

Юрий Бекаревич, Нина Пушкина

MS Office Access 2016

Проектирование и создание
реляционных баз данных

Запросы к базе данных.
Конструирование. Запросы на SQL

Формы для работы с документами
и управления приложением

Отчеты. Печать документов. Вывод
результатов решения задач

Оперативный анализ данных базы
Access в Excel. Табличная модель
данных PowerPivot

Программирование приложений
на языке макросов. Макросы данных

Создание веб-приложения базы
данных на сайтах сервера SharePoint
или в облаке MS Office 365



самоучитель

MS Office Access 2016

Основы
эффективной
работы с базами
данных

Бекаревич Юрий Борисович, доцент экономического факультета Санкт-Петербургского государственного университета, специалист по проектированию и разработке баз данных в сфере экономики.

Пушкина Нина Васильевна, доцент Санкт-Петербургского государственного экономического университета, специалист по базам данных, клиент/серверным технологиям и бизнес-анализу.

Создатели книги имеют большой опыт обучения бакалавров и магистрантов и являются авторами многочисленных трудов в области технологий баз данных и оперативного анализа. В издательстве «БХВ-Петербург» ими подготовлена серия книг по Microsoft Office Access.

Для современных технологий баз данных все более актуальным становится совместное использование удаленных данных. Создание веб-приложения Access позволяет пользователю обратиться к базе данных с любого устройства, имеющего доступ в Интернет.

Авторы знакомят вас с новейшей версией MS Office Access 2016. Дается подробное описание конструирования таблиц и схемы базы данных, запросов для решения задач, а также инструментальных средств разработки объектов приложений — форм и отчетов. Приведенные программы на языке макросов показывают, как автоматизировать работу приложения. Рассмотрена интеграция Access с Excel для оперативного анализа данных в реляционных базах.

ISBN 978-5-9775-3735-3



БХВ-ПЕТЕРБУРГ

191036, Санкт-Петербург,

Гончарная ул., 20

Тел.: (812) 717-10-50,

339-54-17, 339-54-28

E-mail: mail@bhv.ru

Internet: www.bhv.ru



Юрий Бекаревич
Нина Пушкина

самоучитель

MS Office Access 2016

Санкт-Петербург
«БХВ-Петербург»
2017

УДК 004.65
ББК 32.973.26-018.2
Б42

Бекаревич, Ю. Б.

Б42 Самоучитель MS Office Access 2016 / Ю. Б. Бекаревич, Н. В. Пушкина. — СПб.: БХВ-Петербург, 2017. — 480 с.: ил. — (Самоучитель)

ISBN 978-5-9775-3735-3

Самоучитель предназначен для освоения новой версии MS Office Access. Изложены основы разработки базы данных и запросов на SQL, форм, отчетов и макросов, а также другие стандартные технологии реляционных баз данных. Большое внимание уделено освоению интерфейса и инструментальных средств Access 2016. Рассмотрены новые средства разработки веб-приложений, доступных пользователям из браузера с любого устройства, и возможности их автоматической публикации на сайтах сервера SharePoint или в облаке MS Office 365. Приводятся решения задач оперативного анализа на основе данных базы Access средствами Excel и ее надстройки PowerPivot, непосредственно отображающей модель реляционной базы данных. Материал книги иллюстрируется многочисленными примерами.

Для широкого круга пользователей и программистов

УДК 004.65
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Григорий Добин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

Подписано в печать 30.09.16.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 38,7.
Тираж 1500 экз. Заказ № 1578.
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
198034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3735-3

© Бекаревич Ю. Б., Пушкина Н. В., 2017
© Оформление, издательство "БХВ-Петербург", 2017

Оглавление

Предисловие	9
Глава 1. Введение в Access 2016	13
СУБД Microsoft Access и ее реляционная база данных	14
Система управления базами данных	14
Требования к системе для Access 2016	18
Реляционная база данных	19
Определения и понятия	19
Схема данных	21
Объекты Access	22
Размещение базы данных	24
Шаблоны баз данных	25
Мастера Access	28
Средства конструирования объектов	29
Средства программирования	30
Интеграция и использование внешних данных	32
Многопользовательская база данных Access	33
Работа Access с базой данных SQL Server	35
Интернет-технологии	36
Начало работы в Microsoft Access 2016	38
Запуск Access	38
Интерфейс пользователя Access	41
Страницы	42
Лента	43
Коллекция	43
Диалоговые окна	44
Контекстное меню	45
Пользовательская панель инструментов быстрого доступа	45
Область навигации	47
Вкладки документов	50
Строка состояния	50
Панель сообщений	51
Технология Drag and Drop	51
Справка Access	51
Защита баз данных	53
Контрольные вопросы	57
Ответы	58

Глава 2. Проектирование реляционной базы данных.....	61
Этапы проектирования и создания базы данных	61
Построение информационно-логической модели данных	63
Информационные объекты	63
Функциональные зависимости реквизитов	64
Требования нормализации.....	65
Выделение информационных объектов предметной области.....	66
Информационный анализ и определение логической структуры информации	67
Выделение информационных объектов на примере предметной области «Поставка товаров».....	70
Связи информационных объектов.....	80
Тип связи информационных объектов.....	80
Определение связей между информационными объектами.....	82
Информационно-логическая модель предметной области.....	83
Логическая структура реляционной базы данных.....	84
Контрольные вопросы	86
Ответы.....	87
Глава 3. Создание базы данных.....	89
Создание файла базы данных Access	91
Окно Access	93
Лента.....	95
Панель быстрого доступа.....	95
Область навигации.....	96
Область документов	98
Строка состояния.....	100
Создание таблицы базы данных	100
Имя поля и тип данных	101
Общие свойства поля.....	103
Свойства вкладки <i>Подстановка</i>	105
Определение первичного ключа.....	105
Создание таблицы в режиме таблицы.....	107
Добавление полей.....	109
Определение структуры таблицы в режиме конструктора.....	112
Создание таблиц базы данных «Поставка товаров»	113
Создание структуры таблицы	114
Определение составного первичного ключа	120
Использование данных типа <i>Поле объекта OLE</i>	121
Использование данных типа <i>Вложение</i>	121
Использование данных типа <i>Длинный текст</i>	122
Использование данных типа <i>Гиперссылка</i>	123
Использование данных типа <i>Дата и время</i>	125
Маска ввода для даты и текста	125
Непосредственный ввод данных в таблицы.....	126
Макет таблицы	127
Ввод данных в таблицы базы данных	128
Режим ввода записи	130
Вывод строки итогов в таблице	130
Размещение объекта OLE	131
Размещение вложений	132
Ввод логически связанных записей	134
Отображение записей подчиненных таблиц в главной таблице	135

Использование поля со списком при вводе записей	140
Создание поля со списком в режиме таблицы	140
Создание поля со списком в режиме конструктора	143
Схема данных в Access	146
Создание схемы данных	147
Включение таблиц в схему данных	147
Создание связей между таблицами схемы данных	148
Обеспечение целостности данных	150
Каскадное обновление и удаление связанных записей	151
Контрольные вопросы	153
Ответы	154
Глава 4. Запросы.....	157
Однотабличные запросы на выборку	159
Конструирование запросов на выборку с условиями отбора	159
Вычисляемые поля в запросах	164
Параметры в запросах	168
Групповые операции в запросах	170
Запрос с функцией <i>Sum</i>	171
Запрос с функцией <i>Count</i>	173
Условия отбора в запросах с групповыми операциями	174
Отображение строки итогов по столбцу	175
Просмотр инструкции SQL в запросе	175
Контрольные вопросы к разд. «Однотабличные запросы на выборку»	177
Ответы	178
Многотабличные запросы на выборку данных	179
Объединение записей в многотабличном запросе	180
Параметры объединения в инструкциях SQL	184
Ссылки на имена полей различных таблиц в условии отбора	186
Представление данных нарастающим итогом	188
Решение задачи, требующей выполнения нескольких запросов	191
Контрольные вопросы к разд. «Многотабличные запросы на выборку данных»	197
Ответы	199
Запросы на изменение	199
Запрос на создание таблицы	201
Запрос на обновление	202
Обновление полей значениями, рассчитанными в запросе с группировкой	204
Запрос на добавление	207
Добавление данных в связанные таблицы	207
Добавление данных из нескольких таблиц	210
Конструирование запроса на удаление	211
Контрольные вопросы к разд. «Запросы на изменение»	213
Ответы	214
Глава 5. Формы.....	217
Однотабличные формы	219
Создание однотабличной формы	220
Редактирование формы в режиме макета	221
Макеты элементов управления	223
Условное форматирование элементов управления	224
Свойства формы	225
Добавление полей в форму	227

Работа с данными таблицы в режиме формы.....	227
Создание формы на основе запроса	228
Создание разделенной формы	230
Вычисления в форме	232
Вычисления в каждой записи формы	232
Вычисление итоговых значений	234
Многотабличные формы	234
Создание многотабличной формы с помощью мастера	237
Создание одиночной многотабличной формы	242
Создание и редактирование формы в режиме конструктора	245
Создание новой формы конструктором	245
Добавление подчиненной формы	246
Вычисление итогового значения в подчиненной форме и вывод его в текущей записи главной формы	251
Ограничение доступа к данным через форму.....	253
Защита данных поля от изменений.....	253
Установка ограничений на корректировку записей через форму	253
Защита данных подчиненной формы от изменений.....	254
Контрольные вопросы	254
Ответы.....	256
Глава 6. Разработка интерфейса для ввода, просмотра и корректировки документов	259
Этапы разработки интерфейса	260
Определение последовательности загрузки таблиц с документов.....	261
Справочная информация	262
Плановая информация	262
Оперативно-учетная информация	263
Проектирование интерфейса для ввода и корректировки документа.....	264
Определение подсхемы данных.....	264
Разработка макета	265
Создание интерфейса для ввода и корректировки документа.....	267
Доработка интерфейса.....	270
Редактирование формы средствами конструктора	271
Создание кнопок	274
Ограничение доступа к данным таблиц	275
Создание полей со списком	275
Создание поля со списком мастером.....	276
Создание поля со списком в режиме конструктора	280
Вычисления в документе.....	282
Работа с документами.....	283
Выборка документа по его идентификатору	285
Выборка документа по диапазону дат	287
Выборка документов с помощью фильтрации	289
Контрольные вопросы	291
Ответы.....	293
Глава 7. Отчеты.....	295
Основы конструирования отчетов	296
Однотабличные отчеты	298
Доработка отчета в режиме макета	300
Группировка и сортировка данных отчета.....	301

Просмотр и печать отчета.....	305
Представление отчета.....	305
Предварительный просмотр.....	307
Печать отчета.....	309
Изменение источника записей отчета.....	310
Многотабличные отчеты.....	312
Разработка отчета с помощью мастера.....	313
Выбор таблиц для отчета.....	314
Источник записей отчета.....	319
Доработка отчета в режиме конструктора.....	320
Вычисляемые поля в отчете.....	321
Определение параметров в отчете.....	323
Анализ данных отчета средствами фильтрации.....	324
Составные отчеты.....	325
Создание главного отчета.....	327
Создание подчиненного отчета.....	328
Сортировка и группировка записей отчета.....	329
Включение подчиненного отчета.....	330
Доработка составного отчета.....	331
Добавление текущей даты и номера страницы.....	333
Просмотр отчета.....	334
Вывод значений нарастающим итогом.....	335
Контрольные вопросы.....	336
Ответы.....	337
Глава 8. Оперативный анализ данных реляционной базы Access.....	341
Оперативный анализ данных базы Access в Excel.....	344
Отчеты сводных таблиц и диаграмм Excel.....	344
Определение базы данных Access в качестве внешнего источника данных для отчета сводной таблицы.....	348
Отчет сводной таблицы и диаграммы на основе запросов базы данных.....	350
Создание отчета сводной таблицы для анализа отгрузок товара.....	350
Создание отчета сводной таблицы для оценки долга предприятия по договорам.....	355
Отчет сводной таблицы и диаграммы на основе нескольких таблиц базы данных.....	358
Работа с моделью в настройке PowerPivot.....	364
Создание модели и отчета сводной таблицы в PowerPivot.....	367
Контрольные вопросы.....	372
Ответы.....	373
Глава 9. Разработка приложений с использованием макросов.....	375
Конструирование макроса.....	377
Формирование макрокоманд в окне макроса.....	378
Формирование макроса с помощью мыши.....	380
Использование в макросах ссылок на объекты.....	380
Ссылки на объекты и их элементы управления.....	381
Ссылки на свойство объекта.....	381
Ссылки на свойство элемента управления.....	382
Ссылка на подчиненную форму или отчет.....	383
Создание ссылок построителем выражений.....	383
Вложенные макросы.....	384
Создание изолированного макроса.....	385
Внедренный макрос.....	388

Управление последовательностью выполнения макрокоманд	390
Примеры условных выражений	394
Организация выполнения макросов	395
Запуск макроса	395
Выполнение макроса с наступлением события	396
Порядок выполнения макросов, вызываемых событиями	399
Вызов макроса из другого макроса	400
Создание кнопки запуска макроса в форме	400
Создание кнопки запуска макроса с помощью мыши	401
Создание кнопки запуска макроса мастером	401
Макросы данных	402
Именованные макросы	403
Использование макросов данных для решения задач	405
Регистрация событий	412
Контрольные вопросы	413
Ответы	414
Глава 10. Разработка интерфейса для управления приложением	417
Диалоговое окно входа в приложение	418
Главная форма управления приложением	423
Использование форм с вкладками при разработке интерфейса	429
Панель навигации формы с вкладками	434
Формы навигации	435
Контрольные вопросы	437
Ответы	437
Глава 11. Создание веб-приложений	439
Выбор расположения для веб-приложения	440
Добавление таблиц в базу данных веб-приложения	443
Открытие веб-приложения	446
Связывание таблиц	449
Вычисление итоговых значений	452
Создание в форме ссылок на связанную запись главной таблицы	453
Создание поля со списком	455
Создание и использование запросов	457
Создание макросов	461
Создание отчетов	463
Контрольные вопросы	463
Ответы	464
Приложение 1. Структура таблиц базы данных «Поставка товаров»	465
Таблицы справочных данных	465
Таблицы плановых данных	468
Таблицы оперативно-учетных данных	469
Приложение 2. Пример заполненных документов для загрузки в базу данных	
«Поставка товаров»	471
Справочная информация	471
Плановая информация	472
Оперативно-учетная информация	473
Предметный указатель	475

Предисловие

Книга предназначена для изучения технологий, реализуемых новой версией системы управления реляционными базами данных (СУБД) — Microsoft Office Access 2016.

Access 2016, сохраняя функционал разработки локальных баз данных, еще более, чем его предшествующие версии, ориентирован на создание веб-приложений, публикуемых на серверах Microsoft SharePoint, — как корпоративных, так и «облачных» в Office 365. С созданными веб-приложениями пользователи работают уже через браузер, не устанавливая Access на компьютере.

Access позволяет быстро создать веб-приложение для отслеживания важных бизнес-данных. Такое приложение может совместно использоваться сотрудниками компании, работающими в разных городах и странах, на компьютерах различных платформ. Получить веб-приложение на страницах SharePoint можно автоматически — на основе модели данных путем импорта локальной базы. При этом Access 2016 создаст структуру базы данных с формами-представлениями, позволяющими просматривать, добавлять и редактировать данные. Средства навигации и основные команды, встроенные в веб-приложение, позволят сразу начать с ним работать. Завершив создание веб-приложения, автору не потребуется его специально публиковать, — оно публикуется на сайте SharePoint автоматически и готово к работе. Для хранения данных в веб-приложениях Access служат Microsoft SQL Server или SQL Azure. Использование среды SQL Server для хранения данных обеспечивает новый уровень управляемости и масштабируемости.

Для расширения возможностей по обработке данных веб-приложения в Access предусмотрена функция создания локального приложения, подключающегося к базе данных веб-приложения через интерфейс ODBC (Open Database Connectivity). При этом в локальном приложении таблицы базы данных веб-приложения отображаются как связанные и могут быть использованы как обычные таблицы локальной базы данных при решении любых задач предметной области и построении нужных отчетов. Таким образом обеспечивается не только совместная работа удаленных пользователей с базой данных, но и полный цикл обработки данных, размещенных на сайте SharePoint.

В книге продемонстрированы возможности оперативного анализа данных базы Access 2016 в отчетах сводных таблиц и диаграмм расширенными средствами

Excel 2016 и его надстройки PowerPivot. Дело в том, что оперативный анализ данных реляционной базы в версиях, предшествующих Access 2013, обеспечивали содержащиеся в них средства построения сводных таблиц и диаграмм. С их помощью можно было отображать и анализировать данные любой таблицы или запроса. Из версий Access 2013/2016 эти средства исключены. Новые возможности средств анализа данных в Excel теперь позволяют разгрузить оперативные системы и перенести аналитическую платформу, включая источник данных, в отдельную специализированную среду. Отчеты сводных таблиц и сводных диаграмм Excel дают возможность в интерактивном режиме выполнить подробный анализ больших объемов данных базы Access. Эти средства реализуют методику оперативного извлечения нужной информации и формирования соответствующих отчетов, позволяя конечному пользователю оперативно получать ответы на любые аналитические запросы.

В Microsoft Office Excel 2013/2016 и Microsoft Office Excel 365 в качестве источника данных сводных таблиц и диаграмм можно использовать вместо одной таблицы набор взаимосвязанных таблиц, образующих модель данных. Подключение и импорт в Excel базы данных Access 2016 обеспечивает оперативный анализ их в отчетах сводных таблиц и диаграмм Excel с использованием табличной модели данных реляционной базы. При импорте таблиц базы данных Access в отчет сводной таблицы Excel обеспечивается автоматическое создание такой модели. Включение в состав Excel надстройки PowerPivot значительно расширяет возможности по созданию табличных моделей и анализу данных. PowerPivot использует ядро СУБД для обработки огромных массивов данных (сводных таблиц), размещает и обрабатывает их в оперативной памяти, эффективно сжимает при сохранении.

В Access 2016, как и в его предыдущих версиях, можно создавать на компьютере как простые, так и очень сложные базы данных. Конечные пользователи, не прибегая к помощи программистов и администраторов, могут в кратчайшие сроки начать выполнение повседневных задач, работая с документами и реализуя деловые операции бизнес-процессов. В то же время, используя функциональные возможности Access 2016, специалисты отделов информационных технологий смогут обеспечить разработку и поддержку приложений Access, создаваемых на уровне компаний.

Многие пользователи в значительной степени знакомы с возможностями популярного приложения Microsoft Office Excel. Однако если объемы данных значительны, то от организации пользовательских списков в Excel целесообразно переходить к применению специализированных средств создания и использования баз данных Access. Реляционная СУБД, позволяя распределять данные по нескольким взаимосвязанным таблицам, обеспечивает эффективную обработку данных и их актуализацию при минимальном дублировании данных. Обработка и корректировка данных выполняется при использовании форм и отчетов, обеспечивающих удобное для конечного пользователя представление и доступ к данным. В основе таких средств лежит реляционный язык структурированных запросов SQL, который обеспечивает реализацию практически любых запросов к базе данных.

Новая версия Access делает еще доступнее и удобнее использование средств СУБД для начинающих пользователей. Современный интуитивно понятный интерфейс, мно-

гочисленные готовые решения, простые средства создания таблиц базы данных, форм и отчетов в полной мере реализуют эту цель. Работа с усовершенствованным интерфейсом, обеспечивающим быстрый поиск нужных инструментов, позволяет сразу включиться в подготовку профессиональных документов, повышает эффективность, качество работы и сокращает сроки ее выполнения. Необходимые команды организованы в виде наборов вкладок, четко соответствующих функциям, реализуемым в Access, благодаря чему находить их легко и просто.

Предлагаемые шаблоны локальных и веб-приложений помогают пользователю выбрать приложение, в значительной мере соответствующее их потребностям, быстро познакомиться с разнообразными функциями и начать конкретную работу в среде Access 2016. Можно использовать эти стандартные приложения без каких-либо модификаций или настройки, а также адаптировать их в соответствии с задачами пользователя.

Access 2016 предоставляет гибкие средства для работы пользователя с разным уровнем подготовки. Даже новичок может начать создавать таблицы с любым числом полей непосредственным вводом данных без предварительного определения структуры таблиц — тип введенных данных определится автоматически. Тут же, прямо в режиме таблицы, можно вмешаться в этот процесс: присвоить имя, уточнить тип данных, его формат, создать столбцы подстановки.

Кроме создания таблиц в базе данных, Access позволяет организовать связи с внешними данными с помощью интерфейса ODBC. Связанные таблицы отображаются и функционируют аналогично собственным таблицам. Создание связанных таблиц обеспечивает легкость, с которой пользователи Access могут перемещать, объединять и обновлять данные из нескольких разных источников. Через связанные таблицы возможно организовать взаимодействие со всеми распространенными серверами реляционных баз данных, включая Microsoft SQL Server, Oracle, IBM DB2, Informix и Sybase. Возможность создания клиентских приложений — проектов, обеспечивающих доступ к серверам баз данных через интерфейс OLE DB, начиная с Access 2013 исключена.

Интерактивные средства Access обеспечивают простоту разработки как форм для работы с документами предметной области, так и отчетов с результатами решения задач. Построить форму или отчет можно одним щелчком кнопки мыши или с помощью мастера, а для их последующей доработки воспользоваться средствами конструктора. Привлекая мастеров и графические инструменты Access, пользователи, не владеющие специальными навыками, могут весьма успешно разрабатывать приложения баз данных. Представление форм и отчетов в режиме макета позволяет легко настроить их в соответствии с потребностями пользователя, непосредственно наблюдающего отображение данных. Простые в использовании и разнообразные средства фильтрации данных дают возможность, не перестраивая макет, получить отчеты, представляющие данные в самых разных разрезах, и использовать их для анализа данных.

В Access 2016, как и в его предшествующих версиях, предусмотрены средства обеспечения безопасности. Защита базы данных, созданной в файлах форматов

ACCDB или ACCDE, построена на доверии к происхождению ее выполняемых объектов. Некоторые объекты Access — такие как запросы на изменение, макросы, изменяющие базу данных, код VBA и элементы управления ActiveX, полученные из ненадежных источников, выполняясь, могут повредить базу данных. Поэтому база данных открывается для работы с полным набором функциональных возможностей только в том случае, если ее компоненты подписаны надежными издателями, или пользователь придал ей состояние доверенной и берет на себя ответственность за безопасность. Средства обеспечения безопасности баз данных не предусматривают защиты на уровне пользователей.

Книга ориентирована на получение целостного представления о новой версии Access 2016, изучение во взаимосвязи всех его компонентов — от традиционных инструментальных средств до перспективных технологий. Особенностью книги этой серии является демонстрация средств на конкретных примерах. В ней рассматривается проектирование, создание и работа с данными базы, содержащей сведения о поставках товаров гипотетического предприятия. Представлены примеры работы с данными справочных, плановых и оперативно-учетных документов и решения экономических задач. Такой выбор продиктован тем, что одним из характерных применений СУБД является хранение и обработка экономической информации.

Приведенная в книге система взаимосвязанных примеров позволяет наглядно показать возможности средств конструирования объектов и программирования при разработке приложений и организации диалога при решении задач. Показана возможность использовать среду Access 2016 при создании профессиональных приложений, обеспечивающих подготовку системы взаимосвязанных документов, и возможность оперативно выполнять анализ данных.

Выполнение на компьютере в процессе изучения Access 2016 приведенных в книге примеров позволит успешно освоить его инструментарий, приобрести навыки работы с реляционной базой данных и использовать их в любой предметной области. Все примеры содержат подробное описание действий пользователя в процессе выполнения конкретных заданий.

Книга рассчитана на широкий круг пользователей: от студента, изучающего основы технологии баз данных, до разработчиков практических приложений в среде СУБД. Самоучитель сопровождает читателя-пользователя в процессе создания базы данных, ее таблиц и запросов и конструирования объектов приложения: форм, отчетов и макросов, а также интерфейса для управления приложением. Приведенные в конце каждой главы контрольные вопросы с ответами дают возможность проверить степень освоения изученного материала.

Книга позволяет самостоятельно освоить технологии работы с базами данных и может быть использована в качестве учебника по дисциплинам, ориентированным на изучение СУБД.

ГЛАВА 1



Введение в Access 2016

подавляющее большинство деловых операций сегодня записывается, отслеживается и анализируется в виде данных, хранящихся в реляционных системах управления базами данных.

Система управления реляционными базами данных Microsoft Office Access удовлетворяет потребностям самых разных групп пользователей. С помощью мастеров и графических инструментов Access даже пользователи, не владеющие специальными навыками, могут весьма успешно разрабатывать полезные приложения баз данных. Исследования малых и средних предприятий, проведенные различными службами, показали, что Access является одной из самых популярных программ для работы с базами данных.

Приложения для автоматизации работы с электронными таблицами, такие как Microsoft Office Excel, используются на персональных компьютерах с момента их появления и, реализуя мощные вычислительные функции, средства анализа данных и построения диаграмм, позволяют выполнять многие стандартные задачи по обработке табличных данных. Современные приложения для совместной работы, такие как Windows SharePoint, также поддерживают создание и обслуживание списков, доступных через интерфейс веб-обозревателя. Но следует заметить, что ни один из продуктов, поддерживающих таблицы, не обладает всеми достоинствами настоящей реляционной базы данных.

По мере усложнения приложений на основе электронных таблиц или списков возникает вопрос о перемещении данных в стандартные реляционные таблицы, из которых можно выбирать и обрабатывать данные с помощью языка SQL (Structured Query Language). После превращения списков в реляционные данные Access позволяет быстро создавать приложения для решения самых разнообразных задач.

Для решения стандартных задач в Access доступны многочисленные шаблоны баз данных. Мастера и удобные средства конструирования обеспечивают простоту создания в Access реляционных структур данных, а также запросов, форм и отчетов, необходимых для работы с данными. Контекстно-зависимый интерфейс всегда предоставит пользователю элементы, необходимые в данный момент времени.

Простой интуитивно понятный интерфейс пользователя и интерактивные средства разработки в составе Access 2016 делают создание баз данных и приложений в среде этого программного продукта доступным для начинающих пользователей. Любой сотрудник, не имея опыта программирования и обладая ограниченными знаниями в области баз данных, может, используя Access, самостоятельно решать задачи по обработке данных.

В то же время Access удовлетворяет требованиям профессиональных разработчиков и позволяет за незначительное время разрабатывать сложные бизнес-системы.

Средства Access ориентированы на упрощение разработки веб-приложений баз данных, которые позволят сотрудниками компании совместно отслеживать важные бизнес-данные. Благодаря тесной интеграции с Microsoft SharePoint пользовательские веб-приложения просто использовать как в масштабах корпоративной сети предприятия, так и Интернета в целом. Размещение веб-приложения на сайтах SharePoint обеспечивает общий доступ к данным базы через веб-обозреватель и сохраняет возможность доработки веб-приложения в Access.

Для создания веб-приложения и работы с ним помимо Access понадобится SharePoint Online, включенный в Office 365, или корпоративный сервер SharePoint Server с поддержкой служб Access. Данные базы веб-приложения хранятся в базе данных SQL Windows Azure или на сервере SQL Server. Служба SharePoint в пакете Office 365 достаточно проста в управлении и предоставляет усовершенствованные технологии обеспечения безопасности, а также возможность доступа практически из любого расположения.

Интеграция с Office Outlook позволяет использовать электронную почту для сбора или обновления данных в таблицах Access, не прибегая к вводу данных вручную.

СУБД Microsoft Access и ее реляционная база данных

Система управления базами данных

СУБД (система управления базами данных) представляет собой универсальный программный инструмент создания и обслуживания баз данных и приложений пользователя в самых разных предметных областях. СУБД обеспечивает создание, многоаспектный доступ к данным и использование одних и тех же данных различными задачами и приложениями пользователей.

СУБД поддерживаются различные модели данных. *Модель данных* — это метод (принцип) логической организации данных, используемый СУБД. Наиболее известными являются иерархическая, сетевая и реляционная модели.

В СУБД для персональных компьютеров (настольных СУБД) поддерживается преимущественно *реляционная модель*, которую отличает простота и единообразие представления данных простейшими *двумерными таблицами*. Реляционная модель обеспечивает возможность использования в разных СУБД операций обработки

данных, имеющих единую основу, — *алгебру отношений* (реляционную алгебру), и универсального языка структурированных запросов — SQL (Structured Query Language).

Основной логической структурной единицей манипулирования данными является *строка* таблицы — *запись*. *Структура записи* определяется составом входящих в нее *полей*. Совокупность полей записи соответствует логически связанным реквизитам, характеризующим некоторую сущность предметной области.

Типовыми функциями СУБД по манипулированию данными являются выборка, добавление, удаление, изменение данных.

- *Выборка данных* — выборка записей из одной или нескольких взаимосвязанных таблиц в соответствии с заданными условиями.
- *Добавление и удаление данных* — добавление новых записей в таблицы и удаление существующих.
- *Изменение данных* — изменение значений данных в полях существующих записей.

Данные из одной или нескольких взаимосвязанных таблиц могут подвергаться обработке. К операциям *обработки* относятся, например, расчеты в пределах каждой записи, группировка записей в соответствии с заданным критерием группировки и обработка записей выделенных групп с помощью статистических функций, таких как суммирование, определение максимального, подсчет числа записей в группе и т. п.

СУБД *Microsoft Access* является системой управления реляционной базой данных, включающей все необходимые инструментальные средства для создания локальной базы данных, общей базы данных в локальной сети или в Интернете на сайте SharePoint, сохраняемой на SQL-сервере, а также для создания приложения пользователя, работающего с этими базами данных.

База данных Access, создаваемая на локальном компьютере, хранит в файле не только все таблицы с данными, но и объекты приложения: формы, отчеты, а также программный код. Благодаря этому можно создать приложение, целиком хранящееся в одном ACCDB-файле, что существенно упрощает как создание, так и распространение приложений баз данных.

СУБД Access включает разнообразные и многочисленные относительно автономные инструментальные средства, ориентированные на создание объектов базы данных и приложений пользователя.

- *Разнообразные мастера* в режиме ведения диалога с пользователем позволяют создавать объекты и выполнять функции по реорганизации и преобразованию баз данных.
- *Средства программирования* включают язык структурированных запросов SQL, язык макрокоманд и язык объектно-ориентированного программирования для приложений Microsoft Visual Basic for Applications (VBA). VBA является частью семейства Microsoft Visual Basic, которое входит в состав Visual Studio.

□ Средства графического конструирования позволяют создавать объекты базы данных и объекты приложения с помощью многочисленных графических элементов, не прибегая к программированию.

Среди многочисленных средств графического конструирования и диалоговых средств Access следует выделить средства для создания:

- таблиц и схем баз данных, отображающих их связи;
- запросов выборки, отбирающих и объединяющих данные нескольких таблиц в виртуальную таблицу, которая может использоваться во многих задачах приложения;
- запросов на изменение данных базы;
- экранных форм, предназначенных для ввода, просмотра и обработки данных в диалоговом режиме;
- отчетов, предназначенных для просмотра и вывода на печать результатов обработки данных из базы в удобном для пользователя виде.

VBA является базовым компонентом Microsoft Office — он интегрирован в Access, Excel, Outlook, PowerPoint и Word. Все эти приложения, в том числе и локализованные на русском языке, используют англоязычный вариант VBA (включая справочную систему). VBA входит во все варианты поставок Microsoft Office.

VBA представляет собой базовую платформу программирования не только в среде Microsoft Office, но и многих других приложений. VBA содержит средства доступа не только к базам данных Access, но и к базам данных клиент-серверной архитектуры, таким как Microsoft SQL Server, Oracle и др.

Система доступа к данным построена на основе ядра базы данных Access Database Engine. Ядро базы данных используется для создания реляционных баз данных: таблиц для хранения данных и объектов баз данных, таких как индексы; выполняет загрузку, сохранение и извлечение данных в пользовательских и системных базах данных; обеспечивает высокую производительность и улучшенные сетевые характеристики, поддержку двухбайтового представления символов — Unicode, позволяющего использовать символы нескольких национальных алфавитов.

В Access активно развиваются технологические направления, составляющие основу совместного использования корпоративных баз данных.

В Access 2016 обеспечивается работа с многопользовательскими базами данных SQL серверов, реализующими архитектуру «клиент — сервер». Широко известными СУБД, используемыми в этой архитектуре, являются Microsoft SQL Server, Oracle, Sybase SQL Server и др. Эти СУБД являются реляционными SQL-серверами баз данных. Доступ и изменение данных в серверной базе производится по запросам пользователей, обработка которых осуществляется на сервере. Пользователю-клиенту, сделавшему запрос, возвращается только результат выполнения запроса. Access 2016 при работе в качестве клиента для взаимосвязи с SQL-сервером использует специальное программное обеспечение — широко распространенный ин-

терфейс ODBC. Access предоставляет несколько способов взаимодействия приложения с данными сервера на основе интерфейса ODBC:

- *запросы на языке SQL к базам данных SQL Server* — с помощью запросов на SQL можно выполнить выборку данных из таблиц, сохраняемых на сервере, добавление, обновление данных и структур таблиц сервера. Запросы к серверу могут содержать инструкции для создания и удаления таблиц в базе данных и даже баз данных на сервере. Запросы передаются на сервер и выполняются непосредственно на нем. Запросы целесообразно использовать для запуска сохраняемых процедур на сервере;
- *связь с таблицами баз данных SQL Server* — при этом таблицы сервера отображаются в окне базы данных Access для их просмотра и изменения средствами Access. Данные таблиц остаются в формате источника, но представлены как объекты Access. Внешние данные этих объектов можно использовать при создании запросов, форм, отчетов, причем можно объединить внешние данные с данными из объектов Access;
- *импорт объектов SQL Server в текущую базу данных Access* — при импорте объектов создаются их копии. Исходные объекты источника при этом не изменяются;
- *экспорт таблиц базы данных Access в базу данных на SQL Server* — при взаимодействии Access с сервером локальные таблицы базы данных могут быть экспортированы в общую базу данных сервера. Таким образом, можно, создав таблицу в Access, поместить ее в базу данных сервера.

ЗАМЕЧАНИЕ

Начиная с Access 2013, исключена реализованная в предыдущих версиях возможность создания клиентских приложений Microsoft SQL Server — *проектов*, в которых можно было создавать новые базы данных на сервере или использовать существующие и взаимодействовать с ними на основе интерфейса OLE DB.

Интернет-технологии позволяют эффективно распространять и получать доступ к разнородной информации в глобальных и корпоративных сетях. Эти технологии обеспечивают унифицированный доступ к данным различных приложений в разнородных сетях. В роли универсального интерфейса для доступа и работы с данными из внешней среды вне зависимости от аппаратно-программной платформы компьютера пользователя и компьютера-источника данных здесь выступает веб-браузер. Access предоставляет платформу для создания веб-приложений, которые можно совместно использовать в пределах организации или в Интернете с помощью браузера. С веб-страницами приложения можно работать как с формами Access — для ввода и редактирования данных, как с отчетами Access — для отображения иерархически сгруппированных данных. Для работы с приложением на компьютере пользователя не требуется устанавливать Access. Веб-приложения разрабатываются и могут публиковаться как на сайтах собственного сервера SharePoint в интрасети, так и на сайтах SharePoint, предоставляемых Office 365 в «облаке». База данных веб-приложения сохраняется на SQL Server, определенном в SharePoint. Безопас-

ность доступа к данным и управление ими обеспечивается средствами SharePoint и SQL Server.

Требования к системе для Access 2016

Access 2016 входит в состав выпуска Office Professional 2016. Кроме того, можно оформить подписку на «облачные» версии Office 365 — Access входит в состав устанавливаемых приложений Office в следующих выпусках: Office 365 Personal (для 1 пользователя), Office 365 Home (до 5 пользователей ПК или планшетов), Office 365 ProPlus, Office 365 Enterprise E3, Office 365 Enterprise E5.

Для работы Access 2016 рекомендуется использовать компьютер с 32- или 64-разрядным процессором с тактовой частотой 1 ГГц или выше с набором инструкций SSE2. Памяти (RAM) необходимо не менее 1 Гбайт (для 32-разрядных систем) или 2 Гбайт (для 64-разрядных). Компьютер должен работать под управлением одной из операционных систем: Windows 7 с пакетом обновления 1 или более поздней версии, Windows Server 2008 R2 или Windows Server 2012. При установке Access 2016 на планшет нужна Windows 8 или выше.

Для создания и использования веб-приложений помимо Access 2016 понадобится доступ к сайтам корпоративного сервера SharePoint Server 2013/2016 или сайтам SharePoint Online в Office 365 с поддержкой служб Access. Данные базы веб-приложения хранятся на сервере SQL Server 2012 или в базе данных SQL Windows Azure.

Более насыщенные функциональными возможностями новые операционные системы семейства Windows в сочетании с пакетом Office 2016 дают пользователю возможность наиболее эффективно применять компьютер, но их установка не является обязательным условием для работы с пакетом Office 2016. При обновлении с Microsoft Office 2013 на версию 2016 не обязательно обновлять оборудование, хотя может потребоваться выполнить обновление до поддерживаемой операционной системы.

При стандартной установке Access 2016 требуется примерно 3 Гбайт свободного дискового пространства. Часть места на диске будет освобождена после установки при удалении с него исходного загрузочного пакета. В процессе установки на диске необходимо иметь дополнительное пространство примерно такого же объема.

Рекомендуется монитор с разрешением 1280×800 пикселей или с более высоким и с поддержкой 256 цветов.

Для поддержки веб-приложений Access 2016 в среде Office 365 нужна последняя версия браузера Microsoft Edge, так как система Office 365 ориентирована прежде всего на работу с этим браузером, призванным заменить Internet Explorer. Office 365 может работать с предыдущими версиями Edge, однако нет никаких гарантий от возникновения проблем. Пользователям Windows 7 и Windows 8.1 Microsoft Edge не доступен. Он вошел в состав Windows 10 параллельно с Internet Explorer 11.

Office 365 работает с текущей или предыдущей версией Internet Explorer. Рекомендуется использовать последнюю версию Internet Explorer, когда она появляется.

При работе с предыдущими версиями Internet Explorer после выпуска новой никаких гарантий от возникновения проблем не предоставляется.

С января 2016 года начали внедряться изменения для Office 365, которые приводят к существенному ухудшению качества работы для пользователей в Internet Explorer 8 и Internet Explorer 9. Эти нововведения совпадают по времени с изменениями в поддержке браузера Internet Explorer в Windows.

Office 365 также работает с текущими версиями браузеров Safari, Chrome и Firefox.

Реляционная база данных

Определения и понятия

База данных является организованной на машинном носителе совокупностью взаимосвязанных данных и содержит сведения о различных сущностях одной предметной области: реальных объектах, процессах, событиях или явлениях.

Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц — *реляционных таблиц*, называемых также *отношениями*, в каждой из которых содержатся сведения об одной сущности автоматизируемой предметной области.

Логическую структуру реляционной базы данных образует совокупность реляционных таблиц, между которыми установлены логические связи.

В таблицах базы должны сохраняться все данные, необходимые для решения задач предметной области. Причем каждый элемент данных должен храниться в базе только в одном экземпляре. Для создания таблиц, соответствующих реляционной модели данных, используется процесс, называемый *нормализацией данных*. *Нормализация* — это удаление из таблиц повторяющихся данных путем их переноса в новые таблицы, записи которых не содержат повторяющихся значений.

Минимальное дублирование данных в реляционной базе обеспечивает высокую эффективность поддержания базы данных в актуальном и непротиворечивом состоянии, однократный ввод и корректировку данных.

Структура реляционной таблицы определяется составом полей. Каждое *поле* отражает определенную характеристику сущности. Для поля указывается тип и размер элементарного данного, размещаемого в нем, и ряд других свойств. Содержимое поля отображается в столбце таблицы. Столбец таблицы содержит данные одного типа.

Содержание таблицы заключено в ее строках, однотипных по структуре. Каждая строка таблицы содержит данные о конкретном экземпляре сущности и называется *записью*.

Для однозначного определения (*идентификации*) каждой записи таблица должна иметь *уникальный (первичный) ключ*. По значению ключа таблицы отыскивается единственная запись в таблице. Ключ может состоять из одного или нескольких полей таблицы. Значение уникального ключа не может повторяться в нескольких записях.

Логические связи между таблицами дают возможность объединять данные из разных таблиц. Связь каждой пары таблиц задается одинаковыми полями в них — *ключом связи*. Таким образом обеспечивается рациональное хранение недублированных данных и их объединение в соответствии с требованиями решаемых задач.

В нормализованной реляционной базе данных связь двух таблиц характеризуется отношениями записей типа «один-к-одному» (1 : 1) или «один-ко-многим» (1 : M). Отношение 1 : 1 предполагает, что каждой записи одной таблицы соответствует одна запись в другой. Отношение 1 : M предполагает, что каждой записи первой таблицы соответствует много записей во второй, но каждой записи второй таблицы соответствует только одна запись в первой.

Для двух таблиц, находящихся в отношении типа 1 : M, связь устанавливается по уникальному ключу таблицы, представляющей в отношении сторону «один», — *главной таблицы* в связи. Во второй таблице, представляющей в отношении сторону «многие» и называемой *подчиненной*, этот ключ связи может быть либо частью уникального ключа, либо не входить в состав ключа. В подчиненной таблице ключ связи называется еще *внешним ключом*.

На рис. 1.1 показаны две таблицы со списком покупателей и перечнем заключенных договоров, которые находятся в отношении типа 1 : M и логически связаны с помощью общего поля (столбца) **Код покупателя** — ключа связи. Это поле является уникальным ключом в главной таблице — **ПОКУПАТЕЛЬ** и неключевым полем в подчиненной таблице — **ДОГОВОР**.

Связь 1 : M по одноименному полю

Уникальный ключ

Поле

Главная таблица

Подчиненная таблица

ПОКУПАТЕЛЬ						
Код покупателя	ИНН	Наименование	Адрес	Телефон	Номер счета	Б.
П001	7789576511	Компьютер маркет	Москва	(812)345-23-4	7635850976326	Мос
П002						
П004						
П005						
П006						
П007						
П008						
П009						
* Запись № 4 из 8						

ДОГОВОР			
Номер договора	Дата заключения	Код покупателя	Сумма по договору
Д111	11.01.2013	П001	6 414 000,00р.
Д222	05.02.2013	П001	152 280,00р.
Д333	01.01.2013	П002	74 920,00р.
Д555	12.11.2013	П002	9 362,00р.
Д777	14.06.2013	П001	1 579 920,00р.
Д888	23.05.2013	П002	30 000,00р.
Д999	12.09.2013	П002	450 000,00р.
* Запись № 4 из 7			

Рис. 1.1. Взаимосвязанные таблицы реляционной базы данных

Размещение сведений о каждой сущности в отдельной таблице и связывание таблиц позволяет избежать повторения описательных данных в разных таблицах. При этом обеспечивается однократный ввод данных при загрузке и корректировке базы

данных. Если данные двух таблиц в приведенном примере разместить в одной таблице, то каждая запись должна соответствовать одному договору. Причем данные о покупателе (наименование, ИНН, адрес и др.) будут повторяться во всех записях о договорах одного покупателя, что усложнит ввод, корректировку и обеспечение актуального состояния базы данных. При хранении данных в двух таблицах сведения о покупателе хранятся в единственном экземпляре, а в записях таблицы договоров повторяются только значения ключевого поля с кодом покупателя.

В Access реализовано средство просмотра и редактирования связанных записей нескольких таблиц. При этом данные отображаются в иерархическом виде. При раскрытии одного уровня иерархии рядом с записью главной таблицы отображаются связанные записи подчиненной. Для записи подчиненной таблицы также могут быть открыты связанные записи и т. д. Например, для таблиц **ПОКУПАТЕЛЬ** и **ДОГОВОР** (рис. 1.2), связанных отношением 1 : М, для каждой записи таблицы **ПОКУПАТЕЛЬ** могут быть отображены и отредактированы связанные записи в таблице **ДОГОВОР**.

ПОКУПАТЕЛЬ																													
Код покупателя	ИНН	Наименование	Адрес	Телефон	Номер счета																								
☑ П001	7789576611	Компьютер маркет	Москва	(812)345-23-4	7635850976326																								
<table border="1"> <thead> <tr> <th>Номер договора</th> <th>Дата заключения</th> <th>Сумма по договору</th> <th>Код исполнителя</th> </tr> </thead> <tbody> <tr> <td>☑ Д111</td> <td>11.01.2016</td> <td>6 414 000.00р</td> <td>6</td> </tr> <tr> <td>☑ Д222</td> <td>05.02.2016</td> <td>152 280.00р</td> <td>1</td> </tr> <tr> <td>☑ Д777</td> <td>14.06.2016</td> <td>1 579 920.00р</td> <td>1</td> </tr> <tr> <td>* * *</td> <td></td> <td>0.00р</td> <td></td> </tr> </tbody> </table>						Номер договора	Дата заключения	Сумма по договору	Код исполнителя	☑ Д111	11.01.2016	6 414 000.00р	6	☑ Д222	05.02.2016	152 280.00р	1	☑ Д777	14.06.2016	1 579 920.00р	1	* * *		0.00р					
Номер договора	Дата заключения	Сумма по договору	Код исполнителя																										
☑ Д111	11.01.2016	6 414 000.00р	6																										
☑ Д222	05.02.2016	152 280.00р	1																										
☑ Д777	14.06.2016	1 579 920.00р	1																										
* * *		0.00р																											
☑ П002	7898897987	Перспектива	Москва	(995)345-67-8	2938478982394																								
<table border="1"> <thead> <tr> <th>Номер договора</th> <th>Дата заключения</th> <th>Сумма по договору</th> <th>Код исполнителя</th> </tr> </thead> <tbody> <tr> <td>☑ Д333</td> <td>01.01.2016</td> <td>74 920.00р</td> <td>3</td> </tr> <tr> <td>☑ Д555</td> <td>12.11.2016</td> <td>9 352.00р</td> <td>3</td> </tr> <tr> <td>☑ Д888</td> <td>23.05.2016</td> <td>30 000.00р</td> <td>6</td> </tr> <tr> <td>☑ Д999</td> <td>12.09.2016</td> <td>450 000.00р</td> <td>7</td> </tr> <tr> <td>* * *</td> <td></td> <td>0.00р</td> <td></td> </tr> </tbody> </table>						Номер договора	Дата заключения	Сумма по договору	Код исполнителя	☑ Д333	01.01.2016	74 920.00р	3	☑ Д555	12.11.2016	9 352.00р	3	☑ Д888	23.05.2016	30 000.00р	6	☑ Д999	12.09.2016	450 000.00р	7	* * *		0.00р	
Номер договора	Дата заключения	Сумма по договору	Код исполнителя																										
☑ Д333	01.01.2016	74 920.00р	3																										
☑ Д555	12.11.2016	9 352.00р	3																										
☑ Д888	23.05.2016	30 000.00р	6																										
☑ Д999	12.09.2016	450 000.00р	7																										
* * *		0.00р																											
☑ П004	4565756756	Монитор	Санкт-Петербург	(123)45-67	5856878976897																								
☑ П005	7712435678	Компьютер лэнд	Саратов	(123)56-	5979690879087																								
☑ П006	5878798791	Компьютерная техн	Ярославль	(123)45-67	7635850976326																								

Рис. 1.2. Отображение в записях главной таблицы связанных записей подчиненной таблицы

Схема данных

В СУБД Access процесс создания реляционной базы данных включает создание *схемы данных*. Схема данных наглядно отображает логическую структуру базы данных: таблицы и связи между ними, а также обеспечивает использование установленных в ней связей при обработке данных.

Для нормализованной базы данных, основанной на одно-многочисленных и одно-однозначных отношениях между таблицами, в схеме данных для связей таких таблиц по первичному ключу или уникальному индексу главной таблицы могут устанавливаться параметры обеспечения *связной целостности*.

При поддержании целостности взаимосвязанных данных не допускается наличия записи в подчиненной таблице, если в главной таблице отсутствует связанная с ней запись. Соответственно, при первоначальной загрузке базы данных, а также корректировке, добавлении и удалении записей, система допускает выполнение операции только в том случае, если она не приводит к нарушению целостности.

Связи, определенные в схеме данных, автоматически используются для объединения таблиц при разработке многотабличных форм, запросов, отчетов, существенно упрощая процесс их конструирования.

В схеме данных связи могут устанавливаться для любой пары таблиц, имеющих одинаковое поле, позволяющее объединять эти таблицы.

Объекты Access

База данных Access включает следующие сохраняемые в одном ACCDB-файле объекты:

- таблицы, запросы, схемы данных*, непосредственно имеющие отношение к базе данных;
- формы, отчеты, макросы и модули*, называемые объектами приложения.

Формы и отчеты предназначены для типовых процессов обработки данных: просмотра, обновления, поиска по заданным критериям, получения отчетов. Эти объекты приложений конструируются из графических элементов, называемых *элементами управления*. Основные элементы управления служат для отображения полей таблиц, являющихся источниками данных объекта.

Для автоматизации доступа к объектам и их взаимодействия используется программный код. Только с помощью программного кода получается полноценное приложение пользователя, функции которого доступны через меню, панели инструментов и формы. Для создания программного кода служат модули на языке VBA и макросы.

Каждый объект приложения и элемент управления имеет свои свойства, определяя которые, можно настраивать их. С каждым объектом и элементом управления связывается набор событий, которые могут обрабатываться макросами или процедурами обработки событий на VBA, входящими в состав модулей форм, отчетов.

Объекты представлены в области навигации окна базы данных Access. Все операции по работе с объектами собственно базы данных и приложений начинаются в этом окне.

- Таблицы (Tables)* создаются пользователем для хранения данных об одной сущности — одном информационном объекте модели данных предметной области. Таблица состоит из полей (столбцов) и записей (строк). Каждое поле содержит одну характеристику информационного объекта предметной области. В записи собраны сведения об одном экземпляре информационного объекта. Каждая таблица для однозначной идентификации записи должна иметь уникальный — первичный ключ.

База данных Access может включать до 32 768 объектов (в том числе формы, отчеты и т. д.). Одновременно может открываться до 2048 таблиц.

- **Запросы (Queries).** Запросы на выборку служат для выборки нужных данных из одной или нескольких связанных таблиц. Результатом выполнения запроса является виртуальная таблица запроса. В запросе можно указать, какие поля исходных таблиц следует включить в запись таблицы запроса и какие записи нужно отобразить. Таблица запроса может быть использована наряду с другими таблицами базы при обработке данных. Запрос может формироваться с помощью конструктора запросов или инструкции языка SQL. Запросы на изменение позволяют обновлять, удалять или добавлять данные в таблицы, а также создавать новые таблицы на основе существующих.
- **Схема данных (Relationships)** определяет, с помощью каких полей таблицы связываются между собой, как будет выполняться объединение данных этих таблиц, нужно ли проверять связную целостность при добавлении и удалении записей, изменении ключей таблиц. Схемы данных не отображаются в области навигации в окне базы данных Access. Для отображения схемы данных в базах данных Access используется команда **Схема данных (Relationships)**, размещенная на вкладке ленты **Работа с базами данных (Database Tools)** в группе **Отношения**.
- **Формы (Forms)** являются основным средством создания диалогового интерфейса приложения пользователя. Форма может создаваться для работы с электронными документами, сохраняемыми в таблицах базы данных. Вид таких документов может соответствовать привычному для пользователя бумажному документу. Форма используется для разработки интерфейса по управлению приложением. Включаемые в форму процедуры обработки событий позволяют управлять процессом обработки данных в приложении. Такие процедуры хранятся в модуле формы. В формы могут вставляться рисунки, диаграммы, звуковые фрагменты, видео. Возможна разработка форм с набором вкладок, с каждой из которых связано выполнение той или иной функции приложения.
- **Отчеты (Reports)** предназначены для формирования на основе данных базы выходных документов любых форматов, содержащих результаты решения задач пользователя, и вывода их на печать. Как и формы, отчеты могут включать процедуры обработки событий. Использование графических объектов позволяет дополнять данные отчета иллюстрациями. Отчеты обеспечивают возможность анализа данных при использовании фильтрации, агрегирования и представления данных источника в различных разрезах.
- **Макросы (Macros)** являются программами, состоящими из последовательностей макрокоманд, которые выполняются по вызову или при наступлении некоторого события в объекте приложения или его элементе управления. Макросы данных выполняются при наступлении некоторого события в исходных таблицах. Макросы позволяют автоматизировать некоторые действия в приложении пользователя. Создание макросов осуществляется в диалоговом режиме путем выбора нужных макрокоманд и задания их параметров, используемых при выполнении.

Макросы данных используются для добавления логики к данным и сосредоточения ее в исходных таблицах. В веб-приложениях Access, публикуемых на сайтах SharePoint, для программирования можно использовать только макросы, т. к. код VBA несовместим со средствами веб-публикации.

- **Модули (Modules)** содержат процедуры на языке Visual Basic for Applications. Могут создаваться процедуры-подпрограммы, процедуры-функции, которые разрабатываются пользователем для реализации нестандартных функций в приложении пользователя, и процедуры для обработки событий. Использование процедур позволяет создать законченное приложение, имеющее собственный графический интерфейс пользователя, позволяющий запросить выполнение всех функций приложения, обработать все ошибки и нестандартные ситуации.

В Access для удобства пользователя объекты в области навигации базы данных могут быть объединены в *пользовательские группы* по функциональному или иному признаку. Группы содержат ссылки на объекты базы данных различных типов. Группы, в свою очередь, объединяются в *категории*. С помощью такой организации объектов базы данных может быть разработан интерфейс пользовательского приложения, обеспечивающий доступность только к категориям и группам, наглядно и понятно представляющим функциональность приложения.

Размещение базы данных

Все таблицы, а также другие объекты базы данных Access: запросы, формы, отчеты, макросы и модули, построенные для этой базы, и внедренные объекты могут размещаться на диске в одном файле формата ACCDB. Это упрощает технологию ведения базы данных и приложения пользователя, а также обеспечивает высокую компактность размещения всех объектов базы данных на диске и эффективность обработки данных.

Когда база данных открыта, для корректного внесения изменений требуются блокировки данных разных уровней. Контроль над ними осуществляется с помощью *файла блокировки*. Если в MS Access 2010/2016 открыт MDB-файл, для контроля блокирования создается файл с расширением ldb и тем же именем, что у MDB-файла. Для файлов в формате ACCDB блокирование управляется файлом с расширением laccdb. Как LDB-, так и LACCDB-файлы уничтожаются автоматически, когда база данных будет закрыта всеми пользователями.

Введение отдельных блокирующих файлов для файлов Access 2010/2016 и файлов, созданных в более ранних версиях Access, обеспечивает одновременное открытие файлов mdb и accdb с одинаковым именем, и это не приводит к возникновению конфликтов в блокирующем файле, поскольку будут созданы два разных блокирующих файла. Также можно открывать один и тот же файл mdb в Access 2010/2016 и в более ранней версии Access одновременно, поскольку обе версии используют один и тот же блокирующий файл ldb.

База данных Access 2016 может быть превращена в базу данных, доступную только для выполнения, не доступную для изменений и скрывающую свой код. Для этого она должна быть скомпилирована и сохранена в файле формата ACCDE. В процес-

се такого преобразования из базы данных удаляется весь исходный текст программ на VBA, и база сжимается, что значительно сокращает размер файла. В базе данных формата ACCDE код VBA может только выполняться, но просматривать и изменять его нельзя. При этом у пользователей нет разрешений на изменение структуры форм, отчетов или модулей. Для преобразования файла базы данных ACCDB в формат ACCDE на странице, открываемой при выборе значка **Файл** (File), выберите **Сохранить как** (Save As), далее **Сохранить базу данных как** (Save Database As), в группе **Дополнительно** (Advanced) — пункт **Создать ACCDE** (Make ACCDE) (Компиляция исполняемого файла) и нажмите кнопку **Сохранить как** (Save As).

Для преобразования (конвертации) базы данных из предыдущих версий в текущую и наоборот в Access 2016 предназначена команда **Сохранить базу данных как** (Save Database As), доступная в разделе **Сохранить как** (Save As) на странице, открываемой при выборе значка **Файл** (File). Если база данных Access 2007/2016 использует возможности, не поддерживаемые более ранними версиями, выполнение преобразования невозможно.

Сжать и восстановить базу данных (Compact & Repair Database) можно одноименной командой, доступной в разделе **Сведения** (Info) на странице, открываемой при выборе значка **Файл** (File), или на вкладке **Работа с базами данных** (Database Tools) в группе **Сервис** (Tools). Команда реорганизует базу данных на диске с целью улучшения объемно-временных характеристик. Она перезаписывает фрагментированный в результате удаления и добавления объектов файл базы данных и сокращает его размер, а также восстанавливает поврежденную базу данных.

Для обеспечения общего доступа к базе данных предназначена команда **SharePoint**, доступная в разделе **Сохранить как** (Save As) на странице, открываемой при выборе значка **Файл** (File). При выполнении этой команды нужно указать адрес сайта на сервере SharePoint — например: <https://unecon.sharepoint.com>. При этом будет открыт весь контент сайта. Открыв нужный сайт — например, **БИ_1301**, выберите библиотеку **Документы**, где и сохраните базу данных. Теперь достаточно подключиться к серверу SharePoint, чтобы получить доступ к базе данных из любого места. База данных скачивается на локальный компьютер и может быть открыта в Access. При этом выполненные в базе изменения могут быть сохранены на сервере, где делаются доступными всем, получившим доступ к этому документу.

Шаблоны баз данных

Сразу после запуска Access отображается стартовое окно Access 2016 (рис. 1.3) — это место, где можно управлять файлами. В момент открытия здесь представлен список последних использованных баз данных и команда **Открыть другие Файлы** (Open Other Files) баз данных, а справа, в области создания баз данных, предоставлена возможность создать новую базу: **Пустая база данных рабочего стола** (Blank Desktop Database) и **Пользовательское веб-приложение** (Custom Web App). Кроме того, если компьютер подключен к Интернету, в области создания баз данных отображаются многочисленные шаблоны для создания различных типовых баз данных.

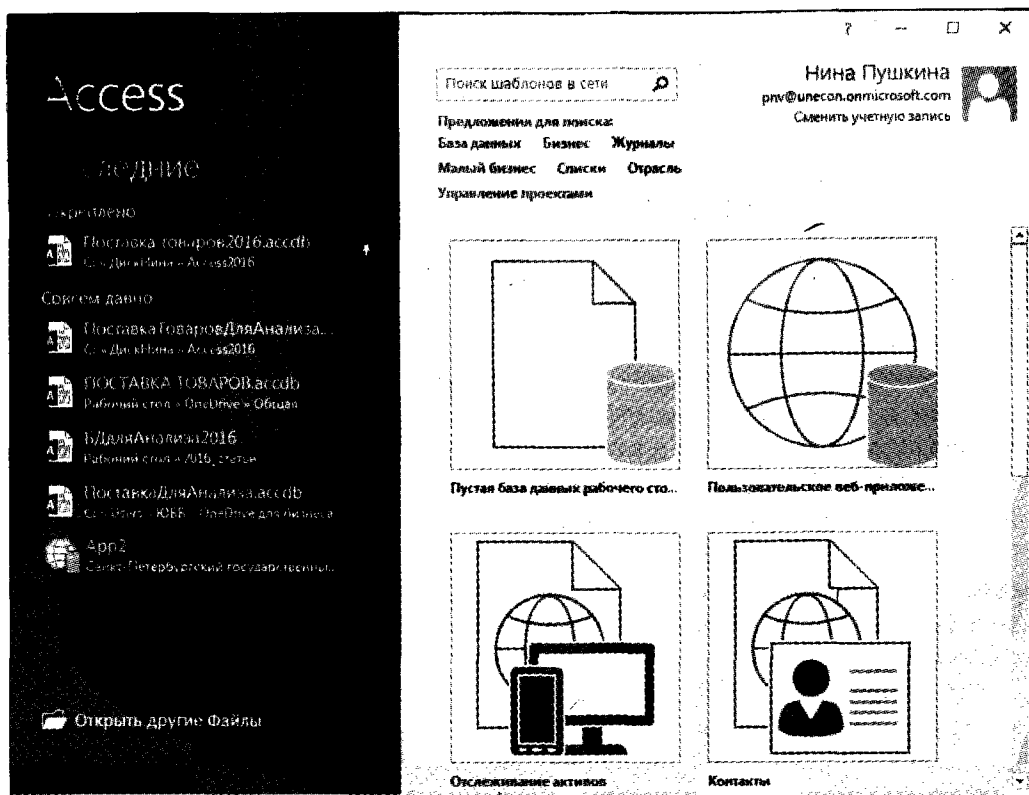


Рис. 1.3. Стартовое окно Access 2016

Шаблоны типовых баз данных включают все необходимые таблицы, формы, запросы и отчеты для предметных областей различных сфер деловой и личной жизни. Эти стандартные приложения можно использовать без какой-либо модификации и настройки, либо взять их за основу и адаптировать в соответствии с характером информации, которую требуется сохранять и обрабатывать. Выбрав нужный шаблон, достаточно определить место, где следует сохранить создаваемую базу данных, и нажать кнопку **Создать (Create)** — автоматически создается и открывается база данных, соответствующая выбранному шаблону, и пользователю остается только ввести данные.

Обратите внимание на два вида шаблонов (рис. 1.4): с помощью одних (левый значок) создаются локальные базы данных, с помощью других (правый значок) — веб-приложения, доступные в Интернете.

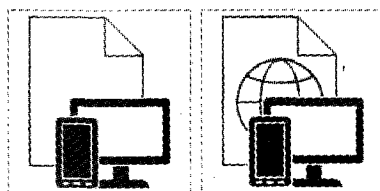


Рис. 1.4. Шаблоны для создания баз данных

Соответственно, при определении расположения базы данных отображаются разные окна. Для размещения веб-приложения необходимо иметь доступ к сайту SharePoint. При этом окно определения местоположения веб-приложения будет иметь вид, показанный на рис. 1.5.

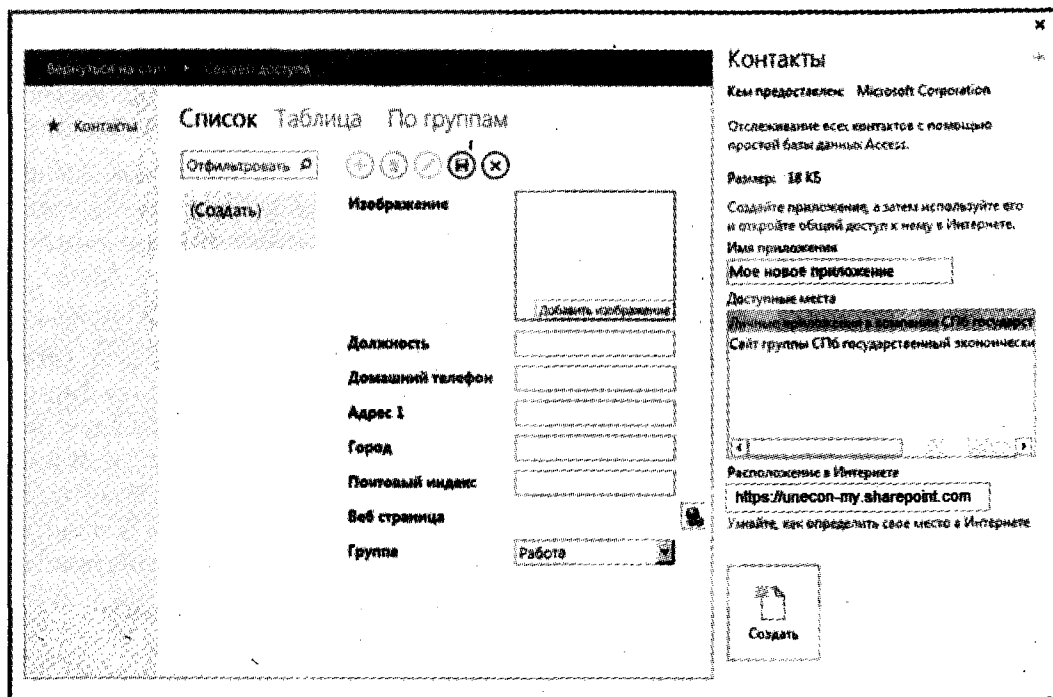


Рис. 1.5. Определение места для веб-приложения, создаваемого на основе шаблона

При выборе шаблона в области просмотра отображаются сведения: кем он предоставлен, размер шаблона, краткое его описание и, если необходимо, дополнительные системные требования.

Типовые базы данных позволят начинающему пользователю познакомиться с основными принципами построения базы данных и приложения пользователя и получить навыки практической работы в среде Access. Работая с типовой базой, пользователь научится просматривать и изменять данные через формы, делать запросы для получения сведений из связанных таблиц, готовить отчеты.

Шаблоны приложений можно найти в Интернете, воспользовавшись полем **Поиск шаблонов в сети** (см. рис. 1.3), где следует указать тему нужного шаблона — например, Образование. В Интернете предлагается множество разнообразных шаблонов для простых и сложных задач, которые можно использовать в коммерческих целях и для собственных нужд.

Шаблоны Access позволяют быстрее создавать приложения и приступать к работе с ними — даже если пользователь не имеет навыков проектирования баз данных и не знает языков программирования.

Однако, используя типовую базу данных, трудно рассчитывать, что она в полной мере удовлетворит потребности пользователя. Базу данных, созданную по шаблону, можно изменить и расширить, но эта работа требует от пользователя практически тех же знаний, что и создание новой базы данных.

Мастера Access

Access располагает разнообразными диалоговыми средствами, которые позволяют создавать объекты базы данных и приложения, не прибегая к программированию.

Множество мастеров Access позволяет автоматизировать процесс создания таблиц базы данных, запросов, форм, отчетов; анализировать таблицы базы данных и выполнять многие другие работы. Практически для любых работ имеется *мастер* (Wizard), который поможет их выполнить.

- *Мастера по созданию форм и отчетов* упрощают и ускоряют процесс создания однотабличных и многотабличных форм и отчетов. Так, выбрав в области навигации таблицу базы данных или запрос, одним щелчком можно создать форму, отображающую поля только одной записи в столбец, или отображающую несколько записей в виде таблицы, или смешанное представление записей. В диалоге с другим мастером пользователю достаточно выбрать таблицы и поля, которые необходимо включить в форму, выбрать источник основной и подчиненной части формы. Мастера по разработке форм и отчетов автоматически создают инструкцию SQL, описывающую источник записей для формы или отчета, что избавляет пользователя от подготовки запроса.
- *Мастер подстановок* (Lookup Wizard) создает в поле таблицы или формы раскрывающийся список значений из полей другой таблицы или запроса для выбора и ввода в поле нужного значения. Созданные в полях таблиц списки наследуются при включении этих полей в форму (поле со списком).
- *Мастера запросов* позволяют создавать простые запросы на выборку или запросы на выборку, в которых выполняются групповые операции над данными из одной или нескольких таблиц.
- *Мастер перекрестных запросов* (Crosstab Query Wizard) формирует из таблицы или запроса базы данных таблицу подобную сводной, в которой одно поле используется в качестве заголовков строк, второе — столбцов, а на их пересечении размещаются итоговые данные, рассчитываемые по значениям третьего поля.
- *Мастер по созданию диаграмм* (Chart Wizard) обеспечивает создание в формах и отчетах диаграмм, базирующихся на данных в таблицах или запросах. На рис. 1.6 приведена форма, позволяющая просматривать в справочнике данные о товаре и синхронно отображать диаграмму с количеством заказанного по договорам (в примере Д111, Д222, Д333, Д777). Вызывается мастер с помощью элемента управления **Диаграмма** (Chart), расположенного на вкладке ленты конструктора форм в группе **Элементы управления** (Controls). Ответив на ряд элементарных вопросов, легко можно получить диаграмму, связанную с выбранными в форме

данными. Двойным щелчком мышью на диаграмме или в контекстном меню может быть выполнен переход к ее редактированию с помощью Microsoft Chart.

- **Мастер кнопок** (Command Button Wizard и Page Command Button Wizard) создает командные кнопки — элементы управления в форме. При построении кнопки мастер предлагает большой набор действий, которые могут быть выполнены при щелчке на кнопке. Для выбранного действия мастер создает встроенный макрос и связывает его с событием **Нажатие кнопки** (On Click). Формы с командными кнопками, связанными с макросами, позволяют реализовать удобный интерфейс пользователя для управления приложением.
- **Мастер по анализу таблиц** (Table Analyzer Wizard) позволяет выполнить нормализацию данных базы. Мастер разделяет ненормализованную таблицу с дублированными данными на две или несколько таблиц меньшего размера, в которых данные сохраняются без повторений. Вызывается мастер с помощью команды **Анализ таблицы** (Table Analyzer), расположенной на стандартной вкладке ленты **Работа с базами данных** в группе **Анализ** (Analyzer).

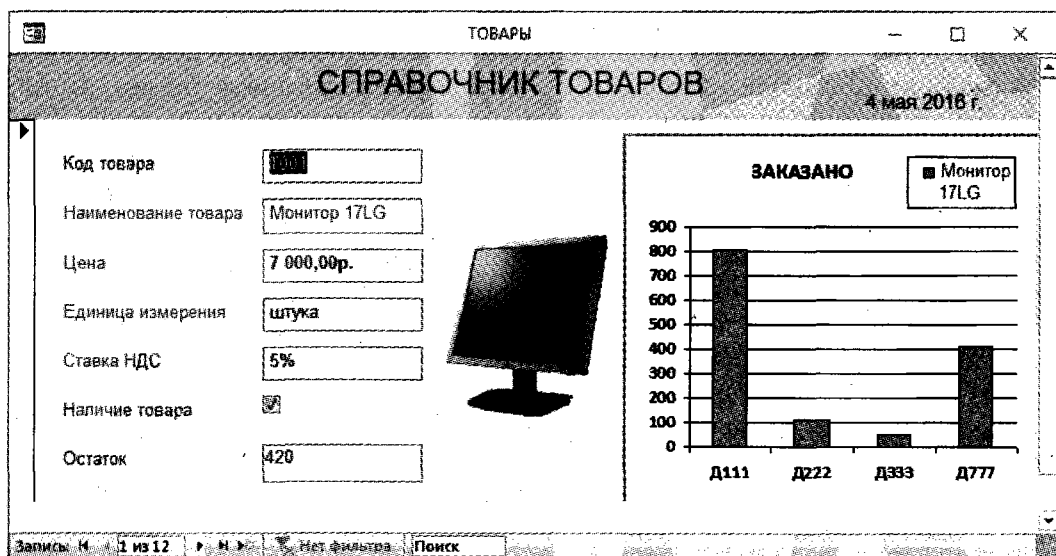


Рис. 1.6. Форма с диаграммой, построенной мастером

Средства конструирования объектов

Для автоматизации процесса создания объектов базы данных — таблиц, схемы базы данных, запросов и объектов приложения: форм, отчетов, макросов — используются специализированные диалоговые графические средства, называемые *конструкторами* (Design). Конструктор предоставляет пользователю набор элементов, с помощью которых быстро создается и модифицируется объект. Для конструирования макета формы, отчета используются *элементы управления*, которые доступны в режиме макета и режиме конструктора на вкладках ленты **Работа с макетами**

форм/отчетов (Form Layout Tools) или Инструменты конструктора форм/отчетов (Form Design Tools).

В режиме макета пользователь, изменяя макет формы или отчета, одновременно видит данные почти в том же виде, в каком они отобразятся в режиме формы или после вывода отчета на печать. Режим макета представляет собой наиболее наглядный режим для изменения макета форм и отчетов. Его можно использовать для внесения практически любых изменений. Поскольку при внесении изменений можно просматривать данные, в этом режиме очень удобно задавать размеры элементов управления и выполнять другие задачи, связанные с внешним видом и удобством формы или отчета. Некоторые задачи невозможно выполнить в режиме макета, и тогда необходимо переключиться в режим конструктора. Access выдает сообщение о том, что для внесения изменений следует переключиться в режим конструктора.

В Access формирование источника данных для формы или отчета может быть выполнено с помощью *списка доступных полей*. При этом оно сводится к простому перетаскиванию полей из таблиц базы данных в нужное место формы/отчета в режиме конструктора или макета.

Для упрощения внесения изменений в объекты базы данных Access предлагает технологию *интеллектуальной замены имен объектов в базе данных*. При этом автоматически исправляются ошибки, вызванные переименованием таблиц, полей, форм, отчетов, запросов, текстовых блоков или других элементов управления. Реализуется эта технология за счет того, что каждый именуемый объект (или элемент) базы данных имеет внутренний уникальный идентификатор, а имя является только псевдонимом. При переименованиях изменяется лишь псевдоним, и при необходимости в других объектах корректируются все ссылки на объект. Для применения этой технологии следует установить параметры *отслеживать автозамену имен (Track name AutoCorrect info)* и *выполнять автозамену имен (Perform name AutoCorrect)* в окне **Параметры Access (Access Options)** на вкладке **Текущая база данных (Current Database)** в разделе **Параметры автозамены имен (Name AutoCorrect Options)**. Окно **Параметры Access (Access Options)** открывается соответствующей командой на странице, открываемой при выборе значка **Файл (File)**.

Средства программирования

Наряду с диалоговыми средствами создания объектов базы данных и объектов приложения, которые позволяют решить многие задачи пользователя, Access располагает мощными средствами программирования. Эти средства могут использоваться как для доработки приложений, созданных диалоговыми средствами, так и для реализации сложных задач и создания приложений с необходимым пользователю интерфейсом в целом. Без использования программного кода практически невозможно получить законченное автоматизированное приложение пользователя.

Одним из средств программирования в Access является язык макрокоманд. Программы, созданные на этом языке, называемые *макросами*, позволяют легко связывать отдельные действия, реализуемые с помощью форм, запросов, отчетов. Макросы управляются событиями, которые вызываются действиями пользователя при

диалоговой работе, прежде всего, с данными через формы — например, нажатие кнопки, изменение данного в поле, — или системой.

Простой язык макрокоманд и конструктор макросов с ясным и понятным интерфейсом, поддерживающий функции IntelliSense, позволяют при малой трудоемкости интегрировать объекты приложения и организовать процесс обработки данных. Конструктор макросов упрощает создание, редактирование и автоматизацию логики базы данных, позволяет сокращать количество ошибок кода.

Помимо обычных макросов конструктор макросов позволяет создавать макросы данных, которые связываются с событиями в таблице и позволяют выполнять определенные действия при изменении, вставке или удалении записи. Например, с помощью макросов можно проверять данные или выполнять вычисления. Макросы данных позволяют добавлять логику к данным и сосредотачивать ее в исходных таблицах. По сути, макросы данных позволяют реализовать триггеры в базе данных Access.

С помощью конструктора макросов и макросов данных можно распространить автоматизацию за пределы клиентского приложения Access, например, на базы данных веб-приложения Access, размещаемого на сайте SharePoint, и другие приложения, обновляющие таблицы.

Наряду с языком макрокоманд Access включает развитую интегрированную среду объектно-ориентированного программирования Visual Basic for Applications (VBA), позволяющую реализовать любые программные решения. Программы на VBA реализуются процедурами, которые объединяются в объектах, называемых *модулями*.

В VBA база данных рассматривается как совокупность объектов (таблиц, форм, отчетов, их элементов и т. д.), имеющих свойства и методы, реализующие заранее определенные действия над объектами. Структурированность объектов базы данных упрощает освоение этого языка и создание приложений. Управление выполнением программ в диалоговых приложениях VBA осуществляется событиями, вызываемыми действиями пользователя или системы.

Среда VBA объединяет разнообразные наглядные графические инструменты: редактор VBA, окно разрабатываемого проекта, окно свойств объектов проекта, окно просмотра объектов, отладчик и др. Все инструменты унифицированы и являются общими для всех приложений Microsoft Office, для Visual Basic, а также продуктов ряда других фирм.

Приложения, разрабатываемые на VBA, могут выполняться только в той среде, в которой поддерживается VBA, в то время как Visual Basic ориентирован на полностью самостоятельную разработку автономно выполняющихся приложений. Язык VBA является производным от самостоятельной системы программирования Visual Basic и имеет с ней много общего. Их синтаксис и интерфейс практически одинаковы.

Заметим, что код VBA несовместим со средствами веб-публикации, поэтому при разработке веб-приложения Access для выполнения задач программирования могут использоваться только макросы.

Интеграция и использование внешних данных

Access продолжает поддерживать технологию OLE (Object Linking and Embedding, связь и внедрение объектов), обеспечивающую возможность интеграции данных различных приложений в составном документе. С помощью OLE пользователь может внедрять объекты другого приложения в объекты базы данных или устанавливать с ними связи. Активизация внедренного объекта запускает программу, которая его создала, и пользователь может изменить объект. При установлении связи с объектом он сохраняется в отдельном файле, а не в базе данных. За счет этого он может обновляться независимо, а в базе данных всегда будет представлена последняя версия объекта. Надо иметь в виду, что при изменении местоположения файла связь с ним должна обновляться пользователем.

Внедряемыми или связываемыми объектами могут быть документы различных приложений Windows: рисунки, графики, электронные таблицы, звуковые или видеофайлы. Например, в таблице наряду с обычными реквизитами, характеризующими информационный объект, может храниться любая графическая информация о нем: схемы, чертежи, диаграммы, рисунки и т. п. Таким образом, в Access расширяется традиционное понятие данных, хранимых в базе и представляющих информационные объекты.

Кроме того, в Access обеспечивается хранение одного или нескольких файлов разных типов: документов Word, презентаций PowerPoint, изображений и т. п. — в поле таблицы базы данных, имеющем тип данного **Вложение (Attachment)**. Вложения позволяют хранить данные более рационально.

Access может использовать данные различных внешних источников. Внешними источниками данных могут служить таблицы других баз данных Access, Oracle и Microsoft SQL Server, электронные таблицы Microsoft Excel, таблицы и списки HTML, файлы XML, списки Windows SharePoint, текстовые файлы и др., находящиеся на локальном компьютере или на сервере в локальной, корпоративной сети или в Интернете.

Для взаимодействия с внешними источниками данных применяется специальное программное обеспечение — *интерфейсы*. Широко используемым интерфейсом является ODBC (Open Database Connectivity, *открытый интерфейс подключения к базам данных*), который служит, прежде всего, для доступа к базам данных.

К источникам данных ODBC могут отправляться запросы, таблицы таких источников данных могут использоваться в базе данных Access, как связанные. Связанные таблицы отображаются в области навигации базы данных, оставаясь в исходном файле вне файла базы данных Access. Их можно использовать при создании запросов, форм, отчетов, причем можно объединять их данные с данными из локальных объектов Access. Связывание позволяет использовать данные другой программы, не импортируя их. При этом просматривать и изменять данные можно как в исходной программе, так и в базе данных Access.

Данные внешних источников могут импортироваться в базу данных Access. На вкладке **Внешние данные (External Data)** в группе **Импорт и связи (Import & Link)**

представлены доступные для этой операции источники данных. Импорт и связывание могут быть выполнены для данных из таких источников, как Access, Excel, база данных ODBC, текстовый файл, XML-файл, список SharePoint, документ HTML, папка Outlook.

Возможен экспорт таблиц, запросов, форм и отчетов, выделенной части объекта в режиме таблицы из базы данных Access в форматы других приложений. Операции экспорта представлены в одноименной группе на вкладке **Внешние данные**.

Выполнение операций импорта или связывания данных, как и экспорта данных из базы, для большинства форматов требует лишь указания, где расположены данные, и выбора способа их хранения в базе данных.

Многопользовательская база данных Access

База данных, как правило, содержит данные, необходимые многим пользователям. Создание многопользовательской базы данных Access и получение одновременного доступа нескольких пользователей к общей базе данных возможно в одноранговой сети персональных компьютеров или в сети с файловым сервером.

Под *одноранговой* понимается сеть, каждый компьютер которой может предоставлять остальным подключенным к сети компьютерам доступ ко всем или некоторым своим ресурсам, т. е. являться сервером и клиентом одновременно. Одноранговая сеть может управляться встроенными компонентами настольных операционных систем от Windows XP до Windows 10.

Сети больших масштабов используют выделенные *файловые серверы*. В сети, поддерживающей концепцию файлового сервера, база данных Access размещается на компьютере, выделенном в качестве файлового сервера, а СУБД Access, как правило, устанавливается на каждый пользовательский компьютер. Обработка данных базы осуществляется на компьютерах пользователей. Поэтому по сети передаются с сервера на компьютер клиента большие объемы данных, что сильно загружает ее и делает невозможным одновременное обслуживание большого числа пользователей.

Работа локальной сети с файловым сервером обеспечивается рядом сетевых операционных систем. Наиболее популярными являются Microsoft Windows Server и Novell NetWare.

Сеть предоставляет аппаратную и программную поддержку обмена данными между компьютерами. Сетевая операционная система обеспечивает защиту базы данных, размещенной в сети, предоставляя пользователям доступ к определенным сетевым ресурсам — например, папкам, файлам, только по предъявлению имени и пароля. Кроме того, для пользователя папки или отдельного файла могут назначаться права доступа, ограничивающие их использование, например, только чтением.

Access не следит за разграничением доступа разных пользователей к базе данных, но автоматически обеспечивает защиту данных от одновременной их корректировки несколькими пользователями. Для обеспечения защиты данных от одновременной их корректировки несколькими пользователями в Access предусматривается

блокировка на уровне страниц, при которой блокируются все записи одной страницы. Дополнительно к блокировке страниц Access поддерживает блокировку на уровне записи. Блокирование не допускает изменения записи другими пользователями, пока текущее изменение не будет закончено.

Выбор режима блокировки на уровне записей по умолчанию в текущей базе данных производится параметром **Открывать базу данных с использованием блокировки на уровне записей** (Open databases using record level locking) в окне **Параметры Access** (Access Options) на вкладке **Параметры клиента** (Client Settings) в группе **Дополнительно** (Advanced). Если флажок этого параметра сброшен, по умолчанию будет использована блокировка на уровне страницы. Выбранный вариант применяется к данным в формах, таблицах и программах, использующих объект Recordset для перебора записей. Этот параметр не применяется к запросам на изменение или программам, выполняющим массовые операции с использованием инструкций SQL.

Прежде чем начать работу нескольких пользователей с общей базой данных, необходимо выполнить разделение локальной базы данных. Разделение базы данных предполагает выделение таблиц текущей базы данных в новую базу. При этом текущая база становится интерфейсной базой данных, а база данных с таблицами помещается в сетевое расположение, указанное при выполнении процедуры разделения.

После разделения базы данных нужно распространить интерфейсную базу данных среди пользователей, чтобы они могли начать работу с ней.

Для разделения базы данных откройте ее и выполните команду **База данных Access** (Access Database) из группы **Перемещение данных** (Move Data) на вкладке **Работа с базами данных** (Database Tools). Первое окно соответствующего мастера приведено на рис. 1.7.

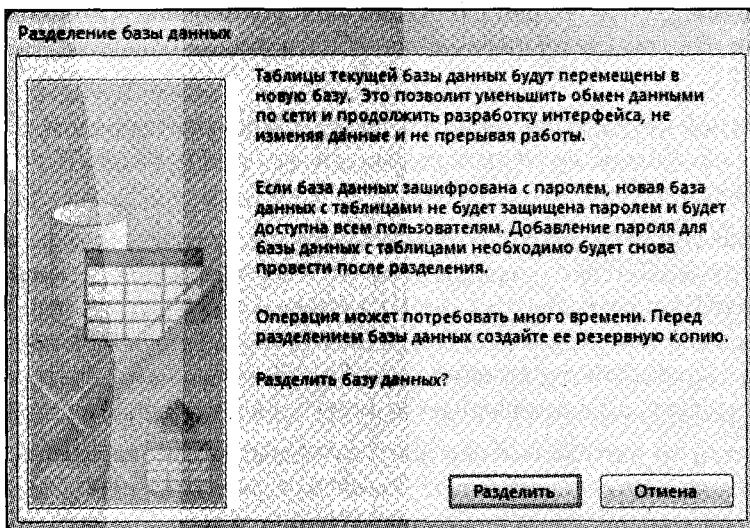


Рис. 1.7. Разделение базы данных

Для выполнения процедуры разделения достаточно в следующем окне **Создание базы данных с таблицами** (Create Back-End Database) указать имя, местоположение новой базы данных с таблицами и нажать кнопку **Разделение** (Split). Поскольку разделяемая база данных преобразуется в интерфейсную, целесообразно предварительно сделать ее копию.

Для базы данных с таблицами рекомендуется использовать имя, предложенное программой Access. Оно представляет собой исходное имя файла, к которому добавлены символы `_be`, указывающие, что это база данных с таблицами.

Чтобы путь к файлу, находящемуся на каком-либо компьютере, был одинаковым при доступе с любого компьютера этой сети, используйте для указания его местоположения путь UNC. Например, `\\Сервер\Общая_папка\ИмяБД_be.accdb`. Выбирайте такое расположение, к которому можно предоставить доступ всем пользователям базы данных.

Так как пользователи, работая в интерфейсной базе данных, получают доступ к общей базе данных с таблицами через связанные таблицы, нельзя произвольно менять ее имя и местоположение. Переместить базу данных с таблицами или заменить ее на другую можно с помощью команды **Диспетчер связанных таблиц** (Linked Table Manager), которая расположена на вкладке **Внешние данные** (External Data) в группе **Импорт и связывание** (Import & Link).

Работа Access с базой данных SQL Server

Работа с общей базой данных в сети с файловым сервером становится неэффективной уже при одновременной работе 15 пользователей. На обеспечение эффективной работы большого числа пользователей с общей базой данных ориентирована технология *«клиент — сервер»*. В этой технологии пользователь-клиент передает со своего компьютера запрос на компьютер сервера, там СУБД обрабатывает запрос и обратно посылает только результат выполнения запроса. Таким образом значительно снижается объем передаваемых по сети данных.

Приложение пользователя разрабатывается и выполняется под управлением СУБД Access на компьютере клиента. Общая база данных размещается на мощном компьютере, где функционирует сервер баз данных, управляемый СУБД SQL Server. Эта СУБД выполняет обработку данных, размещенных на сервере, и отвечает за их целостность и сохранность. Для доступа к данным базы на сервере используется язык структурированных запросов SQL.

SQL-серверы баз данных являются самыми мощными приложениями для сетевой обработки данных. Широко известны серверы баз данных: SQL Server фирмы Microsoft и Oracle Server фирмы Oracle.

Подключение из Access к серверам баз данных SQL может быть осуществлено с помощью драйверов ODBC. Каждому серверу баз данных соответствует свой драйвер ODBC. Драйверы ODBC для MS SQL Server и Oracle SQL Server включены в комплект поставки MS Access.

Использование унифицированного языка запросов SQL позволяет работать с одной и той же базой данных сервера разным пользователям из различных приложений.

Данные из базы могут получать Access, Excel, FoxPro и многие другие приложения, использующие протокол ODBC, посылая запросы на языке SQL серверу баз данных.

Приложение Access взаимодействует с данными, расположенными на сервере, несколькими способами. Можно посылать на сервер запросы на языке SQL сервера. С помощью запроса на выборку можно получать необходимые данные в виде виртуальной таблицы — таблицы запроса — и далее использовать эту таблицу в качестве источника данных в формах или отчетах. Можно посылать на сервер обновленные данные, а также создавать или изменять таблицы в базе данных сервера. Кроме того, в локальной базе данных могут быть созданы связанные таблицы, отображающие данные из таблиц SQL сервера. Эти таблицы могут обрабатываться в приложении наряду с локальными таблицами базы. Возможен импорт таблиц сервера в локальную базу данных и экспорт таблиц локальной базы данных на сервер.

Установка связи с данными из базы данных ODBC выполняется командой **База данных ODBC (ODBC Database)** из группы **Импорт и связи (Import & Link)** на вкладке **Внешние данные (External Data)**.

Интернет-технологии

Пользователи баз данных все больше ориентируются на уникальные возможности быстрого сбора и совместного использования информации, предоставляемые интернет-технологиями. Корпоративные сети, являясь сетями уровня предприятия, также базируются на клиент-серверных и интернет-технологиях. Эти сети могут подключаться или не подключаться к Интернету (рис. 1.8). Технологии Интернета

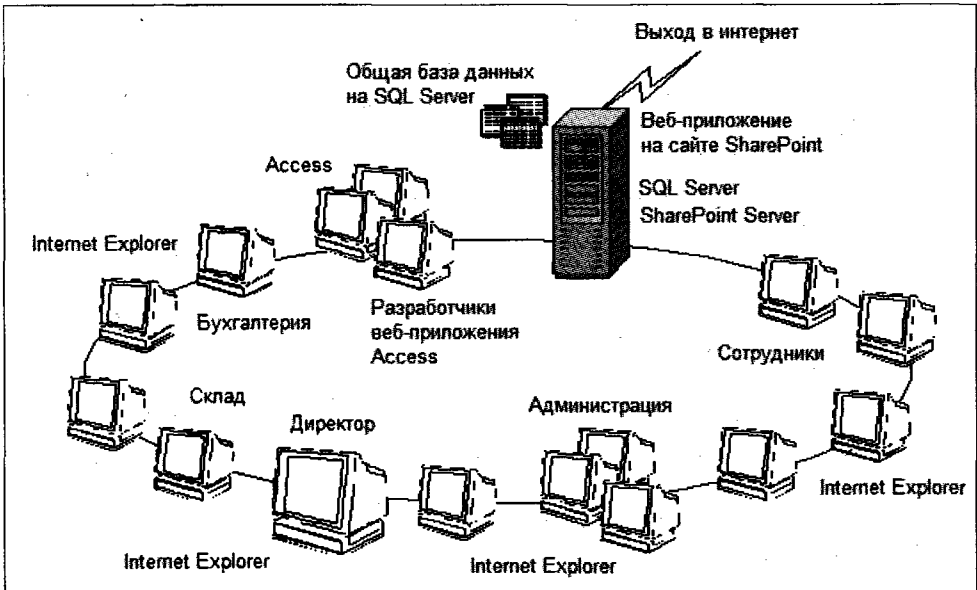


Рис. 1.8. Корпоративная сеть с общей базой данных веб-приложения

позволяют получить доступ к информации всего предприятия со своего рабочего места, не заботясь о совместимости аппаратных и программных платформ, используя обычную программу просмотра — веб-браузер.

Новые технологии в Access 2013/2016 позволяют создавать специальные веб-приложения (для их описания служит термин «app»), опубликованные на сайтах Microsoft SharePoint Server. Они предназначены для организации совместной работы с данными и взаимодействия пользователей. Доступ пользователей к содержимому веб-сайтов осуществляется через веб-браузер или Access 2013/2016 из любой точки корпоративной сети или Интернета. Веб-приложение можно разместить как на сайтах корпоративного сервера SharePoint, так и в «облаках» на сайтах Office 365. Для доступа к сайтам необходимо использовать выпуски Office 365, обеспечивающие среду SharePoint. База данных веб-приложения размещается на SQL Server в корпоративной сети, определенном в SharePoint, или на «облачном» сервере баз данных SQL Windows Azure при работе с Office 365. Применение SQL Server обеспечивает наилучшую производительность и целостность данных. Пользователи могут совместно работать с веб-приложением с помощью стандартного браузера, не устанавливая приложение Access на компьютере. Это делает простым совместный доступ к корпоративной информации в среде настольных систем. В то же время сохраняется возможность изменять, дорабатывать веб-приложение в Access.

На сервере SharePoint службы Access обрабатывают данные и готовят их к просмотру в окне браузера, где пользователи в соответствии со своими правами смогут изменять, добавлять, удалять или только просматривать данные. Права пользователю веб-приложения назначаются с помощью средств SharePoint. Средства Access позволяют скрыть или заблокировать редактирование таблиц базы данных веб-приложения.

Создается веб-приложение, как и база данных для настольных компьютеров, в стартовом окне Access или в любой момент после закрытия активной базы данных выбором команды **Пользовательское веб-приложение** (Custom web app) на странице, открываемой при выборе значка **Файл** (File).

При создании веб-приложения указывается сайт, на котором оно будет храниться и выполняться. Чтобы в окне создания веб-приложения отобразились доступные для его размещения сайты, пользователь должен зарегистрироваться на сайте SharePoint. При использовании Office 365, включающем SharePoint, для регистрации используйте адрес <http://login.microsoftonline.com>. В окне регистрации введите логин и пароль учетной записи организации и нажмите кнопку **Вход** (Sign In). После регистрации сведения об учетной записи сохраняются в Access и могут быть использованы для повторного подключения. Для просмотра учетной записи следует на странице, открываемой при выборе значка **Файл** (File), выбрать опцию **Учетная запись** (Account). Кроме сведений из учетной записи вам будут показаны сведения о подключенных службах и предоставлена возможность сменить учетную запись. При смене учетной записи путем ввода нового адреса может быть добавлена новая учетная запись. По ссылке **Выход** учетная запись удаляется из списка учетных записей в Access.

Для создания таблиц в базе данных веб-приложения следует воспользоваться конструктором таблиц или импортом таблиц из внешних источников. Для создания связи между таблицами служит поле подстановки. Поле подстановки создается с помощью мастера в таблице на стороне связи «многие» и указывает на таблицу на стороне «один». Уже при создании или импорте таблицы создается представление — форма, через которую пользователь в браузере осуществляет просмотр, добавление, изменение, удаление записей таблицы.

После создания веб-приложения посетители сайта SharePoint, используя только браузер и обладая необходимыми разрешениями, могут работать с базой данных. Формы являются основным средством ввода, просмотра и редактирования данных в базе. При открытии формы браузер получает необходимые данные с сервера SharePoint. Данные форм можно фильтровать и сортировать без их повторного получения с сервера. Вся обработка SQL-кода выполняется на сервере, что повышает производительность сети за счет ограничения обмена данными с результирующими наборами. В веб-браузере доступна область навигации, обеспечивающая удобную работу с объектами базы данных.

SharePoint устраняет все проблемы, связанные с параллельным доступом к данным. Работа с приложением базы данных через браузер выполняется на основе разрешений SharePoint, определяющих возможность доступа к тем или иным объектам. Кроме того, средствами SharePoint можно воспользоваться для выполнения таких функций, как запись версий данных, подписка на получение предупреждений при внесении изменений. Таким образом обеспечивается эффективный общий доступ к базе данных, а также предоставляются новые возможности для совместной работы через Интернет.

Начало работы в Microsoft Access 2016

Запуск Access

Для работы с Access 2016 на локальном компьютере пользователя должна быть установлена одна из операционных систем: Windows 7 с пакетом обновления 1 или более поздней версии, Windows Server 2008 R2 или Windows Server 2012 и СУБД Access 2016. Для того чтобы начать работу в СУБД Access, необходимо после загрузки операционной системы ее запустить. Это можно сделать, например, так: в нижней части рабочего стола Windows 7 на панели задач нажать кнопку **Пуск**, в открывшемся меню выбрать **Все программы**. Из списка программ, установленных на компьютере, выбрать **Microsoft Office | Access 2016** и запустить эту СУБД. Access может быть запущен и из списка недавно использовавшихся программ, также представленного в меню **Пуск**.

Для быстрого запуска Access удобно закрепить программу на панели задач или иметь ярлык этой программы на рабочем столе Windows. Создать ярлык можно разными способами. Например, выберите программу Microsoft Access, как при ее запуске. Нажмите правую кнопку мыши. В контекстном меню выберите команду **Отправить** и в ее меню выполните команду **Рабочий стол (создать ярлык)**.

Ярлык будет создан и отобразится на рабочем столе в виде, представленном на рис. 1.9. Теперь запуск Access может быть выполнен двойным щелчком мыши на этом ярлыке.



Рис. 1.9. Ярлык для запуска Access

ЗАМЕЧАНИЕ

Можно запустить Microsoft Access, щелкнув двойным щелчком на файле ранее созданной базы данных Access. При этом сразу будет открыта база данных. Для запуска базы данных, так же как для Access, может быть создан ярлык.

После запуска Access 2016 отображается его стартовое окно (см. рис. 1.3). Как уже отмечалось ранее, на стартовой странице можно создать новую базу данных или открыть существующую. Если ваш компьютер подключен к Интернету, можно создать базу с помощью одного из шаблонов. Таким образом, эта страница обеспечивает быстрый доступ к средствам, позволяющим начать работу с базой данных, в том числе с помощью профессионально разработанных шаблонов.

В левой части стартового окна представлен список последних использованных баз данных и имеется команда **Открыть другие Файлы (Open Other File)**. Справа, в области создания баз данных, предоставлена возможность создать новую базу данных — кнопками **Пустая база данных рабочего стола (Blank desktop database)** и **Пользовательское веб-приложение (Custom web app)**.

При нажатии кнопки **Пустая база данных рабочего стола** создается файл локальной базы данных с указанным именем и в предлагаемой по умолчанию папке. Очевидно, что и имя, и место размещения файла могут быть здесь же изменены.

Открыть существующую базу данных можно, выбрав в левой части окна представления соответствующую команду. Быстро открыть одну из баз данных, с которыми пользователь работал недавно, можно выбрав ее из списка **Последние**. Число последних баз данных для быстрого доступа устанавливается соответствующим параметром на вкладке **Параметры клиента (Client Settings)** (по умолчанию — 4) в окне **Параметры Access (Access Options)** в группе **Вывод на экран (Display)**. Там же устанавливается число баз данных в списке последних файлов (по умолчанию — 25).

В любой момент работы с базой данных щелчком на цветном значке **Файл (File)** открывается страница с командами, относящимися, прежде всего, к базе данных в целом. При открытой базе данных она содержит такие команды, как **Сохранить (Save)** базу данных, **Сохранить как (Save As)** — сохранить копию базы данных в текущем формате или формате другой версии Access, **Заккрыть (Close)** базу данных. В разделе **Сведения (Info)** можно **Сжать и восстановить базу данных (Compact & Repair)**, **Зашифровать с использованием пароля (Encrypt with**

Password), здесь же для ненадежных баз данных выводится **Предупреждение системы безопасности (Security Warning)** (рис. 1.10), а также представлены команды, позволяющие сохранить текущий объект базы данных как новый объект — например, таблицу как форму, выполнить **Печать (Print)** текущего объекта базы данных.

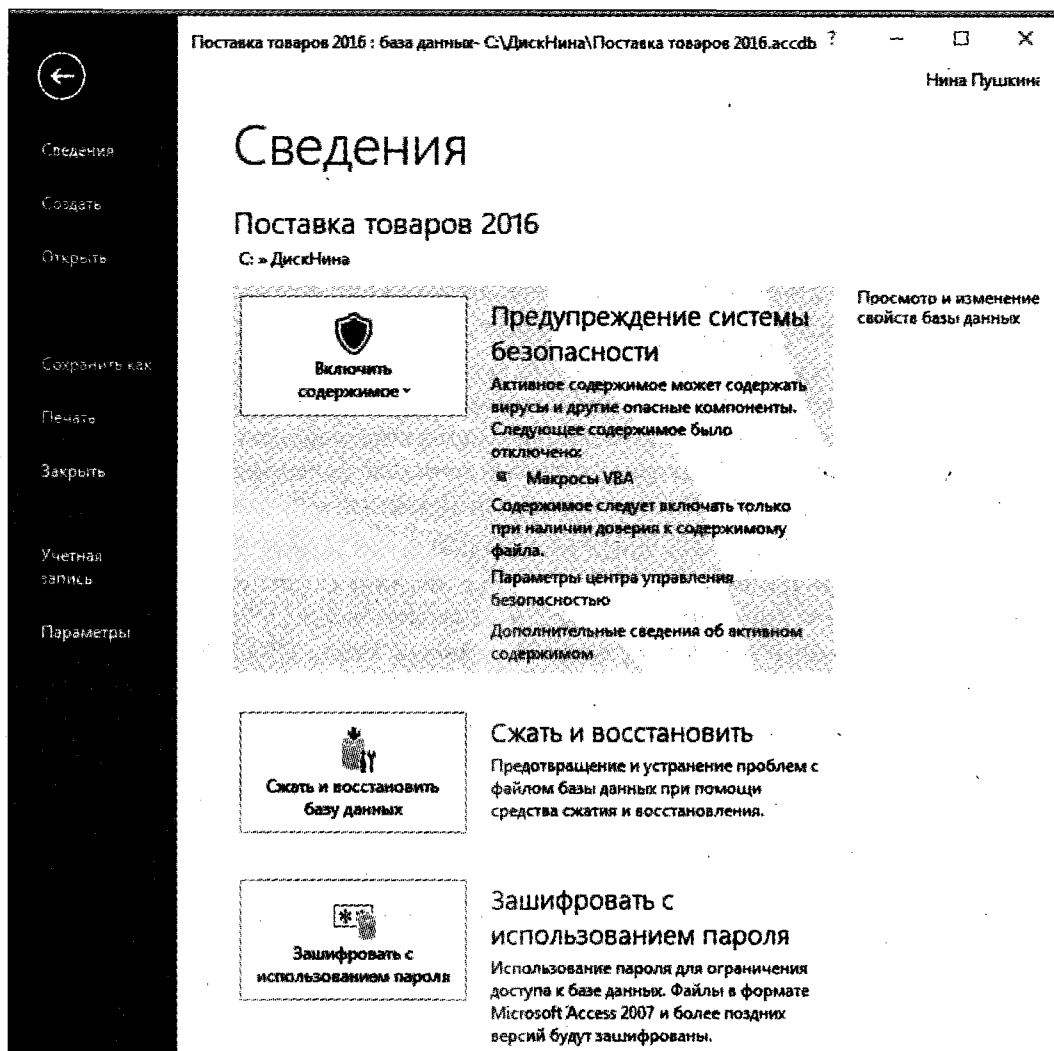


Рис. 1.10. Окно, открываемое щелчком на значке **Файл**

На странице **Параметры** можно выполнить настройку основных параметров Access, параметров текущей базы данных, безопасности документов и компьютера, открыть справку из Интернета (значок вопроса в заголовке окна).

Чтобы при запуске Access вместо отображения стартовой страницы открывалась последняя использованная база данных в окне **Параметры Access (Access Options)**, на вкладке **Параметры клиента (Client Settings)** в разделе **Дополнительно**

(Advanced) выберите **Открывать последнюю использовавшуюся базу данных при запуске Access** (Open last used database when Access starts).

Интерфейс пользователя Access

Для Access разработан интерфейс пользователя, упрощающий доступ к многочисленным функциональным возможностям в процессе создания и работы с объектами базы данных и приложения пользователя.

Основу этого интерфейса составляют *ленты* и *область навигации*. Собранные на одной ленте команды четко соответствуют выполняемым задачам, что позволяет легко находить нужную команду. Интерфейс обеспечивает доступ практически к тысяче команд, но на экране отображаются только те, которые имеют отношение к задаче, выполняемой в текущий момент.

Основные элементы интерфейса пользователя в Access 2016:

- *страницы*, предназначенные для управления файлами баз данных. Стартовая страница отображается при запуске Access и позволяет открыть существующие файлы баз данных или создать новые. В процессе работы доступны страницы, открываемые при щелчке на цветном значке **Файл** (File). Они содержат команды для сохранения, сжатия и восстановления базы данных, определения параметров и ряд других;
- *лента* — широкая полоса, расположенная в верхней части окна Access. Она содержит стандартные вкладки с группами наиболее часто используемых команд, контекстные вкладки, которые появляются только тогда, когда их использование допустимо, и *панель быстрого доступа* — небольшую панель инструментов, на которую можно добавить самые нужные команды.
Контекстные вкладки отображаются наряду со стандартными вкладками в зависимости от того, с каким объектом работает пользователь и какие действия он выполняет. В каждом режиме Access отображает свои вкладки и, соответственно, свой набор команд, представляющий всю его функциональность. Для каждой команды при наведении на ее значок мыши отображаются сведения о ее назначении, а для некоторых команд даже короткая инструкция по ее использованию.
Лента является основой интерфейса пользователя и обеспечивает быстрый доступ к набору команд, применимых к выполняемым в базе данных в текущий момент работам;
- *коллекция (галерея)* — элемент интерфейса, который не просто отображает команды, а показывает набор результатов выполнения этих команд с отображением внешнего вида вариантов выбора;
- *диалоговые окна* могут выводиться при выполнении команд для уточнения операции и передачи параметров. В некоторых группах вкладок ленты имеются кнопки вызова диалоговых окон;
- *контекстное меню* вызывается щелчком правой кнопкой мыши на элементе объекта. Содержит команды, зависящие от контекста — элемента объекта, с которым работает пользователь, или выполняемой задачи;

- *панель быстрого доступа* — единственная панель инструментов, предусмотренная в интерфейсе. Она обеспечивает доступ одним нажатием кнопки к наиболее часто используемым командам. Эта панель настраивается в соответствии с предпочтениями пользователя;
- *область навигации* расположена в левой части окна Access. В ней отображаются объединенные в группы объекты базы данных;
- *вкладки документов* — таблицы, запросы, формы, отчеты и макросы отображаются на вкладках в рабочем пространстве окна Access — окне документов. Этот режим работы по умолчанию используется для всех баз данных, созданных с помощью Access 2016. Для того чтобы открывать каждый объект в отдельном управляемом окне — как в версиях до Access 2007 — следует выбрать параметр окна документов **Перекрытие окон (Overlapping Windows)**;
- *строка состояния* — полоса в нижней части окна программы, в которой отображаются сведения о состоянии объекта и располагаются кнопки, позволяющие изменить режим его представления;
- *мини-панель инструментов* — прозрачный элемент, подключенный к объекту, который появляется над выбранным текстом и позволяет легко отформатировать его;
- *панель сообщений* — это единое средство вывода всех предупреждений системы безопасности. Отображается, когда в открываемой базе данных имеется любое потенциально опасное выполняемое содержимое.

Страницы

Сразу после запуска Access отображается стартовая страница, на которой можно открывать (левая панель окна) или создавать новые файлы баз данных (правая панель окна). На левой панели доступен список недавно использованных последних баз данных, который позволяет быстро выполнить открытие одной из них, и команда **Открыть другие файлы (Open Other Files)**. Всплывающая подсказка каждой базы данных из списка последних отображает местоположение ее файла — полный путь к нему. На правой панели можно создать новую (пустую) базу данных рабочего стола или **Пользовательское веб-приложение (Custom web app)**. Здесь же при доступном Интернете представлены многочисленные шаблоны для создания различных типовых баз данных.

Как уже отмечалось в *разд. «Запуск Access»*, для открытой базы данных при выборе значка **Файл** отображается страница, в левой части которой содержится список команд, применимых ко всей базе данных, — таких как **Сохранить (Save)**, **Сохранить как (Save As)**, **Закреть (Close)** базу данных. Здесь же представлены команды, позволяющие сохранить текущий объект базы данных как новый объект — например, таблицу как форму, выполнить **Печать (Print)** текущего объекта базы данных. Справа в этот момент открыт раздел **Сведения (Info)** с командами: **Сжать и восстановить базу данных (Compact & Repair)**, **Зашифровать с использованием пароля (Encrypt with Password)**, а также с предупреждением системы безопасности для ненадежных баз данных (см. рис. 1.10).

Лента

При открытии базы данных в Access появляется т. н. *лента*, представляющая собой основной элемент интерфейса программы (рис. 1.11). При этом на ней отображаются четыре стандартные *вкладки*: **Главная** (Home), **Создание** (Create), **Внешние данные** (External Data) и **Работа с базами данных** (Database Tools). Вкладки включают логически связанные команды. Однотипные команды вкладок объединены в *группы*. Название группы представлено под набором составляющих ее команд.

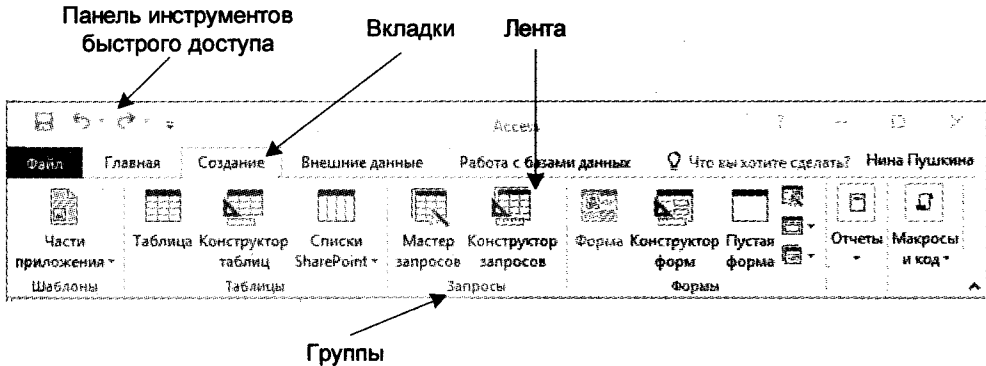


Рис. 1.11. Лента с открытой вкладкой **Создание**

Название команды на вкладке и ее описание можно увидеть во всплывающей подсказке, которая отобразится, если задержать на кнопке указатель мыши. Устанавливая курсор на кнопках, с помощью всплывающих подсказок можно найти нужную операцию. Для того чтобы начать ее выполнение, достаточно щелкнуть на кнопке мышью.

Набор вкладок изменяется при переходе к другому объекту или режиму работы. Например, если открыть таблицу базы данных, на ленте рядом со стандартными вкладками появится контекстная вкладка **Работа с таблицами** (Table Tools) с двумя вкладками второго уровня: **Поля** (Fields) и **Таблица** (Datasheet). Команды стандартных вкладок по-прежнему остаются доступными. Например, щелкнув на команде **Форма** (Form) вкладки **Создание** (Create), вы создадите форму, источником записей которой будет выбрана открытая таблица.

Access, манипулируя набором встроенных вкладок, обеспечивает удобный интерфейс пользователя при выполнении работ с каждым из объектов базы данных. Эти вкладки выводятся в окне Access по умолчанию в соответствии с текущим режимом работы объекта.

Коллекция

Коллекция — это элемент интерфейса, который отображает группу команд, показывая результат выполнения каждой из них. Смысл состоит в том, чтобы предоставить пользователю возможность по внешнему виду найти и выбрать нужные действия, сосредоточившись на результате, а не на поиске команды.

Коллекции различаются по форме и размерам. Примеры коллекций можно видеть на вкладках конструктора форм и отчетов. На рис. 1.12 приведена коллекция кнопки **Темы** (Themes).



Рис. 1.12. Пример коллекции на вкладке конструктора при работе с макетом формы или отчета

Дополнительную более тонкую настройку внешнего вида формы или отчета можно выполнить, воспользовавшись командами вкладок **Формат** (Format) и **Упорядочить** (Arrange).

Диалоговые окна

Диалоговые окна выводятся при выполнении некоторых команд для уточнения операции и передачи параметров. При открытом диалоговом окне нельзя перейти в том же приложении к выполнению других действий. Диалоговое окно имеет постоянные размеры, но может быть перемещено в другое место.

В некоторых группах вкладок ленты в правом нижнем углу группы имеются маленькие значки — кнопки вызова диалоговых окон. Нажатие такой кнопки открывает соответствующее диалоговое окно или область задач, предоставляя дополнительные возможности, относящиеся к этой группе. Пример диалогового окна для выбора дополнительных параметров форматирования открытой таблицы базы данных на вкладке **Главная** (Home) приведен на рис. 1.13.

Пример области задач для отображения содержимого буфера обмена, открывающейся кнопкой в группе **Буфер обмена** (Clipboard) на вкладке **Главная** (Home), приведен на рис. 1.14.

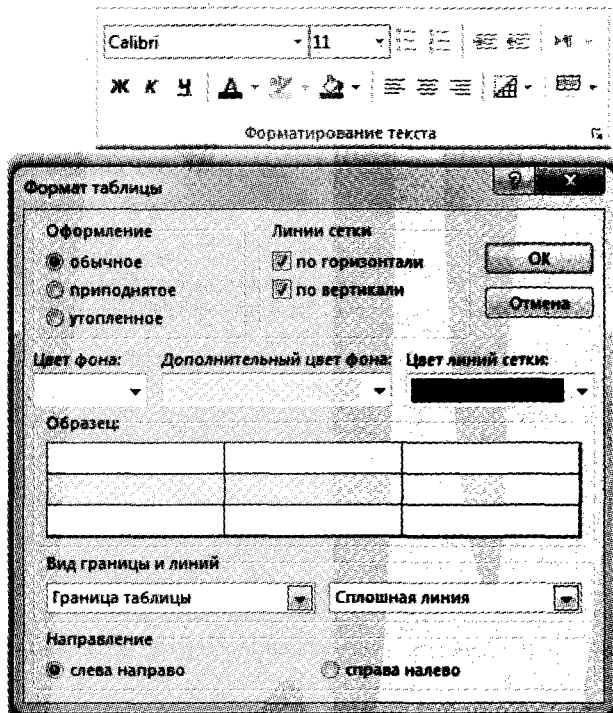


Рис. 1.13. Диалоговое окно Формат таблицы

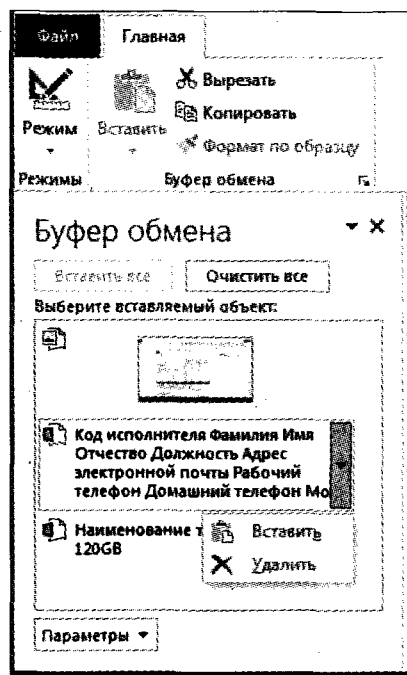


Рис. 1.14. Область задач Буфер обмена

Контекстное меню

В Access практически каждый элемент объекта в любом режиме снабжен динамическим контекстным меню, вызываемым правой кнопкой мыши и содержащим набор команд, применимых в текущей ситуации. Эти меню включают большое число команд, не доступных в других элементах интерфейса Access. Например, команды открытия существующего объекта (таблицы, формы, отчета) в режиме конструктора, режиме таблицы или макета представлены только в контекстном меню этого объекта, выбранного в области навигации, представляющей все объекты базы данных.

Пользовательская панель инструментов быстрого доступа

Пользовательская панель инструментов быстрого доступа (Customize Quick Access Toolbar) первоначально включает лишь кнопки **Сохранить** (Save), **Отменить** (Undo), **Вернуть** (Redo) и кнопку **Настройка панели быстрого доступа** (Customize Quick Access Toolbar), открывающую список команд, которыми может быть дополнена панель (рис. 1.15).

Команды списка **Разместить над лентой** (Place Quick Access Toolbar above the Ribbon) или **Разместить под лентой** (Place Quick Access Toolbar below the Ribbon)

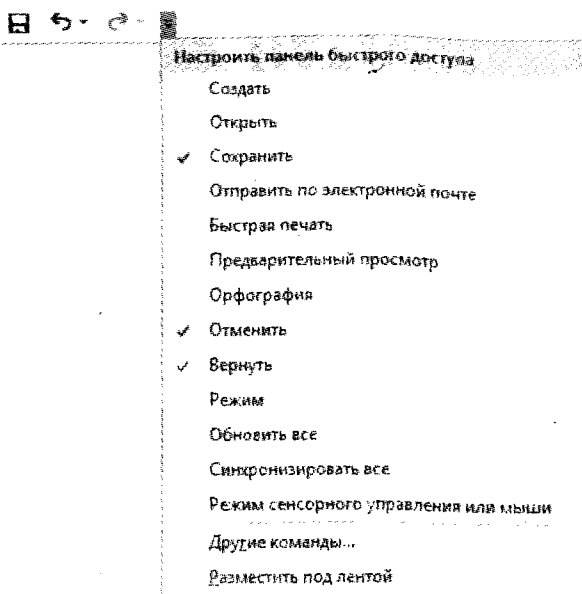


Рис. 1.15. Список команд кнопки **Настройка панели быстрого доступа**

позволяют изменить местоположение этой панели, разместив ее ниже или выше ленты команд.

Строка списка **Другие команды** (More Commands) открывает окно **Параметры Access** (Access Options). В этом окне (рис. 1.16) пользователь может по своему усмотрению дополнить панель быстрого доступа командами, выбирая их из полного списка команд Access. Команды списка разбиты по категориям. Отдельно представлена категория **Часто используемые команды** (Popular Commands). Выбор категории команд, из которой вы собираетесь добавить команды, выполняется в поле **Выбрать команды из** (Choose commands from) в левой части окна.

Для добавления или удаления выбранной команды предназначены кнопки **Добавить** (Add) и **Удалить** (Remove). Для упорядочения кнопок на панели инструментов быстрого доступа воспользуйтесь кнопками перемещения **Вверх** (Move Up) и **Вниз** (Move Down). Настройка может быть выполнена как для текущей базы данных, так и для всех баз данных. Этот выбор осуществляется в списке **Настройка панели быстрого доступа** (Customize Quick Access Toolbar) в правой части окна.

Добавить команду на пользовательскую панель инструментов быстрого доступа можно, не открывая окна **Параметры Access** (Access Options). Контекстное меню *любой кнопки на ленте* включает команду **Добавить на панель быстрого доступа** (Add to Quick Access Toolbar). Контекстное меню *любой кнопки на панели инструментов быстрого доступа* содержит команду **Удалить с панели быстрого доступа** (Remove from Quick Access Toolbar). Эти команды позволяют в оперативном режиме менять состав кнопок, настраивая панель на текущие потребности пользователя.

Используя контекстное меню, на панель инструментов быстрого доступа можно добавить (удалить) не только отдельную команду ленты, но и целую коллекцию, которая представляется одним значком.

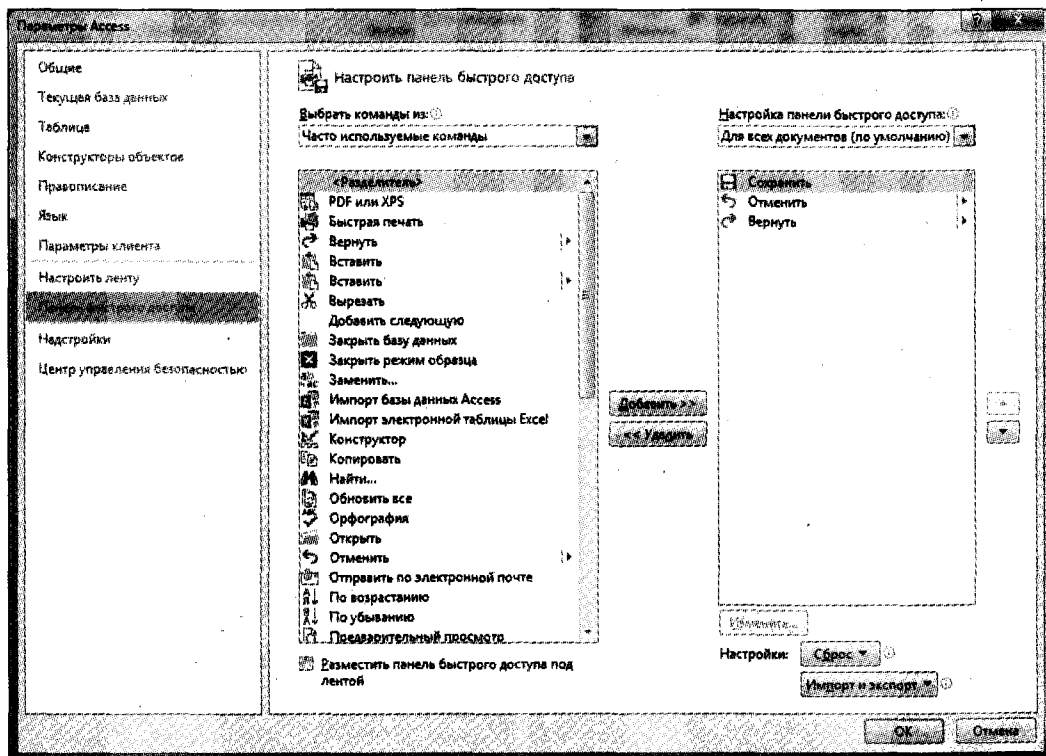


Рис. 1.16. Окно настройки пользовательской панели инструментов быстрого доступа

При последующих открытиях Access добавленные на панель инструментов кнопки будут сохранены.

Вывод команд на панель позволяет выполнить их быстрым и удобным способом из любого окна Access. Однако состав кнопок панели является контекстно-независимым, и поэтому на панели целесообразно размещать только те команды, которые могут быть выполнены в любой момент или используются достаточно часто.

Область навигации

Объекты открытой базы данных представлены в разделе **Область навигации** (Display Navigation Pane). При открытии базы данных все ее объекты: таблицы, формы, отчеты, запросы, макросы и модули — отображаются в области навигации в соответствующих группах. Область навигации обеспечивает быстрый доступ к нужным объектам, которые для удобства пользователя могут быть объединены не только в стандартных категориях и группах, но и в созданных пользователем в соответствии с функциональностью приложения. Состав отображаемых в области навигации объектов может меняться. Можно вывести только нужные категории, отфильтровать группы объектов, выбрать объекты по датам создания или изменения. Для представления в области навигации объектов по группам щелкните на кнопке списка в заголовке области навигации и выберите категорию **Тип объекта** (Object Type) и фильтр **Все объекты Access** (All Access Objects) (рис. 1.17).

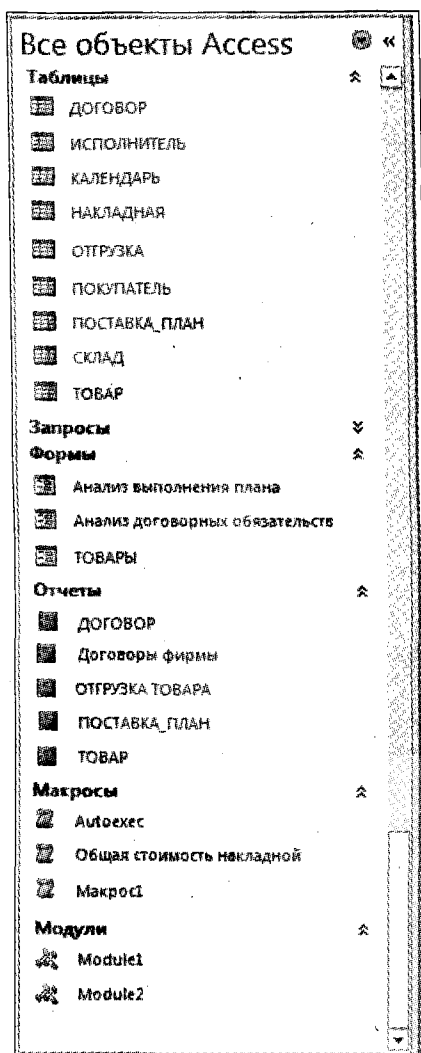



Рис. 1.17. Область навигации с группами объектов

Область навигации используется для того, чтобы открыть объект для работы с ним или для изменения его структуры в режиме конструирования или макета. Открытие объекта может быть выполнено двойным щелчком на нем или с помощью контекстного меню, в котором представлены основные режимы работы с объектом. Только для модуля при двойном щелчке запускается редактор Visual Basic, а не выполнение модуля. Кроме того, выбранный в области навигации объект можно использовать для экспорта в Excel, в RTF-файлы, файлы XML, списки SharePoint, в базы данных различных приложений. Можно также отправить объект по электронной почте в открывающемся окне Microsoft Office Outlook.

Если не нужно просматривать сразу все объекты базы данных, можно создать настраиваемые пользователем категории и группы объектов базы данных и отобразить в области навигации только эти категории и группы. Поскольку в таких груп-

пах доступны, например, только требуемые формы и отчеты и скрыты остальные объекты базы данных, эти группы можно использовать вместо кнопочных форм.

Область навигации всегда остается видимой на экране и не может быть закрыта другими окнами. Область навигации можно убрать с экрана, освободив место для выполнения других работ, например, конструирования формы. Для этого предназначена кнопка  **Открыть/закрыть границу области переходов (Shutter Bar Open/Close Button)**, размещенная в правом верхнем углу заголовка.

Щелкните на кнопке списка в заголовке области навигации. На открывшейся панели будут отображены объекты в области навигации, организованные по категориям (рис. 1.18). Раздел **Переход в категорию (Navigate To Category)** отображает стандартные категории **Тип объекта (Object Type)**, **Таблицы и связанные представления (Tables and Related Views)**, **Дата создания (Created Date)**, **Дата изменения (Modified Date)**.

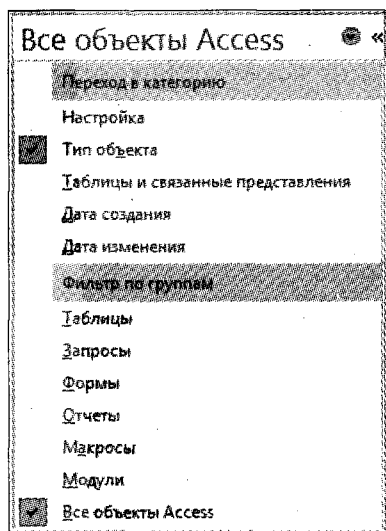


Рис. 1.18. Панель выбора категории и группы объектов для отображения в области навигации

Выбор категории определяет список доступных фильтров. Для категории **Тип объекта (Object Type)** в разделе **Фильтр по группам (Filter By Group)** перечисляются все доступные типы объектов.

При выборе категории **Таблицы и связанные представления (Tables and Related Views)** в разделе **Фильтр по группам (Filter By Group)** открывается возможность выбора таблицы, для которой нужно просмотреть все запросы, в которых она использована. Заметим, что в список представлений включаются не только запросы, но и формы и отчеты, в основе которых лежат запросы, в том числе созданные при их разработке.

В соответствии с названиями категорий может быть произведен отбор объектов по времени создания или изменения.

Чтобы по умолчанию запретить отображение области навигации при открытии базы данных, следует для текущей базы данных в окне параметров Access в разделе **Навигация (Navigation)** снять флажок **Область навигации (Display Navigation Pane)**. Скрывать область навигации целесообразно только при наличии формы или другого метода для запуска объектов в базе данных.

Вкладки документов

Окно документов Access работает с моделью пользовательского интерфейса, называемой *моделью однодокументного интерфейса (SDI)*. Эта модель размещает любой объект базы данных: форму, отчет и т. д. в полном окне документов и образует для каждого объекта свою вкладку. Если открыто несколько объектов, для переключения между этими объектами используются ярлычки их вкладок (рис. 1.19).

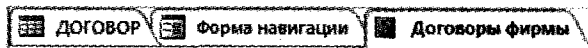


Рис. 1.19. Вкладки окна документов Access

Использование вкладок — это режим работы по умолчанию для всех созданных в Access 2016 баз данных. Для того чтобы открывать каждый объект в отдельном окне, в окне **Параметры Access (Access Options)** выберите **Текущая база данных (Current Database)**, в разделе **Параметры приложений (Application Options)** в группе **Параметры окна документа (Document Window Options)** установите флажок **Перекрытие окон (Overlapping Windows)** вместо **Вкладки документов (Tabbed Documents)**.

Строка состояния

Вдоль нижней границы окна Access может отображаться строка состояния. Эта строка предназначена для вывода текста сообщения, связанного с текущим режимом, состоянием объекта, выполняющейся программой.

Элементы управления в правой части строки состояния позволяют быстро переключать различные режимы активного объекта (рис. 1.20). При просмотре объекта, который поддерживает изменение масштаба (например, отчета в режиме предварительного просмотра), можно регулировать степень увеличения или уменьшения с помощью ползунка в строке состояния.



Рис. 1.20. Строка состояния при просмотре объекта, поддерживающего изменение масштаба

Включить или отключить отображение строки состояния можно в диалоговом окне **Параметры Access (Access Options)**. Для ее отключения в левой области этого окна выберите **Текущая база данных (Current Database)** и в открывшемся окне в разделе **Параметры приложений (Application Options)** снимите флажок **Строка состояния (Status bar)**.

Панель сообщений

В Access 2016, как и других приложениях Microsoft Office этого выпуска, вывод всех предупреждающих сообщений системы безопасности производится не в отдельных окнах, а на панели сообщений. По умолчанию уже при открытии базы данных, если точно неизвестно, что можно доверять ее содержимому, центр управления безопасностью отключает в открываемой базе данных такие потенциально опасные, активные компоненты, как запросы на изменение (запросы, которые добавляют, удаляют или изменяют данные), макросы, элементы управления ActiveX, выражения (функции, возвращающие одно значение), программы на VBA, и выводит предупреждение об этом на панели сообщений (рис. 1.21).

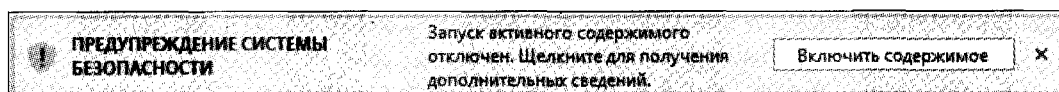


Рис. 1.21. Предупреждающее сообщение системы безопасности при открытии базы данных

Если вы доверяете содержимому базы данных, включите отключенное содержимое, щелкнув на кнопке **Включить содержимое** (Enable this content). Отключенное содержимое будет включено, и база данных откроется с полным набором функциональных возможностей. В противном случае отключенные компоненты не будут работать.

Технология Drag and Drop

Реализованная в Access технология *Drag and Drop* позволяет пользователю переносить объекты базы данных и их элементы с помощью мыши. Например, любую таблицу или запрос можно перенести из области навигации в окно схемы данных. Для установления связи между объектами можно в схеме данных переместить поле из одной таблицы в другую. Для размещения подчиненной формы в главной достаточно перенести в нее ранее созданную форму или даже просто таблицу.

С помощью мыши можно переносить объекты между различными базами данных. При этом необходимо запустить обе задачи Microsoft Access. Возможен перенос таблиц и запросов Access в другие приложения — например, в Microsoft Word и Microsoft Excel. Можно выделить нужные данные в форме или в объекте в режиме таблицы и перенести только их. Можно создать таблицу путем переноса с помощью мыши диапазона ячеек Microsoft Excel в область навигации базы данных Access. Объекты других приложений могут быть перенесены в поле объекта OLE в таблице или форме в режиме формы, а также в форму или отчет в режиме конструктора.

Справка Access

Справочная система Access позволяет получить недостающие сведения и ответы на возникающие в ходе работы вопросы. Отдельное окно справки **Справка Access** (Access Help) открывается щелчком мыши на значке вопроса в заголовке окна

Access. Справочная система требует подключения к Интернету — в автономном режиме она работать не может.

При открытии справки отображается страница со ссылками на разделы, представляющие основы работы с объектами Access. Справка обеспечивает поиск по ключевым словам и подготавливает список гиперссылок на разделы, отвечающие на введенный запрос. Ввод дополнительных ключевых слов сужает список найденных разделов. Ответы представлены в списке в порядке релевантности, т. е. первым отображается наиболее подходящий ответ на вопрос. Раздел, открытый в окне справки, может быть распечатан с помощью соответствующей кнопки в окне справки. Окно справки имеет кнопки перехода вперед и назад по просмотренным разделам.

Для получения справки о применении доступных на ленте команд Access подведите указатель мыши к команде — появится всплывающая подсказка со сведениями о том, какие действия выполняет эта команда. Если наряду с этими сведениями в подсказке отображается ссылка ? **Дополнительные сведения** (Tell me more), можно по ней перейти к справке.

Для получения справки в диалоговом окне, например, многостраничном окне **Параметры Access** (Access Options), используйте значок вопроса в его заголовке. Он открывает в окне справки раздел с описанием параметров, представленных на странице. Для получения таких справок также требуется подключение к Интернету.

На ленте Access 2016 наряду с вкладками появилось текстовое поле **Что вы хотите сделать?** (Tell me what you want to do ...). Ввод в это поле слова и фразы, описывающих ваши намерения, обеспечит вывод списка соответствующих команд. Поле может быть использовано как для поиска подходящей для выполнения некоторой функции команды, так и для поиска известной команды. Кроме того, выводится строка, обеспечивающая переход к получению справки по интересующей теме. Введенные слова используются при поиске как ключевые. При уменьшении окна Access поле **Что вы хотите сделать?** получает имя **Помощник**.

ПРИМЕЧАНИЕ

Поле **Что вы хотите сделать?** (Tell me what you want to do ...) отсутствует на ленте при разработке веб-приложений Access.

Непосредственное обращение к сайту **www.office.com** даст возможность познакомиться с последними нововведениями Microsoft Office, получить доступ к статьям, советам, шаблонам, интерактивным средствам обучения, файлам для загрузки и службам. Предлагаемые курсы обучения включают в себя изучение возможностей продуктов Office. Небольшие видеокурсы по 20–50 минут для самостоятельного обучения со звуком, графическими и анимационными возможностями позволяют приобрести практические навыки.

На сайте **www.office.com** доступны для загрузки многочисленные технические файлы, в том числе драйверы устройств, пакеты обновлений продуктов, файлы с исправлениями, а также полные продукты.

Сайт регулярно обновляется, а содержание узла определяется на основе обратной связи и определенных запросов пользователей Office.

Защита баз данных

Защита баз данных в Access 2016 построена на доверии к происхождению ее выполнимого содержимого. Дело в том, что некоторые компоненты Access, полученные из ненадежных источников, выполняясь, могут повредить базу данных. Поэтому база данных открывается для работы с полным набором функциональных возможностей только, если она имеет состояние доверенной.

База данных Access хотя и хранится в одном файле, представляет собой набор объектов: таблиц, форм, запросов, отчетов, макросов и модулей, которые, как правило, являются взаимозависимыми и, следовательно, выполнение некоторых объектов может повлечь за собой порчу других.

К небезопасным объектам относятся:

- запросы на изменение, которые добавляют, удаляют или изменяют данные в таблицах;
- макрокоманды, изменяющие базу данных или получающие доступ к ресурсам вне базы данных;
- запросы (DDL-запросы), предназначенные для создания или изменения объектов базы данных — таких как таблицы и процедуры;
- SQL-запросы к серверу, которые передаются на сервер баз данных и работают с таблицами непосредственно на нем, минуя ядро базы данных Access;
- элементы управления ActiveX;
- выражения (функции, возвращающие одно значение);
- код VBA.

База данных приобретет состояние доверенной, если помещена в надежное расположение либо имеет цифровую подпись от сертифицированного издателя. *Издатель* — это разработчик макроса, элемента управления ActiveX, надстройки или других расширений приложения, используемых потребителями. *Цифровая подпись* — это электронная зашифрованная печать, удостоверяющая подлинность цифровых данных — таких как сообщения электронной почты, макросы или электронные документы. Подпись подтверждает, что выполняемые компоненты базы данных созданы издателем, и никто другой не изменял их после подписи.

Защита баз данных в Access 2016 реализуется центром управления безопасностью. Центр управления безопасностью отображает диалоговое окно, в котором можно задавать и менять параметры безопасности и конфиденциальности, в том числе определять надежное расположение, надежных издателей. Чтобы открыть это окно, воспользуйтесь кнопкой **Параметры (Options)** на странице, открываемой при выборе значка **Файл (File)**, затем в окне **Параметры Access** откройте группу **Центр управления безопасностью (Trust Center)** и нажмите кнопку **Параметры центра управления безопасностью (Trust Center Settings)**. Параметры, установленные в центре управления безопасностью, определяют поведение новых и существующих баз данных при их открытии в Access. Для установки параметров безопасности открывать базу данных не требуется.

Для создания надежного расположения и добавления туда папки с базами данных выберите **Надежные расположения (Trusted Locations)**, нажмите кнопку **Добавить новое расположение (Add new Location)**, а затем в диалоговом окне **Надежное расположение Microsoft Office (Microsoft Office Trusted Location)** укажите путь к папке надежного расположения на жестком диске или в сети (рис. 1.22). Для перемещения или копирования файла базы данных в надежное расположение используются любые привычные способы — например, средства Проводника Windows. При открытии базы данных, размещенной в надежном расположении, центр управления безопасностью не выполняет ее проверку, не выводит предупреждение системы безопасности и не отключает выполнимое содержимое.

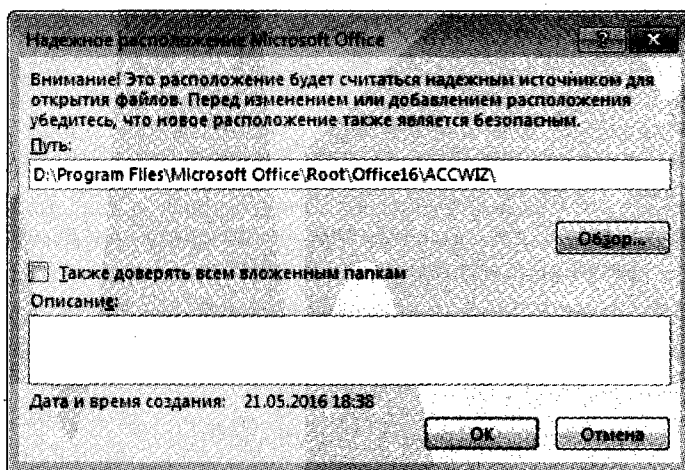


Рис. 1.22. Определение папки в качестве надежного источника для открытия файлов баз данных

Назначение в качестве надежного расположения всей папки Мои документы увеличивает угрозу безопасности и повышает риск атаки хакеров. Любые внешние папки, например сетевые, также не являются безопасными, поэтому не рекомендуется назначать в качестве надежного расположения папку общего пользования сетевого ресурса. При создании надежного расположения, не являющегося локальным ресурсом компьютера, в диалоговом окне **Надежное расположение (Trusted Locations)** нужно установить **Разрешить надежные расположения в моей сети (не рекомендуется) (Allow Trusted Locations on my network (not recommended))**, напоминающее о ненадежности расположения.

Для применения цифровой подписи (шифрованной электронной подписи) к компонентам базы данных, прежде всего, необходимо иметь цифровой сертификат. Если базы данных создаются для коммерческого распространения, нужно получить сертификат в коммерческом центре сертификации, например, в Symantec SSL (ранее VeriSign). Центр сертификации наводит справки об издателе, чтобы удостовериться в его надежности.

Если базу данных планируется использовать в личных целях или в небольшой рабочей группе, можно воспользоваться предусмотренным в операционных системах

Microsoft Windows средством создания сертификатов с собственной подписью, называемым SelfCert.exe. Этот сертификат следует добавить в список надежных источников, а затем подписать базу данных.

Для создания сертификата нажмите кнопку **Пуск** и выберите последовательно пункты **Все программы | Microsoft Office | Средства Microsoft Office | Цифровой сертификат для проектов VBA**. Откроется диалоговое окно **Создание цифрового сертификата**. В поле **Имя сертификата** введите описательное имя сертификата. Нажмите кнопку **ОК**. При появлении сообщения **SelfCert: успех** нажмите кнопку **ОК**.

Для просмотра сертификата в хранилище личных сертификатов откройте браузер Internet Explorer. Выберите в меню **Сервис** команду **Свойства обозревателя** и откройте вкладку **Содержание**. Нажмите кнопку **Сертификаты** и откройте вкладку **Личные**.

При открытии файлов ACCDB или ACCDE Access 2016 сообщает расположение базы данных центру управления безопасностью. Если это расположение надежное, она работает с полным набором функциональных возможностей. При открытии базы данных из более ранней версии Access передает в центр управления безопасностью расположение и цифровую подпись, если она имеется в базе данных.

Центр управления безопасностью проверяет подлинность этих сведений, чтобы определить, имеет ли база данных состояние доверенной, а затем информирует приложение Access о том, как следует ее открывать. Access либо отключает опасные компоненты базы данных, либо открывает с полным набором функциональных возможностей.

Если база данных не имеет состояния доверенной, программные средства центра управления безопасностью по умолчанию, не оценивая компоненты базы данных, отключают все выполняемое содержимое в базе данных и запрашивают у пользователя разрешение на его включение, отображая панель сообщений (см. рис. 1.21).

Если пользователь доверяет источнику или берет на себя ответственность за безопасность базы данных, он разрешает использование опасного содержимого, щелкнув на кнопке **Включить содержимое** (Enable this content). При этом источник автоматически попадает в разряд надежных, и при следующем открытии панель сообщений выводиться уже не будет. Для получения дополнительных сведений щелкните на соответствующей подписи на панели сообщений. На открывшейся странице **Сведений** (Info) о базе данных будет отображено **Предупреждение системы безопасности** (Security Warning). Здесь можно **Включить содержимое** (Enable this content), перейти к работе с параметрами центра управления безопасностью или получить дополнительные сведения об активном содержимом в окне справки (рис. 1.23). В списке кнопки **Включить содержимое** первое значение списка переводит базу данных в разряд надежных, второе позволяет установить защиту от неизвестного содержимого или разрешить его выполнение на время текущего сеанса работы. При повторном открытии такой базы данных будет выводиться запрос по поводу отключения опасного содержимого.

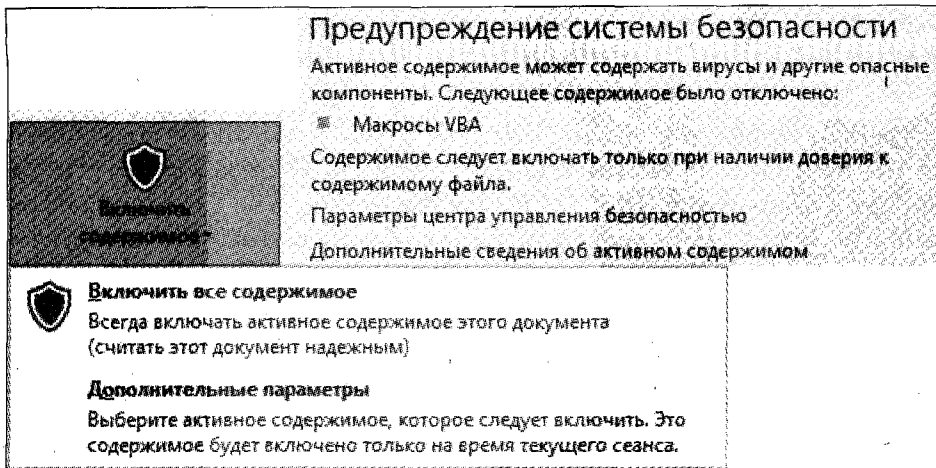


Рис. 1.23. Предупреждение системы безопасности при открытии файла базы данных

При выборе в диалоговом окне центра управления безопасностью вкладки **Надежное расположение** (Trusted Locations) и варианта **Отключить все надежные расположения** (Disable all Trusted Locations) Access отключит все надежные расположения.

Панель сообщений Access является единым средством вывода предупреждений, которое появляется по умолчанию уже при открытии базы данных вне доверенного расположения. Таким образом ограничивается число предупреждающих сообщений, с которыми имеют дело пользователи.

Для включения панели сообщений щелчком на значке **Файл** (File) откройте соответствующую страницу. Выберите **Параметры** (Options). На левой панели диалогового окна **Параметры Access** (Access Options) выберите **Центр управления безопасностью** (Trust Center). На открывшейся вкладке щелкните на кнопке **Параметры центра управления безопасностью** (Trust Center Settings).

В окне **Центра управления безопасностью** (Trust Center) выберите элемент **Панель сообщений** (Message Bar). Установите флажок **Показывать панель сообщений во всех приложениях, если активное содержимое, такое как элементы ActiveX и макросы, заблокированы** (Show the Message Bar in all applications when active content, such as ActiveX controls and macros, has been blocked) и нажмите кнопку **ОК**.

После включения и отображения панели сообщений на ней можно включить содержимое.

Другим средством защиты является *шифрование* баз данных. Это средство объединяет кодирование и пароли баз данных. Чтобы зашифровать базу данных, достаточно установить пароль для шифрования базы данных. Стойкий алгоритм шифрования баз данных в формате Access 2007/2016 исключает несанкционированный их просмотр. Все данные становятся нечитаемыми в других программных средствах, и для того чтобы использовать эту базу данных, пользователи должны вводить пароль.

Шифрование с использованием пароля выполняется для базы данных, открытой в монопольном режиме. Для открытия базы в этом режиме щелкните на значке **Файл (File)** и на открывшейся странице выберите команду **Открыть (Open)**. Далее на странице **Открыть (Open)** выберите **Обзор (Browse)**. В окне **Открытие файла базы данных (Open)** найдите папку, в которой хранится файл, выделите файл, который нужно открыть, щелкните стрелку рядом с кнопкой **Открыть (Open)** и выберите **Монопольно (Open Exclusive)**.

Для шифрования базы данных с использованием пароля на странице, открывающейся выбором значка **Файл (File)**, нажмите кнопку **Сведения (Info)** и выберите **Зашифровать с использованием пароля (Encrypt with Password)**. В открывшемся диалоговом окне **Задание пароля базы данных (Set Database Password)** введите пароль и подтверждение и нажмите кнопку **ОК**. При открытии зашифрованной таким образом базы данных потребуется ввод пароля.

Выполнением команды **Расшифровать базу данных (Decrypt Database)**, в которую преобразовалась команда **Зашифровать с использованием пароля (Encrypt with Password)**, пароль базы данных удаляется. При этом также необходимо ввести пароль.

Пароли должны содержать не менее чем 8 символов, однако надежным паролем может считаться только тот пароль, который сочетает в себе прописные и строчные буквы, цифры и символы и состоит не менее, чем из 14 символов. Важно помнить свой пароль, поскольку восстановить его невозможно.

Контрольные вопросы

1. Изменяется ли набор вкладок ленты при изменении режима обработки объекта в базе данных Access?
2. Для чего предназначена панель быстрого доступа?
3. Какие вкладки ленты выводятся при открытии базы данных?
4. Какая область позволяет отображать все объекты базы?
5. Можно ли управлять набором команд на панели быстрого доступа?
6. Как получить всплывающую подсказку для команды на вкладке ленты?
7. Можно ли изменить состав кнопок на вкладках ленты?
8. Какие объекты составляют базу данных Access?
9. Чем определяется структура реляционной таблицы?
10. Для чего предназначен первичный ключ реляционной таблицы?
11. Какими отношениями характеризуется связь двух таблиц реляционной базы данных?
12. Для чего используется ключ связи?
13. Что предлагает технология интеллектуальной замены имен?

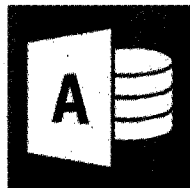
14. Выведется ли сообщение системы безопасности об отключении части функций базы данных при ее открытии из надежного расположения?
15. В файле с каким расширением сохраняется база данных Access 2016?
16. В файле с каким расширением сохраняется исполняемый файл базы данных Access 2016?
17. Можно ли для базы данных Access 2016 в окне документа открывать каждый объект в отдельном окне?
18. Для защиты данных от одновременной корректировки блокировка осуществляется на уровне страницы или записи?
19. С помощью какого интерфейса в базе данных Access 2016 обеспечивается доступ к данным Microsoft SQL Server?
20. С какой целью создаются в Access веб-приложения?
21. Какой режим используется в веб-приложении для создания таблиц?
22. С помощью каких программ пользователи получают возможность работать с веб-приложением в «облаке»?
23. Имеется ли область навигации в веб-приложении?
24. Какой сервер SharePoint можно использовать для создания веб-приложения?
25. Где хранится база данных веб-приложения?

Ответы

1. Да.
2. Для более удобного и быстрого выполнения часто используемых команд.
3. Главная (Home), Создание (Create), Внешние данные (External Data) и Работа с базами данных (Database Tools).
4. Область навигации.
5. Да.
6. Переместить на кнопку курсор.
7. Нет.
8. Таблицы, запросы, формы, отчеты, макросы и модули.
9. Составом полей.
10. Для однозначной идентификации каждой записи таблицы.
11. Отношениями записей типа «один-к-одному» (1 : 1) и «один-ко-многим» (1 : M).
12. Для связи двух таблиц с помощью одинакового поля в них.
13. Отслеживать изменение имен таблиц, полей, запросов, элементов управления и др. и выполнять их автозамену в зависимых объектах.

14. Нет.
15. accdb.
16. accde.
17. Да.
18. Блокировка может осуществляться на обоих уровнях.
19. ODBC.
20. Для публикации и совместного использования приложения базы данных Access на страницах SharePoint с помощью браузера.
21. Режим конструктора.
22. Рекомендуется использовать последнюю версию браузера Microsoft Edge. Работает с текущими версиями браузеров Internet Explorer, Safari, Chrome и Firefox.
23. Да.
24. SharePoint Server 2013 и выше, установленный в сети организации, или SharePoint Online, предоставляемый в облаке Office 365.
25. На SQL Server 2012 и выше, определенном в корпоративном SharePoint Server, или SQL Windows Azure при работе с Office 365.

ГЛАВА 2



Проектирование реляционной базы данных

Реляционная база данных, а именно такой является база данных Access, состоит из взаимосвязанных реляционных таблиц. На этапе проектирования базы данных для выбранной предметной области должна быть определена логическая структура базы данных. Проект логической структуры БД устанавливает состав реляционных таблиц, их структуру и логические связи между таблицами. При формировании структуры каждой таблицы определяется совокупность полей (столбцов), для каждого из которых определяются тип, размер данных и другие свойства. Для таблицы указывается уникальный ключ, который может состоять из одного или нескольких полей. Система выполняет проверку уникальности ключа, обеспечивая целостность данных.

При проектировании базы данных, отвечающей требованиям нормализации, между таблицами определяются логические связи с типом отношений «один-ко-многим» (1 : М). Такие связи позволят осуществлять в Access автоматическое поддержание связной целостности и непротиворечивости данных в базе.

Этапы проектирования и создания базы данных

Для проектирования базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, определять все необходимые источники информации для обеспечения предполагаемых запросов пользователя и решаемых в приложении задач.

Определение состава и структуры данных, которые должны быть загружены в базу данных, осуществляется на основе анализа предметной области. Структура данных предметной области может отображаться *информационно-логической моделью* (ИЛМ). Если при построении такой модели обеспечены требования нормализации данных и она, соответственно, представлена в каноническом виде, далее легко определяется проект логической структуры нормализованной базы данных. На основе канонической модели можно создать реляционную базу без дублирования данных.

К разработке модели данных предметной области существуют два подхода. В первом подходе (аналитическом, или процессном) сначала формулируются основные

задачи, для решения которых строится база, выявляются информационные потребности задач приложения пользователя, и, соответственно, определяются состав и структура информационных объектов модели, а также связи между ними. При втором подходе (интуитивном) сразу устанавливаются типовые информационные объекты предметной области и их взаимосвязи. Наиболее рационально сочетание обоих подходов. Это связано с тем, что на начальном этапе, как правило, нет исчерпывающих сведений обо всех задачах. Использование такой технологии тем более оправдано, что гибкие средства создания реляционной базы данных позволяют на любом этапе разработки внести изменения в базу данных и модифицировать ее структуру без ущерба для введенных ранее данных.

Этапы проектирования и создания базы данных Access иллюстрирует схема, приведенная на рис. 2.1.

В процессе разработки *канонической модели данных* предметной области для проектирования реляционной базы данных необходимо выделить *информационные объекты* (ИО), соответствующие требованиям нормализации данных, и определить связи между ИО с типом отношений 1 : М.

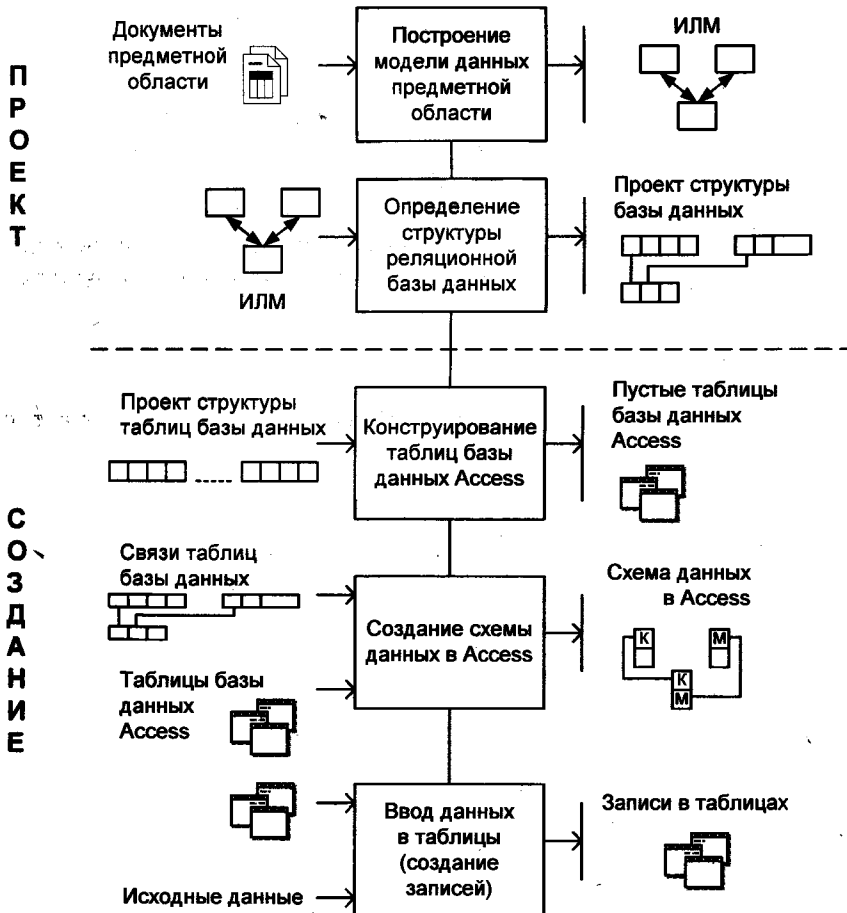


Рис. 2.1. Этапы проектирования и создания базы данных Access

При определении проекта логической структуры реляционной базы данных каждый информационный объект канонической модели предметной области адекватно отображается реляционной таблицей, а связям между двумя информационными объектами соответствуют логические связи между парой соответствующих таблиц. Такие связи устанавливаются по уникальному ключу главной таблицы. Во второй таблице, которая является подчиненной, поле связи может быть либо частью ее уникального ключа, либо не входить в состав ключа вовсе.

В процессе создания базы данных на компьютере сначала осуществляется конструирование ее таблиц средствами Access. Для поддержания целостности данных в каждой таблице определяются ключевое поле и правила проверки значений данных в полях. Далее создается схема данных, в которой устанавливаются логические связи таблиц. В схеме данных базы могут быть заданы параметры поддержания связной целостности данных.

Связная целостность данных означает, что в базе данных установлены и корректно поддерживаются взаимосвязи между записями разных таблиц при загрузке, добавлении и удалении записей в связанных таблицах, а также при изменении значений ключевых полей. При обеспечении связной целостности в подчиненной таблице не может существовать запись, для которой отсутствует связанная запись в главной таблице.

После формирования в Access схемы данных можно приступать к вводу данных в базу — *загрузке* — с документов предметной области, являющихся источниками данных. В практических приложениях пользователя обычно не используется ввод непосредственно в таблицы, а применяются специально создаваемые экранные формы, играющие роль интерфейса пользователя. Фактически новый документ готовится (заполняется) на компьютере с помощью экранной формы и затем сохраняется в базе данных.

Проектирование базы данных, основанное на построении нормализованной модели данных предметной области, позволяет легко получить логическую структуру реляционной базы данных Access, в которой автоматически поддерживается целостность и непротиворечивость данных.

Построение информационно-логической модели данных

Информационно-логическая модель (ИЛМ) отображает данные предметной области в виде совокупности информационных объектов (ИО) и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных. Каждый информационный объект в модели данных должен иметь уникальное имя.

Информационные объекты

Информационный объект — это информационное описание некоторой сущности предметной области: реального объекта, процесса, явления или события. Информационный объект является совокупностью логически взаимосвязанных реквизитов,

представляющих качественные и количественные характеристики сущности. Примерами сущностей являются: товар, поставщик, заказчик, поставка, отгрузка, сотрудник, отдел, студент, преподаватель, кафедра и т. п.

Информационный объект имеет множество реализаций — *экземпляров объекта*. Например, каждый экземпляр информационного объекта *товар* содержит значения реквизитов по товару определенного наименования. Экземпляр объекта должен однозначно определяться среди всего множества экземпляров, т. е. идентифицироваться значением *уникального (первичного) ключа* информационного объекта. Уникальность ключа означает, что любое значение ключа не может повториться в каком-либо другом экземпляре объекта. *Простой ключ* состоит из одного реквизита. *Составной ключ* — из нескольких реквизитов. Таким образом, реквизиты информационного объекта подразделяются на ключевые и описательные, которые являются функционально зависимыми от ключа.

Функциональные зависимости реквизитов

Информационные объекты могут быть выделены на основе описания предметной области путем определения *функциональных зависимостей между реквизитами* предметной области. Функциональная зависимость реквизитов информационного объекта устанавливает соответствие значений ключевых (определяющих) и неключевых (определяемых) реквизитов.

Необходимость установления функциональных зависимостей связана с требованием баз данных по однозначной определяемости любых данных для их размещения и доступа к ним. Например, для однозначной и полной определяемости реквизита *количество* в поставке товара нужно указать, *что* поставляется, — т. е. товар (его идентификатор), и нужно указать, *кем* поставляется, — т. е. указать поставщика (его идентификатор). Если возможны поставки одного товара поставщиком в разные сроки, то нужно также указать, *когда* осуществляется поставка, — т. е. принять срок поставки, как показатель, тоже участвующий в идентификации поставки.

Если не будет выявлена полная функциональная зависимость, реквизит *количество* поставки останется не определенным однозначно. Так, если в функциональных зависимостях количества поставки не определен срок поставки, то нельзя отличить друг от друга несколько поставок одного и того же товара одним и тем же поставщиком, но в разные сроки. В этом случае имеет место неполная идентификация поставки на этапе проектирования. После реализации проекта это приведет к возможности загрузки в базу данных сведений о количестве только по одной поставке (первой введенной), т. к. для одного значения идентификатора Код товара + Код поставщика, т. е. одного товара и одного поставщика, может быть введено только одно значение количества поставляемого. Наглядная иллюстрация этой ситуации, когда срок поставки не указан как ключевой (определяющий поставку), представлена в табл. 2.1.

ЗАМЕЧАНИЕ

При выявлении функциональных зависимостей реквизитов не рассматриваются арифметические зависимости (например, стоимость от количества), поскольку устанавливается только функциональная зависимость, определяющая логические связи описательных и ключевых реквизитов.

Таблица 2.1. Пример идентификации поставок товаров

Реквизит	Код товара	Код поставщика	Срок поставки	Количество поставки	Результат в базе данных
Роль в функциональной зависимости	Ключевой	Ключевой		Зависимый	
	Идентификатор поставки				
Поставка 1	T1	P1	Февраль	10	Введена
Поставка 2	T1	P1	Апрель	20	Не введена

При графическом изображении модели данных каждый информационный объект представляется прямоугольником с обозначением его имени и идентификатора — ключа. Пример такого изображения для информационных объектов ТОВАР и ПОСТАВКА показан на рис. 2.2. Здесь КОД_Т (код товара) — простой ключ объекта ТОВАР, а КОД_Т + КПОСТ (код поставщика) + СРОКП (срок поставки) — составной ключ объекта ПОСТАВКА.



Рис. 2.2. Примеры графического изображения информационных объектов с простым (слева) и составным (справа) ключами

Требования нормализации

Реквизиты каждого информационного объекта канонической модели данных должны отвечать требованиям, соответствующим *третьей нормальной форме* реляционной модели данных:

- информационный объект должен содержать уникальный идентификатор — ключ;
- все описательные реквизиты должны быть взаимонезависимы, т. е. между ними не должно быть функциональных зависимостей;
- все реквизиты, входящие в составной ключ, также должны быть взаимонезависимы;
- каждый описательный реквизит должен *функционально полно* зависеть от ключа, т. е. каждому значению ключа должно соответствовать только одно значение описательного реквизита, а при составном ключе описательные реквизиты должны зависеть целиком от всей совокупности реквизитов, образующих ключ;
- каждый описательный реквизит должен зависеть от ключа *нетранзитивно*, т. е. не должен зависеть через другой промежуточный реквизит.

ЗАМЕЧАНИЕ

В случае транзитивной зависимости между реквизитами информационного объекта можно выполнить расщепление совокупности реквизитов с образованием двух информационных объектов вместо одного.

Выполнение требований нормализации обеспечивает построение канонической модели данных и создание на ее основе реляционной базы данных без дублирования описательных данных, а также возможность автоматического поддержания связной целостности данных средствами СУБД при обновлении базы данных (добавлении и удалении записей, изменении значений в ключевых полях).

Выделение информационных объектов предметной области

Процесс выделения информационных объектов предметной области, отвечающих требованиям нормализации, может производиться как на основе интуитивного, так и формального (аналитического) подхода. При интуитивном подходе сразу устанавливаются информационные объекты, соответствующие реальным объектам. Однако получаемая при этом информационно-логическая модель, как правило, требует дальнейших преобразований, в частности, преобразования много-многозначных связей между объектами. При отсутствии достаточного опыта использования такого подхода возможны существенные ошибки. Последующая проверка выполнения требований нормализации обычно показывает необходимость уточнения структуры информационных объектов.

Теоретические основы аналитического подхода были разработаны и полно изложены известным американским ученым Дж. Мартином в его монографиях по организации баз данных. При аналитическом подходе в ходе исследования предметной области сначала необходимо выявить совокупность данных и различных сведений об объектах и процессах, характеризующих рассматриваемую область, перечень документов, содержащих эти данные, а также комплекс задач и запросов, которые предполагается реализовать. Таким образом, определению структуры базы данных предшествует выявление информационных потребностей приложений пользователя. Основным источником данных являются справочные, плановые и оперативно-учетные документы.

На основе исследования составляется описание предметной области и документов, данные из которых нужно разместить в базе.

Далее выполняется *информационный анализ* предметной области с целью формализации и моделирования данных. Такая формализация необходима для их организации и обработки данных во внутримашинной сфере. При этом должен быть произведен семантический анализ данных и установлены функциональные зависимости реквизитов. Далее на их основе должны быть выявлены информационные объекты и логические взаимосвязи между ними. В результате данные предметной области будут структурированы.

Информационный анализ и определение логической структуры информации

Информационный анализ включает:

- структурирование информации предметной области;
- формализацию и моделирование данных.

Структура информации немашинной сферы находит отражение в ее представлении отдельными структурными единицами — *реквизитами*, их группировке в документах-источниках и упорядочении по классификационным признакам.

Структурирование информации

Рассмотрим структурирование информации применительно к организационно-экономической сфере. Такая информация имеет дискретный характер и поэтому может быть структурирована, т. е. представлена как совокупность отдельных *структурных единиц информации*. Определим важнейшие виды структурных единиц информации:

- реквизит* — простейшая структурная единица информации, неделимая на смысловом уровне, отражающая количественную или качественную характеристику сущности (объекта, процесса и т. п.) предметной области. Можно выделить реквизиты-признаки и реквизиты-основания:
 - *реквизит-признак* позволяет выделить (идентифицировать) объект из множества однотипных объектов (как правило, символическое представление);
 - *реквизит-основание* содержит количественную характеристику объекта, процесса или другой сущности, определяющую их состояние (как правило, числовое значение).

Например, в плане поставок товаров реквизиты-признаки идентифицируют поставку, а реквизиты-основания определяют количество поставляемого товара, его стоимость;

- составная единица информации (СЕИ)* — логически взаимосвязанная совокупность реквизитов.

Примером составной единицы информации может служить документ. Семантика и размещение реквизитов в форме документа определяют роль реквизитов в структуре информации, содержащейся в документе.

В процессе информационного семантического (смыслового) анализа нужно выявить функциональные зависимости реквизитов и определить реквизитный состав информационных объектов.

Для минимизации возможных ошибок целесообразно производить семантический анализ по каждой из форм документов в отдельности. Это связано с тем, что форма документа уже отображает структуру данных, т. к. любой документ объединяет логически взаимосвязанные реквизиты.

Функциональную зависимость реквизитов удобно изобразить графически в виде линий со стрелками, идущих от ключевого (определяющего) реквизита к описа-

тельному реквизиту (определяемому). Ключевой реквизит обычно отмечается особо. Функциональную зависимость можно отображать непосредственно в таблице, где представлен состав реквизитов каждого документа. Это показано на примере реквизитов документа «Справочник товаров» (рис. 2.3), где каждый из описательных реквизитов однозначно определяется ключевым реквизитом Код товара.

Наименование реквизита	Имя реквизита	Функциональные зависимости
Код товара	КОД_ТОВ	
Наименование	НАИМ_ТОВ	
Цена за единицу	ЦЕНА	
Единица измерения	ЕИ	

Рис. 2.3. Функциональная зависимость реквизитов документа «Справочник товаров»

Выделение информационных объектов

На основе описания предметной области необходимо выявить документы-источники и их реквизиты, подлежащие хранению в базе данных. Затем надо перейти к информационному анализу этих документов для определения функциональных зависимостей и выявления информационных объектов.

Рассмотрим порядок действий, которые могут быть использованы для выделения информационных объектов, отвечающих указанным ранее требованиям нормализации.

1. Определение функциональных зависимостей между реквизитами документа.

Для этого анализируется роль реквизитов в структуре информации документа. Сначала целесообразно выявить реквизит (один или несколько), который играет роль общего идентификатора всей информации документа. Как правило, к таким реквизитам относятся номер документа, идентификатор подразделения предприятия, выпускающего документ, и т. п. От такого идентификатора документа будут функционально полно зависеть некоторые описательные реквизиты в общей части документа (например, идентификатор документа-основания). Заметим, что в табличной части документа такие реквизиты, как количество товара и стоимость, будут частично функционально зависимыми от общего идентификатора документа.

Все справочные реквизиты реальных объектов (товаров, предприятий, подразделений и т. п.), как в общей, так и в табличной части, функционально полно определяются идентификаторами этих объектов (см. пример на рис. 2.3). В результате для каждого определяемого реквизита должны быть выявлены реквизиты (ключевые), которые в совокупности однозначно его определяют (одному значению ключа соответствует одно значение описательного реквизита).

Для графического отображения функциональной зависимости проводится линия связи со стрелкой к зависимому реквизиту от определяющего его реквизита.

2. В результате просмотра выявленных функциональных зависимостей выбираются зависимые реквизиты и для каждого из них устанавливаются ключевые реквизиты, т. е. те (один или несколько), которые в совокупности определяют его однозначно.

Такое соответствие описательных и ключевых реквизитов удобно представить в таблице, форма которой с примером заполнения представлена в табл. 2.2.

Таблица 2.2. Соответствие описательных и ключевых реквизитов

Описательные (зависимые) реквизиты	Ключевые реквизиты	Признак ключа	Имя ИО, включающего реквизиты
Количество поставки	Код товара + Код поставщика + Срок поставки	Уникальный составной	ПОСТАВКА
Наименование товара	Код товара	Уникальный простой	ТОВАР
Наименование поставщика	Код поставщика	Уникальный простой	ПОКУПАТЕЛЬ

3. Группируются реквизиты, одинаково зависимые от ключевых реквизитов. Полученные группы зависимых реквизитов вместе с их ключевыми реквизитами образуют реквизитный состав соответствующих информационных объектов.

При следовании приведенным правилам не требуется отдельно преобразовывать транзитивные зависимости реквизитов. При таких зависимостях некоторые реквизиты являются одновременно и зависимыми, и ключевыми и, соответственно, будут представлены в группе зависимых и ключевых.

После выделения информационных объектов необходимо сформировать их окончательное описание. В таком описании кроме состава реквизитов и указания ключа может быть представлена также семантика информационных объектов — их смысловое определение.

ЗАМЕЧАНИЕ

Обычно при следовании приведенным правилам сразу оказываются выделенными объекты, играющие роль связи между объектами, находящимися в отношении «многие-ко-многим» ($M : N$). Соответственно, в модели можно ограничиться рассмотрением только одно-многозначных связей.

Совокупность выделенных информационных объектов после определения связей между объектами позволяет получить информационно-логическую модель, не предусматривающую дальнейших преобразований для создания реляционной базы данных, отвечающей требованиям нормализации.

Выделение информационных объектов на примере предметной области «Поставка товаров»

Рассмотрим выделение информационных объектов на примере предметной области «Поставка товаров».

Описание предметной области

Пусть необходимо построить базу данных, содержащую информацию о планируемых поставках товаров покупателям и фактических отгрузках товаров в соответствии с планом поставок. Такая база данных должна обеспечить подготовку, хранение и обработку данных по договорам с покупателями и по фактическим отгрузкам товаров, а также анализ выполнения договорных обязательств на поставку по срокам и объемам.

Информационное обеспечение такого приложения пользователя включает:

- справочную информацию о поставляемых товарах;
- справочную информацию о покупателях (заказчиках);
- справочную информацию о складах предприятия, где хранится товар;
- данные о плановых поставках товаров;
- оперативно-учетные данные об отгрузках товаров со складов покупателям.

В результате анализа предметной области выявляются документы — источники для создания базы данных.

Справочная информация содержится в документах «Справочник товаров», «Справочник покупателей», «Справочник складов». На рис. 2.4–2.6 приведены формы этих документов.

СПРАВОЧНИК ТОВАРОВ, ПОСТАВЛЯЕМЫХ ФИРМОЙ

Код товара	Наименование	Единица измерения	Цена	НДС
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Рис. 2.4. Форма документа «Справочник товаров, поставляемых фирмой»

СПРАВОЧНИК СКЛАДОВ ФИРМЫ

Фирма /код, наименование/

Код склада	Наименование	Адрес	Отв. лицо
_____	_____	_____	_____
_____	_____	_____	_____

Рис. 2.5. Форма документа «Справочник складов фирмы»

СПРАВОЧНИК ПОКУПАТЕЛЕЙ ФИРМЫ

Код	ИНН	Наименование	Адрес	Тел.	№ расч. счета	Банк
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____

Рис. 2.6. Форма документа «Справочник покупателей фирмы»

Информация о планируемых поставках содержится в договорах. Форма договора приведена на рис. 2.7.

ДОГОВОР № _____

" ____ " _____ 201__ г.

<p style="text-align: center;">Поставщик</p> <p>ИНН, код _____</p> <p>наименование _____</p> <p>адрес _____</p> <p>телефон _____</p> <p>банк _____</p> <p>расчетный счет _____</p>	<p style="text-align: center;">Покупатель</p> <p>ИНН, код _____</p> <p>наименование _____</p> <p>адрес _____</p> <p>телефон _____</p> <p>банк _____</p> <p>расчетный счет _____</p>
---	--

Код товара	Наименование	Единица измерения	Цена	Срок поставки (месяц)	Мин. партия поставки	Кол-во	Сумма

Сумма всего _____

Рис. 2.7. Форма договора на поставку товаров

Учетная информация с данными по фактической отгрузке товаров покупателю со склада фирмы в соответствии с договорами содержится в расходных накладных (рис. 2.8).

Особо отметим, что документы предметной области не только дают возможность выявить структуру данных, но также являются основой для разработки электронных форм ввода/вывода и отчетов для печати документов после их подготовки на компьютере и сохранения в базе данных.

Как отмечалось ранее, информационный анализ для выделения объектов целесообразно производить по каждой из форм документов, содержащих данные, которые должны быть размещены в базе.

НАКЛАДНАЯ № _____

" ____ " _____ 201 ____ г.

<u>Поставщик</u>	<u>Покупатель</u>
ИНН, код _____	ИНН, код _____
наименование _____	наименование _____
<u>Склад-грузоотправитель</u>	
Код _____	
наименование _____	
Договор № ____ от " ____ " _____ 201 ____ г.	

Код товара	Наименование	Единица измерения	Цена	Ставка НДС	Количество	Сумма

Сумма всего _____

Отпустил _____ ФИО материально отв. лица _____

Рис. 2.8. Форма накладной с данными по фактической отгрузке товаров

Выделение объектов справочной информации

Каждый справочник, как правило, содержит только табличную часть, и в ней имеется реквизит, играющий роль идентификатора строки (код или номер).

Например, в справочнике товаров идентификатором является Код товара, который однозначно определяет описательные реквизиты: Наименование товара, Единицы измерения и т. д. В справочнике складов идентификатором является Код склада, который однозначно определяет Наименование склада и другие реквизиты. В справочнике покупателей уникальным идентификатором является ИНН, а также Код, который может иметь небольшую длину и присваивается для удобства внутрисистемного использования.

Анализ справочников товаров и покупателей. Определим функциональные зависимости между реквизитами документа «Справочник товаров», предварительно составив их перечень, как показано на рис. 2.9. Присвоим реквизитам справочника товаров сокращенные обозначения — имена.

Особенность такого простейшего по структуре данных документа состоит в том, что все реквизиты документа содержатся в таблице. Из анализа документа очевидно, что реквизиты Наименование товара (НАИМ_ТОВ), Единица измерения (ЕИ), Цена

(ЦЕНА), Ставка НДС являются описательными. Каждый из них функционально полно зависит только от уникального идентификатора товара — Кода товара (КОД_ТОВ). Таким образом, КОД_ТОВ является ключевым.

Отобразим зависимости каждого описательного реквизита товара от кода товара, который однозначно их определяет, линиями со стрелкой от ключевого реквизита к описательному.

Реквизиты справочника товаров	Имя реквизита	Функциональные зависимости
Код товара	КОД_ТОВ	
Наименование товара	НАИМ_ТОВ	
Единица измерения	ЕИ	
Цена	ЦЕНА	
Ставка НДС	СТАВКА_НДС	

Рис. 2.9. Функциональные зависимости реквизитов справочника товаров

Для документов с такой простейшей структурой данных нет необходимости в дополнительных действиях по выделению информационных объектов, т. к. здесь все реквизиты образуют одну группу реквизитов с общим ключом, т. е. все реквизиты простейшего справочника образуют один информационный объект. Таким объектом для этого документа будет являться объект ТОВАР с ключом Код товара (КОД_ТОВ).

Аналогично легко определить информационный объект по справочнику покупателей. Если в качестве ключа принять Код покупателя (КОД_ПОК), более короткий по сравнению с ИНН, он однозначно определит реквизиты: ИНН, Наименование, Адрес, Телефон, Банк, Номер расчетного счета. Соответственно выделяется информационный объект ПОКУПАТЕЛЬ с ключом Код покупателя.

ЗАМЕЧАНИЕ

ИНН можно использовать в качестве альтернативного ключа и создать по нему уникальный индекс в базе данных.

Реквизитный состав информационных объектов справочника товаров и справочника покупателей представлен в табл. 2.3.

Таблица 2.3. Группировка реквизитов по информационным объектам ТОВАР и ПОКУПАТЕЛЬ

Реквизиты объекта	Признак ключа	Имя информационного объекта	Семантика объекта
КОД_ТОВ	Простой уникальный	ТОВАР	Сведения о поставляемых товарах
НАИМ_ТОВ ЦЕНА ЕИ СТАВКА_НДС			

Таблица 2.3 (окончание)

Реквизиты объекта	Признак ключа	Имя информационного объекта	Семантика объекта
КОД_ПОК	Простой уникальный	ПОКУПАТЕЛЬ	Сведения о покупателях фирмы
ИНН НАИМ_ПОК АДРЕС_ПОК ТЕЛ_НОМ_РСЧ БАНК			

Анализ документа «Справочник складов фирмы». Определим функциональные зависимости между реквизитами документа «Справочник складов фирмы», предварительно составив их перечень (рис. 2.10). Присвоим реквизитам сокращенные обозначения — имена. Все описательные реквизиты склада: Наименование склада (НАИМ_СК), Адрес (АДРЕС_СК), Отв. лицо (ОТВ_ЛИЦО) однозначно определены Кодом склада (КОД_СК), который является ключевым. Обратим внимание, что в общей части документа указан код фирмы (КОД_Ф) — один для всего списка складов. Очевидно, каждому значению кода склада соответствует только одно значение кода фирмы, т. е. можно считать, что имеет место полная функциональная зависимость кода фирмы от кода склада. Наименование фирмы определяется однозначно кодом фирмы.

ЗАМЕЧАНИЕ

Если зависимость между КОД_Ф и КОД_СК не выявлена, то все множество реквизитов документа разделится на два не связанных между собой подмножества, а это не логично для реквизитов одного документа.

Как можно видеть, все установленные функциональные зависимости реквизитов документа «Справочник складов фирмы» на рис. 2.10 отражены.

Заметим, что реквизит КОД_Ф одновременно выступает в роли описательного реквизита в одной связи и ключевого — в другой. Таким образом, здесь мы сталкиваемся

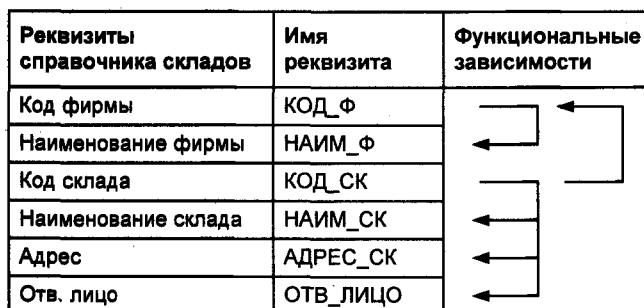


Рис. 2.10. Функциональная зависимость реквизитов справочника складов фирмы

с транзитивной зависимостью. Реквизит `НАИМ_Ф` транзитивно зависит от `КОД_СК` через `КОД_Ф`. Тем не менее, специальных действий по расщеплению этой зависимости при следовании приведенным ранее правилам не потребуется.

Выберем по функциональным связям реквизиты, зависимые от каких-либо других реквизитов, и укажем для них ключевые реквизиты.

Так, при просмотре списка реквизитов сверху находим первый зависимый реквизит `КОД_Ф`, к которому подходит стрелка, и устанавливаем реквизит (ключевой), от которого идет стрелка, — `КОД_СК`. Далее находим второй зависимый (описательный) реквизит `НАИМ_Ф` и устанавливаем его ключевой `КОД_Ф`. Аналогично находим описательный `НАИМ_СК` и устанавливаем его ключевой `КОД_СК` и т. д. Выявленное соответствие описательных и ключевых реквизитов представлено в табл. 2.4.

Таблица 2.4. Соответствие описательных и ключевых реквизитов документа «Справочник складов фирмы»

Описательные (зависимые) реквизиты	Ключевые реквизиты	ИО, включающий реквизиты
КОД_Ф	КОД_СК	СКЛАД
НАИМ_Ф	КОД_Ф	ФИРМА
НАИМ_СК	КОД_СК	СКЛАД
АДРЕС_СК	КОД_СК	СКЛАД
ОТВ_ЛИЦО	КОД_СК	СКЛАД

Сгруппируем реквизиты, зависимые от одних и тех же ключевых реквизитов, и объединим их с ключевыми реквизитами в один информационный объект.

Результат группировки реквизитов документа «Справочник складов фирмы» приведен в табл. 2.5.

Таблица 2.5. Группировка реквизитов по информационным объектам документа «Справочник складов фирмы»

Реквизиты объекта	Признак ключа	Имя информационного объекта	Семантика объекта
КОД_СК	Простой уникальный	СКЛАД	Сведения о складах фирмы
КОД_Ф НАИМ_СК АДРЕС_СК ОТВ_ЛИЦО			
КОД_Ф	Простой уникальный	ФИРМА	Сведения о фирме
НАИМ_Ф			

Таким образом, на основе анализа документа «Справочник складов фирмы» выделены два информационных объекта: ФИРМА и СКЛАД.

Заметим, что особенность объекта ФИРМА состоит в том, что он имеет единственный экземпляр, поэтому этот объект можно не отображать в базе данных отдельной таблицей.

Выделение объектов плановой и учетной информации

Анализ документа «Договор на поставку товаров». Определим функциональные зависимости реквизитов документа «Договор», предварительно составив их перечень (рис. 2.11). Присвоим реквизитам сокращенные обозначения — имена.

Реквизиты документа "Договор"	Имя реквизита	Функциональные зависимости
Номер договора	НОМ_ДОГ	
Дата договора	ДАТА_ДОГ	
Код покупателя	КОД_ПОК	
Сумма по договору	СУММА_ДОГ	
Код товара	КОД_ТОВ	
Срок поставки	СРОК_ПОСТ	
Количество поставки товара	КОЛ_ПОСТ	
Мин. партия поставки товара	МИН_ПОСТ	
Сумма поставки товара	СУММА_ПОСТ	

Рис. 2.11. Функциональные зависимости реквизитов документа «Договор»

Рассмотрим функциональные зависимости между реквизитами общей части документа «Договор». Номер договора присваивается в порядке подготовки нового документа. Этот номер является уникальным среди всех номеров договоров.

Каждый из реквизитов: Дата заключения договора, Код покупателя — имеет единственное значение в договоре. Соответственно, каждый из этих реквизитов однозначно определяется идентификатором документа — Номером договора. Общим идентификатором договора определяется также однозначно реквизит Сумма всего.

Кодом покупателя однозначно определяются описательные реквизиты покупателя: Наименование, ИНН, Адрес, Телефон, Банк, Расчетный счет. В таблице зависимостей их можно не отображать, поскольку информационный объект, образованный этими реквизитами, был уже выделен на основе справочника покупателей.

Описательные реквизиты фирмы, выступающей в этом документе в качестве поставщика, определяются однозначно идентификатором фирмы.

Рассмотрим функциональные зависимости реквизитов табличной части договора. Табличная часть (спецификация договора) содержит реквизиты, имеющие множество значений в соответствующих столбцах, т. к. договор может включать несколько наименований товаров. Среди этих реквизитов особую роль играет Код товара, который фактически является идентификатором каждой строки в документе.

ЗАМЕЧАНИЕ

Поскольку в договоре может быть несколько товаров, номер договора не определяет однозначно товар, указанный в документе, и, соответственно, не может быть установлена функциональная зависимость Кода товара от Номера договора.

Описательные реквизиты товара: Наименование, Единица измерения, Цена однозначно определены Кодом товара. Эти реквизиты можно не включать в таблицу зависимостей, поскольку их взаимосвязи были установлены ранее, при анализе справочника товаров. Что касается реквизитов-оснований документа, таких как Количество поставки товара, Сумма поставки товара, Мин. партия поставки товара — эти реквизиты внутри документа идентифицируются Кодом товара в соответствующей строке, а полная идентификация среди всех договоров образуется добавлением реквизита Номер договора к реквизиту Код товара. Все это справедливо лишь в том случае, если для любого товара Срок поставки может быть только один. Причем в этом случае реквизиты Количество поставки товара, Сумма поставки товара, Мин. партия поставки товара и Срок поставки товара функционально полно будут зависеть от составного ключа Номер договора + Код товара.

Допустим, что в договоре для одного товара возможно несколько сроков поставки, тогда срок поставки должен войти в общий идентификатор для реквизитов Количество поставки товара, Сумма поставки товара и Мин. партия поставки товара. Таким образом, эти реквизиты будут функционально полно зависеть от составного ключа Номер договора + Код товара + Срок поставки (см. рис. 2.11).

Выделение информационных объектов по документу «Договор на поставку товаров». Проанализировав выявленные функциональные взаимосвязи реквизитов, установим, от каких реквизитов зависит каждый реквизит, к которому подходит стрелка. Таким образом, определим соответствие описательных и ключевых реквизитов. Затем сгруппируем реквизиты, зависящие от одинаковых ключевых, и объединим их с ключевыми реквизитами в один информационный объект (ИО). Результат группировки по ИО реквизитов документа «Договор» представлен в табл. 2.6.

Таблица 2.6. Группировка реквизитов по информационным объектам документа «Договор»

Реквизиты объекта	Признак ключа	Имя информационного объекта	Семантика объекта
НОМ_ДОГ	Простой уникальный	ДОГОВОР	Общие сведения по договору
ДАТА_ДОГ КОД_ПОК СУММА_ДОГ			
НОМ_ДОГ + КОД_ТОВ + СРОК_ПОСТ	Составной уникальный	ПОСТАВКА_ПЛАН	Сведения о поставках товаров по договору
КОЛ_ПОСТ МИН_ПОСТ СУММА_ПОСТ			

Анализ документа «Накладная». Рассмотрим функциональные зависимости реквизитов общей части накладной. Номер накладной присваивается в порядке подготовки нового документа. Этот номер можно считать уникальным только среди всех номеров накладных, выписанных на конкретном складе (т. е. он на этом складе не повторяется). Для уникальной идентификации накладных по всей фирме надо принять составной идентификатор Номер накладной + Код склада.

Реквизиты Дата выписки накладной (отгрузки товаров) и Номер договора имеют единственное значение в накладной, и, соответственно, каждый из них однозначно определяется идентификатором накладной (Номер накладной + Код склада). **Общим идентификатором накладной** определяется также однозначно реквизит Сумма всего.

Дата заключения договора однозначно определяется Номером договора, что было уже учтено при анализе договора.

Как было рассмотрено ранее, Код покупателя однозначно определяется в первичном документе «Договор» Номером договора. Поэтому будем учитывать функциональную зависимость Кода покупателя от Номера договора.

ИНН и Наименование покупателя однозначно определяются Кодом покупателя, что уже было учтено при анализе справочника покупателей.

Описательные реквизиты фирмы (ИНН и Наименование фирмы), выступающей в документе в качестве поставщика, определяются однозначно идентификатором фирмы — Кодом фирмы. Как было принято ранее, объект ФИРМА нецелесообразно отображать отдельным объектом в базе данных.

Описательные реквизиты Наименование склада и Отв. лицо (ФИО ответственного лица) однозначно определены Кодом склада, что уже было учтено при анализе справочника складов.

ЗАМЕЧАНИЕ

Номер договора, Код склада играют в документе важную роль связи накладной с соответствующими справочными и плановыми данными. К этим данным относятся данные по складу (с которого отпущен товар), данные по договору (на основании которого выписывается накладная). Как видно из таблиц функциональных зависимостей, реквизит Номер договора с одной стороны является ключевым для соответствующих данных, а с другой — описательным для накладной.

Рассмотрим функциональные зависимости реквизитов табличной части накладной. Табличная часть содержит реквизиты, имеющие множество значений в соответствующих столбцах, т. к. накладная может содержать несколько наименований товаров, отпущенных покупателю. Среди этих реквизитов особую роль играет Код товара, который является идентификатором каждой строки в документе.

ЗАМЕЧАНИЕ

Поскольку в накладной может быть несколько товаров, идентификатор накладной не определяет однозначно товар, указанный в документе, и, соответственно, не может быть установлена функциональная зависимость Кода товара от идентификатора накладной.

Описательные реквизиты товара Наименование, Единица измерения, Цена, Ставка НДС однозначно определены Кодом товара, что уже было учтено при анализе справочника покупателей.

Реквизиты-основания накладной Количество отгруженного товара и Сумма за товар определяют количественные характеристики объекта ОТГРУЗКА ТОВАРОВ. Эти реквизиты внутри одной накладной идентифицируются Кодом товара в соответствующей строке, а полная идентификация на всем множестве накладных образуется добавлением к Коду товара идентификатора накладной. Таким образом, реквизиты Количество отгруженного товара и Сумма за товар однозначно определяются составным идентификатором Номер накладной + Код склада + Код товара.

На рис. 2.12 наглядно представлены все рассмотренные функциональные зависимости реквизитов накладной.

Реквизиты документа "Накладная"	Имя реквизита	Функциональные зависимости
Номер накладной	НОМ_НАКЛ	
Код склада	КОД_СК	
Дата отгрузки	ДАТА_ОТГР	
Номер договора	НОМ_ДОГ	
Сумма всего	СУММА_НАКЛ	
Код товара	КОД_ТОВ	
Количество отгруженного товара	КОЛ_ОТГР	
Сумма за товар	СУММА_ОТГР	

Рис. 2.12. Функциональные зависимости реквизитов накладной

Выделение информационных объектов по накладной. Проанализировав выявленные функциональные взаимосвязи реквизитов, установим, от каких реквизитов зависит каждый реквизит, к которому подходит стрелка. Таким образом, определим соответствие описательных и ключевых реквизитов накладной. Затем сгруппируем реквизиты, зависящие от одинаковых ключевых, и объединим их с ключевыми реквизитами в один информационный объект (ИО). Результат группировки по ИО реквизитов накладной представлен в табл. 2.7.

Таблица 2.7. Группировка реквизитов по информационным объектам документа «Накладная»

Реквизиты ИО	Признак ключа	Имя ИО	Семантика ИО (описание)
НОМ_НАКЛ + КОД_СК	Составной уникальный	НАКЛАДНАЯ	Общие сведения по накладной
ДАТА_ОТГР НОМ_ДОГ СУММА_НАКЛ			

Таблица 2.7 (окончание)

Реквизиты ИО	Признак ключа	Имя ИО	Семантика ИО (описание)
НОМ_НАКЛ + КОД_СК + КОД_ТОВ	Составной уникальный	ОТГРУЗКА	Данные по отгрузке товара
КОЛ_ОТГР СУММА_ОТГР			

Связи информационных объектов

Следующим шагом проектирования после выявления информационных объектов является определение связей между ними. Связь устанавливается между двумя информационными объектами. Наличие связи, как правило, определяется природой реальных объектов, процессов или явлений, отображаемых этими информационными объектами. Связь между объектами существует, если логически взаимосвязаны экземпляры этих информационных объектов. Например, связи имеются между такими парами объектов, как поставщик — товар, склад — товар, кафедра — преподаватель, группа — студент и т. п.

Тип связи информационных объектов

Связи информационных объектов могут быть разного типа:

- одно-однозначные (1 : 1);
- одно-многозначные (1 : M);
- много-многозначные (M : N).

Одно-однозначные связи (1 : 1) имеют место, когда каждому экземпляру первого объекта (А) соответствует только один экземпляр второго объекта (В), и наоборот, каждому экземпляру второго объекта (В) соответствует только один экземпляр первого объекта (А). Следует заметить, что такие объекты легко могут быть объединены в один, структура которого образуется объединением реквизитов обоих исходных объектов, а ключевым реквизитом может быть выбран любой из альтернативных ключей, т. е. ключей исходных объектов. Графическое изображение одно-однозначной связи приведено на рис. 2.13. Примерами одно-однозначных связей являются пары вида: группа — староста, фирма — расчетный счет в банке и т. п.

Одно-многозначные связи (1 : M) — это такие связи, когда каждому экземпляру одного объекта (А) может соответствовать несколько экземпляров другого объекта (В), а каждому экземпляру второго объекта (В) может соответствовать только

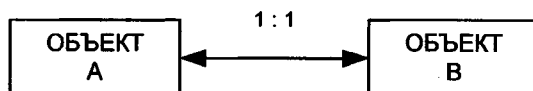


Рис. 2.13. Графическое изображение одно-однозначных отношений объектов

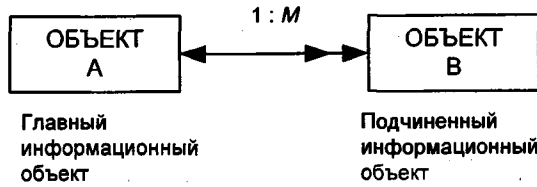


Рис. 2.14. Графическое изображение одно-многочисленных отношений объектов

один экземпляр первого объекта (А). Графическое изображение одно-многочисленной связи приведено на рис. 2.14.

В такой связи объект А является главным, а объект В — подчиненным, т. е. имеет место иерархическая подчиненность объекта В объекту А. Простейшими примерами одно-многочисленных связей объектов являются пары объектов: подразделение — сотрудник, кафедра — преподаватель, группа — студент и т. п.

Много-многочисленные связи (M : N) — это такие связи, когда каждому экземпляру одного объекта (А) могут соответствовать несколько экземпляров второго объекта (В), и наоборот, каждому экземпляру второго объекта (В) может соответствовать несколько экземпляров первого объекта (А). Графическое изображение связи типа M : N показано на рис. 2.15.

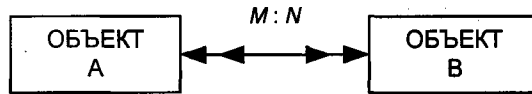


Рис. 2.15. Графическое изображение много-многочисленных отношений объектов

Много-многочисленные связи не могут непосредственно реализовываться в реляционной базе данных. Поэтому, в том случае, когда будут выявлены такие связи, может возникнуть необходимость их преобразования путем введения дополнительного объекта «связки». Исходные объекты будут связаны с этим объектом одно-многочисленными связями. Таким образом, объект-связка является подчиненным в одно-многочисленных связях по отношению к каждому из исходных объектов. Объект-связка должен иметь идентификатор, образованный из идентификаторов исходных объектов K_a и K_b (рис. 2.16).

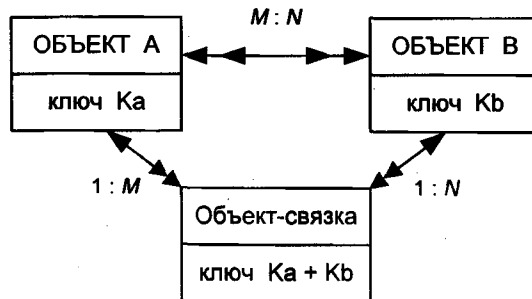


Рис. 2.16. Преобразование связи типа M : N с помощью объекта-связки

При таком подходе к выделению информационных объектов, объект-связка, как правило, выявляется в результате рассмотренного ранее анализа функциональных зависимостей реквизитов.

Примером много-многозначных связей является пара вида «поставщики — товары», если один поставщик поставляет разные наименования товаров, а товар одного наименования может поставляться несколькими поставщиками.

Определение связей между информационными объектами

Определение связей между информационными объектами и типы отношений, которыми они характеризуются, рассмотрим на примере предметной области «Поставка товаров».

Связи между объектами **ПОКУПАТЕЛЬ** → **ДОГОВОР** характеризуются одно-многозначными отношениями (1 : M), т. к. с одним покупателем может быть заключено несколько договоров, а один договор всегда заключается с конкретным покупателем.

Поскольку накладные строго привязаны к конкретному договору, а по одному договору может быть оформлено несколько накладных, между объектами **ДОГОВОР** и **НАКЛАДНАЯ** имеет место связь типа 1 : M.

Характерным случаем одно-многозначных связей являются связи объектов, образованные из документов с табличной частью (спецификацией). В рассматриваемой предметной области по документу «Договор» был выделен объект **ДОГОВОР**, соответствующий общей части документа, и объект **ПОСТАВКА_ПЛАН**, соответствующий строкам табличной части документа. Очевидна одно-многозначная связь между этими объектами **ДОГОВОР** → **ПОСТАВКА_ПЛАН**, поскольку в одном документе может содержаться некоторое множество строк, а каждая строка принадлежит только одному документу.

По документу «Накладная» были выделены два объекта, между которыми также имеет место одно-многозначная связь **НАКЛАДНАЯ** → **ОТГРУЗКА**.

Очевидно наличие связи между объектами **ТОВАР** → **ПОСТАВКА_ПЛАН**. Эту связь также определяют одно-многозначные отношения, поскольку каждый экземпляр поставки (одна из строк спецификации договора) — это данные по одному товару, а товар одного наименования может участвовать в разных плановых поставках товара (одного или разных договоров).

Аналогично устанавливается связь между объектами **ТОВАР** → **ОТГРУЗКА**, которые также находятся в одно-многозначных отношениях.

Связь между объектами **СКЛАД** → **НАКЛАДНАЯ** может быть установлена как одно-многозначная, поскольку по условиям рассматриваемой предметной области на каждом складе выписывается некоторое множество накладных, но каждая накладная выписывается на конкретном складе.

Следует отметить, что объект **ПОСТАВКА_ПЛАН** фактически играет роль объекта-связки в много-многозначных отношениях объектов **ДОГОВОР** и **ТОВАР**, а объект **ОТГРУЗКА**

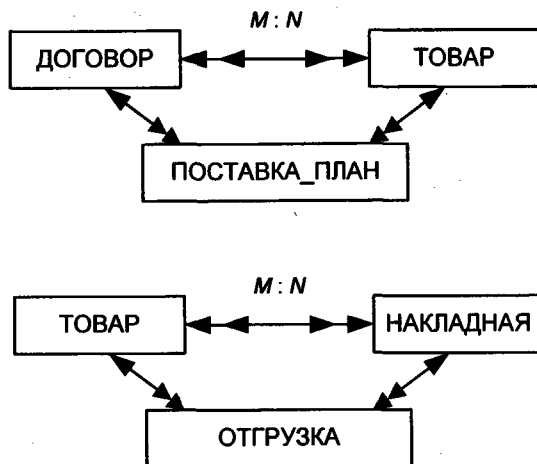


Рис. 2.17. Примеры много-многозначных отношений информационных объектов

играет роль объекта-связки в много-многозначных отношениях объектов НАКЛАДНАЯ и ТОВАР (рис. 2.17).

В табл. 2.8 перечислены все одно-многозначные связи между объектами и, соответственно, определены главные и подчиненные информационные объекты в этих связях.

Таблица 2.8. Связи информационных объектов

Главный объект	Подчиненный объект	Тип связи
ПОКУПАТЕЛЬ	ДОГОВОР	1 : M
ДОГОВОР	ПОСТАВКА_ПЛАН	1 : M
НАКЛАДНАЯ	ОТГРУЗКА	1 : M
ТОВАР	ПОСТАВКА_ПЛАН	1 : M
ТОВАР	ОТГРУЗКА	1 : M
СКЛАД	НАКЛАДНАЯ	1 : M
ДОГОВОР	НАКЛАДНАЯ	1 : M

Информационно-логическая модель предметной области

На рис. 2.18 представлена в каноническом виде информационно-логическая модель (ИЛМ) для рассматриваемой предметной области, построенная в соответствии с выявленными информационными объектами и связями между ними.

Объекты канонической модели размещены по уровням. На нулевом уровне размещаются объекты, не подчиненные никаким другим объектам. Уровень остальных

объектов определяется наиболее длинным путем к объекту от нулевого уровня. Такое размещение объектов дает представление об их иерархической подчиненности, делает модель более наглядной и облегчает понимание одно-многочленных отношений между объектами.

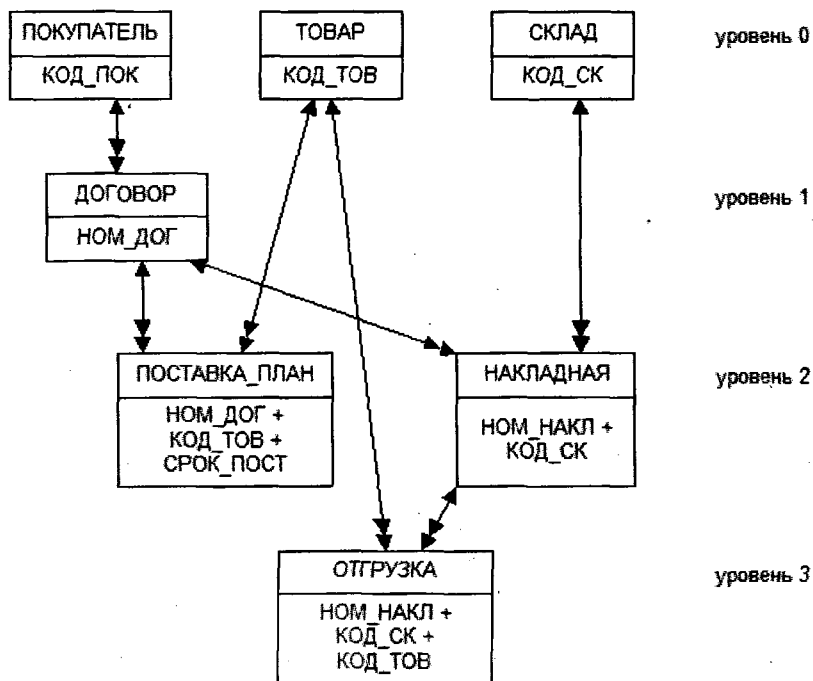


Рис. 2.18. Информационно-логическая модель предметной области «Поставка товаров»

ЗАМЕЧАНИЕ

Для определения уровня объектов на графе ИЛМ можно, условно удалив объекты нулевого уровня, найти объекты первого уровня. К объектам этого уровня следует отнести объекты, не подчиненные теперь никаким другим объектам. Аналогично определяются объекты каждого следующего уровня. При большом количестве объектов в ИЛМ аналогичные действия выполняются на матрице смежности модели.

Логическая структура реляционной базы данных

Логическая структура реляционной базы данных является адекватным отображением полученной информационно-логической модели предметной области. Для канонической модели не требуется дополнительных преобразований. Каждый информационный объект модели данных отображается соответствующей реляционной таблицей. Структура реляционной таблицы определяется реквизитным составом соответствующего информационного объекта, где каждый столбец (поле) соответ-

ствует одному из реквизитов объекта. Ключевые реквизиты объекта образуют уникальный ключ реляционной таблицы. Для каждого столбца таблицы (поля) задаются имя, тип, размер данных и другие свойства. Строки (записи) таблицы соответствуют экземплярам объекта и формируются при загрузке таблицы.

Связи между объектами модели данных реализуются одинаковыми реквизитами — *ключами связи* в соответствующих таблицах. При этом ключом связи типа 1 : M всегда является уникальный ключ главной таблицы. Ключом связи в подчиненной таблице является либо некоторая часть уникального ключа в ней, либо поле, не входящее в состав первичного ключа (например, код фирмы в таблице СКЛАД). Ключ связи в подчиненной таблице называется *внешним ключом*.

Все связи в полученной информационно-логической модели предметной области «Поставка товара» характеризуются отношением типа 1 : M. Соответственно, связь между таблицами ПОКУПАТЕЛЬ и ДОГОВОР осуществляется по коду покупателя (КОД_ПОК), который является уникальным идентификатором главного объекта ПОКУПАТЕЛЬ и не ключевым в объекте ДОГОВОР (см. табл. 2.6).

Связь между таблицами ТОВАР и ПОСТАВКА_ПЛАН осуществляется по уникальному ключу главного объекта ТОВАР — Коду товара, который в подчиненном объекте ПОСТАВКА_ПЛАН является частью ключа (см. табл. 2.6). Аналогично связь между таблицами ТОВАР и ОТГРУЗКА осуществляется по уникальному ключу главного объекта ТОВАР — Коду товара.

Связь между таблицами ДОГОВОР и НАКЛАДНАЯ осуществляется по уникальному ключу главного объекта ДОГОВОР — Номеру договора (НОМ_ДОГ), который в подчиненном объекте НАКЛАДНАЯ не входит в состав ключа (см. табл. 2.7).

Связь между таблицами ДОГОВОР и ПОСТАВКА_ПЛАН осуществляется по уникальному ключу главного объекта ДОГОВОР — Номеру договора (НОМ_ДОГ), который в подчиненном объекте ПОСТАВКА_ПЛАН является частью ключа (см. табл. 2.6).

Аналогично связь между таблицами НАКЛАДНАЯ и ОТГРУЗКА осуществляется по уникальному составному ключу главного объекта НАКЛАДНАЯ — НОМ_НАКЛ + КОД_СК, который в подчиненном объекте ОТГРУЗКА является частью ключа (см. табл. 2.7).

В Access может быть создана *схема данных*, наглядно отображающая логическую структуру базы данных. Определение одно-многозначных связей в этой схеме должно осуществляться в соответствии с построенной моделью данных. Топология схемы данных практически совпадает с топологией информационно-логической модели. Для модели данных предметной области (см. рис. 2.18), построенной в рассмотренном примере, логическая структура базы данных в виде схемы данных Access приведена на рис. 2.19.

На этой схеме прямоугольники отображают таблицы базы данных с полным списком их полей, а связи показывают, по каким полям осуществляется взаимосвязь таблиц. Имена ключевых полей для наглядности выделены полужирным шрифтом и находятся в верхней части полного списка полей каждой таблицы.

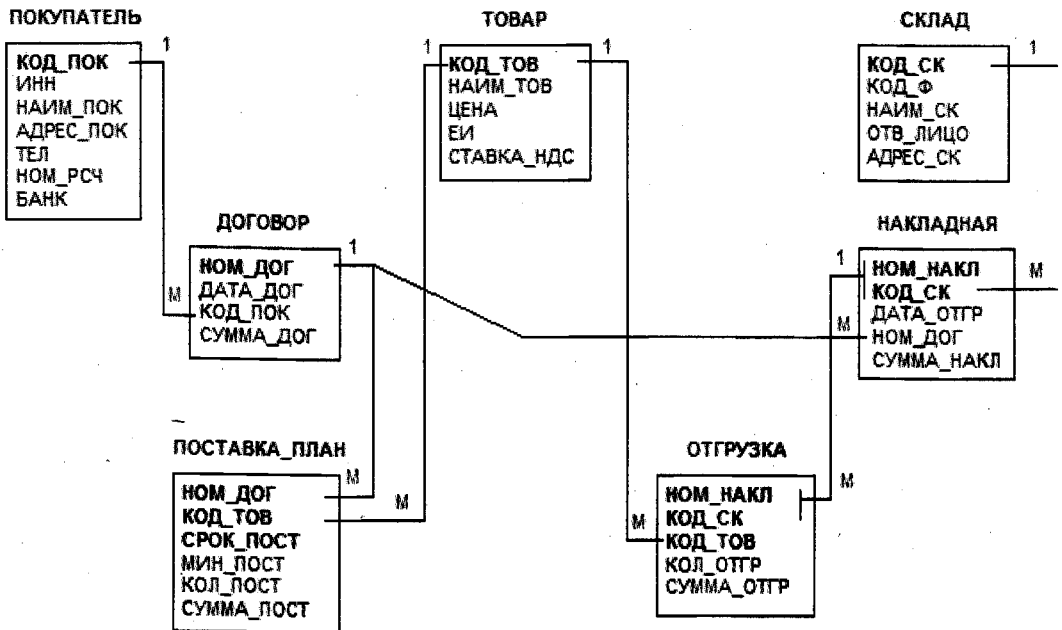


Рис. 2.19. Логическая структура реляционной базы данных предметной области «Поставка товаров»

Контрольные вопросы

1. Что стоит за понятием «сущность предметной области»?
2. Что представляет экземпляр информационного объекта ТОВАР?
3. В каком случае описательный реквизит функционально полно зависит от ключа?
4. Как сказывается нормализация данных на свойствах базы?
5. Должен ли информационный объект, отвечающий требованиям нормализации, иметь уникальный ключ?
6. Могут ли описательные реквизиты, входящие в информационный объект, отвечающий требованиям нормализации, быть зависимыми друг от друга?
7. Может ли описательный реквизит, входящий в информационный объект, отвечающий требованиям нормализации, зависеть от ключа транзитивно?
8. Как устраняется транзитивная зависимость между реквизитами?
9. Верно ли утверждение, что документы являются основным источником данных внемашиной сферы?
10. Перечислите формальные правила выделения информационных объектов.
11. Может ли реквизит одновременно быть ключевым для одних реквизитов и описательным для других?
12. Назовите виды отношений между двумя информационными объектами?

13. В каких отношениях находятся информационные объекты ДОГОВОР и НАКЛАДНАЯ в рассмотренной предметной области?
14. Сколько договоров соответствует одной накладной?
15. В каких отношениях находятся объекты ТОВАР и ОТГРУЗКА?
16. Какой объект является главным в отношении ДОГОВОР и ПОКУПАТЕЛЬ?
17. Как преобразовать много-многозначные отношения ТОВАР и СКЛАД к одно-многозначным?
18. Какой объект играет роль связки между объектами ДОГОВОР и ТОВАР, находящимися в отношении М : N?
19. Какой объект играет роль связки между объектами ТОВАР и НАКЛАДНАЯ, находящимися в отношении М : N?
20. Как образуется ключ в объекте «связка»?
21. По какому полю связываются таблицы, находящиеся в одно-многозначных отношениях?
22. Как можно определить уровень, на котором должен быть размещен объект в канонической модели данных?
23. Сколько таблиц будет в базе данных, построенной на основе нормализованной модели данных предметной области?
24. Может ли повторяться значение первичного ключа в нескольких строках реляционной таблицы?

ОТВЕТЫ

1. Реальный объект, процесс, явление, событие.
2. Совокупность реквизитов, характеризующих конкретный товар.
3. Если одному значению ключа соответствует только одно значение описательного (зависимого) реквизита.
4. Устраняется дублирование описательных данных, обеспечивается их однократный ввод и корректировка, поддержание целостности средствами СУБД.
5. Да.
6. Нет.
7. Нет.
8. Расщеплением информационного объекта на два.
9. Да.
10. Составить перечень реквизитов документа; определить функциональные зависимости между ними; выбрать все зависимые реквизиты с указанием ключа, от которого они зависят функционально полно; объединить реквизиты, зависимые от одинаковых ключевых реквизитов вместе с ключом в один объект.

11. Да.
12. Одно-однозначные, одно-многозначные, много-многозначные.
13. Одно-многозначных.
14. Один.
15. Одно-многозначных.
16. ПОКУПАТЕЛЬ.
17. Через объект «связка» — запас товара на складе с составным ключом КОД_ТОВ + КОД_СК.
18. ПОСТАВКА_ПЛАН.
19. ОТГРУЗКА.
20. Из ключей таблиц, находящихся в отношении $M : N$.
21. По уникальному ключу главной таблицы, в подчиненной таблице это поле не может быть ключом, хотя и может входить в составной ключ.
22. По числу связей в наиболее длинном пути от объектов нулевого уровня к данному объекту.
23. Столько, сколько информационных объектов в ИЛМ данных предметной области.
24. Нет.

ГЛАВА 3



Создание базы данных

Создание новой нормализованной реляционной базы данных Access осуществляется в соответствии с ее структурой, полученной в результате проектирования. Процесс проектирования реляционной базы данных был рассмотрен в *главе 2*. Как там отмечалось, структура реляционной базы данных определяется составом таблиц и их взаимосвязями. Взаимосвязи между двумя таблицами реализуются через ключ связи, входящий в состав полей связываемых таблиц. Напомним, что в нормализованной реляционной базе данных таблицы находятся в отношениях типа «один-ко-многим» или «один-к-одному». Для одно-многозначных отношений в качестве ключа связи, как правило, используется уникальный ключ главной таблицы, а в подчиненной таблице это может быть любое из полей, которое называется *внешним ключом*.



Создание реляционной базы данных начинается с формирования структуры таблиц. При этом определяются состав полей, их имена, тип данных каждого поля, размер поля, ключи, индексы таблицы и другие свойства полей. После определения структуры таблиц создается схема данных, в которой устанавливаются связи между таблицами. Access запоминает и использует эти связи при заполнении таблиц и обработке данных.

При создании базы данных важно задать параметры, в соответствии с которыми Access будет автоматически поддерживать целостность данных. Для этого при определении структуры таблиц должны быть заданы ключевые поля таблиц, указаны ограничения на допустимые значения данных, а при создании схемы данных на основе нормализованных таблиц должны быть заданы параметры поддержания целостности связей базы данных.

Завершается создание базы данных процедурой *загрузки*, т. е. заполнением таблиц конкретными данными. Особое значение имеет технология загрузки взаимосвязанных данных. Удобным инструментом загрузки данных во взаимосвязанные таблицы являются формы ввода/вывода, обеспечивающие интерактивный интерфейс для работы с данными базы. Формы позволяют создать экранный аналог документоисточника, через который можно вводить данные во взаимосвязанные таблицы. Технология загрузки данных из документов с использованием форм будет рассмотрена в *главе 5*. Здесь же рассматривается непосредственный ввод данных в таблицы.

ВНИМАНИЕ!

Пользователь может начинать работу с реляционной базой при любом количестве созданных таблиц еще до создания полной базы, отображающей все объекты модели данных предметной области. База данных может создаваться поэтапно, и в любой момент ее можно дополнять новыми таблицами и вводить связи между таблицами в схему данных, а существующие таблицы могут дополняться новыми полями.

Как уже отмечалось ранее (см. *разд. «Запуск Access» главы 1*), на открывшейся после запуска Access странице (рис. 3.1) можно открыть существующую базу данных или создать новую. Область, предназначенная для открытия файлов базы данных, размещена слева. Здесь отображается список **Последние (Recent)**, демонстрирующий недавно открывавшиеся базы данных. Эти базы могут быть открыты щелчком мыши на них. В конце строки с именем выделенной базы изображается значок кнопки. Щелчком на значке  можно закрепить элемент (базу данных) в списке последних, а щелчком на значке  открепить этот элемент от списка. Ниже размещена ссылка **Открыть другие файлы (Open Other Files)**.

В правой части стартового окна Access размещена область создания базы данных. В ней представлены элементы для создания двух основных объектов Access: новой

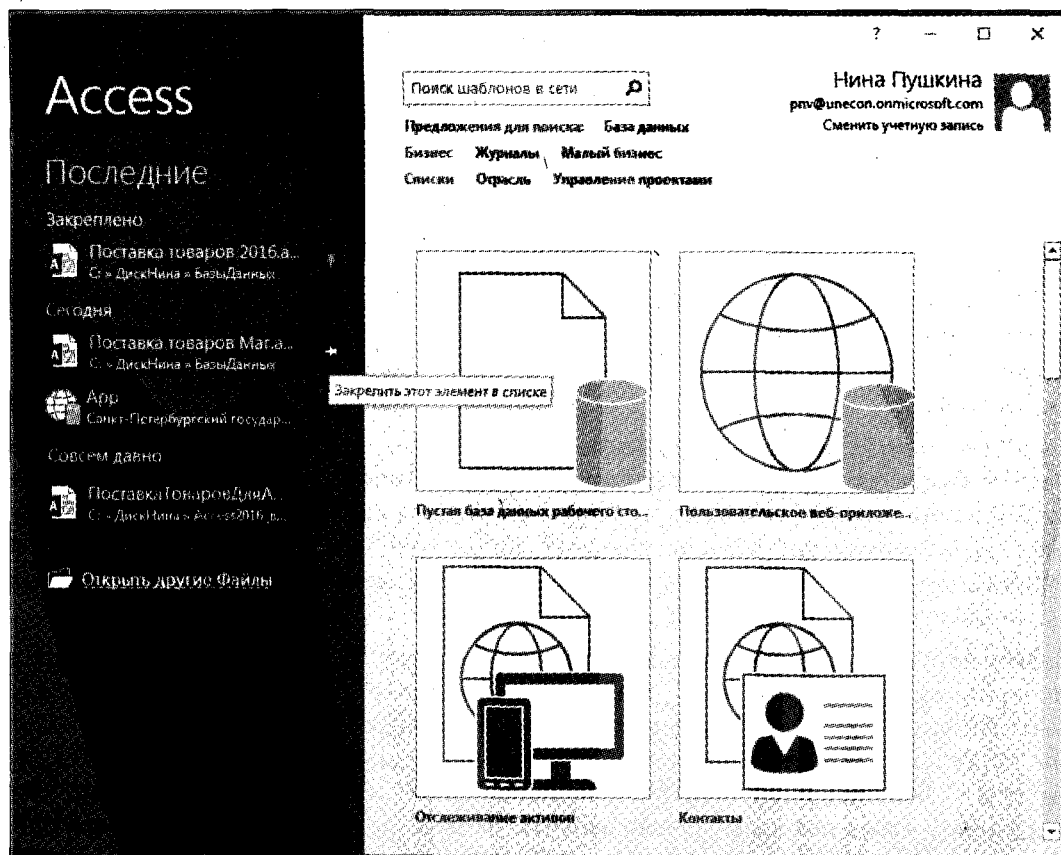


Рис. 3.1. Стартовое окно Access

пустой базы данных рабочего стола и пользовательского веб-приложения. Кроме того, область этой команды содержит многочисленные шаблоны баз данных.

Шаблон — это готовая к использованию база данных, содержащая таблицы, запросы, формы и отчеты, необходимые для выполнения определенной задачи. Эти готовые базы данных позволяют быстро создать приложения, ориентированные на поддержку широкого спектра задач. Такие стандартные приложения можно использовать без какой-либо модификации и настройки, либо, взяв их за основу, приспособить шаблон к своим нуждам и создать новые таблицы, формы, отчеты и другие объекты базы данных.

Шаблоны можно загрузить из Интернета, задав категорию в поле **Поиск шаблонов в сети** (Search for online templates). На основе шаблона можно создать как локальную базу данных рабочего стола, так и веб-приложение Access.

ЗАМЕЧАНИЕ

На рис. 3.1 в правой верхней части окна отображаются сведения из учетной записи, которая была создана при регистрации на сайте SharePoint, что необходимо для создания пользовательского веб-приложения. В списке **Последние** (Recent) отображена база данных App, являющаяся веб-приложением Access 2016.

Создание файла базы данных Access

Если для решения ваших задач использовать шаблон не имеет смысла, можно создать на компьютере базу данных «с нуля». Поскольку Access хранит все таблицы базы данных, а также другие объекты в одном файле, прежде чем приступить к их созданию, необходимо создать файл пустой базы данных.

Для создания файла новой локальной базы данных щелкните в области создания базы данных стартового окна Access на элементе **Пустая база данных рабочего стола** (см. рис. 3.1) — пустая база данных рабочего стола создается на вашем компьютере.

В открывшемся окне (рис. 3.2) введите имя файла в поле **Имя файла** — например, **Поставка товаров 2016**.

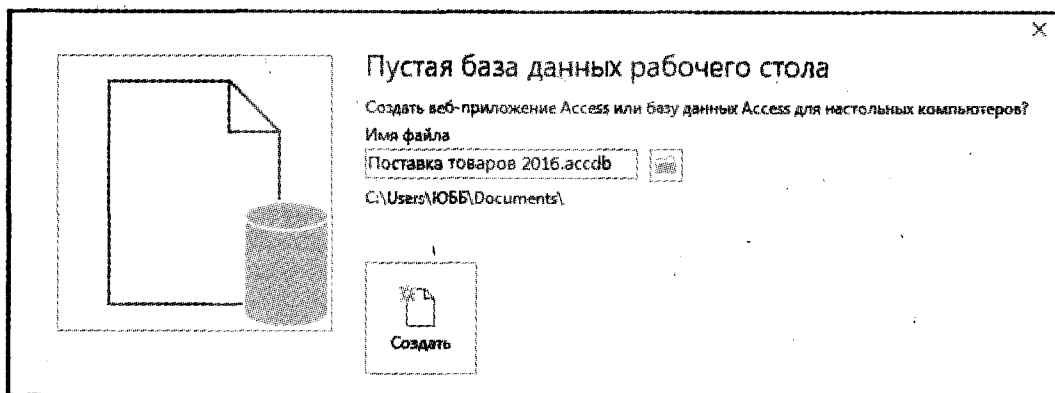



Рис. 3.2. Определение имени и местоположения файла новой базы данных

Задавая имя файла базы данных, следует иметь в виду, что оно не должно содержать символов: \ / : * ? « < > |, а его предельная длина не должна превышать 215 символов, включая пробелы.

Под именем отображается имя папки, в которой по умолчанию сохранится файл базы данных. Если надо сохранить файл в другой папке, щелкните на значке  Поиск расположения для размещения базы данных (Browse for a location to put your database) справа от поля ввода имени файла базы данных и в окне **Файл новой базы данных** (File New Database) откройте нужную папку (рис. 3.3).

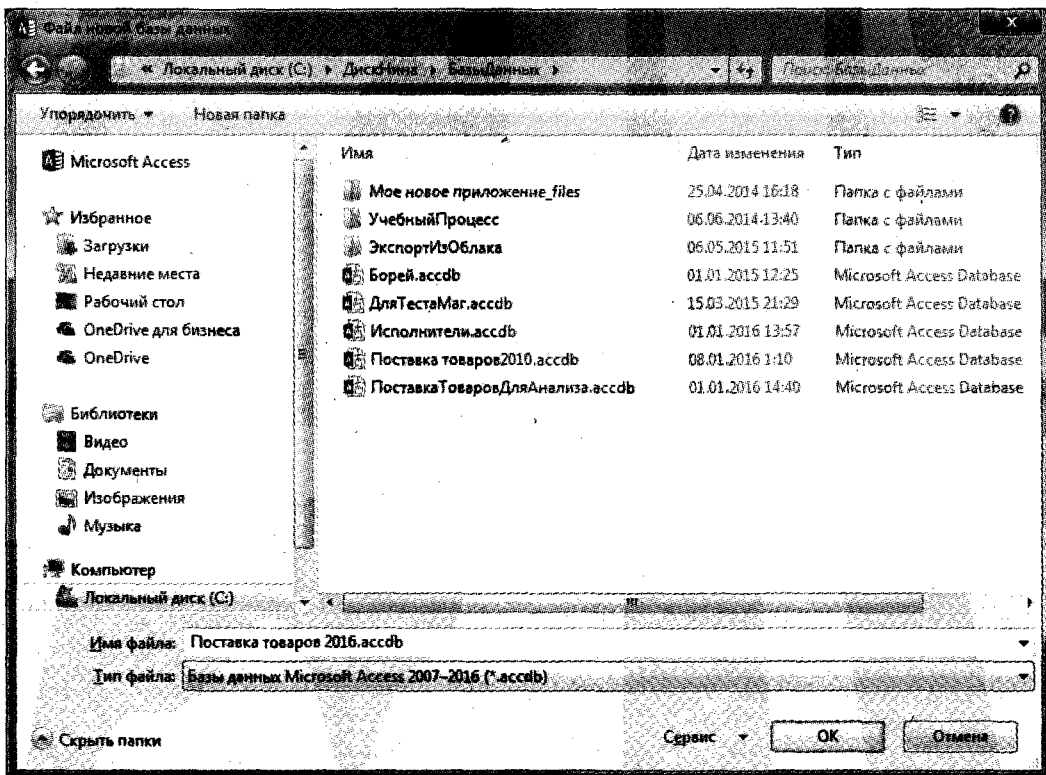


Рис. 3.3. Окно выбора местоположения, имени и формата файла новой базы данных

ЗАМЕЧАНИЕ

Для смены папки **Документы** (Documents), используемой по умолчанию для сохранения файлов новых баз данных, на странице, открываемой выбором значка **Файл** (File), выполните команду **Параметры** (Options) и в открывшемся диалоговом окне на вкладке **Общие** (Popular) в разделе **Создание баз данных** (Creating databases) в поле **Рабочий каталог** (Default Database Folder), используя кнопку **Обзор** (Browse), выберите путь к папке, в которой предполагается хранить новые базы данных.

В окне **Файл новой базы данных** (File New Database) в поле **Тип файла** (Save as type) выберите формат создаваемой базы данных. По умолчанию формат файла имеет значение **База данных Microsoft Office Access 2007–2016 (.accdb)** (Microsoft

Office Access 2007–2016 Databases (.accdb)). В Access 2016 сохранился формат базы данных, впервые принятый в Access 2007, и базы данных в формате Access 2007–2016 сохраняются в файлах с расширением accdb. В предыдущих версиях базы данных сохранялись в файлах с расширением mdb.

ЗАМЕЧАНИЕ

Для изменения формата, выбираемого для новой базы данных по умолчанию, на странице, открываемой выбором значка **Файл** (File), выполните команду **Параметры** (Options) и в открывшемся диалоговом окне на вкладке **Общие** (Popular) в разделе **Создание баз данных** (Creating databases) в поле **Формат файла по умолчанию для пустой базы данных** (Default file format) выберите нужный формат.

Закончив выбор, в окне **Файл новой базы данных** (File New Database), нажмите кнопку **ОК**. Щелчком по кнопке **Создать** (Create) (см. рис. 3.2) завершите процесс создания пустого файла новой базы данных. В результате откроется окно созданной базы данных с пустой таблицей с именем **Таблица1** (Table1) в режиме таблицы (рис. 3.4). Курсор при этом находится в первой пустой ячейке столбца **Щелкните для добавления** (Click to Add). Теперь можно приступить к созданию этой таблицы и других объектов новой оригинальной базы данных.

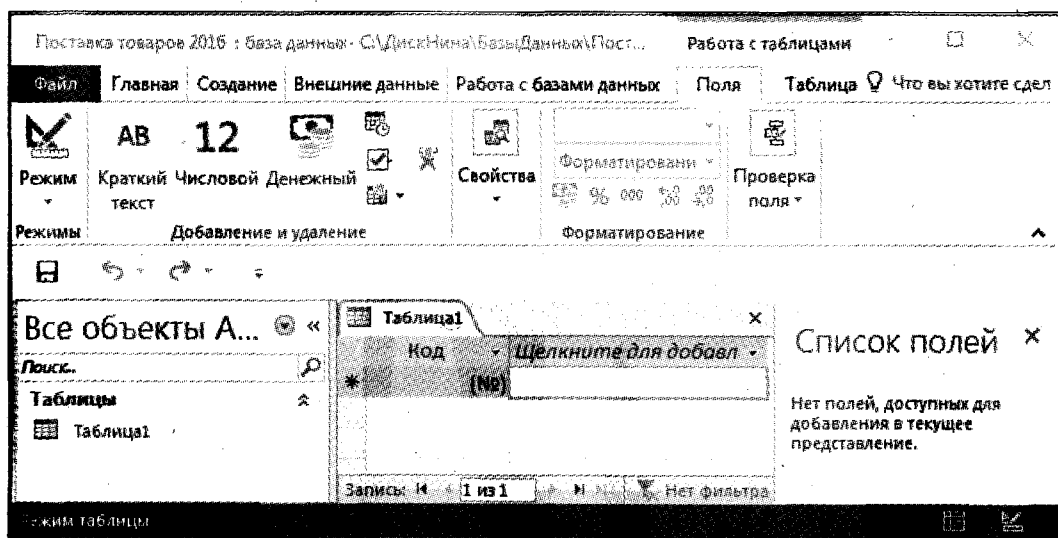


Рис. 3.4. Окно пустой базы данных

Окно Access

Создание новой пустой базы данных приводит к открытию окна Access, в заголовке которого записано: **<Имя БД>: база данных- <полное имя базы данных с указанием местоположения файла и расширения>** (Формат файлов Access 2007–2016). Реквизит **<Имя БД>** соответствует заданному в поле **Имя файла** (см. рис. 3.4).

Ниже заголовка следуют основные компоненты пользовательского интерфейса Access 2016, которые формируют среду работы с базой данных (см. также рис. 1.11).

- **Строка вкладок** определяет доступный функционал Access. Вкладки объединяют логически связанные команды, ориентированные на задачу. С каждой вкладкой связана группа команд, представляемая на соответствующей области ленты.

Исключение представляет собой расположенный в начале строки вкладок значок **Файл (File)**. Выбор этого значка открывает набор команд, отображаемых на отдельной странице (рис. 3.5) и применимых ко всей базе данных. В списке слева представлены такие команды, как **Создать (New)**, **Открыть (Open)**, **Сохранить (Save)**, **Заккрыть (Close)**. Для выбранной команды **Сведения (Info)** отображаются доступные действия: **Сжать и восстановить (Compact & Repair)**, **Зашифровать с использованием пароля (Encrypt with Password)**. Здесь же имеется ссылка для просмотра и изменения свойств базы данных. Стрелка в начале списка позволяет вернуться на предыдущее окно.

В Access 2016 строка вкладок дополнилась полем помощника **Что вы хотите сделать?** С помощью этого поля очень просто перейти к выполнению нужных

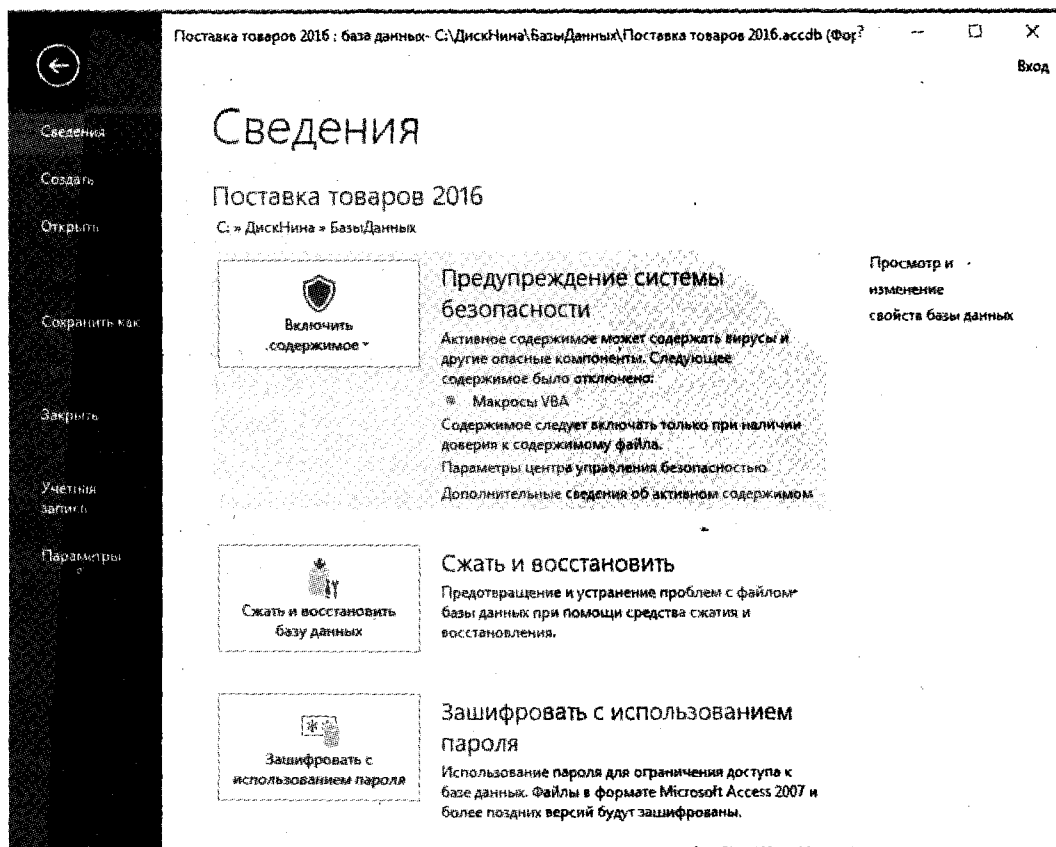


Рис. 3.5. Представление команд, открываемых выбором значка **Файл**

задач. Щелчком на этой вкладке открывается возможность выбрать из выпадающего списка или ввести название команды или действия, к выполнению которого вы хотите перейти. Например, наберите фразу *Конструирование таблицы*, и откроется список команд, которые позволят решить задачу конструирования таблицы и получить справку по этой теме. Ввод имени любой команды обеспечит быстрый переход к ее выполнению.

ЗАМЕЧАНИЕ

Поле помощника отсутствует на ленте при разработке веб-приложений Access.

- **Лента** — это полоса под строкой вкладок, содержащая группы команд выбранной вкладки.
- **Область навигации** — область в левой части окна Access, предназначенная для отображения и работы со всеми объектами открытой базы данных.

Лента


Лента отображает команды активной вкладки и является основным элементом интерфейса в Access, содержащим все инструменты для работы с объектами базы данных (см. также *разд. «Лента» главы 1*).

Четыре *стандартные вкладки*: **Главная (Home)**, **Создание (Create)**, **Внешние данные (External Data)** и **Работа с базами данных (Database Tools)** — отображаются в окне Access и не зависят от выбранного объекта базы данных и режима работы с ним. Команды на ленте каждой вкладки организованы в несколько групп, соответствующих подзадачам вкладки. Некоторые команды могут открывать другие элементы интерфейса — например, коллекции или диалоговые окна.

Остальные виды вкладок ленты являются *контекстно-зависимыми*, соответствуют текущему активному объекту и режиму работы. Набор контекстных вкладок автоматически появляется в интерфейсе наряду со стандартными вкладками по мере необходимости. Названия соответствующих контекстных вкладок выделены цветом. Контекстные вкладки предоставляют команды для работы с выбранным объектом.

В ряде случаев стандартный набор вкладок при переключении на конкретный вид или режим работы заменяется так называемыми *программными вкладками*, например, при предварительном просмотре отчета.

Панель быстрого доступа

Панель быстрого доступа — небольшая панель инструментов, на которую можно добавить самые нужные команды (см. также *разд. «Пользовательская панель инструментов быстрого доступа» главы 1*). Она может размещаться под лентой или над ней и обеспечивает доступ к командам одним нажатием кнопки. Первоначально панель содержит лишь команды **Сохранить (Save)**, **Отменить (Undo)**, **Вернуть (Redo)** и  **Настроить панель быстрого доступа (Customize Quick Access Toolbar)**.

Панель быстрого доступа легко дополнить любыми командами и даже целыми их группами. Для настройки панели можно использовать различные способы. Например, воспользоваться соответствующей кнопкой на панели быстрого доступа и выбрать некоторые команды из ее выпадающего меню. При выборе опции **Другие команды** (More Commands) открывается окно **Параметры Access** (Access Options) с отображенной вкладкой **Панель быстрого доступа** (Quick Access Toolbar), с помощью которой любую команду можно добавить на панель или удалить с нее. Кроме того, добавить на панель быстрого доступа любую команду ленты можно через контекстное меню этой команды. Удалить добавленную команду с панели быстрого доступа можно, выполнив команду **Удалить с панели быстрого доступа** (Remove from Quick Access Toolbar) контекстного меню команды, размещенной на этой панели.

Область навигации

Область навигации (рис. 3.6) служит для отображения объектов открытой базы данных и выполнения работ с ними (см. также *разд. «Область навигации» главы 1*). Действия, которые можно выполнить с объектом, представлены как командами ленты, так и командами его контекстного меню. Команды контекстного меню позволяют открывать объект в нужном режиме, удалять, переименовывать, экспортировать, просматривать его свойства и выполнять некоторые другие действия.

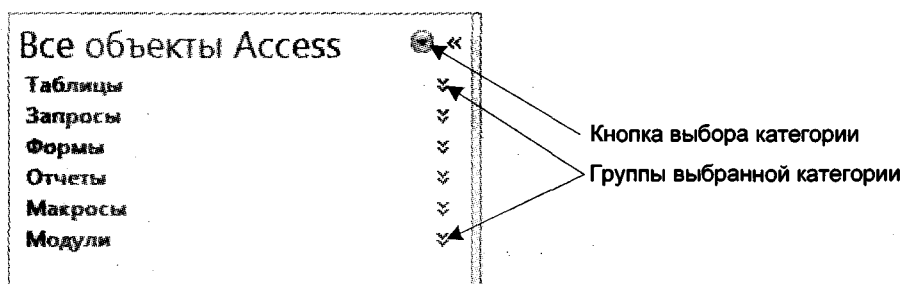


Рис. 3.6. Область навигации со встроенными группами объектов

Все объекты базы данных в области навигации делятся на категории, которые, в свою очередь, содержат группы. На рис. 3.6 показаны группы встроенной категории **Тип объекта** (Object Type), для которой выбран фильтр **Все объекты Access** (All Access Objects). Эта категория группирует компоненты базы данных по их типам: таблицы помещаются в группу **Таблицы** (Tables), формы — в группу **Формы** (Forms) и т. д.

Область навигации можно использовать для упорядочения объектов по типу, дате создания, дате изменения, для отображения таблиц и связанных с ними объектов, а также для создания пользовательских категорий и групп.

Пользовательские категории и группы объектов базы данных (рис. 3.7) позволяют объединить объекты, например, по функциональному назначению, что облегчает пользователю работу с базой данных, реализуя некоторый пользовательский интерфейс.

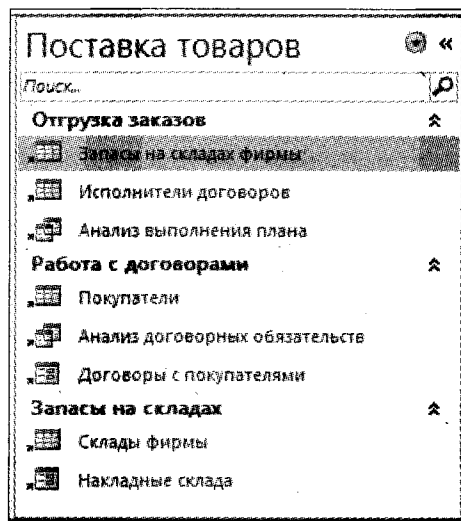


Рис. 3.7. Область навигации с пользовательской категорией **Поставка товаров** и группами объектов

Включить в область навигации новую категорию можно в окне **Параметры навигации** (Navigation Options), открываемом соответствующей командой контекстного меню при щелчке правой кнопкой мыши в пустом пространстве нижней части области навигации. Для добавления категории воспользуйтесь кнопкой **Добавить элемент** (Add Item). Соответствующие кнопки позволят переименовать или удалить категорию. В этом же окне можно добавить, переименовать или удалить группу из категории (рис. 3.8).

Для добавления объектов в группу следует закрыть окно **Параметры навигации** (Navigation Options), в области навигации перейти в созданную категорию, например, **Поставка товаров**, в фильтре по группам выбрать **Показать все**. Далее в группе **Не назначенные объекты** (Unassigned Objects) выбрать нужный объект и по контекстному меню включить его в нужную группу. Если группа не была создана к этому моменту, можно создать ее командой этого же контекстного меню.

После того как группы наполнены объектами, можно в созданной категории работать с объектами всех групп или применить фильтр по группам и работать с объектами выбранной группы. Для того чтобы в категории не отображалась группа **Не назначенные объекты**, следует в окне **Параметры навигации** снять флажок с соответствующей строки.

Поле области навигации **Поиск** (Search) позволяет произвести быстрый поиск объектов в больших базах данных. По мере ввода текста группы, которые не содержат объектов, удовлетворяющих условию поиска, сворачиваются. Чтобы прекратить поиск и восстановить свернутые группы, следует нажать кнопку **Очистить строку поиска** (Clear Search String), расположенную в конце строки **Поиск** (Search).

Для отображения поля **Поиск** (Search) щелкните правой кнопкой мыши в пустом пространстве области навигации и в контекстном меню выберите пункт **Строка поиска** (Search Bar).

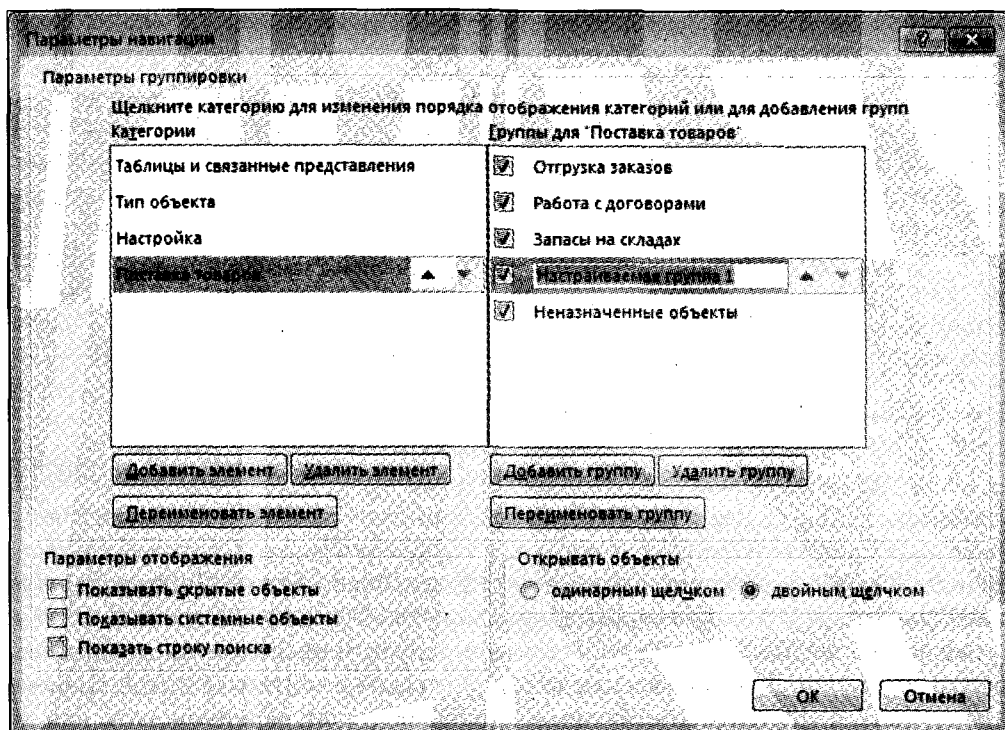


Рис. 3.8. Окно Параметры навигации с пользовательской категорией Поставка товаров и ее группами

Объекты в каждой группе могут отображаться с разной степенью подробности. Например, выбор в контекстном меню группы команды **Просмотр | Сведения** (View By | Details) позволяет отобразить для каждого объекта группы дату его создания и изменения.

Область навигации нельзя скрыть за другими окнами, но можно свернуть, воспользовавшись кнопкой « **Открыть/закрыть границу области переходов** (Shutter Bar Open/Close Button), размещенной в ее правом верхнем углу.

Область документов

Каждый открываемый в любом режиме объект базы данных отображается в области документов. Предусмотрено два вида представления объекта в области документов: **Перекрывание окон** (Overlapping Windows) и **Вкладки** (Tabbed Documents).

Перекрывающиеся окна объектов могут занимать всю область документов, принимать любые размеры или сворачиваться до размеров заголовков, отображаемых внизу области документов. Для перехода к другому документу достаточно щелчка мыши на нем. Если не все документы видны в окне, для перехода к другому документу можно использовать команду **Перейти в другое окно** (Switch Windows) (рис. 3.9), размещенную в группе **Окно** (Window) на вкладке ленты **Главная** (Home).

Начиная с Access 2007, открытые объекты базы данных могут размещаться на различных вкладках области документов (рис. 3.10), а для переключения между открытыми объектами служат ярлычки их вкладок (см. также *разд. «Вкладки документов» главы 1*). При использовании вкладок каждый объект занимает всю область документов и исключается возможность одновременного просмотра нескольких объектов. Команда **Перейти в другое окно** (Switch Windows) делается при этом недоступной.

Переход из режима использования вкладок документов к режиму перекрытия окон и обратно осуществляется путем настройки параметров Access следующим образом. Щелкните на значке **Файл** (File), а затем на открывшейся странице выполните команду **Параметры** (Options). В левой области диалогового окна **Параметры Access** выберите опцию **Текущая база данных** (Current Database). В разделе **Параметры приложения** (Application Options) в группе **Параметры окна документа** (Document Window Options) выберите **Перекрытие окон** (Overlapping Windows) или **Вкладки** (Tabbed Documents). При выборе представления объектов на вкладках автоматически устанавливается флажок **Вкладки документов** (Tabbed Documents). При снятии этого флажка вкладки с названиями документов пропадают, и в области документов отображается один открытый объект.

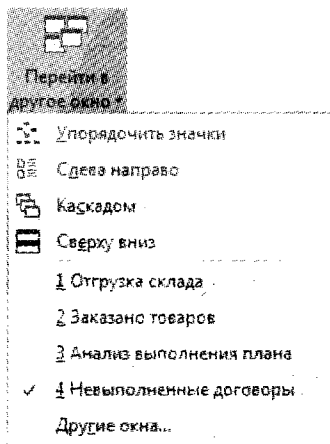


Рис. 3.9. Список команд кнопки **Перейти в другое окно** и открытых документов

ИСПОЛНИТЕЛЬ					
План		Факт		ДОГОВОР	
Номер дог	Дата заключ	Код пошу	Сумма по догов	Код исполн	Щелкните для
D111	11.01.2013	П001	6 414 000,00р.	6	
D222	05.02.2013	П001	152 280,00р.	1	
D333	01.01.2013	П002	74 920,00р.	3	
D555	12.11.2013	П002	9 352,00р.	3	
D777	14.06.2013	П001	1 579 920,00р.	1	
D888	23.05.2013	П002	30 000,00р.	6	
D999	12.09.2013	П002	450 000,00р.	7	
*			0,00р.		

Запись: 14 из 8

Рис. 3.10. Пример набора вкладок документов. Активна вкладка с таблицей **ДОГОВОР**

Чтобы изменения этих параметров вступили в силу, необходимо закрыть и снова открыть базу данных.

ЗАМЕЧАНИЕ

Параметр окна документов **Вкладки** относится к конкретной базе данных и должен устанавливаться отдельно для каждой базы данных. В новых базах данных, созданных в формате Access 2007/2016, вкладки в окне документов отображаются по умолчанию. В базах данных, созданных в предыдущих версиях Access, по умолчанию используются перекрывающиеся окна.

Строка состояния

Вдоль нижней границы окна Access размещена строка состояния, отображающая кнопки, с помощью которых удобно переключаться в различные режимы работы с активным объектом (см. также разд. «*Строка состояния*» главы 1).

Если в окне не отображается строка состояния, откройте окно параметров Access, в его левой области выберите опцию **Текущая база данных** (Current Database) и в разделе **Параметры приложений** (Application Options) установите флажок **Строка состояния** (Status Bar).

Создание таблицы базы данных

Как уже говорилось ранее, таблица содержит данные об одном информационном объекте или сущности — например, сведения о покупателях или товарах. Каждая запись (строка) в таблице включает данные об одном экземпляре сущности — например, о конкретном покупателе. Запись состоит из полей (столбцов), каждое из которых включает один из атрибутов сущности, — таких как наименование, ИНН, адрес и телефон. Для однозначной идентификации записи в таблице определяется уникальный простой или составной ключ. База данных может содержать множество таблиц. Разработка базы данных начинается с создания таблиц.

Создание таблицы БД состоит из двух этапов. На первом этапе определяется ее структура: состав полей, их имена, тип данных каждого поля, размер поля, ключи, индексы таблицы и другие свойства полей. На втором этапе производится создание записей таблицы путем заполнения ее полей данными.

В Access создание таблицы может быть выполнено в одном из двух режимов:

- в *режиме конструктора* (Design View), позволяющем максимально полно определить структуру таблицы;
- в *режиме таблицы* (Datasheet View), предназначенном, прежде всего, для создания, просмотра, поиска, корректировки ее записей и, кроме того, реализующем функции, обеспечивающие определение структуры таблицы.

Рассмотрим основные параметры полей таблицы, которые могут быть заданы, вне зависимости от способа создания таблицы.

Имя поля и тип данных

Для определения поля таблицы обязательно задаются **Имя поля** (Field Name) и **Тип данных** (Data Type).

□ **Имя поля** (Field Name) — каждое поле в таблице должно иметь уникальное имя, удовлетворяющее соглашениям об именах объектов в Access. Оно может являться комбинацией из букв, цифр, пробелов и специальных символов, за исключением точки (.), восклицательного знака (!), надстрочного знака (^) и квадратных скобок ([]). Имя не может начинаться с пробела и содержать управляющие символы с кодами ASCII от 0 до 31. Максимальная длина имени — 64 символа.

Имена полей (а также элементов управления и объектов) могут содержать пробелы, но во избежание конфликтов имен в Microsoft Visual Basic для приложений целесообразно не использовать пробелы в каких бы то ни было именах Access.

□ **Тип данных** (Data Type) — тип данных определяется значениями, которые предполагается хранить в поле, и операциями, которые будут выполняться с этими значениями. В Access допускается использование двенадцати типов данных. Рассмотрим вкратце назначение и допустимые размеры данных всех типов, которые могут назначаться полям таблицы в Access.

- **Короткий текст** (Short Text) — используется для хранения текста или комбинаций алфавитно-цифровых знаков, не применяемых в расчетах (например, код товара). Максимальная длина поля — 255 знаков.
- **Длинный текст** (Long Text) (ранее назывался **Поле МЕМО** (Memo)) — используется для хранения обычного текста или комбинаций алфавитно-цифровых знаков длиной более 255 знаков. Поля с этим типом данных в базах данных формата Access 2007/2016 поддерживают форматирование текста. Это единственный в Access тип данных, обеспечивающий встроенную поддержку отображения и хранения форматированного текста. Максимальный размер поля — 1 Гбайт знаков, хотя через элементы управления в формах и отчетах могут вводиться и отображаться только первые 64 000 символов.
- **Числовой** (Number) — служит для хранения числовых значений (целых или дробных), предназначенных для вычислений. Исключением здесь являются денежные значения, для которых используется отдельный тип данных **Денежный**. Размер поля — 1, 2, 4, 8 и 12 байтов или 16 байтов (если используется для кода репликации) — зависит от типа чисел, вводимых в поле. Для повышения производительности следует задавать минимально достаточный размер поля.
- **Дата/время** (Date/Time) — используется для хранения значений даты и времени в виде 8-байтовых чисел двойной точности с плавающей запятой. Целая часть значения, расположенная слева от десятичной запятой, представляет собой дату. Дробная часть, расположенная справа от десятичной запятой, —

это время. Хранение значений даты и времени в числовом формате позволяет выполнять различные вычисления с этими данными.

- **Денежный (Currency)** — используется для хранения денежных значений с числом знаков до 15 слева от десятичной запятой и с точностью до четырех знаков после десятичной запятой. Этот тип данных применяется в тех случаях, когда значения не должны округляться при вычислениях. Каждое значение поля такого типа занимает 8 байтов.
- **Счетчик (AutoNumber)** — используется для уникальных числовых 4-байтовых значений, которые автоматически вводит Access при добавлении записи. Вводимые числа могут последовательно увеличиваться на указанное приращение или выбираться случайно. Обычно используются для определения суррогатного первичного ключа таблицы.
- **Логический (Yes/No)** — применяется для хранения логических значений, которые могут содержать одно из двух значений: Да/Нет, Истина/Ложь или Вкл/Выкл. Используется 1 для значений Да и 0 для значений Нет. Размер поля равен 1 биту.
- **Поле объекта OLE (OLE Object)** — используется для прикрепления к записям изображений, диаграмм и других объектов ActiveX из приложений Windows, которые затем отображаются в элементах управления форм или отчетов, связанных с этим полем таблицы. Поля объекта OLE позволяют вкладывать в одну запись только один файл. Рекомендуется вместо поля объекта OLE использовать поле с типом данных **Вложение (Attachment)**, которое позволяет вкладывать в одну запись несколько файлов и поддерживает больше типов файлов.

Чтобы в Access просматривать эти изображения, необходимо, чтобы на компьютере, работающем с базой данных, был зарегистрирован OLE-сервер (программа, поддерживающая этот тип файлов). Если для используемого типа файлов OLE-сервер не зарегистрирован, отображается значок поврежденного изображения.

- **Гиперссылка (Hyperlink)** — применяется для хранения ссылок на веб-узлы (URL-адреса), на узлы или файлы интрасети, или локальной сети (UNC-адреса — записи пути стандартного формата), а также на узлы или файлы локального компьютера. Кроме того, можно использовать ссылку на объекты Access, хранящиеся в базе данных. Может содержать до 8192 знаков (каждая часть гиперссылки до 2048 знаков).
- **Вложение (Attachment)** — используется для вложения в поле записи файлов изображений, электронных таблиц, документов, диаграмм и других файлов поддерживаемых типов точно так же, как в сообщения электронной почты. Вложенные файлы можно просматривать и редактировать в соответствии с заданными для поля параметрами. Эти поля не имеют ограничений, связанных с отсутствием зарегистрированных OLE-серверов. Они более рационально используют место для хранения, чем поля с типом данных **Поле объекта**

OLE (OLE Object). Максимальная длина поля для сжатых вложений — 2 Гбайт, для несжатых — примерно 700 Кбайт в зависимости от степени возможного сжатия вложения.

- **Вычисляемый (Calculated)** — предназначен для создания вычисляемых полей: числовых, текстовых, денежных, дата/время, логических. Значение вычисляемого поля определяется выражением, записанным в поле и использующим другие поля текущей записи, некоторые встроенные функции и константы, связанные арифметическими, логическими или строковыми операторами.
- **Мастер подстановок (Lookup Wizard) или Подстановка и отношения** — вызывает мастер подстановок, с помощью которого можно создать поле, позволяющее выбрать значения из списка, построенного на основе значений поля другой таблицы, запроса или фиксированного набора значений. Такое поле отображается как поле со списком. Если список построен на основе поля таблицы или запроса, тип данных и размер создаваемого поля определяется типом данных и размером привязанного столбца, если на основе набора значений — размером текстового поля, содержащего значение. Кроме того, мастер подстановок позволяет определить связь таблиц и включить проверку связной целостности данных.

Общие свойства поля

Основные свойства задаются для каждого поля и зависят от выбранного типа данных. Если открыть таблицу в режиме конструктора, то весь набор свойств выбранного поля будет представлен в нижней части окна на двух вкладках: **Общие (General)** и **Подстановка (Lookup)**. Приведем свойства полей, наиболее важные на первом этапе изучения баз данных.

- **Размер поля (Field Size)** — позволяет для текстового и числового поля уточнить тип данных или размер, задает максимальный размер данных, сохраняемых в поле. Для поля с типом данных **Короткий текст (Short Text)** задается размер от 1 до 255 знаков. Для поля с типом данных **Числовой (Number)** можно задать:
 - **Байт (Byte)** — для целых чисел от 0 до 255, длина поля 1 байт;
 - **Целое (Integer)** — для целых чисел от -32 768 до +32 767, занимает 2 байта;
 - **Длинное целое (Long Integer)** — для целых чисел от -2 147 483 648 до +2 147 483 647, занимает 4 байта;
 - **Одинарное с плавающей точкой (Single)** — для чисел от $-3,4 \times 10^{38}$ до $+3,4 \times 10^{38}$ с точностью до 7 знаков, занимает 4 байта;
 - **Двойное с плавающей точкой (Double)** — для чисел от $-1,797 \times 10^{308}$ до $+1,797 \times 10^{308}$ с числом отображаемых десятичных знаков до 15, занимает 8 байтов;
 - **Действительное (Decimal)** — для целых чисел от -10^{38} до $+10^{38}$ с числом отображаемых десятичных знаков до 28, занимает 12 байтов;

- **Код репликации (Replication ID)** — глобальный уникальный идентификатор (Globally unique identifier, GUID), занимает 16 байтов. Эти длинные генерируемые случайным образом значения обеспечивают малую вероятность их совпадения. Поля такого типа используются Access для создания системных уникальных идентификаторов реплик, наборов реплик, таблиц, записей и других объектов при репликации баз данных. Могут быть использованы в приложениях пользователя для идентификации строк таблицы — например, для идентификации товаров.

Для поля с типом данных **Счетчик (AutoNumber)** можно задать:

- **Длинное целое (Long Integer)** — 4 байта;
- **Код репликации (Replication ID)** — 16 байтов.

Рекомендуется задавать минимально допустимый размер поля, который понадобится для сохраняемых значений, т. к. сохранение таких полей требует меньше памяти, и обработка данных меньшего размера выполняется быстрее.

Изменения в данных, которые происходят вследствие изменения свойства **Размер поля (Field Size)**, нельзя отменить после выполнения сохранения.

- **Формат поля (Format)** — представляет собой формат отображения выбранного типа данных при выводе их на экран или печать в режиме таблицы, в форме или отчете. В Access определены встроенные стандартные форматы отображения для полей с такими типами данных, как **Числовой (Number)**, **Дата/время (Date/Time)**, **Логический (Yes/No)** и **Денежный (Currency)**. Ряд этих форматов совпадает с настройкой региональных форматов, определяемых в окне **Язык и региональные стандарты** Панели управления Windows. Пользователь может создать собственный формат для всех типов данных, кроме **Поле объекта OLE (OLE Object)**, с помощью символов форматирования.
- **Число десятичных знаков (Decimal Places)** — задает для числового и денежного типов данных количество знаков после запятой. Можно задать число от 0 до 15. По умолчанию (значение **Авто (Auto)**) это число определяется установкой в свойстве **Формат поля (Format)**. Следует иметь в виду, что установка этого свойства не действует, если свойство **Формат поля (Format)** не установлено или выбрано значение **Основной (General Number)**. Свойство **Число десятичных знаков (Decimal Places)** влияет только на количество десятичных знаков, отображаемых на экране, и не влияет на число сохраняемых десятичных знаков. Для изменения числа сохраняемых знаков нужно изменить свойство **Размер поля (Field Size)**.
- **Подпись (Caption)** поля задает текст, который выводится в таблицах, формах, отчетах.
- **Описание (Description)** — краткий пользовательский комментарий к полю.
- **Значение по умолчанию (Default Value)** — определяет текст или выражение, значение которого автоматически вводится в поле при создании новой записи. Например, если задана функция `=Now()`, то в поле введется текущая дата и время. При добавлении записи в таблицу можно оставить значение, введенное по умол-

чанию, или ввести другое. Свойство **Значение по умолчанию** (Default Value) используется только при создании новой записи. Максимальная длина значения свойства составляет 255 знаков. Свойство не определено для полей с типами данных: **Счетчик** (AutoNumber), **Поле объекта OLE** (OLE Object), **Вложение** (Attachment).

- **Правило проверки** (Validation Rule) — позволяет осуществлять контроль ввода, задает ограничения на вводимые значения, при нарушении условий запрещает ввод и выводит текст, заданный свойством **Сообщение об ошибке** (ValidationText).
- **Сообщение об ошибке** (ValidationText) — задает текст сообщения, выводимый на экран при нарушении ограничений, заданных свойством **Правило проверки** (Validation Rule).

ВНИМАНИЕ!

Элементы управления, созданные в формах или отчетах на основе поля таблицы, наследуют установленные для этого поля свойства. Благодаря этому не понадобится определять свойства индивидуально для каждого связанного с полем элемента управления.

Свойства вкладки *Подстановка*

В окне конструктора таблиц на вкладке **Подстановка** (Lookup) задается свойство **Тип элемента управления** (Display Control). Это свойство определяет, будет ли отображаться поле в таблице и в элементе управления формы в виде **Поля** (Text Box), **Списка** (List Box) или **Поля со списком** (Combo Box).

Если для поля выбран тип элемента управления **Список** (List Box) или **Поле со списком** (Combo Box), на вкладке **Подстановка** (Lookup) появляются дополнительные свойства, которые определяют источник данных для строк списка и ряд других характеристик списка. Источник данных определяет, откуда брать значения для столбца подстановок. В качестве источника данных для поля со списком выбирается таблица или запрос, с которым осуществляется постоянная связь, что обеспечивает актуальное состояние списка, для списка строится список конкретных значений, разделенных точкой с запятой. Следует заметить, что не все типы данных позволяют использовать поле со списком.

Определение первичного ключа

Каждая таблица в реляционной базе данных должна иметь уникальный (первичный) ключ, однозначно определяющий каждую запись в таблице. Это позволяет быстро найти нужную запись. Кроме того, ключевое поле служит для связи таблиц. Это поле должно содержать уникальные значения — такие как коды или инвентарные номера, и не может содержать значения **Null**. Если для таблицы определен первичный ключ, то Access предотвращает дублирование значений ключа и отсутствие значения в его полях. Ключ может быть простым или составным, включающим несколько полей (до 10).

Для задания первичного ключа таблицы, его изменения или удаления необходимо использовать режим конструктора.

Для определения ключа выделяются поля, составляющие ключ, и на ленте **Работа с таблицами | Конструктор** (Table Tools | Design) в группе **Сервис** (Tools) нажимается кнопка **Ключевое поле** (Primary Key) или выполняется команда контекстного меню поля **Ключевое поле** (Primary Key).

В результате для ключевого поля автоматически строится уникальный индекс с именем **PrimaryKey**, ускоряющий поиск нужной записи по значению ключевого поля. Кроме того, автоматически строятся неуникальные индексы по вторичным ключам таблицы, определяющим связь с главной таблицей. В этом можно убедиться, просмотрев информацию об индексах таблицы. Индекс первичного ключа всегда уникален и не допускает пустых полей в записях. Окно **Индексы** (Indexes) вызывается щелчком на кнопке просмотра и редактирования индексов **Индексы** (Indexes) на ленте **Работа с таблицами | Конструктор** (Table Tools | Design).

ЗАМЕЧАНИЕ

Индексы предназначены для осуществления быстрого поиска требуемых записей в больших таблицах базы данных по значению первичного или вторичного ключа. Индексы можно представить как некоторые внутренние служебные таблицы, содержащие два столбца: первый содержит значение индексируемого поля, а второй — адреса записей, имеющих это значение в индексируемом поле. В индексной таблице производится упорядочение строк по значениям индексируемого поля, и это позволяет использовать методы быстрого поиска строки с заданным значением индексного поля, вместо последовательного просмотра строк таблицы. По адресу, содержащемуся в найденной строке индексной таблицы, осуществляется прямой доступ к искомой записи данных. Допускается не более 32 индексов на таблицу.

На рис. 3.11 представлено окно **Индексы** для таблицы **ДОГОВОР**, подчиненной таблицам **ПОКУПАТЕЛЬ** и **ИСПОЛНИТЕЛЬ** и связанной с ними полем **код_пок** и **Код исполнителя**. Первым представлен уникальный индекс с именем **PrimaryKey**, построенный по ключевому полю **НОМ_ДОГ** — номер договора. В двух следующих строках представлены неуникальные индексы, построенные по вторичным ключам таблицы: **код_пок** и **Код исполнителя**.

Индекс: ДОГОВОР		
Индекс	Имя поля	Порядок сортировки
PrimaryKey	НОМ_ДОГ	По возрастанию
ДОГОВОРКОД_ПОК	КОД_ПОК	По возрастанию
ДОГОВОРКОД_СОТР	Код исполните	По возрастанию

Свойства индекса	
Ключевое поле	Да
Уникальный индекс	Да
Пропуск пустых полей	Нет

Имя данного индекса. Каждый индекс может содержать до 10 полей.

Рис. 3.11. Автоматически созданные индексы таблицы **ДОГОВОР**

В качестве первичного ключа может быть задано поле с типом данных **Счетчик** (AutoNumber). В этом случае при добавлении каждой новой записи в таблицу в это поле автоматически вводятся уникальные целые, последовательно возрастающие, или случайные числа (по умолчанию в поле вводятся последовательные значения). Указание такого поля является наиболее простым способом создания первичного ключа. Значения этого поля нельзя изменить или удалить. В таблице не может быть более одного поля этого типа. Если первичный ключ не установлен пользователем до сохранения вновь созданной таблицы, Access спросит о необходимости создания первичного ключа. При утвердительном ответе Access создаст суррогатный первичный ключ с типом данных **Счетчик** (AutoNumber).

ОБРАТИТЕ ВНИМАНИЕ

Использование в качестве первичного ключа поля с типом данных **Счетчик**: во-первых, не исключает определения в таблице поля, являющегося реальным ключом сущности, — например, такого как **Номер договора** в таблице **ДОГОВОР**, и, во-вторых, требует создания для реального ключа уникального индекса, так как иначе не будет проверяться целостность, и запись с одним и тем же значением реального ключа может быть введена в таблицу неоднократно, — например, запись об одном и том же договоре.

Создание таблицы в режиме таблицы

Рассмотрим, прежде всего, создание таблицы в режиме таблицы. Этот режим значительно проще для неопытного пользователя, а Access предоставляет в этом режиме широкие возможности разработки структуры таблицы, практически мало отличающиеся от доступных в режиме конструирования.

В процессе создания новой базы данных в ней автоматически создается новая пустая таблица с именем **Таблица1** (Table1), которая открывается в области документов в режиме таблицы. На рис. 3.12 показана такая таблица с открытым списком поля **Щелкните для добавления** (Click to Add), обеспечивающим выбор типа данных для поля таблицы.

Одновременно открывается лента **Работа с таблицами | Поля** (Table Tools | Fields), обеспечивающая определение полей таблицы с различными свойствами. На рис. 3.13 представлена эта лента с открытым списком **Другие поля** (More Fields), включающая расширенный список для добавления и удаления полей таблицы с различными типами данных. В конце этого списка имеется раздел **Быстрое добавление полей** (Quick Start), в котором содержатся часто используемые поля, — такие как адрес, телефон, даты начала и окончания. Некоторые из них выполняют добавление поля со списком значений. При открытии списка в поле предоставляется возможность изменить его элементы. Таким образом, пользователь может сразу приступить к разработке первой таблицы базы данных в режиме таблицы, дополняя ее полями с помощью разнообразных и удобных средств.

Для создания следующей таблицы в аналогичном режиме предназначена команда **Таблица** (Table), размещенная на вкладке ленты **Создание** (Create) в группе **Таблицы** (Tables).

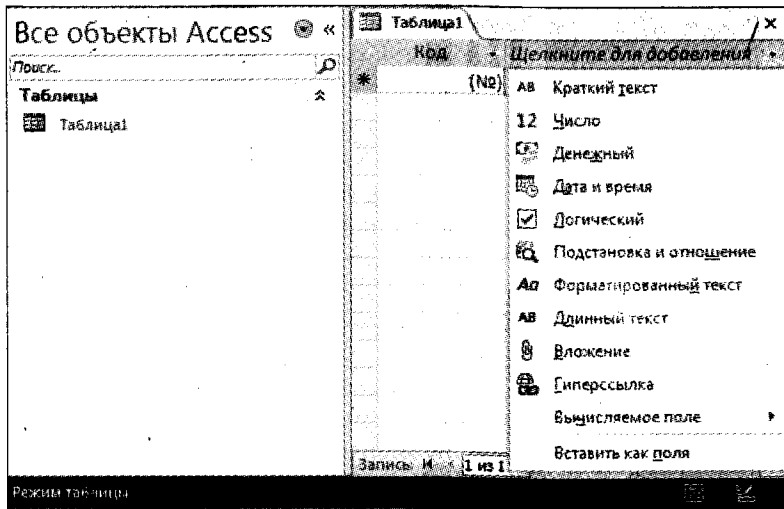


Рис. 3.12. Новая пустая таблица со списком типов данных для добавляемого поля

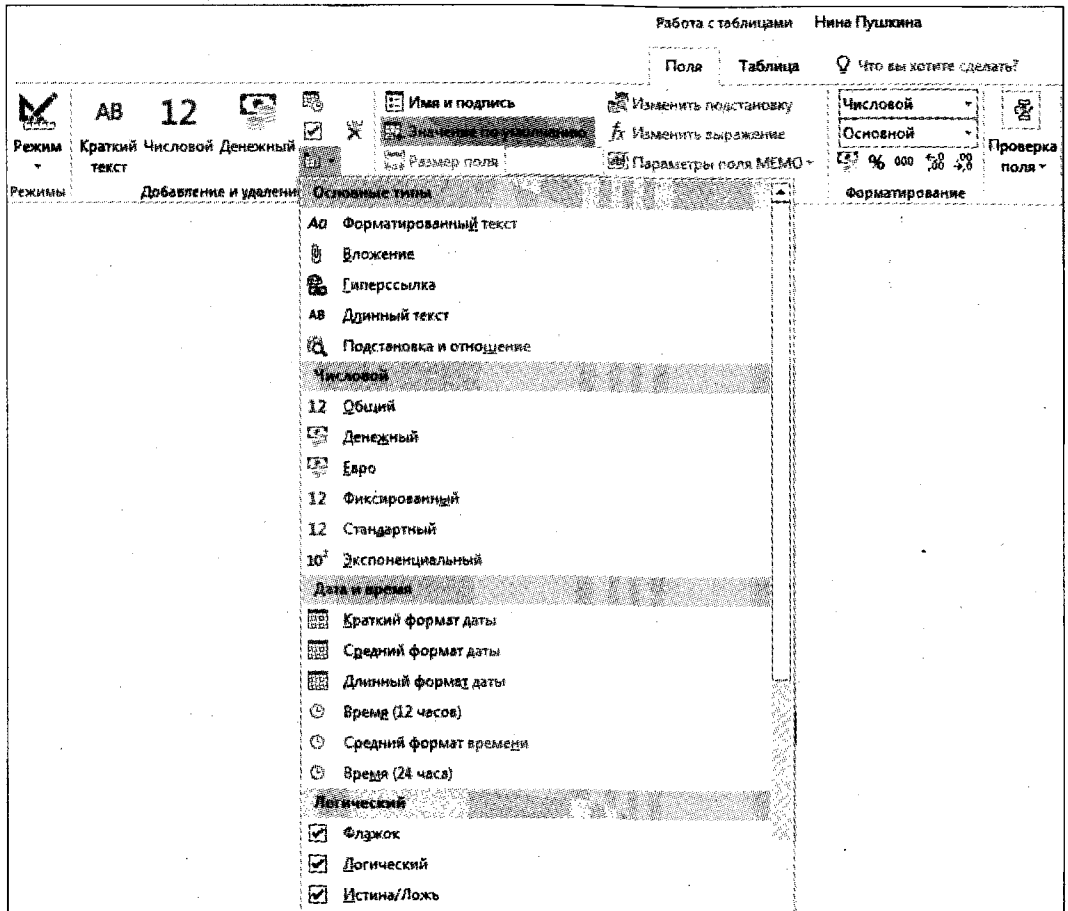


Рис. 3.13. Лента режима Работа с таблицами | Поля (Table Tools | Fields)

В предлагаемой системой таблице определено ключевое поле с типом данных **Счетчик** (AutoNumber), и в нее в режиме таблицы можно добавлять новые поля, наделенные рядом характеристик. Столбец **Щелкните для добавления** (Click to Add) постоянно отображается в режиме таблицы, за исключением случая, когда в таблице не определен первичный ключ.

Как уже было сказано ранее, для каждого поля обязательно задается имя, однозначно определяющее это поле в таблице, и тип данных, соответствующий сохраняемым в этом поле данным. Тип данных определяет значения, которые можно сохранить в поле, операции, которые можно выполнить с данными, а также выделяемый объем памяти. С каждым полем также связывается группа параметров, называемых *свойствами*, которые определяют функциональные характеристики и внешний вид этого поля. Многие свойства поля могут быть определены в режиме таблицы.

Добавление полей

В режиме таблицы определение полей можно выполнить как выбором необходимых команд на ленте **Работа с таблицами | Поля** (Table Tools | Fields) или в списке столбца **Щелкните для добавления** (Click to Add), так и непосредственным вводом данных в столбец **Щелкните для добавления** (см. рис. 3.12). При вводе первого значения Access автоматически определит тип данных (например, дата, число или текст) и добавит новое поле в таблицу. При добавлении полей этим способом Access автоматически назначит им имена: **Поле1** (Field1) для первого, **Поле2** (Field2) для второго и т. д. Чтобы дать полям содержательные имена, измените их. Для этого выполните двойной щелчок на заголовке поля и введите новое имя или, воспользовавшись правой кнопкой мыши, в контекстном меню выберите команду **Переименовать поле** (Rename Column). Имена полей могут содержать до 64 знаков (цифр или букв), включая пробелы.

Для введенных в поле таблицы значений Access распознает и автоматически определяет следующие типы данных:

- для текстовых значений — **Короткий текст** (Short Text) или **Длинный текст** (Long Text);
- для целых числовых значений — **Числовой** (Number), **Длинное целое** (Long Integer);
- для числовых значений типа 45,76 или 34,25 % или 12 % — **Числовой** (Number), **Двойное с плавающей точкой** (Double). Распознаваемый формат чисел зависит от настройки региональных параметров на вашем компьютере. Так если в качестве разделителя целой и дробной частей числа выбрана запятая, а введена точка, полю может быть назначен другой тип данных, — например, **Дата/время** (Date/Time);
- для гиперссылок — **Гиперссылка** (Hyperlink). Допускается использование любого префикса протокола Интернета — например: http://, mailto:;
- для даты и времени — **Дата и время** (Date/Time). Распознаваемый формат даты и времени зависит от настройки региональных параметров на вашем компьютере;

□ для денежных значений типа 12,50 р. — **Денежный** (Currency). Распознаваемый знак валюты зависит от настройки региональных параметров на вашем компьютере.

Если на основании введенных данных Access не может точно определить тип данных, задается тип данных **Короткий текст**.

Access, наряду с определением типа данных, может автоматически задавать значение свойства **Формат** (Format) в зависимости от вида вводимых данных. Это свойство определяет вид отображения и печати данных поля. Например, если ввести 01-январь-2016, будет задан тип данных **Дата и время** (Date/Time) и в свойстве **Формат** (Format) значение **Средний формат даты** (Medium Date). К полю с типом данных **Дата и время** (Date/Time) автоматически привязывается календарь для выбора даты.

После определения структуры таблицы таким простым способом ее всегда можно доработать, воспользовавшись командами ленты **Работа с таблицами | Поля** (Table Tools | Fields) в группах **Свойства** (Properties), **Форматирование** (Formatting) и **Проверка поля** (Field Validation) (см. рис. 3.13). При этом будут доступны почти все используемые Access типы данных.

Пусть необходимо создать таблицу **ПОКУПАТЕЛЬ**, структура которой была определена в *главе 2* и приведена в *приложении 1*. В новой пустой таблице по умолчанию определено поле первичного ключа таблицы с именем **Код** (ID) и типом данных **Счетчик** (AutoNumber). Таблица **ПОКУПАТЕЛЬ** содержит два уникальных поля: код покупателя и ИНН, поэтому можно заменить предлагаемый суррогатный ключ одним из этих полей. Для изменения имени этого поля выполните двойной щелчок на заголовке поля и исправьте имя **Код** (ID) на код_пок. Чтобы изменить тип сохраняемых в поле данных, на вкладке ленты **Работа с таблицами | Поля** (Table Tools | Fields) в группе **Форматирование** (Formatting) откройте список с типами данных и выберите **Короткий текст** (Short Text), а в группе **Свойства** (Properties) измените **Размер поля** (Field Size) по умолчанию (255) на нужный (5).

Для работы с ключевыми полями режим таблицы предоставляет ограниченные возможности. В этом режиме нельзя удалить ключ и нельзя определить новый ключ — ни простой, ни составной. Для выполнения таких операций необходимо использовать режим конструктора. В приведенном примере показано, что в таблице, создаваемой в режиме таблицы, может быть использован предлагаемый системой ключ с типом данных **Счетчик** (AutoNumber) или произведена замена этого ключа любым уникальным полем таблицы с изменением типа данных для этого поля.

Таким образом, если в таблице не предполагается использование суррогатного ключа с типом данных **Счетчик** (AutoNumber), целесообразно процедуру определения ключевых полей перенести в режим конструктора.

Для добавления следующего поля в таблицу **ПОКУПАТЕЛЬ**, например, ИНН, в ячейку под заголовком **Щелкните для добавления** (Click to Add) (см. рис. 3.4) введите значение — например, 0012345678. При вводе такого значения Access автоматически распознает и назначит для этого поля числовой тип данных. Однако, поскольку со значениями поля ИНН не будет производиться никаких арифметических

операций, и к тому же они могут начинаться с нулей, которые в числовых значениях аннулируются, следует изменить тип данных на **Короткий текст** (Short Text). Для текстового поля автоматически выбирается длина, равная 255, что для поля нашей таблицы многовато. Изменить размер поля можно на ленте **Работа с таблицами | Поля** (Table Tools | Fields) в группе **Свойства** (Properties), введя его в соответствующую позицию. Здесь же можно изменить имя поля, указать его подпись, сформировать с помощью построителя выражений значение по умолчанию.

Имя поля, его подпись, отображаемая в заголовке таблицы, а также описание поля могут быть определены с помощью команды **Имя и подпись** (Name & Caption) на вкладке **Работа с таблицами | Поля** (Table Tools | Fields) в группе **Свойства**. Заметим, что если свойство **Подпись** (Caption) поля не определено, в заголовке столбца таблицы отображается его имя.

В группе **Свойства** для поля можно задать также **Значение по умолчанию** (Default Value), которое будет сохраняться в поле, если пользователь не ввел никакого значения.

Для определения типа данных добавляемого поля совсем не обязательно осуществлять ввод данных. Можно, щелкнув в столбце **Щелкните для добавления** (Click to Add), выбрать тип данных нового поля самостоятельно (рис. 3.14).

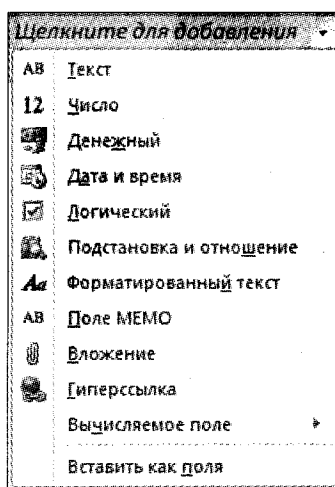


Рис. 3.14. Список типов данных для добавляемого поля

Используя команду **Вставить как поля** (Paste as Fields), можно добавить в таблицу новое поле, скопированное из другой таблицы. Откройте таблицу, из которой будет выполняться копирование, подведите курсор к нужному значению поля и при появлении на нем знака плюс \oplus щелчком мыши выделите его. Скопируйте выделенное поле — копируются конкретное значение поля, его имя и все свойства. Командой **Вставить как поля** добавьте поле в создаваемую таблицу.

Наиболее простым способом добавления полей в таблицу можно считать использование команд группы **Добавление и удаление** (Add & Delete); в которой отдельными кнопками представлены основные типы данных, и в раскрывающемся списке

Другие поля (More Fields) представлены не только недостающие типы данных, но и форматы отображения поля (см. рис. 3.13).

Определив структуру таблицы в режиме таблицы, можно тут же приступить к заполнению ее данными — формированию записей таблицы. Команды по работе с записями таблицы, а также команды оформления внешнего вида таблицы представлены на ленте **Главная (Home)**.

Если таблица была закрыта, открыть ее в режиме таблицы можно, выбрав ее в области навигации и выполнив команду контекстного меню **Открыть (Open)**.

Задание 3.1. Создание таблицы базы данных в режиме таблицы

Завершите разработку структуры таблицы **ПОКУПАТЕЛЬ**, используя *приложение 1*, и введите в нее записи, как показано в *приложении 2*.

Определение структуры таблицы в режиме конструктора

Для создания таблицы в режиме конструктора на вкладке **Создание (Create)** в группе **Таблицы (Tables)** следует выбрать команду **Конструктор таблиц (Table Design)**. В режиме конструктора таблиц открывается окно **Таблица1 (Table1)**, в котором определяется структура таблицы базы данных (рис. 3.15).

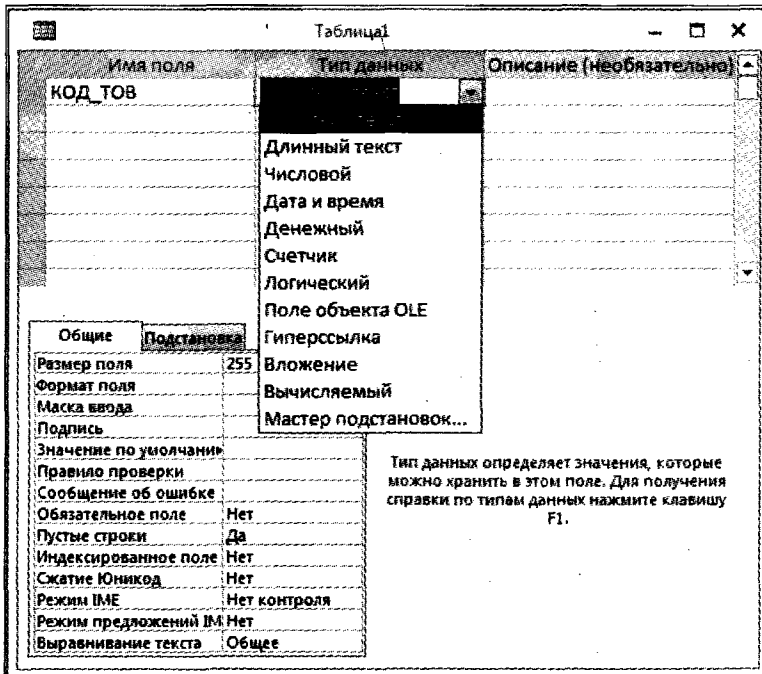


Рис. 3.15. Окно определения структуры таблицы в режиме конструктора

Для перехода из режима таблицы в режим конструктора щелкните на кнопке **Режим** (View), доступной на вкладках **Работа с таблицами | Поля** (Table Tools | Fields) и **Главная** (Home). Кроме того, можно щелкнуть на кнопке **Конструктор** (Design View) в строке состояния или щелкнуть правой кнопкой мыши на заголовке таблицы и в контекстном меню выбрать команду **Конструктор** (Design View).

При переходе в режим конструктора таблиц делается активной лента команд **Работа с таблицами | Конструктор** (Table Tools | Design) (рис. 3.16).

ЗАМЕЧАНИЕ

Назначение кнопки на ленте появляется (всплывает) при наведении на нее курсора мыши.

Для определения поля в окне конструктора таблицы (см. рис. 3.15) задаются **Имя поля** (Field Name), выбирается **Тип данных** (Data Type), вводится **Описание** (Description) — краткий комментарий, в разделе **Свойства поля** (Field Properties) задаются свойства, представленные на двух вкладках: **Общие** (General) и **Подстановка** (Lookup). К общим относятся такие свойства поля, как максимальный размер, формат, подпись, которая выводится в заголовке столбца таблицы, значение по умолчанию и другие. На вкладке **Подстановка** (Lookup) выбирается **Тип элемента управления** (Display Control): поле, список фиксированных значений или поле со списком. Свойства поля зависят от выбранного типа данных. Для отображения свойств поля необходимо установить курсор на строке соответствующего поля. Ранее в этой главе (см. *разд. «Имя поля и тип данных»*) были описаны все типы данных, используемые в Access, а также основные свойства полей.

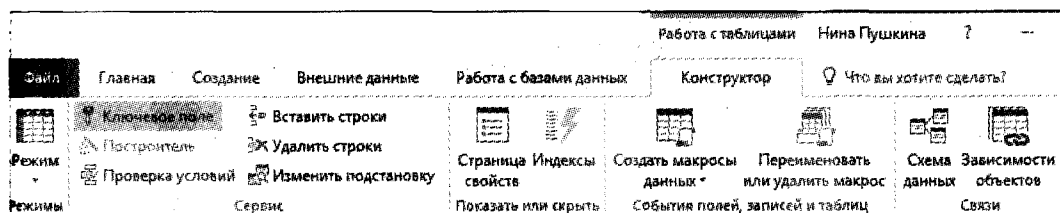


Рис. 3.16. Лента инструментов конструктора таблиц

Создание таблиц базы данных «Поставка товаров»

Рассмотрим последовательность действий при создании таблиц на примере спроектированной в *главе 2* базы данных «Поставка товаров». Структура таблиц этой базы данных приведена в *приложении 1*.

Напомним, что все объекты базы данных Access формируются и сохраняются в одном файле, который создается щелчком мыши в стартовом окне Access на значке **Пустая база данных рабочего стола** (Blank desktop database). При этом в окне **Пустая база данных рабочего стола** (Blank desktop database) задается имя базы

данных (см. рис. 3.2) — назовем ее здесь *Поставка товаров*, и выбирается папка, в которой нужно сохранить файл базы данных. В результате открывается окно новой пустой базы данных *Поставка товаров: база данных* – <полное имя базы данных> (Access 2007–2016), аналогичное приведенному на рис. 3.4.

В соответствии с технологией проектирования реляционной базы данных структура каждой таблицы базы данных «Поставка товаров» должна определяться составом реквизитов соответствующего информационного объекта в ИЛМ. Основные параметры структуры таблиц представлены в табл. П1.1–П1.7 приложения 1.

Создание структуры таблицы

Если ранее в базе данных «Поставка товаров» была создана таблица *ПОКУПАТЕЛЬ* (см. разд. «Создание таблицы в режиме таблицы»), то после открытия базы данных в окне Access таблица *ПОКУПАТЕЛЬ* отобразится в области навигации. Теперь можно приступить к созданию других таблиц базы данных в любом из режимов, предлагаемых Access. Начнем создание таблицы *ТОВАР* с определения ее структуры в режиме конструктора таблиц.

Для этого на вкладке ленты *Создание* (Create) в группе *Таблицы* (Tables) выполним команду *Конструктор таблиц* (Table Design) и в окне конструктора *Таблица1* (Table1) (рис. 3.17) в соответствии с приведенными в табл. 3.1 проектными параметрами структуры определим все поля таблицы *ТОВАР*.

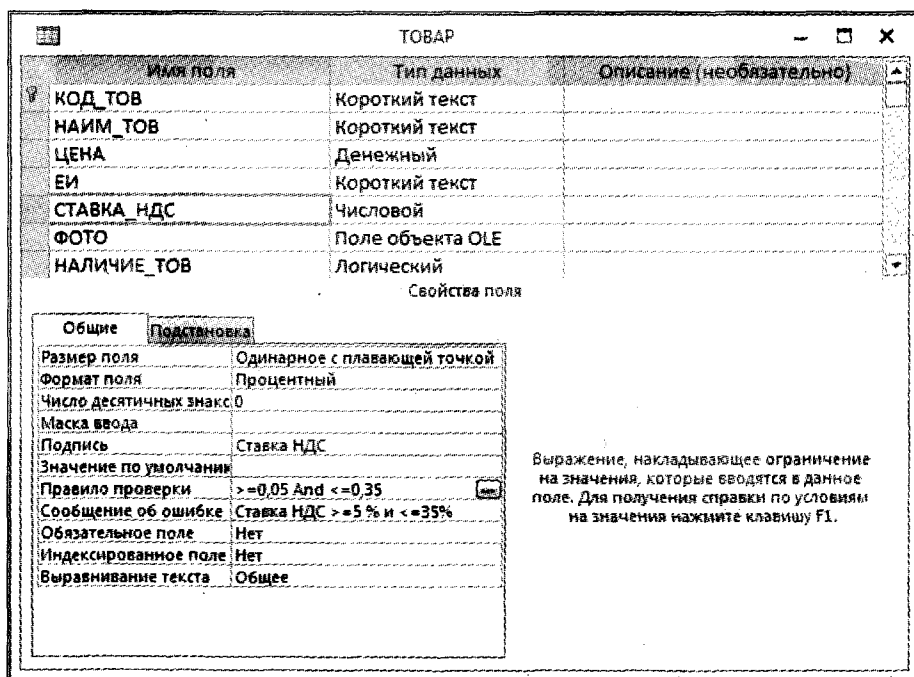


Рис. 3.17. Определение в таблице *ТОВАР* свойств поля *СТАВКА_НДС*

Таблица 3.1. Основные параметры структуры таблицы ТОВАР

Имя поля	Ключевое поле уникальное	Обязательное поле	Тип данных	Размер	Число десятичных знаков	Подпись поля	Правило проверки	Сообщение об ошибке
КОД_ТОВ	Да	Да	Текстовый	5		Код товара		
НАИМ_ТОВ		Нет	Текстовый	25		Наименование товара		
ЦЕНА		Нет	Денежный	Денежный	2	Цена товара	≥ 0 And ≤ 35000	Цена должна быть ≥ 0 и ≤ 35000
ЕИ		Нет	Текстовый	8		Единица измерения		
СТАВКА_НДС		Нет	Числовой	Одинарное с плавающей точкой	0	Ставка НДС	$\geq 0,05$ And $\leq 0,35$	Ставка НДС $\geq 5\%$ и $\leq 35\%$
ФОТО		Нет	Поле объекта OLE	До 2 ГБ		Фото товара		
НАЛИЧИЕ_ТОВ		Нет	Логический			Наличие товара		
ВЛОЖЕНИЯ		Нет	Вложение	До 2 ГБ				
Цена с НДС		Нет	Вычисляемый	Денежный	2			

Для каждого поля таблицы ТОВАР определим **Имя поля** (Field Name), **Тип данных** (Data Type) и на вкладке **Общие** (General) зададим свойства полей:

- для текстовых полей в свойстве **Размер поля** (Field Size) зададим размеры, указанные в соответствующем столбце табл. 3.1;
- для числового поля СТАВКА_НДС выберем в свойстве:
 - **Размер поля** (Field Size) — **Одинарное с плавающей точкой** (Single);
 - **Формат поля** (Format) — **Процентный** (Percent). При отображении одинарных с плавающей точкой чисел, для которых выбран встроенный формат поля **Процентный** (Percent), их значения умножаются на 100 и к ним добавля-

ется знак процента. При этом ввод значений должен осуществляться в формате отображения — например, для ставки НДС: 15 %, 7,5 %;

- Число десятичных знаков (DecimalPlaces) — 0;
- для поля ЦЕНА выберем в свойстве:
- **Формат поля (Format) — Денежный (Currency).** Денежные суммы будут отображаться в соответствии с установленными по умолчанию в Microsoft Windows региональными параметрами в виде 123 456 789,00р. Денежный тип поля используют для предотвращения округления во время вычислений. В денежных полях обеспечивается 15 знаков слева от десятичной запятой и 4 знака справа. Денежное поле занимает 8 байтов;
 - **Число десятичных знаков (DecimalPlaces) — 2;**
- для свойств Подпись (Caption), Правило проверки (Validation Rule), Сообщение об ошибке (Validation Text) — в каждом из полей зададим, как указано в табл. 3.1:
- в поле СТАВКА_ндс будут сохраняться числовые данные **Одинарное с плавающей точкой**, поэтому в правилах проверки должны использоваться соответствующие значения и нельзя вводить их в формате отображения, т. е. в нашем случае в процентах;
 - правило проверки, которое заносится в свойство поля, является логическим выражением. Это выражение проверяется при вводе данных в поле таблицы. Оно может быть сформировано с помощью построителя выражений (рис. 3.18).

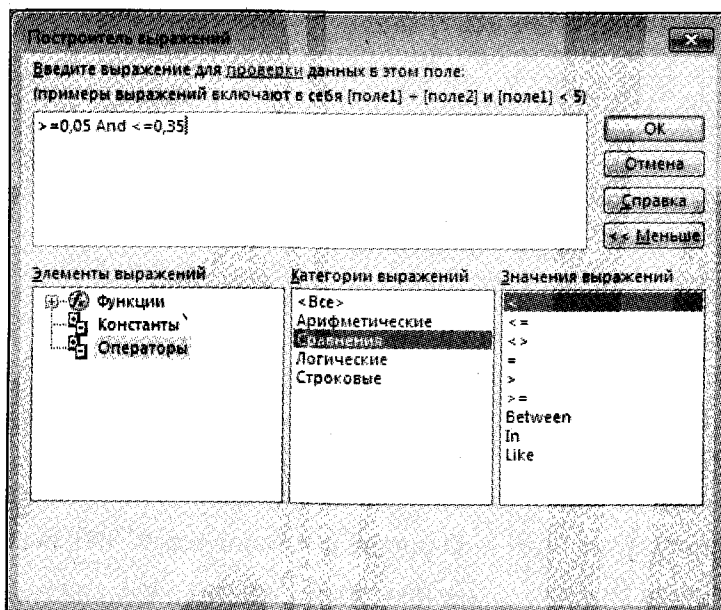


Рис. 3.18. Окно построителя с выражением для проверки данных в поле

в этом свойстве, не прибегая к помощи построителя. Выберите для свойства поля **Тип результата (Result Type)** значение **Денежный (Currency)**.

ЗАМЕЧАНИЕ

Если в некоторых записях таблицы не будет введено значение цены или ставки НДС, а для значения по умолчанию не было указано никаких числовых данных, результат в вычисляемом поле не отобразится. Это связано с тем, что при отсутствии значения в поле оно получает значение Null, с которым не могут производиться вычисления.

Теперь определим первичный ключ таблицы. Выделим поле код_тов, щелкнув кнопкой мыши на области маркировки слева от имени поля, и нажмем кнопку **Ключевое поле (Primary Key)** на вкладке ленты **Конструктор (Design)** в группе **Сервис (Tools)** (см. рис. 3.16). Признаком установки ключа является изображение ключа слева от имени поля. Определим свойства ключевого поля в соответствии с табл. 3.1.

ЗАМЕЧАНИЕ

Для инвентарных номеров и других номеров или кодов, которые часто выступают в качестве ключей и не используются в математических вычислениях, вместо числового целесообразно выбрать текстовый тип данных.

Сохраним созданную структуру таблицы и присвоим имя новой таблице — **ТОВАР**. Для этого выполним команду **Сохранить (Save)** на Панели быстрого доступа (**Customize Quick Access Toolbar**) или на странице, открываемой при выборе значка **Файл**. В окне **Сохранение (Save As)** введем имя таблицы (рис. 3.19).

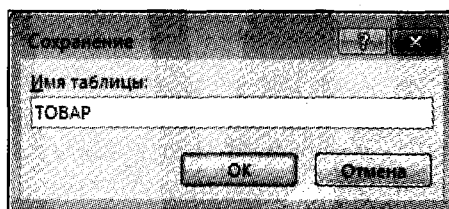


Рис. 3.19. Окно ввода имени сохраняемой таблицы

При сохранении таблицы происходит обновление файла базы данных, в которую помещается созданная таблица. Таблица **ТОВАР** появится в списке объектов **Таблицы (Tables)** в области навигации открытой базы данных «Поставка товаров».

После сохранения структуры таблицы переходите ко второму этапу создания таблицы — созданию записей. Для этого переключитесь в режим таблицы нажатием кнопки **Режим (View)** на ленте конструктора или выбором нужного режима при открытии списка этой кнопки (рис. 3.20). Если сохранение таблицы не было выполнено ранее, при переходе в другой режим откроется окно сохранения таблицы.

В Access, начиная с версии 2007, появились разнообразные средства для добавления и определения свойств полей таблицы в режиме таблицы. Все таблицы в режиме таблицы, если в них определен ключ, содержат пустой столбец с заголовком **Щелкните для добавления (Click to Add)**. Для определения некоторых полей

в таблице достаточно ввести данные в первую пустую ячейку под этим заголовком, и Access сам определит тип данных и некоторые свойства поля. Можно также скопировать и вставить данные в пустой столбец. Можно самостоятельно выбрать нужный тип данных для поля и установить многие его свойства. В Access 2016 на ленте **Работа с таблицами | Поля** (Table Tools | Fields) можно определить множество разнообразных типов данных и их свойств.

Находясь в режиме таблицы, когда отображены значения в полях, удобно задать свойства **Правило проверки поля** (Validation Rule) и **Сообщение об ошибке** (Validation Text), а также сопоставить значения полей записи в правилах проверки записи, например, [ЦенаПродажи]>[ОптоваяЦена]. Для этого следует выбрать поле и открыть список кнопки **Проверка** (Validation) в группе **Проверка поля** (Field Validation). Если в этом списке выбрать **Правило проверки поля** (Field Validation Rule), откроется **Построитель** (Expression Builder), в котором надо ввести выражение для проверки вводимых в поле значений. Выбрав **Сообщение проверки поля** (Field Validation Message), можно ввести текст, который будет выводиться при отклонении вводимых значений от допустимых. Другие команды списка позволят задать выражение для сравнительной проверки значений в полях записи и подготовить сообщение при нарушении условий, заданных выражением. Это выражение сохраняется в свойстве таблицы **Правило проверки** (Validation Rule). Открывается окно свойств таблицы в режиме конструктора кнопкой **Страница свойств** (Property Sheet). Для таблицы может быть введено только одно такое выражение.

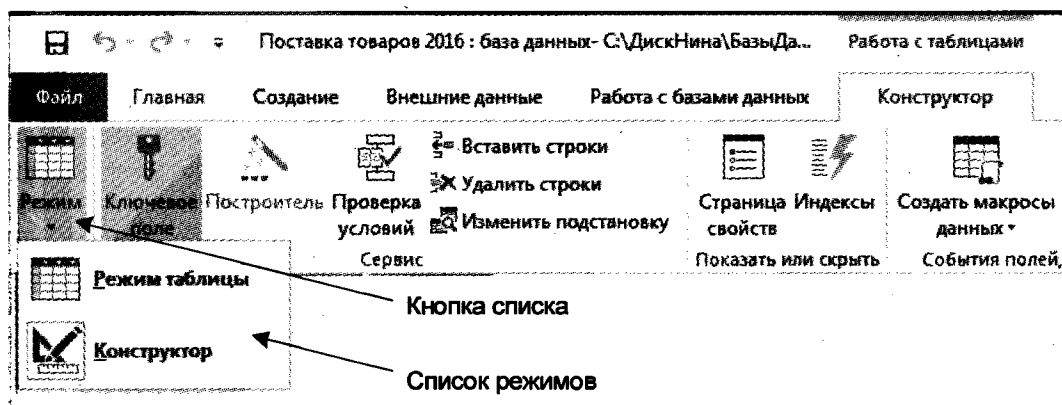


Рис. 3.20. Выбор режима представления таблицы

Как и в режиме конструктора, в режиме таблицы можно удалить столбец. При этом следует помнить, что удаляются все данные столбца, и отменить удаление невозможно. Удаление поля первичного ключа в режиме таблицы невозможно. Для этого необходимо использовать режим конструктора.

Задание 3.2. Создание таблиц базы данных

Создайте структуру таблиц СКЛАД, НАКЛАДНАЯ, ОТГРУЗКА, ДОГОВОР, ПОСТАВКА_ПЛАН, ИСПОЛНИТЕЛЬ. При создании таблиц используйте проект-

ные параметры их структуры (см. приложение 1). Для ввода в таблицы записей используйте данные, приведенные в приложении 2.

* * *

Рассмотрим далее некоторые особенности определения структуры таблиц ТОВАР, НАКЛАДНАЯ, ОТГРУЗКА, ПОКУПАТЕЛЬ, ДОГОВОР, ПОСТАВКА_ПЛАН базы данных «Поставка товаров».

Определение составного первичного ключа

В таблице НАКЛАДНАЯ в составной первичный (уникальный) ключ входят поля НОМ_НАКЛ и КОД_СК. Определение этого ключа возможно только в режиме конструктора таблиц. Выделите оба эти поля, щелкая кнопкой мыши на области маркировки при нажатой клавише <Ctrl>. Затем нажмите на ленте Работа с таблицами | Конструктор (Table Tools | Design) в группе Сервис (Tools) кнопку Ключевое поле (Primary Key).

Аналогично определяются составные ключи в таблицах ОТГРУЗКА и ПОСТАВКА_ПЛАН.

Для ключевого поля автоматически создается уникальный индекс. Для его просмотра выполните команду Индексы (Indexes) на ленте Работа с таблицами | Конструктор (Table Tools | Design) в группе Показать или скрыть (Show/Hide). Окно для просмотра и редактирования индексов таблицы ОТГРУЗКА представлено на рис. 3.21.

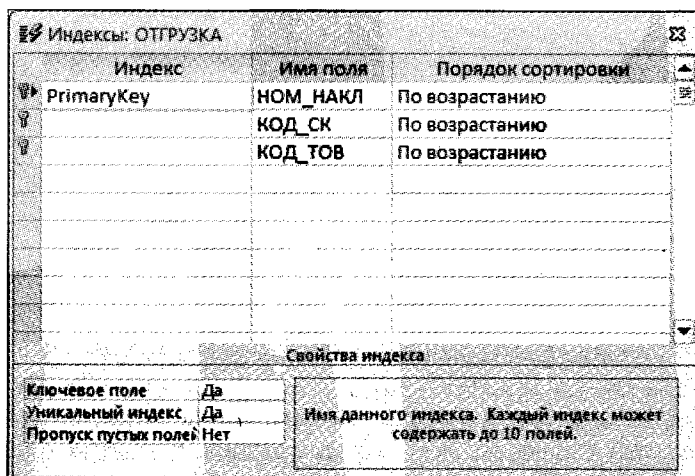


Рис. 3.21. Уникальный индекс по составному первичному ключу таблицы ОТГРУЗКА

Здесь показано, что индексу составного первичного ключа присвоено имя **PrimaryKey**, а в столбце **Имя поля** (Field Name) перечисляются имена полей, составляющих индекс. Индекс первичного ключа всегда уникален и не допускает пустых полей в записях.

Использование данных типа *Поле объекта OLE*

В таблице ТОВАР предусмотрено поле фото, которое должно содержать фотографию товара, хранящуюся в формате графического редактора Paint в файле с расширением bmp. Тип данных такого поля определен как **Поле объекта OLE** (OLE Object). Размещение этого объекта в поле производится на этапе заполнения полей таблицы данными в режиме таблицы или через форму. В режиме таблицы Access отобразит тип объекта **Точечный рисунок**. Отображение объекта возможно только в форме или отчете. Объект может быть внедренным или связанным.

ЗАМЕЧАНИЕ

OLE (Object Linking and Embedding, связывание и внедрение объектов) — это метод передачи информации в виде объектов между приложениями Windows. Поле объекта OLE (OLE Object) является средством, позволяющим установить связь с объектами другого приложения или внедрить объект в базу данных. Объектом может быть документ или его часть, созданная в другом приложении, сохраняющая формат документа-источника и информацию о создавшем его приложении. Объектами могут быть простые и форматированные тексты, рисунки, диаграммы, файлы звукозаписи (WAV), музыки в формате MIDI (музыкально-инструментальный цифровой интерфейс), файлы анимации (FLI, MMM), видеоклипы (AVI), электронные таблицы и другие элементы из различных приложений, поддерживающих это средство. СУБД Access, поддерживая OLE, полностью интегрирована с другими приложениями Microsoft Windows.

Внедренный объект сохраняется в файле базы данных. Двойным щелчком мыши на ячейке, содержащей внедренный объект, предоставляется возможность редактирования объекта средствами приложения, в котором объект был создан.

Связанный объект сохраняется в отдельном файле. Файл объекта можно обновлять независимо от базы данных. Последние изменения будут выведены на экран при следующем открытии формы или отчета. При работе с базой данных также можно просматривать и редактировать объект. Отредактированный связанный объект будет сохраняться в файле объекта, а не в файле базы данных. Связывание объекта удобно при работе с большими объектами, которые нежелательно включать в файл базы данных, а также с объектами, используемыми в различных документах организации, — например, с логотипом. Если связанный файл объекта перемещен, необходимо повторно установить с ним связь.

Использование данных типа *Вложение*

Тип данных **Вложение** (Attachment) можно использовать для хранения нескольких файлов в одном поле, причем в этом поле можно хранить файлы разных типов. Например, в таблице ТОВАР можно в поле с типом данных **Вложение** добавить несколько файлов со сведениями о товаре: сертификат происхождения, сертификат качества, инструкцию по эксплуатации, рекламации, фотографии, схемы и т. д.

Вкладывать файлы можно только в базы данных, созданные в приложении Access 2007/2016 в формате файла ACCDB.

Тип данных **Вложение** (Attachment) предлагается для хранения изображений и документов вместо типа данных **Поле объекта OLE** (OLE Object). Вложениям необ-

ходимо значительно меньше места для хранения и не требуется применения программ, называемых OLE-серверами.

В соответствии с правилами нормализации каждое поле в реляционной базе данных должно содержать только один элемент данных. В противном случае поиск данных будет затруднен или даже невозможен. Вложения нарушают правила разработки базы данных, поскольку в поле можно добавлять несколько файлов — элементов данных. Чтобы избежать этих нарушений, Access по мере добавления файлов вложения создает системные таблицы, которые неявно используются для нормализации данных.

Использование данных типа *Длинный текст*

В таблице ПОКУПАТЕЛЬ предусмотрено поле ОПИСАНИЕ, которое будет содержать текстовое данное большой длины — характеристику покупателя. Для такого поля выберите тип данного — **Длинный текст (Long Text)** (в предыдущих версиях **Поле MEMO (Мемо)**). Ввод данных в это поле можно выполнить непосредственно в поле таблицы, либо через область ввода, вызываемую нажатием комбинации клавиш <Shift>+<F2>.

Если тексты описаний покупателей подготовлены в каком-либо текстовом редакторе (например, Microsoft Word) и хранятся в отдельных файлах, для этого поля может быть задан тип **Поле объекта OLE (OLE Object)** или **Вложение (Attachment)**.

Access 2007/2016 в поле с типом данных **Длинный текст** обеспечивает хранение форматированного текста. Это единственный тип данных в Access, который имеет встроенную поддержку хранения и отображения форматированного текста. Если в поле нужно хранить форматированный текст, создайте поле с длинным текстом и затем в режиме конструктора выберите для свойства **Формат текста (Text Format)** значение **Формат RTF (Rich Text)** вместо **Обычный текст (Plain Text)**. В режиме таблицы для выбора этого свойства предназначена кнопка **Параметры поля MEMO (Memo Settings)** на ленте **Работа с таблицами | Поля (Table Tools | Fields)** в группе **Свойства (Properties)**. Сообщение системы о преобразовании формата показано на рис. 3.22.

Текст поля **Длинный текст** в формате RTF отображается в форматированном виде, хранится и интерпретируется как формат HTML, допускает использование общих

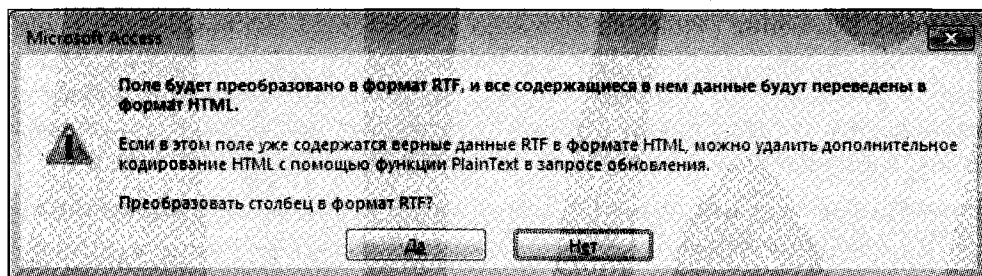


Рис. 3.22. Сообщение при выборе для свойства **Формат текста** значения **Формат RTF**

средств форматирования приложений Microsoft Office, недоступных в обычном тексте. Access автоматически применяет HTML-форматирование к тексту в формате RTF. Использование HTML обусловлено большей степенью его совместимости с полями в формате RTF, хранящимися в списках SharePoint.

К тексту в формате RTF Access допускает применение таких параметров форматирования, как шрифт, размер, цвет, различные виды начертания, выравнивания текста, нумерованные и маркированные списки, уменьшение и увеличение отступа абзаца.

Можно применить форматирование ко всему содержимому поля или его части, редактируя поле в режиме таблицы или в форме. Можно также применить форматирование ко всему полю при просмотре отчета в режиме макета.

Чтобы применить параметры форматирования к тексту RTF, выделите текст, который нужно форматировать, и используйте команды групп **Форматирование текста** (Text Formatting) на ленте **Главная** (Home). Кроме того, после выделения текста мышью появляется мини-панель инструментов, которая содержит кнопки параметров форматирования (рис. 3.23).

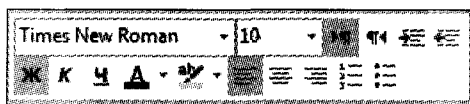


Рис. 3.23. Мини-панель инструментов форматирования текста RTF в поле с длинным текстом

Использование данных типа *Гиперссылка*

Специальный тип данных **Гиперссылка** (Hyperlink) позволяет хранить гиперссылки в полях таблиц базы данных.

Поля с типом данных **Гиперссылка** (Hyperlink) используются для перехода к документам, созданным в различных приложениях Microsoft Office и сохраняемым в файлах на локальных или сетевых дисках, в том числе к другим базам данных, к страницам веб-серверов в Интернете или интранете. При щелчке мышью на гиперссылке осуществляется переход к документу, который открывается создавшим его приложением. Гиперссылки в полях записываются в виде URL- или UNC-адресов и отображаются в поле как подчеркнутый текст, заданный при определении гиперссылки.

В таблице **ПОКУПАТЕЛЬ** предусмотрено поле **лич_док**, в котором может храниться, например, ссылка на некоторый документ, характеризующий покупателя, или ссылка на его веб-страницу или на его базу данных. Для создания гиперссылки в этом поле откройте таблицу в режиме таблицы. Введите в поле текст, который должен отображаться в поле и в то же время определит содержание открываемого по гиперссылке документа, — например, Все о покупателе. Нажмите правую кнопку мыши и выберите в контекстном меню команду **Гиперссылка | Изменить гиперссылку** (Hyperlink | Edit Hyperlink). В открывшемся окне **Изменение гиперссылки** (Edit Hyperlink) выберите папку и файл, который хранит нужный документ (рис. 3.24).

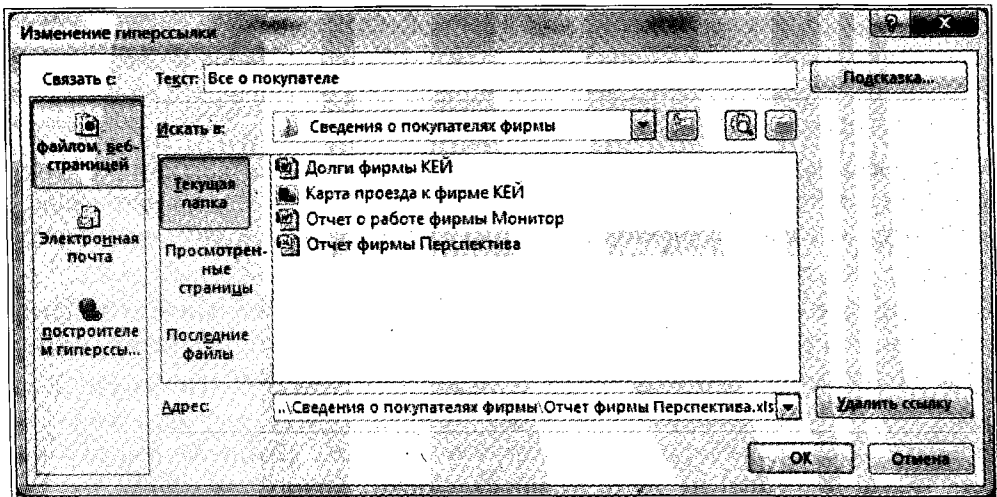


Рис. 3.24. Определение файла, который должен открываться по гиперссылке Все о покупателе

Чтобы добавить гиперссылку для создания сообщений электронной почты, также введите в поле текст — например, Письмо фирме, и в диалоговом окне **Изменение гиперссылки** (Edit Hyperlink) заполните поля, как показано на рис. 3.25.

Щелчком на поле с этой гиперссылкой открывается установленная почтовая программа (например, Outlook) и окно нового сообщения с заполненными полями **Кому** и **Тема**.

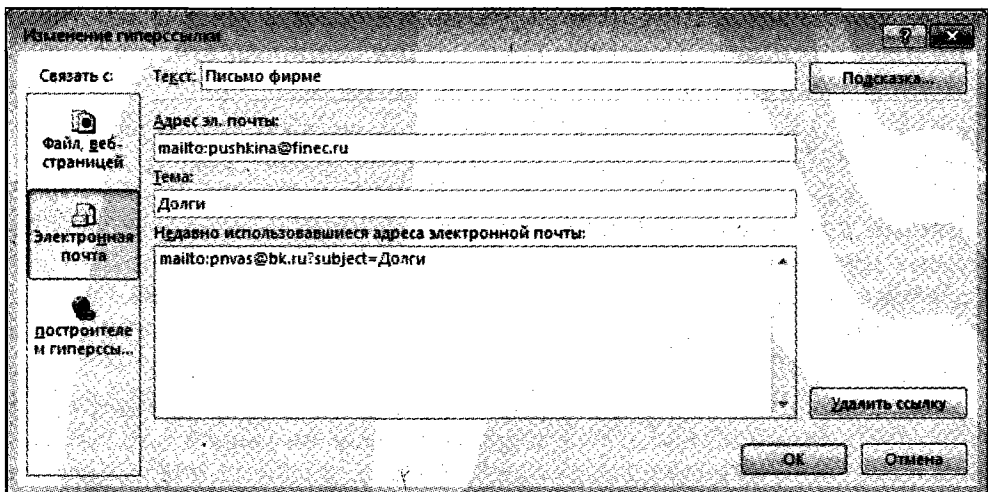


Рис. 3.25. Определение для гиперссылки адреса электронной почты и темы письма

Для создания гиперссылки на веб-страницу запишите в поле, например, такой текст: Страница покупателя#<http://comp.ru/gorod/firm/mircom.htm#Реквизиты>.

По такой гиперссылке будет выполняться переход на страницу фирмы, хранящуюся на сервере в папке, определяемой путем /gorod/firm/ в файле mircom.htm. Причем

будет открыт раздел с именем **Реквизиты**, а в поле гиперссылки будет записано **Страница покупателя**. Если открыть окно **Изменение гиперссылки** (Edit Hyperlink), то можно убедиться, что введенная в поле информация размещена в соответствующих областях: страница покупателя — в области **Текст** (Text), а вся остальная часть — в области **Адрес** (Address). Очевидно, что как отображаемый текст гиперссылки, так и ее адрес могут непосредственно вводиться в окне **Изменение гиперссылки** (Edit Hyperlink). Кроме того, адрес может формироваться путем выбора ранее просмотренных страниц.

Удалить гиперссылку из поля можно командой контекстного меню поля **Гиперссылка** | **Удалить гиперссылку** (Hyperlink | Remove Hyperlink) или простым нажатием на выделенной гиперссылке клавиши <Delete>.

Использование данных типа *Дата и время*

В таблице **ДОГОВОР** для сохранения даты заключения договора предусмотрено поле **ДАТА_ДОГ**. Выберите для этого поля тип данного **Дата и время** (Date/Time). Для отображения данных этого типа может быть выбран один из следующих форматов поля: **Полный** (General Date), **Краткий** (Short Date), **Средний** (Medium Date) или **Длинный формат даты** (Long Date). В режиме конструктора образцы дат в каждом из форматов показаны в строках списка при выборе формата.

В Access 2007/2016 с полем даты автоматически связывается элемент управления **Календарь**, отображаемый при переходе в поле даты значком справа от него. Для ввода даты щелкните на этом значке и в открывшемся календаре выберите нужную дату (рис. 3.26).



Рис. 3.26. Выбор в календаре значения даты для ввода в поле

Маска ввода для даты и текста

Для управления вводом даты в таблице **ДОГОВОР** в свойствах поля **ДАТА_ДОГ** может быть определена маска ввода. Маска ввода задает формат, состоящий из постоянных символов (таких как скобки, точки или дефисы) и специальных символов маски, указывающих, в какие позиции, в каком количестве и какого типа данные могут быть введены. Например, маска для ввода даты в кратком формате имеет вид: 00/00/0000, в среднем формате: 00->L<LL-0000. Воспользуйтесь значком построи-

теля в конце строки свойства **Маска ввода** (Input Mask) для вызова мастера, который позволит без труда сформировать маску ввода.

ОБРАТИТЕ ВНИМАНИЕ

При определении для поля даты маски ввода календарь не отображается.

Для ввода номера телефона в поле **Тел** таблицы **ПОКУПАТЕЛЬ** определите маску (999) 000-0099, также воспользовавшись мастером. Эта маска позволит вводить номера телефонов с кодами городов и без них. Число цифр в номере телефона не может быть менее 5 и более 7. Символ 9 означает, что вместо цифр кода можно ввести пробелы, а последние две цифры номера вводить не обязательно.

Непосредственный ввод данных в таблицы

Определив структуру, можно приступить ко второму этапу создания таблицы — вводу в нее данных.

Непосредственный ввод данных в таблицу производится в **Режиме таблицы** (Datasheet). Переход в режим таблицы из режима конструктора осуществляется нажатием кнопки **Режим** (View) на ленте **Главная** (Home) или **Конструктор** (Design), выполнением команды контекстного меню, вызываемого правой кнопкой мыши на заголовке таблицы, или щелчком мыши на соответствующей кнопке в строке состояния. Если таблица закрыта, выберите ее в области навигации и выполните команду контекстного меню **Открыть** (Open) или просто выполните на таблице двойной щелчок.

В этом режиме отображается лента **Главная** (Home) (рис. 3.27), предназначенная для работы с записями таблицы, их сортировки и фильтрации, поиска в таблице конкретных значений и их замены, определения ряда параметров макета. Кроме того, при переходе в режим таблицы делается доступной лента **Работа с таблицами** (Table Tools), предназначенная для изменения параметров структуры таблицы без перехода в режим конструктора. Лента **Работа с таблицами | Поля** (Table Tools | Fields) представлена ранее на рис. 3.13.

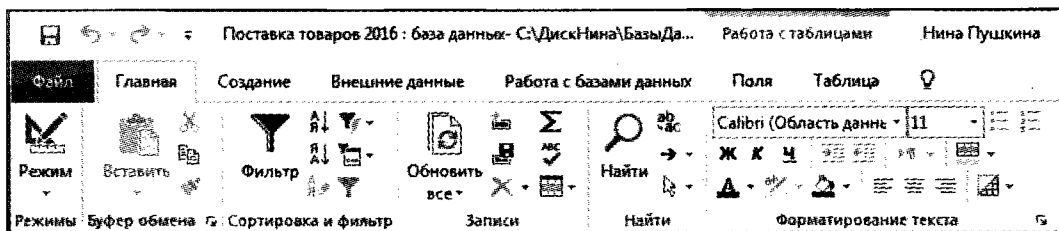


Рис. 3.27. Лента Главная при открытой таблице в режиме таблицы

По умолчанию база данных открывается для работы в многопользовательской среде для совместного доступа с возможностью чтения и записи в базу данных для всех пользователей.

В режиме таблицы предоставляется возможность вводить новые записи в таблицу путем заполнения значениями ее полей.

Очевидно, вводимые в поля таблицы значения данных должны соответствовать типам данных, определенным в структуре, а способ их записи — установленным форматам данных. Значения должны удовлетворять накладываемым ограничениям. После ввода значения в ячейку поля при попытке перейти к другой ячейке Access проверяет, является ли введенное данное допустимым для этого поля. Если введено значение, не соответствующее типу данных поля, Access пытается преобразовать его в правильный тип данных. Если значение не является допустимым и преобразование невозможно — например, нельзя преобразовать текст в число, — появляется предупреждающее сообщение. Для того чтобы выйти из ячейки, следует ввести правильное значение или отменить внесенные изменения клавишей <Esc>.

Обязательно должны вводиться значения в поля первичного ключа и в поля связи, а также в поля, для которых заданы свойства **Обязательное поле** (Required), **Правило проверки** (Validation Rule) или в свойстве **Пустые строки** (Allow Zero Length) выбрано значение **Нет**.

При заполнении связанных таблиц необходимо обеспечить правильность связей между записями этих таблиц. Так, при одно-многозначных связях таблиц и вводе в запись подчиненной таблицы значения поля связи, необходимо отслеживать наличие в главной таблице записей с соответствующими значениями ключевых полей. При изменении или удалении ключевых полей в записях главной таблицы нужно изменять или удалять связанные с ними записи в подчиненных таблицах.

Макет таблицы


Для удобства работы с таблицей можно изменить ее представление на экране. При этом допускается менять ширину столбца, высоту строки, шрифт данных таблицы, цвет текста, линий сетки и фона, оформление, которое может быть обычным, приподнятым или утопленным. Можно выводить на экран только те столбцы, которые нужны для текущей работы, можно зафиксировать столбец при просмотре широких таблиц. Эти параметры отображения таблицы на экране называются *макетом* таблицы и сохраняются вместе с ней.

Настройка макета выполняется в режиме таблицы. При этом могут быть использованы команды ленты **Главная** (Home) или команды контекстного меню.

Многие операции настройки макета можно выполнить непосредственно в таблице с помощью мыши.

- **Изменение ширины столбца.** Для изменения ширины столбца курсор мыши устанавливается на линию, разделяющую имена столбцов. При этом он превращается в планочку со стрелками в обе стороны. Далее границу столбца можно перетянуть в нужное место.
- **Изменение высоты строки.** Для изменения высоты строки курсор мыши устанавливается в области маркировки записи, расположенной слева, на границе

между записями, после чего граница строки перетаскивается на требуемое расстояние. При этом изменяется высота всех строк таблицы.

- *Скрыть/отобразить столбец с экрана.* Убрать столбец можно, перетащив его правую границу влево до исчезновения столбца. Для восстановления отображения скрытого столбца следует установить курсор чуть правее границы столбцов, между которыми размещен скрытый (двунаправленная стрелка при этом имеет разрыв), и перетащить ее вправо.
- *Изменение порядка расположения столбцов на экране.* Столбец выделяется щелчком кнопки мыши на его имени. Протащив курсор мыши поперек столбцов, можно выделить несколько столбцов. Выделенный столбец перетаскивается в новое место при установке курсора на его имени (области маркировки столбца).
- *Изменить ширину столбца, высоту строк, скрыть/отобразить столбцы, не нужные для текущей работы, закрепить/освободить столбцы* можно с помощью соответствующих команд, доступных в контекстном меню выделенного столбца и в списке команд  - Дополнительно (More), размещенной на вкладке ленты Главная (Home) в группе Записи (Records).
- *Сохранение макета таблицы* выполняется командой Сохранить (Save) на вкладке ленты Главная (Home) в группе Записи (Records), на странице, открываемой выбором значка Файл, или при закрытии таблицы после утвердительного ответа на вопрос Сохранить изменения макета таблицы '<имя таблицы>?' (Do you want to save changes to the layout of table <имя таблицы>?).

Ввод данных в таблицы базы данных

Рассмотрим ввод данных на примерах таблиц базы данных «Поставка товаров». В режиме таблицы введем несколько записей в таблицу ТОВАР. В области навигации, представляющей все объекты базы данных «Поставка товаров», установим курсор на таблице ТОВАР и в контекстном меню выберем команду Открыть (Open) или выполним двойной щелчок на имени таблицы — таблица откроется в режиме таблицы (Datasheet View). Заполним записи (строки) открывшейся таблицы (рис. 3.28) в соответствии с названиями полей (столбцов).

Корректность вводимых данных (соответствие заданному типу поля, размеру и условию на значения, которые определены в свойствах полей в режиме конструктора), проверяется автоматически. Отслеживается также уникальность значений ключевых полей.

Отменить ввод значения в поля текущей записи можно, нажав клавишу <Esc> или выполнив команду Отменить (Undo) на панели быстрого доступа. Переход от одного поля к другому можно выполнить клавишами <Enter>, <Tab> или переводом курсора.

Завершение ввода новых значений записи или редактирования осуществляется при переходе к любой другой записи (при смене текущей записи). После перехода к другой записи можно отменить ввод (редактирование) всей записи, также выполнив команду Отменить (Undo) на панели быстрого доступа.

Код товара	Наименование товара	Цена	Единица	Ставка НДС	Фото товара	Наличие	Цена с НДС
T001	Монитор 17LG	7 000,00р.	штука	5%	Bitmap Image	<input checked="" type="checkbox"/>	7 350,00р.
T002	FDD 3,5	500,00р.	коробка	6%	Bitmap Image	<input checked="" type="checkbox"/>	530,00р.
T003	HDD Maxtor 120GB	1 120,00р.	штуки	5%	Bitmap Image	<input checked="" type="checkbox"/>	1 176,00р.
T004	Корпус MiniTower	700,00р.	штука	10%	Bitmap Image	<input checked="" type="checkbox"/>	770,00р.
T005	CD-ROM Panasonic IDE	800,00р.	штуки	30%	Bitmap Image	<input type="checkbox"/>	1 040,00р.
T006	ОЗУ 64М PC100	700,00р.	штука	15%	Bitmap Image	<input type="checkbox"/>	805,00р.
T007	Принтер EPSON ST.A4	2 400,00р.	штука	10%	Bitmap Image	<input type="checkbox"/>	2 640,00р.
T008	СканерAcer	2 338,00р.	штуки	16%	Bitmap Image	<input type="checkbox"/>	2 712,08р.
T009	За. Карта Genius LV	789,00р.	штука	10%	Bitmap Image	<input type="checkbox"/>	867,90р.
T010	Модем Genius эк	1 235,00р.	штука	10%	Bitmap Image	<input checked="" type="checkbox"/>	1 358,50р.
T011	Плоттер	3 000,00р.	штука	10%	Bitmap Image	<input type="checkbox"/>	3 300,00р.
T012	Миникомпьютер	35 000,00р.	штука	15%	Bitmap Image	<input type="checkbox"/>	40 250,00р.
		0,00р.				<input checked="" type="checkbox"/>	

Кнопки перехода по записям
 Маркер строки, в которую вносится новая запись
 Маркер записи, в которую вносятся изменения

Рис. 3.28. Непосредственный ввод записей в таблицу ТОВАР

ВНИМАНИЕ!

Команда **Отменить** (Undo) в зависимости от контекста приобретает различный смысл. По этой команде могут быть отменены только изменения, касающиеся одной последней записи.

- **Добавление записи** в таблицу начинается с заполнения пустой строки, размещенной в конце таблицы и помеченной звездочкой (*). Переход к этой записи можно выполнить также по команде **Создать** (New) на вкладке ленты **Главная** (Home) в группе **Записи** (Records). При создании новой записи должны быть заполнены ключевые поля, поля связи, если установлены связи таблицы с другими таблицами, а также поля, требующие обязательного заполнения в соответствии со свойствами поля.
- **Сохранение новой записи** происходит после перехода к другой записи или выполнения команды **Сохранить** (Save) на ленте **Главная** (Home) в группе **Записи** (Records).
- **Для удаления записи** в таблице ее нужно выделить и в списке кнопки **Удалить** (Delete) на ленте **Главная** (Home) в группе **Записи** (Records) выбрать соответствующую команду. Для исключения ошибочного удаления в Access предусмотрен запрос на подтверждение удаления. После подтверждения на удаление восстановление удаленной строки невозможно.
- **Изменение значений** в полях записи осуществляется непосредственно в ячейках таблицы. Введенное значение проверяется Access при попытке перевода курсора в другое поле. Если значение не является допустимым, появляется предупреждающее сообщение. Для того чтобы выйти из поля, следует ввести правильное значение или отменить внесенные изменения. Отмена изменения значения про-

изводится нажатием клавиши <Esc> или выполнением команды **Отменить** (Undo) на панели быстрого доступа. Откорректированная запись сохраняется после перехода к другой записи или принудительного сохранения командой **Сохранить** (Save) на ленте Главная (Home) в группе **Записи** (Records). *Корректируемая запись* помечается в таблице слева символом карандаша (см. рис. 3.28).


Режим ввода записи

При дополнении таблицы новыми записями может быть использован режим ввода записи, при котором видны только записи, введенные после открытия таблицы. Для реализации этого режима может быть применен фильтр, например, по ключевому полю, в котором отмечается вывод только пустых записей.

Откройте таблицу в режиме таблицы, кнопкой списка на заголовке ключевого поля откройте меню и в области перечисления уникальных значений оставьте помеченным только значение **Пустые** (Blanks). В результате в таблице будет отображена только одна строка новой пустой записи, с которой можно начать ввод нужного числа записей. При этом в свойства таблицы будет записан созданный фильтр.

Если необходимо, чтобы при открытии таблицы применялся сохраненный фильтр и устанавливался режим ввода новых записей, выберите в свойстве таблицы **Фильтр при загрузке** (FilterOnLoad) значение **Да**. Открывается окно свойств таблицы в режиме конструктора кнопкой **Страница свойств** (Property Sheet).

Чтобы вернуться к отображению без фильтра, необходимо удалить фильтр, например, нажав кнопку **С фильтром** (Filtered) на панели переходов по записям в нижней части окна таблицы или выполнив команду **Удалить фильтр** (Filter) на вкладке Главная (Home) в группе **Сортировка и фильтр** (Sort & Filter). При этом фильтр удаляется временно, сохраняя возможность вернуться к исходному представлению. Для повторного применения фильтра служат команда **Применить фильтр** (Toggle Filter) или кнопка **Без фильтра** (Unfiltered).

Для окончательного удаления фильтра необходимо очистить его командой **Удалить все фильтры** (Clear All Filters), размещенной в списке кнопки  **Параметры расширенного фильтра**.

Вывод строки итогов в таблице

Для подсчета числа товаров в таблице или средней, минимальной или максимальной цены товара выберите на ленте Главная (Home) в группе **Записи** (Records) команду **Итоги**. Строка **Итог** будет отображена под строкой со звездочкой. В строке **Итог** щелкните в столбце **Цена** и нажмите кнопку со стрелкой вниз. В открывшемся списке выберите **Количество значений**. Также просмотрите среднее, минимальное и максимальное значение для цены. Повторным выполнением команды **Итоги** скройте строку **Итог** из таблицы. Если выполнить команду **Итоги** еще раз, отобразится строка **Итог** с последним выбранным значением.

Размещение объекта OLE

Выполним размещение объекта OLE на примере поля фото в таблице ТОВАР. Пусть фотографии хранятся в формате графического редактора Paint в файлах с расширением png.

Рассмотрим вариант внедрения объекта в файл базы данных. Установим курсор в соответствующем поле таблицы (см. рис. 3.28). Выполним команду контекстного меню **Вставить объект** (Insert Object). В открывшемся окне (рис. 3.29) в поле **Тип объекта** (Object Type) выберем **Bitmap Image** и отметим переключатель **Создать из файла** (Create from File).

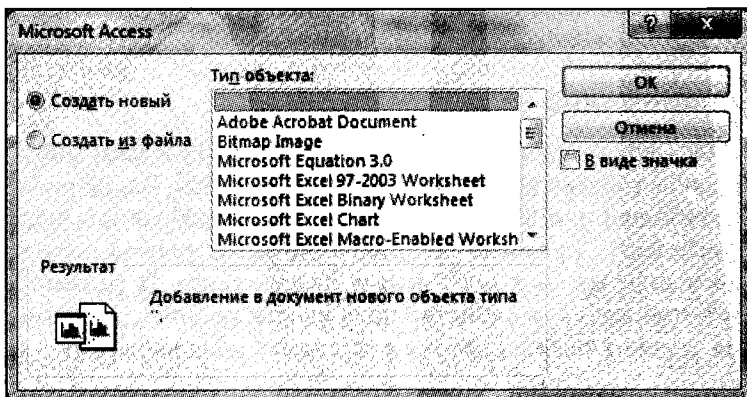


Рис. 3.29. Окно для выбора варианта вставки объекта

В открывшемся окне (рис. 3.30) нужно ввести имя файла с фотографией. Для поиска файла можно воспользоваться кнопкой **Обзор** (Browse), по которой выведется диалоговое окно, позволяющее выбрать необходимый файл.

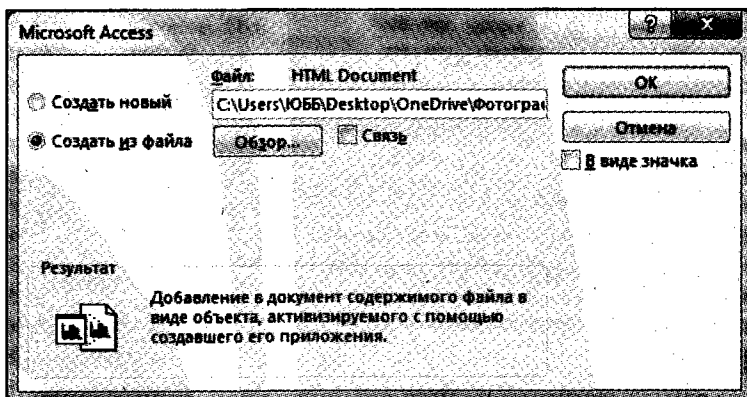


Рис. 3.30. Окно для выбора вставляемого объекта из файла

ВНИМАНИЕ!

Флажок **Связь** (Link) по умолчанию не отмечен и, следовательно, содержимое файла будет введено в поле как внедренный объект. Увидеть содержимое поля можно, выполнив двойной щелчок на нем в таблице, а также через форму или отчет. Дальней-

шие изменения в файле не будут отражаться на встроенном объекте. Для ввода в поле *связанного объекта* надо установить флажок **Связь (Link)**. Это сэкономит место в базе данных и даст возможность отображать все изменения, вносимые в файл другими приложениями.

Результат заполнения таблицы представлен на рис. 3.28, где в соответствующем поле указан вид объекта: **Bitmap Image**. Для просмотра внедренного объекта в создавшем его приложении достаточно в соответствующем поле установить курсор и выполнить двойной щелчок кнопкой мыши.

Для отображения содержимого поля в виде значка, представляющего файл с документом, надо в окне вставки объекта (см. рис. 3.30) установить флажок **В виде значка (Display As Icon)**. Значок может быть использован для представления связанного объекта.

Размещение вложений

Добавим поле вложения в таблицу базы данных **ТОВАР**. Access 2007/2016 предоставляет два способа добавления поля вложения в таблицу. Такое поле можно добавить в режиме таблицы или в режиме конструктора, выбрав для него тип данных **Вложение (Attachment)**. В режиме таблицы в строке заголовка поля вложения отображается значок скрепки, и в него нельзя ввести текст. В режиме конструктора можно изменить имя поля и в свойстве поля ввести подпись, например, **Поле вложений**, которая заменит значок скрепки в заголовке поля вложения.

ЗАМЕЧАНИЕ

Преобразовать поле вложения в поле с другим типом данных невозможно. Такое поле можно только удалить.

Для добавления и изменения вложений служит диалоговое окно **Вложения (Attachments)** (рис. 3.31). Это диалоговое окно можно открыть двойным щелчком в поле вложения в таблице. Файлы вложений сохраняются в базе данных.

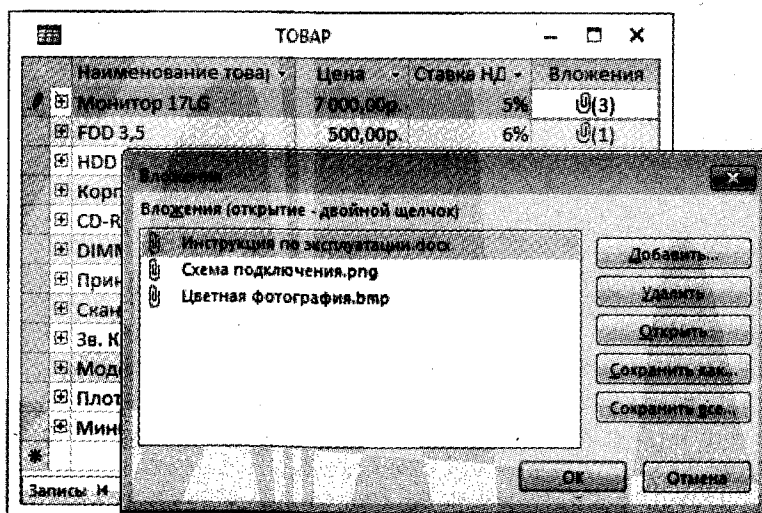


Рис. 3.31. Окно для управления файлами в поле вложений

На рис. 3.31 в окне **Вложения** (Attachments) перечислены добавленные файлы, а в поле вложений указано их число.

Вложения можно просматривать и редактировать, нажав в окне **Вложения** кнопку **Открыть** (Open). Для открытия вложений используются программы, в которых создавались эти файлы, или программы, поддерживающие файлы такого типа. Например, при открытии вложенного в таблицу рисунка BMP запускается программа просмотра фотографий Windows, при открытии документа Word запускается приложение Word, и работа с документом происходит в этом приложении. Если приложение Word не установлено на компьютере, откроется диалоговое окно с предложением выбрать программу для работы с файлом.

Для рисунка, открытого в программе просмотра, выполните команду **Открыть** и выберите программу, с помощью которой можно выполнить редактирование рисунка.

Отредактируйте открытый файл средствами открывшей его программы. Сохраните внесенные в файл изменения и завершите работу программы. При этом все изменения вложенного файла сохранятся во временной копии файла на жестком диске в папке временных файлов Интернета.

Чтобы сохранить изменения в базе данных, вернитесь в Access и в диалоговом окне **Вложения** (Attachments) нажмите кнопку **ОК**. Окно закроется, и появится сообщение с предложением сохранить вложенный файл (рис. 3.32) — нажмите кнопку **Да**.

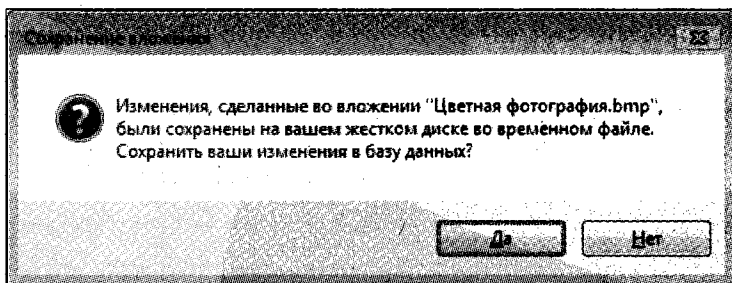



Рис. 3.32. Сообщение при завершении работы с вложениями

Любой из файлов или все файлы, вложенные в поле записи, можно сохранить на жестком диске или в сети. В диалоговом окне **Вложения** (Attachments) нажмите, соответственно, кнопку **Сохранить как** (Save As) или **Сохранить все** (Save All). В диалоговом окне **Сохранить вложения** (Save Attachment) выберите папку для нового расположения файла или всех файлов вложения и нажмите кнопку **Сохранить** (Save).

При использовании поля вложений в форме или отчете в соответствующем элементе управления отображаются и могут просматриваться без использования дополнительного программного обеспечения вложения таких форматов графических файлов, как BMP, PNG, GIF, JPEG, JPG, TIFF, TIF и ряда других, непосредственно поддерживаемых Access. Для неграфических файлов в элементе управления отображается значок программы, создавшей файл. Для элемента управления в форме,

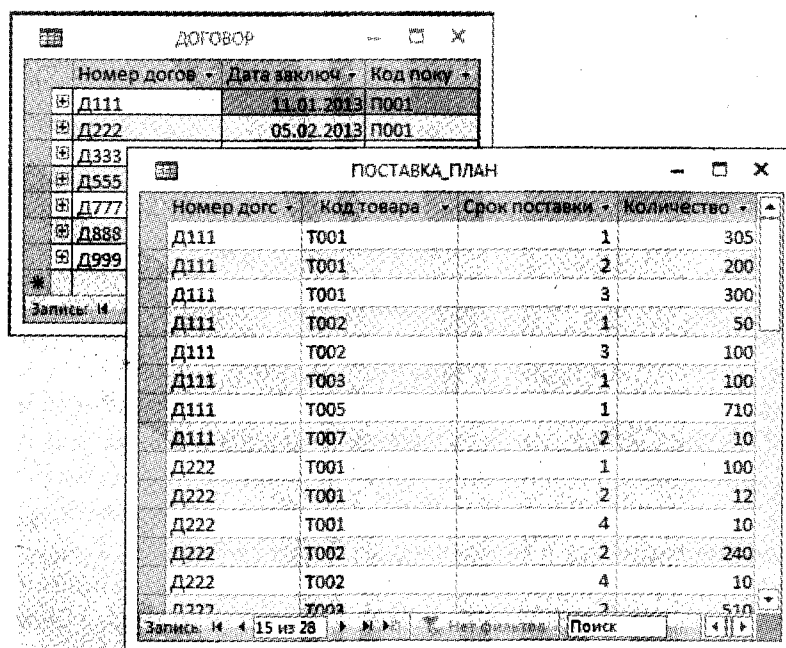
так же, как и в таблице, доступно окно **Вложения** (Attachments) со всеми его командами управления. В формах и отчетах для перехода по файлам и вызова окна **Вложения** используется прозрачная мини-панель инструментов , вызываемая щелчком мыши на элементе управления.

Как правило, в качестве вложений можно использовать файлы, созданные в любых программах Microsoft Office. Кроме того, вложениями могут служить файлы журнала (LOG), текстовые файлы (ТЕХТ, ТХТ) и сжатые ZIP-файлы.

Ввод логически связанных записей

Введем несколько логически взаимосвязанных записей в таблицы ДОГОВОР и ПОСТАВКА_ПЛАН. Логическая связь этих таблиц обеспечивается полем ном_дог — номер договора, входящим в структуру обеих таблиц.

Объекты ДОГОВОР и ПОСТАВКА_ПЛАН связаны одно-многочными отношениями. Связь таких таблиц реализуется в схеме данных с помощью ключа главной таблицы ДОГОВОР. В подчиненной таблице это поле связи является частью ее составного ключа. Пока не создана схема данных, в которой устанавливаются связи между таблицами и их свойства, система не может контролировать логическую взаимосвязь вводимых данных. Поэтому для получения целостной базы, в которой каждая запись подчиненной таблицы имеет логически связанную с ней главную запись, пользователю придется самому отслеживать взаимосвязи записей. При вводе подчиненной записи в таблицу ПОСТАВКА_ПЛАН необходимо проверять наличие в главной таблице ДОГОВОР записи с ключом, значение которого совпадает



Номер догов	Дата заключ	Код поку
Д111	11.01.2013	П001
Д222	05.02.2013	П001
Д333		
Д555		
Д777		
Д888		
Д999		

Номер догс	Код товара	Срок поставки	Количество
Д111	Т001	1	305
Д111	Т001	2	200
Д111	Т001	3	300
Д111	Т002	1	50
Д111	Т002	3	100
Д111	Т003	1	100
Д111	Т005	1	710
Д111	Т007	2	10
Д222	Т001	1	100
Д222	Т001	2	12
Д222	Т001	4	10
Д222	Т002	2	240
Д222	Т002	4	10
Д222	Т002	2	510

Рис. 3.33. Таблицы ДОГОВОР и ПОСТАВКА_ПЛАН

со значением поля связи (внешнего ключа) в подчиненной записи. То есть при добавлении строки плана поставки для договора д111 необходимо, чтобы договор с этим номером уже был представлен в таблице ДОГОВОР.

При непосредственном вводе записей во взаимосвязанные таблицы полезно отобразить на экране обе таблицы (рис. 3.33). Это возможно сделать, только открыв **Параметры Access** и отметив в параметрах окна документов флажок **Перекрывание окон** в разделе **Параметры приложений** для текущей базы данных.

Для удобного отображения открытых таблиц можно воспользоваться одной из команд меню **Сверху вниз** (Tile Horizontally), **Слева направо** (Tile Vertically) или **Каскадом** (Cascade) в списке кнопки **Перейти в другое окно** (Switch Windows) на ленте **Главная** (Home) в группе **Окно** (Window).

Отображение записей подчиненных таблиц в главной таблице

Для более удобного ввода данных в записи подчиненной таблицы в Access имеется возможность при просмотре главной таблицы отображать записи подчиненной таблицы, называемой в этом случае *подтаблицей*. Это позволяет пользователю при добавлении, удалении и изменении записей осуществлять контроль корректности связей в отображаемой цепочке таблиц. Например, если отобразить в таблице ДОГОВОР подчиненную таблицу ПОСТАВКА_ПЛАН, связь между которыми осуществляется по полю ном_дог (номер договора), то при заполнении таблицы ПОСТАВКА_ПЛАН невозможно допустить ошибку в поле ном_дог, т. к. оно даже не отображается в подчиненной таблице. Однако это никоим образом не помогает при заполнении других полей связи таблицы ПОСТАВКА_ПЛАН — таких, например, как код_тов (код товара).

При просмотре таблицы, для которой определена подтаблица, в ней отображается столбец со значками «плюс» (рис. 3.34). Достаточно в строке записи щелкнуть на значке «плюс» (+), чтобы отобразились записи подчиненной таблицы, связанные с текущей записью. При этом плюс на значке преобразуется в минус (-). Щелчком на минусе подчиненные записи закрываются. Таким образом могут быть открыты подчиненные записи каждой записи главной таблицы. Если открыть все записи главной таблицы, вы увидите все записи подчиненной таблицы (при условии, что для связи обеспечивается целостность), разбитые на подмножества, связанные с конкретными записями главной таблицы.

В таблице базы данных Access одновременно можно просматривать данные подтаблиц восьми уровней.

Для определения подтаблицы выполните команду **Подтаблица** (Subdatasheet) в списке кнопки **Дополнительно** (More) на вкладке ленты **Главная** (Home) в группе **Записи** (Records). Отобразится окно **Вставка подтаблицы** (Insert Subdatasheet) (рис. 3.35), в котором следует выбрать подчиненную таблицу или запрос и указать поля связи.

Если в главной и подчиненной таблицах поле связи имеет разные имена, то в окне **Вставка подтаблицы** (Insert Subdatasheet) можно ввести имена полей связи, при составном ключе имена полей разделяются точкой с запятой. Если между таблицами

ДОГОВОР			
Номер догов.	Дата заключ.	Код покупателя	Код исполните
D111	11.01.2013	П001	
Код товара	Срок поставки	Количество	СУММА ПО
T001	1	305	2 135 000,00р.
T001	2	200	1 400 000,00р.
T001	3	300	2 100 000,00р.
T002	1	50	25 000,00р.
T002	3	100	50 000,00р.
T003	1	100	112 000,00р.
T005	1	710	568 000,00р.
T007	2	10	24 000,00р.
*	0	0	0,00р.
D222	05.02.2013	П001	
Код товара	Срок поставки	Количество	СУММА ПО
T001	1	100	658 700,00р.
T001	2	12	79 044,00р.
T001	4	10	0,00р.
T002	2	240	0,00р.
T002	4	10	0,00р.
T003	2	510	25 900,00р.
T004	3	30	27 480,00р.
T010	2	400	0,00р.
*	0	0	0,00р.

Рис. 3.34. Отображение подчиненных записей подтаблицы в главной таблице

ДОГОВОР			
Номер догов.	Дата заключ.	Код поку	Код исполнителя
D111	11.01.2013	П001	6
D222	05.02.2013	П001	1
D333			
D555			
D777			
D888			
D999			
*			

Записи: 14

Таблицы: Запросы Таблицы и запросы

ДОГОВОР
ИСПОЛНИТЕЛЬ
НАКЛАДНАЯ
ОТГРУЗКА
ПОКУПАТЕЛЬ
СКЛАД
ТОВАР

Подчиненные поля:

Основные поля:

Рис. 3.35. Окно для выбора подчиненной таблицы и поля связи с ней

ТОВАР и ПОСТАВКА_ПЛАН не было установлено связи в схеме данных, Access автоматически добавит эту связь после утвердительного ответа пользователя в диалоговом окне (рис. 3.36).

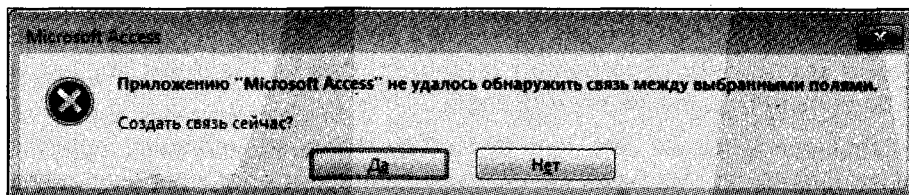


Рис. 3.36. Сообщение системы при определении подтаблицы

Открыть или закрыть все подчиненные записи можно, воспользовавшись командой **Подтаблица | Развернуть все** (Subdatasheet | Expand All) или **Свернуть все** (Collapse All) в списке кнопки **Дополнительно (More)** на ленте **Главная (Home)** в группе **Записи (Records)**.

После сохранения таблицы имя подтаблицы и имена подчиненного и основного полей записываются в свойства главной таблицы.

Для просмотра свойств главной таблицы, например, ДОГОВОР, откройте ее в режиме конструктора и нажмите кнопку **Страница свойств (Property Sheet)**. В строке свойства **Имя подтаблицы (Subdatasheet Name)** указано имя подчиненной таблицы, а в свойствах **Подчиненные поля (Link Child Fields)** и **Основные поля (Link Master Fields)** указано имя ключа связи (простого или составного) (рис. 3.37).

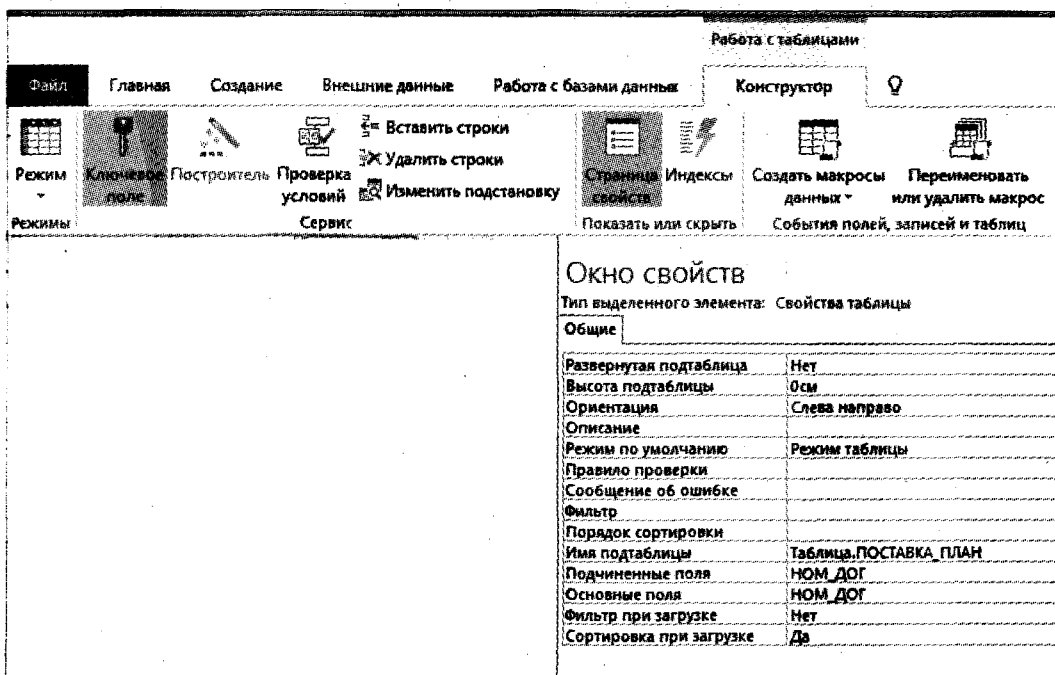


Рис. 3.37. Свойства таблицы ДОГОВОР

В режиме конструктора таблиц можно определить другую подтаблицу, выбрав ее имя из списка в строке свойства **Имя подтаблицы** (Subdatasheet Name) и заменив значения в строках, определяющих связь с ней.

По умолчанию в строке **Имя подтаблицы** (Subdatasheet Name) установлено значение **Авто** (Auto), которое означает, что если в схеме данных установлена связь таблиц, Access автоматически выполнит вывод столбца со значками «плюс» для открытия подчиненных записей по этой связи.

Если у таблицы имеются две или более подчиненные таблицы, определенные в схеме, то при щелчке на знаке «плюс» автоматически откроется окно для выбора подчиненной таблицы. После выбора подчиненной таблицы и сохранения главной таблицы эта связь фиксируется в свойствах.

Для того чтобы в таблице не выводился столбец, позволяющий открывать подчиненные записи, в качестве значения свойства **Имя подтаблицы** (Subdatasheet Name) должно быть установлено **Нет** (None). Значение **Нет** (None) можно установить в режиме таблицы с помощью команды **Подтаблица | Удалить** (Subdatasheet | Remove) в списке кнопки **Дополнительно** (More) на ленте **Главная** (Home) в группе **Записи** (Records).

В подчиненной таблице можно выполнять те же действия, что и в главной: работать с данными и отображать подчиненную более низкого уровня. На рис. 3.38

Код склада	Наименование	Ответственное ли	Адрес склада																																																																					
C01	Главный	Иванов Т.С.	Мичуринская,																																																																					
<table border="1"> <thead> <tr> <th>Номер накладной</th> <th>Дата отгрузки</th> <th>Номер договора</th> <th>Общая сумма</th> </tr> </thead> <tbody> <tr> <td>H001</td> <td>18.01.2010</td> <td>Д111</td> <td>387 338,00р.</td> </tr> <tr> <td colspan="4"> <table border="1"> <thead> <tr> <th>КОД ТОВАРА</th> <th>Количество отгружено</th> <th>Сумма отгрузки</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>50</td> <td>350 000,00р.</td> </tr> <tr> <td>T002</td> <td>10</td> <td>5 000,00р.</td> </tr> <tr> <td>T005</td> <td>2</td> <td>8 000,00р.</td> </tr> <tr> <td>T007</td> <td>10</td> <td>22 000,00р.</td> </tr> <tr> <td>T008</td> <td>10</td> <td>2 338,00р.</td> </tr> <tr> <td>*</td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>H002</td> <td>11.02.2010</td> <td>Д222</td> <td>445 000,00р.</td> </tr> <tr> <td colspan="4"> <table border="1"> <thead> <tr> <th>КОД ТОВАРА</th> <th>Количество отгружено</th> <th>Сумма отгрузки</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>40</td> <td>280 000,00р.</td> </tr> <tr> <td>T002</td> <td>30</td> <td>165 000,00р.</td> </tr> <tr> <td>*</td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>H003</td> <td>27.04.2010</td> <td>Д333</td> <td>42 056,00р.</td> </tr> <tr> <td>H004</td> <td>25.03.2010</td> <td>Д222</td> <td>37 380,00р.</td> </tr> <tr> <td>H005</td> <td>10.01.2010</td> <td>Д222</td> <td>280 000,00р.</td> </tr> <tr> <td>*</td> <td>12.06.2013</td> <td></td> <td>0,00р.</td> </tr> </tbody> </table>				Номер накладной	Дата отгрузки	Номер договора	Общая сумма	H001	18.01.2010	Д111	387 338,00р.	<table border="1"> <thead> <tr> <th>КОД ТОВАРА</th> <th>Количество отгружено</th> <th>Сумма отгрузки</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>50</td> <td>350 000,00р.</td> </tr> <tr> <td>T002</td> <td>10</td> <td>5 000,00р.</td> </tr> <tr> <td>T005</td> <td>2</td> <td>8 000,00р.</td> </tr> <tr> <td>T007</td> <td>10</td> <td>22 000,00р.</td> </tr> <tr> <td>T008</td> <td>10</td> <td>2 338,00р.</td> </tr> <tr> <td>*</td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table>				КОД ТОВАРА	Количество отгружено	Сумма отгрузки	T001	50	350 000,00р.	T002	10	5 000,00р.	T005	2	8 000,00р.	T007	10	22 000,00р.	T008	10	2 338,00р.	*	0	0,00р.	H002	11.02.2010	Д222	445 000,00р.	<table border="1"> <thead> <tr> <th>КОД ТОВАРА</th> <th>Количество отгружено</th> <th>Сумма отгрузки</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>40</td> <td>280 000,00р.</td> </tr> <tr> <td>T002</td> <td>30</td> <td>165 000,00р.</td> </tr> <tr> <td>*</td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table>				КОД ТОВАРА	Количество отгружено	Сумма отгрузки	T001	40	280 000,00р.	T002	30	165 000,00р.	*	0	0,00р.	H003	27.04.2010	Д333	42 056,00р.	H004	25.03.2010	Д222	37 380,00р.	H005	10.01.2010	Д222	280 000,00р.	*	12.06.2013		0,00р.
Номер накладной	Дата отгрузки	Номер договора	Общая сумма																																																																					
H001	18.01.2010	Д111	387 338,00р.																																																																					
<table border="1"> <thead> <tr> <th>КОД ТОВАРА</th> <th>Количество отгружено</th> <th>Сумма отгрузки</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>50</td> <td>350 000,00р.</td> </tr> <tr> <td>T002</td> <td>10</td> <td>5 000,00р.</td> </tr> <tr> <td>T005</td> <td>2</td> <td>8 000,00р.</td> </tr> <tr> <td>T007</td> <td>10</td> <td>22 000,00р.</td> </tr> <tr> <td>T008</td> <td>10</td> <td>2 338,00р.</td> </tr> <tr> <td>*</td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table>				КОД ТОВАРА	Количество отгружено	Сумма отгрузки	T001	50	350 000,00р.	T002	10	5 000,00р.	T005	2	8 000,00р.	T007	10	22 000,00р.	T008	10	2 338,00р.	*	0	0,00р.																																																
КОД ТОВАРА	Количество отгружено	Сумма отгрузки																																																																						
T001	50	350 000,00р.																																																																						
T002	10	5 000,00р.																																																																						
T005	2	8 000,00р.																																																																						
T007	10	22 000,00р.																																																																						
T008	10	2 338,00р.																																																																						
*	0	0,00р.																																																																						
H002	11.02.2010	Д222	445 000,00р.																																																																					
<table border="1"> <thead> <tr> <th>КОД ТОВАРА</th> <th>Количество отгружено</th> <th>Сумма отгрузки</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>40</td> <td>280 000,00р.</td> </tr> <tr> <td>T002</td> <td>30</td> <td>165 000,00р.</td> </tr> <tr> <td>*</td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table>				КОД ТОВАРА	Количество отгружено	Сумма отгрузки	T001	40	280 000,00р.	T002	30	165 000,00р.	*	0	0,00р.																																																									
КОД ТОВАРА	Количество отгружено	Сумма отгрузки																																																																						
T001	40	280 000,00р.																																																																						
T002	30	165 000,00р.																																																																						
*	0	0,00р.																																																																						
H003	27.04.2010	Д333	42 056,00р.																																																																					
H004	25.03.2010	Д222	37 380,00р.																																																																					
H005	10.01.2010	Д222	280 000,00р.																																																																					
*	12.06.2013		0,00р.																																																																					
C02	Оптовый	Петров А.А.	Свердлова, 29																																																																					
C03	Торговый	Смирнов О.Н.	Речной, 38																																																																					

Рис. 3.38. Отображение подтаблиц НАКЛАДНАЯ и ОТГРУЗКА в таблице СКЛАД

в таблице СКЛАД открыты подчиненные записи таблицы НАКЛАДНАЯ и в ней подчиненные записи таблицы ОТГРУЗКА.

Несмотря на то, что в основном через механизм подтаблиц просматриваются связанные записи подчиненных таблиц, система позволяет устанавливать связь и просматривать связанную запись главной таблицы из подчиненной. Например, для подчиненной таблицы ОТГРУЗКА, представленной на рис. 3.38, можно установить связь с главной по отношению к ней таблицей ТОВАР, раскрывающей название товара и другие его характеристики (рис. 3.39).

Код склада	Наименование	Ответственное лиц	Адрес склада																																																																																							
C01	Главный	Иванов Т.С.	Мичуринская, 15																																																																																							
<table border="1"> <thead> <tr> <th>Номер наклад</th> <th>Дата отгруз</th> <th>Номер договора</th> <th>Общая сумма</th> </tr> </thead> <tbody> <tr> <td>H001</td> <td>18.01.2010</td> <td>D111</td> <td>387 338,00р.</td> </tr> <tr> <td colspan="4"> <table border="1"> <thead> <tr> <th>КОД ТОВ</th> <th>Количество отгружено</th> <th>Сумма отгруз</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>50</td> <td>350 000,00р.</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>Монитор 17LG</td> <td>7 000,00р.</td> <td>3%</td> <td>@(3)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table> </td> </tr> <tr> <td>T002</td> <td>10</td> <td>5 000,00р.</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>FDD 3,5</td> <td>500,00р.</td> <td>6%</td> <td>@(1)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table> </td> </tr> <tr> <td>T005</td> <td></td> <td>2</td> <td>8 000,00р.</td> </tr> <tr> <td>T007</td> <td></td> <td>10</td> <td>22 000,00р.</td> </tr> <tr> <td>T008</td> <td></td> <td>10</td> <td>2 338,00р.</td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>H002</td> <td>11.02.2010</td> <td>D222</td> <td>445 000,00р.</td> </tr> <tr> <td>H003</td> <td>27.04.2010</td> <td>D333</td> <td>42 056,00р.</td> </tr> <tr> <td>H004</td> <td>25.03.2010</td> <td>D222</td> <td>37 380,00р.</td> </tr> <tr> <td>H005</td> <td>10.01.2010</td> <td>D222</td> <td>280 000,00р.</td> </tr> <tr> <td>*</td> <td>12.06.2013</td> <td></td> <td>0,00р.</td> </tr> </tbody> </table>				Номер наклад	Дата отгруз	Номер договора	Общая сумма	H001	18.01.2010	D111	387 338,00р.	<table border="1"> <thead> <tr> <th>КОД ТОВ</th> <th>Количество отгружено</th> <th>Сумма отгруз</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>50</td> <td>350 000,00р.</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>Монитор 17LG</td> <td>7 000,00р.</td> <td>3%</td> <td>@(3)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table> </td> </tr> <tr> <td>T002</td> <td>10</td> <td>5 000,00р.</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>FDD 3,5</td> <td>500,00р.</td> <td>6%</td> <td>@(1)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table> </td> </tr> <tr> <td>T005</td> <td></td> <td>2</td> <td>8 000,00р.</td> </tr> <tr> <td>T007</td> <td></td> <td>10</td> <td>22 000,00р.</td> </tr> <tr> <td>T008</td> <td></td> <td>10</td> <td>2 338,00р.</td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table>				КОД ТОВ	Количество отгружено	Сумма отгруз	T001	50	350 000,00р.	<table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>Монитор 17LG</td> <td>7 000,00р.</td> <td>3%</td> <td>@(3)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table>			Наименование	Цена	Ставка НДС	Вложения	Монитор 17LG	7 000,00р.	3%	@(3)	*	0,00р.		@(0)	T002	10	5 000,00р.	<table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>FDD 3,5</td> <td>500,00р.</td> <td>6%</td> <td>@(1)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table>			Наименование	Цена	Ставка НДС	Вложения	FDD 3,5	500,00р.	6%	@(1)	*	0,00р.		@(0)	T005		2	8 000,00р.	T007		10	22 000,00р.	T008		10	2 338,00р.	*		0	0,00р.	H002	11.02.2010	D222	445 000,00р.	H003	27.04.2010	D333	42 056,00р.	H004	25.03.2010	D222	37 380,00р.	H005	10.01.2010	D222	280 000,00р.	*	12.06.2013		0,00р.
Номер наклад	Дата отгруз	Номер договора	Общая сумма																																																																																							
H001	18.01.2010	D111	387 338,00р.																																																																																							
<table border="1"> <thead> <tr> <th>КОД ТОВ</th> <th>Количество отгружено</th> <th>Сумма отгруз</th> </tr> </thead> <tbody> <tr> <td>T001</td> <td>50</td> <td>350 000,00р.</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>Монитор 17LG</td> <td>7 000,00р.</td> <td>3%</td> <td>@(3)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table> </td> </tr> <tr> <td>T002</td> <td>10</td> <td>5 000,00р.</td> </tr> <tr> <td colspan="3"> <table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>FDD 3,5</td> <td>500,00р.</td> <td>6%</td> <td>@(1)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table> </td> </tr> <tr> <td>T005</td> <td></td> <td>2</td> <td>8 000,00р.</td> </tr> <tr> <td>T007</td> <td></td> <td>10</td> <td>22 000,00р.</td> </tr> <tr> <td>T008</td> <td></td> <td>10</td> <td>2 338,00р.</td> </tr> <tr> <td>*</td> <td></td> <td>0</td> <td>0,00р.</td> </tr> </tbody> </table>				КОД ТОВ	Количество отгружено	Сумма отгруз	T001	50	350 000,00р.	<table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>Монитор 17LG</td> <td>7 000,00р.</td> <td>3%</td> <td>@(3)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table>			Наименование	Цена	Ставка НДС	Вложения	Монитор 17LG	7 000,00р.	3%	@(3)	*	0,00р.		@(0)	T002	10	5 000,00р.	<table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>FDD 3,5</td> <td>500,00р.</td> <td>6%</td> <td>@(1)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table>			Наименование	Цена	Ставка НДС	Вложения	FDD 3,5	500,00р.	6%	@(1)	*	0,00р.		@(0)	T005		2	8 000,00р.	T007		10	22 000,00р.	T008		10	2 338,00р.	*		0	0,00р.																																
КОД ТОВ	Количество отгружено	Сумма отгруз																																																																																								
T001	50	350 000,00р.																																																																																								
<table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>Монитор 17LG</td> <td>7 000,00р.</td> <td>3%</td> <td>@(3)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table>			Наименование	Цена	Ставка НДС	Вложения	Монитор 17LG	7 000,00р.	3%	@(3)	*	0,00р.		@(0)																																																																												
Наименование	Цена	Ставка НДС	Вложения																																																																																							
Монитор 17LG	7 000,00р.	3%	@(3)																																																																																							
*	0,00р.		@(0)																																																																																							
T002	10	5 000,00р.																																																																																								
<table border="1"> <thead> <tr> <th>Наименование</th> <th>Цена</th> <th>Ставка НДС</th> <th>Вложения</th> </tr> </thead> <tbody> <tr> <td>FDD 3,5</td> <td>500,00р.</td> <td>6%</td> <td>@(1)</td> </tr> <tr> <td>*</td> <td>0,00р.</td> <td></td> <td>@(0)</td> </tr> </tbody> </table>			Наименование	Цена	Ставка НДС	Вложения	FDD 3,5	500,00р.	6%	@(1)	*	0,00р.		@(0)																																																																												
Наименование	Цена	Ставка НДС	Вложения																																																																																							
FDD 3,5	500,00р.	6%	@(1)																																																																																							
*	0,00р.		@(0)																																																																																							
T005		2	8 000,00р.																																																																																							
T007		10	22 000,00р.																																																																																							
T008		10	2 338,00р.																																																																																							
*		0	0,00р.																																																																																							
H002	11.02.2010	D222	445 000,00р.																																																																																							
H003	27.04.2010	D333	42 056,00р.																																																																																							
H004	25.03.2010	D222	37 380,00р.																																																																																							
H005	10.01.2010	D222	280 000,00р.																																																																																							
*	12.06.2013		0,00р.																																																																																							
C02	Оптовый	Петров А.А.	Свердлова, 29																																																																																							
C03	Торговый	Смирнов О.Н.	Речной, 38																																																																																							
*																																																																																										

Рис. 3.39. Отображение взаимосвязанных записей нескольких таблиц

Задание 3.3. Отображение подтаблиц

Просмотрите подтаблицы всех уровней для таблиц ПОКУПАТЕЛЬ и ТОВАР.

Использование поля со списком при вводе записей

Создание поля со списком в режиме таблицы

Задачу ввода подчиненных записей только при наличии связанной записи в главной таблице можно решить с помощью поля со списком. Для этого достаточно в подчиненной таблице преобразовать поле связи в поле со списком и в качестве источника строк для списка использовать значения ключевого поля из главной таблицы.

Создайте в таблице ПОСТАВКА_ПЛАН для поля ном_дог (номер договора) поле со списком, построенным на основе ключа таблицы ДОГОВОР. Выберите таблицу ПОСТАВКА_ПЛАН в области навигации и откройте ее. Таблица откроется в режиме таблицы и станет доступной вкладка ленты Работа с таблицами | Поля (Table Tools | Fields), на которой представлены команды изменения структуры таблицы, доступные в этом режиме (см. рис. 3.13).

Процесс создания поля со списком в режиме таблицы начните с выполнения команды Подстановка и отношение (Lookup & Relationship) в списке кнопки Другие поля (More Fields) в группе Добавление и удаление (Add & Delete) на ленте Работа с таблицами | Поля. При этом автоматически запускается мастер подстановок.

В открывшемся окне мастера Создание подстановки (Create Lookup) установите тот переключатель, который приведет к построению списка на основе значений из таблицы (рис. 3.40).

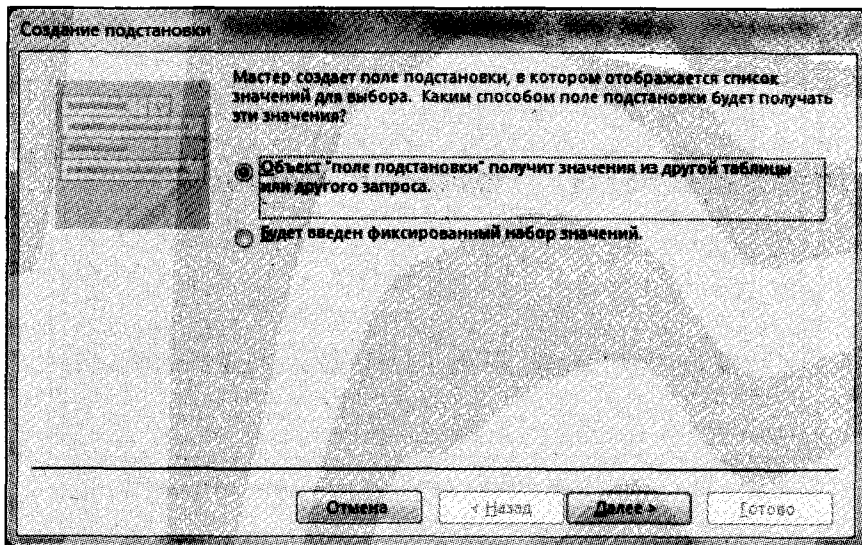


Рис. 3.40. Первое окно мастера подстановок

В следующем окне мастера выберите таблицу ДОГОВОР, на значениях которой будет строиться список поля (рис. 3.41).

Далее выберите столбец, из данных которого будет формироваться список (рис. 3.42).

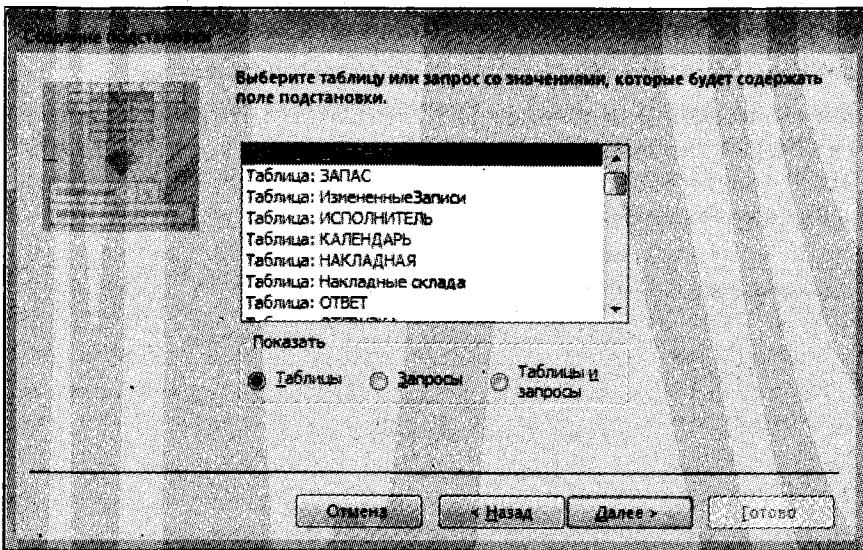


Рис. 3.41. Выбор таблицы, из данных которой будет создаваться список

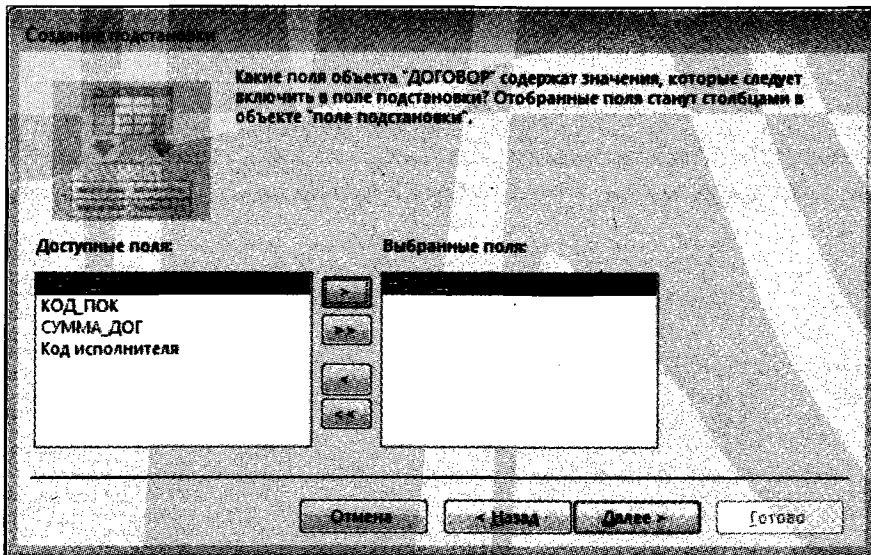


Рис. 3.42. Выбор поля, используемого для формирования списка

В следующих окнах мастера выберите порядок сортировки элементов списка и нужную ширину столбца списка.

Далее мастер предлагает задать имя поля со списком и параметры поддержания целостности для связи таблиц (рис. 3.43) — в нашем примере таблиц ПОСТАВКА_ПЛАН и ДОГОВОР. Связь таблиц сохранится в схеме данных. Для того чтобы увидеть эту связь, нужно открыть схему данных (команда на ленте Работа с таблицами

ми | Таблица | Связи (Table Tools | Table | Relationships)) и в окне **Схема данных** выполнить команду контекстного меню **Отобразить все** (Show All).

На этом создание мастером нового поля со списком завершается.

Теперь при вводе данных в таблицу **ПОСТАВКА_ПЛАН** можно воспользоваться списком, в котором отобразятся все значения поля **НОМ_ДОГ**, содержащиеся в таблице **ДОГОВОР** (рис. 3.44). Выбором нужного значения из списка осуществляется ввод значения в поле номера договора.

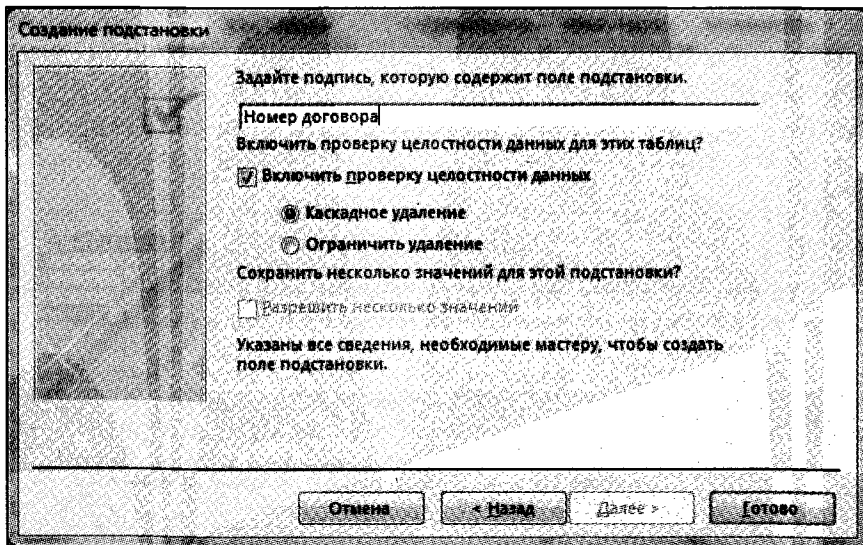


Рис. 3.43. Определение имени поля со списком и проверки связанной целостности данных

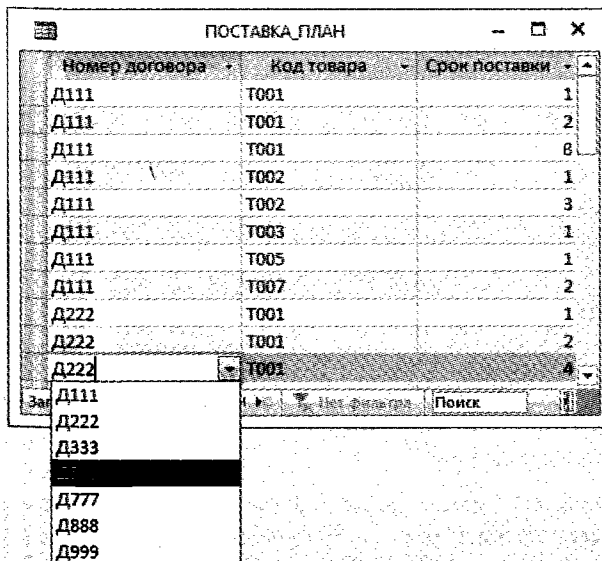


Рис. 3.44. Ввод значения в поле выбором из списка

Если между таблицами не было установлено параметра обеспечения целостности связи таблиц, в поле со списком можно вводить значения, не предусмотренные в списке.

В режиме таблицы предоставляется возможность добавить новое поле с построением для него списка и установлением соответствующего отношения между таблицами и нельзя изменить существующее поле на поле со списком.

Создание поля со списком в режиме таблицы целесообразно при первоначальной разработке структуры таблицы в этом режиме. В противном случае поле, такое как `НОМ_ДОГ`, являющееся полем связи с таблицей `ДОГОВОР`, уже будет создано в таблице, и после создания нового поля со списком его надо будет удалить. В приведенном примере это можно сделать только в режиме конструктора, т. к. поле `НОМ_ДОГ` является частью первичного ключа таблицы. В режиме конструктора поле может быть преобразовано в поле со списком и после разработки структуры таблицы.

Если переключиться в режим конструктора (рис. 3.45), то можно увидеть параметры списка, подготовленные мастером в свойствах нового поля на вкладке **Подстановка** (Lookup).

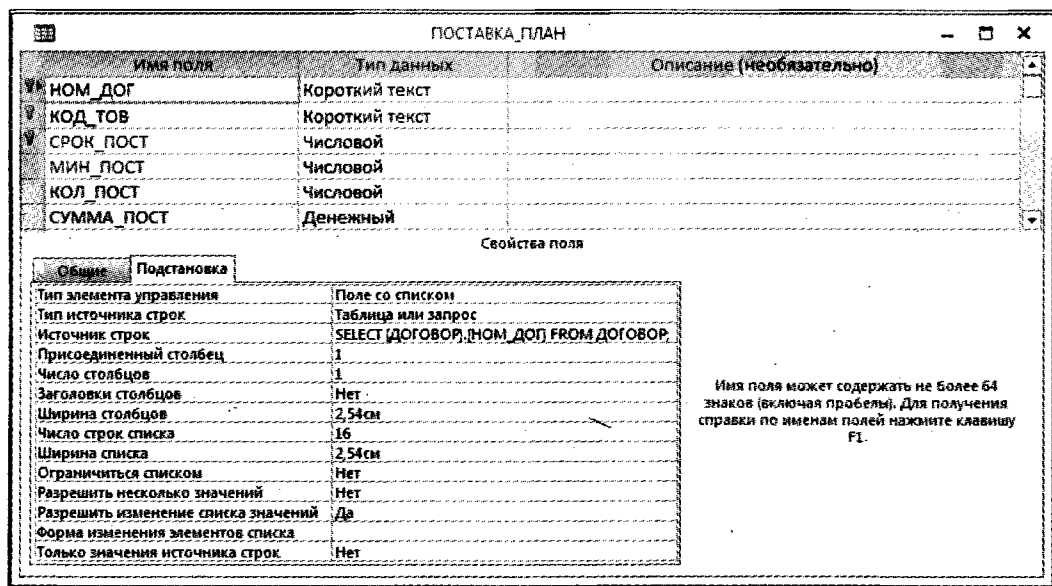


Рис. 3.45. Свойства поля со списком для выбора номера договора

Создание поля со списком в режиме конструктора

Выберите таблицу `ПОСТАВКА_ПЛАН` в области навигации и в контекстном меню таблицы выполните команду **Конструктор** (Design View). Установите курсор в поле `НОМ_ДОГ` на столбце **Тип данных** (Data Type), откройте список и выберите строку **Мастер подстановок** (Lookup Wizard). Откроется окно мастера **Создание**

подстановки (Create Lookup) (см. рис. 3.40). Работа с мастером, рассмотренная в предыдущем разделе, приведет к преобразованию поля `НОМ_ДОГ` в поле со списком. Параметры списка, подготовленные мастером, отобразятся в свойствах поля `НОМ_ДОГ` на вкладке **Подстановка (Lookup)** (см. рис. 3.45).

ЗАМЕЧАНИЕ

Если в базе данных уже создана схема данных и установлена связь таблицы `ПОСТАВКА_ПЛАН` с таблицей `ДОГОВОР`, перед изменением типа данных поля `НОМ_ДОГ` нужно удалить эту связь.

Теперь при вводе данных в таблицу `ПОСТАВКА_ПЛАН` можно воспользоваться созданным списком (см. рис. 3.44).

Ввод в поле `НОМ_ДОГ` ограничивается значениями из списка только в случае, если при работе мастера или в схеме вручную были проставлены параметры обеспечения целостности связи таблиц. Однако, если при определении параметров списка на вкладке **Подстановка (Lookup)** в окне конструктора таблицы (см. рис. 3.45) выбрать в строке **Ограничиться списком (Limit to List)** значение **Да**, ввод значений, не предусмотренных в списке, также сделается невозможным. В случае отклонения от значений списка система выдаст сообщение, показанное на рис. 3.46. Таким образом, пользователь вынужден будет придерживаться только значений, представленных в списке, и в подчиненную таблицу будут введены только связанные записи. То есть, использование списков не только обеспечивает удобный ввод, но и позволяет избежать многих ошибок.

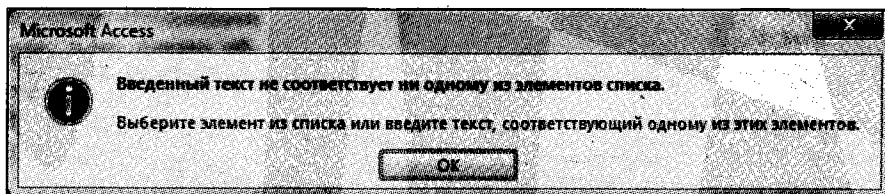


Рис. 3.46. Сообщение о неверно введенном значении

Задание 3.4. Создание поля со списком для существующего поля

Преобразуйте в таблице `ПОСТАВКА_ПЛАН` поле `КОД_ТОВ` (код товара) в поле со списком. Для этого откройте таблицу в режиме конструктора, установите курсор в поле `КОД_ТОВ` на столбце **Тип данных (Data Type)** и в списке этого столбца выберите строку **Мастер подстановок (Lookup Wizard)**. В диалоге с мастером выберите главную по отношению к таблице `ПОСТАВКА_ПЛАН` таблицу `ТОВАР`, на основе данных которой создается список, и поля, включаемые в список: `КОД_ТОВ` — код товара и `НАИМ_ТОВ` — наименование товара. Кроме того, оставьте, как предлагает мастер, помеченным флажок **Скрыть ключевой столбец (рекомендуется) (Hide key column (recommended))**. В результате работы мастера таблица должна приобрести вид, показанный на рис. 3.47.

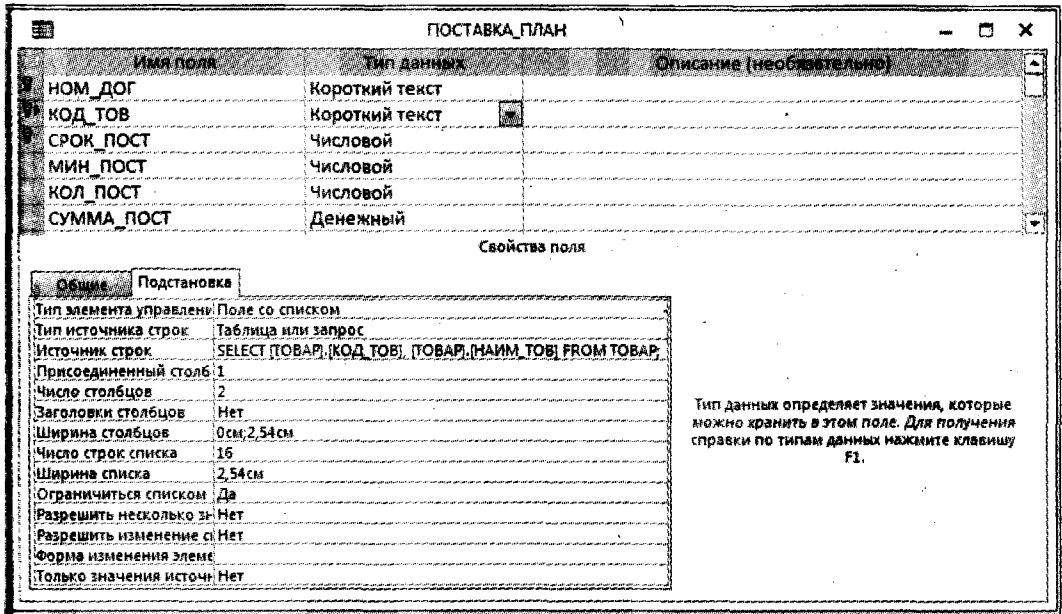


Рис. 3.47. Свойства поля со списком, созданного мастером подстановок для кода товара

При выборе параметра **Скрыть ключевой столбец (рекомендуется)** для первого столбца списка устанавливается ширина, равная нулю. При этом параметр **Ограничиться списком (Limit to List)** может принимать только одно значение — **Да** (рис. 3.48).

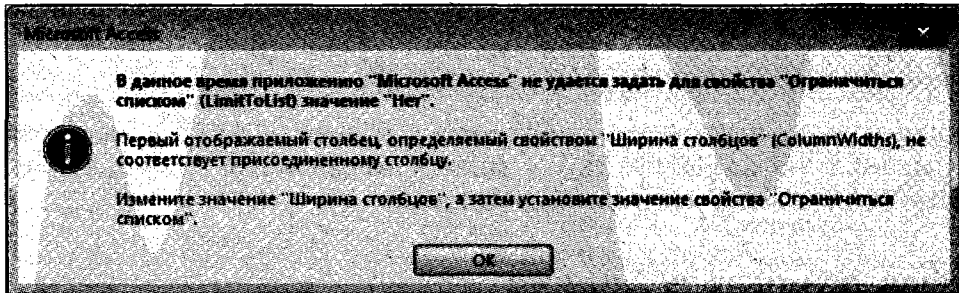


Рис. 3.48. Сообщение при попытке изменить свойство Ограничиться списком

Определение поля со списком с такими параметрами приводит к отображению в поле КОД_ТОВ вместо значений ключевого поля значений второго поля списка — поля НАИМ_ТОВ (наименование товара), взятого из главной таблицы ТОВАР. Тем не менее, выбор наименования товара приведет к вводу соответствующего значения ключа в поле КОД_ТОВ. На рис. 3.49 показано, как в таблице ПОСТАВКА_ПЛАН отображается поле со списком Код товара при нулевой ширине присоединенного столбца.

Таким образом, использование полей со списком не только упрощает ввод данных, но и помогает избежать ошибок при вводе связанных записей в таблицы как за счет

того, что в таблицах вместо ключей могут быть отображены смысловые значения, так и за счет того, что ввод ограничивается набором значений в списке.

ОБРАТИТЕ ВНИМАНИЕ

При создании полей со списком мастер подстановки автоматически формирует связь между таблицей, в которой создается поле со списком, и таблицей, являющейся источником значений списка. Эту связь можно увидеть в Схеме данных (Relationships), которая открывается выполнением команды **Схема данных (Relationships)** в группе **Отношения** на вкладке ленты **Работа с базами данных (Database Tools)**.

Номер договора	Код товара	Сроки поставки	Количество
Д111	Монитор 17LG		305
Д111	Монитор 17LG		200
Д111	Монитор 17LG		300
Д111			50
Д111	FDD 3,5		100
Д111	HDD Maxtor 120GB		100
Д111	Корпус MiniTower		710
Д111	CD-ROM Panasonic IDE		10
Д222	DIMM 64M PC100		100
Д222	Принтер EPSON ST.A4		12
	СканерAcer		
	Зв. Карта Genius Liv		
	Модем Genius ext		
	Плоттер		
	Миникомпьютер		

Рис. 3.49. Отображение поля со списком Код товара при нулевой ширине присоединенного столбца

Задание 3.5. Создание полей со списком

Создайте поле со списком для отображения в таблице **ДОГОВОР** наименований покупателей, в таблице **ОТГРУЗКА** — кодов и наименований товаров, а в таблице **НАКЛАДНАЯ** — кодов и наименований складов.

Схема данных в Access

Реляционная база данных, созданная в соответствии с проектом канонической модели данных предметной области, состоит из нормализованных таблиц, связанных одно-многочисленными отношениями. В такой базе данных обеспечивается отсутствие дублирования описательных данных, их однократный ввод, поддержание целостности данных средствами системы. Связи между таблицами позволяют выполнить объединение данных различных таблиц, необходимое для решения большинства задач ввода, просмотра и корректировки данных, получения информации по запросам и вывода отчетов.

Связи между таблицами устанавливаются в соответствии с проектом логической структуры базы данных (см. рис. 2.19) и запоминаются в схеме данных Access.

Схема данных является не только средством графического отображения логической структуры базы данных, она активно используется системой в процессе обработки данных. Благодаря связям, установленным в схеме данных, разработчику нет необходимости всякий раз сообщать системе о наличии той или иной связи. Однажды указанные в схеме данных связи используются системой автоматически.

Создание схемы данных позволяет упростить конструирование многотабличных форм, запросов, отчетов, а также обеспечить поддержание целостности взаимосвязанных данных при вводе и корректировке данных в таблицах.

Создание схемы данных

Рассмотрим процесс создания схемы данных, соответствующей логической структуре базы данных «Поставка товаров», разработка которой рассмотрена в *главе 2*. Создание схемы данных начинается с выполнения команды **Схема данных (Relationships)** в группе **Отношения** на вкладке ленты **Работа с базами данных (Database Tools)**. В результате выполнения этой команды открывается окно схемы данных и диалоговое окно **Добавление таблицы (Show Table)**, в котором осуществляется выбор таблиц, включаемых в схему (рис. 3.50). Диалоговое окно **Добавление таблицы** откроется автоматически, если в базе данных еще не определена ни одна связь. Если окно не открылось, на ленте **Работа со связями | Конструктор (Relationship Tools | Design)** в группе **Связи (Relationships)** нажмите кнопку **Отобразить таблицу (Show Table)**.

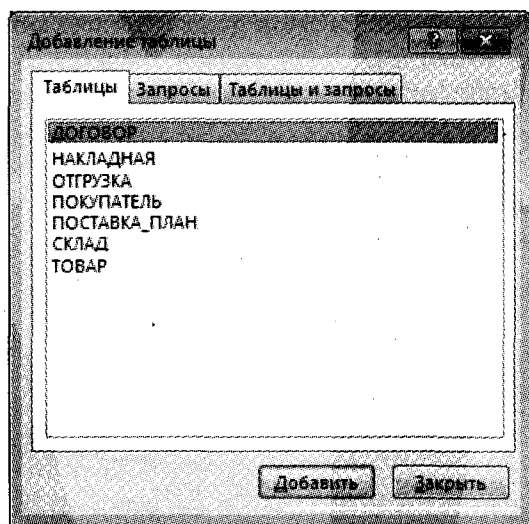


Рис. 3.50. Добавление таблиц в схему данных

Включение таблиц в схему данных

В окне **Добавление таблицы (Show Table)** (см. рис. 3.50) отображены все таблицы и запросы, содержащиеся в базе данных. Выберем вкладку **Таблицы (Tables)** и с помощью кнопки **Добавить (Add)** разместим в окне **Схема данных (Rela-**

tionships) все ранее созданные таблицы базы данных «Поставка товаров», отображенные в окне **Добавление таблицы (Show Table)**. Затем нажмем кнопку **Закрыть (Close)**. В результате в окне **Схема данных (Relationships)** таблицы базы будут представлены окнами со списками своих полей и отмеченными ключами (потом, после добавления всех необходимых связей, мы увидим это на рис. 3.54). Теперь можно приступить к созданию связей между таблицами.

Создание связей между таблицами схемы данных

При создании связей в схеме данных используется проект логической структуры реляционной базы данных, в котором показаны все одно-многочленные связи таблиц. Реализуются связи с помощью добавления в связанные таблицы общих полей, называемых *ключом связи*. При одно-многочленных отношениях между таблицами ключом связи является ключ главной таблицы (простой или составной). В подчиненной таблице он может быть частью уникального ключа или вовсе не входить в состав ключа таблицы. Одно-многочленные связи являются основными в реляционных базах данных. Одно-однозначные связи используются, как правило, при необходимости распределять большое количество полей, определяемых одним и тем же ключом, по разным таблицам, имеющим разный регламент обслуживания.

Создание связей по простому ключу

Установим связь между таблицами **ПОКУПАТЕЛЬ** и **ДОГОВОР**, которые находятся в отношении «один-ко-многим». Устанавливая связи между парой таблиц, находящихся в отношении типа 1 : M, выделим в главной таблице **ПОКУПАТЕЛЬ** ключевое поле **КОД_ПОК**, по которому устанавливается связь. Далее при нажатой кнопке мыши перетащим его в соответствующее поле подчиненной таблицы **ДОГОВОР**. Откроется окно **Изменение связей (Edit Relationships)** (рис. 3.51).

Поскольку поле связи является уникальным ключом в главной таблице, а в подчиненной таблице не является ключевым, Access выявляет отношение «один-ко-многим» между записями этих таблиц. Значение «один-ко-многим» (One-To-Many) отобразится в строке **Тип отношения (Relationship Type)**.

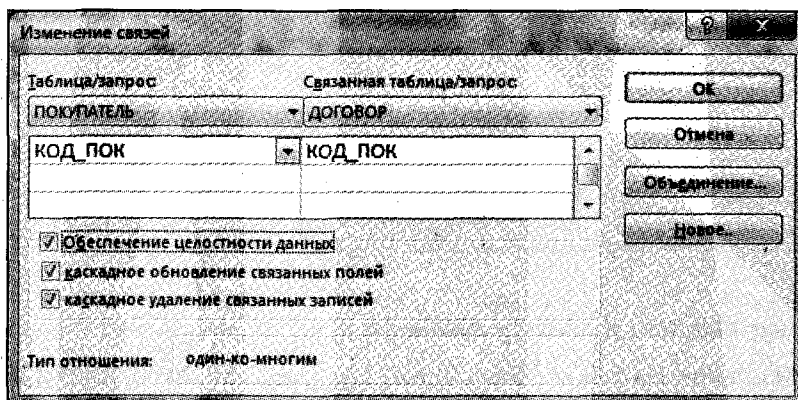


Рис. 3.51. Окно определения параметров связи

ЗАМЕЧАНИЕ

Если поле связи является уникальным ключом в обеих связываемых таблицах, Access выявляет отношение «один-к-одному». Если для связи таблиц вместо ключевого поля главной таблицы используется некоторый уникальный индекс, система также констатирует отношение таблиц как 1 : М или 1 : 1.

Задание 3.6. Создание связей по простому ключу

Добавьте в схему данных и установите связь по простому ключу для других пар таблиц базы данных «Поставка товаров»: СКЛАД → НАКЛАДНАЯ (ключ код_ск), ДОГОВОР → ПОСТАВКА_ПЛАН (ключ ном_дог), ТОВАР → ОТГРУЗКА (ключ код_тов), ТОВАР → ПОСТАВКА_ПЛАН (ключ код_тов), ДОГОВОР → НАКЛАДНАЯ (ключ ном_дог).

ЗАМЕЧАНИЕ

Кнопка Новое (Create New) в окне Изменение связей (Edit Relationships) позволяет перейти к созданию связи между любыми двумя таблицами базы, не выходя в окно схемы данных.

Определение связей по составному ключу

Определим связи между таблицами НАКЛАДНАЯ → ОТГРУЗКА, которые связаны по составному ключу ном_накл + код_ск. Для этого в главной таблице НАКЛАДНАЯ выделим оба этих поля, нажав клавишу <Ctrl>, и перетащим их в подчиненную таблицу ОТГРУЗКА.

В окне Изменение связей (Edit Relationships) (рис. 3.52) для каждого поля составного ключа главной таблицы НАКЛАДНАЯ, названной здесь Таблица/запрос (Table/Query), выберем соответствующее поле подчиненной таблицы ОТГРУЗКА, названной здесь Связанная таблица/запрос (Related Table/Query).

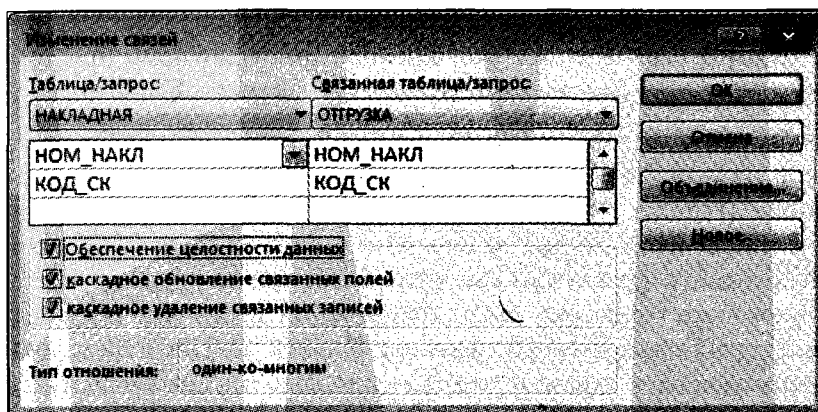


Рис. 3.52. Окно определения связи между таблицами НАКЛАДНАЯ и ОТГРУЗКА

ЗАМЕЧАНИЕ

Для определения связи по суррогатному первичному ключу таблицы с типом данных Счетчик (AutoNumber) необходимо, чтобы в подчиненной таблице был определен ключ связи с типом данных Числовой (Number) и размером Длинное целое (Long Integer).

Обеспечение целостности данных

Если связываемые в схеме данных таблицы находятся в отношении 1 : 1 и 1 : М, для связи можно задать параметр обеспечения связной целостности данных. Обеспечение связной целостности данных означает, что Access при корректировке данных связанных таблиц контролирует соблюдение следующих условий:

- в подчиненную таблицу не может быть добавлена запись с несуществующим в записях главной таблицы значением ключа связи;
- в главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;
- изменение значений ключа связи в записи главной таблицы невозможно, если подчиненная таблица имеет связанные с ней записи, в которых соответственно не меняется ключ связи.

При попытке пользователя нарушить эти условия в операциях добавления и удаления записей или обновления ключевых данных в связанных таблицах, Access не допускает выполнения операции и выводит соответствующее сообщение. Access не позволяет установить параметр целостности для связи таблиц, если ранее введенные в таблицы данные не отвечают требованиям целостности.

Установление между двумя таблицами связи и задание для нее параметров целостности данных возможно только при следующих условиях:

- связываемые поля имеют одинаковый тип данных, при этом имена полей могут быть различными;
- обе таблицы сохраняются в одной базе данных Access;
- главная таблица связывается с подчиненной по первичному простому или составному ключу либо по уникальному индексу главной таблицы.

Для каждой из ранее установленных связей базы данных «Поставка товаров» установим в окне **Изменение связей** (Edit Relationships) флажок **Обеспечение целостности данных** (Enforce Referential Integrity). Установление связи с параметром обеспечения целостности данных возможно только в том случае, если таблицы ранее были заполнены корректными данными. В противном случае появится сообщение о невозможности установить связь. Например, если эти условия целостности не соблюдены для ранее загруженных данных таблиц ПОКУПАТЕЛЬ и ДОГОВОР, то при попытке установить параметр обеспечения целостности выдается сообщение, показанное на рис. 3.53.

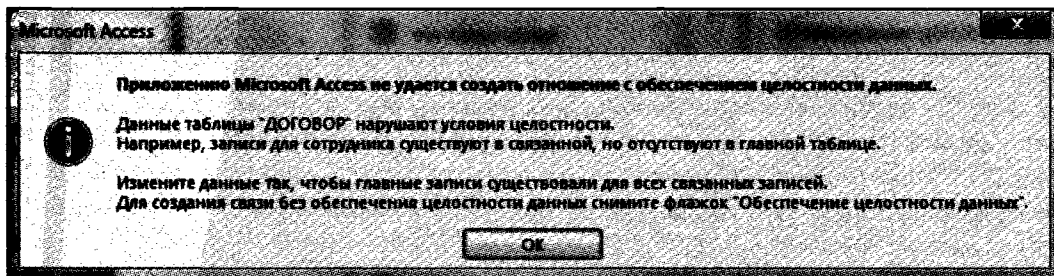


Рис. 3.53. Сообщение о невозможности установить параметр обеспечения целостности для связи

Каскадное обновление и удаление связанных записей

Если для выбранной связи обеспечивается поддержание целостности, можно задать режим *каскадного удаления связанных записей* и режим *каскадного обновления связанных полей*. Такие параметры делают возможным в главной таблице, соответственно, удаление записей и изменение значения в ключевом поле, т. к. при этих параметрах система автоматически выполнит необходимые изменения в подчиненных таблицах, обеспечив сохрание свойств целостности базы данных.

- В режиме *каскадного удаления связанных записей* при удалении записи из главной таблицы будут автоматически удаляться все связанные записи в подчиненных таблицах. При удалении записи из главной таблицы выполняется каскадное удаление подчиненных записей на всех уровнях, если этот режим задан на каждом уровне.
- В режиме *каскадного обновления связанных полей* при изменении значения ключевого поля в записи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях.

Установить в окне **Изменение связей** (Edit Relationships) (см. рис. 3.51) флажки **каскадное обновление связанных полей** (Cascade Update Related Fields) и **каскадное удаление связанных записей** (Cascade Delete Related Records) можно только после задания параметра обеспечения целостности данных.

После создания связей изображения таблиц могут перемещаться в пределах рабочего пространства окна схемы данных. Перемещения и изменения размеров окон со списками полей таблиц в окне схемы данных осуществляются принятыми в Windows способами. На рис. 3.54 показана схема базы данных «Поставка товаров», где таблицы размещены в соответствии с размещением информационных объектов канонической модели данных, полученной при проектировании базы данных в главе 2 (см. рис. 2.18).

В созданной схеме данных БД «Поставка товаров» (см. рис. 3.54) все связи отмечены символами 1 или ∞. Это свидетельствует о том, что одно-многочленные связи установлены правильно (по-простому или составному ключу), и для них задан параметр обеспечения целостности данных.

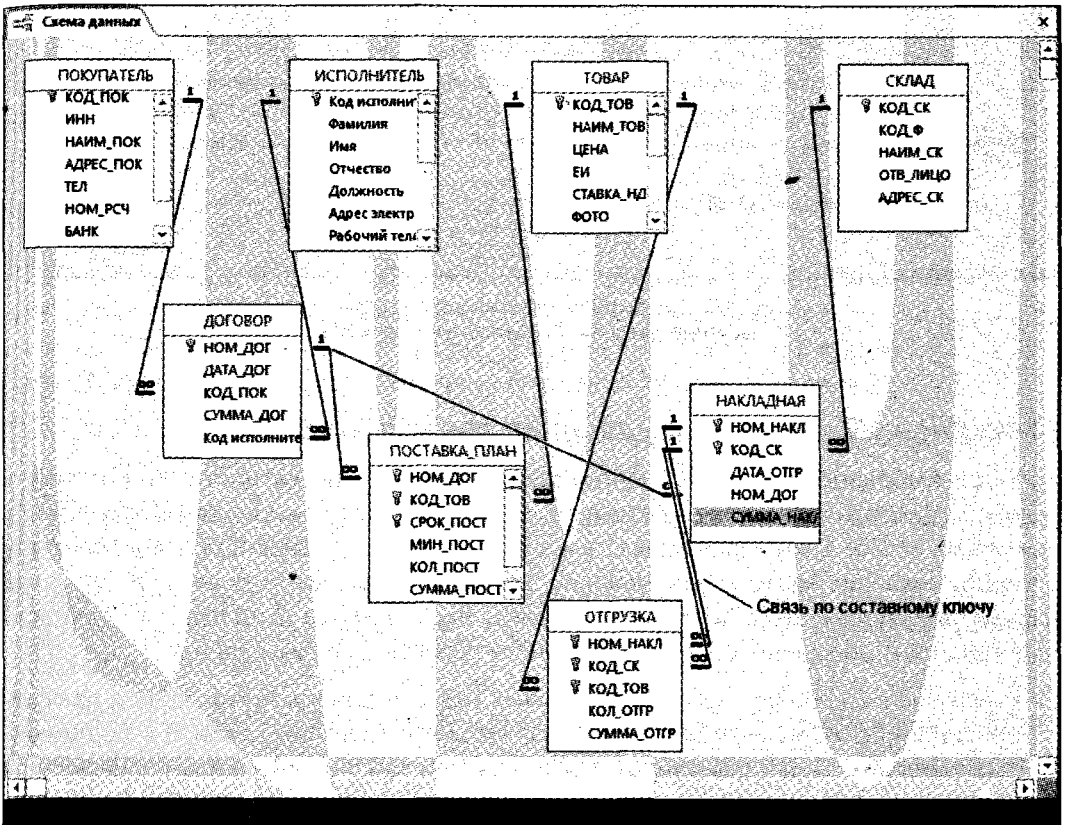


Рис. 3.54. Схема данных БД «Поставка товаров»

Задание 3.7. Проверка поддержания целостности в базе данных

Если в схеме данных определена связь таблиц, и для нее установлены параметры обеспечения целостности, при вводе и корректировке данных в таких взаимосвязанных таблицах пользователь не сможет определить записи, нарушающие требования целостности. Проверьте, как обеспечивается поддержание целостности при внесении изменений в таблицы ПОКУПАТЕЛЬ—ДОГОВОР, связанные одномногочисленными отношениями.

Проверка автоматического поддержания целостности при изменении значений ключей связи в таблицах

Откройте главную таблицу связи ПОКУПАТЕЛЬ в режиме таблицы. Измените значение ключевого поля код_пок (код покупателя) в одной из записей. Убедитесь, что во всех записях подчиненной таблицы ДОГОВОР для договоров, заключенных с этим покупателем, автоматически также изменится значение поля код_пок. Изменение происходит, т. к. был установлен флажок **каскадное обновление связанных полей** (Cascade Update Related Fields) (см. рис. 3.51). Причем это изменение осуществляется мгновенно, как только изменяемая запись перестает быть текущей.

Проверка при добавлении записей в подчиненную таблицу

Убедитесь, что невозможно включить новую запись в подчиненную таблицу ДОГОВОР со значением ключа связи код_пок, не представленным в таблице ПОКУПАТЕЛЬ. Измените значение ключа связи код_пок в подчиненной таблице ДОГОВОР на значение, не существующее в записях таблицы ПОКУПАТЕЛЬ, и убедитесь, что такое изменение запрещено, т. к. при поддержании целостности не может существовать запись подчиненной таблицы с ключом связи, которого нет в главной таблице.

Проверка при удалении записи в главной таблице

Убедитесь, что вместе с удалением записи в главной таблице ПОКУПАТЕЛЬ удаляются все подчиненные записи в таблице ДОГОВОР, т. к. был установлен флажок **каскадное удаление связанных записей** (Cascade Delete Related Records).

Заметим, что если каскадное удаление не разрешено, невозможно удалить запись в главной таблице, если имеются связанные с ней записи в подчиненной.

Контрольные вопросы

1. В файле какого типа сохраняется база данных Access 2016?
2. Какие объекты представлены в области навигации базы данных?
3. Чем определяется состав полей таблицы?
4. Чем определяется структура таблицы?
5. В каком режиме наиболее полно определяются параметры структуры таблицы?
6. В каком свойстве поля задаются ограничения на его значения?
7. На что влияет свойство **Формат поля** (Format)?
8. Что формирует система на основе заданного первичного ключа?
9. Для чего служат индексы таблицы?
10. Значение какого свойства поля используется при отображении заголовка столбца таблицы?
11. Может ли ключ иметь повторяющиеся значения?
12. Можно ли в строке таблицы не вводить значение ключа?
13. Могут ли в таблице повторяться значения в одном из полей составного ключа?
14. Какой тип данных имеет суррогатный ключ?
15. Какой тип данных должен иметь ключ связи в подчиненной таблице, если главная таблица имеет суррогатный ключ?
16. В каком режиме осуществляется ввод данных в таблицу?
17. Назовите кнопки перехода по записям таблицы.
18. В каком режиме визуально выполняется настройка макета таблицы?

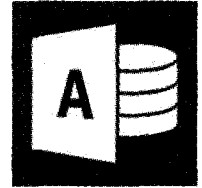
19. С помощью какой команды выполняется внедрение объекта в поле с типом данных OLE?
20. Для чего создается схема данных базы Access?
21. В каких отношениях должны находиться таблицы, чтобы для них можно было установить параметры обеспечения связной целостности данных?
22. По какому полю должна быть установлена связь между таблицами, чтобы появилась возможность установить параметры обеспечения связной целостности данных?
23. Можно ли в подчиненную таблицу ДОГОВОР ввести запись о договоре с покупателем, который не представлен в таблице ПОКУПАТЕЛЬ, если для этих таблиц обеспечивается целостность данных?
24. Можно ли удалить запись о покупателе, если в таблице ДОГОВОР представлены записи о договорах с этим покупателем и не установлен флажок каскадное удаление связанных записей (Cascade Delete Related Records)?
25. Что произойдет при изменении значения ключевого поля в главной таблице, если для ее связи с подчиненной установлен флажок каскадное обновление связанных полей (Cascade Update Related Fields)?
26. Для чего предназначен значок + (плюс) в левом столбце открытой таблицы?
27. Какая команда позволяет открыть в главной таблице связанные записи нужной подчиненной таблицы?
28. Какое свойство таблицы определяет подтаблицу, из которой выводятся связанные записи при щелчке на значке + (плюс) в левом столбце открытой таблицы?

Ответы

1. ACCDB.
2. Таблицы, запросы, формы, отчеты, макросы и модули.
3. Реквизитным составом информационного объекта (сущности) информационно-логической модели предметной области.
4. Составом ее полей, для которых заданы имя, тип данных и свойства.
5. В режиме конструктора.
6. **Правило проверки (Validation Rule).**
7. На вид отображаемого данного.
8. Уникальный индекс с именем **PrimaryKey**.
9. Для более быстрого поиска записи (записей) по заданному значению индексируемого поля (полей).
10. **Подпись (Caption).**
11. Нет.

12. Нет.
13. Да.
14. Суррогатный ключ имеет тип данных **Счетчик (AutoNumber)**.
15. **Числовой (Number)** с размером **Длинное целое (Long Integer)**.
16. В режиме таблицы.
17. **Первая запись (First Record)**, **Предыдущая запись (Previous Record)**, **Следующая запись (Next Record)**, **Последняя запись (Last Record)**, **Новая (пустая) запись (New (blank) Record)**.
18. В режиме таблицы.
19. С помощью команды контекстного меню **Вставить объект (Insert Object)**.
20. Схема данных наглядно отображает связи таблиц базы. Установленные в схеме связи автоматически используются при обработке связанных таблиц. В схеме данных можно запросить проверку связной целостности данных, задать способ объединения записей.
21. 1 : M или 1 : 1.
22. По первичному (простому или составному) ключу главной таблицы или уникальному индексу главной таблицы.
23. Нет.
24. Нет.
25. Обновятся значения связанных полей во всех подчиненных записях.
26. Для открытия записей выбранной подтаблицы.
27. Подтаблица (Subdatasheet), размещенная в списке кнопки **Дополнительно (More)** на вкладке ленты **Главная (Home)** в группе **Записи (Records)**.
28. **Имя подтаблицы (Subdatasheet Name)**.

ГЛАВА 4



Запросы

Запросы являются основным инструментом выборки, обновления и обработки данных в таблицах базы данных.

Access в соответствии с концепцией реляционных баз данных для выполнения запросов использует язык структурированных запросов SQL (Structured Query Language). С помощью инструкций языка SQL в Access реализуется любой запрос.

В то же время Access позволяет создавать запросы, не прибегая к записи инструкций языка SQL. Простейшие запросы могут быть созданы с помощью мастера, практически любой запрос можно создать в режиме графического конструктора. При создании запроса этими средствами Access сам автоматически создает эквивалентную инструкцию SQL, которую можно увидеть, переключившись в режим SQL. Конструктор позволяет создавать запросы простым и удобным способом, а просмотр этих запросов в режиме SQL — понять и освоить синтаксис основных инструкций языка SQL, реализованного в Access.

Основным видом запроса является *запрос на выборку*. Результатом выполнения этого запроса становится новая таблица, которая существует до закрытия запроса. Структура такой таблицы определяется выбранными из одной или нескольких взаимосвязанных таблиц полями. Записи формируются путем объединения записей таблиц, на которых построен запрос. Способ объединения записей таблиц указывается при определении их связи в схеме данных или при создании запроса. Условия отбора, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц. Запрос на выборку позволяет сформировать пользовательское представление о данных, не обязательно отвечающее требованиям нормализации.

В Access может быть создано несколько видов запроса:

□ *запрос на выборку* — выбирает данные из одной таблицы (запроса) или нескольких взаимосвязанных таблиц и других запросов. Результатом является таблица, которая существует до закрытия запроса. Структура записи таблицы запроса определяется набором выбранных полей. Записи таблицы запроса образуются путем объединения записей таблиц, выбранных в запросе, и в соответствии с заданными условиями отбора;

- *запрос на создание таблицы* — выбирает данные из взаимосвязанных таблиц и других запросов, но, в отличие от запроса на выборку, результат сохраняет в новой постоянной таблице;
- *запросы на обновление, добавление, удаление* — являются запросами действия, в результате выполнения которых изменяются данные в таблицах.

Запрос в режиме конструктора содержит *схему данных*, отображающую используемые таблицы, и *бланк запроса*, в котором конструируется структура таблицы запроса и условия выборки записей (рис. 4.1).

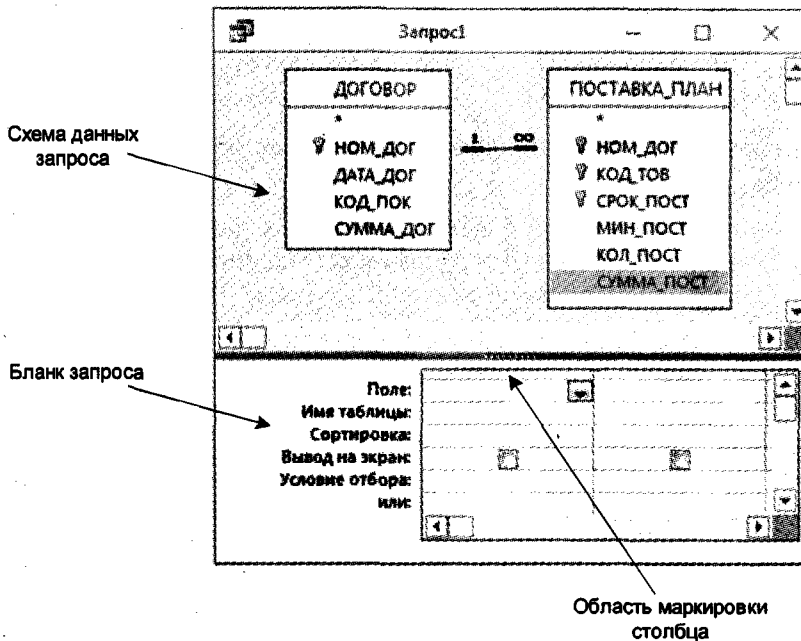


Рис. 4.1. Окно конструктора запросов

С помощью запроса можно выполнить следующие виды обработки данных:

- включить в таблицу запроса выбранные пользователем поля таблицы;
- выбрать записи, удовлетворяющие условиям отбора;
- сформировать на основе объединения записей взаимосвязанных таблиц новую виртуальную таблицу;
- произвести вычисления в каждой из полученных записей;
- сгруппировать записи, которые имеют одинаковые значения в одном или нескольких полях, одновременно выполнить над другими полями группы статистические функции и в результате включить одну запись для каждой группы;
- создать новую таблицу базы данных, используя данные из существующих таблиц;
- произвести обновление полей в выбранном подмножестве записей;

- удалить выбранное подмножество записей из таблицы базы данных;
- добавить выбранное подмножество записей в другую таблицу.

Последовательное выполнение ряда запросов позволяет решать достаточно сложные задачи, не прибегая к программированию.

Запросы служат источниками записей для других запросов, форм, отчетов. С помощью запроса можно собрать из нескольких таблиц полные сведения для формирования некоторого документа предметной области. Далее можно использовать его, например, для создания формы — электронного представления этого документа. Если форма или отчет создаются мастером на основе нескольких взаимосвязанных таблиц, для которых определены связи в схеме данных, то для них в качестве источника записей автоматически формируется запрос. Ненормализованная с повторяющимися данными таблица запроса не хранится в базе данных — хранится только сам запрос, который может быть использован различными пользователями для выполнения своих задач.

Однотабличные запросы на выборку

В простейшем случае запрос реализует выбор из одной таблицы нужных полей, записей, соответствующих заданным условиям отбора, и просмотр результатов выполнения запроса.

Конструирование запросов на выборку с условиями отбора

Рассмотрим процесс конструирования однотабличного запроса на выборку на примере получения информации из таблицы ТОВАР базы данных «Поставка товаров».

Задача 1

Пусть необходимо выбрать ряд характеристик товара по его наименованию.

1. Для создания запроса в окне базы данных выберите вкладку ленты **Создание** (Create) и в группе **Запросы** нажмите кнопку **Конструктор запросов** (Design View) — откроется пустое окно запроса на выборку в режиме конструктора **ЗапросN** (QueryN) и диалоговое окно **Добавление таблицы** (Show Table) (рис. 4.2).
2. В окне **Добавление таблицы** (Show Table) выберите таблицу ТОВАР и нажмите кнопку **Добавить** (Add) — выбранная таблица будет отображена в области схемы данных запроса. Закройте окно **Добавление таблицы** (Show Table), нажав кнопку **Закрывать** (Close).

В результате выполненных действий в верхней панели окна конструктора запросов (см. рис. 4.1) появится *схема данных запроса*, которая включает выбранные для этого запроса таблицы. В нашем случае — одну таблицу ТОВАР. Таблица представлена списком полей. Первая строка в списке полей таблицы, отмечен-

ная звездочкой (*), обозначает все множество полей таблицы. Нижняя панель представляет собой *бланк запроса*, который нужно заполнить.

Кроме того, на ленте появляется и автоматически активизируется новая вкладка **Работа с запросами | Конструктор (Query Tools | Design)** (на рис. 4.3 представлена часть этой вкладки), на которой цветом выделен тип создаваемого запроса — **Выборка (Select)**. По умолчанию первоначально всегда создается запрос на выборку. Команды этой вкладки представляют инструментарий для выполнения необходимых действий при создании запроса. Вкладка открывается, когда в режиме конструктора создается новый запрос или редактируется существующий.

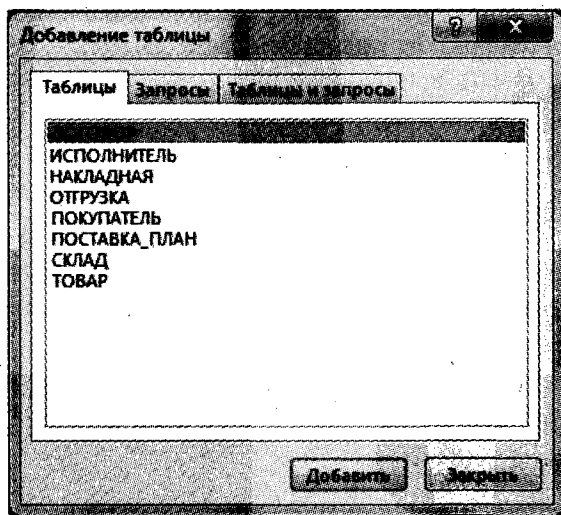


Рис. 4.2. Окно выбора таблиц и запросов для схемы данных запроса

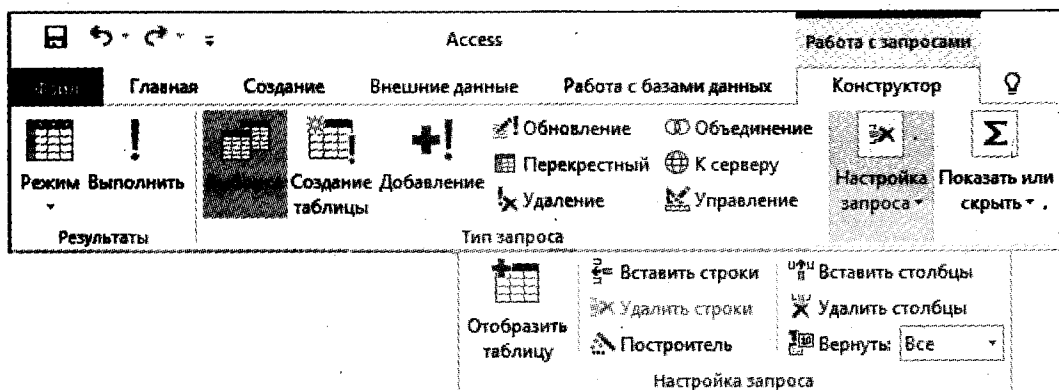



Рис. 4.3. Лента инструментов конструктора запросов

- Для удаления любой таблицы из схемы данных запроса установите на нее курсор мыши и нажмите клавишу <Delete>. Для добавления — нажмите кнопку **Отобразить таблицу**  в группе **Настройка запроса (Query Setup)** на вкладке

ке Работа с запросами | Конструктор (Query Tools | Design) или выполните команду Добавить таблицу (Show Table) в контекстном меню, вызываемом на схеме данных запроса.

4. В окне конструктора (рис. 4.4) последовательно перетащите из списка полей таблицы ТОВАР поля НАИМ_ТОВ, ЦЕНА, НАЛИЧИЕ_ТОВ в столбцы бланка запроса в строку Поле (Field).

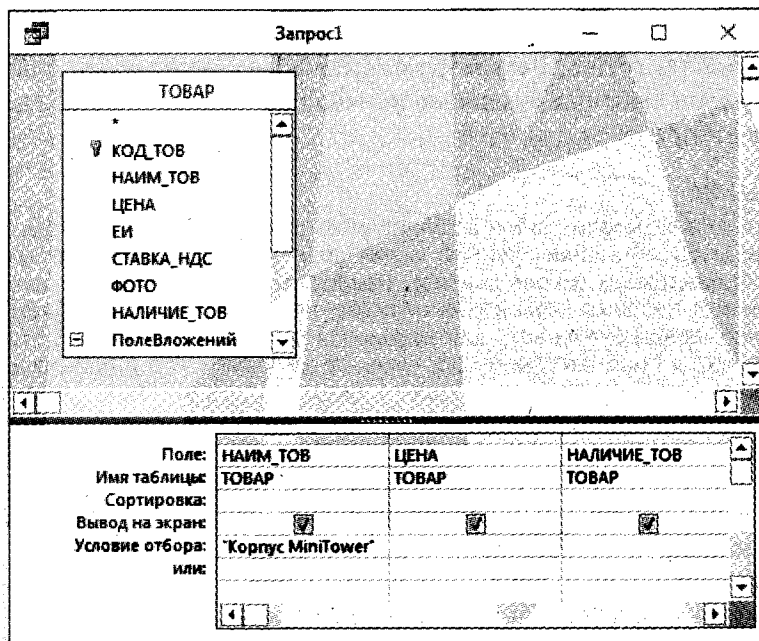



Рис. 4.4. Окно конструктора запроса на выборку

5. Для включения нужных полей из таблицы в соответствующие столбцы запроса можно воспользоваться следующими приемами:
 - в первой строке бланка запроса Поле (Field) щелчком мыши вызвать появление кнопки списка и выбрать из списка нужное поле. Список содержит поля таблиц, представленных в схеме данных запроса;
 - щелкнуть двойным щелчком на имени поля таблицы в схеме данных запроса;
 - для включения всех полей таблицы можно перетащить или щелкнуть двойным щелчком на символе * (звездочка) в списке полей таблицы в схеме данных запроса.
6. Если вы по ошибке перетащили в бланк запроса ненужное поле, удалите его. Для этого переместите курсор в область маркировки столбца сверху, где он примет вид черной стрелки, направленной вниз, и щелкните кнопкой мыши. Столбец выделится. Нажмите клавишу <Delete> или выполните команду Удалить столбцы (Delete Columns) в группе Настройка запроса (Query Setup).

7. В строке **Вывод на экран (Show)** отметьте требуемые поля, иначе они не будут включены в таблицу запроса.
8. Запишите в строке **Условие отбора (Criteria)** наименование товара, как показано в бланке запроса на рис. 4.4. Так как выражение в условии отбора не содержит оператора, то по умолчанию используется оператор =. Используемое в выражении текстовое значение вводится в двойных кавычках, которые добавляются автоматически.
9. Выполните запрос, щелкнув на кнопке **! Выполнить (Run)** или на кнопке  **Режим (View)** в группе **Результаты (Results)** — на экране откроется окно запроса в режиме таблицы с записью из таблицы **ТОВАР**, отвечающей заданным условиям отбора.

ЗАМЕЧАНИЕ

Окно запроса в режиме таблицы аналогично окну просмотра таблицы базы данных. Через некоторые таблицы запроса может производиться изменение данных базовой таблицы, лежащей в основе запроса. Запрос, просматриваемый в режиме таблицы, в отличие от таблицы базы данных Access, не имеет столбца **Добавить поле (Add New Field)**, предназначенного для изменения структуры таблицы. В этом режиме на вкладке ленты Главная (Home) доступны те же кнопки, что и при открытии таблицы базы данных.

10. Если при вводе сложного наименования товара вы допустили неточность, товар не будет найден в таблице. Использование *операторов шаблона* — звездочка * и вопросительный знак ? (стандарт ANSI-89, используемый для запросов по умолчанию) или знак процента % и подчеркивания _ (ANSI-92, рекомендуемый как стандарт для SQL Server) — упрощает поиск нужных строк и позволяет избежать многих ошибок. Введите вместо полного имени товара **Корпус*** или **Корпус%**. Выполните запрос. Если в поле наименования товара одно значение начинается со слова «Корпус», результат выполнения запроса будет таким же, как в предыдущем случае. После выполнения запроса введенное выражение будет дополнено оператором **ALike** "Корпус%". Этот оператор позволяет использовать символы шаблона при поиске в текстовых полях.
11. Если необходимо найти несколько товаров, используйте оператор **In**. Он позволяет выполнить проверку на равенство любому значению из списка, который задается в круглых скобках. Запишите в строке условий отбора **In** ("Корпус MiniTower"; "HDD Maxtor 20GB"; "FDD 3,5"). В таблице запроса будет выведено три строки. В операторе **In** не допускается использование символов шаблона.
12. Сохраните запрос, щелкнув на значке **Файл** и выполнив команду **Сохранить (Save)**. В окне **Сохранение (Save As)** введите имя запроса **Пример1**. Заметим, что имя запроса не должно совпадать не только с именами имеющихся запросов, но и с именами таблиц в базе данных. Сохранить запрос можно также командой **Сохранить (Save)** на пользовательской панели инструментов быстрого доступа (**Customize Quick Access Toolbar**).
13. Закройте текущий запрос по команде контекстного меню **Заккрыть (Close)** или нажав кнопку окна запроса **Заккрыть (Close)**.

14. Выполните сохраненный запрос, выделив запрос в области навигации и выбрав в контекстном меню команду **Открыть** (Open).
15. Для редактирования запроса выделите его в области навигации и выполните в контекстном меню команду **Конструктор** (Design View).

Задача 2

Пусть надо выбрать товары, цена которых не более 1000 руб., и НДС не более 10 %, а также выбрать товары, цена которых более 2500 руб. Результат должен содержать наименование товара (НАИМ_ТОВАР), его цену (ЦЕНА) и НДС (СТАВКА_НДС).

1. Создайте новый запрос в режиме конструктора, добавьте таблицу ТОВАР. В окне конструктора (рис. 4.5) последовательно перетащите из списка полей таблицы ТОВАР в бланк запроса поля НАИМ_ТОВАР, ЦЕНА, СТАВКА_НДС.
2. Запишите **Условие отбора** (Criteria), как показано в бланке запроса на рис. 4.5. Между условиями, записанными в одной строке, выполняется логическая операция AND. Между условиями, записанными в разных строках, выполняется логическая операция OR.

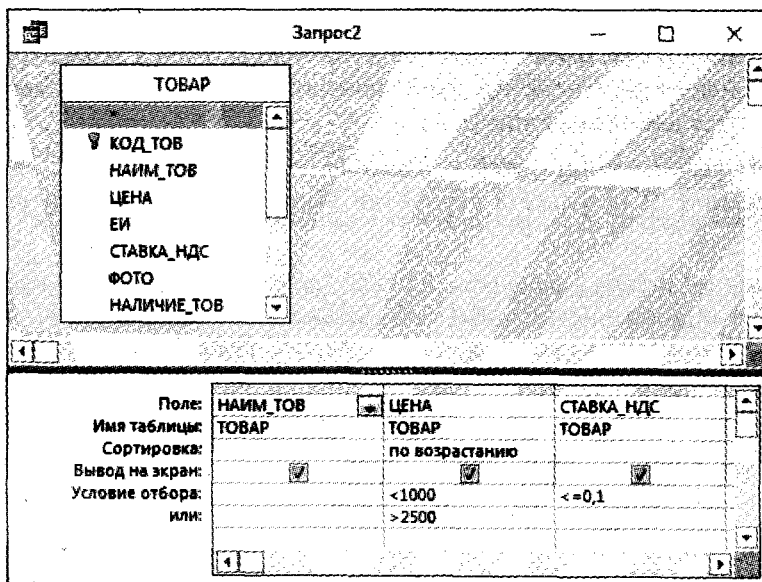


Рис. 4.5. Окно конструктора запроса на выборку с логическими операциями в условии отбора

3. Выполните запрос, щелкнув на кнопке **Выполнить** (Run) в группе **Результаты** (Results). На экране появится окно запроса в режиме таблицы с записями из таблицы ТОВАР, отвечающими заданным условиям отбора.
4. Сохраните запрос, выполнив соответствующую команду в контекстном меню запроса, которое вызывается при установке курсора на заголовок запроса. Дайте ему имя **Пример2**.

Задача 3

Пусть надо выбрать все накладные за заданный период. Результат должен содержать номер накладной (НОМ_НАК), код склада (КОД_СК), дату отгрузки (ДАТА_ОТГР) и общую стоимость отгруженного товара (СУММА_НАКЛ).

1. Создайте новый запрос в режиме конструктора, добавьте таблицу НАКЛАДНАЯ. В окне конструктора последовательно перетащите из списка полей таблицы НАКЛАДНАЯ в бланк запроса все необходимые поля.
2. Для поля ДАТА_ОТГР в строке **Условие отбора (Criteria)** запишите `Between #11.01.2016# And #31.03.2016#`. Оператор `Between` задает интервал дат (в ANSI-92 вместо знака `#` используются одинарные кавычки `'`). Кроме того, этот оператор позволяет задать интервал для числового значения.

Вычисляемые поля в запросах

В запросе, как и в таблице, для каждой записи могут производиться вычисления с числовыми, строковыми значениями или значениями дат с использованием данных из одного или нескольких полей. Результат вычисления образует в таблице запроса новое вычисляемое поле. Для создания вычисляемого поля нужно ввести выражение в ячейку строки **Поле (Field)** пустого столбца бланка запроса. При каждом выполнении запроса на основе текущих значений полей, используемых в выражении, вычисляются новые значения поля.

В выражениях вычисляемых полей помимо имен полей могут использоваться константы и функции. В результате обработки выражения может получаться только одно значение.

Задача 1

В таблице ТОВАР имеются поля ЦЕНА и СТАВКА_НДС. Вычислите цену с учетом НДС и сравните ее с полученной в вычисляемом поле таблицы Цена с НДС.

1. Создайте в режиме конструктора запрос на выборку для таблицы ТОВАР. Перетащите в бланк запроса поля НАИМ_ТОВ, ЦЕНА, СТАВКА_НДС и Цена с НДС (рис. 4.6).
2. Для подсчета цены с учетом НДС создайте вычисляемое поле, записав в пустой ячейке строки **Поле (Field)** выражение `[ЦЕНА]+[ЦЕНА]*[СТАВКА_НДС]`.
3. Для отбора записей со значением выше 5000 в вычисляемом поле в строку **Условие отбора (Criteria)** введите `> 5000`.
4. После ввода выражения система по умолчанию формирует имя вычисляемого поля `Выражение1`, которое становится заголовком столбца в таблице с результатами выполнения запроса. Это имя вставится перед выражением `[ЦЕНА]+[ЦЕНА]*[СТАВКА_НДС]`. Для каждого нового вычисляемого поля в запросе номер выражения увеличивается на единицу. Имя вычисляемого поля отделяется от выражения двоеточием. Для изменения имени установите курсор мыши в вычисляемом поле бланка запроса и нажмите правую кнопку мыши. В контекстно-зависимом меню выберите **Свойства (Properties)** поля и в строку

Подпись (Caption) введите новое имя поля: Цена с НДС. Теперь в таблице с результатами выполнения запроса в заголовке вычисляемого столбца отобразится это имя. Имя поля может быть исправлено также непосредственно в бланке запроса.

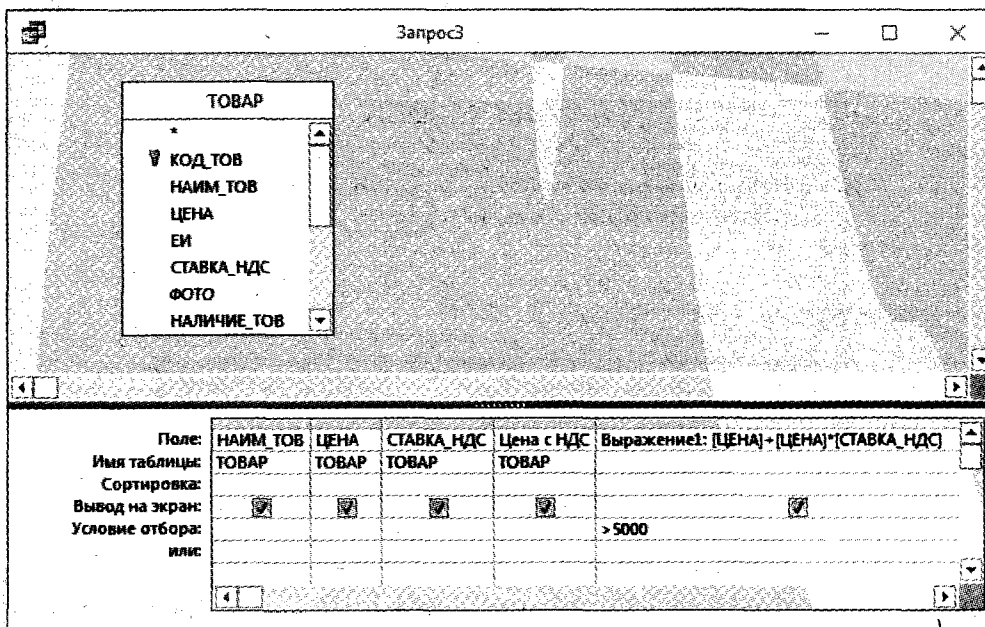


Рис. 4.6. Запрос с вычисляемым полем и условием отбора по его значению

- Для отображения результата выполнения запроса щелкните на кнопке **Выполнить (Run)** в группе **Результаты (Results)**. Вычисляемое поле таблицы и запроса имеют одинаковые значения.
- Измените в одной из записей запроса цену товара. Значения в обоих вычисляемых полях будут моментально пересчитаны.
- Для формирования сложного выражения в вычисляемом поле или условии отбора целесообразно использовать построитель выражений. Построитель позволяет выбрать необходимые в выражении имена полей из таблиц, запросов, операторы, функции. Удалите выражение в вычисляемом поле и используйте построитель для его формирования.
- Вызовите построитель выражений (Expression Builder), нажав кнопку **Построитель (Builder)** в группе **Настройка запроса (Query Setup)** ленты **Конструктор** или выбрав **Построить (Build)** в контекстно-зависимом меню. Курсор мыши должен быть установлен предварительно в ячейке ввода выражения.
- В левой части окна **Построитель выражений (Expression Builder)** (рис. 4.7) выберите таблицу **ТОВАР**, на которой построен запрос. Справа отобразится список ее полей. Последовательно выбирайте нужные поля и операторы, двойным щелчком вставляя их в выражение. Выражение сформируется в верхней части

окна. Обратите внимание — построитель перед именем поля указал имя таблицы, которой оно принадлежит, и отделил его от имени поля восклицательным знаком.

ЗАМЕЧАНИЕ

Если вычисляемое поле строится в ранее сохраненном запросе, в построителе выражений автоматически выбирается этот запрос и доступны для использования его поля.

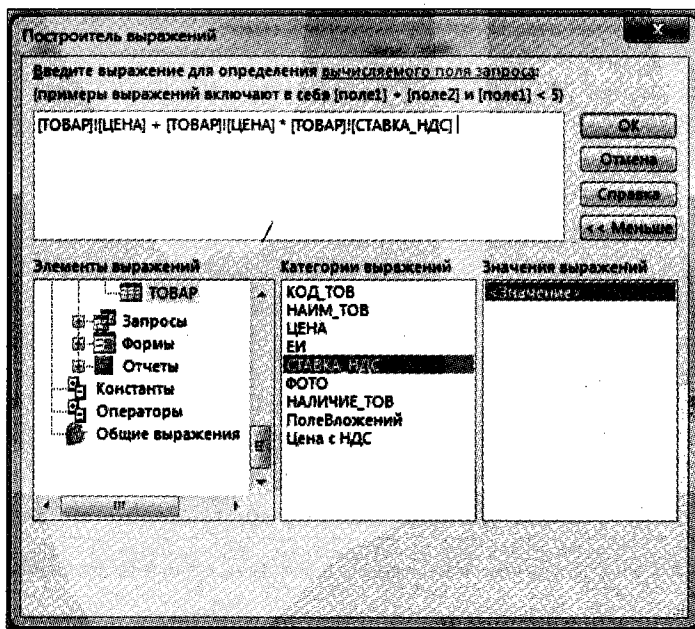


Рис. 4.7. Окно построителя выражений при формировании вычисляемого поля запроса

10. Завершите процесс построения выражения в вычисляемом поле, щелкнув на кнопке **ОК**.
11. Сохраните запрос под именем **Цена с НДС** и закройте его.
12. Выполните сохраненный запрос, выделив его в области навигации и выбрав в контекстном меню команду **Открыть (Open)**.

В вычисляемых полях и условиях отбора можно использовать *встроенные функции*. В Access определено более 150 таких функций.

Задача 2

Пусть необходимо выбрать все накладные, по которым производилась отгрузка в заданном месяце. В таблице **НАКЛАДНАЯ** дата отгрузки хранится в поле **ДАТА_ОТГР** с типом данных **Дата и время (Date/Time)**.

1. Создайте в режиме конструктора запрос на выборку данных из таблицы **НАКЛАДНАЯ**. Перетащите в бланк запроса поля **НОМ_НАКЛ** и **КОД_СК** (рис. 4.8).

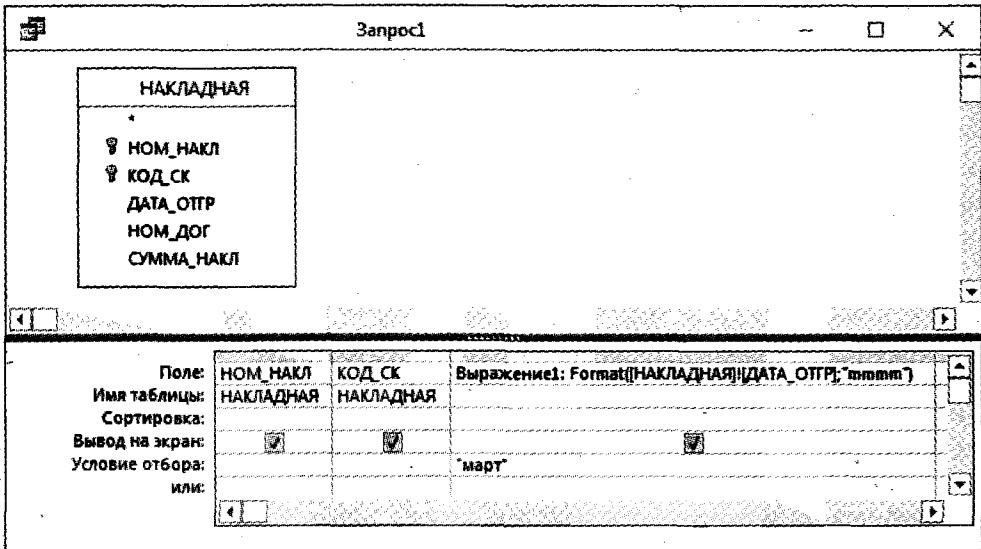


Рис. 4.8. Запрос с функцией выделения из даты полного названия месяца в вычисляемом поле

2. Создайте вычисляемое поле, записав в пустой ячейке строки **Поле (Field)** одно из выражений:

- `Format([НАКЛАДНАЯ].[ДАТА_ОТПР];"mmmm")` — эта функция возвратит полное название месяца;
- ИЛИ `Format([НАКЛАДНАЯ].[ДАТА_ОТПР];"mm")` — эта функция возвратит номер месяца.

ЗАМЕЧАНИЕ

Если необходимо выделить не один элемент из даты, запишите в функции **Format** несколько символов форматирования, разделяя их знаком \. Например, функция `Format([НАКЛАДНАЯ].[ДАТА_ОТПР];"mmmm\ yyyy")` возвратит полное название месяца и четыре цифры года. После знака \ нужно поставить пробел.

3. Для отбора накладных, выписанных в заданном месяце, в вычисляемом поле в строку **Условие отбора (Criteria)** введите название месяца, например `март` (рис. 4.8), или номер месяца, например `3`, — в соответствии с тем, как в выражении определен параметр функции `Format`.
4. Выполните запрос, нажав кнопку **Выполнить (Run)** в группе **Результаты (Results)** на вкладке ленты **Работа с запросами | Конструктор (Query Tools | Design)**. В таблице запроса отобразятся накладные, выписанные в заданном месяце.
5. Дополните запрос полем `ДАТА_ОТПР`, чтобы убедиться, что функция извлекает из даты месяц в заданном формате.
6. Запишите в вычисляемом поле функцию `Month([НАКЛАДНАЯ].[ДАТА_ОТПР])` и убедитесь, что эта функция возвращает выделенный из даты номер месяца.

7. Для выборки всех строк, относящихся ко второму кварталу, в строку **Условие отбора (Criteria)** введите оператор `Between 4 And 6`, определяющий, попадает ли значение выражения в указанный интервал.
8. Запишите в вычисляемом поле выражение `MonthName (Month (НАКЛАДНАЯ!ДАТА_ОТПР))` и убедитесь, что функция `MonthName` преобразует номер месяца в его полное название.

Параметры в запросах

В предыдущих примерах выражение в условии отбора вводилось в бланке запроса. При этом, чтобы задать новое значение в условии отбора, нужно было повторно открыть запрос в режиме конструктора и ввести его. При решении практических задач значительно удобнее вводить выражение в условии отбора в процессе выполнения запроса в диалоге с системой, не переходя в режим конструктора. Обеспечить такой диалог можно с помощью *параметра запроса*. Имя параметра запроса задается в строке **Условие отбора (Criteria)** в квадратных скобках. При выполнении запроса это имя появится в диалоговом окне **Введите значение параметра (Enter Parameter Value)**.

1. Замените в условии отбора рассмотренного ранее запроса (см. рис. 4.8) название месяца март на имя параметра: `[Название месяца]`.
2. Выполните запрос. Открывшееся диалоговое окно (рис. 4.9) позволит ввести значение параметра запроса: **Название месяца**.

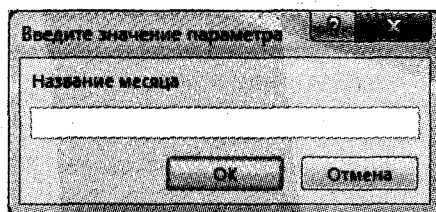


Рис. 4.9. Диалоговое окно ввода значения параметра

3. В запросе может быть определено несколько параметров. Например, для отбора записей по двум месяцам в условии отбора вычисляемого поля запишите два параметра, связанные логической операцией `OR`: `[Название месяца] OR [Еще одно название месяца]`.
 4. Для определения числового интервала используйте параметры в операторе `Between [Номер первого месяца периода] and [Номер последнего месяца периода]`.
- Параметры запроса могут быть использованы не только отдельно, но и как операнды выражений в условиях отбора или вычисляемых полях.

1. Создайте в режиме конструктора запрос на выборку для таблицы **ТОВАР**. Перетащите в бланк запроса поля `НАИМ_ТОВ` и `ЦЕНА`.

2. Для увеличения цены на заданный процент в вычисляемое поле запишите выражение с параметром запроса [На сколько процентов увеличить?] (рис. 4.10):
 $ЦЕНА+ЦЕНА*[На\ сколько\ процентов\ увеличить?]/100$.

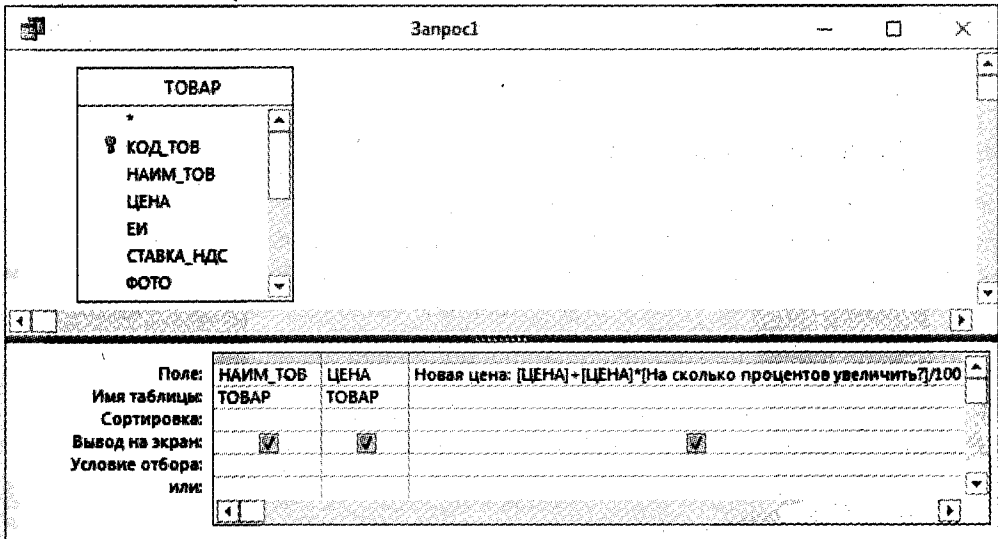


Рис. 4.10. Использование параметра в выражении вычисляемого поля

ОБРАТИТЕ ВНИМАНИЕ

Через этот запрос вы сможете показать увеличенные цены в таблице запроса или использовать их при построении форм, отчетов, но они не будут внесены в поле таблицы ТОВАР.

После выполнения запроса в таблице отображается результат вычисления с использованием введенного значения параметра. Однако значение параметра при этом не выводится. Для отображения в таблице запроса введенного значения параметра дополните запрос еще одним вычисляемым полем, в котором запишите, например, такое выражение:

Увеличение: "На " & [На сколько процентов увеличить?] & "%"

Теперь в таблице запроса появится поле Увеличение, в котором будет записано, например, при вводе 10, — На 10%. В дальнейшем такое поле можно будет использовать при создании форм или отчетов.

Чтобы избежать ошибок при вводе в качестве значения параметра данных неправильного типа, следует указывать тип данных для параметра запроса. Это позволит получать подробные сообщения об ошибках в случае ввода данных неправильного типа, например, ввода текста, когда требуются числовые данные.

Определите тип данных для параметра [На сколько процентов увеличить?] в последнем запросе.

1. Откройте запрос в режиме конструктора и на соответствующей вкладке в группе **Показать или скрыть (Show/Hide)** нажмите кнопку **Параметры (Parameters)**.

2. В окне **Параметры запроса** (Query Parameters) (рис. 4.11) в столбце **Параметр** (Parameter) введите имя параметра запроса, для которого требуется указать тип данных. Имя должно соответствовать используемому в бланке запроса.
3. В столбце **Тип данных** (Data Type) выберите тип данных.

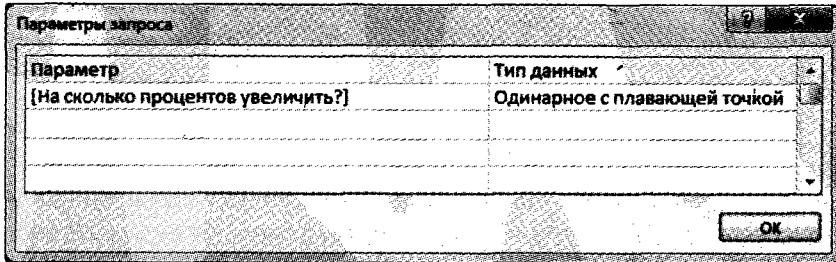


Рис. 4.11. Определение типа данных параметра запроса

ЗАМЕЧАНИЕ

Если для параметра указан текстовый тип данных, любое введенное значение интерпретируется как текст, и сообщение об ошибке не отображается.

Групповые операции в запросах

Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и вычислить итоговые данные для каждой из групп по другим полям, используя одну из статистических функций. Статистические функции применимы, прежде всего, к полям с типом данных **Числовой** (Number), **Денежный** (Currency), **Дата и время** (Date/Time). В Access предусматривается девять статистических функций.

- Sum — сумма значений некоторого поля для группы;
- Avg — среднее от всех значений поля в группе;
- Max, Min — максимальное, минимальное значение поля в группе;
- Count — число значений поля в группе без учета пустых значений;
- StDev — среднееквадратичное отклонение от среднего значения поля в группе;
- Var — дисперсия значений поля в группе;
- First и Last — значение поля из первой или последней записи в группе.

Результат запроса с использованием групповых операций содержит по одной записи для каждой группы. В запрос, прежде всего, включаются поля, по которым производится группировка, и поля, для которых выполняются статистические функции. Кроме этих полей в запрос могут включаться поля, по которым задаются условия обора.

Рассмотрим конструирование однотабличного запроса с групповой операцией на примере таблицы ПОСТАВКА_ПЛАН.

Запрос с функцией Sum

Задача

Определите, какое суммарное количество каждого из товаров должно быть поставлено покупателям по всем договорам. Все данные о запланированном к поставке количестве товара указаны в таблице ПОСТАВКА_ПЛАН.

1. Создайте в режиме конструктора запрос на выборку данных из таблицы ПОСТАВКА_ПЛАН.
2. Из списка таблицы перетащите в бланк запроса поле КОД_ТОВ (код товара) — по этому полю будет производиться группировка записей таблицы.
3. Перетащите в бланк запроса поле КОЛ_ПОСТ, по которому будет подсчитываться суммарное количество каждого из товаров, заказанных покупателями во всех договорах.
4. Выполните команду **Итоги (Totals)** из группы **Показать или скрыть (Show/Hide)**. В бланке запроса появится новая строка **Групповая операция (Total)** со значением **Группировка (Group By)** в обоих полях запроса.
5. В столбце КОЛ_ПОСТ замените слово **Группировка (Group By)** на функцию **Sum**. Для этого нажмите кнопку списка и выберите эту функцию. Бланк запроса примет вид, показанный на рис. 4.12.

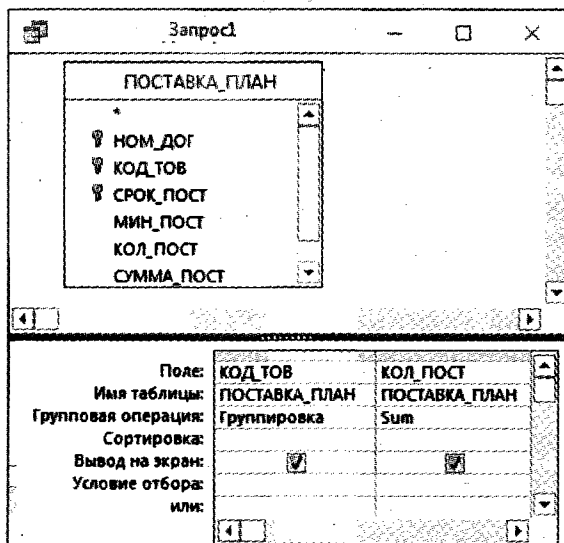


Рис. 4.12. Запрос с группировкой по коду товара и суммированием количеств в группе

Код товара	Sum-КОЛ_ПОСТ
	2387
T002	780
T003	750
T004	30
T005	750
T006	200
T007	10
T008	4
T010	400

Рис. 4.13. Результат подсчета суммарного количества каждого из товаров

6. Для отображения результата запроса (рис. 4.13) щелкните на кнопке **Выполнить (Run)** в группе **Результаты (Results)**.
7. Замените подпись поля **Sum-КОЛ_ПОСТ** на **Заказано товаров**. Для этого перейдите в режим конструктора, в бланке запроса установите курсор мыши на

поле КОД_ПОСТ и нажмите правую кнопку. В контекстном меню выберите Свойства (Properties). В окне Свойства поля (Field Properties) введите в строке Подпись (Caption): Заказано товаров. Для открытия окна свойств может быть выполнена команда Страница свойств (Property Sheet) в группе Показать или скрыть (Show/Hide).

8. Сохраните запрос под именем Заказано товаров.
9. Чтобы подсчитать количество товаров, заказанных в каждом месяце, выполните группировку по двум полям: КОД_ТОВ и СРОК_ПОСТ, в котором хранится месяц поставки (рис. 4.14).

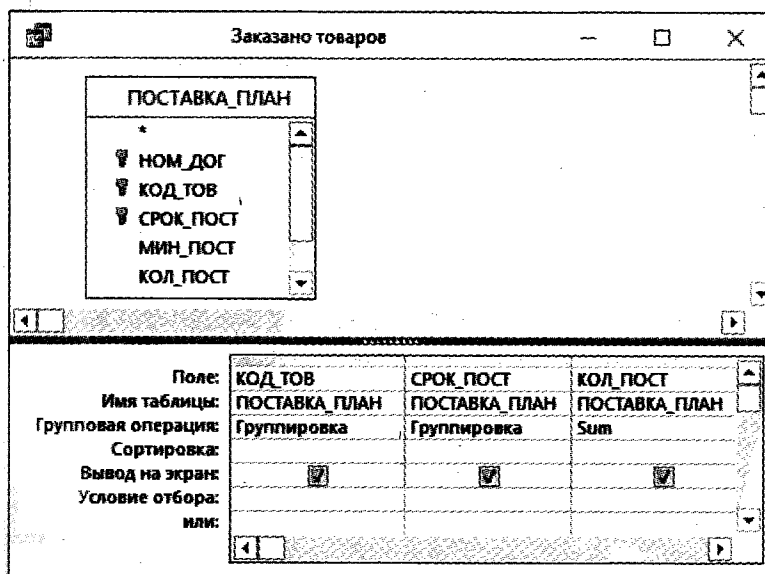


Рис. 4.14. Запрос с группировкой по двум полям

10. Чтобы подсчитать количество товаров, заказанных в заданном месяце, предыдущий запрос дополните вводом параметра запроса в условие отбора (рис. 4.15).

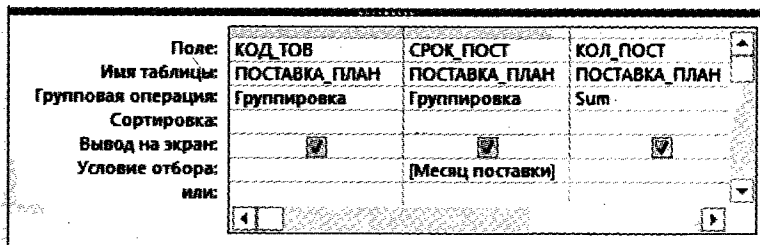


Рис. 4.15. Запрос с группировкой по двум полям и параметром запроса

Запрос с функцией Count

Задача

Определите, сколько раз отгружался товар по каждому из договоров. Факт отгрузки фиксируется записью в таблице НАКЛАДНАЯ.

1. Создайте запрос на выборку на основе таблицы НАКЛАДНАЯ.
2. Из списка полей таблицы НАКЛАДНАЯ перетащите в бланк запроса поле `НОМ_ДОГ`. По этому полю должна производиться группировка.
3. По сути, смысл задачи сводится к подсчету в таблице числа строк с одинаковыми номерами договоров, поэтому неважно, по какому полю будет вычисляться функция `Count`. Перетащите в бланк запроса любое поле, например опять `НОМ_ДОГ`.
4. Выполните команду **Итоги (Totals)** из группы **Показать или скрыть (Show/Hide)**. Замените слово **Группировка (Group By)** в одном из столбцов с именем `НОМ_ДОГ` на функцию `Count`. Бланк запроса примет вид, показанный на рис. 4.16.
5. Сохраните запрос под именем **Число отгрузок по договорам**. Выполните запрос. Результат запроса показан на рис. 4.17.

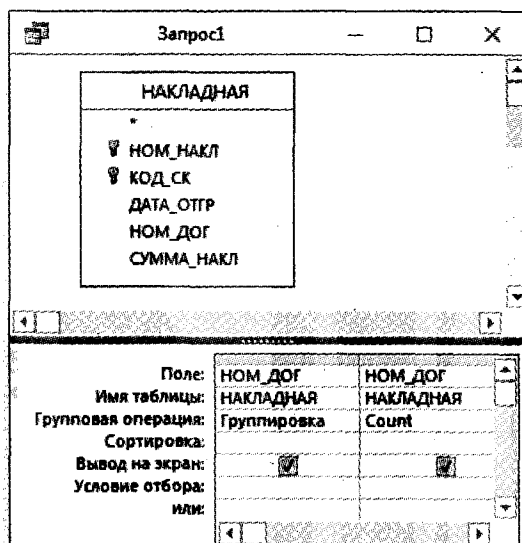


Рис. 4.16. Запрос для подсчета числа отгрузок по договорам

Номер догово	Count-НОМ_ДОГ
Д222	7
Д333	1

Рис. 4.17. Результат подсчета числа отгрузок по договорам

Условия отбора в запросах с групповыми операциями

В запрос с групповыми операциями помимо полей, по которым производится группировка и выполнение статистических функций для групп, можно из таблиц включать поля для задания условий отбора записей, включаемых в группы.

Задача

Подсчитайте, сколько накладных было выписано по каждому из договоров и какова общая стоимость товаров, отгруженных по этим накладным. В расчете учитывайте только накладные на сумму более 10 000 руб.

1. Для подсчета общей стоимости товаров дополните бланк запроса **Число отгрузок по договорам** (см. рис. 4.16, 4.17) полем СУММА_НАКЛ и замените в нем слово **Группировка** (Group By) на функцию Sum.
2. Затем вторично включите поле СУММА_НАКЛ в бланк запроса и замените слово **Группировка** (Group By) словом **Условие** (Where), выбрав его из списка. Обратите внимание, в этом столбце не должен быть отмечен флажок **Вывод на экран** (Show). В строку **Условие отбора** (Criteria) введите выражение >10000 (рис. 4.18).

Поле:	НОМ_ДОГ	НОМ_ДОГ	СУММА_НАКЛ	СУММА_НАКЛ
Имя таблицы:	НАКЛАДНАЯ	НАКЛАДНАЯ	НАКЛАДНАЯ	НАКЛАДНАЯ
Групповая операция:	Группировка	Count	Sum	Условие
Сортировка:				
Вывод на экран:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				>10000
или:				

Рис. 4.18. Запрос с группировкой записей, соответствующих условию отбора

3. Выполните запрос и убедитесь, что число накладных для некоторых договоров уменьшилось, и общая стоимость отгруженного по договору также считается только с учетом накладных на сумму более 10 000 (рис. 4.19).

Номер договора	Count-НОМ_ДОГ	Sum-СУММА_НАКЛ
Д222	7	1 233 906,00р.
Д333	1	216 427,00р.

Записи: 1 из 3 | Нет фильтра | Поиск

Рис. 4.19. Результат выполнения запроса

4. Чтобы отобразить только нужные группы записей, введите условие отбора в поле, по которому производится группировка, или в поле, где записана функция. Например, чтобы отобразить договоры с заданными номерами, введите в условие

отбора поля `НОМ_ДОГ` строку `In ("Д111"; "Д333")`. Чтобы отобразить договоры с заданной общей стоимостью отгрузки по ним, введите в поле `СУММА_НАКЛ` условие `>10 000`.

Отображение строки итогов по столбцу

Строка итогов используется для быстрого расчета и отображения в столбце таблицы или запроса в режиме таблицы таких значений, как итоговая сумма, среднее, минимальное и максимальное, количество значений.

1. Для добавления строки итогов в таблицу запроса откройте запрос **Число отгрузок по договорам** в режиме таблицы. На вкладке ленты **Главная (Home)** в группе **Записи** выполните команду **Итоги (Totals)**. В таблице отобразится строка **Итог (Total)**.
2. В строке **Итог (Total)** нажмите кнопку списка в столбце, для которого требуется выполнить расчет, и выберите в списке, например, **Сумма (Sum)**. Результат вычисления в этом столбце итоговой суммы показан на рис. 4.20.

Номер догово	Счит-НОМ_ДОГ	Sum-СУММА_НАКЛ
Д111	3	659 338,00р.
Д222	7	1 233 906,00р.
Д333	1	216 427,00р.
Итог		2 109 671,00р.

Записи: И | Итоги | Нет Ф

Нет

- Сумма
- Среднее
- Количество значений
- Максимальное значение
- Минимальное значение
- Стандартное отклонение
- Дисперсия

Рис. 4.20. Отображение строки итогов с расчетом суммы по столбцу

ЗАМЕЧАНИЕ

Кнопка списка расположена с другой стороны ячейки для того, чтобы избежать нарушений выравнивания текста и чисел.

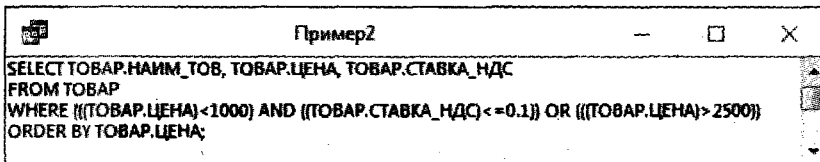
3. Для того чтобы скрыть строку итогов, повторно выполните команду **Итоги (Totals)**.

Просмотр инструкции SQL в запросе

1. Откройте в режиме конструктора рассмотренный ранее запрос (см. рис. 4.5), сохраненный под именем **Пример2**.
2. Для перехода в **Режим SQL (SQL View)** выберите из списка кнопки **Режим (View)** соответствующее значение. Можно также воспользоваться кнопкой

Режим SQL в строке состояния. Access выведет для запроса, созданного в режиме конструктора, эквивалентную инструкцию SQL (рис. 4.21).

- Инструкция `SELECT` используется для формирования таблицы запроса, структура которой определяется выбранными из таблиц базы данных полями, а набор записей сформулированными условиями отбора.
- Список имен нужных полей следует непосредственно за ключевым словом `SELECT`.
- Предложение `FROM` определяет таблицы или запросы, которые служат источником данных для запроса, и способ объединения их записей, если запрос построен на нескольких таблицах.
- В предложении `WHERE` задаются условия отбора записей, которые полностью соответствуют сформулированным в задании на создание запроса (см. рис. 4.5).
- Предложение `ORDER BY` указывает, по каким полям должна быть выполнена сортировка записей в таблице запроса. При отсутствии сортировки этого предложения нет в инструкции. Такая инструкция указывает на сортировку по возрастанию по полю `ЦЕНА` из таблицы `ТОВАР`.



```
SELECT ТОВАР.НАИМ_ТОВ, ТОВАР.ЦЕНА, ТОВАР.СТАВКА_НДС
FROM ТОВАР
WHERE ((ТОВАР.ЦЕНА)<1000) AND ((ТОВАР.СТАВКА_НДС)<=0.1)) OR (((ТОВАР.ЦЕНА)>2500))
ORDER BY ТОВАР.ЦЕНА;
```

Рис. 4.21. Запрос с логическими операциями в условии отбора в режиме SQL

3. Перейдите в режим конструктора и внесите изменения в бланк запроса — например, добавьте поле код товара: `КОД_ТОВ`.
4. Вернитесь в режим SQL и посмотрите, как изменилась инструкция SQL.
5. Чтобы убедиться, что запросы можно редактировать в режиме SQL, а не только в режиме конструктора, внесите изменения в инструкцию SQL, удалив из списка полей `КОД_ТОВ`.
6. Затем перейдите в режим конструктора и убедитесь, что из бланка запроса удалено поле `КОД_ТОВ`.
7. Откройте в режиме конструктора запрос **Число отгрузок по договорам**, в котором используется только функция подсчета числа договоров.
8. Перейдите в режим SQL. Инструкция SQL в этом запросе будет записана, как показано на рис. 4.22.

В инструкции `SELECT` определен список полей для таблицы запроса. Это поле `НОМ_ДОГ`, по которому производится группировка, и вычисляемое поле, содержащее результат выполнения функции `Count`, примененной к полю `НОМ_ДОГ`. Резервированное слово `AS` позволяет определить для поля с результатом вычисления псевдоним `Count-НОМ_ДОГ`, который станет заголовком столбца.

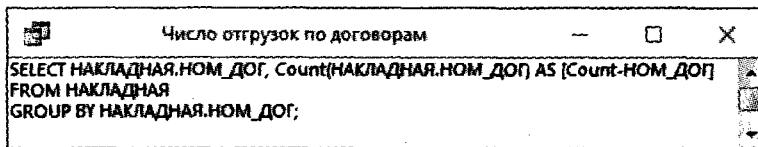


Рис. 4.22. Запрос с использованием функции Count в режиме SQL

9. Измените предложенный по умолчанию псевдоним на `Число отгрузок` и убедитесь, что в таблице запроса в соответствующем столбце отображается это имя.

Обратите внимание — в инструкции наряду с именами полей, включаемых в результат, допускается использование функций. Возможно включение и более сложных выражений, определяющих содержимое вычисляемых полей.

Предложение `GROUP BY` указывает, что необходимо объединить записи с одинаковыми значениями в указанных полях (`НАКЛАДНАЯ.НОМ_ДОГ`) в одну группу и представить ее одной записью.

Контрольные вопросы

к разд. «Однотабличные запросы на выборку»

1. Какая команда позволяет добавить необходимые таблицы в запрос?
2. Как включить в запрос нужные поля таблицы?
3. Можно ли разместить поле в бланке запроса, щелкнув двойным щелчком на нем в списке полей таблицы?
4. Можно ли заменить поле в бланке запроса выбором его из раскрывающегося списка?
5. Какие записи составляют таблицу запроса на выборку?
6. Какая логическая операция применяется к условиям отбора, записанным для различных полей в одной строке?
7. Какая логическая операция применяется к условиям отбора, записанным для одного поля в различных строках?
8. Какие операторы сравнения и логические операторы могут быть использованы при записи условия отбора?
9. Как выполнить запрос, не открывая его в режиме конструктора?
10. Можно ли использовать в выражении условия отбора имя поля?
11. Поля каких таблиц используются в выражении вычисляемого поля?
12. Где хранятся значения вычисляемого поля?
13. Где задается имя вычисляемого поля?
14. Как задать параметр запроса?
15. Из каких записей образуется группа при использовании групповой операции?

16. Какие поля выбираются для группировки записей?
17. Можно ли задать условия отбора записей, включаемых в группы?
18. Какое значение должно быть выбрано в строке **Групповые операции (Total)** для поля, по которому задается условие отбора?
19. В каком столбце бланка запроса записывается условие отбора нужных групп?
20. Как просмотреть инструкцию SQL, построенную конструктором или мастером запросов?
21. Какая инструкция SQL соответствует запросу на выборку?
22. Изменяет ли инструкция `SELECT` данные в базе?
23. В каком предложении инструкции SQL указываются таблицы, на которых создается запрос?
24. В каком предложении инструкции SQL задаются условия отбора записей?
25. Можно ли в списке полей инструкции `SELECT` использовать функции?
26. В каком предложении инструкции SQL задаются поля, по которым должна быть выполнена группировка записей?
27. Каким знаком разделяются имена полей в списке инструкции `SELECT`?
28. Можно ли в инструкции `SELECT` использовать символ звездочки (*) для отбора всех полей таблицы?
29. Какой оператор позволяет задать интервал выбираемых значений в условии отбора?
30. Приведите пример использования оператора `Between` для выбора интервала дат.
31. Имеет ли смысл производить группировку по ключу?
32. Сколько записей сформируется в таблице запроса при группировке по полю `код_тов` в таблице `ПОСТАВКА_ПЛАН`, содержащей все сведения о заказанных товарах?
33. Какая команда позволяет добавить в таблицу запроса строку итогов?

Ответы

1. **Отобразить таблицу...** (Show Table...) на вкладке конструктора запросов или в контекстном меню схемы данных запроса **Добавить таблицу (Show Table...)**.
2. Разместив их в столбцах бланка запроса.
3. Да.
4. Да.
5. Соответствующие заданным условиям отбора.
6. `AND` (И).
7. `OR` (ИЛИ).
8. `=`, `<`, `>`, `<>`, `<=`, `>=`, `Between`, `In`, `Like`, `And`, `Or`, `Not`.

9. Выделив запрос в области навигации базы данных и выполнив команду **Открыть** (Open) в его контекстном меню.
10. Да, заключив его в квадратные скобки.
11. Поля таблицы, в которой создается вычисляемое поле.
12. Только в записях таблицы запроса.
13. В его свойстве **Подпись** (Caption) или непосредственно перед выражением вычисляемого поля.
14. Записав имя параметра, заключенное в квадратные скобки, в условии отбора или выражении вычисляемого поля.
15. Из записей с одинаковыми значениями в полях группировки.
16. Поля, имеющие повторяющиеся значения.
17. Да.
18. **Условие** (Where).
19. В поле, по которому производится группировка или для которого выполняется функция.
20. Выбрав для запроса в режиме конструктора режим SQL из списка кнопки **Режим** (View) в группе **Результаты** (Results).
21. SELECT.
22. Нет.
23. FROM.
24. WHERE.
25. Да.
26. GROUP BY.
27. Запятой.
28. Да.
29. Between.
30. Between #11.01.2015# And #31.03.2016#
31. Нет.
32. Сколько видов товара было заказано.
33. Команда **Итоги** (Totals) на вкладке ленты **Главная** (Home) в группе **Записи**.

Многотабличные запросы на выборку данных

Многотабличный запрос позволяет сформировать записи результата путем объединения записей из нескольких взаимосвязанных таблиц базы данных и выбора из них нужных полей и записей. Многотабличный запрос часто осуществляет объединение данных, которые на этапе проектирования были разделены на множество таблиц, отвечающих требованиям нормализации. В нормализованных таблицах,

прежде всего, обеспечивалось отсутствие повторяемости описательных данных в базе, там повторяются только значения ключевых полей. В результате выполнения запроса формируется ненормализованная таблица с повторяющимися данными, в которой каждая запись собирает необходимые данные из разных таблиц.

Например, при объединении двух нормализованных связанных одно-многочными отношениями таблиц, для которых обеспечивается связанная целостность, результирующая запись образуется на основе записи подчиненной таблицы, в которую добавляются поля из связанной записи главной таблицы. Подобное объединение формирует ненормализованную таблицу, в которой число записей равно числу записей в подчиненной таблице. При этом данные главной таблицы дублируются в различных записях результирующей таблицы.

При конструировании многотабличного запроса важнейшим условием является правильное представление о том, как идет объединение записей таблиц при формировании результата.

В Access имеются возможности задать способ объединения записей двух связанных таблиц как при создании общей схемы базы данных, так и в схеме данных запроса. Для любой пары связанных таблиц может быть выбран один из трех способов объединения записей:

- способ 1* — объединение только тех записей, в которых значения связанных полей обеих таблиц совпадают (выбирается по умолчанию);
- способ 2* — объединение тех записей, в которых значения связанных полей обеих таблиц совпадают, а также объединение всех записей из первой таблицы, для которых нет связанных во второй, с пустой записью второй таблицы;
- способ 3* — объединение тех записей, в которых значения связанных полей обеих таблиц совпадают, а также объединение всех записей из второй таблицы, для которых нет связанных в первой, с пустой записью первой таблицы.

При установлении связи между таблицами в схеме данных базы по умолчанию выбран первый способ объединения. При обработке таблиц используется этот способ объединения, если он специально не переопределен.

ОБРАТИТЕ ВНИМАНИЕ

Если между таблицами в запросе не установлена связь, то для Access остается неизвестным, какие записи одной таблицы связаны с записями другой таблицы, и в запросе формируются все комбинации записей таблиц, т. е. объединяются все со всеми. Например, если одна таблица содержит 10 записей, а другая 4, то в таблице запроса будет 40 записей (10×4). Такое объединение называется «полным объединением», или «декартовым произведением».

Объединение записей в многотабличном запросе

Задача

Рассмотрим технологию конструирования многотабличного запроса на выборку для решения задачи расчета суммарного количества каждого из товаров, которое должно быть поставлено покупателям по договорам. В таблице запроса выведем

помимо кода товара также и его наименование. Для реализации такого запроса необходимы таблицы **ТОВАР** и **ПОСТАВКА_ПЛАН**, находящиеся в отношении 1 : М. Ранее был создан запрос **Заказано товаров** (см. рис. 4.12 и 4.13), решающий аналогичную задачу, но каждый из товаров был представлен только своим кодом из таблицы **ПОСТАВКА_ПЛАН**. Добавив таблицу **ТОВАР**, можно для каждого кода товара получить его наименование. Прежде чем группировать записи для подсчета суммарного количества каждого из товаров, посмотрим, как образуются эти записи при объединении двух таблиц.

1. Для создания запроса выполните команду **Конструктор запросов (Query Design)**, расположенную на ленте **Создание (Create)** в группе **Запросы**.
2. В окне **Добавление таблицы (Show Table)** (см. рис. 4.2) выберите таблицы **ПОСТАВКА_ПЛАН** и **ТОВАР** и добавьте их в запрос.
3. Закройте окно **Добавление таблицы (Show Table)**.

В результате в схеме данных запроса будут представлены таблицы **ТОВАР** и **ПОСТАВКА_ПЛАН**, между которыми, в соответствии со схемой данных, автоматически установлена связь 1 : М по полю **код_тов** с обеспечением целостности.

ЗАМЕЧАНИЕ

Если в схеме данных не определена связь между таблицами, но таблицы имеют поля с одинаковыми именами и одинаковым типом данных, Access может автоматически установить эту связь. Связи, которые не были установлены Access автоматически, может установить пользователь непосредственно в схеме данных запроса, перетаскивая задействованные в связи поля из списка полей одной таблицы в список полей другой.

4. Щелчком мыши выделите линию связи таблиц и нажмите правую кнопку. В контекстном меню выберите **Параметры объединения (Join Properties)**. В открывшемся окне для связываемых таблиц по умолчанию выбран первый способ объединения (рис. 4.23).
5. Перетащите в бланк запроса из таблицы **ПОСТАВКА_ПЛАН** поле **код_тов** и **кол_пост**, а из таблицы **ТОВАР** — поле **НАИМ_ТОВ**.

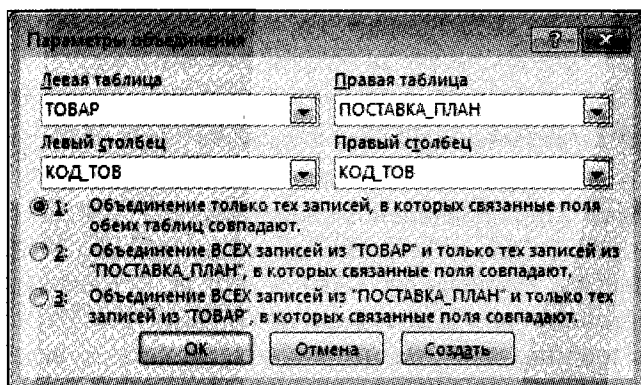



Рис. 4.23. Окно выбора способа объединения связываемых таблиц

ЗАМЕЧАНИЕ

В ряде случаев в бланке запроса наряду с именем поля нужно отображать имя соответствующей таблицы — например, когда поля имеют одинаковые имена в разных таблицах. Для того чтобы в бланке запроса, наряду с именем поля, видеть имя таблицы, выполните команду  **Имена таблиц (Table Names)** на ленте конструктора в группе **Показать или скрыть (Show/Hide)**.

6. Выполните запрос, щелкнув на кнопке **Выполнить (Run)** или **Режим (View)** на ленте конструктора запросов в группе **Результаты (Results)**. Число записей в таблице запроса будет равно числу записей в подчиненной таблице **ПОСТАВКА_ПЛАН**.

Если для связи таблиц не было задано параметра обеспечения целостности данных, в таблицу **ПОСТАВКА_ПЛАН** можно было бы включить записи с кодами товара, не содержащимися в таблице **ТОВАР**. Однако такие записи не были бы включены в таблицу запроса, т. к. первый способ объединения требует наличия связанных записей в таблице **ТОВАР**. В результате по ряду заказанных товаров необходимые сведения были бы потеряны.

7. Снова откройте для связи таблиц окно **Параметры объединения (Join Properties)** (см. рис. 4.23) и выберите второй способ объединения записей. Обратите внимание — на линии связи появилась стрелка, направленная от таблицы **ТОВАР** к таблице **ПОСТАВКА_ПЛАН** (рис. 4.24). Она указывает на необходимость включения в таблицу запроса дополнительных записей, сформированных из тех записей таблицы **ТОВАР**, для которых нет связанных в таблице **ПОСТАВКА_ПЛАН**.

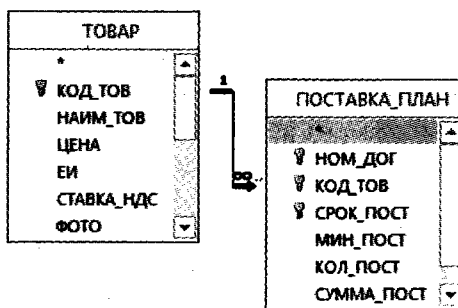


Рис. 4.24. Схема данных запроса при объединении таблиц вторым способом

8. Выполните запрос. При объединении таблиц вторым способом к записям, полученным первым способом, добавились записи из таблицы **ТОВАР**, не имеющие связанных записей в таблице **ПОСТАВКА_ПЛАН**. Каждая такая запись дополнена полями **Количество** и **Код товара** из таблицы **ПОСТАВКА_ПЛАН**, которые содержат значения Null. Такая ситуация возникает вследствие того, что в плане отсутствуют заказы на ряд товаров. На рис. 4.25 показаны три записи с пустыми значениями в полях с количеством и кодом товара.

Код товара	Наименование товара	Количество
T002	FDD 3,5	80
T002	FDD 3,5	100
T002	FDD 3,5	200
T003	HDD Maxtor 120GB	100
T003	HDD Maxtor 120GB	510
T003	HDD Maxtor 120GB	140
T004	Корпус MiniTower	30
T005	CD-ROM Panasonic IDE	710
T005	CD-ROM Panasonic IDE	40
T006	DIMM 64M PC100	200
T007	Принтер EPSON ST.A4	10
T008	СканерAcer	4
	Зв. Карта Genius Liv	
T010	Модем Genius ext	400
	Плоттер	
	Миникомпьютер	

Рис. 4.25. Результат объединения таблиц ТОВАР и ПОСТАВКА_ПЛАН вторым способом

ОБРАТИТЕ ВНИМАНИЕ

Null — это константа, которая означает, что поле не содержит данных. Поле получает значение Null, когда неизвестно его содержимое. Такое поле не следует путать с полем, содержащим пустую строку, хотя выглядят они одинаково. Значение Null можно ввести в поле или использовать в выражениях и запросах для указания отсутствующих или неизвестных данных. Пустая строка служит для указания того, что строковое значение для этого поля отсутствует. Для ввода пустой строки с клавиатуры следует ввести два символа прямых кавычек без пробела ("").

При выборе третьего способа объединения записей в таблице запроса к записям, полученным первым способом, добавились бы записи из таблицы ПОСТАВКА_ПЛАН, не имеющие связанных записей в таблице ТОВАР. Каждая такая запись дополнена полем НАИМ_ТОВ, которое содержат значения Null. Такая ситуация возникает, если в таблицу ТОВАР не внесены сведения о новых товарах, или договор заключен на несуществующий товар. В рассматриваемом примере в таблице ПОСТАВКА_ПЛАН не может существовать записей с указанием товаров, отсутствующих в таблице ТОВАР, т. к. для связи этих таблиц установлен параметр обеспечения целостности.

- Рассмотрев, как формируются записи запроса, перейдите к выполнению их группировки. Это позволит решить поставленную задачу. Выполните команду **Итоги** (Totals) из группы **Показать или скрыть** (Show/Hide). В бланке запроса в строке **Групповая операция** (Total) в столбце **КОД_ПОСТ** замените слово **Группировка** (Group By) на функцию Sum.
- Выполните запрос. На рис. 4.26 приведена таблица запроса, полученная путем объединения записей таблиц ПОСТАВКА_ПЛАН и ТОВАР вторым способом и последующей их группировки. Таблица содержит не только суммарное коли-

чество каждого из заказанных товаров — при выбранном способе объединения записей в результат попадают и те товары, по которым не было сделано заказов.

Код товара	Наименование товара	Sum-КОЛ_П
	Зв. Карта Genius Liv	
	Миникомпьютер	
	Плоттер	
T001	Монитор 17LG	2387
T002	FDD 3,5	780
T003	HDD Maxtor 120GB	750
T004	Корпус MiniTower	30
T005	CD-ROM Panasonic IDE	750
T006	DIMM 64M PC100	200
T007	Принтер EPSON ST.A4	10
T008	Сканер Acer	4
T010	Модем Genius ext	400

Записи: 11 из 12 | Нет фильтра | Поиск

Рис 4.26. Результат группировки при объединении записей таблиц вторым способом

- Чтобы отобразить только те товары, по которым не сделано заказов, введите в столбец, где подсчитывается суммарное количество, в строку **Условие отбора** (Criteria) значение Null. Выполните запрос, чтобы проверить результат.

Параметры объединения в инструкциях SQL

- Откройте в режиме конструктора созданный в предыдущей задаче запрос, где для исходных таблиц ТОВАР и ПОСТАВКА_ПЛАН установлен второй способ объединения и для поля КОЛ_ПОСТ задано условие отбора Is Null.
- Перейдите в режим SQL, выбрав его из списка кнопки **Режим (View)**. Инструкция SQL для этого запроса будет иметь вид:

```
SELECT ПОСТАВКА_ПЛАН.КОД_ТОВ, ТОВАР.НАИМ_ТОВ,
       Sum(ПОСТАВКА_ПЛАН.КОЛ_ПОСТ) AS [Sum-КОЛ_ПОСТ]
FROM ТОВАР LEFT JOIN ПОСТАВКА_ПЛАН
ON ТОВАР.КОД_ТОВ = ПОСТАВКА_ПЛАН.КОД_ТОВ
GROUP BY ПОСТАВКА_ПЛАН.КОД_ТОВ, ТОВАР.НАИМ_ТОВ
HAVING (((Sum(ПОСТАВКА_ПЛАН.КОЛ_ПОСТ)) Is Null));
```

Приведенная инструкция SELECT использует для формирования структуры таблицы запроса, помимо двух полей из таблиц базы данных, статистическую функцию Sum. С помощью предложения AS полю, формируемому этой функцией, присваивается альтернативное имя.

ЗАМЕЧАНИЕ

Имя поля, введенное в его свойство Подпись (Caption), автоматически не переносится в инструкцию SQL. Чтобы в инструкции SELECT в предложении AS полю присваивалось альтернативное имя, оно должно быть в бланке запроса введено перед именем поля в следующем виде: Заказано товаров:КОЛ_ПОСТ.

Предложение FROM рассматриваемой инструкции SELECT определяет таблицы, которые служат источником данных для этого запроса, и способ их объединения: LEFT JOIN. При этом указывается по какому полю связываются таблицы:

```
ON ТОВАР.КОД_ТОВ = ПОСТАВКА_ПЛАН.КОД_ТОВ
```

Операция LEFT JOIN используется для создания *левого внешнего объединения*. Левое внешнее объединение включает помимо связанных записей все записи из первой (левой) таблицы, даже если для них нет связанных записей во второй (правой) таблице.

- Вернитесь в режим конструктора и измените способ объединения таблиц запроса, выбрав вместо второго параметра объединения третий. Предложение FROM для такого запроса будет иметь вид:

```
FROM ТОВАР RIGHT JOIN ПОСТАВКА_ПЛАН
ON ТОВАР.КОД_ТОВ = ПОСТАВКА_ПЛАН.КОД_ТОВ
```

Операция RIGHT JOIN используется для создания *правого внешнего объединения*. Правое внешнее объединение включает помимо связанных записей все записи из второй (правой) таблицы, даже если для них нет связанных записей в первой (левой) таблице.

Если просмотреть запрос, в котором выбран первый способ объединения таблиц, то в предложении FROM будет указана соответствующая операция INNER JOIN.

Предложение HAVING задает условия отбора групп, определяя какие из сгруппированных записей следует включить в таблицу запроса. Предложение HAVING аналогично предложению WHERE, которым определяется выбор записей.

- В режиме конструктора дополните запрос полем ЦЕНА, в строке **Групповая операция** замените слово **Группировка (Group By)** словом **Условие (Where)** и введите для него условие отбора: >1000. В поле с суммарным количеством товара удалите в условии отбора значение Is Null. Инструкция SELECT будет дополнена предложением WHERE.
- В условии отбора поля с суммарным количеством товара введите: >40 — осуществится отбор соответствующих сгруппированных записей. Инструкция SELECT будет дополнена предложением HAVING и примет вид:

```
SELECT ПОСТАВКА_ПЛАН.КОД_ТОВ, ТОВАР.НАИМ_ТОВ,
Sum(ПОСТАВКА_ПЛАН.КОЛ_ПОСТ) AS [Sum-КОЛ_ПОСТ]
FROM ТОВАР RIGHT JOIN ПОСТАВКА_ПЛАН
ON ТОВАР.КОД_ТОВ = ПОСТАВКА_ПЛАН.КОД_ТОВ
WHERE (((ТОВАР.ЦЕНА)>1000))
GROUP BY ПОСТАВКА_ПЛАН.КОД_ТОВ, ТОВАР.НАИМ_ТОВ
HAVING (((Sum(ПОСТАВКА_ПЛАН.КОЛ_ПОСТ))>40));
```

При выполнении такого запроса сначала в соответствии с предложением WHERE отбираются нужные записи, затем они группируются с помощью предложения GROUP BY, и после этого в соответствии с предложением HAVING производится отбор групп записей.

Ссылки на имена полей различных таблиц в условии отбора

В условии отбора в качестве операндов могут использоваться не только конкретные значения для отбора по значениям полей, но и ссылки на имена полей таблиц, на которых основывается запрос.

Задача

Пусть необходимо выбрать из таблицы ОТГРУЗКА записи, в которых указанная стоимость товара не соответствует произведению количества отгруженного на цену, указанную для этого товара в таблице ТОВАР, и рассчитать величину отклонения.

1. Для решения этой задачи создайте запрос, представленный на рис. 4.27.
2. Для отбора записей с неверно указанной суммой отгрузки в строке **Условие отбора (Criteria)** для поля СУММА_ОТПР запишите выражение:

<>[КОЛ_ОТПР]*[ЦЕНА]

Указание имен таблиц в ссылках на поля является не обязательным, потому что поля имеют уникальные имена. В противном случае при ссылке на поле перед его именем необходимо указывать имя таблицы, например: [ТОВАР].[ЦЕНА].

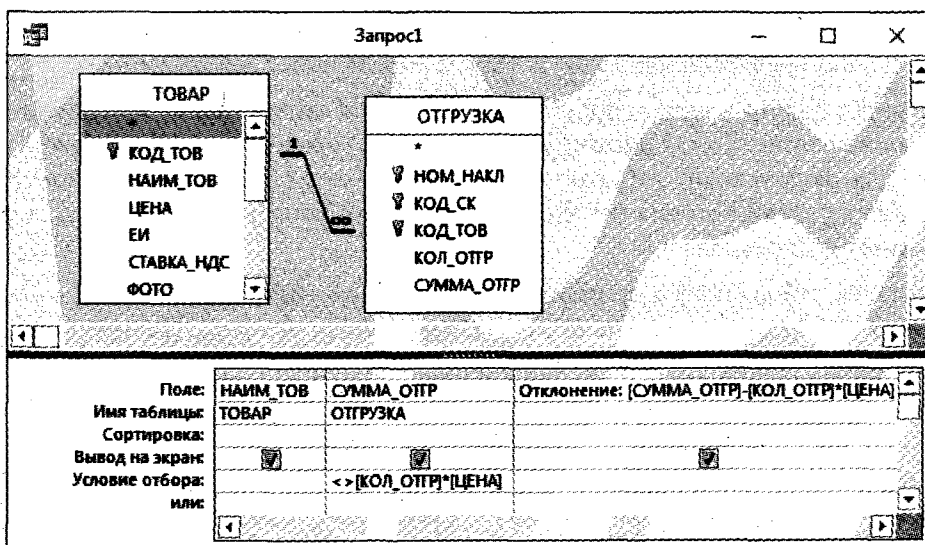


Рис. 4.27. Запрос с использованием в условиях отбора имен полей из разных таблиц

3. Для расчета величины, на которую отличается значение суммы в таблице от правильного, создайте вычисляемое поле, записав выражение:

[СУММА_ОТПР]-[КОЛ_ОТПР]*[ЦЕНА]

4. Замените сформированное по умолчанию имя вычисляемого поля **Выражение1 (Expr1)** на **Отклонение**.

Результаты выполнения запроса, в который добавлены еще поля с номером накладной **НОМ_НАКЛ** и кодом склада **КОД_СК**, приведены на рис. 4.28.

Код накладной	Код склада	Наименование товара	Сумма отгружено	Отклонение
H002	C01	FDD 3,5	165 000,00р.	150 000,00р.
H003	C03	FDD 3,5	14 000,00р.	13 000,00р.
H002	C02	HDD Maxtor 120GB	20 160,00р.	11 200,00р.
H003	C01	HDD Maxtor 120GB	14 000,00р.	-19 600,00р.
H001	C01	CD-ROM Panasonic IDE	8 000,00р.	6 400,00р.
H001	C02	CD-ROM Panasonic IDE	8 000,00р.	6 400,00р.
H003	C02	CD-ROM Panasonic IDE	8 000,00р.	6 400,00р.
H001	C01	Принтер EPSON ST.A4	22 000,00р.	-2 000,00р.
H001	C01	СканерAcer	2 338,00р.	-21 042,00р.

Рис. 4.28. Записи об отгрузке товаров с неверно указанной суммой

ЗАМЕЧАНИЕ

Если результат выполнения запроса не содержит записей, то это означает, что все суммы подсчитаны правильно, т. е. равны произведению количества на цену товара.

Просмотрите запрос в режиме SQL. Инструкция SELECT для этого запроса имеет вид:

```
SELECT ОТГРУЗКА.НОМ_НАКЛ, ОТГРУЗКА.КОД_СК, ТОВАР.НАИМ_ТОВАР,
    ОТГРУЗКА.СУММА_ОТГР, [СУММА_ОТГР]-[КОЛ_ОТГР]*[ЦЕНА] AS Отклонение
FROM ТОВАР INNER JOIN ОТГРУЗКА ON ТОВАР.КОД_ТОВАР = ОТГРУЗКА.КОД_ТОВАР
WHERE ((ОТГРУЗКА.СУММА_ОТГР) <> [КОЛ_ОТГР]*[ЦЕНА]);
```

В инструкции SELECT наряду с полями таблиц, включаемыми в таблицу запроса, представлено вычисляемое поле, которому присвоено имя **Отклонение**.

Задание 4.1. Создание многотабличного запроса

Подсчитайте суммарное количество каждого из товаров, которое должно быть поставлено заданному покупателю в каждом из месяцев. Результат должен содержать наименование покупателя, месяц поставки, наименование заказанного товара и количество. Используйте таблицы **ПОСТАВКА_ПЛАН**, **ТОВАР**, **ПОКУПАТЕЛЬ**, а также таблицу **ДОГОВОР**, через которую осуществляется связь таблицы **ПОСТАВКА_ПЛАН** с таблицей **ПОКУПАТЕЛЬ**.

Задание 4.2. Создание многотабличного запроса с параметрами

Подсчитайте общее количество каждого из товаров, отгруженное в заданный период. Результат должен содержать наименование товара, количество отгруженного

и заданный период. Используйте таблицы ОТГРУЗКА, ТОВАР и НАКЛАДНАЯ. Основой для образования записей этого запроса является таблица ОТГРУЗКА, подчиненная таблицам ТОВАР и НАКЛАДНАЯ. Код товара в таблице ОТГРУЗКА может быть заменен на наименование, выбранное из связанной записи таблицы ТОВАР. Каждая запись из таблицы ОТГРУЗКА может быть дополнена датой отгрузки товара из связанной записи в таблице НАКЛАДНАЯ.

Представление данных нарастающим итогом

Данные о плановых поставках товаров заказчику в соответствии с договорами хранятся в таблице ПОСТАВКА_ПЛАН. Ранее было показано, как с помощью несложного запроса группировки подсчитать, какое суммарное количество каждого из товаров должно быть поставлено заказчиком (выпущено и отгружено предприятием) в каждом отдельном месяце. Однако часто необходимо предоставлять не только помесечный план поставок, но и план поставок товаров нарастающим итогом. В таком плане указывается суммарное количество товара, которое необходимо поставить к концу каждого месяца от начала года, т. е. количество от месяца к месяцу будет нарастать. На рис. 4.29 представлены результаты выполнения запросов, содержащие данные о плане поставок товаров в каждом месяце и нарастающим итогом. Рассматривается план поставок на три месяца.

Код товара	Срок поставки	Сум-КОЛ_ПОСТ
T001	1	413
T001	2	512
T001	3	450
T002	1	150
T002	2	320
T002	3	300
T003	1	240
T003	2	510
T004	3	30
T005	1	750
T006	2	200
T007	2	10

Код товара	Номер месяца	Сум-КОЛ_ПОСТ
T001	1	413
T001	2	927
T001	3	1377
T002	1	150
T002	2	470
T002	3	770
T003	1	240
T003	2	750
T003	3	750
T004	3	30
T005	1	750
T005	2	750
T005	3	750
T006	2	200
T006	3	200
T007	2	10
T007	3	10

Рис. 4.29. Данные о плане поставок товаров в каждом месяце (слева) и нарастающим итогом (справа)

Создайте запрос, позволяющий получить план поставок товаров нарастающим итогом (рис. 4.30).

- Прежде чем конструировать запрос, создайте дополнительную таблицу КАЛЕНДАРЬ, которая содержит строки с номерами и наименованиями трех месяцев.

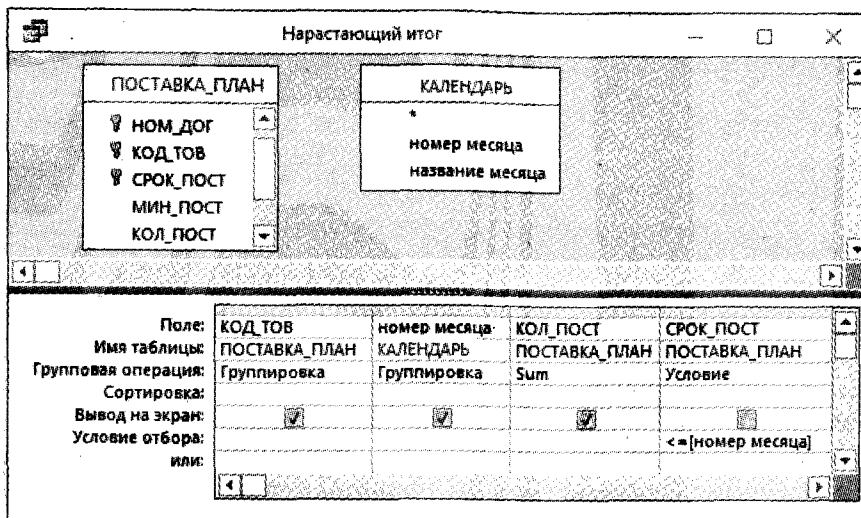


Рис. 4.30. Запрос для представления плана поставок изделий нарастающим итогом в режиме конструктора

2. Включите эту таблицу в запрос наряду с таблицей ПОСТАВКА_ПЛАН.
3. Между таблицами не устанавливайте никакого способа объединения записей. Это приведет к объединению записей таблиц с помощью операции «декартово произведение». При этом каждая строка таблицы ПОСТАВКА_ПЛАН будет дополнена выбранными полями из календаря и включена в результат столько раз, сколько месяцев в календаре, т. е. если в таблице ПОСТАВКА_ПЛАН 5 строк, «декартово произведение» даст 5×3 строк, т. к. в календаре содержится три строки.
4. Включите в бланк запроса поля, необходимые для выполнения отбора нужных строк, группировки и подсчета суммарных количеств в группах (см. рис. 4.30).
5. Введите в поле СРОК_ПОСТ условие отбора <=[номер месяца], чтобы оставить только те строки, в которых срок поставки товара меньше или равен номеру месяца из календаря. Это позволит получить для каждого месяца в календаре группу, включающую строки поставок товара за все предыдущие месяцы.
6. Выполните группировку по коду товара и номеру месяца.
7. Чтобы подсчитать суммарное количество каждого из товаров, которое должно быть выпущено за период до конца каждого месяца, выберите функцию Sum для поля с количеством.
8. Чтобы представить работу алгоритма вычисления нарастающего итога в более наглядном виде, сократите число строк в таблицах ПОСТАВКА_ПЛАН и КАЛЕНДАРЬ. Пусть в таблице ПОСТАВКА_ПЛАН хранится пять строк, представляющих план поставок одного изделия на три месяца, а в таблице КАЛЕНДАРЬ — три строки, представляющие три месяца (рис. 4.31).

The image shows two windows from a database application. The left window, titled 'ПОСТАВКА_ПЛАН', contains a table with columns: 'Номер договора', 'Код товара', 'Срок поставки', and 'Количество'. The data rows are: (D111, T001, 1, 100), (D111, T001, 2, 20), (D111, T001, 3, 30), (D222, T001, 1, 50), and (D222, T001, 2, 10). The status bar shows 'Запись: 4 из 6'. The right window, titled 'КАЛЕНДАРЬ', contains a table with columns: 'Номер месяца' and 'Название месяца'. The data rows are: (1, январь), (2, февраль), (3, март), and (4, *). The status bar shows 'Запись: 4 из 4'.

Номер договора	Код товара	Срок поставки	Количество
D111	T001	1	100
D111	T001	2	20
D111	T001	3	30
D222	T001	1	50
D222	T001	2	10
*		0	0

Номер месяца	Название месяца
1	январь
2	февраль
3	март
4	*

Рис. 4.31. Таблицы с данными о плане поставок товаров (слева) и календарь (справа)

«Декартово произведение» этих таблиц представлено на рис. 4.32 и объединяет каждую строку плана с каждым месяцем календаря. Там же показан результат выполнения операции отбора записей, содержащий только 11 строк. В соответствии с заданным условием отбора из «декартова произведения» остаются только те строки, в которых срок поставки меньше или равен номеру месяца из календаря.

В результате группировки по коду товара и номеру месяца из календаря для каждого товара формируется три группы по числу месяцев в календаре. Таким образом, для каждого товара будет получено столько строк, сколько месяцев включает календарь, а суммарное количество будет равно запланированному к выпуску на конец каждого месяца (рис. 4.33).

The image shows two windows from a database application. The left window, titled 'Запрос1', contains a table with columns: 'Код товара', 'Срок поставки', 'Количество', and 'Номер месяца'. The data rows are: (T001, 1, 100, 1), (T001, 1, 100, 2), (T001, 1, 100, 3), (T001, 2, 20, 1), (T001, 2, 20, 2), (T001, 2, 20, 3), (T001, 3, 30, 1), (T001, 3, 30, 2), (T001, 3, 30, 3), (T001, 1, 50, 1), (T001, 1, 50, 2), (T001, 1, 50, 3), (T001, 2, 10, 1), (T001, 2, 10, 2), (T001, 2, 10, 3). The status bar shows 'Запись: 1 из 15'. The right window, titled 'Запрос2', contains a table with columns: 'Код товара', 'Номер месяца', and 'Количество'. The data rows are: (T001, 1, 100), (T001, 2, 100), (T001, 3, 100), (T001, 2, 20), (T001, 3, 20), (T001, 3, 30), (T001, 1, 50), (T001, 2, 50), (T001, 3, 50), (T001, 2, 10), (T001, 3, 10). The status bar shows 'Запись: 1 из 11'.

Код товара	Срок поставки	Количество	Номер месяца
T001	1	100	1
T001	1	100	2
T001	1	100	3
T001	2	20	1
T001	2	20	2
T001	2	20	3
T001	3	30	1
T001	3	30	2
T001	3	30	3
T001	1	50	1
T001	1	50	2
T001	1	50	3
T001	2	10	1
T001	2	10	2
T001	2	10	3

Код товара	Номер месяца	Количество
T001	1	100
T001	2	100
T001	3	100
T001	2	20
T001	3	20
T001	3	30
T001	1	50
T001	2	50
T001	3	50
T001	2	10
T001	3	10

Рис. 4.32. «Декартово произведение» таблиц ПОСТАВКА_ПЛАН и КАЛЕНДАРЬ до (слева) и после (справа) отбора записей в соответствии с условиями

ОБРАТИТЕ ВНИМАНИЕ

При выполнении запроса сначала производится объединение записей и их отбор в соответствии с заданными условиями и только затем группировка полученных записей.

Код товара	Номер месяца	Sum-КОЛ_П
T001	1	150
T001	2	180
T001	3	210

Рис. 4.33. План поставок товара нарастающим итогом

9. Просмотрите запрос в режиме SQL. Инструкция SELECT для этого запроса имеет вид:

```
SELECT ПОСТАВКА_ПЛАН.КОД_ТОВ, КАЛЕНДАРЬ.[номер месяца],
       Sum(ПОСТАВКА_ПЛАН.КОЛ_ПОСТ) AS [Sum-КОЛ_ПОСТ]
FROM ПОСТАВКА_ПЛАН, КАЛЕНДАРЬ
WHERE ((ПОСТАВКА_ПЛАН.СРОК_ПОСТ)<=[номер месяца]))
GROUP BY ПОСТАВКА_ПЛАН.КОД_ТОВ, КАЛЕНДАРЬ.[номер месяца];
```

В инструкции SELECT в предложении FROM ПОСТАВКА_ПЛАН, КАЛЕНДАРЬ имена таблиц разделены запятой, что и определяет операцию «декартово произведение» при объединении записей таблиц.

Предложение WHERE ПОСТАВКА_ПЛАН.СРОК_ПОСТ<=[номер месяца] определяет отбор нужных записей. И наконец, предложение GROUP BY определяет поля группировки. Суммирование количества для группы выполняется соответствующей функцией, записанной в списке полей инструкции.

Тот же результат может быть получен, если в инструкции SQL заменить приведенные в примере предложения FROM и WHERE одним предложением:

```
FROM ПОСТАВКА_ПЛАН INNER JOIN КАЛЕНДАРЬ ON
ПОСТАВКА_ПЛАН.СРОК_ПОСТ<=КАЛЕНДАРЬ.[номер месяца]
```

Однако такая инструкция не может быть представлена в режиме конструктора Access. Выражение объединения, использующее оператор <=, не поддерживается конструктором. Адекватное отображение в режиме конструктора Access инструкции с выражением объединения возможно только при использовании для сравнения полей связи знак =.

Решение задачи, требующей выполнения нескольких запросов

Чтобы решить сложную задачу, чаще всего необходимо разбить ее на несколько подзадач, алгоритм каждой из которых может быть реализован выполнением одного достаточно простого запроса. Для каждого из таких запросов определяются входные и выходные данные. В простейшем случае выходные данные предшествующего запроса являются входными для следующего, построенного на нем запроса. При этом, выполнив только последний запрос в цепочке построенных друг на друге запросов, вы инициируете последовательное выполнение всех запросов

цепочки и полное решение задачи. Для представления алгоритмов решения задач целесообразно использовать функционально-технологическую схему, на которой приводится цепочка всех запросов с описанием и входными и выходными данными.

Задача

Проанализируйте выполнение общего плана поставок каждого из товаров на конец заданного месяца. При решении этой задачи подсчитайте количество товара, запланированного к поставке и отгруженного покупателям, и получите величину недопоставки каждого из товаров на конец заданного месяца.

1. Данные о плановых поставках хранятся в таблице ПОСТАВКА_ПЛАН. На ее основе создайте запрос для подсчета суммарного количества каждого из товаров, запланированных к поставке на конец заданного месяца (рис. 4.34).

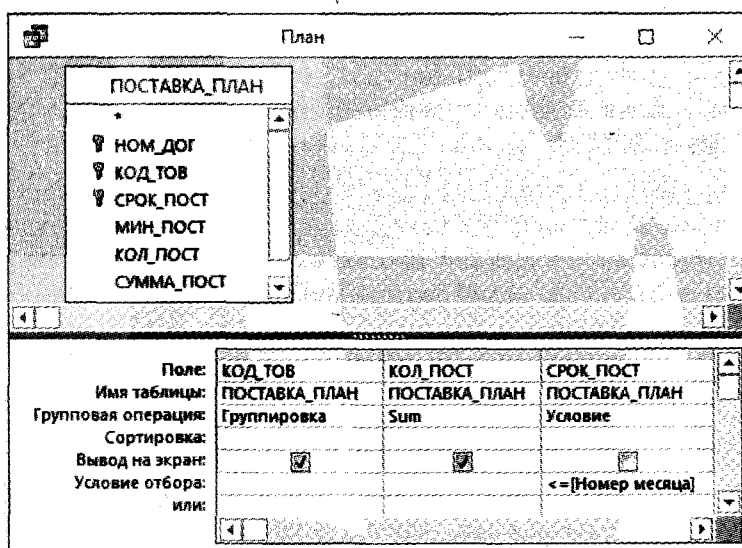


Рис. 4.34. Подсчет суммарного количества товаров, запланированных к поставке

2. Сохраните запрос под именем План.

В запросе План производится группировка записей таблицы по полю **Код товара**. Для операции используются только записи, в которых срок поставки (месяц) имеет значение меньше или равно заданному параметром запроса с именем **Номер месяца**. В каждой группе записей о поставках одного товара суммируются количества, запланированные к поставке.

3. Данные о фактически поставленных, отгруженных покупателям товарах, хранятся в таблице ОТГРУЗКА. На ее основе создайте запрос для подсчета суммарного количества отгруженных товаров. Для отбора поставок, выполненных до конца заданного месяца, в запрос добавьте таблицу НАКЛАДНАЯ, в которой хранится дата отгрузки товаров (рис. 4.35).

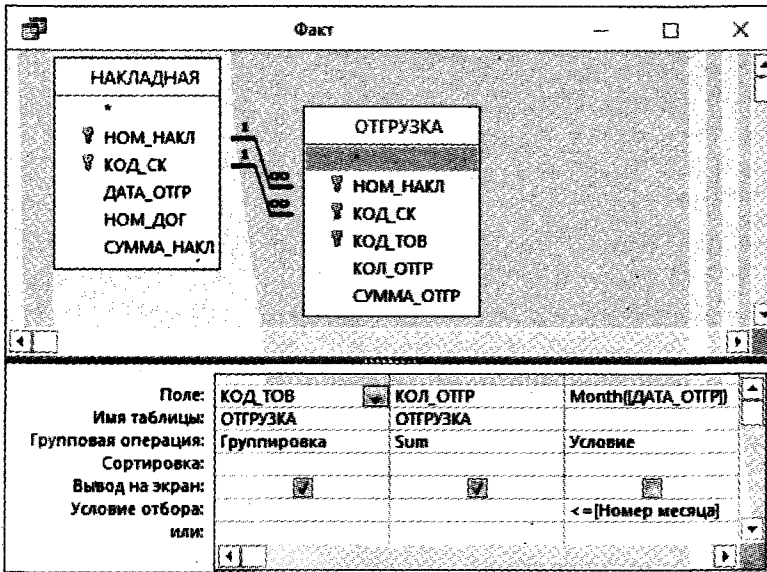


Рис. 4.35. Подсчет суммарного количества отгруженных товаров

4. Сохраните запрос под именем **Факт**.
5. Таблицы, на которых построен запрос **Факт**, находятся в отношении «один-многим». Для их связи по составному ключу установлены параметры обеспечения целостности. В результате объединения этих таблиц первым способом формируется таблица запроса с числом записей, равным числу записей в подчиненной таблице **ОТГРУЗКА**. Причем каждая запись об отгрузках товара дополняется датой из связанной записи главной таблицы **НАКЛАДНАЯ**.
6. Для отбора только тех накладных, по которым отгружался товар до конца заданного месяца, из даты отгрузки с помощью функции **Month** выделяется номер месяца и для этого вычисляемого поля в условии отбора вводится параметр запроса с именем **Номер месяца**, совпадающим с именем параметра в предыдущем запросе.
7. Для сравнения количества запланированного к поставке и отгруженного создайте новый запрос. Добавьте в него два предыдущих запроса: **План** и **Факт**.
8. Поскольку некоторые из запланированных товаров могли не отгружаться и в то же время могла производиться отгрузка товаров, которые не были запланированы, добавьте в запрос таблицу **ТОВАР**, в которой представлена вся номенклатура товаров фирмы. Свяжите эту таблицу с каждым из запросов по полю **КОД_ТОВ** и укажите на объединение ее записей с записями таблиц запросов **План** и **Факт** вторым способом (рис. 4.36). Такое объединение таблиц приведет к включению в результат не только тех записей о товарах, по которым были определены планы поставки и отгрузки, но и тех записей:
 - по которым был определен план поставки, но эти товары не отгружались вовсе;

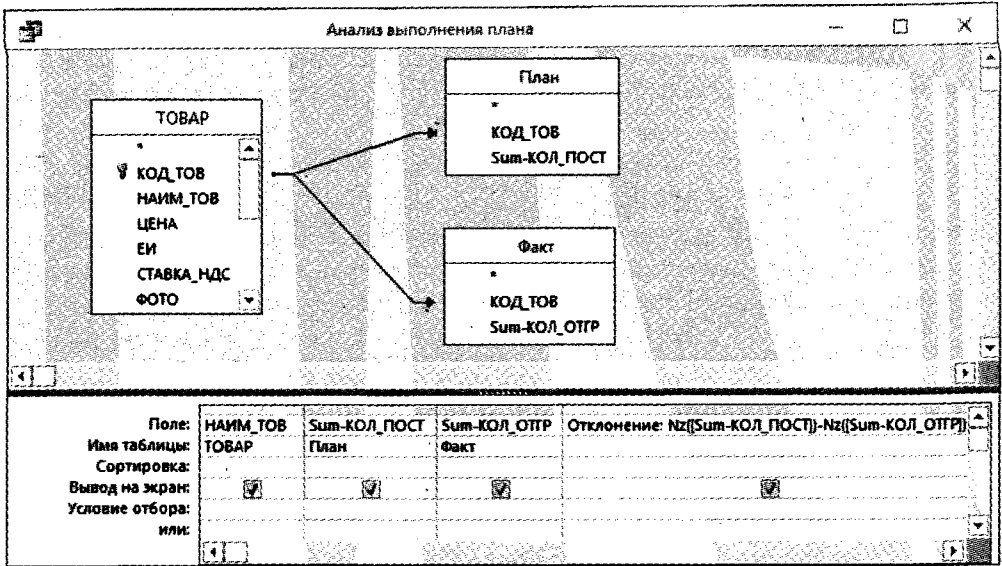


Рис. 4.36. Вычисление разности плановых и фактических поставок товаров

- которые отгрузались несмотря на то, что не были запланированы к поставке;
 - которые вообще не планировались к поставке и не отгрузались.
9. Обратите внимание, что при формировании записей вторым способом объединения в таблицу запроса включаются и те записи из левой таблицы (ТОВАР), для которых нет связанных записей в правой таблице (например, План). При этом в формируемой записи поля, выбираемые из правой таблицы, заполняются значением Null.
 10. Учитывая, что во всех пустых полях таблицы запроса определено значение Null, для получения разности между запланированным и фактически поставленным количеством товара создайте вычисляемое поле с выражением $Nz([Sum-КОЛ_ПОСТ]) - Nz([Sum-КОЛ_ОТПР])$. Присвойте вычисляемому полю имя Отклонение. Функция Nz возвращает существующее в поле значение или — для поля со значением Null — новое значение, указанное вторым аргументом. Если второй аргумент не указан, по умолчанию для числового поля со значением Null возвращается 0, а для символического — пустая строка.
 11. Результатом арифметических операций с полем Null является Null. Убедитесь в этом, исключив из выражения функцию Nz.
 12. Сохраните запрос под именем Анализ выполнения плана.
 13. Выполните запрос. В таблице запроса Анализ выполнения плана (рис. 4.37) представлен весь список товаров фирмы. Показано, что по большинству заказанных товаров выполнялась отгрузка, по одному виду товара отгрузок не было, один вид товара был отгружен без предварительного оформления договора, и два вида товаров вообще не были заказаны в договорах.

Наименование товара	Sum-КОЛ_ПОСТ	Sum-КОЛ_ОТПР	Отклонени
Монитор 17LG	1377	202	1175
FDD 3,5	770	167	603
HDD Maxtor 120GB	750	42	708
Корпус MiniTower	30	100	-70
CD-ROM Panasonic IDE	750	12	738
DIMM 64M PC100	200	11	189
Принтер EPSON ST.A4	10	10	0
СканерAcer	4	32	-28
Зв. Карта Genius Liv		13	-13
Модем Genius ext	400		400
Плоттер			0
Миникомпьютер			0

Рис. 4.37. Результат вычисления разности плановых и фактических поставок товаров

ОБРАТИТЕ ВНИМАНИЕ

Выполнение запроса Анализ выполнения плана инициирует выполнение запросов План и Факт, и нет необходимости в их предварительном выполнении.

- Запишите в условие отбора поля Sum-КОЛ_ПОСТ значение Not Null. Отобразятся только те строки, которые относятся к товарам, на которые были заключены договоры.
- Для того чтобы явно увидеть, в какой последовательности производится объединение таблиц запроса, просмотрите запрос в режиме SQL. Инструкция SELECT для этого запроса имеет вид:

```
SELECT ТОВАР.НАИМ_ТОВАР, План.[Sum-КОЛ_ПОСТ], Факт.[Sum-КОЛ_ОТПР],
       NZ([Sum-КОЛ_ПОСТ]) - NZ([Sum-КОЛ_ОТПР]) AS Отклонение
FROM (ТОВАР LEFT JOIN План ON ТОВАР.КОД_ТОВАР = План.КОД_ТОВАР)
     LEFT JOIN Факт ON ТОВАР.КОД_ТОВАР = Факт.КОД_ТОВАР;
```

В предложении FROM указано, что сначала будет производиться левое внешнее объединение таблиц ТОВАР и План по полю связи Код товара (ТОВАР LEFT JOIN План ON ТОВАР.КОД_ТОВАР = План.КОД_ТОВАР). Затем будет производиться левое внешнее объединение таблицы, полученной в результате первого объединения, и таблицы Факт по тому же полю связи.

Объединение таблиц в рассматриваемом запросе дает правильный результат только потому, что в каждой из объединяемых таблиц КОД_ТОВАР имеет уникальные неповторяющиеся значения: в таблице ТОВАР поле КОД_ТОВАР определено ключевым изначально, а в таблицах запросов План и Факт поле КОД_ТОВАР также имеет уникальные значения, т. к. они получены в результате группировки по этому полю.

- Попробуйте подсчитать суммарное количество плановых и фактических поставок товаров одним запросом, представленным на рис. 4.38.

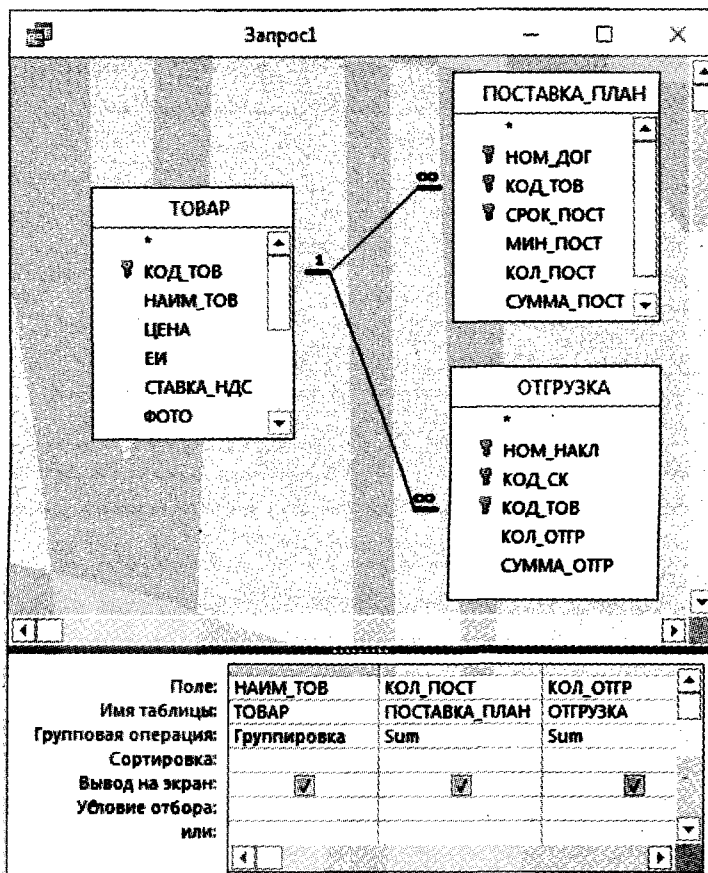


Рис. 4.38. Запрос, объединяющий записи таблиц, находящихся в отношении M : N

17. Проанализируйте результат выполнения запроса и убедитесь, что он неправильный. Неверный результат образуется вследствие того, что в многотабличных запросах выполнение группировки производится после формирования записей путем объединения записей таблиц и их выборки в соответствии с условиями отбора.

Объединение записей таблиц ПОСТАВКА_ПЛАН и ОТГРУЗКА по полю КОД_ТОВ, значения которого многократно повторяются в каждой из таблиц, приводит к неоднократному включению в результат как записей из первой таблицы, так и записей из второй таблицы. Например, даже в результате объединения показанных на рис. 4.39 таблиц будет получена таблица, в которой количество запланированного к поставке будет повторено дважды, и при группировке будет получено удвоенное количество плановой поставки. Если в первую таблицу включить неоднократно повторяющиеся записи по одному и тому же товару, в результат объединения будут повторно включаться одни и те же записи об отгрузках. При группировке это приведет к неверному подсчету количества отгруженного.

Таблица ПОСТАВКА_ПЛАН:

КОД_ТОВ	КОЛ_ПОСТ
T1	100
T2	100

Таблица ОТГРУЗКА:

КОД_ТОВ	КОЛ_ОТГР
T1	10
T1	50
T2	50
T2	60

Результат объединения:

КОД_ТОВ	КОЛ_ПОСТ	КОЛ_ОТГР
T1	100	10
T1	100	50
T2	100	50
T2	100	60

Рис. 4.39. Пример объединения записей таблиц по полю с повторяющимися значениями

Задание 4.3

Проанализируйте план поставок товара заданному покупателю.

Задание 4.4

Проанализируйте план поставок товара в стоимостном выражении.

Задание 4.5

Подсчитайте, на какую сумму отгружен товар покупателям, которые не заключали договоров.

Задание 4.6

Подсчитайте, на какую сумму покупатели недополучили товар на текущую дату.

Контрольные вопросы

к разд. «Многотабличные запросы на выборку данных»

1. Какой способ объединения записей двух таблиц по умолчанию устанавливается при определении их связи в схеме данных?
2. Мешает ли флажок **Обеспечение целостности данных** (Enforce Referential Integrity) установлению параметра объединения?
3. Какой командой открывается возможность выбора параметра объединения таблиц в запросе?

4. Запрос построен на двух таблицах, находящихся в отношениях 1 : М, для которых установлен параметр обеспечения целостности. Первая таблица имеет пять записей, а вторая 10, причем 10 записей второй таблицы подчинены трем записям первой. Сколько записей будет включено в таблицу запроса, если для исходных таблиц выбран первый способ объединения записей, и если — второй?
5. Изменится ли параметр объединения таблиц, установленный в запросе, если изменить его в схеме данных базы?
6. Какая команда позволяет отобразить в бланке запроса строку с именами таблиц?
7. Какие поля представлены в списке, открываемом в строке Поле (Field) бланка запроса?
8. Сколько записей будет содержать таблица запроса, если она построена на двух несвязанных таблицах по 5 записей?
9. Какая операция будет применена к записям объединяемых в запросе таблиц, если между ними не установлена связь?
10. Можно ли в инструкции SELECT записать выражение объединения таблиц, использующее оператор `<=` ?
11. Будет ли инструкция SELECT с выражением объединения таблиц, использующим оператор `<=`, адекватно отображена в режиме конструктора Access?
12. Как вводится параметр в запрос?
13. Как в инструкции SQL указать на использование конкретного поля, если в таблицах, на которых строится запрос, имеются одинаковые имена?
14. Допускается ли группировка записей запроса по нескольким полям?
15. Нужно ли последовательно выполнять запрос1, запрос2, запрос3, если запрос3 построен на запросе2, а запрос2 на запросе1?
16. В каком предложении и какой операцией задается способ объединения записей таблиц в инструкции SQL?
17. После какого ключевого слова в предложении FROM указываются поля связи таблиц?
18. Как в предложении FROM указывается вложенность операций объединения таблиц?
19. Можно ли в инструкции SELECT в качестве значения условия отбора использовать параметр?
20. Каким предложением определяется необходимость в сортировке?
21. Какая операция выполняется первой в многотабличном запросе с группировкой: объединение записей таблиц или группировка?
22. Как устанавливаются параметры объединения таблиц в запросе?

Ответы

1. Первый, при котором объединяются только те записи, в которых связанные поля обеих таблиц совпадают.
2. Нет.
3. В схеме данных запроса в контекстном меню выделенной связи таблиц кнопкой **Параметры объединения...** (Join Properties) открывается соответствующее окно, где и выполняется выбор типа объединения.
4. 10 и 12 соответственно.
5. Нет.
6. Команда **Имена таблиц** (Table Names) в группе **Показать или скрыть** (Show/Hide) на вкладке конструктора запросов.
7. Поля всех таблиц запроса.
8. 25.
9. «Полное объединение» или «декартово произведение».
10. Да.
11. Нет.
12. Как текстовая строка, заключенная в квадратные скобки.
13. Перед именем поля поставить имя таблицы и разделить их точкой.
14. Да.
15. Нет. Чтобы получить результат, достаточно выполнить запрос 3.
16. В предложении FROM операциями INNER JOIN, LEFT JOIN и RIGHT JOIN.
17. ON.
18. С помощью круглых скобок.
19. Да.
20. ORDER BY.
21. Объединение записей таблиц.
22. С помощью команды **Параметры объединения** (Join Properties) в контекстном меню линии связи таблиц.

Запросы на изменение

К запросам на изменение относятся запросы на обновление данных в записях таблицы базы, на добавление и удаление записей из таблицы, а также запросы на создание таблицы на основе сформированных запросом записей.

Для разработки запросов на изменение в Access можно использовать конструктор. Процесс создания любого запроса на изменение начинается с создания запроса на выборку, который после добавления в него необходимых таблиц нужно преобразовать в запрос на изменение.

По умолчанию Access 2016 в целях обеспечения безопасности, как правило, блокирует выполнение всех запросов на изменение. Если при выполнении запроса на изменение ничего не происходит, проверьте, не появляется ли в строке состояния Access сообщение:

Данное действие или событие заблокировано в режиме отключения.

Если отображается это сообщение и под лентой — панель сообщений (рис. 4.40), для включения заблокированных запросов можно нажать на ней кнопку **Включить содержимое** (Enable Content). После этого выполнение запроса станет доступным.

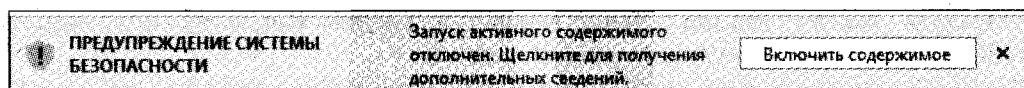


Рис. 4.40. Панель сообщений (Message Bar)

Если панель сообщений была закрыта и больше не отображается, нажмите на значок **Файл** (File) и на открывшейся странице **Сведения** (Info) в блоке **Предупреждение системы безопасности** (Security Warning) нажмите кнопку **Включить содержимое** (Enable Content) (рис. 4.41).

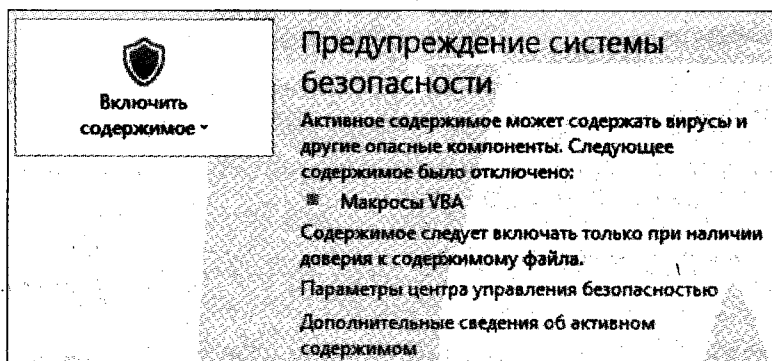


Рис. 4.41. Блок включения потенциально опасного содержимого

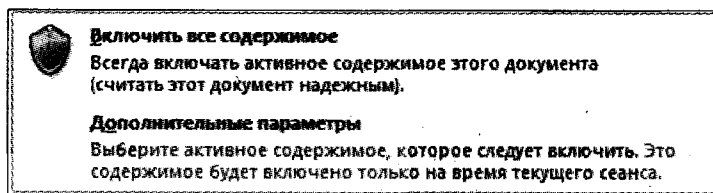


Рис. 4.42. Выбор варианта включения потенциально опасного содержимого

В списке **Включить содержимое** (Enable Content) (рис. 4.42) можно **Включить все содержимое** (Enable All Content) открытой базы данных. Это приведет к тому, что при повторных открытиях базы ни панель сообщения, ни сообщения о невозмож-

ности выполнения запроса на изменение выводиться не будут, т. к. база данных будет отнесена к разряду надежных. То же самое происходит, если была нажата кнопка **Включить содержимое** (Enable Content) на панели сообщений.

Выбор в списке **Включить содержимое** (Enable Content) строки **Дополнительные параметры** (Advanced Options) открывает окно параметров безопасности Microsoft Office, в котором можно включить опасное содержимое только на время сеанса. При следующем открытии базы данных опять появится панель сообщений и, если не включать содержимое, запросы на изменение выполняться не будут.

Чтобы вернуть возможность управления содержимым, соответствующей командой на странице, открывающейся выбором значка **Файл**, откройте окно **Параметры** (Options), щелкните на строке **Центр управления безопасностью** (Trust Center) и далее выберите опцию **Параметры центра управления безопасностью** (Trust Center Settings). В окне центра на странице **Надежные документы** (Trusted Documents) в строке **Сбросить пометку о надежности для всех надежных документов** (Clear all Trusted Documents so that they are no longer trusted) нажмите кнопку **Очистить** (Clear).

ОБРАТИТЕ ВНИМАНИЕ!

Единое средство вывода предупреждений системы безопасности — панель сообщений — по умолчанию появляется при открытии базы данных Access 2016 вне доверенного расположения. Если точно известно, что можно доверять содержимому базы данных, включите все отключенные потенциально опасные, активные компоненты: запросы на изменение, макросы, элементы управления ActiveX, некоторые выражения и программы на VBA — при открытии базы данных, содержащей один или несколько этих компонентов.

Запрос на создание таблицы

Запрос на создание таблицы используется для сохранения результата выполнения запроса. Этот вид запроса основан на запросе на выборку, но, в отличие от него, сохраняет в базе данных новую таблицу с результатами запроса.

Сформируем запрос на создание таблицы на примере ранее полученного запроса на выборку с групповыми вычислениями **Число отгрузок по договорам** (см. рис. 4.16 и 4.17).

1. В области навигации выделите названный запрос и с помощью команды контекстного меню откройте его в режиме конструктора.
2. Преобразуйте этот запрос на выборку в запрос на создание таблицы, выполнив команду **Создание таблицы** (Make Table) в группе **Тип запроса** (Query Type) на вкладке конструктора или выбрав команду контекстного меню запроса **Тип запроса | Создание таблицы** (Query Type | Make Table Query).
3. В окне **Создание таблицы** (Make Table) введите имя создаваемой таблицы: **Число отгрузок** (рис. 4.43).
4. Для того чтобы просмотреть, какие записи будут помещены в новую таблицу, щелкните по кнопке **Режим** (View) на ленте конструктора запросов в группе **Результаты** (Results).

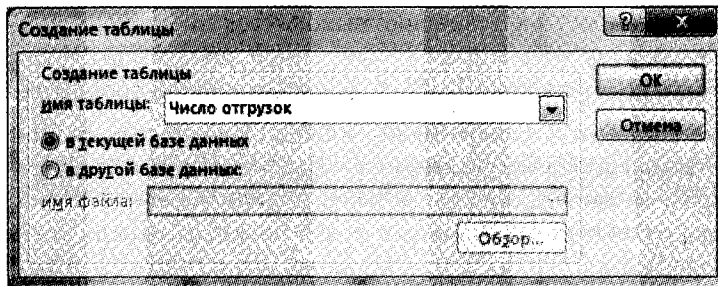


Рис. 4.43. Определение имени таблицы, создаваемой в запросе

5. Выполните запрос, чтобы таблица `Число отгрузок` была сохранена в базе данных. Теперь эту таблицу можно увидеть в списке таблиц области навигации.
6. Перейдите в режим SQL. Эквивалентная запросу на создание таблицы инструкция `SELECT ... INTO` будет записана следующим образом:

```
SELECT НАКЛАДНАЯ.НОМ_ДОГ, Count(НАКЛАДНАЯ.НОМ_ДОГ AS [Count-НОМ_ДОГ]
      INTO [Число отгрузок]
FROM НАКЛАДНАЯ
GROUP BY НАКЛАДНАЯ.НОМ_ДОГ;
```

Инструкция `SELECT ... INTO` аналогично инструкции `SELECT` выполняет выборку данных, но, в отличие от нее, для сохранения выбранных данных создает новую таблицу, имя которой указывается в предложении `INTO`. Записи новой таблицы формируются как в обычном запросе на выборку. Структура записи определяется перечисленными в инструкции `SELECT` полями таблиц, указанными в предложении `FROM`. Допустимо использование предложения `GROUP BY`, определяющего поле группировки.

В рассмотренном примере список полей новой таблицы включает два поля, одно из них вычисляемое с помощью статистической функции `Count`. Если перед предложением `GROUP BY` поместить предложение `WHERE`, группы будут формироваться из записей, отобранных в соответствии с условиями, заданными в предложении `WHERE`.

Запрос на обновление

Для обновления данных в полях базовых таблиц может быть использован запрос **Обновление** (`Update`). Изменения вносятся в группу записей, выбираемых с помощью указанных пользователем условий отбора. Обновляемые поля и значения для изменений в этих полях определяются в бланке запроса в строке **Обновление** (`Update To`).

Задача

Рассчитайте стоимость товара в каждой строке таблицы `ОТГРУЗКА` и сохраните ее в поле `СУММА_ОТГР` этой же таблицы.

1. Для формирования запроса на обновление сначала создайте запрос **Выборка** (`Select`) на основе двух таблиц: обновляемой таблицы `ОТГРУЗКА` и таблицы

ТОВАР, в которой содержится необходимая для вычисления стоимости цена товара.

- Преобразуйте запрос на выборку в запрос на обновление, щелкнув на кнопке **Обновление (Update)**, размещенной на вкладке ленты **Конструктор (Design)**, или выбрав команду **Обновление (Update)** из списка **Тип запроса (Query Type)** в контекстном меню запроса. После выполнения этой команды в бланке запроса появляется строка **Обновление (Update To)** (рис. 4.44).

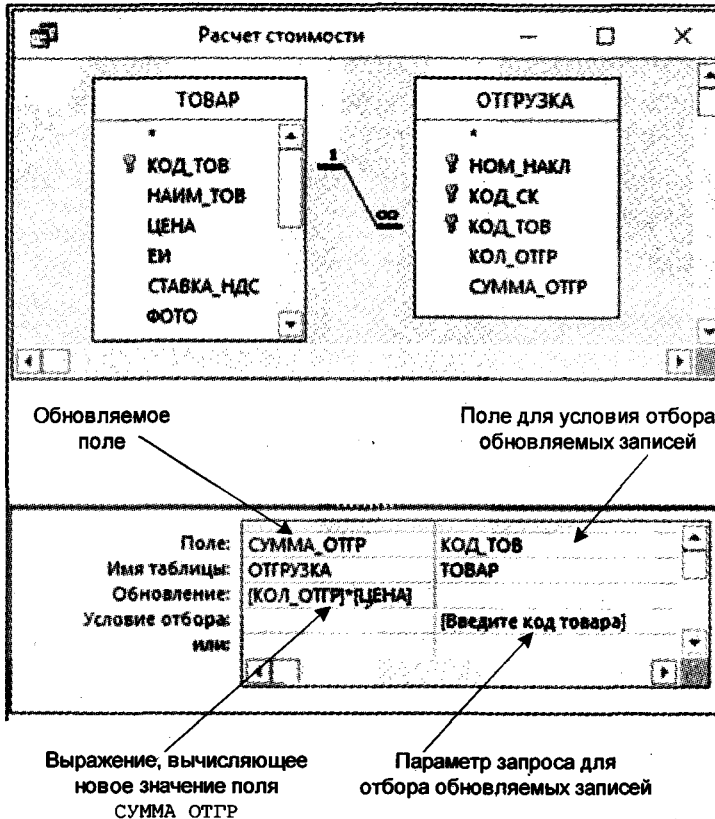


Рис. 4.44. Пример запроса на обновление

- Заполните бланк запроса. Перетащите обновляемое поле `СУММА_ОТПР` из списка таблицы `ОТГРУЗКА` в строку **Поле**. В строку **Обновление (Update To)** введите выражение `[ЦЕНА] * [КОЛ_ОТПР]`, которое рассчитывает значение для обновления.
- Просмотрите содержимое обновляемого поля `СУММА_ОТПР` перед выполнением запроса, нажав кнопку **Режим (View)** на ленте конструктора запросов в группе **Результаты (Results)**.
- Для обновления содержимого поля `СУММА_ОТПР` выполните запрос, нажав кнопку **Выполнить (Run)** на вкладке ленты **Конструктор (Design)**. Откроется диалоговое окно с сообщением о числе обновляемых записей и вопросом о продолжении операции обновления. Подтвердите обновление записей.

6. Просмотрите содержимое обновляемого поля СУММА_ОТГР после выполнения запроса. Для этого переключитесь после выполнения запроса в режим таблицы, воспользовавшись кнопкой **Режим таблицы (Datasheet View)** в строке состояния или нажмите кнопку **Режим (View)** на вкладке ленты.

Рассмотренный запрос позволяет автоматизировать расчет стоимости товара, указанного в каждой строке спецификации накладной, — записи таблицы ОТГРУЗКА.

7. Если обновлять нужно только некоторые строки таблицы, задайте условия отбора обновляемых записей. Для этого дополните бланк запроса полем, по которому требуется произвести отбор записей. Перетащите поле КОД_ТОВАР в бланк запроса и введите в строку **Условие отбора (Criteria)** параметр [Введите код товара] (см. рис. 4.44).
8. Выполните запрос. Обновление будет выполнено только для записей с введенным кодом товара.
9. Сохраните запрос под именем **Расчет стоимости**.

ОБРАТИТЕ ВНИМАНИЕ

При использовании в запросе на обновление таблиц, находящихся в отношении 1 : М, обновлять можно только содержимое полей таблицы со стороны «многие».

10. Перейдите в режим SQL. Эквивалентная запросу на обновление инструкция UPDATE будет записана следующим образом:

```
UPDATE ТОВАР INNER JOIN ОТГРУЗКА ON ТОВАР.КОД_ТОВАР = ОТГРУЗКА.КОД_ТОВАР
SET ОТГРУЗКА.СУММА_ОТГР = [ЦЕНА]*[КОЛ_ОТГР]
WHERE (((ТОВАР.КОД_ТОВАР)=[Введите код товара]));
```

Имена таблиц, используемых в запросе, и способ их объединения задаются непосредственно за именем инструкции UPDATE. Инструкция UPDATE обновляет указанное в предложении SET поле ОТГРУЗКА.СУММА_ОТГР, присваивая ему значение, заданное выражением [ЦЕНА]*[КОЛ_ОТГР]. Обновление происходит во всех записях, которые удовлетворяют условию отбора, заданному в предложении WHERE, то есть в этом примере — во всех записях с введенным значением кода товара.

Обновление полей значениями, рассчитанными в запросе с группировкой

В запросе на обновление не могут быть использованы значения, вычисляемые в запросе с группировкой. Чтобы обойти это ограничение, следует преобразовать запрос с группировкой в запрос на создание таблицы.

Задача

Пусть необходимо рассчитать общую стоимость товара, отгруженного по каждой накладной, и сохранить ее в соответствующем поле таблицы НАКЛАДНАЯ. Решение этой задачи состоит из двух этапов. На первом этапе на основе данных из таб-

лицы ОТГРУЗКА подсчитывается общая стоимость накладной, на втором — полученными значениями обновляется соответствующее поле в таблице НАКЛАДНАЯ.

ЗАМЕЧАНИЕ

Рассмотренный ранее запрос Расчет стоимости (см. рис. 4.44) позволяет автоматизировать расчет стоимости товара, указанного в каждой строке спецификации накладной, — записи таблицы ОТГРУЗКА. Для этого достаточно убрать в нем параметр условия отбора.

1. Для подсчета общей стоимости товара, отгруженного по каждой накладной, создайте запрос на основе таблицы ОТГРУЗКА.
2. Включите в бланк запроса поля, определяющие составной ключ накладной: номер накладной (НОМ_НАКЛ), код склада (КОД_СК) и поле СУММА_ОТГР.
3. Выполните группировку по двум полям, определяющим составной ключ накладной. Это приведет к включению в группу всех записей об отгрузках товаров по одной накладной. Для поля СУММА_ОТГР выберите статистическую функцию Sum. Для группы будет подсчитана стоимость всех товаров, отгруженных по накладной.
4. Теперь значения стоимости накладных, рассчитанные с использованием статистической функции Sum, нужно сохранить в соответствующем поле таблицы НАКЛАДНАЯ. Однако запрос на обновление поля в таблице НАКЛАДНАЯ непосредственно на запросе с группировкой построить нельзя. Поэтому преобразуйте запрос на выборку в запрос на создание таблицы. Создаваемой в запросе таблице дайте имя Стоимость накладной. Далее эту таблицу используйте для обновления поля СУММА_НАКЛ в накладной.
5. Сохраните запрос под именем Расчет стоимости накладной (рис. 4.45).

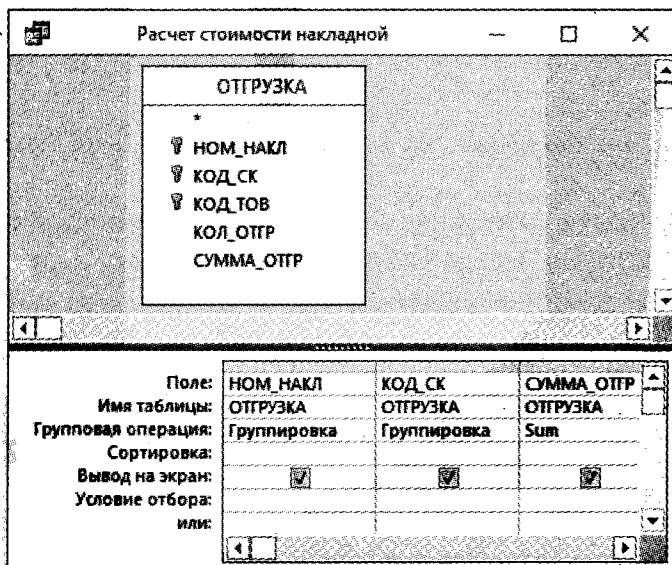


Рис. 4.45. Расчет стоимости всего товара, отгруженного по накладной, и сохранение его в новой таблице

6. Создайте запрос на обновление на основе обновляемой таблицы базы данных НАКЛАДНАЯ и новой таблицы Стоимость накладной, содержащей данные для обновления.
7. В схеме данных запроса установите связь таблиц по полям, однозначно идентифицирующим накладную: НОМ_НАК и КОД_СК.
8. В бланк запроса включите единственное обновляемое в таблице НАКЛАДНАЯ поле СУММА_НАКЛ, в строке Обновление (Update To) введите имя поля Sum-СУММА_ОТПР из таблицы Стоимость накладной, из которого будет выбираться значение для обновления (рис. 4.46).
9. Сохраните запрос под именем Обновление стоимости накладной.

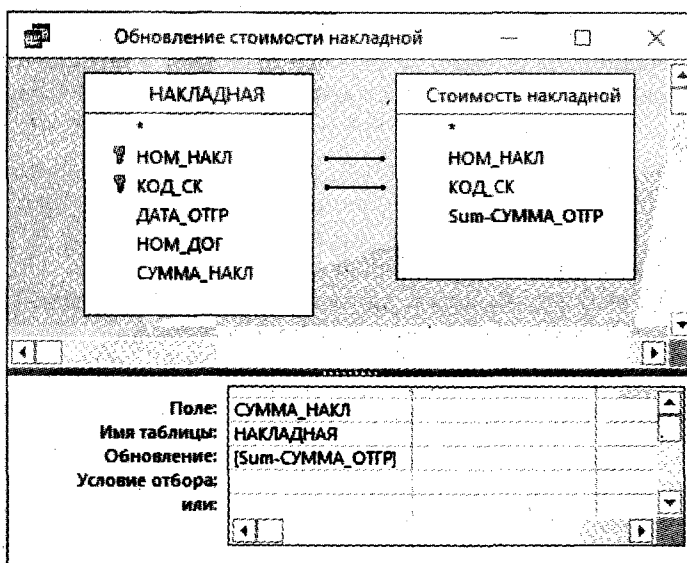


Рис. 4.46. Обновление в накладной поля с общей стоимостью товара

10. Для решения задачи вычисления общей стоимости накладной последовательно выполните сначала запрос на создание таблицы Расчет стоимости накладной, а затем запрос на Обновление стоимости накладной.

ЗАМЕЧАНИЕ

Если в таблице ОТГРУЗКА в поле СУММА_ОТПР не хранится значение стоимости каждого товара, для расчета общей стоимости товаров одной накладной дополните запрос Расчет стоимости накладной таблицей ТОВАР и создайте вычисляемое поле с выражением $[ЦЕНА] * [КОЛ_ОТПР]$. Выполните группировку по полям НОМ_НАКЛ и КОД_СК. Статистическую функцию Sum используйте для вычисляемого поля.

Для предварительного расчета стоимости каждого товара в таблице ОТГРУЗКА можно выполнить запрос Расчет стоимости (см. рис. 4.44) без условий отбора.

Запрос на добавление

С помощью запроса **Добавление** (Append) производится добавление записей в таблицу базы данных. Добавляемые записи формируются из одной или нескольких таблиц базы данных в запросе на выборку. При использовании нескольких таблиц их записи объединяются в соответствии с указанным способом объединения.

Структура записи таблицы, из которой добавляются записи, может не полностью совпадать со структурой записи дополняемой таблицы. Добавляемые записи обязательно должны включать значения ключевых полей. В добавляемых записях может быть меньше полей, если на поля в таблице, куда добавляются записи, не наложено требование по обязательности их заполнения. Допускается несоответствие типов полей, если возможно преобразование типа данных добавляемого поля в тип данных поля, в которое добавляются значения. В бланк запроса могут быть включены поля для задания условий отбора добавляемых записей. В качестве значений полей добавляемых записей может быть использовано вычисляемое выражение.

Добавление данных в связанные таблицы

Задача

В рассматриваемой базе данных «Поставка товаров» учет накладных ведется в таблицах **НАКЛАДНАЯ** и **ОТГРУЗКА**. Предположим, что это централизованная база данных предприятия. На складах предприятия имеются свои базы данных, в которых ведется учет накладных, по которым отгружаются товары в течение дня. Данные этих накладных сохраняются в двух таблицах, которые имеют имена **Накладные склада** и **Отгрузка склада** и такие же ключевые поля, как и аналогичные таблицы централизованной базы. В конце каждого дня в централизованную базу данных импортируются таблицы складов. После чего с помощью запросов на добавление накладные, оформленные на складе, включаются в централизованную базу данных. Сначала выполняется добавление записей в главную таблицу **НАКЛАДНАЯ**. Только после этого могут быть добавлены связанные записи в подчиненную таблицу **ОТГРУЗКА**. Это обусловлено тем, что в схеме данных базы для связи таблиц **НАКЛАДНАЯ** и **ОТГРУЗКА** установлен параметр **Обеспечение целостности данных** (Enforce Referential Integrity), делающий возможным добавление подчиненных записей только в том случае, если в главной таблице уже имеется связанная запись.

Рассмотрим создание запроса на добавление в таблицы централизованной базы данных накладных, оформленных на складе и уже импортированных в нее.

1. Создайте запрос на основе таблицы **Накладные склада**, из которой будут добавляться записи. Для этого выполните команду **Конструктор запросов** (Query Design) на ленте **Создание** (Create) в группе **Запросы** (Queries). По умолчанию будет создан запрос на выборку, что подтверждается выделенным типом запроса — **Выборка** (Select) — на ленте **Работа с запросами** | **Конструктор** (Query Tools | Design).
2. Превратите запрос на выборку в запрос на добавление, выбрав тип запроса **Добавление** (Append). При этом открывается диалоговое окно **Добавление** (Append To) (рис. 4.47).

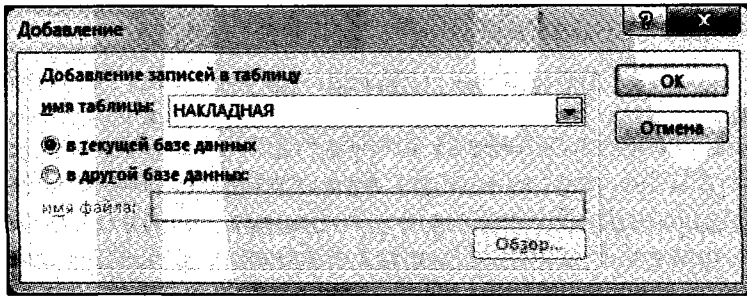


Рис. 4.47. Окно для ввода имени таблицы, в которую добавляются записи

3. В окне **Добавление** в поле **имя таблицы** (Table Name) введите или выберите из списка имя таблицы-получателя **НАКЛАДНАЯ**. В эту таблицу будут добавляться записи.
4. Так как таблица — получатель добавляемых записей — находится в открытой базе данных, отметьте переключатель **в текущей базе данных** (Current Database). Для таблицы, находящейся в другой базе данных, надо отметить переключатель **в другой базе данных** (Another Database) и ввести имя файла базы данных. При необходимости надо указать путь к другой базе данных. Помимо базы данных Access, можно указать базу данных Microsoft FoxPro, Paradox, dBASE или ввести строку подключения для базы данных SQL Server.

ЗАМЕЧАНИЕ

При выборе в качестве таблицы-получателя другой базы данных возможно добавление записей в централизованную базу без предварительного импорта в нее таблиц склада.

5. После преобразования запроса в его бланке появляется строка **Добавление** (Append To) (рис. 4.48).
6. Для формирования добавляемых в таблицу **НАКЛАДНАЯ** записей обязательно включите в бланк запроса из таблицы **Накладные склада** поля **НОМ_НАК** и **КОД_СК**, составляющие ключ, и поле связи **НОМ_ДОГ**. Включите также поля **ДАТА_ОТПР** и **СУММА_НАКЛ**.
7. Поскольку в этом примере имена полей обеих таблиц совпадают, Access автоматически сформирует в строке **Добавление** (Append To) имена полей таблицы **НАКЛАДНАЯ**, в которые будут добавляться значения из соответствующих полей таблицы **Накладные склада**. Если эти имена не совпадают, откройте в каждой ячейке список полей и выберите нужное имя.
8. Для предварительного просмотра записей, которые планируется добавить в таблицу **НАКЛАДНАЯ**, нажмите кнопку **Режим** (View) на вкладке ленты **Конструктор** (Design) или **Режим таблицы** (Datasheet View) в строке состояния. Возврат в режим конструктора запросов производится аналогично.
9. Для добавления записей выполните запрос, нажав кнопку **Выполнить** (Run) на вкладке ленты. При этом откроется диалоговое окно с сообщением о числе добавляемых записей и вопросом о продолжении операции добавления.

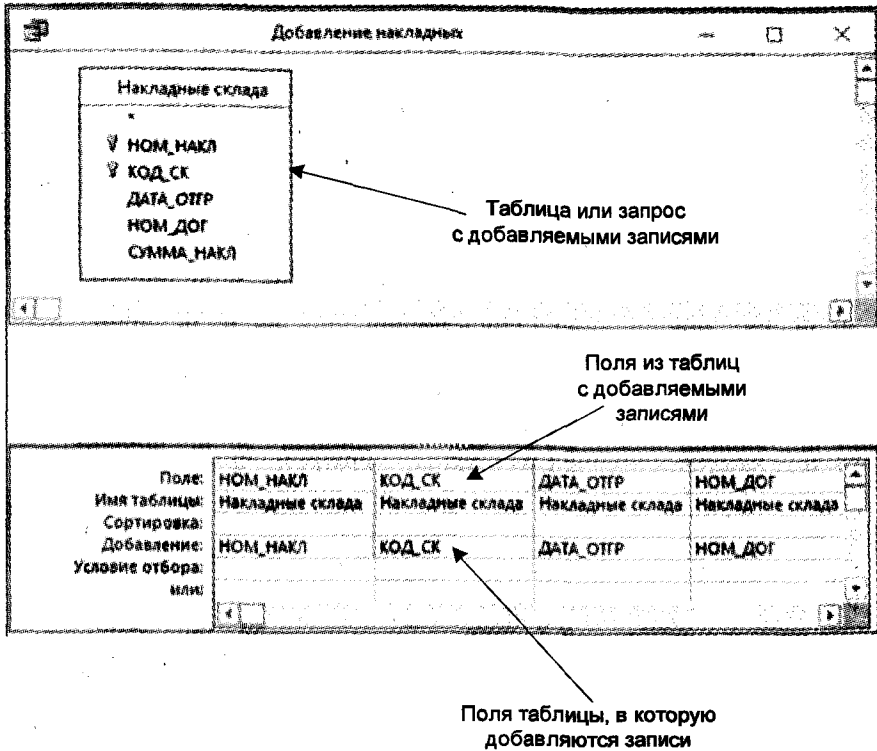


Рис. 4.48. Запрос на добавление записей

ЗАМЕЧАНИЕ

Если таблица, в которую добавляются записи, содержит ключевое поле, добавляемые записи должны содержать такое же поле. Те записи, добавление которых приведет к появлению совпадающих или пустых значений в ключевом поле, не будут добавлены. Записи не добавляются и в случае, если невозможно преобразование типа данных в добавляемых полях или не выполняются условия на значения.

10. Перейдите в режим SQL. Эквивалентная запросу на добавление инструкция INSERT INTO имеет вид:

```
INSERT INTO НАКЛАДНАЯ (НОМ_НАКЛ, КОД_СК, ДАТА_ОТПР, НОМ_ДОГ, СУММА_НАКЛ)
SELECT [Накладные склада].НОМ_НАКЛ, [Накладные склада].КОД_СК,
      [Накладные склада].ДАТА, [Накладные склада].НОМ_ДОГ,
      [Накладные склада].СУММА_НАКЛ
FROM [Накладные склада];
```

Инструкция INSERT INTO определяет поля таблицы НАКЛАДНАЯ, которые должны заполниться значениями из полей добавляемых записей. Поскольку в запросе в качестве источника добавляемых записей указана таблица Накладные склада, конструктор создал на ее основе запрос на выборку, который определяет поля добавляемых записей. Этот запрос представлен инструкцией SELECT. Список имен полей и порядок их перечисления для обеих таблиц должен совпадать.

Задание 4.7. Создайте запрос на добавление записей в таблицу ОТГРУЗКА из таблицы Отгрузка склада

Учтите при этом, что добавлять записи в подчиненную таблицу ОТГРУЗКА из таблицы Отгрузка склада можно только после дополнения таблицы НАКЛАДНАЯ записями из таблицы Накладные склада.

Добавление данных из нескольких таблиц

Задача

Необходимо создать новую накладную на отгрузку товара покупателю. Пусть товары по накладным отгружаются один раз в месяц в полном соответствии с планом поставок. Это означает, что по каждому договору выписывается только одна накладная в месяц, а спецификация накладной должна полностью совпадать со спецификацией договора на заданный месяц. Прежде чем заполнять спецификацию накладной, создайте в таблице НАКЛАДНАЯ новую запись, заполнив поля **Номер накладной**, **Номер склада**, **Номер договора** и **Дата отгрузки**. Только после этого можно приступить к добавлению связанных записей в таблицу ОТГРУЗКА, сформировав эти записи на основании записей, выбранных из таблицы ПОСТАВКА_ПЛАН.

1. Создайте запрос на добавление записей в таблицу ОТГРУЗКА на основе таблиц НАКЛАДНАЯ и ПОСТАВКА_ПЛАН. Установите между ними связь по полю ном_дог. Таблица НАКЛАДНАЯ включена в запрос для того, чтобы сформировать в добавляемых записях ключевое поле, включающее ном_накл и код_ск.
2. В запрос включите поля ном_накл, код_ск, код_тов, кол_пост, сумма_пост, которые составят запись, добавляемую в таблицу ОТГРУЗКА, а также срок_пост, ном_дог и вычисляемое поле Month(ДАТА_ОТПР) для задания условий отбора добавляемых записей (рис. 4.49).

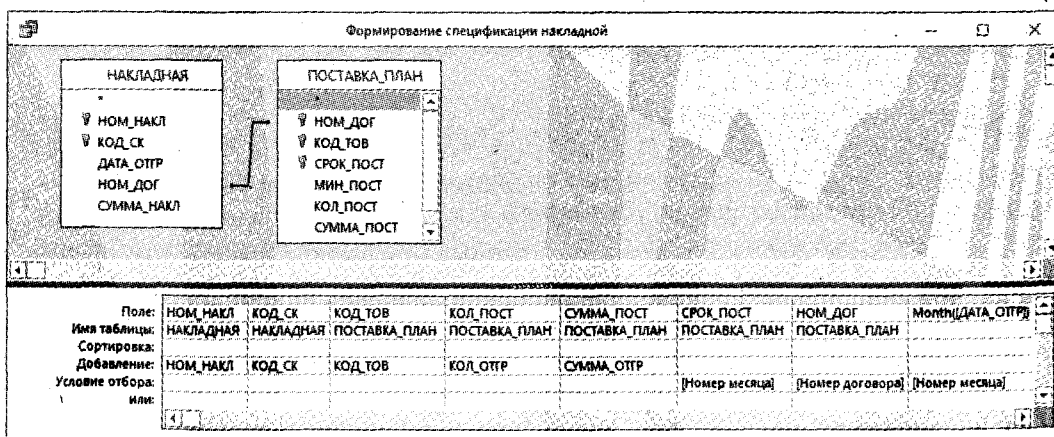


Рис. 4.49. Добавление в таблицу ОТГРУЗКА записей, соответствующих заданному договору и месяцу

3. Задайте параметры запроса, которые позволят ввести номер месяца и номер договора для выборки формируемой накладной. По значениям этих параметров из таблицы ПОСТАВКА_ПЛАН должны быть выбраны все записи, необходимые для добавления в таблицу ОТГРУЗКА.
4. В каждом месяце по договору выписывается только одна накладная, поэтому по введенным номеру договора и номеру месяца из таблицы НАКЛАДНАЯ будет выбрана только одна запись — та, которая была добавлена в таблицу при формировании новой накладной. При объединении записей таблиц запроса ключевые поля записи из таблицы НАКЛАДНАЯ будут добавлены ко всем записям, выбранным из таблицы ПОСТАВКА_ПЛАН. В результате записи обретут составной ключ — НОМ_НАКЛ, КОД_СК, КОД_ТОВ — необходимый для их добавления в таблицу ОТГРУЗКА.
5. Для предварительного просмотра записей, которые будут добавлены в таблицу, нажмите на вкладке ленты кнопку **Режим (View)**. Для добавления записей нажмите кнопку **Выполнить (Run)**.
6. Перейдите в режим SQL. Эквивалентная запросу на добавление инструкция INSERT INTO будет записана следующим образом:

```
INSERT INTO ОТГРУЗКА (НОМ_НАКЛ, КОД_СК, КОД_ТОВ, КОЛ_ОТГР, СУММА_ОТГР)
SELECT НАКЛАДНАЯ.НОМ_НАКЛ, НАКЛАДНАЯ.КОД_СК,
       ПОСТАВКА_ПЛАН.КОД_ТОВ, ПОСТАВКА_ПЛАН.КОЛ_ПОСТ,
       ПОСТАВКА_ПЛАН.СУММА_ПОСТ
FROM НАКЛАДНАЯ INNER JOIN ПОСТАВКА_ПЛАН ON
      НАКЛАДНАЯ.НОМ_ДОГ = ПОСТАВКА_ПЛАН.НОМ_ДОГ
WHERE ((ПОСТАВКА_ПЛАН.СРОК_ПОСТ)=[Номер месяца]) AND
       ((ПОСТАВКА_ПЛАН.НОМ_ДОГ)=[Номер договора]) AND
       (Month([ДАТА_ОТГР])=[Номер месяца]);
```

Инструкция INSERT INTO определяет поля записей, добавляемых в таблицу ОТГРУЗКА. Структура и количество добавляемых записей определяется числом записей, возвращаемых инструкцией выбора SELECT. Список имен полей и порядок их перечисления для таблицы ОТГРУЗКА, куда добавляются записи, и для таблицы, определяемой инструкцией SELECT, должен совпадать.

Конструирование запроса на удаление

Запрос на удаление (Delete Query) позволяет удалить записи из одной таблицы или из нескольких взаимосвязанных таблиц, для связи которых установлен параметр обеспечения целостности данных **Каскадное удаление связанных записей (Cascade Delete Related Records)**. В схему данных запроса включается таблица, из которой должны удаляться записи, и связанные с ней таблицы, для полей которых требуется задать условия отбора удаляемых записей.

Задача

Пусть необходимо из справочника удалить запись о покупателе с кодом П003. В схеме данных базы «Поставка товаров» для всех связей установлен параметр

обеспечения целостности **Каскадное удаление связанных записей** (Cascade Delete Related Records), поэтому удаление записи из таблицы **ПОКУПАТЕЛЬ** повлечет за собой удаление из базы всех подчиненных записей на всех уровнях иерархии. Автоматически будут удалены все договоры этого покупателя, а также все накладные, выписанные по этим договорам, т. е. будут удалены связанные записи из таблиц **ДОГОВОР**, **ПОСТАВКА_ПЛАН**, **НАКЛАДНАЯ** и **ОТГРУЗКА**.

1. Создайте запрос на выборку. Добавьте в схему данных запроса таблицу **ПОКУПАТЕЛЬ**, из которой удаляются записи.
2. Преобразуйте запрос на выборку в запрос на удаление, выбрав на вкладке ленты **Конструктор** (Design) тип запроса **Удаление** (Delete) или выполнив команду контекстного меню запроса **Тип запроса | Удаление** (Query Type | Delete Query).
3. После преобразования запроса в запрос на удаление в его бланке появляется строка **Удаление** (Delete) (рис. 4.50).



Рис. 4.50. Запрос на удаление записи в главной таблице

4. Далее приступайте к формированию бланка запроса. Переместите символ звездочки (*) из списка полей таблицы **ПОКУПАТЕЛЬ** в бланк запроса — в строке **Удаление** (Delete) в столбце этого поля появится значение **Из** (From).
5. Для того чтобы задать условия отбора удаляемых записей, переместите с помощью мыши в бланк запроса поле **КОД_ПОК** из таблицы **ПОКУПАТЕЛЬ** — в строке **Удаление** (Delete) под именем этого поля появится значение **Условие** (Where). Запишите в строку **Условие отбора** (Criteria) для этого поля значение **П003**.
6. Следует иметь в виду, что удаленные записи нельзя восстановить. Поэтому перед удалением записей выполните предварительный просмотр удаляемых записей, нажав кнопку **Режим** (View) на вкладке ленты.

7. Для удаления записей на вкладке ленты нажмите кнопку **Выполнить (Run)**. При выполнении запроса будет сообщено об удалении одной записи. Откройте таблицу **ПОКУПАТЕЛЬ** и подчиненные ей таблицы на всех уровнях и убедитесь, что в них также удалены записи, связанные с этим покупателем.

Результаты работы запроса на удаление зависят от отношений между таблицами и установленных в схеме базы данных параметров обеспечения целостности для их связей.

Если параметры обеспечения целостности для связей таблицы не установлены вообще, то записи удаляются только в указанной в бланке запроса таблице и вне зависимости от ее логических связей.

Если в запросе объединяются две или более таблицы, находящиеся в отношении 1 : M, то удалить можно только записи одной из таблиц на стороне «многие».

Если между таблицами, находящимися в отношении 1 : M, установлена связь с поддержанием целостности, но не задан параметр **Каскадное удаление связанных записей (Cascade Delete Related Records)**, сначала необходимо удалить записи подчиненной таблицы и только после этого можно выполнить удаление записей в главной таблице. То есть, удаление записи главной таблицы возможно только, если в подчиненной таблице нет связанных записей. Если параметр **Каскадное удаление связанных записей (Cascade Delete Related Records)** задан, то для удаления записей главной таблицы и связанных с ними подчиненных записей достаточно указать в запросе удаление записей главной таблицы.

8. Откройте запрос на удаление, представленный на рис. 4.50, в режиме конструктора и перейдите в режим SQL. Эквивалентная этому запросу инструкция **DELETE** будет записана следующим образом:

```
DELETE ПОКУПАТЕЛЬ.*, ПОКУПАТЕЛЬ.КОД_ПОК  
FROM ПОКУПАТЕЛЬ  
WHERE ((ПОКУПАТЕЛЬ.КОД_ПОК)="П003");
```

Инструкция **DELETE** удаляет из таблицы записи, удовлетворяющие условию отбора, заданному в предложении **WHERE**. Если в предложении **FROM** объединяются две или более таблицы, то удалить можно только записи одной из таблиц. Для того чтобы указать, из какой именно таблицы должны быть удалены записи, в список полей включается конструкция **имя_таблицы.*** (например, **ПОСТАВКА_ПЛАН.***). Для однотабличного запроса эта конструкция может быть опущена.

Контрольные вопросы

к разд. «Запросы на изменение»

1. Можно ли запросы на изменение создать с помощью мастера?
2. С создания запроса какого типа начинается конструирование запроса на изменение?
3. Как просмотреть в запросе на обновление записи, которые будут изменены в таблице?

4. Определен ли в структуре таблицы, созданной запросом, ключ?
5. Какой командой осуществляется преобразование запроса на выборку в запрос на обновление?
6. В какой строке запроса на обновление указывается новое значение обновляемого поля?
7. Можно ли для подсчета нового значения обновляемого поля использовать выражение?
8. Какая инструкция языка SQL соответствует запросу на создание таблицы?
9. Можно ли в новую таблицу, создаваемую запросом, включать поля нескольких таблиц?
10. Какая инструкция языка SQL соответствует запросу на обновление?
11. Можно ли просмотреть содержимое обновляемых полей до и после обновления, не выходя из режима конструктора запросов?
12. В какой строке запроса на добавление указывается, откуда должны выбираться значения полей добавляемых записей?
13. Где указывается имя таблицы, в которую добавляются записи?
14. В какой строке запроса на добавление указывается, в какие поля должны попадать поля добавляемой записи?
15. Какая инструкция языка SQL соответствует запросу на добавление?
16. Вслед за каким словом в инструкции INSERT записывается имя таблицы, в которую добавляются записи?
17. С помощью какого предложения формируются добавляемые записи в инструкции INSERT?
18. Могут ли добавляемые записи формироваться на основе нескольких таблиц?
19. Можно ли одним запросом удалить записи из нескольких таблиц?
20. Если в запросе на удаление использованы главная таблица и две подчиненные, находящиеся с главной в отношении 1 : М, из какой таблицы могут удаляться записи?
21. Каким образом в запросе на удаление указывается таблица, из которой удаляются записи?
22. Какая инструкция языка SQL соответствует запросу на удаление?
23. Если в предложении FROM инструкции DELETE указана одна таблица, нужно ли использовать в списке полей конструкцию имя_таблицы.*?
24. Какие поля включаются в список полей запроса на удаление?

Ответы

1. Нет.
2. Запроса на выборку.

3. Нажав в режиме конструктора запросов кнопку **Режим (View)**.
4. Нет.
5. Командой **Обновление (Update Query)** на ленте конструктора запросов или из списка контекстного меню запроса **Тип запроса (Query Type)**.
6. **Обновление (Update To)**.
7. Да.
8. `SELECT ... INTO.`
9. Да.
10. `UPDATE.`
11. Да.
12. В строке **Поле (Field)** указываются имена полей, а в строке **Имя таблицы (Table)** — имена таблиц, из которых выбираются данные для добавления.
13. В диалоговом окне **Добавление (Append)**, открываемом при преобразовании запроса на выборку в запрос на добавление.
14. **Добавление (Append To)**.
15. `INSERT INTO.`
16. `INTO.`
17. С помощью инструкции выбора `SELECT`.
18. Да, как в любом запросе на выборку.
19. Нет. Исключением является автоматическое удаление наряду с записью главной таблицы связанных записей подчиненных таблиц, для которых установлен флажок **каскадное удаление связанных записей (Cascade Delete Related Records)**.
20. Из любой, но только одной подчиненной таблицы.
21. Размещением в строке **Поле (Field)** бланка запроса конструкции `имя_таблицы.*` и в строке **Удаление (Delete)** значения **Из (From)**.
22. `DELETE.`
23. Нет.
24. Все поля таблицы, из которой удаляются записи, и, если необходимо, поля, для которых формулируются условия отбора.

ГЛАВА 5



Формы

Формы являются основой разработки диалоговых приложений пользователя для работы с базой данных. Формы, адекватные формам первичных документов, позволяют выполнять загрузку справочных, плановых и оперативно-учетных данных, в любой момент просматривать и редактировать содержимое ранее введенных в базу данных документов, оформлять новые документы.

ЗАМЕЧАНИЕ

Вопросы создания графического диалогового интерфейса пользователя для работы с документами, сохраняемыми в базе данных, рассматриваются в *главе 6*.

Формы обеспечивают удобную работу с данными одной или нескольких взаимосвязанных таблиц. Данные из таблиц выводятся на экран с использованием макета формы, разработанного пользователем. Работая с формой, пользователь может добавлять, удалять и изменять записи таблиц, получать расчетные данные. В процессе работы может осуществляться контроль вводимых данных, могут проверяться ограничения на доступ к данным, выводиться необходимые дополнительные сведения.

Форма состоит из элементов управления, которые отображают поля таблиц и графические элементы, не связанные с полями таблиц. Элементы управления предназначены, прежде всего, для разработки макета формы: размещения полей таблиц и запросов, надписей, внедряемых объектов (рисунков, диаграмм), вычисляемых полей, а также кнопок, выполняющих печать и открывающих другие объекты или задачи.

Как форма в целом, так и каждый из ее элементов обладают множеством свойств. Посредством их изменения можно настроить внешний вид, размер, местоположение элементов в форме, определить источник данных формы, режим ввода/вывода, привязать к элементу выражение, макрос или программу. Набор свойств доступен в соответствующем окне, где они разбиты на категории, каждая из которых представлена на своей вкладке. Основными вкладками в окне свойств являются:

Макет — представляет свойства, ориентированные на определение внешнего вида формы или ее элементов;

- **Данные** — представляет свойства для определения источника данных формы или ее элементов, режима использования формы (только ввод, разрешение на изменение, добавление, удаление и т. п.);
- **События** — представляет события, возникающие при работе с конкретным объектом или элементом: нажатие кнопки мыши, перевод фокуса на элемент управления, изменение данных в элементе управления (до обновления, после обновления), открытие или закрытие формы и т. д. Они могут быть инициированы пользователем или системой. Одно действие может запустить несколько разных событий. С событием могут связываться макрос или процедура обработки события на языке VBA, выполняющие некоторые действия или рассчитывающие значения. С их помощью можно организовать открытие связанной формы, обновление данных таблицы расчетными значениями, печать формы, вывод отчета. Запрограммировав вызов различных объектов базы данных, можно автоматизировать выполнение задач приложения.

Для быстрого создания формы предназначены мастера Access. Однако точное формирование макета формы, отвечающего заданным требованиям, дополнение его процедурами обработки событий, возникающих в форме, обеспечивается средствами конструирования. Конструктор форм можно использовать как для создания новой формы, так и для редактирования формы, созданной мастером. Для удобной настройки внешнего вида формы Access предоставляет режим макета.

В процессе создания формы определяется источник данных формы (таблицы или/и запросы). Из источника данных выбираются поля, которые должны быть представлены в форме, осуществляется их размещение в форме, создаются вычисляемые поля. Для оформления формы создаются графические элементы: кнопки, выключатели, элементы, поясняющий текст и рисунки. Для настройки различных элементов формы используется типовой набор их свойств.

Формы в Access могут быть представлены в четырех режимах.

- **Режим формы** предназначен для ввода, просмотра и корректировки данных таблиц, на которых основана форма.
- **Режим макета** обеспечивает просмотр данных почти в таком виде, в каком они отображаются в режиме формы, и в то же время позволяет изменять внешний вид формы. В этом режиме элементы формы становятся выделяемыми, их можно перетаскивать в другие места, редактировать содержимое надписей полей, изменять формат, размер и т. п. Режим макета позволяет удобно настраивать внешний вид формы и может использоваться для внесения большинства структурных изменений. В Access 2010/2016 появилась возможность в режиме макета выполнять действия, ранее доступные только в режиме конструктора. В режиме макета доступна лента **Конструктор (Design)**. И все-таки, если некоторую задачу невозможно выполнить в режиме макета, следует переключиться в режим конструктора. В ряде случаев Access отображает сообщение о том, что для внесения изменений надо переключиться в режим конструктора.
- **Режим конструктора** предназначен для разработки формы с помощью полного набора инструментов, обеспечивающего более детальную проработку структуры

формы, использование всех элементов управления. В этом режиме форму можно разработать с нуля или доработать ее после создания мастером. Просмотр данных при внесении изменений в этом режиме не предусматривается.

- **Режим таблицы** обеспечивает просмотр данных, выбранных в форме из источника данных в табличном виде. Если форма содержит подчиненную форму, в таблице, соответствующей главной форме, могут быть отображены соответствующие подчиненные записи.

Однотабличные формы

Однотабличная форма предназначена для загрузки, просмотра и корректировки данных одной таблицы. Источником данных такой формы служит единственная таблица. Однотабличная форма создается одним щелчком мыши с помощью команд автоматического создания формы: **Форма (Form)**, **Разделенная форма (Split Form)** или **Несколько элементов (Multiple Items)**, размещенных на вкладке ленты **Создание (Create)** в группе **Формы (Forms)** (рис. 5.1). Для последующей настройки формы в соответствии с требованиями пользователя ее можно доработать в режиме макета или конструктора. Можно удалить из формы ненужные поля, изменить расположение элементов управления и подобрать их размеры, добавить новые элементы управления, произвести вычисления, задать свойства формы и ее элементов управления.

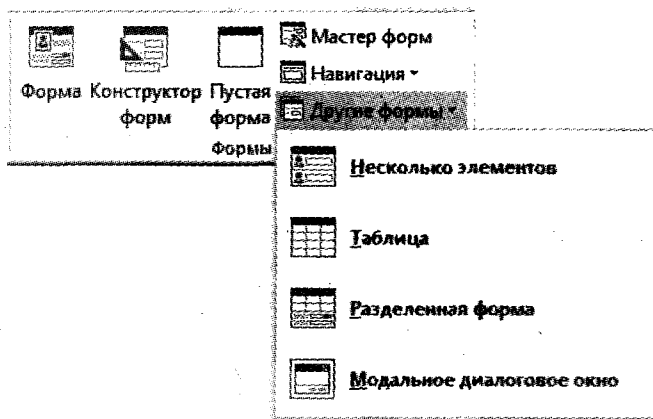


Рис. 5.1. Команды группы **Формы** на вкладке ленты **Создание**

Формы, созданные с помощью указанных команд, отличаются, прежде всего, способом отображения записей базового источника данных:

- в форме, созданной с помощью команды **Форма (Form)**, одновременно отображается только одна запись, поэтому ее целесообразно использовать, например, в случае, когда таблица включает поля с данными, требующими большого окна для отображения (длинный текст, диаграмма, фотография). Поля отображаемой записи располагаются внутри одного раздела формы в столбик (макет в столбик) с надписью слева от каждого поля;

- в форме, созданной с помощью команды **Несколько элементов** (Multiple Items), отображается сразу несколько записей, и все поля записи размещаются в одной строке, что удобно для ввода данных из простых справочников, имеющих только табличную часть. Этот макет формы аналогичен отображению записей таблицы в режиме таблицы, однако при этом предоставляются многочисленные возможности по настройке отображения и дополнения новыми элементами. Например, в полях такой формы отображаются рисунки;
- в форме, созданной с помощью команды **Разделенная форма** (Split Form), одновременно отображаются данные в двух представлениях: в одном ее разделе записи отображаются в виде таблицы, в другом — выводится единственная выделенная в таблице запись, предназначенная для удобной работы с ее данными.

Общим для этих команд является то, что они автоматически создают для выбранной таблицы форму, не вступая в диалог с пользователем, и сразу выводят на экран форму в режиме макета.

Создать однотоабличную форму можно с помощью команды **Мастер форм** (Form Wizard), размещенной на вкладке ленты **Создание** (Create) в группе **Формы** (Forms). В диалоговых окнах мастера пользователь выбирает таблицу — источник записей формы, поля, которые надо включать в форму, способ отображения записей, стиль оформления.

Создание однотоабличной формы

Создайте однотоабличную форму СПРАВОЧНИК ТОВАРОВ для ввода, просмотра и корректировки данных таблицы ТОВАР в базе данных «Поставка товаров». Поскольку в таблице есть поле с фотографией, требующее большой области для отображения, одновременно выводите в форме поля только одной записи таблицы.

Чтобы источником записей формы стала таблица ТОВАР, выберите ее в области навигации и выполните команду **Форма** (Form) на вкладке ленты **Создание** (Create). Эта команда обеспечит автоматическое создание формы на основе только одной выбранной таблицы.

ЗАМЕЧАНИЕ

Если таблица имеет только одну подчиненную таблицу, с которой она находится в отношении 1 : М, то на основе подчиненной таблицы автоматически создается внедренный элемент, отображающий записи подчиненной таблицы, связанные с текущей записью главной. Если необходимо построить однотоабличную форму, следует удалить связи. Таблице ТОВАР в схеме данных подчинено две таблицы, поэтому сразу создается однотоабличная форма.

Завершается работа команды отображением формы в режиме макета (рис. 5.2). При этом в форме отображаются поля одной записи таблицы, расположенные в макете *в столбик*. Заголовок формы соответствует имени таблицы источника ТОВАР. Скрытые в таблице поля, например, командой **Скрыть поля** (Hide Fields) из контекстного меню выделенного столбца, не включаются в форму.

Код товара	T001	Наличие товара	<input checked="" type="checkbox"/>
Наименование товара	Монитор 17LG	Вложения	
Цена	7 000,00р.	Цена с НДС	7 350,00р.
Единица измерения	штука	Остаток	420
Ставка НДС	5%	Стоимость	2 940 000,00р.
Фото товара			

Рис. 5.2. Форма, полученная по команде Форма для работы с таблицей ТОВАР

Редактирование формы в режиме макета

В режиме макета можно просматривать данные практически так же, как в режиме формы, и в то же время вносить изменения. Это средство функционирует по принципу WYSIWYG (что видим, то и получаем) и позволяет вносить изменения в форму и тут же видеть результат внесенных изменений, что очень удобно для уточнения местоположения, размера, шрифта элементов управления, изменения текста надписей и выполнения других задач, связанных с внешним видом и удобством формы. При отображении формы в режиме макета появляются вкладки ленты инструментов **Работа с макетами форм | Конструктор (Form Layout Tools | Design)**, **Упорядочить (Arrange)** и **Формат (Format)** (рис. 5.3–5.6), которые сохраняются на экране, пока активно окно формы и не выполнено переключение в другой режим.

При уменьшении размеров окна Access сворачиваются группы ленты. При этом команды группы можно открыть с помощью значка списка.

1. Для открытия формы ТОВАР в режиме макета выберите ее в области навигации и в контекстном меню щелкните на строке **Режим макета (Layout View)**. Если форма открыта, и необходимо переключиться в режим макета, щелкните на соответствующем значке строки состояния или выполните команду **Режим макета** из списка кнопки **Режим (View)** на ленте **Конструктор (Design)** или **Главная (Home)**.
2. Измените заголовок формы на **СПРАВОЧНИК ТОВАРОВ**. Для перехода к редактированию заголовка щелкните на значке **Заголовок (Title)** в группе **Колоннотитулы** на ленте **Конструктор (Design)** или просто сделайте двойной щелчок на заголовке.

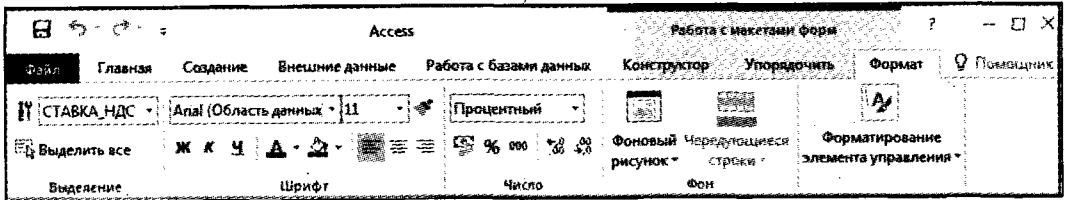


Рис. 5.3. Вкладка ленты Работа с макетами форм | Формат

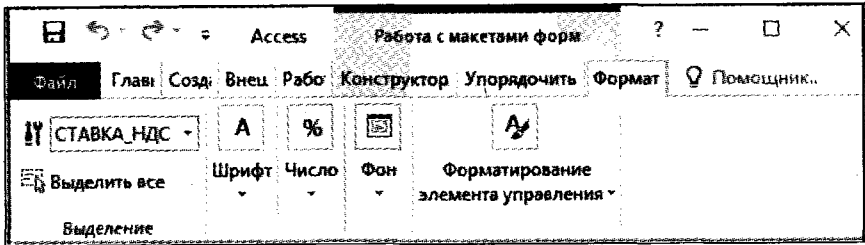


Рис. 5.4. Свернутые группы вкладки ленты Формат

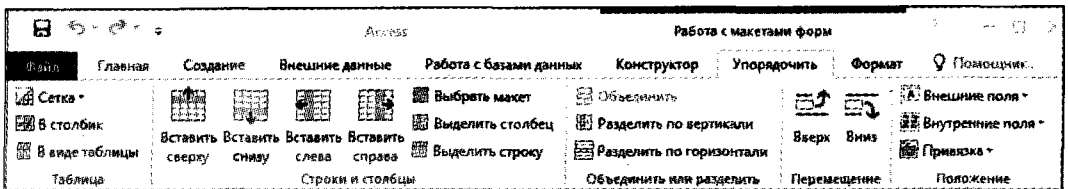


Рис. 5.5. Вкладка ленты Работа с макетами форм | Упорядочить

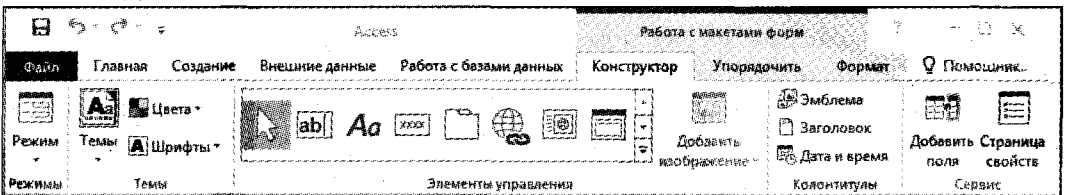


Рис. 5.6. Вкладка ленты Работа с макетами форм | Конструктор

3. Замените эмблему, которая автоматически вставилась в форму при ее создании. Для этого щелкните на значке Эмблема (Logo) и выберите свой рисунок.
4. Вставьте в форму дату. Для этого щелкните на кнопке Дата и время (Date and Time) на ленте Конструктор (Design) в группе Колонтитулы (Header/Footer). В открывшемся окне Дата и время выберите формат даты, уберите флажок Формат времени (Include Time) и нажмите кнопку ОК.
5. Access включает множество разнообразных тем, которые можно применить к форме для придания ей нужного стиля. В режиме макета на ленте Конструктор (Design) представлена коллекция тем. Выберите понравившуюся вам тему и щелкните на ней, чтобы применить ее к форме.

Макеты элементов управления

Макет элементов управления — это объединение элементов управления в группу, для которой произведено выравнивание по вертикали и горизонтали для единообразного оформления формы. В форме может быть несколько макетов элементов управления.

Макет можно рассматривать как таблицу, в столбцах которой размещены элементы управления. Естественно в таблице можно менять только ширину всего столбца или всей строки, а не отдельных ее ячеек. В макете, как и в таблице, можно перемещать значение из одной ячейки в другую.

В форме ТОВАР системой создан макет (см. рис. 5.2), в котором поля и их подписи размещены в четырех столбцах. Подписи полей в форме соответствуют заданным в соответствующих свойствах при определении структуры.

Выделите макет формы, щелкнув на звездочке в левом верхнем углу макета или выполнив команду **Выбрать макет** (Select layout) в группе **Строки и столбцы** (Rows & Columns) на ленте **Упорядочить** (Arrange). Команды доступны только при размещении курсора на одном из элементов макета. Затем выполните команду **Сетка** (Gridlines) в группе **Таблица** (Table) на ленте **Упорядочить**. Выберите в списке команды опцию **Все** (Both), что означает показать все линии таблицы, а также цвет, тип границы и ширину линий такими, чтобы была отчетливо видна таблица, в которую заключены элементы управления формы.

В рамках одного макета невозможно изменение ширины и высоты отдельных элементов — местоположение элемента может меняться только в пределах макета. В то же время макет при перетаскивании элементов для изменения их местоположения может легко расширяться путем добавления новых строк и столбцов.

Элемент можно удалить из макета командой контекстного меню **Макет | Удалить макет**, и тогда будет обеспечено его свободное перемещение и изменение размеров. Выделив весь макет и выполнив ту же команду, можно полностью разгруппировать элементы управления. Выделив нужную группу элементов, можно создать новый макет в столбик или табличный соответствующими командами на ленте **Упорядочить** (Arrange) в группе **Таблица** (Table). Именно макет позволит вам выравнивать все его элементы.

1. Измените неоправданно большую ширину полей формы. Для этого щелкните на любом из них и перетащите его правую границу. Изменится ширина всех полей в столбце формы. Это объясняется тем, что при создании формы с помощью команды **Форма** (Form) автоматически создается макет в столбик.
2. Удалите поле с фотографией из макета формы командой контекстного меню **Макет | Удалить макет** (Layout | Remove layout), а подпись поля удалите совсем, выделив ее и нажав клавишу <Delete>.
3. Выделите оставшиеся в двух правых столбцах поля с их подписями и переместите под последний элемент первого столбца. Макет имеет способность расширяться, и поэтому перемещенные элементы дополняют два первых столбца

макета. При этом из макета удалятся два правых столбца, в которых не останутся элементов.

4. Выделяя широкие строки полей с кодом товара, ценой и вложениями, уменьшите их высоту.
5. Измените местоположение и размер фотографии в форме, а подпись этого поля удалите. Чтобы убрать рамку вокруг фотографии, выделите поле и на вкладке ленты **Формат (Format)** в группе **Форматирование элемента управления (Control Formatting)** в списке команды **Контур фигуры (Shape Outline)** выберите **Прозрачный (Transparent)**.
6. Подпись поля с вложениями измените на Сертификат качества.

Условное форматирование элементов управления

Для изменения внешнего вида элемента управления в форме в зависимости от одного или нескольких условий используйте условное форматирование. Измените в форме ТОВАР цвет денежных значений в поле ЦЕНА на красный, когда они оказываются выше заданной величины, например, 2000 руб.

1. Выберите поле ЦЕНА, в котором нужно изменить цвет значения при заданном условии. На вкладке ленты **Формат (Format)** в группе **Форматирование элементов управления (Control Formatting)** выберите команду **Условное форматирование (Conditional Formatting)** — откроется диалоговое окно **Диспетчер правил условного форматирования (Conditional Formatting Rules Manager)** (рис. 5.7).

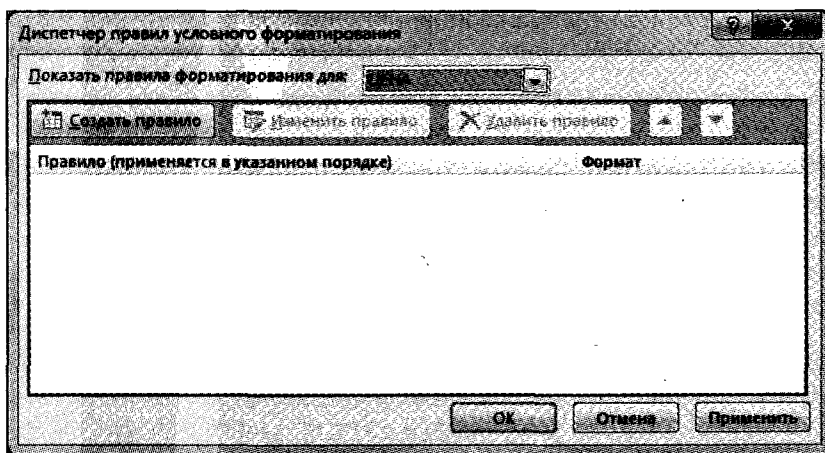


Рис. 5.7. Окно с пустым списком правил форматирования

2. Чтобы создать первое правило условного форматирования, щелкните на соответствующей кнопке.
3. В окне **Новое правило форматирования (New Rule)** (рис. 5.8) выберите тип правила и сформируйте описания правил, согласно которым будет производиться заданное форматирование поля.

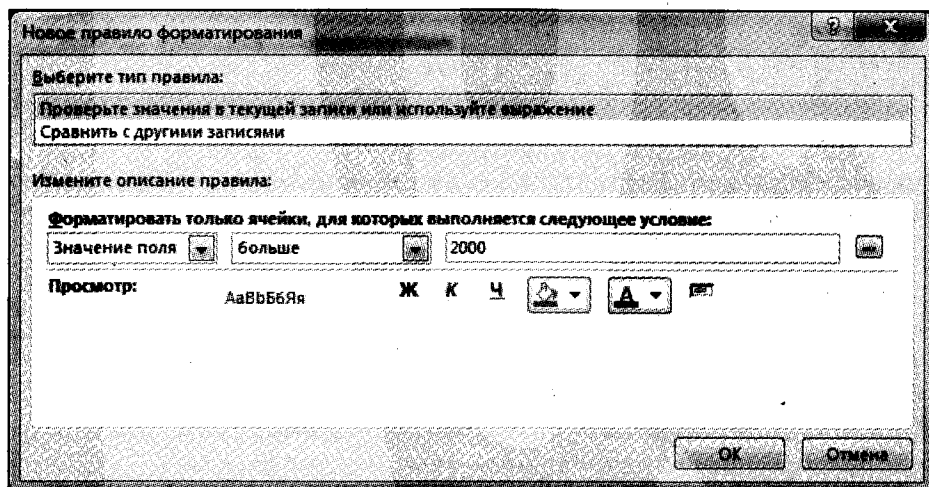


Рис. 5.8. Окно определения условий форматирования и форматов поля

4. В первом поле со списком выберите пункт **Значение поля** (Field Value Is), во втором выберите тип сравнения: **больше** (greater than) и введите постоянное значение в третье поле — 2000, не используя знака денежных единиц.
5. Выберите начертание шрифта, цвет и другие параметры форматирования, используемые при выполнении условия. Нажмите кнопку **ОК**. Выбранное форматирование применяется только в тех случаях, когда значение элемента управления удовлетворяет условию.
6. Для каждого элемента управления можно указать несколько условий. Чтобы добавить новое условие или удалить существующее, нажмите соответствующие кнопки в окне **Диспетчера правил условного форматирования** (Conditional Formatting Rules Manager).

После выполненных изменений макета форма примет вид, показанный на рис. 5.9. Сейчас здесь выделен один макет в столбик. Выделение макета выполняется щелчком в его левом верхнем углу.

Свойства формы

Редактирование формы, ее разделов и входящих в нее элементов управления может быть произведено не только графическими средствами, но и путем изменения их свойств. Для изменения свойств предназначена **Страница свойств** (Property Sheet).

1. Откройте страницу свойств, выбрав соответствующую кнопку на вкладке ленты **Конструктор** (Design) в группе **Сервис** (Tools). В окне свойств (рис. 5.10) в списке поля **Тип выделенного элемента** (Selection type) отображаются все элементы формы.
2. Выберите в списке строку **Форма** (Form). Чтобы убедиться в действенности свойств элементов, в открывшихся свойствах формы на вкладке **Макет** (Format)



Рис. 5.9. Форма ТОВАР в режиме макета после доработки

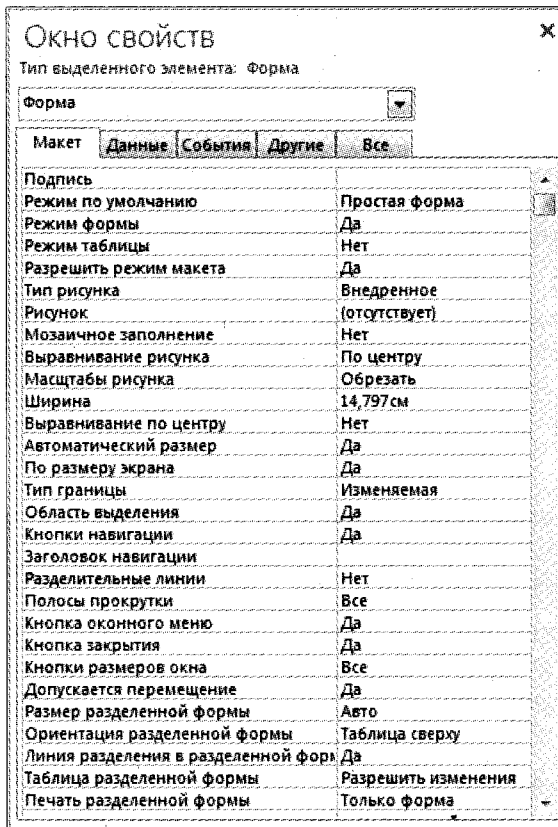


Рис. 5.10. Окно свойств формы

в строке **Область выделения (Record Selectors)** выберите значение **Нет**. Область выделения записи, расположенная слева в области данных, исчезнет из формы.

3. Верните область выделения в форму.

Следует заметить, что изменение многих свойств формы и ее элементов управления невозможно в режиме макета.


Добавление полей в форму

В режиме макета возможно удаление полей из формы и добавление их в нее. Для удаления поля достаточно выделить его вместе с наименованием и в контекстном меню выполнить команду **Удалить (Delete)**. Предварительно можно удалить поле из макета. Для добавления полей в форму щелкните на кнопке **Добавить поля (Add Existing Fields)** на вкладке ленты **Конструктор (Design)** в группе **Сервис (Tools)** — отобразится область **Список полей (Field List)** базовой таблицы или запроса, источника данных формы. Добавить поле в активный макет формы можно двойным щелчком на нем. Для размещения поля в любом макете формы перетащите его туда из области **Список полей (Field List)**.

В режиме макета в группе **Элементы управления (Controls)** на вкладке ленты **Конструктор** представлен набор элементов, для включения которых в форму нет необходимости переходить в режим конструктора. Этот набор содержит элементы для создания поля, надписи, списка, поля со списком, флажка, кнопки и т. п. Набор элементов управления режима макета в сравнении с доступным в режиме конструктора несколько усечен.

Работа с данными таблицы в режиме формы

Созданная форма **ТОВАР** предназначена для ввода, просмотра, корректировки и удаления данных, сохраняемых в одноименной таблице. Основным режимом работы с данными базовой таблицы является режим формы. В режиме макета предусмотрен только просмотр данных, обеспечивающий удобную настройку элементов.

1. Для перехода в режим формы из режима макета щелкните на кнопке **Режим формы (Form View)** в строке состояния или выполните команду **Режим формы** контекстного меню формы. Если форма закрыта, выберите ее в области навигации в группе **Формы (Forms)** и в контекстном меню нажмите кнопку **Открыть (Open)**.
2. Просмотрите записи таблицы **ТОВАР**, используя стандартные кнопки панели перехода по записям внизу формы или аналогичные команды в списке кнопки **Перейти (Go To)** на вкладке ленты **Главная (Home)** в группе **Найти (Find)**. Измените некоторые значения в полях записи.
3. Для создания новой записи в таблице **ТОВАР** сделайте текущей новую запись, воспользовавшись кнопкой  на панели перехода по записям или кнопкой **Создать (New)** на вкладке ленты **Главная (Home)** в группе **Записи (Records)**, и введите значения в поля формы. Значения, вводимые в поля формы, должны

соответствовать типам данных и их свойствам, заданным при определении структуры таблицы.

4. Для завершения создания (редактирования) записи таблицы достаточно перейти к другой записи или выполнить команду **Сохранить** (Save) на вкладке ленты **Главная** (Home) в группе **Записи** (Records). Запись может быть сохранена только при условии ввода значений ключевых полей таблицы, а также полей, которые были определены в свойствах поля как обязательные.
5. Чтобы удалить запись, выделите ее, щелкнув на области выделения в левой части области данных формы или выполнив команду **Выбрать | Выделить** (Select | Select) в группе **Найти** (Find), и выполните команду **Удалить | Удалить запись** (Delete | Delete Record) в группе **Записи** (Records).
6. Закройте форму. Откройте таблицу ТОВАР и убедитесь, что все сделанные изменения внесены в таблицу.

Задание 5.1. Создание однотобличной формы

Создайте однотобличную форму для работы со справочником покупателей, отредактируйте ее в режиме макета и произведите добавление записей в таблицу. Если в базе данных «Поставка товаров», выбрав таблицу **ПОКУПАТЕЛЬ**, для создания формы использовать команду **Форма** (Form), автоматически будет создана форма, содержащая встроенную подчиненную таблицу **ДОГОВОР**. Источником записей главной формы будет таблица **ПОКУПАТЕЛЬ**. Такое поведение команды **Форма** вызвано тем, что таблица **ПОКУПАТЕЛЬ** имеет единственную подчиненную таблицу **ДОГОВОР**, с которой она находится в отношениях 1 : М, и эта связь определена в схеме данных. В главной форме будет отображаться запись таблицы **ПОКУПАТЕЛЬ**, а в подчиненной — связанные с ней записи из таблицы **ДОГОВОР**.

Для создания формы, отображающей сведения только о покупателе, прежде чем воспользоваться командой **Форма** (Form), которая без вмешательства пользователя создаст однотобличную форму в столбец, следует в схеме данных удалить связь таблицы **ПОКУПАТЕЛЬ** с таблицей **ДОГОВОР**.

Кроме того, можно воспользоваться услугами мастера форм, который позволяет определить в качестве источника записей формы любое число таблиц, выбрать из них необходимые поля и способ их отображения в форме: в один столбец или выровненный для одновременного отображения одной записи, табличный и ленточный для отображения сразу всех записей.

Создание формы на основе запроса

Если в базе данных уже имеется запрос, в котором выбраны таблицы, нужные для получения результата, определены поля, которые необходимо включить в результат, параметры, сделаны вычисления, целесообразно создавать форму на основе такого запроса.

В запросе с именем **Накладные месяца** (рис. 5.11) из таблицы **НАКЛАДНАЯ** выбираются все накладные, по которым производилась отгрузка в заданном месяце.

В вычисляемом поле из даты в накладной извлекается номер месяца. Определение параметра в условии отбора этого поля с именем [Номер месяца] обеспечит выбор накладных заданного месяца.

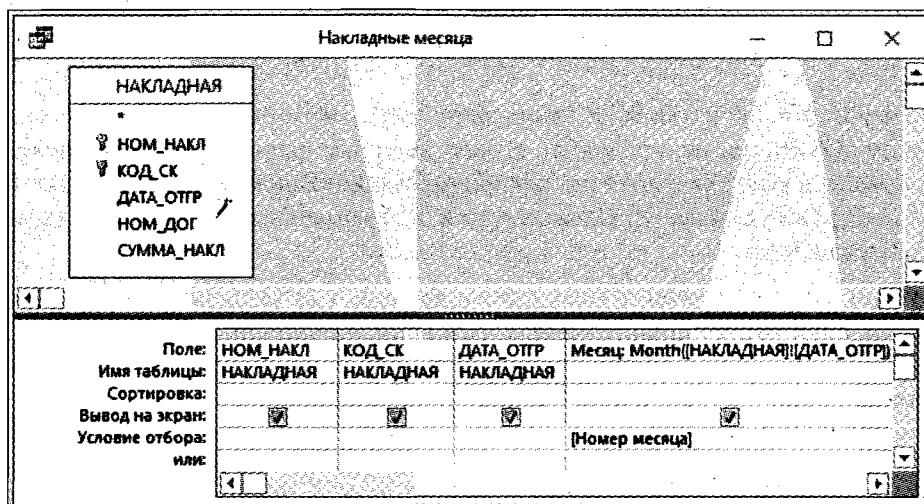


Рис. 5.11. Запрос для выборки накладных, выписанных в заданном месяце

Выбор запроса с параметром в качестве источника записей позволит создать форму, обеспечивающую диалог с пользователем и вывод только тех записей, которые соответствуют заданным в диалоге условиям отбора.

Для создания такой формы можно использовать любую из команд автоматического создания формы одним щелчком мыши: **Форма (Form)**, **Разделенная форма (Split Form)** или **Несколько элементов (Multiple Items)** (см. рис. 5.1), размещенных в группе **Формы (Forms)** на вкладке ленты **Создание (Create)**.

1. Выделив запрос в области навигации, щелкните на кнопке **Несколько элементов (Multiple Items)**. В результате выполнения этой команды откроется диалоговое окно **Введите значение параметра (Enter Parameter Value)**.
2. Введите номер месяца, например, 3. Форма откроется в режиме макета и отобразит записи, соответствующие введенному значению параметра (рис. 5.12).



Рис. 5.12. Форма, полученная по команде **Несколько элементов**

По команде **Несколько элементов (Multiple Items)** в форме автоматически создается табличный макет. В табличных макетах элементы расположены в двух разделах формы: надписи располагаются в заголовке формы, поля записей — по строкам в области данных.

3. Сохраните форму, воспользовавшись командой контекстного меню формы, и закройте ее.
4. Выберите форму в области навигации, выполните в ее контекстном меню команду **Открыть** или щелкните на ней двойным щелчком — откроется диалоговое окно для ввода значения параметра. Введите нужное значение. В открывшейся форме будут представлены записи, выбранные в соответствии с заданным условием отбора.
5. Для построенной формы открыт режим ввода новых записей. Очевидно, это находится в противоречии с заданным условием отбора, поэтому в режиме макета целесообразно открыть **Страницу свойств (Property Sheet)** формы и на вкладке **Данные (Data)** в строке **Разрешить добавление (Allow Additions)** выбрать значение **Нет**. После изменения свойства в форме не будет отображаться строка новой записи, а также померкнут соответствующий значок на панели перехода по записям и кнопка **Создать (New)** в группе **Записи (Records)** на вкладке ленты **Главная (Home)**. В приведенном примере добавление новой записи вообще невозможно, т. к. в форме не представлено обязательное поле вторичного ключа — номер договора.

Создание разделенной формы

Разделенная форма позволяет синхронно отображать данные одного источника в двух представлениях: в режиме формы и в режиме таблицы. В любой части разделенной формы можно производить изменение данных: добавление, обновление или удаление, которые тут же будут отображаться в другом представлении. При выделении поля в одной части формы выделяется то же поле в другой части.

Создайте однотабличную разделенную форму для работы с данными таблицы **ТОВАР**. Выберите таблицу **ТОВАР** в области навигации и выполните команду **Разделенная форма (Split Form)** на вкладке ленты **Создание (Create)** в группе **Формы (Forms)**. Эта команда обеспечит автоматическое создание формы на основе только одной выбранной таблицы. Созданная форма отобразится в режиме макета.

Работа с разделенной формой реализует преимущества обоих типов представления данных. Например, можно воспользоваться табличной частью формы, чтобы быстро найти запись о конкретном товаре, а затем просмотреть или изменить ее в другой части формы.

Если в вашей базе данных существуют формы, основанные на одной таблице или запросе, совсем не обязательно создавать разделенную форму с тем же источником данных заново, достаточно изменить свойство этой формы **Режим по умолчанию (Default View)**. В нашей базе данных ранее была создана форма **ТОВАР** для просмотра и изменения сведений о товарах. Откройте форму в режиме конструктора.

Для этого выберите ее в области навигации и в контекстном меню выполните команду **Конструктор (Design View)**. На вкладке ленты **Конструктор (Design)** в группе **Сервис (Tools)** выполните команду **Свойства страницы (Property Sheet)**. Измените на вкладке **Макет (Format)** значение свойства формы **Режим по умолчанию (Default View)** с **Простая форма (Single Form)** на **Разделенная форма (Split Form)** — форма ТОВАР примет вид, представленный на рис. 5.13.

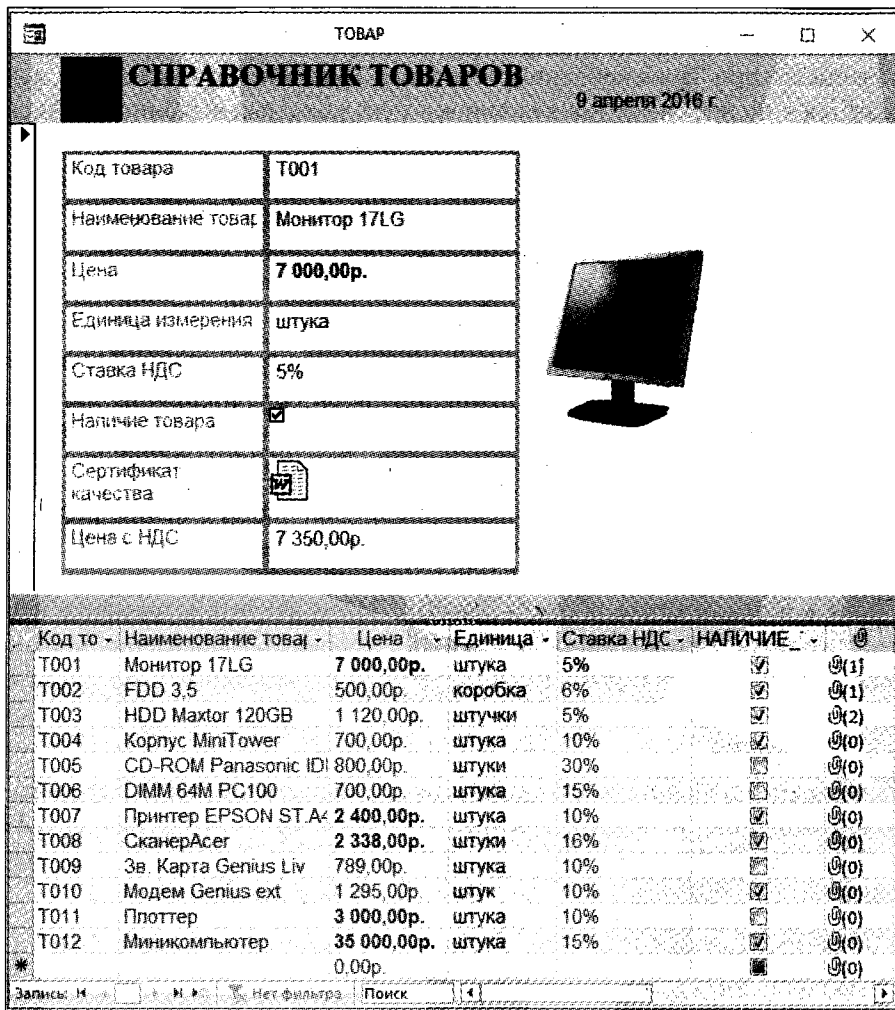


Рис. 5.13. Разделенная форма ТОВАР

Расположение таблицы в окне разделенной формы определяется в свойстве формы **Ориентация разделенной формы (Split Form Orientation)** на вкладке **Макет**. Свойство может принимать значения **Таблица сверху (Datasheet on Top)**, **Таблица снизу (Datasheet on Bottom)**, **Таблица слева (Datasheet on Left)** и **Таблица справа (Datasheet on Right)**. Изменить значение этого свойства можно только в режиме конструктора.

Задание 5.2. Создание разделенной формы

Преобразуйте созданную в задании 5.1 однотабличную форму для работы со справочником покупателей в разделенную форму. Добавьте в базу данных нового покупателя.

Вычисления в форме

Вычисления в форме могут осуществляться как в каждой записи формы, так и для группы записей при формировании итоговых величин. Вычисляемые величины отображаются в поле формы, но, в отличие от создаваемых в таблице вычисляемых полей, не сохраняются в таблице. Для сохранения результатов вычислений в таблице базы данных требуется подготовка макроса или процедуры на VBA (Visual Basic for Applications). Примеры макросов, обеспечивающих обновление поля таблицы значениями, вычисленными в форме, будут рассмотрены в *главе 9*.

Вычисления в каждой записи формы

Чтобы произвести вычисления на основе данных каждой записи формы, необходимо создать элемент управления вычисляемое поле, источником данных которого является выражение для расчета (создание вычисляемого поля в форме аналогично созданию такого поля в запросе).

Для создания такого элемента управления откройте форму в режиме макета или конструктора. Оба режима на ленте конструктора имеют в группе **Элементы управления** (Controls) кнопку **Поле** (Text Box). Нажмите ее и поместите вычисляемое поле в нужном месте области данных. Затем введите выражение в элемент вычисляемого поля, называемый **Свободный** (Unbound). Выражение должно начинаться со знака равенства (=). В качестве операндов выражения чаще всего используются имена полей, константы и функции, а в качестве операторов — знаки арифметических операций.

Пусть необходимо подсчитать и отобразить в форме величину НДС каждого товара в денежном выражении. Откройте созданную ранее простую форму **ТОВАР** в режиме конструктора. Создайте вычисляемый элемент управления и запишите в него выражение:

```
= [ЦЕНА] * [СТАВКА_НДС]
```

В связанную с полем надпись запишите: *Стоимость НДС*.

Выражение из элемента управления будет сохранено в свойстве **Данные** (Control Source), размещенном на вкладке **Данные** (Data). В режиме макета выражение нужно записать непосредственно в это свойство. Добавить вычисляемое поле в имеющийся в форме макет значительно проще в режиме макета.

Измените формат поля, выбрав на вкладке **Макет** (Format) в списке свойства **Формат поля** (Format) значение **Денежный** (Currency). Используйте **Формат по образцу** (Format Painter) для оформления нового элемента в стиле всех других элементов.

ТОВАР

Заголовок формы

СПРАВОЧНИК ТОВАРОВ (Дата)


Область данных


Код товара	КОД_ТОВ
Наименование товара	НАИМ_ТОВ
Цена	ЦЕНА
Единица измерения	ЕИ
Ставка НДС	СТАВКА_НДС
Наличие товара	<input checked="" type="checkbox"/>
Сертификат качества	ПолеВложенный
Цена с НДС	Цена с НДС
Стоимость НДС	=ЦЕНА*СТАВКА_НДС

Рис. 5.14. Форма с вычисляемым полем в режиме конструктора

ТОВАР

СПРАВОЧНИК ТОВАРОВ 9 апреля 2016 г.

Код товара	T001
Наименование товара	Монитор 17LG
Цена	7 000,00р.
Единица измерения	штука
Ставка НДС	5%
Наличие товара	<input checked="" type="checkbox"/>
Сертификат качества	
Цена с НДС	7 350,00р.
Стоимость НДС	350,00р.



Запись № 1 из 13 Нет фильтра Поиск

Рис. 5.15. Форма с вычисляемым полем в режиме просмотра

Форма в режиме конструктора и в режиме просмотра, полученная после создания вычисляемого поля, приведена на рис. 5.14 и 5.15 соответственно.

Вычисление итоговых значений

Вычисление итоговых значений в форме выполняется в примечании формы с помощью встроенных статистических функций, записываемых в выражениях в вычисляемых элементах управления.

Откройте форму ТОВАР в режиме конструктора. Чтобы поместить вычисляемое поле в область примечания формы, расширьте ее, установив курсор на границе так, чтобы он принял вид двунаправленной стрелки, и перетащив границу вниз. Затем щелкните на кнопке **Поле (Text Box)** в группе **Элементы управления (Controls)** и в области примечания вычертите вычисляемый элемент управления.

Для расчета среднего значения цены всех товаров запишите в вычисляемый элемент управления выражение:

= Avg ([ЦЕНА])

и в надпись: Средняя цена товара (рис. 5.16). Измените формат поля на **Денежный (Currency)**.

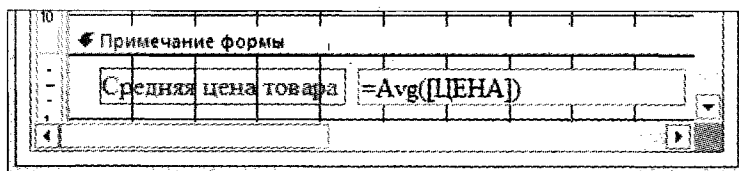


Рис. 5.16. Примечание формы в режиме конструктора с полем для вычисления итогов

После перехода в режим просмотра в форме отображается результат расчета (рис. 5.17).

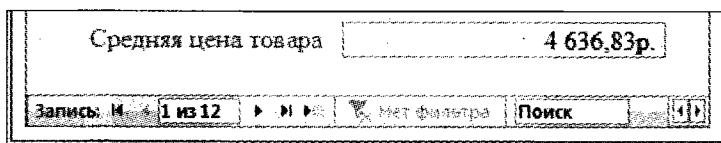


Рис. 5.17. Форма в режиме просмотра с расчетным полем для итоговых значений

Многотабличные формы

Многотабличная форма создается для работы с данными нескольких взаимосвязанных таблиц. Она может быть *одиночной*, построенной на многотабличном запросе, или *составной*, включающей главную форму и подчиненные ей формы.

Источником данных одиночной формы всегда является многотабличный запрос. Если такая форма строится мастером, то он позволит определить набор взаимосвя-

занных таблиц и сам создаст многотабличный запрос — источник записей формы. Форма может быть простой, отображающей одну запись в столбик, или ленточной, отображающей все записи в табличном виде с надписями в заголовке формы. Для создания одиночной формы могут быть использованы команды **Форма (Form)**, **Несколько элементов (Multiple Items)** или мастер форм.

Составная форма может состоять из главной формы и одной или нескольких подчиненных включаемых форм на одном или нескольких уровнях иерархии. Источником записей главной формы может быть таблица или запрос. Подчиненная форма строится на основе таблицы или запроса, подчиненного источнику записей главной формы. Как правило, источники записей главной и подчиненной форм находятся в отношении 1 : М. Подчиненная форма отображает данные из всех записей подчиненного источника, которые связаны с записью, отображаемой в главной форме.

Для разработки составной формы можно воспользоваться следующими средствами:

1. На основе главной таблицы создать командой **Форма** простую форму с макетом в столбик.
2. На основе подчиненной таблицы командой **Несколько элементов** создать ленточную форму — это многозаписевая форма.
3. Закрыть вторую форму, оставив первую открытой в режиме макета.
4. Перетащить вторую форму из области навигации в первую форму — она отобразится в элементе управления **Подчиненная форма/отчет (Subform/Subreport)**.
5. Если не установилась связь между формами, следует в главной форме открыть свойства элемента управления **Подчиненная форма/отчет** и на вкладке **Данные (Data)** заполнить строки **Основные поля (Link Master Fields)** и **Подчиненные поля (Link Child Fields)**. Это можно сделать с помощью построителя, вызываемого в строке свойства **Основные поля**.

Воспользовавшись приведенным алгоритмом, несложно создать составную форму на основе таблиц **НАКЛАДНАЯ** и **ОТГРУЗКА**, находящихся в отношении 1 : М (рис. 5.18). Связь этих форм осуществляется по составному ключу **НОМ_НАКЛ** и **КОД_СК**. В строке свойств они записываются через точку с запятой.

Очевидно, для разработки полноценной формы для работы с накладными необходимо дополнить главную форму данными из таблиц **ДОГОВОР** и **ПОКУПАТЕЛЬ**, а подчиненную форму — данными из таблицы **ТОВАР**.

Решить эту задачу можно, выполнив следующие действия:

1. Создать запросы на выборку для источника записей главной формы на основе таблиц **НАКЛАДНАЯ**, **ДОГОВОР** и **ПОКУПАТЕЛЬ**, подчиненной формы — на основе таблиц **ОТГРУЗКА** и **ТОВАР**, выбирая необходимые поля.
2. На основе первого запроса создать простую форму командой **Форма (Form)**, на основе второго — ленточную командой **Несколько элементов (Multiple Items)**.
3. Закрыть вторую форму и затем перетащить ее из области навигации в первую.

Накладная

НАКЛАДНАЯ

Номер накладной: H001
 Код склада: C01
 Дата отгрузки: 18.01.2016
 Номер договора: D111
 Общая сумма по накладной: 387 338,00р.

ОТГРУЗКА

Код накладной	Код склада	КОД ТОВ	Количество отгружено	Сумма отгружено
H001	C01	T001	50	350 000,00р.
H001	C01	T002	10	5 000,00р.
H001	C01	T005	2	8 000,00р.
H001	C01	T007	10	22 000,00р.
H001	C01	T008	10	2 338,00р.
*	H001	C01	0	0,00р.

Записи: 1 из 5 | Без фильтра | Поиск

Рис. 5.18. Составная форма в режиме макета

НакладнаяДоговорПок

НакладнаяДоговорПок

Номер накладной: H001
 Код склада: C01
 Дата отгрузки: 18.01.2016
 Номер договора: D111
 Общая сумма по накладной: 387 338,00р.

Дата заключения договор: 11.01.2013
 Наименование покупателя: Компьютер маркет
 Адрес: Москва
 НОМ_РЧН: 7635850976326453656.
 БАНК: Мост

ОтгрузкаТовар

Код накладной	Код склада	КОД ТОВ	Наименование товара	Цена	Количество отгружено
H001	C01	T001	Монитор 17LG	7 000,00р.	50
H001	C01	T002	FDD 3,5	500,00р.	10
H001	C01	T005	CD-ROM Panasonic	800,00р.	2
H001	C01	T007	Принтер EPSON ST.A4	2 400,00р.	10
H001	C01	T008	СканерAcer	2 338,00р.	10
*					

Записи: 1 из 5 | Без фильтра | Поиск

Рис. 5.19. Составная форма для работы с накладными

4. Для проверки и установки связи главной и подчиненной форм открыть свойства элемента управления **Подчиненная форма/отчет (Subform/Subreport)** и на вкладке **Данные (Data)**, если необходимо, заполнить строки **Основные поля (Link Master Fields)** и **Подчиненные поля (Link Child Fields)**.

На рис. 5.19 приведена составная форма, в которой источником записей простой главной формы является первый запрос с именем **НакладнаяДоговорПок**, а источником записей ленточной подчиненной формы — второй запрос с именем **ОтгрузкаТовар**.

Создание многотабличной формы с помощью мастера

Основным средством создания многотабличной формы можно считать мастер форм, который, запросив у пользователя сведения о включаемых в форму полях из нескольких взаимосвязанных таблиц и/или запросов, создает составную или одиночную форму. При этом мастер на основе выбранных взаимосвязанных таблиц может сам создать запросы на выборку, используемые в качестве источника записей формы.

Полученная с помощью мастера форма при необходимости может быть отредактирована в режиме макета или конструктора. Режим конструктора позволяет детально просмотреть структуру формы и настроить любой ее раздел. Некоторые задачи удобнее выполнять в режиме конструктора, а не макета, некоторые могут выполняться только в режиме конструктора.

С помощью мастера создайте форму для работы с данными о покупателях и их договорах. Очевидно, такая форма должна строиться на основе двух таблиц: **ПОКУПАТЕЛЬ** и **ДОГОВОР**, находящихся в отношении 1 : М и связанных полем код_пок (код покупателя). Для одновременной работы с записью главной и несколькими связанными записями подчиненной таблицы целесообразно построить многотабличную составную форму, в которой в главную форму будет встроена подчиненная форма.

Для вызова мастера форм выполните на вкладке ленты **Создание (Create)** в группе **Формы (Forms)** команду **Мастер форм (Form Wizard)** — отобразится окно мастера **Создание форм (Form Wizard)**, представленное на рис. 5.20.

Если предварительно в области навигации не была выбрана таблица **ПОКУПАТЕЛЬ**, которая представляет сторону «один» в отношении 1 : М с таблицей **ДОГОВОР**, выберите ее в раскрывающемся списке **Таблицы и запросы (Tables and queries)** диалогового окна мастера. Далее отберите из списка **Доступные поля (Available Fields)** необходимые в форме поля и переходите к выбору полей из подчиненной таблицы **ДОГОВОР**.

ЗАМЕЧАНИЕ

Порядок выбора таблиц не имеет принципиального значения.

Из таблицы **ПОКУПАТЕЛЬ** обязательно выберите ключевое поле код_пок (код покупателя) и все другие обязательные поля, иначе форма будет непригодна для вво-

да сведений о новых покупателях. Из таблицы ДОГОВОР обязательно выберите ключевое поле код_дог (код договора), а код покупателя (поле связи таблиц) выбирать не следует, поскольку значения этого поля станут тогда неоднократно повторяться в подчиненной форме, если покупатель заключил несколько договоров, а ввод значения этого вторичного ключа в таблицу ДОГОВОР обеспечивается в форме автоматически.

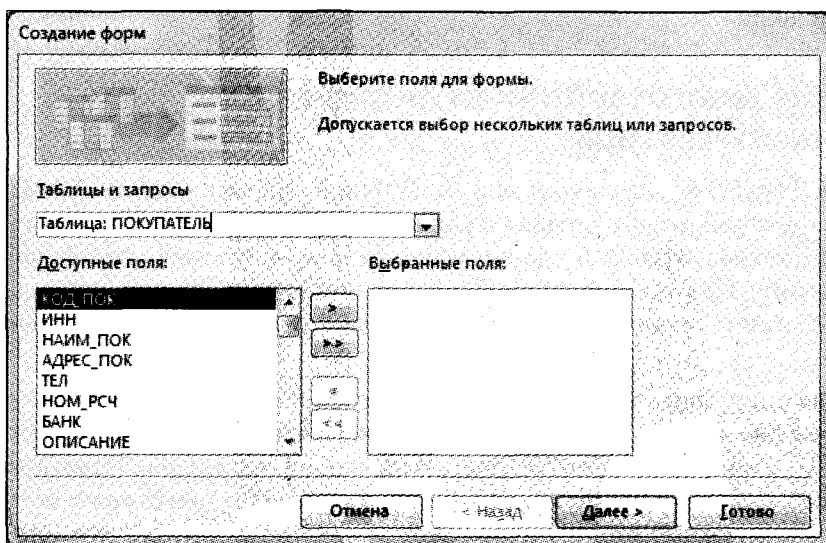


Рис. 5.20. Выбор таблиц и полей для создаваемой формы

ВНИМАНИЕ!

Если выбраны таблицы, для которых не определена связь в схеме данных, появится сообщение о невозможности создать форму. При этом мастер или предлагает изменить состав полей, или выводит схему данных для возможного определения нужных связей. После изменения связей мастер должен запускаться заново.

После выбора полей для обеих таблиц и нажатия кнопки **Далее** (Next) в окне **Создание форм** (Form Wizard) (рис. 5.21) в списке **Выберите тип представления данных** (How do you want to view your data) надо выделить имя таблицы **ПОКУПАТЕЛЬ**, которая является источником данных главной формы. Поскольку эта таблица была выбрана первой, то она уже выделена.

Таблица **ПОКУПАТЕЛЬ**, как источник данных основной формы, в нашем случае является главной по отношению к другой выбранной для формы таблице **ДОГОВОР**, поэтому в окне **Создание форм** (Form Wizard) предлагается выбрать один из двух возможных вариантов подключения форм:

- для включения подчиненной формы в главную надо отметить переключатель **Подчиненные формы** (Form with subform(s));
- для включения кнопки, вызывающей подчиненную форму, надо выбрать **Связанные формы** (Linked forms).

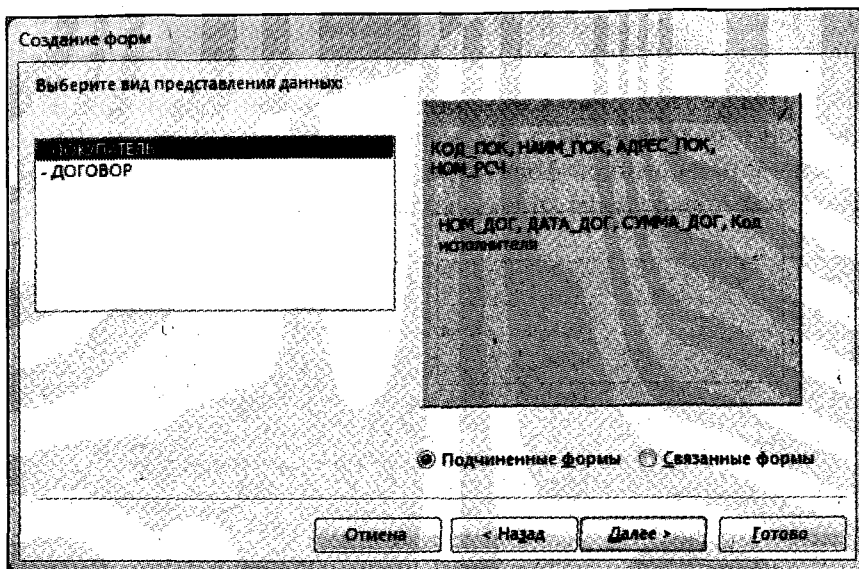


Рис. 5.21. Выбор варианта включения подчиненной формы

Выберем первый вариант с непосредственным включением подчиненной формы в главную. На рис. 5.21 видно, как распределяются поля в основной и подчиненной частях формы при выборе этого варианта.

В следующем диалоговом окне предоставляется возможность выбрать вид отображения данных в подчиненной форме: **ленточный** (tabular) или **табличный** (datasheet) (рис. 5.22). При выборе ленточного вида для списка выводимых в под-

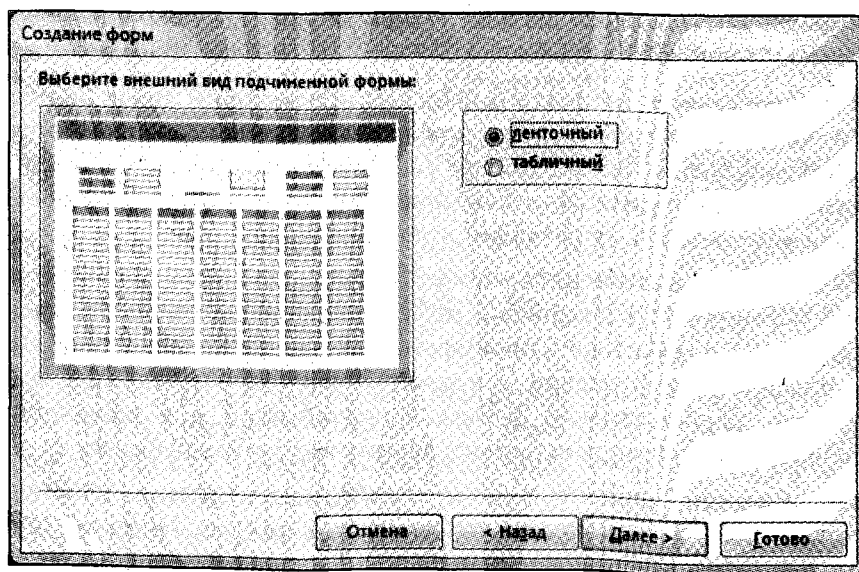


Рис. 5.22. Выбор ленточного вида для подчиненной формы

чиненной форме записей названия столбцов (полей) берутся из подписи, заданной в общих свойствах поля при конструировании подчиненной таблицы.

В последнем диалоговом окне **Создание форм (Form Wizard)** можно изменить имена основной и подчиненной форм, если был выбран вариант с непосредственным включением подчиненной формы (рис. 5.23), или имя связанной формы, если был выбран вариант включения кнопки, вызывающей связанную форму.

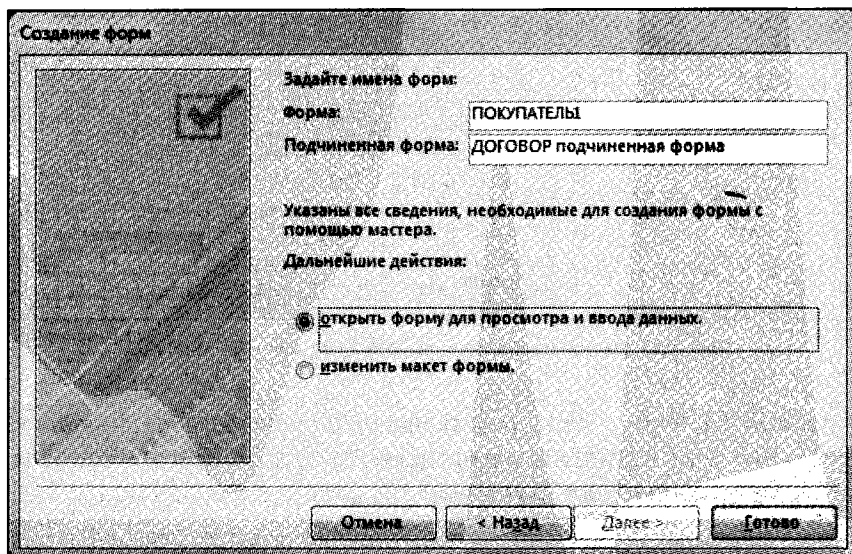


Рис. 5.23. Задание имен форм и выбора режима отображения

В том же окне можно выбирать дальнейшие действия: **открыть форму для просмотра и ввода данных (Open the form to view or enter information)** или **изменить макет формы (Modify the form's design)**.

Если был выбран вариант **открыть форму для просмотра и ввода данных (Open the form to view or enter information)**, после нажатия кнопки **Готово (Finish)** мастер завершит создание формы и откроет ее в режиме формы (рис. 5.24), предназначенном для работы с данными таблиц, на которых построена форма. Сохранение обеих форм, созданных мастером, производится автоматически.

При выборе варианта **изменить макет формы (Modify the form's design)** (см. рис. 5.23) форма после завершения работы мастера выводится в режиме конструктора (рис. 5.25), позволяющем произвести нужную доработку. Форма состоит из главной и подчиненной: в главной — в макете в столбик представлены выбранные поля источника данных главной формы, в подчиненной — в табличном макете представлены выбранные поля источника данных подчиненной формы.

В процессе доработки формы, используя технику редактирования, можно перемещать поля в главной форме, менять их свойства, в том числе шрифт и размеры, подписи полей, формировать текст в заголовке формы. Аналогичные действия по доработке можно выполнять для подчиненной формы.

ПОКУПАТЕЛЬ

Код покупателя: П001
ИНН: 778957651111
Наименование: Компьютер маркет
Адрес: Москва
НОМ_РСЧ: 78358509763264536567
БАНК: Мост

ДОГОВОР

Номер договора	Дата заключения	Сумма по договору	Код исполнителя
Д111	11.01.2016	6 414 000,00р.	6
Д222	05.02.2016	152 280,00р.	1
Д777	14.06.2016	1 579 820,00р.	1
*		0,00р.	

Записи: 1 из 3 | Нет фильтра | Поиск

Рис. 5.24. Форма, открытая для просмотра, корректировки и ввода данных

Следует отметить исключительную простоту и универсальность всех действий по изменению размеров, перемещению любого элемента, редактированию подписей и подобных действий. На этапе доработки можно выполнить и более сложные действия по редактированию формы и настроить составную форму в соответствии с любыми требованиями пользователя. Далее будут рассмотрены более подробно возможности конструктора при создании и редактировании форм.

Задача 5.3. Создание многотабличной формы второго уровня

Включите в созданную мастером составную форму **Покупатель**, представленную на рис. 5.24 и 5.25, подчиненную форму второго уровня. Создайте эту форму на основе таблицы **ПОСТАВКА_ПЛАН**, которая содержит сведения о заказах по договору. Таблицы **ДОГОВОР** и **ПОСТАВКА_ПЛАН** находятся в отношениях 1 : М.

Подчиненную форму создайте командой **Несколько элементов**, выделив предварительно таблицу **ПОСТАВКА_ПЛАН**. Перетащите ее в область данных подчиненной формы **ДОГОВОР**. Access, выдав сообщение «Свойство **Режим по умолчанию** формы с подчиненной формой не может иметь значение **Ленточная форма**», сам заменит значение **Ленточная форма** на **Простая форма** и вставит подчиненную форму в форму **ДОГОВОР**.

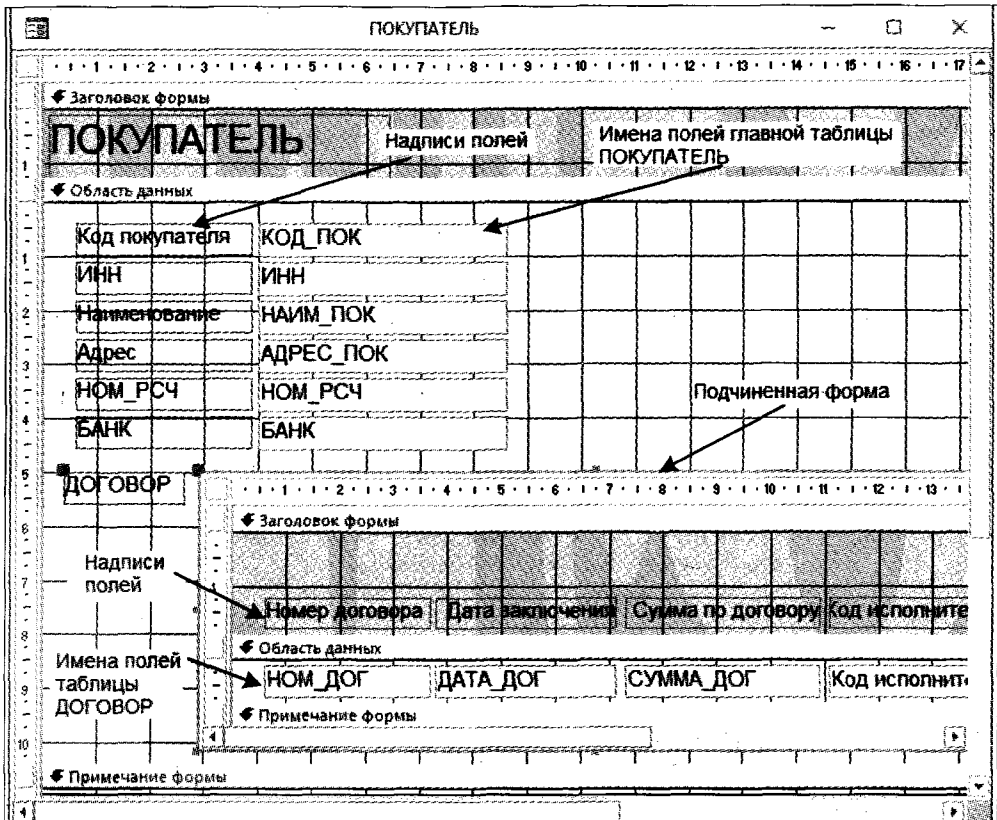


Рис. 5.25. Форма в режиме конструктора после завершения работы мастера

Создание одиночной многотабличной формы

Одиночную форму, включающую поля из нескольких связанных таблиц, позволяет быстро построить инструмент **Пустая форма (Blank Form)**. Выполните команду **Пустая форма** на вкладке ленты **Создать (Create)** в группе **Формы (Forms)**. Откроется пустая форма в режиме макета и отобразится область **Список полей (Field List)**. В списке перечислены все таблицы базы данных, и предоставляется возможность открыть список полей каждой из них. Возможно для открытия списка таблиц понадобится нажать **Показать все таблицы**. Чтобы добавить поле в форму, выполните на нем двойной щелчок или перетащите его в форму. Для отображения каждого поля Access создает в форме соответствующий элемент управления и привязывает его к полю. Кроме того, для элемента управления создается присоединенная надпись.

Важно понимать, что создание формы начинается с определения источника записей формы. Добавление полей в форму из области **Список полей (Field List)** автоматически решает эту задачу, поскольку при добавлении полей в форму создается инструкция SQL `SELECT`. Эта инструкция записывается в строку свойств формы **Источник записей (Record Source)** на вкладке **Данные (Data)**. В графическом представ-

лении ее можно просмотреть в построителе запросов, который вызывается командой **Изменение источника записи** (Edit Record Source) контекстного меню области **Список полей** (Field List). При последующих добавлениях полей из списка источник записей автоматически корректируется.

Создание одиночной формы следует начинать с включения полей подчиненной таблицы. Эта таблица определит набор записей, отображаемых через форму, и ее запись станет основой для образования записи формы. Далее можно добавлять поля из главных по отношению к ней таблиц.

Создайте форму для просмотра накладных, выписанных для отгрузки товаров. При этом включите в просматриваемую накладную сведения о договоре, на основании которого делается отгрузка, о покупателе и складе, с которого отгружается товар.

Выполните команду **Пустая форма** (Blank Form). В области **Список полей** (Field List) откройте список полей таблицы НАКЛАДНАЯ, щелкнув на знаке «плюс» (+) рядом с ее именем. Добавьте в форму необходимые поля. Далее последовательно добавляйте поля из таблиц ДОГОВОР, ПОКУПАТЕЛЬ и СКЛАД. После добавления полей из всех таблиц область **Список полей** примет вид, показанный на рис. 5.26.

ОБРАТИТЕ ВНИМАНИЕ

При создании одиночной формы добавление в нее полей должно начинаться с подчиненной таблицы нижнего уровня. В нашем случае — с таблицы НАКЛАДНАЯ, которая является подчиненной таблицам ДОГОВОР и СКЛАД, а таблица ДОГОВОР подчинена таблице ПОКУПАТЕЛЬ.

Если при добавлении поля некоторой таблицы обнаруживается, что эта таблица не связана с таблицей, поля которой уже включены в форму, будет выведено окно для определения связи между ними. На рис. 5.27 показано такое окно при добавлении поля из таблицы ПОКУПАТЕЛЬ в случае, когда в форму уже включены поля из договора, а в схеме данных отсутствует связь между таблицами.

На основе выбранных таблиц автоматически создается источник записей формы — запрос на выборку. В этом запросе устанавливаются и созданные при перетаскивании полей в форму связи между таблицами.

Просмотреть и изменить запрос, являющийся источником записей формы, можно по команде **Изменение источника записи** (Edit Record Source) контекстного меню области **Список полей** (Field List). Соответствующую инструкцию SQL можно просмотреть в свойстве формы **Источник записей** (Record Source), представленном на вкладке **Данные** (Data).

Пользуясь инструментами группы **Колонтитулы** (Header/Footer) на вкладке **Работа с макетами форм | Конструктор** (Form Layout Tools | Design), добавьте в форму заголовок, эмблему компании, дату и время. Первоначально в пустой форме представлен только один раздел — **Область данных** (Detail). Сохраните форму под именем Накладная_одиночная.

Форма, построенная с помощью инструмента **Пустая форма** (Blank Form), представлена на рис. 5.28.

Список полей X

Показать только поля в текущем источнике записей

Поля, доступные в данном представлении:

- ДОГОВОР
 - НОМ_ДОГ
 - ДАТА_ДОГ
 - КОД_ПОК
 - СУММА_ДОГ
 - НОМ_ДОГ_ДОГОВОР
 - КОД_ПОК_ДОГОВОР
- НАКЛАДНАЯ
 - НОМ_НАКЛ
 - КОД_СК
 - ДАТА_ОТПР
 - НОМ_ДОГ
 - СУММА_НАКЛ
- ПОКУПАТЕЛЬ
 - КОД_ПОК
 - ИНН
 - НАИМ_ПОК
 - АДРЕС_ПОК
 - ТЕЛ
 - НОМ_РСЧ
 - БАНК
- СКЛАД
 - КОД_СК
 - КОД_Ф
 - НАИМ_СК

Доступные поля в связанной таблице:

- ИСПОЛНИТЕЛЬ
- ОТГРУЗКА
- ПОСТАВКА_ПЛАН

Доступные поля в другой таблице:

- КАЛЕНДАРЬ
- ТОВАР

Рис. 5.26. Область
Список полей при создании
одиночной формы Накладная

Определение связи

Выберите таблицу для связи с "ПОКУПАТЕЛЬ":

Как следует связать "ПОКУПАТЕЛЬ" с этой таблицей?

Это поле в "ПОКУПАТЕЛЬ"...

Связано с данным полем в "ДОГОВОР"...

КОД_ПОК	↔	КОД_ПОК
	↔	

Одной записи в "ПОКУПАТЕЛЬ" соответствует одна запись в "ДОГОВОР"
 Многим записям в "ПОКУПАТЕЛЬ" соответствует одна запись в "ДОГОВОР"
 Одной записи в "ПОКУПАТЕЛЬ" соответствует много записей в "ДОГОВОР"

Рис. 5.27. Определение связи между таблицами при создании одиночной формы Накладная

Накладная	
Код покупателя	П001
Наименование	Компьютер маркет
Номер накладной	Н001
Код склада	С01
Наименование склада	Главный
Ответственное лицо	Иванов Т.С.
Дата отгрузки	18.01.2016
Общая сумма по накладной	387 338,00р.
Номер договора	Д111
Дата заключения	11.01.2016
Сумма по договору	6 414 000,00р.

Записи: 1 из 12 Поиск

Рис. 5.28. Одиночная форма, созданная с помощью инструмента Пустая форма

Создание и редактирование формы в режиме конструктора

Средствами конструктора форм можно создать любую форму с нуля. В режиме конструктора также можно в любой момент отредактировать ранее разработанную любыми средствами форму. При создании и редактировании формы может выполняться: определение источника данных формы, добавление новых полей и надписей, включение полей со списком, создание кнопок, добавление подчиненных форм, внедрение объектов из других приложений — например, рисунков, диаграмм и т. п. Наиболее точная и полная настройка структуры и внешнего вида всех разделов и элементов формы производится в режиме конструктора. Ряд элементов управления доступен только в режиме конструктора.

Создание новой формы конструктором

Конструирование формы начинается выполнением команды **Конструктор форм** (Form Design) на вкладке ленты **Создать** (Create) в группе **Формы** (Forms). В результате выполнения команды открывается окно пустой формы в режиме конструктора с именем формы по умолчанию **Форма1** (Form1). В пустой форме конструктора представлен только один раздел — **Область данных** (Detail).

Начать конструирование формы следует с определения ее источника данных. Источник данных может быть создан различными способами. Можно открыть

свойства формы и на вкладке **Данные (Data)** в строке **Источник записей (Record Source)** открыть список и выбрать таблицу или запрос. Здесь же, воспользовавшись услугами построителя запросов, можно создать запрос в режиме конструктора, а соответствующая инструкция SQL будет записана в строку свойства. Наиболее простым способом определения источника записей является использование области **Список полей (Field List)**, в которой отображаются поля всех таблиц базы данных. При перетаскивании нужных полей в форму автоматически строится запрос на выборку — источник записей формы. Для отображения списка полей щелкните на кнопке **Добавить поля (Add Existing Fields)** на вкладке ленты **Конструктор (Design)** в группе **Сервис (Tools)**.



Создайте форму для работы с накладными, в которой будет представлена информация об отгруженных товарах.

Очевидно, что такая форма должна строиться на основе главной таблицы **НАКЛАДНАЯ** с общими данными о накладной и подчиненной таблицы **ОТГРУЗКА** со сведениями об отгруженных по накладной товарах. Соответственно, должна быть построена составная форма, в которой главной является форма, созданная на основе таблицы **НАКЛАДНАЯ**, а подчиненной — форма на основе таблицы **ОТГРУЗКА**.

Для создания главной формы достаточно перетащить поля из списка таблицы **НАКЛАДНАЯ** в нужное место области данных пустой формы. Как только вы перетащили первое поле в форму, в области списка полей появился новый раздел **Поля, доступные в данном представлении (Fields Available for this View)**, в который перемещается таблица **НАКЛАДНАЯ**. Построенный на этой таблице запрос на выборку, называемый *представлением*, будет автоматически определен в качестве базового источника данных формы. Если необходимо дополнить запись таблицы **НАКЛАДНАЯ** более подробными сведениями о договоре и/или покупателе, перетащите в область данных недостающие поля соответствующих таблиц. Автоматически откорректируется источник данных формы.

В результате перетаскивания поля в форме появляется элемент управления, в котором отобразится имя поля таблицы. В присоединенной надписи будет использовано значение из свойства поля **Подпись (Caption)**, определенное при конструировании таблицы.

Добавление подчиненной формы

Откройте главную форму в режиме конструктора. Для добавления подчиненной формы воспользуйтесь элементом управления  **Подчиненная форма/отчет (Subform/Subreport)** на вкладке ленты **Конструктор (Design)** и поместите его в нужном месте главной формы, растянув курсором до нужного размера. По умолчанию к созданию подчиненной формы подключается мастер. Для этого необходимо, чтобы была включена кнопка  **Использовать мастера (Use Control Wizards)**, находящаяся в группе **Элементы управления (Controls)**. Мастер помогает включить в качестве подчиненной ранее созданную форму или построить новую на основе таблицы или запроса.

ЗАМЕЧАНИЕ

Если кнопка **Использовать мастера** (Use Control Wizards) не отображается в группе **Элементы управления** (Controls), откройте список этой группы. По умолчанию использование мастера включено.

В окне мастера выберите форму **ОТГРУЗКА** (рис. 5.29), которая должна быть ранее построена на основе соответствующей таблицы с помощью команды **Несколько элементов** (Multiple Items). Источник данных подчиненной формы **ОТГРУЗКА** должен содержать поля связи: **НОМ_НАКЛ** (номер накладной), **КОД_СК** (код склада), хотя в форме они могут и не отображаться. Это необходимо для установления связи главной и подчиненной форм.

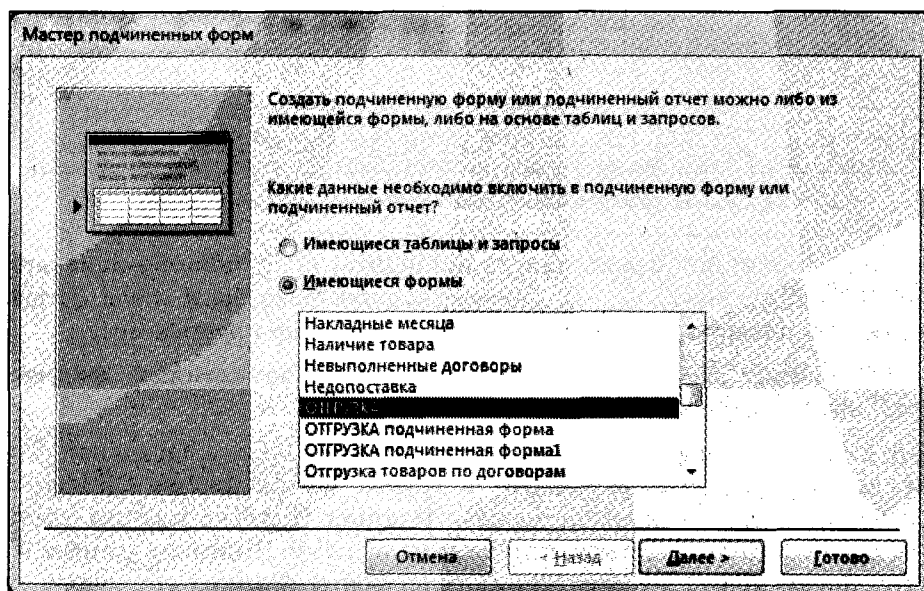


Рис. 5.29. Окно мастера для включения формы в качестве подчиненной

ЗАМЕЧАНИЕ

В окне мастера подчиненных форм предлагается выбор из имеющихся таблиц или запросов, однако отображаются только формы. Для выбора из имеющихся таблиц или запросов следует отметить этот переключатель и перейти далее. Только в открывающемся после этого окне возможен выбор таблиц и/или запросов и их полей. С этого начинается создание мастером новой — в нашем случае подчиненной — формы.

После выбора подчиненной формы **ОТГРУЗКА** в следующем окне мастер предлагает выбрать вариант связи главной и подчиненной форм: **НОМ_НАКЛ**, **КОД_СК** (рис. 5.30). Согласитесь с ним. После определения имени подчиненной формы в следующем окне процесс создания подчиненной формы завершается.

Доработайте форму в режиме конструктора. Проверьте правильность установленной связи в свойствах элемента **Подчиненная форма/отчет** (Subform/Subreport) на вкладке **Данные** (Data).

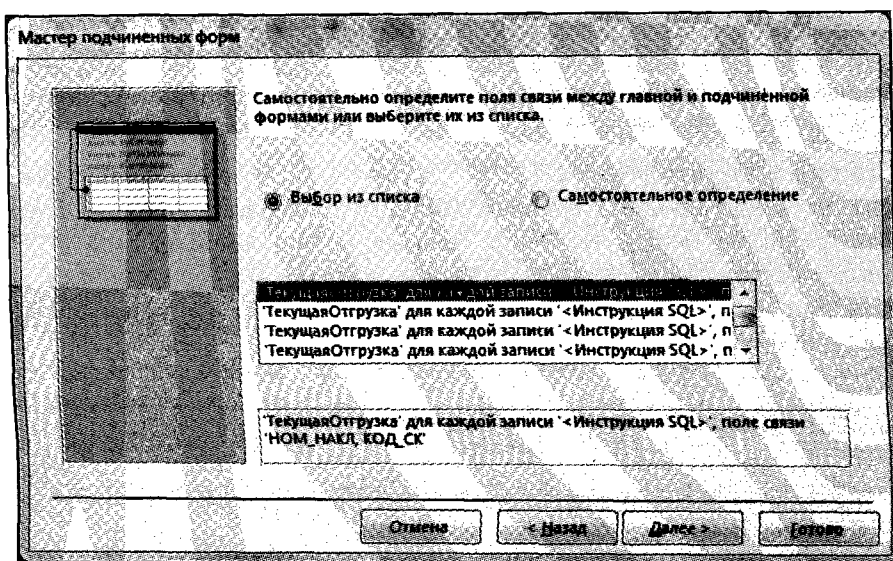


Рис. 5.30. Окно мастера для определения поля связи с подчиненной формой

Удалите из подчиненной формы (не из источника) поля, повторяющие значения главной формы: `НОМ_НАКЛ`, `КОД_СК`. После удаления некоторого поля оставшиеся в форме поля автоматически сдвигаются влево, не оставляя пустых мест.

Если необходимо, измените размер элемента **Подчиненная форма/отчет** (Subform/Subreport), получивший имя `Отгрузка`.

Добавьте в форму заголовок и примечание, щелкнув на кнопке **Заголовок** (Title) на вкладке ленты **Конструктор** (Design) в группе **Колонтитулы** (Header/Footer). В заголовок вставится элемент управления **Надпись** (Label) с текстом, соответствующим имени формы. Измените этот текст на нужный. Выделите элемент и измените его шрифт, используя средства вкладки ленты **Формат** (Format) в группе **Шрифт** (Font) или ленты **Главная** (Home) в группе **Форматирование текста** (Text Formatting).

Если номер накладной нужно отображать в заголовке, перетащите поле `НОМ_НАКЛ` в область заголовка и измените его надпись и формат.

Удобным средством настройки внешнего вида является группировка/разгруппировка элементов формы в макете. Выделите, удерживая нажатой клавишу `<Shift>`, все поля главной формы **Накладная**, надписи можно не включать. Нажмите кнопку **В столбик** (Stacked) в группе **Таблица** (Table) на вкладке **Упорядочить** (Arrange) — выделенные поля будут включены в один макет.

Для улучшения дизайна формы можно, используя свойства формы, на вкладке **Макет** (Format) убрать область выделения записи (слева), полосу прокрутки, кнопки перехода по записям, разделительные линии между областью заголовка, данных и примечаний и т. д.

Форма в режиме конструктора и просмотра, полученная после включения подчиненной формы мастером, приведена на рис. 5.31 и 5.32 соответственно.

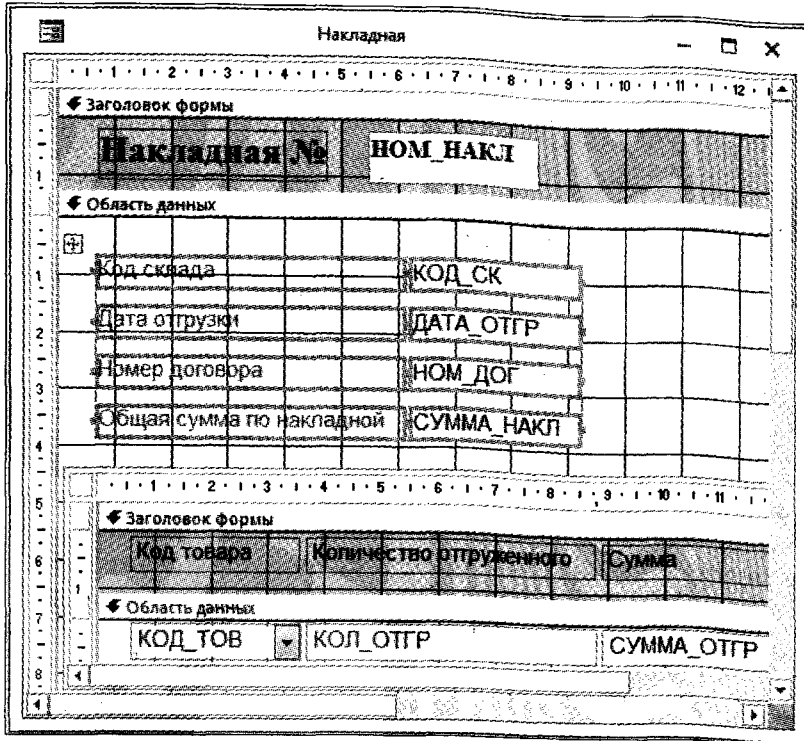


Рис. 5.31. Форма в режиме конструктора после включения подчиненной формы мастером

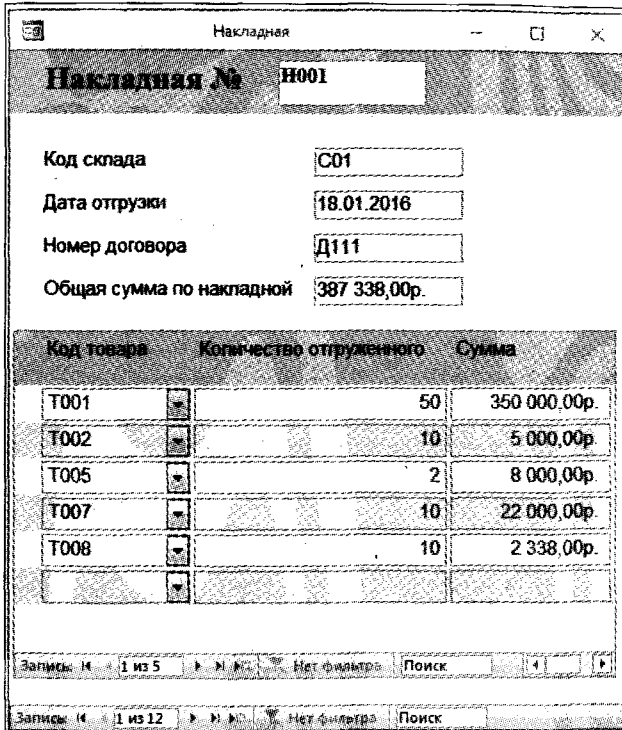


Рис. 5.32. Форма в режиме просмотра

Рассмотрим вариант встраивания подчиненной формы без использования мастера. Выключите мастер форм, нажав кнопку **Использовать мастера** (Use Control Wizards). Вставьте в главную форму элемент управления **Подчиненная форма/отчет** (Subform/Subreport) — в форме отобразится **Свободный** (Unbound) элемент подчиненной формы с именем **ВнедренныйN** (ChildN).

Для определения источника данных подчиненной формы откройте ее свойства. Если в форме выделен внедренный элемент, в окне свойств отобразится **Тип выделенного элемента: Подчиненная форма/отчет** и в поле со списком **ВнедренныйN** (ChildN). В противном случае можно самостоятельно выбрать внедренный объект из списка. Перейдите на вкладку **Данные** (Data). В строке **Объект-источник** (Source Object) выберите из списка имен форму, которая будет подчиненной. В нашем примере — форму **ОТПРУЗКА**. После этого в рамке подчиненной формы отобразится выбранная форма в режиме конструктора.

Если ранее в схеме данных была определена связь таблиц, соответствующих формам, связь между главной и подчиненной формами устанавливается автоматически (достаточно щелкнуть на значке в конце строки свойства **Подчиненные поля** (Link Child Fields) или **Основные поля** (Link Master Fields)). Если связи между таблицами не определены, то в окне свойств в строках **Подчиненные поля** и **Основные поля** следует ввести имена полей, по которым связываются таблицы. В рассмотренном примере связь реализуется полями **НОМ_НАКЛ** (номер накладной) и **КОД_СК** (код склада) (рис. 5.33). Имена полей в строке свойств разделяются точкой с запятой. Аналогичным образом можно вместо формы вставить таблицу **ОТПРУЗКА**.

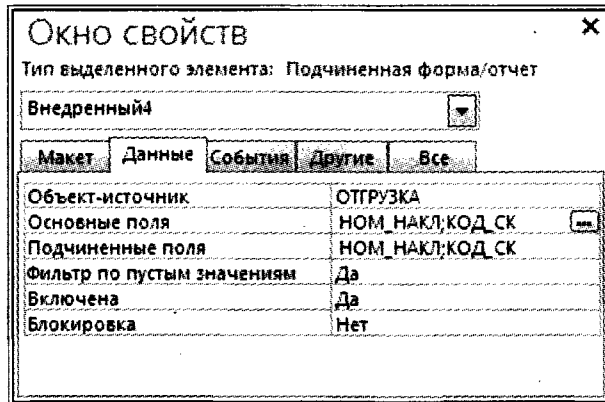


Рис. 5.33. Окно свойств подчиненной формы

Самый простой способ — создание подчиненной формы путем перетаскивания одной формы в другую. Откройте в режиме конструктора форму, которая должна быть главной. Перетащите форму или таблицу, на основе которой нужно создать подчиненную форму, из области навигации в главную форму.

Вычисление итогового значения в подчиненной форме и вывод его в текущей записи главной формы

Вычисление на основе данных каждой записи формы, а также вычисление итоговых значений в форме с помощью встроенных статистических функций было рассмотрено в предыдущих разделах главы. Теперь рассмотрим, как итоговые значения, вычисленные в подчиненной форме, отобразить или использовать в дальнейших расчетах в главной форме.

Если главная форма и ее подчиненная форма построены на основе таблиц, между которыми установлена связь типа с 1 : М, в основной форме выводится одна текущая запись, а в подчиненной форме — связанные с ней записи.

Пусть необходимо подсчитать, на какую сумму заключены договоры с каждым из покупателей, и отобразить ее в форме. Откройте в режиме конструктора созданную ранее форму ПОКУПАТЕЛЬ (см. рис. 5.24 и 5.25), включающую одну подчиненную форму ДОГОВОР. Рассчитайте общую сумму по договорам одного покупателя в подчиненной форме. Для этого в примечании подчиненной формы ДОГОВОР

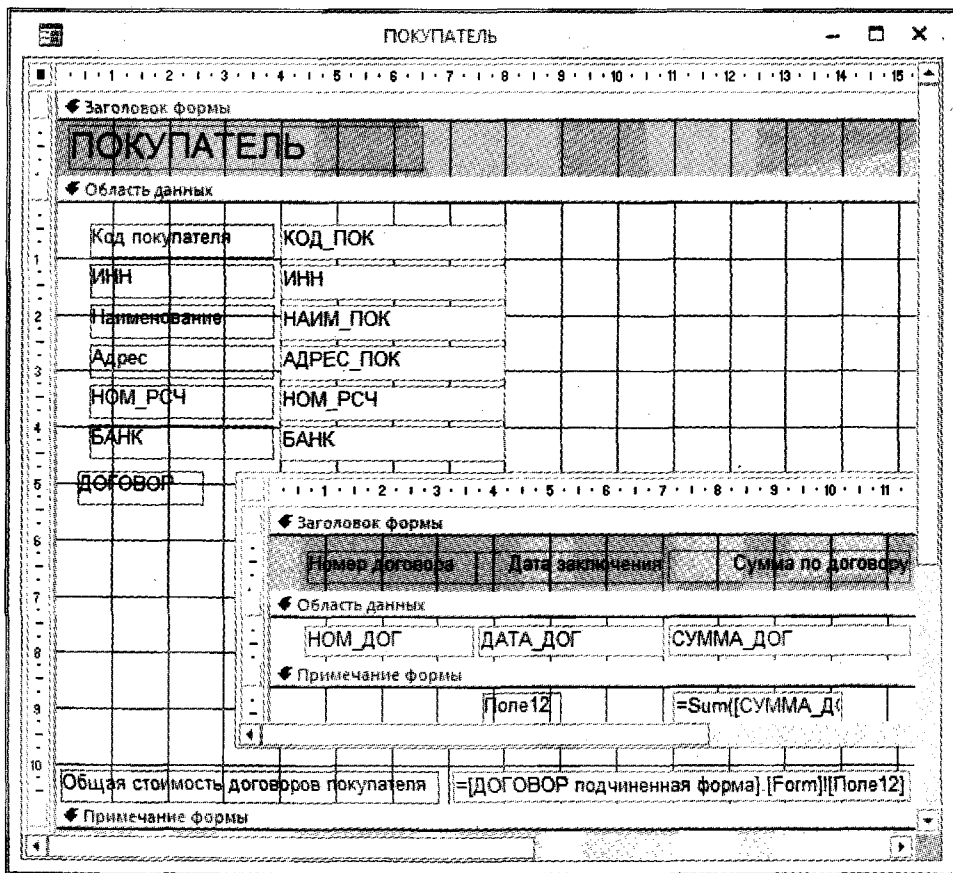


Рис. 5.34. Вычисление итогового значения в подчиненной форме и отображение его в главной форме

создайте вычисляемый элемент управления и запишите в него выражение (рис. 5.34):

```
=Sum([СУММА_ДОГ])
```

Для отображения результата вычисления в главной форме создайте там вычисляемый элемент управления и запишите в него выражение (см. рис. 5.34):

```
=[ДОГОВОР подчиненная форма].[Form]![Поле12]
```

Это выражение является ссылкой на содержащий общую стоимость договоров элемент управления Поле12 в примечании подчиненной формы. Общий формат такой ссылки имеет вид:

```
=[Имя подчиненной формы].[Form]![Имя поля в подчиненной форме]
```

В надпись вычисляемого элемента управления введите: Общая стоимость договоров покупателя. В свойстве **Формат поля (Format)** на вкладке **Макет (Format)** измените значение на **Денежный (Currency)**.

Чтобы не отображалось примечание подчиненной формы, установите для его свойства **Вывод на экран (Display When)** значение **Нет**.

Форма с вычисляемым элементом управления, отображающим в главной форме общую стоимость договоров покупателя, рассчитанную в подчиненной форме, представлена на рис. 5.35.

The screenshot shows a Microsoft Access form titled "ПОКУПАТЕЛЬ". The form contains several fields for customer information and a table of contracts. The total contract value is displayed as 8,146,200.00p.

Код покупателя	П001
ИНН	778957651111
Наименование	Компьютер маркет
Адрес	Москва
НОМ_РСЧ	76358609763264536567
БАНК	Мост

Номер договора	Дата заключения	Сумма по договору
Д111	11.01.2013	6 414 000.00p.
Д222	05.02.2013	152 280.00p.
Д777	14.06.2013	1 579 920.00p.
		0.00p.

Общая стоимость договоров покупателя: 8 146 200.00p.

Рис. 5.35. Отображение в главной форме итогового значения, вычисленного в подчиненной форме

Ограничение доступа к данным через форму

Защита данных поля от изменений

Для защиты данных поля от изменения используется свойство **Блокировка** (Locked). Блокировка может быть установлена для любого поля формы. Чтобы защитить поле, надо открыть форму в режиме макета или конструктора, установить курсор в его рамке и, например, с помощью контекстно-зависимого меню вызвать свойства поля. В окне свойств на вкладке **Данные** (Data) в строке **Блокировка** (Locked) выберите **Да** (Yes). По умолчанию для всех полей в свойстве **Блокировка** (Locked) устанавливается значение **Нет** (No). После установки этого свойства поле доступно только для чтения.

Установка ограничений на корректировку записей через форму

Для того чтобы при работе через форму записи были доступны только для чтения, следует в свойствах формы на вкладке **Данные** (Data) (рис. 5.36) в строках **Разрешить добавление** (Allow Additions), **Разрешить удаление** (Allow Deletions) и **Разрешить изменение** (Allow Edits) задать значения **Нет** (No). Сделать записи доступными только для чтения можно также, выбрав для свойства **Тип набора записей** (Recordset Type) значение **Статический набор** (Snapshot). Указанные свойства могут устанавливаться независимо друг от друга. Например, при запрете на изменение записей может быть разрешено добавление и удаление записей.

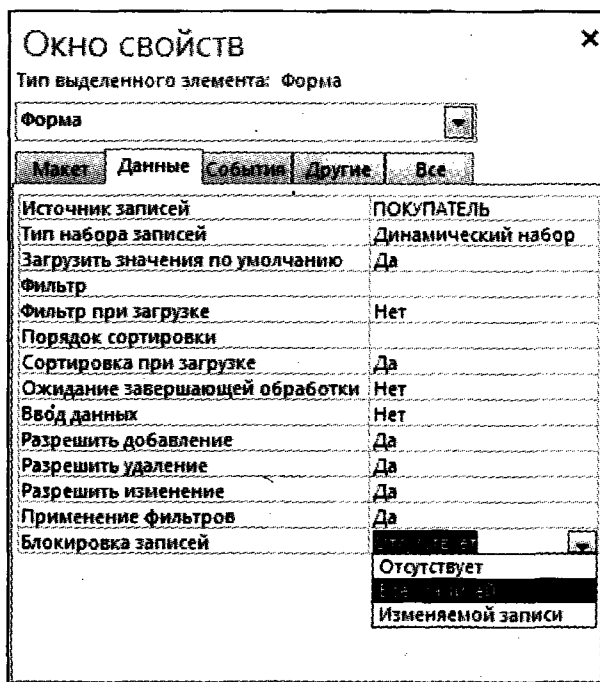


Рис. 5.36. Установка ограничений на корректировку записей

ЗАМЕЧАНИЕ

Свойство формы **Блокировка записей** (Record Locks) определяет способы блокировки записей при попытке двух пользователей одновременно изменить записи в некотором наборе записей сетевой базы данных. Когда один пользователь изменяет запись, автоматически может блокироваться только изменяемая запись или все записи набора. При этом другие пользователи не могут изменять или только одну запись, или весь набор до завершения работы с ней первого пользователя.

Свойство **Ввод данных** (Data Entry) при значении **Да** определяет режим формы, при котором разрешен только ввод новых записей, просмотр существующих записей при этом недоступен. При открытии формы выводится пустая запись, с которой можно начать ввод новых записей. С вновь введенными записями можно выполнять все разрешенные действия: удалять, изменять их. При новом открытии формы снова делается доступным только режим ввода новых записей.

Защита данных подчиненной формы от изменений

Защитить содержимое всех полей подчиненной формы, сделав его доступным только для чтения, позволяет свойство **Блокировка** (Locked), установленное для элемента управления **Подчиненная форма/отчет** (Subform/Subreport).

Выделите элемент **Подчиненная форма/отчет** (Subform/Subreport). В окне свойств на вкладке **Данные** (Data) в строке **Блокировка** (Locked) выберите **Да** (Yes). При этом станет невозможным изменение, добавление и удаление данных в полях подчиненной формы. Кроме того, станет невозможным добавление новых записей.

Выделите элемент **Подчиненная форма/отчет** (Subform/Subreport). Установите для свойства **Включена** (Enabled) (в предыдущей версии **Доступ**) значение **Нет**. Это приведет не только к запрету на любые изменения, но и к невозможности перемещаться по записям, просматривать записи за пределами рамки подчиненной формы (см. рис. 5.33).

При выделении подчиненной формы в ее свойствах, как и в свойствах любой другой формы, можно установить ограничения на корректировку записей через форму (см. рис. 5.36).

Контрольные вопросы

1. Может ли форма, созданная мастером, содержать данные из нескольких таблиц?
2. Как в форме установить защиту от изменения данных поля?
3. В каком разделе свойств формы определяются основные параметры ее вида?
4. Какое свойство обеспечивает открытие формы в режиме добавления записей, при котором не возможен просмотр ранее введенных записей?
5. Какой внешний вид подчиненной формы нужно выбрать, чтобы в ней выводились надписи полей, определенные в свойствах таблиц?
6. Какой элемент управления позволяет создать вычисляемое поле в форме?
7. С какого знака начинается выражение, записываемое в вычисляемое поле?

8. Как вызвать построитель для формирования выражения в вычисляемом поле?
9. В каком разделе формы размещается вычисляемое поле для расчета итогового значения для подчиненной формы?
10. Можно ли при расчете итогового значения использовать в аргументе функции Sum имя другого вычисляемого элемента управления?
11. Как отобразить в главной форме итоговое значение, рассчитанное в подчиненной форме?
12. Какой формат имеет ссылка на вычисляемое поле в подчиненной форме?
13. Сохраняется ли значение вычисляемого поля в таблице?
14. Что нужно сделать, если не работает связь основной и подчиненной форм?
15. Должны ли поля, по которым устанавливается связь между основной и подчиненной формами, иметь одинаковые имена?
16. Может ли связь между основной и подчиненной формами устанавливаться по нескольким полям?
17. Можно ли путем перетаскивания таблицы из окна базы данных в форму создать в ней подчиненную форму?
18. Может ли мастер построить форму на основе несвязанных таблиц?
19. Какие записи отображаются в подчиненной форме?
20. Может ли мастер построить форму на основе ранее созданного запроса?
21. Можно ли при создании новой записи в форме СПРАВОЧНИК ТОВАРОВ не вводить значение кода товара?
22. Какие поля входят в список доступных в форме?
23. В каком элементе управления размещается подчиненная форма?
24. В каких свойствах указываются поля связи основной и подчиненной формы?
25. Какая команда позволяет создавать форму на основе выбранной таблицы?
26. Какая вкладка ленты активируется при открытии формы?
27. Откуда выбирается текст при автоматическом формировании надписи поля в форме?
28. Какие вкладки ленты появляются при переходе в режим конструирования формы?
29. Для чего предназначена группа **Элементы управления (Controls)** конструктора форм?
30. Как в режиме конструктора просмотреть доступные в форме поля?
31. Где размещена кнопка, по которой создается элемент управления **Надпись (Label)**?
32. По какой команде просматриваются свойства элемента управления?
33. Какая кнопка позволяет переключаться из режима конструктора в режим формы и обратно?

34. Можно ли в форме с помощью кнопок перехода по записям сделать текущей новую запись?
35. В каких отношениях, как правило, находятся таблицы — источники основной и подчиненной формы?
36. Какие поля записеобразующей таблицы должны быть обязательно включены в источник записей подчиненной формы, чтобы через нее можно было вводить новые записи?
37. Можно ли через многотабличную форму осуществить ввод данных сразу в несколько таблиц?

Ответы

1. Да.
2. В свойствах поля на вкладке **Данные (Data)** в строке **Блокировка (Locked)** выбрать значение **Да (Yes)**.
3. На вкладке **Макет (Format)**.
4. Свойство формы **Ввод данных (Data Entry)** на вкладке **Данные (Data)**.
5. **Ленточный (Tabular)**.
6. Элемент управления **Поле (Text Box)**, создаваемый по соответствующей кнопке на панели элементов.
7. Со знака равенства (=).
8. Открыть свойства поля и на вкладке **Данные (Data)** в конце строки **Данные (Control Source)** вызвать построитель выражений.
9. В области примечания формы.
10. Нет.
11. Создать новое вычисляемое поле в главной форме и разместить в нем ссылку на вычисляемое поле в подчиненной форме.
12. =[Имя подчиненной формы].[Форм]![Имя вычисляемого поля в подчиненной форме].
13. Нет.
14. Установить правильные значения в свойствах элемента управления **Подчиненная форма/отчет (Subform/Subreport)**, **Подчиненные поля (Link Child Fields)** и **Основные поля (Link Master Fields)**.
15. Нет.
16. Да.
17. Да, если установлен мастер подчиненных форм.
18. Нет.
19. Связанные с записью, отображаемой в главной форме.
20. Да.

21. Нет. Код товара является ключом в таблице ТОВАР.
22. Все поля, выбранные из таблиц или запросов, на основе которых формируется источник записей формы.
23. **Подчиненная форма/отчет** (Subform/ Subreport).
24. В свойствах элемента управления **Подчиненная форма/отчет** (Subform/ Subreport): **Подчиненные поля** (Link Child Fields) и **Основные поля** (Link Master Fields).
25. **Форма** (Form), **Несколько элементов** (Multiple Items), **Разделенная форма** (Split Form).
26. **Главная** (Home).
27. Из соответствующего свойства поля, заданного при определении структуры таблицы.
28. Три вкладки, объединенные заголовком **Инструменты конструктора форм** (Form Design Tools): **Конструктор** (Design), **Упорядочить** (Arrange), **Формат** (Format).
29. Для создания в форме полей и других элементов управления.
30. По кнопке **Добавить поля** (Add Existing Fields) на лентах **Инструменты конструктора форм** или **Работа с макетами форм | Конструктор** (Form Design Tools | Design) в группе **Сервис** (Tools).
31. На лентах **Инструменты конструктора форм** или **Работа с макетами форм | Конструктор** (Form Design Tools | Design) в группе **Элементы управления** (Controls).
32. **Страница свойств** (Property Sheet).
33. Кнопка **Режим** (View) на вкладке ленты **Главная** (Home) или **Конструктор** (Design), а также в строке состояния.
34. Да.
35. Один-ко-многим.
36. Все поля, составляющие ключ, и поля, обязательные для заполнения.
37. Да.

ГЛАВА 6



Разработка интерфейса для ввода, просмотра и корректировки документов

Диалоговые приложения пользователя баз данных, прежде всего, должны обеспечивать удобный графический интерфейс для работы с документами предметной области. Основным средством создания такого интерфейса являются *формы* (см. главу 5). Формы, адекватные формам первичных документов, позволяют выполнять загрузку справочных, плановых и оперативно-учетных данных, в любой момент просматривать и редактировать содержимое ранее введенных в базу данных документов. Формы могут использоваться и для оформления документов на компьютере — например, накладных на отгрузку товара, счетов на оплату в соответствии с договорами и т. д. В процессе оформления документа выполняется и ввод его в базу данных. Кроме того, может быть предусмотрена распечатка подготовленного документа средствами отчетов.

Инструментарий разработки форм предоставляет широкие возможности по созданию графического диалогового интерфейса пользователя для работы с документами, сохраняемыми в базе данных. Такой интерфейс является основой работы с базой данных в практическом приложении пользователя. После окончательного создания приложения пользователь, как правило, не работает непосредственно с таблицами базы данных. Да и разработчик приложения часто ограничивает полностью или частично непосредственный доступ пользователя к таблицам.

При конструировании интерфейса пользователя целесообразно использование определенного для формы и ее элементов управления набора типовых событий. Эти события инициируются пользователем в процессе работы с формой. Для обработки событий могут применяться макросы или процедуры VBA. Это позволяет вместо разработки и выполнения больших программ приложения разрабатывать и запускать небольшие макросы или процедуры обработки событий при возникновении некоторых событий (нажатие кнопки, выбор в списке и т. п.). Таким образом создается приложение, управляемое действиями пользователя и реализующее связь алгоритмов небольших программ в единый алгоритм, реализующий комплекс задач приложения. Например, связанные с полем события, такие как **Изменение** (On Change), **До обновления** (Before Update), **После обновления** (After Update), или

связанные с записью события, такие как Удаление (Delete), До обновления (Before Update), После обновления (After Update), могут быть использованы для подключения процедур, обеспечивающих автоматический перерасчет (изменение) показателей в таблицах.

При создании некоторых элементов формы автоматически формируются типовые процедуры обработки событий. Таким элементом является, например, кнопка. С ее событием **Нажатие кнопки** (On Click) связывается выполнение таких категорий действий, как:

- переходы по записям источника формы, обработка записей (добавление, удаление, печать, восстановление);
- работа с формой (закрытие, открытие других форм, изменение фильтра, обновление данных, печать формы);
- работа с отчетом (печать, просмотр, отправка, вывод в файл);
- запуск запроса, макроса, печать таблицы, автонабор номера.

Этапы разработки интерфейса

Технология создания целостной базы, в которой между таблицами установлены связи и определены параметры поддержания целостности, предполагает упорядочение первоначальной загрузки взаимосвязанных таблиц.

Технология поддержания такой базы данных в актуальном состоянии требует обеспечения процесса ввода оперативных данных и обновления существующих данных.

При этом должен быть разработан удобный интерфейс пользователя, обеспечивающий важнейший аспект технологии работы с базой данных — однократный и корректный ввод взаимосвязанных данных. Использование экранных форм — электронных аналогов первичных документов, являющихся источниками для загрузки справочных, плановых и оперативных учетных данных, — позволяет решить эти задачи.

Прежде чем вводить, отображать или корректировать данные таблиц через экранную форму, надо ее спроектировать и сконструировать. Далее рассматриваются основы проектирования форм для создания удобного интерфейса пользователя, обеспечивающего работу с документами. Подробно описана технология разработки формы для первоначального ввода, просмотра и обновления документов в базе данных.

В процессе разработки технологии загрузки базы данных и проектирования форм целесообразно определить:

- перечень документов-источников, сохраняемых в базе и содержащих необходимые данные для загрузки таблиц базы данных;
- таблицы — объекты загрузки для каждого документа-источника;
- содержание и последовательность загрузки. При этом необходимо учитывать, что для обеспечения связной целостности главные таблицы должны быть загружены ранее подчиненных;

- подсхему данных каждой формы (фрагмент схемы данных), состоящую из таблиц, необходимых для создания электронного документа. При этом для много-табличной (составной) формы выбирается:
 - таблица, которая будет базовым источником записей главной формы, и таблицы для отображения справочных данных в этой части формы;
 - таблица, которая будет источником записей подчиненной формы, включаемой в главную форму, и таблицы для отображения справочных данных в подчиненной форме;
- макет формы, т. е. ее общую структуру, соответствующую структуре документа-источника и выбранной подсхеме данных. При этом распределяется пространство формы для размещения включаемых подчиненных форм;
- состав и размещение элементов, связанных с полями таблиц, и надписей для каждой из частей составной формы. При этом:
 - в главной форме обязательно надо выводить ключевые поля таблицы-источника данных, например, в форме ДОГОВОР — идентификатор документа (номер договора);
 - в подчиненной форме предусмотреть отображение только тех полей составного ключа таблицы, которых нет в таблице-источнике главной формы, например, в подчиненной договору форме ПОСТАВКА_ПЛАН — код товара и срок поставки.

ОБРАТИТЕ ВНИМАНИЕ

Источник данных подчиненной формы должен содержать все составляющие ключ поля, даже если они не отображаются в форме.

После выполнения перечисленных пунктов и получения макета формы можно приступить к разработке форм средствами Access.

Определение последовательности загрузки таблиц с документов

При разработке форм, обеспечивающих загрузку взаимосвязанных таблиц базы данных, следует соблюдать требования к последовательности загрузки записей в таблицы, которые диктуются их связями и установленными параметрами поддержания целостности. Эти требования можно сформулировать следующим образом:

- независимо могут создаваться записи таблиц, которые не подчинены каким-либо другим таблицам в схеме данных;
- запись таблицы, подчиненной каким-либо другим таблицам, может создаваться при наличии связанных с ней записей в главных таблицах, записи главной таблицы должны быть загружены ранее (таблицы справочных данных) или должны создаваться вместе с подчиненной записью в одной форме.

В соответствии с этими требованиями можно рекомендовать в практических приложениях предусмотреть сначала ввод в базу справочных данных, а затем данных плановых и оперативно-учетных документов. Это связано с тем, что таблицы с плановыми и оперативно-учетными данными в схеме данных являются подчиненными таблицам справочных данных, которые, как правило, находятся на верхнем уровне.

Рассмотрим технологию загрузки на примере базы данных «Поставка товара». Таблицы базы данных и связи между ними отображены в схеме данных, приведенной в *главе 2* на рис. 2.19.

Документы-источники загрузки базы данных «Поставка товара» перечислены при описании предметной области в *главе 2*. Определим объекты загрузки базы данных — взаимосвязанные таблицы, подлежащие загрузке с каждого документа предметной области, и последовательность их загрузки.

Справочная информация

Для документов справочной информации в базе данных «Поставка товаров» следует выделить следующие объекты загрузки:

- таблица ТОВАР — загрузка этой таблицы производится из документа «Справочник товаров», содержащего сведения о товарах, поставляемых фирмой;
- таблица СКЛАД — загрузка этой таблицы производится из документа «Справочник складов», содержащего сведения о складах фирмы;
- таблица ПОКУПАТЕЛЬ — загрузка этой таблицы производится из документа «Справочник покупателей», содержащего сведения о покупателях фирмы.

Таблицы справочной информации ПОКУПАТЕЛЬ, ТОВАР, СКЛАД на схеме данных находятся на верхнем уровне и не подчинены другим таблицам, поэтому их загрузка производится в любой последовательности.

Плановая информация

Из документа «Договор», содержащего условно-постоянную плановую информацию, целесообразно одновременно вводить не только общие сведения о договоре, но и данные о плановых поставках по договору. В соответствии с этим следует выделить единый объект загрузки: таблицы ДОГОВОР — ПОСТАВКА_ПЛАН. Загрузка записей этих таблиц производится одновременно из документа «Договор», что обеспечит формирование взаимосвязей записей этих таблиц. При этом обеспечивается однократный ввод значений идентификатора договора `НОМ_ДОГ` для всех товаров документа.

Загрузка таблицы ДОГОВОР может производиться после загрузки таблицы ПОКУПАТЕЛЬ, т. к. таблица ДОГОВОР в схеме данных подчинена таблице ПОКУПАТЕЛЬ.

Загрузка таблицы ПОСТАВКА_ПЛАН может производиться только после загрузки таблиц ДОГОВОР и ТОВАР, т. к. таблица ПОСТАВКА_ПЛАН подчинена этим таблицам.

Оперативно-учетная информация

Из документа «Накладная», содержащего оперативно-учетную информацию, как и в предыдущем случае, целесообразно одновременно вводить общие сведения о накладной и данные об отгрузках товара по накладной. В соответствии с этим следует выделить единый объект загрузки: таблицы НАКЛАДНАЯ — ОТГРУЗКА. Загрузка записей этих таблиц производится одновременно из документа «Накладная», что обеспечит формирование взаимосвязей записей этих таблиц. При этом осуществляется однократный ввод значений идентификатора накладной — НОМ_НАКЛ и КОД_СК для всех отгружаемых по накладной товаров.

Загрузка таблицы НАКЛАДНАЯ может производиться только после загрузки таблиц ДОГОВОР и СКЛАД, т. к. таблица НАКЛАДНАЯ в схеме данных подчинена этим таблицам.

Загрузка таблицы ОТГРУЗКА может производиться только после загрузки таблиц НАКЛАДНАЯ и ТОВАР, т. к. таблица ОТГРУЗКА подчинена этим таблицам.

ЗАМЕЧАНИЕ

Загрузка таблицы СКЛАД может быть осуществлена и после загрузки данных по договорам, поскольку ни по каким путям в схеме данных таблицы ДОГОВОР и ПОСТАВКА_ПЛАН не подчинены таблице СКЛАД.

Таким образом, определена последовательность этапов загрузки базы данных «Поставка товаров», а также объекты загрузки на отдельных этапах и соответствующие документы-источники данных. Технология загрузки базы данных «Поставка товаров» обобщена в табл. 6.1.

Таблица 6.1. Технология загрузки базы данных «Поставка товаров»

Таблицы БД — объекты загрузки	Документы-источники	Вид информации	Этап загрузки	Примечание
ПОКУПАТЕЛЬ	Справочник покупателей	Справочная	I	Независимые этапы
ТОВАР	Справочник товаров	Справочная	I	
СКЛАД	Справочник складов	Справочная	I или II	
ДОГОВОР — ПОСТАВКА_ПЛАН	Договоры	Плановая	II	
НАКЛАДНАЯ — ОТГРУЗКА	Накладные	Оперативно-учетная	III	

После определения этапов загрузки базы данных можно приступить к определению подсхемы данных для каждого этапа загрузки, к проектированию макета форм и их созданию средствами Access.

Проектирование интерфейса для ввода и корректировки документа

Ввод и корректировка справочных данных могут быть осуществлены через простые формы с макетом в столбец или табличный, в которых для проверки значений в полях заданы ограничения. Для ввода и корректировки данных плановых и оперативно-учетных документов пользователю нужно разработать удобный экраный интерфейс, который позволит минимизировать операции по вводу данных и контролировать их достоверность и корректность. При этом необходимо ограничиваться вводом только идентификаторов и количественных показателей. Справочные данные (наименования, нормативы, цены, тарифные ставки и т. п.) не могут вводиться с этих документов, а должны только отображаться в форме из ранее созданных таблиц справочной информации. Отображение справочных данных позволяет осуществлять визуальный контроль правильности вводимых с плановых или оперативно-учетных документов данных, в которых обычно присутствуют справочные данные.

Разработка интерфейса требует для каждого документа выполнить проектирование формы.

Рассмотрим процесс проектирования формы для ввода, просмотра и корректировки данных о договорах фирмы. Форма служит электронным документом, вид которого должен соответствовать виду бумажного документа. Вид документа «Договор» был приведен в *главе 2* на рис. 2.7.

В соответствии с этапами загрузки базы данных («Поставка товаров», определенными ранее (см. табл. 6.1), загрузка данных из документа «Договор» должна производиться в таблицы ДОГОВОР и ПОСТАВКА_ПЛАН после загрузки таблиц со справочными данными ПОКУПАТЕЛЬ и ТОВАР, что обеспечит установление связей загружаемых записей с соответствующими записями этих таблиц.

При проектировании формы определяется подсхема данных, включающая объект загрузки формы, и общая структура формы — проект макета и размещение реквизитов в соответствии со структурой документа «Договор» и подсхемой данных, учитываются особенности назначения и работы с формой.

Определение подсхемы данных

Выбор подсхемы данных для построения формы-аналога документа «Договор» (назовем ее ДОГОВОРЫ С ПОКУПАТЕЛЯМИ) определяется следующими соображениями.

- Загрузка данных по договорам должна производиться в таблицы ДОГОВОР и ПОСТАВКА_ПЛАН, находящиеся в отношении 1 : М, следовательно, эти таблицы — объекты загрузки — надо включить в подсхему данных формы.
- В форме должны отображаться справочные данные по покупателям и товарам, указанным в договоре, поэтому в подсхему надо включить также таблицы ПОКУПАТЕЛЬ и ТОВАР, главные по отношению к таблицам ДОГОВОР и ПОСТАВКА_ПЛАН.

- Так как форма обеспечивает загрузку двух таблиц, связанных отношением 1 : М, главная в отношении таблица ДОГОВОР должна быть источником записей основной формы, подчиненная ПОСТАВКА_ПЛАН — источником записей подчиненной формы. Для отображения справочных данных в основной форме должна использоваться таблица ПОКУПАТЕЛЬ. Для отображения справочных данных в подчиненной форме должна использоваться таблица ТОВАР.

Таким образом, подсьема данных для формы ввода/вывода договоров фирмы должна иметь вид, показанный на рис. 6.1.

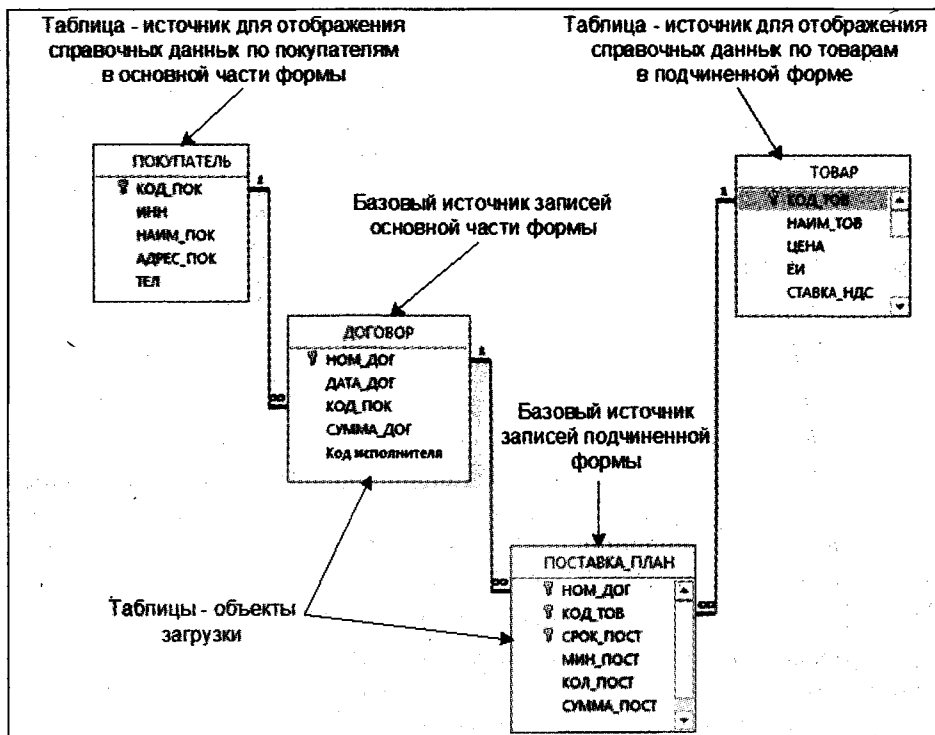


Рис. 6.1. Подсьема данных для формы ввода/вывода договоров фирмы

ОБРАТИТЕ ВНИМАНИЕ

В процессе проектирования базы данных (см. главу 2) все реквизиты документа были развиты на подмножества, составляющие таблицы базы данных. Например, реквизиты документа «Договор» были распределены по таблицам ДОГОВОР, ПОСТАВКА_ПЛАН, ПОКУПАТЕЛЬ и ТОВАР. Очевидно, для того чтобы форма ДОГОВОР отображала полный документ, ее подсьема данных должна включать все эти таблицы.

Разработка макета

Макет формы разрабатывается в соответствии со структурой документа и полученной подсьемой данных. Макет формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ приведен на рис. 6.2.

ДОГОВОР №

от

Покупатель

код адрес

наименование телефон

ИНН расчетный счет

банк

Код товара	Наименование	Ед изм	Цена	Срок поставки (месяц)	Мин. партия поставки	Количество	Сумма
<input type="text" value="КОД ТОВ"/>	<input type="text" value="НАИМ ТОВ"/>	<input type="text" value="ЕИ"/>	<input type="text" value="ЦЕНА"/>	<input type="text" value="СРОК ПОСТ"/>	<input type="text" value="МИН ПОСТ"/>	<input type="text" value="КОЛ ПОСТ"/>	<input type="text" value="СУММА ПОСТ"/>
ПОСТАВКА_ПЛАН	ТОВАР		ПОСТАВКА_ПЛАН				
Сумма всего							<input type="text" value="СУММА ДОГ (ДОГОВОР)"/>

Рис. 6.2. Проект макета формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ

В соответствии с определенными объектами загрузки многотабличная форма ДОГОВОРЫ С ПОКУПАТЕЛЯМИ должна состоять из двух форм: главной и включенной в нее подчиненной формы.

Источником записей главной формы станет ДОГОВОР, а таблица ПОКУПАТЕЛЬ будет использована для отображения справочной информации. Через эту часть многотабличной формы выполняется ввод, просмотр и корректировка общих сведений о договоре. Число доступных записей определяется количеством записей в таблице ДОГОВОР.

Источником записей подчиненной формы станет таблица ПОСТАВКА_ПЛАН, а таблица ТОВАР будет использована для отображения справочной информации. Через эту часть многотабличной формы выполняется ввод, просмотр и корректировка данных о плановых поставках по договорам. Общее число доступных записей определяется количеством записей в таблице ПОСТАВКА_ПЛАН. Одновременно в подчиненной форме должны отображаться только записи, связанные с открытым в главной форме договором.

В макете, как для главной, так и для подчиненной форм, определяется состав и размещение элементов документа, а также их связь с полями таблиц подсхемы данных:

- в основной форме обязательно должны размещаться: ключевое поле ном_дог и поле кода покупателя код_пок, по которому устанавливается связь с таблицей ПОКУПАТЕЛЬ. Эти поля должны быть взяты из таблицы ДОГОВОР — источ-

ника записей основной части формы. В противном случае нельзя будет ввести новую запись в таблицу ДОГОВОР;

- в подчиненной форме должно обеспечиваться формирование ключа таблицы ПОСТАВКА_ПЛАН: ном_дог, код_тов, срок_пост. В противном случае станет невозможным ввод новых записей. Однако в подчиненной форме можно разместить только те ключевые поля таблицы, которых нет в основной форме, т. е. обязательно нужно разместить только поля код_тов (код товара) и срок_пост (срок поставки) из таблицы ПОСТАВКА_ПЛАН, а поле ном_дог (номер договора) можно не размещать.

ВНИМАНИЕ!

Если поле код_тов выбрать из таблицы ТОВАР, то невозможно будет сформировать новую запись в таблице ПОСТАВКА_ПЛАН.

Многотабличная форма, соответствующая этому макету, обеспечит удобный интерфейс для ввода, просмотра и корректировки данных о договорах, позволит минимизировать операции по вводу данных и контролировать их достоверность и корректность. Вводить нужно только идентификаторы и количественные показатели. Справочные данные (наименования, цена и т. п.) не потребуются вводить из документа, они отобразятся в форме из ранее загруженных таблиц справочной информации. Отображение справочных данных позволяет осуществлять визуальный контроль правильности вводимых из документа данных.

После разработки макета можно приступить к созданию многотабличной формы средствами Access.

Создание интерфейса для ввода и корректировки документа

С помощью мастера форм создайте в соответствии с результатами проектирования форму ДОГОВОРЫ С ПОКУПАТЕЛЯМИ, состоящую из главной формы ДОГОВОР и включенной подчиненной формы ПЛАН ПОСТАВОК. Многотабличная форма после редактирования в конструкторе должна обеспечивать удобный ввод новых записей из документа «Договор» в таблицы ДОГОВОР и ПОСТАВКА_ПЛАН, просмотр и корректировку существующих договоров. Доступ к отображаемым в форме полям справочных данных о покупателе и товарах должен ограничиваться только чтением, т. к. значения этих полей не должны вводиться и корректироваться при вводе договоров.

Откройте базу данных «Поставка товаров». В области навигации выберите таблицу ДОГОВОР, которая будет служить источником записей основной формы. Для вызова мастера форм выполните на вкладке ленты Создание (Create) в группе Формы (Forms) команду Мастер форм (Form Wizard).

В открывшемся окне мастера Создание форм (Form Wizard) в списке Таблицы и запросы (Tables/Queries) будет отображена выбранная таблица ДОГОВОР и ее поля. Выберите из таблицы ДОГОВОР в списке Доступные поля (Available Fields) те

поля, которые вошли в спроектированный макет формы, перемещая их в область **Выбранные поля (Selected Fields)**.

Последовательно выбирайте из списка **Таблицы и запросы (Tables/Queries)** таблицы **ПОКУПАТЕЛЬ**, **ПОСТАВКА ПЛАН**, **ТОВАР** и включайте в область **Выбранные поля (Selected Fields)** нужные поля этих таблиц (рис. 6.3). Нажмите кнопку **Далее (Next)**.

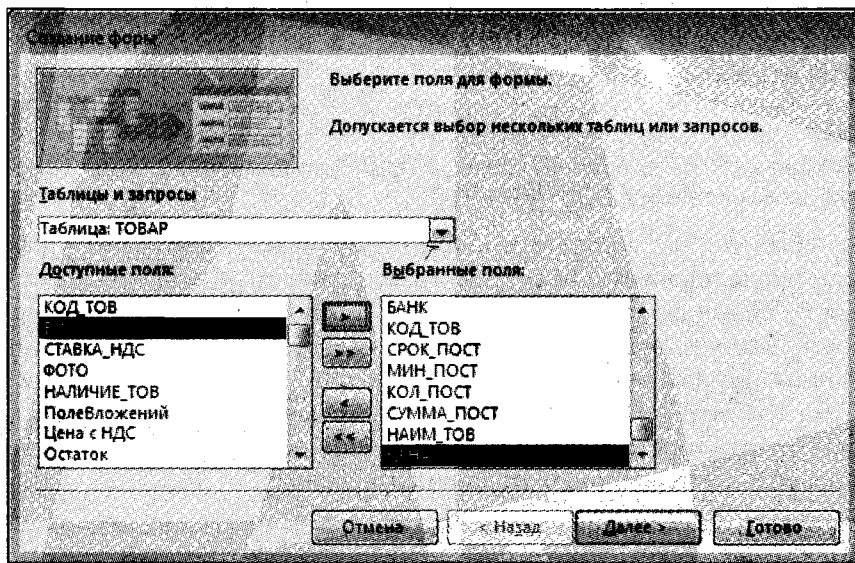


Рис. 6.3. Окно мастера форм при выборе полей из таблиц, составляющих подсхему формы

Для формирования основной формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** включите:

загружаемые поля таблицы **ДОГОВОР**:

- **НОМ_ДОГ** — номер договора (уникальный ключ);
- **ДАТА_ДОГ** — дата заключения договора;
- **КОД_ПОК** — код покупателя (поле связи);
- **СУММА_ДОГ** — сумма всего по договору;

поля из таблицы **ПОКУПАТЕЛЬ**, позволяющие отобразить справочную информацию о покупателе, с которым заключен договор:

- **НАИМ_ПОК** — наименование покупателя;
- **ИНН**;
- **АДРЕС** и другие справочные реквизиты покупателя.

Для формирования подчиненной формы включите поля:

загружаемые поля таблицы **ПОСТАВКА ПЛАН**:

- **КОД_ТОВ** — код товара (часть уникального ключа);
- **СРОК_ПОСТ** — срок поставки (часть уникального ключа);

- МИН_ПОСТ — минимальная партия поставки;
- КОЛ_ПОСТ — плановое количество поставки товара;
- СУММ_ПОСТ — сумма поставки товара.

Для формирования записи таблицы ПОСТАВКА_ПЛАН помимо ввода значений в поля КОД_ТОВ и СРОК_ПОСТ необходим ввод значения в поле НОМ_ДОГ, также входящего в составной ключ таблицы. Однако Access сам включает это загружаемое поле в источник записей подчиненной формы и при вводе новых записей обеспечивает его заполнение значением из главной таблицы;

□ поля из таблицы ТОВАР, позволяющие отобразить справочную информацию о товарах, поставляемых по договору:

- НАИМ_ТОВ — наименование товара;
- ЕИ — единица измерения;
- ЦЕНА — цена товара.

В следующем диалоговом окне мастера (рис. 6.4) уже выделена таблица ДОГОВОР, предлагаемая в качестве базового источника для создания основной формы, и схематично отображен макет формы с перечнем полей в основной и подчиненной форме. Кроме того, мастером отмечен переключатель **Подчиненные формы** (Form with subform(s)) — вариант непосредственного включения подчиненной формы в главную. Переключатель **Связанные формы** (Linked forms) определяет вызов подчиненной формы по кнопке.

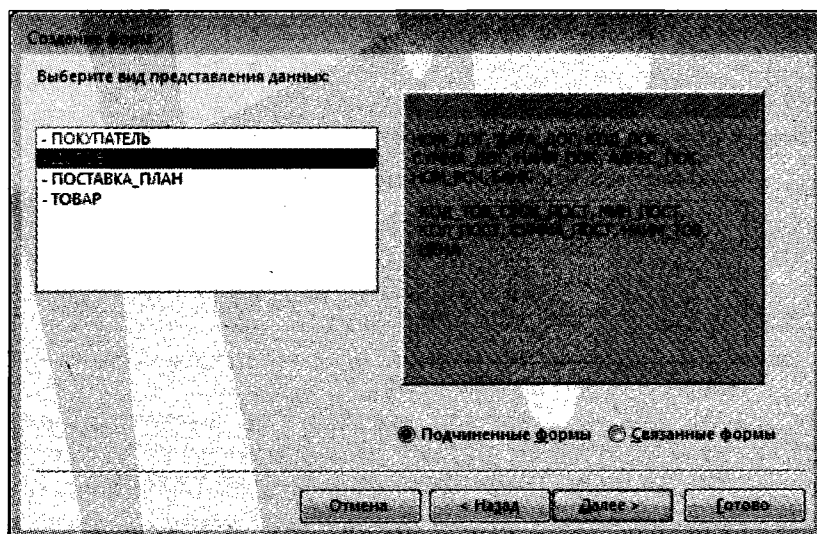


Рис. 6.4. Окно мастера форм при выборе вида представления данных

Для получения многозаписевой подчиненной формы с отображением полей каждой записи в одной строке, а надписей (определенных в свойствах таблиц) в заголовках столбцов, выберите в следующем окне мастера вид формы **Ленточный** (Tabular).

В последнем окне мастера задайте имя главной формы — **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**, а также имя подчиненной формы — **ПЛАН ПОСТАВОК**. Выберите дальнейшие действия мастера — **Открыть форму для просмотра и ввода данных** (Open the form to view or enter information). Это позволит после завершения работы мастера сразу приступить к работе с договорами.

Для завершения создания формы мастером нажмите кнопку **Готово** (Finish). Сохранение формы производится автоматически.

В созданной форме (рис. 6.5) выводятся данные по договорам, которые были ранее загружены в базу. Главная форма отображает сведения из одной текущей записи таблицы **ДОГОВОР**, в которую добавлены справочные данные из таблицы **ПОКУПАТЕЛЬ**. В подчиненной форме отображаются записи плановых поставок товаров, связанные с текущей записью главной формы. Каждая запись в подчиненной форме образуется на основе одной записи базового источника — таблицы **ПОСТАВКА_ПЛАН**, в которую добавляются справочные данные из таблицы **ТОВАР**.

ДОГОВОРЫ С ПОКУПАТЕЛЯМИ

Номер договора: Д333 Наименование: Перспектива

Дата заключения: 01.01.2016 ТЕЛ: (995)345-67-89

Код покупателя: П002 Адрес: Москва

ИНН: 789689798798 НОМ_РСЧ: 29384789823942938489

Сумма по договору: 74 920,00р. БАНК: Кредит

ПЛАН ПОСТАВОК

Код товара	Срок поставки	Наименование товара	Цена	Количество	СУММА ПОСТ
T001	3	Монитор 17LG	7 000,00р.	50	350 000,00р.
T002	2	FDD 3,5	500,00р.	80	4 000,00р.
T005	1	CD-ROM Panasonic	800,00р.	40	32 000,00р.
T005	2	CD-ROM Panasonic	800,00р.	10	8 000,00р.
*					

Запись: N 4 из 4 Поиск

Запись: N 4 из 7 Поиск

Рис. 6.5. Форма с подчиненной формой, созданная мастером

Доработка интерфейса

Чтобы подготовить более удобный интерфейс для работы с документом «Договор», соответствующий макету формы, отредактируйте созданную мастером форму средствами конструктора.

Редактирование формы средствами конструктора

Откройте созданную мастером многотабличную форму **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** в режиме конструктора. Для этого выберите ее в области навигации и в контекстном меню выполните команду **Конструктор (Design)**. Если форма была открыта, то для перехода в режим конструктора выберите **Конструктор (Design View)** из списка кнопки **Режим (View)** на вкладке ленты **Главная (Home)**.

В главной форме переместите и отредактируйте поля, чтобы получилось так, как показано на рис. 6.6. Уточните текст надписей полей, шрифт и размеры полей и надписей.

ДОГОВОРЫ С ПОКУПАТЕЛЯМИ

Закрыть форму **ДОГОВОР № Д333** от 01.01.2016 Следующая запись
Предыдущая запись

Покупатель **Перспектива**

Код покупателя: T002 Телефон: (995)345-87-89
 ИНН: 789889798798 Номер счета: 29384789823942938489
 Адрес: Москва Банк: Кредит

Код товара	Срок поставки	Наименование товара	Цена	Количество	Сумма поставки
T001	3	Монитор 17LG	7 000,00р.	50	350 000,00р.
T002	2	FDD 3,5	500,00р.	80	4 000,00р.
T005	1	CD-ROM Panasonic	800,00р.	40	32 000,00р.
T005	2	CD-ROM Panasonic	800,00р.	10	8 000,00р.
*					

Забить | 1 из 4 | Поиск

Сумма по договору 74 920,00р.

Забить | 5 из 7 | Поиск

Рис. 6.6. Отредактированная форма документа «Договор» в режиме просмотра

Чтобы увеличить рабочее пространство, уберите с экрана область навигации, щелкнув на кнопке **Открыть/закрыть границу области переходов (Shutter Bar Open/Close Button)** в ее правом верхнем углу. Для редактирования внешнего вида элементов формы переключайтесь в режим макета, т. к. в этом режиме можно сразу видеть результаты выполнения многих операций.

Для упорядочения, изменения размеров, перемещения элементов формы удобно объединять их в макет. Если элементы управления формы не включены в макет, для образования нового макета выделите нужные элементы и выполните команду контекстного меню **Макет | В столбик (Layout | Stacked)** или **Табличный (Tabular)**.

Изменение отдельного элемента требует удаления его из макета. Выделите нужный элемент вместе с надписью и выполните команду **Макет | Удалить (Layout | Remove)** в контекстном меню. Для перемещения макета выделите его, щелкнув в левом верхнем углу области макета или выполните команду **Выбрать макет (Select Layout)** на вкладке ленты **Упорядочить (Arrange)**, после этого макет можно перетащить в любое место. Команда **Выбрать макет** доступна, если макет есть в форме, и курсор установлен на одном из его элементов.

Для придания элементам одинакового вида удобно копировать форматирование одного элемента и применять его к другим элементам. Это позволяет сделать команда **Формат по образцу (Format Painter)**, размещенная на вкладках **Главная (Home)** или **Формат (Format)** в группе **Шрифт (Font)**. Если на значке команды **Формат по образцу** щелкнуть двойным щелчком, скопированный формат можно применять несколько раз. Закончив выполнение форматирования по образцу, повторно щелкните на значке команды.

Измените порядок отображения полей в подчиненной форме. Для того чтобы было удобно изменять размер, перемещать, удалять и добавлять поля, образуйте из них табличный макет. Тогда при выполнении любого из перечисленных действий все другие поля макета автоматически встают на новые места (сдвигаются или наоборот раздвигаются), сохраняя правильное размещение в строке.

Удалите элемент с надписью подчиненной формы. Для этого выделите надпись и сначала удалите ее из макета, выполнив в контекстном меню команду **Макет | Удалить (Layout | Remove)**, затем в контекстном меню выполните команду **Удалить (Delete)**.

Просмотрите источник записей главной формы. Для этого откройте свойства формы, выполнив соответствующую команду в контекстном меню. Открыть свойства формы также можно, выбрав в окне свойств в списке поля **Тип выделенного элемента (Selection Type)** строку **Форма (Form)**. На вкладке **Данные (Data)** в строке **Источник записей (Record Source)** представлена инструкция SQL, записанная мастером в качестве источника записей главной формы. Чтобы просмотреть инструкцию SQL в режиме конструктора запросов, нажмите значок построителя в конце строки. Соответствующий запрос в режиме конструктора показан на рис. 6.7.

Аналогичным образом просмотрите, какая таблица является источником записей подчиненной формы. Инструкция SQL в режиме конструктора запросов, являющаяся источником записей подчиненной формы, показана на рис. 6.8.

Если в главную или подчиненную форму необходимо добавить поля, представленные в источнике записей, установите курсор на соответствующей форме, нажмите кнопку **Добавить поля (Add Existing Fields)** на вкладке ленты **Конструктор (Design)** и из открывшегося списка перетащите нужное поле в форму.

В список полей включены только те поля таблиц, которые были выбраны при работе мастера или в запросе. Для добавления недостающих полей в источник записей формы, главной или подчиненной формы, можно изменить просмотренные запросы (см. рис. 6.7 и 6.8).

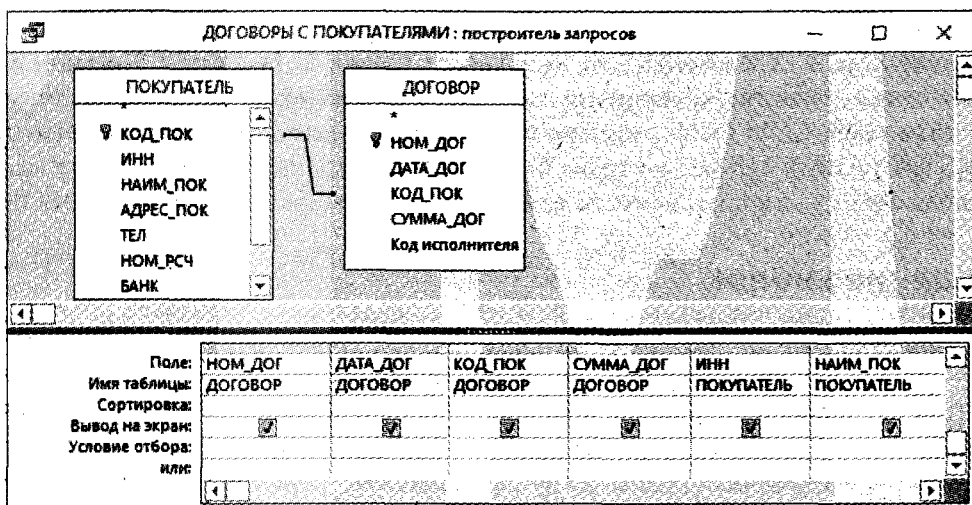


Рис. 6.7. Запрос — источник записей основной формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ

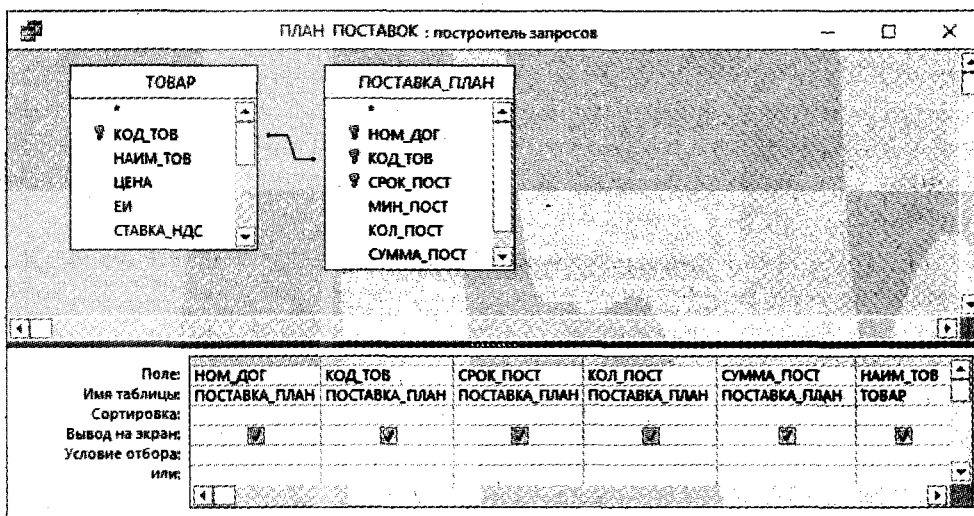


Рис. 6.8. Запрос — источник записей подчиненной формы ПЛАН ПОСТАВОК

Однако проще щелкнуть в области **Список полей (Field List)** на значке **Показать все таблицы (Show All tables)** — в разделе **Поля, доступные в данном представлении (Fields available for this view)** отобразятся все поля таблиц, включенных в подсхему данных формы (главной или подчиненной). Кроме того, в других разделах отобразятся остальные таблицы базы данных. Перетаскивая поля из любого раздела области, вы автоматически меняете запрос источника записей формы. Для отображения этого запроса в графическом виде также можно не открывать окно свойств и затем построитель запросов. Достаточно выполнить команду **Изменение источника записи (Edit Record Source)** в контекстном меню области **Список полей (Field List)**.

Проверьте связь подчиненной формы с главной. Для этого откройте свойства подчиненной формы, предварительно выделив ее рамку. В свойствах **Подчиненная форма/отчет (Subform/Subreport)** на вкладке **Данные (Data)** в строке **Подчиненные поля (Link Child Fields)** указано имя поля связи **НОМ_ДОГ** из подчиненной таблицы **ПОСТАВКА_ПЛАН**, в строке **Основные поля (Link Master Fields)** указано имя поля связи — в нашем примере тоже **НОМ_ДОГ** из главной таблицы **ДОГОВОР**.

Создание кнопок

Дополните форму новыми элементами управления — кнопками. Создайте в области заголовка главной формы кнопки для перехода к следующему и предыдущему договору (т. е. для перехода к другой записи источника главной формы таблицы **ДОГОВОР**).

В режиме конструктора нажмите в группе **Элементы управления (Controls)** кнопку **Использовать мастера (Use Control Wizards)**, а затем **Кнопка (Button)**. Не отпуская курсора, перенесите кнопку в нужное место и вычертите ее рамку. Запустится мастер создания кнопок (**Command Button Wizard**) (рис. 6.8).

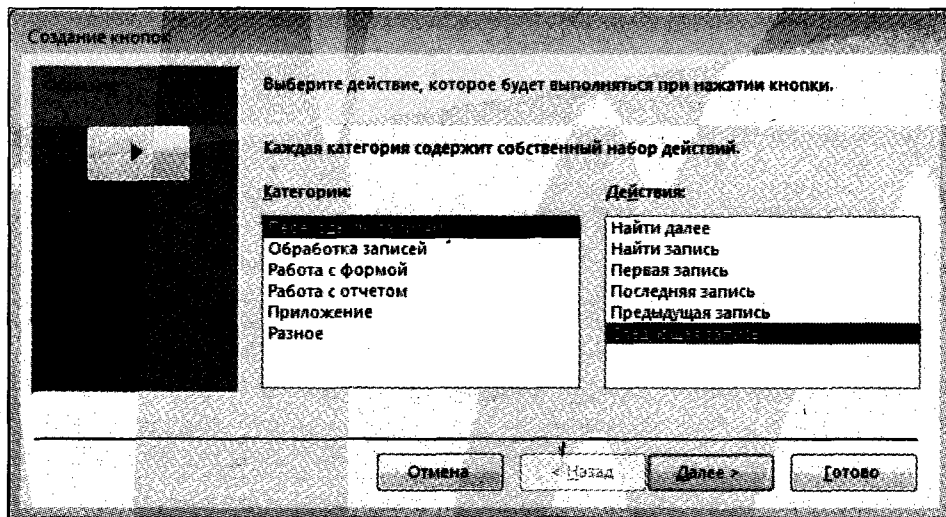


Рис. 6.9. Окно мастера создания кнопок

В окне мастера кнопок в группе **Категории (Categories)** выберите строку **Переходы по записям (Record Navigation)**, в группе **Действия (Actions)** — действие, которое должно выполняться при нажатии кнопки **Предыдущая запись (Go to Previous Record)**.

В следующем диалоговом окне мастера выберите вид кнопки с текстом на ней **Текст (Text)** или с рисунком **Рисунок (Picture)** и выберите рисунок из списка. После завершения работы мастера кнопка с выбранным текстом или рисунком встраивается в форму.

Аналогичные действия произведите для встраивания кнопки перехода к следующей записи, выбрав соответственно в области Действия (Actions) — Следующая запись (Go to Next Record). Размер и надписи кнопок редактируются, как и другие элементы. Создайте также кнопку для закрытия формы.

Отредактированная форма в режиме просмотра приведена на рис. 6.6.

Ограничение доступа к данным таблиц

Поля таблиц справочной информации ПОКУПАТЕЛЬ и ТОВАР должны использоваться в форме ДОГОВОРЫ С ПОКУПАТЕЛЯМИ только для отображения. Поэтому целесообразно защитить их от произвольных изменений при работе с формой. В главной форме такая защита нужна для полей таблицы ПОКУПАТЕЛЬ (НАИМ_ПОК, ИНН, АДРЕС и др.), в подчиненной форме — для полей таблицы ТОВАР (НАИМ_ТОВ, ЦЕНА и др.).

Для защиты поля выделите его (можно выделить несколько полей) и откройте окно свойств. В окне свойств на вкладке Данные (Data) в строке Блокировка (Locked) выберите Да (Yes). После установки этого свойства поле доступно только для чтения.

Если необходимо установить режим, при котором возможно только добавление новых договоров в базу данных и запрещен просмотр существующих, откройте свойства формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ и на вкладке Данные (Data) в строке Ввод данных (Data Entry) выберите значение Да.

Создание полей со списком

Как уже было сказано ранее, при загрузке договоров через разработанную форму обязательно должны вводиться ключевые поля таблиц — объектов загрузки — и ключи связи с таблицами справочных данных. Для удобства ввода значений в поля связи целесообразно соответствующие элементы управления создавать как Поле со списком (Combo Box). Поле со списком объединяет поле формы, в которое нужно ввести данное, и список записей из связанной главной таблицы. В списке можно выбрать нужное значение и ввести его в поле формы (рис. 6.10).

При этом повышается достоверность вводимой информации и снижается вероятность ошибок при вводе. Отображение данных из справочных таблиц при вводе идентификатора обеспечивает ввод в поле связи загружаемой подчиненной записи только тех значений, которые присутствуют в записях главной таблицы. Это обеспечивает успешное завершение операции ввода при установленном параметре целостности для связи таблиц в схеме данных.

Создайте с помощью мастера поле со списком для ввода в таблицу ДОГОВОР только тех значений кода покупателя код_пок (см. рис. 6.10), которые есть в таблице ПОКУПАТЕЛЬ. Сразу после ввода через поле со списком кода покупателя в форме отобразятся все реквизиты покупателя — проверьте соответствие кода и наименования покупателя в документе «Договор».

После ввода кода покупателя в таблицу ДОГОВОР

Закреть форму

ДОГОВОР № Д111 от 11.01.2016

Следующая запись

Предыдущая запись

Покупатель: **Компьютер маркет**

Код покупателя: П001

ИНН:

Адрес:

Телефон: (812)345-23-45

Номер счета: 76358508763264536567

Банк: Мост

Код товара	Срок поставки	Товар	Цена	Количество	Сумма постав
T001	1	Монитор 17LG	7 000,00р	305	2 135 000,00

Список записей из таблицы ПОКУПАТЕЛЬ

Список записей справочной таблицы ПОКУПАТЕЛЬ для ввода кода в запись таблицы ДОГОВОР

Рис. 6.10. Поле со списком записей справочной таблицы ПОКУПАТЕЛЬ для ввода кода в запись таблицы ДОГОВОР

Создание поля со списком мастером

В открытой в режиме макета или конструктора форме нажмите в группе **Элементы управления (Controls)** кнопку **Использовать мастера (Use Control Wizards)**, а затем кнопку **Поле со списком (Combo Box)**, переместите курсор в нужное место, нажмите кнопку мыши и, не отпуская ее, вычертите рамку элемента. После того как вы отпустите кнопку мыши, запустится мастер и откроется диалоговое окно **Создание полей со списком (Combo Box Wizard)**.

В окне мастера определите способ, которым список поля получает свои значения. Для формирования списка из связанной главной таблицы выберите **Объект "поле со списком" получит значения из другой таблицы или другого запроса (I want the combo box to look up the values in a table or query)** (рис. 6.11).

В следующем окне выберите таблицу **ПОКУПАТЕЛЬ**, которая будет поставлять значения в список поля (рис. 6.12).

Затем выберите ключевое поле **КОД_ПОК**, а также поле **НАИМ_ПОК** для отображения кода и наименования в строках списка (рис. 6.13).

В следующем окне мастера выберите порядок сортировки значений в списке. Далее настройте ширину столбцов списка и вариант отображения ключевого столбца (рис. 6.14).

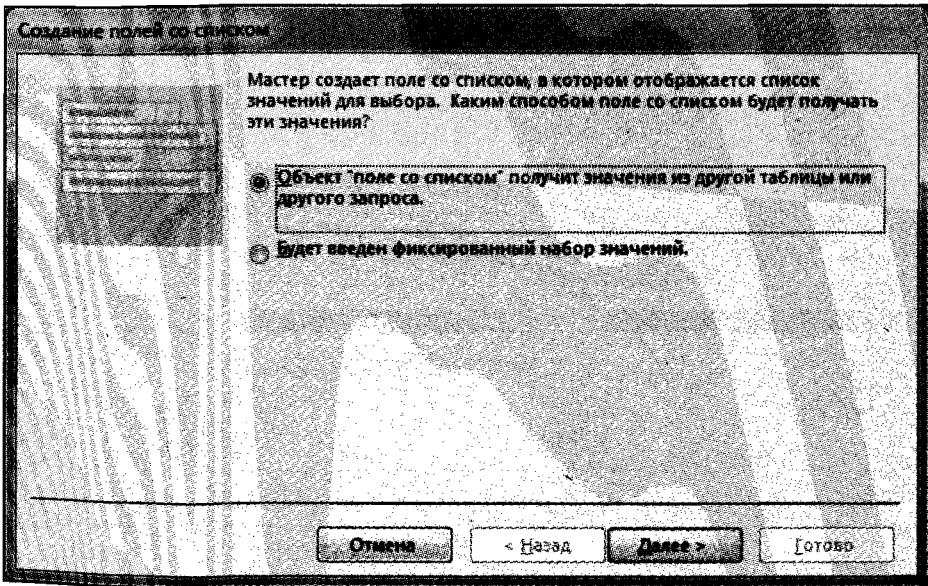


Рис. 6.11. Выбор способа формирования списка

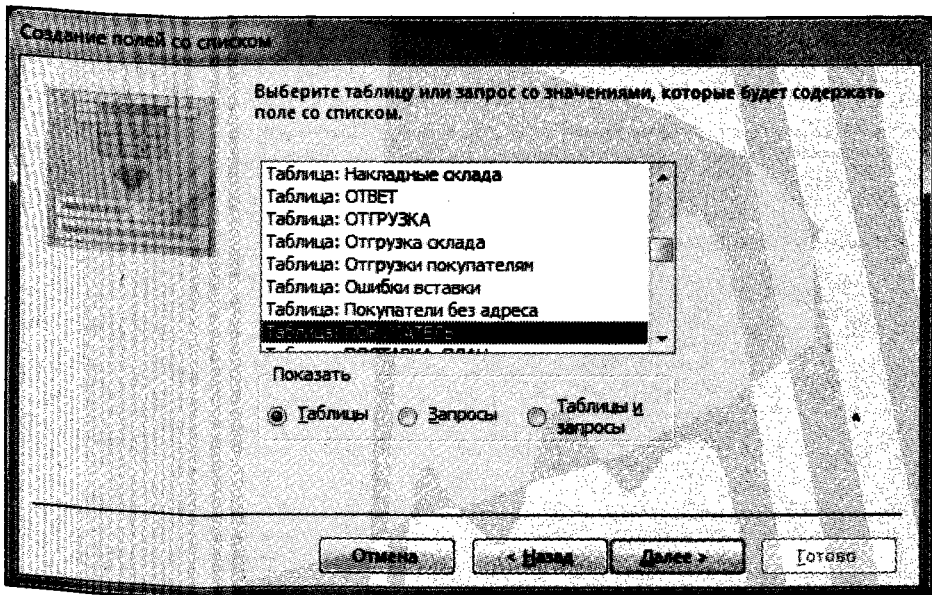


Рис. 6.12. Выбор источника данных для формирования списка

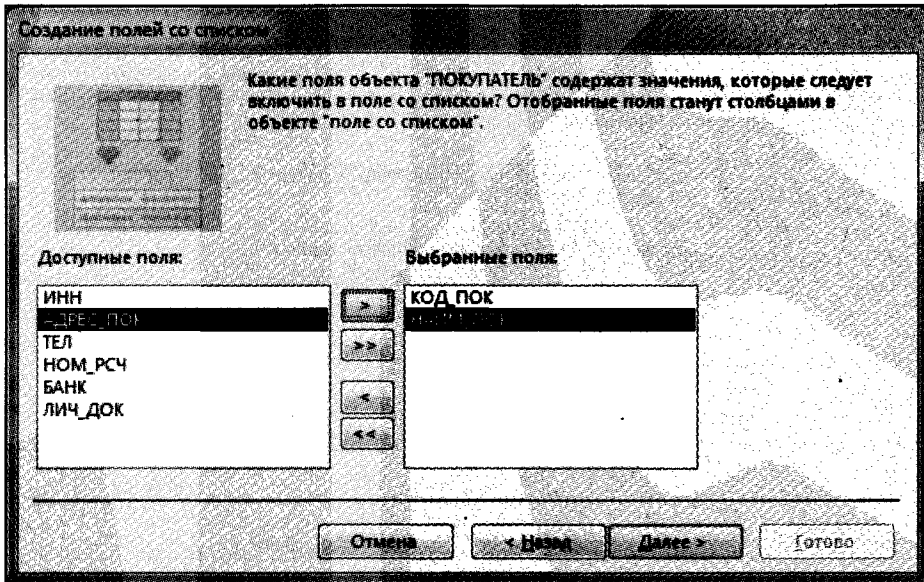


Рис. 6.13. Выбор полей для формирования записи списка

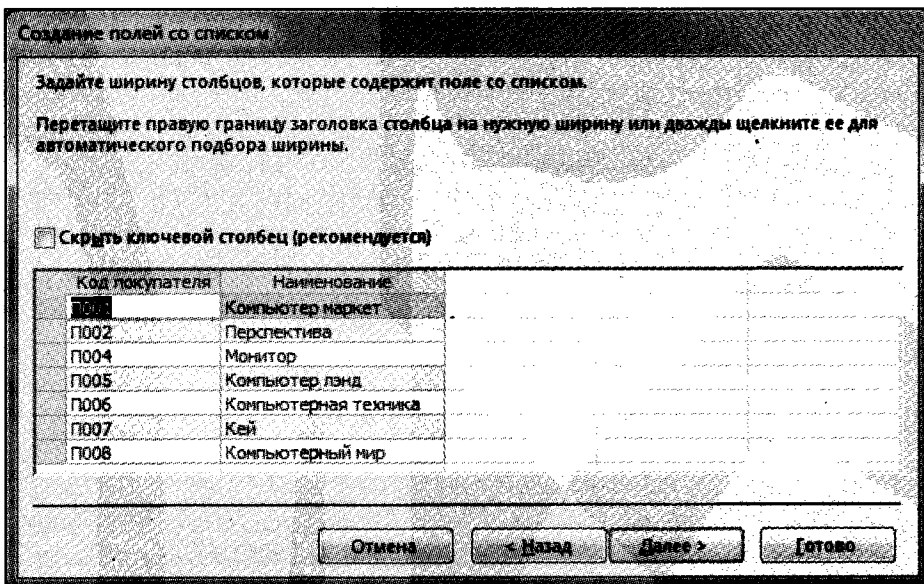


Рис. 6.14. Окно для оформления столбцов списка

Затем выберите поле списка код_пок, являющееся ключом связанной таблицы ПОКУПАТЕЛЬ. Из этого поля будет выбираться значение для ввода в поле формы и сохранения его в записи таблицы ДОГОВОР (рис. 6.15).

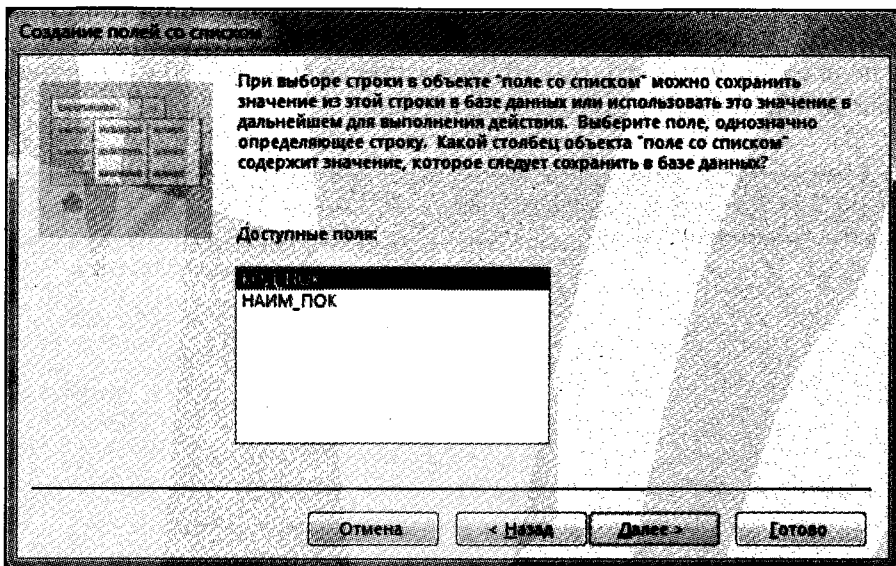


Рис. 6.15. Окно выбора поля источника значений для поля формы

В следующем окне (рис. 6.16) отметьте переключатель **Сохранить в поле** (Store that value in this field) и выберите поле формы код_пок (поле таблицы ДОГОВОР), в которое будут вводиться значения из списка.

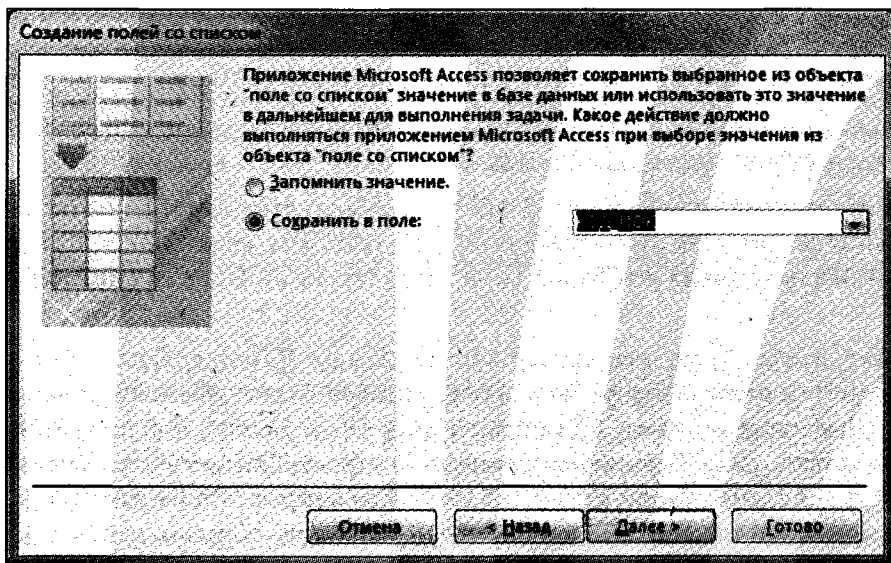


Рис. 6.16. Выбор поля формы, в которое вводится значение из списка

Далее введите надпись поля со списком: Код покупателя (рис. 6.17).

Нажмите кнопку **Готово** (Finish). В результате будет создано новое поле со списком КОД_ПОК. Замените поле с кодом покупателя на созданное мастером поле со списком. Использование поля со списком покупателей в режиме формы иллюстрирует рис. 6.10. Если потребуется настройка ширины столбцов списка, в окне свойств поля со списком в строках **Ширина списка** (List Width) и **Ширина столбцов** (Column Width) задайте подходящие значения.

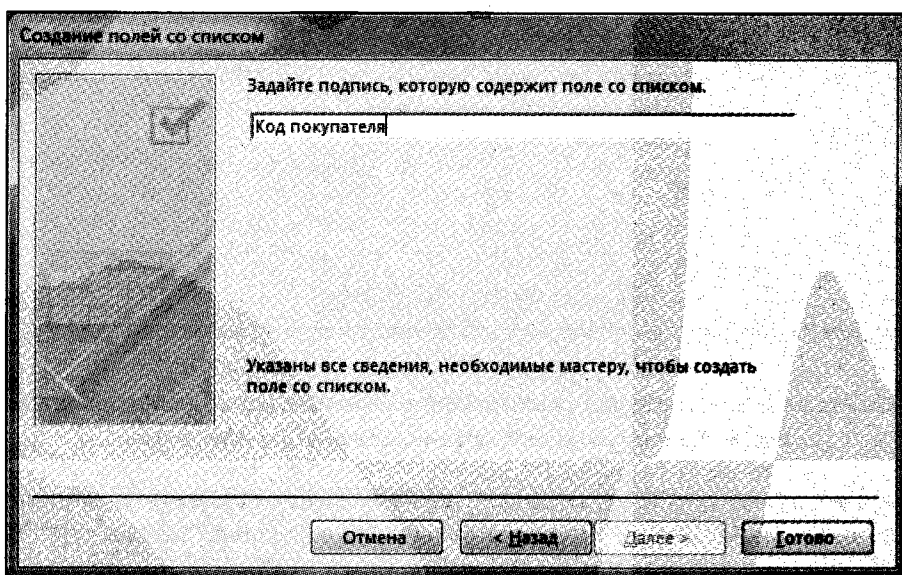


Рис. 6.17. Определение подписи поля со списком

Создание поля со списком в режиме конструктора

Мастер создает новое поле со списком. В режиме конструктора можно преобразовать существующее поле в поле со списком.

В подчиненной форме ПЛАН ПОСТАВОК преобразуйте поле КОД_ТОВ (код товара) в поле со списком. Для этого выберите в контекстном меню этого поля команду **Преобразовать элемент в** (Change To) и далее строку — **Поле со списком** (Combo Box).

Затем откройте свойства поля (рис. 6.18). На вкладке **Данные** (Data) в строке **Данные** (Control Source) будет указано поле, которое будет заполняться в форме — КОД_ТОВ. Выберите в строке **Источник строк** (Row Source) таблицу ТОВАР, из записей которой надо формировать список. В строке **Тип источника строк** (Row Source Type) по умолчанию стоит **Таблица или запрос** (Table/Query), а в строке **Присоединенный столбец** (Bound Column) — 1. Таким образом, чтобы сформировать поле со списком, может оказаться достаточным преобразовать обычное поле и выбрать его источник строк.

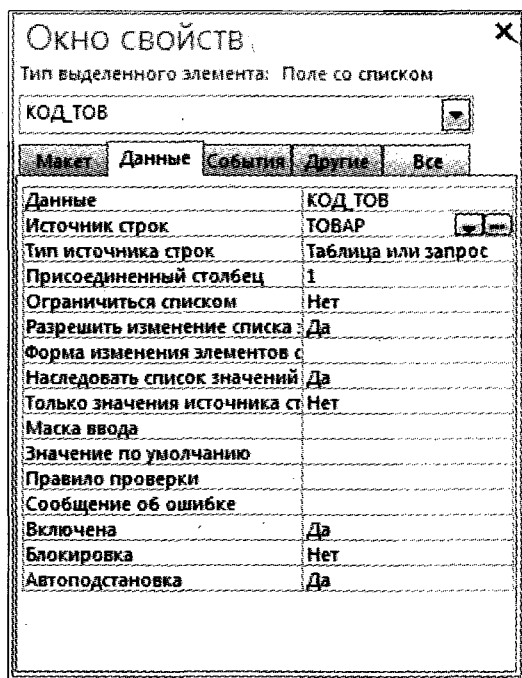


Рис. 6.18. Определение свойств создаваемого поля со списком без использования мастера

ОБРАТИТЕ ВНИМАНИЕ

Созданный таким образом список поля будет включать значения первого указанного в структуре таблицы столбца. В приведенном примере таким полем является поле КОД_ТОВ (код товара) — ключ таблицы ТОВАР, поэтому список поля будет построен правильно. Если же на первом месте в структуре таблицы окажется другое поле, список будет содержать неверные значения. В таком случае нельзя в качестве источника строк выбирать таблицу. Нужно с помощью построителя запросов или вручную записать в качестве источника строк инструкцию SQL, выбирающую из таблицы нужное поле — в нашем случае поле с кодом товара.

Если в списке необходимо отображать два столбца: код товара и его наименование, в конце строки свойства **Источник строк** (Row Source) щелкните на кнопке **Построитель запросов** (Query Builder). В бланке запроса должна присутствовать таблица ТОВАР. Включите в запрос ее поля КОД_ТОВ и НАИМ_ТОВ. Закройте окно построителя выражений, подтвердив сохранение инструкции SQL созданного запроса и обновление свойства. Теперь в строке **Источник строк** будет записана инструкция SQL:

```
SELECT ТОВАР.КОД_ТОВ, ТОВАР.НАИМ_ТОВ FROM ТОВАР;
```

Можно просто записать такую инструкцию в строку свойств.

В окне свойств поля на вкладке **Макет** (Format) замените значение свойства поля **Число столбцов** (Column Count) с 1 на 2. Кроме того, установите ширину столбцов и ширину списка, указав, например, 2, 5 и 7 см соответственно.

Для создания нового поля со списком без помощи мастера надо перетащить в форму элемент управления **Поле со списком** (Combo Box), когда кнопка **Использовать мастера** (Use Control Wizards) не активна.

Вычисления в документе

Вычисления в форме могут осуществляться как в каждой записи формы, так и для группы записей, при формировании итоговых величин. Расчетные величины только отображаются в вычисляемых полях формы. Для сохранения результатов в таблице базы данных может быть использован макрос или процедура обработки событий на VBA.

Рассчитайте в форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** значения стоимости заказанного товара в каждой строке подчиненной формы, а также общую стоимость заказанного в договоре товара. Разместите соответствующие вычисляемые поля, как предусмотрено в макете формы.

ЗАМЕЧАНИЕ

Если в форме отображается поле `СУММА_ПОСТ` из таблицы `ПОСТАВКА_ПЛАН`, поместите вычисляемое поле рядом. Сравните значения в поле `СУММА_ПОСТ` с расчетными и при необходимости подправьте. Если поле `СУММА_ПОСТ` пусто, скопируйте в него рассчитанное значение, и оно сохранится в поле таблицы.

Чтобы произвести вычисления на основе данных одной записи, создайте в подчиненной форме вычисляемое поле и запишите туда выражение: `=КОЛ_ПОСТ*ЦЕНА`. Для этого откройте форму в режиме макета. Проверьте наличие макета **В виде таблицы** в подчиненной форме. Нажмите на ленте **Конструктор (Design)** в группе **Элементы управления (Controls)** кнопку **Поле (Text Box)** и поместите этот свободный элемент управления, не связанный с источником данных, после поля с суммой поставки из таблицы. Проследите, чтобы при вставке элемента рядом с существующим полем появилась желтая вертикальная линия. Это позволит вставить новый столбец непосредственно в макет формы.

Откройте свойства нового поля и в строку **Данные (Control Source)** на одноименной вкладке запишите выражение. При этом можно воспользоваться построителем выражений. Сразу после ввода выражения в вычисляемом поле подчиненной формы появятся вычисленные значения. Чтобы ввести выражение непосредственно в элемент управления **Свободный (Unbound)**, нужно переключиться в режим конструктора. Выражение должно начинаться со знака равенства. В качестве операндов выражения чаще всего используются имена полей, константы, функции, а в качестве операторов — знаки арифметических операций.

Вычисление итоговых значений для группы записей подчиненной формы выполняется с помощью встроенных статистических функций. Эти функции записываются в вычисляемом поле области примечания подчиненной формы, которая доступна только в режиме конструктора. Для подсчета общей стоимости договора создайте в этой области свободный элемент управления и запишите туда выражение, содержащее статистическую функцию подсчета суммарной стоимости товаров: `=Sum(КОЛ_ПОСТ*ЦЕНА)`. В качестве аргумента статистической функции нельзя использовать имена других вычисляемых элементов управления формы. Ссылаться можно только на имена полей, которые являются источником данных.

Для отображения результата вычисления в главной форме создайте там свободный элемент управления и запишите в него выражение:

```
=[ПЛАН ПОСТАВОК].Form! [ПолеN]
```

Это выражение является ссылкой на элемент управления ПолеN в примечании подчиненной формы, содержащий суммарную стоимость товаров.

Чтобы элементы управления из области примечаний подчиненной формы не отображались при просмотре формы, установите в свойстве примечания **Вывод на экран (Visible)** значение **Нет**.

Перейдите в режим просмотра в форме и убедитесь, что результаты расчетов отображаются правильно.

Работа с документами

Форма **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** предназначена для ввода, просмотра, корректировки и удаления данных о договорах. В основной форме отображаются данные из одной записи таблицы **ДОГОВОР**, дополненные данными из таблицы **ПОКУПАТЕЛЬ**. В подчиненной форме выводятся связанные с договором записи из таблицы **ПОСТАВКА_ПЛАН**. Эти записи дополнены данными из таблицы **ТОВАР**.

Чтобы начать работу с формой, выберите ее в области навигации и в контекстном меню выполните команду **Открыть (Open)**.

Для просмотра и корректировки данных конкретного договора сделайте нужную запись текущей. Для этого воспользуйтесь созданными в форме кнопками **Предыдущий договор** и **Следующий договор** или стандартными кнопками перехода по записям, предусмотренными как для основной, так и для подчиненной формы.


При большом числе записей используйте команду поиска нужной записи. Для этого установите курсор на поле с номером договора и нажмите кнопку **Найти (Find)** в соответствующей группе вкладки **Главная (Home)**. В диалоговом окне **Поиск и замена (Find and Replace)** задайте номер нужного договора. После нажатия в этом окне кнопки **Найти далее (Find Next)** в форме отобразятся данные о заданном договоре.

ВНИМАНИЕ!

Грубой ошибкой является попытка перейти к нужной записи путем ввода нового значения в поле идентификации текущей записи — например, нового значения номера договора в форме **ДОГОВОР**. Такие действия могут привести лишь к изменению значения идентификатора в текущей записи.

Для просмотра договоров в порядке возрастания или убывания их номеров установите курсор на поле с номером договора и нажмите кнопку **Сортировка по возрастанию (Sort Ascending)** или **Сортировка по убыванию (Sort Descending)** в группе **Сортировка и фильтрация (Sort & Filter)** на вкладке ленты **Главная (Home)**.

Рассмотрим ввод в базу данных нового документа «Договор». Для загрузки данных о новом договоре через форму **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** нажмите на

нижней панели кнопок перехода по записям кнопку  **Новая запись (New (blank) record)**. Текущей становится пустая запись источника основной формы — таблицы ДОГОВОР и пустая запись источника подчиненной формы — таблицы ПОСТАВКА_ПЛАН.

В главной форме заполните обязательные поля: номер договора (ключ таблицы ДОГОВОР) и код покупателя (поле связи с таблицей ПОКУПАТЕЛЬ), а также поле даты, загружаемой в таблицу ДОГОВОР. Справочные поля из таблицы ПОКУПАТЕЛЬ автоматически отобразятся после ввода кода покупателя.

С полем типа данных **Дата/время (Date/Time)** связывается календарь. В форме при выборе поля с датой заключения договора справа от него отображается значок календаря. Выбранная в календаре дата вводится в поле `ДАТА_ДОГ`. Однако, если для поля определена маска ввода, календарь недоступен для использования. Если маска ввода была задана в свойствах таблицы, она автоматически наследуется полем в форме. Если после создания формы удалить маску в поле таблицы, она сохранится в поле формы. Для удаления маски в поле формы откройте свойства поля и очистите соответствующую строку на вкладке **Данные (Data)**.

ЗАМЕЧАНИЕ

При наличии календаря сохраняется возможность ручного ввода даты в поле.

Сформированная запись автоматически сохранится в таблице ДОГОВОР при переходе к другой записи договора или в подчиненную форму.

Поле суммы по договору может быть рассчитано и введено в таблицу ДОГОВОР после ввода данных в подчиненной форме.

ЗАМЕЧАНИЕ

Сохранить новую запись можно, выполнив команду **Сохранить (Save)** в группе **Записи (Records)** на вкладке ленты **Главная (Home)** или просто щелкнув мышью на области выделения записи.

Перейдите к формированию записей в подчиненной форме. В каждой новой записи введите значения обязательных полей: кода товара (ключевое), месяца поставки (ключевое), а также полей: минимальная партия поставки и количество. После ввода кода товара справочные данные: наименование, единица измерения и цена из таблицы ТОВАР — отобразятся автоматически. Сформированная запись сохранится в таблице ПОСТАВКА_ПЛАН при переходе к другой записи.

Если в строке подчиненной формы создано вычисляемое поле для расчета стоимости поставки по каждому товару, значение стоимости будет автоматически вычислено после заполнения полей **Цена** и **Количество**.

Если в примечании подчиненной формы создано вычисляемое поле для расчета общей стоимости поставки, и на него сделана ссылка в вычисляемом поле главной формы, после сохранения записи в подчиненной форме в главной форме отобразится общая сумма по договору. После завершения ввода записей в подчиненную форму договора общая сумма из вычисляемого поля может быть скопирована и вставлена в поле таблицы `СУММА_ПОСТ`.

ЗАМЕЧАНИЕ

Для расчета стоимостей с автоматическим сохранением результата в соответствующих полях таблиц ПОСТАВКА_ПЛАН и ДОГОВОР должны быть подготовлены процедуры обработки событий на VBA или макросы.

Рассмотренные ранее способы перехода по записям формы не обеспечивают пользователя достаточно удобным интерфейсом для выборки нужных документов. При работе с формами — электронными аналогами документов — пользователю часто необходимо иметь средства, обеспечивающие быстрый поиск нужного документа по заданным условиям отбора. Например, могут задаваться условия отбора для выборки документа по его номеру или по заданному временному диапазону дат. В общем случае целесообразно предусмотреть выборку объекта (товара, покупателя, склада, договора и т. п.) по заданному идентификатору или по диапазону количественных показателей.

Выборка документа по его идентификатору

Рассмотрим организацию выборки нужного договора по заданному номеру. Предусмотрим возможность ввода номера договора в диалоге с пользователем.

Для этого в источнике записей главной формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ — запросе на выборку из записей двух взаимосвязанных таблиц — в поле номера договора **НОМ_ДОГ** определим параметр запроса [Номер договора] (рис. 6.19). Представленное на рисунке окно построителя запросов открывается при щелчке на соответствующем значке в строке свойства формы **Источник записей** (Record Source).

Теперь при открытии формы будет выводиться диалоговое окно (рис. 6.20) для ввода значения параметра запроса (**Номер договора**).

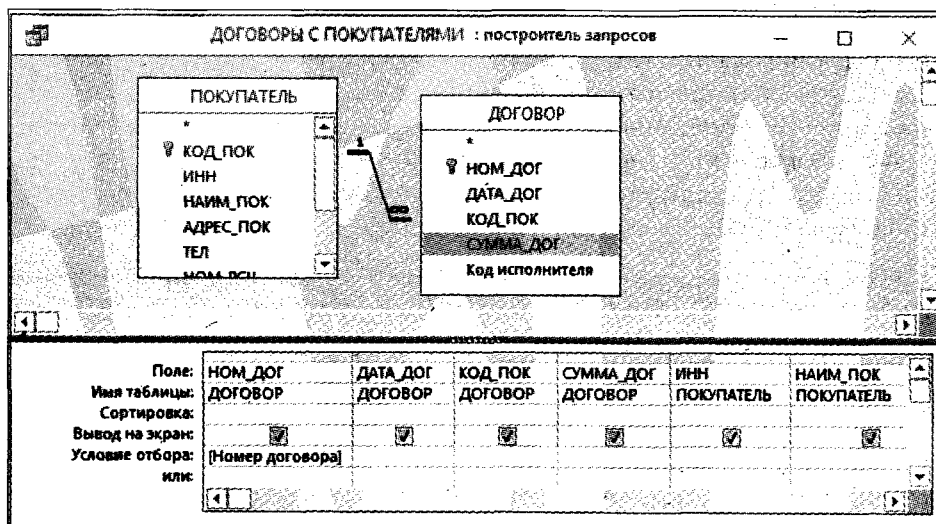


Рис. 6.19. Источник записей главной формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ

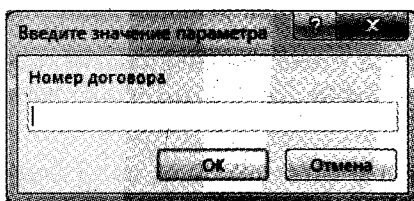


Рис. 6.20. Диалоговое окно ввода значения параметра запроса

После ввода нужного номера договора в форме будут доступны данные соответствующего договора. Однако для перехода к просмотру другого договора потребует-ся заново открыть форму.

Для удобства пользователя желательно создание такого интерфейса, который обеспечит возможность выбора документа при постоянно открытой форме. Для этих целей может быть подготовлена форма интерфейса, в которую встраивается форма документа. Форма интерфейса должна включать элемент управления, который обеспечивает ввод значения параметра запроса.

Создадим пустую форму (без источника записей), воспользовавшись командой **Конструктор форм (Form Design)** в группе **Формы (Forms)** на вкладке ленты **Создание (Create)**, и присвоим ей имя **Выборка договоров**. Создадим в форме, открытой

Рис. 6.21. Форма для вывода договора по выбираемому из списка номеру в режиме конструктора

в режиме конструктора, с помощью мастера элемент управления — поле со списком. Выберем в качестве источника формирования значений списка таблицу ДОГОВОР и ее поле — номер договора. Полю со списком в его свойствах дадим имя, совпадающее с названием параметра запроса: Номер договора (см. рис. 6.19). В надписи поля запишем обращение к пользователю: Введите номер договора. Перетащим из окна базы данных форму ДОГОВОРЫ С ПОКУПАТЕЛЯМИ в область данных формы Выборка договоров. Полученная составная форма в режиме конструктора приведена на рис. 6.21.

Удалим надпись встроенной формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ. В свойствах макета формы Выборка договоров для строк Область выделения (Record Selectors), Полосы прокрутки (Scroll Bars), Кнопки навигации (Navigation Buttons) выберем значение Нет.

В режиме формы при выборе из списка номера нужного договора он отобразится во встроенной форме ДОГОВОРЫ С ПОКУПАТЕЛЯМИ (рис. 6.22).

Код товара	Срок поставки	Наименование товара	Цена	Количество	Сумма поставки
T001	3	Монитор 17LG	7 000,00р.	50	350 000,00р.
T002	2	FDD 3,5	500,00р.	80	4 000,00р.
T005	1	CD-ROM Panasonic	800,00р.	40	32 000,00р.
T005	2	CD-ROM Panasonic	800,00р.	10	8 000,00р.

Рис. 6.22. Интерфейс, обеспечивающий выборку нужного договора в поле со списком

Выборка документа по диапазону дат

Для формы с именем ДОГОВОРЫ С ПОКУПАТЕЛЯМИ предусмотрим возможность выборки договоров, заключенных в заданном диапазоне дат. Для этого подготовим интерфейс, аналогичный рассмотренному в предыдущем примере.

В запросе — источнике записей главной формы (см. рис. 6.19) — удалим ранее заданный параметр запроса [Номер договора] и зададим в поле даты заключения договора `Дата_Дог` два параметра запроса. Параметры определим в выражении `Between [ДатаНачальная] And [ДатаКонечная]`, записанном в условии отбора.

Как в предыдущем примере, создадим форму без источника записей, в которой разместим два свободных поля, не связанных с каким-либо источником данных. В свойствах одному полю дадим имя параметра `ДатаНачальная`, второму — `ДатаКонечная`. Надписи полей изменим на `Дата начала периода` и `Дата окончания периода` соответственно. Перетащим из окна базы данных форму `ДОГОВОРЫ С ПОКУПАТЕЛЯМИ` в область данных формы.

Для связи полей ввода дат диапазона с календарем выберите для каждого поля в свойстве `Формат поля (Format)` на вкладке `Макет (Format)` нужный формат, например, `Краткий формат даты (Short Date)`.

В режиме формы после ввода дат начала и конца диапазона во встроенной форме отобразятся выбранные договоры (рис. 6.23). Просмотрев договоры одного диапазона, тут же введите другой. В форме станут доступными договоры, заключенные во вновь указанный диапазон.

Выборка договоров по диапазону дат

Задайте период для выборки договоров

Дата начала: 01.01.2016 Дата окончания: 08.06.2016

Закрывать форму **ДОГОВОР № Д111**

Попытатель: **Компьютер маркет**

Код покупателя: Т001 Телефон: 7023345-23-45

ИНН: 778957651111 Номер счета: 763585097632645385

Адрес: Москва Банк: Мост

Код товара	Срок поставки	Наименование товара	Цена	Количество	Сумма поставки
T001	1	Монитор 17LG	7 000,00p.	305	2 135 000,00p.
T001	2	Монитор 17LG	7 000,00p.	200	1 400 000,00p.
T001	3	Монитор 17LG	7 000,00p.	300	2 100 000,00p.
T002	1	FDD 3.5	500,00p.	50	25 000,00p.
T002	3	FDD 3.5	500,00p.	100	50 000,00p.

Записи: 1 из 3 Поиск

Сумма по договору: 8 414 000,00p.

Рис. 6.23. Интерфейс, обеспечивающий выборку документов по диапазону дат

Выборка документов с помощью фильтрации

Используйте фильтры для выборки нужных договоров или различных сведений из них.

В открытой для работы с договорами форме установите курсор на поле с номером договора. Щелкните на кнопке **Фильтр (Filter)** в группе **Сортировка и фильтр (Sort & Filter)** ленты **Главная (Home)**. В открывшемся окне откройте список **Текстовые фильтры (Text Filters)**. Для поля доступны эти фильтры, потому что поле имеет текстовый тип данных. В списке щелкните на опции **Содержит (Contains)** — откроется настраиваемый фильтр (рис. 6.24). Введите нужный номер договора и нажмите кнопку **ОК**. В форме отобразится выбранный договор. Открыть список **Текстовые фильтры** можно также, воспользовавшись контекстным меню поля.

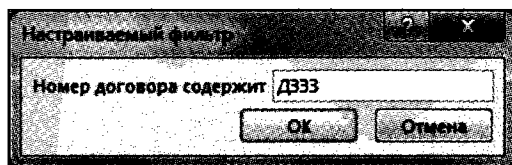


Рис. 6.24. Диалоговое окно задания условия отбора для фильтра

Чтобы снять фильтр с формы, используйте команду контекстного меню поля **Снять фильтр с Номер договора (Clear filter from НОМ_ДОГ)**. Если необходимо, задайте новые условия отбора по полю с номером договора.

Чтобы отобразить договоры за заданный период, в контекстном меню поля с датой заключения договора откройте **Фильтры дат (Date Filters)**. Список фильтров для дат позволяет осуществить выборку по самым разным критериям (рис. 6.25).

ОБРАТИТЕ ВНИМАНИЕ

Итоговые данные по договору подсчитываются только для выбранных в результате фильтрации данных.

Для отбора в договорах сведений только о заданном товаре используйте текстовый фильтр по полю с наименованием товара. Для отбора заданного товара, заказанного в заданном месяце, не снимая фильтра с наименования товара, выполните фильтрацию по полю **СРОК_ПОСТ** (срок поставки). Числовой фильтр по полю **СРОК_ПОСТ** позволит отобразить данные в любом заданном интервале. При использовании фильтров легко проверить запланированные поставки заданного товара в договорах и, если необходимо, откорректировать или дополнить их новыми строками.

Таким образом, использование фильтра позволит просматривать данные в различном представлении.

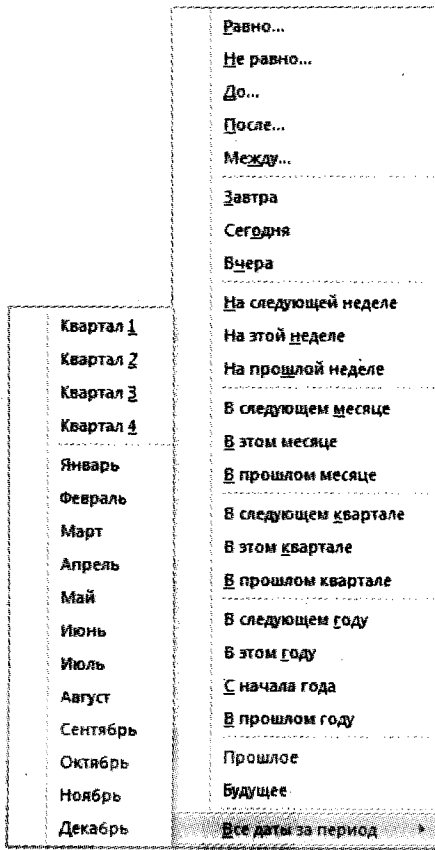


Рис. 6.25. Фильтры для выборки записей по полю с типом данных Дата/Время

Задание 6.1. Создание интерфейса для работы с документом «Накладная»

Создайте составную многотабличную форму для ввода и корректировки в базе данных накладных на отгрузку товаров.

Подсхема для такой формы приведена на рис. 6.26. В результате загрузки данных по накладным должны создаваться только записи таблиц НАКЛАДНАЯ и ОТГРУЗКА, находящихся в отношении 1 : М.

1. Осуществите проектирование интерфейса для загрузки данных документа «Накладная»:
 - определите общую структуру составной формы в соответствии с подсхемой данных, приведенной на рис. 6.26;
 - разместите реквизиты в макете формы так, чтобы обеспечить удобный ввод данных с документа, а также отображение справочной информации по договору, покупателю и товару.
2. Создайте экранную форму, адекватную виду документа, через которую будет осуществляться ввод, добавление и изменение записей таблиц НАКЛАДНАЯ и ОТГРУЗКА.
3. Загрузите данные с документа «Накладная» через построенную форму.

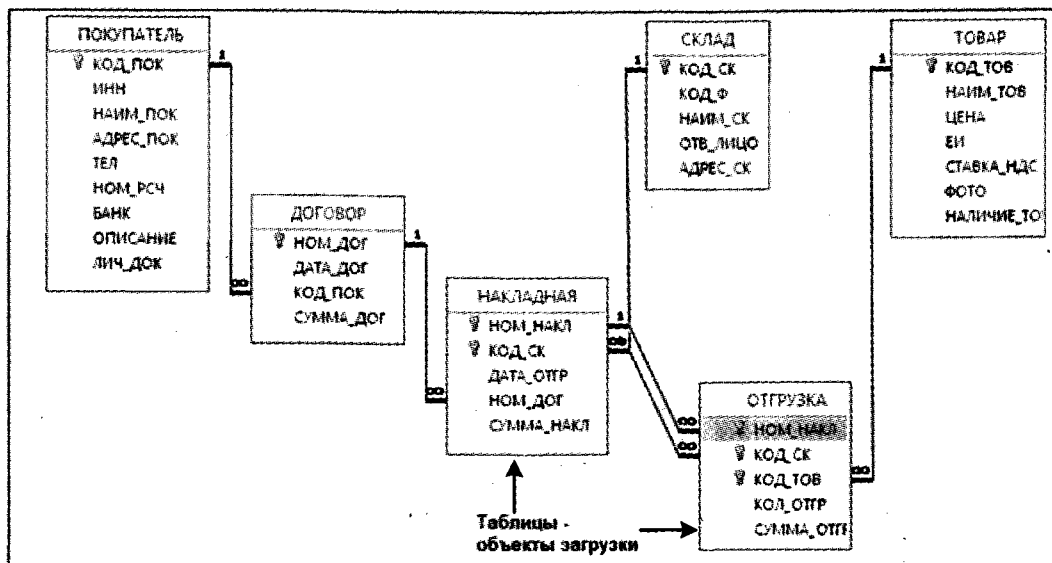


Рис. 6.26. Подсхема для составной формы по работе с накладными

Контрольные вопросы

1. Значения какой таблицы содержит поле со списком **Код покупателя** на рис. 6.10?
2. В запись какой таблицы вводится значение, выбранное в поле со списком **Код покупателя** на рис. 6.10?
3. Может ли поле со списком включать значения нескольких полей одной записи?
4. Можно ли с помощью одного поля со списком сразу выбрать несколько значений и ввести их в несколько полей?
5. В каком случае при создании поля со списком не подключается мастер?
6. Откуда может получать значения поле со списком?
7. Можно ли для поля со списком пользоваться командами поиска и сортировки?
8. Где можно посмотреть информацию об источнике данных для поля со списком?
9. Где хранится имя поля, в которое должно вводиться значение, выбранное в списке?
10. Какое свойство приводит к открытию формы в режиме добавления записей, при котором не возможен просмотр ранее введенных записей?
11. Какой элемент управления позволяет создать вычисляемое поле в форме?
12. С какого знака начинается выражение, записываемое в вычисляемое поле?
13. В каком разделе формы размещается вычисляемое поле для расчета итогового значения для подчиненной формы?

14. Вычисление итоговых значений возможно в любой многозаписевой форме? В каком разделе формы?
15. Можно ли при расчете итогового значения использовать в аргументе функции имя другого вычисляемого элемента управления?
16. Как отобразить итоговое значение, рассчитанное в подчиненной форме, в главной форме?
17. Какой формат имеет ссылка на вычисляемое поле в подчиненной форме?
18. Сохраняется ли значение вычисляемого поля формы в таблице?
19. Что нужно сделать, если не работает связь основной и подчиненной форм?
20. Должны ли поля, по которым устанавливается связь между основной и подчиненной формами, иметь одинаковые имена?
21. Может ли связь между основной и подчиненной формами устанавливаться по нескольким полям?
22. Что указывает мастер в качестве источника записей в форме, содержащей поля нескольких взаимосвязанных таблиц?
23. Как просмотреть запрос — источник записей многотабличной формы в режиме конструктора?
24. Можно ли, изменив инструкцию SQL или запрос в режиме конструктора, повлиять на состав доступных в форме полей?
25. В каком случае к созданию кнопки не подключается мастер?
26. Можно ли изменить вид кнопки после завершения работы мастера?
27. Какие таблицы можно дополнить данными при работе в форме ДОГОВОРЫ С ПОКУПАТЕЛЯМИ?
28. Какие поля входят в список доступных в форме?
29. В каком элементе управления размещается подчиненная форма?
30. В каких свойствах указываются поля связи основной и подчиненной формы?
31. Сохраняет ли в форме поле такие свойства, как Поле со списком (List Box), Список (Combo Box), заданные при его определении в таблице?
32. Где хранится ссылка на источник записей формы?
33. В какой последовательности загружаются таблицы базы данных при установленных параметрах обеспечения целостности связанных записей?
34. Какой объект нужно использовать для первоначальной загрузки и корректировки взаимосвязанных таблиц базы данных?
35. В каких отношениях, как правило, находятся таблицы — источники главной и подчиненной формы?
36. Какие поля записеобразующей таблицы должны быть обязательно включены в подчиненную форму, чтобы через нее можно было вводить новые записи?

37. Должны ли поля, входящие в ключ подчиненной формы, но не отображенные в ней, включаться в источник записей?
38. Можно ли через многотабличную форму осуществить ввод данных в несколько таблиц?
39. Каким требованиям должен отвечать интерфейс пользователя для работы с документами, сохраняемыми в базе данных?
40. Назовите основные этапы проектирования формы для работы с документами, сохраняемыми в базе.
41. Какими средствами целесообразно создавать интерфейс для подготовки и ввода данных документа?
42. Можно ли через форму НАКЛАДНАЯ ввести запись об отгрузке товара, отсутствующего в таблице ТОВАР?

Ответы

1. Значения из поля **Код покупателя** таблицы **ПОКУПАТЕЛЬ**.
2. В поле **Код покупателя** текущей записи таблицы **ДОГОВОР**.
3. Да.
4. Нет.
5. Мастер не установлен на компьютере или на панели элементов не нажата кнопка **Использовать мастера (Use Control Wizards)**.
6. Из таблицы, запроса, фиксированного набора значений.
7. Да.
8. В свойствах поля со списком, на вкладке **Данные (Data)** в строке **Источник строк (Row Source)**.
9. В свойствах поля со списком, на вкладке **Данные (Data)** в строке **Данные (Control Source)**.
10. Свойство формы **Ввод данных (Data Entry)** на вкладке **Данные (Data)**.
11. Элемент управления **Поле (Text Box)**, создаваемый по соответствующей кнопке на панели элементов.
12. Со знака равенства (=).
13. В области примечания формы.
14. Да. В области примечания формы.
15. Нет.
16. Создать новое вычисляемое поле в главной форме и разместить в нем ссылку на вычисляемое поле в подчиненной форме.
17. =[Имя подчиненной формы].[Form]![Имя вычисляемого поля в подчиненной форме].
18. Нет.

19. Установить правильные значения в свойствах элемента управления **Подчиненная форма/отчет** (Subform/Subreport) **Подчиненные поля** (Link Child Fields) и **Основные поля** (Link Master Fields).
20. Нет.
21. Да.
22. Инструкцию `select` языка SQL.
23. Вызвав построитель запросов, кнопкой в конце строки **Источник записей** (Record Source) в окне свойств формы.
24. Да.
25. Если предварительно на вкладке ленты **Конструктор** (Design) в группе **Элементы управления** (Controls) не нажата кнопка **Использовать мастера** (Use Control Wizards).
26. Да.
27. ДОГОВОР и ПОСТАВКА_ПЛАН.
28. Все поля, выбранные из таблиц или запросов, на основе которых формируется источник записей формы.
29. **Подчиненная форма/отчет** (Subform/Subreport).
30. В свойствах **Подчиненные поля** (Link Child Fields) и **Основные поля** (Link Master Fields) элемента управления **Подчиненная форма/отчет** (Subform/Subreport).
31. Да.
32. В свойствах формы на вкладке **Данные** (Data) в строке **Источник записей** (Record Source).
33. Сначала главные таблицы, затем подчиненные.
34. Многотабличную форму.
35. Один-ко-многим.
36. Все поля, составляющие ключ и не представленные в главной форме.
37. Обязательно, наличие недостающих полей ключа в главной форме обеспечивает автоматический их ввод в запись подчиненной формы.
38. Да.
39. Интерфейс должен быть построен на основе форм, соответствующих виду документа, в которых наряду с основными данными документа отображаются справочные данные.
40. Определение подсхемы данных формы с указанием роли таблиц и макета формы с указанием связи элементов формы с полями таблиц.
41. Мастером форм с последующей доработкой в режиме макета и/или конструктора.
42. При установленных параметрах обеспечения связной целостности — нет.

ГЛАВА 7



Отчеты

Средства Access по разработке отчетов предназначены для подготовки макета отчета, в соответствии с которым осуществляется вывод данных из определенного источника записей в виде выходного печатного документа. Эти средства позволяют создавать отчет любой сложности, обеспечивающий вывод взаимосвязанных данных из многих таблиц, их группировку, вычисления, фильтрацию и анализ данных в любых разрезах. При этом могут быть выполнены самые высокие требования к оформлению выходного документа.

Перед началом создания отчета пользователь должен спроектировать его макет. В макете устанавливается состав и содержание разделов отчета, размещение в нем значений, выводимых из полей таблиц (запросов) базы данных, и вычисляемых реквизитов, определяются поля, по которым нужно группировать данные. Для каждого уровня группировки задаются заголовки и примечания, вычисляемые итоговые значения. Кроме того, заголовки и подписи реквизитов отчета соответствующим образом оформляются.

Отчет может создаваться с помощью мастера или в режиме конструктора отчетов. Во многих случаях удобно использовать мастер отчетов. Созданный мастером отчет можно доработать в режиме конструктора. Простые отчеты на основе одной текущей таблицы или запроса могут создаваться автоматически.

При необходимости вывода в отчете результатов решения задачи, в качестве основы для отчета может быть использован многотабличный запрос, представляющий эти результаты. На запрос могут быть возложены наиболее сложные виды выборки и предварительной обработки данных. Разнообразные возможности конструктора отчетов позволяют легко структурировать и оформить полученные в запросе данные.

Access позволяет создать профессионально оформленные отчеты не только с помощью мастера или конструктора, но и в режиме макета. При этом простыми средствами перетаскивания в отчет нужных полей из таблиц базы данных строится запрос — источник записей отчета, а использование свойств WYSIWYG дает возможность сразу увидеть, как именно будут выглядеть содержащиеся в нем данные на странице отчета, и усовершенствовать макет.

Основы конструирования отчетов

Чтобы правильно создавать отчеты, необходимо понимать назначение каждого его раздела. Например, от выбора раздела, в который будет помещен вычисляемый элемент управления, зависит способ вычисления результата.

Создание и изменение макета отчета осуществляется в расчете на структуру отчета. Пустые разделы отчета, открытого в режиме конструктора, показаны на рис. 7.1.

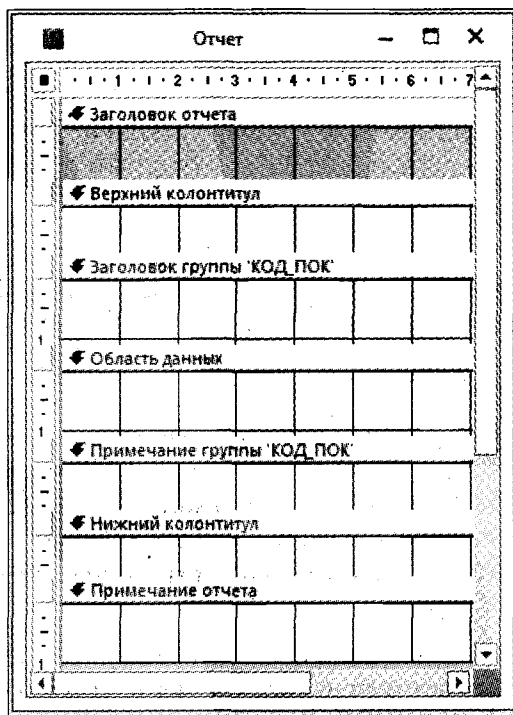


Рис. 7.1. Разделы отчета в режиме конструктора

Рассмотрим назначение каждого из разделов:

- **Заголовок отчета (Report Header)** обычно включает эмблему (логотип) компании, название отчета, дату. Заголовок отображается перед верхним колонтитулом только один раз в начале отчета;
- **Верхний колонтитул (Page Header)** отображается вверху каждой страницы и используется в случае, когда нужно, чтобы название отчета и другая общая информация повторялись на каждой странице;
- **Заголовок группы (<имя поля> Header)** используется при группировке записей отчета для вывода названия группы и однократного отображения полей, по которым производится группировка. Отображается перед каждой новой группой записей. Например, если отчет сгруппирован по покупателям, в заголовке группы можно указать название покупателя, а также адрес, телефон и другие реквизиты. Допускается до 10 уровней группировки выводимых записей;

- **Область данных (Detail)** отображает записи из источника данных, составляющие основное содержание отчета;
- **Примечание группы** (<имя поля> Footer) служит для отображения итогов и другой сводной информации по группе в конце каждой группы записей. Если поместить в примечание группы вычисляемый элемент управления, использующий статистическую функцию Sum, сумма будет рассчитываться для текущей группы;
- **Нижний колонтитул (Page Footer)** применяется для нумерации страниц и отображения другой информации внизу каждой страницы;
- **Примечание отчета (Report Footer)** служит для отображения итогов и другой сводной информации по всему отчету один раз в конце отчета. Если в примечании отчета поместить вычисляемый элемент управления, использующий статистическую функцию Sum, сумма рассчитывается для всего отчета.

ЗАМЕЧАНИЕ

В конструкторе **Примечание отчета (Report Footer)** находится под нижним колонтитулом. Однако при печати и в режиме предварительного просмотра оно помещается над нижним колонтитулом последней страницы.

Вставка или удаление заголовка и примечания отчета могут быть выполнены командой контекстного меню отчета **Заголовок/примечание отчета (Report Header/Footer)**. Кроме того, заголовок и примечание отчета вставляются при добавлении в отчет элементов управления **Заголовок (Title)**, **Эмблема (Logo)** или **Дата и время (Date & Time)**. Для включения или удаления колонтитулов можно воспользоваться командой контекстного меню **Колонтитулы страницы (Page Header/Footer)**.

Заголовок и примечание группы добавляются в отчет после определения его источника записей и выбора в области **Группировка, сортировка и итоги (Group, Sort, and Total)** поля группировки и поля, по которому следует рассчитать итоги. Область **Группировка, сортировка и итоги** выводится под макетом отчета по команде **Группировка (Group & Sort)** на ленте **Конструктор** или по команде **Сортировка и группировка (Group & Sort)** из контекстного меню режима конструирования.

При создании отчета его разделы нужно заполнить элементами в соответствии с разработанным пользователем макетом отчета. В заголовок помещается текст из шапки макета отчета. В верхний и нижний колонтитулы обычно помещают надписи с поясняющим текстом, в том числе заголовки столбцов отчета, номера страниц. При определении содержания этих разделов следует исходить из требований к оформлению отдельных страниц отчета.

Поля таблиц базы данных или запросов с неповторяющимися значениями размещаются в области данных, которой можно придать вид табличной части отчета. Поля с повторяющимися значениями, по которым производится группировка записей, целесообразно размещать в заголовке группы. Здесь же отображаются данные, которые позволяют идентифицировать группу.

Элементами разделов отчета, кроме полей таблиц или запросов, на которых строится отчет, являются также тексты подписей, внедряемые объекты, линии, прямоугольники и т. п.

Для каждого элемента, а также раздела и отчета в целом, могут быть уточнены свойства. Технология размещения элементов и определения их свойств практически такая же, как и при разработке форм.

Инструменты разработки отчетов представлены на лентах, которые связаны с режимами изменения и просмотра отчетов, и появляются при переходе из одного режима в другой.

В Access существуют два представления, в которых можно вносить изменения в отчет: режим макета и режим конструктора. Режим макета является наиболее удобным для внесения изменений в отчет, поскольку пользователь сразу видит данные отчета. В этом режиме доступны большинство инструментов, необходимых для его настройки. В нем можно изменить ширину столбцов, поменять их местами, добавить или изменить уровни группировки и итоги. Можно разместить в макете отчета новые поля, а также задать свойства отчета и элементов управления.

В режиме конструктора отображаются разделы отчета и доступны дополнительные инструменты и возможности разработки. Переходите в режим конструктора, если не удается выполнить необходимые изменения в режиме макета. В определенных случаях в Access отображается сообщение о том, что для внесения изменений следует переключиться в режим конструктора.

Просматривать отчет можно в режимах **Представление отчета (Report View)**, **Предварительный просмотр (Print Preview)** или **Макет (Layout View)**. В режиме **Представление отчета** можно отфильтровать данные для отображения только заданных строк, найти нужные данные, скопировать текст отчета или его часть в буфер обмена. Режим предварительного просмотра предназначен для просмотра отчета перед печатью. В этом режиме можно увеличивать масштаб для просмотра деталей или уменьшать его для проверки размещения данных на странице, изменять параметры страницы. Режим макета позволяет, просматривая данные отчета, изменять его.

Однотабличные отчеты

Наиболее простым способом создания отчета на основе таблицы или запроса является использование команды **Отчет (Report)**. С помощью этого средства отчет формируется без диалога с пользователем и выводит все поля выбранного источника. Рассмотрим создание однотабличного отчета о договорах покупателей фирмы.

Пусть в результате проектирования макета отчета «Договоры фирмы» определены следующие требования к отчету:

- в отчете должны последовательно выводиться со своими заголовками списки договоров для каждого покупателя;
- список договоров каждого покупателя должен начинаться с новой страницы;

- записи о договорах должны выводиться в порядке возрастания их номеров;
 - по каждому покупателю и фирме в целом должны выводиться итоговые суммы.
- Выберем в области навигации таблицу ДОГОВОР, данные из которой будут источником записей отчета. Выполним на ленте Создание (Create) в группе Отчеты (Reports) команду Отчет (Report).

Access автоматически создаст отчет и отобразит его в режиме макета (рис. 7.2). В отчете представлены все записи таблицы ДОГОВОР.

Номер договора	Дата	Код покупателя	Сумма по договору	Код исполнителя
D111	11.01.2016	П001	6 414 000,00p	6
D222	05.02.2016	П001	152 280,00p	1
D333	01.01.2016	П002	74 920,00p	3
D555	12.11.2016	П002	9 352,00p	3
D777	14.06.2016	П001	1 579 920,00p	1
D888	23.05.2016	П002	30 000,00p	6
D999	12.09.2016	П002	450 000,00p	7
			8 710 472,00p	

12 апреля 2016
18:02:4

Страница 1 из 1

Рис. 7.2. Отчет, созданный соответствующей командой (режим макета)

В режиме конструктора отчетов (рис. 7.3) представлены разделы отчета и размещение в них полей таблицы — источника записей отчета. В области данных отчета размещены все поля таблицы. Это обеспечивает вывод в отчете всех записей в табличном виде. Размещение заголовков столбцов в верхнем колонтитуле обеспечивает вывод их на каждой странице отчета. Для добавления номера страницы в нижнем колонтитуле в вычисляемом элементе сформировано выражение: ="Страница " & [Page] & " из " & [Pages].

В заголовке отчета размещены: эмблема, название отчета, текущая дата и время. Для добавления в отчет текущей даты использована встроенная функция =Date() из категории Дата/время (Date/Time). В свойствах этого элемента на вкладке Макет (Format) в строке Формат поля (Format) установлен формат отображения даты — **Длинный формат даты (Long Date)**. Для добавления в отчет текущего времени применена встроенная функция =Time().

В примечании отчета в вычисляемом элементе управления записана статистическая функция =Sum([СУММА_ДОГ]), рассчитывающая стоимость всех договоров, представленных в отчете.

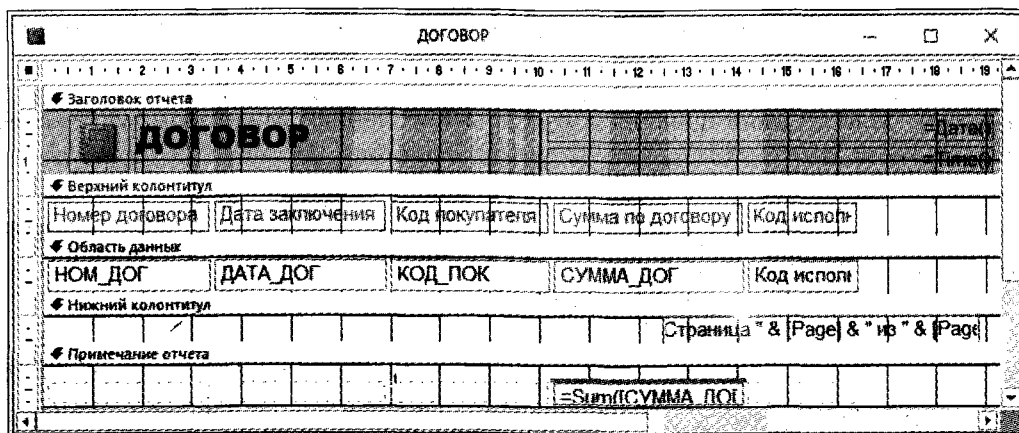


Рис. 7.3. Разделы однотабличного отчета в режиме конструктора

Доработка отчета в режиме макета

В режиме макета (см. рис. 7.2) легко привести созданный отчет в соответствие с заданными требованиями и размерами страницы. Ориентируясь на фактические данные отчета, можно отрегулировать ширину столбцов, высоту строк, изменить их порядок, добавить уровни группировки и итоговые значения. Можно добавить в отчет новые поля, а также изменить свойства отчета и входящих в него элементов управления. Границы страницы отчета в режиме макета отмечены пунктирной линией. Это позволяет контролировать размещение элементов отчета в заданных рамках.

Измените название отчета на *Договоры фирмы*. Для этого выполните двойной щелчок на нем и введите новое название. Для замены эмблемы удалите старую и, щелкнув на значке *Эмблема (Logo)* на вкладке ленты *Конструктор (Design)*, выберите нужный рисунок — эмблема будет добавлена в то же место заголовка отчета. При желании ее можно перетащить в другое место и изменить размер.

Элементы управления отчета «Договор» организованы в табличный макет. Табличные макеты элементов всегда охватывают несколько разделов отчета — в нашем случае надписи расположены в верхнем колонтитуле выше области данных, в которой расположены поля данных, а в примечании отчета расположен вычисляемый элемент, также включенный в этот макет. Выделите макет отчета и измените размер шрифта, выбрав его на вкладке *Главная (Home)* в группе *Форматирование текста (Text Formatting)* или на вкладке *Формат (Format)* в группе *Шрифт (Font)*, — все элементы макета изменят шрифт на выбранный.

Для изменения размеров элементов управления макета можно выделить весь макет, и тогда все его элементы будут менять размер и местоположение. Можно выделить отдельный столбец макета и изменять ширину только этого столбца и перемещать его внутри макета, перетаскивая на нужное место. При перетаскивании вертикальная полоска указывает, где элемент будет помещен, если отпустить кнопку мыши. Аналогично можно изменить и высоту строки.

При необходимости изменять положение и размер отдельных элементов макета их нужно удалить из макета и создать новый дополнительный макет. Выделите элемент с общей суммой по договорам и в контекстном меню выполните команду **Макет | Удалить макет (Layout | Remove Layout)**. Теперь стало возможным изменение ширины только этого элемента.

Для выделения раздела отчета щелкните слева от него, и он будет обрамлен жирной линией. Воспользовавшись кнопкой **Заливка фигуры (Shape Fill)** в группе **Форматирование элемента управления (Control Formatting)** на вкладке ленты **Формат (Format)**, выберите из коллекции нужный цвет.

Используйте условное форматирование для изменения внешнего вида значения в поле **Сумма по договору** в зависимости от одного или нескольких условий. Измените цвет денежных значений на красный, когда они оказываются выше заданной величины, например, 400 000 руб. Для этого щелкните в поле `СУММА_ДОГ` и выполните команду **Условное форматирование (Conditional Formatting)** на ленте **Формат (Format)** в группе **Форматирование элемента управления**. В окне **Диспетчер правил условного форматирования (Conditional Formatting Rules Manager)** в поле **Показать правила форматирования для (Show formatting rules for)** будет указано имя поля `СУММА_ДОГ`, для которого нужно создать правила. Щелкните на кнопке **Создать правило (New Rule)**. В окне **Новое правило форматирования (New Formatting Rule)** измените описание правила, оставив в первом поле **Значение поля (Field Value Is)**, выбрав во втором **больше (greater than)** и в третьем введя значение 400000. В области **Просмотр (Preview)** выберите формат, используемый при выполнении условия для значений поля: жирный шрифт и красный цвет.

Для изменения внешнего вида отчета в целом используйте богатый выбор коллекции **Темы** на ленте **Конструктор (Design)**.

Для изменения параметров страницы выполняйте команды соответствующей ленты. Пунктирной линией в отчете отмечены границы страницы. Отображение этой линии регулируется кнопкой **Показать поля (Show Margins)** на вкладке ленты **Параметры страницы (Page Setup)**. Для выбора размеров полей страницы может быть использована коллекция, отображаемая при нажатии кнопки **Поля (Margins)**. На этой же ленте можно выбрать размер бумаги, ориентацию страницы и ряд других параметров.

Группировка и сортировка данных отчета

Для выполнения группировки и сортировки данных, отображаемых в отчете, предназначена область в самом низу окна документов **Группировка, сортировка и итоги (Group, Sort, and Total)** (рис. 7.4). Если область не отображена, выполните команду **Группировка (Group & Sort)** на ленте **Конструктор (Design)**. Первоначально в отчете, созданном с помощью средства **Отчет (Report)**, нет ни группировки, ни сортировки, и область содержит лишь кнопки **Добавить группировку (Add a group)** и **Добавить сортировку (Add a sort)**.

В режиме макета выполните группировку данных отчета по полю `КОД_ПОК` (код покупателя). Щелкните на кнопке **Добавить группировку (Add a group)**. В области

Группировка, сортировка и итоги (Group, Sort, and Total) появится новая строка и будет отображен список доступных полей (рис. 7.5).

Щелкните для группировки в поле **КОД_ПОК** (код покупателя) — вид отчета немедленно изменится: в отчет для каждой группы добавится строка со значением кода покупателя в первом столбце, а в области **Группировка, сортировка и итоги** (Group, Sort, and Total) появится строка с указанием на группировку по полю **КОД_ПОК**.

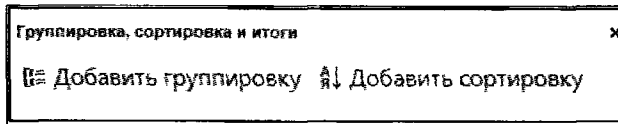


Рис. 7.4. Область управления группировкой и сортировкой в отчете

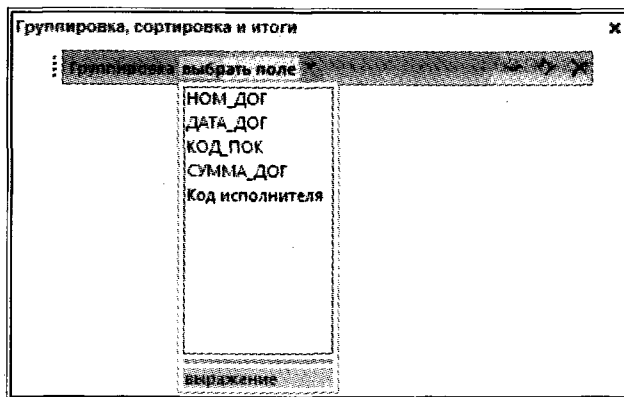


Рис. 7.5. Выбор поля группировки в отчете

Если переключиться в режим конструктора, то вы увидите новый раздел отчета **Заголовок группы 'КОД_ПОК'**. В отчете можно определить до 10 уровней группировки и сортировки.

Вернитесь в режим макета. Для каждого уровня группировки существует ряд параметров. Для отображения всех параметров щелкните **Больше** (More) на уровне, который нужно изменить (рис. 7.6).

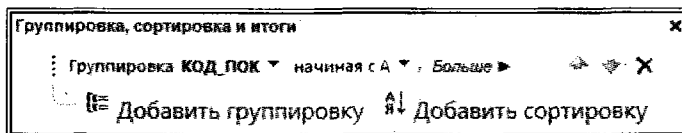


Рис. 7.6. Строка группировки первого уровня по полю **КОД_ПОК**

Кнопки, расположенные в строке группировки справа, позволяют повысить (стрелка вверх), понизить (стрелка вниз) или удалить уровень группировки.

Для добавления в отчет итоговых сумм по каждому покупателю и общей суммы нажмите кнопку **Больше** (greater than) и откройте список значений параметра

с итогами: СУММА_ДОГ (with СУММА_ДОГ totaled). В поле списка **Итог по полю** (Total On) выберите поле СУММА_ДОГ, по которому нужно рассчитать итоговые значения. Отметьте флажок **Показать общий итог** (Show Grand Total) и определите место отображения итоговых значений по каждому покупателю выбором флажка **Показать промежуточный итог в примечании группы** (Show in group footer) (рис. 7.7).

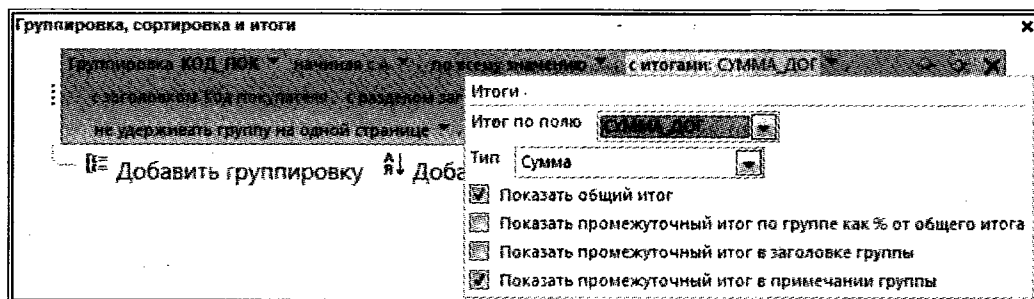


Рис. 7.7. Строка группировки с открытым списком параметра с итогами: СУММА_ДОГ

Для закрытия всплывающего окна **Итоги** (Totals) щелкните любое место за его пределами — в отчет для каждой группы немедленно добавится строка с итоговым значением в столбце **Сумма по договору**.

Если переключиться в режим конструктора, то вы увидите новый раздел отчета **Примечание группы 'КОД_ПОК'**, в котором размещено вычисляемое поле с выражением для подсчета общей суммы по договорам группы — договорам покупателя.

Можно подвести итоги для нескольких полей группы. Для этого следует выбирать каждое поле в раскрывающемся списке **Итог по полю** (Total On) и нужные параметры.

По умолчанию интервал группировки выбирается **по всему значению** (by entire value) (см. рис. 7.7). Этот параметр определяет способ группировки записей. Например, можно выполнить группировку по первой букве текстового поля, так что будут объединены в одну группу элементы, начинающиеся с буквы А, затем — с Б и т. д. Для поля даты можно задать группировку по дню, неделе, месяцу, кварталу или по специальному интервалу.

Параметр с заголовком (with title) позволяет изменить заголовок поля группировки. Чтобы изменить заголовок, щелкните выделенный синим цветом текст (Код покупателя) за словами **с заголовком**. В окне **Область ввода** (Zoom) введите новый заголовок и нажмите кнопку **ОК**.

Для установления в группе сортировки по номеру договора щелкните на кнопке **Добавить сортировку** (Add a sort) и в списке доступных полей — на строке **НОМ_ДОГ**. Строки отчета немедленно отсортируются, а в области **Группировка, сортировка и итоги** (Group, Sort, and Total) появится новая строка с указанием на сортировку по полю **НОМ_ДОГ** (рис. 7.8).

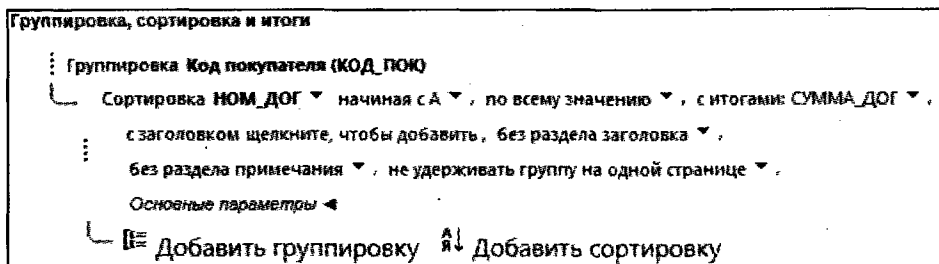


Рис. 7.8. Строка сортировки по полю НОМ_ДОГ

В качестве значения параметра порядок сортировки по умолчанию выбирается **начиная с А** (With A on top). Для изменения значения выберите его из раскрывающегося списка.

Отчет в режиме макета после внесения перечисленных изменений представлен на рис. 7.9.

Код покупателя	Номер договора	Дата	Сумма по договору	Код исполнителя
П001				
	Д777	14.06.2016	1 579 920,00р.	1
	Д222	05.02.2016	152 280,00р.	1
	Д111	11.01.2016	6 414 000,00р.	6
			8 146 200,00р.	
П002				
	Д999	12.09.2016	450 000,00р.	7
	Д888	23.05.2016	30 000,00р.	6
	Д555	12.11.2016	9 352,00р.	3
	Д333	01.01.2016	74 920,00р.	3
			564 272,00р.	
			8 710 472,00р.	

Рис. 7.9. Отчет с группировкой по полю Код покупателя и итогами для каждого покупателя

Если в отчете необходимо отобразить только итоговые данные (рис. 7.10), выполните команду **Без подробностей** (Hide Details) на ленте **Конструктор** (Design) в группе **Группировка и итоги** (Grouping & Totals). Повторное выполнение команды **Без подробностей** восстанавливает строки отчета, поскольку записи только скрываются, а элементы управления в скрытом разделе не удаляются.

Код покупателя	Номер договора	Дата	Сумма по договору	Код исполнителя
П001			8 146 200,00р.	
П002			564 272,00р.	
			8 710 472,00р.	

Рис. 7.10. Отчет только с итоговыми данными по каждому покупателю и фирме в целом

Просмотр и печать отчета

Просмотр отчета возможен в одном из трех режимов:

- режим **Представление отчета (Report View)** предназначен для работы с отображаемым на экране отчетом и обеспечивает выборку записей по заданным условиям, поиск данных по образцу, копирование части данных в буфер обмена;
- режим **Предварительный просмотр (Print Preview)** показывает, в каком виде отчет будет выведен на печать, и обеспечивает необходимую настройку параметров страницы;
- Режим макета (Layout View)**, как было показано в предыдущем разделе, обеспечивает просмотр отчета при одновременной настройке макета отчета.

Представление отчета

Для перехода в режим **Представление отчета (Report View)** выберите его из списка кнопки **Режим (View)** на вкладке ленты **Главная (Home)**. Если отчет закрыт, выберите его в области навигации и выполните двойной щелчок или в контекстном меню выполните команду **Открыть (Open)**. Для режима **Представление отчета** не существует специальной ленты, и доступные команды представлены на ленте **Главная (Home)**.

В этом режиме, так же как в режиме макета, для отбора нужных записей в отчете можно использовать фильтр. После применения фильтра в представление включаются только те записи, которые удовлетворяют заданным условиям отбора. Прочие записи останутся скрыты до тех пор, пока фильтр не будет удален.

Например, можно применить фильтр, чтобы отобразить все договоры, заключенные в заданном периоде. Установите курсор в столбце с датой заключения договора. Выполните команду **Фильтр (Filter)** на вкладке ленты **Главная (Home)** в группе **Сортировка и фильтр (Sort & Filter)**. В открывшемся окне откройте список

Фильтры дат (Date Filters) и щелкните на строке **В прошлом месяце** (Last Month) (рис. 7.11) — в отчете отобразятся только те договоры, которые были заключены в указанном периоде, и только для этих договоров будут рассчитаны итоговые значения.

Щелкните в списке **Фильтры дат** (Date Filters) на строке **После...** (After...) и откроется окно **Настраиваемый фильтр** (Custom Filter), в котором можно задать дату для отбора договоров, заключенных после ее наступления (рис. 7.12).

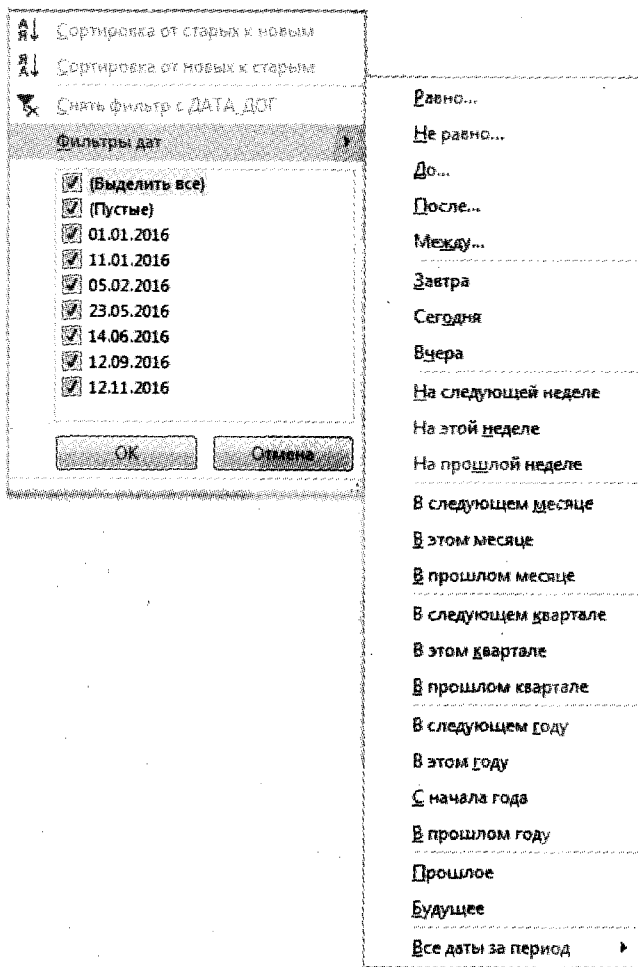


Рис. 7.11. Фильтр для столбца с датами

Для задания в условии отбора диапазона дат (рис. 7.13) используйте команду **Между...** (Between...) из списка кнопки **Фильтры дат** (Date Filters) (см. рис. 7.11).

Чтобы вернуться к отображению всех строк отчета, удалите фильтр, выполнив команду **Удалить фильтр** (Toggle Filter). После этого кнопка команды получает имя **Применить фильтр** (Toggle Filter). Выполнение этой команды позволит снова задействовать фильтр. Для окончательного удаления фильтра выполните команду

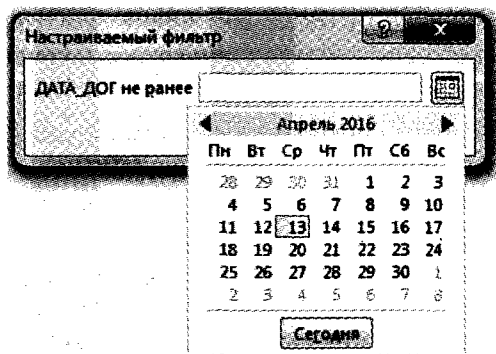


Рис. 7.12. Настраиваемый фильтр для столбца с датами

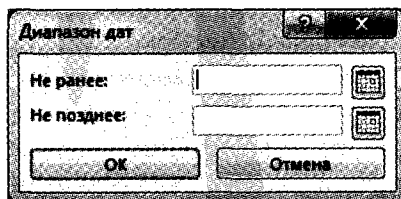


Рис. 7.13. Выбор диапазона дат для условия отбора строк отчета

Удалить все фильтры (Clear All Filters) из списка кнопки **Дополнительно (Advanced Filter Options)**.

Для изменения, сохранения и загрузки ранее сохраненных фильтров используйте команду **Расширенный фильтр (Advanced)** из списка кнопки **Дополнительно (Advanced Filter Options)**. Так, в открывшемся запросе можно изменить условия отбора и, выполнив в контекстном меню запроса команду **Применить фильтр (Toggle Filter)**, обновить содержимое отчета. Здесь же выполняются команды сохранения фильтра как запроса и его загрузки из запроса.

В зависимости от типа данных в выбранном столбце доступны различные фильтры. Установите курсор в столбец с суммой договора, и в окне фильтра вместо **Фильтры дат (Date Filters)** отобразятся **Числовые фильтры (Number Filters)**. Для столбца с текстом отобразятся **Текстовые фильтры (Text Filters)**.

Таким образом, использование фильтров позволяет, не создавая новых более специфичных отчетов, выполнять анализ данных, просматривая наборы записей, сформированные при различных условиях отбора.

ОБРАТИТЕ ВНИМАНИЕ

Выполнение группировок, добавление итогов для групп, просмотр данных отчета с подробностями или без них невозможны в режиме представления отчета. Для выполнения этих операций используйте режим макета.

Предварительный просмотр

Для предварительного просмотра страницы отчета и внесения в нее изменений перед печатью предназначен соответствующий режим. Предварительный просмотр позволяет убедиться, что отчет будет напечатан правильно. Переход из режима макета в режим предварительного просмотра отчета осуществляется с помощью кнопки **Режим (View)** на ленте **Главная (Home)**. При этом все ленты делаются недоступными, открыта только лента **Предварительный просмотр (Print Preview)**.

Для просмотра ранее созданного отчета выберите в области навигации нужный отчет и в контекстном меню выполните команду **Предварительный просмотр (Print**

Preview). Отчет отобразится на экране таким, каким он будет напечатан. Для перехода от одной страницы отчета к другой можно использовать стандартное поле номера страницы в левом нижнем углу окна отчета.

Прежде чем переходить к печати отчета, необходимо уточнить параметры его страниц. Команда **Параметры страницы** (Page Setup), доступная в режиме предварительного просмотра отчета в группе **Разметка страницы** (Page Layout), открывает диалоговое окно **Параметры страницы** (Page Setup) (рис. 7.14), в котором можно выбрать принтер, задать формат и ориентацию (книжная, альбомная) бумаги, размер полей (вкладка **Параметры печати** (Print Options)), а также число, размер и макет столбцов. Эти же параметры представлены в группе **Размер страницы** (Page Size) отдельными кнопками. Кнопки **Поля** (Margins) и **Размер** (Size) открывают наглядные коллекции возможных вариантов полей страницы и размера бумаги.

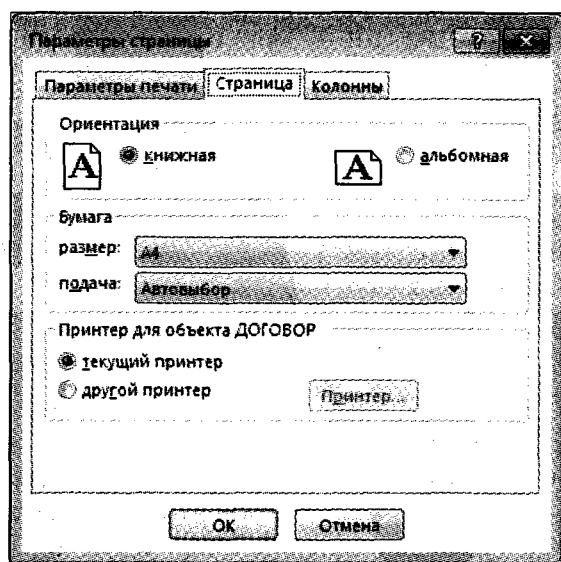


Рис. 7.14. Выбор параметров страницы для печати отчета

ЗАМЕЧАНИЕ

Окно **Параметры страницы** (Page Setup) открывается, только если на компьютере установлен локальный или сетевой принтер (причем установленный принтер может быть и недоступен). В противном случае выводится сообщение о необходимости установить его.

В примере выбрана бумага формата А4, книжная ориентация страниц отчета и использование принтера по умолчанию. Если для печати отчета нужно выбрать другой доступный принтер, отметьте соответствующий переключатель и нажмите кнопку **Принтер** (Printer). В раскрывающемся списке **Имя** (Name) открывшегося окна (рис. 7.15) выберите из доступных принтеров нужный и установите для него необходимые свойства. Access сохраняет заданные параметры страницы вместе с отчетом.

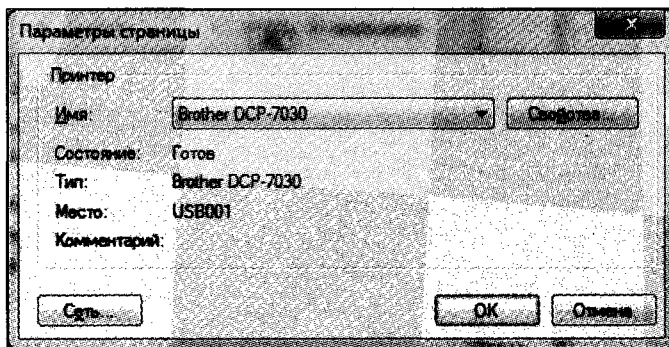


Рис. 7.15. Выбор принтера для печати отчета

Размещение разделов отчета на страницах регулируется их свойствами. Для вывода отчета по каждой группе с новой страницы нужно открыть свойства заголовка группы. На вкладке **Макет (Format)** в строке **Конец страницы (Force New Page)** выбрать значение **До раздела (Before Section)**. Сведения, которые необходимо отображать на каждой странице, размещаются в верхнем и нижнем колонтитулах. Для вывода заголовка отчета на отдельной странице в его свойстве **Конец страницы (Force New Page)** выбирается значение **После раздела (After Section)**.

Если в рассматриваемом примере требуется печатать отчет для каждого покупателя на отдельной странице, установите для заголовка группы код_пок в свойстве макета **Конец страницы (Force New Page)** значение **До раздела (Before Section)**.

Печать отчета

Для вывода отчета на печать нужно в режиме предварительного просмотра нажать кнопку **Печать (Print)**. В диалоговом окне печати (рис. 7.16) можно выбрать принтер, на который будет отправлен отчет, настроить его свойства, дополнительно проверить и при необходимости настроить параметры страницы, выбрать для печати отдельные страницы отчета, распечатать заданное число копий, вывести отчет в файл, который может распечатываться в другое время. Установив нужные для печати параметры, отчет можно отправить в очередь печати выбранного принтера.

Отчет печатается с использованием настроек, заданных в диалоговом окне **Печать (Print)**.

Чтобы напечатать отчет, его не обязательно открывать. Если нет необходимости проверить отчет, подправить его макет или убедиться, что будут напечатаны нужные данные, выберите отчет, который нужно напечатать, в области навигации. Нажмите кнопку **Файл (File)**, а затем выберите пункт **Печать (Print)**. Будет предоставлена возможность быстро напечатать отчет на используемом по умолчанию принтере без внесения каких-либо изменений, или отобразить диалоговое окно **Печать** для выбора принтера и параметров печати, или перейти к предварительному просмотру отчета и внесению изменений.

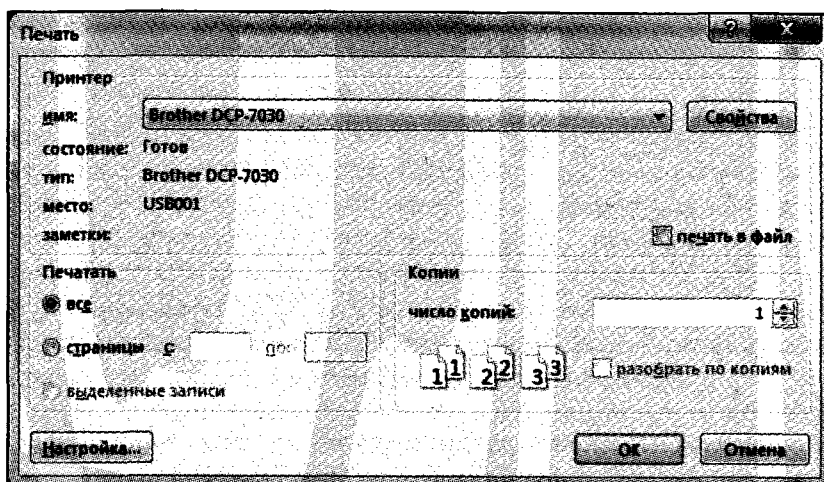


Рис. 7.16. Диалоговое окно Печать отчета

Изменение источника записей отчета

В режиме макета Access обеспечивает не только форматирование, но и изменение источника записей отчета простыми средствами добавления полей из списка всех таблиц базы данных.

Пусть необходимо в отчете о договорах покупателей вывести подробные сведения о покупателе.

Откройте отчет «Договоры фирмы» (см. рис. 7.9) в режиме макета. На вкладке ленты **Конструктор** (Design) в группе **Сервис** (Tools) выберите команду **Добавить поля** (Add Existing Fields) — в области **Список полей** (Field List) отобразятся поля, доступные в этом отчете. Щелкните в верхней части области на строке **Показать все таблицы** (Show all tables). В разделе **Доступные поля в связанной таблице** (Fields available in related tables) найдите таблицу **ПОКУПАТЕЛЬ**, в которой содержатся все сведения о покупателе.

Чтобы отобразить список полей таблицы, щелкните на знаке «плюс» (+) перед ее именем.

Для добавления нужного поля к отчету достаточно перетащить его из области **Список полей** (Field List) в нужный раздел отчета. Добавьте поле **НАИМ_ПОК** в заголовок группы **КОД_ПОК**. Чтобы поле попало в заголовок группы **КОД_ПОК**, поместите его над значением поля с кодом покупателя (рис. 7.17).

ЗАМЕЧАНИЕ

Двойной щелчок в поле приводит к его размещению в конце области данных.

После перемещения поля в отчет Access создает соответствующий элемент управления, связанный с полем. Кроме того, создается присоединенная подпись. Оба элемента управления размещаются в заголовке группы.



Рис. 7.17. Добавление поля в отчет в режиме макета

В списке полей таблица **ПОКУПАТЕЛЬ** переместится из раздела **Доступные поля в связанной таблице** (Fields available in related tables) в раздел **Поля, доступные в данном представлении** (Fields available for this view). При этом автоматически изменяется источник записей отчета. В соответствующий запрос на выборку — представление — добавляется таблица **ПОКУПАТЕЛЬ** и перемещенное поле. Таким образом, поля текущего источника записей будут дополнены полем **ИМЯ_ПОК**.

Из области полей можно продолжить перемещение нужных полей в отчет. При этом источник записей отчета будет автоматически подправляться.

Просмотреть источник записей можно в свойствах отчета. Откройте свойства отчета, щелкнув на кнопке **Страница свойств** (Property Sheet) на ленте **Конструктор** (Design). В окне свойств на вкладке **Данные** (Data) в строке **Источник записей** (Record Source) будет записана следующая инструкция SQL:

```
SELECT ДОГОВОР.*, ПОКУПАТЕЛЬ.ИМЯ_ПОК
FROM ПОКУПАТЕЛЬ INNER JOIN ДОГОВОР
ON ПОКУПАТЕЛЬ.КОД_ПОК = ДОГОВОР.КОД_ПОК;
```

Команда **Изменение источника записей** (Edit Record Source) из контекстного меню области **Список полей** (Field List) позволяет просмотреть и изменить соответствующий запрос в режиме конструктора.

Отчет в режиме предварительного просмотра после добавления в него поля из таблицы **ПОКУПАТЕЛЬ** и незначительных изменений принял вид, показанный на рис. 7.18.

Код покупателя	Номер договора	Дата	Сумма по договору	Код исполнителя
П001	Компьютер маркет			
	Д777	14.06.2016	1 579 920,00р	1
	Д222	05.02.2016	152 280,00р	1
	Д111	11.01.2016	6 414 000,00р	6
			8 146 200,00р	

Рис. 7.18. Первая страница отчета в режиме предварительного просмотра

Многотабличные отчеты

В предыдущем разделе отчет, автоматически созданный на одной таблице ДОГОВОР, был с помощью простого добавления полей из связанной таблицы превращен в отчет на основе запроса, объединяющего поля двух таблиц. Таким образом, было показано, как, не создавая заранее запрос, можно строить отчет на основе данных нескольких связанных таблиц непосредственно в режиме макета.

В рассмотренном примере в отчет добавлялись поля из таблицы ПОКУПАТЕЛЬ — главной по отношению к таблице ДОГОВОР — источнику записей отчета. Причем в отчете предварительно была выполнена группировка по коду покупателя, поэтому задача сводилась к добавлению полей в существующий заголовок группы.

В большинстве случаев создание отчета на основе нескольких таблиц, как путем перетаскивания полей таблиц в отчет, так и на основе ранее созданного запроса, требует значительной работы по настройке отчета. Некоторые операции по настройке такого отчета могут быть выполнены только в режиме конструктора.

Мастер отчетов позволяет в режиме диалога с пользователем создать многотабличный отчет путем выбора необходимых таблиц и полей, определения полей группировки, итоговых значений для записей. Создание отчета мастером является простой процедурой, а полученный отчет без больших усилий может быть приведен к желаемому виду.

Многотабличные отчеты, так же как формы, могут состоять из главного отчета и включаемого в него подчиненного отчета. Для каждого из этих отчетов в качестве источника данных выбираются своя таблица или запрос, построенный на нескольких таблицах. Чтобы обеспечить соответствие записей, выводящихся в подчиненном отчете, записям в главном отчете, устанавливается связь подчиненного отчета с главным. Однако мастер отчетов, в отличие от мастера форм, при выборе всех

необходимых таблиц — например, для разработки и печати документа «Договор», не создает такого составного отчета. Мастер отчетов решает эту задачу с помощью группировки.

Создать составной отчет можно, воспользовавшись мастером сначала для создания главного отчета, а затем для создания подчиненного отчета. При создании мастером главного отчета следует выбрать макет в столбец, а при создании подчиненного — макет табличный. На последнем этапе, открыв главный отчет в режиме макета или конструктора, надо перетащить в него подчиненный и, открыв свойства данных подчиненного отчета, установить его связь с главным.

Настройка отображения полученного отчета выполняется такими же средствами, как и настройка форм, достаточно просто, хотя и требует понимания назначения каждого из разделов отчета. Например, чтобы убрать заголовок/примечание подчиненного отчета, достаточно выполнить соответствующую команду контекстного меню. Чтобы отобразить названия столбцов в подчиненном отчете, достаточно в свойствах его макета для свойства **Отображать колонтитулы страниц** (Show Page Header and Page Footer) выбрать **Да**. Чтобы отчет содержал на одной странице сведения из одной записи главной формы и связанных с ней записей подчиненной формы, т. е. отображал один документ, — например, выводил данные об одном договоре, достаточно установить для области данных главного отчета в свойстве макета **Конец страницы** (Force New Page) значение **До раздела** (Before Section). Свойство **Конец страницы** (Force New Page) со значением **После раздела** (Section) для заголовка главного отчета позволит отображать и печатать его на отдельной странице.

Рассмотрим создание многотабличного отчета сложной структуры с помощью мастера, который создаст на основе выбранных пользователем таблиц запрос — источник записей отчета, разместит в отчете поля, создаст необходимые группировки, итоги и т. д. При этом отчет за счет группировки, выполненной в нем, может приобрести такой же вид, как и отчет, состоящий из главной и подчиненной табличной частей, т. е. группировкой можно заменить создание составного отчета.

Для придания отчету окончательного вида рассмотрим возможности инструментария Access, предоставляемого в режиме макета и конструктора.

Разработка отчета с помощью мастера

Пусть необходимо подготовить отчет, позволяющий распечатать документ «Договор», на основе данных, ранее загруженных в базу с помощью формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** (см. главу 6).

Предположим, что макет отчета должен иметь вид, представленный на рис. 7.19. В соответствии с проектом макета в главной части отчета предполагается выводить данные по договору, включая сведения о покупателе, заключившем договор. Эти данные содержатся в таблицах **ДОГОВОР** и **ПОКУПАТЕЛЬ**. В табличной части необходимо вывести построчно данные о заказанных в договоре товарах, которые содержатся в таблицах **ПОСТАВКА_ПЛАН** и **ТОВАР**.

Договор № _____		от _____					
Покупатель _____	Адрес _____	Телефон _____					
ИНН _____							
Банк _____							
Расчетный счет _____							
Наименование товара	Срок поставки	Мин. партия	Количество	Цена	Единица измерения	Ставка НДС	Стоимость
_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____
							Общая сумма _____

Рис. 7.19. Проект макета отчета для вывода данных по договору фирмы

Выбор таблиц для отчета

Для создания отчета мастером выберем в области навигации таблицу ДОГОВОР, необходимую для формирования источника записей отчета. Первая выбранная таблица определит имя и заголовок отчета по умолчанию, а также вид представления данных в отчете.

Выполним команду **Мастер отчетов (Report Wizard)** в группе **Отчеты (Reports)** на вкладке ленты **Создание (Create)**. В первом открывшемся окне мастера **Создание отчетов (Report Wizard)** отобразится таблица ДОГОВОР и список ее полей (рис. 7.20). В этом окне необходимо выбрать поля этой таблицы, а также все другие таблицы и поля, включаемые в отчет.

В примере в соответствии с проектом отчета (см. рис. 7.20) выберем таблицы:

ДОГОВОР, из нее поля:

- номер договора — НОМ_ДОГ;
- дата заключения договора — ДАТА_ДОГ;

ПОКУПАТЕЛЬ, из нее поля:

- идентификационный номер налогоплательщика — ИНН;
- наименование покупателя — НАИМ_ПОК;
- адрес покупателя — АДРЕС_ПОК;
- телефон — ТЕЛ;
- банк, который обслуживает счет покупателя, — БАНК;
- номер расчетного счета покупателя — НОМ_РСЧ;

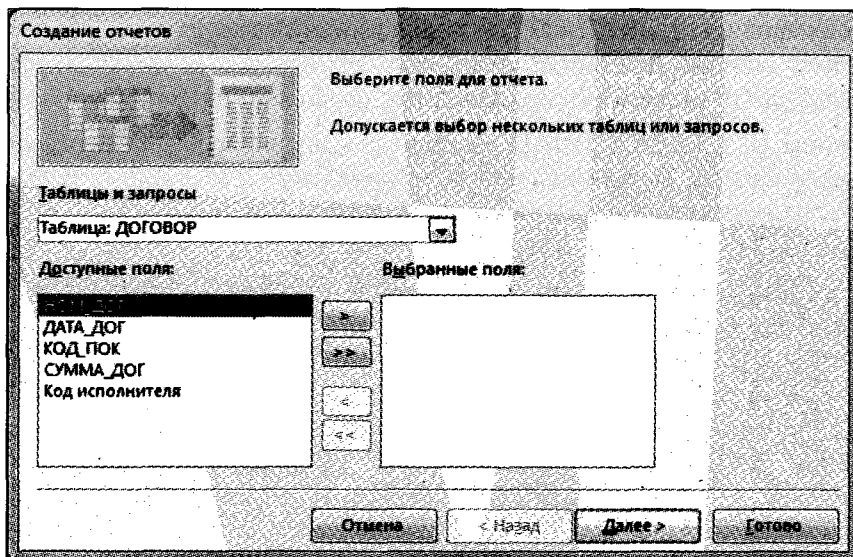


Рис. 7.20. Первое окно мастера создания отчетов

□ **ПОСТАВКА_ПЛАН**, из нее поля:

- срок поставки — СРОК_ПОСТ;
- минимальная партия поставки — МИН_ПОСТ;
- количество товара — КОЛ_ПОСТ;

□ **ТОВАР**, из нее поля:

- наименование товара — НАИМ_ТОВ;
- цена — ЦЕНА;
- единица измерения — ЕИ;
- ставка НДС — СТАВКА_НДС.

При выборе взаимоовязанных таблиц мастер отчетов в соответствии со схемой базы данных «Поставка товаров» (см. рис. 3.54 в главе 3) автоматически установит связи между ними и построит запрос, объединяющий записи этих таблиц и включающий в таблицу запроса указанные поля. Этот запрос будет использован в качестве источника данных отчета, а соответствующая инструкция SQL записана мастером в свойство отчета **Источник записей (Record Source)**. Объединение записей таблиц будет производиться способом, указанным параметрами объединения для каждой из связей в схеме данных. Подсхема данных, включающая таблицы, на которых будет строиться отчет, приведена на рис. 7.21.

В окне мастера (рис. 7.22) выберем вид представления данных в отчете. Чтобы получить отчет, заданный в рассматриваемом примере, выберем таблицу ДОГОВОР в качестве основы главной части отчета и увидим, что мастером на основе полей этой таблицы будет осуществлена группировка. К полям группировки будут отне-

сены и поля таблицы ПОКУПАТЕЛЬ, поскольку эта таблица является главной по отношению к таблице ДОГОВОР. Это позволяет дополнить записи подчиненной таблицы ДОГОВОР сведениями о покупателе, заключившем договор.

Поля таблицы ПОСТАВКА_ПЛАН составят основу записей для построения табличной части отчета. Причем записи о товарах, перечисленных в этой таблице, будут дополнены реквизитами из таблицы ТОВАР.

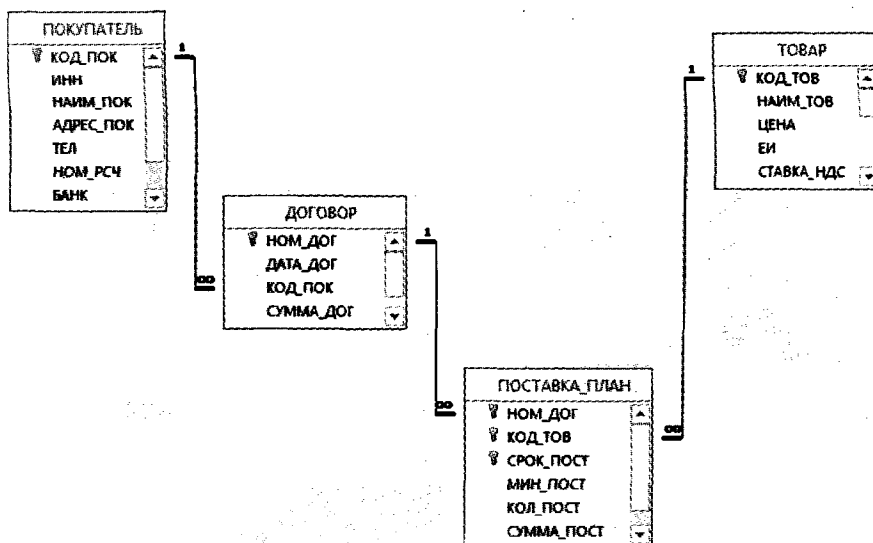


Рис. 7.21. Подсхема данных отчета «Договоры фирмы»

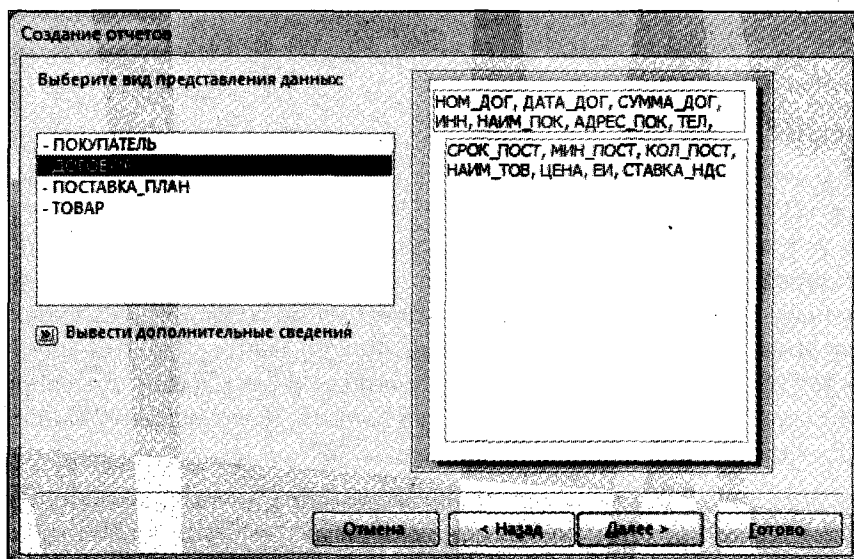


Рис. 7.22. Окно мастера отчетов при выборе вида представления данных

Если, выбирая вид представления данных, остановиться на таблице ПОКУПАТЕЛЬ, мастер создаст два уровня группировки: на первом будут собраны данные о покупателе, на втором — о его договорах (рис. 7.23).

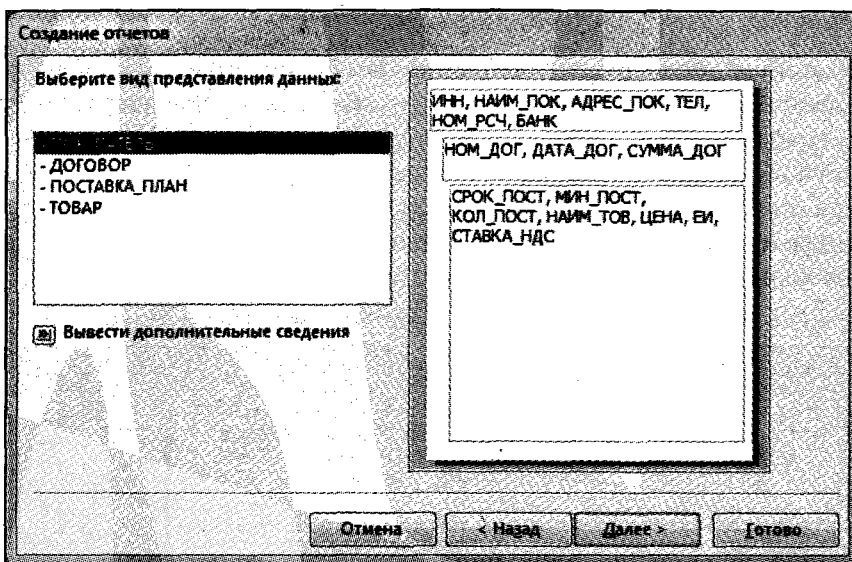


Рис. 7.23. Окно мастера отчетов при выборе представления с двумя уровнями группировки

В следующем окне мастера (рис. 7.24) можно добавить уровни группировки. Если в предыдущем окне группировка производилась в соответствии с уровнями иерархии таблиц в схеме данных, что позволило отразить их подчиненность в отчете, то

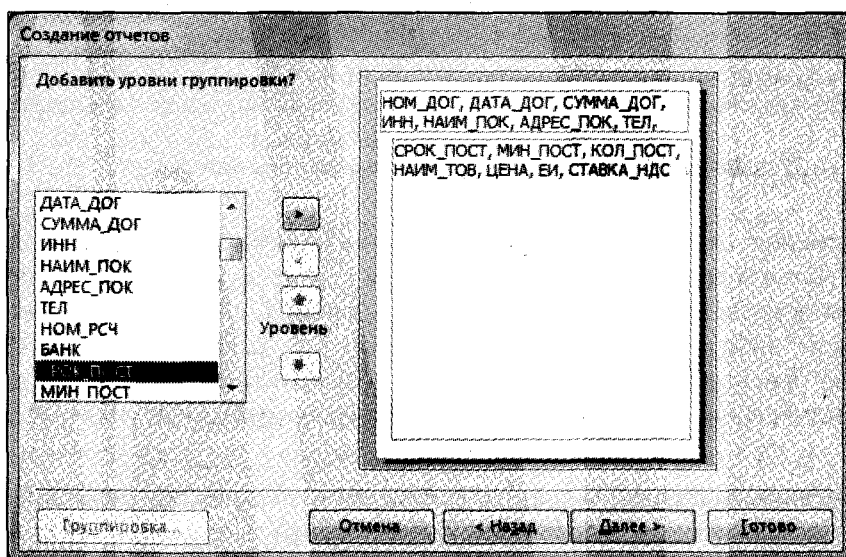


Рис. 7.24. Окно мастера отчетов при добавлении новых уровней группировки

в этом окне добавление группировки может быть выполнено по любому из полей таблицы на любом уровне иерархии. Очевидно, что группировка должна иметь смысл. Например, в наборе строк из таблицы ПОСТАВКА_ПЛАН можно выполнить группировку по сроку поставки. В отчете, создаваемом мастером, можно определить до четырех уровней группировки.

Чтобы добавить уровень группировки, выполните двойной щелчок на добавляемом в отчет имени поля. Для удаления уровня группировки — двойной щелчок на нем на экране страницы в правой части диалогового окна. Для добавления и удаления уровней группировки можно также воспользоваться кнопками со стрелками. Таким образом добавляется очередной уровень группировки, который отображается вложенным в предыдущий уровень группировки. Пример с включением уровня группировки по полю СРОК_ПОСТ (срок поставки) показан на рис. 7.25.

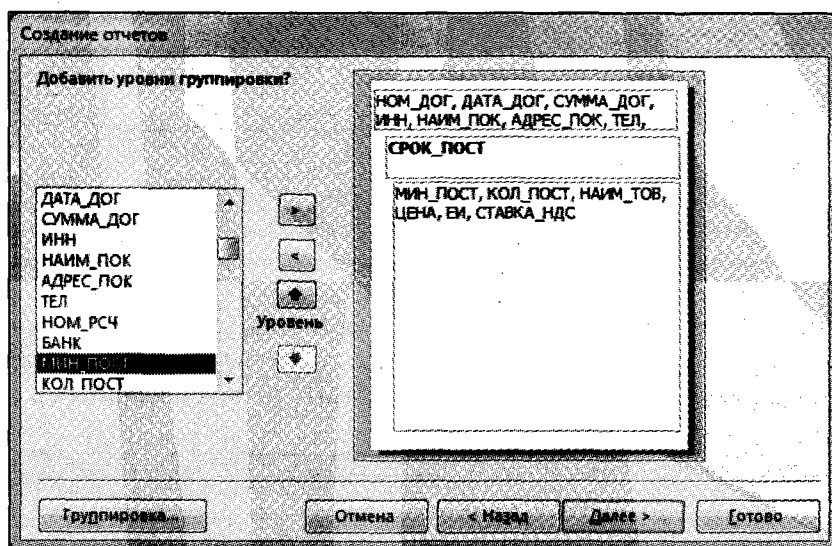


Рис. 7.25. Окно мастера отчетов при добавлении уровня группировки по сроку поставки

Для построения отчета остановимся на варианте группировки, показанном на рис. 7.25. Далее в диалоге с мастером выберем сортировку по наименованию товара, причем подведение итогов выполнять не станем, т. к. в нашем примере это не имеет какого-либо заметного смысла. Выберем вид макета структура (Stepped), альбомную ориентацию страниц отчета, попросим мастера настроить ширину полей для размещения на одной странице и выберем требуемый стиль оформления отчета.

Дадим отчету имя «Договоры фирмы» и в качестве дальнейших действий выберем **Изменить макет отчета** (Modify the report's design). Результат работы мастера представлен на рис. 7.26.

Рис. 7.26. Отчет на основе нескольких взаимосвязанных таблиц (режим конструктора)

Источник записей отчета

Мастер на основе выбранных таблиц создает запрос. Просмотрим его, открыв свойства отчета. Для этого в режиме конструктора в любом месте окна отчета откроем контекстное меню и выполним команду **Свойства отчета (Report Properties)**. В окне свойств отчета на вкладке **Данные (Data)** в строке **Источник записей (Record Source)** содержится инструкция SQL:

```
SELECT [ДОГОВОР].[НОМ_ДОГ], [ДОГОВОР].[ДАТА_ДОГ], [ПОКУПАТЕЛЬ].[ИНН],
[ПОКУПАТЕЛЬ].[НАИМ_ПОК], [ПОКУПАТЕЛЬ].[АДРЕС_ПОК],
[ПОКУПАТЕЛЬ].[ТЕЛ], [ПОКУПАТЕЛЬ].[НОМ_РСЧ], [ПОКУПАТЕЛЬ].[БАНК],
[ПОСТАВКА_ПЛАН].[СРОК_ПОСТ], [ПОСТАВКА_ПЛАН].[МИН_ПОСТ],
[ПОСТАВКА_ПЛАН].[КОЛ_ПОСТ], [ТОВАР].[НАИМ_ТОВ], [ТОВАР].[ЦЕНА],
[ТОВАР].[ЕИ], [ТОВАР].[СТАВКА_НДС]
FROM ТОВАР INNER JOIN ((ПОКУПАТЕЛЬ INNER JOIN ДОГОВОР
ON ПОКУПАТЕЛЬ.КОД_ПОК=ДОГОВОР.КОД_ПОК) INNER JOIN
ПОСТАВКА_ПЛАН ON ДОГОВОР.НОМ_ДОГ=ПОСТАВКА_ПЛАН.НОМ_ДОГ)
ON ТОВАР.КОД_ТОВ=ПОСТАВКА_ПЛАН.КОД_ТОВ;
```

Инструкция **SELECT** определяет список полей запроса на выборку, а ее предложение **FROM** — последовательность и способ объединения таблиц.

Значком построителя, размещенным справа в строке **Источник записей (Record Source)**, можно открыть инструкцию SQL в режиме конструктора запросов. Запрос, являющийся источником записей отчета, можно изменить в любом из этих режимов: добавить или удалить поля, таблицы.

ЗАМЕЧАНИЕ

Если отчет создается в режиме конструктора или с помощью средства Пустой отчет (Blank Report), которое отличается тем, что открывает пустой отчет в режиме макета, прежде всего, необходимо определить источник записей. Задается источник записей в свойствах отчета, как в рассмотренном примере, путем записи инструкции SQL или с помощью построителя запросов. Кроме того, запрос может быть создан путем перетаскивания полей из области Список полей (Field List), доступной как в режиме конструктора, так и в режиме макета.

Доработка отчета в режиме конструктора

Для того чтобы отчет соответствовал проекту, его надо доработать в режиме конструктора.

После создания отчета мастером надписи полей группировки и их значения размещены в заголовке группы (см. рис. 7.26). Здесь же размещены надписи столбцов

ДОГОВОРЫ ФИРМЫ

Заголовок отчета

ДОГОВОРЫ ФИРМЫ

Верхний колонтитул

Заголовок группы 'НОМ ДОГ'

Договор №	НОМ ДОГ	от	ДАТА ДОГ
-----------	---------	----	----------

Покупатель

НАИМ ПОК	Адрес	АДРЕС ПОК	Тел	ТЕЛ
----------	-------	-----------	-----	-----

ИНН

ИНН

БАНК

БАНК

НОМ РСЧ

НОМ РСЧ

Заголовок группы 'СРОК ПОСТ'

Срок поставки	СРОК ПОСТ
---------------	-----------

Количество	Наименование	Цена	Единица	Ставка НДС	Стоимость
	товара		измерения		

Область данных

КОЛ ПОСТ	НАИМ ТОВ	ЦЕНА	ЕИ	СТАВКА НДС	= [ЦЕНА] * [КОЛ ПОСТ]
----------	----------	------	----	------------	-----------------------

Примечание группы 'СРОК ПОСТ'

Сумма договора за месяц	= Sum([ЦЕНА] * [КОЛ ПОСТ])
-------------------------	----------------------------

Примечание группы 'НОМ ДОГ'

Общая сумма договора	= Sum([ЦЕНА] * [КОЛ ПОСТ])
= ИИ([Поле59] > 1000000; "Обратите внимание; ")	

Нижний колонтитул

=Now()	= "Стр. " & [Page] & " из " & [Pages]
--------	---------------------------------------

Примечание отчета

Рис. 7.27. Отчет о договорах фирмы, доработанный в режиме конструктора

табличной части отчета. Это позволяет выводить названные элементы один раз в начале каждого договора.

Используя средства конструктора отчетов, разместим и отформатируем надписи и поля в разделах отчета, как показано на рис. 7.27.

Чтобы каждый договор отображался с новой страницы отчета, установим для заголовка группы **НОМ_ДОГ** значение свойства макета **Конец страницы** (Force New Page) равным **До раздела** (Before Section). Если договор занимает несколько страниц, для отображения реквизитов договора и заголовков столбцов на каждой странице установим для заголовка группы значение **Да** в свойстве макета **Повторение раздела** (Repeat Section). При этом на каждой странице одного договора будут повторяться как надписи столбцов табличной части отчета, так и все общие реквизиты договора.

Если необходимо в отчете сформировать титульную страницу, установите в свойствах заголовка отчета на вкладке **Макет** (Format) в строке **Конец страницы** (Force New Page) значение **После раздела** (After Section). Увеличив размер области заголовка отчета до размера страницы, можно придать титульной странице необходимый вид.

Вычисляемые поля в отчете

Рассчитаем стоимость каждого из товаров, заказанных в договоре. Для этого дополним перечень полей области данных вычисляемым полем и запишем в него выражение: $=[\text{ЦЕНА}] * [\text{КОЛ_ПОСТ}]$. Вычисляемое поле создадим, открыв отчет в режиме конструктора. Для этого щелкнем на кнопке **Поле** (Text Box) на ленте **Конструктор** (Design) в группе **Элементы управления** (Controls) и перетащим его в конец области данных. В созданном таким образом элементе **Свободный** (Unbound) запишем нужное выражение. Вырежем надпись этого поля, вставим в заголовок группы и переместим в нужное место. Заменяем ее значение на **Стоимость**.

Это выражение будет внесено в свойства элемента. Откроем свойства элемента на вкладке **Данные** (Data) и увидим выражение в строке **Данные** (Control Source). На вкладке **Макет** (Format) в строке **Число десятичных знаков** (Decimal Places) выберем точность значения — **0**, в строке **Формат поля** (Format) — денежный формат отображения значения поля.

Рассчитаем стоимость всех товаров, заказанных в договоре. Для создания вычисляемого поля с итоговым значением по группе записей его надо разместить в разделе примечания группы. Чтобы отобразить примечание группы, откроем область **Группировка, сортировка и итоги** (Group, Sort, and Total), щелкнув на кнопке **Группировка** (Group & Sort) на вкладке ленты **Конструктор** (Design). В строке **Группировка НОМ_ДОГ** (Group on НОМ_ДОГ) откроем дополнительные параметры группировки, щелкнув на значке **Больше** (More), и выберем значение параметра с **разделом примечания** (with a footer section) в списке **без раздела примечания** (without a footer section).

В режиме конструктора отчета создадим в примечании группы, предварительно расширив его, вычисляемый элемент **Свободный** (Unbound). В этот элемент может быть записано выражение суммирования, расчета среднего арифметического, под-

счета числа записей в группе и др. для группы. Запишем в него выражение: $=\text{Sum}([\text{КОЛ_ПОСТ}] * [\text{ЦЕНА}])$. Так же как для вычисляемого поля **Стоимость**, установим точность значения и формат отображения. Подпись изменим на **Общая сумма договора**.

Если необходимо подсчитать общую сумму по всем договорам, создайте вычисляемое поле с таким же выражением, как при подсчете общей суммы по одному договору: $=\text{Sum}([\text{КОЛ_ПОСТ}] * [\text{ЦЕНА}])$, но в области примечания или заголовка отчета. Подпись поля изменим на **Общая сумма по всем договорам**.

Пусть необходимо особо пометить договоры, общая сумма которых составляет более миллиона рублей. Создадим в примечании группы вычисляемое поле и запишем в него выражение: $=\text{Iif}([\text{Поле52}] > 1000000; \text{'Обратите_внимание'}; \text{''})$, где Поле52 — имя поля со значением общей суммы, созданного ранее. Для этого вычисляемого поля выберем новые значения для шрифта, размера шрифта, насыщенности и цвета текста. Надпись поля удалим. Для договора на сумму более миллиона рублей в отчете будет выводиться фраза **Обратите внимание**, оформленная в соответствии со сделанным выбором. Для остальных договоров никакие слова не

ДОГОВОРЫ ФИРМЫ					
Договор №	D111			от	11.01.2016
Покупатель	Компьютер маркет	Адрес	Москва	Тел.	(812)345-23-45
ИНН	778957651111				
БАНК	Мост				
НОМ_РСЧ	7635850976326453656				
Срок поставки		1			
Количество	Наименование товара	Цена	Единица измерения	Ставка НДС	Стоимость
50	FDD 3.5	500.00р.	коробка	6%	25 000р.
100	HDD Maxtor 12	1 120.00р.	штучки	5%	112 000р.
710	CD-ROM Paha	800.00р.	штучки	30%	568 000р.
305	Монитор 17LG	7 000.00р.	штука	5%	2 135 000р.
Сумма договора за месяц					2 840 000р.
Срок поставки		2			
Количество	Наименование товара	Цена	Единица измерения	Ставка НДС	Стоимость
200	Монитор 17LG	7 000.00р.	штука	5%	1 400 000р.
10	Принтер EPS	2 400.00р.	штука	10%	24 000р.
Сумма договора за месяц					1 424 000р.
Общая сумма договора					4 264 000р.
Обратите_внимание					
19 июля 2016 г.					
					Стр 2 из 8

Рис. 7.28. Страница отчета с данными договора в режиме предварительного просмотра

выводятся, т. к. третий операнд функции `iff` задает в качестве значения пустую строку.

Страница отчета с данными по одному договору в режиме предварительного просмотра приведена на рис. 7.28. В этом примере заголовок отчета «Договоры фирмы» и все добавленные в него сведения, рисунки, графики выводятся на отдельном титульном листе один раз на первой странице отчета.

Для нумерации строк в табличной части отчета создайте в области данных вычисляемое поле, в котором запишите выражение `=1`. Затем для этого поля установите на вкладке **Данные (Data)** в свойстве **Сумма с накоплением (Running Sum)** значение **Для группы (Over Group)**. Не следует использовать это свойство для вычисляемых полей примечания.

Вывод в отчете итогов по группе и отчету в целом может быть реализован и просто командой **Σ Итоги** в группе **Группировка и итоги** на ленте **Конструктор**. Заметим, что по вычисляемому полю, такому как **Стоимость**, таким образом вывести итоговые значения нельзя. При использовании вычисляемых полей с итогами можно просмотреть отчет без подробностей, т. е. с отображением только итоговых данных.

Определение параметров в отчете

Пусть необходимо в отчете получать данные только о заданном договоре. Эта задача может быть решена с помощью ввода параметра в запрос, на котором основывается отчет.

Откроем отчет «Договоры фирмы» в режиме конструктора. В свойствах отчета на вкладке **Данные (Data)** в конце строки **Источник записей (Record Source)** щелкнем на кнопке построителя. В окне построителя запросов перейдем к полю с наименованием покупателя — `НАИМ_ПОК`. В строке **Условие отбора (Criteria)** определим параметр, записав его имя в квадратных скобках, например, `[Наименование покупателя]`.

Теперь при просмотре отчета будет выводиться диалоговое окно ввода значения параметра (рис. 7.29). После ввода значения параметра отчет будет содержать договоры только одного покупателя.

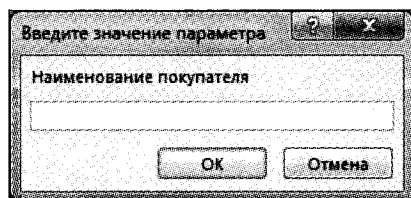


Рис. 7.29. Диалоговое окно ввода значения параметра отчета

Чтобы введенные значения параметров отображались в просматриваемом или напечатанном отчете, выполните следующие действия. Откройте отчет в режиме конструктора. Вставьте в отчет элемент управления *вычисляемое поле*. Для этого нажмите на вкладке **Конструктор (Design)** кнопку элемента управления **Поле (Text**

Box) и вычертите элемент в месте, куда нужно его поместить. Запишите в это новое поле выражение, содержащее только имя параметра. Параметр должен записываться точно так же, как он записан в условии отбора запроса, — например: =[Наименование покупателя].

Для просмотра введенного значения параметра в отчете откройте его в режиме предварительного просмотра. Значение параметра отобразится в точности так, как было введено в диалоговом окне.

Анализ данных отчета средствами фильтрации

В отчете с параметром вы получили сведения лишь о договорах одного покупателя и не можете всесторонне в интерактивном режиме проанализировать данные о всех договорах фирмы. Для выборки и анализа различных сведений из договоров вам поможет использование фильтров.

Удалите параметр из запроса — источника записей отчета. Откройте отчет в режиме **Представление отчета (Report View)**. Пользуйтесь для переключения режимов строкой состояния.

Установите курсор на поле с номером договора. Щелкните на кнопке **Фильтр (Filter)**. В открывшемся окне откройте список **Текстовые фильтры (Text Filters)**. Для поля доступны эти фильтры, потому что поле имеет текстовый тип данных. В списке выберите опцию **Содержит (Contains)** — откроется настраиваемый фильтр (рис. 7.30), в котором укажите нужный номер договора. Для открытия списка **Текстовые фильтры** можно также воспользоваться контекстным меню поля — в отчете отобразится выбранный договор.

Чтобы снять фильтр с отчета, используйте соответствующую команду контекстного меню поля. Если необходимо, задайте новые условия отбора по полю с номером договора.

Чтобы отобразить договоры за заданный период, в контекстном меню поля с датой заключения договора откройте **Фильтр дат (Date Filters)**.

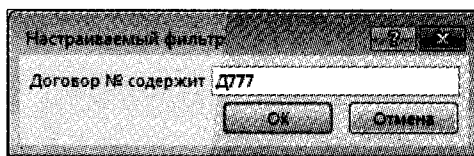


Рис. 7.30. Диалоговое окно задания условия отбора для фильтра

ОБРАТИТЕ ВНИМАНИЕ

Итоговые данные по договору подсчитываются только для выбранных в результате фильтрации данных.

Для отбора договоров, в которых заказан заданный товар, используйте текстовый фильтр по полю с наименованием товара. Для отбора заданного товара, заказанного в заданном месяце, не снимая фильтра с наименования товара, выполните фильтра-

цию по полю СРОК_ПОСТ (срок поставки). Числовой фильтр позволит отобразить данные в любом заданном интервале.

Таким образом, использование фильтра позволит без изменения отчета получать данные в различном представлении. Самые сложные данные можно представить в удобном для понимания виде, создавая в интерактивном режиме отчеты для определенных целей.

Задание 7.1. Создайте отчет для вывода договоров покупателя

Создайте с помощью мастера отчет для вывода договоров, сгруппированных по заказчикам — покупателям. Обеспечьте вывод итоговых значений по каждому договору и по покупателю в целом. В предметной области указано, что один покупатель может заключать с фирмой несколько договоров.

Задание 7.2. Создайте отчет для вывода накладных

Создайте отчет для вывода накладных, сгруппированных по договорам и месяцам. Обеспечьте вывод итоговых значений по накладным месяца и по накладным договора в целом. По одному договору может выписываться много накладных. Обеспечьте вывод отчета об отгрузках по заданному договору.

Составные отчеты

Пусть необходимо получить отчет об отгрузках товаров за весь учетный период. Отчет должен содержать все данные о товаре и список всех отгрузок товара, сгруппированных по месяцам (рис. 7.31).

Для создания отчета, соответствующего заданию, потребуется использовать данные из следующих взаимосвязанных таблиц:

□ ТОВАР, поля:

- код товара — КОД_ТОВ;
- наименование товара — НАИМ_ТОВ;
- цена — ЦЕНА;
- единица измерения — ЕИ;
- ставка НДС — СТАВКА_НДС;
- фотография товара — ФОТО;

□ ОТГРУЗКА, поля:

- код склада — КОД_СК;
- код товара — КОД_ТОВ;
- количество отгруженного товара — КОЛ_ОТГР;

□ НАКЛАДНАЯ, поле:

- дата отгрузки — ДАТА_ОТГР.

ОТГРУЗКА ТОВАРА

Код товара _____
 Наименование товара _____
 Цена _____
 Единица измерения _____
 Ставка НДС _____

Фото товара

	Код склада	Код товара	Количество	Дата
Месяц	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Число отгрузок: _____ Количество отгруженного: _____

ИТОГО ОТГРУЖЕНО _____

Рис. 7.31. Проект макета для создания отчета об отгрузке товаров

Связи указанных таблиц определены в подсхеме данных, представленной на рис. 7.32.

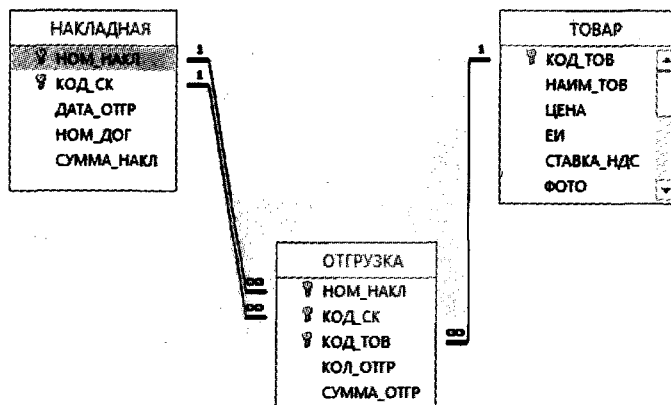


Рис. 7.32. Подсхема данных отчета об отгрузке товаров

Очевидно, что главную часть отчета должны составить данные из таблицы ТОВАР, табличную часть отчета — данные из подчиненной таблицы ОТГРУЗКА. Причем записи таблицы ОТГРУЗКА должны быть дополнены датой, которая хранится в главной по отношению к ней таблице НАКЛАДНАЯ.

В соответствии с этим создадим составной отчет, содержащий главный отчет и включенный в него подчиненный отчет.

Создание главного отчета

Главная часть составного отчета должна содержать все сведения о товаре, которые хранятся в таблице базы данных ТОВАР, и может быть создана как обычный однотоабличный отчет о товарах фирмы. Для автоматического создания отчета, включающего все поля одной таблицы, достаточно выбрать в области навигации таблицу ТОВАР и выполнить команду **Отчет (Report)** на вкладке ленты **Создание (Create)**. Таблица ТОВАР будет указана в качестве источника данных этого отчета.

Созданный однотоабличный отчет откроется в режиме макета. В качестве заголовка отчета будет использовано имя таблицы. Он включает все поля таблицы. Присоединенные элементы управления для каждого поля отчета связаны с подписями и образуют один табличный макет, размещенный в области данных.

В режиме макета пунктирными линиями показаны границы текста на странице, что позволяет правильно выбрать местоположение и размер элементов управления.

Можно создать отчет и с помощью команды **Пустой отчет (Blank Report)**, которая открывает пустой макет, и команды **Список полей (Field List)**. Перетащив нужные поля таблицы в макет отчета, можно получить такой же отчет, как и по команде **Отчет (Report)**. Область **Список полей** может быть открыта командой **Добавить поля (Add Existing Fields)** на вкладке ленты **Конструктор (Design)** в группе **Сервис (Tools)**.

Поскольку в главной части отчета должны отображаться данные только одной записи, преобразуем его макет в столбик. Для этого выделим табличный макет, щелкнув в его левом верхнем углу, и выполним команду **В столбик (Stacked)** на ленте **Упорядочить (Arrange)** в группе **Таблица (Table)**.

Для того чтобы сделать отчет более компактным, переместим поле с фотографией товара в верхнюю правую часть отчета, освободив место для подчиненного отчета. Перемещать отдельный элемент управления за пределы макета, а также удалять из отчета можно только после его удаления из макета. Для удаления из макета поля с фотографией и его надписи выделим их и выполним соответствующую команду контекстного меню. После исключения элементов управления из макета, не снимая выделения, переместим их в новое место.

Удалим подпись поля с фотографией, выделив ее и нажав клавишу **<Delete>**.

Размещение элементов в макете значительно упрощает их упорядочение в отчете, но требует выполнения операций по объединению элементов управления в макет и удаления из него.

Переключимся в режим конструктора, чтобы хорошо видеть структуру разделов отчета. Для переключения щелкнем на соответствующем значке в строке состояния. Отчет с данными о товарах в режиме конструктора представлен на рис. 7.33.

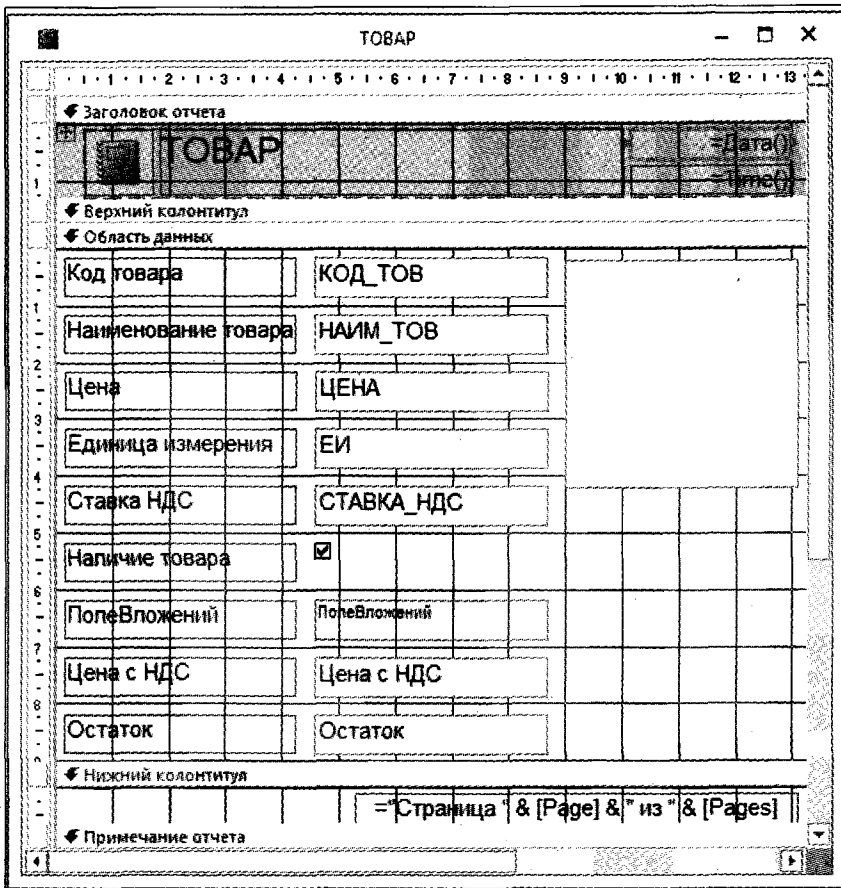


Рис. 7.33. Отчет о товарах фирмы в режиме конструктора

Создание подчиненного отчета

Табличная часть отчета должна содержать все сведения об отгрузках, которые хранятся в таблицах ОТГРУЗКА и НАКЛАДНАЯ. Создадим на основе этих таблиц отчет, который будет включен в главный отчет в качестве подчиненного.

Создадим с помощью команды **Отчет (Report)** по таблице ОТГРУЗКА однотоабличный отчет, который откроется в режиме макета. Из области **Список полей (Field List)** добавим в отчет из таблицы НАКЛАДНАЯ поле даты отгрузки: `ДАТА_ОТГР` — источник записей отчета автоматически дополнится выбранным полем.

Табличный макет этого отчета вполне подходит для реализации нужного представления. Подберем ширину элементов управления таким образом, чтобы все столбцы отчета поместились на странице, по умолчанию рассчитанной на размер бумаги А4 и книжную ориентацию. Для того чтобы получить возможность уменьшить ширину столбцов, сохранив полное отображение содержимого, разместим подписи в заголовках столбцов в двух строках, увеличив их высоту.

Для задания интервала между элементами управления и линиями сетки макета выделите макет отчета, щелкнув в его левом верхнем углу, и выберите нужный интервал из списка команды **Внутренние поля** (Control Padding), размещенной на ленте **Упорядочить** (Arrange) в группе **Положение** (Position).

Сортировка и группировка записей отчета

Для группировки данных отчета по дате отгрузки выполним команду **Группировка** (Group & Sort) на ленте **Конструктор** (Design). В открывшейся области **Группировка, сортировка и итоги** (Group, Sort, and Total) щелкнем на кнопке **Добавить группировку** (Add a group). Выберем для предлагаемого уровня группировки поле **ДАТА_ОТПР**. Откроем список, щелкнув на параметре **по кварталам** (by quarter), и выберем **по месяцам** (by month) (рис. 7.34). Для закрытия списка щелкните на любом месте за его пределами.

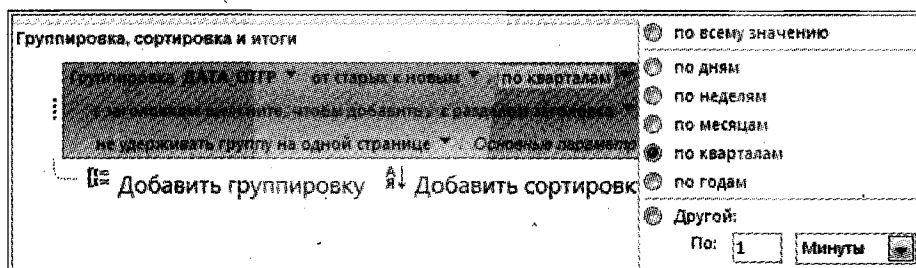


Рис. 7.34. Выбор для поля с датой группировки по месяцам

Для добавления в отчет итоговых сумм по каждому месяцу и общей суммы щелкните на кнопке **Больше** (More), в области параметров откройте список значений параметра с **итогами**: **СУММА_ОТПР** (with СУММА_ОТПР totaled). В поле списка **Итог по полю** (Total On) выберите поле **СУММА_ОТПР**, по которому нужно рассчитать итоговые значения. Отметьте флажок **Показать общий итог** (Show Grand Total) и определите место отображения итоговых значений по каждому месяцу выбором флажка **Показать в примечании группы** (Show in group footer) (рис. 7.35).

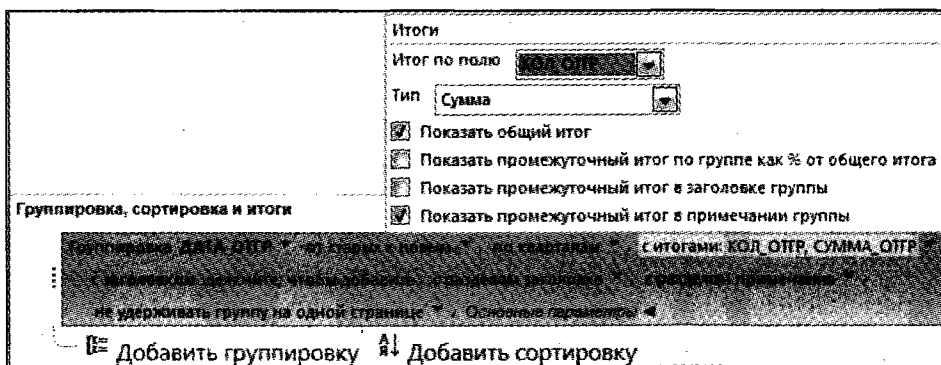


Рис. 7.35. Группировка с итогами: по двум полям: КОЛ_ОТПР и СУММА_ОТПР

Поскольку нам нужно подсчитать итоги и по количеству отгруженного, в раскрываемом списке с **итогами**: выберем поле КОД_ОТГР и отметим те же флажки.

ЗАМЕЧАНИЕ

В этом отчете расчет итоговых значений для поля с количеством отгруженного не имеет смысла, т. к. происходит сложение количеств разных товаров. Однако после вставки этого отчета в главный отчет, из общего набора строк будет отбираться только строки, связанные с конкретным товаром, и, следовательно, будет суммироваться количество для одного товара.

В качестве значения параметра **Порядок сортировки** по умолчанию выбирается устраивающее нас значение **от старых к новым (from oldest to newest)**.

Отчет в режиме макета после внесения перечисленных изменений представлен на рис. 7.36.

Закроем отчет, сохранив его под данным по умолчанию именем **ОТГРУЗКА**.

	Дата отгрузки	Код накладной	Код склада	Количество отгружено	Сумма отгружено
Январь 2016					
	18.01.2016	Н001	С01	10	22 000,00р.
	18.01.2016	Н001	С01	50	350 000,00р.
	18.01.2016	Н001	С01	10	2 338,00р.
	18.01.2016	Н001	С01	2	8 000,00р.
	18.01.2016	Н001	С01	10	5 000,00р.
	Итого за месяц			82	387 338,00р.
Февраль 2016					
	21.02.2016	Н004	С02	10	70 000,00р.
	21.02.2016	Н004	С02	10	5 000,00р.
	04.02.2016	Н001	С03	10	7 000,00р.
	21.02.2016	Н004	С02	1	1 120,00р.
	21.02.2016	Н004	С02	100	70 000,00р.
	11.02.2016	Н002	С02	6	4 800,00р.
	11.02.2016	Н002	С02	55	27 500,00р.
	11.02.2016	Н002	С01	40	280 000,00р.
	11.02.2016	Н002	С01	30	165 000,00р.

Рис. 7.36. Подчиненный отчет ОТГРУЗКА в режиме макета

Включение подчиненного отчета

Если отчет **ТОВАР** закрыт, откройте его в режиме конструктора. В области навигации выделите имя отчета **ОТГРУЗКА** и перетащите в его область данных главного отчета **ТОВАР** — отчет **ОТГРУЗКА** встраивается в качестве подчиненного в главный отчет, где он доступен для редактирования.

Access автоматически устанавливает связь между главным и подчиненным отчетами по полю КОД_ТОВАР (код товара), т. к. она определена в схеме данных. При этом не имеет значения, включено ли поле связи в подчиненный отчет, но необходимо, чтобы оно было в его источнике записей. Проверить имена полей связи можно в свойствах элемента **Подчиненная форма/отчет** (Subform/Subreport) на вкладке **Данные** (Data) в строках **Подчиненные поля** (Link Child Fields) и **Основные поля** (Link Master Fields) (рис. 7.37). Не путайте свойства подчиненного отчета и элемента **Подчиненная форма/отчет** (Subform/Subreport), содержащего подчиненный отчет.

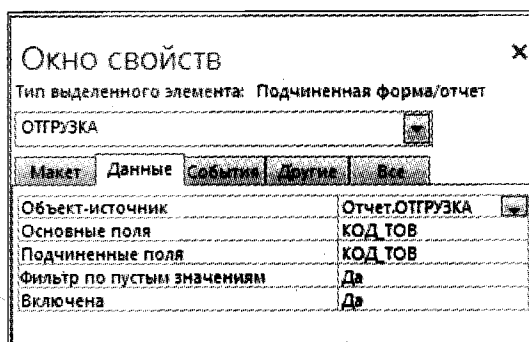


Рис. 7.37. Свойства элемента Подчиненная форма/отчет

Установленная связь отчетов ТОВАР и ОТПРУЗКА позволяет для каждой записи в отчете ТОВАР отображать только связанные записи в подчиненном отчете ОТПРУЗКА.

Для включения подчиненного отчета можно также воспользоваться услугами мастера подчиненных отчетов. Инициировать работу мастера можно на вкладке ленты **Конструктор** (Design) в коллекции **Элементы управления** кнопкой **Подчиненная форма/отчет** (Subform/Subreport). При этом должна быть выключена кнопка **Использовать мастера** (Use Control Wizards). Мастер создает в области данных основного отчета элемент **Свободный** (Unbound) с надписью **ВнедренныйN** (ChildN) и предлагает выбрать существующий подчиненный отчет или создать его на основе таблиц и запросов. Далее определяются поля связи основного и подчиненного отчетов, имя подчиненного отчета, и на этом работа мастера завершается. Свободный элемент превращается в элемент, называемый **Подчиненная форма/отчет**, и в его свойствах на вкладке **Данные** (Data) в строке **Объект-источник** (Source Object) указывается имя встроенного отчета, а в строках **Подчиненные поля** (Link Child Fields), **Основные поля** (Link Master Fields) указываются поля связи основного и подчиненного отчетов.

Доработка составного отчета

В режиме конструктора и макета в составном отчете доступны для редактирования как основной, так и подчиненный отчеты.

Над рамкой подчиненного отчета размещен элемент с его именем (ОТПРУЗКА). Удалим этот элемент.

Верхний колонтитул подчиненного отчета ОТГРУЗКА, в котором размещены подписи полей, не отображается при просмотре составного отчета. Перейдем в режим конструктора и для отображения заголовков столбцов табличной части отчета перенесем названия столбцов из верхнего колонтитула в заголовок группы ДАТА_ОТГР. Удалим содержимое заголовка подчиненного отчета.

После этого в подчиненном отчете удалим оба колонтитула, вызвав контекстное меню в любом месте внутри подчиненного отчета и выполнив команду **Колонтитулы страницы (Page Header/Footer)**.

Если размер подчиненного отчета не соответствует установленной ширине страницы или не позволяет отобразить часть данных, измените его, щелкнув на элементе управления подчиненного отчета и перетаскив маркеры размера в нужные позиции.

Используя инструменты на вкладках лент **Конструктор (Design)**, **Формат (Format)**, **Упорядочить (Arrange)** в режиме макета, а также обращаясь к свойствам элементов отчета, откорректируем подписи, изменим формат полей и подписей, проведем недостающие линии.

Отчет ТОВАР в режиме конструктора после внедрения подчиненного отчета ОТГРУЗКА и доработки показан на рис. 7.38.

ТОВАР

Заголовок отчета

ТОВАР

Верхний колонтитул

Область данных

Отгрузка товара	НАИМ_ТОВ	Код	КОД_ТОВ
Цена	ЦЕНА		
Единица измерения	ЕИ		
Ставка НДС	СТАВКА_НДС		
Наличие товара	<input checked="" type="checkbox"/>		

Заголовок отчета

Заголовок группы 'ДАТА_ОТГР'

-MonthName([Date of Shipment])

Дата отгрузки	Код накладной	Код склада	Код товара	Количество отгружено	Сумма отгружено
---------------	---------------	------------	------------	----------------------	-----------------

Область данных

ДАТА_ОТГР	НОМ_НАКЛ	КОД_СК	КОД_Т	КОЛ_ОТГР	СУММА_ОТГР
-----------	----------	--------	-------	----------	------------

Примечание группы 'ДАТА_ОТГР'

Итого за месяц

Примечание отчета

Общий итог по товару

Нижний колонтитул

Страница 1 & [Page] & из [Pages]

Рис. 7.38. Отчет ТОВАР со встроенным подчиненным отчетом ОТГРУЗКА (режим конструктора)

Добавление текущей даты и номера страницы

При автоматическом создании отчета с помощью инструмента **Отчет** или с помощью мастера отчетов в отчет всегда включается текущая дата и номер страницы. Эти элементы управления можно перенести в нужный раздел отчета или удалить вовсе. Для отображения текущей даты используется вычисляемое поле со встроенной функцией (Built-In Functions) `Date()` из категории **Дата и время** (Date/Time). Текущая дата, как любое другое вычисляемое поле, может быть создана с помощью кнопки **Поле** (Text Box) и записи в свободный элемент функции: `=Date()`. Для просмотра и выбора встроенных функций может быть использован построитель выражений, который вызывается в окне свойств на вкладке **Данные** (Data) в строке **Данные** (Control Source). В свойствах этого элемента на вкладке **Макет** (Format) в строке **Формат поля** (Format) можно выбрать формат отображения даты. Подпись этого поля можно удалить. Включите дату в область верхнего колонтитула главного отчета, и она будет выводиться на каждой странице отчета.

Для добавления номера страницы в область нижнего колонтитула мастер в свободном элементе сформировал выражение: `"Страница " & [Page] & " из " & [Pages]`. Это выражение может быть записано в свободный элемент отчета с помощью построителя выражений. В левой колонке нижней части окна построителя выберем элемент **Общие выражения** (Common Expressions). В средней колонке — тип элемента выражения **Страница N из M** (Page N of M). В правой колонке отобразится выражение, которое и следует вставить в вычисляемый элемент отчета.

Существуют и другие способы формирования поля даты и номера страницы. Добавить в раздел отчета поле текущей даты и времени можно, выполнив в режиме макета на вкладке ленты **Конструктор** (Design) команду **Дата и время** (Date & Time). Установка в диалоговом окне **Дата и время** (Date and Time) флажков **Формат даты** (Include Date) и/или **Формат времени** (Include Time) позволяет вставить текущую дату и/или текущее время и выбрать нужный вариант форматов (рис. 7.39).

В результате в отчете будет создано поле, в свойствах которого на вкладке **Данные** (Data) в строке **Данные** (Control Source) будет записано выражение: `=Date()`.

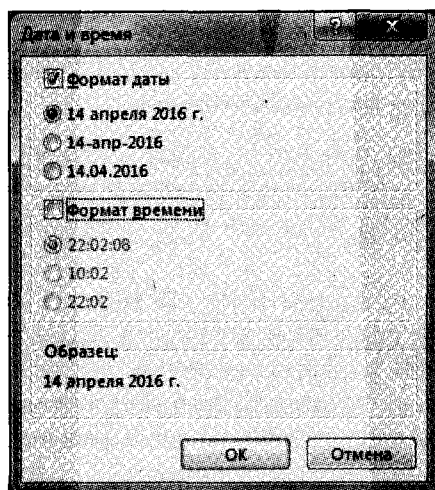


Рис. 7.39. Окно выбора формата даты и/или времени

Добавить в отчет поле нумерации страниц можно, выполнив команду **Номера страниц** (Insert Page Number). В диалоговом окне **Номера страниц** (Page Numbers) выбираются параметры, определяющие формат, расположение и выравнивание номеров страниц (рис. 7.40). Для печати номера страницы на первой странице устанавливается флажок **Отображать номер на первой странице** (Show Number on First Page).

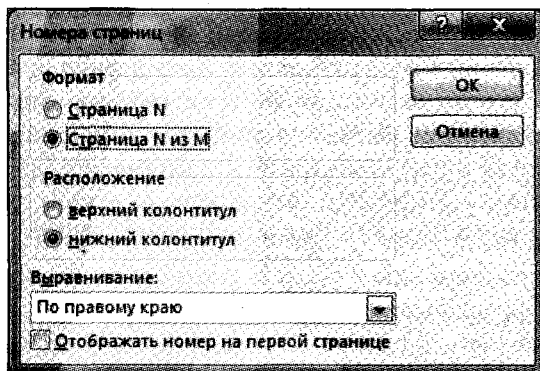


Рис. 7.40. Окно выбора параметров для номеров страниц

Просмотр отчета

Перейдем от конструирования к просмотру отчета, содержащего подчиненный отчет, щелкнув на кнопке **Предварительный просмотр** (Report View) в строке состояния. Страница составного отчета об отгрузке товаров в режиме предварительного просмотра представлена на рис. 7.41. В отчете записи об отгрузке товара сгруппированы по месяцам, причем учтено, что сведения об отгрузках могут сохраняться не один год.

Чтобы сведения об отгрузках каждого товара отображались с новой страницы, установите для области данных главного отчета на вкладке **Все** в свойстве **Конец страницы** (Force New Page) значение **До раздела** (Before Section).

Вывод отчета на печать можно осуществить нажатием кнопки **Печать** (Print) на вкладке ленты **Предварительный просмотр** (Report View).

В составном отчете реализуется режим, при котором запись главного отчета выводится и в том случае, когда для нее нет связанных записей в подчиненном отчете. В рассматриваемом примере отчет будет включать и те товары, для которых не было отгрузок.

Если аналогичный рассмотренному отчет построить, исходя из предварительно созданного запроса на основе трех таблиц, можно исключить вывод записей о товарах, по которым не было отгрузки или, наоборот, включить в отчет только их. Для этого достаточно в запросе определить нужный способ объединения записей таблиц **ТОВАР** и **ОТГРУЗКА**. Создание аналогичных запросов рассмотрено в *главе 4 «Запросы»*.


ТОВАР					
ОТГРУЗКА ТОВАРА		Монитор 17LG		14 апреля 2016 г.	
Цена	7 000,00р.		код T001		
Единица измерения	штука				
Ставка НДС	5%				
Наличие товара	<input checked="" type="checkbox"/>				
Дата отгрузки	Код накладной	Код склада	Количество отгружено	Сумма отгружено	
Январь 2016					
18.01.2016	H001	CO1	50	350 000,00р.	
Итого за месяц			50	350 000,00р.	
Февраль 2016					
10.02.2016	H005	CO1	40	280 000,00р.	
21.02.2016	H004	CO2	10	70 000,00р.	
11.02.2016	H002	CO2	30	210 000,00р.	
11.02.2016	H002	CO1	40	280 000,00р.	
Итого за месяц			120	840 000,00р.	
Март 2016					
25.03.2016	H004	CO1	2	14 000,00р.	
24.03.2016	H002	CO8	30	210 000,00р.	
Итого за месяц			32	224 000,00р.	
Общий итог по товару				1 414 000,00р.	

Рис. 7.41. Отчет об отгрузке товаров в режиме предварительного просмотра

Вывод значений нарастающим итогом

В отчетах можно вывести значения некоторого поля записи или итогового поля группировки нарастающим итогом. В рассматриваемом примере можно накапливать количество каждого отгруженного товара от месяца к месяцу. Так, если в первом месяце количество отгруженного равно 12, во втором — 4, в третьем — 7, используя свойство поля Сумма с накоплением (Running Sum) со значением Для всего (Over All), можно получить значения сумм с накоплением: для первого месяца — 12, для второго — 16, для третьего — 23. Это свойство размещено на вкладке Данные (Data).

Задание 7.3. Создайте отчет о заказанных покупателями товарах

В отчет включите сведения о покупателе и список товаров, заказанных им во всех договорах. В список включите наименование товара, сроки его поставки, количество заказанного и стоимость нарастающим итогом (от месяца к месяцу). Разра-

ботайте отчет на основе запроса на выборку. В запросе используйте таблицу **ПОСТАВКА_ПЛАН**, в которой содержатся сведения обо всех заказах товара и указан срок поставки, таблицу **ТОВАР**, а также таблицы **ДОГОВОР** и **ПОКУПАТЕЛЬ**. В отчете выполните группировку по покупателю и сроку поставки. Для накопления суммарных значений количества и стоимости используйте свойство **Сумма с накоплением (Running Sum)**. Выполните фильтрацию данных отчета по покупателю, товару, месяцу поставки.

Контрольные вопросы

1. Из каких разделов состоит отчет?
2. Назовите основные ленты, используемые при конструировании отчета?
3. Как просмотреть список доступных в конструкторе отчетов полей и вставить нужное поле в отчет?
4. Какое значение размещается в подписи поля при его перетаскивании в отчет?
5. Возможно ли выполнить в отчете группировку записей, не отображая заголовка и (или) примечания группы?
6. В каком порядке сортируются группы по умолчанию?
7. Чем отличается группировка в запросе от группировки, выполненной в отчете?
8. Где целесообразно размещать значение поля, по которому производится группировка?
9. Где должно размещаться вычисляемое поле с итоговым значением, рассчитанным для группы?
10. Какая функция позволяет включить в отчет дату?
11. Какой элемент, по какой кнопке создается для размещения в нем вычисляемого поля?
12. Какая вкладка ленты позволяет выбрать размер страницы отчета, ее полей?
13. Можно ли с помощью перетаскивания включить в отчет подчиненный отчет?
14. В каком режиме должен быть открыт отчет, чтобы в него можно было включить подчиненный?
15. Нужно ли устанавливать связь между главным и подчиненным отчетами, если связь соответствующих таблиц определена в схеме данных?
16. В свойствах какого элемента сохраняются поля связи отчетов?
17. Какое условие должно быть выполнено, чтобы при включении подчиненного отчета с помощью кнопки **Подчиненная форма/отчет (Subform/Subreport)** заработал мастер?
18. В какой строке свойств элемента **Подчиненная форма/отчет (Subform/Subreport)** указывается источник данных?
19. В какой строке свойств отчета указывается источник данных?

20. В каком месте отчета нужно щелкнуть мышью, чтобы сразу открыть его свойства?
21. Какой элемент отчета должен быть выделен, чтобы в свойствах просмотреть поля связи подчиненного отчета с главным?
22. На основе каких объектов строит отчет мастер?
23. Возможен ли выбор полей из различных таблиц и запросов при построении отчета мастером?
24. Сколько уровней группировки может определить мастер в отчете?
25. В каком случае мастер не выведет кнопку **Итоги** (Summary Options), которая позволяет указать, какие именно итоговые значения нужно вывести в отчете?
26. Какие функции позволяет использовать мастер для подведения итогов?
27. Что будет использовано в качестве источника записей при построении отчета мастером на основе нескольких взаимосвязанных таблиц?
28. Как просмотреть созданную мастером инструкцию SQL в конструкторе запросов?
29. Чем определяется состав полей, доступных в многотабличном отчете, построенном мастером?
30. Каким свойством надо воспользоваться для того, чтобы каждая группа печаталась с новой страницы?
31. Какое свойство поля нужно использовать, чтобы выводить его значения нарастающим итогом?
32. За счет чего повторяющиеся значения в таблице запроса могут быть отображены в отчете только один раз?
33. Будет ли при просмотре отчета выводиться диалоговое окно ввода параметра, если отчет создан на запросе с параметром?
34. В какой строке бланка запроса указывается наименование параметра?
35. Как просмотреть список полей, доступных в отчете?
36. Какие поля содержит список доступных в отчете полей, если отчет построен на запросе?
37. Какое значение получит вычисляемое поле, если в него записана встроенная функция управления: `IIf([СУММА_ПОСТ]>100000;'Особое внимание';''),` а в текущей записи поля сумма поставки имеет значение «70000»?

ОТВЕТЫ

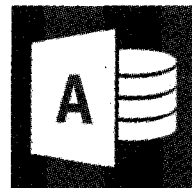
1. Заголовка отчета, верхнего колонтитула, заголовка группы, области данных, а также примечания группы, нижнего колонтитула и примечания отчета.
2. Группы **Инструменты конструктора отчетов** (Report Design Tools) и **Работа с макетом отчетов** (Report Layout Tools) включают одинаковые ленты: **Конст-**

руктор, Формат, Упорядочить, Параметры страницы (Design, Format, Arrange, Page Setup).

3. В режиме макета или конструктора отчетов нажать на ленте **Конструктор** в группе **Сервис (Tools)** кнопку **Добавить поля (Add Existing Fields)**. Перетащить из списка полей нужное в раздел отчета.
4. Подпись берется из свойства поля, заданного в таблице.
5. Нет. Только выбор одного из этих разделов приводит к выделению групп с одинаковыми значениями в выбранном поле.
6. По возрастанию.
7. В отчете записи группы не объединяются в одну запись. Один раз выводятся только поля, по которым осуществляется группировка.
8. В заголовке группы.
9. В примечании или заголовке группы.
10. Встроенная функция `Date()`.
11. Свободный элемент создается по кнопке **Поле (Text Box)**.
12. **Параметры страницы (Page Setup)**.
13. Да.
14. В режиме конструктора.
15. Нет. Устанавливается автоматически.
16. В свойствах элемента **Подчиненная форма/отчет (Subform/Subreport)** на вкладке **Данные (Data)**.
17. Должна быть включена кнопка **Использовать мастера (Use Control Wizards)**.
18. **Объект-источник (Source Object)**.
19. **Источник записей (Source Records)**.
20. На пересечении линеек.
21. **Подчиненная форма/отчет (Subform/Subreport)**.
22. Таблиц и запросов.
23. Да. Таблицы и запросы должны быть связаны непосредственно или через не включаемые в отчет объекты.
24. 4.
25. Если в отчете не предусмотрена группировка или не включено ни одно числовое поле.
26. `Sum, Avg, Min, Max`.
27. Инструкция SQL, запрос.
28. Щелчком мыши на кнопке **Построитель (Builder)** в строке **Источник записей (Record Source)** свойств отчета.

29. Перечисленными в инструкции **Select** полями.
30. Свойством заголовка группы **Конец страницы** (Force New Page) на вкладке **Макет** (Format).
31. Свойство **Сумма с накоплением** (Running Sum).
32. За счет группировки.
33. Да.
34. В строке **Условие отбора** (Criteria).
35. Открыв отчет в режиме макета или конструктора и выполнив на ленте **Конструктор** команду **Добавить поля** (Add Existing Fields).
36. Все поля запроса.
37. Пустая строка.

ГЛАВА 8



Оперативный анализ данных реляционной базы Access

Результаты решения различных задач на данных реляционной базы, как правило, представляются в двумерных ненормализованных таблицах с многократно повторяющимися данными. Просматривать и анализировать такие многомерные таблицы очень сложно.

Даже нормализованные таблицы содержат повторяющиеся данные, такие как ключи связи. Например, в рассматриваемом примере базы данных подчиненная таблица ОТГРУЗКА содержит ключи связи с таблицами ТОВАР и НАКЛАДНАЯ, которые, естественно, многократно повторяются. Объединение таблицы ОТГРУЗКА с таблицами ТОВАР и НАКЛАДНАЯ дает таблицу с более полными сведениями, необходимыми при решении задач, и с большим числом повторяющихся описательных данных.

Отчеты являются традиционным средством представления результатов решения задач и служат поставщиком данных для управления предприятием и контроля его деятельности. Отчеты, созданные в рамках оперативных баз данных, позволяют не только представлять данные в печатном виде, но и обеспечивают интерактивный анализ данных отчета, отображаемого на экране. При этом возможна фильтрация данных по любому из полей отчета, группировка и расчет итогов. В предыдущей главе подробно рассмотрены эти средства анализа данных. Однако необходимая при принятии решений многоаспектность и скорость анализа, требующая рассмотрения данных под разными углами зрения, не позволяют аналитику ограничиваться регламентированными отчетами. Невозможно предусмотреть и обеспечить аналитика всем многообразием отчетов, которые давали бы ответы на все возникающие у него вопросы.

Изменение структуры отчета может потребовать значительного времени и привлечения программиста. Даже изменение поля группировки требует навыков, не свойственных конечному пользователю. В результате аналитик не может оперативно проверить свои идеи, требующие выполнения многочисленных и все новых срезов данных. Кроме того, при увеличении числа отчетов вряд ли возможно организовать эффективный поиск нужных отчетов и навигацию по ним.

Интерактивные отчеты добавляют не так много гибкости к регламентным отчетам и не позволяют получить желаемого представления данных.

Информационные системы на основе баз данных также в первую очередь фокусируются на оперативной обработке транзакций (Online Transaction Processing, OLTP) в процессе учета материальных, финансовых, людских ресурсов и регистрации различных операций и событий. Однако автоматизация учета обеспечивает решение лишь оперативных, тактических, но не стратегических задач, а потому не удовлетворяет в полной мере потребностям бизнеса.

На оказание помощи руководителю в принятии своевременных решений ориентированы технологии оперативной аналитической обработки данных (On-Line Analytical Processing, OLAP). Эти технологии позволяют осуществлять стратегический обзор ситуации, в реальном времени получать ответы на вопросы, интересующие пользователя-аналитика, и представлять результаты этого анализа в удобном для восприятия и принятия решений виде. Средства OLAP предназначены как для быстрого получения ответов на произвольные запросы, так и для быстрого составления отчетности по агрегированным показателям процессов в различных разрезах и с произвольной глубиной детализации оперативных данных.

Оперативный анализ данных обеспечивает получение различных показателей деятельности предприятия, зависящих от большого количества параметров. В отличие от фиксированных форм отчетов, часто используемых менеджерами всех уровней, эти средства могут в оперативном режиме предоставлять все необходимые данные для принятия управленческих решений.

Оперативность при аналитической обработке данных, прежде всего, нужна для поддержания необходимого ритма работы человека в процессе анализа и принятия решений. Она реализуется за счет интерактивности, непосредственного доступа к данным, возможности самостоятельно, без посредников сформулировать запрос к данным для получения динамических отчетов.

Оперативность и гибкость создания динамических отчетов требуют, чтобы «на лету» можно было выбрать, задать или изменить:

- состав показателей, связанные с ними разрезы и параметры;
- вид таблицы — т. е. размещение показателей и параметров разрезов в строках, столбцах или заголовке отчета (фактов и измерений — в терминологии многомерных моделей данных);
- уровни детализации (агрегирования) разрезов и конкретных значений параметров;
- условия отбора значений фактов и измерений;
- формулы расчета;
- форматирование табличного представления;
- графическое оформление результатов.

Средства оперативного анализа данных должны обладать следующими основными преимуществами:

- гибкими механизмами детализации и агрегации данных по различным разрезам;
- высокой скоростью ответа на запрос;

- возможностью моделирования и быстрого создания аналитических отчетов, не прибегая к помощи программистов;
- вычислительной платформой, включающей статистические и другие сложные функции.

Для оперативного анализа данных в версиях, предшествующих Access 2013, содержались средства построения сводных таблиц и диаграмм. В них можно было отображать данные любой таблицы, запроса и даже источника данных формы. Из версий Access 2013/2016 эти средства исключены. В Microsoft Office сводные таблицы и диаграммы всегда входили в состав средств Excel, который позиционируется не только как инструмент для ведения и вычислений в электронных таблицах, но и как основное средство конечного пользователя для анализа данных. Аналитическая платформа Excel постоянно совершенствуется.

Интерактивный, быстрый и подробный анализ больших объемов данных обеспечивают сводные таблицы и диаграммы Excel. Сводные таблицы позволяют быстро получать ответы на нерегламентированные запросы пользователей-аналитиков, простыми средствами формировать отчетность по агрегированным показателям процессов в различных разрезах с произвольной глубиной детализации оперативных данных и представлением результатов анализа в виде, удобном для восприятия и принятия решений.

С помощью названных средств пользователи имеют возможность свободно анализировать данные, не ограничиваясь predetermined алгоритмами и не прибегая к помощи программистов. Простым перетаскиванием полей источника данных изменяется макет сводной таблицы и диаграммы, при этом автоматически выполняются новые группировки и вычисление итогов по выбранным параметрам.

Начиная с Excel 2013, в качестве источника данных сводных таблиц вместо одной таблицы могут использоваться несколько. Для определения взаимосвязей таблиц строится модель данных. Создание модели данных значительно расширило возможности оперативного анализа в Excel. При выборе в качестве источника таблиц реляционной базы данных модель может строиться автоматически на основе установленных в базе данных связей между таблицами. Таким образом, пользователь-аналитик освобождается от необходимости погружаться в структуру взаимосвязей таблиц данных и гарантирован от ошибок, т. к. структура базы данных подготавливается опытными ИТ-специалистами.

Модель данных можно использовать во всей книге Excel в любом количестве сводных таблиц и диаграмм. Для этой модели фундаментальными являются связи между таблицами, определяющие пути навигации и вычисления данных в отчете сводной таблицы.

При подключении к базе данных в одной сводной таблице можно, меняя макет, анализировать многие показатели, зависящие от разного набора параметров, можно вывести несколько показателей, зависящих от одинакового набора параметров. В то же время, если необходимо, можно для каждой задачи анализа создать свой отчет сводной таблицы.

Работа в популярном программном продукте обеспечивает специалистам конкретной предметной области комфортную среду и простоту обработки данных и позво-

ляет им опираться на свои навыки и накопленный опыт при интерпретации данных. При использовании хорошо знакомого и достаточно простого средства естественным образом расширяется круг участников процесса подготовки и принятия решений. В процесс вовлекаются не только руководство организации, бизнес-аналитики, но также менеджеры среднего звена и простые специалисты. Средствами доступа к необходимой информации вооружаются все категории лиц, принимающих решения.

Следует отметить, что пользователь, работая со сводными таблицами и диаграммами Excel, доверяет результатам анализа, т. к. понимает применяемые методы и умеет интерпретировать результаты в практической плоскости. В отличие от систем, в которых задействованы сложные модели, и далеко не очевидна взаимосвязь входа и выхода, работа со сводными таблицами не требует специальной подготовки пользователей. Кроме того, существуют объективные трудности при создании самих моделей.

Имея инструментарий, обладающий такими комбинаторными свойствами, пользователь может проводить интерактивное исследование данных «с чистого листа», создавая нужные динамические отчеты и представления данных, видоизменяя ранее подготовленные динамические отчеты. Он получает возможность менять структуру отчета, переставляя строки и столбцы таблицы, переключаться с табличного представления на графическое, убирать ненужные сведения и детализировать нужные, переходить с детальных данных на итоги. Таким образом, в ходе анализа он переберет много частных отчетов, но зафиксирует (сохранит или напечатает) только приемлемые варианты или решения.

Новые возможности средств анализа данных в Excel позволяют разгрузить оперативные системы и перенести аналитическую платформу, включая источник данных, в отдельную специализированную среду.

Оперативный анализ данных базы Access в Excel

Для интерактивного, быстрого и подробного анализа больших объемов данных базы Access целесообразно использовать отчеты сводных таблиц и сводных диаграмм Excel. Отчеты сводных таблиц позволяют конечному пользователю оперативно получать ответы на произвольные вопросы по данным. Они предоставляют методику оперативного извлечения нужной информации из больших массивов данных и формирования соответствующих отчетов. Отчеты сводных таблиц для представления агрегатных данных широко применяются на практике.

Отчеты сводных таблиц и диаграмм Excel

Отчет сводной таблицы Excel (Pivot Table) является интерактивным перекрестным отчетом, содержащим итоговые данные и выполняющим анализ данных, расположенных в книге или импортированных из внешних источников данных.

Отчет сводной таблицы используется для суммирования, статистического анализа при исследовании большого количества исходных сводных данных, содержащих повторяющиеся значения, и представления их в удобном для понимания виде. Строки и столбцы подобной интерактивной таблицы базируются на данных из столбцов исходной таблицы. Ячейки области значений сводной таблицы представляют собой вычисленные значения на основе одного из столбцов исходной таблицы. По умолчанию для вычислений по столбцам с числовыми значениями используется функция СУММ, с текстовыми — функция СЧЁТ. При этом вычисление производится для тех строк исходной таблицы, в которых значения столбцов совпадают с именами строки и столбца сводной таблицы, на пересечении которых находится нужная ячейка, т. е. выполняется группировка строк исходной таблицы по значениям полей, используемых в строках и столбцах сводной таблицы, и вычисление итогов по полю, которое выводится в области значений.

Отчет сводной диаграммы Excel используется для визуализации данных и упрощения процедуры сравнений, поиска закономерностей и тенденций. Диаграмма, как и отчет сводной таблицы, выполняя интерактивный анализ данных, позволяет пользователю изменять представление данных, выполнять фильтрацию данных по различным параметрам, просматривать данные с различными уровнями детализации, выполнять группировку и подведение итогов по различным полям, перестраивая макет диаграммы, перетаскивая поля и показывая или скрывая элементы полей.

Создание отчета сводной таблицы начинается с выбора источника данных. В общем случае в качестве источника данных сводных таблиц Excel могут быть использованы списки или диапазоны данных Excel, таблицы или запросы внешних реляционных баз данных и OLAP-кубы служб аналитики, сохраняемые в многомерной базе данных аналитических систем, — например, SQL Server, Oracle и др.

Выбор в качестве основного источника для оперативного анализа данных реляционных баз обосновывается наличием их в подавляющем большинстве предприятий и накоплением в них основных объемов данных со сведениями о функционировании предприятия. Без оперативных учетных систем не было бы детальных первичных данных — нечего было бы анализировать.

В Microsoft Office включены средства (драйверы), которые позволяют извлекать данные из различных источников, в том числе из Microsoft Office Access. Для извлечения данных из источников, для которых в Microsoft Office соответствующие средства не предоставлены, можно использовать средства других производителей.

Выбрав источник, пользователь может приступить к созданию макета сводной таблицы, определяя, какие из полей таблицы источника данных будут участвовать в формировании строк, столбцов и страниц (фильтра) и какие из них будут использованы для вычисления суммарных значений. При этом в случае применения в качестве суммируемых нечисловых полей можно вычислить количество соответствующих записей в исходной таблице. Осуществить разработку макета можно путем перетаскивания мышью наименований полей из списка полей исходной таблицы в соответствующие области макета сводной таблицы.

На рис. 8.1 приведен пример отчета сводной таблицы для анализа стоимости заказанных товаров по различным покупателям и договорам по всем или некоторым месяцам года.

1	А	В	С	Д	Е	Ф	Г
2	Месяц отгрузки заказанных товаров	4					
3	Сумма	Покупатель/Договор					
4		Компьютер маркет		Компьютер маркет Итого	Перспектива	Перспектива Итого	Общий итог
5	Наименование товара	Д111	Д222		Д333		
6	FDD 3,5		3 600р.	3 600р.			3 600р.
7	Зв. Карта Genius Liv		7 890р.	7 890р.			7 890р.
8	Монитор 17LG	2 000р.		2 000р.	2 000р.	2 000р.	4 000р.
9	Принтер EPSON ST.A4	24 000р.	24 000р.	48 000р.			48 000р.
10	СканерAcer				28 056р.	28 056р.	28 056р.
11	Общий итог	26 000р.	35 490р.	61 490р.	28 056р.	28 056р.	91 546р.

Рис. 8.1. Сводная таблица для анализа суммарной стоимости любого из заказанных товаров

Приведенный отчет сводной таблицы позволяет проанализировать связанные итоги. С его помощью можно легко сравнить сумму заказа одного товара за выбранный период с суммой по другому товару, либо вычислить общий итог заказов, либо сравнить суммы заказов по различным покупателям и договорам.

Отчет построен на основе запроса базы данных Access «Поставка товаров», рассматриваемой в книге. Запрос дополняет записи таблицы ПОСТАВКА ПЛАН наименованиями товаров и покупателей из таблиц ТОВАР и ПОКУПАТЕЛЬ.

Список полей запроса базы данных — источника данных сводной таблицы представлен на рис. 8.2.

Рис. 8.2. Список полей (столбцов) исходной таблицы, доступных для построения макета сводной таблицы

В отчете сводной таблицы выбранный столбец (поле) исходной таблицы становится столбцом или строкой сводной таблицы, по их значениям производится группировка записей таблицы. Для образовавшихся групп итоги рассчитываются по полю, выбранному для области значений, и размещаются в области значений на пересечении строки и столбца.

В примере область значений содержит суммарные значения поля стоимости заказов. Каждая ячейка содержит сумму значений по столбцу Сумма для строк исходных данных с одинаковыми значениями в полях `НАИМ_ТОВАР` (наименование товара), `НОМ_ДОГ` (номер договора) и `НАИМ_ПОК` (наименование покупателя). Причем для суммирования выбираются только строки заказов, попадающие в указанный период. По умолчанию для получения данных в области значений сводной таблицы исходные данные рассчитываются следующим образом: для числовых значений используется функция `СУММ`, а для текстовых — `СЧЕТ`.

В созданном отчете сводной таблицы простым выбором значений в поле строк (**Наименование товара** ↓), поле столбца (**Покупатель/Договор** ↓) и поле страницы (фильтра) (**Месяц отгрузки заказанных товаров** ↓) осуществляется фильтрация данных исходной таблицы и перерасчет итогов.

Для всестороннего анализа данных возможно динамическое изменение макета. Существует возможность изменять заголовки строк, столбцов, а также полей, определяющих страницу. Создавать и быстро модифицировать макет можно, выбирая и перетаскивая поля из списка полей источника в нужные области. При каждом изменении макета сводная таблица немедленно выполняет вычисления заново в соответствии с новым расположением данных.

Сводные диаграммы служат для наглядного графического представления анализируемых данных, облегчая для пользователей сравнение и выявление тенденций и закономерностей в них (рис. 8.3).

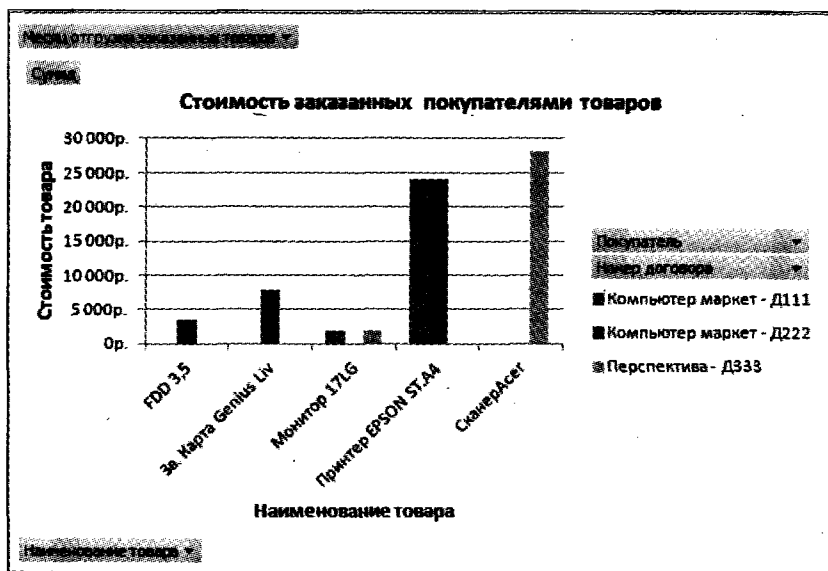


Рис. 8.3. Гистограмма, соответствующая отчету сводной таблицы

Вставка сводной диаграммы потребует от вас только выбора соответствующего типа диаграммы — например, гистограммы. В новой версии Excel в сводной диаграмме, также как в отчете сводной таблицы, для получения нужных показателей можно осуществлять выбор значений для полей строк, столбцов и фильтра.

Содержимое сводной таблицы и диаграммы синхронизировано. Выполненные в сводной таблице изменения сразу отображаются в диаграмме и наоборот.

В общем случае отчет сводной таблицы позволяет:

- выполнять запросы к большим массивам данных дружественными по отношению к пользователю способами;
- подводить промежуточные итоги и применять статистические функции к данным, суммировать данные по категориям и подкатегориям, а также создавать для них дополнительные вычисления и формулы;
- разворачивать и свертывать уровни представления данных для привлечения внимания к результатам, а также разворачивать отчеты с целью получения подробных сведений из итоговых данных по нужным областям;
- конвертировать строки в столбцы или столбцы в строки («вращение») для просмотра различных сводных данных по исходным данным;
- фильтровать, сортировать, группировать наиболее важные и часто используемые подмножества данных для привлечения внимания к нужным сведениям;
- представлять краткие, наглядные и аннотированные отчеты в сети или в напечатанном виде.

Определение базы данных Access в качестве внешнего источника данных для отчета сводной таблицы

Рассмотрим возможности использования реляционной базы данных Microsoft Office Access в качестве внешнего источника данных сводной таблицы Excel. Существуют два варианта работы с таким источником: первый основывается на использовании одной таблицы (запроса, представления) базы данных, второй — нескольких.

В первом случае, единственно доступном в версиях до Excel 2013, для представления в списке полей сводной таблицы всех необходимых для анализа показателей (фактов), параметров, влияющих на них (измерений), иерархий измерений, обеспечивающих детализацию и агрегацию фактов, требуется предварительная подготовка в базе данных таблицы или запроса необходимой структуры. В запросе собираются данные из нескольких нормализованных таблиц, создавая многомерную структуру данных, ориентированную на выполнение задач анализа.

Во втором случае, реализованном в последних версиях Excel, появилась возможность при определении источника выбирать несколько таблиц из различных источников, определять связи между ними и на их основе создавать табличную модель данных, по-прежнему предоставляя пользователю возможность путем простого пе-

ретаскивания полей строить сводную таблицу. При использовании в качестве источника данных реляционной базы, вместе с таблицами в Excel импортируется реализованная в базе модель данных, освобождая конечного пользователя от необходимости создавать модель вручную и гарантируя ее правильность. Следует заметить, что такое автоматическое построение модели накладывает некоторые ограничения на структуру базы данных.

Как в первом, так и во втором случае, исходные данные базы импортируются в книгу Excel. В дальнейшем они могут обновляться в соответствии с требуемым регламентом.

Создание отчета сводной таблицы Excel, связанного с внешним источником данных, начинается с определения этого источника. На ленте Excel **Данные (Data)** в группе **Получение внешних данных (Get External Data)** выберите тип источника данных — для работы с базой данных Access **Из Access (From Access)**. В открывшемся окне **Выбор источника данных (Select Data Source)** определите конкретное местоположение источника и его имя. Для получения доступа к данным выбранного источника выполните команду **Открыть (Open)**. Если в качестве источника была выбрана некоторая база данных, откроется окно **Выбор таблицы (Select Table)**, где представлены все таблицы и запросы базы данных. Выделив нужную таблицу, можно начать импорт данных источника. Флажок **Разрешить выбор нескольких таблиц (Enable selection of multiple tables)** позволяет импортировать любое число таблиц. Далее выбирается один из трех способов представления данных в книге: **Таблица (Table)**, **Отчет сводной таблицы (PivotTable Report)** или **Сводная диаграмма (PivotChart)**. Возможно только создать подключение с целью его дальнейшего использования в различных книгах.

Выберите **Отчет сводной таблицы** и приступайте к его построению, воспользовавшись списком полей, полученным из источника данных. Отчет сводной таблицы состоит из 4 областей: **СТРОКИ (Rows)**, **СТОЛБЦЫ (Columns)**, **Σ ЗНАЧЕНИЯ (Values)** и **ФИЛЬТРЫ (Filters)**. Перетаскиванием в эти области полей источника данных строится макет отчета сводной таблицы.

В результате определения внешнего источника данных для отчета сводной таблицы в книге создается подключение, кроме того, создается файл подключения, доступный для использования в других книгах. Файл подключения имеет расширение **odc** (от **Microsoft Office Data Connection**). Использование файлов подключения удобно при совместном использовании соединения и для более простого администрирования источников данных. После подключения к внешним данным книги Excel могут автоматически обновляться из исходного источника данных при его изменении.

Файлы подключения могут храниться в сети или — по умолчанию — на локальном компьютере в папке **Мои документы | Мои источники данных**. Сведения о файле подключения и источнике данных можно посмотреть, выполнив команду **Подключения (Connections)** из группы **Подключения (Connections)** на ленте **Данные (Data)**. В открывшемся окне **Подключения к книге (Workbook Connections)** выберите имя подключения и нажмите кнопку **Свойства (Properties)**. Размещение и имя источника данных указано в окне **Свойства подключения (Connection Properties)**

на вкладке **Определения (Definition)** — в поле **Строка подключения (Connection string)**, например: `DataSource=E:\Access 16\ Поставка товаров.accdb`. Там же можно посмотреть имя провайдера.

Сведения об источнике данных, сохраненные в файле подключения, позволяют выполнить обновление анализируемых данных путем импорта измененных в источнике данных. Для этого достаточно выполнить одну из команд, отображаемых по кнопке **Обновить все (Refresh All)** на ленте **Данные** в группе **Подключения**. Возможность подключения к внешним данным на конкретном компьютере может быть отключена. Для обновления данных при открытии книги необходимо разрешить подключение данных, используя **Параметры центра управления безопасностью**, либо помещая книгу в список надежных расположений.

Если на локальном компьютере или в сети существуют ранее использовавшиеся подключения к источникам данных, создание отчета сводной таблицы можно начать с команды **Сводная таблица (PivotTable)** на ленте **Вставка (Insert)** в группе **Таблица (Tables)**. По этой команде открывается окно выбора данных для анализа, где можно указать диапазон данных Excel или выбрать подключение к внешнему источнику данных. В открывшемся окне **Существующие подключения (Existing Connections)** выбирается нужный файл подключения. Далее можно указать, куда следует поместить отчет сводной таблицы. В указанном месте книги откроется список полей источника данных, доступных для построения отчета сводной таблицы.

Для соединения с рядом внешних, предварительно указанных, источников данных может использоваться мастер подключения данных. Чтобы открыть мастер подключения данных, нажмите кнопку **Из других источников (From Other Sources)** на вкладке **Данные (Data)** в группе **Получить внешние данные (Get External Data)**, а затем выберите пункт **Из мастера подключения данных**.

Отчет сводной таблицы и диаграммы на основе запросов базы данных

Создание отчета сводной таблицы для анализа отгрузок товара

Пусть необходимо провести всесторонний анализ деятельности предприятия в части отгрузки товаров покупателям по договорам в различные периоды. Создайте сводную таблицу и диаграмму, связанную с реляционной базой данных Access «Поставка товаров», в которой накоплены оперативные данные по отгрузкам товаров. В качестве внешнего источника данных сводной таблицы используйте запрос.

На рис. 8.4 приведена подсхема базы данных «Поставка товаров», разработанная для хранения накладных — документов по отгрузке товаров. Она имеет вид «снежинки». В центре «снежинки» лежит таблица фактов, с которой связаны таблицы измерений. Измерения в таблице фактов представлены кодами. Данные для каждого измерения берутся из таблицы-справочника. Таблица **ОТГРУЗКА** содержит фак-

ты о количестве отгруженного товара по накладной (КОЛ_ОТГР) и о стоимости этого товара (СУММА_ОТГР), остальные таблицы содержат значения измерений. Схема «снежинка» определяет возможность работы с иерархическими уровнями, определяющими степень детализации данных.

Откройте в Access базу данных «Поставка товаров». Создайте многотабличный запрос на выборку полей, необходимых для выполнения анализа (рис. 8.5). Для об-

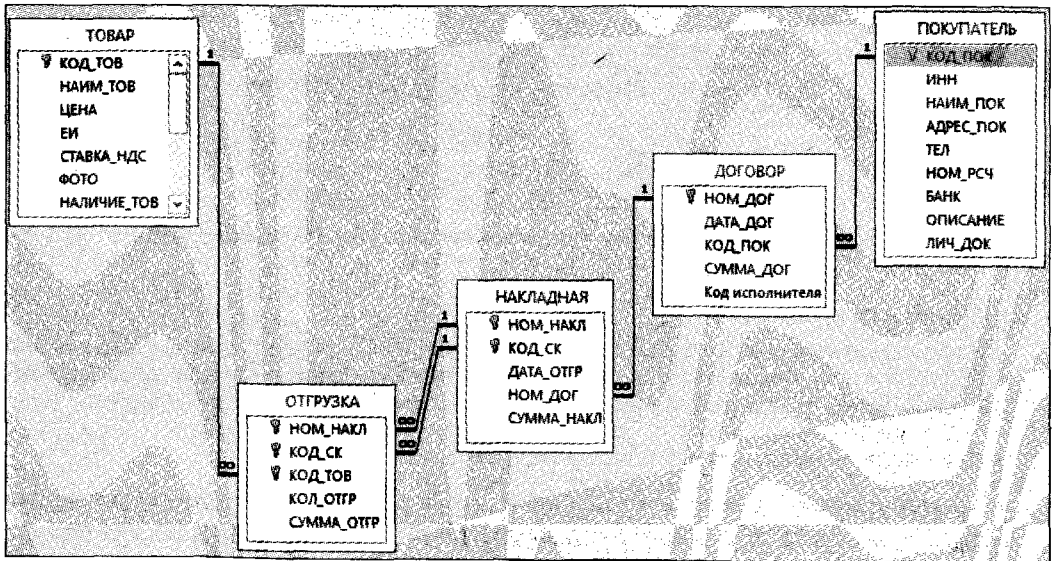


Рис. 8.4. Схема базы данных «Поставка товаров»

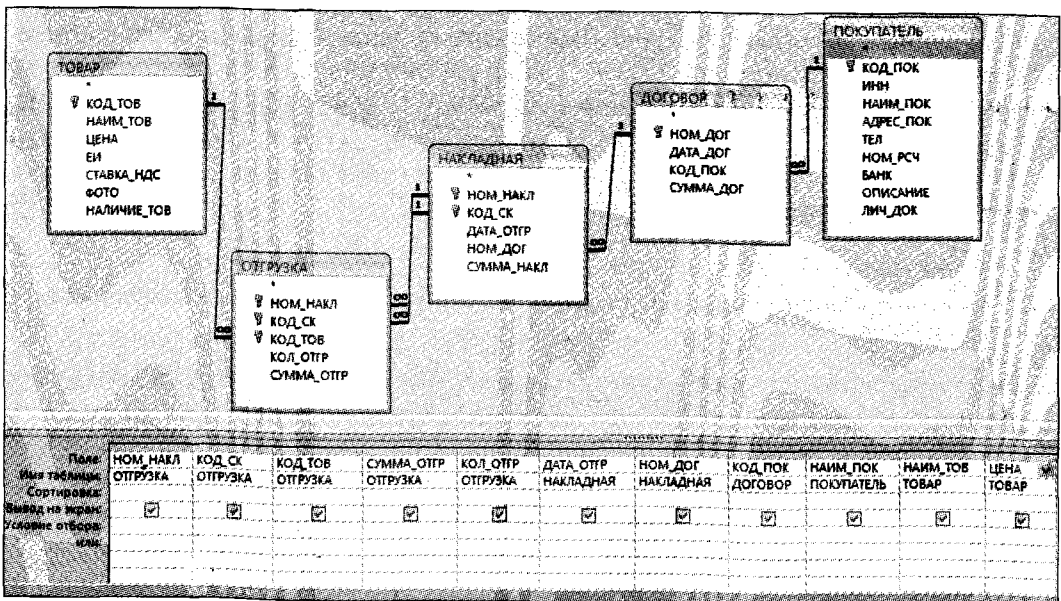


Рис. 8.5. Запрос к базе данных «Поставка товаров» в режиме конструктора

разования уровней иерархии, по дате отгрузки дополните запрос вычисляемыми полями Месяц и Год (рис. 8.6).

В запросе, показанном на рис. 8.5, использованы все таблицы подсхемы и выбраны поля, которые представляют интерес при анализе отгрузок по договорам.

Сохраните запрос под именем *Массив* для анализа. Результатом выполнения запроса будет многомерный массив, представленный в виде ненормализованной таблицы с многочисленными повторяющимися значениями в полях.

Месяц: Month([ДАТА_ОТПР])	Год: Year([ДАТА_ОТПР])
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 8.6. Вычисляемые поля для определения месяца и года по указанной дате

ОБРАТИТЕ ВНИМАНИЕ

В запросе выполняется объединение данных различных ранее нормализованных таблиц, при этом обеспечивается выбор только нужных полей, создание новых вычисляемых полей, отбор только нужных записей, например, за последний квартал.

Откройте Excel. Создайте отчет сводной таблицы.

1. Определите для открытой пустой книги Excel в качестве внешнего источника данных для отчета сводной таблицы базу данных Access «Поставка товаров». Для этого на ленте **Данные (Data)** в группе **Получение внешних данных (Get External Data)** выберите тип источника данных из **Access** и в открывшемся окне **Выбор источника данных (Select Data Source)** — конкретное местоположение базы данных «Поставка товаров». Выполните команду **Открыть (Open)**. Откроется окно **Выбор таблицы (Select Table)**, где представлены все таблицы и запросы базы данных. Выделите запрос *Массив* для анализа и нажмите кнопку **ОК**. Выберите способ представления данных в книге **Отчет сводной таблицы (PivotTable Report)** и место для размещения сводной таблицы. Нажмите кнопку **ОК**, чтобы начать импорт данных источника.

ЗАМЕЧАНИЕ

Если окно **Выбор таблицы** не открывается, вероятно подключаемая база данных открыта. Закройте ее.

2. После завершения процедуры импорта отобразится лента **Работа со сводными таблицами | Анализ (PivotTable Tools | Analyze)**, на которой можно начать создание отчета сводной таблицы. Перетащите поля исходной таблицы из списка **Поля сводной таблицы (PivotTable Fields)** (рис. 8.7) в области сводной таблицы: **ФИЛЬТРЫ**, **Σ ЗНАЧЕНИЯ**, **СТРОКИ** и **СТОЛБЦЫ**. Измените имена областей сводной таблицы, как показано на рис. 8.8.
3. Создайте сводную диаграмму на основе данных сводной таблицы. Для этого выполните команду **Сводная диаграмма (PivotChart)** на ленте **Работа со сводными**

Поля сводной таблицы [X] [v]

АКТИВНАЯ | **ВСЕ**

Выберите поля для добавления в отчет: [шестеренка]

Поиск [лупа]

- НОМ_НАКЛ
- КОД_СК
- КОД_ТОВ
- СУММА_ОТГР
- КОЛ_ОТГР
- Дата отгрузки
- НОМ_ДОГ
- КОД_ПОК
- НАИМ_ПОК
- НАИМ_ТОВ
- ЦЕНА
- Месяц
- Год

Перетащите поля в нужную область:

ФИЛЬТРЫ	СТОЛБЦЫ
Год [v]	НАИМ_ПОК [v]
Месяц [v]	НОМ_ДОГ [v]
Дата отгрузки [v]	
СТРОКИ	ЗНАЧЕНИЯ
НАИМ_ТОВ [v]	Суммарное количество отгруженного [v]

Отложить обновление макета [ОБНОВИТЬ]

Рис. 8.7. Заполнение областей сводной таблицы полями из запроса базы данных

	A	B	C	D	E	F	G
1	Год	2012	[v]				
2	Месяц	4	[v]				
3	Дата отгрузки	(Все)	[v]				
4							
5	Суммарное количество отгруженного	Наименование покупателя [v]					
6		Компьютер маркет		Компьютер маркет	Перспектива		
7	Наименование товара [v]	Итого	Итого	Итого	Итого	Общий итог	
8	FDD 3,5	Д111	Д222	Д333			
9	Зв. Карта Genius Liv		10	10			10
10	Монитор 17LG		10	10			10
11	Принтер EPSON ST.A4		2	2	2	2	4
12	СканерAcer	10	10	20			20
					12	12	12

Рис. 8.8. Отчет сводной таблицы, связанный с запросом

ми таблицами | Анализ (PivotTable Tools | Analyze) или воспользуйтесь командами ленты Вставка (Insert). Выбрав нужный вид диаграммы в группе Диаграммы (PivotTable), разместите ее на отдельном листе. Сводная диаграмма, соответствующая выбранным в сводной таблице значениям параметров, отобразится, как показано на рис. 8.9.

- Используя кнопку в конце наименования столбца **Наименование покупателя**, выбирайте поля **НАИМ_ПОК** или **НОМ_ДОГ** для сортировки и фильтрации значений в отчете сводной таблицы и диаграммы.
- На новом листе повторно создайте отчет сводной таблицы, выбрав тот же источник данных (или скопируйте ранее созданный отчет). Перестройте отчет в соответствии с рис. 8.10. Для сравнения данных по годам и месяцам переимес-

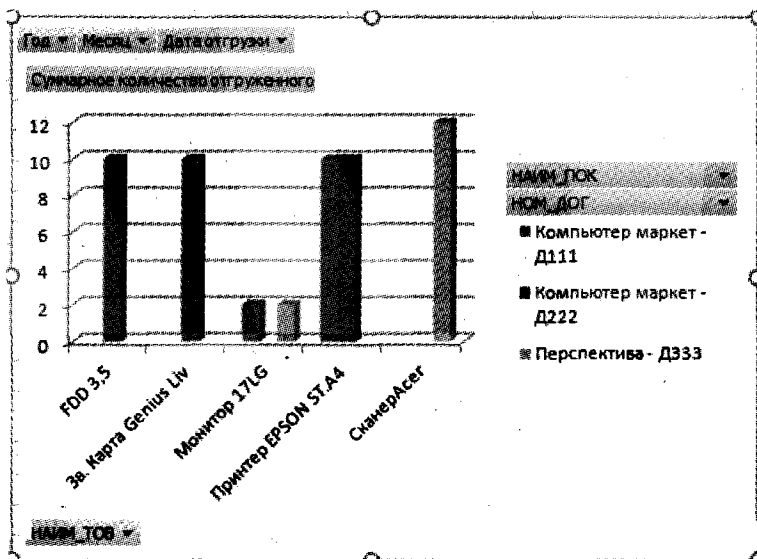


Рис. 8.9. Сводная диаграмма, соответствующая сводной таблице на рис. 8.8

	A	B	C	D	E
1					
2					
3	НАИМ_ТОВ	Монитор 17LG			
4					
5	Сумма отгрузки	Наименование покупателя			
6		Компьютер маркет	Перспектива	Перспектива	Итого
7	Дата		Д333		Общий итог
8	2011	20 000р.			20 000р.
9	2012	7 000р.	4 000р.	4 000р.	11 000р.
10	2	5 000р.			5 000р.
11	3		2 000р.	2 000р.	2 000р.
12	4	2 000р.	2 000р.	2 000р.	4 000р.
13	Общий итог	27 000р.	4 000р.	4 000р.	31 000р.

Рис. 8.10. Сводная таблица для сравнительного анализа данных по годам и месяцам

тите их в область строк или столбцов. Наименование товара поместите в фильтр, поле СУММА_ОТГР — в область значений.

Создание отчета сводной таблицы для оценки долга предприятия по договорам

Создайте сводную таблицу и диаграмму для анализа стоимости долга предприятия (недопоставок) в разрезе покупателей, договоров по всем или некоторым месяцам года с вычислением итогов. В качестве источника данных используйте запрос базы данных «Поставка товаров».

Для получения сведений о долгах предприятия необходимо подсчитать:

1. Суммарную стоимость товаров, указанных в договорах, нарастающим итогом.
2. Суммарную стоимость товаров, отгруженных в соответствии с договорами, нарастающим итогом.
3. Получить разность между первой и второй суммами.

На рис. 8.11 представлен запрос Стоимость договоров, на рис. 8.12 — запрос Стоимость отгрузки по договорам, а на рис. 8.13 — запрос Недоплата для подсчета месячного долга предприятия покупателям в стоимостном выражении нарастающим итогом. Вычисляемые поля запроса Недоплата для расчета долга по договорам пред-

Стоимость договоров

ТОВАР

- КОД_ТОВ
- НАИМ_ТОВ
- ЦЕНА
- ЕИ
- СТАВКА_НДС
- ФОТО
- НАЛИЧИЕ_ТОВ

ДОГОВОР

- НОМ_ДОГ
- ДАТА_ДОГ
- КОД_ПОК
- СУММА_ДОГ

ПОКУПАТЕЛЬ

- КОД_ПОК
- ИНН
- НАИМ_ПОК
- АДРЕС_ПОК
- ТЕЛ
- НОМ_РСЧ
- БАНК
- ОПИСАНИЕ
- WEB_АДРЕС

ПОСТАВКА_ПЛАН

- НОМ_ДОГ
- КОД_ТОВ
- СРОК_ПОСТ
- МИН_ПОСТ
- КОЛ_ПОСТ
- СУММА_ПОСТ

Календарь

- Номер месяца
- Название месяца

Поле:	НОМ_ДОГ	НАИМ_ПОК	Номер месяца	СРОК_ПОСТ	SummaПодогаворы: Sum([КОЛ_ПОСТ]*[ЦЕНА])
Имя таблицы:	ПОСТАВКА_ПЛАН	ПОКУПАТЕЛЬ	Календарь	ПОСТАВКА_ПЛАН	Выражение
Групповая операция:	Группировка	Группировка	Группировка	Условие	
Сортировка:					
Выход на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				<= [Номер месяца]	

Рис. 8.11. Запрос Стоимость договоров к базе данных «Поставка товаров»

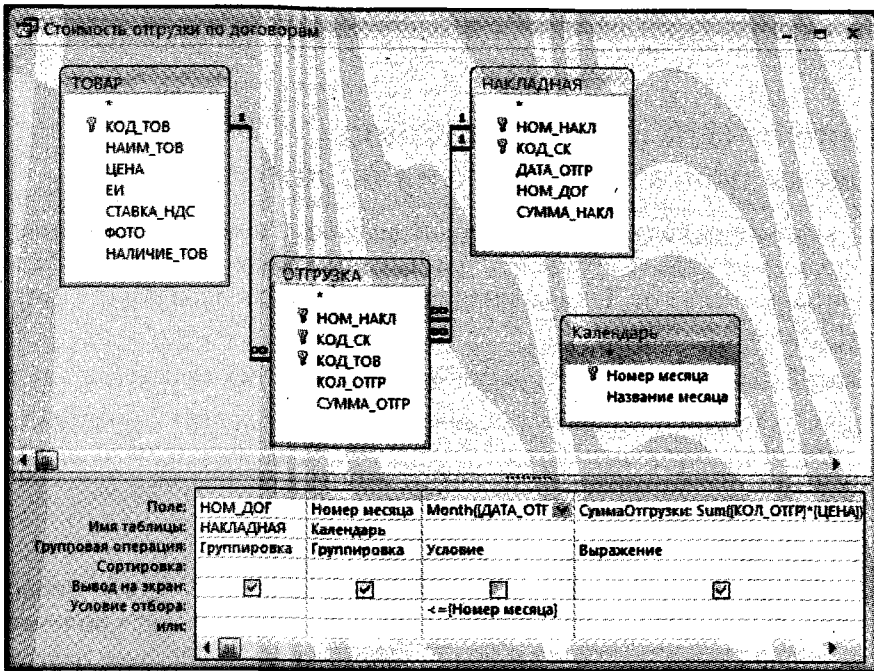


Рис. 8.12. Запрос Стоимость отгрузки по договорам к базе данных «Поставка товаров»

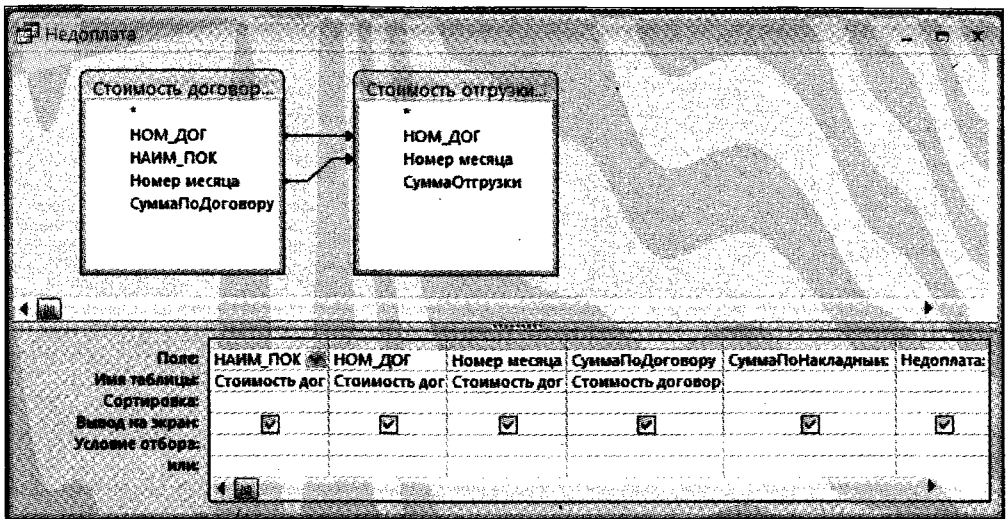


Рис. 8.13. Запрос вычисления долга по договорам в стоимостном выражении

ставлены на рис. 8.14. Результатом выполнения запросов будет многомерный массив с именем Недоплата, представленный в виде многомерной ненормализованной таблицы.

На рис. 8.15 и 8.16 показан один из вариантов отчета сводной таблицы и сводной диаграммы, которые построены на основе запроса Недоплата базы данных «Поставка товаров».

На основе последнего запроса (см. рис. 8.13 и 8.14) предложите другие варианты анализа данных и создайте отчеты сводной таблицы и соответствующие им диаграммы.

Выполните изменения в базе данных и, обновив отчет сводной таблицы, убедитесь, что работаете с новыми данными. Сохраните обновленный массив данных для последующей работы в Excel.

Задание 8.1. Создание запроса и отчета сводной таблицы

В базе данных «Поставка товаров» создайте запрос для анализа заказов покупателей в разрезе товаров и месяцев. Обеспечьте возможность отображения в сводной таблице товаров, не заказанных ни одним из покупателей, и покупателей, не сделавших заказов.

Отчет сводной таблицы и диаграммы на основе нескольких таблиц базы данных

В новых версиях Microsoft Office Excel 2013/2016 и Microsoft Office Excel 365 в качестве источника данных сводной таблицы можно использовать несколько взаимосвязанных таблиц. При импорте таблиц автоматически создается модель данных, включающая все таблицы. Связи между таблицами модели могут создаваться уже при импорте или позже вручную. Модель данных, в которой определены связи таблиц, позволяет создавать сводные таблицы, использующие поля из любой таблицы, даже если таблицы происходят из различных источников.

Для анализа деятельности предприятия по заказам и отгрузкам товаров покупателям создайте отчет сводной таблицы Excel, связанный с базой данных Access «Поставка товаров». При определении внешнего источника данных выберите эту базу данных. В окне **Выбор таблицы** (Select Table) установите флажок **Разрешить выбор нескольких таблиц** (Enable selection of multiple tables) и выберите все таблицы базы данных (рис. 8.17).

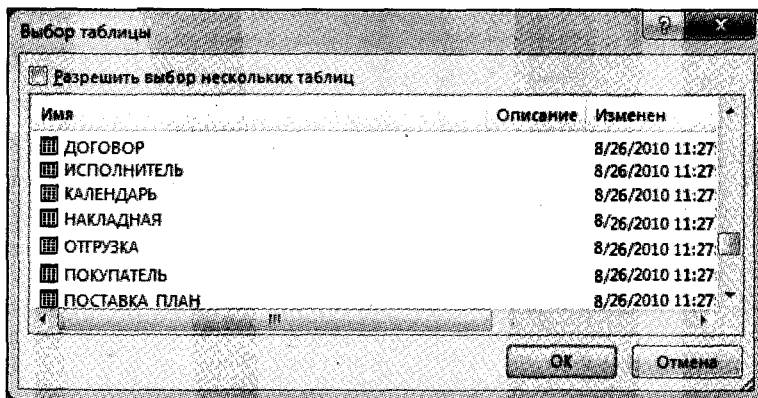


Рис. 8.17. Окно выбора таблиц базы данных для выполнения оперативного анализа

Далее в окне **Импорт данных (Select Table)** выберите **Отчет сводной таблицы (PivotTable Report)** и лист для размещения данных. Выбранные таблицы будут импортированы в книгу Excel.

Выбор из базы данных нескольких таблиц обеспечивает формирование списка полей сводной таблицы, в котором наглядно отображены все таблицы. Одни таблицы представляют факты, другие — измерения. На рис. 8.18 показан пример списка полей таблиц фактов и измерений для анализа заказов и отгрузок товаров по договорам покупателей.

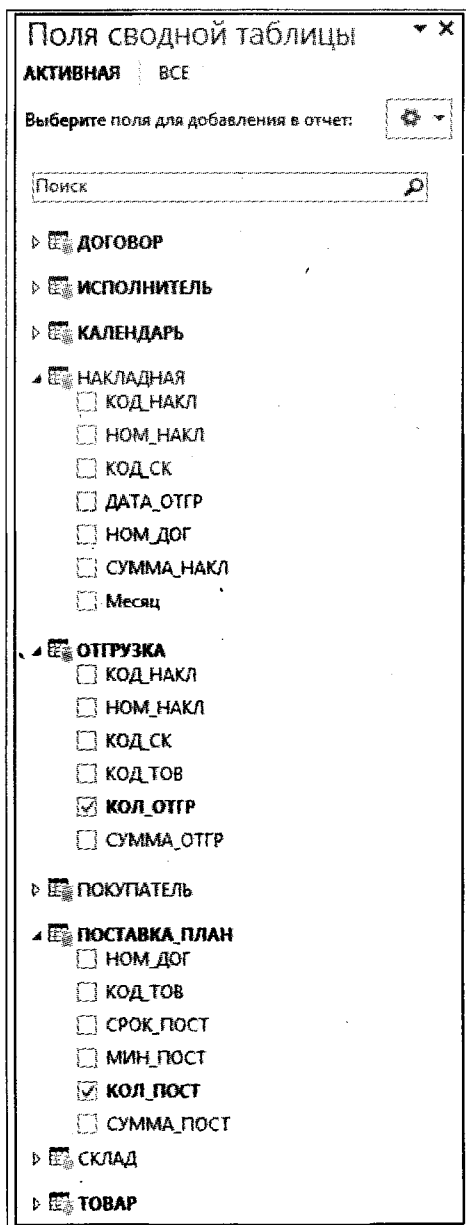


Рис. 8.18. Список таблиц базы данных

Если в базе определены связи таблиц, они вместе с таблицами импортируются в книгу Excel, где на их основе создается модель данных источника сводной таблицы. Если связи не определены в базе данных, их можно создать в модели данных самостоятельно. Модель данных в Excel представляется списком связей между парами таблиц в окне **Управление связями** (Manage Relationships) (рис. 8.19). Это окно вызывается командой **Отношения** (Relationships) на вкладке **Данные** (Data) в группе **Работа с данными** (Data Tools).

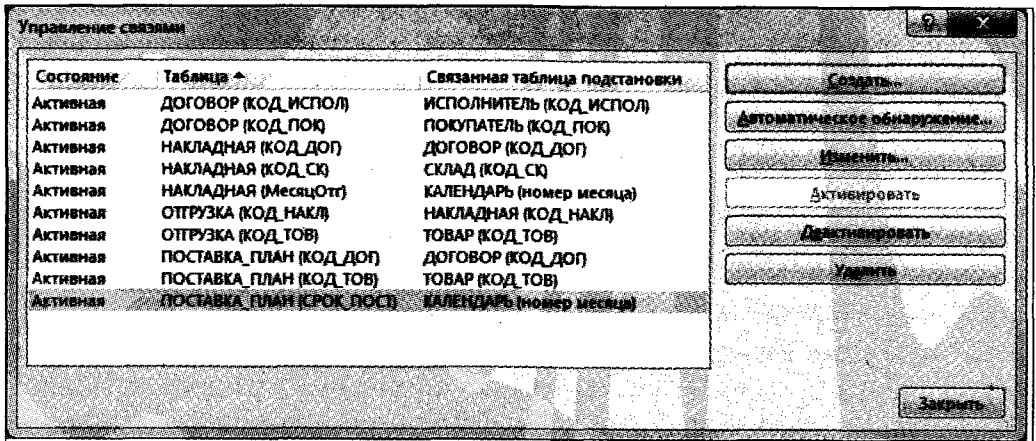


Рис. 8.19. Модель источника данных сводной таблицы

В модели данных, как и в реляционных базах данных, связи таблиц могут быть «один-к-одному» или «один-ко-многим», но не «многие-ко-многим». Связь «многие-ко-многим» приводит к ошибкам. Типы данных в полях связи должны быть совместимыми.

Для создания или изменения связи предназначены соответствующие кнопки в окне **Управление связями** (Manage Relationships). На рис. 8.20 показано окно создания связи между таблицами **ПОКУПАТЕЛЬ** и **ДОГОВОР**, находящимися в отношении 1 : М (один покупатель может заключить несколько договоров, и каждый договор заключен с одним покупателем).

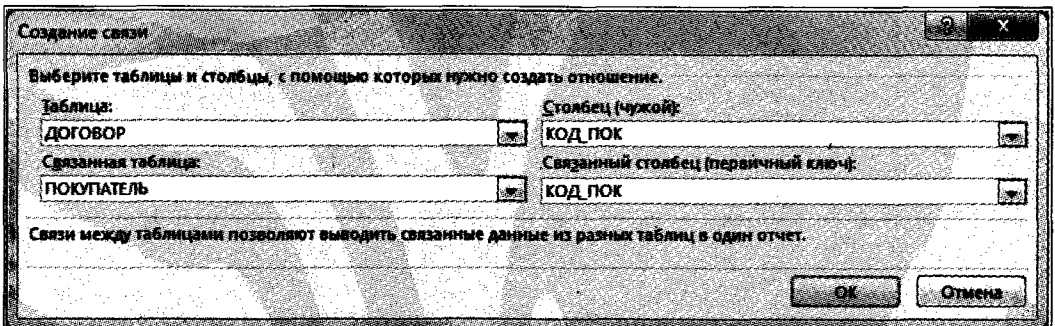


Рис. 8.20. Создание связи между таблицами **ПОКУПАТЕЛЬ** и **ДОГОВОР**

Следует обратить внимание, что связь между таблицами модели может устанавливаться только по одному полю. Если в таблицах реляционной базы данных имеются составные ключи, для связи можно использовать суррогатный ключ — поле с типом данных Счетчик. В следующем далее примере (см. рис. 8.23) в таблице НАКЛАДНАЯ в качестве такого поля использован уникальный идентификатор — код накладной (КОД_НАКЛ).

В таблицах модели данных для связи таблиц допускается использование полей с уникальными значениями и любыми типами данных. Поэтому в качестве таких полей могут быть использованы простые реальные ключи таблицы наряду с ключами с типом данных Счетчик. Для таблиц нижнего уровня в иерархии таблиц, а в модели это таблицы фактов, ключи могут быть и составными, поскольку у них нет подчиненных таблиц и их не надо использовать в качестве ключей связи. Кроме того, реальные ключевые поля часто не используются при анализе данных, и их можно исключить из таблиц модели наряду с другими не нужными полями, если для связи задействован суррогатный ключ таблицы. Это позволит сократить объем требуемой памяти и обеспечит большую скорость анализа.

Таблица НАКЛАДНАЯ включает вычисляемое поле Месяц, по которому в базе данных нельзя установить связь с другой таблицей. После импорта таблиц в книгу Excel поле теряет статус вычисляемого, и необходимая связь может быть установлена.

Использование модели данных избавляет от необходимости предварительного создания запросов и, что особенно важно, обеспечивает возможность на одном источнике решать все необходимые задачи анализа.

Подход к анализу данных, основанный на моделях, включающих любое количество взаимосвязанных таблиц данных, обеспечивает решение в сводных таблицах всего спектра аналитических задач предметной области.

Вариант отчета сводной таблицы (рис. 8.21), построенной на основе представленной модели данных, наглядно демонстрирует возможности автоматического агрегирования данных разных таблиц без подготовки SQL-запросов. Агрегирование выполнено как по количеству заказанных товаров, так и по количеству отгруженных товаров по исполнителям и договорам за выбранный период.

Месяц	Исполнитель		Итого		Итого
	Ильина	Д222	Ильина	Д777	Итого
Наименование товара	Количество по договору	Количество по накладным	Количество по договору	Количество по накладным	Итого
Корпус MiniTower	30				30
Монитор 17LG	20	12	110	15	130
Сканер Acer	130	10			130

Рис. 8.21. Отчет сводной таблицы с агрегированными данными разных таблиц

Представленная сводная таблица содержит поля из пяти разных таблиц. Эта задача оказалась настолько легкой благодаря заранее созданным связям между таблицами. Так как связи между таблицами существовали в источнике данных и были импортированы все таблицы сразу, приложение Excel смогло воссоздать эти связи в модели.

ЗАМЕЧАНИЕ

При разработке макета сводной таблицы можно использовать поля из импортированных таблиц, не проверяя наличия необходимых связей. Если приложению Excel не удастся определить, как внести поле в отчет сводной таблицы, будет предложено создать связь между таблицами.

Для наглядного отображения результатов фильтрации и агрегирования анализируемых данных постройте диаграмму, связанную со сводной таблицей и синхронно отображающую все ее изменения. На рис. 8.22 приведена диаграмма, соответствующая сводной таблице, показанной на рис. 8.21. В диаграмме, так же как и в сводной таблице, можно изменять значения параметров (измерений) для получения новых показателей, и все выполненные изменения будут отображены в сводной таблице.

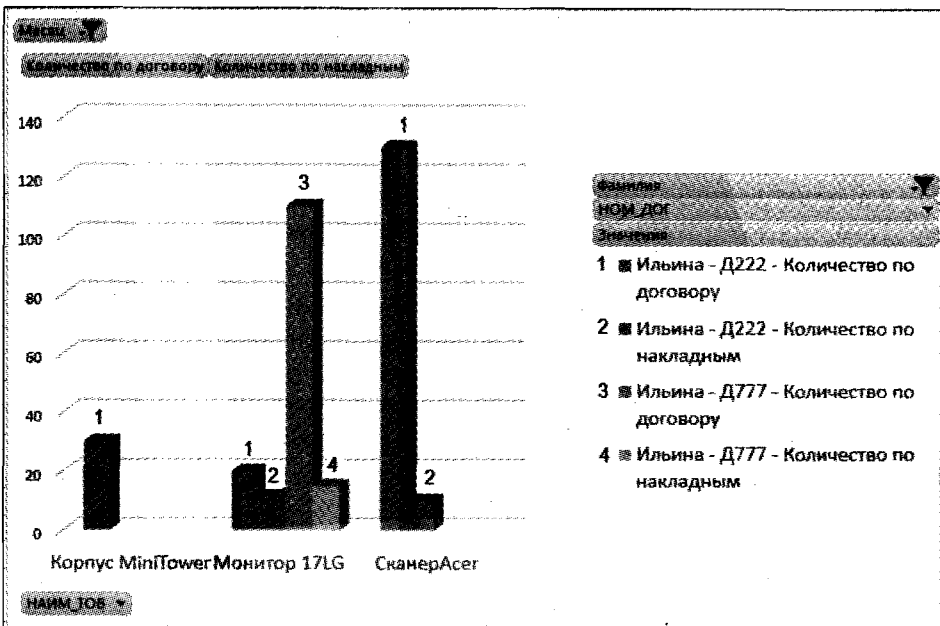


Рис. 8.22. Сводная диаграмма для сравнения количественных показателей заказанного и отгруженного товара

На рис. 8.23 показана схема базы данных, использованная при создании модели данных приведенного сводного отчета и диаграммы. Для отображения модели данных в графическом виде нужна надстройка PowerPivot для Excel (см. далее).

При использовании одной таблицы в качестве источника данных сводной таблицы для решения такой задачи нужно создать несколько запросов. Для группировки и

суммирования количеств по договорам и отгрузкам нужны отдельные запросы, т. к. таблицы ПОСТАВКА_ПЛАН и ОТГРУЗКА размещены на разных ветвях схемы базы данных и находятся в отношении М : N. Для агрегирования данных по заказам (договорам) нужен один запрос, по отгрузкам — второй. Кроме того, нужен запрос, объединяющий результаты выполнения двух предыдущих.

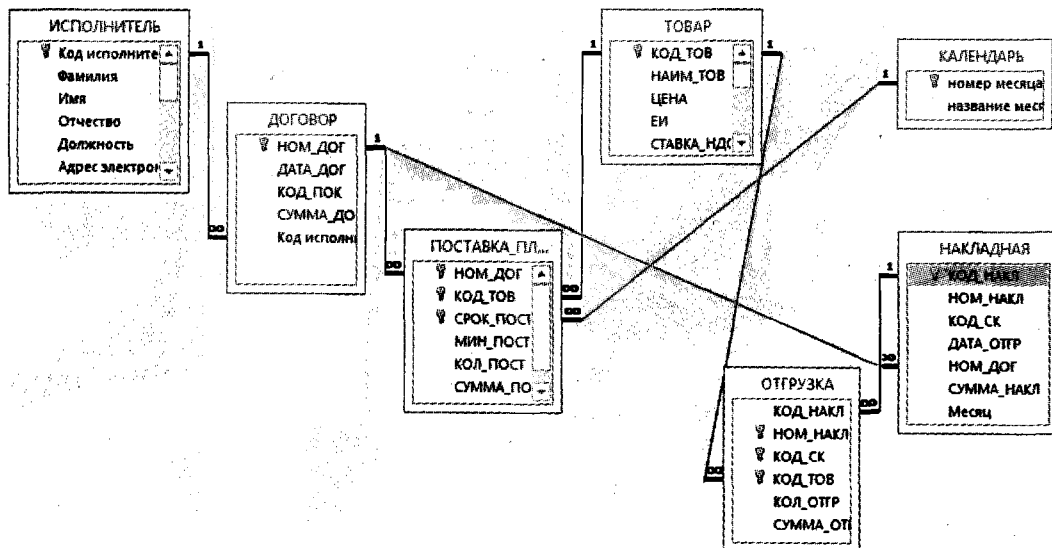


Рис. 8.23. Схема базы данных, задействованной при создании сводной таблицы

Новые возможности оперативного анализа данных в Excel обеспечивают агрегирование данных в соответствии с моделью, как по ветви, ведущей к заказам, так и по ветви, ведущей к отгрузкам, непосредственно при обращении к данным соответствующих таблиц фактов.

Отсутствие возможности получения бизнес-информации из оперативных систем без создания сложных запросов приводило к необходимости в привлечении специалистов ИТ-службы и, как следствие, к усложнению анализа бизнес-информации. Оперативный анализ, основанный на моделях данных, осуществляется без предварительной подготовки запросов в оперативной базе данных, что в значительной степени упрощает самостоятельное создание модели данных анализа конечными пользователями.

Создание тщательно продуманной модели данных упрощается за счет импортируемых из оперативной базы данных гарантированно корректных связей. Агрегирование фактических данных выполняется автоматически, следуя за логикой пользователя, причем в одном отчете может быть получено несколько агрегатов по разным таблицам. Модель данных позволяет собрать все необходимые схемы данных и выполнять анализ различных показателей по многим параметрам в одной сводной таблице, вместо того чтобы создавать несколько запросов и использовать их в разных сводных таблицах.

Работа с моделью в надстройке PowerPivot

PowerPivot — это надстройка для Excel, предназначенная для создания сложных моделей и эффективного анализа данных. Модель данных, являющуюся набором таблиц со связями, можно создать и в Excel, и в PowerPivot. Одна и та же модель данных используется и в книге Excel, и в окне PowerPivot, вне зависимости от того, где она создана. Все данные, импортируемые в Excel, доступны также в PowerPivot, и наоборот.

Импортированные из различных источников данные сохраняются в аналитической базе данных в книге Excel. Мощная локальная подсистема выполняет загрузку, обновление данных из базы и запросы к ним. Данные, внедренные в книгу Excel, сразу становятся доступными для сводных таблиц, сводных диаграмм, где они используются для агрегирования и взаимодействия с ними. Представление данных и взаимодействие обеспечиваются приложением Excel. Данные и объекты представления Excel хранятся внутри одного файла книги. PowerPivot поддерживает файлы размером до 2 Гбайт и позволяет работать с данными в памяти объемом до 4 Гбайт.

Если модель данных содержит незначительное количество таблиц, для ее представления достаточно списочного отображения таблиц и их связей. Но для больших моделей сложно представить ее структуру без схемы. Наглядное представление всех таблиц и их связей в модели данных обеспечивают диаграммы в PowerPivot. С помощью диаграммы можно не только определить связи таблиц, но и быстро определить, какие таблицы должны быть задействованы при решении конкретной аналитической задачи.

Кроме диаграмм, PowerPivot включает разнообразные и мощные средства для разработки и анализа больших объемов данных из различных источников, в том числе средства для просмотра и фильтрации таблиц модели данных, создания вычисляемых столбцов и мер, создания иерархий для столбцов измерения, ключевых показателей эффективности.

Надстройка PowerPivot доступна в Microsoft Office Professional Plus. Она встроена в Excel, но не включена. Для включения PowerPivot выберите опцию меню **Файл | Параметры | Надстройки** (File | Options | Add-Ins). В поле **Управление (Manage)** выберите **Надстройки COM (COM Add-Ins)**, а затем **Перейти (Go)**. Установите флажок **Microsoft Office PowerPivot для Excel** и нажмите кнопку **ОК** — на ленте появится вкладка **PowerPivot** (рис. 8.24). Выберите эту вкладку.

На вкладке **PowerPivot** можно работать с мерами и ключевыми показателями эффективности (KPI), добавлять таблицу Excel в модель для реализации ее связи с таблицей в модели и автоматического обновления последней.

Для перехода в окно **PowerPivot** нажмите на вкладке кнопку **Управление (Manage)**. Для отображения модели данных сводной таблицы в виде схемы в окне PowerPivot выполните команду **Представление диаграммы (Diagram View)**. На рис. 8.25 показана диаграмма модели данных источника сводной таблицы, рассмотренной ранее. Здесь так же, как в схеме базы данных Access, наглядно представлены все таблицы и связи между ними, а установить связь можно простым перетаскиванием поля одной таблицы в другую.

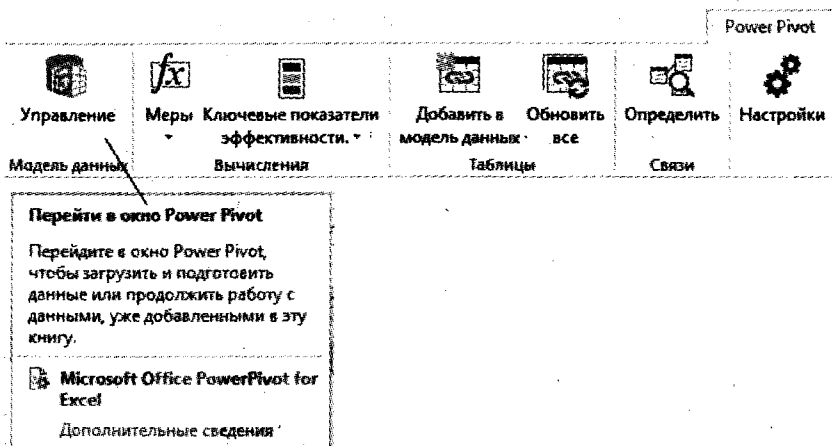


Рис. 8.24. Вкладка ленты надстройки PowerPivot в Excel 2016

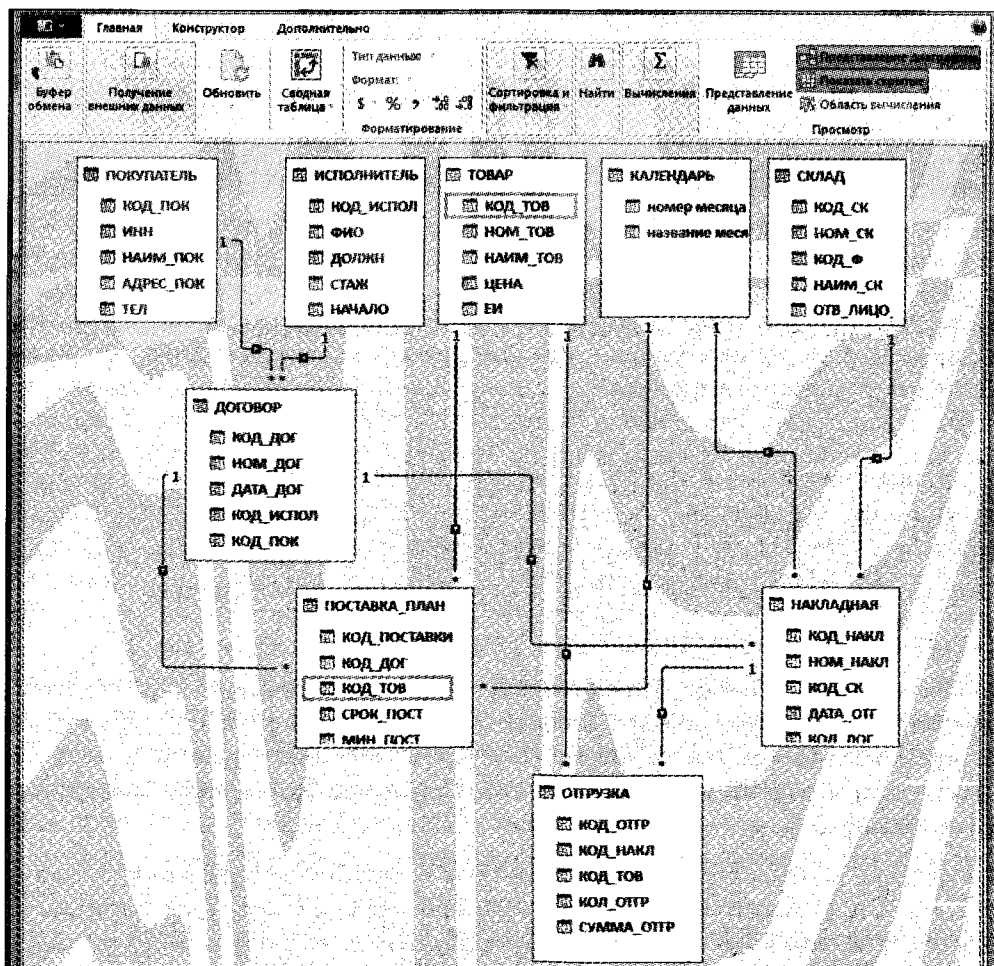


Рис. 8.25. Представление диаграммы модели данных

Для просмотра, сортировки, фильтрации данных всех таблиц модели, дополнения их вычисляемыми столбцами и мерами (вычисляемыми полями) выполните команду **Представление данных (Data View)**. На рис. 8.26 в нижней части отображены вкладки для всех таблиц модели и показана таблица **ДОГОВОР** с отображением для нее связей с таблицами **НАКЛАДНАЯ** и **ПОСТАВКА_ПЛАН**.

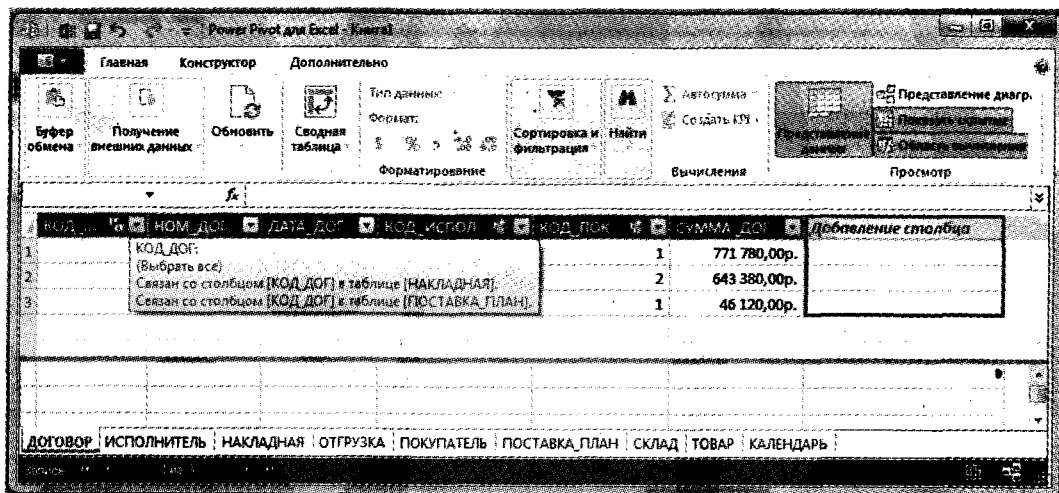


Рис. 8.26. Представление данных таблицы **ДОГОВОР**

Для добавления в таблицу вычисляемого столбца достаточно щелкнуть на поле **Добавление столбца (Add Column)** и в строке ввода формул записать выражение, определяющее значения столбца и начинающееся со знака равенства. Формула автоматически применяется ко всему столбцу. Значения в столбце вычисляются для каждой строки сразу после ввода формулы. Повторно значения столбца вычисляются при обновлении базовых данных. В выражении могут быть использованы полные имена столбцов таблицы, включающие имя таблицы и столбца. Можно создавать вычисляемые столбцы на основе других вычисляемых столбцов и мер. Например, для добавления в таблицу **НАКЛАДНАЯ** столбца со значениями номера месяца отгрузки, можно в вычисляемом столбце записать выражение:

```
=MONTH ('НАКЛАДНАЯ' [ДАТА_ОТГ])
```

В любом столбце таблицы можно создать *меру* (вычисляемое поле с итогами), которое, наряду с другими полями, отобразится в списке полей, доступных в сводной таблице. Таким вычисляемым полем может быть, например, стоимость всех остатков товара, сумма заказов, сумма отгрузок. Пример создания меры приводится далее (см. рис. 8.33).

Ввод формул в вычисляемые столбцы и поля выполняется с помощью языка формул **DAX**. Он не является языком программирования. Для создания формулы **DAX** вводится знак равенства и затем имя функции или выражение и необходимые значения или аргументы. **DAX** предусматривает ряд функций, используемых в формулах Excel, включая разнообразные функции, которые позволяют работать со стро-

ками, выполнять вычисления с использованием даты и времени или создавать условные значения. Однако формулы DAX отличаются от формул Excel:

- ❑ для вычислений на уровне строк предусмотрены функции, позволяющие использовать текущее значение строки или связанное с ним значение для вычислений, зависящих от контекста;
- ❑ включен тип функции, результатом выполнения которой является таблица, а не единичное значение. Эти функции также можно использовать для передачи данных другим функциям;
- ❑ функции операций со временем позволяют выполнять вычисления с использованием диапазонов данных и сравнивать результаты по параллельным периодам времени.

DAX включает и дополнительные функции, предназначенные для работы с реляционными данными и создания динамических агрегатов.

Создание модели и отчета сводной таблицы в PowerPivot

В PowerPivot можно как просмотреть и доработать модель данных ранее созданной сводной таблицы, так и создать модель с нуля. Создание модели данных начинается уже в процессе импорта таблиц, для каждой из которых можно изменить состав столбцов и выполнить фильтрацию строк. При импорте из реляционной базы данных, в которой установлены связи между таблицами, они импортируются вместе с таблицами. Таким образом модель данных может быть создана автоматически.

Для импорта таблиц из различных источников данных воспользуйтесь одной из команд группы **Получение внешних данных** (Get External Data) на вкладке **Главная** (Home) в окне PowerPivot. Для перехода в это окно выполните в окне Excel на вкладке **PowerPivot** команду **Управление** (Manage). Для импорта таблиц из реляционной базы данных Access выберите **Из базы данных | Из Access** (From Database | From Access) — откроется окно **Мастер импорта таблиц** (Table Import Wizard). Мастер импорта внешних данных обеспечит подключение к нужным источникам данных, в нашем случае к базе данных Access, выбор таблиц и фильтрацию данных при добавлении их в файл Excel. В окне мастера укажите имя базы данных, проверьте соединение и перейдите к следующему окну мастера, где осуществляется **Выбор способа импорта данных** (Choose How to Import the Data). Можно импортировать таблицы и существующие в базе запросы или написать на языке SQL запрос, определяющий выборку нужных данных из таблиц базы. В последнем случае в модель импортируется таблица с именем запроса, содержащая результаты выполнения запроса.

На рис. 8.27 показано окно мастера для выбора таблиц и запросов (представлений) из базы данных «Поставка товаров», которые нужно импортировать. Для автоматического выбора связанных таблиц базы данных воспользуйтесь кнопкой **Выбрать**

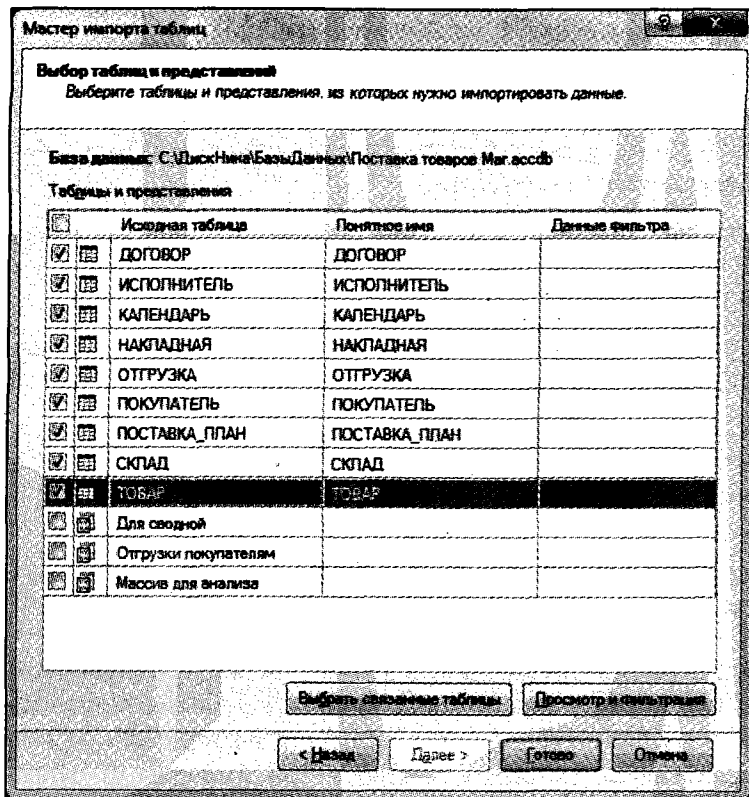


Рис. 8.27. Выбор связанных таблиц базы данных «Поставка товаров»

связанные таблицы (Select Related Tables). Перед выбором связанных таблиц необходимо выбрать как минимум одну таблицу или представление. Имена выбираемых таблиц отмечаются флажками. Таким образом можно в модели данных сводной таблицы получить полную схему базы данных.

Для просмотра и фильтрации данных таблицы выберите ее и нажмите соответствующую кнопку. В открывшемся окне (рис. 8.28) отобразится выделенная таблица (на рис. 8.27 это таблица ТОВАР). Здесь можно снять флажки со столбцов, которые не нужно включать в таблицу при импорте, и путем выбора значений в столбцах определить строки для включения. Выполните фильтрацию данных для всех таблиц.

Завершив выбор и фильтрацию таблиц, нажмите кнопку ОК, чтобы приступить к импорту таблиц. На рис. 8.29 показано последнее окно мастера с результатами импорта таблиц.

Закройте окно. Выполните проверку, если необходимо, добавьте связи между таблицами, вычисляемые элементы, а затем с помощью соответствующей команды создайте сводную таблицу. Сводная таблица создается только на пустом новом листе и не может создаваться на уже имеющемся листе со сводной таблицей (рис. 8.30).

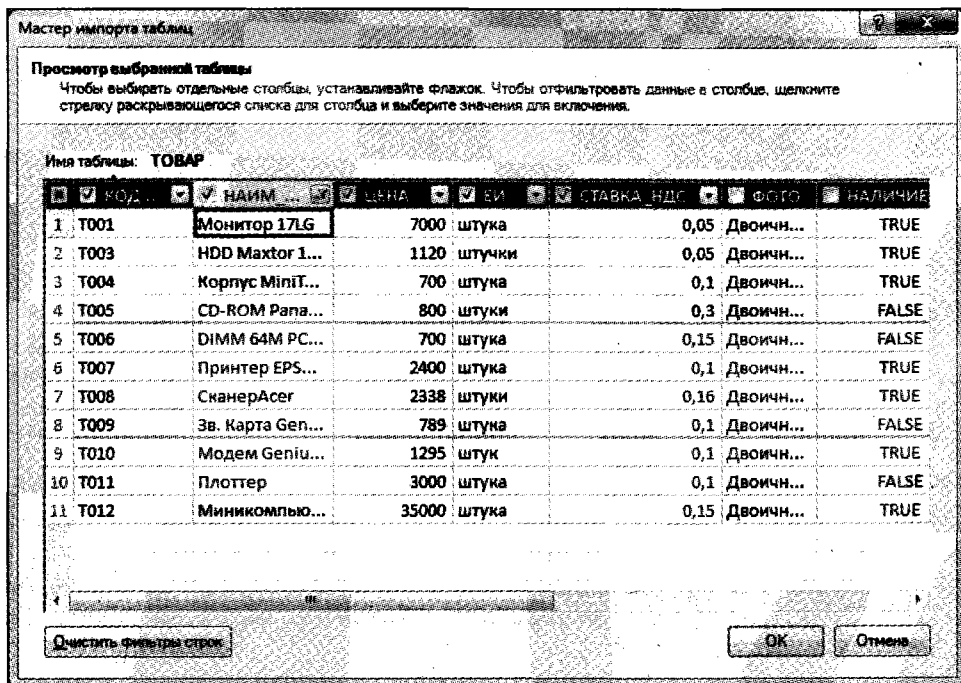


Рис. 8.28. Окно мастера для выбора столбцов и строк импортируемой таблицы ТОВАР

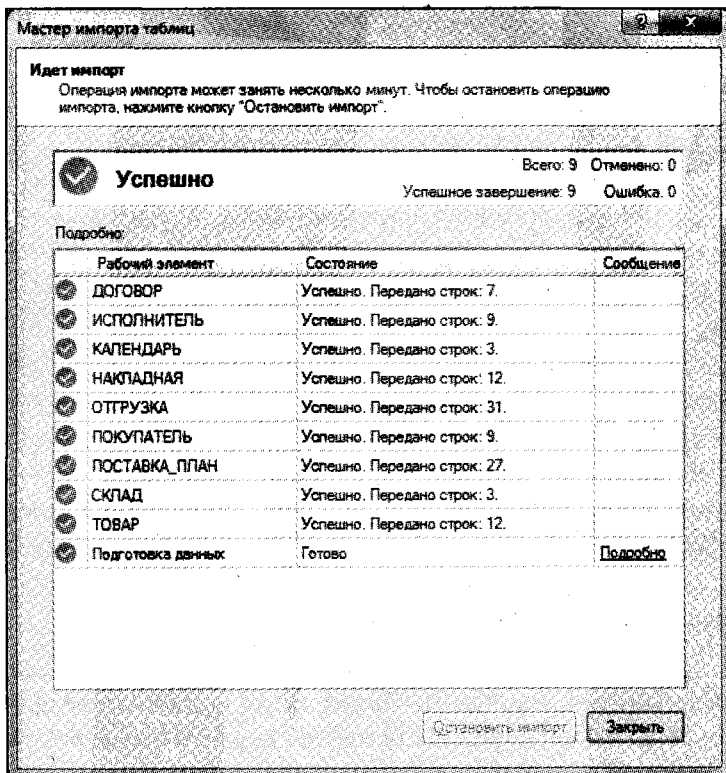


Рис. 8.29. Окно мастера с результатами импорта таблиц базы данных «Поставка товаров»



Рис. 8.30. Сообщение при создании сводной таблицы

ОБРАТИТЕ ВНИМАНИЕ

Настройка PowerPivot обеспечивает включение в модель данных таблиц из многих источников, а также отбор только нужных столбцов и строк в каждой из них. В PowerPivot в любой момент из любого внешнего источника можно добавить новые таблицы, которые включаются в модель, и нужно лишь установить их связи с другими таблицами.

Для каждой импортированной таблицы создается запрос, который используется при выполнении обновления таблицы данными из источника. Инструкцию SQL в запросе можно просмотреть в окне **Изменение свойств таблицы** (Edit Table Properties), открыв его на ленте **Конструктор** (Design) командой **Свойства таблицы** (Table Properties) и переключившись в **Редактор запросов** (Query Editor) (рис. 8.31).

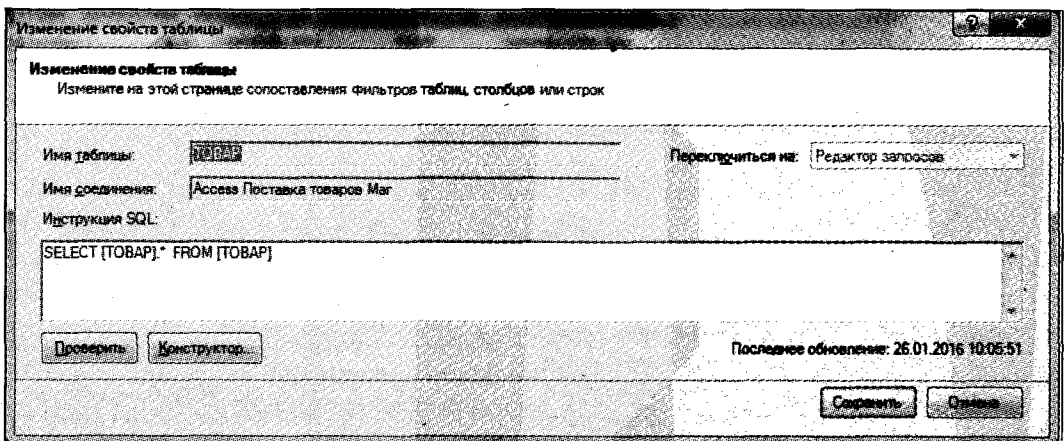


Рис. 8.31. Запрос для импортированной таблицы ТОВАР

В редакторе запросов можно изменить инструкцию SQL, например, сократив список импортируемых столбцов:

```
SELECT [ТОВАР].[КОД_ТОВ], [ТОВАР].[НАИМ_ТОВ], [ТОВАР].[ЦЕНА], [ТОВАР].[ЕИ],
[ТОВАР].[СТАВКА_НДС], [ТОВАР].[НАЛИЧИЕ_ТОВ], [ТОВАР].[ОСТАТОК]
FROM [ТОВАР]
```

Записав в инструкцию предложение с условиями отбора, можно отфильтровать импортируемые данные. Результат нового представления таблицы отобразится сразу после сохранения запроса.

Для анализа данных на основе созданной модели создайте отчет сводной таблицы или/и диаграммы, воспользовавшись соответствующей командой на вкладке Главная (Home) окна PowerPivot. На новом или существующем листе откроются Поля сводной таблицы (PivotTable Fields), представленные набором таблиц со списками доступных полей, и пустой макет сводной таблицы.

Чтобы получить в сводной таблице сведения о суммах, на которые покупателям в соответствии с договорами должны быть отгружены товары за указанный период, и суммах, на которые фактически отгружены товары, а также суммы долга, перетащите поля из списка в макет сводной таблицы, как показано на рис. 8.32. В сводной таблице полям значений, которые вычислены на основе полей СУММ_ПОСТ и СУММ_ОТГР, присвоены пользовательские имена Сумма по договору и Сумма по накладной соответственно.

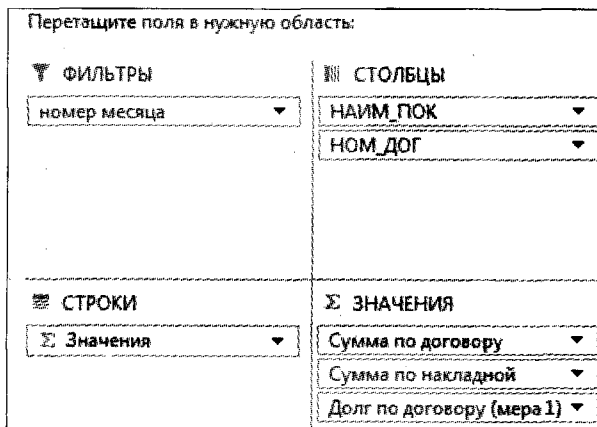


Рис. 8.32. Определение макета сводной таблицы

Для вычисления долга по договору создайте меру, воспользовавшись соответствующей командой на вкладке PowerPivot. В окне Мера (Measure) выберите таблицу ПОСТАВКА_ПЛАН. Здесь же можно присвоить мере удобное имя. Запишите в поле формулы выражение:

=SUM('ПОСТАВКА_ПЛАН' [СУММА_ПОСТ]) -SUM('ОТГРУЗКА' [СУММА_ОТГР])

Проверив формулу, нажмите кнопку ОК — в списке полей таблицы ПОСТАВКА_ПЛАН появится новое вычисляемое поле по умолчанию с именем мера 1. Автоматически это поле будет добавлено в область значений сводной таблицы (рис. 8.33).

номер месяца	1						
Покупатель/Договор	Компьютер маркет		Компьютер маркет		Перспектива	Перспектива	Общий итог
Значения	Д111	Д333	Итого	Д222	Итого		
Сумма по договору	596 120,00р.	500 000,00р.	1 096 120,00р.	50 000,00р.	50 000,00р.	1 146 120,00р.	
Сумма по накладной	50 000,00р.		50 000,00р.	18 876,00р.	18 876,00р.	68 876,00р.	
Долг по договору (мера 1)	546 120,00р.	500 000,00р.	1 046 120,00р.	31 124,00р.	31 124,00р.	1 077 244,00р.	

Рис. 8.33. Отчет сводной таблицы с вычисленной мерой

ЗАМЕЧАНИЕ

Настройка Excel PowerPivot позволяет в таблице источника данных на основе зависимостей атрибутов создавать иерархии. Для рассматриваемого источника данных можно создать иерархию с уровнями Покупатель→Договор. Такая иерархия обеспечит удобный способ выбора значений связанных атрибутов при анализе данных с различной степенью детализации.

Задание 8.2. Создание отчета сводной таблицы на модели данных

На основе модели данных, включающей все таблицы базы данных Access «Поставка товаров» (см. рис. 8.25 и 8.26), последовательно создайте макеты для анализа в различных разрезах заказов и отгрузок товара покупателям. Покажите возможность анализа нескольких фактов в одной сводной таблице. Обеспечьте просмотр как детальных, так и агрегированных данных по различным параметрам.

Создайте сводную таблицу:

- для подсчета общего количества каждого товара, отгруженного с каждого склада, и со всех складов в целом;
- для анализа отгрузок товара покупателям в количественном и стоимостном выражении в заданный период времени;
- для анализа заказов товара покупателями в количественном и стоимостном выражении в заданный период времени;
- для сравнительного анализа заказов и отгрузок товара покупателям в количественном и стоимостном выражении в заданный период времени.

Контрольные вопросы

1. Можно ли, используя сводную таблицу, изменять данные в ее источнике — объекте базы данных?
2. Какие объекты базы данных Access могут быть представлены в источнике данных сводной таблицы и диаграммы?
3. Назовите области сводной таблицы, в которые могут быть размещены поля источника данных?
4. Как открыть список полей, доступных для использования в сводной таблице?
5. Можно ли использовать в сводной таблице Excel 2013/2016 несколько таблиц базы данных?
6. Необходимо ли устанавливать связи между таблицами источника данных сводной таблицы?
7. В каких отношениях могут находиться таблицы источника данных сводной таблицы?
8. При анализе данных в сводных таблицах данные выбираются непосредственно из базы данных?

9. Какая надстройка Excel позволяет представлять модель данных сводной таблицы в виде схемы?
10. Допустимы ли связи таблиц в модели данных сводной таблицы по нескольким полям?
11. Возможно ли использование нескольких полей источника в одной области сводной таблицы?
12. Отразятся ли изменения в данных источника на ранее созданной сводной таблице или диаграмме?

Ответы

1. Нет.
2. Таблицы и запросы.
3. Области: **СТРОКИ, СТОЛБЦЫ, ФИЛЬТРЫ** и **Σ ЗНАЧЕНИЯ**.
4. Командой **Список полей (Field List)** в группе **Показать (Show/Hide)** на вкладке ленты **Работа со сводными таблицами | Анализ (PivotTable Tools | Design)**.
5. Да.
6. Да.
7. 1 : 1 и 1 : M.
8. Нет. При определении источника данных таблицы импортируются в книгу Excel.
9. Надстройка **PowerPivot** для Excel, доступная в **Microsoft Office Professional Plus**.
10. Нет.
11. Да.
12. Только при обновлении источника данных.

ГЛАВА 9



Разработка приложений с использованием макросов

Как показано в предыдущих главах книги, база данных Access — это не только комплект таблиц, хранящих взаимосвязанные данные, но и набор средств для работы с данными, таких как запросы, формы и отчеты. При разработке приложения с помощью Access они составляют его основу. При этом формы образуют основу интерфейса, обеспечивая интерактивный ввод, просмотр и изменение данных. Также может быть создана форма для управления приложением, представляющая все его подзадачи и обеспечивающая пользователя простым доступом ко всем его функциям. Отчеты же обеспечивают обобщенное представление результатов обработки данных и их вывод на экран и печать. В терминологии Access формы и отчеты называются *объектами приложения*.

Для автоматизации выполнения задач, связи различных объектов, создания, редактирования и автоматизации более сложной логики приложений необходимо использовать *средства программирования*. Объектно-ориентированный язык программирования VBA (Visual Basic for Applications), являясь общим средством программирования для всего семейства Microsoft Office, позволяет создавать в среде Access целостные графические диалоговые приложения пользователя с большими возможностями по управлению ими и по контролю за их выполнением, решать и автоматизировать выполнение самых сложных задач.

Более простым языком программирования, позволяющим добавлять функциональные возможности и автоматизировать выполнение задач приложения, является *язык макросов*. Использование макросов для автоматизации управления реакцией приложения на действия пользователя в формах или отчетах позволяет создавать полноценные интерактивные приложения без написания кода на VBA.

Макросы обеспечивают выполнение части команд, доступных в VBA, а создание макроса для пользователей оказывается проще, нежели написание кода VBA. Конструктор макросов снабжен ясным и понятным интерфейсом, поддерживающим функции IntelliSense, и позволяет существенно упростить процесс добавления функциональных возможностей в приложение базы данных.

В веб-приложениях Access, основанных на базах данных, опубликованных в SharePoint, для программирования используются только макросы. Код VBA несовместим со средствами веб-публикации.

Появившиеся в Access 2010 *макросы данных* обеспечивают добавление логики к данным в исходных таблицах. Для связи макросов данных с действиями по добавлению, обновлению, удалению записей в таблице достаточно при открытой таблице на вкладке **Работа с таблицами | Таблица** (Table Tools | Table) лишь щелкнуть на нужной кнопке (что будет показано далее). Сосредоточение бизнес-логики в таблицах позволяет распространить автоматизацию за пределы клиентского приложения Access на базы данных веб-приложений Access и другие приложения, обновляющие таблицы Access.

Макрос (от слова «макрокоманда») — это программа, состоящая из последовательности макрокоманд. А *макрокоманда* — это инструкция, ориентированная на выполнение определенного действия над объектами Access и их элементами.

Например, макрокомандой можно открыть форму или отчет, напечатать отчет, запустить на выполнение запрос, применить фильтр, присвоить значение, создать свое меню, организовать выполнение различных ветвей алгоритма в зависимости от условий. Макрокоманда **ЗапускКомандыМеню** (RunMenuCommand) позволяет выполнить любые встроенные команды Access, которые выводятся на вкладках ленты или в контекстных меню и соответствуют режиму выполнения макрокоманды. Имеющийся в Access набор макрокоманд (более 50) реализует практически любые действия, которые необходимы для решения задач.

Макрос может быть представлен как отдельный объект, который отображается в области навигации в группе **Макросы** (Macros), — это *изолированный макрос*. Кроме того, макрос может быть связан с любым событием в форме, отчете или элементе управления и тогда он внедряется в форму или отчет — это *внедренный макрос*. Такой макрос не отображается как объект в группе **Макросы** (Macros), а становится компонентом формы или отчета.

Изолированные макросы могут запускаться на выполнение прямо из области навигации. Возможно решение задач с помощью цепочки взаимосвязанных макросов, первый из которых запускается на выполнение из области навигации, — пользователь запускает главный макрос на выполнение, и далее все управление выполнением задачи осуществляется изнутри макроса: макрос сам открывает нужные объекты, выбирает и обрабатывает данные, а также вызывает другие макросы, следуя алгоритму, приводящему к решению задачи. При необходимости из макроса может быть инициирован диалог с пользователем. Для перехода по различным ветвям макроса используется блок управления **Если** (If).

Изолированный макрос может выполняться в ответ на многочисленные виды событий, возникающих в формах, отчетах и их элементах управления. Внедренный макрос всегда связывается с событием и сохраняется в форме или отчете. События наступают, прежде всего, при выполнении пользователем определенных действий с объектами. Примерами событий являются: изменение данных в поле, открытие или закрытие формы или отчета, нажатие кнопки в форме и просто передача фокуса от одного поля к другому. Связь макросов с событиями позволяет автоматизировать приложения, используя макросы для открытия форм, печати отчетов, выполнения последовательности запросов, для выполнения действий, зависящих от значений некоторого поля в базе данных, для вывода пользовательских сообщений или

отключения предупреждающих сообщений во время выполнения запросов действия и многого другого. Сохранение внедренных макросов вместе с формами и отчетами упрощает управление объектами приложения.


Программы на языке макросов реализуют алгоритмы решения отдельных задач приложения. Механизм связывания макросов с событиями в объектах позволяет объединить разрозненные задачи приложения в единый комплекс, управляемый пользователем. Пользователь, выполняя различные действия, прежде всего в формах, инициирует выполнение макросов, автоматизирующих решение связанных с действиями пользователя задач.

Конструирование макроса

Создание макросов осуществляется в диалоговом режиме и сводится к записи в окне конструктора макроса последовательности макрокоманд, для которых задаются аргументы. Каждому макросу присваивается имя. При выполнении макроса макрокоманды выполняются последовательно в порядке их расположения, обрабатывая данные или объекты, указанные в аргументах макрокоманд. Как уже отмечалось, для изменения порядка выполнения макрокоманд может быть использован логический блок управления Если (If).

Выполнение изолированных макросов может сводиться к его открытию, как это делается и для других объектов базы данных. Помимо этого, Access предоставляет возможность автоматически инициировать выполнение макроса при наступлении некоторого события. Для связи макроса с событием достаточно в окне свойств объекта или его элемента управления внести в строку этого события имя макроса или создать внедренный макрос. События, с которыми можно связать макрос, представлены в свойствах форм и отчетов и их элементов управления.

Создание изолированного макроса, являющегося отдельным объектом базы данных, начинается с выполнения команды **Макрос (Macros)** на ленте **Создание (Create)** в группе **Макросы и код (Macros & Code)**. В результате выполнения команды открывается окно макроса и каталог макрокоманд (рис. 9.1).

Для создания внедренного макроса надо выбрать в области навигации форму или отчет и выполнить в его контекстном меню команду **Конструктор (Design)** или **Режим макета (Layout View)**. Далее открыть окно свойств и выбрать элемент управления, раздел или форму (отчет) целиком, на вкладке **События (Event)** выбрать событие, по которому будет выполняться макрос, и нажать кнопку строителя . В диалоговом окне **Построитель (Choose Builder)** выделить пункт **Макросы (Macro Builder)** и нажать кнопку **ОК**. В результате откроется окно макроса, такое же, как при создании изолированного макроса.

Внедренные макросы отличаются от изолированных макросов тем, что они выполняются при наступлении события, с которым связаны, и хранятся в формах и отчетах. Они не отображаются в виде объектов в группе **Макросы (Macros)** в области навигации. Это упрощает управление базой данных, поскольку не нужно следить за тем, какие макросы относятся к какой форме или отчету. Внедренные макросы

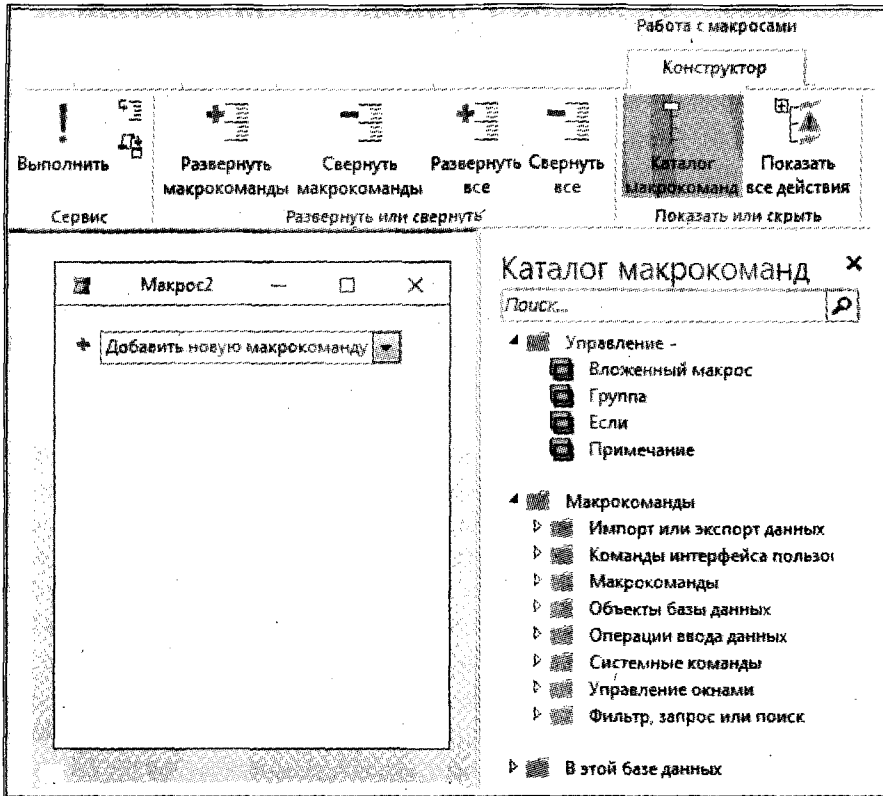


Рис. 9.1. Окно конструирования макроса и Каталог макрокоманд

сохраняются в составе формы или отчета и при их копировании, импорте или экспорте.

Формирование макрокоманд в окне макроса

При создании нового макроса в его окне отображается поле **Добавить новую макрокоманду** (Add New Action) с раскрывающимся списком (см. рис. 9.1). В списке представлен весь доступный набор макрокоманд. Этот набор можно изменять, нажимая кнопку **Показать все действия** (Show All Action) на ленте конструктора макросов в группе **Показать или скрыть** (Show/Hide). Если кнопка не нажата, то в набор не включаются так называемые *небезопасные макрокоманды*. К ним относятся макрокоманды, изменяющие базу данных или получающие доступ к ресурсам вне базы. Это позволяет пользователю при необходимости исключить использование небезопасных макрокоманд в своем приложении и таким образом обрести возможность открывать базу данных с полным набором функций, даже если она не получила статус доверенной. Как известно, доверенной база данных становится, например, в случае, если она получена из источника, включенного в список надежных.

Для постоянного отображения в окне конструктора макросов списка всех доступных макрокоманд нужно нажать кнопку **Каталог макрокоманд** (Action Catalog).

В окне каталога для простоты поиска макрокоманды разбиты на функциональные группы и обеспечена очень удобная возможность поиска макрокоманды по имени (см. рис. 9.1). Кроме того, в каталоге имеется раздел **В этой базе данных** (In this Database), где представлены не только все макросы из области навигации, но и макросы, внедренные в различные формы и отчеты. Содержимое этих макросов также может быть скопировано в другие конструируемые макросы.

Для ввода макрокоманды в поле можно нажать кнопку раскрытия списка макрокоманд и выбрать нужную. Имя макрокоманды можно ввести с клавиатуры, при этом система помогает сформировать его. Можно в каталоге выбрать макрокоманду и перетащить ее в окно макроса. Место размещения новой макрокоманды отмечается там оранжевой полосой.

Окно конструирования внедренного макроса отличается от окна конструирования изолированного макроса только записью в заголовке окна. Для изолированного макроса там записано имя макроса, а для внедренного: имя формы (отчета): имя элемента: свойство события, например, ТОВАР: ЦЕНА: После обновления.

После ввода макрокоманды в макросе отображается блок, содержащий имя макрокоманды и строки ее аргументов (рис. 9.2). Значения аргументов задаются путем выбора их из списка, открывающегося в строке аргумента, с помощью строителя или вручную. Для выбранного аргумента или макрокоманды выводятся всплывающие подсказки.

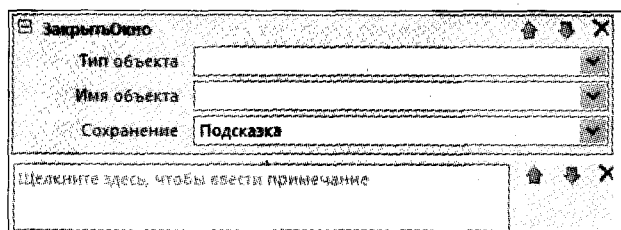


Рис. 9.2. Блок конструирования макрокоманды и ввода комментария

Знак минус слева от имени макрокоманды позволяет скрыть ее аргументы. Зеленые стрелки в правой части блока позволяют перемещать макрокоманду выше или ниже других макрокоманд. Здесь же имеется значок удаления макрокоманды.

Для ввода в макрос комментария используется блок **Примечание** (Comment), размещенный в окне каталога макрокоманд в разделе **Управление** (Program Flow). При перетаскивании этого блока в макрос создается пустой блок, в который и вводится нужный комментарий. После завершения ввода комментарий отображается строкой зеленого цвета, заключенной в знаки /* и */.

Каждая новая макрокоманда макроса добавляется в нужное место макроса, и проще всего это сделать перетаскиванием ее из каталога макрокоманд. Порядок размещения макрокоманд в бланке определяет последовательность их выполнения.

После ввода всех макрокоманд в макрос его надо сохранить, воспользовавшись командой **Сохранить** (Save).

В режиме конструктора для выполнения макроса предназначена команда **Выполнить** (Run). Если макрос уже закрыт, то для его выполнения надо в области навигации выбрать макрос и в контекстном меню выбрать команду **Выполнить** (Run). Кроме того, для выполнения макроса можно использовать кнопку **Выполнить макрос** (Run Macro) на вкладке ленты **Работа с базами данных** (Database Tools) в группе **Макрос** (Macro). Таким образом, реализованная макросом задача может решаться по инициативе пользователя.

Выполнение внедренного макроса происходит автоматически при наступлении события, с которым он связан.

Изолированный макрос, так же как внедренный, может быть связан с любым событием формы (отчета) или их элементов. Для этого, вместо того чтобы вызывать для создания макроса построитель, нужно выбрать ранее созданный изолированный макрос из списка в строке свойства события.

Для просмотра и редактирования существующего изолированного макроса надо выбрать его в области навигации и в контекстном меню нажать кнопку **Конструктор** (Design View). Для редактирования внедренного макроса нужно так же, как при его создании, воспользоваться кнопкой построителя в строке свойства события.

Формирование макроса с помощью мыши

Создание некоторых макрокоманд в макросе может быть выполнено путем переноса с помощью мыши объекта базы данных из области навигации в окно макроса. Макрокоманда по умолчанию создается со значениями аргументов, соответствующими выбранному объекту. Например, при перетаскивании таблицы создается макрокоманда **ОткрытьТаблицу** (OpenTable), открывающая таблицу в режиме таблицы со значением **Изменение** (Edit) для аргумента **Режим данных** (Data Mode).

Возможно перемещение любых объектов, представленных в области навигации базы данных. Соответственно перемещаемому объекту могут быть созданы макрокоманды **ОткрытьТаблицу** (OpenTable), **ОткрытьФорму** (OpenForm), **ОткрытьЗапрос** (OpenQuery), **ОткрытьОтчет** (OpenReport), **ОткрытьМодуль** (OpenModule) или **ЗапускМакроса** (RunMacro).

Использование в макросах ссылок на объекты

В программах на языке макросов, как и на языке VBA, при обработке данных в базе необходимо уметь правильно ссылаться на эти данные. В Access определены объекты, через которые предоставляется возможность получить доступ к данным базы. К таким объектам относятся формы и отчеты. Ссылки на формы, отчеты, их элементы управления и свойства формируются по определенным правилам. Построитель выражений позволяет сформировать такие ссылки простым выбором объекта, элемента управления и свойства в списках. Ко времени выполнения в макросах выражения со ссылкой на объект этот объект должен быть открыт.

Ссылки на объекты и их элементы управления

Ссылаться на объекты можно по имени, но нужно учесть, что в Access объекты объединяются в *семейства*. Формы объединены в семейство **Формы** (Forms), отчеты — в семейство **Отчеты** (Reports). Поэтому ссылка на объект включает имя семейства и — через восклицательный знак — имя объекта. Если имя включает пробелы или специальные символы, его надо брать в квадратные скобки. Например, для ссылки на форму надо записать:

```
Forms! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ] .
```

В ссылке на элемент управления вслед за именем объекта через восклицательный знак записывается имя элемента управления, заключенное в квадратные скобки.

Например, ссылку на поле в форме надо записать так:

```
Forms! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ] ! [СУММА_ДОГ]
```

Ссылку на элемент управления **Надпись** (Caption) с именем **Цена_Надпись** надо записать так:

```
Forms! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ] ! [Цена_Надпись]
```

Обратите внимание, что имя этого элемента управления не совпадает с подписью, выводимой в форме. Имя записывается в свойстве элемента управления на вкладке **Другие** (Other) в строке **Имя** (Name) (рис. 9.3). Так, в форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** элемент управления **Надпись** (Caption) с именем **Цена_Надпись** отображается как **ЦЕНА**. Это значение записано на вкладке **Макет** (Format) в свойстве **Подпись**. Проверяйте имя элемента управления в окне свойств — оно может по умолчанию принимать и другие значения, например **Надпись15**.

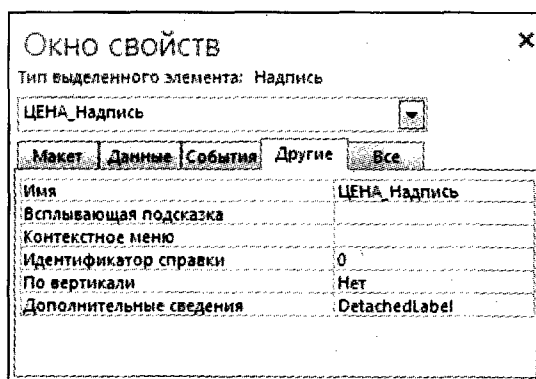


Рис. 9.3. Окно свойств элемента управления **Надпись**

Ссылки на свойство объекта

В ссылке на свойство объекта вслед за именем объекта через точку записывается имя свойства. Перечень свойств формы или отчета можно посмотреть, открыв объект в режиме конструктора или макета и вызвав окно его свойств. Многие свойства

названы несколькими словами с пробелами между ними. Истинное имя свойства таких пробелов не имеет, поэтому в ссылках оно записывается без них. Например, имя свойства **Область выделения** (Record Selectors) должно записываться так: **ОбластьВыделения** (RecordSelectors), имя свойства **Полосы прокрутки** (Scroll Bars): **ПолосыПрокрутки** (ScrollBars).

Ссылку на это свойство формы надо записать так:

```
Forms! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ].ScrollBars
```

ВНИМАНИЕ!

В русифицированной версии Access в строках окна свойств указаны русские имена свойств. В ссылках же следует использовать английский вариант имени свойства.

Ссылки на свойство элемента управления

Для записи ссылки на свойство элемента управления нужно дополнить ссылку на элемент управления через точку именем свойства. Различные типы элементов управления имеют разные свойства.

Например, ссылку на свойство **ВыводНаЭкран** (английский вариант имени: Visible) элемента управления с именем **Цена_Надпись**, которое соответствует строке **Вывод на экран** (Visible) на вкладке **Макет** (Format) (рис. 9.4), надо записать так:

```
Forms! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ]![Цена_Надпись].Visible
```

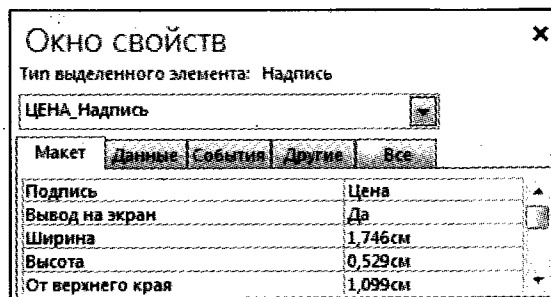


Рис. 9.4. Отображение свойства Visible в окне свойств

В общем виде правило записи ссылок может быть сформулировано следующим образом: оператор ! (восклицательный знак) указывает, что следующий за ним элемент является элементом, определяемым пользователем. Например, оператор ! ставится перед ссылкой на открытую форму, на открытый отчет или элемент управления в открытой форме или отчете. Оператор . (точка) обычно указывает, что следующий за ним элемент определен в Access. Например, оператор . (точка) ставится перед ссылкой на свойства форм, отчетов и элементов управления.

Ссылка на подчиненную форму или отчет

Подчиненная форма (или отчет) рассматривается в Access как элемент управления формы (отчета). Форма (отчет) после внедрения содержится в элементе управления **Подчиненная форма/отчет** (Subform/Subreport). Поэтому ссылка на подчиненную форму (или отчет) записывается как ссылка на элемент управления формы (отчета).

Элемент управления **Подчиненная форма/отчет** (Subform/Subreport) имеет специальное свойство **Форма** (Form) или, соответственно, **Отчет** (Report). Это свойство позволяет ссылаться на элементы управления подчиненных объектов и их свойства. Полная ссылка на свойство элемента управления в подчиненной форме имеет в общем виде следующую структуру:

```
Forms! [Имя_формы] ! [Элемент_Подчиненная_форма] .Form! [Элемент_подчиненной_формы] .Имя_свойства
```

Заметим, что при ссылках на элемент управления в подчиненной форме или подчиненном отчете не обязательно указывать свойство **Form** или **Report**.

Далее приведена ссылка на элемент управления: поле **СУММА ПОСТ** в подчиненной форме **ПЛАН ПОСТАВОК**, встроенной в форму **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**:

```
Forms! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ] ! [ПЛАН ПОСТАВОК] ! [СУММА ПОСТ]
```

Следует заметить, что не всегда нужно в ссылке использовать полное имя элемента управления. Так, для расчета стоимости товара в текущей записи подчиненной формы **ПЛАН ПОСТАВОК** и его сохранения в поле **СУММА ПОСТ** достаточно в аргументах макрокоманды **ЗадатьЗначение** (SetValue) записать: в строке **Элемент** (Item) — имя **[СУММА ПОСТ]**, а в строке **Выражение** (Expression) — **[ЦЕНА] * [КОЛ ПОСТ]** (рис. 9.5).

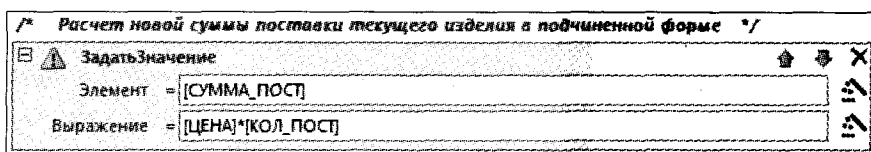



Рис. 9.5. Ссылки на имена элементов управления текущей формы **ПЛАН ПОСТАВОК**

Если макрос вызывается при наступлении некоторого события, например, при обновлении пользователем полей **КОЛ ПОСТ** или **ЦЕНА** в форме **ПЛАН ПОСТАВОК**, то при передаче управления в макрос текущим объектом остается эта форма. Поэтому в макросе для ссылок достаточно использовать только имена полей.

Создание ссылок построителем выражений

Ссылки на объекты, элементы управления и свойства удобно создавать с помощью построителя выражений.

Чтобы вызвать построитель выражений, надо выбрать в окне макроса строку аргумента макрокоманды, в которую требуется ввести выражение, и нажать появившуюся кнопку построителя .

В открывшемся окне **Построитель выражений** (Expression Builder) в поле записи выражений нужно сформировать выражение. Для этого в левом поле в нижней половине окна построителя раскройте двойным щелчком кнопки мыши папку, содержащую объекты. Затем выберите папку необходимого объекта. В среднем поле выберите нужный элемент управления объекта, а в правом — **Значение** (Value), если формируется ссылка на элемент управления, или нужное свойство, и двойным щелчком вставьте ссылку в выражение.

Закончив создание выражения, нажмите кнопку **ОК** — Access скопирует созданное выражение в ту строку, из которой был вызван построитель выражений.

Если строка аргумента, из которой вызывается построитель выражений, уже содержит выражение, то оно автоматически копируется в поле выражений, где может быть отредактировано.

Заметим, что если нужный объект или свойство не появляется в нижней части окна построителя выражений, это означает невозможность их использовать в том контексте, в котором был вызван построитель выражений.

Вложенные макросы

Access предоставляет возможность создавать *вложенные* макросы как в изолированном, так и во внедренном макросе. Вложенный макрос имеет имя и может содержать любые макрокоманды. В области навигации макрос с вложенными макросами отображается как один объект. Целесообразно объединять несколько макросов в одном, если они связаны с решением одной задачи или используются при работе с одним объектом. Работать с макросами, объединенными в группу, часто оказывается удобнее, чем с несколькими отдельными макросами.

Вызывается вложенный макрос с помощью макрокоманды **ЗапускМакроса** (RunMacro) или в ответ на событие. Для ссылки на вложенный макрос используется следующий синтаксис:

ИмяГруппыМакросов.ИмяВложенногоМакроса

Для включения в макрос вложенного макроса нужно выбрать соответствующую макрокоманду из раскрывающегося списка в поле **Добавить новую макрокоманду** (Add New Action) или перетащить **Вложенный макрос** (Submacro) в нужное место из раздела **Управление** (Program Flow) каталога макрокоманд. В макросе отобразится блок (рис. 9.6), в котором по умолчанию вложенному макросу присвоено имя **Sub1**, предоставляется возможность добавления макрокоманд и вставлен признак конца вложенного макроса.

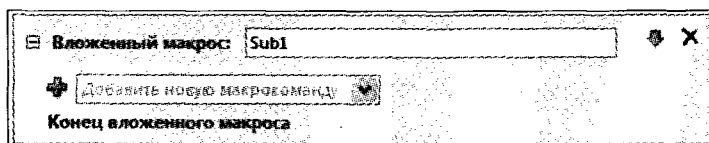


Рис. 9.6. Блок пустого вложенного макроса

Создание изолированного макроса

Рассмотрим на примере создание изолированного макроса, включающего вложенные макросы, которые запускаются на выполнение при инициировании пользователем некоторых событий.

Задача

Пусть при просмотре данных о договоре необходимо иметь возможность отображать информацию о накладных, по которым отгружался товар в соответствии с этим договором.

Для решения задачи используем формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** и **НАКЛАДНАЯ**, созданные ранее (см. главы 5 и 6). Обе формы имеют поле с номером договора **НОМ_ДОГ**. При работе с формой **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** должна открываться форма **НАКЛАДНАЯ**, в которой представлены только соответствующие просматриваемому договору накладные.

Создайте изолированный макрос с именем **Накладные**, включающий вложенные макросы **М1** и **М2**. Пусть макрос **М1** при открытии формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** открывает форму **НАКЛАДНАЯ**, а макрос **М2** — фильтрует записи с накладными по значению поля **НОМ_ДОГ**, взятому из текущей записи формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**. Макрос **М2** должен выполняться, когда пользователь, работая в форме с договорами, инициирует некоторое событие — например, **Двойное нажатие кнопки** (On Dbl Click) или **Вход** (On Enter) для поля с номером договора. Если необходимо отображать накладные только при переходе к другому договору, свяжите макрос **М2** с событием формы **Текущая запись** (On Current). Создание группы позволяет объединить макросы, предназначенные для решения одной задачи, и упростить сопровождение приложения.

1. Для создания изолированного макроса выполните команду **Макрос** (Macro) на вкладке ленты **Создание** (Create) в группе **Макросы и код** (Macros & Code).
2. В открывшемся окне макроса с помощью макрокоманды управления **Вложенный макрос** (Submacro) создайте вложенные макросы с именами **М1** и **М2** (рис. 9.7).
3. Макрос **М1** дополните макрокомандой **ОткрытьФорму** (OpenForm) **НАКЛАДНАЯ**.
4. Макрос **М2** дополните:
 - макрокомандой **ВыделитьОбъект** (SelectObject) **НАКЛАДНАЯ**, которая делает объект текущим и позволит выполнять для него макрокоманды;
 - макрокомандой **ПоказатьВсеЗаписи** (ShowAllRecords), которая отменит фильтр, примененный ранее к форме;
 - макрокомандой **ПрименитьФильтр** (ApplyFilter), которая отфильтрует записи формы **НАКЛАДНАЯ** в соответствии с условием отбора.

В открывшемся окне **Построитель выражений** (Expression Builder) в поле записи выражений нужно сформировать выражение. Для этого в левом поле в нижней половине окна построителя раскройте двойным щелчком кнопки мыши папку, содержащую объекты. Затем выберите папку необходимого объекта. В среднем поле выберите нужный элемент управления объекта, а в правом — **Значение** (Value), если формируется ссылка на элемент управления, или нужное свойство, и двойным щелчком вставьте ссылку в выражение.

Закончив создание выражения, нажмите кнопку **ОК** — Access скопирует созданное выражение в ту строку, из которой был вызван построитель выражений.

Если строка аргумента, из которой вызывается построитель выражений, уже содержит выражение, то оно автоматически копируется в поле выражений, где может быть отредактировано.

Заметим, что если нужный объект или свойство не появляется в нижней части окна построителя выражений, это означает невозможность их использовать в том контексте, в котором был вызван построитель выражений.

Вложенные макросы

Access предоставляет возможность создавать *вложенные* макросы как в изолированном, так и во внедренном макросе. Вложенный макрос имеет имя и может содержать любые макрокоманды. В области навигации макрос с вложенными макросами отображается как один объект. Целесообразно объединять несколько макросов в одном, если они связаны с решением одной задачи или используются при работе с одним объектом. Работать с макросами, объединенными в группу, часто оказывается удобнее, чем с несколькими отдельными макросами.

Вызывается вложенный макрос с помощью макрокоманды **ЗапускМакроса** (RunMacro) или в ответ на событие. Для ссылки на вложенный макрос используется следующий синтаксис:

ИмяГруппыМакросов.ИмяВложенногоМакроса

Для включения в макрос вложенного макроса нужно выбрать соответствующую макрокоманду из раскрывающегося списка в поле **Добавить новую макрокоманду** (Add New Action) или перетащить **Вложенный макрос** (Submacro) в нужное место из раздела **Управление** (Program Flow) каталога макрокоманд. В макросе отобразится блок (рис. 9.6), в котором по умолчанию вложенному макросу присвоено имя **Sub1**, предоставляется возможность добавления макрокоманд и вставлен признак конца вложенного макроса.

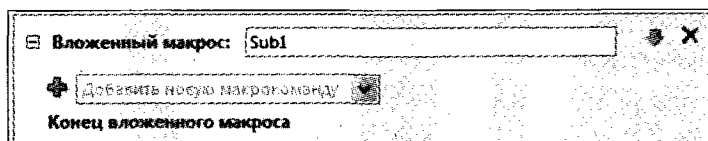


Рис. 9.6. Блок пустого вложенного макроса

Создание изолированного макроса

Рассмотрим на примере создание изолированного макроса, включающего вложенные макросы, которые запускаются на выполнение при инициировании пользователем некоторых событий.

Задача

Пусть при просмотре данных о договоре необходимо иметь возможность отображать информацию о накладных, по которым отгружался товар в соответствии с этим договором.

Для решения задачи используем формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** и **НАКЛАДНАЯ**, созданные ранее (см. главы 5 и 6). Обе формы имеют поле с номером договора **НОМ_ДОГ**. При работе с формой **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** должна открываться форма **НАКЛАДНАЯ**, в которой представлены только соответствующие просматриваемому договору накладные.

Создайте изолированный макрос с именем **Накладные**, включающий вложенные макросы **М1** и **М2**. Пусть макрос **М1** при открытии формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** открывает форму **НАКЛАДНАЯ**, а макрос **М2** — фильтрует записи с накладными по значению поля **НОМ_ДОГ**, взятому из текущей записи формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**. Макрос **М2** должен выполняться, когда пользователь, работая в форме с договорами, инициирует некоторое событие — например, **Двойное нажатие кнопки** (On Dbl Click) или **Вход** (On Enter) для поля с номером договора. Если необходимо отображать накладные только при переходе к другому договору, свяжите макрос **М2** с событием формы **Текущая запись** (On Current). Создание группы позволяет объединить макросы, предназначенные для решения одной задачи, и упростить сопровождение приложения.

1. Для создания изолированного макроса выполните команду **Макрос (Macro)** на вкладке ленты **Создание (Create)** в группе **Макросы и код (Macros & Code)**.
2. В открывшемся окне макроса с помощью макрокоманды управления **Вложенный макрос (Submacro)** создайте вложенные макросы с именами **М1** и **М2** (рис. 9.7).
3. Макрос **М1** дополните макрокомандой **ОткрытьФорму (OpenForm)** **НАКЛАДНАЯ**.
4. Макрос **М2** дополните:
 - макрокомандой **ВыделитьОбъект (SelectObject)** **НАКЛАДНАЯ**, которая делает объект текущим и позволит выполнять для него макрокоманды;
 - макрокомандой **ПоказатьВсеЗаписи (ShowAllRecords)**, которая отменит фильтр, примененный ранее к форме;
 - макрокомандой **ПрименитьФильтр (ApplyFilter)**, которая отфильтрует записи формы **НАКЛАДНАЯ** в соответствии с условием отбора.

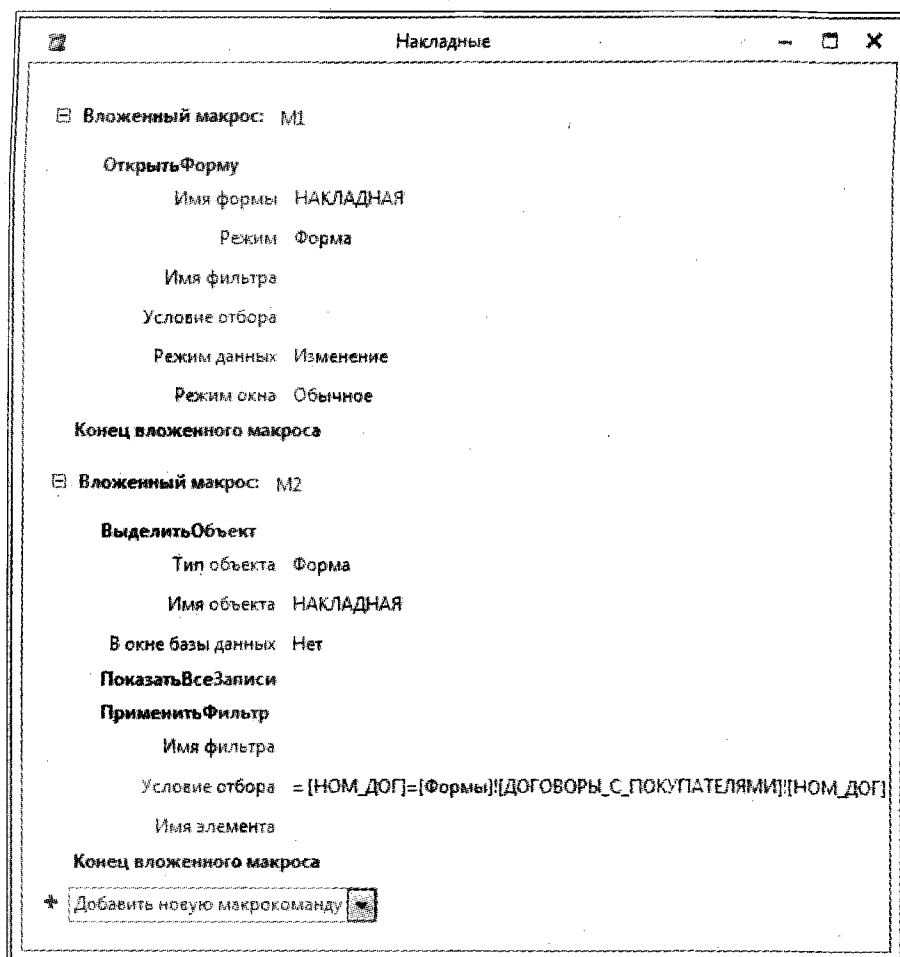


Рис. 9.7. Макрос, организующий синхронный просмотр данных в двух формах

5. Сохраните группу макросов под именем **Накладные**. Это имя будет выводиться в списке макросов в области навигации.
6. Для запуска макроса **M1** установите связь события **Открытие** (On Open) формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** с макросом. Откройте форму в режиме макета или конструктора, щелкните на вкладке ленты конструктора в группе **Сервис** (Tools) на кнопке **Страница свойств** (Property Sheet). В окне свойств формы на вкладке **События** (Event) в качестве значения свойства **Открытие** (On Open) выберите из списка имя макроса **Накладные.M1** (рис. 9.8).
7. Для запуска макроса **M2** при двойном щелчке пользователя в поле с номером договора в свойствах поля **НОМ_ДОГ** формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** в качестве значения события **Двойное нажатие кнопки** (On Dbl Click) выберите **Накладные.M2**.

Событие **Двойное нажатие кнопки** (On Dbl Click) для элемента управления **НОМ_ДОГ** возникает, когда пользователь дважды нажимает и быстро отпускает

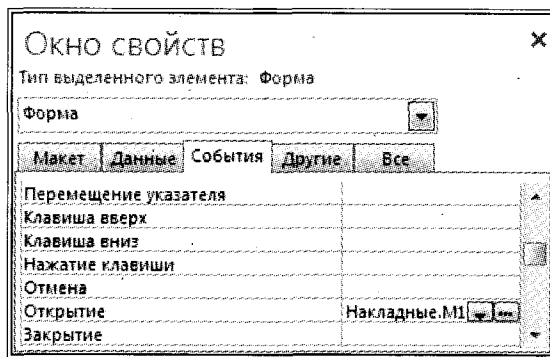


Рис. 9.8. Связь вложенного макроса со свойством формы ДОГОВОРЫ С ПОКУПАТЕЛЯМИ

левую кнопку мыши в тот момент, когда курсор мыши установлен на поле `НОМ_ДОГ` или присоединенной к нему надписи.

8. Чтобы проверить работу макросов, откройте форму **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**. На экране будет отображен результат работы макроса **M1** — открытая форма **НАКЛАДНАЯ**. При этом в форме будут доступны все записи.
9. Выполните двойной щелчок по полю с номером договора в форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**. Результат выполнения макроса **M2** отобразится в форме **НАКЛАДНАЯ**. К ней применен фильтр, в котором в качестве условий отбора записано `[НОМ_ДОГ]=Формы! [ДОГОВОРЫ С ПОКУПАТЕЛЯМИ] ! [НОМ_ДОГ]`, и поэтому доступны только записи со значением номера договора из формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**.

ЗАМЕЧАНИЕ

Для ссылки на элемент управления в текущем объекте достаточно указать только имя элемента, для ссылки на элемент управления других открытых объектов должен использоваться полный синтаксис. Пользуйтесь построителем для формирования сложной ссылки на элемент. Построитель вызывается щелчком на кнопке в правой части строки аргумента.

10. Чтобы макрос **M2** выполнялся, когда пользователь, работая в форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**, входит в поле `НОМ_ДОГ`:
 - удалите связь макроса **M2** с событием **Двойное нажатие кнопки** (On Db! Click) для элемента управления `НОМ_ДОГ`;
 - установите его связь с событием **Вход** (On Enter) этого же элемента управления.

Событие **Вход** (On Enter) наступает при перемещении курсора в поле `НОМ_ДОГ` и при переходе к другой записи, если текущим является поле `НОМ_ДОГ`. Кроме того, при открытии формы, наряду с событием **Открытие** (On Open), для текущего элемента управления формы наступает событие **Вход** (On Enter). При открытии формы текущим элементом управления является поле, которое стоит первым в списке, определяющем последовательность перехода по элементам управления области данных. Если поле `НОМ_ДОГ` размещено в области данных

формы и стоит на первом месте в списке, то оно будет текущим, и для него наступит событие **Вход** (On Enter). В результате при открытии формы выполнятся оба макроса.

11. Просмотрите последовательность перехода по элементам управления области данных в форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**, выполнив команду **Переходы** (Tab Order) на вкладке **Инструменты конструктора форм | Конструктор** (Form Design Tools | Design) в группе **Сервис** (Tools).
12. Если поле `ном_дог` в списке последовательности переходов стоит не на первом месте, переместите его.

ОБРАТИТЕ ВНИМАНИЕ!

Одно действие пользователя может приводить к возникновению нескольких событий. При этом важно знать последовательность наступления этих событий, т. к. это определяет и последовательность выполнения макросов.

13. Если в форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** поле `ном_дог` размещено в области данных и стоит первым в списке последовательности переходов, то при открытии формы и при переходе от записи к записи синхронно отображаются соответствующие записи формы **НАКЛАДНАЯ**. Курсор должен оставаться в поле `ном_дог`. Убедитесь в этом.

Внедренный макрос

Внедренный макрос всегда связывается с событием и сохраняется в форме или отчете. Рассмотрим создание внедренного макроса на примере.

Задача

В базе данных «Поставка товаров» для просмотра, ввода и корректировки данных по договорам создана многотабличная форма **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**. Она состоит из главной формы, построенной на таблицах **ДОГОВОР** и **ПОКУПАТЕЛЬ**, и подчиненной формы, построенной на таблицах **ПОСТАВКА_ПЛАН** и **ТОВАР** и представляющей спецификацию договора.

Пусть необходимо при добавлении или удалении строк спецификации договора, а также при изменении значения поля `кол_пост` (количество поставки), автоматизировать расчет суммы поставки в каждой строке спецификации (поле `сумма_пост`) и общей суммы по договору (поле `сумма_дог`) с сохранением результатов в таблицах **ПОСТАВКА_ПЛАН** и **ДОГОВОР** соответственно.

1. Для расчета и обновления полей `сумма_пост` и `сумма_дог` в таблицах **ПОСТАВКА_ПЛАН** и **ДОГОВОР** при изменении в форме значения поля `кол_пост` (оно размещается в подчиненной форме) создайте следующий внедренный макрос обработки события **После обновления** (After Update) для этого поля (рис. 9.9).
2. Убедитесь, что этот макрос правильно производит расчет также при добавлении новой строки в подчиненную форму.

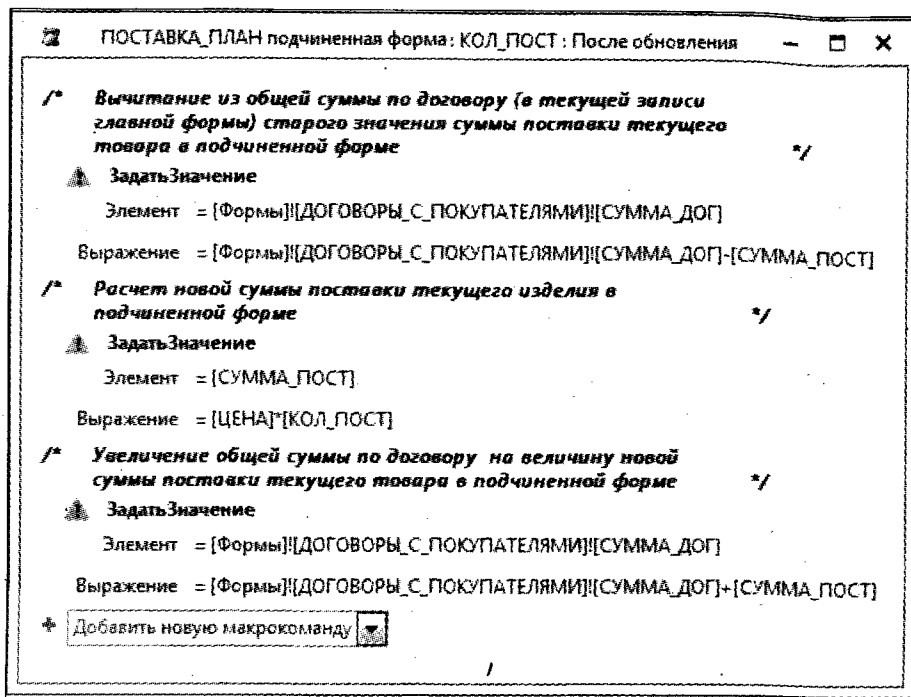


Рис. 9.9. Окно конструирования внедренного макроса

Для расчета и обновления данных в таблице ДОГОВОР при удалении строки из подчиненной формы должен быть создан внедренный макрос обработки события **Удаление (Delete)** для подчиненной формы.

- Для перехода к созданию этого макроса щелкните мышью на подчиненной форме и в контекстном меню откройте ее свойства. На вкладке **События (Event)** в строке **Удаление (Delete)** щелкните на значке **Построитель (Choose Builder)**, выберите пункт **Макросы (Macro Builder)** и затем нажмите кнопку **ОК**. В открывшееся окно макроса введите макрокоманду **ЗадатьЗначение (SetValue)** (рис. 9.10).

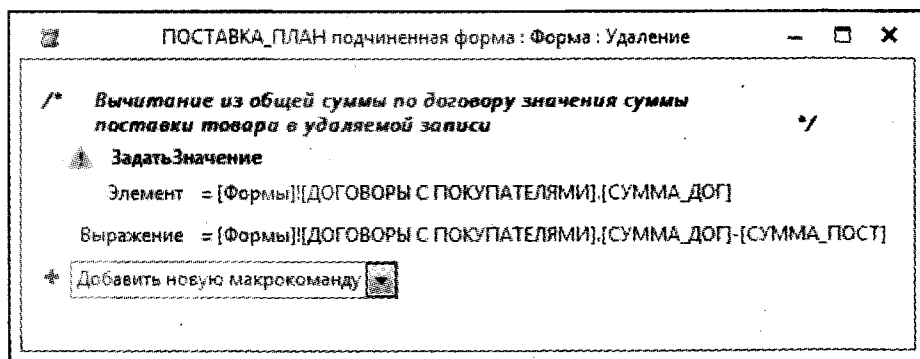



Рис. 9.10. Внедренный макрос обработки события Удаление

Убедитесь в том, что после сохранения внедренного макроса его имя отобразится в окне **Каталога макрокоманд (Action Catalog)** в разделе **В этой базе данных (In this Database)** под именем формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**.

ОБРАТИТЕ ВНИМАНИЕ

Чтобы отобразить полный список макрокоманд, выполнение которых возможно только в базе данных со статусом доверенной, щелкните на кнопке **Показать все действия (Show All Actions)**. Когда доступен полный список макрокоманд, кнопка **Показать все действия (Show All Actions)** выделена. Небезопасные макрокоманды помечены в списке каталога макрокоманд и в теле макроса значком .

Управление последовательностью выполнения макрокоманд

В макросе макрокоманды выполняются в порядке их расположения. Однако для реализации алгоритма в программах часто необходимо нарушить последовательность их выполнения в зависимости от некоторых условий. Условия позволяют определить порядок передачи управления между макрокомандами в макросе и обеспечивают выполнение определенных ветвей алгоритма. Например, в макросе проверяется значение поля в форме на соответствие заданным условиям, и для одних значений может потребоваться вывести сообщение, а для других значений произвести вывод отчета.

Для изменения порядка выполнения макрокоманд может быть использован управляющий блок:

Если условное выражение **то** [макрокоманды_то] **иначе** [макрокоманды_иначе]
Конец блока "Если" (If ... Then ... Else ... End If)

Блок обеспечивает выполнение одной или другой группы макрокоманд в зависимости от значения условного выражения.

Условное выражение является логическим выражением, которое возвращает значение **Истина** (True) или **Ложь** (False). Если условное выражение имеет значение **Истина**, то выполняются [макрокоманды_то], в противном случае — [макрокоманды_иначе].

Для включения в макрос блока **Если** (If) нужно выбрать его из раскрывающегося списка в поле **Добавить новую макрокоманду (Add New Action)** или перетащить в нужное место из каталога макрокоманд. Блок для выполнения набора макрокоманд, если условное выражение истинно, отобразится в макросе в виде, представленном на рис. 9.11. В этом блоке отсутствует блок **Иначе** (Else). Если условное выражение ложно, то набор макрокоманд не выполнится, и управление будет сразу передано на макрокоманду следующую за блоком **Если** (If).

После выполнения **Добавить блок "Иначе"** (Add Else) (см. рис. 9.11) мы получим блок **Если ... то ... иначе ... Конец блока "Если"** (If ... Then ... Else ... End If), который позволит выполнить набор макрокоманд **то**, если условное выражение истинно, и набор макрокоманд **Иначе** в противном случае (рис. 9.12).

На рис. 9.11 и 9.12 видно, что можно Добавить блок "Иначе если" (Add Else If) и тогда в набор макрокоманд **Иначе** будет включен новый блок **Если**, который позволит организовать проверку нового условного выражения и выполнение нового набора макрокоманд. Структура блока в этом случае может быть записана в следующем виде:

Если условное выражение **то** [макрокоманды_то]
иначе **Если** условное выражение **то** [макрокоманды_то] **Конец блока "Если"**

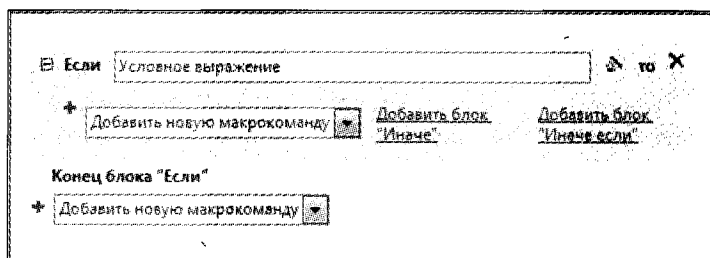


Рис. 9.11. Блок Если ... то ... Конец блока "Если"

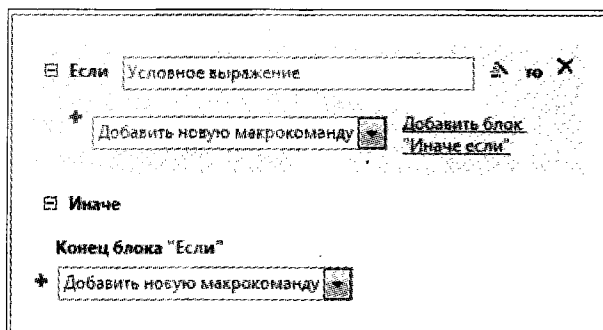


Рис. 9.12. Блок Если ... то ... иначе ... Конец блока "Если"

Таким образом, условие позволяет пропустить или выполнить связанный с первым условием набор макрокоманд и перейти к проверке следующего условия. Добавляя новые логические блоки, можно реализовать самые сложные алгоритмы, связанные с условными переходами к различной логике.

Допускается ввод в поле **Условное выражение** (Conditional expression) значений **Нет** (No), **Да** (Yes) или **Ложь** (False), **Истина** (True). Этот прием может использоваться при отладке макросов, например для того, чтобы пропустить выполнение соответствующей группы макрокоманд.

Рассмотрим создание блока управления последовательностью выполнения макрокоманд в макросе на примере.

Задача

Пусть необходимо проверить наличие на складе некоторого заказанного в договоре товара, и в случае его отсутствия вывести отчет об отгрузках этого товара. В форме **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ**, включающей подчиненную форму с перечнем

заказанных товаров, проверим наличие одного из товаров на складе. В случае отсутствия товара, выведем отчет о фактически произведенных по накладным отгрузкам товара для просматриваемого договора.

В источник записей подчиненной формы *ПОСТАВКА_ПЛАН подчиненная форма*, построенный на таблицах *ПОСТАВКА_ПЛАН* и *ТОВАР*, включите поле *НАЛИЧИЕ_ТОВ*, не отображая его в форме. Подготовьте отчет об отгрузках товара на таблице *ОТГРУЗКА*, которую дополните датой и номером договора из таблицы *НАКЛАДНАЯ*, а для поля *КОЛ_ОТГР* (Количество отгруженного) предусмотрите итоговую величину.

1. Для реализации этой задачи создайте изолированный макрос, представленный на рис. 9.13. Свяжите его выполнение с событием **Получение фокуса** (On Got Focus) для поля *КОД_ТОВ* (код товара) в *ПОСТАВКА_ПЛАН подчиненная форма*.

```

Если [Формы]![ДОГОВОР]![ПОСТАВКА_ПЛАН подчиненная форма]![НАЛИЧИЕ_ТОВ]= Да то
    ОкноСообщения
        Сообщение Товар на складе есть
        Сигнал Да
        Тип Предупреждающее!
        Заголовок
        ОстановитьМакрос
Иначе
    ОкноСообщения
        Сообщение Товара на складе нет!
        Сигнал Да
        Тип Информационное
        Заголовок Товара нет!
    ОткрытьОтчет
        Имя отчета ОТГРУЗКА_ТОВАРА
        Режим Просмотр
        Имя фильтра
        Условие отбора = [НОМ_ДОГ]=[Формы]![ДОГОВОР]![НОМ_ДОГ] And
        [КОД_ТОВ]=[Формы]![ДОГОВОР]![ПОСТАВКА_ПЛАН подчиненная форма]![КОД_ТОВ]
        Режим окна Обычное
    Конец блока "Если"
Если MsgBox("Закреть отчет?",1)=1 то
    ЗакретьОкно
        Тип объекта Отчет
        Имя объекта ОТГРУЗКА_ТОВАРА
        Сохранение Подсказка
    Конец блока "Если"
Добавить новую макроскоманду

```

Рис. 9.13. Пример макроса с блоками Если

2. В макросе в первом блоке **Если ... то ... иначе ... Конец блока "Если"** записано условное выражение для проверки значения в поле `НАЛИЧИЕ_ТОВАР`, которое имеет тип данных **Логический** (Yes/No). В зависимости от результата проверки условия будут выполнены различные ветви программы:
- если логическое выражение возвращает значение **Истина** (True), выполняются макрокоманды **то: ОкноСообщения** (MessageBox) и **ОстановитьМакрос** (StopMacro). Первая выводит сообщение о том, что товар, на который был переведен фокус, есть на складе. Вторая завершает выполнение макроса;
 - при возврате значения **Ложь** (False) выполняются макрокоманды **Иначе: ОкноСообщения** (MessageBox) и **ОткрытьОтчет** (OpenReport). Первая макрокоманда выводит сообщение об отсутствии товара на складе, вторая открывает отчет об отгрузках по открытому договору того товара, на который был переведен фокус.
3. После этого выполняется второй блок **Если ... то ... Конец блока "Если"**, в котором проверяется ответ пользователя на выведенное в диалоговом окне предложение закрыть отчет (рис. 9.14). Вывод диалогового окна обеспечивается функцией `MsgBox('Закреть отчет?';1)`. При утвердительном ответе функция возвращает значение 1, и отчет закрывается. На этом макрос завершает свою работу.

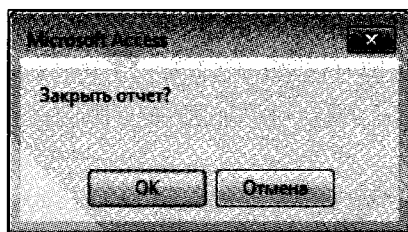


Рис. 9.14. Диалоговое окно функции `MsgBox('Закреть отчет?';1)`

4. В приведенном примере переместите второй блок **Если** на одну позицию выше, щелкнув на соответствующей зеленой стрелке в правой части блока. Он переместится в набор макрокоманд **Иначе**. При этом можно удалить из набора **то** макрокоманду **ОстановитьМакрос** (StopMacro).
5. Убедитесь, что и теперь макрос при перемещении фокуса на новый код товара выполняется правильно.
6. Чтобы сделать структуру программы более наглядной, щелкните на команде **Свернуть макрокоманды** (Collapse Actions). При этом макрос свернет макрокоманды, отобразив их имена и аргументы в строках, а блоки оставит несвернутыми.
7. Убедитесь в том, что после сохранения этого изолированного макроса его имя отобразится в списке макросов области навигации, а также в разделе **В этой базе данных** (In this Database) в окне каталога макрокоманд.

Примеры условных выражений

Рассмотрим примеры условных выражений в макросах.

□ Проверка значений полей в текущей записи формы:

- значения в поле ЦЕНА:

```
ЦЕНА > = 2000
```

- нахождение значения в поле ДАТА_ДОГ (дата заключения договора) в интервале не раньше 2-фев-16 и не позже 2-мар-16:

```
[ДАТА_ДОГ] Between #2-фев-16# And #2-мар-16#
```

- значения в поле СУММА на равенство пустому (Null) значению:

```
IsNull([СУММА])
```

или

```
[СУММА] Is Null
```

□ Проверка одновременного выполнения двух условий на равенство заданным значениям в полях НАИМ_ТОВ и КОЛ_ПОСТ текущей записи:

```
[НАИМ_ТОВ]="Монитор" And [КОЛ_ПОСТ]>15
```

ЗАМЕЧАНИЕ

В условиях могут использоваться функции, например, DCount, DMax, MsgBox.

□ Подсчет числа выписанных накладных и сравнение его с заданным:

```
DCount(Формы![НАКЛАДНАЯ]![НОМ_НАКЛ]; 'НАКЛАДНАЯ')>50
```

или

```
DCount('*', 'НАКЛАДНАЯ')>50
```

□ Выбор максимальной цены товара и сравнение ее с заданным значением:

```
DMax(Формы![ТОВАР]![ЦЕНА]; 'ТОВАР')> 500
```

□ Функция MsgBox может использоваться для вывода на экран диалогового окна, содержащего нужное сообщение, и кнопок ОК и Отмена (Cancel), нажатие которых формирует одно или другое значение функции. Сравнение значения, сформированного функцией MsgBox, с заданным значением позволяет выбрать пользователю вариант действий:

```
MsgBox("Выдать справку?";1)=1
```

Первый параметр функции задает выводимое на экран сообщение, единица на месте второго параметра определяет отображение в диалоговом окне кнопок ОК и Отмена (Cancel). Если пользователь нажимает кнопку ОК, функция возвращает значение 1. Если нажимает кнопку Отмена (Cancel), функция возвращает значение 2.

ЗАМЕЧАНИЕ

В условиях могут использоваться значения элементов управления из объектов, которые не являются текущими в данный момент.

- Сравнение значения поля КОД_ПОК в текущей записи формы ПОКУПАТЕЛЬ со значением аналогичного поля в открытой форме ДОГОВОР:

Forms! [ДОГОВОР]! [КОД_ПОК] = [КОД_ПОК]

Организация выполнения макросов

При запуске любого макроса выполнение начинается с первой макрокоманды и следует по алгоритму, реализуемому макросом. В процессе выполнения макроса в блоках управления **Если** проверяются условия, и в зависимости от результата выполняются те или иные макрокоманды. При этом, следуя алгоритму, макрос выполняется по одному из определенных путей, из множества возможных. Путь выбирается в зависимости от результата проверки условий. Макрокоманда **ЗапускМакроса** (RunMacro) вызывает на выполнение другие макросы. При вызове другого макроса ему передается управление. Вызванный макрос может выполняться несколько раз. После выполнения вызванного макроса управление возвращается к вызывающему макросу и продолжается выполнение его макрокоманд.

Изолированные и вложенные макросы могут запускаться как из области навигации, так и через механизм расширенной обработки событий. Внедренные макросы запускаются только путем установления связи с событием. Access распознает определенные события, к которым может привязываться запуск макроса или процедуры обработки событий на VBA. Событиями, например, являются открытие отчета, изменение данных в поле формы, перевод фокуса на другую запись или поле в форме, щелчок мышью на элементе управления. Для формы определены события формы, раздела формы, записи и элемента управления. Для отчета определены только события отчета и раздела отчета.

Порядок выполнения макросов, связанных с событиями, зависит от порядка возникновения событий и в значительной степени определяется действиями пользователя в формах. Таким образом, управление программой осуществляется пользователем, который выполняет действия, а программа реагирует на них.

Запуск макроса

Для запуска изолированного макроса, открытого в режиме конструктора, надо на ленте нажать кнопку **Выполнить** (Run).

Для запуска изолированного макроса из области навигации надо установить курсор на имя макроса и выполнить на нем двойной щелчок или нажать кнопку **Выполнить** (Run) в его контекстном меню. Можно, не выбирая макроса в области навигации, на ленте **Работа с базами данных** (Database Tools) в группе **Макрос** (Macro) щелкнуть на кнопке **Выполнить макрос** (Run Macro) — откроется диалоговое окно **Запуск макроса** (Run Macro), в котором в поле со списком **Имя макроса** (Macro Name) можно выбрать нужный макрос (рис. 9.15).

В диалоговом окне **Запуск макроса** (Run Macro) в поле со списком **Имя макроса** (Macro Name) отображаются не только имена макросов, представленных в области

навигации, но имена вложенных макросов. В поле **Имя макроса** (Macro Name) имя вложенного макроса записывается так:

ИмяГруппыМакросов. ИмяВложенногоМакроса

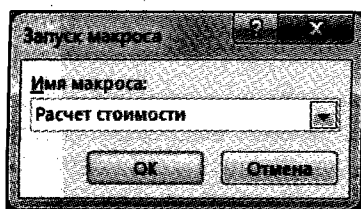


Рис. 9.15. Окно выбора запускаемого макроса

Access позволяет создать макрос, автоматически запускающийся при каждом открытии базы данных. Этот макрос должен иметь имя **AutoExec**. В процессе открытия базы данных Access осуществляет поиск макроса с этим именем и, если такой макрос существует, автоматически запускает его.

В макрос **AutoExec** целесообразно поместить макрокоманды, которые подготовят нужную рабочую среду для пользователя, откроют формы и, если необходимо, другие объекты базы данных, разместят их на экране в удобном виде. Этот макрос создается как любой другой макрос и сохраняется под именем **AutoExec**. При следующем открытии базы данных Access автоматически запустит его.

Если требуется открыть базу данных, не выполняя при этом макрос **AutoExec**, надо открывать ее при нажатой клавише **<Shift>**. Клавиша **<Shift>** позволяет отключать макрос **AutoExec**, если в диалоговом окне **Параметры Access** (Access Options) на вкладке **Текущая база данных** (Current Database) в разделе **Параметры приложений** (Application Options) установлен флажок **Использовать специальные клавиши Access** (Use Access Special Keys).

Выполнение макроса с наступлением события

Наряду с рассмотренными способами запуска макросов, в Access предусмотрена возможность автоматического запуска макроса при возникновении события в форме, отчете или их элементах управления. Для этого необходимо в свойствах этих объектов в строку соответствующего события ввести имя макроса или организовать создание внедренного макроса.

Связь события с программой позволяет организовать взаимодействие различных компонентов приложения. Причем, поскольку события чаще всего возникают в результате действий пользователя, управление приложением осуществляется по его инициативе.

Пусть необходима обработка данных при *добавлении записей* через форму. Если эта обработка реализуется некоторым макросом, то для автоматического запуска этого макроса можно использовать события **До вставки** (Before Insert) и **После вставки** (After Insert), которые являются событиями формы:

- событие **До вставки** (Before Insert) наступает при вводе пользователем первого символа в новую запись, но до фактического добавления записи в базу данных;
- событие **После вставки** (After Insert) наступает после добавления новой записи в базу данных. Пример вызова вложенного макроса, размещенного в области навигации, для обработки события **После вставки** приведен на рис. 9.16.

Пусть необходима обработка данных при *обновлении значения* в поле записи через форму. Для автоматического запуска макроса, выполняющего такую обработку, можно использовать события **До обновления** (Before Update) и **После обновления** (After Update), которые определены как для формы, так и для ее элементов управления:

- событие **До обновления** (Before Update) наступает, когда пользователь приступает к обновлению данных в элементе управления или записи;
- событие **После обновления** (After Update) наступает после обновления данных в элементе управления или записи. Это событие возникает при потере фокуса элементом управления, т. е. при переходе к другому элементу или после сохранения записи. Сохранение записи выполняется при переходе к другой записи или при выполнении пользователем команды **Записи | Сохранить** (Records | Save) на вкладке ленты **Главная** (Home). Событие возникает для новых и существующих записей.

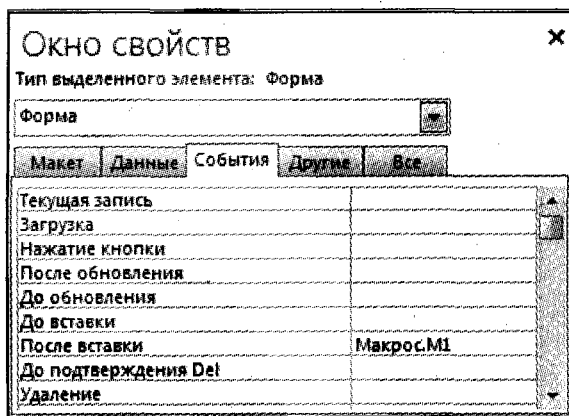


Рис. 9.16. Связь вложенного макроса с событием формы

Например, в подчиненной форме **ПОСТАВКА_ПЛАН** формы **ДОГОВОРЫ С ПОКУПАТЕЛЯМИ** необходимо при обновлении значений в полях **КОЛ_ПОСТ** или **ЦЕНА** рассчитать и записать новое значение в поле **СУММА_ПОСТ = ЦЕНА*КОЛ_ПОСТ**. Пусть в базе данных имеется изолированный макрос **Расчет стоимости**, в котором макрокомандой **ЗадатьЗначение** (SetValue) выполняется требующийся расчет значения поля **СУММА_ПОСТ**. Для выполнения такого макроса при наступлении события **После обновления** (After Update) в свойствах полей **КОЛ_ПОСТ** и **ЦЕНА** на вкладке **События** (Event) в строке **После обновления** выберите имя макроса (рис. 9.17).

Теперь при обновлении значения в поле КОЛ_ПОСТ или ЦЕНА выполнится макрос **Расчет стоимости**, и результат вычисления выражения сохранится в таблице ПОСТАВКА_ПЛАН в поле СУММА_ПОСТ.

Посмотрите, как записывается макрокоманда **ЗадатьЗначение** (SetValue), выполняющая расчет значения поля СУММА_ПОСТ = ЦЕНА*КОЛ_ПОСТ, во внедренном макросе, показанном на рис. 9.9.

В приведенном примере макрос хранится как отдельный объект, представленный в области навигации. При большом числе объектов становится трудно управлять макросами, связь которых с конкретными формами или отчетами не очевидна. Значительно удобнее использовать внедренные макросы, сохраняемые в формах или отчетах.

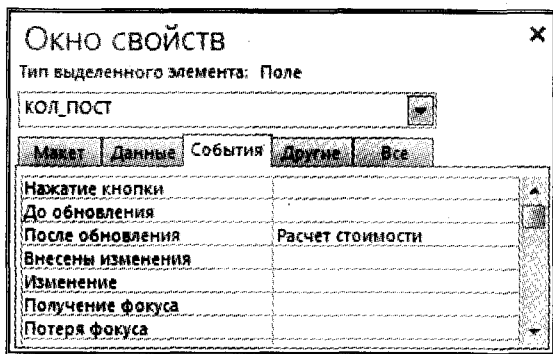


Рис. 9.17. Связь макроса с событием элемента управления

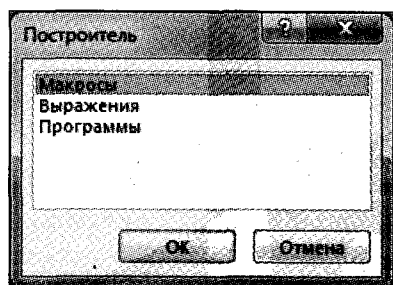


Рис. 9.18. Начало процесса по созданию внедренного макроса

Для внедрения макроса в подчиненную форму ПОСТАВКА_ПЛАН откройте ее в режиме конструктора или макета. Откройте окно свойств для поля КОЛ_ПОСТ и в пустой строке события После обновления (After Update) нажмите кнопку построителя. В окне Построитель (Choose Builder) выберите Макросы (Macro Builder) (рис. 9.18) — откроется окно конструктора макросов, такое же, как для изолированного макроса. Для вычисления значения суммы добавьте новую макрокоманду **ЗадатьЗначение** (SetValue) (рис. 9.19). Если этой макрокоманды нет в списке, проверьте выделена ли кнопка **Показать все действия** (Show All Actions) на ленте конструктора макросов в группе **Показать или скрыть** (Show/Hide).

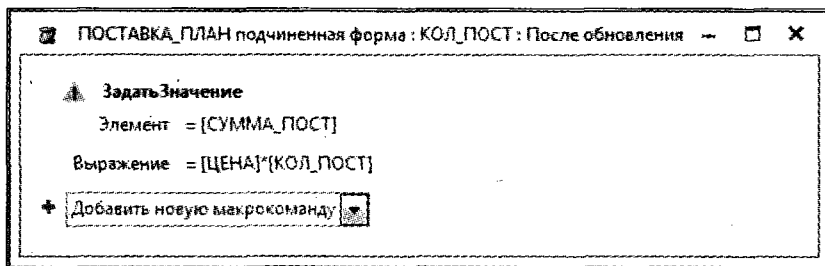


Рис. 9.19. Внедренный макрос, выполняющийся при наступлении в поле КОЛ_ПОСТ события После обновления

В окне свойств поля `КОЛ_ПОСТ` в строке события **После обновления** (After Update) будет записано **[Внедренный макрос]** ([Embedded Macro]) (рис. 9.20).

Для одной и той же формы или отчета могут одновременно использоваться и изолированные, и внедренные макросы.

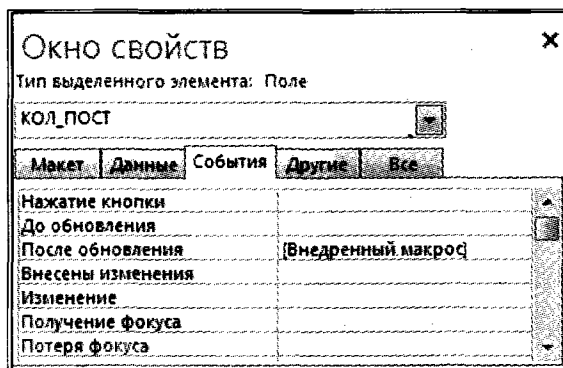


Рис. 9.20. Связь внедренного макроса с событием

Порядок выполнения макросов, вызываемых событиями

Для правильного определения порядка выполнения макросов, которые запускаются при возникновении событий, необходимо понимание того, когда и в какой последовательности возникают события. Каждое отдельное действие, например переход на другой элемент управления, может явиться причиной ряда событий, возникающих в определенной последовательности. Если созданы макросы, которые требуется выполнять в некотором порядке, пользователь должен быть уверен, что события, вызывающие эти макросы, возникают в том же порядке.

Поясним последовательность возникновения событий при закрытии формы и при переходе с одного элемента управления на другой.

Когда пользователь закрывает форму, для текущего элемента управления возникают события **Выход** (On Exit) и **Потеря фокуса** (On Lost Focus), а также события формы, такие как **Выгрузка** (On Unload), **Отключение** (On Deactivate) и **Закрытие** (On Close). Порядок возникновения событий будет следующим:

Выход ⇒ **Потеря фокуса** ⇒ **Выгрузка** ⇒ **Отключение** ⇒ **Закрытие**

Когда пользователь переводит курсор (фокус) на другой элемент управления, для элемента, с которого переводится курсор, сначала возникает событие **Выход** (On Exit), а затем **Потеря фокуса** (On Lost Focus). Для элемента, на который переводится курсор, сначала возникает событие **Вход** (On Enter), а затем **Получение фокуса** (On Got Focus).

Вызов макроса из другого макроса

Если некоторый набор макрокоманд будет использоваться в нескольких макросах, то вместо того, чтобы включать эти команды в каждый макрос, пользователь имеет возможность создать макрос, содержащий эти макрокоманды, и вызывать его из других макросов.

Вызов макроса выполняется макрокомандой **ЗапускМакроса** (RunMacro). Макрокоманда вводится в той позиции макроса, из которой должен вызываться другой макрос.

Аргументы макрокоманды **ЗапускМакроса** (RunMacro):

- **Имя макроса** (Macro Name) — если запускается вложенный макрос, имя задается с указанием имени группы:

ИмяГруппыМакросов.ИмяВложенногоМакроса

- **Число повторов** (Repeat Count) вызова макроса — если этот аргумент и аргумент **Условие повтора** (Repeat Expression) не заданы, вызванный макрос выполняется один раз;
- **Условие повтора** (Repeat Expression) — логическое выражение. При значении **Ложь** (False) повторное выполнение вызванного макроса не производится и возобновляется выполнение вызывающего макроса. Если заданы значения обоих аргументов, **Число повторов** (Repeat Count) и **Условие повтора** (Repeat Expression), вызываемый макрос будет выполняться до тех пор, пока выражение, заданное аргументом **Условие повтора** (Repeat Expression), не получит значение **Ложь** (False), либо пока не исчерпается число указанных повторов.

Макрокоманда **ЗапускМакроса** (RunMacro) с помощью аргументов **Число повторов** (Repeat Count) и **Условие повтора** (Repeat Expression) позволяет реализовать циклы в программах на языке макросов.

При запуске исходного макроса его макрокоманды выполняются поочередно до тех пор, пока не наступит очередь макрокоманды **ЗапускМакроса** (RunMacro). В этот момент происходит вызов второго макроса. После завершения выполнения макрокоманд вызванного макроса выполнение вызывающего макроса возобновляется со следующей макрокоманды.

Допускается выполнение макрокоманды **ЗапускМакроса** (RunMacro) в вызванном макросе. Это означает, что сначала вызывается макрос А, в котором в свою очередь вызывается макрос В, и т. д. По завершении вызванного макроса выполняется следующая макрокоманда вызывающего макроса.

Создание кнопки запуска макроса в форме

Можно предусмотреть запуск изолированного макроса из формы путем нажатия пользователем кнопки. Связь кнопки с макросом определяется в свойствах кнопки на вкладке **События** (Event) в строке **Нажатие кнопки** (On Click).

Создание кнопки запуска макроса с помощью мыши

Для создания кнопки запуска макроса с помощью мыши надо в области навигации выбрать макрос, который необходимо запустить при нажатии кнопки, и перетащить его в форму, открытую в режиме конструктора, — в форму будет помещена кнопка, по нажатию которой станет выполняться этот макрос. Кнопка будет иметь подпись, соответствующую имени макроса.

Если макрос содержит вложенные макросы, при использовании этого метода организуется запуск по кнопке только первого вложенного макроса. Для запуска другого вложенного макроса необходимо в окне свойств кнопки на вкладке **События** (Event) в строке **Нажатие кнопки** (On Click) выбрать вложенный макрос (рис. 9.21).

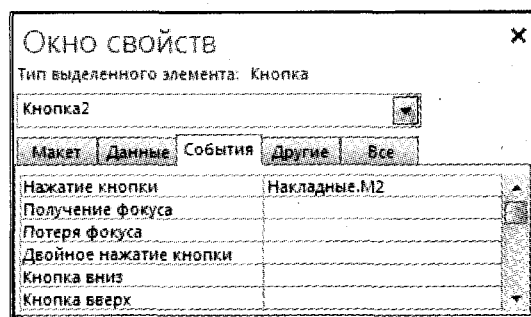


Рис. 9.21. Окно свойств кнопки, вызывающей выполнение макроса

Создание кнопки запуска макроса мастером

Кнопка для запуска макроса может быть создана в форме с помощью мастера. Для этого надо нажать на вкладке ленты конструктора форм в группе **Элементы управления** (Controls) кнопку **Кнопка** (Command Button). После вычерчивания кнопки в форме открывается диалоговое окно мастера **Создание кнопок** (Command Button Wizard), в котором надо выбрать категорию **Разное** (Miscellaneous) и действие **Выполнить макрос** (Run Macro). Далее следует выбрать макрос, который будет запускаться при нажатии кнопки. Мастер позволяет выбрать рисунок или задать текст подписи на кнопке, по умолчанию кнопка получает подпись **Выполнить макрос**.

Мастер создает для кнопки внедренный макрос, что отражается в свойстве **Нажатие кнопки** (On Click) записью **[Внедренный макрос]** ([Embedded Macro]). Созданный мастером внедренный макрос содержит одну макрокоманду запуска изолированного макроса, выбранного в ходе работы мастера.

Очевидно, что для кнопки так же, как для любого другого элемента формы, можно создать любой макрос.

Макросы данных

Макросы данных предназначены для добавления логики к базовым таблицам. Макросы данных связываются с событиями в базовой таблице и выполняются при изменении, вставке или удалении записи. Такие макросы позволяют проверять данные или выполнять вычисления. Привязывая логику к данным и сосредотачивая ее в базовых таблицах, макросы данных обеспечивают доступ к ней из объектов приложения. Так, выполнение операций с записями в формах, построенных на базовых таблицах, инициирует выполнение макросов данных.

Макросы данных, по сути, позволяют реализовать в базе данных функции, которые в серверных базах данных реализуются с помощью особого вида хранимых процедур — *триггеров*. Триггер также связывается с определенной операцией в базовой таблице и вызывается каждый раз при выполнении такой операции.

С помощью макросов данных, как и триггеров, могут выполняться действия по обеспечению реляционной целостности данных, не поддерживаемые стандартными средствами. Макрос данных не имеет ни параметров, ни возвращаемого значения.

При программировании в приложении получить доступ к данным базы можно через формы и отчеты. В макросах данных в отличие от макросов, привязанных к событиям в объектах приложения, для доступа к данным используются ссылки на поля таблиц. Обращение к данным через поля объектов приложения не допустимо.

Список макрокоманд, используемых в макросах данных, существенно короче предлагаемого в обычных макросах. Однако для управления выполнением макрокоманд кроме блока управления **Если** в макросах данных может использоваться блок данных **ДляКаждойЗаписи** (ForEachRecord). Этот блок, реализуя циклы обработки записей, обеспечивает простоту решения многих задач бизнес-логики.

Если ваш компьютер настроен для отправки и приема сообщений электронной почты, в макросе данных можно использовать макрокоманду **ОтправитьПочту** (SendEmail). Если такая настройка на компьютере не выполнена, системой будет выдано сообщение о том, что не удалось отправить сообщение электронной почты.

Чтобы начать создание макроса данных, необходимо открыть таблицу в режиме таблицы. При этом на вкладке ленты **Работа с таблицами | Таблица** (Table Tools | Table) отображаются кнопки создания макросов данных, которые будут выполняться **До изменения** (Before Change), **После обновления** (After Update) записи и т. д. (рис. 9.22).

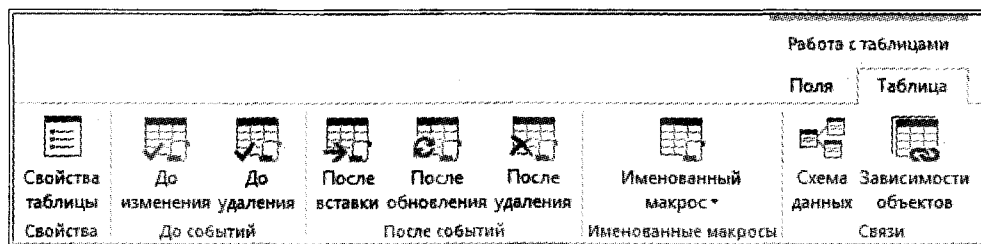


Рис. 9.22. Команды работы с макросами данных на вкладке Таблица

Для каждой таблицы может быть создано по одному макросу данных на каждое из представленных на ленте действий. Щелчком мыши на кнопке открывается окно создания макроса данных. На рис. 9.23 показано окно создания макроса данных таблицы ТОВАР, который будет выполняться После удаления (After Delete) записи.

Список макрокоманд для макроса данных не только короче предлагаемого для макросов базы данных, но и отличается от него, а в разделе каталога макрокоманд **В этой базе данных (In this Database)** отображаются только именованные макросы данных, созданные для таблиц.

Макросы данных внедряются в таблицу. Если для некоторого действия в таблице создан макрос данных, то соответствующая кнопка на ленте конструктора выделяется цветом. Щелчком на такой кнопке открывается макрос данных для просмотра или редактирования в конструкторе.

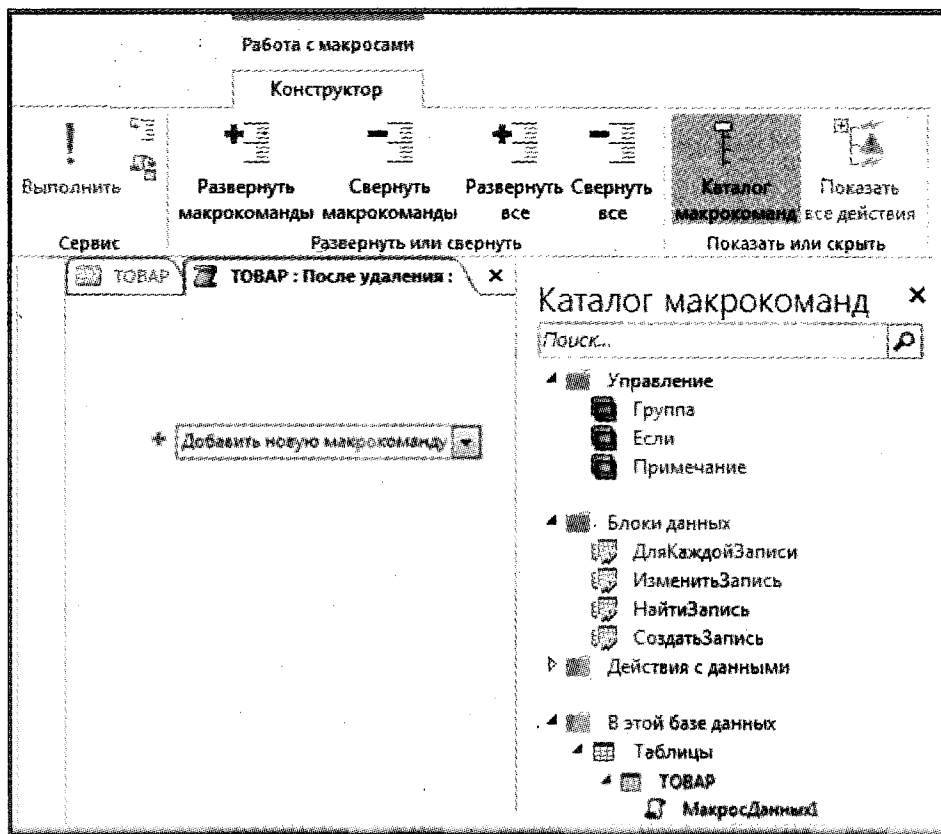


Рис. 9.23. Окно макроса данных с каталогом макрокоманд

Именованные макросы

Кроме макросов данных, связанных с отображенными на вкладке действиями, могут быть созданы именованные макросы данных. Эти макросы данных могут выполняться только при вызове из других макросов данных. В именованном мак-

росе могут определяться параметры. Значения параметрам присваиваются в вызывающем макросе.

Щелчком мыши на кнопке **Именованный макрос (Named Macro)** открывается список команд, показанный на рис. 9.24. Первая команда списка позволяет **Создать именованный макрос (Create Named Macro)**, вторая — **Изменить именованный макрос (Edit Named Macro)**, третья — **Переименовать или удалить макрос (Rename/Delete Macro)**, т. е. просмотреть все таблицы и содержащиеся в них макросы и переименовать или удалить их.

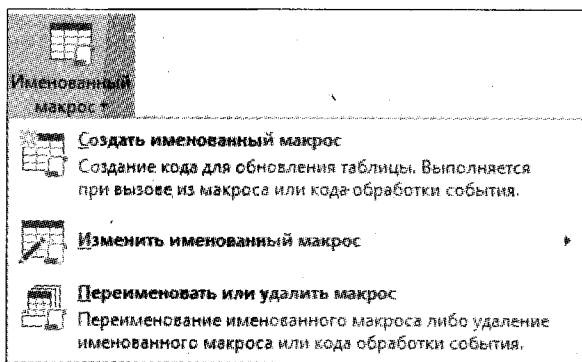


Рис. 9.24. Список команд именованного макроса

Первый именованный макрос по умолчанию имеет имя **МакросДанных1**. При обращении к нему оно дополняется именем таблицы, в которой он был создан, и записывается, например, так: **ТОВАР.МакросДанных1**.

Запуск именованного макроса выполняется макрокомандой **ЗапускМакросаДанных (RunDataMacro)**, доступной как из макроса данных, так и из любых других макросов приложения. Например, работая в форме, можно не выполнять вычисления через обращения к полям формы, а вызвать именованный макрос данных,

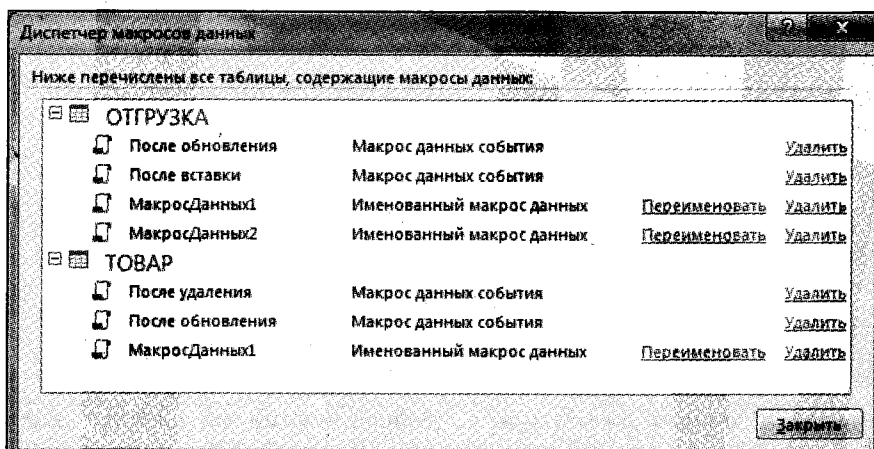


Рис. 9.25. Список таблиц открытой базы, содержащих макросы данных

который выполнит вычисления непосредственно в таблицах. Изменения в таблицах тут же отобразятся в форме.

Командой **Переименовать или удалить макрос** (Rename/Delete Macro) открывается окно диспетчера макросов данных, представленное на рис. 9.25. В нем перечислены все таблицы базы, содержащие макросы данных. Именованные макросы также отображаются в окне диспетчера макросов данных. Переименовать можно только именованные макросы.

Использование макросов данных для решения задач

При разработке практических приложений часто необходимо при изменениях в одних документах решать задачи, позволяющие актуализировать текущие данные в других. Рассмотрим примеры таких задач.

Задача 1

Рассмотрим создание макроса данных для таблицы ТОВАР, в котором после обновления записи проверяется изменение цены товара и выполняется перерасчет стоимости отгруженного товара в таблице ОТГРУЗКА.

1. Откройте таблицу ТОВАР и на вкладке ленты Работа с таблицами | Таблица (Table Tools | Table) щелкните на кнопке После обновления (AfterUpdate). В открывшееся окно макроса данных введите макрокоманды, выбирая их в каталоге и заполняя аргументы, как показано на рис. 9.26.

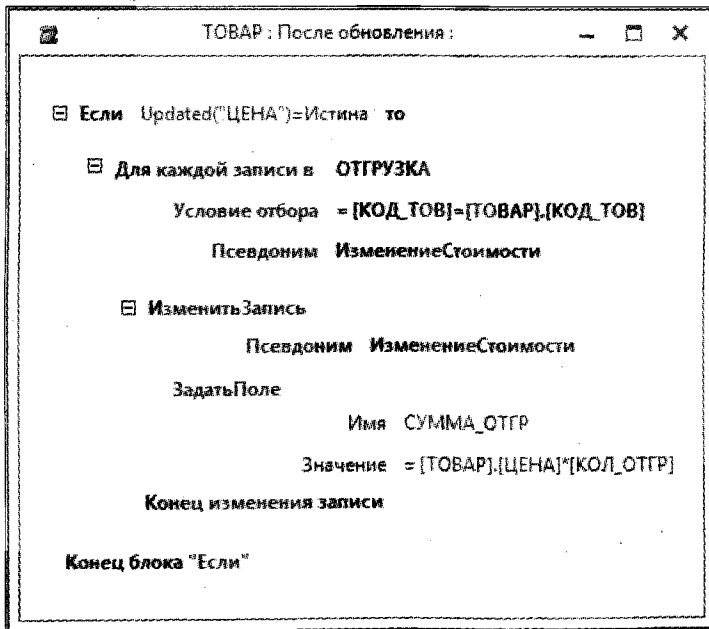


Рис. 9.26. Макрос данных для обновления поля в записях таблицы ОТГРУЗКА

2. В блоке **Если** в логическом выражении условия записано выражение `Updated ("ЦЕНА")= Истина`. Функция `Updated ("имя поля")` проверяет, изменялось ли значение в определенном поле. При истинном значении выражения макрокомандой **Для каждой записи в** (`ForEachRecord`) организуется цикл, в котором для каждой записи таблицы `ОТПРУЗКА`, содержащей значение кода товара, равное значению из обновленной записи таблицы `ТОВАР`, макрокомандой **ИзменитьЗапись** (`EditRecord`) рассчитывается новое значение в поле `СУММА_ОТПР`. Обратите внимание, макрокоманда **ИзменитьЗапись** должна быть вложена в макрокоманду **Для каждой записи в**, иначе не будет найден псевдоним текущей записи цикла. Макрокоманда **ЗадатьПоле** (`SetField`), в свою очередь, должна быть вложена в макрокоманду **ИзменитьЗапись**.

Задача 2

Рассмотрим создание именованного макроса с параметрами для таблицы `ТОВАР`. Макрос должен обеспечить сохранение сведений об обновлениях цены товаров в таблице `ТОВАР` в некоторой таблице, например, `ИзмененныеЗаписи`. С помощью параметров именованный макрос будет получать данные об обновлении цены товара в записи таблицы `ТОВАР`.

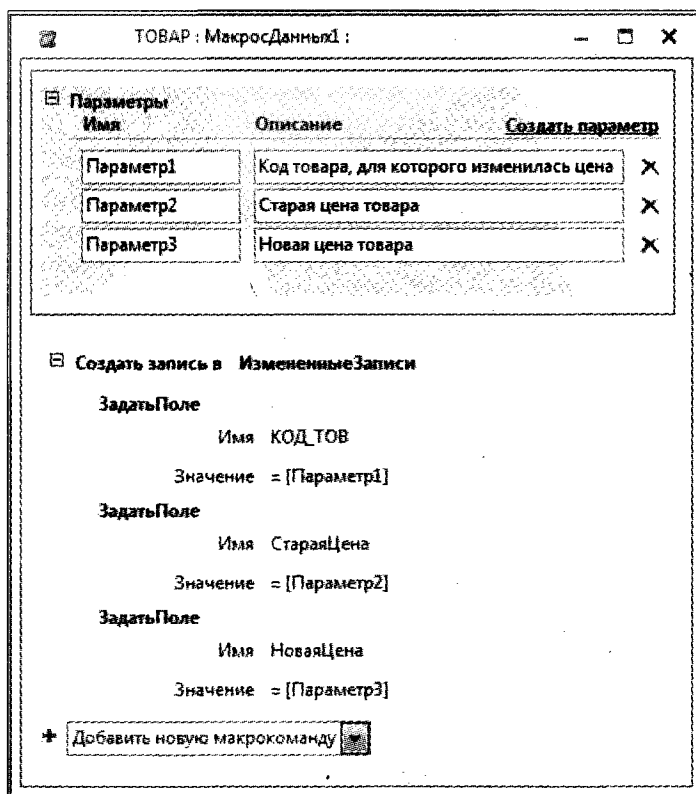


Рис. 9.27. Именованный макрос данных с параметрами

1. Создайте таблицу **ИзмененныеЗаписи**, содержащую поля **Код товара**, **Старая цена**, **Новая цена** и ключ с типом данных **Счетчик (AutoNumber)**. Использование такого ключа позволит включать в таблицу неоднократные изменения записи с одним и тем же кодом товара.
2. При открытой таблице **ТОВАР** создайте именованный макрос, выбрав соответствующую команду на вкладке **Работа с таблицами | Таблица (Table Tools | Table)**.
3. В макросе для получения из вызывающего макроса значений кода товара и его старой и новой цены создайте соответствующие параметры. В верхней части именованного макроса щелкните на значке **+** в строке **Параметры (Parameters)** и на отобразившейся ссылке **Создать параметр (Create Parameter)**. Параметрам дайте имена, как показано на рис. 9.27. Далее выберите макрокоманду **СоздатьЗапись (CreateRecord)**. В макросе она отображается как **Создать запись в (Create a Record In)**. В аргументе укажите имя таблицы, в которой должна создаваться запись: **ИзмененныеЗаписи**. Три макрокоманды **ЗадатьПоле (SetField)** обеспечивают присваивание полям таблицы **ИзмененныеЗаписи** значений, полученных из вызывающего макроса.
4. Закройте макрос. По умолчанию макрос получит имя **ТОВАР.МакросДанных1**.

Задача 3

Организируйте вызов именованного макроса **ТОВАР.МакросДанных1** из ранее созданного для таблицы **ТОВАР** макроса данных, связанного с событием **ПослеОбновления (AfterUpdate)** (см. рис. 9.26). В этом макросе проверяется изменение цены товара в текущей записи таблицы **ТОВАР**. Присвойте параметрам вызываемого макроса значения кода, старой и новой цены товара.

1. Откройте таблицу **ТОВАР**. Щелкните на событии **ПослеОбновления (AfterUpdate)** и дополните ранее созданный для него макрос данных макрокомандой **Запуск-МакросаДанных (RunDataMacro)** (рис. 9.28). Эта макрокоманда размещается в блоке **Если ... то** вслед за макрокомандой **Для каждой записи в (ForEachRecord)**. Укажите имя вызываемого именованного макроса: **ТОВАР.МакросДанных1**.
2. Поскольку в вызываемом макросе определены параметры, после указания его имени автоматически выведутся строки для присваивания параметрам значений. Если список параметров в вызываемом макросе изменялся, щелкните на расположенной справа ссылке **Обновить параметры (UpdateParameters)**. Присвойте параметрам значения из обновленной записи таблицы **ТОВАР**. Для получения значения старой цены используйте функцию **[Old]. [ЦЕНА]**. Закройте макрос, сохранив внесенные изменения.
3. Измените в таблице **ТОВАР** цену одного из товаров и убедитесь, что в таблице **ОТПРУЗКА** во всех записях с кодом товара с обновленной ценой сумма отгрузки пересчиталась, а в таблицу **ИзмененныеЗаписи** внесена новая запись. Обновления в таблицы вносятся вне зависимости от того, открыты они или закрыты. Если таблица открыта, может понадобиться обновить ее отображение, например, закрыв ее и повторно открыв.

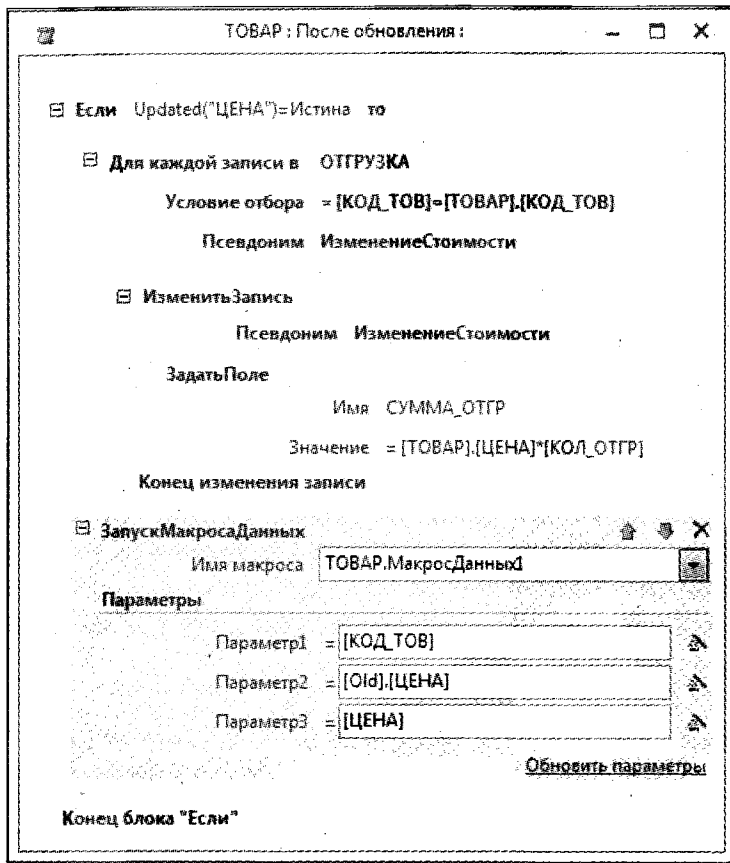


Рис. 9.28. Вызов именованного макроса данных и передача ему параметров

Задание 9.1

Измените макросы, приведенные на рис. 9.27 и 9.28, таким образом, чтобы в таблицу *ИзмененныеЗаписи* помимо старой и новой цены записывалась дата и время изменения цены товара. Для этого дополните таблицу полем *Дата*, именованный макрос новым параметром, а в макросе данных в макрокоманде *ЗапускМакросаДанных* (*RunDataMacro*) присвойте новому параметру значение функции *Now()*.

Задача 4

Пусть необходимо при отгрузке товара по накладной уменьшать его остаток на складе. Предположим, в нашей базе данных склад и товар находятся в отношении 1 : М. Если товар хранится только на одном складе, остаток товара можно разместить в поле таблицы *ТОВАР*. Дополните таблицу *ТОВАР* полем *Остаток*.

1. Откройте таблицу *ОТГРУЗКА* в режиме таблицы и щелчком мыши на кнопке **ПослеВставки** (*AfterInsert*) создайте макрос данных, который будет выполняться при добавлении в таблицу новых записей. В макросе используйте макро-

команду **Найти запись в** (Look Up a Record In) с аргументом **ТОВАР** и для найденной записи — макрокоманду **ИзменитьЗапись** (EditRecord). В рамках последней с помощью макрокоманды **ЗадатьПоле** (SetField) присвойте полю **Остаток** новое значение, уменьшенное на количество отгруженного товара (рис. 9.29).

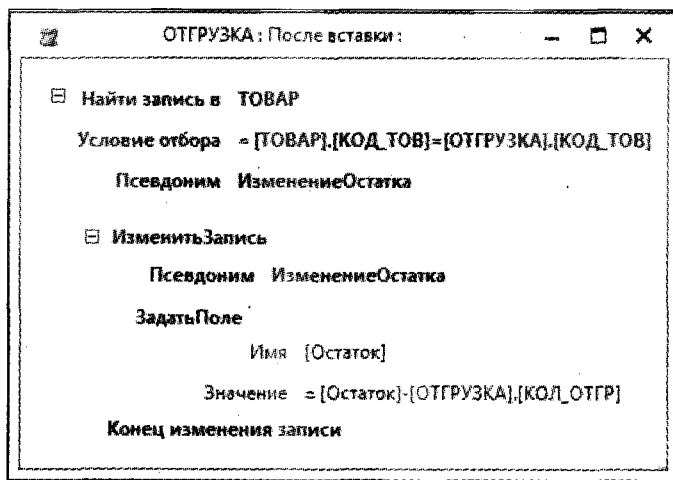


Рис. 9.29. Макрос данных, выполняющийся после вставки записи в таблицу ОТГРУЗКА

2. Убедитесь, что при добавлении новой записи в таблицу ОТГРУЗКА через форму также будет выполняться макрос данных таблицы. Откройте форму Накладная, построенную на таблицах НАКЛАДНАЯ и ОТГРУЗКА. В последней хранятся данные из спецификации накладной. Дополните одну из накладных строкой спецификации, указав, какой товар и в каком количестве отгружается по накладной. Откройте таблицу ТОВАР, и в поле **Остаток** вы увидите обновленное значение.

Задача 5

Пусть необходимо при добавлении новой записи об отгрузке товара в таблицу ОТГРУЗКА пересчитать общее количество товара, отгруженного по договору. Такой расчет позволит отследить выполнение плана отгрузки по каждому товару. Плановое количество каждого товара, которое должно быть отгружено покупателю в соответствии с договором в том или ином месяце, хранится в записях таблицы ПОСТАВКА_ПЛАН. Напомним, что поля **Номер договора**, **Код товара** и **Месяц отгрузки** составляют ключ этой таблицы. Для сохранения количества товара, уже отгруженного по договору в конкретном месяце, предусмотрите в таблице ПОСТАВКА_ПЛАН дополнительное поле **ВыполнениеПлана**.

Разделите решение этой задачи между двумя макросами, связанными с таблицей ОТГРУЗКА: макросом *ОТГРУЗКА : После вставки*, выполняющимся при добавлении записи в таблицу ОТГРУЗКА, и именованным макросом *ОТГРУЗКА : МакросДанных1*, осуществляющим добавление отгруженного количества к значе-

нию поля `ВыполнениеПлана` в соответствующей записи таблицы `ПОСТАВКА_ПЛАН` при его вызове из первого макроса.

1. Сначала создайте именованный макрос, в котором определите четыре параметра. Три необходимы для получения из вызывающего макроса значений для поиска обновляемой записи — это поля: Номер договора, Код товара и Месяц отгрузки, и один: Количество отгруженного — для увеличения количества в поле `ВыполнениеПлана` найденной записи. В этом макросе будет выполняться поиск обновляемой записи в таблице `ПОСТАВКА_ПЛАН` и присваиваться новое численное значение полю `ВыполнениеПлана`.

На рис. 9.30 представлен именованный макрос с параметрами, по умолчанию получивший имя *ОТГРУЗКА : МакросДанных1*. Макрокоманда **Найти запись в** (Look Up a Record In) таблице `ПОСТАВКА_ПЛАН` возвращает запись, соответствующую заданному условию отбора:

```
([ПОСТАВКА_ПЛАН].[НОМ_ДОГ]=[Параметр4] And
[ПОСТАВКА_ПЛАН].[СРОК_ПОСТ]=[Параметр3] And
[ПОСТАВКА_ПЛАН].[КОД_ТОВ]=[Параметр1])
```

Макрокоманда **ИзменитьЗапись** (EditRecord), входящая в блок макрокоманды **Найти запись в** (Look Up a Record In), ссылаясь на псевдоним найденной запи-

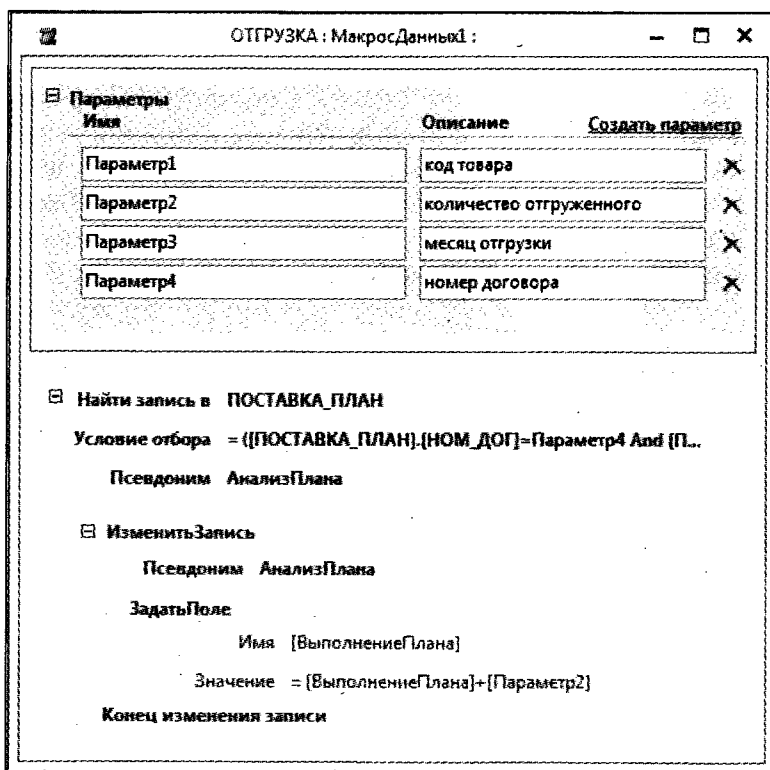


Рис. 9.30. Именованный макрос поиска и обновления записи в таблице `ПОСТАВКА_ПЛАН`

си, выполняет макрокоманду **ЗадатьПоле (SetField)**, которая прибавляет указанное в записи об отгрузке количество к имеющемуся в поле **ВыполнениеПлана**.

ЗАМЕЧАНИЕ

Чтобы значение выражения могло быть вычислено при первой отгрузке товара, при создании записи в таблице **ПОСТАВКА_ПЛАН** полю **ВыполнениеПлана** по умолчанию должно присваиваться нулевое значение.

2. Затем создайте макрос **ОТГРУЗКА** : После вставки, который должен выполняться при наступлении события **ПослеВставки (AfterInsert)** записи в таблице **ОТГРУЗКА** — вызывать именованный макрос и подготавливать значения его параметров.

Этот макрос, так же как при расчете остатка товара на складе (см. задачу 4), должен выполняться при наступлении в таблице **ОТГРУЗКА** события **ПослеВставки (AfterInsert)**.

3. При открытой таблице **ОТГРУЗКА** в режиме таблицы щелчком мыши на кнопке **ПослеВставки (AfterInsert)** откройте существующий макрос данных (см. рис. 9.29). Дополните макрос макрокомандами, которые позволят сформировать значения необходимых в именованном макросе параметров, и макрокомандой вызова именованного макроса (рис. 9.31). Код товара и количество отгруженного выбираются из вставленной новой записи таблицы **ОТГРУЗКА**, поэтому достаточно присвоить их значения соответствующим параметрам. Два недостающих для формирования значений параметров поля: **Номер договора** и **Месяц отгрузки** надо выбрать из таблицы **НАКЛАДНАЯ**.
4. Чтобы найти запись в накладной, нужно задать значения ее ключевых полей: **Номер накладной** и **Код склада**. Значения этих полей имеются в добавленной в таблицу **ОТГРУЗКА** записи. Однако использование имен этих полей в логическом выражении условия отбора с оператором **And** не допустимо. Поэтому определите локальные переменные (в примере **ЛокП1** и **ЛокП2**) и присвойте им значения из соответствующих полей записи таблицы **ОТГРУЗКА**. После этого сформируйте логическое выражение, как показано на рис. 9.31, и заключите его в круглые скобки. Аргумент **ПсевдонимНакладной** макрокоманды **Найти запись в (Look Up a Record In)** таблице **НАКЛАДНАЯ** можно опустить, т. к. далее он в рассматриваемой программе не используется.
5. Чтобы запомнить значения полей из найденной в таблице **НАКЛАДНАЯ** записи, определите в рамках макрокоманды **Найти запись в** локальные переменные. Для выделения из даты отгрузки номера месяца, необходимого при поиске записи в таблице **ПОСТАВКА_ПЛАН**, используйте в выражении, присваиваемом переменной, функцию **Month**.
6. После того как значения всех параметров подготовлены и доступны, вызовите именованный макрос **ОТГРУЗКА.МакросДанных1**. Для вывода строк с параметрами и присвоения им значений щелкайте по надписи **Обновить параметры (UpdateParameters)**.

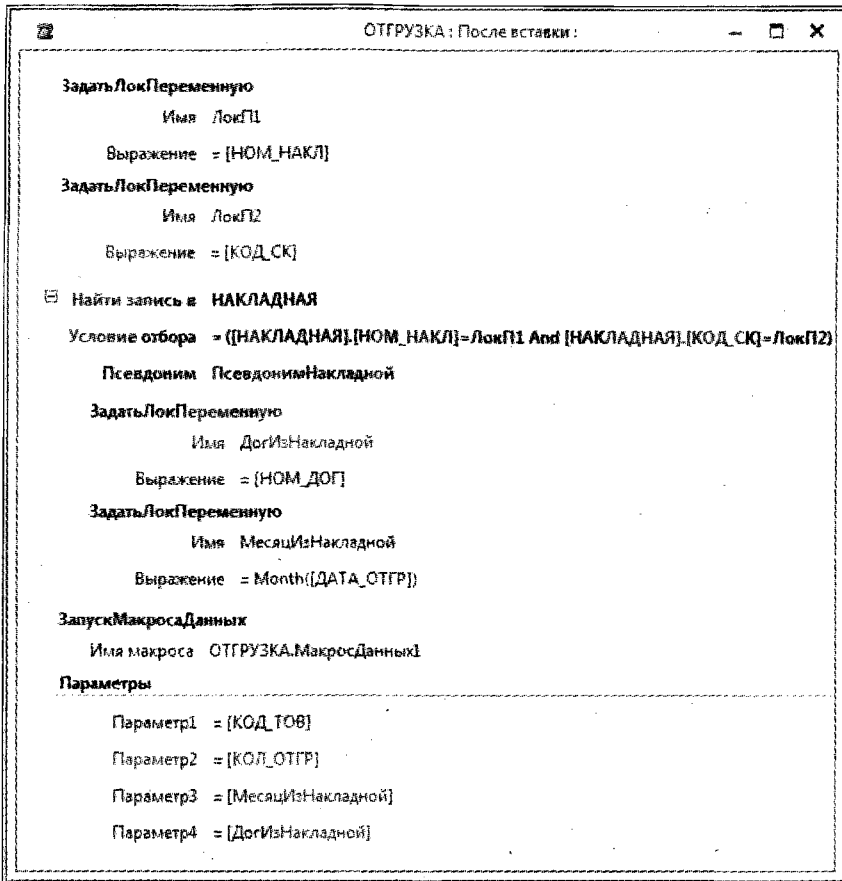


Рис. 9.31. Вызов именованного макроса с параметрами

ЗАМЕЧАНИЕ

Если в логическом выражении условия отбора использованы логические операторы And, Or или другие, то вместо имен полей нужно использовать локальные переменные и заключать такое выражение в круглые скобки.

Задание 9.2

Решите задачу подсчета количества товара, уже отгруженного по договору, в поле `ВыполнениеПлана` таблицы `ПОСТАВКА_ПЛАН`, не используя именованного макроса. Выполните поиск обновляемой записи и увеличение значения в поле `ВыполнениеПлана` непосредственно в макросе *ОТГРУЗКА : После вставки*.

Регистрация событий

Возникающие при выполнении макросов ошибки заносятся системой в журнал регистрации — системную таблицу с именем `USysApplicationLog`. При этом в строке состояния отображается надпись **Новые ошибки приложения**. Щелкните на ней,

чтобы просмотреть эти ошибки. Обнаруженные системой ошибки отмечены в столбце **Category** значением **Execution**.

Используя макрокоманду **РегистрацияСобытия (LogEvent)**, пользователь может включать в журнал свои замечания. Они отмечаются в столбце категории значением **User**.

Если в параметрах навигации установить значок **Показать системные объекты (Show System Objects)**, то таблица журнала регистрации будет отображаться в области навигации наряду с пользовательскими таблицами. Щелкните правой кнопкой мыши в нижней части области навигации в ее свободном пространстве и в контекстном меню найдите команду **Параметры навигации (Navigation Options)**. В окне **Параметры навигации** устанавливаются **Параметры отображения (Display Options)**, в том числе **Показать системные объекты**.

При отсутствии ошибок, обнаруживаемых системой, просмотреть пользовательские сообщения в журнале регистрации можно, открыв соответствующую таблицу в области навигации.

Задание 9.3

Измените рассмотренный ранее макрос данных **ПослеВставки (AfterInsert)** для таблицы **ОТГРУЗКА** так, чтобы в поле **ВыполнениеПлана** отображалось недопоставленное количество товара.

Задание 9.4

Создайте макрос данных, который будет выполняться при изменении цены в таблице **ТОВАР** и изменять стоимость в таблице **ПОСТАВКА_ПЛАН** для договоров, заключенных после заданной даты, для которых еще не было отгрузок.

Контрольные вопросы

1. Чем определяется последовательность выполнения макрокоманд?
2. Какая макрокоманда позволяет изменить текущую запись в объекте?
3. Должен ли объект, в котором меняется текущая запись, быть текущим?
4. Сделается ли объект текущим после выполнения макрокоманды, изменяющей текущую запись в нем?
5. Какая макрокоманда позволяет изменить значение поля записи?
6. Какой объект будет текущим в макросе, если он не установлен явно?
7. Можно ли в логическом выражении условия использовать функцию?
8. Как записывается ссылка на элемент управления в форме?
9. В ссылке на свойство каким знаком отделяется имя свойства от имени элемента управления?

10. Как организовать последовательное выполнение нескольких запросов действия?
11. Какие аргументы имеет макрокоманда **ОткрытьЗапрос** (OpenQuery)?
12. Какое значение должен иметь аргумент **Режим** (View), чтобы макрокоманда **ОткрытьЗапрос** (OpenQuery) приводила к выполнению запроса?
13. Какая макрокоманда позволяет из макроса вывести на экран сообщение?
14. Какая команда позволяет начать выполнение макроса из области навигации?
15. Как просмотреть все внедренные в формы макросы?
16. Как просмотреть, какие именованные макросы имеются в базе данных?
17. Как вставить содержимое одного макроса в другой?
18. Как организовать выполнение макроса при открытии формы?
19. Какая макрокоманда позволяет создать в макросе группу макросов, каждый из которых доступен для выполнения?
20. Как записывается ссылка на макрос, который включен в группу?
21. Какая макрокоманда позволяет выполнить фильтрацию записей?
22. К какому объекту применяется команда фильтрации записей?
23. Какая макрокоманда позволяет снять действие фильтра?
24. В каком случае в выражении макрокоманды **ЗадатьЗначение** (SetValue) при записи ссылки на поле формы должен использоваться полный синтаксис?
25. Как записывается ссылка на поле КОЛ_ОТПР формы Отгрузки покупателям, которая не является текущей?
26. Может ли одно действие пользователя приводить к возникновению нескольких событий?
27. Отображается ли внедренный макрос среди объектов области навигации?
28. С какими объектами работает макрос данных?
29. При каких действиях в таблице выполняются макросы данных?
30. Можно ли для таблицы создать несколько макросов данных **ПослеВставки** (AfterInsert)?
31. Можно ли именованный макрос данных вызвать из внедренного макроса формы?
32. Какая макрокоманда позволяет организовать цикл в макросе данных?

Ответы

1. Порядком размещения макрокоманд в программе и условиями, заданными в блоке управления последовательностью выполнения макрокоманд Если (If).
2. **НаЗапись** (GoToRecord).
3. Нет.

4. Нет.
5. **ЗадатьЗначение** (SetValue).
6. Объект, из которого вызывается макрос.
7. Да.
8. Forms! [Имя_формы]! [Имя_элемента_управления].
9. Точкой.
10. Записать в макросе соответствующее число макрокоманд **ОткрытьЗапрос** (OpenQuery).
11. **Имя запроса** (Query Name), **Режим** (View) и **Режим данных** (Data Mode).
12. **Таблица** (Datasheet).
13. **ОкноСообщения** (MsgBox).
14. **Выполнить** (Run) в контекстном меню или на вкладке **Работа с базами данных** (Database Tools).
15. В каталоге макрокоманд в блоке **В этой базе данных/Формы** (In this Database|Forms).
16. В каталоге макрокоманд в блоке **В этой базе данных/Таблицы** (In this Database|Tables).
17. Открыть принимающий макрос и, выбрав вставляемый в каталоге макрокоманд в блоке **В этой базе данных** (In this Database), выполнить двойной щелчок.
18. Записать в строку свойства события формы **Открытие** (On Open) имя макроса или создать внедренный макрос.
19. **Вложенный макрос** (Submacro).
20. *ИмяГруппыМакросов.имяВложенногоМакроса.*
21. **ПрименитьФильтр** (ApplyFilter).
22. К текущему объекту.
23. **ПоказатьВсеЗаписи** (ShowAllRecords).
24. Если форма не является текущей.
25. [Формы]! [Отгрузки покупателям]! [КОЛ_ОТГР]
26. Да.
27. Нет.
28. С таблицами.
29. При указанных на вкладке **Работа с таблицами | Таблица** (In this Database | Tables).
30. Нет.
31. Да.
32. **ДляКаждойЗаписи** (ForEachRecord).

ГЛАВА 10



Разработка интерфейса для управления приложением

Для представления пользователю задач и возможностей приложения в целом необходим понятный, удобный и дружелюбный интерфейс. Интерфейс объединяет в себе все элементы и компоненты приложения, делая простым доступ пользователя к его функционалу. В Access разработка пользовательского интерфейса для управления приложением баз данных обеспечивается инструментальными средствами форм. Основанный на формах интерфейс интуитивно понятен и не требует объяснений, как им пользоваться.

По своей структуре формы интерфейса, как правило, сложнее, чем формы для представления отдельного электронного документа. Они могут включать разнообразные компоненты: поля со списком, ссылки на нужные пользователю документы, диаграммы, отражающие контролируемые показатели деятельности предприятия, вкладки с документами и т. п. Для разработки интерфейса приложения удобно использовать формы навигации, позволяющие на многоуровневых вкладках размещать формы и отчеты.

Интерфейс приложения может включать в качестве основного компонента какой-либо электронный документ предметной области, при работе с которым одновременно решается ряд смежных задач. При этом могут открываться другие документы, заполняться новые, выполняться процедуры проверки допустимости вводимых значений и обновления расчетных величин, хранимых в базе. Примерами таких задач в рассматриваемой базе данных могут быть проверка при выписке накладной на отгрузку товара степени выполнения договорных обязательств, проверка допустимости отгрузки с учетом наличия товара на складе и многое другое.

Разнообразные средства разработки форм позволяют создавать удобный интерфейс управления приложением, предоставляющий пользователю все необходимое для выполнения его функциональных обязанностей. Формы обеспечивают возможность включать в интерфейс элементы управления для открытия разных документов, производить необходимые запросы, анализ текущего состояния тех или иных объектов предметной области. Например, при решении вопроса о возможности отгрузки товара проверять запасы на складе и оценивать допустимость отгрузки с точки зрения требований к нормативным запасам, при формировании накладной на отгрузку использовать данные из соответствующих договоров. Таким образом,

в пределах одного интерфейса могут быть выполнены разнообразные проверки, автоматически не обеспечиваемые структурой базы данных. Обычно аналогичные интерфейсы предлагаются в рамках проблемно ориентированных прикладных систем.

Рассмотрим некоторые инструменты, которые могут быть использованы при разработке интерфейса управления приложением баз данных с помощью форм.

Диалоговое окно входа в приложение

Создадим пользовательское диалоговое окно, которое будет отображаться при открытии приложения. В окне обеспечим ввод имени исполнителя, приступающего к работе, вход в основную форму управления приложением и сохранение кода исполнителя с помощью встроенного макроса. Сохраненный код позволит при открытии других объектов выбирать из них сведения, касающиеся только этого исполнителя. Вид такого диалогового окна приведен на рис. 10.1.

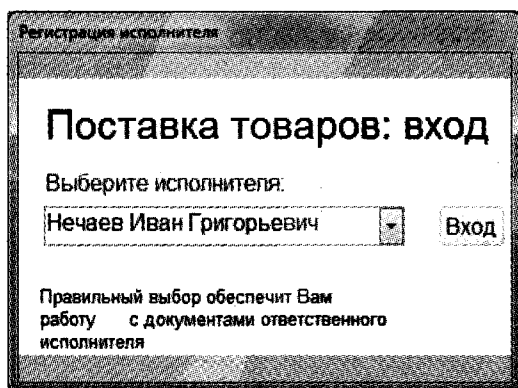


Рис. 10.1. Пользовательское диалоговое окно входа в приложение

Поскольку диалоговое окно, которое нужно разработать, требует ввода исполнителя работ, будем исходить из того, что в рассматриваемой предметной области для выполнения всех работ по ведению договоров решено назначать ответственного исполнителя. Это решение потребует дополнения базы данных таблицей ИСПОЛНИТЕЛЬ и внесения нового поля в существующую таблицу ДОГОВОР для связи ее с исполнителем. Реляционные базы данных позволяют выполнить эти изменения в структуре в любой момент без какого-либо ущерба для ранее разработанных объектов приложения. Для автоматической идентификации исполнителя определим ключевое поле Код исполнителя с типом данных Счетчик (AutoNumber) (рис. 10.2). Таблицу ДОГОВОР дополним полем связи Код исполнителя с типом данных Числовой (Number) и размером поля Длинное целое (Long Integer).

1. Для получения пользовательского диалогового окна нужно создать модальную всплывающую форму. Сначала создайте пустую форму, воспользовавшись соответствующей командой на ленте Создание (Create) в группе Формы (Forms). Пустая форма не имеет источника записей и одновременно с ее открытием в ре-

- жиме макета открывается список полей, отображающих все таблицы базы данных. Они доступны для использования в этой форме и позволяют создать ее источник записей. В нашем случае список полей можно закрыть.
- Поскольку разрабатываемая форма предназначена для запоминания идентификатора пользователя, приступающего к работе с приложением, создайте в ней поле со списком. Для этого выберите на ленте **Работа с макетами форм | Конструктор (Form Layout Tools | Design)** элемент управления **Поле со списком (Combo Box)** и поместите его в области данных — единственном доступном разделе пустой формы. С помощью мастера определите в качестве источника строк поля со списком таблицу **ИСПОЛНИТЕЛЬ** и выберите поля: Код исполнителя, Фамилия, Имя, Отчество и Должность.
 - Для созданного поля со списком мастер построит запрос на выборку заданных полей из таблицы **ИСПОЛНИТЕЛЬ**, который можно просмотреть в строке свойства **Источник строк (Row Source)** на вкладке **Данные (Data)**, как в виде инструкции **SELECT**, так и в режиме конструктора запросов. Поскольку в поле со списком желательно отображать фамилию, имя и отчество исполнителя в едином поле, объедините эти поля в вычисляемом поле. Для этого в режиме конструктора запросов создайте вычисляемое поле, записав выражение `[Фамилия] & " " & [Имя] & " " & [Отчество]`. Поля с фамилией, именем и отчеством удалите. Закройте конструктор запросов, сохранив сделанные в нем изменения.

ИСПОЛНИТЕЛЬ

Имя поля	Тип данных	Описание (необязательно)
Код исполнителя	Счетчик	
Фамилия	Короткий текст	
Имя	Короткий текст	
Отчество	Короткий текст	
Должность	Короткий текст	
Адрес электронной почты	Короткий текст	
Рабочий телефон	Короткий текст	
Домашний телефон	Короткий текст	
Мобильный телефон	Короткий текст	
Адрес	Короткий текст	
Веб-страница	Гиперссылка	
Характеристика	Длинный текст	
Вложения	Вложение	

Свойства поля

Общие		Подстановка	
Размер поля	Длинное целое		
Новые значения	Последователь		
Формат поля			
Подпись			
Индексированное поле	Да (Совпадения)		
Выравнивание текста	Общее		

Имя поля может содержать не более 64 знаков (включая пробелы). Для получения справки по именам полей нажмите клавишу F1.

Рис. 10.2. Структура таблицы ИСПОЛНИТЕЛЬ

4. В окне свойств поля со списком на вкладке **Макет (Format)** измените **Число столбцов (Column Count)** на 3 (код исполнителя, вычисляемое поле и должность). В свойстве **Ширина столбцов (Column Widths)** уберите значения для несуществующих столбцов и увеличьте ширину второго столбца списка, чтобы в нем размещались значения вычисляемого поля.
5. Настройте ширину самого поля со списком. Измените надпись поля со списком на **Выберите исполнителя**.
6. Для того чтобы в поле со списком по умолчанию выводилось значение первого исполнителя в таблице, в строку свойства **Значение по умолчанию (Default Value)** на вкладке **Данные (Data)** введите выражение: `=DFirst("[Код исполнителя]"; "[ИСПОЛНИТЕЛЬ]")`.
7. Сохраните форму под именем **Регистрация исполнителя**. Перейдите в режим формы и убедитесь, что в поле со списком отображается фамилия, имя и отчество в одном поле, а должность — в другом.
8. Далее создайте в форме кнопку с именем **Вход**, которая обеспечит сохранение во временной переменной идентификатора исполнителя, выбранного в поле со списком, и переход к форме управления приложением. Для этого перейдите в режим конструктора, создайте кнопку и с событием **Нажатие кнопки (On Click)** свяжите внедренный макрос.
9. Введите во внедренный макрос, связанный с нажатием кнопки, макрокоманды, как показано на рис. 10.3.

Определение в макросе временной переменной, сохраняющей значение идентификатора текущего исполнителя, позволит использовать ее в других макросах приложения для выбора документов, назначенных для исполнения этому сотруднику. Макрокоманда **Открыть Форму (Open Form)** обеспечивает переход к работе в форме **Управление приложением**, если в форме **Регистрация исполнителя** выбран исполнитель.

ЗАМЕЧАНИЕ

Создание формы **Управление приложением** будет рассмотрено далее.

10. Используйте свойства формы для приведения ее к виду, показанному на рис. 10.1.
11. Для того чтобы форма стала пользовательским диалоговым окном, задайте на вкладке **Другие (Other)** свойствам **Модальное окно (Modal)** и **Всплывающее окно (Pop Up)** значение **Да (Yes)**, а на вкладке **Макет (Format)** свойству **Тип границы (Border Style)** — **Окна диалога (Dialog)**.

При открытии формы с этими свойствами делается недоступной работа со всеми объектами базы данных и отключается доступ ко всем инструментам на лентах, пока не будет закрыта форма. Окно такой формы не является вкладкой, как другие окна. Таким образом, в нашем примере удастся потребовать от пользователя выбора исполнителя, необходимого для продолжения работы в вызываемом окне управления приложением.

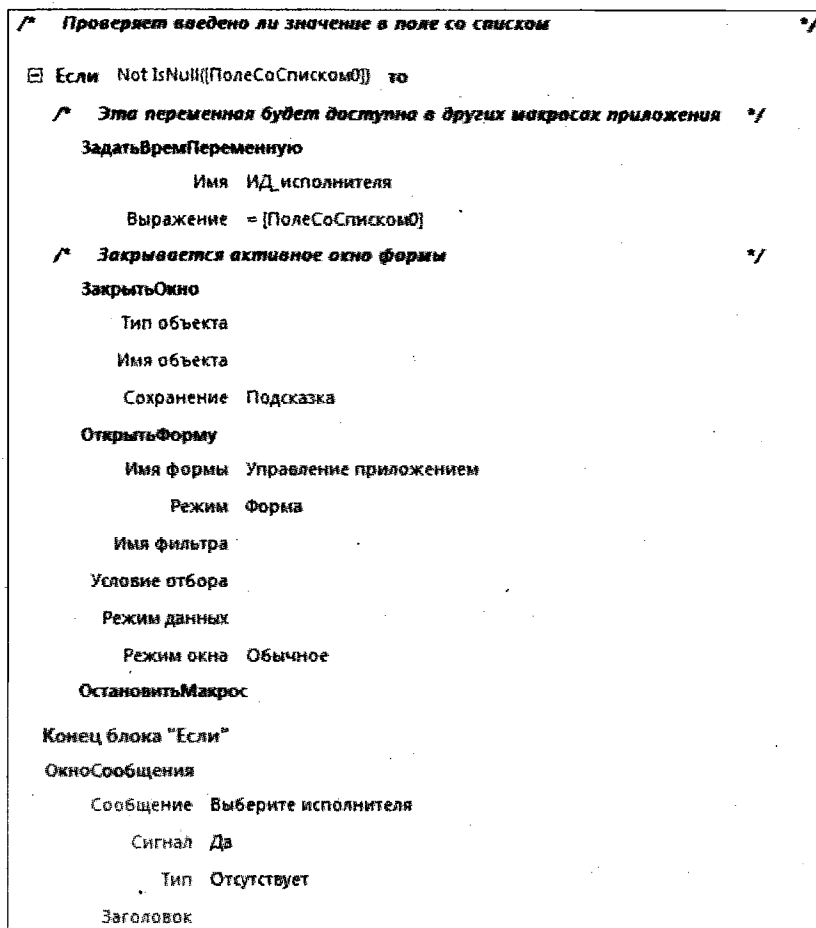


Рис. 10.3. Внедренный макрос, обеспечивающий вход в форму Управление приложением

ЗАМЕЧАНИЕ

В модальной форме нельзя переключаться из режима формы в другие режимы, однако из контекстного меню или после закрытия формы из области навигации можно открыть форму в режиме конструктора или макета. Форма в режиме конструктора или режиме макета не является модальной.

12. Для того чтобы форма отображалась сразу после открытия базы данных, создайте автоматически запускающийся при этом макрос с именем **Autoexec** и включите в него макрокоманду **ОткрытьФорму** (Open Form), открывающую форму **Регистрация исполнителя** (рис. 10.5).
13. Если база данных не отнесена к разряду надежных, то при ее открытии будет выведена панель сообщений, предлагающая включить потенциально опасное содержимое (см. главу 1). Чтобы привлечь внимание пользователя на ненадежность базы, включите в макрос **Autoexec** макрокоманду **Если** (If), в условном выражении которой выполните проверку текущей базы данных на надежность: `Not [CurrentProject].[IsTrusted]`. В случае ненадежности базы данных выведи-

те окно формы **Заставка** (рис. 10.4) с сообщением о необходимости включить содержимое и только при ее надежности обеспечьте выполнение макркоманды **ОткрытьФорму** (Open Form).

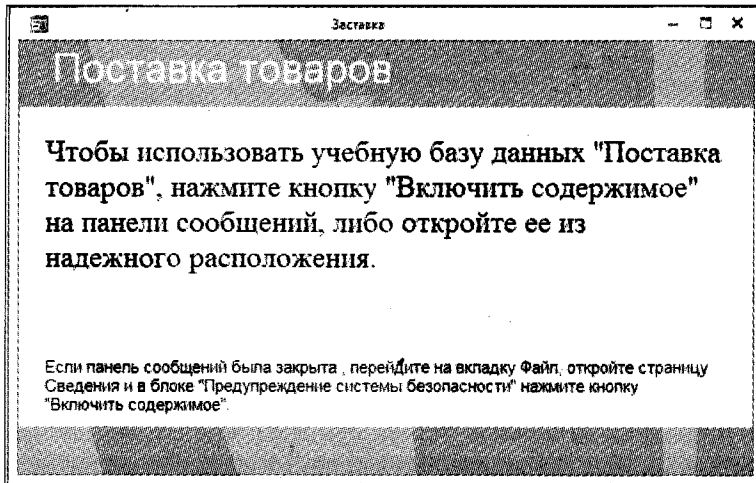


Рис. 10.4. Окно, предлагающее включить базу данных в список надежных

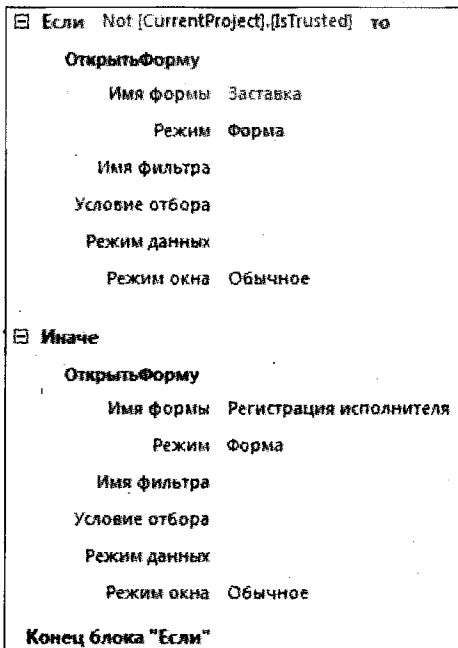


Рис. 10.5. Макрос Autoexec с проверкой надежности базы данных

Нажатием кнопки **Включить содержимое** (Enable Content) вы присваиваете базе данных статус надежной. При этом повторяется проверка надежности базы данных и открывается форма **Управление приложением**. При следующем открытии базы данных из того же места панель сообщений выводиться не будет.

ЗАМЕЧАНИЕ

Если при открытии базы данных нет необходимости в выполнении каких-либо дополнительных действий, для автоматического открытия некоторой формы можно не создавать макрос **Autoexec**, а просто указать имя формы в параметрах **Access**. Для этого откройте вкладку **Файл (File)** и выберите пункт **Параметры (Options)**. Далее выберите пункт **Текущая база данных (Current Database)** и в группе **Параметры приложений (Application Options)** в списке поля **Форма просмотра (Display Form)** выберите имя формы, которая должна отображаться при открытии базы данных.

Главная форма управления приложением

Создадим главную форму управления приложением, в которой будут представлены:

- данные о невыполненных договорных обязательствах, за которые отвечает приступивший к работе исполнитель;
- данные о запасе товаров на складе;
- ссылки на наиболее часто используемые документы и отчеты;
- диаграмма стоимости товаров, не отгруженных по договорам.

Кроме того, предоставим возможность выбирать другого исполнителя и, соответственно, отображать связанные с ним данные.

Такая форма управления приложением в окончателном виде приведена на рис. 10.6.

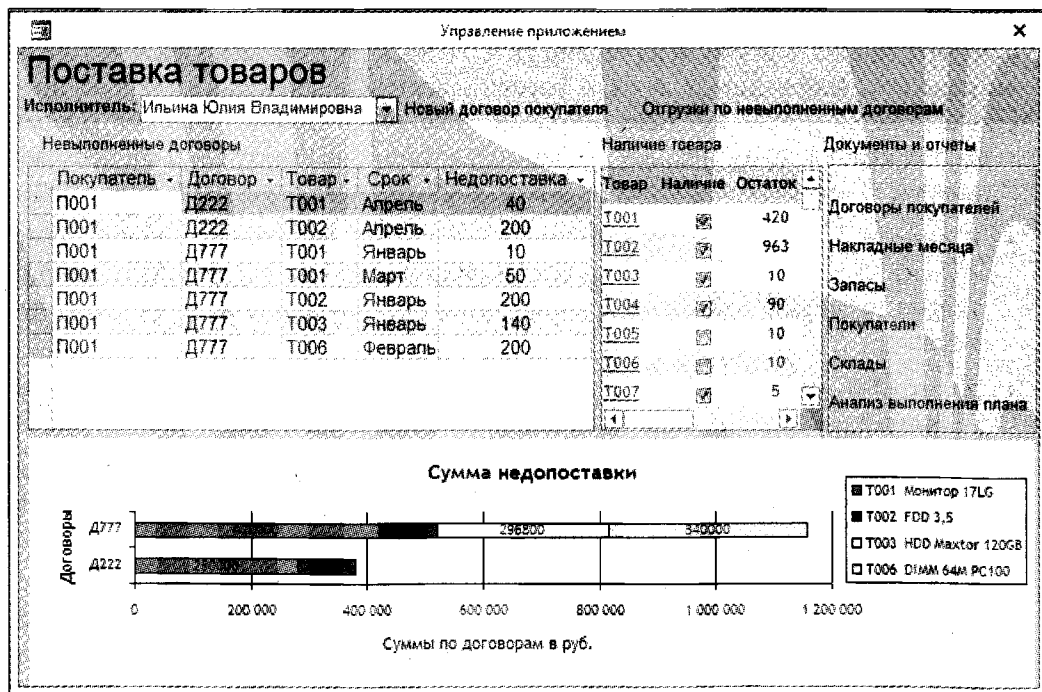


Рис. 10.6. Форма управления приложением

1. *Создайте пустую форму. Дополните открывшуюся в режиме макета форму заголовком, выполнив соответствующую команду на ленте **Конструктор (Design)** в группе **Колонтитулы (Header/Footer)**.*
2. *Замените значение в надписи заголовка на **Поставка товаров**.*
3. *Скопируйте поле со списком из созданной ранее формы **Регистрация исполнителя**, открыв ее в режиме конструктора. В разрабатываемой форме перейдите в режим конструктора, увеличьте ширину заголовка формы и вставьте скопированное поле со списком в заголовок. Чтобы в поле со списком отображался исполнитель, выбранный в окне регистрации, в свойствах поля на вкладке **Данные (Data)** измените **Значение по умолчанию (Default Value)** на: `=[TempVars]![ИД_исполнителя]`. Используемая здесь временная переменная была определена в макросе кнопки **Вход из формы Регистрация исполнителя**.*
4. *Дополните заголовок формы кнопками **Новый договор покупателя** и **Отгрузки по невыполненным договорам**. Для каждой кнопки свяжите с событием **Нажатие кнопки (On Click)** вызов внедренного макроса с макрокомандой, открывающей соответствующую форму. Для открытия формы в режиме ввода нового договора установите в макрокоманде **ОткрытьФорму (OpenForm)** **Режим данных (Data Mode)** — **Добавление (Add)**. Для формы **Отгрузки по невыполненным договорам** установите **Режим данных (Data Mode)** — **Только чтение (Read Only)**.*

Для вывода в форме управления приложением данных о невыполненных договорах используйте подчиненную форму **Невыполненные договоры**, построенную на запросе.

1. *Создайте запрос, в котором из таблиц **ПОСТАВКА_ПЛАН**, **ДОГОВОР** и **ТОВАР** выбираются необходимые поля и рассчитывается недопоставка. Запрос представлен на рис. 10.7. Условие отбора в поле **Недопоставка** позволит выбрать только те строки спецификации договора, в которых не выполнен план поставок. В запросе, кроме отображенных на рисунке полей, используйте еще три поля:*
 - *в первом — выберите срок поставки **СРОК_ПОСТ**, запишите условие отбора `<=Month(Date())` и снимите флажок **Вывод на экран**. Это позволит выбрать только те записи, в которых срок поставки просрочен или истечет в текущем месяце;*
 - *во втором — для вывода срока поставки в виде наименования месяца создайте вычисляемое поле: `Выражение1: MonthName([СРОК_ПОСТ]);`*
 - *в третьем — для объединения кода и наименования изделия создайте вычисляемое поле `Объединение: [ПОСТАВКА_ПЛАН]![КОД_ТОВ] & " " & [НАИМ_ТОВ`. Это поле будет использовано в легенде диаграммы.*
2. *Создайте форму с помощью мастера. В диалоге с мастером определите в качестве источника записей запрос **Недопоставка** и выберите необходимые поля (см. разд. **Невыполненные договоры** на рис. 10.6). Выберите внешний вид формы **табличный (datasheet)**. Задайте имя формы: **Невыполненные договоры**. Созданную форму закройте.*

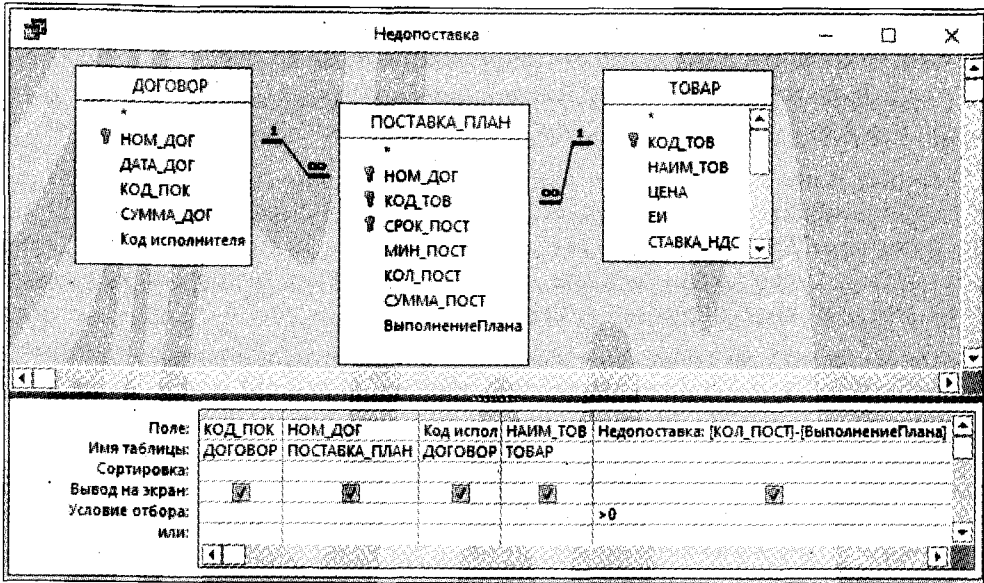


Рис. 10.7. Запрос — источник записей подчиненной формы Невыполненные договоры

3. Перетащите созданную форму из области навигации в область данных пустой формы, открытой в режиме макета или конструктора.
4. В свойствах вставленной подчиненной формы **Невыполненные договоры** на вкладке **Данные (Data)** введите **ПолеСоСписком0** в свойство **Основные поля (Link Master Fields)** и **Код исполнителя** в свойство **Подчиненные поля (Link Child Fields)**. Проверьте:

- действительно ли поле со списком имеет указанное имя (в свойствах этого поля, на вкладке **Другие (Other)**);
- включен ли в список полей, доступных в этой подчиненной форме, **Код исполнителя**. Этот список можно открыть кнопкой **Добавить поля (Add Existing Field)** на ленте **Конструктор (Design)**, если форма открыта в режиме конструктора, а также на ленте **Работа с макетами форм | Режим конструктора (Form Layout Tools | Design)**, если форма открыта в режиме макета.

После определения этих свойств в подчиненной форме будут выводиться только записи, связанные с выбранным в поле со списком исполнителем.

5. Создайте надпись **Невыполненные договоры** и настройте элементы подчиненной формы, как показано на рис. 10.6.

Для вывода в форме управления приложением данных о запасе товаров на складе используйте подчиненную форму, построенную на таблице **ТОВАР**.

1. Создайте форму с помощью мастера форм. В диалоге с мастером определите в качестве источника записей таблицу **ТОВАР**. Выберите доступные поля: **Код товара**, **Наличие товара**, **Остаток** и **Наименование товара**. Выберите внешний вид формы **ленточный (Continuous Forms)**. Задайте имя формы: **Наличие товара**. Созданную форму закройте.

2. Перетащите созданную форму из области навигации в область данных пустой формы, открытой в режиме макета или конструктора.
3. В созданной таким образом подчиненной форме измените заголовок, оставив в нем только надписи полей.
4. Оформите поля с кодом и наименованием товара как гиперссылки. Для этого в свойстве макета **Отображать как гиперссылку** (Display As Hyperlink) выберите значение **Только на экране** (Screen Only). Теперь поля на экране будут выглядеть как гиперссылки. Однако по щелчку на такой «гиперссылке» никаких действий пока выполняться не будет.
5. Чтобы по щелчку на таком поле выполнялись нужные действия, создайте в окне свойств поля для события **Нажатие кнопки** (OnClick) вложенный макрос (рис. 10.8). В макросе предусмотрите открытие формы ТОВАРЫ, в которой представлены сведения о выбранном в вызывающей форме товаре и диаграмма заказов на этот товар по договорам (см. рис. 1.6 в главе 1).

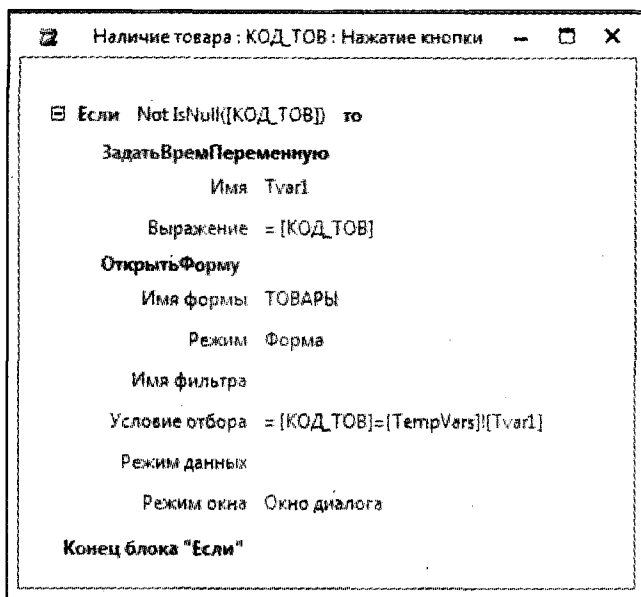


Рис. 10.8. Макрос, вложенный в форму *Наличие товара*

6. Во вложенном макросе для ссылки на поле КОД_ТОВ в вызывающей форме *Наличие товара* используйте временную переменную. Это позволит правильно сформировать условие отбора.
7. Аналогичным образом оформите поле *Наименование товара* (на рис. 10.6 это поле, следующее за полем *Остаток*, не отображено). Поскольку для этого поля целесообразно предусмотреть выполнение такого же макроса, как и для поля *Код товара*, скопируйте туда вложенный макрос *КОД_ТОВ : Нажатие кнопки*:
 - в окне свойств поля *НАИМ_ТОВ* на вкладке **События** (Event) в строке **Нажатие кнопки** (OnClick) щелкните на значке построителя и выберите **Макросы**;

- в открывшемся окне конструктора макросов откройте в каталоге макрокоманд раздел **В этой базе данных (In this Database)**, откройте список форм и список формы **Наличие товара**, выберите макрос **КОД_ТОВАР.OnClick** и выполните команду его контекстного меню **Добавить копию макроса (Add Copy of Macro)**.
8. Создайте надпись **Наличие товара** и настройте элементы подчиненной формы, как показано на рис. 10.6.

Для вывода в форме управления приложением ссылок на наиболее часто используемые документы и отчеты используйте набор кнопок, заключенных в прямоугольник.

1. Создайте кнопки, используя соответствующий элемент управления на ленте конструктора. В диалоге с мастером выберите одно из действий, которое будет выполняться при нажатии кнопки: **Открыть форму (Open Form)** или **Открыть отчет (Open Report)**. Эти действия содержатся в категориях **Работа с формой (Form Operations)** и **Работа с отчетом (Report Operations)**.
2. Выберите форму или отчет, для открытия которого предназначена кнопка. Введите текст, который нужно разместить на кнопке, соответствующий названию документа или отчета. Мастер для открытия выбранного объекта создаст внедренный макрос, который будет связан с событием **Нажатие кнопки (OnClick)**.
3. Чтобы кнопка приняла вид простой надписи, выберите в свойствах макета **Тип границы (Border Style)** — **Отсутствует (Transparent)**, а **Тип фона (Back Style)** — **Прозрачный (Transparent)**.
4. Создайте надпись **Документы и отчеты**, используя соответствующий элемент управления на ленте конструктора.

Для вывода в форме управления приложением диаграммы стоимости товаров, не отгруженных по договорам, используйте рассмотренный выше запрос **Недоставка** (см. рис. 10.7). В режиме SQL этот запрос имеет вид:

```
SELECT ДОГОВОР.КОД_ПОК, ПОСТАВКА_ПЛАН.НОМ_ДОГ, ДОГОВОР.[Код исполнителя],
ТОВАР.НАИМ_ТОВ, [КОЛ_ПОСТ]-[ВыполнениеПлана] AS Недоставка,
ПОСТАВКА_ПЛАН.КОД_ТОВ AS Товар, ([КОЛ_ПОСТ]-[ВыполнениеПлана])*[ЦЕНА] AS
Стоимость, MonthName([СРОК_ПОСТ]) AS Выражение1, [ПОСТАВКА_ПЛАН]![КОД_ТОВ] & "
" & [НАИМ_ТОВ] AS Объединение
FROM ДОГОВОР INNER JOIN (ТОВАР INNER JOIN ПОСТАВКА_ПЛАН ON ТОВАР.КОД_ТОВ =
ПОСТАВКА_ПЛАН.КОД_ТОВ) ON ДОГОВОР.НОМ_ДОГ = ПОСТАВКА_ПЛАН.НОМ_ДОГ
WHERE ((([КОЛ_ПОСТ]-[ВыполнениеПлана])>0) AND
((ПОСТАВКА_ПЛАН.СРОК_ПОСТ)<=Month(Date())));
```

Для создания диаграммы откройте форму в режиме конструктора и с помощью элемента управления **Диаграмма (Chart)** на ленте конструктора нарисуйте область диаграммы.

1. В диалоге с мастером выберите запрос **Недоставка** и поля, которые необходимы для построения диаграммы.
2. Выберите тип диаграммы **Линейчатая (Bar Chart)** и тип отображения данных на ней.

2. Перетащите созданную форму из области навигации в область данных пустой формы, открытой в режиме макета или конструктора.
3. В созданной таким образом подчиненной форме измените заголовок, оставив в нем только надписи полей.
4. Оформите поля с кодом и наименованием товара как гиперссылки. Для этого в свойстве макета **Отображать как гиперссылку** (Display As Hyperlink) выберите значение **Только на экране** (Screen Only). Теперь поля на экране будут выглядеть как гиперссылки. Однако по щелчку на такой «гиперссылке» никаких действий пока выполняться не будет.
5. Чтобы по щелчку на таком поле выполнялись нужные действия, создайте в окне свойств поля для события **Нажатие кнопки** (OnClick) внедренный макрос (рис. 10.8). В макросе предусмотрите открытие формы ТОВАРЫ, в которой представлены сведения о выбранном в вызывающей форме товаре и диаграмма заказов на этот товар по договорам (см. рис. 1.6 в главе 1).

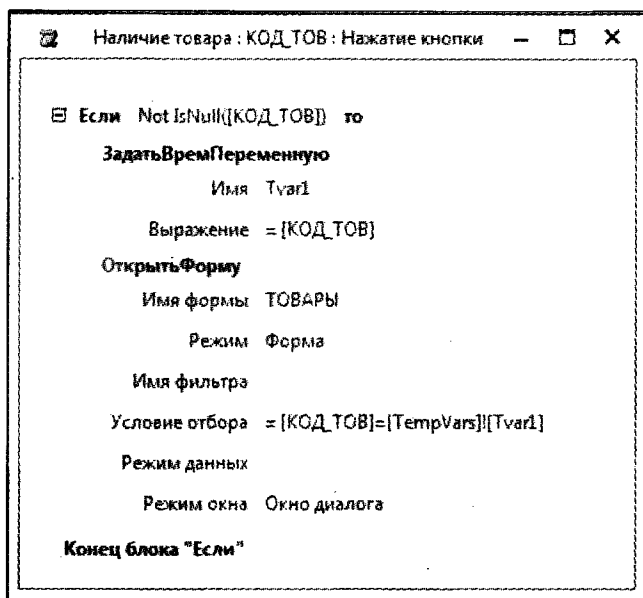


Рис. 10.8. Макрос, внедренный в форму Наличие товара

6. Во внедренном макросе для ссылки на поле КОД_ТОВ в вызывающей форме Наличие товара используйте временную переменную. Это позволит правильно сформировать условие отбора.
7. Аналогичным образом оформите поле Наименование товара (на рис. 10.6 это поле, следующее за полем Остаток, не отображено). Поскольку для этого поля целесообразно предусмотреть выполнение такого же макроса, как и для поля Код товара, скопируйте туда внедренный макрос *КОД_ТОВ : Нажатие кнопки*.
 - в окне свойств поля НАИМ_ТОВ на вкладке **События** (Event) в строке **Нажатие кнопки** (OnClick) щелкните на значке построителя и выберите **Макросы**;

- Для изменения диаграммы при переходе между значениями поля со списком выберите поля, связывающие его с диаграммой: ПолеСоСписком0 (проверьте имя поля со списком) и Код исполнителя.

В области диаграммы отобразится макет. Для просмотра диаграммы перейдите в режим формы.

- Для изменения диаграммы в режиме конструктора выделите ее и выполните команду контекстного меню **Объект Chart | Правка (Chart Object | Edit)**. Двойной щелчок мыши в области диаграммы также позволяет перейти в режим ее изменения. В режиме изменения область диаграммы обрамлена серой рамкой. Чтобы отражался вклад каждой категории в общей сумме, выполните на ленте работы с диаграммой команду **Диаграмма | Тип диаграммы (Chart Type)** и выберите **Линейчатая диаграмма с накоплением (Stacked Bar)**.
- Выполнив команду контекстного меню **Параметры диаграммы (Chart Options)**, настройте вид диаграммы: определите заголовки осей, подписи данных.
- Для возврата в форму щелкните в любой ее части за пределами диаграммы.
- Если при создании диаграммы не была установлена ее связь с полем со списком, выделите диаграмму и откройте ее свойства. На вкладке **Данные (Data)** введите имена полей связи. В строку **Основные поля (Link Master Fields)** введите ПолеСоСписком0, в строку **Подчиненные поля (Link Child Fields)** — Код исполнителя.
- В строке **Источник строк (Row Source)** диаграммы просмотрите перекрестный запрос. В режиме SQL инструкция TRANSFORM этого источника строк будет записана в виде:

```
TRANSFORM Sum(Недопоставка.[Стоимость]) AS Сумма_Стоимость
SELECT Недопоставка.[НОМ_ДОГ]
FROM Недопоставка
GROUP BY Недопоставка.[НОМ_ДОГ]
PIVOT Недопоставка.Объединение;
```

Инструкция TRANSFORM указывает, по какому полю и с помощью какой статистической функции рассчитываются значения рядов данных диаграммы (столбцов):

```
Sum(Недопоставка.[Стоимость]) AS Сумма_Стоимость.
```

Встроенная инструкция SELECT указывает поле Недопоставка.[НОМ_ДОГ], значения которого используются на оси категорий.

Предложение GROUP BY указывает, что значения поля НОМ_ДОГ используются в заголовках строк, и по этому полю выполняется группировка.

Предложение PIVOT указывает, что значения поля Объединение (см. запрос Недопоставка в режиме SQL) могут использоваться в именах рядов данных. По этому полю также выполняется группировка. Кроме того, предложение PIVOT определяет поле, значения которого будут использованы в легенде диаграммы,

определяющей названия рядов данных. Заметим, что в предложении PIVOT вместо поля может использоваться выражение или набор фиксированных значений.

Использование форм с вкладками при разработке интерфейса

При разработке интерфейса приложения для представления функционально связанных объектов в одной форме может быть использован набор вкладок. Размещение на страницах вкладок различных подчиненных форм/отчетов позволяет удобно организовать работу специалиста с необходимыми при решении конкретных задач документами. При этом они легко доступны, и нет необходимости отвлекаться на их поиск.

Создадим форму, на вкладках которой представим данные, необходимые для оценки ситуации по отгрузке каждого заказанного в договорах товара:

- данные о запланированных по договорам отгрузках;
- данные о фактических отгрузках товара;
- данные по анализу итогов отгрузки товара.

Такая форма с именем **Отгрузка товаров по договорам** в окончательном виде приведена на рис. 10.9. На вкладках формы могут размещаться как формы, так и отчеты, в приведенном примере они связаны с основной формой первой вкладки, т. е. отображают данные об отгрузке товара, выбранного на первой вкладке.

1. Создайте пустую форму. Дополните открывшуюся в режиме макета форму заголовком, выполнив соответствующую команду на ленте **Конструктор (Design)** в группе **Колонтитулы (Header/Footer)**.
2. Замените значение в надписи заголовка на: **ВСЕ ОБ ОТГРУЗКЕ ТОВАРА**.
3. На ленте **Конструктор (Design)** щелкните на элементе **Вкладка (Tab Control)** и вычертите в области данных формы область вкладок — в форме будет создано две вкладки с произвольными подписями типа: **Вкладка1** и **Вкладка2**.
4. Одновременно с созданием пустой формы открывается список полей, в котором отображена запись: **Нет полей доступных для добавления в текущее представление (No fields available to be added to the current view)**. При этом возможно открытие списка всех таблиц базы данных. Щелкните на ссылке **Показать все таблицы (Show all tables)** и переместите в область первой вкладки поля таблицы **ТОВАР**, как показано на рис. 10.9. Если список полей не отобразился после создания пустой формы, выполните команду **Добавить поля (Add Existing Field)** на ленте **Конструктор (Design)** в группе **Сервис (Tools)**.
5. После перемещения полей на вкладку таблица **ТОВАР** будет отображена в разделе списка полей: **Поля, доступные в данном представлении (Fields available for this view)**, а для формы в качестве источника записей создан запрос на выборку — представление, записанное в соответствующей строке свойств формы инструкцией SQL на вкладке **Данные (Data)**.

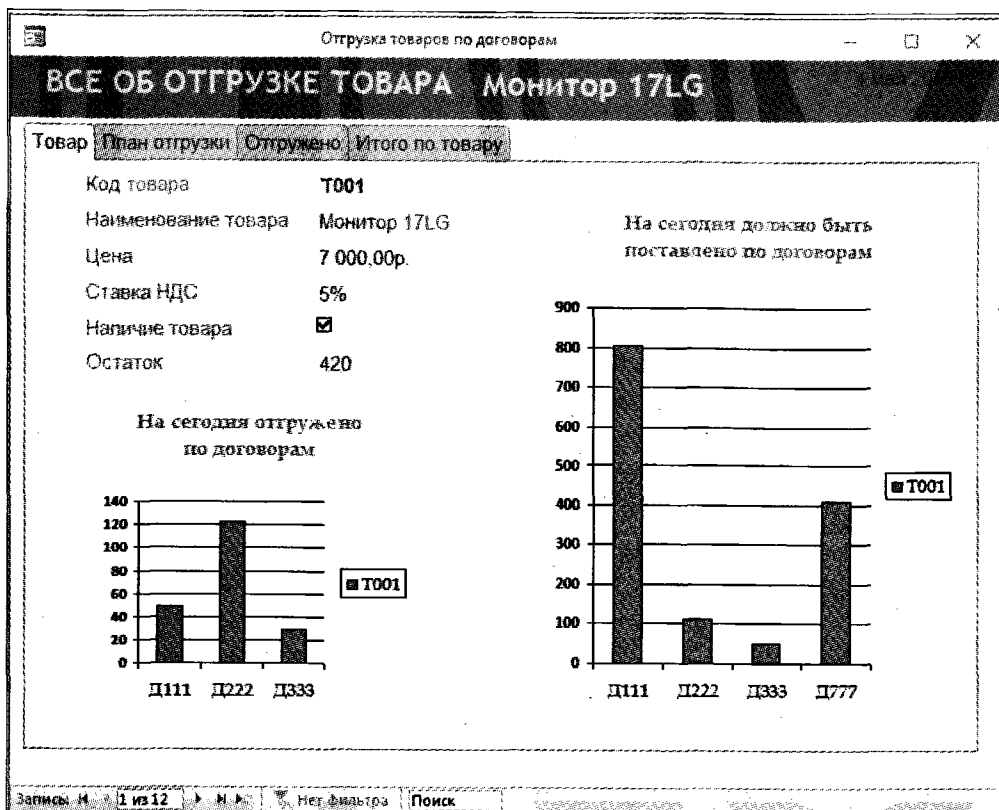


Рис. 10.9. Форма с вкладками для получения сведений об отгрузках товара

- Измените отображаемую подпись вкладки на **Товар**. Для этого откройте свойства вкладки и в строке **Подпись** (Caption) на вкладке свойств **Макет** (Format) введите: Товар.

Постройте диаграммы для отображения на вкладке сведений о плановых и фактических отгрузках товара. Для создания источника строк диаграмм используйте запросы, представленные на рис. 10.10 (в этом запросе отбираются записи со сроком поставки товаров до текущего месяца) и 10.11.

- Для создания диаграммы перейдите в режим **Конструктор** и на ленте выберите элемент управления **Диаграмма** (Chart). Вычертите область размещения диаграммы на вкладке — ее рамку. Предварительно увеличьте область, занимаемую вкладкой.
- В диалоговом окне запущившегося мастера создания диаграмм выберите **нужный** запрос: ТекущийПланПоставок.
- В следующем окне мастера выберите поля с данными, которые необходимо отобразить в диаграмме: **НОМ_ДОГ** — номер договора, **КОД_ТОВ** — код товара, **КОЛ_ПОСТ** — количество, которое необходимо поставить.
- В следующем окне выберите тип диаграммы: **Гистограмма** (Column Chart).

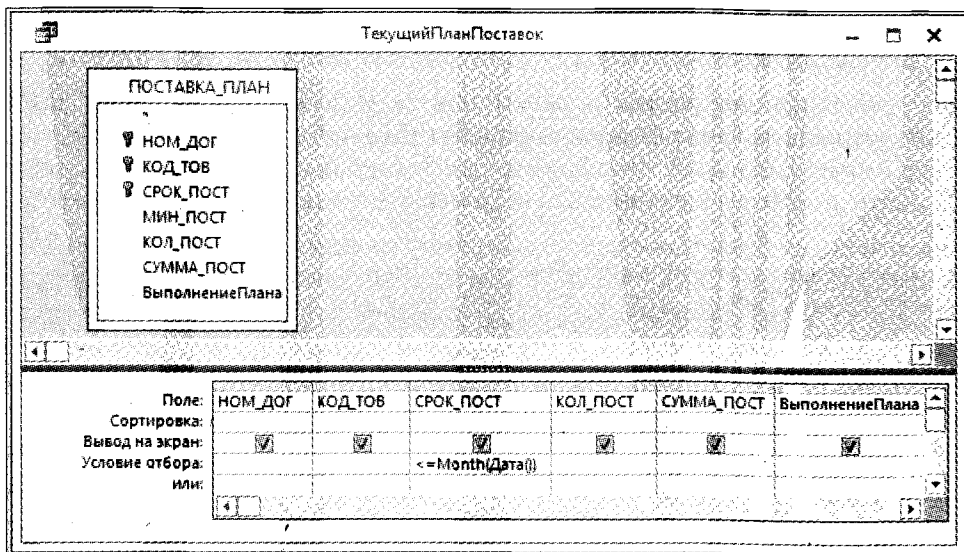


Рис. 10.10. Запрос для получения сведений о запланированных поставках товара

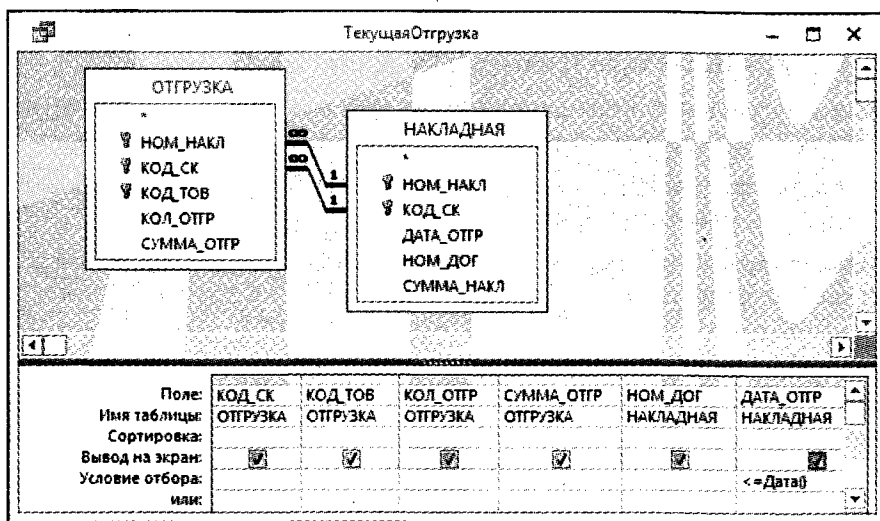


Рис. 10.11. Запрос для получения сведений о реальных отгрузках товара на текущую дату

- Оставьте предложенный мастером тип отображения данных на диаграмме, т. к. он соответствует заданному в форме на вкладке **Товар**. Номера договоров отобразятся на оси, код товара — в рядах, суммарное количество товара, поставленного по каждому из договоров, — в области данных.
- В следующем окне создания диаграммы для изменения диаграммы при переходе между записями выберите поле, связывающее документ с диаграммой: КОД_ТОВ.
- Задайте название диаграммы: На сегодня должно быть поставлено по договорам.

8. Для правильного отображения всех элементов диаграммы измените размеры рамки диаграммы.
9. Для изменения диаграммы в целом или ее отдельных элементов, находясь в режиме конструктора формы, выполните двойной щелчок на ней. Доступный теперь инструментарий используйте для настройки диаграммы. Для возврата в конструктор форм щелкните в любом месте формы за пределами диаграммы.
10. Аналогичным образом постройте диаграмму на основе запроса Текущая отгрузка, выбрав поля с данными, которые необходимо отобразить в диаграмме: `НОМ_ДОГ` — номер договора, `КОЛ_ОТГР` — количество, которое было отгружено, `КОД_ТОВ` — код товара. Если тип отображения данных на диаграмме не соответствует желаемому, переместите поля с диаграммы в список и затем переместите из списка поле `НОМ_ДОГ` на **Оси (Axis)**, поле `КОД_ТОВ` — в **Ряды (Data Series)** и поле `КОЛ_ОТГР` — в **Данные (Data)**.
11. Для отображения в заголовке основной формы наименования выбранного товара создайте там поле, в которое запишите выражение: `=[НАИМ_ТОВ]`.

На следующей вкладке организуйте вывод подробных данных о запланированных по договорам отгрузках выбранного товара. Для этого:

1. Создайте на основе запроса ТекущийПланПоставок ленточную форму, выбрав этот запрос и выполнив на ленте **Создание (Create)** в списке **Другие формы (More Forms)** команду **Несколько элементов (Multiple Items)**.
2. Вставьте в заголовок формы дату, переместите в него поле `КОД_ТОВ` — код товара, поскольку на вкладке отображаться будет только один вид товара.
3. В примечании формы создайте поле, в которое введите выражение для подсчета общего количества отгруженного товара: `=Sum([КОЛ_ПОСТ])`.
4. Перетащите созданную форму на вторую вкладку формы, открытой в режиме макета или конструктора.
5. Если при этом не установилась связь с основной записью на первой вкладке, откройте свойства вставленной формы и на вкладке **Данные (Data)** щелкните на значке построителя в строке **Основные поля (Link Master Fields)**. В открывшемся окне (рис. 10.12) в качестве поля связи должно быть выбрано поле `КОД_ТОВ`. Очевидно, что можно `КОД_ТОВ` ввести в строки свойств **Основные поля (Link Master Fields)** и **Подчиненные поля (Link Child Fields)** вручную, не обращаясь к помощи построителя.

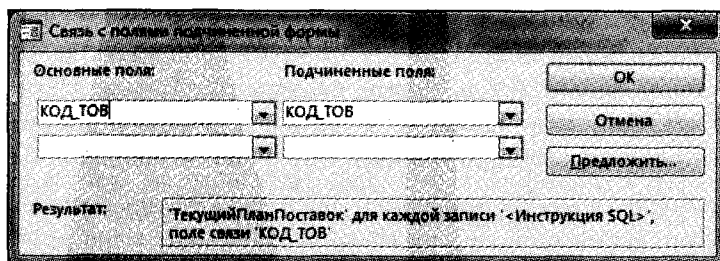


Рис. 10.12. Определение связи форм на первой и второй вкладках

6. Измените отображаемую подпись вкладки на **План отгрузки**. Для этого откройте свойства вкладки и в строке **Подпись (Caption)** на вкладке свойств **Макет (Format)** введите: План отгрузки.

Форма с подробными данными о запланированных по договорам отгрузках выбранного товара, размещенная на второй вкладке, представлена на рис. 10.13.

На	04.05.2016	Товар	Т001	
Номер договора	Срок поставки	Количество	Сумма	Выполнено
Д111	1	305	2 135 000,00р.	191
Д111	2	200	1 400 000,00р.	100
Д111	3	300	2 100 000,00р.	0
Д222	1	100	700 000,00р.	140
Д222	2	12	84 000,00р.	42
Д333	3	50	350 000,00р.	0
Д777	1	10	70 000,00р.	10
Д777	2	300	2 100 000,00р.	300
Д777	3	100	700 000,00р.	0
Итого запланировано:		1377		

Рис. 10.13. Форма с данными о запланированных поставках товара на вкладке **План отгрузки**

Задание 10.1. Добавление новой вкладки с данными о фактических отгрузках товара

Добавьте новую вкладку и организуйте на ней вывод подробных данных о фактических отгрузках выбранного товара по договорам. Для этого:

1. Для добавления новой вкладки в форму щелкните на имеющейся вкладке и выполните команду контекстного меню **Вставить вкладку (Insert Page)**.
2. Так же, как в приведенном ранее примере, создайте на основе запроса ТекущаяОтгрузка ленточную форму, доработайте форму, вставьте ее на новую вкладку, установите необходимую связь этой формы с записью первой вкладки и измените подпись вкладки на **Отгружено**.
3. Если возникла необходимость изменения последовательности размещения вкладок в форме, используйте команду контекстного меню любой из вкладок **После**

довательность вкладок (Page Order). В открывающемся окне **Порядок страниц (Page Order)** любая из вкладок может быть перемещена в нужное место.

Задание 10.2. Добавление новой вкладки с данными о выполнении плана отгрузки

Добавьте еще одну вкладку: **Итого по товару**. Организуйте на ней вывод итоговых данных о плановых, фактических отгрузках выбранного товара и отклонении от плана. Для получения результата используйте запросы, позволяющие подсчитать общее количество каждого из товаров, запланированного на текущий месяц, фактически отгруженного на текущую дату, и разность между ними (аналогичные запросы для решения этой задачи: План, Факт и Анализ выполнения плана приведены в главе 4, не используйте в них параметров). На основе последнего запроса создайте отчет, который так же просто, как форма, размещается на странице вкладки.

Задание 10.3. Создание формы с вкладками

Создайте форму на основе таблицы ИСПОЛНИТЕЛЬ и в нее вставьте вкладки со связанными формами и отчетами о деятельности этого исполнителя. В заголовке формы выведите должность и фамилию исполнителя.

Панель навигации формы с вкладками

Заголовок и панель навигации (кнопки перехода по записям) основной формы, размещенной на первой вкладке, сохраняются на всех других вкладках формы. Это позволяет переходить к другой записи основной формы и просматривать связанные данные на любой другой вкладке — например, со сведениями по отгрузкам любого товара, не возвращаясь для выбора к первой вкладке.

Пользуйтесь фильтрами для перехода к нужной записи.

1. На вкладке **Товары** установите курсор на поле с кодом товара и в контекстном меню выполните команду **Текстовые фильтры | Равно (Text Filters | Equals)**.
2. В диалоговом окне **Настраиваемый фильтр (Custom Filter)** введите нужный код товара (рис. 10.14) — в форме отобразится только запись с введенным кодом товара.

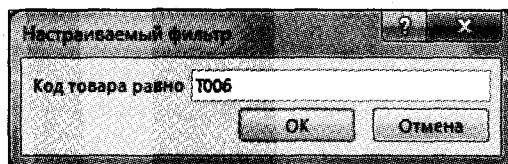


Рис. 10.14. Диалоговое окно ввода значения для фильтра

3. Уберите фильтр, щелкнув мышью на кнопке панели навигации **С фильтром (Filtered)**. Для повторного использования последнего фильтра щелкните на кнопке **Без фильтра (Unfiltered)**.

4. Чтобы совсем удалить фильтр, выполните команду контекстного меню поля с кодом товара **Снять фильтр с Код товара** (Clear filter from Код товара). При этом кнопка фильтра тускнеет и получает имя **Нет фильтра** (No Filter).

Текстовые фильтры позволяют отобразить записи не только по равенству значений в поле, а и по значительно более сложным критериям, таким как **Начинается с...**, **Заканчивается на...** (Ends With...) и т. д.

Для числовых полей используйте **Числовые фильтры** (Number Filters), позволяющие отобразить записи со значениями, заданными условием фильтрации.

Для перехода к нужной записи можно воспользоваться инструментами поиска.

На панели навигации щелкните в поле **Поиск** (Search). Курсор может быть установлен в любом поле. Начинайте ввод нужного наименования товара. По мере ввода значения будет отображаться запись с наименованием товара, начинающимся с введенных символов. Если нет товара с наименованием, начинающимся с введенного символа или символов, текущей остается старая запись. Можно вводить и код товара. Поиск осуществляется по всем полям записей, в том числе и по числовым. Если ввести одну или несколько букв или цифр, повторным нажатием на клавишу <Enter> можно просмотреть все значения, начинающиеся с них.

Задание 10.4. Использование фильтра для отбора записей

С помощью фильтра отберите только те товары, которые имеются в наличии. Установите курсор на поле **Наличие товара** и в контекстном меню выполните команду **Выделено** (Is Selected) или **Не выделено** (Is Not Selected), в зависимости от того, свидетельствует значение о наличии товара или нет. Будут отобраны все записи, в которых значение в поле **Наличие товара** равно или не равно значению в текущей записи.

Формы навигации

Форма навигации предназначена для создания на ней горизонтальных, вертикальных или и тех, и других вкладок одновременно. На этих вкладках могут размещаться формы и отчеты.

Создаются формы навигации командами из списка **Навигация** (Navigation), расположенного на ленте **Создание** (Create) в группе **Формы** (Forms).

Для размещения формы или отчета на странице вкладки достаточно выбрать его в области навигации базы данных и перетащить в ряд кнопок навигации перед кнопкой **Создать** (Add New). Автоматически будет создана кнопка навигации с именем формы/отчета, и перемещаемый объект размещен на странице вкладки. Для размещения объекта на странице вкладки можно просто ввести имя формы или отчета вместо слова **Создать** (Add New) на кнопке навигации. Впоследствии можно изменить имя на кнопке навигации, при этом на странице сохранится ранее размещенный объект. Прежний объект на странице вкладки сохраняется, даже если имя на кнопке навигации меняется на совпадающее с именем некоторой другой формы или отчета.

Для удаления формы/отчета на странице вкладки нужно выделить кнопку навигации и выполнить команду контекстного меню Удалить (Delete) или просто нажать клавишу <Delete>.

Размещенная на одной странице вкладки подчиненная форма/отчет не связывается с объектами, размещенными на страницах других вкладок.

Создайте форму навигации с горизонтальным расположением вкладок в два уровня. Для этого выполните соответствующую команду, расположенную на ленте Создание (Create) в группе Формы (Forms) в списке Навигация (Navigation).

Кнопки навигации первого уровня используйте для открытия группы вкладок второго уровня. Дайте этим кнопкам имена, которые будут определять суть содержимого группы, например, Справочники, Плановые документы, Оперативные

Форма навигации


Справочники | Плановые документы | Оперативные документы | **Отчеты**

Отгрузки каждого товара по месяцам | Все отгрузки по месяцам | Статистика договоров покупателей

4 мая 2016 г.

ОТГРУЗКА ТОВАРА **Монитор 17LG** **Код T001**

Цена	7 000,00р.	
Единица измерения	штука	
Ставка НДС	5%	
Наличие товара	<input checked="" type="checkbox"/>	



	Дата отгрузки	Код накладной	Код склада	Количество отгружено	Сумма отгружено
Январь 2016					
	18.01.2016	Н001	С01	50	350 000,00р.
	Итого за месяц			50	350 000,00р.
Февраль 2016					
	10.02.2016	Н005	С01	40	280 000,00р.
	21.02.2016	Н004	С02	10	70 000,00р.
	11.02.2016	Н002	С02	30	210 000,00р.
	11.02.2016	Н002	С01	40	280 000,00р.
	Итого за месяц			120	840 000,00р.
Март 2016					
	25.03.2016	Н004	С01	2	14 000,00р.
	24.03.2016	Н002	С03	30	210 000,00р.
	Итого за месяц			32	224 000,00р.
Общий итог по товару					1 414 000,00р.

Рис. 10.15. Форма навигации с горизонтальными вкладками в два уровня

документы, Отчеты (рис. 10.15). Имена не должны совпадать с именами имеющихся в базе форм и отчетов.

Разместите на страницах вкладок второго уровня формы справочников базы данных, формы для работы с плановыми документами — такими как договоры, оперативно-учетными документами — такими как накладные, отчеты, позволяющими проанализировать текущую ситуацию. Для размещения нужных форм и отчетов на страницах вкладок достаточно перетаскивать их в нужные места в ряду кнопок навигации или вводить их имена на кнопках навигации.

Формы, размещаемые на страницах вкладок, в свою очередь могут включать вкладки, обеспечивая таким образом дополнительные уровни иерархии.

Очевидно, формы навигации должны настраиваться на работу специалиста с определенным кругом обязанностей, и в организации для каждой группы сотрудников, имеющих схожие функциональные обязанности, могут быть разработаны соответствующие формы навигации.

Контрольные вопросы

1. К каким последствиям приводят значения свойств формы **Модальное окно** (Modal) и **Всплывающее окно** (PopUp)?
2. Можно ли в поле со списком в качестве источника строк использовать вычисляемое поле?
3. Может ли в поле со списком вводиться значение по умолчанию?
4. В какой момент выполняется макрос с именем **Autoexec**?
5. К каким последствиям приводит нажатие кнопки **Включить содержимое** (Enable Content) на панели сообщений?
6. Какая инструкция языка SQL записывается в строке **Источник строк** (RowSource) диаграммы?
7. Возможно ли установление связи между записями различных вкладок?
8. Сохраняются ли на второй и последующих вкладках кнопки перехода по записям формы с первой вкладки?
9. Какие объекты могут размещаться на вкладках формы навигации?
10. Можно ли поместить форму на вкладку формы навигации, заменив на кнопке навигации слово **Создать** (Add New) именем формы?
11. Допускает ли форма навигации создание нескольких уровней вкладок?

Ответы

1. При открытии формы с этими свойствами делается недоступной работа со всеми объектами базы данных и отключается доступ ко всем инструментам на лентах.
2. Да.

3. Да.
4. В момент открытия базы данных.
5. Разрешает выполнение в базе данных опасного содержимого, такого как запросы на изменение.
6. Инструкция `TRANSFORM`.
7. Да.
8. Да.
9. Формы и отчеты.
10. Да.
11. Да.

ГЛАВА 11



Создание веб-приложений

Начиная с версии 2013, Access включает средства, упрощающие создание и развертывание веб-приложений на сайтах Microsoft SharePoint для совместной работы пользователей с данными базы. Новая модель приложения Access позволяет пользователям с небольшим опытом программирования или вовсе без него быстро в интерактивном режиме создавать веб-приложения, ориентированные на работу с данными базы.

Публикация веб-приложения на страницах Microsoft SharePoint обеспечивает доступ к нему из корпоративной сети или через Интернет с любого устройства с помощью веб-браузера, не устанавливая приложение Access на компьютере.

Для создания, публикации и использования веб-приложения, помимо обычного приложения Access 2013/2016, понадобится сервер SharePoint Server 2013/2016 с поддержкой служб Access, установленный в сети компании, или облачная версия SharePoint Online 2013/2016 с поддержкой служб Access, поставляемая отдельно или в составе Office 365. При публикации веб-приложения на сайте SharePoint Server, размещаемого компанией локально, база данных создается на сервере SQL Server 2012 или более поздней версии, который определен для SharePoint. При создании веб-приложения в «облаке» база данных создается на сервере SQL Azure. В базе данных сервера хранятся все объекты и данные, необходимые для приложения, — такие как таблицы, формы, запросы и макросы. Каждый раз, когда пользователи входят в приложение, вводят или изменяют данные, они взаимодействуют с этой базой данных.

На сервере SharePoint службы Access обрабатывают данные и готовят их к просмотру в окне браузера, где пользователи в соответствии со своими правами могут изменять, добавлять, удалять или только просматривать данные. Права пользователя веб-приложения назначаются с помощью семи уровней разрешений (ролей) SharePoint, которые определяют, что может делать тот или иной пользователь или группа пользователей с контентом сайта. По умолчанию каждый сайт SharePoint включает некоторые группы, для которых определяются уровни разрешений. Так для сайта группы автоматически предусмотрены три стандартные группы: Посетители, Участники, Владельцы. Группа Владельцы имеет уровень разрешений — *Полный доступ*, Участники — *Изменить*, Посетители — *Чтение*. Только полный

доступ позволяет добавлять и изменять элементы самого сайта — этот уровень разрешений включает все доступные в SharePoint разрешения. Участники могут изменять данные, но не могут изменять структуру сайта, а Посетители — только просматривать существующие данные. Администратор может определить собственные группы и уровни разрешений для них.

В Access для ограничения доступа к данным базы веб-приложения предусмотрены функции блокировки и скрытия таблиц. При блокировке таблицы возможен только просмотр данных таблицы. Значки, обеспечивающие переход к редактированию, удалению, добавлению записей при просмотре таблицы в режиме списка, не доступны. Скрытая таблица не видна при работе с веб-приложением в браузере. В Access скрытая таблица перемещается в конец списка таблиц приложения и выделяется пунктирной линией и блеклым цветом.

Очевидно, что бизнес-процессы практически любой предметной области не ограничиваются интерактивным вводом, просмотром и изменением данных базы многими пользователями, располагающими только доступом в Интернет и браузером. Для обеспечения возможности расширенной обработки данных веб-приложения в Access предусмотрена функция создания локального приложения, подключающегося к базе данных веб-приложения через интерфейс ODBC. При этом в локальном приложении таблицы базы данных веб-приложения отображаются как связанные. Такие таблицы могут быть использованы как обычные таблицы локальной базы данных при решении любых задач предметной области и построении нужных отчетов. Таким образом, рассматриваемая технология не только обеспечивает совместную работу удаленных пользователей с базой данных, но и обеспечивает полный цикл обработки данных, размещенных в «облаке».

Выбор расположения для веб-приложения

Для разработки веб-приложения в стартовом окне Access в области создания выберите **Пользовательское веб-приложение** (Custom web app). В открывшемся окне необходимо дать приложению имя и выбрать для него расположение в Интернете — сайт SharePoint, на котором будет публиковаться приложение при разработке и на который будут приходить и работать пользователи приложения. Если вы используете SharePoint в «облаке» как отдельное предложение или как часть Office 365 или локальный сайт SharePoint 2013 с поддержкой служб Access, в списке **Доступные места** (Available Locations) должно быть указано хотя бы одно расположение. Если в списке **Доступные места** отсутствует нужное расположение, введите в поле **URL-адрес сайта SharePoint или Office 365** (SharePoint or Office 365 site URL) альтернативный адрес доступного сайта (рис. 11.1).

ЗАМЕЧАНИЕ

Протокол HTTPS — это расширенный протокол HTTP, поддерживающий шифрование.

При использовании сайта SharePoint с поддержкой служб Access, установленного в своей организации, в списке **Доступные места** (Available Locations) и в поле **URL-адрес сайта SharePoint или Office 365** (SharePoint or Office 365 site URL) вы

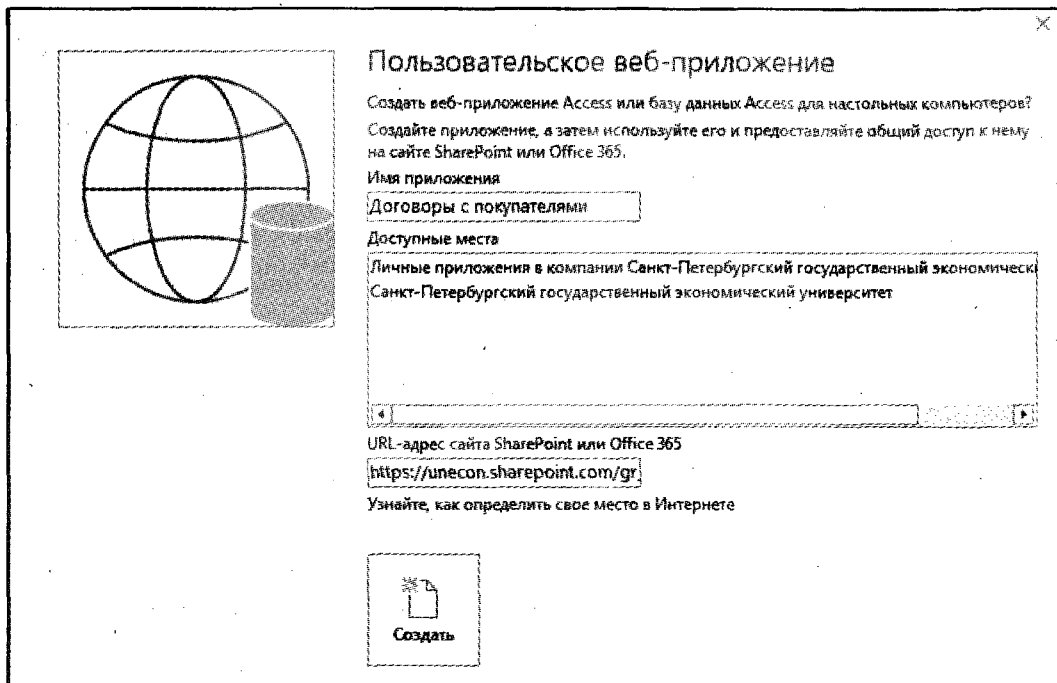


Рис. 11.1. Окно выбора сайта SharePoint для расположения веб-приложения

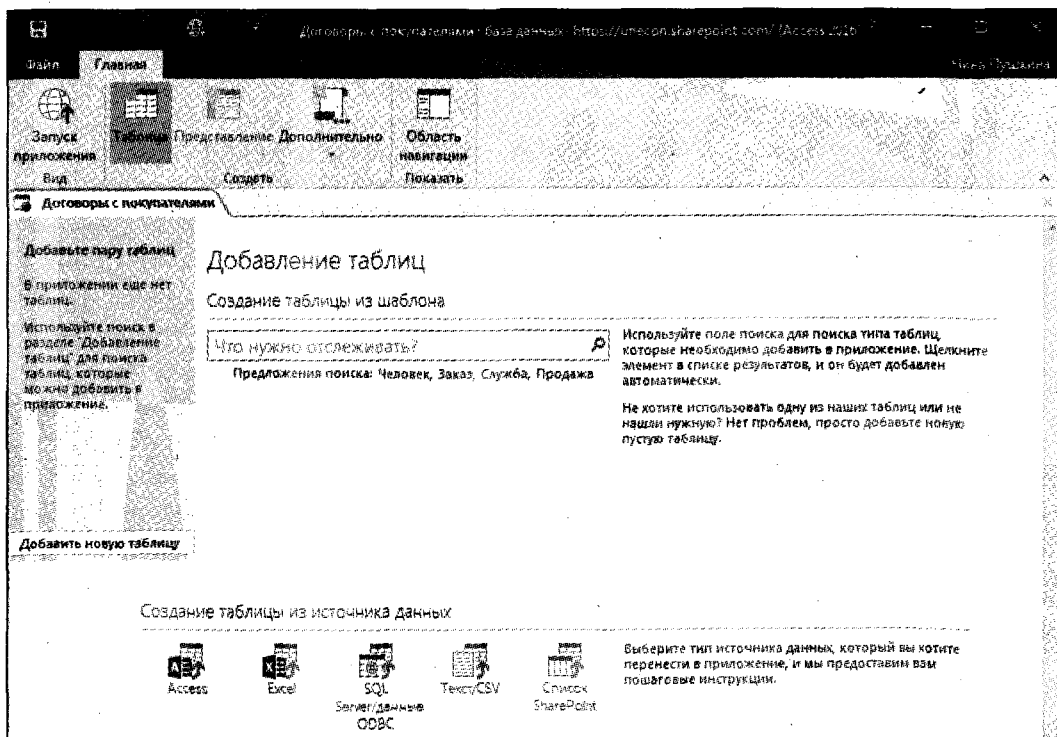


Рис. 11.2. Окно разработки веб-приложения Договоры с покупателями

увидите корневую папку сайта. Если администратор сайта в вашей организации указал другие места для приложений, они будут перечислены в списке. Так, на рис. 11.1 в списке **Доступные места** (Available Locations) показаны сайты, где могут создавать приложения при использовании Office 365 пользователи СПбГЭУ. При выборе сайта группы приложение будет доступно всем пользователям сайта.

Выбрав расположение и заполнив поле **Имя приложения** (App Name) — например, **Договоры с покупателями**, нажмите кнопку **Создать** (Create). Чтобы создать веб-приложение Access, необходимо располагать разрешением уровня *Полный доступ* на веб-сайте.

В открывшемся окне **Добавление таблиц** (рис. 11.2) можно начать разрабатывать веб-приложение.

ЗАМЕЧАНИЕ

Для просмотра адреса сайта веб-приложения, расположения SQL Server и имени базы данных созданного веб-приложения выполните переход **Файл | Сведения** (File | Info). На рис. 11.3 показан пример расположения веб-приложения при работе в Office 365.

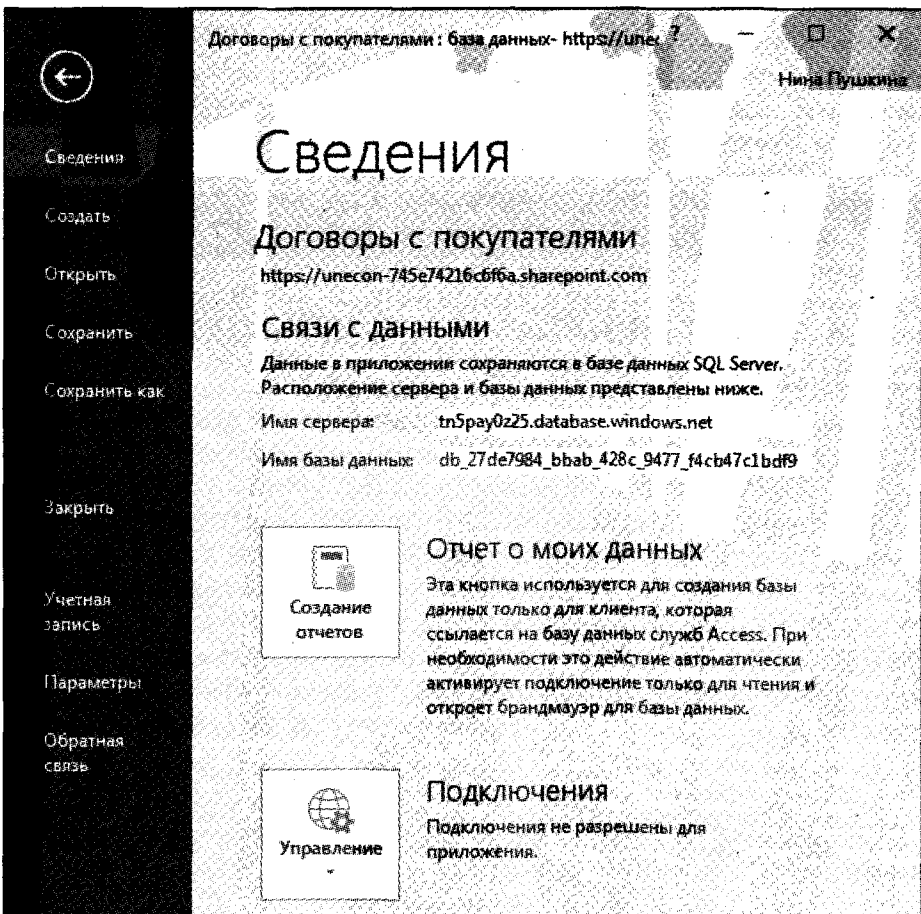


Рис. 11.3. Сведения о расположении сайта и базы данных веб-приложения

Добавление таблиц в базу данных веб-приложения

Разработка веб-приложения начинается с добавления таблиц в базу данных. Для быстрого добавления таблиц используются предварительно разработанные шаблоны, которые могут содержать готовые связанные таблицы.

Чтобы добавить таблицу-шаблон, введите в поле поиска описание данных, которые вы собираетесь хранить в таблице — например, Продажи. Нажмите кнопку **Поиск** и выберите подходящую таблицу из списка — например, Заказы. Если у шаблона есть связанные таблицы, то Access также добавит в базу часто используемые связанные таблицы.

Добавить таблицы в пользовательское веб-приложение можно, импортировав их из внешних источников данных, таких как базы данных Access, файлы Microsoft Excel, любые другие источники, которые используют для доступа к данным интерфейс ODBC, например, SQL Server, текстовые файлы и списки SharePoint (см. рис. 11.2).

Так же, как в локальной базе данных, можно начать разработку и с пустых таблиц. Чтобы добавить новую пустую таблицу, нажмите в пустом приложении кнопку **Таблица (Table)** или ссылку **добавьте новую пустую таблицу** — откроется окно конструктора таблиц. На рис. 11.4 представлено окно конструктора при создании таблицы **Товар**.

В таблице автоматически создается первичное ключевое поле **Код (ID)** с типом данных **Счетчик (AutoNumber)**. Это поле нельзя удалить или изменить, но можно поменять имя этого поля. Определение другого поля первичного ключа невозможно. Целесообразно для поля с кодом товара создать уникальный индекс, используя кнопку ленты конструктора таблиц **Индексы (Indexes)**. Определение уникального индекса обеспечит целостность данных, не позволив ввести в таблицу базы данных товар с одинаковым кодом дважды.

Открытый список **Тип данных (Data Type)** показывает недопустимость использования в таблицах базы данных веб-приложения типов данных **Поле объекта OLE (OLE Object)** и **Вложение (Attachment)**, доступных в базах данных Access для настольных компьютеров.

Новые таблицы веб-приложения отображаются в его списке таблиц. Кроме того, как и все объекты текущего приложения, отображаются они и в области навигации в левой части окна. Для отображения этой области используйте соответствующую кнопку на ленте **Главная (Home)**.

Для ввода, просмотра, удаления данных из таблицы перейдите в режим таблицы. Для этого выберите таблицу в списке приложения и выполните команду контекстного меню **Просмотреть данные**. Если таблица открыта в режиме конструктора, выполните команду **Режим (View)** на ленте конструктора.

В контекстном меню таблицы, наряду с командой **Просмотреть данные**, представлены команды:

- Изменить таблицу**, обеспечивающая переход к изменению структуры таблицы в режиме конструктора;

- Скрыть/Показать**, позволяющая скрыть таблицу при работе в веб-приложении;
- Заблокировать/Разблокировать**, позволяющая установить режим доступа только просмотр к данным таблицы при работе в веб-приложении;
- а также команды **Переименовать** и **Удалить** таблицу.

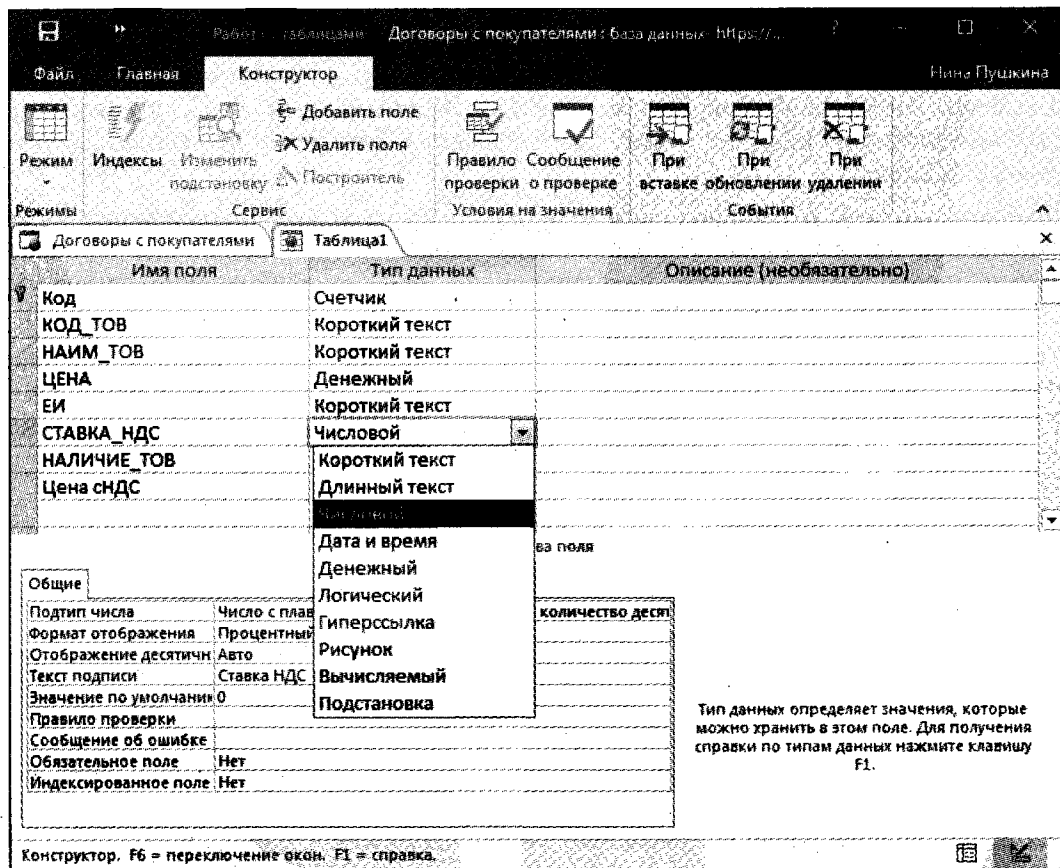


Рис. 11.4. Окно конструктора создания таблицы

Выполните импорт таблиц из созданной ранее базы данных «Поставка товаров». Для перехода к окну добавления таблиц из любого режима нажмите кнопку **Таблица** (Table) на ленте **Главная** (Home). Для импорта таблиц из базы данных Access в области **Создание таблицы из источника данных** нажмите кнопку **Access**. В открывшемся окне **Внешние данные — База данных Access (Get External Data — Access Database)** укажите расположение источника данных.

В следующем окне **Импорт объектов (Import Objects)** выберите таблицы **ДОГОВОР**, **ПОКУПАТЕЛЬ**, **ИСПОЛНИТЕЛЬ**, **ПОСТАВКА_ПЛАН**, **КАЛЕНДАРЬ**.

Мастер создаст новые таблицы в базе данных веб-приложения и импортирует в них данные из таблиц источника. Не импортируются вычисляемые поля, правила про-

верки, значения по умолчанию и поля с такими типами данных, как **Поле объекта OLE (OLE Object)** и **Вложение (Attachment)**.

Для установления связи между таблицами веб-приложения используются поля подстановки. При этом в главной таблице ключевое поле должно иметь тип данных **Счетчик**, а поле подстановки в подчиненной таблице — числовой тип данных. Построение схемы данных в веб-приложении Access не предусмотрено.

При импорте таблиц, имеющих ключи с типом данных, отличным от **Счетчик (AutoNumber)**, автоматически добавляется поле первичного ключа **Код** с типом данных **Счетчик**. Поскольку при этом в подчиненных таблицах отсутствует подходящее для связи поле, создание связи невозможно.

В таблицах базы данных «Поставка товаров» определены реальные ключи, поэтому при их импорте в базу данных веб-приложения добавляются поля первичного ключа с типом данных **Счетчик**. При этом в подчиненной таблице отсутствует поле, которое может быть использовано для его связи с главной таблицей. Измените в созданных импортом таблицах типы данных полей связи и сами данные на числовые и определите для них тип данных **Подстановка (Lookup)**.

ВНИМАНИЕ!

Если в базе данных, из которой выполняется импорт, для таблиц определены ключи с типом данных **Счетчик** и в подчиненных таблицах — соответствующие поля связи с числовым типом данных, в базе данных веб-приложения автоматически будут созданы поля подстановки и обеспечена связь таблиц, аналогичная связям в исходной базе данных.

После завершения импорта таблиц они будут отображены в списке таблиц веб-приложения и области навигации.

Одновременно с созданием таблицы в веб-приложении автоматически выполняется создание форм, предназначенных для представления данных таблицы в браузере. Одна форма предназначена для отображения данных таблицы в виде списка, вторая — в виде таблицы, что и подчеркивается в их именах. Например, формы таблицы **ДОГОВОР** имеют имена **ДОГОВОР Список** и **ДОГОВОР Таблица**.

Задание 11.1. Определение связей при импорте таблиц из базы данных

Сделайте копию базы данных «Поставка товаров». Определите для ее таблиц суррогатные ключи с типом данных **Счетчик**. Для реальных ключей таблиц создайте уникальные индексы. В подчиненных таблицах определите поля связи с главными таблицами и введите в них нужные данные. Создайте схему данных. Создайте новое веб-приложение. Импортируйте в него таблицы из новой базы данных «Поставка товаров». Убедитесь, что в веб-приложении поля связи в подчиненных таблицах получили тип данных **Подстановка (Lookup)** и за счет этого автоматически установились связи между таблицами.

Открытие веб-приложения

Для использования разработанных в Access веб-приложений достаточно иметь веб-браузер и подключение к Интернету. Устанавливать Access на устройства пользователей не требуется. Пользователи для работы с веб-приложением должны иметь в SharePoint учетную запись, поскольку управление безопасностью осуществляется в инфраструктуре SharePoint.

Зарегистрируйтесь на сайте SharePoint. При использовании Office 365, включающем SharePoint, для регистрации используйте адрес <https://login.microsoftonline.com>. В окне регистрации введите логин и пароль учетной записи организации и нажмите кнопку **Войти** (Sign In).

В окне Office 365 откройте страницу **Сайты** и далее выберите сайт, на котором опубликовано приложение, в рассматриваемом примере — **Сайт группы**.

Откройте **Содержимое сайта** и созданное веб-приложение Access **Договоры с покупателями**. Если приложение открывалось ранее, его можно найти в списке **Последние**.

ЗАМЕЧАНИЕ

На сайте SharePoint можно создать документы таких приложений Office, как Word, Excel, PowerPoint, OneNote, или просто перетащить на сайт файлы, в том числе и файл с базой данных Access. Открываются и обрабатываются такие файлы соответствующими приложениями. Только база данных веб-приложения, созданная в Access 2013/2016, доступна в Интернете через браузер.

В браузере откроется окно веб-приложения Access. В нем имена таблиц представлены слева, а формы-представления для каждой таблицы — в верхней части окна справа. Первоначально отображается форма первой в списке таблицы с первой записью в ней. На рис. 11.5 — это первая запись из таблицы **ДОГОВОР**.

В окне веб-приложения доступны для фильтрации, добавления, удаления, редактирования записи всех таблиц приложения через два вида форм: **Список** (List) и **Таблица** (Datasheet).

Если вы обладаете соответствующими правами и на компьютере установлен Access 2016, для перехода от работы с веб-приложением в браузере к его изменению и доработке в Access нажмите кнопку **Настроить в Access** (Settings) в конце заголовка приложения (рис. 11.6).

Внизу экрана отобразится запрос на сохранение или открытие веб-приложения с сайта SharePoint в Access (Do you to open App..accdw (220 bytes) from pnvas-db5-e3be7a9e115.sharepoint.com?) (рис. 11.7).

Кнопка **Открыть** обеспечивает переход в Access. При нажатии кнопки **Сохранить** сохраняется небольшой файл Access (ACCDW). Файл формата ACCDW является XML-файлом. В нем содержится ссылка на сайт SharePoint, и при его открытии в Access выполняется открытие файла базы данных с сервера Microsoft SharePoint и выводится сообщение, представленное на рис. 11.8. Сохранение файла в формате ACCDW дает простую возможность пользователям делиться файлами баз данных,

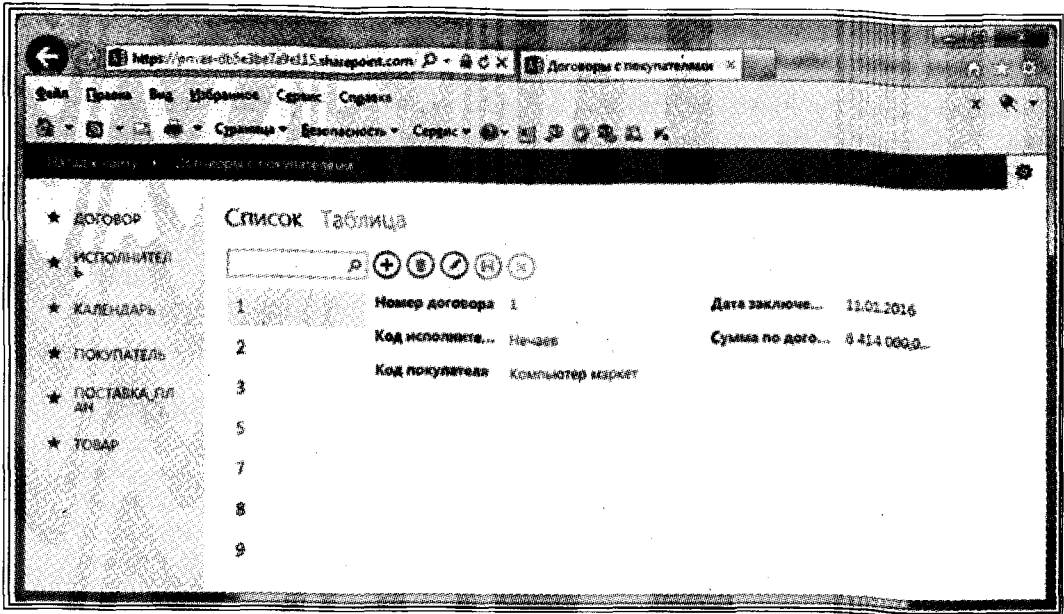


Рис. 11.5. Окно веб-приложения Access 2016 в браузере



Рис. 11.6. Кнопка перехода к настройке веб-приложения в Access

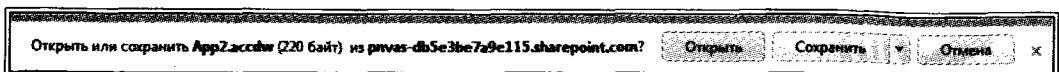


Рис. 11.7. Запрос на выполнение веб-приложения

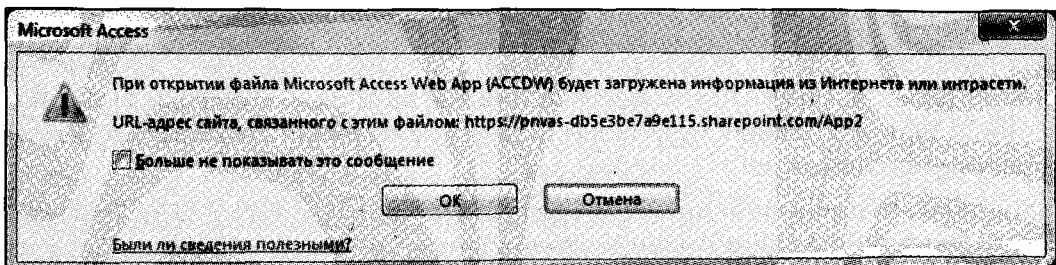


Рис. 11.8. Сообщение при открытии веб-приложения с помощью файла ACCDW

которые хранятся удаленно. Поскольку файлы формата ACCDW — по сути только ярлыки, их выполнение требует соединения с сайтом SharePoint.

После открытия приложения в Access в левой части окна может отображаться область навигации со списком всех его объектов: таблиц, запросов, форм-представлений, макросов. В правой части окна под названием приложения отображается список таблиц и для текущей таблицы — форма **Список (List)** с кнопкой **Изменить** (рис. 11.9). По кнопке **Изменить** на новой вкладке открывается макет соответствующей формы, и можно начать его доработку. В примере откроется макет формы **ПОКУПАТЕЛЬ Список**. Нажав кнопку **Таблица (Table)**, можно аналогичным образом начать доработку структуры таблицы-представления.

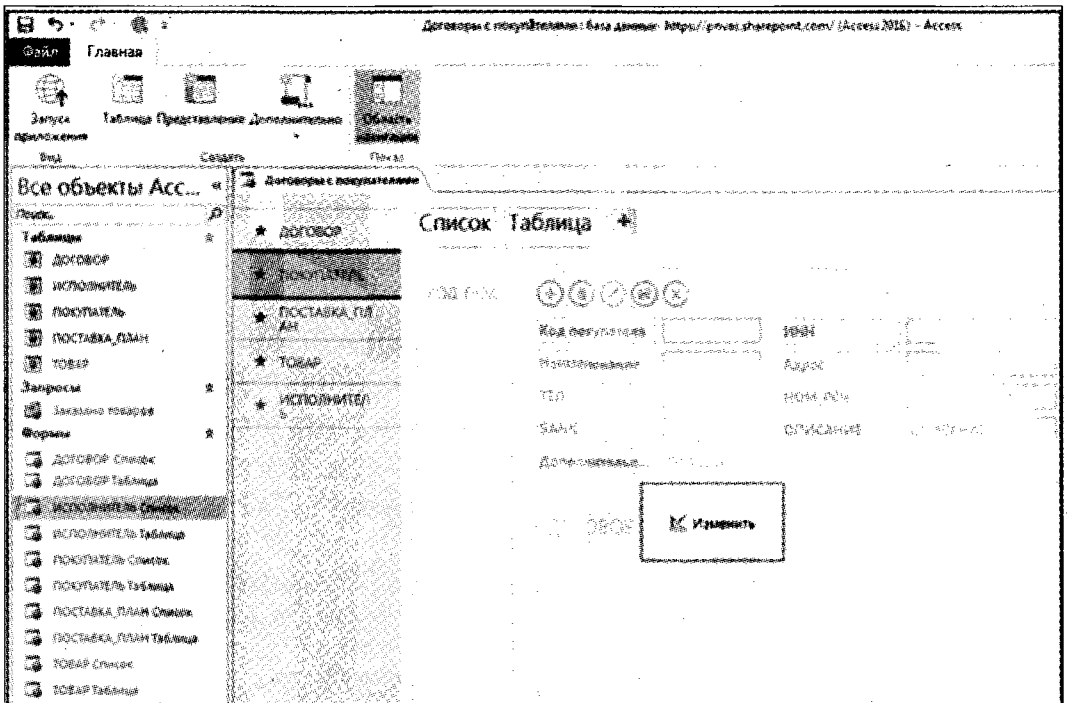


Рис. 11.9. Окно с основной вкладкой разрабатываемого веб-приложения

Для завершения работы с веб-приложением в браузере щелкните в заголовке на ссылке **Назад к сайту**. Для завершения работы с сайтами SharePoint выполните команду **Выход** из списка, открываемого щелчком на имени пользователя, зарегистрированного на сайте.

Если нужно только доработать веб-приложение, открывайте его в Access. При этом, чтобы войти на сайт SharePoint, где хранится приложение, может понадобится переключиться на соответствующую учетную запись или добавить новую учетную запись. Для этого используйте ссылку **Сменить учетную запись (Switch Account)**, размещенную в отображаемой учетной записи. При открытии Access отображается учетная запись, которая была использована последней. В окне **Учетные записи** |

переключить учетную запись (Accounts | Switch Account) выберите или добавьте нужную учетную запись.

Если вы ранее открывали приложение, просто найдите его в списке последних. Имейте в виду, что приложение может приобрести имя App с указанием под ним сайта размещения.

Связывание таблиц

Чтобы создать связи между таблицами, создайте в подчиненных таблицах поля подстановки. Установите связь таблиц ПОСТАВКА_ПЛАН и ДОГОВОР.

ЗАМЕЧАНИЕ

Подстановку можно создать только для полей с числовым типом данных. Если таблицы создавались импортом из базы данных «Поставка товаров», в которой использовались реальные ключи таблиц, поле Код договора в таблице ПОСТАВКА_ПЛАН должно быть изменено. Для него нужно определить числовой тип данных и ввести данные, соответствующие значениям из ключевого поля главной таблицы ДОГОВОР.

Откройте в режиме конструктора таблицу ПОСТАВКА_ПЛАН и выберите для поля Код договора тип данного Подстановка (Lookup). В первом открывшемся окне Создание подстановки (Lookup Wizard) укажите, что поле подстановки должно получать значения из другой таблицы или запроса. В следующем окне (рис. 11.10) выберите таблицу ДОГОВОР, из которой поле подстановки будет получать значения. Выберите значение, которое должно отображаться в поле подстановки: Код договора, и действие, которое обеспечивает поддержание связной целостности при удалении записи из таблицы ДОГОВОР.

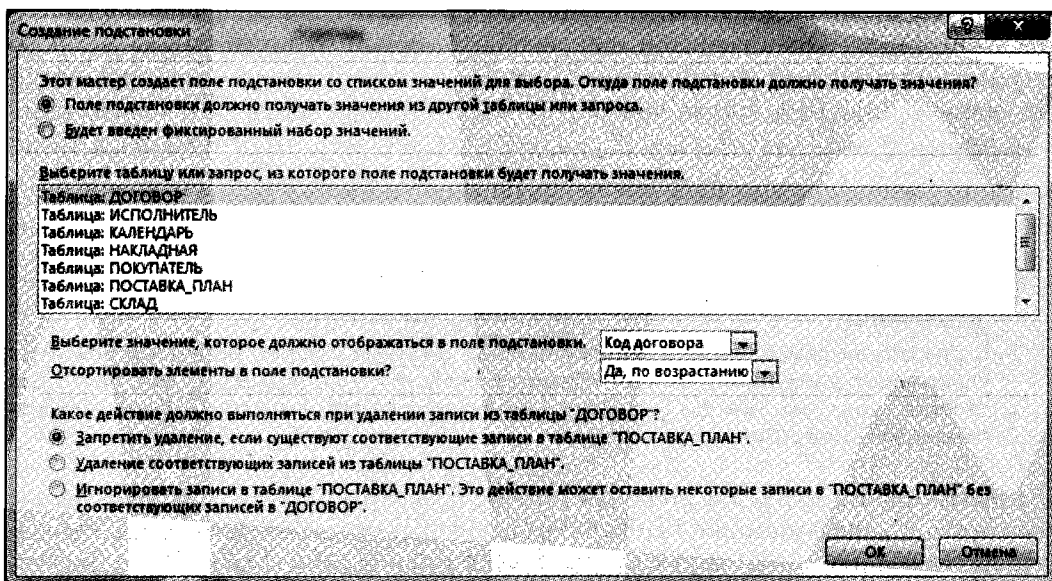


Рис. 11.10. Окно мастера создания поля подстановки, обеспечивающего связь таблиц

После установления связи таблиц в форме-представлении таблицы ДОГОВОР автоматически появится подчиненная форма со сведениями из подчиненной таблицы ПОСТАВКА_ПЛАН, а в списке полей таблица ПОСТАВКА_ПЛАН будет объявлена связанной. Убедитесь в этом, открыв форму ДОГОВОР Список (рис. 11.11).

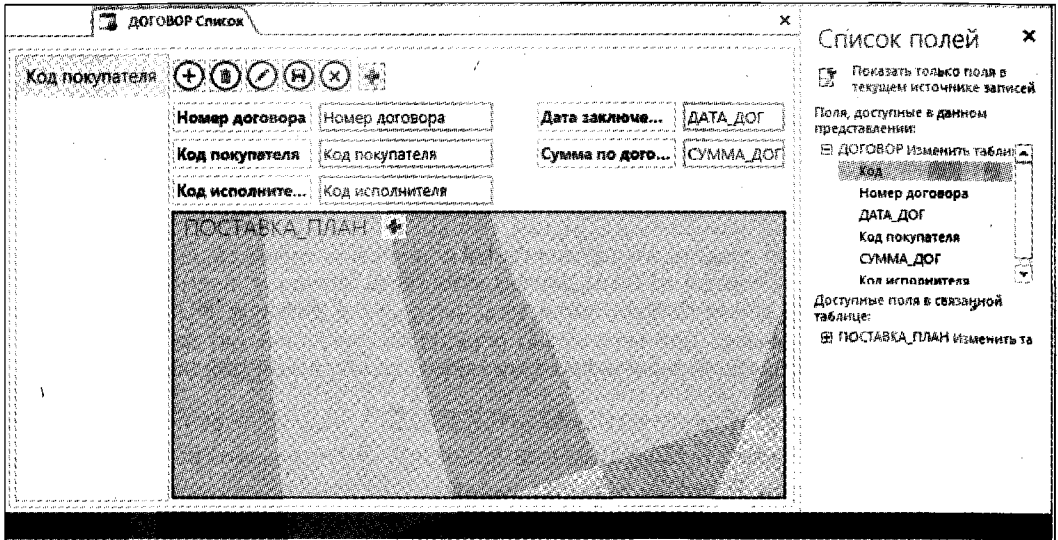



Рис. 11.11. Макет формы-представления после установления связи таблиц ДОГОВОР и ПОСТАВКА_ПЛАН

В главной части формы ДОГОВОР Список выделяется основное поле. В примере это поле Код покупателя, в котором при работе с формой будет отображаться список всех значений из этого поля (см. рис. 11.11). Выбором из этого списка нужного значения открывается соответствующая запись договора. Если нужно просматривать записи, используя вместо кодов покупателей номера договоров, можно заменить основное поле Код покупателя на поле Номер договора. Для этого достаточно выделить поле Код покупателя, нажать появившуюся кнопку  Данные и в открывшемся окне в поле Основной выбрать вместо поля с кодом покупателя поле с номером договора.

ЗАМЕЧАНИЕ

В разрабатываемых формах веб-приложения свойства каждого раздела и элемента управления размещены не на страницах свойств, а на выносках для настройки свойств возле них. Для большинства элементов доступны выноски Данные, Форматирование, Действия.

Для просмотра и работы с формой достаточно перейти в браузер, т. к. приложение уже готово к использованию. Его публикация выполняется автоматически. Выполните на ленте Главная (Home) команду Запуск приложения (Launch App). Зарегистрируйтесь на сайте вашего приложения и откройте его. В браузере для таблицы ДОГОВОР на вкладке Список отобразится форма для работы с договором и под-

чиненной таблицей с заказами. На рис. 11.12 основным полем, по значениям которого осуществляется переход к записям о договорах, осталось поле Код покупателя. Из списка этого поля видно, что договоры заключались только покупателями 1 и 2, причем первый заключил 3 договора, а второй 5.

★ ДОГОВОР

★ ПОКУПАТЕЛЬ

★ ПОСТАВКА_ПЛАН

★ ТОВАР

Список Таблица

Номер договора 1

Дата заключе... 11.01.2016

Код покупателя 1

Сумма по дого... 6414 000,00р.

Код исполните... 6

ПОСТАВКА_ПЛАН

КОДУЙТЕЛ	СРОК ПОСТАВКИ	Мин. партии	Количество
1	1	10	305
1	2	5	200
1	3	5	300
2	1	10	10
2	3	5	100

Добавить ПОСТАВКА_ПЛАН

Рис. 11.12. Отображение формы-представления ДОГОВОР Список в браузере

ПОСТАВКА_ПЛАН

ДАННЫЕ

Источник данных ПОСТАВКА_ПЛАН

Связанное поле Номер договора

Первое поле Код товара

Первый заголовок Код товара

Второе поле СРОК ПОСТ

Второй заголовок Срок поставки

Третье поле СУММА ПОСТ

Третий заголовок Стоимость

Четвертое поле КОЛ ПОСТ

Четвертый заголовок Количество

Всплывающее окно ПОСТАВКА_ПЛАН Спис

Поле сортировки Код товара

Порядок сортировки По возрастанию

Рис. 11.13. Окно выбора 4-х полей для отображения в подчиненной части формы ДОГОВОР Список

Вернитесь в Access и в подчиненной форме **ПОСТАВКА_ПЛАН** формы **ДОГОВОР Список** замените поле **Мин. партия** на **Стоимость** заказанного товара. Для этого нажмите кнопку **Изменить**, выделите заголовок элемента формы-представления **ПОСТАВКА_ПЛАН** и нажмите значок **Данные**.

Заполните поля, как показано на рис. 11.13, — теперь в браузере в подчиненной форме будут отображаться выбранные в области **ДАННЫЕ** поля.

Не забывайте сохранять выполненные в Access изменения и обновлять отображаемые в браузере данные. Нажатие кнопки **Сохранить** на панели быстрого доступа сохраняет изменения на сервере SharePoint Server или сайте Office 365. Необходимость публикации как дополнительного действия отпадает.

Задача 11.2. Создание подчиненной формы

Установите связь таблиц **ПОСТАВКА_ПЛАН** и **ТОВАР**. Для этого измените в таблице **ПОСТАВКА_ПЛАН** значения и тип данных поля связи **Код товара** на **Подстановка (Lookup)**. Убедитесь, что после установления связи в форме **ТОВАР Список** появилась подчиненная форма **ПОСТАВКА_ПЛАН**.

Вычисление итоговых значений

Вычислите итоговое значение для поля **КОЛ_ПОСТ** на вкладке подчиненной формы **ПОСТАВКА_ПЛАН** в форме **ТОВАР Список**. При выделении элемента управления **Вкладка** появляются кнопки **Данные**, **Форматирование** и **Вычисление**. Воспользовавшись командой **Вычисление** для подсчета итоговых значений вкладки, подсчитайте общее количество заказанного товара. В открывшемся окне **ВЫЧИСЛЕНИЕ** выберите поле и одну из трех статистических функций (**Сумма**, **Количество**, **Среднее**) для вычисления по этому полю (рис. 11.14).

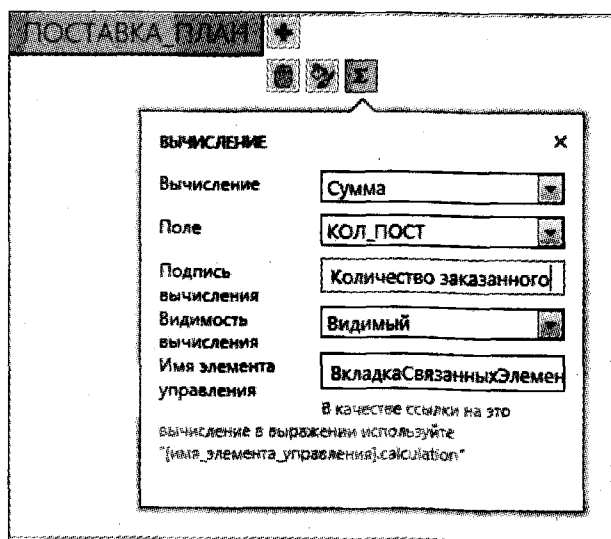


Рис. 11.14. Добавление итогов к вкладке **ПОСТАВКА_ПЛАН** в форме **ТОВАР Список**

★ ДОГОВОР

★ ПОКУПАТЕЛЬ

★ ПОСТАВКА_ПЛАН

★ ТОВАР

★ ИСПОЛНИТЕЛЬ

Список Таблица

Фильтрация списка...

Наименование ... CD-ROM Panasonic IDE

Единица изме... штуки

Наличие товара

Остаток 10

Цена 800,00р.

Ставка НДС 30%

Цена с НДС 1 040,00р.

Стоимость 8 000,00р.

Зв. Карта Genius Liv

Корпус MiniTower

Миникомпьютер

Модем Genius ext

Монитор 17LG

ПОСТАВКА_ПЛАН

Номер договора	Срок поставки	Мень. партия	Количество
1	1	0	710
3	1	10	40

Добавить ПОСТАВКА_ПЛАН

Количество заказанного: 750

Рис. 11.15. Отображение итогового значения по полю Количество

Результат этого вычисления в браузере представлен на рис. 11.15.

Задание 11.3. Создание в форме поля с итоговыми значениями


Обеспечьте вывод итоговой суммы по всем договорам исполнителя. Используйте в форме **ИСПОЛНИТЕЛЬ Список** на вкладке **ДОГОВОР** кнопку **Вычисление**. Вкладка **ДОГОВОР** отобразится в форме при установлении связи таблицы **ДОГОВОР** с таблицей **ИСПОЛНИТЕЛЬ**.

Создание в форме ссылок на связанную запись главной таблицы

Установите связь таблиц **ДОГОВОР** и **ИСПОЛНИТЕЛЬ**. Откройте в режиме конструктора таблицу **ДОГОВОР** и выберите для поля **Код исполнителя** тип данного **Подстановка (Lookup)**. В открывшемся окне **Создание подстановки (Lookup Wizard)** выберите таблицу **ИСПОЛНИТЕЛЬ**, из которой поле подстановки будет получать значения. Выберите значение, которое должно отображаться в поле подстановки: фамилия и действие, которое должно выполняться при удалении записи из таблицы **ИСПОЛНИТЕЛЬ**, — **Запретить удаление**, если существуют соответствующие записи в таблице **ДОГОВОР** (Prevent delete if there are corresponding records in the "ДОГОВОР" table?).

Установленная связь позволит дополнить форму **ИСПОЛНИТЕЛЬ Список** сведениями из подчиненной таблицы **ДОГОВОР**. Кроме того, эта связь позволит в форме **ДОГОВОР Список** создать ссылку на связанную запись в главной таблице **ИСПОЛНИТЕЛЬ**.

Откройте форму **ДОГОВОР Список**. Если эта форма содержит поле **Код исполнителя** до изменения его типа данных на **Подстановка**, удалите его и затем заново разместите его в форме, перетащив из списка полей. Выделите вновь вставленное

поле Код исполнителя, нажмите значок  **Данные**. Автоматически значение поля Код исполнителя преобразуется в ссылку, по которой будет отображаться соответствующая запись об исполнителе с подчиненным списком всех договоров, сопровождаемых этим исполнителем. На рис. 11.16 показано, как при этом заполняются поля области **ДАнные**.

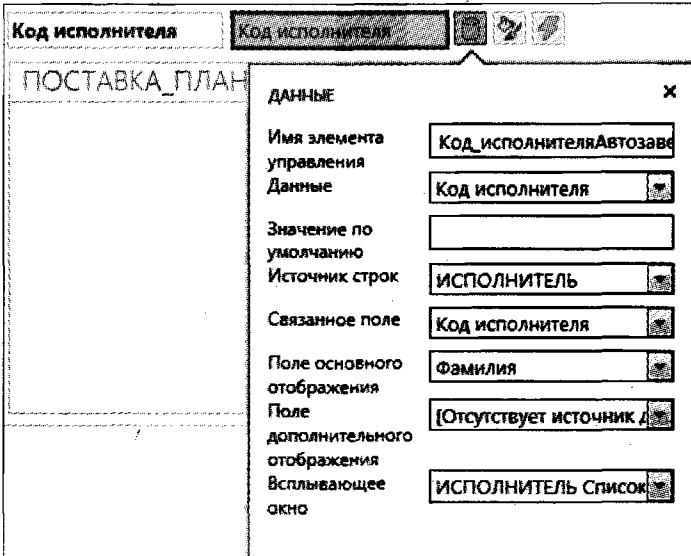


Рис. 11.16. Подготовка ссылки, обеспечивающей вывод данных об исполнителе

Перейдите к работе с формой в браузере. Щелкните по ссылке на исполнителя (на рис. 11.17 это фамилия исполнителя текущего договора **Нечаев**) — отобразится запись со сведениями об исполнителе и всех договорах, с которыми он работает.

Для просмотра в браузере всех записей таблицы одновременно выберите ее из списка и нажмите вкладку **Таблица** (см. рис. 11.12 или 11.17). Макет таблицы размещается на вкладке **Таблица** (см. рис. 11.9) при ее создании в Access. При необходимости изменить структуру макета, например, дополнить таблицу данными из связанных таблиц, работая в Access, выберите таблицу и нажмите вкладку **Таблица** — отобразится макет таблицы и список полей. Дополните макет таблицы **ДОГОВОР Таблица** полями **Имя**, **Отчество** и **Рабочий телефон** из связанной таблицы **ИСПОЛНИТЕЛЬ**, перетаскивая их в макет из списка полей.

Задание 11.4. Добавление в форму ссылки для получения сведений о покупателе

Создайте в форме **ДОГОВОР Список** ссылку для отображения сведений о покупателе. Для этого определите связь таблиц **ДОГОВОР** и **ПОКУПАТЕЛЬ** и, выполнив в форме команду **Данные** для поля **Код покупателя**, заполните открывшееся окно.

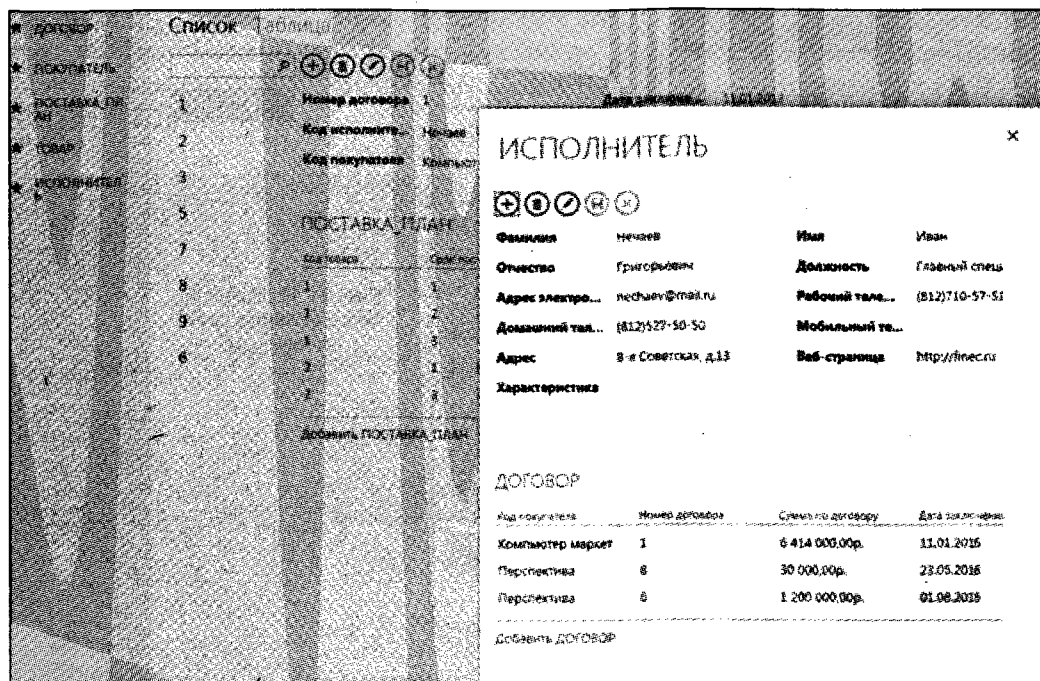



Рис. 11.17. Отображение формы ДОГОВОР Список сведений об исполнителе

Создание поля со списком

Создайте в форме ДОГОВОР Список поле со списком, обеспечивающее ввод кода покупателя выбором из списка. Выберите форму и нажмите кнопку **Изменить**. На ленте конструктора представлений щелкните на элементе управления **Поле со списком** (Combo box). Для нового пустого поля со списком выполните команду **Данные** и заполните поля выноски, как показано на рис. 11.18. В поле **Данные** выберите **Код покупателя** из таблицы ДОГОВОР, в поле **Источник строк** — **ПОКУПАТЕЛЬ**, а в **Связанном поле** — **Код** из таблицы **ПОКУПАТЕЛЬ**.

Сохраните изменения в форме и для работы с новым полем перейдите в браузер. При просмотре данных в форме ДОГОВОР Список поле со списком отображается как простое поле. Работа со списком этого поля доступна в режиме добавления и изменения записей в главной форме. На рис. 11.19 форма представлена в режиме изменения записи.

Обратите внимание, что в форме слово **Список** изменено на **Договоры с покупателями**, надпись поля **Номер договора** изменена на **ДОГОВОР**, надпись поля **Дата заключения** на **от**. Изменен и заголовок подчиненной формы **ПОСТАВКА_ПЛАН**. Изменение подписей выполняется непосредственно в элементах управления. Для изменения заголовка подчиненной формы можно использовать команду **Форматировать**. Изменение в главной форме слова **Список** может быть выполнено командой **Переименовать** на выноске этого элемента **Параметры/Действия** .

КОД ПОКУПАТЕЛЯ

ДАНИЕ X

Имя элемента управления: Поле со списком

Данные: Код покупателя

Значение по умолчанию: []

Тип источника строк: Таблица или запрос

Источник строк: ПОКУПАТЕЛЬ

Связанное поле: Код

Поле отображения: НАИМ_ПОК

Всплывающее окно: []

Родительский элемент управления: []

Рис. 11.18. Выноска со свойствами поля со списком

Договоры с покупателями Таблица

[] [P] [+] [v] [A] [H] [X]

1	ДОГОВОР	[1] от	11.01.2016
2	Код исполните...	Нечаев	
3	Код покупателя	Компьютер маркет	
5		Сумма по дого...	6 414 000,00р
7	Заказы		
8	Код товара	Срок поставки	Стоимость
9	CD-ROM Panasoni...	1	568 000,00р.
6	FDD 3,5	1	5 000,00р.
	FDD 3,5	3	50 000,00р.
	HDD Maxtor 120GB	1	112 000,00р.
	Корпус MiniTower	1	2 135 000,00р.
	Добавить Заказы		Общая сумма: 6 414 000,00р.

Рис. 11.19. Отображение поля со списком в браузере

Замените в подчиненной форме **Заказы** (см. рис. 11.19) поле Код товара на поле со списком. Перейдите в браузер, нажмите ссылку **Добавить Заказы**. При добавлении записи в подчиненной форме в поле со списком Код товара отображаются все товары (рис. 11.20).

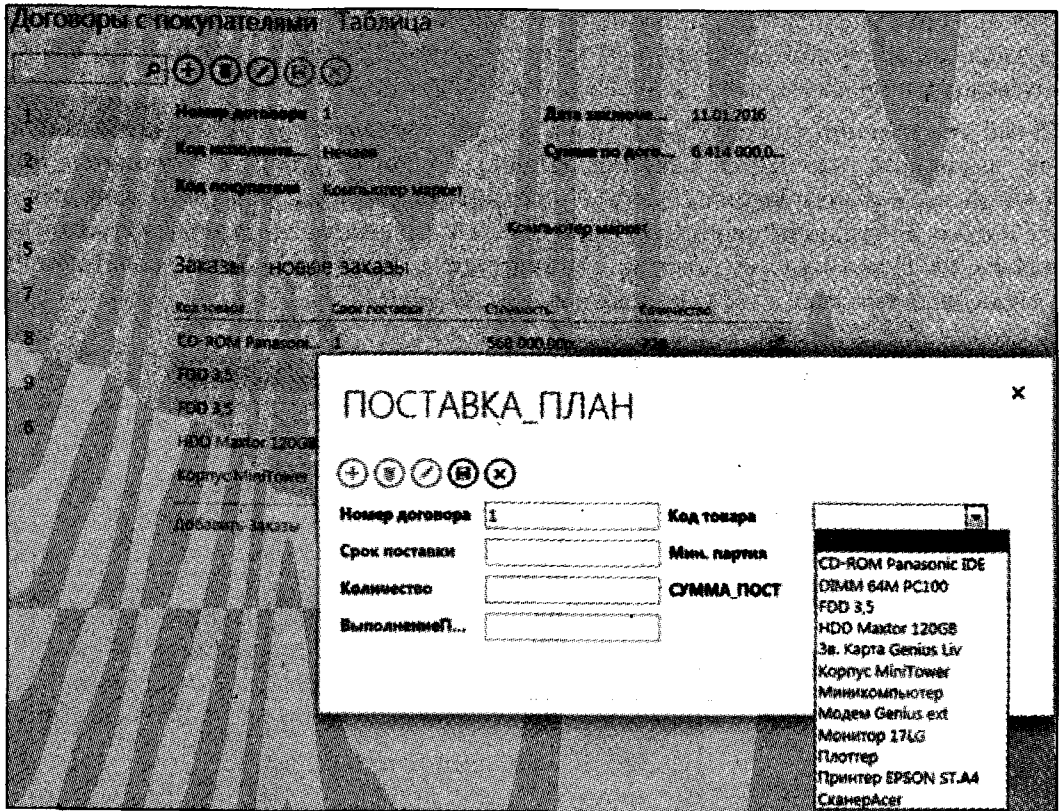


Рис. 11.20. Окно добавления записи в таблицу подчиненной формы ПОСТАВКА_ПЛАН

Задача 11.5. Добавление в форму поля со списком

Добавьте поле со списком в одну из главных форм приложения. Убедитесь, что в браузере работа со списком этого поля доступна в режиме добавления и изменения записей в главной форме. Только в этих режимах поле отображается как поле со списком, иначе — как простое поле.

Создание и использование запросов

В базе данных веб-приложения, как и в локальной базе данных, с помощью запросов на выборку можно сформировать пользовательское представление о данных. Запрос строится на основе одной или нескольких взаимосвязанных таблиц. Результатом выполнения такого запроса становится новая таблица, которая существует до

закрытия запроса. Структура таблицы запроса определяется выбранными из одной или нескольких взаимосвязанных таблиц полями. Записи формируются путем объединения записей таблиц, на которых построен запрос. Способ объединения записей таблиц указывается при определении их связи в запросе. Условия отбора, сформулированные в запросе, позволяют фильтровать записи, составляющие результат объединения таблиц. В запросах базы данных веб-приложения, так же как в запросах локальной базы данных, могут содержаться вычисляемые поля, выполняться группировка записей, использоваться параметры запроса.

В веб-приложении запросы создаются в режиме конструктора. Для создания запроса выполните на ленте Главная (Home) в списке кнопки Дополнительно (Advanced) команду Запрос (Query) — в окне конструктора откроется бланк запроса и окно Добавление таблицы (Show Table), позволяющее выбрать таблицы и запросы, на которых нужно построить новый запрос (рис. 11.21). Лента конструктора запросов отображает его функциональные возможности.

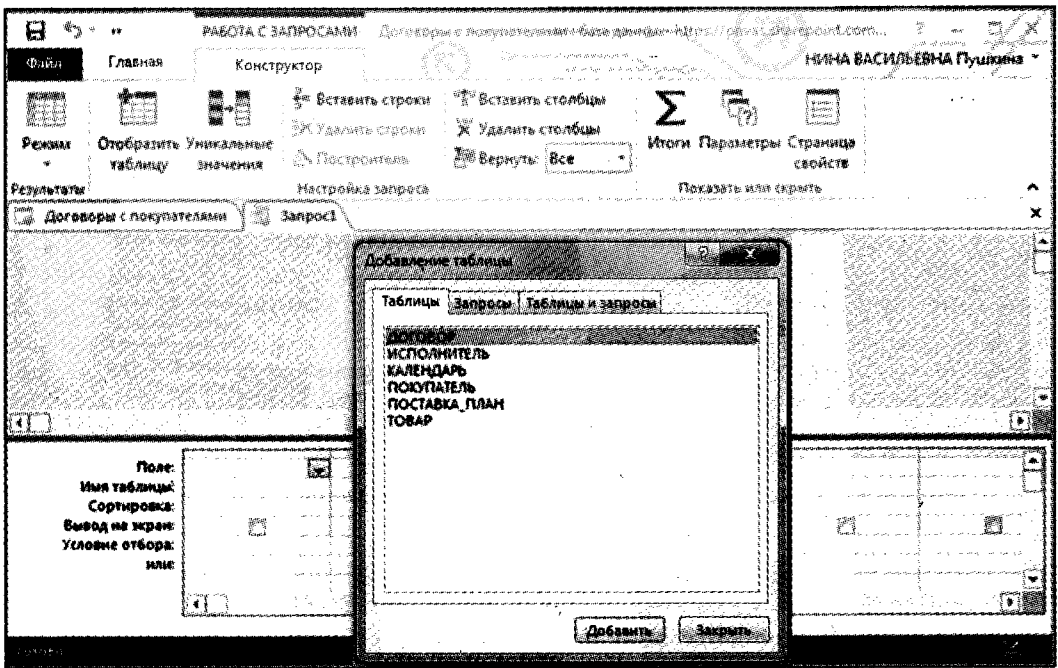


Рис. 11.21. Первое окно создания запроса в базе данных веб-приложения

Если в приложении была установлена связь таблиц, она автоматически отобразится в бланке запроса.

Как и в локальной базе данных, после добавления таблиц в запрос между ними могут быть установлены и не предусмотренные ранее связи. Для каждой из этих таблиц можно установить параметры объединения.

В базе данных веб-приложения **Договоры с покупателями** создайте запрос, в котором подсчитайте, какое количество и на какую сумму каждого из товаров было

заказано по всем договорам. Для этого в окне **Добавление таблицы** выберите таблицу **ПОСТАВКА_ПЛАН**. Заполните бланк запроса, как показано на рис. 11.22, и при сохранении дайте ему имя **Заказано товаров**.

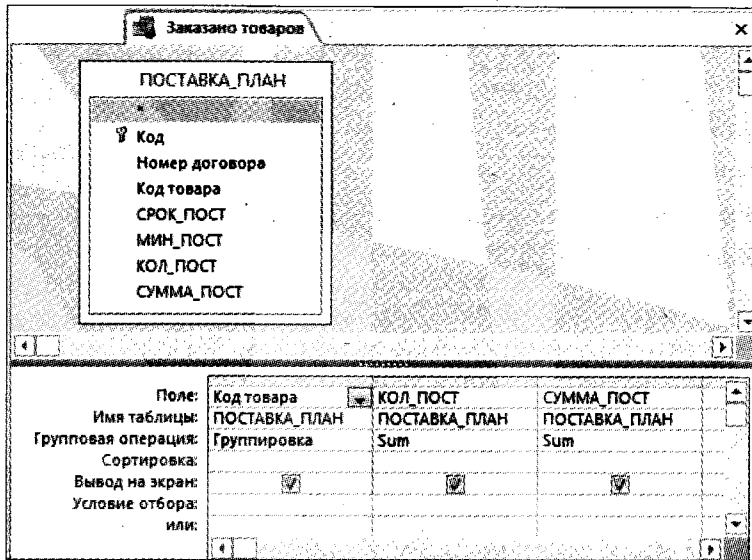




Рис. 11.22. Запрос для подсчета общего количества и стоимости заказанных товаров

Используйте запрос в форме-представлении **ТОВАР Список** для отображения результатов выполнения запроса: общего количества и суммы заказанного товара. Откройте в Access соответствующий макет формы и дополните его элементом управления для связанных элементов. Для этого достаточно щелкнуть на расположенном на ленте конструктора представления значке  **Элемент управления для связанных элементов (Related items Control)**. Далее выделите новый элемент управления и, щелкнув на значке  **Добавление вкладки**, определите ее заголовок и источник данных (рис. 11.23).

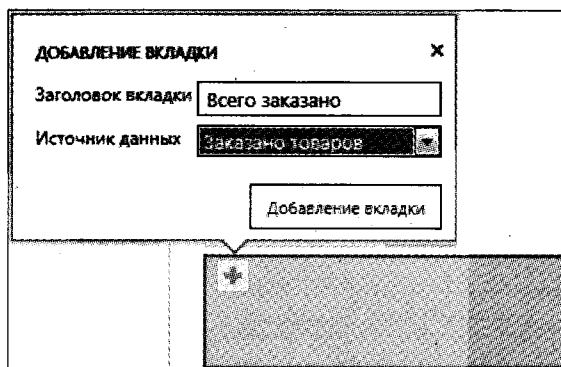


Рис. 11.23. Определение источника данных для вкладки

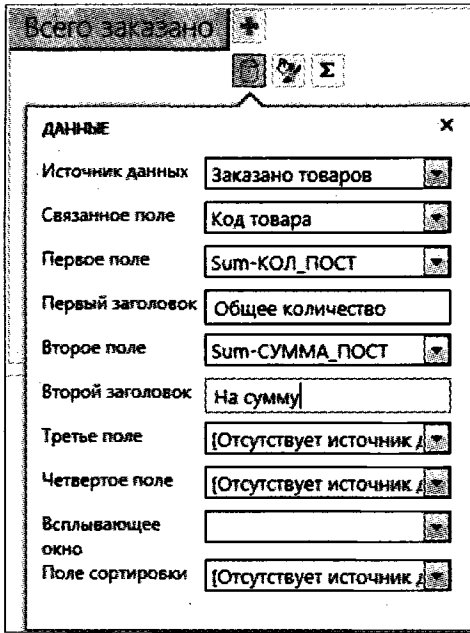


Рис. 11.24. Определение полей источника данных для вывода на вкладке

Выделив элемент **Всего заказано**, нажмите кнопку **Данные** и в открывшемся окне выберите поля, которые нужно вывести на вкладке (рис. 11.24), — теперь в форме **ТОВАР Список** для каждого выбираемого товара будет отображаться не только список заказов, но и новая вкладка с общими данными по этому товару (рис. 11.25).

- ★ ДОГОВОР
- ★ ПОКУПАТЕЛЬ
- ★ ПОСТАВКА_ПЛАН
- ★ ТОВАР
- ★ ИСПОЛНИТЕЛЬ

Список Таблица

Фильтрация списка...

CD-ROM Panasonic IDE

DIMM 64M PC100

FDD 3,5

HDD Maxtor 120GB

Зв. Карта Genius Liv

Корпус MiniTower

Миникомпьютер

Модем Genius ext

Монитор 17LG

Плоттер

Принтер EPSON ST.A4

СканерAcer

Наименование ... FDD 3,5

Единица изме... коробка

Наличие товара

Остаток 963

Цена 500,00р.

Ставка НДС 6%

Цена с НДС 530,00р.

Стоимость 481 500,00р.

ПОСТАВКА_ПЛАН

номер договора	Срок поставки	Минь партия	Количество
1	1	10	10
1	3	5	100
2	2	120	240
2	4	0	10
3	2	34	80

Добавить ПОСТАВКА_ПЛАН

Всего заказано

Общее количество	На сумму
740	370000

Рис. 11.25. Форма ТОВАР Список с новой вкладкой Всего заказано в браузере

Для формы **ТОВАР Список** новая вкладка может быть добавлена к существующей вкладке **ПОСТАВКА_ПЛАН** без включения в нее дополнительного связанного элемента управления. Для этого предназначена кнопка со знаком + на вкладке **ПОСТАВКА_ПЛАН** (см. рис. 11.11). В отличие от вкладки дополнительного элемента управления, ее данные будут отображаться не постоянно, а по требованию, т. е. при переходе на нее.

Создание макросов

Для реализации более сложной логики приложений, автоматизации выполнения задач, связи различных объектов, добавления функциональных возможностей необходимо использовать доступный при разработке веб-приложений язык макросов. С помощью макросов в веб-приложении можно, например, открыть другую форму, применить фильтр или создать новую запись.

В веб-приложении могут быть созданы макросы трех типов:

- внедренные макросы* — привязаны непосредственно к событиям в элементах управления или формах-представлениях и сохраняются в них;
- изолированные макросы* — представлены наряду с другими объектами приложения в области навигации в группе **Макросы**, однако их запуск возможен только из другого макроса;
- макросы данных* — отличаются от изолированных макросов набором допустимых макрокоманд и служат для добавления или изменения данных в таблицах.

Внедренные макросы запускаются тогда, когда в форме-представлении или его элементе управления происходят определенные события — например, открытие представления, переход к другой записи, нажатие кнопки или обновление значения в поле. Такие макросы являются частью формы, в которую они внедрены.

Создайте внедренный макрос, который при изменении исполнителя договора отменит это действие и выведет соответствующее сообщение. Откройте макет формы **ДОГОВОР Список**. Выделите элемент управления — поле **Код исполнителя** — с событием которого нужно связать макрос. Нажмите кнопку **Действия** — откроется диалоговое окно **Действия** с событиями для поля **Код исполнителя**, к которым можно привязать макрос (рис. 11.26).

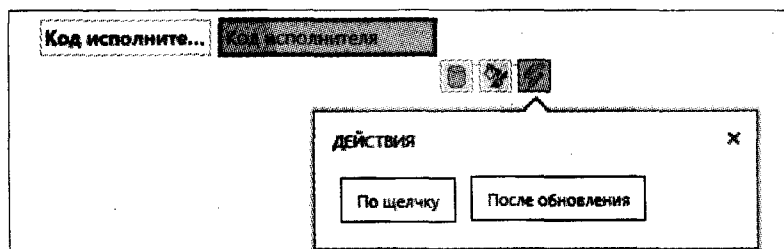


Рис. 11.26. События, доступные для поля формы

Выберите событие **После обновления** — откроется пустой макрос в режиме конструктора. Добавьте в него макрокоманды **ОтменитьЗапись** (UndoRecord) из группы **Операции ввода/вывода** (Data Entry Operations) и **Окно сообщения** (MessageBox) из группы **Команды интерфейса пользователя** (User Interface Commands). Введите текст сообщения: **Изменение исполнителя недопустимо!**

При повторном открытии окна **Действия** кнопка события, для которого существует внедренный макрос, будет окрашена зеленым цветом. Щелчком на ней макрос откроется для редактирования.

Теперь при работе в браузере операция изменения значения в поле **Код исполнителя** будет отменена и выведется **Сообщение с веб-страницы** (Message from webpage) (рис. 11.27).

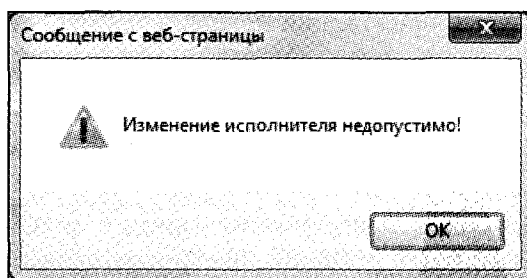


Рис. 11.27. Сообщение об обновлении поля

События формы-представления (рис. 11.28), к которым можно привязать внедренный макрос, открываются при выделении формы щелчком мыши за ее пределами и нажатии кнопки **Действия**.

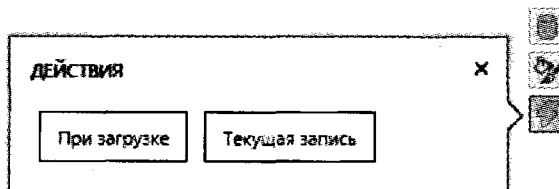


Рис. 11.28. Окно действий для формы

Создание изолированного макроса выполняется командой **Главная | Дополнительно | Макрос** (Home | Advanced | Macro). При этом также открывается пустой макрос в режиме конструктора, где можно добавить макрокоманды. При сохранении изолированного макроса вводится **Имя макроса** (Macro Name). Вызывается изолированный макрос из другого макроса макрокомандой **ЗапускМакроса** (RunMacro).

Макросы данных создаются командой **Главная | Дополнительно | Макросы данных** (Home | Advanced | Data Macro).


Для ссылок на поля в макросе используется формат `[ИмяТаблицы].[ИмяПоля]`.

Задание 11.6. Создание в форме изолированного макроса

Создайте и организуйте выполнение изолированного макроса для открытия формы **ДОГОВОР Список** с заданным номером договора.

Создание отчетов

Накопленная в базе данных веб-приложения информация должна обрабатываться, обеспечивая решение задач предметной области и вывод результатов в различных отчетах. Для такой расширенной обработки данных веб-приложения в Access 2013/2016 предусмотрена функция создания локального приложения, подключенного к базе данных веб-приложения через интерфейс ODBC. Создание такого приложения сводится к выполнению команды **Создание отчетов** (Create Reports), доступной на странице **Сведения** (Info), открывающейся выбором значка **Файл** (см. рис. 11.3).

В локальном приложении Access таблицы базы данных веб-приложения отображаются как связанные и помечены значком, например, * договор. Приложение можно дополнить локальными таблицами. Связанные таблицы могут использоваться в объектах локального приложения Access так же как локальные и вместе с ними. Связь локального приложения с базой данных веб-приложения обеспечивает автоматическое отображение новых данных в отчете при добавлении или изменении их через веб-приложение.

Задание 11.7. Создание отчета на данных базы веб-приложения

Создайте отчет с данными о договорах фирмы, сгруппированных по покупателям. Вид этого отчета представлен на рис. 7.9 в главе 7.

Задание 11.8. Создание составного отчета на данных базы веб-приложения

Создайте составной отчет с данными об отгрузке товаров. Вид этого отчета представлен на рис. 7.41 в главе 7.

Контрольные вопросы

1. На сайтах какого приложения может располагаться веб-приложение?
2. Где хранится объект базы данных веб-приложения?
3. Под управлением какой операционной системы может работать Access 2016?
4. Какой тип данного допустим для ключевого поля в таблице веб-приложения?
5. Какие объекты автоматически создаются в веб-приложении при добавлении таблицы?
6. Как реализуется связь таблиц в веб-приложении?

7. Допустимо ли использование в таблицах веб-приложения полей с типами данных **Длинный текст** (Long Text), **Поле объекта OLE** (OLE Object), **Вложение** (Attachment)?
8. Какого типа запросы могут быть созданы в веб-приложении?
9. Допустимо ли в запросах веб-приложения использование вычисляемых полей, группировки, параметров запроса?
10. Какого типа макросы могут создаваться в веб-приложении?
11. Как в Access выбрать подключение к нужному сайту?
12. Какие события определены в веб-приложениях для поля с данными?
13. Какие события определены в веб-приложениях для формы-представления?
14. Данные из какой таблицы отображаются в поле со списком?
15. Какой командой обеспечивается создание локального приложения Access, связанного с базой данных веб-приложения?
16. Какой интерфейс обеспечивает связь локального приложения Access с базой данных веб-приложения?

Ответы

1. На сайтах корпоративного сервера SharePoint со службами Access или сайтах Office 365.
2. На SQL Server 2012 и выше или SQL Windows Azure.
3. Windows 7 и выше.
4. Только **Счетчик** (AutoNumber).
5. Формы двух типов: **Список** (List) и **Таблица** (Datasheet) для представления данных в браузере.
6. Через определение в подчиненной таблице для поля связи типа данных **Подстановка**.
7. **Длинный текст** (Long Text) — да. Два других — нет.
8. Запросы на выборку.
9. Да.
10. Внедренные, изолированные и макросы данных.
11. Чтобы подключиться к сайту с нужным веб-приложением, надо выбрать или добавить соответствующую учетную запись.
12. **По щелчку** и **После обновления**.
13. **При загрузке** и **Текущая запись**.
14. Главной по отношению к таблице, в которой содержится поле со списком.
15. Командой **Отчет о моих данных** (Report on my Data), доступной на странице **Сведения** (Info) вкладки **Файл**.
16. ODBC.

ПРИЛОЖЕНИЕ 1

Структура таблиц базы данных «Поставка товаров»

Таблицы справочных данных

Таблица П1.1. Описание свойств полей таблицы ТОВАР

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Формат	Число дес. знаков	Подпись поля
КОД_ТОВ	Ключ простой	Да	Текстовый	5			Код товара
НАИМ_ТОВ		Нет	Текстовый	25			Наименование товара
ЦЕНА		Нет	Денежный		Денежный	2	Цена
ЕИ		Нет	Текстовый	8			Единица измерения
СТАВКА_НДС		Нет	Числовой	Одинарное с плавающей точкой	Процентный	0	Ставка НДС
ФОТО		Нет	Поле объекта OLE	До 1 Гбайт			Фото товара
НАЛИЧИЕ_ТОВ		Нет	Логический	1 бит	Да/Нет		Наличие товара на складе
ВЛОЖЕНИЯ		Нет	Вложение	700 Кбайт			Поле вложений
Цена с НДС		Нет	Вычисляемый		Денежный	2	

Требования к определению полей табл. П1.1:

1: Для поля СТАВКА_НДС нужно предусмотреть:

- **Условие на значение:** $\geq 0,05$ And $\leq 0,35$;
- **Сообщение об ошибке:** «Ставка НДС должна быть $\geq 5\%$ и $\leq 35\%$ ».

2. Для поля ЦЕНА нужно предусмотреть:
 - **Условие на значение:** ≥ 0 And ≤ 35000 ;
 - **Сообщение об ошибке:** «Цена должна быть ≥ 0 и ≤ 35000 ».
3. В поле Цена с НДС записать выражение $[ЦЕНА] + [СТАВКА_НДС] * [ЦЕНА]$.
4. Для этой таблицы и остальных — если для поля указан признак первичного ключа, это означает следующее:
 - когда ключ «простой», поле надо выделить и присвоить признак ключа (нажатием кнопки **Ключевое поле**), при этом в свойстве **Индексированное поле** автоматически установится значение **Да (Совпадения не допускаются)**;
 - когда ключ «составной», надо выделить все поля, образующие первичный ключ, и только после этого присвоить признак ключа. При этом в свойстве **Индексированное поле** для каждого поля, входящего в ключ, нужно оставить значение по умолчанию **Нет**.
5. Неуказанные в таблице параметры в конструкторе таблиц должны сохранить значение по умолчанию.

Таблица П1.2. Описание свойств полей таблицы СКЛАД

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Подпись поля
КОД_СКЛ	Ключ простой	Да	Текстовый	5	Номер склада
КОД_Ф		Нет	Текстовый	5	Код фирмы
НАИМ_СК		Нет	Текстовый	20	Наименование склада
ОТВ_ЛИЦО		Нет	Текстовый	20	Ответственное лицо
АДРЕС_СК		Нет	Текстовый	20	Адрес склада

Таблица П1.3. Описание свойств полей таблицы ПОКУПАТЕЛЬ

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Подпись поля
КОД_ПОК	Ключ простой	Да	Текстовый	5	Код покупателя
ИНН		Нет	Текстовый	12	
НАИМ_ПОК		Нет	Текстовый	20	Наименование
АДРЕС_ПОК		Нет	Текстовый	20	Адрес
ТЕЛ		Нет	Текстовый	10	Телефон
НОМ_РСЧ		Нет	Текстовый	20	Номер расчетного счета
БАНК		Нет	Текстовый	10	

Таблица П1.3 (окончание)

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Подпись поля
ОПИСАНИЕ		Нет	Поле Длинный текст		
WEB_АДРЕС		Нет	Гиперссылка		

Требования к заполнению табл. П1.3:

- Для поля ТЕЛ следует задать маску ввода: \ (999\)\ 000\ -0099.
- Для поля ИНН нужно выполнить следующие операции:
 - указать маску ввода: 000000000000;
 - в свойстве **Индексированное поле** выбрать значение **Да (Совпадения не допускаются)**.

Таблица П1.4. Описание свойств полей таблицы ИСПОЛНИТЕЛЬ

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Подпись поля
Код исполнителя	Ключ простой	Да	Счетчик	4	
Фамилия		Да	Текстовый	20	
Имя		Нет	Текстовый	10	
Отчество		Нет	Текстовый	15	
Должность		Нет	Текстовый	10	
E-mail		Нет	Текстовый	20	
Рабочий телефон		Нет	Текстовый	15	
Домашний телефон		Нет	Текстовый	15	
Мобильный телефон		Нет	Текстовый	15	
Адрес		Нет	Текстовый	40	
Веб-страница		Нет	Гиперссылка		
Характеристика		Нет	Длинный текст		

Требование к заполнению табл. П1.4: для полей Фамилия, Имя, Отчество создайте составной индекс.

Таблицы плановых данных

Таблица П1.5. Описание свойств полей таблицы ДОГОВОР

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Формат	Число дес. знаков	Подпись поля
НОМ_ДОГ	Ключ простой	Да	Текстовый	5			Номер договора
ДАТА_ДОГ		Нет	Дата/время	5	Краткий формат даты		Дата
КОД_ПОК		Да	Текстовый	5			Код покупателя
СУММА_ДОГ		Нет	Денежный		Денежный	Авто	Сумма по договору

Таблица П1.6. Описание свойств полей таблицы ПОСТАВКА_ПЛАН

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Формат	Число дес. знаков	Подпись поля
НОМ_ДОГ	Ключ составной	Да	Текстовый	5			Номер договора
КОД_ТОВ		Да	Текстовый	5			Код товара
СРОК_ПОСТ		Да	Числовой	Байт	Фиксированный		Срок поставки (№ месяца)
МИН_ПОСТ		Нет	Числовой	Целое		Авто	Минимальная партия поставки
КОЛ_ПОСТ		Нет	Числовой	Длинное целое		Авто	Количество поставки
СУММА_ПОСТ		Нет	Денежный		Денежный	Авто	Сумма поставки

Таблицы оперативно-учетных данных

Таблица П1.7. Описание свойств полей таблицы НАКЛАДНАЯ

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Формат	Число дес. знаков	Подпись поля
НОМ_НАКЛ	Ключ составной	Да	Текстовый	5			Номер накладной
КОД_СК		Да	Текстовый	5			Код склада
ДАТА_ОТГ		Нет	Дата/ время		Краткий формат даты		Дата отгрузки
НОМ_ДОГ		Да	Текстовый	5			Номер договора
СУММА_НАКЛ		Нет	Денежный		Денежный	Авто	Сумма по накладной

Требование к заполнению табл. П1.7: необходимо предусмотреть для поля ДАТА_ОТГ ввод значения по умолчанию текущей даты: Date().

Таблица П1.8. Описание свойств полей таблицы ОТГРУЗКА

Имя поля	Признак первичного ключа	Обязательное поле	Тип данных	Размер	Формат	Число дес. знаков	Подпись поля
НОМ_НАКЛ	Ключ составной	Да	Текстовый	5			Номер накладной
КОД_СК		Да	Текстовый	5			Код склада
КОД_ТОВ		Да	Текстовый	5			Код товара
КОЛ_ОТГР		Нет	Числовой	Длинное целое		Авто	Количество
СУММА_ОТГР		Нет	Денежный		Денежный	Авто	Сумма по товару

ПРИЛОЖЕНИЕ 2

Пример заполненных документов для загрузки в базу данных «Поставка товаров»

Справочная информация

Код товара	Наименование товара	Цена	Единица изм.	Ставка НДС
T001	Монитор 17LG	6 587,00р.	штука	5%
T002	FDD 3,5	363,00р.	коробка	20%
T003	HDD Maxtor20GB	2 590,00р.	штука	10%
T004	Корпус MiniTower	1 916,00р.	штука	20%
T005	CD-ROM Panasonic IDE	1 153,00р.	штука	30%
T006	DIMM64MPC100	360,00р.	штука	15%
T007	Принтер HP LaserJet122C	5 432,00р.	штука	10%
T008	Сканер Acer	2 338,00р.	штука	15%
T009	Зв. Карта Genius Liv	789,00р.	штука	5%
T010	Модем Genius ext	1 295,00р.	штука	5%
T011	Плоттер	3 000,00р.	штука	10%
T012	Миникомпьютер	39 000,00р.	штука	15%

Номер склада	Код фирмы	Наименование	Отв. лицо	Адрес скл.
01	100	Главный	Иванов Т.С.	Мичуринская,
02	100	Оптовый	Петров А.А.	Свердлова, 29
03	100	Торговый	Смирнов О.Н.	Речной, 38

Код покупателя	ИНН	Наименование	Адрес	Телефон	Номер расч. счета	Банк
П001	778865432223	Компьютер маркет	Садовая, 22	(812)310-34-62	342456788903	Мост
П002	123865432223	Перспектива	Фонтанка, 67	(812)310-78-62	542456788982	Кредит
П003	543865432223	Инфоцентр	Большой, 34	(812)980-22-62	242456788965	ВТБ
П004	912865432223	Монитор	Средний, 37	(812)650-22-45	942456788987	Надежда
П005	935865432223	Компьютер лэнд	Тульский, 23	(812)310-22-62	142456788999	Инвест
П006	778887632223	Компьютерная техника	Малый, 12	(812)310-67-62	842456788909	Мост

Справочник исполнителей						
Фамилия	Имя	Отчество	Должность	E-mail	Рабочий телефон	Домашний телефон
Ильина	Юлия	Юрьевна	Начальник отдела	julij@mail.ru	(812)310-22-62	(812)510-72-62
Минаков	Владимир	Федорович	Ведущий специалист	m-m@bk.ru	(812)310-22-02	(812)610-65-62
Куликов	Евгений	Викторович	Главный специалист	evgeny@bk.ru	(812)310-22-61	(812)310-22-88
Нечаев	Иван	Петрович	Специалист	nech@bk.ru	(812)310-22-77	(812)310-22-09

Плановая информация

ДОГОВОР N		Д111	От		05.01.2016
ПОКУПАТЕЛЬ		Перспектива			
Код покупателя		Г002			
Код товара	Наименование товара	Месяц поставки	Мин. партия поставки	Количество поставки	
T001	Монитор 17LG	1	10	100	
T001	Монитор 17LG	2	5	20	
T001	Монитор 17LG	3	5	30	
T002	FDD 3,5	1	10	50	
T002	FDD 3,5	3	5	10	

Вводятся только те реквизиты, которые обведены.

ДОГОВОР N		Д222	От		25.02.2016
ПОКУПАТЕЛЬ		Компьютер маркет			
Код покупателя		Г001			
Код товара	Наименование товара	Месяц поставки	Мин. партия поставки	Количество поставки	
T001	Монитор 17LG	1	10	100	
T001	Монитор 17LG	2	4	12	
T003	HDD Maxtor 20GB	2	5	10	
T004	Корпус MiniTower	3	15	30	

ДОГОВОР № От

ПОКУПАТЕЛЬ

Код покупателя

Код товара	Наименование товара	Месяц поставки	Мин. партия поставки	Количество поставки
T005	CD-ROM Panasonic IDE	1	10	40

Оперативно-учетная информация

НАКЛАДНАЯ № от

Номер склада

Номер договора

Код товара	Наименование товара	Количество
T001	Монитор 17LG	10
T002	FDD 3,5	40

НАКЛАДНАЯ № от

Номер склада

Номер договора

Код товара	Наименование товара	Количество
T001	Монитор 17LG	10
T002	FDD 3,5	230

НАКЛАДНАЯ № от

Номер склада

Номер договора

Код товара	Наименование товара	Количество
T002	FDD 3,5	12

НАКЛАДНАЯ №	0004	от	25.03.2016
Номер склада	01		
Номер договора	Д222		
Код товара	Наименование товара	Количество	
T004	Корпус MiniTower	4	
T001	Монитор 17LG	5	

НАКЛАДНАЯ №	0005	от	13.05.2016
Номер склада	02		
Номер договора	Д222		
Код товара	Наименование товара	Количество	
T001	Монитор 17LG	10	
T003	HDD Maxtor20GB	8	

Предметный указатель

A

AutoExec 421

D

Drag and Drop 51

O

ODBC 35

OLE 32, 121

◇ Object 102, 103, 105, 131, 443, 445, 464

P

PowerPivot

◇ для Excel 364

◇ мастер импорта таблиц 368

◇ отображение модели данных 364

◇ получение внешних данных 367

S

SharePoint Server 2013 439

SQL 35, 157

◇ инструкция TRANSFORM 428

◇ Server 35

Structured Query Language 15

A

Автозамена имен объектов 30

Автоотчет 327

Б

База данных

◇ загрузка 260

◇ зашифровать паролем 57

◇ конвертация 25

◇ логическая структура 61, 63

▫ пример 84

◇ многопользовательская 33

◇ монопольный режим 57

◇ параметры автозамены имен 30

◇ перемещение таблиц 34

◇ разделение 34

◇ реляционная 19

▫ структура 89

◇ сжать и восстановить 25

◇ создание 62, 89

◇ технология загрузки 262

◇ типовая 26

◇ формат ACCDE 24

◇ формат LACCDB 24

◇ этапы проектирования 62

Бланк запроса 182

Блокировка

◇ записей 254

◇ на уровне записи 34

◇ на уровне страниц 34

В

- Веб-приложение 439
- ◇ Access
 - ссылка на запись главной таблицы 454
- ◇ SQL Server 2012 439
- ◇ SQL Windows Azure 439
- ◇ выбор расположения 440
- ◇ выход 448
- ◇ вычисление итогов 452
- ◇ добавление таблиц 443
- ◇ запросы 458
- ◇ защита таблиц 444
- ◇ ключевое поле таблицы 443
- ◇ недопустимые типы данных 443
- ◇ открыть в Access 447
- ◇ открыть в браузере 446, 450
- ◇ подчиненная форма 450
- ◇ поле со списком 455
- ◇ поля подстановки 449
- ◇ сведения о расположении 442
- ◇ связи таблиц 449
- ◇ сменить учетную запись 449
- ◇ события форм 462
- ◇ события элементов управления 461
- ◇ создание отчетов 463
- ◇ создание форм-представлений 445
- ◇ таблица
 - заблокировать 444
 - контекстное меню 443
 - скрыть 444
- ◇ типы макросов 461
- ◇ уникальный индекс таблицы 443
- Вкладки документов 42, 50
- Включить содержимое 422
- Вложенные макросы 384
- Внедренный макрос 376, 377, 379, 388, 398, 426
- Внешний ключ 20
- Временная переменная 420, 426
- Всплывающее окно 420
- Вычисления: в форме 232

Г

- Главная таблица 20
- Группировка в отчете 301
- Групповые операции 170

Д

- Декартово произведение 190
- Диалоговое окно 41, 44
- Доступные шаблоны 25, 91

З

- Запись 15, 19
 - ◇ добавление 129
 - ◇ изменение значений 129
 - ◇ удаление 129
- Запросы 23, 157
 - ◇ бланк 158
 - ◇ в веб-приложении 457
 - ◇ групповые операции 170
 - условия отбора 174
 - ◇ итоги 175
 - ◇ к серверу SQL 17
 - ◇ конструирование 170
 - ◇ многотабличные 179, 180
 - декартово произведение 180, 190
 - ◇ на выборку 157
 - ◇ на добавление 207
 - ◇ на изменение 199
 - ◇ на обновление 158, 202
 - ◇ на создание таблицы 158, 201
 - ◇ на удаление 211
 - ◇ нарастающий итог 188
 - ◇ объединение записей 181
 - ◇ однотоабличные 170
 - ◇ параметры 168
 - ◇ режим SQL 175, 184, 187, 191, 195, 202, 204, 209, 211, 213
 - ◇ режим таблицы 162, 204
 - ◇ тип данных параметра 169
 - ◇ условия отбора 174, 186
- Защита баз данных 53
 - ◇ надежное расположение 54
 - ◇ небезопасные компоненты 53
 - ◇ Панель сообщений 55
 - ◇ Центр управления безопасностью 53

И

- Изолированный макрос 376
- Импорт 17
- Инструкции SQL 175, 184, 187, 191, 195
 - ◇ DELETE 213
 - ◇ INSERT INTO 209, 211
 - ◇ UPDATE 204

Инструкции SQL на создание
таблицы 202

Информационно-логическая модель 61, 63

- ◇ пример 83
 - ◇ разработка 61
- Информационный объект 62, 63
- ◇ выделение 66, 68
 - пример 70
 - ◇ главный 81
 - ◇ ключ 64
 - ◇ объект-связка 81
 - ◇ подчиненный 81
 - ◇ связи 80
 - ◇ экземпляр 64

Информация

- ◇ оперативно-учетная 263
 - ◇ плановая 262
 - ◇ справочная 262
- Итоги 302

К

Каскадное

- ◇ обновление связанных записей 151
- ◇ удаление связанных записей 151

Ключ

- ◇ внешний 89
- ◇ первичный 105
- ◇ связи 20, 85

Кнопка: свойства 137

Коллекция 41, 43

Команда: Подтаблица 135, 155

Конструктор 29

- ◇ запросов
 - панель инструментов 160
- ◇ форм 271

Контекстное меню 45

Корпоративная сеть 36

Л

Лента 41, 43, 95, 221

- ◇ Работа с макетами форм 221
 - ◇ Упорядочить 225
- Локальная сеть 33
- ◇ Microsoft Windows Server 33
 - ◇ NetWare Novell 33
 - ◇ одноранговая 33
 - ◇ с файловым сервером 33

М

Макет

- ◇ отчета 295
 - ◇ таблицы 127
 - ◇ элементов управления 223
- Макрокоманда 376
- ◇ ЗадатьЗначение 415
 - ◇ ЗапускМакроса 395, 400
 - ◇ каталог 378
 - ◇ НаЗапись 414
 - ◇ ОткрытьЗапрос 414, 415
 - ◇ Сообщение 415

Макросы 376

- ◇ выполнение
 - порядок 399
 - ◇ AutoExec 396
 - ◇ блок Если 390
 - пример 392
 - ◇ блок-схема выполнения 390
 - ◇ в веб-приложении 461
 - ◇ вложенные 384
 - ◇ внедренный макрос 376, 377
 - ◇ вызов 400
 - ◇ выполнение 395
 - ◇ запуск 395
 - автоматический 396
 - кнопкой 401
 - нажатием кнопки 400
 - через события 395, 396
 - ◇ изолированный макрос 376
 - ◇ создание 377
 - ◇ ссылки на объекты 380
 - ◇ условные выражения 394
- Макросы данных 376, 402
- ◇ журнал регистрации 412
 - ◇ именованные 403, 410
 - ◇ пример 405, 407, 408, 411
- Маска ввода 125
- ◇ мастер 126

Мастер

- ◇ диаграмм 28
- ◇ по анализу таблиц 29
- ◇ подстановок 28, 140, 143
- ◇ форм 220, 237, 267

Мастера Access 28

Мини-панель

- ◇ инструментов 42
 - ◇ форматирования текста RTF 123
- Модальное окно 420

Модель данных 14

Модули 24

Н

Набор вкладок 429

Надежность базы данных 421

Надстройка PowerPivot 364

Нормализация данных 19, 63

◇ требования 65

О

Обеспечение безопасности 200

Область

◇ документов 98

◇ навигации 42, 47, 96

◇ новая категория 97

Объект OLE

◇ внедренный

Объект OLE

◇ внедренный 121

▫ пример 131

◇ связанный 121

Объекты Access 22

Окно

◇ Access 93

◇ конструктора запросов 159

◇ макросов 378

Оперативный анализ данных 363

◇ технологии 342

Отчеты 23, 295

◇ вычисляемые поля 321

◇ группировка 297, 301, 329

◇ добавление даты 333

◇ добавление номеров страниц 333

◇ источник записей 311, 315, 319, 326

◇ итоги 302

▫ без подробностей 304

◇ мастер отчетов 314

◇ многотабличные 312

◇ нарастающий итог 335

◇ однотабличные 298

◇ параметры 323

◇ печать 309

▫ представление отчета 308

◇ подчиненные 328, 331

◇ проектирование макета 295

◇ просмотр 305, 307

◇ разделы 296

◇ режимы 298

▫ Предварительный просмотр 298, 307, 323, 334

▫ Представление отчета 298, 305

▫ режим конструктора 298, 320

▫ режим макета 298, 330

◇ свойства 338

◇ сортировка 303

◇ составные 325

▫ главный отчет 327

▫ подчиненный отчет 328

▫ режим конструктора 332

◇ сумма с накоплением 335

◇ фильтрация 289, 305, 324

П

Панель

◇ быстрого доступа 42, 95, 96

◇ инструментов

▫ таблица в режиме таблицы 126

▫ быстрого доступа 45

◇ сообщений 42, 51, 55, 421

◇ элементов 29

Параметры Access 53

◇ автозамена имен 30

◇ блокировка на уровне записи 34

◇ запроса 168

◇ клиента 40

◇ навигации 97

◇ настройка 46

◇ окна документа 50

◇ Центр управления безопасностью 53

Первичный ключ 19, 118. См. Уникальный ключ

◇ составной 120

Подтаблица 135

Подчиненная

◇ таблица 20

◇ форма 235

Поле 19

◇ Метод 122

◇ Вложение

▫ пример 121

▫ размещение 132

◇ гиперссылка 123

◇ Дата/время 125

◇ Мастер подстановок 140, 143

◇ общие свойства 115

▫ подпись 104

▫ правило проверки 105

- размер 104
- сообщение об ошибке 105
- формат 104
- ◇ общие свойства 103, 113
 - вставка 131, 154
 - объекта OLE
 - определение 101, 113
 - пример 121
 - формат 104
 - число десятичных знаков 104
- ◇ свойства 113
- ◇ со списком 105, 140, 275
- ◇ соглашение об именах 101
- ◇ тип данных 144
- ◇ типы данных 101
- ◇ в веб-приложении 455
- Построитель выражений 165, 383
- ◇ условие на значение 116
- Предметная область 61
 - ◇ документы 70, 71
 - ◇ пример описания 70
- Представления Backstage
 - ◇ Окно 26
- Предупреждение системы безопасности 51
- Преобразование базы данных 25
- Проект Access 11

Р

- Разделение базы данных 34
- Расположение в Интернете 440
- Режим
 - ◇ ввода записи 130
 - ◇ макета 221
 - ◇ перекрытия окон 99
- Реквизиты
 - ◇ зависимые 64, 87
 - ◇ ключевые 64
 - ◇ описательные 64
 - ◇ транзитивная зависимость 65, 69
 - ◇ функциональная зависимость
 - пример 72, 74, 76, 78, 79
- Реляционная
 - ◇ модель 14
 - ◇ таблица 19

С

- Сводные таблицы и диаграммы 343
 - ◇ источник данных
 - запросы 350

- ◇ источник данных 345, 348
- ◇ макет 345
- ◇ модель данных 343
 - в PowerPivot 364
 - источника 349, 358
- Свойства поля 143, 144
- Связи информационных объектов 80
 - ◇ много-многозначные 81
 - примеры 82, 83
 - ◇ одно-многозначные 80
 - примеры 82, 85
 - ◇ одно-однозначные 80
 - примеры 82
- Связная целостность 21
- Связь и внедрение объектов (OLE) 32, 121, 131
 - Связь с таблицами SQL сервера 17
 - Связь таблиц 20, 146, 148
 - ◇ по составному ключу 149
 - ◇ по суррогатному ключу 150
- Сервер баз данных SQL 35
- Сжать и восстановить базу данных 25
- Система управления базой данных (СУБД) 14
 - Службы Access 37
 - Создание диаграммы 427, 430
 - Список 105
 - Справочная система Access 51, 52
 - Ссылка
 - ◇ на подчиненную форму/отчет 383
 - ◇ на свойство 381
 - ◇ на элемент управления 252, 381
 - ◇ создание построителем 383
- Строка состояния 42, 50
- Структура
 - ◇ записи 15
 - ◇ реляционной таблицы 19
- Суррогатный ключ 102
 - ◇ Связь таблиц 150
- Схема данных 21, 147
 - ◇ определение связей 148
 - по составному ключу 149
 - ◇ пример 85, 152
 - ◇ создание 147
 - пример 147

Т

- Таблицы БД 22
 - ◇ ввод данных 126
 - пример 128
 - логически связанных 135

Таблицы БД (прод.)

- ◇ итоги 130
- ◇ макет 127
- ◇ мастер подстановок 143
- ◇ первичный ключ 105
 - счетчик 107
 - индексы 106
 - создание 100
 - структура 89, 100
- ◇ режим конструктора 112
- ◇ режим таблицы 118
- ◇ структура 112
- ◇ создание: пример 114
- ◇ тип данных

У

Уникальный ключ 19

- ◇ определение 105
- ◇ простой 65
- ◇ составной 65

Ф

Файл базы данных 24

- ◇ имя 92
- ◇ рабочий каталог 92
- ◇ создание 91
- ◇ формат 93

Файловый сервер 33

Фильтр 130

Форма навигации 435

Формат даты 125

Формы 23

- ◇ автоформат 222
- ◇ вкладки 429
- ◇ всплывающие 420
- ◇ выборка документа 285
- ◇ вычисление итогов 234
- ◇ вычисления 232, 282
- ◇ добавить поле 272
- ◇ загрузка БД 261
- ◇ защита данных 253
- ◇ источник записей 272
- ◇ кнопки управления 401
- ◇ конструирование 245, 271
- ◇ конструирование поля со списком 280
- ◇ макет 261

- ◇ мастер 237, 267
 - поле со списком 276
- ◇ многотабличные 234
 - макет данных 266
 - подсхема данных 264
 - создание 267
- ◇ модальные 420
- ◇ на основе запроса 228
- ◇ несколько элементов 219, 229
- ◇ одиночная многотабличная 242
- ◇ однотоабличные 219
- ◇ панель навигации 434
- ◇ параметры 229
- ◇ подготовка к созданию 260
- ◇ подчиненные 235, 246, 269
 - вычисления 251
 - защита данных 254
 - связь 250
- ◇ поле
 - защита от изменений 253
 - со списком 275
- ◇ пустая форма 243
- ◇ работа с документом 283
- ◇ разделенная форма 219
- ◇ разделенные 230, 231
- ◇ режим формы 227
- ◇ режимы 218
- ◇ свойства 230
- ◇ события 218, 259
- ◇ создание кнопок 274
- ◇ условное форматирование 224
- ◇ элемент управления 227
 - поле со списком 281
 - свойства 217

Ц

Целостность данных 63, 150

- ◇ задание параметров 150
- ◇ проверка 152
- ◇ условия установления 150

Э

Элемент управления

- ◇ свойства 401
- ◇ вставить диаграмму 28