



Министерство образования
Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра «Экономика строительства»

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ACCESS

**Лабораторный практикум
для студентов специальности 1-27 01 01
«Экономика и организация производства»**

Минск 2005

Министерство образования Республики Беларусь
БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Кафедра «Экономика строительства»

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ACCESS

Лабораторный практикум
для студентов специальности 1–27 01 01
«Экономика и организация производства»

М и н с к 2 0 0 5

УДК ~~681.3.066~~(076.5)

~~ББК 32.97.7~~

Л12

Составители:

О.О. Гурьева, Л.П. Гусева

Рецензенты:

Л.А. Сюсюткина, Т.Н. Водоносова

Л 12 Система управления базами данных Access: лабораторный практикум для студ. спец. 1–27 01 01 «Экономика и организация производства» / Сост.: О.О. Гурьева, Л.П. Гусева. – Мн.: БНТУ, 2004. – 47 с.

Лабораторный практикум содержит задания, позволяющие освоить основные приемы работы и возможности системы управления базами данных Access.

Издание предназначено для студентов II курса специальности 1–27 01 01 «Экономика и организация производства».

В в е д е н и е

Учебным планом предусмотрен лабораторный практикум по дисциплине «Сетевые технологии и базы данных», включающий самостоятельное выполнение лабораторных работ в компьютерном классе.

Практикум содержит задания для выполнения лабораторных работ, каждая из которых посвящена конкретной теме. По усмотрению преподавателя задания могут выполняться за одно или несколько занятий и предполагают как создание новых баз данных, так и использование лицензионной учебной базы данных Tutorial.

Задания выполняются в среде Access 2000, которая является одной из популярнейших *систем управления базами данных (СУБД)*. Современные СУБД представляют собой совокупность языковых и программных средств, обеспечивающих создание, поддержание, доступ к данным со стороны пользователей, и прикладных программ – *приложений*.

СУБД MS Access 2000 является *реляционной базой данных (БД)*. Реляционная база данных представляет собой множество взаимосвязанных двумерных таблиц — *реляционных таблиц*, называемых *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 MS Access 2000 реализованы все основные функциональные возможности СУБД:

1) *определение данных* — т.е. можно определить, какая именно информация будет храниться в базе данных, задать структуру данных, их тип, а также указать, как эти данные связаны между собой;

2) *обработка данных* — т.е. можно выбирать любые характеристики объектов, фильтровать и сортировать данные, объединять их с другой связанной с ними информацией и вычислять итоговые значения;

3) *управление данными* — т.е. можно указать, кому разрешено знакомиться с данными, изменять их или добавлять новую информацию, определить правила коллективного использования данных.

Система MS Access — типичная настольная СУБД. На небольшом предприятии или фирме ресурсов MS Access вместе с ресурсами остальных программных продуктов Microsoft Office достаточно для обслуживания всего делопроизводства.

Простые и в то же время мощные средства MS Access могут использоваться для:

- 1) бухгалтерского учета;
- 2) ввода заказов;
- 3) ведения информации о клиентах;
- 4) ведения информации о деловых контактах.

Поскольку СУБД MS Access может работать в режиме коллективного доступа к базам данных, она является хорошим средством разработки приложений для рабочих групп, которые хранят данные на серверах локальных сетей своих подразделений и в то же время периодически пользуются приложениями других подразделений, сохраняя свои данные на серверах учреждения или предприятия. Если речь идет о небольших рабочих группах, то хранение и коллективный доступ к данным могут осуществляться только при помощи MS Access. Большие приложения для хранения данных используют специальный сервер (например, SQL Server), а MS Access на рабочей станции выступает в этом случае в роли клиента. Кроме того, в крупных учреждениях MS Access может использоваться как пользовательская среда для обработки данных — создания собственных запросов на поиск нужной информации, построения диаграмм, отчетов и т.д.

Для автоматизации проектирования объектов БД MS Access 2000 имеет специализированные программы — Конструкторы и Мастера. *Конструктор* предоставляет пользователю ряд инструментальных средств, с помощью которых можно быстро и просто создавать и модифицировать объекты БД; *Мастер* де-

лает это по-другому: задает пользователю ряд вопросов и на основе его ответов создает вполне законченный объект БД.

Справочная система Access

Справочная система Access аналогична тем, что используются другими Windows-приложениями. После запуска Access можно воспользоваться любым видом справки, если обратиться к пункту меню *Справка*.

Выбор в нем пункта *Справка* по Microsoft Access приводит к появлению окна справки с тремя вкладками. Вкладка *Содержание* используется для просмотра оглавления справки и поиска информации по темам справки; вкладка *Мастер ответов* — для ввода своего вопроса; вкладка *Указатель* — для поиска информации по конкретным словам или фразам.

Выбор пункта *Показать помощника* приводит к появлению окна с анимационным персонажем. Помощник дает советы для конкретной рабочей ситуации и отыскивает необходимые разделы справочной документации в ответ на вопросы, вводимые в поле ввода. Его можно временно скрыть по команде *Справка/Скрыть помощника* или отключить на весь сеанс работы в Access, если в его выноске нажать кнопку *Параметры* и снять флажок *Использовать помощника*.

Пункт *Что это такое?* предназначен для получения краткой справки о назначении элементов в главном окне Access. Его выбор приводит к появлению указателя мыши в виде стрелки с вопросительным знаком; щелчок им по интересующему элементу дает сведения о нем.

Пункт *Office на Web* позволяет соединиться с Web-узлом корпорации Microsoft для получения обновленных сведений о программе.

Объекты базы данных и их размещение

СУБД Access 2000 ориентирована на работу с объектами БД. Основными объектами MS Access являются таблицы, за-

просы, формы, отчеты, страницы доступа к данным, макросы и модули.

Таблица — это основная структура, предназначенная для хранения информации в БД. В таблицах БД хранятся все данные, необходимые для решения задач предметной области. Каждый элемент данных должен храниться в базе только в одном экземпляре. Минимальное дублирование данных в реляционной БД обеспечивает высокую эффективность поддержания БД в актуальном и непротиворечивом состоянии, однократный ввод и корректировку данных.

Запрос — это требование на отбор данных, хранящихся в таблицах, выполнение вычислений над данными, изменения в БД.

Форма — созданный на экране шаблон, используемый для ввода, просмотра и редактирования записей БД.

Отчет — отображение на принтере или на экране информации из БД в виде, удобном для ее восприятия и анализа пользователем.

Страница доступа к данным — диалоговая Web-страница, которая поддерживает динамическую связь с БД и позволяет просматривать, редактировать и вводить данные в базу, работая в окне браузера Internet Explorer.

Макрос — последовательность макрокоманд для автоматизации выполнения операций в среде Access без программирования.

Модуль — это программа для работы с БД, написанная на языке Visual Basic for Applications (VBA).

Объекты БД могут быть объединены в именованные группы объектов по функциональному или иному признаку.

Все объекты, за исключением страниц доступа к данным, можно хранить в одном файле — файле БД с расширением .Mdb. Страницы доступа сохраняются как самостоятельные файлы с расширением .html отдельно от БД, с которой они связаны, а в файле базы данных размещаются только ярлыки на них. Отчеты можно сохранять в файле с расширением .snp (формат снимка отчета) и распространять среди пользователей по электронной

почте. С целью защиты форм, отчетов и модулей VBA БД может быть сохранена в файле приложения с расширением .mde; при этом БД сжимается, оптимизируется использование памяти и повышается быстродействие БД. Проект размещается в файле с расширением .adp на компьютере пользователя.

Основные термины и определения

Таблица — объект базы данных, используемый для хранения данных. Каждая таблица состоит из строк и столбцов, которые принято называть соответственно записями и полями.

Запись — строка таблицы базы данных, в которой собрана вся информация о конкретном предмете. Например, в таблице «Студенты» базы данных «Кафедра» это — информация о конкретном студенте — его фамилия, год его рождения, средний балл при поступлении в вуз и т.д.

Поле — столбец таблицы базы данных, составляющий часть записи, которая отводится для отдельной характеристики предмета. Так, в предыдущем примере полями являются фамилия студента, год его рождения, средний балл при поступлении в вуз и т.д. Порядок расположения полей с указанием их имен, тип хранимых в них данных, размер этих данных определяют *структуру* таблицы.

Первичный ключ (ключевое поле) — поле (совокупность полей), значения которого позволяют однозначно определить (идентифицировать) каждую запись таблицы. Например, в таблице «Сотрудники» значения поля «Табельный номер» являются уникальными, и поэтому оно может быть ключом для данной таблицы.

Для повышения эффективности поиска требуемых данных используются индексы.

Индекс — это внутренняя таблица приложения Access, имеющая два столбца: в первом находятся значения выражения, содержащего все поля, включенные в индекс, во втором — местоположение каждой записи таблицы с данным значением ин-

дексного выражения. Таким образом, Access сначала осуществляет поиск данных именно в индексе; если же индекс отсутствует, он просматривает все записи таблицы.

После определения таблиц необходимо указать Access, какие действия следует предпринимать для объединения содержимого различных таблиц, т.е. установить связи между таблицами.

Межтабличная связь — это отношение, устанавливаемое между полями двух таблиц. Эта связь может быть установлена при наличии в таблицах полей, содержащих совпадающие данные. Поля не обязательно должны иметь одинаковые имена, но необходимо, чтобы совпадали типы данных и размеры связующих полей.

В нормализованной реляционной БД связь между двумя таблицами характеризуется отношениями записей типа *один-к-одному* (1:1) или *один-ко-многим* (1:N). Отношение 1:1 предполагает, что каждой записи одной таблицы соответствует одна запись к другой. Отношение 1:N предполагает, что каждой записи первой таблицы соответствует много записей во второй, но каждой записи второй таблицы соответствует только одна запись в первой. Например, связь между таблицами «Читатели» и «Заказы» в базе данных *Библиотека* — связь типа *один-ко-многим*, т.к. один читатель может сделать много заказов, но любой заказ относится только к одному читателю. Если при этом связующее поле в одной из таблиц является ключевым, такая таблица называется *главной*, или *родительской*; вторая таблица, участвующая в связи, называется *подчиненной*. При этом связующее поле подчиненной таблицы обычно называют внешним ключом. *Внешний ключ* — это одно или несколько полей в таблице, содержащих ссылку на поле (или поля) первичного ключа в другой. Для ускорения поиска информации рекомендуется создавать индексы по внешним ключам.

Связь типа *многие-ко-многим* (N:M) — это межтабличное отношение, при котором каждой записи в первой таблице могут соответствовать несколько записей во второй таблице, и каждой записи во второй таблице — несколько записей в первой табли-

це. При наличии подобной связи лучше разбить ее на две связи типа *один-ко-многим* с помощью дополнительной таблицы.

Access 2000 поддерживает 3 типа связей (или отношений): *один-ко-многим*, *один-к-одному*, *многие-ко-многим*. При определении связей между таблицами в Access 2000 можно установить следующие параметры:

- 1) обеспечение целостности данных;
- 2) каскадное обновление связанных полей;
- 3) каскадное удаление связанных записей.

При этом Access автоматически будет отслеживать *целостность* данных, при которой не допускается наличия в базе данных подчиненной записи без связанной с ней главной. Второй параметр означает, что при изменении значения связующего поля в главной таблице соответствующие поля подчиненной таблицы будут автоматически обновлены. Третья опция означает, что в случае удаления записи из главной таблицы все связанные с ней записи из подчиненной таблицы будут удалены автоматически.

Тема 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

При создании БД необходимо решить вопрос о ее *наиболее эффективной структуре*. Правильно спроектированная БД позволяет:

- 1) обеспечить быстрый доступ к данным;
- 2) исключить ненужное повторение данных, которое приводит к нерациональному использованию дискового пространства компьютера и может явиться причиной ошибок при вводе;
- 3) обеспечить целостность данных таким образом, чтобы при изменениях в одном объекте автоматически происходили соответствующие изменения в связанных с ним объектах.

К основным принципам проектирования БД можно отнести отсутствие в таблице повторяющихся групп полей и повторяющихся записей.

Проектирование БД начинается с определения всех объектов, сведения о которых будут включены в базу, и их атрибу-

тов. Атрибуты затем сводятся в одну таблицу — исходное отношение. Например, надо создать БД для учета заказов в библиотеке. Вся информация для этой задачи может быть размещена в исходное отношение — таблицу «Заказы». Это могут быть следующие данные:

- 1) номер заказа по порядку;
- 2) фамилия, имя, отчество читателя, сделавшего заказ;
- 3) его должность;
- 4) место работы;
- 5) рабочий телефон;
- 6) домашний адрес;
- 7) домашний телефон;
- 8) название книги;
- 9) автор;
- 10) год издания;
- 11) стоимость;
- 12) дата заказа.

Информацию в полях рекомендуется хранить в атомарном виде, т.е. в виде минимально возможных элементов, так как наличие в одном поле нескольких элементов затрудняет извлечение отдельного элемента из этого поля.

Исходное отношение, построенное по перечисленным данным, содержит избыточное дублирование данных. Явная избыточность заключается в том, что строки с данными о читателе, берущем разные книги, повторяются соответствующее число раз. То же можно сказать о повторении данных о книге, взятой разными читателями. Избыточность данных приводит к значительным затратам на время ввода повторяющихся данных, к росту числа ошибок при их вводе и времени изменения какого-либо данного, так как придется корректировать все записи, содержащие это данное. Кроме того, избыточность данных приведет к увеличению размера БД.

Средством исключения избыточности в таблицах является их *нормализация*.

Первое правило при проектировании БД: в таблицах должны отсутствовать повторяющиеся группы полей. Этого можно

добиться, поместив поля с повторяющейся информацией о читателях в отдельную таблицу, присвоив каждому читателю уникальный код, записываемый в новое поле *КодЧитателя*. В исходном отношении в таблице «Заказы» останется информация о заказах и книгах, а информация о читателе будет заменена полем *КодЧитателя*. В результате будут получены две таблицы: «Читатели» и «Заказы». Таблица «Читатели» будет иметь атрибуты: *КодЧитателя*, фамилия, имя, отчество читателя, его должность, место работы, рабочий телефон, домашний адрес, домашний телефон. Исходная таблица «Заказы» будет иметь атрибуты: номер заказа, *КодЧитателя*, название книги, автор, год издания, стоимость, дата заказа.

Информацию о книге также необходимо выделить в отдельную таблицу «Книги», добавив в нее дополнительное поле, однозначно идентифицирующее книгу, — поле *КодКниги*. При этом информацию о книге надо исключить из исходной таблицы, заменив ее одним атрибутом — *КодКниги*. В результате в исходной таблице «Заказы» останутся четыре атрибута: номер заказа, *КодЧитателя*, *КодКниги*, дата заказа.

Поскольку в отдельный заказ может быть включено несколько книг, имеет смысл информацию о заказанных книгах выделить в отдельную таблицу и фиксировать в ней код заказа. В результате новая таблица «Заказанные книги» будет содержать поля: *КодЗаказа*, *КодКниги*; таблица «Заказы» — поля: *КодЗаказа*, *Дата заказа*, *КодЧитателя*.

Второе правило при проектировании БД: таблица не должна иметь повторяющихся записей. Для выполнения этого правила в таблице должен быть уникальный идентификатор (*первичный ключ*). В рассматриваемом примере атрибут *КодЧитателя* является уникальным в таблице «Читатели» и может быть первичным ключом. Первичным ключом может быть *КодКниги* в таблице «Книги» и *КодЗаказа* в таблице «Заказы». В таблице «Заказанные книги» уникальной является комбинация ключей *КодЗаказа* и *КодКниги*, поэтому первичный ключ здесь — составной и включает в себя поля *КодЗаказа* и *Код-*

Книги. В результате описанных операций окончательный проект БД включает в себя четыре таблицы.

Процессы разработки конкретных программных приложений в среде Access, в первую очередь, определяются спецификой каждой из автоматизируемых предметных областей. Но для большинства из них можно выделить ряд типичных этапов:

- 1) разработка и описание структур таблиц данных;
- 2) разработка схемы данных и задание системы взаимосвязей между таблицами;
- 3) разработка системы запросов к таблицам БД и их интеграция в схему данных;
- 4) разработка экранных форм ввода/вывода данных;
- 5) разработка системы отчетов по данным;
- 6) разработка программных расширений для базы данных, решающих специфические задачи по обработке содержащейся в ней информации, с помощью инструментов макросов и модулей;
- 7) разработка системы защиты данных, прав, ограничений по доступу.

Между этими этапами существует большое количество обратных связей.

З а д а н и е 1

Спроектировать таблицы по данным заданного варианта.

Вариант 1. Номер зачетки; фамилия, имя, отчество студента; группа; предмет; оценка; год рождения; адрес.

Вариант 2. Фамилия, имя, отчество владельца квартиры; его адрес; сколько человек прописано; наименование коммунальной услуги; стоимость коммунальной услуги.

Вариант 3. Фамилия, имя, отчество сотрудника; должность; оклад; адрес; семейное положение; автобиография; дата приема на работу; дата рождения.

Вариант 4. Наименование лекарства; аннотация; место хранения; дата поступления; приход; остаток на конец месяца; фирма-производитель; поставщик.

Вариант 5. Фамилия, имя, отчество рабочего; номер бригады; табельный номер; специальность; тарифная ставка; семейное положение; адрес; дата рождения; дата приема на работу.

Вариант 6. Наименование отдела магазина; наименование продукта; реализовано за первую декаду месяца; стоимость продукта; реализовано за вторую декаду месяца; характеристика продукта; производитель продукта; реализовано за третью декаду месяца.

Вариант 7. Наименование предприятия; наименование изделий, поставляемых предприятием; характеристика изделий; адрес предприятия; директор; телефон отдела снабжения; план поставок предприятия.

Вариант 8. Номер склада; грузополучатель; ответственное лицо грузополучателя; его телефон; местоположение склада; главный кладовщик; его рабочий телефон; отгружено товара.

Вариант 9. Спроектировать базу данных в предметной области «Футбол в Белоруссии» (не более трех таблиц).

Вариант 10. Спроектировать базу данных в предметной области «Агентство недвижимости» (не более трех таблиц).

Тема 2. СОЗДАНИЕ И КОРРЕКТИРОВКА БАЗ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ MS ACCESS 2000

2.1. Пользовательский интерфейс Access

Для работы с интерфейсом Access необходимо запустить СУБД MS Access 2000. На экране появляется окно запуска. Для создания файла новой базы данных переключатель устанавливают в положение *Новая база данных*. В появившемся окне *Файл новой базы данных* надо указать, на каком диске (на диске D:, в папке *Мои документы* или на своем диске A:), в какой папке, под каким именем создается файл БД, и нажать кнопку [Создать]. Выполнение одной из предложенных операций вызывает главное окно Access.

В строке заголовка отображается имя программы и содержатся кнопки для управления окном. Строка меню содержит несколько пунктов. Их перечень и содержание изменяются в зависимости от режимов работы системы. Панель инструментов представляет собой набор графических кнопок, назначение которых — ускоренный вызов команд меню. Access имеет большое количество панелей инструментов, используемых в различных режимах работы. Панели инструментов, соответствующие некоторому режиму, выводятся на экран автоматически при переходе в этот режим.

2.2. Создание таблицы

В приложении Access 2000 существуют 3 способа *создания таблицы*:

- 1) в режиме таблицы;
- 2) с помощью Мастера;
- 3) в режиме Конструктора.

Наиболее просто таблицы создаются в режиме таблицы выбором строки *Создание таблицы путем ввода данных* в рабочем поле окна БД или строки *Режим таблицы* в окне *Новая таблица*.

После выбора этого режима сразу открывается пустая таблица, в которую можно ввести данные. При сохранении этой таблицы Access анализирует данные и автоматически присваивает соответствующий тип данных каждому полю. Таблица имеет 20 столбцов и 31 строку. Полям таблицы по умолчанию присваиваются имена *Поле1*, *Поле2* и т. д.

Любое поле этой таблицы можно переименовать в соответствии с требованиями пользователя, непосредственно редактируя имена в заголовке столбцов. Для этого следует дважды щелкнуть кнопкой мыши на области выделения столбца, содержащей его имя, или выполнить команду *Формат — Переименовать столбец*. В результате курсор будет установлен на наименовании поля, и откроется возможность его переимено-

вания. Для сохранения нового имени надо щелкнуть мышкой в любом месте таблицы.

По окончании ввода данных во все нужные столбцы таблицу необходимо сохранить. Созданные в этом режиме таблицы требуют доработки в режиме Конструктора.

Создание таблицы с помощью Мастера сводится к созданию новой таблицы на основе образца из деловой или личной сферы. Мастер автоматически создает таблицу по одному из шаблонов. Пользователю предлагается для выбора более 50 образцов таблиц, предназначенных для использования в различных целях. Каждая из них содержит определенный набор полей, из которого пользователь может выбрать нужные. Включаемые в таблицу поля могут быть переименованы. Начать работу Мастера можно, выбрав строку *Создание таблицы с помощью Мастера* в рабочем поле объекта *Таблицы* окна БД или строку *Мастер таблиц* в окне *Новая таблица*. После создания таблицы Мастером можно в любое время доработать структуру таблицы в режиме конструктора.

Создание таблицы в режиме Конструктора начинается с выбора строки *Создание таблицы в режиме Конструктора* в рабочем поле окна БД или *Конструктор* в окне *Новая таблица*.

При выборе режима *Конструктор* появляется окно *Таблица1: таблица*, в котором определяется структура таблицы БД. При переходе в режим *Конструктор таблиц* меняется состав команд меню и появляется панель инструментов *Конструктор таблиц*.

В этом окне надо описать структуру таблицы, т.е. указать имя поля, выбрать соответствующий тип поля из раскрывающегося списка, задать размер поля и другие свойства в окне *Свойства поля*, которое становится доступным после выбора типа поля. Если создается главная таблица, необходимо определить в ней *первичный ключ*. При создании подчиненной таблицы надо учитывать тип связи между таблицами. Для связи типа 1:1 для *вторичного ключа* задается свойство *Индексированное поле со значением "Да (Совпадения не допускаются)"*, для связи типа

1:М — поле со значением “*Да (Допускаются совпадения)*”. После описания структуры таблицы окно Конструктора закрывается, таблица сохраняется, и ей присваивается имя.

Для заполнения созданной таблицы конкретными данными она открывается в режиме таблицы посредством кнопки [Открыть]. Переход к новой записи вызывает автоматическое сохранение предыдущей записи.

З а д а н и е 2

Требуется создать БД *Библиотека*, состоящую из 3 таблиц: «Книги», «Читатели», «Заказы». Таблицу «Книги» создать, используя режим *Создание таблицы с помощью Мастера*, таблицу «Читатели» — в режиме Конструктора, таблицу «Заказы» — в режиме таблицы. Ввести данные в таблицы и сохранить их. В таблицах «Книги» и «Читатели» должно быть не менее 10 записей.

Для создания таблицы «Книги»:

1) выбрать кнопку [Таблицы](панель *Объекты*), кнопку [Создать] (панель инструментов окна БД), в диалоговом окне (ДО) *Новая таблица* выбрать *Мастер Таблиц*, [ОК];

2) выбрать образец таблицы для создания собственной таблицы: в диалоговом окне *Создание таблиц* установить переключатель *Личные*, в списке *Образцы таблиц* выбрать «Книги»;

3) включить в новую таблицу поле *КодКниги*, поле *Переводчик* (переименовать его в *Автор*), *МестоПубликации* (переименовать его в *Город*), *Издательство*, *ГодИздания*, *ЦенаПокупки* (переименовать его в *Стоимость*), *ЧислоСтраниц*, кнопка [Далее];

4) в качестве имени таблицы оставить имя «Книги», в разделе *Выберите способ определения ключа* установить переключатель *Microsoft Access автоматически определяет ключ*, кнопка [Далее];

5) ввести две записи в таблицу «Книги»;

б) ввести в структуру таблицы «Книги» следующие изменения: переименовать поле *ЧислоСтраниц* в *Количество страниц*; добавить новое поле *Название* (текстовый размер 50), вставив его перед полем *Город*; поменять местами поля *Стоимость* и *Количество страниц*; для поля *Стоимость* изменить значение свойства *Число десятичных знаков* на 0;

7) перейти в режим таблицы;

8) заполнить таблицу данными.

2. Для создания в режиме Конструктора таблицы «Читатели» необходимо:

1) выбрать кнопку [Создать] на панели инструментов окна базы данных; в ДО *Новая таблица* выбрать *Конструктор*, кнопка [ОК];

2) определить поле *КодЧитателя* (указать имя поля, тип данных - *Счетчик*);

3) определить поля *Фамилия*, *Имя*, *Отчество*, *Должность* (указать имя поля, тип данных - текстовый, размер поля - по умолчанию);

4) определить поле *Место работы* (указать имя поля, тип данных - текстовый, размер поля - 100);

5) определить поле *Индекс* (указать имя поля, тип данных - текстовый, размер поля – 10); , задать маску ввода на вкладке *Общие свойства поля* в нижней части окна в строке *Маска ввода* (для этого ввести: 000000; ;_); по *Справке* определить, что означает такая маска;

б) определить поле *Город* (указать имя поля, тип данных - текстовый, размер поля – 50); определить значение, автоматически добавляемое в поле для новой записи (щелчок мыши; в строке *Значения по умолчанию* ввести "Минск");

7) определить поле *Домашний адрес* как поле *Место работы*;

8) определить поле *Домашний телефон* (для этого указать имя поля, тип данных – текстовый, размер поля – 15); задать маску ввода: !\ (999) "999\ -99\ -99; ;_); по *Справке* определить, что означает такая маска;

9) создать копию поля *Домашний телефон* и поместить ее перед полем *Город*: переименовать копию в *Рабочий телефон*;

10) задать первичный ключ таблицы (выделить поле *КодЧитателя*, меню *Правка/Ключевое поле* или соответствующую пиктограмму на панели инструментов; обратить внимание, что ключевое поле автоматически индексируется; ознакомиться с назначением индекса и способами его создания в *Справке*. Для этого выполнить команду *Справка/Справка по MS Access/вкладка Содержание/пункт Создание и разработка таблиц/ Работа с первичными ключами и индексами/Создание индекса для ускорения поиска и сортировки записей*;

11) перейти в режим таблицы;

12) заполнить таблицу данными;

13) завершить работу с таблицей «Читатели».

3. Для создания таблицы «Заказы» в режиме таблицы необходимо определить поля (полями этой таблицы являются: *КодЗаказа*, *КодЧитателя*, *КодКниги*, *Дата заказа*, *Отметка о возврате*); скорректировать структуру таблицы следующим образом в режиме Конструктора:

1) определить поле *КодЗаказа* (указать тип данных - *Счетчик*);

2) определить поле *КодЧитателя* (указать тип данных — числовой, размер поля — *Длинное целое*; удалить значение 0, автоматически добавляемое в поле для новой записи, клавишей [Delete] или [Backspace]; создать индекс по этому полю — *Совпадения допускаются*);

3) определить поле *КодКниги* (указать тип данных — числовой, размер поля — *Длинное целое*, удалить значение 0, автоматически добавляемое в поле для новой записи, клавишей [Delete] или [Backspace]);

4) определить поле *ДатаЗаказа* (указать тип данных — *Дата/время*, формат вывода значений даты на экран и на печать — *Краткий формат даты*, маску ввода с использованием Мастера по созданию масок ввода — *Краткий формат даты*);

5) определить поле *ОтметкаОВозврате* (указать тип данных — логический, формат вывода значений — *Да/Нет*, значение, автоматически добавляемое в поле для новой записи, — *Нет*, тип элемента управления для вывода поля — *Флажок*);

- 6) определить первичный ключ — *КодЗаказа*;
- 7) перейти в режим таблицы;
- 8) заполнить таблицу данными с учетом того, что один читатель может прийти в библиотеку несколько раз за разными книгами, и одна книга может быть взята разными читателями;
- 9) завершить работу с таблицей «Заказы».

Тема 3. СОЗДАНИЕ СХЕМЫ ДАННЫХ

Схема данных является удобным и наглядным средством отображения логических связей таблиц БД, используемых для объединения записей связанных таблиц при любой обработке. Это упрощает процесс конструирования многотабличных запросов, форм и отчетов. Если создается связь с поддержкой целостности данных, то можно предусмотреть в БД следующие каскадные операции (т.е. операции по корректировке БД):

1) каскадное обновление связанных полей — изменение значений ключа в записи главной таблицы, приводящее к автоматическому изменению значений вторичного ключа в подчиненных записях;

2) каскадное удаление связанных записей — удаление записи из главной таблицы, приводящее к автоматическому удалению всех подчиненных записей.

Для создания схемы данных необходимо:

1) открыть окно *Схема данных* по команде *Сервис/Схема данных*; найти на панели инструментов соответствующую пиктограмму;

2) определить таблицы, между которыми необходимо создать связи; в ДО *Добавление таблицы* выделить протяжкой мыши все связываемые таблицы или щелчком мыши – первую таблицу; нажать клавишу [Shift] и, не отпуская, щелчком мыши выделить последнюю таблицу;

3) нажать кнопки [Добавить] и [Заккрыть];

4) установить связи между таблицами по одноименным полям, перетащив первичный ключ на вторичный;

- 5) установить флажок *Обеспечение целостности данных*;
- 6) нажать кнопку [Создать];
- 7) сохранить изменения схемы данных и закрыть окно *Схема данных*, в котором между таблицами установится связь, обозначенная на схеме (например, 1:1 или 1:∞).

При создании схемы данных возможны неточные действия пользователя: например, добавление лишней таблицы. В этом случае из окна *Схема данных* надо выделить эту таблицу и нажать [Delete]. Для удаления линии связи, образовавшейся при перетаскивании первичного ключа на вторичный, ее тоже надо выделить и нажать [Delete]. Для печати схемы данных используется команда *Файл / Печать схемы данных*. Для изменения параметров существующей связи следует:

- 1) открыть окно *Схема данных*;
- 2) сделать двойной щелчок по линии связи, подлежащей изменению;
- 3) изменить параметры связи в появившемся окне *Изменение связей*.

З а д а н и е 3

Создать схему данных БД *Библиотека*.

З а д а н и е 4

Открыть учебную БД Tutorial. Ознакомиться с данными этой базы данных, создать ее схему данных.

Тема 4. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ

Запрос — это объект БД, который используется для извлечения нужной информации из одной либо нескольких таблиц БД или для выполнения определенных действий с данными. Например, чтобы из БД *Библиотека* получить информацию о книгах всех издательств, расположенных в Минске, необходимо сформировать запрос, в котором будут указаны конкретные

условия отбора («Город Минск»). Для записи подобных условий поиска данных используются специальные операторы.

При выполнении запроса из всей совокупности информации будут отобраны и выведены на экран в табличном виде только те данные, которые удовлетворяют поставленным условиям. Представленный на экране результат выполнения запроса называют *выборкой*, или *динамической таблицей*. В БД запрос сохраняется в виде *конкретных условий отбора*, т.е. при каждом выполнении запроса выборка формируется всякий раз заново на основе реальных таблиц с учетом всех последних изменений данных.

Запросы предназначены для просмотра, анализа и изменения данных, хранящихся в БД. Они используются для создания форм, отчетов, таблиц, других запросов и страниц доступа к данным. Запросы бывают разных типов:

- 1) запросы на выборку;
- 2) перекрестные запросы;
- 3) запросы действия;
- 4) запросы SQL.

Наиболее часто используемым типом запроса является *запрос на выборку*, предназначенный для отбора данных из одной или нескольких таблиц по заданным условиям и отображающий отобранные данные в виде динамической таблицы. Такие запросы можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

З а д а н и е 5

Создать запрос *Выбор_1* в режиме *Конструктор*. Его результатом должна быть динамическая таблица, содержащая данные всех полей таблицы Office плюс значения нового поля с именем *Новое*, которые рассчитываются по формуле: $YtdSales + YtdSales/2$. В динамической таблице должны быть выведены записи для значений поля *YtdSales*, больших 212000. Данные следует отсортировать по полю *State*. Вывод

всех полей из таблицы «Office» осуществляется выбором символа *, который расположен первым в таблице «Office». Сохранить запрос.

З а д а н и е 6

Создать запрос *Выбор_2* в режиме *Конструктор*. Его результатом должна быть динамическая таблица, содержащая следующие данные: шифр офиса (Ono), место расположения офиса (City) из таблицы «Office», названия компаний, относящихся к этому офису (Company), из таблицы «Customer», дату (Date) и числовое значение Itotal из таблицы «Invoices». Упорядочить значения шифра офиса. Вывести перечисленные данные для компаний Info Ltd и Citco Ltd. Сохранить запрос.

З а д а н и е 7

Открыть предыдущий запрос, модифицировать его и определить количество записей, в которых названия компаний начинаются с символов B, I, Q. Модифицировать запрос и вывести данные для компаний, название которых начинается с символов System. Сколько таких записей? Сохранить запрос под именем *Выбор_3*, используя команду *Файл/Сохранить как...*

З а д а н и е 8

Открыть предыдущий запрос. Изменить его и определить количество записей, в названии компаний которых слово Systems находится в конце названия. Отобразить данные для дат, находящихся в промежутке 11.05.90 и 17.05.90. Сколько таких записей? Сохранить запрос под именем *Выбор_4*.

З а д а н и е 9

Открыть запрос *Выбор_2*. Убрать условия отбора. Создать новое вычисляемое поле Itot, значение которого вычисляется

по формуле Itotal (из таблицы «Invoices»)*Ono (из таблицы «Office»). Сохранить запрос под именем Выбор_5. Определить количество записей, в которых значение поля Itot больше или равно 8361, меньше 11665 и равно 14223. Закрывать запрос без сохранения условий отбора.

З а д а н и е 10

Открыть предыдущий запрос *Выбор_5*. Удалить поля Company и Idate. Упорядочить значения поля Ono. Просмотреть результирующую таблицу. Использовать команду *Вид/Групповые операции*, чтобы вывести для каждого офиса значение суммы значений поля Itotal, среднее значений поля Itot. Просмотреть результирующую таблицу. Вывести количество городов, в которых расположены филиалы офисов с одним шифром. Сохранить запрос под именем *Выбор_6*.

З а д а н и е 11

Построить запрос *Выбор_7* по таблицам «Invoices» и «Detail». Результирующая таблица должна содержать поля: Ino, Cno, Price, Line, *Итого*. Значение поля *Итого* вычисляется по формуле: значения поля Price умножаются на значения поля Line. Отобразить записи, для которых значения поля Ino находятся в промежутке 1100 и 1200. Для каждого значения поля Ino вывести значение суммы по полю Price, минимальное значение поля *Итого* и количество записей, участвующих в подведении итогов. Сохранить запрос под именем *Выбор_7*.

К запросам на выборку относятся *запросы с параметрами* — запросы, которые при выполнении отображают в собственном диалоговом окне приглашение ввести данные, — например, условие для отбора записей или значение, которое требуется вставить в поле. Диалоговое окно создается MS Access автоматически после задания параметра и определения типа вводимых данных. Для поля, по которому проводится выборка данных, задается параметр в строке "Условие отбора" в квад-

ратных скобках, который представляет собой приглашение для ввода значения. Например: [Введите, пожалуйста, фамилию студента]. По команде *Запрос/Параметры* можно дополнительно задать тип вводимых данных: в нашем примере вводимая фамилия имеет тип "текстовый". Чтобы запросить у пользователя один или несколько символов для поиска записей, которые начинаются с этих символов или содержат их, создается запрос с параметрами, использующий оператор Like и подстановочный знак (*). Например, выражение: Like [Введите первый символ для поиска:] & "*" будет выполнять поиск слов, начинающихся с введенного символа. Выражение: Like "*" & [Введите любой символ для поиска:] & "*" выполняет поиск слов, которые содержат указанный символ.

З а д а н и е 1 2

Создать запрос с параметром по таблицам «Office», «Customer», «Invoices». Для задаваемого штата вывести количество офисов (Ono), среднее значение поля Itotal и максимальное значение поля Itot. Параметром является фраза [Введите название штата]. По команде *Запрос/Параметры* задать тип вводимых данных.

З а д а н и е 1 3

Вывести для задаваемых начальным символом городов из таблицы «Office» (Ono) и из таблицы «Customer» название компаний; город и штат, в которых расположены компании; числовое поле Ytdpurch, заданное в промежутке значений 1000...10000. Сколько получится записей для городов, начинающихся с символа "N"; какое значение Ytdpurch минимально, какое максимально? Сколько получится записей для городов, начинающихся на символ "B", на символы "Bo"? Сколько получится записей для городов, начинающихся на символ "M"? Какой город – первый по алфавиту? Какой последний? Сколько записей получится для городов, начинающихся на символы "Mi"?

З а д а н и е 14

Вывести для штатов, содержащих символ "Т", из таблиц «Customer», «Invoices», «Detail» суммарное значение поля Itotal, количество деталей, определяемое по их шифру (поле Pno), максимальное значение цены деталей (поле Price). Определить количество записей. Вывести эти данные для штатов, содержащих символ "А". Определить количество записей. Вывести эти данные для штатов, содержащих и символ "О". Определить количество записей.

Перекрестные запросы группируют данные, представляя их для анализа в более наглядном и компактном виде. В таком запросе данные выводятся подобно сводным таблицам приложения «Excel»: каждому столбцу и каждой строке соответствует поле исходной таблицы. При создании перекрестного запроса для каждой из пар строка/столбец исходной таблицы вычисляются значения в соответствии с заданной функцией. Данные могут быть разбиты на группы, для которых выполняются групповые вычисления (суммы, количества записей и средних значений). Перекрестный запрос создается с помощью Мастера или самостоятельно в режиме Конструктора. В этом режиме можно указать поля, значения которых будут заголовками столбцов и строк, а также поле, значения которого следует использовать в вычислениях. Мастер перекрестного запроса вызывается из диалогового окна *Новый запрос*.

З а д а н и е 15

С помощью Мастера создать перекрестный запрос *Запр_перек_1*. Результат этого запроса представляет собой таблицу, в которой поле Company (из таблицы «Customer») составляет заголовки строк, поле Ono (таблица «Office») – заголовки столбцов, а поле Itotal (таблица «Invoices») – суммарное значение для каждого офиса. Создание перекрестного запроса начинается с создания запроса на выборку по заданным полям;

далее производится операция группировки данных по шифру офиса; затем, используя клавишу [Тип запроса], выбирают тип - перекрестный. В соответствии с заданием задаются заголовки строк, столбцов, выводимые значения.

З а д а н и е 1 6

В режиме Конструктора создать перекрестный запрос *Запр_перек_2*. Данные предыдущего задания вывести для компаний, название которых начинается на символы "Во".

З а д а н и е 1 7

В режиме Конструктора создать перекрестный запрос *Запр_перек_3*. Данные предыдущего задания вывести для компаний, в названии которых присутствует слово «Computer».

Запросы действия или *запросы на изменения* — это запросы, в результате действия которых изменяются данные самой БД. К таким относятся *запрос на обновление, запрос на удаление, запрос на добавление, запрос на создание таблиц*. Результаты работы этих запросов являются необратимыми. Для предотвращения ошибок, связанных с этим обстоятельством, необходимо сделать резервную копию таблицы, прежде чем применить к ней запрос на изменение. Лучше всего первоначально запустить запрос на изменение в качестве запроса на выборку. Убедившись, что он выделяет необходимые записи, надо преобразовать его в запрос на изменение и запустить на выполнение. Ознакомиться с созданием запросов действия в справке Access.

З а д а н и е 1 8

Создать копию таблицы «Customer» и назвать новую таблицу «Cust1».

Удалить из таблицы «Cust1» все записи, создав для этого запрос на удаление.

З а д а н и е 19

Создать запрос на добавление записей в таблицу «Cust1». Для этого создать запрос на выборку по таблице «Customer», отобразить поля Company, City, State, Ytdpurch. Отобразить записи для штата Оризона (OZ). Проверить результаты: должно быть отобрано 8 записей. Используя клавишу [Тип запроса], в режиме Конструктора выбрать запрос на добавление записей. Записи добавлять в таблицу «Cust1». Продемонстрировать выполнение запроса. Дважды выполнить этот запрос, каждый раз проверяя результат выполнения.

З а д а н и е 20

Создать запрос на обновление значений поля Ytdpurch таблицы «Cust1». Новое значение этого поля вычисляется по формуле: Ytdpurch*2,3.

З а д а н и е 21

Создать запрос на поиск повторяющихся записей в таблице «Customer» (значения повторяются в поле State), дополнительно вывести названия компаний и их адреса.

З а д а н и е 22

Создать таблицу «Новая», которая должна содержать поля Ono, Ytdsales, Company, Itotal. Выполнить запрос, проверить результат. Модифицировать запрос, вводя условие отбора для поля Ono.

Тема 5. ФОРМЫ В ПРИЛОЖЕНИИ ACCESS

5.1. Назначение и способы проектирования форм

Для ввода, редактирования и просмотра данных Access предоставляет удобное и широко используемое средство —

формы, которые позволяют ограничить объем информации, отображаемой на экране, и представить ее в требуемом виде. Создавая форму, можно выбрать, из каких таблиц, запросов, какие поля и в какой последовательности должны быть в ней представлены. Можно разбить их на логически связанные группы, задать удобное расположение на экране, вставить иллюстрации, графически представлять хранящуюся в базе данных информацию. Таким образом, форма — объект базы данных, позволяющий создать удобный пользовательский интерфейс для работы с данными.

Создание форм требует дополнительных усилий, однако потраченное время возмещается за счет уменьшения ошибок при вводе, удобства доступа к информации, наглядности ее представления. Кроме того, они могут служить защитой базы данных от действий неквалифицированных пользователей.

Форма для ввода представляет собой бланк, подлежащий заполнению. При этом появляется возможность осуществить контроль вводимых данных и исключить ввод неверных. Бланк-форма упрощает процесс заполнения базы данными (благодаря чему в БД может вводить информацию пользователь невысокой квалификации), позволяет ограничить доступ пользователя к информации БД, заблокировав служебные и засекреченные поля или записи.

Формы могут быть простыми и составными (включающими другие формы), содержать различные элементы — поля БД, подписи к ним, списки, флажки, переключатели, кнопки, вкладки и другие элементы управления. Их можно спроектировать на базе одной или нескольких таблиц и/или запросов; на основе одной таблицы или запроса можно построить несколько форм. Имена полей в форме берутся из описания таблицы, а сами поля пользователь располагает в соответствии со своими вкусами и требованиями и вносит различные элементы оформления — линии, рисунки, заливку и др.

Форма может быть создана "вручную" — с помощью Конструктора форм, автоматизированным способом — с помощью

Мастера форм и автоматически — с использованием автоформы. *Конструктор форм* предоставляет пользователю набор инструментов, с помощью которого пользователь может создать форму соответственно своим вкусам и требованиям. *Мастер форм* руководит процессом проектирования форм, задает пользователю вопросы о структуре и оформлении формы, предлагая выбор из нескольких вариантов. *Автоформа* автоматически создает на основе выбранной таблицы одну из форм: в столбец, ленточную, табличную.

5.2. Инструментальные средства конструкторов форм

Для создания формы с помощью Конструктора необходимо открыть окно Конструктора формы, выполнив определенную последовательность действий:

окно *База данных/объекты Формы*/[Создать]/окно *Новая форма*/выбрать *Конструктор*/указать *источник формы* — таблицу или запрос/[ОК].

Окно Конструктора формы содержит следующие инструментальные средства конструирования:

1) панель Конструктора с набором графических кнопок, включаемая/выключаемая по команде *Вид/Панели инструментов*;

2) разделы проекта, в которых размещают различные графические объекты, определяющие вид и содержание формы, называемые *элементами управления*;

3) панель элементов управления, включаемая/выключаемая по команде *Вид/Панель элементов*;

4) список полей таблицы или запроса, служащих источником данных для формы, включаемый/выключаемый по команде *Вид/Список полей*;

5) сетка, отображаемая в разделах проекта для удобства проектирования, включаемая/ выключаемая по команде *Вид/Сетка*;

6) линейки горизонтальная и вертикальная, включаемые/выключаемые по команде *Вид/Линейка*.

Инструментальные средства конструирования предназначены для удобства проектирования форм.

По умолчанию в окне Конструктора формы выводится раздел проекта *Область данных*, в котором размещают поля БД из таблицы или запроса, а также вычисляемые поля. Кроме данного раздела, при конструировании могут быть востребованы и другие разделы:

Заголовок формы и *Примечание формы*, включаемые/выключаемые попарно соответственно по командам: *Вид / Заголовок / Примечание формы*;

Верхний колонтитул и *Нижний колонтитул*, включаемые/выключаемые попарно по команде *Вид / Колонтитулы*.

Заголовок формы содержит сведения, общие для всех записей (название формы, дата, время). *Примечание формы* отображает сведения, общие для всех записей (инструкции по работе с формой, командные кнопки, общие суммы в счетах). *Верхний колонтитул* и *Нижний колонтитул* используются в случае конструирования формы табличного вида: в *Верхнем колонтитуле* отображаются названия столбцов, в *Нижнем колонтитуле* — номер страницы, дата. Изменение высоты и ширины раздела производится путем перетаскивания соответственно нижней и правой границ раздела.

5.3. Элементы управления и работа с ними

Разделы проекта формы наполняются различными графическими объектами — *элементами управления*, используемыми для отображения данных, выполнения действий или в качестве украшений.

Элементы управления делятся на связанные и несвязанные. *Связанные элементы управления* присоединены к полю базовой таблицы или запроса и используются для просмотра, ввода или редактирования значений из полей БД. Ввод связанного элемента управления в проект формы осуществляется путем перетаскивания из окна со списком полей нужного поля в

раздел *Область данных*. В результате появляются два прямоугольника — подпись поля и поле.

Маркеры перемещения служат для отдельного перемещения подписи и поля. При установке на них указателя мыши появляется изображение руки с вытянутым указательным пальцем, и тогда следует выполнить перемещение.

Для одновременного перемещения подписи и поля указатель мыши устанавливается на границу выделенной подписи или поля. Когда появляется изображение раскрытой ладони, производят перетаскивание. Для удаления подписи осуществляют ее выделение и нажимают [Delete]; для удаления поля с подписью выделяют поле и нажимают [Delete].

Для *несвязанных элементов управления* источника данных не существует. Они используются для отображения линий, прямоугольников, рисунков. Ввод в проект формы такого элемента управления осуществляется следующим образом: производится щелчок по кнопке нужного элемента управления на панели элементов, а затем – щелчок в том разделе проекта, где требуется разместить этот элемент.

Выделение группы элементов управления производится путем щелчков по элементам управления, входящим в группу, при нажатой клавише [Shift] или очерчивания рамки вокруг этой группы. Чтобы изменить размер элементов группы, выровнять их, изменить интервал между ними, необходимо воспользоваться пунктом меню *Формат*. Возможны перенос и копирование элементов управления в другой раздел проекта посредством команд для работы с буфером обмена.

Для придания элементу управления необходимых свойств его выделяют и вводят команду *Вид/Свойства* или вызывают контекстное меню и выбирают в нем пункт *Свойства*. В результате появляется окно свойств элемента управления, в котором можно произвести его форматирование, например, с помощью вкладки *Макет*.

5.4. Рекомендации для создания формы

При необходимости спроектировать форму, в которой должны быть определенные заголовки, поля БД, вычисляемое поле и комментарий, следует:

1) открыть окно *Конструктора формы*, не забыв указать источник формы;

2) настроить это окно для работы, включив (если не включены) линейки, сетку, панель элементов и три раздела: *Заголовок формы*, *Область данных*, *Примечание формы*;

3) задать заголовок формы, для чего в раздел *Заголовок формы* вставить элемент управления *Надпись* и ввести текст заголовка;

4) из окна со списком полей перетащить в раздел *Область данных* те поля БД, значения которых должны просматриваться по форме, и расположить их в нужном порядке;

5) создать вычисляемое поле, для чего вставить в раздел *Область данных* свободное поле, используя элемент управления *Поле*; в само поле ввести выражение для расчета, а в подпись — имя вычисляемого поля;

6) задать необходимый комментарий в разделе *Примечание формы*, используя элемент управления *Надпись*;

7) произвести желаемое оформление формы.

5.5. Работа с формой

Спроектированная форма приобретает реальный вид в окне формы, открываемом из окна БД с помощью кнопки [*Открыть*] и из окна Конструктора по команде *Вид / Режим формы* или *Вид / Режим таблицы*. В режиме формы в форме отображается только одна запись, а в режиме таблицы — группа записей. В обоих режимах можно редактировать данные, просматривать их с помощью кнопок перехода к записям (к предыдущей, к следующей, к первой, к последней). Кнопка перехода к пустой записи позволяет добавлять в базу новые данные.

Форму можно сохранить, введя в окне Конструктора или в окне формы одну из команд: *Файл / Сохранить как, Файл / Сохранить* или просто закрыв эти окна.

Форма может быть предварительно просмотрена из окна БД, из окна Конструктора и из окна формы по команде *Файл / Предварительный просмотр* и распечатана по команде *Файл / Печать*. При печати ее заголовок появляется только в начале первой страницы, а ее примечание — в конце последней. *Верхний* и *нижний колонтитулы* видны только при печати соответственно в начале и в конце каждой страницы. Изменения в форму вносятся в окне Конструктора.

З а д а н и е 23

Создать по таблице «Detail» экранную табличную автоформу. В режиме *Конструктор* расширить раздел *Примечание*. Вставить в этот раздел элемент управления *Календарь*. В рабочей области выделить все поля, щелкнув мышью слева на линейке. Для всех полей задать цвет фона — желтый, тип фона — обычный, оформление — утопленное, цвет границы и цвет текста — темно-зеленый, шрифт — Times Roman Cyr, размер шрифта — 11, насыщенность — плотный, выравнивание текста — по центру. Использовать для этого контекстное меню: команда *Свойства/вкладка Макет* диалогового окна *Несколько элементов управления*. Расширить раздел *Заголовок формы*. Сдвинуть все элементы этого раздела вниз на 2,5 см. Создать заголовок формы — Detail. У этого элемента задать размер шрифта — 14, сверхжирный, Courier Cyr, цвет фона — желтый, цвет текста — темно-зеленый, оформление — приподнятое. Сохранить форму под именем Detail. Просмотреть форму в режиме таблицы и в режиме формы. Определить количество записей, выводимых в экранную форму. Сохранить под именем *Форма_1*.

З а д а н и е 24

Создать по таблице «Invoices» экранную автоформу в столбец. В режиме *Конструктор* расширить разделы *Заголовок*

формы и Примечание формы. В разделе *Заголовок формы* создать надпись “Таблица Invoices”, в разделе *Примечание формы* — “Автоформа в столбец”, задать размер области данных — 6 см. Просмотреть форму в режиме таблицы и в режиме формы. Сохранить под именем *Форма_2*.

З а д а н и е 2 5


Создать экранную форму, в которой из таблицы «Customer» выводятся поля Company, Address, City, State, Zip, Lat, Long, Ytdpurch, из таблицы «Office» — Ono, Address, City, State, Zip, Phone. Использовать для этого в качестве источника данных соответствующий запрос. Использовать Мастер форм. Просмотреть форму в режиме формы. Сохранить под именем *Форма_3*.

З а д а н и е 2 6

Создать по таблицам «Office» и «Customer» экранную форму с подчиненной формой, используя Мастер форм. Выводимые поля взять из задания 2. В подчиненной форме изменить заголовок Company на *Компании*. Вставить гиперссылки для вызова документа из приложений Word, Excel, PowerPoint. Просмотреть форму в режиме формы. Сохранить под именем *Форма_4*.

З а д а н и е 2 7

Создать связанные формы по таблицам «Customer» (выводимые поля из задания 2) и «Invoices» (выводимые поля — Idate, Itotal, Salesman). Основную форму сохранить под именем Cust_s_inv, связанную — под именем *Связанная*. В основной форме создать вычисляемое поле *Сумма*, которое содержит сумму значений полей Lat, Long, Ytdpurch, для чего перейти в режим *Конструктор*. Элемент управления *Поле пе-*

ретащить в область данных формы Cust_s_inv. Изменить надпись на *Сумма*. Выделить поле, вызвать контекстное меню, выбрать команду *Свойства...*, вкладка *Данные*, строка *Данные*. Выбрать *Построитель выражений*,  *Все формы*, Cust_s_inv, поля Lat + Long + Ytdpurch. Просмотреть форму в режиме формы. Применить фильтр *Расширенный* по полю State = CA.

З а д а н и е 28

Построить плоскую гистограмму по запросу *Для_формы* по таблице «Office», содержащему поля Ono, City, Ytdsales и вычисляемое поле *Расчет* = (Ytdsales + Zmin)/10. Гистограмма должна отображать объемы продаж (Ytdsales) и рассчитанную величину (*Расчет*) по офисам компаний (Ono) или по городам (City), в которых расположены эти офисы. Просмотреть гистограмму в режиме формы. Изменить тип гистограммы в режиме Конструктора на объемный. Отредактировать заголовок гистограммы, удалить легенду, изменить шрифты, развернуть гистограмму. Просмотреть результат в режиме формы. Сохранить под именем *Диаграмма*.

Тема 6. ОТЧЕТЫ В ПРИЛОЖЕНИИ ACCESS

6.1. Создание отчета

Отчет является важным средством извлечения информации из БД и вывода ее на экран или на печать в виде, удобном для восприятия и анализа пользователем. В отчете можно сортировать и группировать данные, осуществлять расчеты в строках и проводить итоговые вычисления над группами строк и над всеми строками с использованием статистических функций. Отчет может основываться на множестве таблиц, представлять сложные зависимости между различными наборами данных, быть составным, — т.е. включать другие отчеты.

Access предоставляет большие возможности для оформления отчетов: шрифтовое, фоновое и цветовое оформление, обрамление, рисунки, деловую графику, вставку объектов других приложений.

Существует 3 способа создания отчета: с помощью Конструктора, с помощью Мастера отчетов и автоматическое — автоотчеты. Конструктор дает возможность самостоятельного проектирования отчетов; Мастер отчетов позволяет создать отчет на основе ответов пользователя на вопросы, касающиеся структуры, содержания и оформления отчета; автоотчет создает отчет в столбце и ленточный.

Для создания отчета с помощью Конструктора необходимо открыть окно Конструктора отчета, выполнив следующие действия:

открыть БД/ выбрать объект *Отчеты*/в меню окна базы данных выбрать *Создать*/в открывшемся окне *Новый отчет* выбрать команду *Конструктор*/указать источник отчета — таблицу, запрос или другой отчет/нажать [ОК].

Так как методика проектирования формы и отчета одинакова, то и окно Конструктора отчета содержит те же инструментальные средства конструирования, что и окно Конструктора формы:

1) панель Конструктора с набором графических кнопок, включаемая/выключаемая по команде *Вид/Панели инструментов*;

2) разделы проекта, в которых размещаются различные элементы управления: заголовок отчета, верхний колонтитул, заголовок группы, область данных, примечание группы, нижний колонтитул, примечание отчета;

3) панель элементов управления, включаемая/выключаемая по команде *Вид/Панель элементов*;

4) список полей таблицы или запроса, служащих источником данных для отчета, включаемый/выключаемый по команде *Вид/Список полей*;

5) сетка, отображаемая в разделах проекта для удобства проектирования, включаемая/ выключаемая по команде *Вид/Сетка*;

б) линейки — горизонтальная и вертикальная, включаемые/выключаемые по команде *Вид/Линейка*, также предназначенные для удобства проектирования отчетов.

6.2. Работа с отчетом

Спроектированный отчет можно предварительно просмотреть в одном из двух представлений: общий вид и страничный вид. Общий вид доступен только из окна Конструктора по команде *Вид/Образец* и позволяет оценить внешний вид отчета, не вникая в подробности. Страничный вид дает возможность увидеть отчет таким, каким он будет после печати. Страничное представление отчета можно получить из окна Конструктора по команде *Вид/Предварительный просмотр*, а также из окна БД по команде *Файл/Предварительный просмотр*.

Печать отчета можно выполнить из окна БД, из окна Конструктора или из окна предварительного просмотра с помощью команды *Файл/Печать*. Установка параметров печати осуществляется перед печатью по команде *Файл/Параметры страницы*. Изменения в отчет вносятся в окне Конструктора.

З а д а н и е 29

Создать и просмотреть ленточный в столбец автоотчет по данным двух таблиц: «Office», «Invoices». Для выполнения задания использовать соответствующий запрос. В режиме Конструктора в заголовок отчета вставить объект WordArt, используя *Буфер обмена*. Сохранить отчет под именем *Отчет_1*.

З а д а н и е 30

Создать с помощью Мастера отчет по таблице «Customer». Выбрать для отчета поля Company, Address, City, State, Lat, Long. Задать группирование по полям State, City. Внутри группы City упорядочить данные по полю Company. По числовому полю Lat задать подведение итогов, по группе State — сумми-

рование, по числовому полю Long определить максимальное и минимальное значение в группе данных State. Отредактировать отчет в режиме Конструктора: выделить область примечаний группы City; вызвать контекстное меню, выбрать команду *Сортировка и группировка*; в появившемся диалоговом окне для поля City убрать строку *Примечание группы*, задав для него значение *Нет*. Сохранить отчет под именем *Отчет_2*.

З а д а н и е 3 1

Построить отчет с помощью Мастера на базе запроса, построенного по таблицам «Customer» и «Invoices», включающего поля: Company, State, City, Lat, Long, Idate, Itotal. Данные запроса должны быть упорядочены по полю Company. При построении отчета использовать данные предыдущего задания. В режиме Конструктора отредактировать отчет, выделив цветной рамкой заголовок группы — слово State — и итоговые данные по этой группе. Сохранить отчет под именем *Отчет_3*.

З а д а н и е 3 2

Построить многотабличный отчет с помощью Мастера по таблицам «Customer» и «Invoices», используя данные предыдущего задания. Подготовить в графическом редакторе логотип фирмы, готовящей отчет. В режиме Конструктора вставить логотип в заголовок отчета, убрать итоги по полю City, уменьшить размеры надписи и поля State, передвинуть их к левому краю области, в которой они находятся. Задать одинаковый цвет тексту поля State и итогам по этому полю. В раздел *Примечание отчета* вставить элемент управления *Календарь*. Сохранить отчет под именем *Отчет_4*.

З а д а н и е 3 3

В режиме *Конструктор* построить многотабличный отчет по таблицам «Office» и «Customer», используя для этого пред-

варительно созданный запрос, в который включены поля: Ono, City (из таблицы «Office»), Company, City, State, Ytdpurch, Lat, Long (из таблицы «Customer»). В отчет добавить новое вычисляемое поле $Formula = (Ytdpurch + Lat)/Long$ и сгруппировать его по полям Ono, City (из таблицы «Office»). Подвести итоги по группе и общие итоги по отчету для полей Long, Lat, Ytdpurch, Formula, найти минимальное, максимальное, среднее значения и сумму для соответствующих полей.

Для создания отчета необходимо:

1) открыть окно Конструктора отчета, не забыв указать источник отчета;

2) настроить окно для работы, включив (если не включены) линейки, сетку, панель элементов и все разделы;

3) задать заголовок отчета, для чего в раздел *Заголовок отчета* вставить элемент управления *Надпись* и ввести текст заголовка «Отчет по офисам компании»;

4) спроектировать строки отчета:

а) из окна со списком полей перетащить в раздел *Область данных* поля, значения которых должны выводиться в строках отчета;

б) в разделе создать вычисляемое поле, значения которого будут выводиться в расчетных столбцах, для чего вставить свободное поле, используя элемент управления *Поле*, и ввести в него необходимое выражение;

в) подписи всех полей переместить из раздела *Область данных* в раздел *Верхний колонтитул*;

г) поля в разделе *Область данных* расположить в строку в нужном порядке, выровнять их по верхнему краю раздела и сделать интервалы между ними равными по горизонтали;

5) спроектировать шапку отчета:

а) в разделе *Верхний колонтитул* подписи полей расположить в строку над соответствующими полями;

б) названия столбцов выровнять по верхнему краю раздела;

б) спроектировать частные итоги:

а) в раздел *Заголовок группы* переместить названия и значения полей Ono, City (из таблицы «Office»);

б) в раздел *Примечание группы* вставить свободные поля, используя элемент управления *Поле*, и в них ввести выражения для подсчета частных итогов; подписи этих полей удалить;

в) в строке слева ввести название для строк с частными итогами при помощи элемента управления *Надпись*;

7) спроектировать общие итоги:

а) в раздел *Примечание отчета* вставить свободные поля, используя элемент управления *Поле*, и в них ввести выражения для подсчета общих итогов; подписи этих полей удалить;

б) созданные вычисляемые поля расположить в строку под полями, по которым подводятся общие итоги, и выровнять их по верхнему краю раздела; в строке слева ввести название для строки с общими итогами при помощи элемента управления *Надпись*;

8) предусмотреть нумерацию страниц отчета, воспользовавшись командой *Вставка/Номера страниц*;

9) произвести оформление отчета, например:

а) в разделе *Заголовок отчета* провести нижнюю и верхнюю линии шапки отчета, используя элемент управления *Линия*;

б) выделить разными шрифтами шапку отчета, вычисляемые поля для частных и общих итогов;

10) осуществить предварительный просмотр отчета; если отчет при просмотре соответствует заданию, сохранить и напечатать его, если нет, — модифицировать его, используя Конструктор отчета.

З а д а н и е 3 4

Подготовить отчет для печати каждой группы на отдельной странице. Для выполнения задания воспользоваться *Справкой*.

Тема 7. СТРАНИЦЫ ДОСТУПА К ДАННЫМ

Страницы доступа к данным — специальный тип Web-страниц, предназначенный для просмотра и работы с данны-

ми, хранящимися в базе данных MS Access, через Internet или Интрасеть.

Страницы доступа к данным разрабатываются в MS Access, однако представляют собой отдельный HTML-документ, хранящийся за пределами базы данных. При создании страницы доступа к данным MS Access автоматически добавляет ярлык на нее в окно базы данных. Таким образом, доступ к странице можно получить как из MS Internet Explorer (для работы с данными через компьютерную сеть), так и из MS Access.

Разработка страниц доступа к данным аналогична разработке форм и отчетов. Страницы доступа к данным используются для просмотра, ввода и редактирования, наглядного отображения, анализа данных. Более подробно эта тема рассматривается в курсе “Сетевые технологии”.

З а д а н и е 35

С помощью Мастера создать по таблице «Office» страницу доступа к данным и просмотреть ее в MS Internet Explorer.

Тема 8. МАКРОСЫ

Для автоматизации часто выполняемых операций служат макросы. **Макрос** — объект базы данных, представляющий собой последовательность операций, собранных в виде инструкций таким образом, чтобы их можно было выполнять, используя только одну команду. Макрос запускается, чтобы выполнить всю последовательность необходимых действий, каждое из которых реализуется определенной макрокомандой. Создание макросов осуществляется путем выбора нужных макрокоманд и задания дополнительных параметров, используемых ими при выполнении.

Для запуска макроса на выполнение чаще всего используются формы. Макросы, как правило, автоматически выполняются в ответ на определенное событие. Например, с помощью

макроста можно осуществить проверку значения какого-либо поля при изменении его содержимого, открыть форму, запрос, распечатать отчет или запустить другой макрос.

Технология создания макросов в приложении Access отличается от технологии создания их в приложениях Excel, Word, PowerPoint. Макросы создаются в окне *Макрос*, для открытия которого в окне БД надо выбрать объекты *Макросы* и кнопку [Создать] в меню окна БД. Откроется окно создания макрокоманд в режиме *Конструктор*, имеющее по умолчанию два столбца — *Макрокоманда* и *Примечания*.

Макрокоманды можно ввести путем:

- 1) ввода их с клавиатуры;
- 2) выбора их имен из раскрывающегося списка;
- 3) перетаскивания объекта из окна БД в ячейку (при этом автоматически добавляется команда открытия объекта).

Наиболее просто использовать второй или третий способ ввода макрокоманд. В списке *Макрокоманды* выбираются необходимые макрокоманды, при этом в нижней части окна *Макрос* открываются аргументы, соответствующие выбранной макрокоманде, указывающие, к какому объекту (таблица, запрос, форма, отчет) применяется данная макрокоманда, и задающие условия выполнения действия. Например, макрос должен открывать таблицу "Customer". В списке макрокоманд надо отыскать команду *ОткрытьТаблицу*. Для этого достаточно набрать первую букву искомой макрокоманды, и список представит все команды, начинающиеся на эту букву, затем набрать вторую букву и т.д. Как только команда будет выбрана, внизу окна появится список аргументов этой команды: *Имя таблицы*, *Режим*, *Режим данных*. Для аргумента *Имя таблицы* надо задать имя Customer, для аргумента *Режим* — выбрать *Таблицу*, *Конструктор* или *Просмотр* (в нашем случае — *Таблицу*), для аргумента *Режим данных* задать один из режимов: *Изменение*, *Добавление*, *Только для чтения* (в нашем примере — *Добавление*).

Для работы с одним объектом БД могут понадобиться несколько макросов, каждый из которых автоматизирует от-

дельный процесс. Один и тот же отчет может быть нужен для использования в режиме просмотра, для печати, редактирования, выборки из отчета определенных записей с целью переноса в таблицу Excel, в документ Word или в подготавливаемую презентацию в PowerPoint. Такие макросы могут быть созданы отдельно и запускаться по мере надобности. Можно объединить их в один макрос под общим именем, создав внутри него соответствующую группу макросов и присвоив каждой группе уникальное имя, которое записывается в отдельном столбце окна *Конструктор макроса*. Столбец выводится при нажатии пиктограммы *Имена макросов*.

З а д а н и е 3 6

Создать макрос, который открывает таблицу Office, запрос *Выбор_1*, форму 5, используя перетаскивание объектов. Макрос размещает открытые окна вместе с окном БД в окне приложения Access по макрокоманде *ВыполнитьКоманду*. Способ размещения указывается в аргументе: *РядомВертикально* или *РядомГоризонтально*.

З а д а н и е 3 7

Создать макрос под именем *Объекты_Свернуть*; в нем создать макрогруппу *Таблица*, в которой открывается таблица «Customer», которая по команде *Свернуть* сворачивается. Создать макрогруппу *Запрос*, в которой открывается и сворачивается запрос, выбранный исполнителями. Создать макрогруппу *Форма*, в которой открывается форма *Форма_3*. К ней применяется фильтр *Запрос_ТХ*, в котором данные таблицы Customer выбраны для штата ТХ. Создать макрогруппу *Горизонт*, в которой выполняется команда, располагающая в окне Access открытые окна по горизонтали. Запуск макрогрупп осуществляется после выполнения команды *Сервис/Макрос/Выполнить макрос... Запустить все макрогруппы*. Отредактировать мак-

рогруппу *Таблица*, поставив первой команду *Свернуть*. Запустить макрогруппу. Для чего была использована эта команда?

З а д а н и е 38

Создать макрос под именем *Отчеты*, а в нем – макрогруппу *Просмотр*, в которой *Отчет_1* открывается в режиме *Просмотр*. Создать макрогруппу *Печать* для печати отчета *Отчет_1*; макрогруппу *Редактор*, в которой должен редактироваться *Отчет_2*; макрогруппу *Выбор* для выборки из *Отчет_3* записей, относящихся к штатам Мичиган (MI) и Нью-Йорк (NY). Перенести отобранные данные в таблицу "Excel", и по ним построить диаграмму.

З а д а н и е 39

В форме *Cust_s_inv* создать кнопку, открывающую созданную в задании 6 (тема 5) гистограмму, для чего необходимо создать макрос, открывающий форму *Диаграмма*. Кнопку разместить в области *Примечание формы*, для чего на панели элементов отключить кнопку [Мастер], нажать элемент *Кнопка* и в области *Примечание формы* нарисовать курсором мыши кнопку. В свойствах кнопки задать выполнение созданного макроса при нажатии кнопки, для чего надо выделить кнопку, в пиктографическом меню выбрать пиктограмму *Свойства*, в диалоговом окне *Кнопка* на вкладке *Макет* в строке *Подпись* задать надпись на кнопке — *Диаграмма*. На вкладке *События*, в строке *Нажатие кнопки* выбрать имя макроса, открывающего форму *Диаграмма*. С помощью Мастера создать кнопку *Назад* для закрытия формы *Диаграмма*.

Л и т е р а т у р а

1. Бекаревич, Ю.Б., Пушкина, Н.В. MS Access 2002. – СПб.: БХВ-Санкт-Петербург, 2002.
2. Вейскас, Дж. Эффективная работа с Microsoft Access 2000. – СПб: Издательство «Питер», 2000.
3. Экономическая информатика: учебник/ Под ред. П.В. Коныховского, Д.Н. Колесова. – СПб.: Питер, 2000.
4. Гончаров, А. Access 97 в примерах. – СПб.: Питер, 1997.
5. Оскерко, В.С. Технологии организации, хранения и обработки данных. – Мн.: БГЭУ, 2002.
6. Основы компьютерных технологий в образовании: В 3 ч. Ч. 3. Технологии обработки данных: учеб. пособие / Г.М. Троян [и др.]; под ред. Г.М. Троян. – Мн.: РИВШ БГУ, 2002.

С о д е р ж а н и е

В в е д е н и е	3
Справочная система Access	5
Объекты базы данных и их размещение	5
Основные термины и определения	7
Тема 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	9
Тема 2. СОЗДАНИЕ И КОРРЕКТИРОВКА БАЗЫ ДАННЫХ В СИСТЕМЕ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ MS ACCESS 2000	13
2.1. Пользовательский интерфейс Access	13
2.2. Создание таблицы	14
Тема 3. СОЗДАНИЕ СХЕМЫ ДАННЫХ	19
Тема 4. СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ЗАПРОСОВ ...	20
Тема 5. ФОРМЫ В ПРИЛОЖЕНИИ ACCESS	27
5.1. Назначение и способы проектирования форм	27
5.2. Инструментальные средства конструкторов форм	29
5.3. Элементы управления и работа с ними	30
5.4. Рекомендации для создания формы	32
5.5. Работа с формой	32
Тема 6. ОТЧЕТЫ В ПРИЛОЖЕНИИ ACCESS	35
6.1. Создание отчета	35
6.2. Работа с отчетом	37
Тема 7. СТРАНИЦЫ ДОСТУПА К ДАННЫМ	40
Тема 8. МАКРОСЫ	41
Л и т е р а т у р а	45

Учебное издание

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ACCESS

Лабораторный практикум
для студентов специальности 1–27 01 01
«Экономика и организация производства»

Составители:
ГУРЬЕВА Ольга Олеговна
ГУСЕВА Лариса Петровна

Редактор Т.А. Палилова
Компьютерная верстка Л.Н. Юргилевич

Подписано в печать 19.10.2005.

Формат 60x84 1/16. Бумага офсетная.

Отпечатано на ризографе. Гарнитура Таймс.

Усл. печ. л. 2,8. Уч.-изд. л. 2,2. Тираж 100. Заказ 28.

Издатель и полиграфическое исполнение:
Белорусский национальный технический университет.
ЛИ № 02330/0056957 от 01.04.2004.
220013, Минск, проспект Независимости, 65.