Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра автоматического управления

М.А. Антипова, С.В. Снисаренко, А.А. Седушкин

ИНФОРМАЦИОННЫЕ ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ

Методическое пособие

к лабораторным работам для студентов специальностей
53 01 03 «Автоматическое управление в технических системах» и
53 01 07 «Информационные технологии и управление в технических системах»
всех форм обучения

Минск 2003

УДК 681.5 (075.8)
ББК 32.965 я 73
А 72

Р е ц е н з е н т :
зав. кафедрой теоретических основ электротехники БГУИР,
д-р техн. наук, проф. Л.Ю. Шилин

Антипова М.А.

А 72 Информационные основы систем управления: Метод. пособие к лаб. работам для студентов спец. 53 01 03 «Автоматическое управление в технических системах» и 53 01 07 «Информационные технологии и управление в технических системах» всех форм обучения / М.А. Антипова, С.В. Снисаренко, А.А. Седушкин. – Мн.: БГУИР, 2003. – 64 с.

ISBN 985-444-503-8.

В пособии приведены описание и порядок выполнения шести лабораторных работ по курсу «Информационные основы систем управления». Содержание работ определено рабочей программой курса в соответствии с учебными планами специальностей 53 01 03 «Автоматическое управление в технических системах» и 53 01 07 «Информационные технологии и управление в технических системах».

В каждой лабораторной работе изложены краткие теоретические сведения, представлены задания и методики выполнения работ, приведены примеры работы в системе управления базами данных Microsoft Access.

УДК 681.5 (075.8)
ББК 32.965 я 73

ISBN 985-444-503-8

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Хранение информации – одна из важнейших функций компьютера. Самым распространенным средством такого хранения являются базы данных. База данных – это файл специального формата, содержащий информацию, структурированную заданным образом.

Структура базы данных. Большинство баз данных имеют табличную структуру. Как мы знаем, в табличной структуре адрес данных определяется пересечением строк и столбцов. В базах данных столбцы называются полями, а строки – записями. Поля образуют структуру базы данных, а записи составляют информацию, которая в ней содержится. Для того чтобы легко усвоить понятие структуры базы данных, надо представить себе пустую базу, в которой пока еще нет никаких данных. Несмотря на то, что данных в базе нет, информация в ней все-таки есть. Это структура базы, т.е. набор полей. Они определяют, что будет записано в эту базу и в каком виде.

Простейшие базы данных. Простейшие базы можно создавать, не прибегая к специальным программным средствам. Чтобы файл считался базой данных, информация в нем должна иметь структуру (поля) и быть форматирована так, чтобы содержимое соседних полей легко различалось. Простейшие базы можно создавать даже в текстовом редакторе «Блокнот», т.е. обычный текстовый файл при определенном форматировании тоже может считаться базой данных. Существует по крайней мере два формата текстовых баз данных: с заданным разделителем; с фиксированной длиной поля. Несмотря на «примитивность» таких текстовых баз данных, мощные системы управления базами данных позволяют импортировать подобные файлы и преобразовывать их в «настоящие» базы данных.

СУБД Access. Системы управления базами данных (СУБД) – это программные средства, с помощью которых можно создавать базы данных, наполнять их и работать с ними. В мире существует немало различных систем управления базами данных. Многие из них на самом деле являются не законченными продуктами, а специализированными языками программирования, с помощью которых каждый, освоивший данный язык, может сам создавать такие структуры, какие ему удобны, и вводить в них необходимые элементы управления. К подобным языкам относятся Clipper, Paradox, FoxPro и др. Необходимость программировать всегда сдерживала широкое внедрение баз данных в малом бизнесе. Крупные предприятия могли позволить себе сделать заказ на программирование специализированной системы «под себя». Малым предприятиям зачастую не по силам было не только решить, но даже и правильно сформулировать эту задачу. Положение изменилось с появлением в составе пакета Microsoft Office системы управления базами данных Access. С помощью Access обычные пользователи получили удобное средство для создания и эксплуатации достаточно мощных баз данных без необходимости что-либо программировать. В то же время работа с Access

не исключает возможности программирования. При желании систему можно развивать и настраивать собственными силами. Для этого надо владеть основами программирования на языке Visual Basic. Еще одним дополнительным достоинством Access является интегрированность этой программы с Excel, Word и другими программами пакета Office . Данные, созданные в разных приложениях, входящих в этот пакет, легко импортируются и экспортируются из одного приложения в другое. Мы будем рассматривать работу СУБД на примере Access XP, которая установлена в учебных лабораториях.

ЛАБОРАТОРНАЯ РАБОТА № 1

ЗНАКОМСТВО С СУБД ACCESS. ПРЕДМЕТНАЯ ОБЛАСТЬ КАК ПЕРВЫЙ ЭТАП ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

Цель работы: познакомиться с интерфейсом СУБД Access, получить начальные навыки работы с ее компонентами путем создания базы данных при помощи шаблона, научиться разрабатывать предметную область по предложенной теме индивидуального задания.

Краткие теоретические сведения

Объекты Access. Исходное окно Access отличается простотой и лаконичностью. Шесть кнопок слева представляют шесть видов объектов, с которыми работает программа. Таблицы – основные объекты базы данных. В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Запросы – это специальные структуры, предназначенные для обработки данных базы. С помощью запросов данные упорядочивают, фильтруют, отбирают, изменяют, объединяют, т.е. обрабатывают. Формы – это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся. Отчеты – это «формы наоборот». С их помощью данные выдают на принтер в удобном и наглядном виде. Макросы – это макрокоманды. Если какие-то операции с базой производятся особенно часто, имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш. Модули – это программные процедуры, написанные на языке Visual Basic. Если стандартных средств Access не хватает для удовлетворения требований заказчика, программист может расширить возможности системы, написав для этого необходимые модули или использовав готовые.

Режимы работы с Access. С организационной точки зрения в работе с любой базой данных есть два разных режима: проектировочный и эксплуатационный (пользовательский). Создатель базы имеет право создавать в ней новые объекты (например таблицы), задавать их структуру, менять свойства полей, устанавливать необходимые связи. Он работает со структурой базы и имеет полный доступ к базе. У одной базы может быть один, два или несколько разработчиков. Пользователь базы – это лицо, которое наполняет ее информацией с помощью форм, обрабатывает данные с помощью запросов и получает результат в виде результирующих таблиц или отчетов. У одной базы могут быть миллионы пользователей, и, конечно, доступ к структуре базы для них закрыт.

На стартовом окне базы данных кроме шести кнопок для основных объектов есть три командные кнопки: «Открыть», «Конструктор», «Создать». С их помощью и выбирается режим работы с базой.

1. Кнопка «Открыть» открывает избранный объект. Если это таблица, то ее можно просмотреть, внести новые записи или изменить те, что были внесены ранее.

2. Кнопка «Конструктор» тоже открывает избранный объект, но по-другому. Она открывает его структуру и позволяет править не содержимое, а устройство. Если это таблица, в нее можно вводить новые поля или изменять свойства существующих полей. Если это форма, в ней можно изменять или создавать элементы управления. Очевидно, что этот режим служит не для пользователей базы, а для ее разработчиков.

3. Действие командной кнопки «Создать» соответствует ее названию. Она служит для создания новых объектов. Этот элемент управления тоже предназначен для проектировщиков базы. Таблицы, запросы, формы и отчеты можно создавать несколькими разными способами: автоматически, вручную или с помощью Мастера. О достоинствах и недостатках этих методов мы поговорим при более подробном рассмотрении объектов Access.

Прежде чем приступать к созданию системы автоматизированной обработки информации, разработчик должен сформировать понятия о предметах, фактах и событиях, которыми будет оперировать данная система. Поэтому первым этапом проектирования любой базы данных является определение и разработка предметной области. Напомним определение:

Предметная область – часть реального мира, подлежащая изучению с целью организации управления и, в конечном счете, автоматизации. Предметная область представляется множеством фрагментов, например, предприятие – цехами, дирекцией, бухгалтерией и т.д. Каждый фрагмент предметной области характеризуется множеством объектов и процессов, использующих объекты, а также множеством пользователей, характеризующихся различными взглядами на предметную область.

В теории проектирования информационных систем предметную область (или, если угодно, весь реальный мир в целом) принято рассматривать в виде трех представлений:

1) представление предметной области в том виде, как она реально существует;

2) как ее воспринимает человек (имеется в виду проектировщик базы данных);

3) как она может быть описана с помощью символов.

Иначе говоря, мы имеем дело с реальностью, описанием (представлением) реальности и данными, которые отражают это представление.

Отсюда вытекают основные этапы, на которые разбивается процесс проектирования базы данных информационной системы:

1. Функциональное моделирование – построение диаграмм потоков данных (иллюстрация предметной области).

2. Концептуальное проектирование – сбор, анализ и редактирование требований к данным. Для этого осуществляются следующие мероприятия:

• обследование предметной области, изучение информационной структуры;

• выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами; моделирование и интеграция всех представлений.

По окончании данного этапа получаем концептуальную модель, инвариантную к структуре базы данных. Часто она представляется в виде модели «сущность—связь».

3. Логическое проектирование – преобразование требований к данным в структуры данных. На выходе получаем СУБД, ориентированную структуру базы данных и спецификации прикладных программ. На этом этапе часто моделируют базы данных применительно к различным СУБД и проводят сравнительный анализ моделей.

4. Физическое проектирование – определение особенностей хранения данных, методов доступа и т.д.

Более подробно остановимся на первом этапе. Одним из методов представления предметной области является построение диаграммы потоков данных Йордана – Де Марко.

Диаграммы потоков данных (DFD – Data Flow Diagramm) строятся из следующих элементов, представленных в табл. 1.

Таблица 1

Элемент	Описание	Обозначение
Функция	Действие, выполняемое моделируемой системой	Имя функции
Поток данных	Объект, над которым выполняется действие. Может быть информационным (логическим) или управляющим. (Управляющие потоки обозначаются пунктирной линией со стрелкой)	Имя объекта →
Хранилище данных	Структура для хранения информационных объектов	Имя объекта
Внешняя сущность	Внешний по отношению к системе объект, обменивающийся с нею потоками данных	Имя внешнего объекта

Такой тип обозначений элементов DFD-диаграммы получил название «нотация Йордана – Де Марко», по именам разработавших его специалистов.

Функции, хранилища и внешние сущности на DFD-диаграмме связываются дугами, представляющими потоки данных. Дуги могут разветвляться или сливаться, что означает соответственно разделение потока

данных на части либо слияние объектов. При интерпретации DFD-диаграммы используются следующие правила:

1. Функции преобразуют входящие потоки данных в выходящие.

2. Хранилища данных не изменяют потоки данных, а служат только для хранения поступающих объектов.

3. Преобразования потоков данных во внешних сущностях игнорируются.

Помимо этого, для каждого информационного потока и хранилища определяются связанные с ними элементы данных. Каждому элементу данных присваивается имя, также для него может быть указан тип данных и формат. Именно эта информация является исходной на следующем этапе проектирования – построении модели «сущность—связь». При этом, как правило, информационные хранилища преобразуются в сущности, проектировщику остается только решить вопрос с использованием элементов данных, не связанных с хранилищами.

Построим DFD-диаграмму для предприятия, строящего свою деятельность по принципу «изготовление на заказ» (рис. 1). На основании полученных заказов формируется план выпуска продукции на определенный период. В соответствии с этим планом определяются потребность в комплектующих изделиях и материалах, а также график загрузки производственного оборудования. После изготовления продукции и проведения платежей готовая продукция отправляется заказчику.

Эта диаграмма представляет самый верхний уровень функциональной модели. Естественно, это весьма грубое описание предметной области. Уточнение модели производится путем детализации необходимых функций на DFD-диаграмме следующего уровня.

Задание к лабораторной работе

1. Создать базу данных «Заказы на работу» при помощи шаблона по приведенному ниже примеру.

Пример создания базы данных. Запустить на выполнение приложение Microsoft Access. Открыть меню «Файл» и в окне «Создание» выбрать «Создание при помощи шаблона – Общие шаблоны», где на вкладке «База данных» отображается галерея шаблонов для создания БД, используемых в различных прикладных областях.

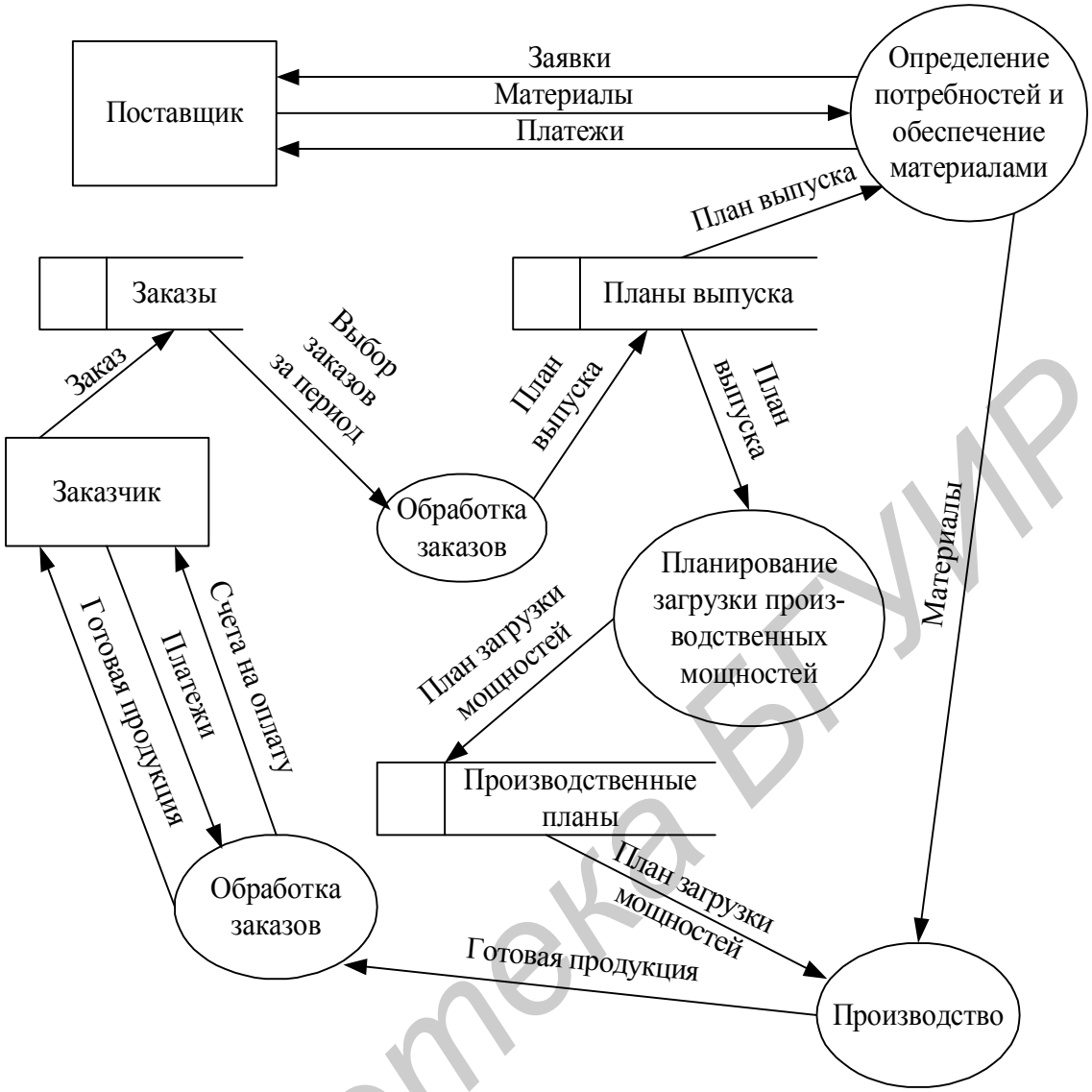

Рис. 1. Пример DFD-диаграммы

Выбрать шаблон «Заказы на работы». В появившемся на экране диалоговом окне «Файл новой базы данных» выбрать диск и папку, в которых будет сохраняться создаваемая БД, в разделе «Имя файла» будет предложено имя файла для создаваемой БД.

Через небольшой промежуток времени появится окно «Создание баз данных», в котором будет указано, какие сведения будет содержать создаваемая БД. Нажать «Далее».

Появившееся на экране следующее окно содержит в левой части имена таблиц, которые будут созданы, в правой части список обязательных и необязательных полей для выбранной таблицы. Выполнить щелчок по кнопке «Далее».

В следующих окнах выбрать вид оформления экрана и вид оформления отчета на печати, просмотрев все предлагаемые варианты, затем выполнить щелчок по кнопке «Далее».

В следующем окне ввести название базы данных или воспользоваться стандартным. Если необходимо использовать рисунок для оформления отчетов, включить параметр «Да» в разделе «Добавить рисунок во все отчеты?» При этом станет активной кнопка, позволяющая получить доступ к дискам и папкам для выбора нужного для вставки рисунка. Затем выполнить щелчок по кнопке «Далее», а в следующем окне — по кнопке «Готово».

Для создания всех объектов базы данных потребуется некоторое время. После завершения процесса создания БД на экране появится кнопочная форма для работы с БД «Заказы на работы».

Ознакомиться с каждым из представленных пунктов кнопочной формы.

В левом нижнем углу экрана будет расположена свернутое до размеров пиктограммы окно самой базы данных. Восстановить размер окна щелчком по соответствующей кнопке. Выбрать вкладку «Таблицы». Выбрать первую таблицу из списка таблиц.

Выполнить щелчок по кнопке «Открыть». Информация будет представлена в режиме Таблицы. В верхней строке в качестве заголовков столбцов представлены имена полей. Каждая следующая строка представляет собой запись в таблице. Добавить в каждую из таблиц соответствующие данные (по 3-4 записи в таблицу) и закрыть окно.

Выполнить щелчок по кнопке «Конструктор». В окне будет представлена структура выбранной таблицы. В верхней части окна конструктора представлены имена полей и типы, в разделе «Свойства поля» – свойства выбранного поля. Слева от имени первого поля расположен символ «ключ», обозначающий, что данное поле является ключевым. Завершив знакомство со структурой таблицы, закрыть окно Конструктора.

Перейти на вкладку «Запросы», выполнить щелчок по кнопке «Создать», выбрать «Простой запрос» и создать его на основе таблицы «Сотрудники». Просмотреть созданный запрос в режиме конструктора, поработать со строкой «Условие отбора», создать и сохранить несколько запросов с условиями отбора в различных полях.

Закрыть БД, выполнив щелчок по соответствующей кнопке .

2. Определить предметную область и разработать диаграмму Йордана – Де Марко для своего варианта индивидуального задания. Варианты заданий приведены в прил. 1.

ЛАБОРАТОРНАЯ РАБОТА № 2

ПРЕДСТАВЛЕНИЕ ДАННЫХ С ПОМОЩЬЮ МОДЕЛИ «СУЩНОСТЬ—СВЯЗЬ». РАБОТА С ТАБЛИЦАМИ MICROSOFT ACCESS

Цель работы: изучение принципов построения модели «сущность—связь». Получение навыков работы по созданию структуры таблиц, модификации структуры таблиц, заполнению таблиц. Создание ключевых полей, индексированных полей, установка связей между таблицами. Удаление информации из связанных таблиц и восстановление этой информации.

Краткие теоретические сведения

Разработка модели «сущность—связь». Предметная область, которая была рассмотрена в предыдущей лабораторной работе, представляет собой схематическое изображение процессов, происходящих в реальном мире. Вторым способом представления данных, независимо от реализующего его программного обеспечения, является модель «сущность—связь».

Сущность (entity) – это объект, который может быть идентифицирован неким способом, отличающим его от других объектов. Примеры: конкретный человек, предприятие, событие и т.д.

Набор сущностей (entity set) – множество сущностей одного типа (обладающих одинаковыми свойствами). Примеры: все люди, предприятия, праздники и т.д. Наборы сущностей необязательно должны быть непересекающимися. Например, сущность, принадлежащая к набору «Мужчины», также принадлежит набору «Люди».

Связь (relationship) – это ассоциация, установленная между несколькими сущностями. Примеры:

- Поскольку каждый сотрудник работает в каком-либо отделе, между сущностями «Сотрудник» и «Отдел» существует связь "работает в …" или «Отдел—Работник».
- Так как один из работников отдела является его руководителем, то между сущностями «Сотрудник» и «Отдел» имеется связь "руководит …" или «Отдел—Руководитель».
- Могут существовать и связи между сущностями одного типа, например связь «Родитель – Потомок» между двумя сущностями «Человек».

Связь также может иметь атрибуты. Например, для связи «Отдел—Работник» можно задать атрибут «Стаж_работы_в_отделе».

Очень важным свойством модели «сущность—связь» является то, что она может быть представлена в виде графической схемы. Это значительно

облегчает анализ предметной области. Мы будем использовать нотацию Чена—Мартина. В табл. 2 приведен список обозначений.

Таблица 2

Условные обозначения нотации Чена—Мартина

Обозначение	Значение
Имя сущности	Набор независимых сущностей
Имя сущности	Набор зависимых сущностей
Имя атрибута	Атрибут
Имя атрибута	Ключевой атрибут
Имя связи	Набор связей

Атрибуты с сущностями и сущности со связями соединяются прямыми линиями. То число сущностей, которое может быть ассоциировано через набор связей с другой сущностью, называют степенью связи. Рассмотрение степеней особенно полезно для бинарных связей. Могут существовать следующие степени бинарных связей:

- один к одному (обозначается 1 : 1). Это означает, что в такой связи сущности с одной ролью всегда соответствует не более одной сущности с другой ролью. В рассмотренном нами примере это связь «руководит», поскольку в каждом отделе может быть только один начальник, а сотрудник может руководить только в одном отделе. Так как степень связи для каждой сущности равна 1, то они соединяются одной линией.

Важной характеристикой является класс принадлежности входящих в нее сущностей или кардинальность связи. Так как в каждом отделе обязательно должен быть руководитель, то каждой сущности «Отдел» непременно должна соответствовать сущность «Сотрудник». Однако не каждый сотрудник является руководителем отдела, следовательно, в данной связи не каждая сущность «Сотрудник» имеет ассоциированную с ней сущность «Отдел» (рис. 2).

Рис. 2. Пример связи «один к одному»

Таким образом, говорят, что сущность «Сотрудник» имеет обязательный класс принадлежности – ⟊, а сущность «Отдел» имеет необязательный класс принадлежности – ⟜.

- Много к одному (n : 1). Предположим, что рассматриваемое нами предприятие строит свою деятельность на основании контрактов, заключаемых с заказчиками. Этот факт отображается в модели «сущность – связь» с помощью связи «Контракт—Заказчик», объединяющей сущности «Контракт (номер, срок исполнения, сумма)» и «Заказчик (наименование, адрес)». Так как с одним заказчиком может быть заключено более одного контракта, то связь «Контракт—заказчик» между этими сущностями будет иметь степень n : 1 (рис. 3).

Рис. 3. Пример связи «многие к одному»

- многие ко многим (n : n). В этом случае каждая из ассоциированных сущностей может быть представлена любым количеством экземпляров. Пусть на рассматриваемом нами предприятии для выполнения каждого контракта создается рабочая группа, в которую входят сотрудники разных отделов. Поскольку каждый сотрудник может входить в несколько (в том числе и ни в одну) рабочих групп, а каждая группа должна включать не менее одного сотрудника, то связь между сущностями «Сотрудник» и «Рабочая_группа» имеет степень n : n (рис. 4).

Рис.4. Пример связи «многие ко многим»

В качестве примера построим диаграмму, отображающую связь на предприятии сотрудников, отделов и должностей. Каждый сотрудник может иметь более чем одну должность (и работать более чем в одном отделе), причем может занимать неполную ставку. В то же время одну и ту же должность могут занимать одновременно несколько сотрудников. В результате этих рассуждений мы должны ввести наборы сущностей «Сотрудник», «Отдел», «Должность» и набор связей «работает в». Для сущности «Сотрудник» введем атрибуты (свойства) «Табельный номер», «Имя сотрудника»; для сущности «Отдел» – «Имя отдела»; для сущности «Должность» – «Имя должности» и «Оклад». Также для связи «работает в» установим атрибут «ставка» (рис. 5).

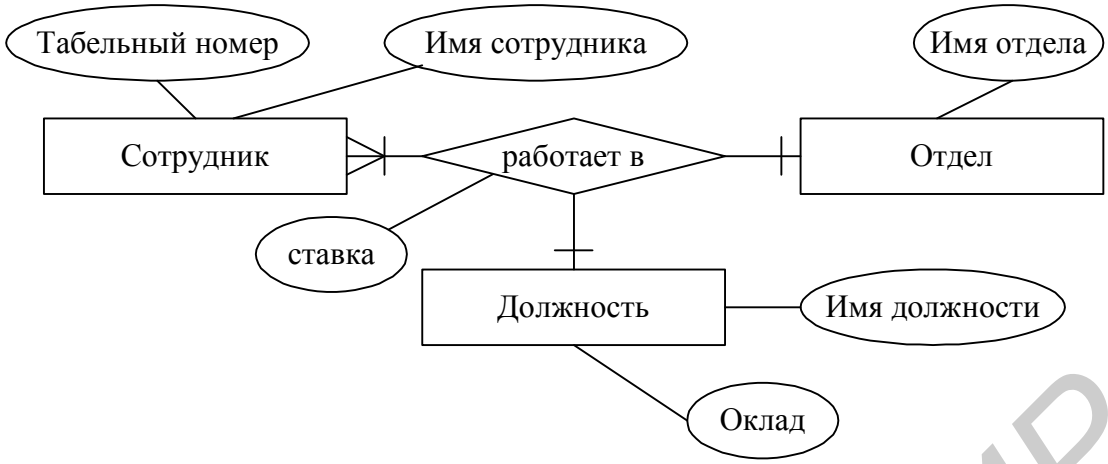

Рис. 5. Пример диаграммы «сущность—связь»

В приведенном примере связь «работает в» является тринарной (связывает три сущности). Для реализации базы данных на Access нам требуется, чтобы присутствовали только бинарные связи. Кроме того, атрибуты могут принадлежать только сущностям.

Введем дополнительную сущность «Штатная единица» с атрибутом «ставка». В результате получим модель (рис. 6).

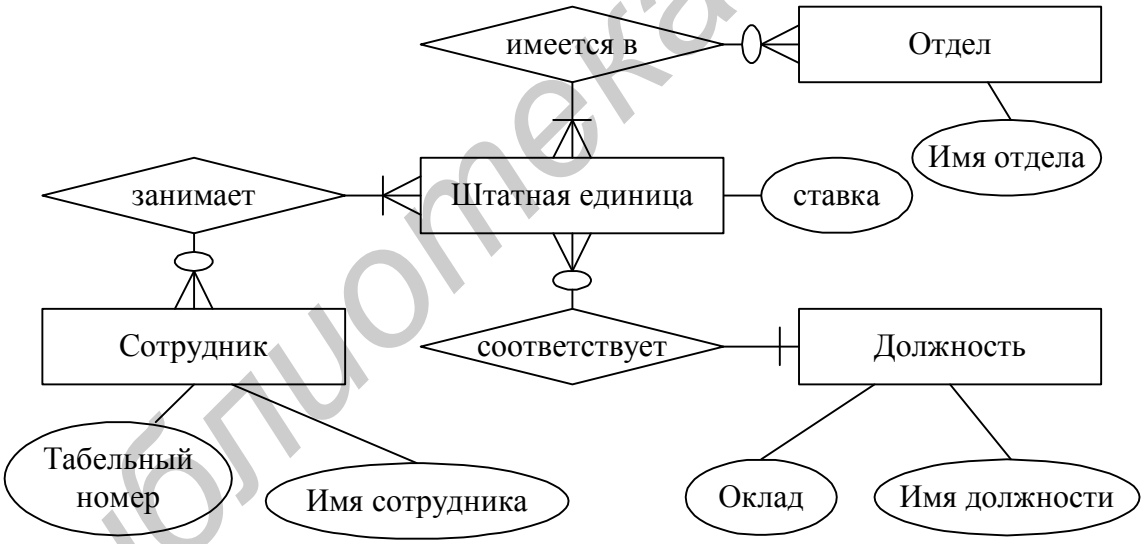

Рис. 6. Пример диаграммы «сущность—связь»

Создание базы данных в среде Microsoft Access. Итак, разработав модель «сущность—связь», переходим к созданию базы данных в Microsoft Access. В этом случае каждой сущности будет соответствовать таблица (табл. 3-6). Атрибуты сущностей будут определены соответствующими полями таблиц. Для организации связей между таблицами в Microsoft Access используются

ключевые поля и схема данных. В приведенных ниже примерах ключевым полем является поле «Код». Связь между таблицами показана на рис. 7.

Таблица 3

Таблица «Отделы»

Код	Имя отдела
1	Бухгалтерия
2	Отдел автоматизации

Таблица 4

Таблица «Должности»

Код	Имя должности	Оклад, р.
1	Руководитель	200 000
2	Инженер	100 000
3	Бухгалтер	150 000

Таблица 5

Таблица «Сотрудники»

Код	Табельный номер	Имя сотрудника
1	6630943	Иванов И. И.
2	7346290	Петров П.П.
3	2578244	Сидоров С.С.

Таблица 6

Таблица «Штатные единицы»

Код	Код сотрудника	Код отдела	Код должности	Ставка, р.
1	1 *(Иванов И.И)*	1 *(Бухгалтерия)*	1 *(Руководитель)*	150 000
2	2 *(Петров П.П.)*	1 *(Бухгалтерия)*	2 *(Бухгалтер)*	100 000
3	2 *(Петров П.П.)*	2 *(Отдел автоматизации)*	1 *(Руководитель)*	150 000
4	3 *(Сидоров С.С.)*	2 *(Отдел автоматизации)*	1 *(Инженер)*	120 000

Поля «Код сотрудника», «Код отдела», «Код должности» содержат числовые значения. Значения устанавливаются в соответствии с ключевым полем соответствующей таблицы. То есть в поле «Код сотрудника» записано значение «2», которое является ссылкой на запись в таблице «Сотрудники» с кодом «2» – Петрова П.П.

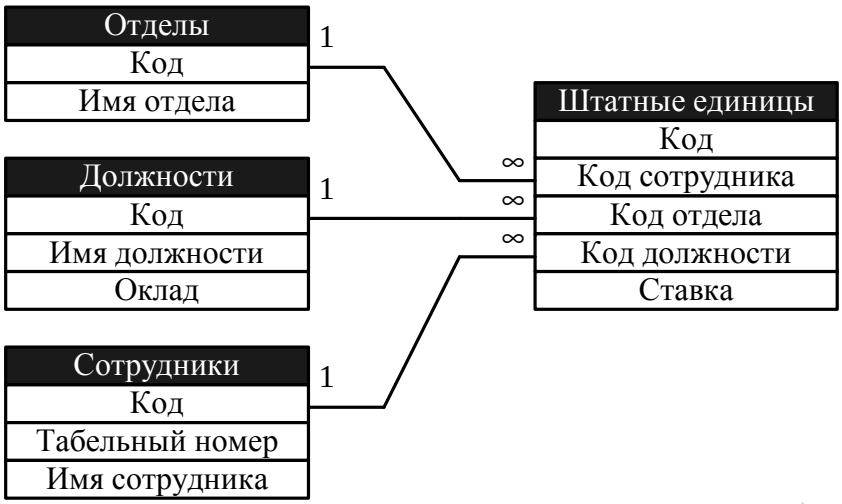

Рис. 7. Связь между таблицами в Microsoft Access

Рассмотрим основные этапы создания таблиц.

Определение структуры таблицы в режиме Конструктора. Для каждого поля конкретной таблицы необходимо определить его название, тип и размер и тщательно проверить, удовлетворяет ли диапазон значений выбранного типа тем значениям, которые может реально принимать данное поле. При необходимости для некоторых полей можно установить «Условие на значение» и задать сообщение, выдаваемое на экран в случае несоответствия введенного значения заданному условию или присвоить значения, принимаемые по умолчанию. Можно также определить формат вводимой информации для конкретных полей. В соответствии с этим заполняется соответствующей информацией каждый из разделов создаваемой таблицы: Имя поля, Тип данных и Описание. Раздел описаний необязателен для заполнения, но информация, введенная в данный раздел будет отображаться в строке состояния при вводе данных в соответствующие поля, т.е., по сути, может быть использована в качестве контекстной подсказки при вводе данных.

Поля уникальные и ключевые. Создание базы данных всегда начинается с разработки структуры ее таблиц. Структура должна быть такой, чтобы при работе с базой требовалось вводить в нее как можно меньше данных. Если ввод каких-то данных приходится повторять неоднократно, базу делают из нескольких связанных таблиц. Структуру каждой таблицы разрабатывают отдельно. Для того чтобы связи между таблицами работали надежно, и по записи из одной таблицы можно было однозначно найти запись в другой таблице, надо предусмотреть в таблице уникальные поля.

Уникальное поле – это поле, значения в котором не могут повторяться. Например, если из таблицы «Выдача книг» известно, что заказ сделал читатель Иванов. Но в таблице «Читатели» может быть несколько разных Ивановых и компьютер не разберется, кто же из них сделал данный заказ, это означает, что поле «Фамилия» не является уникальным и потому его нельзя использовать для

связи между таблицами. Поле номера телефона – более удачный кандидат на звание уникального поля, но одним телефоном могут пользоваться несколько разных людей. Если ни одно поле таблицы не приемлемо в качестве уникального, его можно создать искусственно.

В рассматриваемом примере в таблице «Читатели» создано поле «Код читателя», который присваивается с его вводом в базу данных. Его и использовали для связи между таблицами. Скорее всего, поле «Код читателя» окажется уникальным и проблем со связями между таблицами не возникнет, но было бы неплохо, если бы компьютер мог просигнализировать в том случае, если вдруг записи в этом поле повторятся. Для этого существует понятие «ключевое поле». При создании структуры таблиц одно поле (или одну комбинацию полей) можно назначить ключевым. С ключевыми полями компьютер работает особо. Он проверяет их уникальность и быстрее выполняет сортировку по таким полям. Ключевое поле – очевидный кандидат для создания связей. Иногда ключевое поле называют первичным ключом. Если при создании таблицы автор не задал ключевое поле, система управления базой данных вежливо напомнит о его необходимости. В качестве первичного ключа в таблицах часто используют поле, имеющее тип «Счетчик». Ввести два одинаковых значения в такое поле нельзя по определению, поскольку приращение значения поля производится автоматически. Структура связей между таблицами называется схемой данных.

Заполнение таблиц. Для этого необходимо открыть таблицу в режиме таблицы и заполнить поля необходимой информацией. Сохранение не требуется, так как оно производится сразу при переходе к следующей записи.

Установка связей между таблицами. Выполнить команду меню «Сервис → Схема данных».

Появится окно «Схема данных». Если связи устанавливаются впервые, оно будет содержать диалоговое окно «Добавление таблицы». Если окно «Добавление таблицы» отсутствует, его можно открыть, выбрав «Связи → Добавить таблицу» или выбрав пиктограмму «Добавить таблицу».

Выбрать таблицы, которые будут использоваться для установки связей, затем выполнить щелчок на кнопке для добавления таблиц в окно «Схема данных».

Для создания связей между таблицами переместить поле (или поля), которое необходимо связать, на соответствующее поле другой таблицы. В большинстве связей ключевое поле первой таблицы связывается с аналогичным полем второй таблицы. После перемещения поля появится диалоговое окно «Связи».

В диалоговом окне представлены названия таблиц, между которыми устанавливаются связи, и имена полей для связи. Полям, на основе которых создаются связи между таблицами, необязательно иметь одинаковые имена, однако они должны быть одного типа и размера. Исключение составляют поля счетчиков, которые можно связывать с числовыми полями.

Для автоматической поддержки целостности БД установить флажок «Обеспечение целостности данных». Кроме этого флажка в окне представлены и другие:

- «Каскадное обновление связанных полей». При включении данного режима изменения, сделанные в связанном поле первой таблицы, автоматически вносятся в поля связанной таблицы, содержащей те же данные.
- «Каскадное удаление связанных полей». При включении данного режима удаление записей в первой таблице приводит к удалению соответствующих записей связанной таблицы.

Затем следует выполнить щелчок на кнопке «Ok», закрыть окно «Связи». При запросе о сохранении связи выполнить щелчок на кнопке «Да».

Для выполнения лабораторных работ в качестве примера будет использоваться база данных «Библиотека», которая состоит из пяти таблиц, представленных в прил. 2.

На примере этой базы данных рассмотрим понятия нормализации и избыточности.

Теоретически для учета заказов книг в библиотеке можно было бы создать одну таблицу и вносить в нее информацию о читателе, о книге, об издательстве, дате заказа и т. д. Но очевидно, что такая таблица будет иметь существенные недостатки. В ней будет повторяться информация о читателе при заказе им очередной книги и данные о книге при заказе её новым читателем, также придется дублировать информацию об издательствах. Наличие повторяющихся полей или записей в базе данных называется избыточностью. Для того чтобы избавиться от избыточности, используют процесс нормализации, который на практике заключается в разбиении таблицы на несколько отдельных таблиц и установке связей между ними по соответствующим полям. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

В результате рассматриваемая база данных была разбита на пять таблиц, связанных между собой отношениями «один ко многим».

Задание к лабораторной работе

1. Открыть БД «Библиотека».

а) В режиме Конструктора открыть поочередно все таблицы и обосновать выбор ключевых полей для каждой из них, а также выбор типа данных в остальных полях;

б) изменить формат поля «Дата заказа» и установить соответствующий шаблон для ввода в это поле значений;

в) открыть «Схему данных» для этой базы данных и объяснить, как и почему были установлены приведенные связи между таблицами;

г) самостоятельно добавить несколько полей в таблицы БД (по собственному усмотрению) и внести в них информацию;

д) создать новую таблицу «Журналы», которая будет содержать поля: код журнала (счетчик, ключевое), название (текстовое), количество страниц (целое), год и месяц выпуска (дата/время);

е) добавить новую таблицу на схему данных и установить связь типа «один ко многим» с таблицей «Выдача книг», предварительно добавив в нее новое поле, по которому будет устанавливаться связь;

ж) дополнить таблицу «Выдача книг» заказами на журналы и книги и заказами только на журналы. Подумайте, какие из свойств полей дают возможность реализовать поставленную задачу.

2. Представить модель «сущность—связь» по индивидуальному заданию и предварительно определить структуру таблиц вашей базы данных. Согласовать с преподавателем.

3. Сформировать таблицы в режиме Конструктора, определить тип и основные свойства полей, установить связи между таблицами в соответствии с моделью «сущность—связь».

ЛАБОРАТОРНАЯ РАБОТА № 3

СОЗДАНИЕ УСЛОВНЫХ ЗАПРОСОВ И ЗАПРОСОВ НА ВЫБОРКУ

Цель работы: создание фильтров, условных запросов и запросов на выборку на основе учебной базы «Библиотека».

Краткие теоретические сведения

При работе с базами данных постоянно требуется организовывать просмотр, сортировку, фильтрацию, выборку данных и вычисление значений как в пределах одной таблицы, так и по всей базе данных. Например, нам нужно просмотреть все заказы, сделанные какой-либо фирмой, либо все заказы, сделанные за определенный месяц и т.п. В Microsoft Access поставленные задачи можно решать при помощи фильтров либо при помощи запросов.

Фильтры. Фильтры обычно используются при работе с одной таблицей. Они находят широкое применение при проектировании и эксплуатации баз данных в Microsoft Access. Преимуществом перед запросами является их простота. Кроме того, фильтры не занимают лишнего места на диске, в отличие от запросов. Для фильтрации данных в таблицах можно воспользоваться следующими способами:

*Поиск данных***.** Чтобы выполнить простой поиск по одному полю, сначала выделите его (поместив курсор на заголовок поля, при этом записи в поле изменят цвет на инверсный). Затем в меню при помощи команд «Правка → Найти» перейдите к диалоговому окну «Поиск в поле». В поле ввода «Образец» укажите значение, которое Access должен найти. В образе поиска можно использовать подстановочные символы. Символ ***** (звездочка) заменяет строку любой длины, а ? (знак вопроса) – любой произвольный символ. Например, «*****AB??DE*****» совпадает с «*ABERDEEN*» и «*TAB IDEA*», но не с «*LAB DEPARTMENT*».

Фильтр по выделенному. Предположим, вам нужно в базе данных «Библиотека» выбрать все записи таблицы «Издательства», в которых в качестве города указан город Москва. Найдите одну запись, в которой указан город Москва, щелкните правой кнопкой мыши по этому полю и выберите «Фильтр по выделенному» либо через меню «Запись → Фильтр → Фильтр по выделенному». В результате будут отображены все записи, у которых поле «Город» имеет значение «Москва». Опция «Исключить выделенное» – наоборот, оставляет на экране поля, значения которых не совпадают с выделенной записью.

Чтобы отфильтровать записи по значениям в различных полях, используют опцию (пиктограмму) «Изменить фильтр», которая позволяет сформировать строку с необходимыми значениями с помощью простейших алгебраических и

логических операторов. Просмотреть результат работы такого фильтра можно после нажатия на пиктограмму «Применить фильтр**».**

Запросы. При выполнении запроса происходит составление набора записей, содержащего отобранные данные. В большинстве случаев с набором записей можно работать как с обычной таблицей: вы можете просматривать и выбирать информацию, печатать и даже обновлять данные. Однако в отличие от реальной таблицы набор записей физически не существует в базе данных. Access создает его из данных таблиц только во время выполнения запроса.

Одно из преимуществ запросов состоит в том, что они позволяют достаточно быстро отобрать необходимые данные из нескольких связанных таблиц. Но запросы полезны и при работе с одной таблицей. Все приемы, используемые при работе с единственной таблицей, годятся и для сложных многотабличных запросов.

Существует немало различных видов запросов, но самые простые из них и к тому же используемые наиболее часто – это запросы на выборку. С них и принято начинать знакомство с созданием запросов. Цель запроса на выборку состоит в создании результирующей таблицы, в которой отображаются только нужные по условию запроса данные из базовых таблиц. Как и другие объекты Access, запросы можно создавать автоматически с помощью Мастера или вручную в режиме Конструктора.

Для создания запросов к базам данных существует специальный язык запросов. Он называется SQL (Structured Query Language – структурированный язык запросов). Но Access использует более простое средство, которое называется бланком запроса по образцу. С его помощью можно сформировать запрос простыми приемами, перетаскивая элементы запроса между окнами.

Рассмотрим выбор данных из одной таблицы на примере учебной базы данных «Библиотека». В данном примере организуем выборку читателей с фамилией «Бобров». Такой запрос будет называться условным.

Перейдите на вкладку «Запросы» и выберете «Создание запроса в режиме конструктора». Далее вам будет предложен список таблиц. Выберите таблицу «Читатели» и нажмите кнопку «Добавить». Так как мы ограничиваемся только одной таблицей, закройте это диалоговое окно. Приступим к заполнению конструктора запроса.

В строке запроса «Поле» в первом столбце выберите поле «Имя», во втором – «Отчество», в третьем – «Фамилия». Тот же результат будет получен, если последовательно выбирать эти поля в таблице двойным щелчком по нужному полю. В первом столбце, в строке «Условие отбора» введите фамилию: Бобров. Запрос должен иметь следующий вид (рис. 8).

Поле:	Имя	Отчество	Фамилия
Имя таблицы:	Читатели	Читатели	Читатели
Сортировка:			
Вывод на экран:	**ü**	**ü**	**ü**
Условие отбора:	Бобров		
или:			

Рис. 8. Параметры запроса на выборку

Закройте окно конструктора запроса. Перед закрытием Access запросит имя, под которым следует сохранить запрос. Введите «Бобров». Для того чтобы просмотреть результат работы запроса, наведите на него указатель и нажмите кнопку «Открыть». В результате появится окно (рис. 9).

	Имя	**Отчество**	**Фамилия**
4	Бобров	Виктор	Иванович
Σ			

Рис. 9. Результат выполнения запроса на выборку

Теперь рассмотрим пример выборки данных из нескольких таблиц. Просмотрим читателей, которые в 1996 г. заказали «Сборник задач» М.И Сканави. При этом пусть нам требуется вывести только название книги и фамилию читателя.

Для начала по вышеописанной процедуре откройте окно конструктора нового запроса и добавьте таблицы «Читатели», «Книги», «Выдача книг».

Переходим к заполнению параметров запроса. Так как нам следует организовать выборку по фамилиям читателей, названию книги и дате заказа, вводим соответствующие поля в бланк запроса. Далее вводим условие отбора для поля «Название книги» – «Сборник задач», а в условии отбора в поле «Дата заказа» – «Between 1.01.96 and 31.12.96». В данном случае мы используем оператор «Between … and», который организует выборку данных в указанном промежутке значений. Существует множество операторов для задания условия отбора, с которыми вы можете ознакомиться, используя справочную систему Access.

Строка «Вывод на экран» в бланке запроса предназначена для скрытия отображения поля в режиме просмотра запроса. Например, скроем поле «Дата заказа» в рассматриваемом примере, так как оно необходимо только для задания условия отбора. Бланк запроса должен иметь следующий вид (рис. 10).

Поле:	Имя	Название	Дата заказа
Имя таблицы:	Читатели	Книги	Выдача книг
Сортировка:			
Вывод на экран:	**ü**	**ü**	
Условие отбора:		Сборник задач	Between 1.01.96 and 31.12.96
или:			

Рис. 10. Многотабличный запрос на выборку

Сохраните запрос под именем «Сборник задач в 1996 году» и запустите его. В результате появится результат в следующем виде (рис. 11).

	Имя	**Название**
4	Федосенко	Сборник задач
	Захаров	Сборник задач
Σ		

Рис. 11. Результат многотабличного запроса на выборку

Вычисления в запросах. Поле, содержимое которого является результатом расчета по содержимому других полей, называется вычисляемым полем. Прежде чем мы научимся создавать и использовать вычисляемые поля, следует обратить внимание на то, что вычисляемое поле существует только в результирующей таблице запроса. В исходных (базовых) таблицах такое поле не создается, и при работе обычного запроса таблицы не изменяются.

Для создания запроса, производящего вычисления, служит тот же самый бланк запроса по образцу. Разница только в том, что в одном из столбцов вместо имени поля записывают формулу. В формулу входят заключенные в квадратные скобки названия полей, участвующих в расчете, а также знаки математических операций, например: сумма продажи: [Цена] * [Количество].

В узкий столбец непросто записать длинную формулу, но если нажать комбинацию клавиш SHIFT+F2, то открывается вспомогательное диалоговое окно, которое называется «Область ввода». В нем можно ввести сколь угодно длинную формулу, а потом щелчком по кнопке «ОК» перенести ее в бланк запроса по образцу.

Если включить отображение вычисляемого поля, результаты расчетов будут выдаваться в результирующей таблице.

Ничто не мешает сделать вычисляемое поле полем сортировки, чтобы не только получать новые результаты, но и анализировать их.

Для построения условий отбора могут применяться элементарные математические функции (=, <, >, <=, >=, <>), логические функции сцепки and

(и) or (или), функция Like «текст» для выбора по значению текстового поля, функция выбора по списку In (список значений через запятые).

Приведем примеры использования некоторых операторов:

- Москва or Минск – издательства Москвы или Минска.
- Not Москва – все кроме Москвы.
- Like «С*» – все текстовые записи, которые начинаются с буквы «С».
- <>1100 – все кроме указанного значения.
- >#01/03/98# – начиная с указанной даты.

Для выборки данных, связанных с определенной датой, можно воспользоваться встроенными функциями обработки даты Microsoft Access. Это функции Date, Day, Month, Year, DatePart. Рассмотрим каждую из них.

Date () – возвращает текущую дату. Обычно используют, если требуется связать запрос с текущей датой, месяцем, годом и т.п.

Day ([дата]), Month ([дата]), Year ([дата]) – для указанной даты возвращают целочисленное значение дня, месяца и года.

DatePart (интервал; [дата/время]) – возвращает для указанной даты или времени целочисленное значение, заданное параметром «интервал». Этот параметр – строковая переменная. Допустимые значения: уууу – год, Q – квартал, m – месяц, Y – день года, D – день месяца, w – день недели, ww – неделя, h – часы, n – минуты, s – секунды.

Пример: ***DatePart («Q», Date ())*** – при этом будет выведен номер квартала для текущей даты.

Используя вышеприведенные функции, предыдущий запрос можно привести к следующему виду (рис. 12).

Поле:	Имя	Название	Выражение1: Year (Дата заказа)
Имя таблицы:	Читатели	Книги	Выдача книг
Сортировка:			
Вывод на экран:	**ü**	**ü**	
Условие отбора:		Сборник задач	1996
или:			

Рис. 12. Создание выражений в полях запроса

Задание к лабораторной работе

1. Ознакомиться на практике с методами использования фильтров в СУБД Microsoft Access. Продемонстрировать выборку по трем начальным буквам фамилии, по двум полям одновременно.

2. При выполнении следующих заданий ознакомиться на практике с методикой создания условных запросов и запросов на выборку:

- Запрос должен выводить только Ф.И.О. читателей в алфавитном порядке и их домашние телефоны.

- Запрос должен осуществлять выбор книг издательства "Мир" в алфавитном порядке.
- Запрос должен осуществлять поиск книг по заданному сочетанию букв «упр» в теме.
- Запрос должен выводить в одном поле Ф.И.О. читателей, заказавших книги в 2002 г.

Чтобы вывести Ф.И.О. в одном поле, создадим выражение в поле запроса по формуле: [Фамилия]&" "&[Имя]&" "&[Отчество]. После нажатия на «Ввод», если нет ошибки, Access перед формулой добавит фразу «Выражение1:». Во второй строке этой колонки необходимо выбрать таблицу «Читатели» и установить отображение на экран.

3. Создать и сохранить запросы с использованием встроенных функций обработки даты в поле «Дата заказа», которые позволят произвести следующие выборки из нескольких таблиц:

- Информация о заказах на книги за последние 30 дней.
- Все книги, заказанные после 15 числа любого месяца.
- Читатели, сделавшие заказы на книги в 1998 году.
- Читатели, сделавшие заказы на книги в январе 1997 года.
- Читатели, сделавшие заказы на книги в текущем месяце текущего года.
- Читатели, сделавшие заказы на книги в первом квартале 1997 г.

3. Выполнить запрос на выборку в индивидуальном задании. Самостоятельно разработать задание и создать запрос с применением встроенных функций обработки данных в поле с типом дата/время для своей базы данных.

ЛАБОРАТОРНАЯ РАБОТА № 4

СОЗДАНИЕ ИТОГОВЫХ, ПАРАМЕТРИЧЕСКИХ И ПЕРЕКРЕСТНЫХ ЗАПРОСОВ. ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ЗАПРОСАМ

Цель работы: создание итоговых, параметрических и перекрестных запросов и отчетов по запросам на основе учебной базы «Библиотека».

Краткие теоретические сведения

Создание итоговых запросов. Иногда нас интересуют не отдельные записи таблицы, а итоговые значения по группам данных. Например, может понадобиться общая сумма книг для всех читателей за последний месяц или вы хотите узнать средний объем заказов по каждому месяцу прошлого года. Ответы на подобные вопросы дает итоговый запрос. Для задания вычислений итоговых значений нажмите кнопку «Σ» (групповые операции) на панели инструментов Конструктора запросов, чтобы в бланке запроса появилась строка «Групповая операция».

Теперь записи по каждому полю будут группироваться, но итоги подводиться не будут. Если выполнить запрос, то вы получите набор записей, содержащий по одной строке для каждого уникального значения поля запроса, но без итогов. Для вычисления итогов замените установку «Группировка» в строке «Групповая операция» на конкретные итоговые функции.

Access предоставляет девять функций, обеспечивающих выполнение групповых операций. Вы можете задать нужную вам функцию, выбрав ее в раскрывающемся списке. Ниже перечислены итоговые функции Access:

1. Sum – возвращает сумму всех значений данного поля в каждой группе. Используется только для числовых или денежных полей.

2. Avg – возвращает среднее арифметическое всех значений данного поля в каждой группе. Используется только для числовых или денежных полей. При вычислении функции Access исключает значения Null.

3. Min – возвращает наименьшее значение, найденное в данном поле внутри каждой группы, для числовых полей возвращает наименьшее значение, для текстовых полей – наименьшее из символьных значений независимо от регистра. Access игнорирует значения Null.

4. Max – возвращает наибольшее значение, найденное в данном поле внутри каждой группы, для числовых полей возвращает наибольшее значение, для текстовых полей – наибольшее из символьных значений независимо от регистра. Access игнорирует значения Null.

5. Count – возвращает число записей, в которых значения данного поля отличны от Null. Чтобы подсчитать число записей в каждой группе с учетом значений Null, введите в строку «Поле» специальное выражение COUNT (*).

6. StDev – возвращает стандартное отклонение всех значений данного поля в каждой группе. Эта функция применяется только к числовым или денежным полям. Если в группе меньше двух строк, Access возвращает значение Null.

7. Var – возвращает дисперсию значений данного поля в каждой группе. Эта функция применима только к числовым или денежным полям. Если в группе менее двух строк, Access возвращает значение Null.

8. First – возвращает первое значение данного поля в группе.

9. Last – возвращает последнее значение данного поля в группе.

Создание параметрических запросов. Запрос с параметрами – это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, например, условие для возвращения записей или значение, которое должно содержаться в поле. Можно создать запрос, в результате которого выводится приглашение на ввод нескольких данных, например двух дат. В результате будут возвращены все записи, находящиеся между указанными двумя датами.

Также запросы с параметрами удобно использовать в качестве основы для форм и отчетов. Например, на основе запроса с параметрами можно создать месячный отчет о заказах. При выводе данного отчета на экране появится приглашение ввести месяц, заказы за который интересуют пользователя. После ввода месяца на экране будет представлен требуемый отчет.

Для создания параметрического запроса необходимо сначала создать любой другой запрос или использовать уже имеющийся. А в нем для каждого поля, которое предполагается использовать как параметр, вводится в ячейку строки «Условие отбора – текст приглашения», заключенный в квадратные скобки. Это приглашение будет выводиться при запуске запроса. Текст подсказки должен отличаться от имени поля, но может включать его.

Для поля, в котором отображаются даты, можно вывести приглашения «Введите начальную дату:» и «Введите конечную дату:» для определения диапазона отбираемых значений. Для этого в ячейку строки «Условие отбора» вводится выражение Between [Введите начальную дату:] And [Введите конечную дату:].

Замечание. При создании параметрических запросов необходимо определить тип данных для параметров. Это делается с помощью вкладки «Запрос» в пункте «Параметры запроса», где в колонке «Параметр» полностью указывается содержимое квадратных скобок, а затем выбирается тип данных.

Создание перекрестных запросов. Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой – заголовки строк.

Перекрестный запрос создается с помощью Мастера или самостоятельно в режиме Конструктора запроса. В бланке запроса можно указать поля, значения

которых будут заголовками столбцов и строк, а также поле, значения которого следует использовать в вычислениях.

В перекрестном запросе, в котором заголовками столбцов являются, например, названия месяцев, можно вывести их в хронологическом, а не в алфавитном порядке или вывести столбцы только с января по июнь.

Для этого:

1) создайте перекрестный запрос с помощью или без помощи Мастера;
2) отобразите его в режиме Конструктора;
3) выберите фоновую область окна Конструктора запроса вне бланка запроса и списка полей;
4) откройте окно свойств запроса нажатием кнопки «Свойства» на панели инструментов;
5) в ячейку свойства «Заголовки столбцов» введите заголовки столбцов в том порядке, в котором их следует выводить в запросе. Заголовки столбцов следует разделять точкой с запятой или использовать символ разделителя списка данной страны. (Для того чтобы найти конкретный символ разделителя списка, откройте окно Язык и стандарты панели управления Windows).

Введенные заголовки столбцов должны точно соответствовать заголовкам столбцов в запросе в режиме таблицы.

Примечания:

1. Если перекрестный запрос выполняется часто или если он используется как базовый для формы или отчета, то описанные выше действия по заданию фиксированных заголовков столбцов позволяют уменьшить время выполнения запроса.

2. При частом использовании одних и тех же заголовков в разных запросах создайте таблицу с одним текстовым полем, в которой сохраняются заголовки столбцов. После этого открывайте таблицу и копируйте заголовки в ячейку свойства «Заголовки столбцов» (Column Headings).

Отчеты. Отчет является эффективным средством представления данных в печатном формате. Имея возможность управлять размером и внешним видом всех элементов отчета, пользователь может отобразить сведения желаемым образом. Большинство отчетов являются присоединенными к одной или нескольким таблицам и запросам из базы данных. Источником записей отчета являются поля в базовых таблицах и запросах. Отчет не должен включать все поля из каждой таблицы или запроса, на основе которых он создается.

Присоединенный отчет получает данные из базового источника записей. Другие данные, такие, как заголовок, дата и номера страниц, сохраняются в макете отчета.

Создание отчетов с помощью Мастера. Выбрав в диалоговом окне «База данных» вкладку «Отчеты» и щелкнув по кнопке «Создать», мы получаем диалоговое окно «Новый отчет», позволяющее создать отчет автоматически (автоотчет), с помощью Мастера или вручную. Операция создания отчета по запросам настолько проста, что сводится к одному щелчку левой кнопки мыши. При отсутствии принтера отчеты создавать все-таки можно. Достаточно

выполнить программную установку с помощью команды операционной системы: Пуск > Настройка > Принтеры > Установка принтера, после чего установить драйвер принтера, либо взяв его с гибкого диска, либо выбрав один из драйверов, прилагающихся к самой операционной системе.

Структура отчета. Отчеты состоят из разделов, а разделы могут содержать элементы управления. Но, в отличие от форм, разделов в отчетах больше, а элементов управления, наоборот, меньше. Со структурой отчета проще всего ознакомиться, создав какой-либо отчет с помощью Мастера, а затем открыв его в режиме Конструктора. Структура отчета состоит из пяти разделов: заголовка отчета, верхнего колонтитула, области данных, нижнего колонтитула и примечания отчета.

Раздел заголовка служит для печати общего заголовка отчета. Раздел верхнего колонтитула можно использовать для печати подзаголовков, если отчет имеет сложную структуру и занимает много страниц. Здесь можно также помещать и колонцифры (номера страниц), если это не сделано в нижнем колонтитуле. В области данных размещают элементы управления, связанные с содержимым полей таблиц базы. В эти элементы управления выдаются данные из таблиц для печати на принтере. Раздел нижнего колонтитула используют для тех же целей, что и раздел верхнего колонтитула. Раздел «Примечания» используют для размещения дополнительной информации.

Задание к лабораторной работе

1. Создать следующие запросы:

а) Итоговый запрос, который выводит список читателей и общее количество заказанных ими книг:

- закладка «Запросы», «Создание запроса в режиме Конструктора»;
- добавить таблицы «Читатели», «Выдача книг» и закрыть окно;
- в бланк запроса добавить из таблицы «Читатели» поле «Фамилия», а из таблицы «Выдача книг» – «Код книги»;
- выбрать из меню «Вид – Групповые операции». По умолчанию в новой строке бланка для всех полей устанавливается элемент «Группировка» и при выполнении запроса итоги не подводятся;
- для получения итогов для поля «Код книги» в строке «Групповая операция» выбрать функцию Count;
- в поле заголовка этого столбца перед полем «Код книги» ввести «Общее кол-во книг»;
- установить сортировку по убыванию и сохранить запрос.

б) Параметрический запрос, который выводит в алфавитном порядке список читателей, заказавших книги в определенный интервал времени, который будет устанавливаться пользователем:

- создать обычный запрос с помощью Конструктора;

- добавить таблицы «Читатели» и «Выдача книг», закрыть окно;
- в бланк запроса добавить из таблицы «Читатели» поля «Фамилия», «Имя», «Отчество», а из таблицы «Выдача книг»—«Дата заказа»;
- для поля «Фамилия» установить сортировку по возрастанию;
- в строке «Условия отбора» для поля «Дата заказа» ввести выражение: Between [Введите начальную дату:] And [Введите конечную дату:];
- далее необходимо указать тип каждого параметра: выбирать из меню «Запрос» пункт «Параметры» и в открывшемся окне в столбец «Параметры» ввести имя каждого параметра (в точном соответствии с бланком запроса), а во втором столбце выбрать нужный тип каждого параметра (Дата\Время) и щелкнуть по кнопке «ОК»;
- сохранить и выполнить запрос.

в) Параметрический запрос, который выводит книги по первым буквам фамилии автора:

- сначала создать обычный запрос;
- добавить таблицы «Книги » и «Издательства»;
- выбрать поля: «Автор», «Название», «Год издания книги», «Город» и «Название издательства»;
- для поля «Автор» установить сортировку по возрастанию;
- в строке условия отбора для поля «Фамилия» ввести выражение: Like [Введите первые буквы фамилии:] &«*»;
- далее указать тип параметра;
- Сохранить и выполнить запрос.

г) Перекрестный запрос, который содержит список издательств и для каждого издательства выводит количество книг, изданных за каждый год работы библиотеки:

- закладка «Запросы», «Создание запроса в режиме Конструктора»;
- добавить таблицы «Книги » и «Издательства»;
- выбрать поля «Год издания» и «Код книги», «Название издательства»;
- в меню «Запрос» выбрать пункт «Перекрестный» (в бланке появится две новые строки «Групповая операция» и «Перекрестная таблица»).
- в строке «Перекрестная таблица» для поля «Название издательства» установить «Заголовки строк», а для поля «Год издания» – «Заголовки столбцов». В строке «Групповая операция» по ним осуществляется «Группировка»;
- для поля «Код книги» в строке «Перекрестная таблица» установить «Значение», а в строке «Групповая операция» – Count;
- добавить итоговый столбец: еще раз добавить в бланк поля «Код книги». В строке «Перекрестная таблица» установить «Заголовок строк», а в строке «Групповая операция» выбрать Count. Переименовать столбец в «Итого»;
- сохранить и выполнить запрос.

2. Создать и сохранить следующие запросы в базе данных «Библиотека».

а) Итоговые запросы:

- список городов и наименований издательств, общая сумма стоимости книг по группам (вычисляемому полю присвоить имя «Общая стоимость книг»);
- список фамилий читателей и для каждого читателя – количество книг, заказанных им в 1999 г. (вычисляемому полю присвоить имя «Общее количество книг»);
- список наименований издательств, фамилии авторов и общее количество книг, изданных каждым издательством (вычисляемому полю присвоить имя «Общее количество книг»).

б) Параметрические запросы:

- условием выбора являются книги определенного года издания;
- поиск книг по ключевому слову в теме. Запрос создать на основании таблиц «Книги», «Издательства», «Темы»;
- список книг, заказанных в определенный месяц текущего года.

в) Перекрестные запросы:

- сводная таблица должна содержать фамилии читателей и для каждого читателя должно быть указано общее количество заказанных им книг, а также количество книг по каждому году;
- сводная таблица должна содержать информацию о количестве выдач каждой книги за определенный год, а также суммарное количество выдач каждой книги по месяцам данного года;
- сводная таблица должна содержать список всех издательств, общее количество изданных им книг, а также количество книг, изданных за каждый отдельный год.

3. Выполнить пункты по итоговым, параметрическим и перекрестным запросам в индивидуальном задании.

4. По каждому запросу создать отчет с помощью Мастера, в случае необходимости провести его более точную настройку в режиме Конструктора.

ЛАБОРАТОРНАЯ РАБОТА № 5

СОЗДАНИЕ ФОРМ. ПРИВЯЗКА ИНФОРМАЦИОННЫХ ПОЛЕЙ ЧЕРЕЗ ВЗАИМОСВЯЗИ

Цель работы: создание форм и элементов управления с помощью Мастера и Конструктора. Привязка информационных полей через взаимосвязи.

Краткие теоретические сведения

Обычно разработчик базы данных создает структуру таблиц и запросов, но заполнением таблиц информацией он не занимается. Для этого есть специальные кадры, выполняющие функции наборщиков. Для упрощения их труда разработчик базы может подготовить специальные объекты – формы. Форма представляет собой некий электронный бланк, в котором имеются поля для ввода данных. Наборщик вводит данные в эти поля, и данные автоматически заносятся в таблицы базы.

Зачем нужны формы? Данные в таблицу можно вносить и без помощи каких-либо форм, но существуют по крайней мере четыре причины, которые делают формы незаменимым средством ввода данных в базу. Во-первых, малоквалифицированному персоналу нельзя предоставлять доступ к таблицам (самому ценному из того, что есть в базе). Представьте, что будет, если новичок «наведет порядок» в таблице банка, хранящей расчетные счета клиентов. Во-вторых, разные люди могут иметь разные права доступа к информации, содержащейся в таблицах. Например, один имеет право вводить только имена и адреса клиентов, другой – только номера их расчетных счетов, а третий – только денежные суммы, хранящиеся на этих счетах. Сговор между этими людьми должен быть исключен. Для ввода данных им предоставляют разные формы, хотя данные из форм могут поступать в одну таблицу. В-третьих, ввод данных в таблицу – чрезвычайно утомительное занятие. Уже после нескольких часов работы люди делают ошибки. Ввод данных в форму проще. Здесь многое можно автоматизировать. К тому же элементы управления форм настраивают таким образом, чтобы при вводе данных выполнялась их первичная проверка. И, наконец, в-четвертых, надо вспомнить, откуда берется информация для баз данных. Как правило, ее берут из бумажных бланков (анкеты, заявления, накладные, счета, описи, ведомости, справки и т. п.). Экранные формы можно сделать точной копией бумажных бланков, с которых происходит ввод данных. Благодаря этому во много раз уменьшается количество ошибок при вводе и значительно снижается утомляемость персонала.

Создание форм. Как и другие объекты Access, формы можно создавать вручную или автоматически, причем несколькими способами. В отличие от таблиц и запросов формы состоят из многочисленных элементов управления, и от того, насколько аккуратно эти элементы расположены на экране, зависит

внешний вид формы. Автоматические средства позволяют создавать аккуратные формы и не задают пользователю лишних вопросов. Начинать работу лучше с них.

Создание формы с помощью Мастера. С помощью Мастера форма создается всего в четыре этапа:

1) выбор полей, данные для которых можно будет вводить в форме;
2) выбор внешнего вида формы (один из четырех);
3) выбор фонового рисунка формы (один из десяти);
4) задание имени формы.

Все эти пункты достаточно хорошо объяснены в Мастере и не требуют никаких пояснений. Готовую форму можно сразу же использовать для просмотра существующих записей или для ввода новых. Структуру формы составляют ее разделы, а разделы содержат элементы управления. Создадим форму с помощью Мастера, взяв за основу все поля из таблицы «Выдача книг».

Разделы формы. Самый простой способ познакомиться с разделами формы состоит в том, чтобы взять готовую форму, например созданную с помощью Мастера, и посмотреть ее структуру в режиме Конструктора. Обратите внимание на то, что рядом с ней открывается панель элементов, содержащая заготовки и инструменты для создания элементов управления формы. Размеры разделов и размеры рабочего поля формы можно изменять с помощью мыши. В структуре формы четко видны три раздела: раздел заголовка формы, область данных и раздел примечания формы.

Все, что содержится в области данных, является элементами управления. В рассматриваемом случае присутствуют элементы управления только двух типов: связанное поле (то, что в него вводится, поступает и в одноименное поле таблицы) и присоединенная надпись (называется так, поскольку перемещается вместе со своим элементом управления). Содержание присоединенной надписи совпадает с названием связанного поля, но это можно и изменить.

Создание надписей. Редактирование форм состоит в создании новых или изменении имеющихся элементов управления, а также в изменении их взаимного расположения.

При рассмотрении приемов создания новых элементов управления воспользуемся тем фактом, что Мастер, создавший форму, не заполнил ее раздел заголовка. Перетащив вниз разделительную границу между заголовком и областью данных, мы можем освободить вверху достаточно места для создания крупной надписи. На панели элементов существует специальный элемент управления для создания заголовков, который называется «Надпись». Щелкнув по нему, а потом по форме, мы получаем текстовую рамку, в которую можно вводить произвольный текст. При вводе текста не надо заботиться о его форматировании. Неважно, как он выглядит и где расположен. Закончив ввод, надо нажать клавишу «ENTER», после чего можно приступать к оформлению текста. Для форматирования элемента управления его надо сначала выделить. Для этого служит инструмент «Выбор объектов». При выделении элемента управления вокруг него образуется рамка с восемью маркерами (по углам и по

центрам сторон рамки). Рамку можно растягивать или сжимать методом перетаскивания границ. При наведении на маркер указатель мыши меняет форму, принимая вид открытой ладони. В этот момент рамку можно перемещать. Особую роль играет левый верхний маркер рамки. При наведении на него указатель мыши принимает форму указательного пальца. О роли этого маркера мы расскажем чуть позже. Когда объект выделен, можно изменять параметры шрифта, метод выравнивания текста и другие элементы форматирования. Это выполняют обычными средствами форматирования, доступными через соответствующую панель инструментов Access. Если щелкнуть на выделенном элементе правой кнопкой мыши, откроется его контекстное меню, в котором имеются дополнительные возможности изменения оформления.

Создание и редактирование связанных полей. Заголовок таблицы, который мы только что создали, не связан ни с одним из полей таблицы. Поэтому элемент управления «Надпись» еще называют свободным полем. Текст, введенный в него, остается неизменным независимо от того, какую запись в этот момент просматривают в форме. Совсем иначе обстоит дело с элементами управления, в которых отображается содержимое полей таблицы. Такие элементы управления называют связанными полями. Для их создания служит элемент «Поле» на панели инструментов. При создании связанного поля вместе с ним одновременно образуется еще один элемент управления – «Присоединенная надпись». Она перемещается вместе со связанным полем и образует с ним единое целое. Оторвать поле от присоединенной надписи позволяет уже упомянутый маркер, расположенный в левом верхнем углу. При наведении на него указатель мыши принимает форму указательного пальца. В этот момент связанное поле можно оторвать от присоединенной надписи и перемещать отдельно. Перемещать элементы управления и изменять их размеры с помощью мыши не слишком удобно. Гораздо удобнее использовать для этой цели курсорные клавиши в комбинации с клавишами «SHIFT» или «CTRL». В первом случае происходит изменение размеров элемента управления, а во втором – изменение его расположения. Чтобы элементы управления располагались в форме ровными рядами, существуют специальные команды выравнивания. Сначала надо выделить группу элементов управления с помощью инструмента «Выбор объектов» (группа выбирается при нажатой клавише SHIFT), а потом дать команду «Формат – Выровнять и выбрать метод выравнивания».

Прочие элементы управления формы. При создании формы вручную элементы управления размещают на ней так, как удобно проектировщику. Созданные элементы управления формы выравнивают с помощью команды «Формат – Выровнять». Кроме рассмотренных выше элементов управления «Надпись» и «Поле», существует еще несколько полезных элементов управления:

1) переключатели (с ними можно связать команды, например, выполняющие фильтрацию);

2) флажки (действуют аналогично переключателям, но, в отличие от них, допускают множественный выбор, удобны для управления режимами сортировки данных);

3) список (может содержать фиксированный набор значений или значения из заданного поля одной из таблиц, позволяет не вводить данные, а выбирать их из списка);

4) поле со списком (применяется так же, как и список, но занимает меньше места в форме, поскольку список открывается только после щелчка на раскрывающей кнопке);

5) командные кнопки (с каждой из них можно связать какую-либо полезную команду, например, команду поиска записи, перехода между записями и др.);

6) вкладки (позволяют разместить много информации на ограниченной площади, на них размещают другие элементы управления);

7) поле объекта OLE (служит для размещения внешнего объекта, соответствующего принятой в Windows концепции связывания и внедрения объектов). Объектом, как правило, является иллюстрация, например фотография, но это может быть и видеозапись, и музыкальный фрагмент, и голосовое сообщение).

Существуют два типа полей для размещения объектов OLE: свободная рамка объекта и присоединенная рамка объекта. В первом случае рамка не связана ни с каким полем таблиц базы данных. Объект, находящийся в ней, выполняет роль иллюстрации и служит для оформления формы. С присоединенной рамкой связано одно из полей таблицы. В ней отображается содержимое этого поля. Это содержимое может меняться при переходе от одной записи к другой.

Привязка информационных полей через взаимосвязи. Во второй лабораторной работе мы рассмотрели простейший пример базы данных предприятия, в которую входили таблицы «Сотрудники», «Отделы», «Должности», «Штатные единицы». При этом таблица «Штатные единицы» формировалась на основе ключевых полей остальных таблиц. Таблица имела следующий вид (табл. 7).

Таблица 7

Таблица «Штатные единицы»

Код	Код сотрудника	Код отдела	Код должности	Ставка
1	1	1	1	150
2	2	1	2	100
3	2	2	1	150
4	3	2	1	120

Очевидно, что такой вид таблицы неудобен для заполнения. Если у вас будет несколько десятков работников, отделов и должностей, их будет невозможно удержать в памяти. Но и переключаться между таблицами, чтобы

посмотреть, какой записи какой номер соответствует, тоже неудобно. Лучше всего было бы заполнять таблицу, имеющую следующий вид (табл. 8).

Таблица 8

Таблица «Штатные единицы», форма для заполнения

Код	Код сотрудника	Код отдела	Код должности	Ставка
1	Иванов И.И	Бухгалтерия	Руководитель	150 000
2	Петров П.П.	Бухгалтерия	Бухгалтер	100 000
3	Петров П.П.	Отдел автоматизации	Руководитель	150 000
4	Сидоров С.С.	Отдел автоматизации	Инженер	120 000

Для того чтобы исходная табл. 7 приняла такой вид, используют привязку информационных полей через взаимосвязи. Рассмотрим привязку на примере таблиц «Сотрудники» и «Штатные единицы». Нужно, чтобы числовые значения поля «Код сотрудника» таблицы «Штатные единицы» заменялись на соответствующие значения из поля «Имя сотрудника» таблицы «Сотрудники». Поле «Код» таблицы «Сотрудники» имеет тип «Счетчик», а поле «Код сотрудника» таблицы «Штатные единицы» – числовой тип.

Необходимо выполнить такую последовательность действий: открыть таблицу «Штатные единицы» в режиме Конструктора, выбрать поле «Код сотрудника» и перейти на вкладку «Подстановка». Указать тип элемента управления «Поле со списком». В качестве «Источника строк» выбрать таблицу «Сотрудники». Таблица «Сотрудники» содержит поля «Код», «Табельный номер» и «Имя сотрудника». Поскольку нам достаточно видеть поле «Имя сотрудника», в строке «Ширина столбцов» введите 0; в этом случае обнуление означает, что первые два столбца выводиться не будут. В целом параметры подстановки должны иметь следующий вид (рис. 13).

Тип элемента управления	Поле со списком
Тип источника строк	Таблица/запрос
Источник строк	Сотрудники
Присоединенный столбец	1
Число столбцов	3
Заглавия столбцов	Нет
Ширина столбцов	0см;0см
Число строк списка	8
Ширина списка	Авто
Ограничиться списком	Да

Рис. 13. Параметры подстановки для поля «Код сотрудника»

Открытие таблицы в режиме просмотра позволяет получить следующий результат (табл. 9).

Таблица 9

Таблица «Штатные единицы» в режиме просмотра

Код	Код сотрудника	Код отдела	Код должности	Ставка
1	Иванов И.И	1	1	150 000
2	Петров П.П.	1	2	100 000
3	Петров П.П.	2	1	150 000
4	Сидоров С.С.	2	1	120 000

Настройка взаимосвязи оставшихся полей устанавливается аналогичным способом. Тот же результат можно получить, если настроить на соответствующей форме элемент управления «Поле со списком».

Задание к лабораторной работе

1. Привязать через поля-коды реальные информационные данные о читателях (Фамилия, Имя, Отчество и Домашний телефон), о книгах (Название, Автор) в таблице «Выдача книг».
2. Создать автоформы по всем таблицам БД «Библиотека».
3. Создать формы по всем ранее созданным запросам, с подключением соответствующих отчетов (с кнопками перехода на просмотр и печать отчета).
4. Создать формы на основе необходимых полей таблиц БД «Библиотека», которые позволят:

а) осуществлять в наиболее удобной форме оформление нового заказа на книги;

б) осуществлять поиск и вывод в подчиненной форме информации о книге по выбранной теме.

5. Создать главную форму, с которой будет осуществляться переход на все другие объекты БД. Предусмотреть возврат из второстепенных форм на главную. На главной форме должны размещаться:

а) кнопка перехода на форму для работы с автоформами таблиц (причем переход между формами таблиц должен быть осуществлен с учетом связей между таблицами базы данных);

б) кнопки открытия форм из пункта 4;

в) кнопка перехода на форму просмотра отчетов по запросам;

г) кнопка выхода из приложения;

д) кнопка просмотра информации о БД и ее разработчике.

ЛАБОРАТОРНАЯ РАБОТА № 6

НАПИСАНИЕ ПРОЦЕДУР ОБРАБОТКИ СОБЫТИЙ НА VISUAL BASIC FOR APPLICATION

Цель работы: изучить основные принципы написания программ на VBA, создать форму для удаления и добавления полей в таблицы БД «Библиотека» на основе применения полученных знаний.

Краткие теоретические сведения

Visual Basic for Applications представляет собой полуфункциональный язык программирования, являющийся неотъемлемой составной частью Access. Этот язык используется для разработки приложений, предназначенных для манипулирования БД и для настройки пользовательского интерфейса. VBA – это структурированный язык программирования высокого уровня. В нем, как и в других языках, есть операторы проверки условий, циклического выполнения повторяющихся операций, а также обмена данными с памятью и дисками. В языке VBA реализованы общие принципы объектно-ориентированного программирования. Это означает, что пользовательская среда, управляемая приложением, не подвергается изменениям путем выполнения последовательности процедур и операторов, но реагирует на события, связанные с различными объектами: полями ввода, кнопками, разделами форм и отчетов. В языке VBA программный код привязан непосредственно к объектам и срабатывает тогда, когда случается определенное событие. Все программирование в Windows основано именно на отклике на то или иное событие в системе.

Создание кода VBA с помощью Мастеров элементов управления. Одним из эффективных способов использования VBA является привязка кода к кнопкам, добавляемым в форму с помощью Мастеров элементов управления. Чтобы добавить кнопку с фрагментом кода, а затем просмотреть код, необходимо выполнить следующее:

1. Открыть БД «Библиотека».
2. Выбрать таблицу и выполнить щелчок на кнопке «Автоформа» стандартной панели инструментов. Для таблицы будет создана форма по умолчанию.
3. Сохранить созданную форму, выполнив щелчок на кнопке закрытия окна, а затем на запрос о сохранении ответить утвердительно и ввести имя созданной формы или оставить предложенное для сохранения имя формы.
4. Выбрать вкладку «Формы». Выбрать созданную форму и открыть ее в режиме Конструктора.
5. Выбрать в меню «Вид» команду «Панель элементов», если панель элементов не отображена на экране.

6. Выполнить щелчок на элементе «Кнопка».

7. Выполнить щелчок на свободном месте формы, чтобы вставить в нее кнопку. При этом откроется диалоговое окно Мастера кнопок, показанное на рис. 14.

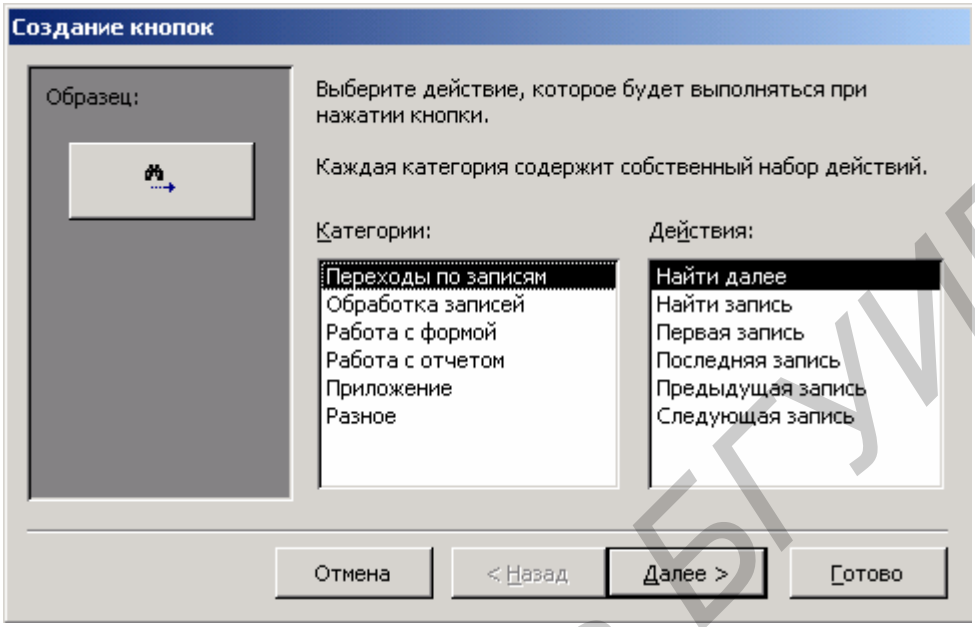

Рис. 14. Окно мастера создания кнопок

8. В списке «Категории» выбрать пункт «Переходы по записям», а затем в списке «Действия →Найти далее» После щелчка по кнопке «Готово» в форму будет добавлена кнопка, выполняющая поиск.

9. Выполнить щелчок по элементу «Кнопка».

10. Вставить кнопку на свободном месте формы ниже предыдущей, созданной ранее. Снова откроется окно кнопок.

11. В списке «Категории» выбрать пункт «Работа с формой», а в списке «Действия» – пункт «Закрытие формы». После щелчка по кнопке «Готово» в форму будет добавлена кнопка, выполняющая закрытие формы.

12. Сохранить форму, выбрав команду «Файл → Сохранить». После этого можно запустить форму и проверить работу добавленных кнопок.

13. Переключиться в режим Конструктора, выполнив щелчок по кнопке «Конструктор». Выбрать команду «Вид → Программа» или выполнить щелчок по кнопке «Программа» на панели инструментов. Откроется окно редактора Visual Basic for Applications, представленное на рис. 15.

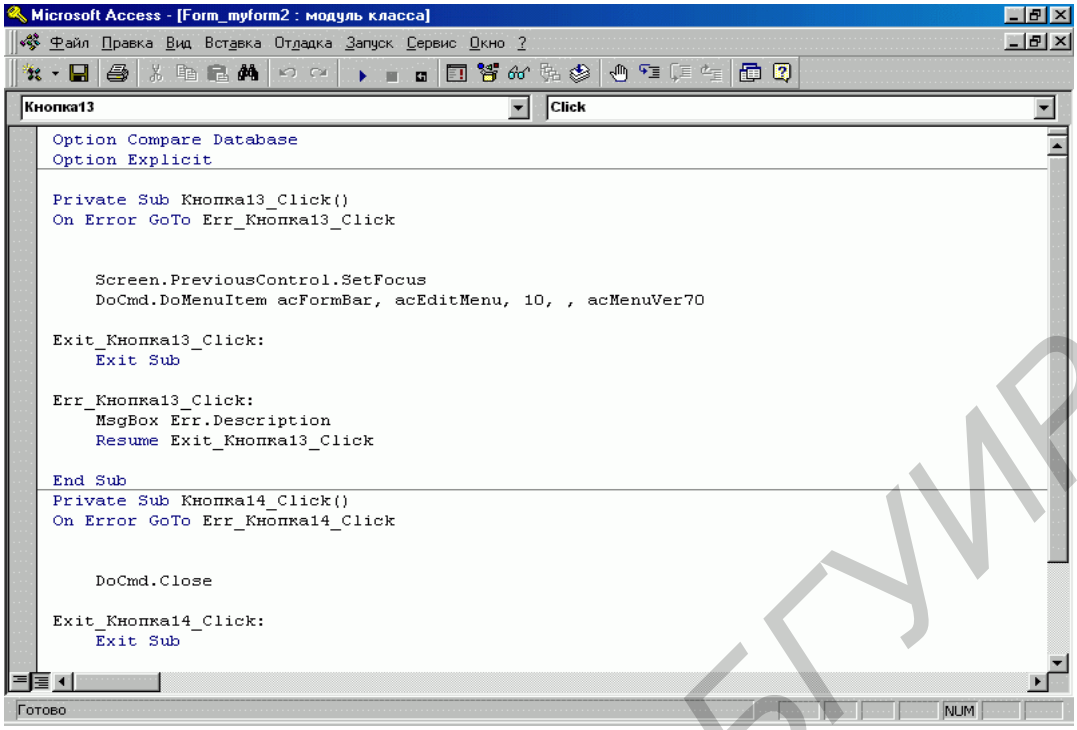

Рис. 15. Окно редактирования программных модулей

В этом окне можно видеть три фрагмента программы на языке VBA: раздел объявлений (в верхней части окна), который содержит код, относящийся ко всей форме в целом, а также два раздела кода, привязанных к двум кнопкам.

Первый раздел начинается с заголовка процедуры:

```
Private Sub Кнопка13_Click ()
```

Заканчивается процедура оператором: End Sub. Операторы, заключенные между ними, выполняются после щелчка по кнопке «Найти». Основная часть работы выполняется следующим оператором:

```
DoCmd.DoMenuItem acFormBar, AcEditMenu, 10, AcMenuVer70
```

Этот оператор выполняет те же действия, что и команда меню «Правка →Найти», т.е. выводит на экран диалоговое окно «Поиск в поле», с помощью которого можно выполнить поиск.

Открыв модуль VBA, его можно редактировать как текст в любом текстовом редакторе. В окне редактора Visual Basic в код можно вставить текст из другого файла. Для этого выбрать команду «Вставка → Файл», а в открывшемся диалоговом окне выбрать нужный файл и выполнить щелчок по кнопке «ОК».

Модули на языке Visual Basic можно выводить на печать. Для этого следует открыть модуль и выбрать команду «Файл → Печать».

Модуль является структурным элементом программы, написанной на языке VBA. Это совокупность объявлений и процедур, объединенных в единое целое. Существуют модули трех типов: стандартные модули, модули формы и модули отчетов. В отличие от стандартного модуля, который создается таким же образом, как и любой другой объект БД, и может выполнять практически любые вычисления, модули форм и отчетов разрабатываются для обработки событий, связанных с элементами формы или отчета.

Каждый модуль состоит из области описания и одной или нескольких процедур. Процедура представляет собой последовательность операторов, которые часто называют программными кодами. Входящие в модуль процедуры объединены общей областью описания. В ней описываются данные и объекты, которые являются общепринятыми для процедур модуля. Иерархия указанных объектов такова:

- база данных;
- модуль;
- область описания;
- процедура;
- код;
- оператор.

Процедуры делятся на две категории: процедуры-подпрограммы (подпрограммы) и процедуры-функции (функции).

Процедура–подпрограмма активизируется при обращении к ней по имени, вследствие чего выполняется определенная последовательность операторов (инструкций). Подпрограмму используют, например, для задания свойства формы или заполнения списка значениями, полученными в результате вычислений.

Процедура-функция после выполнения возвращает некоторое значение, которое можно применять в операторах и выражениях в качестве переменной. Например, функции могут возвращать значение, используемое по умолчанию для некоторого поля, или вычислять сложный критерий в рамках запроса. В модуль можно включать любое количество функций и подпрограмм.

Процедурой называется целостная структурная единица кода на языке VBA. Каждая процедура состоит из операторов, в которых применяются встроенные в Access функции, методы и свойства, предназначенные для выполнения каких-либо операций над данными. Например, к свойству Click кнопки «Найти» приведенного ранее примера привязана следующая процедура:

```
Private Sub Command6_Click ( )
On  Error GoTo Err_Command6_Click
        Screen.PreviosControl.SetFocus
DoCmd.DoMenuItem acFormBar, acEditMenu, 10, ,  acMenuVer70
Exit Command6_Click:
        Exit Sub
Err_Command6_Click:
```

```
        MsgBox Err.Description
        Resume Exit_Command6_Click
End Sub
```

Для обращения к перечисленным объектам, а также для обозначения операторов используются ключевые слова, которые записываются с прописной буквы (например Function).

Теоретически каждая процедура может быть вызвана из любого модуля, а функция – из таких объектов, как форма, запрос или отчет. Наряду с общедоступными процедурами (Public), которыми являются все процедуры по умолчанию, существуют локальные, или личные, процедуры (Private), доступные только в том модуле, в котором они описаны.

Для передачи значений из вызывающих операторов в вызываемые процедуры служат аргументы. С помощью аргументов ведется контроль за выполнением процедуры, устанавливается способ получения результата, определяются параметры вычислений и т.д.

Создание процедур обработки событий. Начинающим разработчикам приложений Access понадобится набор процедур для отклика на различные события, например, внесение изменений в поле или щелчок на кнопке. Код процедуры обработки события привязывается к событию, которое может произойти при работе с формой, отчетом или элементом управления. Результат выполнения процедуры-функции обычно применяется:

- в качестве значения по умолчанию для поля таблицы;
- в качестве значения критерия для запросов или фильтров;
- в качестве содержимого поля.

Хотя конкретный вид процедуры зависит от события и желаемого отклика, можно предложить общую последовательность действий по созданию процедуры обработки события:

1. Открыть нужную форму или отчет в режиме Конструктора. Если необходимо привязать процедуру к элементу управления, выбрать этот элемент.

2. Выбрать команду «Вид →Свойства», чтобы открыть окно свойств формы, отчета или элемента управления.

3. Открыть вкладку «События».

4. Выполнить щелчок на строке свойства того события, которое должно запускать процедуру. Например, если процедура должна запускаться в ответ на изменение пользователем данных, выполнить щелчок на свойстве «После обновления».

5. Выполнить щелчок по кнопке с тремя точками, чтобы открыть диалоговое окно «Построитель».

6. В этом окне выполнить двойной щелчок мышью на строке «Программы». Откроется окно редактора Visual Basic, в котором автоматически появится начальная и конечная строки процедуры.

7. Ввести операторы, образующие код процедуры.

Для повышения удобочитаемости кода процедуры, вводимого между ограничительными операторами, применяют приемы структурирования. Один из таких приемов заключается в том, что все операторы процедуры записываются с отступом от начала строки. Для создания отступа можно использовать клавишу «Tab». По умолчанию позиции табуляции установлены через четыре символа. Отступы образуются и посредством команды «Увеличить отступ» из меню «Правка». В результате активизации этой команды строка смещается вправо на то количество символов, которое указано в поле интервал табуляции. Команда «Уменьшить отступ» служит для выполнения противоположного действия.

Посредством отступов выделяют вложенные циклы и условные операторы внутри процедуры.

Тексты программ принято снабжать комментариями. В начале каждой строки комментария ставится апостроф, и такие строки не влияют на выполнение программы, а при синтаксическом анализе и компилировании – пропускаются. Наряду с отступами Access позволяет использовать различные шрифты и цвета для выделения фрагментов текста модуля. Эти параметры устанавливаются на вкладке «Модуль» диалогового окна «Параметры».

Редактирование в окне модуля. Код модуля и тексты подпрограмм/функций редактируются так же, как документы в обычном текстовом редакторе. Для поиска процедуры в тексте модуля используется список процедур, находящийся в верхней части окна модуля.

При перемещении курсора из строки кода программа автоматически проверяет синтаксис этой строки и в случае обнаружения ошибки выводит на экран соответствующее сообщение. Для отказа от такой проверки следует выключить опцию проверка синтаксиса на вкладке «Модуль» диалогового окна «Параметры».

Для быстрого поиска и замены фрагментов кода применяются команды «Найти» и «Заменить» из меню «Правка».

В Access при редактировании фрагментов кодов кроме общепринятых используются дополнительные комбинации клавиш: Ctrl+Y позволяет вставить в буфер строку, в которой находится курсор, не выполняя его маркировку; F3 и Shift + F3 позволяют просмотреть все фрагменты модуля, в которых встречается искомая последовательность символов. F3 дублирует команду «Найти далее» из меню «Правка» и кнопку «Найти далее» окна поиска.

Аргументы процедуры. Благодаря аргументам пользователь имеет возможность управлять выполнением процедуры. При описании процедуры, зависящей от аргументов, имя аргумента принято вводить в скобках за именем процедуры в строке с ключевым словом Function/Sub. Например ввод коэффициента при пересчете рублей в доллары:

```
Function Рубли_в_Доллары (Коэффициент)
```

При вызове такой функции значение аргумента указывается в скобках после имени функции. Например:

```
=Рубли_в_Доллары (4500)
```

Вызов процедуры, имеющей аргументы, должен сопровождаться заданием значений для всех объявленных аргументов. Эти значения перечисляются в том же порядке, что и аргументы при объявлении, и разделяются запятыми. В пользовательских функциях или модулях в качестве аргументов могут применяться выражения.

Основные операторы и переменные VBA. Переменные используются для сохранения значений величин, изменяющихся в процессе выполнения программы. Каждая переменная имеет имя, по которому к ней обращаются. Правила присвоения имен аналогично правилу присвоения имен для полей. Присвоение значения для переменной осуществляется посредством оператора присваивания. В левой части оператора присваивания – имя переменной, а в правой – значение или выражение.

Переменные могут создаваться автоматически, по мере появления в процедуре (неявное объявление). По умолчанию неявно объявленные переменные имеют тип Variant. Внутри такой переменной кроме значения хранится индикатор типа значения.

Неявное объявление типа допустимо только в случае отсутствия оператора Option Explicit в области описания модуля. Чтобы при создании нового модуля предотвратить появление этого оператора в области описания, следует отключить опцию «Явное описание переменных» на вкладке «Модуль» диалогового окна «Параметры».

Рекомендуется описывать все переменные явно, это позволит избежать ошибок, связанных с преобразованием типов данных. Явное описание типа производится оператором Dim, после которого указывается имя переменной и ее тип. Если при явном объявлении переменной не указан тип данных, а задано только имя переменной то будет создана переменная типа Variant, которой можно присваивать цепочки символов, числа с плавающей точкой, значения даты и времени. Access однозначно идентифицирует присваиваемые значения, поскольку наряду с ними он хранит в переменной и признак типа.

Типы данных и размер хранимых значений, которые используются в Access, можно найти в справочной информации приложения.

Для объявления нескольких переменных можно пользоваться одним оператором Dim, перечисляя их через запятую.

При объявлении переменной следует помнить об области действия. Переменные, объявленные в процедуре, можно использовать только внутри этой процедуры. Если переменная должна быть доступна во всех процедурах одного модуля, ее необходимо объявить в области описания модуля. Здесь также можно применять оператор Dim. Наибольшую область действия имеет переменная, объявленная глобальной (с префиксом Global) в области описания

модуля. Синтаксис оператора Global не отличается от синтаксиса оператора Dim.

В отличие от переменной константа содержит фиксированное значение, которое не может быть изменено в процессе выполнения программы. Согласно правилам хорошего тона, константы, как и переменные, следует объявлять явно, указывая их имена и значения. Для объявления константы используется оператор Const. Объявленную константу можно использовать в программе, обращаясь к ней по имени. Например:

```
Const Число_Пи = 3.1415926
```

Условные операторы. В языке VBA, как и в других языках программирования, основными элементами, управляющими ходом выполнения процедуры, являются условные операторы. Наиболее простой из них – оператор If . . . Then:

```
If  Полина = "Ж" Then
  Поздравление = "С 8 марта!"
End if
```

Если условие, заданное выражением между ключевыми словами If и Then, выполняется, соблюдаются инструкции внутри блока, ограниченного ключевыми словами Then и End if. Иначе операторы между ключевыми словами не выполняются, а Access перейдет к обработке оператора, который следует за ключевыми слова End if. В случае необходимости произвести два различных действия (одно при соблюдении условия, а второе – нет), надлежит воспользоваться полной формой оператора If:

```
If  выражение Then
  Оператор1
Else
  Оператор2
End if
```

Если условие соблюдается, выполняется Оператор1 (или группа операторов), расположенный между ключевыми словами Then и Else, а если не соблюдается – Оператор2 (или группа операторов), расположенный между ключевыми словами Else и End if.

Однако не всегда возможны два варианта решения. Учитывая это, VBA предоставляет в распоряжение пользователей оператор Select Case, предназначенный для выбора одного из множеств вариантов решений:

```
Select Case Город
  Case "Киев"
```

```
    Код_МГТС = "044"
  Case "Москва"
    Код_МГТС = "095"
  Case "Запорожье"
    Код_МГТС = "061-2"
  Case Else
    Print "Я не знаю такого города!"
End Select
```

При выполнении этого оператора проверяется значение переменной «Город». В зависимости от результата проверки переменной «Код_МГТС» присваивается телефонной код некоторого города. Если значение переменной «Город» не совпадает ни с одним из значений, перечисленных в строках с ключевым словом Case, производится действие, указанное между ключевыми словами Case Else и End Select.

Операторы цикла. Цикл используется для многократного повторения одной или нескольких инструкций. Количество повторений цикла связано с некоторым условием. В VBA предусмотрено несколько разновидностей циклов. Простейшим примером циклической конструкции является так называемый цикл по счетчику:

```
For Счетчик = 1 To 10
    Print Счетчик
Next Счетчик
```

Цикл по счетчику ограничивается ключевыми словами For и Next. После ключевого слова For указывается имя переменной, которая будет выполнять роль счетчика, после знака равенства – начальное значение счетчика, а после ключевого слова To – конечное значение счетчика.

Еще одна разновидность цикла – While-цикл. Условие выполнения команд внутри такого цикла определяется некоторым условным оператором:

```
Do While Счетчик <> 10
  Print
Loop
```

Разновидностью цикла Do является Until-цикл, который выполняется, пока условное выражение ложно:

```
Do Until Счетчик=10
    Print
  Счетчик=Счетчик+1
Loop
```

Задание к лабораторной работе

1. Написать программу для добавления полей в таблицу, следуя приведенной ниже методике (в данном примере поля будут добавляться только в таблицу «Книги»):

а) Создать новую форму в режиме конструктора, в которой будут размещаться элементы управления, предназначенные для добавления поля.

б) Выбрать элемент типа «поле» на панели элементов управления для ввода имени добавляемого поля и разместить его в области данных формы.

в) Далее необходимо изменить свойства этого элемента управления. Правой кнопкой мыши щелкнуть по этому полю и выбрать параметр «Свойства». В качестве имени указать «Field_Name».

г) Отредактировать надпись слева от созданного поля ввода. В свойствах изменить подпись на «Название поля:». Для того чтобы подпись вмещалась полностью, в «Свойствах» ширину поля указать равной 2,6 см.

д) Теперь следует создать элемент управления, в котором будет содержаться перечень возможных типов для добавляемого поля. Выбрать на панели элементов «Поле со списком» и разместить его в форме. В предложенном списке способов заполнения элементов списка выбрать «Будет введен фиксированный набор значений» и нажать кнопку «Далее». Число столбцов оставить равным единице. В качестве элементов списка указать «Текстовый», «Числовой», «Дата/время». Нажать кнопку «Далее», в качестве подписи указать «Тип поля:» и нажать кнопку «Готово».

е) В свойствах созданного поля со списком изменить «Название поля» на «Field_Type».

ж) Теперь следует создать кнопку, в которой будет содержаться программа добавления поля. В панели элементов выбрать элемент управления «Кнопка» и разместить ее в форме. Будет предложен список действий, которые должны выполняться при нажатии на данную кнопку. Поскольку в Microsoft Access нет готовых программ для добавления полей, нажать кнопку «Отмена».

з) Перейти к свойствам созданной кнопки. «Подпись» изменить на «Добавление поля», «Имя» на «Add». В событиях выбрать поле «Нажатие кнопки». Слева от поля появится кнопка с тремя точками. Нажать на нее. В появившемся диалоговом окне выбрать «Программы» и нажать кнопку «ОК». В результате вы перейдете к окну редактирования программных модулей. Оно содержит следующие строки:

```
Option Compare Database
Option Explicit

Private Sub Add_Click ()
End Sub
```

Курсор должен находиться после строки «Private Sub Add_Click ()».С этой позиции будем писать программу обработки нажатия кнопки.

и) Для начала необходимо получить доступ ко всей базе данных в целом и к нужной таблице в частности. Кроме того, потребуется переменная, содержащая информацию о новом поле. Поэтому нужно объявить переменные, необходимые для создания поля:

```
Dim dbs As Database, tdf As TableDef, fld As Field
```

Переменная dbs имеет тип Database и предназначена для получения указателя на базу данных, tdf имеет тип TableDef – для получения указателя на таблицу, fld имеет тип Field – для получения указателей на поля таблицы.

к) Установим указатель на текущую базу данных:

```
Set dbs = CurrentDb
```

Ключевое слово «CurrentDb» означает, что информацию будем получать по текущей базе данных.

л) Установить указатель на таблицу:

```
Set tdf = dbs.TableDefs ("Книги")
```

Обратите внимание, что после ввода точки после «dbs» был выведен перечень переменных и функций текущей базы данных. Функция TableDefs предназначена для получения указателя на требуемую таблицу из заданной базы данных.

м) Теперь в зависимости от выбранного типа поля организовать создание нового поля в таблице. Для организации условного перехода внутри программы воспользуемся оператором Select Case:

```
Select Case Me.Field_Type.Value
 Case "Текстовый"
  Set fld=tdf.CreateField (Me.Field_Name, dbText)
 Case "Числовой"
  Set fld=tdf.CreateField (Me.Field_Name, dbLong)
 Case "Дата/время"
  Set fld=tdf.CreateField (Me.Field_Name, dbDate)
End Select
```

То есть в зависимости от значения поля со списком Field_Type происходит создание поля с выбранным типом. Для создания поля используется метод в CreateField. Созданное поле размещается в переменной «fld». Ключевое слово «Me.» означает, что следующие за ним поля принадлежат текущей форме.

н) Для добавления поля в таблицу воспользуемся командами:

```
tdf.Fields.Append fld
tdf.Fields.Refresh
```

Первая команда предназначена для добавления информации о новом поле в таблицу, вторая – для обновления произведенных изменений. Итак, вся программа имеет вид:

```
Private Sub Add_Click ()

Dim dbs As Database, tdf As TableDef, fld As Field

Set dbs = CurrentDb
Set tdf = dbs.TableDefs ("Книги")
Select Case Types.Value
    Case "Текстовый"
        Set fld = tdf.CreateField (Name, dbText)
    Case "Числовой"
        Set fld = tdf.CreateField (Name, dbLong)
    Case "Дата/время"
        Set fld = tdf.CreateField (Name, dbDate)
End Select

tdf.Fields.Append fld
tdf.Fields.Refresh

End Sub
```

о) После написания программы закрыть текущее окно.

п) Сохранить форму и открыть ее в обычном режиме.

р) Проверить работу программы: в качестве названия поля ввести «Test», в качестве типа указать «Текстовый». Нажать кнопку «Добавление поля», а затем открыть таблицу «Книги» и убедиться в присутствии нового поля «Test».

2. Написать программу для удаления полей из таблицы, следуя приведенной ниже методике (в данном примере поля будут удаляться только из таблицы «Книги»).

а) Открыть новую форму в режиме Конструктора.

б) Выбрать на панели элементов управления «Список» и разместить его в форме.

в) Из предлагаемых способов заполнения элементов списка выбрать «Будет введен фиксированный набор значений» и нажать кнопку «Далее». Число столбцов оставить равным единице. В качестве элементов списка указать «Автор», «Название», «Код издательства», «Объем», «Год издания», «Стоимость», «Test». Поле «Код» не добавлять, так как оно является ключевым

и не может быть удалено из таблицы. Нажать кнопку «Далее», в качестве подписи указать «Список полей» и нажать кнопку «Готово». При необходимости выровнять размер подписи.

г) В свойствах задать списку имя «Fields».

д) Создать кнопку, в которой будет содержаться программа удаления поля. На панели элементов выбрать элемент управления «Кнопка» и разместить ее в форме. Вам будет предложен список действий, которые должны выполняться при нажатии на данную кнопку. Поскольку в Microsoft Access нет готовых программ для удаления полей, нажать кнопку «Отмена».

е) В свойствах созданной кнопки «Подпись» изменить на «Удаление поля», «Имя» на «Del» и открыть программу обработки события в поле «Нажатие кнопки». Курсор находится после строки «Private Sub Del_Click ()».С этой позиции следует писать программу обработки события «Нажатие кнопки».

ж) Переменные, необходимые для создания поля:

```
Dim dbs As Database, tdf As TableDef
```

з) Указатель на текущую базу данных:

```
Set dbs = CurrentDb
```

и) Указатель на таблицу:

```
Set tdf = dbs.TableDefs ("Книги")
```

к) Удаление выбранного поля:

```
tdf.Fields.Delete (Me.Fields.Value)
```

л) Обновление списка полей в таблице:

```
tdf.Fields.Refresh
```

м) Программа должна иметь вид:

```
Private Sub Del_Click ()

Dim dbs As Database, tdf As TableDef

Set dbs = CurrentDb
Set tdf = dbs.TableDefs ("Книги")
```

```
    tdf.Fields.Delete (Me.Fields.Value)
    tdf.Fields.Refresh

    End Sub
```

н) Сохранить форму и открыть ее в режиме просмотра.

о) В качестве примера выбрать созданное нами в предыдущем примере поле «Test» и нажать кнопку «Удаление поля». Список не изменился, но если открыть таблицу «Книги», то можно убедиться, что поле «Test» удалено.

3. Создать форму (см. рис. 16) , в которой:

а) Реализовать добавление и удаление любого поля (кроме ключевых) из любой таблицы.

б) В качестве возможных типов добавляемых полей должны быть: «Текстовый», «Числовой», «Дата/время», «Денежный», «Логический», «Поле MEMO».

в) Предусмотреть защиту от удаления ключевых полей и защиту от добавления уже существующего поля, либо поля с неуказанным именем или типом данных.

г) Обеспечить обновление списка полей и поля ввода на форме после добавления/удаления полей в выбранной таблице.

д) Перед удалением поля должно предварительно появиться диалоговое окно («Message Box») для подтверждения удаления. Поле удалять только в случае подтверждения удаления.

е) Предусмотреть возможность открытия любой таблицы из созданной формы для просмотра внесенных изменений.

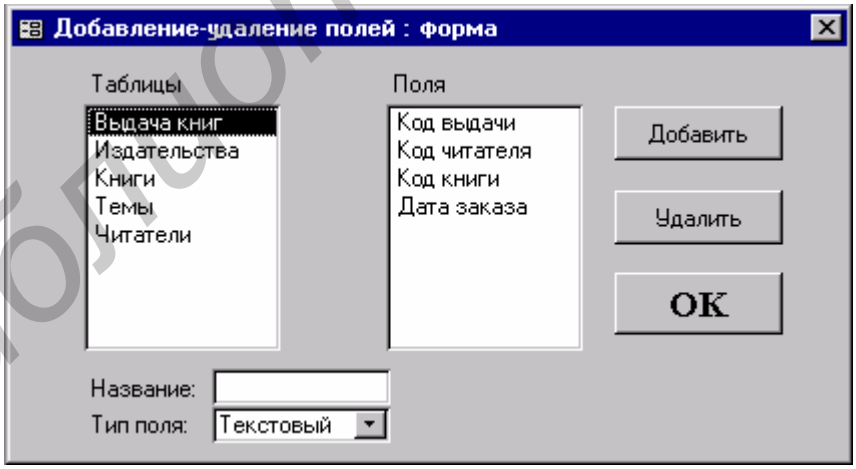

Рис. 16. Форма для добавления и удаления полей

ЛИТЕРАТУРА

1. Харитонова С.А. Access 2000. Разработка приложений. – СПб.: BHV–Санкт-Петербург, 2000. – 832 с.

2. Access 2000. Базы данных и приложения: Лекции и упражнения. – Киев: ДиаСофт, 2000. – 512 с.

3. Новалис С. Access 2000: Руководство по VBA: Пер. А. Киселева. – М.: ЛОРИ, 2001. – 506 с.

4. Боровиков В.В. Access 2002. Программирование и разработка баз данных и приложений., –М.: Солон - Р, 2002. – 560 с.

5. Профессиональное программирование в Microsoft Access 2002. – М: Вильямс, 2002. – 992 с.

6. Веремеенко Е.Г. Лабораторные работы по курсу «Базы данных и знаний». Ч. 1. URL:http://khpi-iip.mipk.kharkiv.edu.

7. Кириллов В.В. Проектирование реляционных баз данных. URL: http://khpi-iip.mipk.kharkiv.edu.

8. СУБД Access. Краткий учебный курс URL: http://www.test-center.narod.ru.

ПРИЛОЖЕНИЕ 1

ВАРИАНТЫ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ

Вариант 001. Создайте БД отдела кадров университета. Ориентировочные таблицы-составляющие: «Сотрудники», «Штатное расписание», «Контакты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Доценты» (запрос – условная выборка); «Сводка количества работающих на каждой должности» (итоговый запрос); «Сотрудники, нуждающиеся в продлении контракта» (параметрический запрос); «Динамика текучести кадров» (перекрестный запрос).

Вариант 002. Создайте БД галантерейного магазина. Ориентировочные таблицы-составляющие: «Партии товара», «Поставщики», «Продажи». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Залежавшийся товар» (запрос – условная выборка); «Рейтинг спроса по фирмам» (итоговый запрос); «Поставщики партий заданного объема» (параметрический запрос); «Динамика продаж по базовым видам продукции» (перекрестный запрос).

Вариант 003. Создайте БД отдела доставки почтового отделения. Ориентировочные таблицы-составляющие: «Подписчики», «Периодика», «Подписки». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Подписчики газеты "Вечерний Минск"» (запрос – условная выборка); «Количество подписок на каждое издание» (итоговый запрос); «Подписки дешевле заданной стоимости» (параметрический запрос); «Динамика цен на подписку на заданные издания» (перекрестный запрос).

Вариант 004. Создайте БД универмага. Ориентировочные таблицы-составляющие: «Товары», «Отделы», «Продажа». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список отделов, реализующих парфюмерию» (запрос – условная выборка); «Сводка продаж по отделам» (итоговый запрос); «Список продукции в заданном отделе» (параметрический запрос); «Динамика продаж по отделам» (перекрестный запрос).

Вариант 005. Создайте БД штатного расписания предприятия. Ориентировочные таблицы-составляющие: «Отделы», «Должности», «Сотрудники». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список сотрудников, не занятых в основном производстве» (запрос – условная выборка); «Сводка по заработной плате каждого отдела» (итоговый запрос); «Список сотрудников заданной должности» (параметрический запрос); «Динамика освобождения рабочих мест по мере выхода сотрудников на пенсию» (перекрестный запрос).

Вариант 006. Создайте БД фирмы по производству пиломатериалов. Ориентировочные таблицы-составляющие: «Изделия», «Сырье», «Продажи». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Реализованные изделия за последнюю неделю» (запрос – условная выборка); «Сводка расхода сырья» (итоговый запрос); «Продажи изделий заданной породы древесины» (параметрический запрос); «Динамика продаж изделий по группам» (перекрестный запрос).

Вариант 007. Создайте БД программы выпуска деталей литейного цеха. Ориентировочные таблицы-составляющие: «Детали», «Материал», «Технологические процессы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Чугунные отливки большого объема» (запрос – условная выборка); «Расход материала в плановом периоде» (итоговый запрос); «Процессы выплавки деталей заданных габаритов» (параметрический запрос); «Динамика выпуска деталей (по материалу)» (перекрестный запрос).

Вариант 008. Создайте БД инфекционного отделения городской больницы. Ориентировочные таблицы-составляющие: «Койко-место», «Больные», «Диагнозы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Больные-пенсионеры» (запрос – условная выборка); «Количество больных в каждой палате» (итоговый запрос); «Палаты больных с заданной температурой» (параметрический запрос); «Динамика заболеваемости» (перекрестный запрос).

Вариант 009. Создайте БД процессов обработки партий деталей. Ориентировочные таблицы-составляющие: «Детали», «Оборудование», «Технологические карты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список оборудования для высококачественной обработки деталей» (запрос – условная выборка); «Загруженность оборудования» (итоговый запрос); «Партии, проходящие заданную операцию» (параметрический запрос); «Количество партий на оборудование по заданным интервалам времени» (перекрестный запрос).

Вариант 010. Создайте БД фирмы по оптовой реализации бытовой техники. Ориентировочные таблицы-составляющие: «Продукция», «Клиенты», «Заказы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Крупнейшие партии» (запрос – условная выборка), «Сводка по технике, закупленной каждым клиентом» (итоговый запрос); «Продукция, реализованная в заданный период времени» (параметрический запрос); «Динамика популярности фирм-изготовителей» (перекрестный запрос).

Вариант 011. Создайте БД оборудования НИИ. Ориентировочные таблицы-составляющие: «Оборудование», «Исследовательские работы», «Акты проведения работ». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Наиболее загруженное оборудование» (запрос – условная выборка); «Количество часов наработки» (итоговый запрос); «Проведение работ на заданной единице оборудования» (параметрический запрос); «Распределение общего количества единиц оборудования по темам исследований во времени» (перекрестный запрос).

Вариант 012. Создайте БД фирмы по производству столярных работ. Ориентировочные таблицы-составляющие: «Продукция/услуги», «Клиенты», «Договора». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Постоянные клиенты фирмы» (запрос – условная выборка); «Прибыль по каждому виду продукции/услуги» (итоговый запрос); «Продукция/услуги, реализованные на заданную сумму» (параметрический запрос); «Динамика пользовательского спроса продукции» (перекрестный запрос).

Вариант 013. Создайте БД агентства недвижимости. Ориентировочные таблицы-составляющие: «Объекты недвижимости», «Покупка», «Продажа». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список объектов, предлагаемых к продаже» (запрос – условная выборка); «Сальдо по видам объектов» (итоговый запрос); «Объекты заданной стоимости» (параметрический запрос); «Динамика продаж по районированию объектов» (перекрестный запрос).

Вариант 014. Создайте БД фирмы-распространителя программного обеспечения. Ориентировочные таблицы-составляющие: «Программное обеспечение», «Клиенты», «Производимые работы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список клиентов, программному обеспечению которых предстоит обновление» (запрос – условная выборка); «Сводка реализованного программного обеспечения» (итоговый запрос); «Список клиентов, купивших продукцию заданного вида» (параметрический запрос); «Динамика обновления версий» (перекрестный запрос).

Вариант 015. Создайте БД инструментального склада. Ориентировочные таблицы-составляющие: «Обслуживаемое оборудование», «Инструмент», «Технологические карты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Инструмент с высокой степенью используемости» (запрос – условная выборка); «Количество единиц инструмента для каждого оборудования» (итоговый запрос); «Список инструмента, подверженного повышенному износу» (параметрический запрос); «Количество единиц инструмента для токарного оборудования по технологическим картам» (перекрестный запрос).

Вариант 016. Создайте БД страховой фирмы. Ориентировочные таблицы-составляющие: «Виды страховок», «Клиенты/объекты страховки», «Страховая деятельность». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Клиенты, застраховавшие свою жизнь за последний месяц» (запрос – условная выборка); «Сводка полученных/выплаченных сумм страховок по клиентам» (итоговый запрос); «Объекты, застрахованные на заданную сумму» (параметрический запрос); «Динамика заключения страховых договоров (по основным видам)» (перекрестный запрос).

Вариант 017. Создайте БД подъемно-транспортного оборудования машиностроительного предприятия. Ориентировочные таблицы-составляющие: «Техника», «Производимые работы», «Ремонт». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Работа техники высокой грузоподъемности» (запрос – условная выборка); «Сводка часов простоя единиц оборудования во внеплановом ремонте» (итоговый запрос); «Список техники с заданным коэффициентом загрузки» (параметрический запрос); «Динамика поломок» (перекрестный запрос).

Вариант 018. Создайте БД музея. Ориентировочные таблицы-составляющие: «Экспонаты», «Авторы», «Экспозиции». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Малоизвестные экспонаты» (запрос – условная выборка); «Выставлено работ по авторам» (итоговый запрос); «Экспозиции работ повышенной ценности» (параметрический запрос); «Динамика экспозиции работ по видам» (перекрестный запрос).

Вариант 019. Создайте БД НИИ. Ориентировочные таблицы-составляющие: «Сотрудники», «Научно-исследовательские разработки», «Штатное расписание». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Сотрудники, задействованные в научно-исследовательских разработках» (запрос – условная выборка); «Выплаченная заработная плата (по отделам)» (итоговый запрос); «Список ответственных по научно-исследовательским разработкам» (параметрический запрос); «Динамика ведения научных работ по месяцам» (перекрестный запрос).

Вариант 020. Создайте БД студии видеозаписи. Ориентировочные таблицы-составляющие: «Режиссеры/Актеры», «Фильмы», «Продажи». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Оскар» (запрос – условная выборка); «Рейтинг продаж по актерам» (итоговый запрос); «Фильмы с заданным актерским дуэтом» (параметрический запрос); «Динамика продаж по жанрам» (перекрестный запрос).

Вариант 021. Создайте БД деканата. Ориентировочные таблицы-составляющие: «Список курсов», «Список студентов», «Сессии». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Неуспевающие студенты» (запрос – условная выборка); «Средний балл по предметам» (итоговый запрос); «Студенты, имеющие хорошие оценки по заданному предмету» (параметрический запрос); «Динамика сдачи сессий по группам» (перекрестный запрос).

Вариант 022. Создайте БД автозаправочной станции. Ориентировочные таблицы-составляющие: «Горюче-смазочные материалы», «Поставщики», «Накладные». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Закупки дизельного топлива» (запрос – условная выборка); «Затраты по видам продукции» (итоговый запрос); «Поставщики дешевой смазки» (параметрический запрос); «Динамика закупочных цен по основным видам продукции» (перекрестный запрос).

Вариант 023. Создайте БД выставки продукции. Ориентировочные таблицы-составляющие: «Продукция», «Предприятие», «Выставочное место». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Предприятия, арендовавшие наибольшие выставочные площади» (запрос – условная выборка); «Сводка по продукции/предприятиям» (итоговый запрос); «Продукция заданного предприятия» (параметрический запрос); «Количество выставляемой продукции по предприятиям» (перекрестный запрос).

Вариант 024. Создайте БД полиграфической фирмы. Ориентировочные таблицы-составляющие: «Материалы», «Техника», «Контракты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Материалы заказов малых объемов» (запрос – условная выборка); «Сводка расходов материалов» (итоговый запрос); «Заказы на заданную сумму» (параметрический запрос); «Динамика загруженности оборудования» (перекрестный запрос).

Вариант 025. Создайте БД фирмы-поставщика медицинской техники. Ориентировочные таблицы-составляющие: «Ассортимент», «Заказчики», «Контракты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Последние поступления» (запрос – условная выборка); «Рейтинг заказчиков по общим суммам контрактов» (итоговый запрос); «Клиенты, заказавшие

заданный вид изделия» (параметрический запрос); «Динамика реализации заданных единиц ассортимента» (перекрестный запрос).

Вариант 026. Создайте БД планово-финансового отдела. Ориентировочные таблицы-составляющие: «Участки», «Работники», «Наряды». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Опасные работы» (запрос – условная выборка); «Начисление заработной платы по участкам» (итоговый запрос); «Списки работников по заданным датам/участкам» (параметрический запрос); «Количество работников на участках по месяцам» (перекрестный запрос).

Вариант 027. Создайте БД строительной фирмы. Ориентировочные таблицы-составляющие: «Объекты», «Этапы выполнения», «Стройматериалы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Текущие этапы работы» (запрос – условная выборка); «Сроки строительства объектов» (итоговый запрос); «Объекты заданного процента завершенности» (параметрический запрос); «Распределение материалов по объектам» (перекрестный запрос).

Вариант 028. Создайте БД опытного цеха. Ориентировочные таблицы-составляющие: «Производимые изделия», «Оборудование», «Технологические карты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Изделия/карты фрезерной обработки» (запрос – условная выборка); «Количество изделий по каждому оборудованию» (итоговый запрос); «Карты изделий заданных габаритов» (параметрический запрос); «Распределение количества единиц оборудования по заданным изделиям во времени» (перекрестный запрос).

Вариант 029. Создайте БД механообрабатывающего цеха. Ориентировочные таблицы-составляющие: «Детали», «Материалы», «Оснастка». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Оснастка для обработки деталей высокой твердости» (запрос – условная выборка); «Расход материалов по видам деталей» (итоговый запрос); «Детали заданного материала» (параметрический запрос); «Распределение количества деталей по материалу и требуемым видам оснастки» (перекрестный запрос).

Вариант 030. Создайте БД фирмы по установке и обслуживанию ЛВС. Ориентировочные таблицы-составляющие: «Сетевые конфигурации», «Клиенты», «Договора». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список клиентов, с которыми необходимо перезаключать договора» (запрос – условная выборка); «Протяженность кабеля у каждого клиента» (итоговый запрос); «Список ЛВС с заданным количеством станций» (параметрический запрос); «Динамика спроса базовых сетевых конфигураций» (перекрестный запрос).

Вариант 031. Создайте БД фирмы по продаже стройматериалов. Ориентировочные таблицы-составляющие: «Продукция», «Заказчики», «Контракты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Продажа кирпича» (запрос – условная выборка); «Сводка сумм контрактов по каждому заказчику» (итоговый запрос); «Список продукции, заказанной в заданный период» (параметрический запрос); «Динамика заключения контрактов по месяцам» (перекрестный запрос).

Вариант 032. Создайте БД ветеринарной станции. Ориентировочные таблицы-составляющие: «План ветеринарных работ», «Обслуживаемые хозяйства», «Проведенные работы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Список хозяйств, запланированных на ближайшую неделю» (запрос – условная выборка); «Объемы проведенных работ в каждом хозяйстве» (итоговый запрос); «Выполнение плана по хозяйствам на заданное число» (параметрический запрос); «Распределение количества осмотренных животных по дням недели» (перекрестный запрос).

Вариант 033. Создайте БД фирмы по проведению буровых работ. Ориентировочные таблицы-составляющие: «Услуги-расценки», «Клиенты», «Контракты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Контракты на бурение глубинных скважин» (запрос – условная выборка); «Количество контрактов на каждый вид работ» (итоговый запрос); «Контракты с заданным клиентом» (параметрический запрос); «Метраж пробуренных скважин за последние 10 недель» (перекрестный запрос).

Вариант 034. Создайте БД загрузки аудиторий. Ориентировочные таблицы-составляющие: «Аудитории», «Учебные дисциплины», «Группы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Поточные (лекционные) аудитории» (запрос – условная выборка); «Рейтинг дисциплин по загрузке аудиторий» (итоговый запрос); «Аудитории заданной дисциплины» (параметрический запрос); «Динамика занятости заданной группой аудиторий по дням недели» (перекрестный запрос).

Вариант 035. Создайте БД трикотажной фабрики. Ориентировочные таблицы-составляющие: «Сырье», «Изделия», «Сбыт изделий». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Сбыт изделий, содержащих искусственные волокна» (запрос – условная выборка); «Рейтинг себестоимости изделий» (итоговый запрос); «Картина сбыта изделий заданного размера» (параметрический запрос); «Динамика сбыта шерстяных изделий по месяцам» (перекрестный запрос).

Вариант 036. Создайте БД ателье головных уборов. Ориентировочные таблицы-составляющие: «Изделия», «Клиенты», «Квитанции». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Заказчики зимнего ассортимента» (запрос – условная выборка); «Популярность моделей» (итоговый запрос); «Модели шляп заданной стоимости» (параметрический запрос); «Количество заказов заданных моделей по месяцам» (перекрестный запрос).

Вариант 037. Создайте БД гостиницы. Ориентировочные таблицы-составляющие: «Номера», «Клиенты», «Счета». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Богатые клиенты» (запрос – условная выборка); «Рейтинг загрузки номеров» (итоговый запрос); «Счета указанного клиента» (параметрический запрос); «Динамика наплыва посетителей по месяцам» (перекрестный запрос).

Вариант 038. Создайте БД жилищного коммунального хозяйства. Ориентировочные таблицы-составляющие: «Специалисты», «Жилищный фонд», «Мероприятия». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Деятельность, запланированная на ближайшую неделю» (запрос – условная выборка); «Рейтинг работников по количеству мероприятий» (итоговый запрос); «Мероприятия, проведенные заданным работником» (параметрический запрос); «Количество проведенных мероприятий по месяцам» (перекрестный запрос).

Вариант 039. Создайте БД стоматологической поликлиники. Ориентировочные таблицы-составляющие: «Врачи», «Пациенты», «Обслуживание». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Протезирование» (запрос – условная выборка); «Рейтинг услуг» (итоговый запрос); «Пациенты заданного врача» (параметрический запрос); «Динамика обращений за помощью по месяцам года» (перекрестный запрос).

Вариант 040. Создайте БД сборочного процесса. Ориентировочные таблицы-составляющие: «Комплектующие», «Изделия», «Технологические процессы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Изделия, требующие микросборки» (запрос – условная выборка); «Количество комплектующих в каждом изделии» (итоговый запрос); «Информация по изделиям заданного технологического процесса» (параметрический запрос); «Динамика сборки изделий за заданный интервал времени планового периода» (перекрестный запрос).

Вариант 041. Создайте БД мебельной фабрики. Ориентировочные таблицы-составляющие: «Изделия», «Заказчики», «Контракты». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Контракты по корпусной мебели» (запрос – условная выборка); «Рейтинг продукции» (итоговый запрос); «Последние контракты» (параметрический запрос); «Динамика популярности заданных видов продукции» (перекрестный запрос).

Вариант 042. Создайте БД кабельного завода. Ориентировочные таблицы-составляющие: «Сырье», «Продукция», «Технологические процессы». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Монтажные провода» (запрос – условная выборка); «Затраты сырья по видам продукции» (итоговый запрос); «Исходные материалы для производства проводников заданного типа» (параметрический запрос); «Расход меди на погонный метр в зависимости от толщины кабеля по видам продукции» (перекрестный запрос).

Вариант 043. Создайте БД механизированной колонны. Ориентировочные таблицы-составляющие: «Техника», «Обслуживающий персонал», «Путевки/наряды». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Экскаваторщики» (запрос – условная выборка); «Отработано персоналом часов за неделю» (итоговый запрос); «Наряды в заданный промежуток времени» (параметрический запрос); «Динамика загруженности техники» (перекрестный запрос).

Вариант 044. Создайте БД санатория. Ориентировочные таблицы-составляющие: «Оздоровительные программы», «Отдыхающие», «Диагностируемые заболевания». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Программы по желудочно-кишечным заболеваниям» (запрос – условная выборка); «Сводка отдыхающих за плановый период» (итоговый запрос); «Отдыхающие, завершающие курс оздоровления» (параметрический запрос); «Динамика заболеваний по месяцам» (перекрестный запрос).

Вариант 045. Создайте БД геологоразведочной экспедиции. Ориентировочные таблицы-составляющие: «Регионы», «Карты», «Экспедиции». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Экспедиции по Беларуси» (запрос – условная выборка); «Протяженность маршрутов по регионам» (итоговый запрос); «Перечень карт заданного региона» (параметрический запрос); «Количество листов карт по каждой экспедиции» (перекрестный запрос).

Вариант 046. Создайте БД фирмы по автоматизации производства. Ориентировочные таблицы-составляющие: «Каталог устройств и программного обеспечения», «Клиенты», «Договора». Общее количество полей в базе – не менее 20, общее количество записей – не менее 30.

Создайте отчеты: «Услуги, оказанные в текущем месяце» (запрос – условная выборка); «Сводка доходов по виду услуг за отчетный период» (итоговый запрос); «Договора по установке программного обеспечения» (параметрический запрос); «Динамика спроса по месяцам» (перекрестный запрос).

ПРИЛОЖЕНИЕ 2

ПЕРЕЧЕНЬ ТАБЛИЦ УЧЕБНОЙ БАЗЫ ДАННЫХ «БИБЛИОТЕКА»

Таблица 1

Таблица «Издательства»

Код издательства	Название	Город
1	Наука	Москва
2	Мир	Москва
3	Радио и связь	Москва
4	Машиностроение	Киев

Таблица 2

Таблица «Книги»

Код книги	Автор	Название	Код издательства	Объем	Год издания	Стоимость
1	Беспалько	Педагогика	2	340	1994	9 900 р.
2	Сканави	Сборник задач	2	634	1992	6 600 р.
3	Арсак	Программирование	1	273	1989	19 800 р.
4	Перминов	Язык АДА	3	278	1987	4 400 р.
5	Грибанов	Операционные машины	3	446	1991	13 200 р.
6	Ульман	БД на Паскале	4	563	1992	3 200 р.

Таблица 3

Таблица «Выдача книг»

Код выдачи	Код читателя	Код книги	Дата заказа
1	1	1	01.09.96
2	1	2	01.03.97
3	1	5	20.04.97
4	2	1	04.11.95
5	2	3	20.12.96
6	3	1	03.08.95
7	3	2	20.01.96
8	4	2	03.02.96
9	5	1	04.10.96
10	6	3	18.03.97
11	7	4	23.04.97
12	7	6	06.05.97
13	8	5	01.03.97
14	8	6	04.05.97
15	9	6	27.04.97
16	10	5	20.10.97

Таблица 4

Таблица «Темы»

Код темы	Код книги	Название темы
1	1	Личность человека
2	1	Проектирование ППС
3	1	Технология обучения
4	1	Анализ учебного процесса
5	2	Уравнения
6	2	Прогрессии
7	2	Геометрические задачи
8	3	Игры с числами
9	3	Игры без стратегии
10	3	Комбинаторные задачи
11	3	Стратегия без игры
12	4	Программные модули
13	4	Лексика
14	4	Предопределенные типы
15	4	Операторы
16	5	Структура ОС ЕС
17	5	Управление заданиями
18	5	Управление задачами
19	5	Управление данными
20	6	Операции с поставщиками
21	6	Бухгалтерская книга
22	6	Платежная ведомость
23	6	Реляционная алгебра
24	6	Правила нормализации

Таблица 5

Таблица «Читатели»

Код читателя	Фамилия	Имя	Отчество	Адрес	Домашний телефон
1	Минкевич	Андрей	Яковлевич	Ул. Московская, 17-19	233-33-33
2	Гуляев	Сергей	Дмитриевич	Ул. Козлова, 7-38	236-93-79
3	Федосенко	Олег	Георгиевич	Ул. Богдановича, 102-34	232-23-16
4	Захаров	Михаил	Степанович	Ул. Плеханова, 34-67	266-19-75
5	Бобров	Виктор	Иванович	Ул. Казинца, 23-47	245-42-26
6	Гусакович	Елена	Анатольевна	Пр. Скорины, 5-8	
7	Семенова	Галина	Ивановна	Ул. Денисовская, 20-47	251-49-67
8	Савченко	Ольга	Павловна	Пр. Пушкина, 30-41	231-99-06
9	Мицкевич	Мария	Степановна	Ул. Чкалова, 12-56	220-12-35
10	Карпович	Татьяна	Дмитриевна	Ул. Малинина, 118-56	234-80-34

Св. план 2003, поз. 19

Учебное издание

Антипова Марина Александровна,
Снисаренко Светлана Валерьевна,
Седушкин Артём Анатольевич

ИНФОРМАЦИОННЫЕ ОСНОВЫ СИСТЕМ УПРАВЛЕНИЯ

Методическое пособие
к лабораторным работам для студентов специальностей 53 01 03
«Автоматическое управление в технических системах» и 53 01 07
«Информационные технологии и управление в технических системах»
всех форм обучения

Редактор Т.А. Лейко
Корректор Е.Н. Батурчик
Компьютерная верстка Т.В. Шестакова

Подписано в печать 29.05.2003. Формат 60x84 1/16. Бумага офсетная.
Печать ризографическая. Гарнитура «Таймс». Усл. печ. л. 3,84.
Уч.-изд. л. 3,5. Тираж 200 экз. Заказ 99.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Лицензия ЛП № 156 от 30.12.2002.
Лицензия ЛВ № 509 от 03.08.2001.
220013, Минск, П. Бровки, 6.