



**Пример оформления лабораторного практикума
(фрагмент)**

**СИСТЕМЫ УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

МИНСК 2006

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«МИНСКИЙ ГОСУДАРСТВЕННЫЙ ВЫСШИЙ
РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Лабораторный практикум
для студентов специальности 1-08 01 01-07
«Профессиональное обучение. (Информатика)»

МИНСК 2006

УДК 681.3(075)
ББК 32.973.26–018.2я7
С40

Рекомендовано к изданию кафедрой информатики и Научно-методическим советом Учреждения образования «Минский государственный высший радиотехнический колледж»

С о с т а в и т е л и:
Е. А. Лазицкас, С. Е. Специан,
ассистенты кафедры информатики МГВРК

Р е ц е н з е н т
Ю. А. Скудняков, зав. кафедрой информатики МГВРК,
канд. техн. наук, доцент

С40 **Системы** управления базами данных : лаб. практикум для студентов специальности 1-08 01 01-07 «Профессиональное обучение. (Информатика)» / сост. Е. А. Лазицкас, С. Е. Специан. – Мн. : МГВРК, 2006. – 88 с.
ISBN 985-6754-52-6

В лабораторном практикуме приведены указания по выполнению лабораторных работ, описания 14-ти работ, каждая из которых содержит цель, теоретические сведения, необходимые для выполнения задания, порядок выполнения, контрольные вопросы.

Практикум предназначен для студентов и преподавателей колледжа.

УДК 681.3(075)
ББК 32.973.26–018.2я7

ISBN 985-6754-52-6

© Лазицкас Е. А., Специан С. Е., 2006
© Оформление. Учреждение образования «Минский государственный высший радиотехнический колледж», 2006

Предисловие

В состав данного лабораторного практикума включены 14 лабораторных работ, рассчитанных на 30 академических часов.

Предлагаемые работы должны выполняться в следующем порядке:

- первые десять работ в среде системы управления базами данных (СУБД) Access;

- последующие четыре работы должны быть выполнены в InterBase.

Задания, сформулированные в лабораторных работах, рассчитаны на первый и второй семестры изучения дисциплины СУБД.

Указания по выполнению лабораторных работ

Перед выполнением лабораторной работы студенты должны ознакомиться с предлагаемыми теоретическими сведениями, необходимыми для успешного выполнения конкретной работы. Следует внимательно ознакомиться с заданием, обдумать действия по выполнению работы, изучить при необходимости дополнительный теоретический материал, а затем приступить к выполнению задания.

Результаты выполнения всех лабораторных работ необходимо сохранить на магнитном носителе, так как они будут использоваться для выполнения последующих работ.

Отчет по лабораторной работе включает:

- демонстрацию созданной базы данных (БД);
- демонстрацию запланированных в задании действий над БД;
- устные пояснения по ходу выполнения лабораторной работы;
- ответы на контрольные вопросы по лабораторной работе.

К выполнению следующей работы студенты допускаются только после сдачи отчета по предыдущей лабораторной работе.

При выполнении лабораторных работ преподавателем могут выдаваться индивидуальные задания по различным темам лабораторного практикума, которые выполняются и дорабатываются в процессе изучения новых возможностей работы в СУБД.

ЛАБОРАТОРНАЯ РАБОТА 1

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Цель работы: закрепление знаний в теории нормализации баз данных; формирование знаний описания предметной области базы данных; формирование умений и навыков проектирования и разработки БД.

1. Краткие теоретические сведения

Предметная область – это часть реального мира, данные о которой необходимо отразить в БД. Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т. д.

Следует научиться выделять в предметной области важные понятия и малозначащие или вообще не значащие данные. Так, если в качестве предметной области выбрать учет товаров на складе, то понятия «накладная» и «счет-фактура» являются важными понятиями, а наличие у сотрудницы, принимающей накладные, двоих детей – это для учета товаров неважно. Однако, с точки зрения отдела кадров, данные о наличии детей являются важными. Таким образом, важность данных зависит от выбора предметной области.

Модель предметной области – это знания о предметной области. Модель предметной области описывает процессы, происходящие в предметной области, и данные, используемые этими процессами. Знания об этих процессах должны быть выражены с помощью текстовых описаний предметной области (наборов должностных инструкций, правил ведения дел в компании и т. п.).

При проектировании реляционной БД надо решить вопрос о наиболее эффективной структуре данных. Основные цели, которые при этом преследуются:

- 1) обеспечить быстрый доступ к данным в таблице;
- 2) исключить ненужное повторение;
- 3) обеспечить целостность данных.

Процесс уменьшения избыточной информации называется *нормализацией*.

Теория нормализации оперирует пятью нормальными формами таблиц. Эти формы предназначены для уменьшения избы-

точной информации от первой до пятой нормальной формы. Поэтому каждая последующая нормальная форма должна удовлетворять требованиям предыдущих форм и некоторым дополнительным условиям. При практическом проектировании БД четвертая и пятая нормальные формы чаще всего не используются.

Условия первой нормальной формы:

- 1) отношение не должно иметь повторяющихся записей;
- 2) в отношении должны отсутствовать повторяющиеся группы полей.

Условия второй нормальной формы:

- 1) отношение удовлетворяет условиям первой нормальной формы;
- 2) любое неключевое поле однозначно идентифицируется полным набором ключевых полей.

Условия третьей нормальной формы:

- 1) отношение удовлетворяет условиям второй нормальной формы;
- 2) ни одно из неключевых полей отношения не идентифицируется с помощью другого неключевого поля.

2. Практическая часть

Порядок выполнения работы

1. Выбрать вариант задания для выполнения работы.
2. Изучить предметную область БД.
3. Разработать структуру БД, приведя ее к третьей нормальной форме (БД должна состоять не менее чем из пяти таблиц).

Описание предметной области тематических заданий

Вариант 1

Разработать БД «АРМ библиотекаря», предназначенную для автоматизации работы библиотекаря колледжа. БД должна вести учет:

- 1) справочника книг, разбитых по тематике (добавление, удаление, редактирование информации о книге);
- 2) каждого экземпляра, находящегося в библиотеке (у каждого переплета должен быть собственный номер, шифр, обозначающий положение книги в хранилище);

3) перемещения книг (в библиотеке, на руках, в читальном зале, списана);

4) книг, подлежащих списанию (определить условие для списания книг, при списании обязательно указывать причину);

5) читателей библиотеки (добавление, удаление, редактирование информации о читателе);

6) карточек читателей (выдавать карточку читателей – список книг, заказываемых читателем, дату заказа книги, дату возврата, выдавать список читателей, вовремя не сдавших книги, информировать читателей, начислять штрафные санкции);

7) поставщиков книг (добавлять, редактировать и удалять информацию о поставщиках, у которых колледж заказывает книги);

8) поставок книг (когда и на какую сумму были произведены закупки книг).

Вариант 2

Разработать БД «АРМ работника склада», предназначенную для автоматизации работы сотрудника склада. БД должна вести учет:

1) заказчиков (добавление, редактирование и удаление информации обо всех заказчиках данной организации);

2) заказов (когда, по какой цене, кем и кому были отгружены товары);

3) продукции (какие товары, по какой цене, в каком количестве находятся на складе, формировать отпускную цену);

4) поставок (когда, кем, в каком количестве и по какой цене были произведены поставки товаров);

5) поставщиков (добавление, удаление и редактирование информации о всех поставщиках продукции);

6) работников склада (учет грузчиков, начисление им заработной платы);

7) прибыли склада (по какой цене закуплена продукция, по какой цене продана продукция, стоимость суток хранения единицы продукции).

Вариант 3

Разработать БД «АРМ администратора ателье по ремонту оргтехники». БД должна вести учет:

1) клиентов ателье (добавление, удаление и редактирование информации обо всех клиентах ателье);

2) техники, сданной в ремонт (какая техника, когда сдана в ремонт, какой ремонт необходим, в какой срок выполнится, степень готовности и информация о мастере, который будет выполнять работу);

3) комплектующих (стоимость комплектующих, использованных при ремонте, отдельный заказ комплектующих);

4) проделанной работы (какая техника, стоимость работы по категориям, формирование общей стоимости заказа, учет скидок и налогов при формировании цены, вид оплаты);

5) работников ателье (формирование заработной платы работникам в зависимости от объема выполненных работ).

Вариант 4

Разработать БД «АРМ администратора автосалона». БД должна вести учет:

1) автомобилей, находящихся в автосалоне (добавление, удаление и редактирование информации обо всех автомобилях в автосалоне или в гараже);

2) поставщиков автомобилей (добавление, удаление и редактирование информации обо всех поставщиках автомобилей);

3) клиентов автосалона (добавление, удаление и редактирование информации обо всех клиентах автосалона);

4) поставок (когда, кем, в каком количестве и по какой цене были произведены поставки автомобилей);

5) заказов (когда, по какой цене, кем и кому были проданы автомобили, в какой форме производится оплата);

6) продажи запчастей для автомобилей (заказ запчастей, если их нет на складе);

7) услуги тюнинга автомобилей (какой автомобиль, состав услуг, формирование стоимости услуг).

Вариант 5

Разработать БД «АРМ администратора ресторана». БД должна вести учет:

1) распределения столиков (на сколько мест, где расположены, учитывать, какая наценка идет на столик в зависимости от его расположения);

2) клиентов ресторана (добавление, удаление и редактирование информации обо всех клиентах ресторана);

3) предварительных заказов на столики (распределение предварительных заказов на столики, предоплата, меню на столик);

4) меню (формирование меню на конкретный день, стоимость блюд, напитков);

5) заказов на конкретный столик (формирование счета в зависимости от заказанных блюд, принятие оплаты).

Варианта 6

Разработать БД «АРМ сотрудника ЖЭСа». БД должна вести учет:

1) всех домов, подчиняющихся ЖЭСу (учет информации о количестве квартир, подъездов, этажей);

2) квартиросъемщиков (добавление, удаление и редактирование информации обо всех квартиросъемщиках ЖЭСа);

3) стоимости всех услуг ЖЭСа (стоимость единицы жилплощади, единицы горячей и холодной воды и т. д.);

4) льготных квартиросъемщиков ЖЭСа;

5) стоимости оплаты за квартиру (формирование квитанции об оплате с учетом параметров квартиры и льготников);

6) должников по оплате (начисление пени).

Вариант 7

Разработать БД «АРМ администратора аэропорта». БД должна вести учет:

1) всех самолетов (добавление, удаление и редактирование информации обо всех видах самолетов, обслуживаемых аэропортом);

2) всех рейсов (добавление, удаление и редактирование информации обо всех рейсах с указанием даты отправления и прибытия, вида самолета, фамилий членов экипажа);

3) билетов (добавление, удаление и редактирование информации о распределении билетов в зависимости от типа самолета, рейса и др.);

4) предварительных заказов билетов (формирование предварительного заказа билетов на конкретный рейс, предоплата билета);

5) стоимости билетов (формирование стоимости билета в зависимости от типа самолета, рейса, класса салона);

б) пассажиров (добавление, удаление и редактирование информации обо всех пассажирах аэропорта, их багаже, номерах билетов).

Вариант 8

Разработать БД «Учебный процесс в колледже». БД должна вести учет:

1) всех кафедр колледжа (добавление, удаление и редактирование информации обо всех кафедрах колледжа);

2) всех преподавателей колледжа (добавление, удаление и редактирование информации обо всех преподавателях колледжа);

3) всех групп колледжа (добавление, удаление и редактирование информации обо всех группах и отделениях колледжа);

4) всех студентов колледжа (добавление, удаление и редактирование информации обо всех студентах колледжа);

5) успеваемости студентов колледжа (информация о предмете, преподавателе, номере семестра, виде занятия (зачет, экзамен), группе, студенте и итоговой оценке);

б) изучения предметов (в какой группе, какой преподаватель и какой предмет читает с указанием часов на практические и лабораторные работы).

Контрольные вопросы

1. Дайте определение СУБД.
2. Опишите структуру реляционной базы данных.
3. В чем разница между первичным и внешним ключом?
4. Что такое нормализация?

ЛАБОРАТОРНАЯ РАБОТА 2

СРЕДА MICROSOFT ACCESS. РАЗРАБОТКА СТРУКТУРЫ БД. СОЗДАНИЕ И МОДИФИКАЦИЯ ТАБЛИЦ

Цель работы: формирование знаний, умений и навыков работы с окном и объектами СУБД Microsoft Access; закрепление умений в проектировании БД, определении типов данных атрибутов отношений и организации связей между отношениями.

1. Краткие теоретические сведения

1.1. Знакомство с окном СУБД Microsoft Access

На рис. 2.1 показан общий вид окна БД.

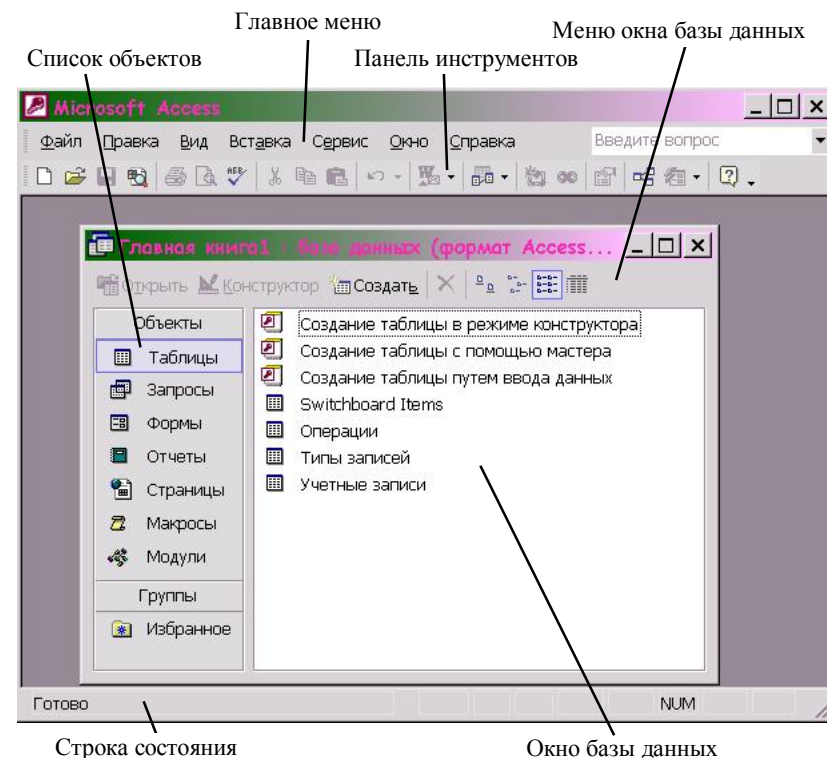


Рис. 2.1. Общий вид окна БД

С помощью *окна базы данных* можно управлять элементами БД: оно предоставляет доступ ко всем средствам БД, включая инструменты для создания новых и изменения содержащихся в БД объектов. Доступ к объектам осуществляется через *список объектов*.

Основные объекты СУБД Microsoft Access:

- **Таблица** – это набор данных по конкретной теме (например, товары или поставщики). Использование отдельной таблицы для каждой темы означает, что соответствующие данные сохранены только один раз, что делает базу данных более эффек-

тивной и уменьшает число ошибок при вводе данных;

- **Запросы** используются для просмотра, изменения и анализа данных различными способами. Запросы также можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным;

- **Формы** используются для визуального отображения данных из таблиц в удобном для пользователя виде;

- **Отчет** – эффективный способ представления данных в печатном формате. Имея возможность управлять размером и внешним видом всех элементов отчета, пользователь может отобразить сведения желаемым образом;

- **Страницы доступа к данным** представляют собой специальный тип Web-страниц, предназначенный для просмотра и работы через Интернет или интрасеть с данными, хранящимися в базах данных Microsoft Access или базах данных Microsoft SQL Server. Страница доступа к данным может также включать данные из других источников, таких как Microsoft Excel;

- **Макросом** называют набор из одной или более макрокоманд, выполняющих определенные операции, такие как открытие форм или печать отчетов. Макросы могут быть полезны для автоматизации часто выполняемых задач. Например, при нажатии пользователем кнопки можно запустить макрос, который распечатает отчет;

- **Модуль** – это набор описаний и процедур на языке Visual Basic for Application для приложений, собранных в одну программную единицу.


1.2. Создание БД

В Microsoft Access поддерживаются два способа создания БД. Имеется возможность создать пустую БД, а затем добавить в нее таблицы, формы, отчеты и другие объекты. Этот способ является наиболее гибким, но требует отдельного определения каждого элемента БД. Можно также воспользоваться мастером баз данных для создания всех необходимых таблиц, форм и отчетов для БД выбранного типа. Это простейший способ начального создания БД. В обоих случаях созданную БД можно в любое время изменить и расширить.

Этапы создания БД без использования мастера баз данных:

1) при запуске Microsoft Access открывается диалоговое ок-

но, в котором предлагается создать новую БД или открыть существующую. Выберите в этом диалоговом окне переключатель **Новая база данных** и нажмите кнопку **OK**;

2) если БД уже была открыта или если было закрыто диалоговое окно, выводимое при запуске Microsoft Access, нажмите кнопку **Создать**  на панели инструментов и дважды щелкните значок **База данных** на вкладке **Общие**;

3) укажите имя и каталог БД и нажмите кнопку **Создать**.

1.3. Создание таблицы БД

СУБД MS Access позволяет создавать таблицы с помощью мастера или конструктора.

Мастером называется специальное средство, по шагам в диалоговом режиме помогающее пользователю создавать или модифицировать объект или элемент БД.

Конструктор – режим создания или модификации объектов БД. Конструктор имеет свое окно, в котором расположен бланк для задания свойств объекта.

С помощью мастера вы не только создаете таблицу, сколько собираете ее. Существуют два способа создания таблицы без помощи мастера таблиц.

Первый способ – сделать это путем ввода данных или в режиме таблицы. Access выводит на экран пустую таблицу, и вы вводите данные. По мере ввода данных Access просматривает их и определяет типы полей на основе этой информации.

Второй, более совершенный способ, – использование режима конструктора.

Для того чтобы создать новую таблицу при помощи конструктора, необходимо:

1) в левой части окна БД выбрать вкладку **Таблицы**, а затем дважды щелкнуть на элементе **Создание** таблицы в режиме конструктора;

2) ввести имя поля и нажать клавишу **Tab**, чтобы переместить курсор дальше;

3) щелкнуть на кнопке раскрывающегося списка, чтобы просмотреть список всех доступных типов полей. Выбрать нужное поле. Обратите внимание на вкладку в нижней левой части окна. В зависимости от типа данных вашего поля установить нужные параметры;

4) перейти в поле **Описание** и ввести с клавиатуры краткое и четкое описание создаваемого поля. Информация данного поля служит автоматической подсказкой по ходу работы;

5) ввести все поля по аналогии с пунктами 2–4;

6) сохранить таблицу.

Определение необходимых в таблице полей

Поле – элемент таблицы, содержащий определенную информацию, например, фамилию. Термин «поле» применяется как для столбца, так и для ячейки таблицы.

Свойства полей:

- **Размер поля** – до 255 символов для текстового поля;

- **Формат поля** определяет, в каком виде число, набираемое на клавиатуре, выводится на экран;

- **Маска ввода** используется в том случае, если все значения поля должны иметь одинаковый формат. По желанию можно воспользоваться готовой маской или создать новую;

- **Подпись** – второй идентификатор поля. Используется для заголовка столбца в таблице (но не для второго названия поля);

- **Значение по умолчанию** – значение поля, автоматически вводимое в таблицу при добавлении записи;

- **Условие на значение** служит для автоматического контроля значения поля, вводимого в таблицу: оно должно удовлетворять условию, заданному пользователем. Например, **>1981 AND < 1998**. Значение поля должно находиться в рамках между 1981 и 1998. Если условие не выполняется, выводятся сообщение об ошибке и допустимые значения поля;

- **Сообщение об ошибке**. Здесь формулируется свой текст сообщения об ошибке;

- **Обязательное поле** – логическое свойство. При значении **Да** в поле обязательно должны быть введены данные;

- **Пустые строки** – логическое свойство. Значение **Да** указывает, что в поле можно вводить значение нулевой длины, которое обозначается как две кавычки;

- **Индексированное поле** указывает, является ли поле индексом.

В Microsoft Access для связывания сведений, хранящихся в разных таблицах (например, для связывания клиента со всеми его заказами), каждая таблица БД должна содержать поля или набор полей, однозначно определяющих каждую запись. Такое

поле или набор полей называют *первичным ключом*.

Первичный ключ – это одно или несколько полей (в очень редких случаях), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с вторичными ключами в других таблицах.

Если вы создадите таблицу без первичного ключа, Access автоматически напомним об этом и спросит, не хотите ли вы создать ключ. При утвердительном ответе программа создаст поле Счетчик в начале таблицы и определит его в качестве первичного ключа.

Для задания первичного ключа следует:

1) выбрать поле, которое определено в качестве первичного ключа;

2) нажать на панели инструментов кнопку .


Ключевое поле должно быть обязательным и индексированным.

1.4. Установление связей между таблицами

После разбиения сведений на таблицы и определения ключевых полей необходимо выбрать способ, которым Microsoft Access будет вновь объединять связанные сведения. Для этого следует определить связи между таблицами базы данных Microsoft Access.

Для установления связей между таблицами необходимо выполнить следующие действия:

1) из системного меню БД выполнить команду **Сервис** ®

Схема данных или щелкнуть на кнопке , расположенной на панели инструментов;

2) в результате появится окно **Схема данных**;

3) выбрать из системного меню команду **Связи** ® **Добавить таблицу**;

4) на экране появится диалоговое окно **Добавление таблицы** и список таблиц текущей БД;

5) добавить таблицы, которые планируется связать;

6) закрыть диалоговое окно **Добавление таблицы**;

7) выбрать, какие две таблицы нужно связать сначала.

В окне каждой из таблиц прокрутить список полей до тех пор,

пока на экране не появятся оба связующих поля;

8) в Access, прежде чем установить связь, нужно увидеть на экране два связующих поля;

9) поместить курсор мыши на то поле, с которого хотите начать связь и перемещать указатель от одного связующего поля к другому (зажав правую кнопку мыши);

10) указатель примет вид прямоугольника. Когда он окажется напротив связующего поля, отпустить кнопку мыши;

11) в результате появится диалоговое окно **Изменение связей**, детально описывающее устанавливаемую связь (ссылочная целостность, каскадное удаление и обновление);

12) убедитесь в том, что вы правильно выбрали таблицу и имя поля. Если вы уверены в этом, щелкните на кнопке *Создать*, иначе – выберите **Отмена**, и связь создана не будет;

13) проделайте пункты 2–7 нужное количество раз, для того чтобы связать все необходимые таблицы между собой;

14) закройте окно **Схема данных**.

Между таблицами существуют следующие виды связей:

- **Один к одному** – одна запись основной таблицы связана только с одной записью второй таблицы;

- **Один ко многим** – одной записи основной таблицы соответствует несколько записей подчиненной таблицы;

- **Многие к одному** – нескольким записям основной таблицы соответствует одна запись подчиненной таблицы;

- **Многие ко многим** – нескольким записям основной таблицы соответствует несколько записей подчиненной таблицы;

- Не определено – **СУБД не в состоянии определить вид связи между таблицами**.

После того, как определены все поля, в том числе и ключевые, установлены необходимые связи между таблицами, можно наполнять БД информацией – вводить данные в таблицы.

Для этого необходимо перейти в режим таблицы (**Вид** ® **Таблица**).

В данном режиме имеется возможность:

- добавления записей;
- копирования или дублирования записей;
- удаления ненужных записей;
- редактирования записей;
- вставки новых строк;

- поиска информации;

- автоматической замены данных и другие возможности.

2. Практическая часть

Порядок выполнения работы

1. Загрузить СУБД Microsoft Access. Изучить элементы интерфейса пользователя.

2. Создать пустую БД согласно варианту.

3. Создать таблицы в режиме конструктора. Указать атрибуты таблиц и для каждого определить собственные свойства.

4. Определить для каждой таблицы первичный ключ.

5. Установить необходимые связи между таблицами, детально описать установленные связи (ссылочная целостность, каскадное удаление и обновление).

6. Заполнить таблицы (не менее 10 записей в каждую таблицу).

7. Сохранить созданную БД на дискете для дальнейшего использования.

Контрольные вопросы

1. Основные возможности СУБД Microsoft Access.

2. Что такое ссылочная целостность?

3. Типы данных СУБД Microsoft Access.

4. Способы создания объектов БД в СУБД Microsoft Access.