

XXXXXXXX

XX

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

XXXXXXXXXXXXXXXX

**БАЗЫ ДАННЫХ**

**Учебно-методическое пособие**

**Томск ....**

**XXXXXXXXXXXX**

Базы данных: Учебно-методическое пособие. – Томск: Томский государственный университет систем управления и радиоэлектроники, 20XX. – XX с.

© XXXXXXXXXXXX

© ТУСУР, каф. АСУ, 20XX

## СОДЕРЖАНИЕ

	Стр.
1 ВВЕДЕНИЕ .....	4
2 Общие сведения об MS Access.....	4
3 Предварительная подготовка.....	6
4 Описания лабораторных работ .....	7
ЛР № 1 Таблицы и схема данных .....	7
ЛР № 2 Запросы на выборку данных.....	20
ЛР № 3 Формы для просмотра и ввода данных.....	26
ЛР № 4 Отчёты.....	30

## **1 Введение**

Настоящее пособие содержит методические указания к выполнению лабораторных работ по дисциплине «Базы данных». Пособие адресовано студентам кафедры АСУ ТУСУР.

Цель лабораторного практикума – освоить основные приёмы реализации реляционных баз данных и компонентов приложений.

Практикум состоит из четырёх лабораторных работ (ЛР). На весь цикл студент получает одно индивидуальное задание (проект), содержащее описание структуры базы данных и функций приложения пользователя. В ходе выполнения ЛР этот проект должен быть реализован. Работы выполняются в среде реляционной СУБД MS Access. Индивидуальные задания выдаются преподавателем.

Количество аудиторных часов, отведённое для выполнения ЛР, указано в её описании. Для того чтобы уложиться в отведённое время необходимо предварительно изучить соответствующий раздел индивидуального задания и общее описание работы в настоящем пособии.

В конце каждого занятия следует скопировать текущее состояние проекта (.accdb-файл) в папку на сетевом диске, указанную преподавателем. Отчёт о лабораторной работе необходимо сдать преподавателю до начала следующего занятия. Отчёт должен быть оформлен в соответствии с требованиями стандарта «Стандарт ОС ТУСУР 01-2013 по направлениям подготовки и специальностям технического профиля». На титульном листе отчёта обязательно должна быть личная подпись автора и дата представления.

Ниже описаны отдельные лабораторные работы цикла и сформулированы требования к содержанию отчётов.

## **2 Общие сведения об MS Access**

Программа MS Access является приложением ОС Windows и представляет собой полнофункциональную реляционную СУБД. Она реализует среду разработки приложений баз данных и обеспечивает управление данными и приложениями.

Базой данных (БД) в терминологии разработчиков Access является именованный набор объектов, сохраняемых в одном файле <имя БД>. accdb. В настоящем пособии термин «база данных» употребляется исключительно в этом смысле.

Основными объектами базы данных являются таблицы, формы, запросы, отчёты, макросы и модули. Объекты создаются разработчиком БД с помощью соответствующих инструментов визуальной среды разработки – конструкторов, мастеров, строителей и т.п. Определения объектов сохраняются в .accdb-файле вместе с данными пользователя и доступны для просмотра и модификации.

**Таблица** – это единственная структурная единица хранения данных пользователя. Понятие таблицы полностью соответствует понятию отношения реляционной модели данных. Над таблицами определены операции выборки, вставки, модификации и удаления. СУБД выполняет эти операции на уровне *наборов записей* (подмножеств строк таблиц). Кроме этих основных операций выполняются сортировка, фильтрация и печать наборов записей. Обычно база данных содержит несколько таблиц. Таблицы могут быть связаны друг с другом по типу «родитель-потомок». СУБД поддерживает все определённые разработчиком БД связи таблиц и правила ссылочной целостности.

**Форма** представляет собой диалоговое окно, содержащее графические объекты, называемые *элементами управления*. Это поля, кнопки, переключатели и т.п. Обычно форма связана с *источником данных* – базовой таблицей или запросом. В *полях* отображаются значения столбцов из записей источника данных. Другие элементы управления используются для перемещения по записям источника данных, запуска процедур обработки данных и т.п.

**Запрос** – это формализованная запись требования на выполнение некоторого действия над данными в таблицах. Можно создавать запросы на выборку, добавление, модификацию и удаление данных. В результате исполнения запроса на выборку создаётся набор записей из данных, содержащихся в одной или нескольких таблицах. Его можно использовать как источник данных для формы или отчёта. Запросы на обновление данных изменяют состояние *одной* базовой таблицы. Можно также создавать так называемые управляющие запросы, которые используются для создания, удаления или изменения таблиц и индексов в текущей базе данных.

**Отчёт** представляет собой определение формата отображения набора записей, полученного в результате исполнения запроса на выборку.

**Макрос** – последовательность макрокоманд встроенного языка ядра базы данных Microsoft Jet. Макрокоманды выполняют определённые операции, например, открытие формы, печать отчёта, задание нового значения свойства элемента управления и т.п. В современные версии MS Access механизм макрокоманд включается для совместимости с более ранними версиями. Для автоматизации часто решаемых задач обработки данных рекомендуется использовать язык Visual Basic for Application (VBA).

**Модуль** - это набор объявлений и процедур на языке VBA, собранных в одну программную единицу. Каждая процедура в модуле может быть либо процедурой-функцией Function, либо процедурой Sub. Существует два основных типа модулей: модули класса и стандартные модули. Модуль класса связывается с определённым объектом БД – формой или отчётом. Он содержит процедуры, запускаемые в ответ на определённое событие в объекте. Процедуры модуля класса доступны только в том объекте, с которым связан модуль. В стандартных модулях содержатся общие процедуры, не связанные ни с каким объектом, а также часто используемые процедуры, которые могут быть запущены из любого окна базы данных. Локальные модули классов могут вызывать процедуры стандартного модуля.

Более подробные сведения об MS Access Вы получите в процессе выполнения лабораторных работ из справочной системы.

### 3 Предварительная подготовка

1. Создайте новую папку **S:\БД**.
2. В папке **S:\БД** создайте две папки с именами **Лаб** и **КПр**.
3. Папку **S:\БД\КПр** пока оставьте пустой. В ней будут храниться материалы курсового проекта.
4. В папку **S:\БД\Лаб** скопируйте текст настоящего руководства и Ваше индивидуальное задание.
5. Создайте шаблон титульного листа отчёта о лабораторной работе и сохраните его в этой же папке.

Теперь можно начинать.

## 4 Описания лабораторных работ

### ЛР № 1 Таблицы и схема данных

#### **Цели:**

- ознакомиться с интерфейсом СУБД MS Access;
- научиться создавать макеты таблиц и схемы данных.

#### **Порядок выполнения работы**

На выполнение работы выделено 4 часа.

1. При подготовке к занятию **прочитайте** раздел «Общие сведения о таблицах и связях».
2. **Создайте макеты всех таблиц**, определённых в индивидуальном задании в точном соответствии со **спецификациями**.
3. **Определите связи** таблиц и **правила** ссылочной целостности.
4. **Введите** в каждую таблицу **3..5 строк** с пробными данными и **убедитесь** в том, что поддерживаются **все** ограничения, которые указаны в задании.
5. **Сохраните файл** проекта в Вашей папке и скопируйте его в указанный преподавателем раздел сетевого диска.
6. **Напишите отчёт** о лабораторной работе и **сдайте** его преподавателю **до начала** следующего занятия.

#### **Общие сведения о таблицах и связях**

#### Что такое таблица

##### Определение

**Таблицей** в MS Access называется объект базы данных, представляющий собой совокупность строк и столбцов, на пересечении которых размещаются значения простых типов данных.

При этом:

- Все значения столбца принадлежат одному **домéну**, т.е. **имеют один тип данных** и удовлетворяют одному набору ограничений.

- Каждая таблица имеет **уникальное имя** и сохраняется в системном каталоге под определением (**макет**).
- **Макет** можно понимать как **список** имён **столбцов** таблицы (полей) с указанием их свойств.
- Все **значения** данных **сохраняются** в БД Access только как **значения полей таблиц**. Эти таблицы называются **базовыми**.

На рисунке 4.1 можно увидеть окно конструктора таблиц MS Access с указателями на имя таблицы, домен и имя отдельного столбца.

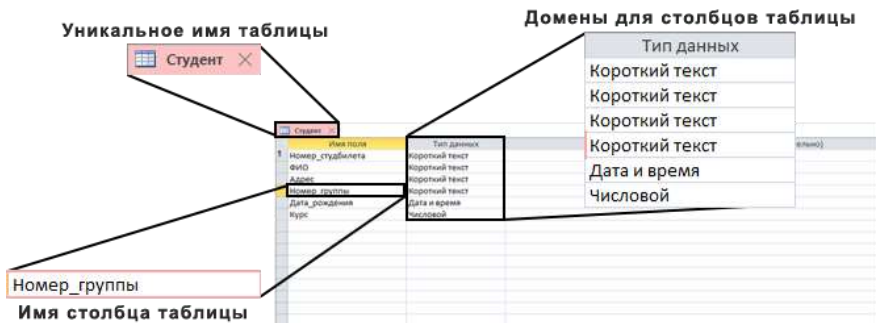


Рисунок 4.1 – Представление таблицы в конструкторе Access

### Для чего предназначена таблица

В таблице сохраняются сведения, относящиеся к **одной** и **только одной** теме (объекту или факту). Например:

- таблица **СТУДЕНТ** содержит **личные** данные студентов,
- таблица **ДИСЦИПЛИНА** содержит сведения об учебных **дисциплинах**, изучаемых студентами,
- таблица **УСПЕВАЕМОСТЬ** содержит сведения об **оценках**, полученных студентами на экзаменах (фактах сдачи экзаменов) и т.п.

На рисунке 4.2 представлен пример таблицы для сохранения данных о поставщике.



Код	Название	Город базис	Телефон	Статус	Информация
4125	ОАО "ТОМСКИЕ ПОСТАВКИ"	Томск	89XXXXXXX	70	Дорого
5621	ООО "Надежные поставки"	Томск	89XXXXXXX	10	Надёжно
5862	ООО "АА"	Алматы	89XXXXXXX	100	Нормально
7865	ОАО "РотДос"	Москва	89XXXXXXX	80	Далеко, дорого, ненадёжно
9565	ООО "ТомскАвтоДост"	Томск	89XXXXXXX	80	Есть скидки

Рисунок 4.2 – Таблица с данными поставщика

### Замечание


Таблица **СТУДЕНТ** **может** содержать поля **Номер\_студбилета**, **Фамилия**, **Имя**, **Отчество**, **Группа** и **не может** содержать поле **Оценка**. Это поле характеризует успеваемость студента и должно быть определено в таблице **УСПЕВАЕМОСТЬ**.

### Что такое первичный ключ таблицы

#### Определение

Для **каждой** таблицы **должен** быть определён **первичный ключ**. **Первичный ключ** – это поле (группа полей), значения которого **идентифицируют строки** таблицы. Первичный ключ может быть **простым** и **составным**.

На рисунке 4.3 представлены примеры простого и составного первичных ключей.



## Первичный ключ

- Идентифицирует строки таблицы
- Каждое значение ключа - **уникально** для данной таблицы
- Не может** принимать значение NULL

#### Простой ключ

Номер студбилета	ФИО	Адрес
43432	Федоров А.А.	Ленина, 23
43742	Васильев Г.Ю.	Пушкина, 10
438417	Иванов И.И.	Пушкина, 10

Любого студента можно однозначно определить по номеру его студенческого билета.

#### Составной ключ

Номер студбилета	Аббревиатура дисциплины	Дата	Оценка
4394317	СНАОД	14.05.2020	отлично
4394317	ООП	17.05.2020	хорошо
434313	СНАОД	23.05.2020	отлично

Результаты сдачи любого экзамена можно однозначно определить по номеру студенческого билета и названию дисциплины.

Рисунок 4.3 – Первичный ключ

Первичный ключ обладает **тремя важными** свойствами:


1. Ни при каких условиях **таблица не может содержать** двух **строк с одинаковыми** значениями **первичного ключа** – **свойство уникальности**.
2. **Составной** первичный ключ **не содержит подмножества** полей, обладающего свойством **уникальности** – **свойство избыточности**.
3. Первичный ключ **не может принимать** неопределённые (**NULL**) значения.

#### Замечание

Первичным ключом **нельзя объявить** произвольное подмножество полей таблицы. **Заключение** о наличии или отсутствии **свойств** первичного ключа у некоторого подмножества полей можно сделать только исходя из **смысла** данных, сохраняемых в таблице.


Примеры правильного и неправильного выбора первичного ключа можно увидеть на рисунках 4.4 и 4.5.

### Примеры



Студент	Номер_студбилета	Номер_груп	ФИО	Адрес
+	434312	434-3	Федоров А.А.	Ленина, 21
+	434311	434-3	Петров Ю.А.	Пушкина, 19
+	434310	434-3	Скворцова Е.В.	Гагарина, 2

Поле "Номер\_группы" не может быть первичным ключом, так его значения повторяются для студентов, зачисленных в одну группу.

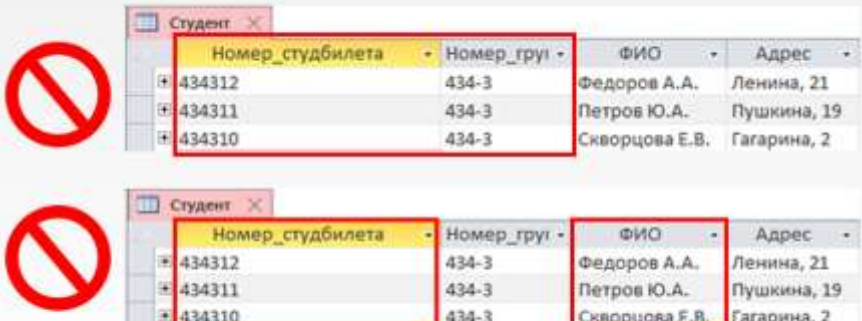


Студент	Номер_студбилета	Номер_груп	ФИО	Адрес
+	434312	434-3	Федоров А.А.	Ленина, 21
+	434311	434-3	Петров Ю.А.	Пушкина, 19
+	434310	434-3	Скворцова Е.В.	Гагарина, 2

Значения поля "Номер\_студбилета" уникальны для всей таблицы "Студент". Это позволяет назначить данное поле первичным ключом таблицы.

Рисунок 4.4 – Пример назначения первичных ключей

## Примеры



Two examples of database tables are shown, each with a red prohibition sign (a circle with a diagonal line) to its left, indicating that the primary key assignment is incorrect.

**Example 1:** The table has columns: Номер\_студбилета, Номер\_группы, ФИО, and Адрес. The primary key is assigned to Номер\_студбилета. This is incorrect because the field 'Номер\_студбилета' is not unique across all rows.

Номер_студбилета	Номер_группы	ФИО	Адрес
434312	434-3	Федоров А.А.	Ленина, 21
434311	434-3	Петров Ю.А.	Пушкина, 19
434310	434-3	Скворцова Е.В.	Гагарина, 2

**Example 2:** The table has columns: Номер\_студбилета, Номер\_группы, ФИО, and Адрес. The primary key is assigned to Номер\_группы. This is incorrect because the field 'Номер\_группы' is not unique across all rows.

Номер_студбилета	Номер_группы	ФИО	Адрес
434312	434-3	Федоров А.А.	Ленина, 21
434311	434-3	Петров Ю.А.	Пушкина, 19
434310	434-3	Скворцова Е.В.	Гагарина, 2

Пары полей "Номер\_студбилета", "Номер\_группы" и "Номер\_студбилета", "ФИО" нельзя назначить первичными ключами, так как поле "Номер\_студбилета" в каждой паре само обладает свойством уникальности.

Рисунок 4.5 – Пример назначения составных первичных ключей

### Для чего нужен первичный ключ

Поскольку значения первичных ключей идентифицируют строки таблицы, их можно использовать:

- Для быстрого поиска отдельных строк.
- Для организации связей таблиц.

#### Замечание

Для ускорения поиска записей Access создаёт *уникальные индексы* по первичным ключам таблиц. Они хранят информацию о размещении отдельных строк таблиц в физической базе данных.

## Связи таблиц и внешние ключи

### Определение

Для организации связей в состав полей таблицы-потомка включается дубликат первичного ключа родителя. Он называется здесь **внешним ключом**.

Значения внешнего ключа – это ссылки на строки родительской таблицы, содержащие такие же значения первичного ключа. Ссылки реализуются как неуникальные индексы по внешнему ключу. Эти индексы хранят информацию о связи, т.е. соответствии строк различных таблиц.

На рисунке 4.6 представлена схема данных с таблицами «Студент» и «Успеваемость». Линия с утолщёнными концами обозначает связь между таблицами, или соответствие определённых строк этих таблиц.

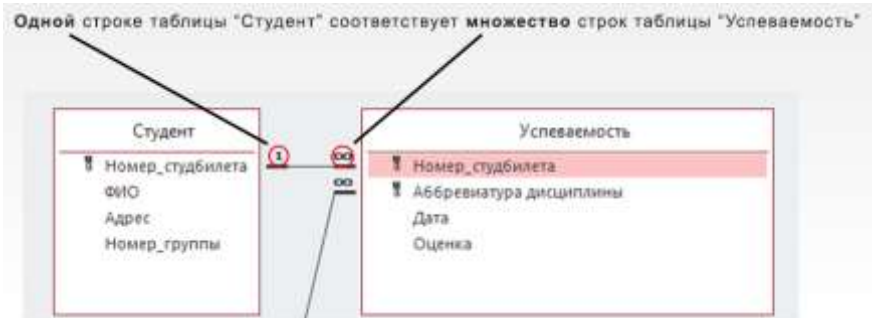


Рисунок 4.6 – Связь двух таблиц

На рисунке 4.7 показаны строки из двух таблиц – «Студент» и «Успеваемость». Видно, что поле «Номер\_студбилета» таблицы «Успеваемость» ссылается на такое же поле родительской таблицы.

Таблица родитель - "Студент"			
Номер_студбилета	Номер_группы	ФИО	Адрес
434312	434-3	Федоров А.А.	Ленина, 21

Таблица потомок - "Успеваемость"			
Номер_студбилета	Аббревиатура дисциплины	Дата	Оценка
434312	ООП	15.01.2020	хорошо
434312	СИАОД	23.01.2020	отлично

Рисунок 4.7 – Строки из таблиц «Студент» и «Успеваемость»

Для того чтобы **установить** соответствие **строк** родителя (таблица «Студент») и потомка (таблица «Успеваемость»), достаточно **включить** в схему потомка столбец «**Номер\_студбилета**».

### Ссылочная целостность

#### Определение

**Значения внешнего ключа** в строках таблицы-потомка **не могут** быть произвольными. Они **должны** принадлежать множеству значений **родительского ключа** в существующих строках родительской таблицы. Это условие называется **требованием ссылочной целостности**.

Далее на рисунке 4.8 продемонстрирован пример нарушения ссылочной целостности. Строка таблицы «Успеваемость» ссылается на несуществующий номер студенческого билета.

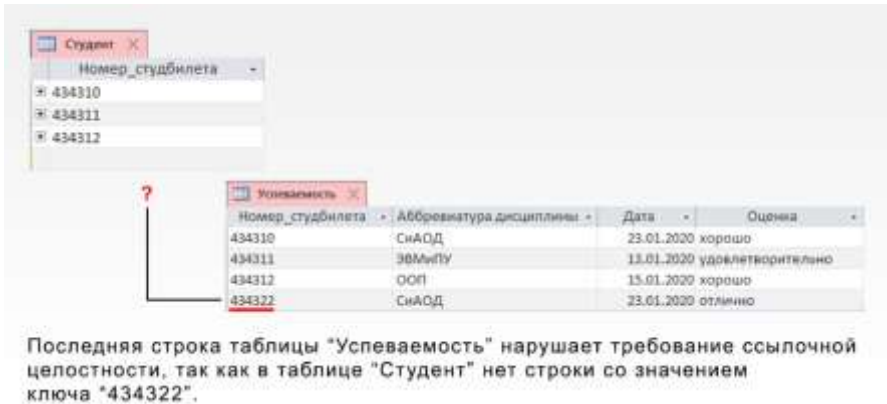


Рисунок 4.8 – Нарушение ссылочной целостности

### Правила ссылочной целостности

#### Замечание

Одна из **главных** задач СУБД – **поддерживать** ссылочную **целостность** при выполнении операций **обновления данных**. Она будет это делать, если для **каждой** пары родитель-потомок **определены правила** совместного обновления множеств значений первичного и внешнего ключей.

Ссылочная целостность может быть нарушена в следующих случаях:

- При попытке **добавления строки** в таблицу-потомок.
- При попытке **изменения значения** внешнего ключа в существующей строке **таблицы-потомка**.
- При попытке **удаления строки** из родительской таблицы.
- При попытке **изменения значения** первичного ключа в существующей строке **родительской таблицы**.

В **двух первых** случаях СУБД **должна проверить**, принадлежит ли новое значение внешнего ключа множеству существующих зна-

чений родительского ключа. Если **да**, то обновление таблицы-потомка **будет выполнено, иначе – отвергнуто**.

В **двух последних** случаях **вариантов** правил обновления гораздо **больше**. Они определяются требованиями пользователя и **должны быть реализованы разработчиком** БД в виде специальных программ – **триггеров ссылочной целостности**.

#### Замечание

Однако любая реляционная СУБД имеет **набор** стандартных **правил** ссылочной целостности, которые **поддерживаются ядром**. В частности, MS Access **поддерживает**:

- правила **отложенного** (Restricted) обновления/удаления – что означает **запрет** обновления/удаления значения **родительского** ключа, если на него **есть ссылка** в таблице-потомке;
- правила **каскадного** (Cascaded) обновления/удаления – что означает **автоматическое** распространение операции над значением **родительского** ключа на те **строки потомка**, которые на него **ссылаются**.

### ***Начало работы***

Запустите MS Access. Система предложит создать новую БД или открыть существующую.

Выберите пункт *Новая база данных* и щелкните по кнопке *ОК*. В открывшемся окне выберите папку S:\БД\Лаб, задайте имя Вашей базы данных (проекта) и щёлкните по кнопке *Создать*.

Перед Вами открылось окно базы данных. Вы получили возможность создавать объекты Вашего проекта – таблицы, запросы, формы, отчёты и т.п. Начать следует с создания таблиц.

В любой момент Вы можете прекратить работу, выполнив пункт меню *Файл/Закреть*. Текущее состояние проекта будет сохранено автоматически. Для возобновления работы над проектом достаточно щёлкнуть левой кнопкой мыши по имени файла проекта в папке S:\БД\Лаб.

### **Создание таблиц**

Создать макет базовой таблицы можно через меню *Вставка/Таблица...*, либо выбрав в окне БД вкладку *Таблицы* и нажав на кнопку *Создать*. В обоих случаях Вы получите окно выбора способа создания новой таблицы. Этих способов пять. Последние два используются для создания связей с внешними базами данных MS Access. Пока не обращайтесь на них внимания.

В *Режиме таблицы* можно ввести несколько строк непосредственно в пустую таблицу. При сохранении новой таблицы данные анализируются, и каждому полю присваивается необходимый тип данных и формат. Никакие другие свойства полей в этом режиме не могут быть заданы.

Используя *Конструктор*, можно определить все параметры макета таблицы явно. Независимо от способа, примененного для создания таблицы, всегда имеется возможность использовать *Конструктор* для редактирования существующего макета таблицы.

С помощью *Мастера* можно выбрать подходящий образец таблицы из предлагаемого системой списка и/или поля создаваемой таблицы из множества полей предлагаемых таблиц.

Ниже подробно описаны упомянутые способы создания таблиц. Используйте их все в своей работе. При возникновении проблем обращайтесь к справочной системе MS Access, раздел «*Создание таблиц*».

Закончив описание структуры таблицы, введите в неё несколько пробных записей, чтобы убедиться в том, что их формат соответствует Вашему замыслу. Особенно не усердствуйте, так как их, скорее всего, придётся удалить при редактировании макета или при создании связей таблиц.

#### **Задание 1.1 Создание таблицы путем ввода данных**

– Переключитесь в окно базы данных. Для переключения из другого окна в окно базы данных нажмите клавишу *F11*.

– На вкладке *Таблица* нажмите кнопку *Создать*.

– Дважды щелкните левой кнопкой мыши по элементу *Режим таблицы*. На экране появится пустая таблица, состоящая из двадцати столбцов и тридцати строк.



– По умолчанию столбцам присвоены имена *Поле1*, *Поле2* и т. д. Замените их осмысленными. Для переименования столбца дважды щелкните по его названию и введите имя.

– Введите несколько строк в таблицу. В каждый столбец вводите данные определенного типа. Например, в первый – номера студенческих билетов, а во второй – фамилии студентов. Все пустые столбцы с неизмененными именами будут удалены при сохранении таблицы.

**Замечание 1.1.** При создании макета в режиме таблицы тип данных каждого поля определяется типом введенных значений. Автоматически устанавливаются свойства поля этого типа, принятые в MS Access по умолчанию. Поэтому до наполнения реальными данными созданный макет таблицы следует отредактировать в режиме конструктора.

– Закончив ввод данных во все нужные столбцы, нажмите кнопку *Сохранить* на панели инструментов, чтобы сохранить таблицу. Прежде чем выполнить операцию, система попросит у Вас разрешения на создание поля первичного ключа. Лучше отказать ей по причинам, изложенным ниже. Первичный ключ следует определить в режиме конструктора.

Состав столбцов таблицы и некоторые их свойства можно изменить в режиме таблицы. Щёлкните правой кнопкой мыши по имени столбца. Появится всплывающее меню. Изучите описания его команд в справочной системе, попытайтесь их исполнить и посмотрите, что получится.

### **Задание 1.2. Создание таблицы в режиме конструктора**

- Переключитесь в окно базы данных.
- На вкладке *Таблица* нажмите кнопку *Создать*.
- Дважды щелкните элемент *Конструктор*.

Перед Вами бланк конструктора таблицы. Строки верхней части бланка предназначены для описания полей создаваемой таблицы. Вкладки в левой нижней части используются для описания свойств полей. В правой нижней части выводится контекстная подсказка.

– Введите имя поля и выберите тип данных. Желательно также описать смысл поля. Это облегчит Вам дальнейшую работу.

– В открывшейся вкладке *Общие* задайте свойства поля. Состав свойств зависит от типа данных поля. Для вызова контекстной справки о свойстве установите указатель в соответствующую строку вкладки и нажмите *F1*.

– Повторите эти действия нужное число раз. Работая с *Конструктором*, обращайте внимание на подсказки в правой нижней части бланка.

– Определите первичный ключ таблицы. Процедура определения описана в разделе справочной системы MS Access «*Ключ по одному полю*».

**Замечание 1.2.** Обязательно определяйте первичный ключ каждой таблицы. В противном случае при сохранении таблицы выдается вопрос, нужно ли его создавать. Если Вы согласитесь, то система автоматически создаст ключевое поле, но

а) Вы не сможете управлять его значениями и

б) эти значения не будут иметь смысла в предметной области приложения.

Будет создан так называемый *суррогатный ключ*, который используется системой исключительно для быстрого поиска записей и определения связей. Требование целостности сущности поддерживаться фактически не будет.

– Для сохранения таблицы нажмите кнопку *Сохранить* на панели инструментов, а затем введите имя таблицы.

– Откройте созданную таблицу в режиме таблицы и введите несколько пробных строк. Убедитесь в том, что созданный макет соответствует Вашему замыслу.

### **Создание альтернативных ключей**

Таблица может содержать *альтернативный ключ (АК)*, т.е., поле (группу полей), значения которого (которой) не могут повторяться в различных строках. Чтобы определить альтернативный ключ, нужно создать уникальный индекс.

Если АК простой, то для этого достаточно открыть таблицу в режиме конструктора, выделить поле АК и во вкладке *Общие* установить свойство *Индексированное поле* = *Да (Совпадения не допускаются)*.

Если АК составной, то нужно щёлкнуть по рамке окна конструктора правой кнопкой мыши и выполнить команду *Индексы*. В

открывшемся окне Вы увидите все индексы, определённые для текущей таблицы. Введите в пустой строке имя создаваемого индекса в столбец *Индекс*, а затем в этой же и последующих строках выберите из выпадающего списка имена полей компонентов индекса. Затем определите свойства нового индекса.

### **Использование подстановок**

Таблица может содержать поле, значения которого должны выбираться из некоторого конечного множества. Например, значения внешнего ключа должны выбираться из множества существующих значений родительского ключа. В подобных случаях рекомендуется связать с полем *элемент управления*, который будет использоваться для отображения данных.

- Откройте таблицу в режиме Конструктора.
- Выберите нужное поле и откройте вкладку *Подстановка*.
- Из выпадающего списка выберите тип элемента управления

*Список* или *Поле со списком*.

Тип *Список* используется для подстановки значений из постоянного списка, например, из списка должностей.

Тип *Поле со списком* используется, если список подстановок зависит от состояния БД, например, если это список значений родительского ключа.

Для получения более подробных сведений обратитесь к разделу *Свойство «Тип элемента управления» (DisplayControl)* справочной системы.

### **Задание 1.3. Создание таблицы при помощи мастера таблиц**

- Переключитесь в окно базы данных.
- На вкладке *Таблица* нажмите кнопку *Создать*.
- Дважды щелкните элемент *Мастер таблиц*.

Следуйте инструкциям, выдаваемым в диалоговых окнах *Мастера*.

**Замечание 1.3.** При создании макета с помощью *Мастера* поля новой таблицы наследуют свойства полей использованных образцов таблиц. Просмотрите созданный макет таблицы в режиме *Конструктора* и, если нужно, отредактируйте его.

### **Задание 1.4. Определение связей между таблицами**

Создавая таблицы, Вы ввели в некоторые из них поля имеющие смысл внешних ключей. Однако редактор свойств таблиц не содержит средств определения внешних ключей, поэтому системе ничего о них не известно. Для того чтобы определить связи таблиц, выполните пункт меню *Сервис/Схема данных...* или щелкните по иконке схемы на инструментальной панели. Находясь в окне схемы, вызовите справочную систему MS Access, и изучите раздел *«Работа со схемой данных»*. Затем создайте связи таблиц и определите разумные правила ссылочной целостности.

### **Требования к содержанию отчёта**

Отчёт о ЛР1 должен содержать следующие разделы.

- 1 Цель лабораторной работы.
- 2 Описание таблиц БД из индивидуального задания.
- 3 Описание процесса реализации таблиц и схемы.
- 4 Выводы.

В разделе 3 следует привести описание использованных Вами средств MS Access и наиболее рациональной (по Вашему мнению!) последовательности действий, направленных на реализацию схемы БД. Здесь же должен быть приведён снимок схемы данных и перечень установленных правил ссылочной целостности с их обоснованиями.

В разделе 4 перечислите то, чему Вы научились в ходе выполнения работы.

## **ЛР № 2 Запросы на выборку данных**

**Ц е л ь :** научиться создавать и редактировать запросы на выборку данных с помощью построителя запросов QBE.

### **Порядок выполнения работы**

На выполнение работы выделено 4 часа.

– При подготовке к занятию прочтите раздел «Общие сведения о запросах на выборку данных».

– Создайте определения всех запросов, указанных в индивидуальном задании в точном соответствии со спецификациями.

– Сохраните новое состояние проекта (под старым именем!!!) в Вашей папке и скопируйте его в указанный преподавателем раздел сетевого диска.

– Напишите отчёт о лабораторной работе и сдайте его преподавателю до начала следующего занятия.

### ***Общие сведения о запросах на выборку данных***

Что такое запрос. Запрос – это адресованное СУБД требование пользователя (приложения) на выполнение некоторых действий над БД. Для реляционных СУБД эти требования формулируются на языке SQL. По характеру действий выделяют три категории запросов: на выборку данных, на обновление данных, на редактирование схемы БД. MS ACCESS имеет средства создания и обработки запросов всех этих категорий. Здесь Вы должны освоить только работу с запросами на выборку.

Что такое запрос на выборку. Запрос на выборку (далее запрос) требует извлечения из БД некоторого подмножества значений хранимых данных из одной или нескольких таблиц. В результате обработки запроса в рабочем буфере СУБД создаётся неименованная временная таблица. Она содержит значения хранимых данных, удовлетворяющие условиям запроса. Эта таблица отражает текущее состояние БД, и существует до тех пор, пока она нужна приложению, издавшему запрос.

Для чего нужны запросы. С помощью запросов можно визуально просматривать и анализировать данные из нескольких таблиц. Однако в основном они используются в качестве источников данных для форм, отчётов и прикладных программ.

Как выглядит запрос. Для MS ACCESS запрос – это объект базы данных, содержащий текст SQL-оператора выборки данных. Оператор выглядит как структурированная фраза на английском языке. Она содержит несколько стандартных ключевых слов, между которыми автор запроса может вписать свои слова по определённым правилам. Общий вид этой фразы (оператора выборки) таков.

```
SELECT <целевой список запроса>
FROM <источник данных>
WHERE <условие отбора строк источника данных>
GROUP BY <список полей группирования строк>
HAVING <условие отбора групп строк>
ORDER BY <список полей сортировки результата выборки>;
```

### Примеры запросов.

*Получить упорядоченный по фамилиям список студентов группы 123:*

```
SELECT СТУДЕНТ.Фамилия
FROM СТУДЕНТ
WHERE СТУДЕНТ.Группа = 123
ORDER BY СТУДЕНТ.Фамилия;
```

Это читается так. «Из таблицы СТУДЕНТ выбрать строки, содержащие значение 123 в столбце Группа. Поместить в выходную таблицу значения столбца Фамилия из этого множества строк. Упорядочить строки выходной таблицы по значениям столбца Фамилия.»

Результат может выглядеть так

Фамилия
Бабкин
Вабкин
Габкин
Дабкин

*Для каждой группы получить число студентов. Результат упорядочить по численности групп.*

```
SELECT СТУДЕНТ.Группа, Count(*) AS Численность
FROM СТУДЕНТ
GROUP BY СТУДЕНТ.Группа
ORDER BY Численность;
```

“Строки таблицы СТУДЕНТ сгруппировать по значениям столбца Группа. Подсчитать число строк в каждой группе строк (функция Count(\*)) и поместить вместе со значением столбца Группа в выходную таблицу. Упорядочить строки выходной таблицы по значениям столбца Численность.”

Результат может выглядеть так

Группа	Численность
123	13
120	15
121	19

Временные и сохраняемые запросы. Запрос может создаваться для однократного или многократного использования. В первом случае определение запроса не сохраняется в схеме базы данных. Однако определение можно сохранить как постоянный объект схемы, присвоив ему уникальное имя. Такой запрос называ-

ется *представлением* (View). На представление можно ссылаться в прикладных программах и определениях запросов, форм и отчётов как на таблицу. Однако, в отличие от таблицы, представление *не содержит собственных данных*. Представление может быть активировано многократно. Если между двумя последовательными активациями содержимое базовых таблиц изменилось, то результаты выборки будут различными. Другими словами, представление *всегда отображает текущее состояние базовых таблиц*.

### ***Начало работы***

Формулировки запросов на русском языке приведены в индивидуальном задании.

Загрузите Windows и запустите MS Access. Откройте проект, созданный в предыдущем сеансе. В открывшемся окне щёлкните по вкладке *Запросы*, а затем по кнопке *Создать*. Перед Вами список режимов работы построителя запросов MS Access.

*Конструктор* – универсальное средство, обеспечивающее создание запросов любой сложности. Он может работать в режимах табличного построителя запросов (QBE) и редактора текстов запросов на языке SQL.

*Простой запрос* – мастер, автоматически создающий запросы для получения данных из полей одной или нескольких таблиц или запросов. С его помощью можно также вычислять суммы или средние арифметические значений столбца для всех строк или определенных групп строк, находить максимальное и минимальное значение в столбце, подсчитывать число строк в таблице или определённой группе строк. Однако нельзя ограничить количество записей, возвращаемых этим запросом, с помощью условий отбора.

*Перекрёстный запрос*, *Повторяющиеся записи* и *Записи без подчинённых* – это также мастера запросов особых типов. Вы будете использовать в дальнейшем мастер перекрёстных запросов для создания отчёта специального вида.

Запрос, созданный с помощью любого мастера, можно отредактировать и настроить в режиме конструктора. Например, можно добавить условия отбора в результат работы мастера простого запроса.

Выполняя задание, пользуйтесь разделом справочной системы MS Access *Создание, изменение или просмотр результатов запроса*.

### ***Создание простого запроса с помощью мастера***

– В окне базы данных выберите вкладку *Запросы* и нажмите кнопку *Создать*.

– В диалоговом окне *Новый запрос* выберите *Простой запрос* и нажмите кнопку *ОК*.

– Укажите имя таблицы или запроса, на котором должен быть основан создаваемый запрос, а затем выберите поля, данные которых нужно использовать.

– Если необходимо, укажите дополнительную таблицу или запрос и выберите нужные поля. Повторяйте этот шаг до тех пор, пока не будут выбраны все необходимые поля.

Следуйте инструкциям диалоговых окон мастера. Последнее диалоговое окно позволяет либо запустить запрос, либо открыть его в режиме конструктора.

### ***Использование QBE для создания и редактирования запросов.***

– В диалоговом окне *Новый запрос* выберите *Конструктор* и нажмите кнопку *ОК*.

– В диалоговом окне *Добавление таблицы* выберите вкладку, содержащую нужные объекты – таблицы, запросы или и то, и другое.

– Выберите имя объекта, добавляемого в запрос. Для одновременного выбора нескольких объектов выбирайте объекты при нажатой клавише *CTRL*. Чтобы выбрать группу соседних объектов, выберите имя первого объекта, нажмите клавишу *SHIFT* и, удерживая ее, выберите имя последнего объекта группы.

– Нажмите кнопку *Добавить*, а затем кнопку *Заккрыть*. Можно также добавить объект в запрос путём перетаскивания его имени из окна базы данных в верхнюю половину окна запроса в режиме конструктора.

**Замечание 2.1.** Если все выбранные объекты являются базовыми таблицами, соединёнными в схеме данных по типу «родитель – потомок», то по умолчанию источником данных запроса считается *естественное соединение* этих таблиц. Если же среди выбранных



объектов имеется запрос, то нужно определить его соединения с другими объектами. Для этого захватите мышью имя родительского поля в верхней части окна построителя запросов и перетащите его на имя поля-потомка. Затем дважды щёлкните по линии связи и выберите тип соединения.

Теперь нужно выбрать поля источника данных, которые будут использоваться в целевом списке запроса или в условиях отбора. Для этого используется таблица в нижней половине окна построителя запросов.

- Установите указатель в левый верхний угол таблицы.
- Разверните список доступных полей источника данных и выберите нужное поле
- Переместите указатель в соседний столбец. Повторяйте шаги 6, 7 до тех пор, пока не выберете все нужные поля.
- В строке *Вывод на экран* таблицы погасите флажки под именами полей, не входящих в список выбора запроса.
- Прочитайте раздел справки *«Добавление или изменение условия отбора»* и задайте ограничения на значения выбранных полей источника, которые должны удовлетворяться в отбираемых запросом строках.
- Откройте созданный запрос в режиме SQL и переведите текст на русский язык.
- Исполните запрос и убедитесь в том, что результат соответствует спецификации.

### ***Требования к содержанию отчёта***

Отчёт о ЛР2 должен содержать следующие разделы.

- 1 Цель лабораторной работы.
- 2 Описание работы
  - формулировки запросов из индивидуального задания;
  - тексты запросов на SQL с интерпретацией на русском языке;
- 3 Выводы.

### **ЛР № 3 Формы для просмотра и ввода данных**

**Ц е л ь :** Научиться создавать экранные формы и обработчики событий.

#### ***Порядок выполнения работы***

На выполнение работы выделено 8 часов.

– При подготовке к занятию прочтите раздел «*Общие сведения о формах*».

– Создайте макеты всех форм, указанных в индивидуальном задании, в точном соответствии со спецификациями.

– Сохраните новое состояние проекта (под старым именем!!!) в Вашей папке и скопируйте его в указанный преподавателем раздел сетевого диска.

– Напишите отчёт о лабораторной работе и сдайте его преподавателю до начала следующего занятия.

#### ***Общие сведения о формах***

Что такое форма. Формой в MS Access называется диалоговое окно, содержащее графические объекты, называемые *элементами управления*. Формы обеспечивают интерфейс конечного пользователя и приложения. Они принимают действия пользователя, отображают текущее состояние источника данных формы, сообщения приложения и т.д.

Что такое источник данных формы. Это объект базы данных (базовая таблица или запрос), для обработки которого предназначена форма. При загрузке формы создаётся временный набор записей (объект *Recordset*), сформированный из записей источника данных. Значения полей этих записей отображаются в форме. Если источником данных является базовая таблица, то пользователь может изменять эти значения, удалять записи, добавлять новые. Все обновления данных происходят не в базовой таблице, а в связанном с формой объекте *Recordset*. Если обновлённые данные удовлетворяют ограничениям целостности, то они сохраняются в базовой таблице. В противном случае обновления базовой таблицы не происходит.

Форма может не иметь источника данных, если она предназначена для управления приложением.

Что такое элемент управления. Это графический объект, предназначенный для ввода/отображения значений хранимых данных, запуска процедуры, задания режима исполнения приложения и т.п. Каждый элемент управления имеет уникальное имя и набор свойств. На имя можно ссылаться в процедурах обработки данных и макросах. Свойства можно изменять как вручную, так и программно. С любым элементом управления (и с формой как целым) могут быть связаны обработчики событий. Существует три основных типа элементов управления: поле, подчинённая форма и кнопка.

Что такое поле. Это графический объект, в котором отображаются значения определённого типа данных. Поле может быть *связанным* или *свободным*. В связанном поле отображаются значения столбца источника данных формы. Свободное поле – это свободная переменная, используемая в процедурах обработки данных. Значения этой переменной могут вводиться пользователем с клавиатуры или отображать результаты вычислений и т.п. С полем может быть связан набор подстановочных значений (список, раскрывающийся список или поле со списком). В этом случае пользователь может выбирать нужное значение поля, а не вводить его с клавиатуры.

Что такое подчинённая форма. Это форма, вставленная в другую форму. Первичная форма называется *главной* формой. Подчинённые формы используются для вывода данных из таблиц или запросов, связанных с источником данных главной формы отношением «один-ко-многим». В подчинённой форме выводятся только те записи, которые связаны с текущей записью главной формы. Каждая новая запись, вводимая в подчинённую форму, автоматически связывается с текущей записью главной формы. Тем самым обеспечивается ссылочная целостность при обновлении связанных таблиц.

Что такое кнопка. Кнопки в форме используются для выполнения действия или набора действий. Например, можно создать кнопку, изменяющую свойства формы или её элементов управления, открывающую другую форму, запускающую процедуру обработки данных и т.п. Чтобы определить действие для кнопки в

форме, следует написать *обработчик события* и вложить его в свойство кнопки *Нажатие кнопки (OnClick)*.

Что такое событие. Событием называют определенное действие, возникающее при работе с конкретным объектом. Например, нажатие кнопки мыши, обновление данных, открытие или закрытие формы, получение или потеря фокуса элементом управления и т.д. Обычно события возникают в результате действий пользователя.

Что такое обработчик события. Это макрос или программная процедура, автоматически запускаемая при наступлении события. Пользователь может написать собственные обработчики событий и связать их с элементами управления формы. Например, если нужно, чтобы после ввода значения в некоторое поле П1 автоматически перевычислялось значение поля П2, то можно связать процедуру перевычисления со свойством *После обновления* поля П1.

### ***Начало работы***

Спецификации форм, которые Вы должны создать, приведены в индивидуальном задании. Чтобы создать форму, откройте в окне базы данных закладку *Форма* и нажмите кнопку *Создать*. Перед Вами окно создания новой формы. Здесь нужно выбрать один из режимов создания формы и указать источник данных, т.е. таблицу или запрос, значения полей которого будут отображаться в полях формы.

Мастера *Диаграмма* и *Сводная таблица* предназначены для создания специальных форм просмотра данных. Здесь они нам не понадобятся.

Другие мастера позволяют создавать формы просмотра и редактирования полей таблиц на основе шаблонов. Конструктор обеспечивает создание нешаблонных форм и редактирование форм, созданных мастерами.

**Замечание 3.1.** Форма не содержит собственных данных. *поля формы связаны с полями таблиц* – базовых или виртуальных, – и все значения данных, отображающиеся в полях формы, *выбираются из таблиц*. Значения, введенные в поля формы и удовлетворяющие ограничениям целостности данных, автоматически сохраняются как значения соответствующих полей (базовых) таблиц.

Ваша задача сейчас – научиться создавать простые формы. Используйте для этого мастера. Созданную с помощью мастера форму всегда можно отредактировать с помощью *Конструктора*, изменив дизайн, добавив или удалив элементы управления, изменив их свойства и т.п.

**Задание 3.1.** Создать форму для просмотра, добавления, удаления и обновления записей указанной таблицы.

**Задание 3.2.** Создать форму, обеспечивающую *только просмотр* записей;

Для того чтобы выполнить эти задания, с помощью какого-либо мастера создайте макет формы. Затем, предварительно сохранив, откройте его в режиме *Конструктора*. Щёлкните правой кнопкой мыши на черном квадратике в левом верхнем углу открывшегося окна и выберите в появившемся всплывающем меню опцию *Свойства*. В открывшемся окне изучите закладку *Данные*. Чтобы получить контекстную справку о свойстве, установите указатель в соответствующую строку и нажмите *F1*.

**Задание 3.3.** Создать управляемую форму в соответствии со спецификациями, содержащимися в индивидуальном задании.

Во вкладке *Формы* окна базы данных выберите форму, созданную в предыдущем задании и скопируйте её в это же окно, сохранив под другим именем. Откройте копию в режиме *Конструктора* и добавьте нужные кнопки и обработчики событий.

**Задание 3.4.** Создать форму, содержащую подчинённую форму в соответствии со спецификациями, содержащимися в индивидуальном задании.

Создайте макет формы мастером, выбрав поля из двух таблиц. Откройте макет конструктором и добавьте в него необходимые элементы управления и обработчики событий.

**Рекомендация.** Для того чтобы изучить некоторые способы организации диалога с помощью форм, выполните действие *Файл/Создать базу данных...* В открывшемся окне выберите закладку *Базы данных* и щёлкните по любой иконке в появившемся окне. Далее следуйте указаниям системы. Будет сгенерирован выбранный Вами проект – база данных и приложение. Исследуйте формы этого проекта в режиме исполнения и в режиме *Конструктора*. Используйте систему помощи MS Access.

### ***Требования к содержанию отчёта***

Отчёт по ЛР3 должен содержать следующие разделы.

- 1 Цель лабораторной работы.
- 2 Описание требований к формам.
- 3 Описание процесса реализации форм.
- 4 Выводы.

В разделе 3 следует привести снимки созданных форм и описание использованных Вами средств MS Access.

В разделе 4 перечислите то, чему Вы научились в ходе выполнения работы.

### **ЛР № 4 Отчёты.**

**Ц е л ь :** Научиться создавать отчёты в среде MS Access.

#### ***Порядок выполнения работы***

На выполнение работы выделено 4 часа.

– При подготовке к занятию прочтите раздел «*Общие сведения об отчётах*».

– Создайте шаблоны отчётов, указанных в индивидуальном задании, в точном соответствии со спецификациями.

– Сохраните новое состояние проекта (под старым именем!!!) в Вашей папке и скопируйте его в указанный преподавателем раздел сетевого диска.

– Напишите отчёт о лабораторной работе и сдайте его преподавателю.

#### ***Общие сведения об отчётах***

Что такое отчёт. Отчётом в MS Access называется результат исполнения запроса на выборку данных, отображённый в предопределённом формате. С точки зрения пользователя отчёт представляет собой готовый к публикации документ, оформленный в соответствии с принятыми правилами.

Элементы управления отчёта. В шаблоне отчёта используются следующие элементы управления: поле, надпись, подчинённый отчёт, графический элемент оформления (линия, рисунок и т.п.).

Поле может быть связанным или свободным. В связанных полях отображаются значения соответствующих полей источника данных. Свободное поле – это свободная переменная. Она создаётся

разработчиком и используется в какой-либо процедуре обработки данных, связанной с отчётом.

Что такое подчинённый отчёт. Это отчёт, вставленный в другой отчёт. Чаще всего подчинённый отчёт используют для отображения результатов агрегированной обработки данных совместно с результатами выборки индивидуальных значений.

Более подробные сведения об отчётах можно получить из раздела справки «Создание, изменение и печать отчетов».

### ***Начало работы***

Чтобы создать отчёт, откройте в окне базы данных закладку *Отчёт* и нажмите кнопку *Создать*. Перед Вами окно создания нового отчёта. Здесь нужно выбрать один из режимов создания отчёта и указать источник данных, т.е. таблицы или запросы, содержащие данные, которые должны быть отображены в отчёте. Для того чтобы «пощупать» средства создания отчётов, возьмите любую базовую таблицу и создайте на её основе простейший отчёт *Мастером* и *Конструктором*.

### ***Создание отчётов***

Спецификации отчётов, которые Вы должны создать, приведены в индивидуальном задании. Удобнее всего создавать отчёт на базе одиночного запроса или таблицы.

***Задание 4.1.*** Создать простой отчёт, содержащий агрегированные данные.

- Создайте один или несколько запросов, возвращающих значения данных, которые должны отображаться в полях отчёта.
- Запустите *Мастер* и создайте «заготовку» отчёта.
- Откройте «заготовку» *Конструктором* и приведите формат в соответствие со спецификациями индивидуального задания.

***Задание 4.2.*** Создать перекрёстный отчёт.

Для этого нужно выполнить следующие действия.

- Создать запрос на выборку элементов данных, которые отображаются в перекрёстном отчёте.
- Средствами соответствующего мастера создать на базе этого запроса перекрёстный запрос.
- Использовать перекрёстный запрос как источник данных для отчёта.

***Требования к содержанию отчёта***

Отчёт по ЛР4 должен содержать следующие разделы.

- 1 Цель лабораторной работы.
- 2 Описание требований к отчётам.
- 3 Описание процесса реализации отчётов.
- 4 Выводы.

В разделе 3 следует привести распечатки созданных отчётов и описание использованных Вами средств MS Access.

В разделе 4 перечислите то, чему Вы научились в ходе выполнения работы.