

## Практическая работа №1

### СОЗДАНИЕ И ОСНОВНЫЕ ПРИЕМЫ РЕДАКТИРОВАНИЯ ТАБЛИЦ

**Название работы:** Построение информационных технологий производственных процессов. Создание базы данных (БД) с помощью Мастера, ее редактирование и внесение данных.

#### Цель работы:

1. использовать технологии сбора, размещения, хранения, накопления, преобразования и передачи данных в профессионально ориентированных информационных системах;
2. использовать в профессиональной деятельности различные виды программного обеспечения, в т.ч. специального;

#### Общие сведения

#### 1. Основные принципы проектирования баз данных

##### 1.1 Основные определения

Современные информационные системы характеризуются большими объемами хранимых данных, их сложной организацией, а также высокими требованиями к скорости и эффективности обработки этих данных. Это становится возможным при использовании специальных программных средств - систем управления базами данных (СУБД).

**База данных (БД)** - это поименованная совокупность данных, относящихся к определенной предметной области.

**Система управления базами данных (СУБД)** - это комплекс программных и языковых средств, необходимых для создания, обработки баз данных и поддержания их в актуальном состоянии организации поиска в них необходимой информации.

Почти все современные СУБД основаны на **реляционной** модели данных. Название "реляционная" связано с тем, что каждая запись в такой базе данных содержит информацию, относящуюся (related) только к одному объекту. Кроме того, с данными о двух объектах можно работать как с единым целым, основанным на значениях связанных между собой данных. **Все данные в реляционной БД представлены в виде таблиц.** Каждая строка таблицы содержит информацию только об одном объекте и называется **записью**. Столбец таблицы содержит однотипную для всех записей информацию и называется **полем**.

Для успешного функционирования базы данных важна правильная организация данных в ней. При определении структуры данных в базе выделяют следующие основные понятия.

**Класс объектов** - совокупность объектов, обладающих одинаковым набором свойств. Например, в базе данных о ВУЗе классами объектов являются студенты, преподаватели, предметы. Для каждого отдельного объекта из данного класса объектов в таблице создается отдельная запись.

**Свойство (атрибут)** - определенная часть информации о некотором объекте. Хранится в виде столбца (поля) таблицы. Например, *фамилия, имя, отчество* - это свойства для объекта *Студент*.

**Связь (отношение)** - способ, которым связана информация о разных объектах.

## 1.2 Типы связей между объектами

Основным структурным компонентом базы данных, как правило, является таблица. При определении состава таблиц следует руководствоваться правилом: **в каждой таблице должны храниться данные только об одном классе объектов**. Например, в одной таблице нельзя хранить анкетные данные студента и фамилии преподавателей, которым он сдавал экзамены, т.к. это свойства разных классов объектов.

Если в базе данных должна содержаться информация о разных классах объектов, то она должна быть разбита на отдельные таблицы. Связь между таблицами осуществляется с помощью общих полей.

Связи между любыми двумя таблицами относятся к одному из трех типов: один- к- одному (1:1), один- ко- многим (1:M) и много- ко- многим (M:M).

### **Связь типа "один- к- одному" (1:1)**

При этом типе связи каждой записи в одной таблице соответствует **не** более одной записи в другой таблице. Этот вид связи встречается довольно редко. В основном в тех случаях, когда часть информации об объекте либо редко используется, либо является конфиденциальной (такая информация хранится в отдельной таблице, которая защищена от несанкционированного доступа).

### **Связь типа "один- ко- многим" (1:M)**

При таком типе связи каждой записи в одной таблице соответствует несколько записей в связанной таблице. Этот наиболее распространенный тип связей. Для его реализации используются две таблицы. Одна из них представляет сторону "один", другая - сторону "много". Например, нужно иметь информацию о студентах и результатах сдачи ими экзаменов (дата сдачи, предмет, оценка и т.д.). Если все это хранить в одной таблице, то ее объем неоправданно возрастет, т.к. в ней для каждой записи об очередном экзамене должны повторяться все анкетные сведения о студенте. Поскольку *Студент* и *Экзамены* - это разные классы объектов, то и свойства их должны храниться в разных таблицах.

Решением этой задачи является создание двух таблиц. Условно назовем их *Студенты* и *Экзамены*. В каждой из них хранятся соответствующие свойства. Для связи этих таблиц нужно использовать только часть информации о студенте, сдающем экзамен. Но она должна однозначно определять каждого студента среди всех. Такой информацией может явиться, например, номер зачетки (он уникален для каждого студента).

В таблице со стороны "один" (в нашем примере *Студенты*) такие поля называются **ключевыми**. **Основное требование к значениям в ключевых полях - это их уникальность для каждой записи (т.е. они не должны повторяться)**.

### **Связь типа "много- ко- многим" (M:M)**

При таком типе связи множеству записей в одной таблице соответствует множество записей в связанной таблице. Большинство современных СУБД непосредственно не поддерживают такой тип связи. Для его реализации такая связь разбивается на две связи типа один- ко- многим. Соответственно, для хранения информации потребуется уже три таблицы: две со стороны "много" и одна со стороны "один". Связь между этими тремя таблицами также осуществляется по общим полям.

## 2. Структура MS Access

Первая версия MS Access была создана в 1993 г. фирмой Microsoft. MS Access - это функционально полная реляционная СУБД, работающая в среде Windows. Access позволяет создавать сложные базы данных, определяя структуру таблиц, связи между ними. Access обладает совершенной системой создания запросов, отчетов и форм любой сложности. В Access, как любом приложении Windows, можно использовать все возможности обмена данными между приложениями (DDE и OLE), что позволяет включить в базу данных графическую и (или) звуковую информацию.

В Access база данных включает в себя все объекты, связанные с хранимыми данными (таблицы, формы, отчеты, запросы, макросы, модули). **Все** объекты Access хранятся в одном файле с расширением **.mdb**, **.accdb** В таблицах хранятся данные, которые можно просматривать, редактировать, добавлять. Используя формы, можно выводить данные на экран в удобном виде, просматривать и изменять их. Запросы позволяют быстро выбирать необходимую информацию из таблиц. С помощью отчетов можно создавать различные виды документов для вывода на печать, макросы и модули позволяют автоматизировать работу с базой данных.

## 3. Справочная система MS Access

В Access реализована мощная система оперативной помощи, которая предоставляет пользователям возможность получения справочной информации без прерывания работы с программой. В Access предусмотрено несколько способов быстрого доступа к справочной информации с помощью команд из меню **Справка**.


Выбор пункта **Справка** осуществляется нажатием **кнопки F1 на клавиатуре: Microsoft Office Access** выводит на экран диалоговое окно справочной системы с активной строкой поиска, в которой нужно ввести слово или группу слов и нажать **Enter** или кнопку Поиск.

Кнопка **Показать оглавление** открывает страницу с перечнем папок, содержащих разделы справочной системы, этой вкладкой удобно пользоваться при систематическом изучении разделов Access. При единичных обращениях к справке удобнее пользоваться строкой поиска.

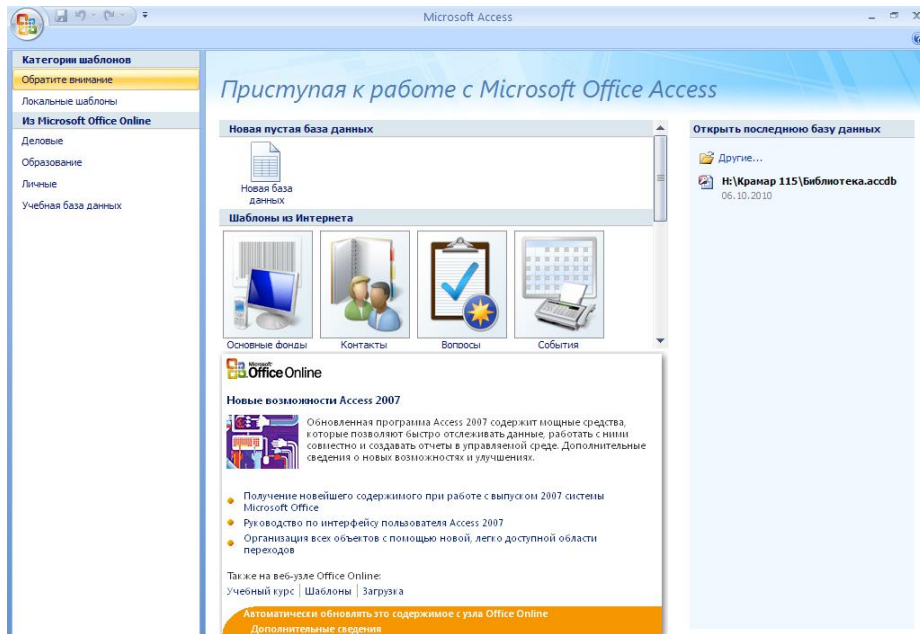


## 4. Начало работы с MS Access

Запустить Access можно несколькими способами:

- двойным щелчком мыши по значку  Microsoft Access на рабочем столе;
- командой **Пуск\ Все программы\ Microsoft Office/ Microsoft Office Access 2007**;
- с помощью **Проводника** ( уже имеющуюся базу данных);

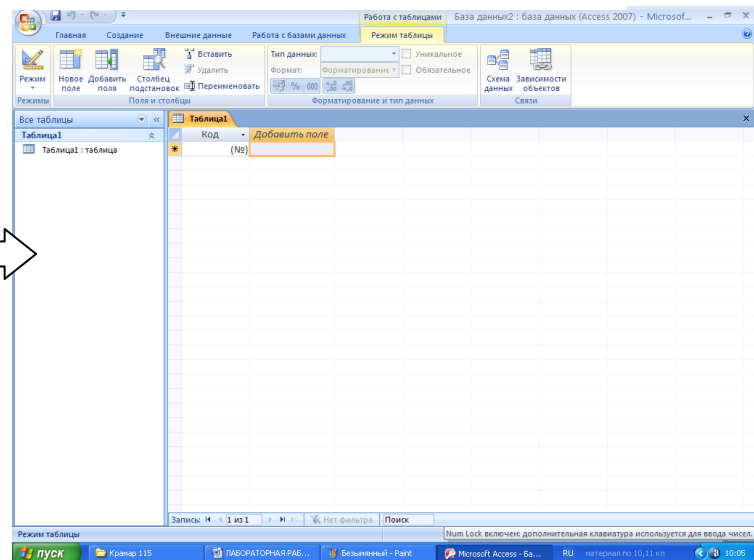
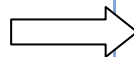
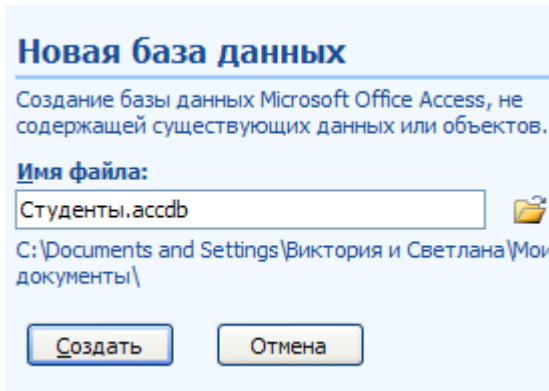
После запуска на экране появится рабочий экран программы Microsoft Access 2007, в левой части которого расположены **Категории шаблонов**, в правой части экрана



указаны уже имеющиеся базы данных на компьютере, которые можно открыть, в центре: создание шаблонов БД из интернета и справочная информация.

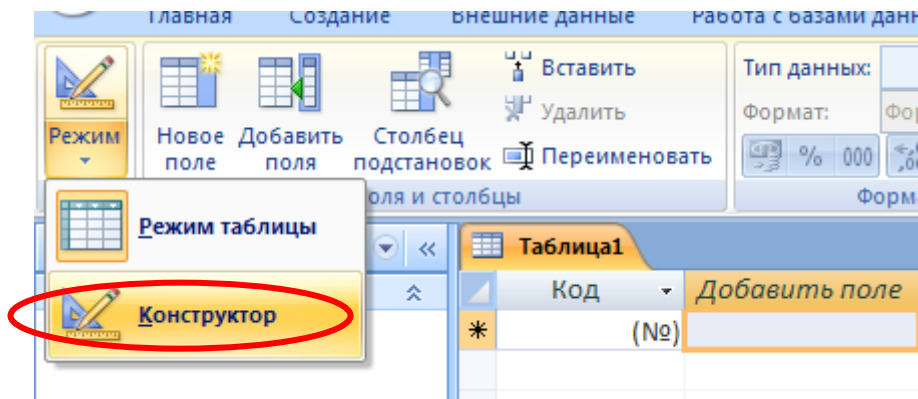
Создать новый файл можно, щелкнув по ссылке **Новая база**

**данных**. После нажатия кнопки **Новая база данных**. Access 2007 запрашивает имя для файла (вы указываете **Студенты** ), в котором будет храниться база данных, его местоположение, и открывает окно новой базы данных.



## 5. Создание новой таблицы с помощью Конструктора

Чтобы приступить к заполнению таблицы базы данных необходимо сначала дать названия полям и назначить им тип данных. Для этого необходимо перейти в «Режим»+ «Конструктор»:

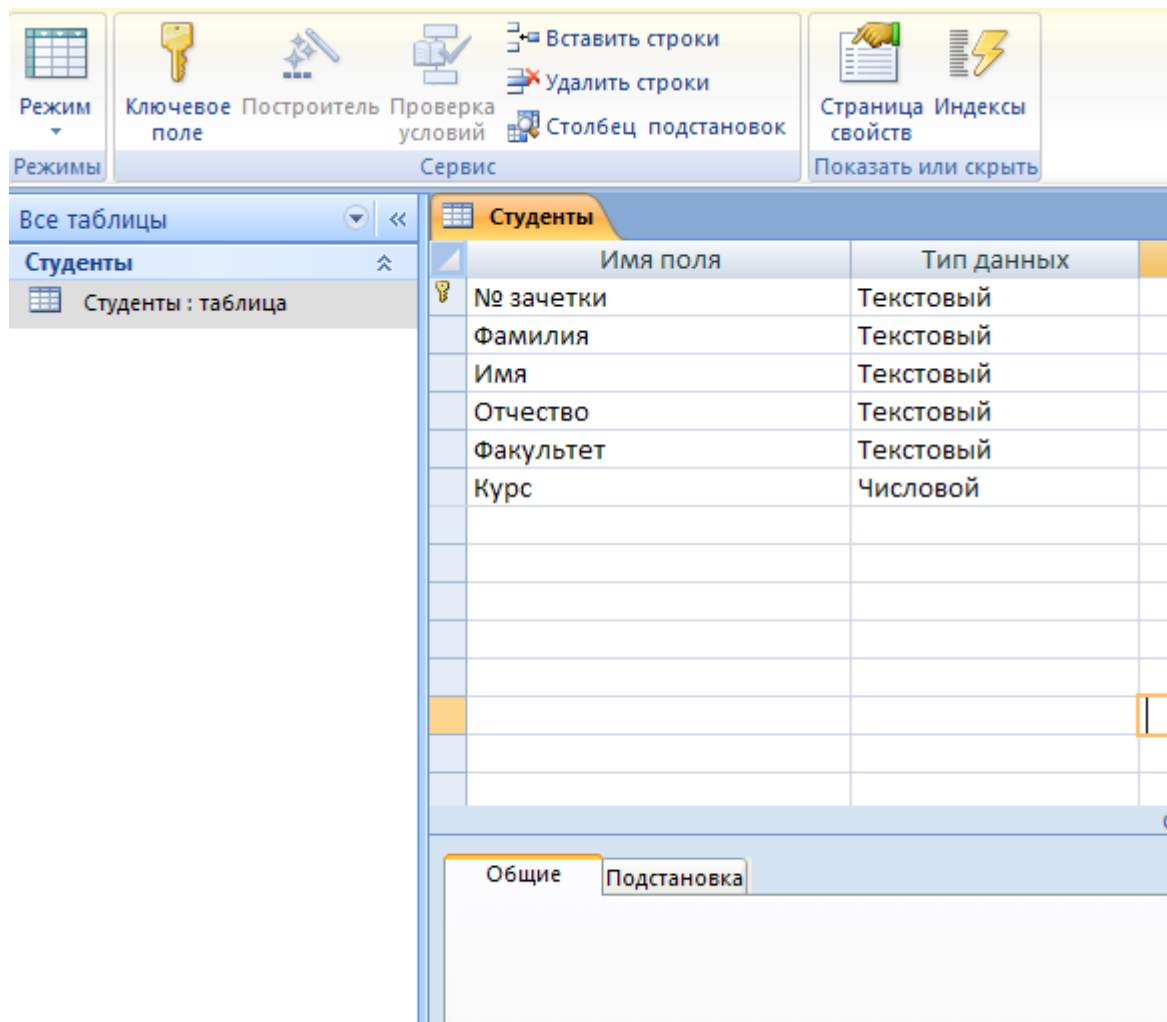


В появившемся окне введите название вашей таблицы и нажмите ОК.

После этого Access выводит окно *Конструктора таблицы*, в котором задаются имена, типы и свойства полей для создаваемой таблицы. По умолчанию первое поле таблицы создается под именем КОД с типом данных Счетчик и предназначено быть **ключевым полем** (обратите внимание на значок КЛЮЧ слева от записи, который можно установить и для другой записи) (в данном столбце будет производиться нумерация всех введенных записей (строк) таблицы. Если это устраивает, то продолжаем заполнять имена полей и типов данных (выбирается из ниспадающего списка). Иначе переименовываем.

Имя поля не должно превышать 64 символа и в нем нельзя использовать символы *пробел, точка, запятая, восклицательный знак*.

Окно вашей базы данных в режиме конструкторы должно выглядеть следующим образом:



Каждая строка в столбце Тип данных является полем со списком, элементами которого являются типы данных Access (таблица 1). Тип поля определяется характером вводимых в него данных.

Таблица 1. Типы данных MS Access 2007

Тип данных	Использование
Текстовый	Алфавитно-цифровые данные (до 255 символов)
Мемо	Алфавитно-цифровые данные-предложения, абзацы, тексты (до 64 000 символов)
Числовой	Различные числовые данные (целое, длинное целое, с плавающей точкой)
Дата\ Время	Дата или время в одном из предлагаемых Access форматов
Денежный	Денежные суммы, хранящиеся с 8 знаками в десятичной части. В целой части каждые три разряда разделяются запятой.
Счетчик	Уникальное длинное целое, создаваемое Access для каждой новой записи
Логический	Логические данные, имеющие значения <i>Истина</i> или <i>Ложь</i>
Объект OLE	Картинки, диаграммы и другие объекты OLE из приложений Windows
Гиперссылка	В полях этого типа хранятся гиперссылки, которые представляют собой путь к файлу на жестком диске, либо адрес в сетях Internet или Intranet.

Среди типов данных Access есть специальный тип - *Счетчик*. В поле этого типа Access автоматически нумерует строки таблицы в возрастающей последовательности. Редактировать значения такого поля нельзя.

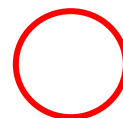
Кроме вышеперечисленных типов данных в списке есть элемент *Мастер подстановок*, который позволяет представить значения полей в виде простого или комбинированного списка. Дополнительные свойства такого поля представлены на вкладке **Подстановка** окна конструктора таблиц.

Каждое поле обладает индивидуальными свойствами, по которым можно установить, как должны сохраняться, отображаться и обрабатываться данные. Набор свойств поля зависит от выбранного типа данных. Для определения свойств поля используется бланк Свойства поля в нижней части окна конструктора таблиц (таблица 2).

Свойство ПОЛЯ	Использование
Размер поля	определяется только для текстовых и Мемо- полей; указывает максимальное количество символов в данном поле. По умолчанию длина текстового поля составляет 50 символов
Формат поля	определяется для полей числового, денежного типа, полей типа <i>Счетчик</i> и <i>Дата\Время</i> . Выбирается один из форматов представления данных.
Число десятичных знаков	определяет количество разрядов в дробной части числа.
Маска ввода	определяет шаблон для ввода данных. Например, можно установить разделители при вводе телефонного номера
Подпись поля	содержит надпись, которая может быть выведена рядом с полем в форме или отчете (данная надпись может и не совпадать с именем поля, а также может содержать поясняющие сведения).
Значение по умолчанию	содержит значение, устанавливаемое по умолчанию в данном поле таблицы. Например, если в поле <i>Город</i> ввести значение по умолчанию <i>Уфа</i> , то при вводе записей о проживающих в Уфе, это поле можно пропускать, а соответствующее значение ( <i>Уфа</i> ) будет введено автоматически. Это облегчает ввод значений, повторяющихся чаще других.
Условие на значение	определяет множество значений, которые пользователь может вводить в это поле при заполнении таблицы. Это свойство позволяет избежать ввода недопустимых в данном поле значений. Например, если стипендия студента не может превышать 250 р., то для этого поля можно задать условие на значение: $\leq 250$ .
Сообщение об ошибке	определяет сообщение, которое появляется на экране в случае ввода недопустимого значения.
Обязательное поле	установка, указывающая на то, что данное поле требует обязательного заполнения для каждой записи. Например, поле <i>Домашний телефон</i> может быть пустым для некоторых записей ( значение <i>Нет</i> в данном свойстве). А поле <i>Фамилия</i> не может быть пустым ни для одной записи (значение <i>Да</i> ).
Пустые строки	установка, которая определяет, допускается ли ввод в данное поле пустых строк.
Индексированное поле	определяет простые индексы для ускорения поиска записей.

Таблица 2. Свойства полей MS Access 2007

Для сохранения структуры таблицы нужно нажать на значок крестика и согласиться с сохранением структуры таблицы



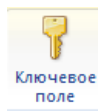
Имя поля	Тип данных	Описание
№ зачетки	Текстовый	
Фамилия	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Факультет	Текстовый	

## 6. Определение ключевых полей

При создании многотабличной базы данных важно правильно определить связи между таблицами и задать **ключи**.

Создавая таблицы с помощью **Конструктора**, Access может автоматически создать первичный ключ, поставив перед первым указанным в структуре полей поле **Код** и задав для него тип *Счетчик*. Для этого при сохранении таблицы в запросе на автоматическое определение ключа нужно ответить **Да**.

Но далеко не всегда такой автоматически определенный ключ бывает корректен. В большинстве случаев ключи должен определять сам разработчик БД. Для того, чтобы определить какое-либо поле таблицы ключевым, нужно установить курсор в строке с



именем этого поля и нажать на кнопку **Ключевое поле** на панели инструментов. После этого в строке рядом с именем этого поля появится изображение ключа. Повторным нажатием на эту кнопку ключ можно убрать.

## 7. Использование режима таблицы

### 7.1 Ввод, редактирование и просмотр данных

**Открываем нашу таблицу двойным щелчком мыши и приступаем к заполнению данных:**

№ зачетки	Фамилия	Имя	Отчество	Факультет
125221	Иванов	Александр		

Используя горизонтальную и вертикальную полосы прокрутки, можно перемещаться на новые поля или записи таблицы.

В режиме таблицы можно не только просматривать данные, но и редактировать их. Для этого используются обычные приемы редактирования и клавиши перемещения между записями.

Если нужно удалить запись целиком, то нужно выделить ее щелчком на области выделения записи (поле маркера) и нажать клавишу **Delete**.

Для облегчения ввода и редактирования данных Access дает несколько "горячих"



клавиш, назначение которых можно посмотреть в разделе **Работа с данными** справочной системы Access 2007( вызов кнопкой F1) .

**Мастер таблиц** предлагает множество готовых структур таблиц по разным тематикам. Пользователь может выбрать подходящую таблицу и ее поля. При необходимости созданную таким образом таблицу можно откорректировать, открыв ее в режиме конструктора.

## 7.2 Использование Мастера подстановок при вводе данных в таблицы

**Мастер подстановок** позволяет формировать для нужного поля список значений, который может содержать данные другой таблицы или запроса, либо состоять из фиксированного набора значений. В обоих случаях **Мастер подстановок** облегчает ввод данных и обеспечивает их достоверность.

Например, в базе данных **Студенты-Экзамены** поле **Номер зачетки** является общим для обеих таблиц. При заполнении этого поля в неключевой таблице **Экзамены** важно, чтобы его значения для каждого студента точно соответствовали значениям в ключевой таблице. Для этого значения поля **Номер зачетки** в таблице **Экзамены** лучше всего представить в виде списка. Это избавит от необходимости печатать данные при заполнении этой таблицы, предотвратит появление ошибок при наборе и сделает ввод данных более наглядным (рис.5).

### *Рис.5 Использование комбинированного списка*

Комбинированный список для поля **Номер зачетки** формируется на основе данных связанной таблицы **Студенты**. Для создания комбинированного списка нужно войти в режим конструктора таблицы **Экзамены**, перейти в колонку **Тип данных** для поля **Номер зачетки**. Из списка доступных типов полей выбрать элемент **Мастер подстановок**.

Первое диалоговое окно **Мастера подстановок** предлагает выбрать источник формирования списка: на основе данных таблицы\запроса или фиксированный набор значений. В данном случае нужно выбрать первый вариант.

В следующем окне из приведенного списка таблиц\запросов следует выбрать таблицу\запрос, являющуюся источником данных для списка. В нашем примере такой таблицей является таблица **Студенты**, т.к. она послужит источником данных для списка номеров зачетов.

В третьем окне Мастера подстановок из списка **Доступные поля** нужно выбрать поля, значения которых фигурируют в списке. В данном случае можно выбрать поля **Фамилия** и **Имя**, которые сделают список более информативным. После этого можно нажать кнопку **Готово** для завершения процесса проектирования комбинированного списка. Сохранив проект таблицы, можно открыть таблицу **Экзамены** и просмотреть результат работы.

Другая возможность использования **Мастера подстановок** - это создание списков с фиксированными значениями. Он используется в тех случаях, когда набор значений в поле строго ограничен. Например, поле **Курс** в таблице **Студенты** может иметь ограниченный набор значений (количество курсов на факультете). Никаких других значений в этом поле быть не может. Чтобы ввести единообразие в записях и избежать

ошибок при вводе, это поле также можно представить в виде списка, формируемого **Мастером подстановок**. Процесс создания списка похож на описанный выше, но на первом шаге **Мастера подстановок** следует выбрать опцию **Фиксированный набор значений**. На следующем шаге нужно создать список этих значений (в данном примере записать в список все курсы - рис. 6). После чего можно завершить процесс создания списка кнопкой **Готово** и сохранить изменения в структуре.

*Рис. 6. Использование Мастера подстановки.*

Перейдите в режим таблицы. Теперь при вводе данных в поле **Факультет** нет необходимости печатать значения, а достаточно просто выбрать из предлагаемого списка нужное.

Чтобы добавить значения в фиксированный список, нужно

1. перейти в режим Конструктора,
2. **Свойства поля - Подстановка,**
3. вписать недостающую запись в **Источник строк**.

#### 9. Определение связи таблиц

Для определения связей нужно закрыть все (!) таблицы, перейти в окно базы данных и выполнить команду **Сервис\Схема данных** или на панели инструментов нажать кнопку  $\text{пф } m3$  **Схема данных**. После этого Access откроет окно **Схема данных** и окно

#### **Добавление**

**таблицы**. Если окно **Добавление таблицы** не появляется автоматически, то для ее вызова

$\Phi$  выберите

команду **Связи/Добавить таблицу** или нажмите кнопку  $\text{ш}$  **Отобразить**

**таблицу** на панели инструментов.

В окне **Добавление таблицы** нужно выделить имя таблицы, добавляемую в схему данных, и нажать на кнопку **Добавить**. После этого данное окно можно закрыть.

В окне **Схема данных** появятся имена всех указанных таблиц вместе со списками полей. Имена ключевых полей выделены полужирным шрифтом.

Для создания связи нужно выделить ключевое поле главной (ключевой) таблицы, нажать кнопку мыши, перетащить ее курсор на аналогичное поле в связываемой (подчиненной) таблице, после чего кнопку мыши отпустить. В результате появится диалоговое окно **Связи** (рис.6).

*Рис. 7 Диалоговое окно Связи*

В этом окне **Access** заполнил первую строку именем поля, по которому связывались таблицы. Чтобы в связанных таблицах не нарушалась целостность данных, нужно щелкнуть по флажку **Обеспечение целостности данных**. После этого **Access** сделает невозможным запись в неключевую таблицу такого значения общего поля, которого нет в таблице ключевой.

После установления целостности данных **Access** активизирует две дополнительные опции: **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**.

Если выбрать первую опцию, то при изменении какого-либо значения ключевого поля в ключевой таблице **Access** автоматически обновит значения этого поля для соответствующих записей во всех связанных таблицах. Например, если у одного из

студентов изменился номер зачетки в таблице **Студенты**, то он автоматически должен поменяться и в таблице **ЛичныеДанные**.

Выбор второй опции при удалении одной из записей в ключевой таблице приведет к удалению тех записей в таблице со стороны "много", которые имеют такое же значение ключа. Например, если из таблицы **Студенты** удалить запись об одном из студентов, то записи о результатах сданных им экзаменов будут удалены автоматически.

Для завершения процесса создания связей, нужно щелкнуть по кнопке **Создать**. Access нарисует линию между таблицами в окне **Схема данных**, указывающую на наличие связи между ними. На конце линии у таблицы со стороны "один" будет нарисована цифра 1, а на другом конце, у таблицы со стороны "много" - символ бесконечности ∞. Установленные связи необходимо сохранить.

## 10. Отображение подчиненных таблиц

Если таблица имеет подчиненные таблицы, то между областью выделения записи и первым столбцом таблицы существует узкий столбец, в котором для каждой записи отображается значок плюс «+». Если этот столбец не отобразился, выполните команду **Вставка/Подтаблица**.

При наличии одной подчиненной таблицы щелчок на значке левой кнопкой мыши открывает часть подчиненной таблицы, содержащей только записи, связанные с записью в главной таблице. Значок «+» при этом меняется на «-».

При наличии нескольких подчиненных таблиц щелчок на значке плюс открывает диалоговое окно **Вставка подтаблицы** со списком подчиненных таблиц для указания имени таблицы.

### Задания

#### 1. Создание структуры таблицы Студенты

1. Создайте новую БД **РЭНГМ** и сохраните на своем сетевом диске
2. Создайте структуру ключевой таблицы **Студенты** в режиме конструктора. Структура таблицы **Студенты** может быть следующей:

Имя поля	Тип данных
№ зачетки	Текстовый
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Факультет	Текстовый
Курс	Числовой
Группа	Числовой
Стипендия	Денежный

Поле **№\_зачетки** в таблице **Студенты** объявить ключевым.

3. Определить свойства полей на вкладке "Общие" следующим образом:

Свойства	Поля						
	№ зачетки	Фамилия	Имя, Отчество	Факультет	Курс	Группа	Стипендия
Размер	6	30	12,30	30			
Формат поля							денежный
Значение по умолчанию				нач. кл.			

Условие на значение					> 0 And <=5		
Сообщение об ошибке					не верно!		
Обязат. поле	да	да	нет	нет	нет	нет	нет

5. Вводить данные в таблицу **Студенты** рекомендуется в режиме таблицы. В одной из записей попробовать внести в поле **Курс** значение, большее 5.

6. Определить тип поля **Курс** с помощью Мастера подстановок, взяв в качестве источника данных *фиксированный набор значений*: 1 и 2

7. Тип поля **Группа** определите Мастером подстановок, используя для подстановки *фиксированный набор значений* -1, 2, 3

## 2. Создание структуры таблицы ЛичныеДанные

1. Создайте структуру таблицы **ЛичныеДанные** с помощью Мастера таблиц.

2. Используйте категорию *Личные*, образцы таблиц - *Адреса*. Добавьте поля **КодАдреса** (сразу же его переименуйте с помощью кнопки «Переименовать поле» на **№\_зачетки**), **Город**, **ДомашнийТелефон**, **ДатаРождения**, **Фотография**. Нажмите Далее >

- В следующем окне измените используемое по умолчанию имя таблицы на **ЛичныеДанные**. При определении ключа выберите «Пользователь определяет ключ самостоятельно» и нажмите Готово

3. В режиме **Конструктора** отмените ключевое поле. В этой таблице ключевых полей быть не должно.

4. Тип данных для поля **№\_зачетки** выберите Текстовый.

5. У вас должно получиться следующее:

Обратите внимание, что для поля **ДатаРождения** выбран тип **Дата/время**, а в свойствах поля определены **Формат поля** и **Маска ввода**. Откройте выпадающие списки и посмотрите, какие еще существуют **Форматы поля** и **Маски ввода**.

Тип поля **Фотография** - **Поле объекта OLE**. Посмотрите в **Таблице 1 «Типы данных»**, для чего используется этот тип.

6. Тип поля **№\_зачетки** определите Мастером подстановок, используя для подстановки данные из таблицы **Студенты**. В качестве доступных при подстановке полей выбрать **Фамилию** и **Имя**. Введите подпись столбца подстановки - **Фамилия**.

7. Для поля **Город** введите значение по умолчанию - Саратов.

8. В режиме таблицы введите данные в таблицу **ЛичныеДанные**.

Для нескольких записей использовать значение по умолчанию в поле **Город**.

Для вставки объекта OLE в поле **Фотография** используйте команду меню **Вставка/Объект**. Выберите объект **Точечный рисунок** (см. рис.ниже), если это будет рисунок в Paint (после создания рисунка достаточно просто закрыть окно - рисунок сохранится автоматически). Если объект находится в каком-либо файле, установите переключатель в положение **Создать из файла** и найдите этот файл.3. **Создание связи между таблицами.**

1. Проверьте, что обе таблицы закрыты.

2. Вызовите окно **Схема данных** и добавьте в схему данных обе таблицы (**Студенты** и **ЛичныеДанные**).

3. Установите в окне схемы данных связь между таблицами по полю **№\_зачетки** (в таблице **ЛичныеДанные** оно отображается как **Фамилия**).

4. В окне **Связи** включите переключатель **Определение целостности данных**. После этого нужно включить опции **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**. После этого закройте окно **Связи**, подтвердив сохранение.

5. Для проверки соблюдения целостности данных при работе с таблицами нужно:

- изменить значение поля **Фамилия** для одной из записей в таблице **Студенты**. Перейти в таблицу **ЛичныеДанные** и проверить, изменилось ли в ней значение общего поля для соответствующих записей;
- удалить одну из записей в таблице **Студенты** (например, последнюю). Перейти в таблицу **ЛичныеДанные** и проверить, удалены ли в ней соответствующие записи.

**Внимание!** После редактирования таблицы **Студенты** нужно сначала сохранить в ней изменения, а затем, переходить в неключевую таблицу.

6. Откройте таблицу **Студенты**. Для нескольких записей отобразите соответствующие записи из связанной таблицы **ЛичныеДанные**.

### **Контрольные вопросы**

1. В чем заключается роль дисциплины «Информационные технологии в профессиональной деятельности» в современном мире?
2. Что такое «Информационные технологии»?
3. Назовите приоритетные направления дисциплины «Информационные технологии в профессиональной деятельности»
4. Перечислите неразрывные части дисциплины «Информационные технологии в профессиональной деятельности»
5. Определение БД и СУБД.
6. Основные характеристики БД.
7. Типы связей "один-к-одному" и "один-ко-многим": как они реализуются в реляционных базах данных?
8. В каких режимах можно создать структуру таблицы в Access 2000?
9. Какая информация указывается в структуре таблицы?
10. Какие ограничения налагаются на имя поля?
11. Что такое ключевое поле и какие требования предъявляются к нему?
12. Как в СУБД Access можно определить ключевое поле?
13. Как отменить ключ?