

## Базы данных

*База данных (БД)* – это поименованная совокупность структурированных данных, относящихся к определенной предметной области. Под *предметной областью* будем понимать некоторую часть реального мира, информация о которой представлена в базе данных. Для взаимодействия пользователей с базами данных используются СУБД. Современные СУБД содержат:

- 1) Набор средств для поддержки таблиц с окном между связанными таблицами;
- 2) Развитый пользовательский интерфейс, который позволяет выводить и модифицировать информацию, выполнять поиск и представлять информацию в текстовом или графическом виде;
- 3) Средства программирования высшего уровня, с помощью которого мы можем создать собственное приложение.

*Система управления базами данных (СУБД)* – это комплекс программных средств, предназначенных для создания и сопровождения базы данных. Можно выделить комплекс основных понятий, с помощью которых описываются принципы построения баз данных. Рассмотрим эти понятия.

*Информационный объект* – это описание некоторого реального объекта, явления, процесса, события в виде совокупности логически связанных *атрибутов*. Например, информационный объект СТУДЕНТ может быть представлен в базе данных с помощью следующих атрибутов: Код студента, Фамилия, Имя, Отчество, Год рождения. Это можно записать так:

СТУДЕНТ (Код студента, Фамилия, Имя, Отчество, Год рождения)

Ядром любой базы данных является модель данных. *Модель данных* – это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними. Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную.

*Иерархическая модель данных* представляет собой совокупность элементов данных, расположенных в порядке их подчинения и образующих по структуре перевернутое дерево (рис. 1). К основным понятиям иерархической модели данных относятся: уровень, узел и связь. Узел – это совокупность атрибутов данных, описывающих информационный объект.

Иерархическая структура должна удовлетворять следующим требованиям:

- каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне;
- существует только один корневой узел на самом верхнем уровне, не подчиненный никакому другому узлу;
- к каждому узлу существует ровно один путь от корневого узла

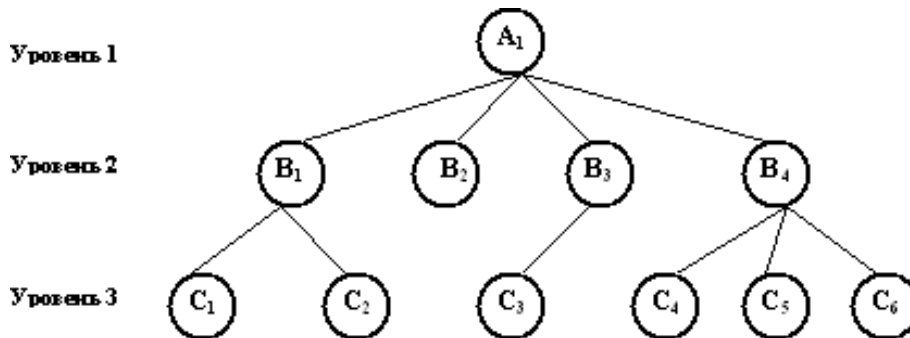


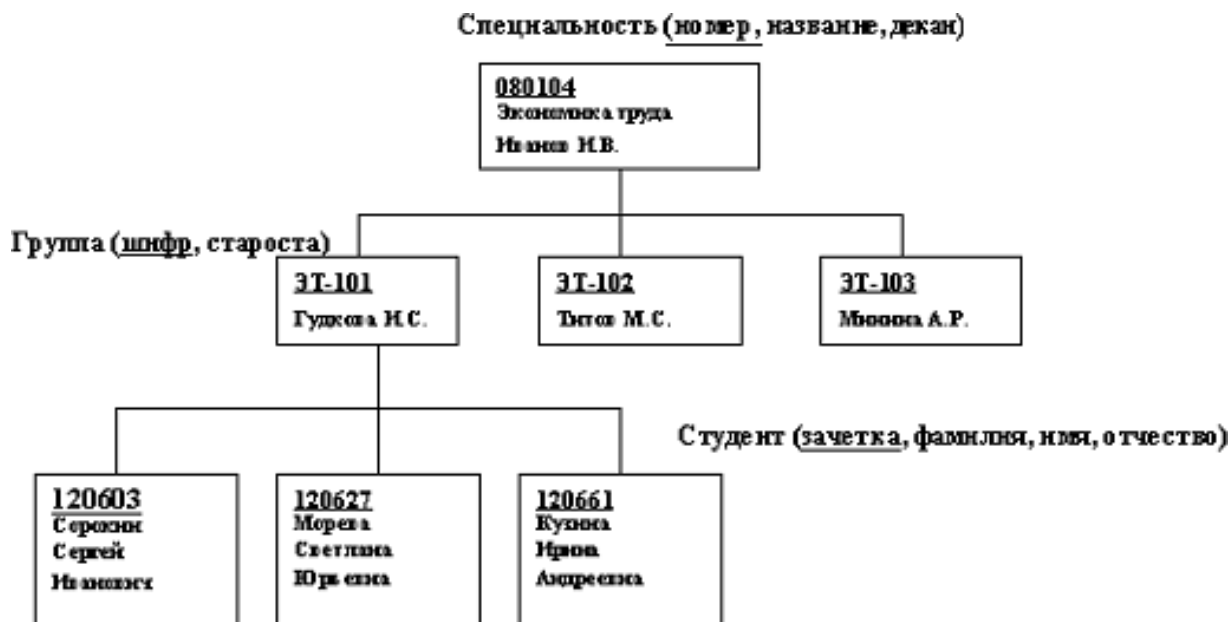
Рис. 1.  
даных

Графическое  
изображение  
иерархической структуры

Иллюстрация использования иерархической модели для построения базы данных «Институт» представлена на рис. 2.

Рис. 2. Пример иерархической структуры данных

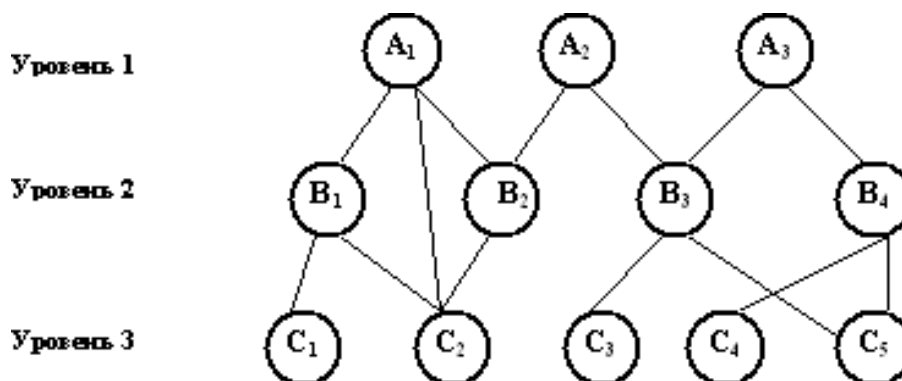
Информация базы данных структурирована в виде иерархических деревьев, количество которых равно количеству специальностей в институте. На первом уровне находится информационный объект Специальность (



номер, название, декан). Информационные объекты второго уровня – Группа (шифр, староста), информационные объекты третьего уровня – Студент (зачетка, фамилия, имя, отчество). Подчеркиванием выделен атрибут, который однозначно определяет каждый экземпляр информационного объекта. Например, атрибут номер однозначно определяет конкретную специальность.

*Сетевая модель данных* основана на тех же основных понятиях (уровень, узел, связь), что и иерархическая модель, но в сетевой модели каждый узел может быть связан с любым другим узлом. На рис. 3 схематически изображена сетевая структура организации данных.

Рис. 3. Графическое изображение сетевой структуры данных



Примером сетевой структуры может служить структура базы данных, содержащей сведения о студентах, занимающихся в спортивных секциях. Возможно участие одного студента в нескольких секциях, возможно также участие нескольких студентов в одной секции. Графическое представление описанной в примере сетевой структуры, состоящей из двух типов информационных объектов, показано на рис. 4

Рис. 4. Пример сетевой структуры данных

*Реляционная модель данных* использует организацию данных в виде двумерных таблиц. Каждая такая таблица, называемая *реляционной таблицей* или *отношением*, представляет собой двумерный массив и обладает следующими свойствами:

- все столбцы в таблице однородные, т.е. все элементы в одном столбце имеют одинаковый тип и максимально допустимый размер;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись (рис. 5). *Поле* (столбец реляционной таблицы) – элементарная единица логической организации данных, которая соответствует конкретному атрибуту информационного объекта. *Запись* (строка реляционной таблицы) – совокупность логически связанных полей, соответствующая конкретному экземпляру информационного объекта.

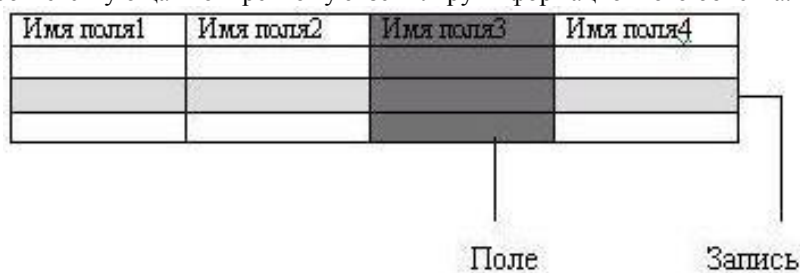


Рис. 5. Основные структурные элементы реляционной таблицы

Например, в виде реляционной таблицы можно представить информацию о студентах, обучающихся в вузе (рис. 6).

Рис. 6. Пример реляционной таблиц

Номер личного дела	Фамилия	Имя	Отчество	Пол	Дата рождения	Специальность
16493	Сергеев	Петр	Михайлович	м	01.01.86	080104
16593	Петрова	Анна	Владимировна	ж	15.03.85	080102
16693	Анохин	Андрей	Борисович	м	14.04.86	080104

## Основы проектирования реляционных баз данных

Основными объектами любой базы данных являются таблицы. Таблицы базы данных создаются таким образом, чтобы каждая из них содержала информацию об одном информационном объекте. Между таблицами должны быть установлены реляционные связи. Установка таких связей делает возможным выполнение одновременной обработки данных из нескольких таблиц.

*Первичный ключ* реляционной таблицы – это поле или группа полей, которые позволяют однозначно определить каждую запись (строку) в таблице. Первичный ключ должен обладать двумя свойствами:

- *однозначная идентификация записи* – запись должна однозначно определяться значением ключа;
- *отсутствие избыточности* – никакое поле нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации записи.

Если первичный ключ состоит из одного поля, то он называется *простым ключом* или *ключевым полем*. Если первичный ключ состоит из нескольких полей, то говорят, что таблица имеет *составной ключ*.

Для установки связей между таблицами используются ключевые поля. Можно связать две реляционные таблицы, если ключ одной связываемой таблицы ввести в состав ключа другой таблицы (возможно совпадение ключей). Можно ключевое поле одной связываемой таблицы ввести в структуру другой таблицы так, что оно в этой таблице не будет ключевым. В этом случае это поле называется *внешним ключом*.

Пример 4.1. Рассмотрим базу данных «Деканат», состоящую из трех таблиц: СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ (рис. 7).

В таблице СТУДЕНТ ключевым полем является поле Номер личного дела, оно однозначно определяет каждую запись таблицы, если предположить, что не существует двух или более личных дел с одинаковыми номерами. В таблице СЕССИЯ ключевым полем также является поле Номер личного дела. В столбце Результат содержатся числа 0, 1 или 2 в зависимости от того, получает ли студент стипендию по результатам сессии (0 – студент не получает стипендию, 1 – обычная стипендия, 2 – увеличенная в 2 раза стипендия).

Таблица СТИПЕНДИЯ содержит информацию о проценте начисляемой студенту стипендии в зависимости от результата сдачи сессии. В этой таблице ключевым является поле Результат.

Таблицы СТУДЕНТ и СЕССИЯ имеют совпадающие первичные ключи, что дает возможность легко организовать связь между ними по полю Номер личного дела. Таблица СЕССИЯ имеет первичный ключ Номер личного дела и содержит внешний ключ Результат, который позволяет обеспечить ее связь с таблицей СТИПЕНДИЯ.

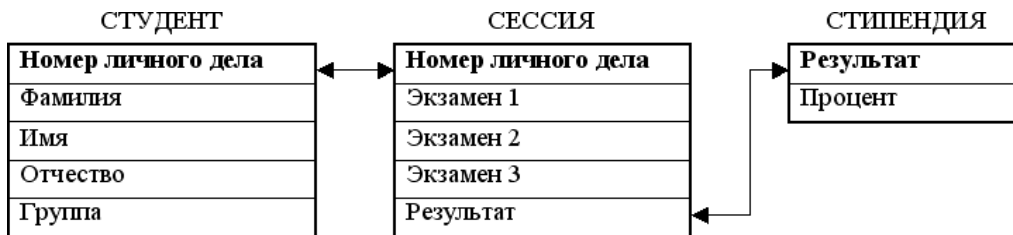


Рис. 7. Реляционные таблицы базы данных «Деканат»

Для наглядности представления связей между таблицами перейдем к представлению таблиц в виде структур этих таблиц, т.е. будем указывать только имена полей таблиц (рис. 8).

Таблица СТУДЕНТ

Номер личного дела	Фамилия	Имя	Отчество	Группа
120603	Свиридова	Анна	Алексеевна	ЭТ-101
120608	Коробов	Игорь	Андреевич	ЭТ-102
...	...	...	...	...

Таблица СЕССИЯ

Номер личного дела	Экзамен 1	Экзамен 2	Экзамен 3	Результат
120603	3	4	4	0
120608	5	5	5	2
120634	4	5	4	1
...	...	...	...	

Таблица СТИПЕНДИЯ

Результат	Процент
0	0
1	100
2	200

Рис. 8. Пример реляционной структуры данных

В зависимости от того, какие ключи (первичные или внешние) используются для установки связи между таблицами, различают три типа связей между таблицами реляционной базы данных: «один-к-одному» (1 – 1), «один-ко-многим» (1 – М), «многие-ко-многим» (М – М)

Считается, что запись таблицы А связана с записью таблицы В, если в обеих таблицах эти записи содержат одно и то же значение в поле, по которому установлена связь между таблицами.

Между таблицами А и В установлена связь «один-к-одному», если каждая запись в таблице А может иметь не более одной связанной с ней записи в таблице В и наоборот, каждая запись в таблице В может иметь не более одной связанной с ней записи в таблице А. В этом случае для связи используются первичные ключи связываемых таблиц.

Рассмотренные ранее таблицы СТУДЕНТ и СЕССИЯ связаны по типу один-к-одному» (рис. 9).

Рис. 9. Пример таблиц, связанных по типу «один-к-одному»

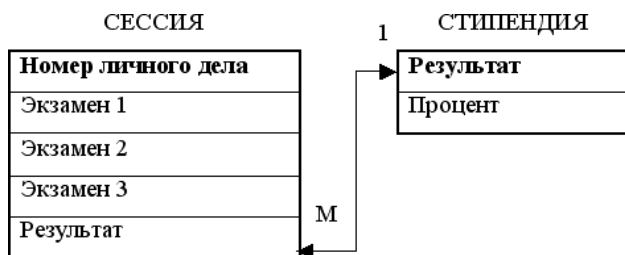
Между таблицами А и В установлена связь «один-ко-многим», если каждая запись в таблице А может быть связана с несколькими записями таблицы В, но каждая запись в таблице В не может быть связана более чем с одной записью таблицы А. Таблица А в этом случае называется *главной* таблицей, а

таблица В – *подчиненной*. В этом случае для связи используется поле, которое является первичным ключом таблицы, находящейся на стороне отношения

«один» (таблица А), и являющееся внешним ключом в таблице, находящейся на стороне отношения «многие» (таблица В)

Например, рассмотренные ранее таблицы СТИПЕНДИЯ и СЕССИЯ связаны по типу «один-ко-многим» (рис. 10). При этом на стороне «один» находится таблица СТИПЕНДИЯ, а на стороне «многие» – таблица СЕССИЯ. Связь устанавливается по полю Результат. Каждая запись таблицы СТИПЕНДИЯ может иметь много связанных с ней записей в таблице СЕССИЯ, иначе говоря, в таблице СЕССИЯ может быть много строк с заданным значением в поле Результат (например, со значением 1). В то же время, если взять любую строку в таблице СЕССИЯ, то для нее найдется не более одной строки в таблице СТИПЕНДИЯ с таким же значением в поле Результат.

Рис. 10. Пример таблиц, находящихся



Между таблицами А и В установлена *связь «многие-ко-многим»*, если каждой записи таблицы А может соответствовать несколько записей в таблице В, и наоборот, каждой записи таблицы В может соответствовать несколько записей в таблице А. *Такая связь всегда реализуется с помощью третьей связующей таблицы* (назовем ее – таблица С). Связь «многие-ко-многим» представляет собой комбинацию двух связей типа «один-ко-многим»: между таблицами А и С (А – главная, С – подчиненная) и между таблицами В и С (В – главная, С – подчиненная).

Примером могут служить таблицы ЧИТАТЕЛИ и КНИГИ базы данных БИБЛИОТЕКА (рис. 11). Связь между ними организуется с помощью таблицы АБОНЕМЕНТ. Каждой записи в таблице ЧИТАТЕЛИ могут соответствовать в отношении «один-ко-многим» несколько записей в таблице КНИГИ, и наоборот, каждой записи таблицы КНИГИ соответствовать несколько записей в таблице ЧИТАТЕЛИ. Соответствие устанавливается с помощью таблицы АБОНЕМЕНТ. При этом между таблицами ЧИТАТЕЛИ и АБОНЕМЕНТ установлена связь «один-ко-многим», в которой таблица ЧИТАТЕЛИ является главной. Аналогично между таблицами КНИГИ и АБОНЕМЕНТ установлена связь «один-ко-многим», в которой таблица КНИГИ является главной.

Рис. 11. Пример таблиц, находящихся в отношении «многие-ко-многим»

## Компоненты MS Access.

Рассмотрим основные приемы разработки баз данных и работы с ними на примере СУБД Microsoft Access. База Данных в Microsoft Access – это один файл с расширением .mdb, содержащий таблицы, запросы и другие *объекты* базы данных. Рассмотрим эти объекты более подробно.

M.S.Access состоит из отдельных компонентов, которые используются для хранения и представления информации. Этими компонентами являются таблица, форма, запросы, отчеты, макросы, модули и страницы. Для создания форм и отчетов используются конструкторы, поэтому эти конструкторы называются:

**Конструкторскими объектами.** Конструкторские объекты являются составными объектами, то есть они состоят из более мелких объектов, таких как: кнопки, диаграммы, рамки и так далее, которые называются:

**Элементами управления.** К элементам управления относятся: надписи, прямоугольники и линии, поля и списки, кнопки, переключения, включения и флажки, графические объекты, OLE объекты.

**Таблицы** являются основной базой данных. В M.S.Access вся информация содержится в таблице.

**Форма** используется для ввода и просмотра таблиц в окне конструктора. Форма позволяет ограничить объём информации отображаемой на экране.

**Отчеты** используются для отображения информации содержащихся в базе данных. С помощью конструктора отчетов мы можем создать свой собственный отчет.

**Запрос** является средством извлечения информации из базы данных.

**Макросы** предназначаются для автоматических или часто выполняемых операций. Каждый макрос содержит один или несколько макрокоманд.

## Знакомство с главным окном MS Access

M.S.Access хранит все наши таблицы, макросы и модули в базе данных, поэтому при первом запуске в M.S.Access на экран выводится окно базы данных и панель инструментов. Экран состоит из пяти областей: заголовок, строка меню, панель инструментов, строка состояний и окно базы данных. Заголовок находится в верхней части главного окна. Меню M.S.Access меняется в зависимости оттого, что мы хотим сделать. Также как и в других приложениях Windows. Эти меню следуют определённому порядку и стандарту. В M.S.Access имеется множество самых разнообразных панелей инструментов, которые содержат набор кнопок, состав которых зависит от назначения конкретных инструментов. При первом входе в M.S.Access на экране находится и инструкция называемая базой данных.

### **Строка состояния**

Строка состояния находится в нижней части главного окна. В левой части строки состояния может отображаться описание панели таблиц, форма или другая получаемая информация. Окно базы данных состоит из семи приложений: таблица, запросы, форма, отчеты, макросы, модули, страницы. В правой части окна находятся кнопки: открыть, конструктор, создать. Кнопка открыть предназначается для открытия выбранного объекта. Этим объектом может быть: таблица, запросы, отчеты и так далее. При открытии таблицы открывается окно таблицы, и мы можем просмотреть и редактировать данные таблицы. Открытие запроса к выполнению выбранного запроса и просмотра результата выбора. При переходе на вкладку отчета кнопки открыть на просмотр. Кнопка конструктор предназначена для модификации выбранного объекта, кнопка создание для его создания.

### **Создание базы данных**

M.S.Access является системой управления работой базы данных. В работе базы данных в нынешнее время наиболее распространено и является промышленным стандартом.

Теория работы базы данных была разработана в начале семидесятых годов. База данных в M.S.Access представляет собой совокупность средств для ввода, хранения, просмотра, выборки и управления информацией. К этим средствам относятся таблица, форма, запросы, отчеты, макросы.

### **Создание пустой базы данных**

Для создания новой базы данных выполняем следующие действия:

1. Выбираем наш файл создать базу данных; на экране открывается окно диалога создание, содержимое второй вкладки общие базы данных. Общие позволяет создать новую пустую базу данных. База данных позволяет выбрать образец базы данных содержащих большинство требуемых объектов и создать базу данных с помощью мастера.
2. Для создания новой базы данных переходим на вкладку общие и нажимаем кнопку ОК, на экране открывается окно файл/новая база данных.
3. Выбираем из раскрывающегося списка папка, папку, которая предлагает сохранить создаваемую базу данных, а в поле имя файла выводим имя базы данных. После ввода имени созданной базы данных нажимаем кнопку создание данного окна диалога, на экране появляется окно базы данных. Окно базы данных состоит из семи вкладок

### **Строка состояния**

Строка состояния находится в нижней части главного окна. В левой части строки состояния может отображаться описание панели таблиц, форма или другая получаемая информация. Окно базы данных состоит из семи приложений: таблица, запросы, форма, отчеты, макросы, модули, страницы. В правой части окна находятся кнопки: открыть, конструктор, создать. Кнопка открыть предназначается для открытия выбранного объекта. Этим объектом может быть: таблица, запросы, отчеты и так далее. При открытии таблицы открывается окно таблицы, и мы можем просмотреть и редактировать данные таблицы. Открытие запроса к выполнению выбранного запроса и просмотра результата выбора. При переходе на вкладку отчета кнопки открыть на просмотр. Кнопка конструктор предназначена для модификации выбранного объекта, кнопка создание для его создания.

### Создание базы данных

M.S.Access является системой управления работой базы данных. В работе базы данных в нынешнее время наиболее распространено и является промышленным стандартом.

Теория работы базы данных была разработана в начале семидесятых годов. База данных в M.S.Access представляет собой совокупность средств для ввода, хранения, просмотра, выборки и управления информацией. К этим средствам относятся таблица, форма, запросы, отчеты, макросы.

### Создание пустой базы данных

Для создания новой базы данных выполняем следующие действия:

4. Выбираем наш файл создать базу данных; на экране открывается окно диалога создание, содержимое второй вкладки общие базы данных. Общие позволяет создать новую пустую базу данных. База данных позволяет выбрать образец базы данных содержащих большинство требуемых объектов и создать базу данных с помощью мастера.
5. Для создания новой базы данных переходим на вкладку общие и нажимаем кнопку ОК, на экране открывается окно файл/новая база данных.
6. Выбираем из раскрывающегося списка папка, папку, которая предлагает сохранить создаваемую базу данных, а в поле имя файла выводим имя базы данных. После ввода имени созданной базы данных нажимаем кнопку создание данного окна диалога, на экране появляется окно базы данных. Окно базы данных состоит из семи вкладок

### **Основы работы с СУБД Microsoft Access.**

Объекты базы данных Microsoft Access

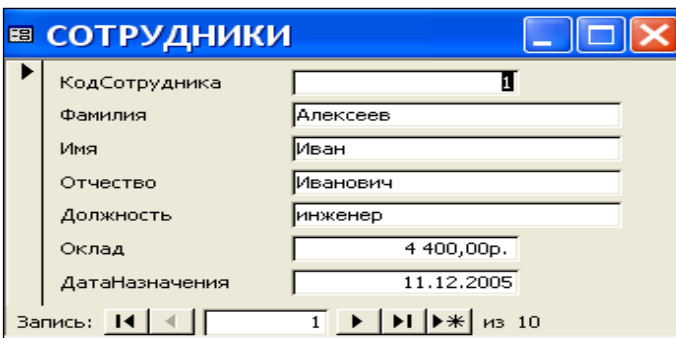
*Таблицы* – это основные объекты любой базы данных. В таблицах хранятся данные. На рис. 15 показан пример таблицы базы данных в режиме просмотра данных.



	КодСотрудника	Фамилия	Имя	Отчество	Должность	Оклад	ДатаНазначения
+	1	Алексеев	Иван	Иванович	инженер	4 400,00р.	11.12.2005
+	2	Алехин	Игорь	Иванович	инженер	8 800,00р.	11.12.2005
+	9	Булкин	Алексей	Игоревич	инженер	3 000,00р.	21.11.2006
+	4	Зотова	Инна	Андреевна	инженер	3 300,00р.	13.01.2005
+	5	Иванов	Иван	Иванович	бухгалтер	4 400,00р.	26.09.2006
+	6	Петров	Петр	Петрович	бухгалтер	11 000,00р.	31.12.2005

Рис. 15. Пример таблицы базы данных

*Запросы* – это специальные средства для отбора и анализа данных. На рис.16 показан пример выполнения запроса на выборку данных из таблицы СОТРУДНИКИ согласно критерию отбора Должность = Инженер.



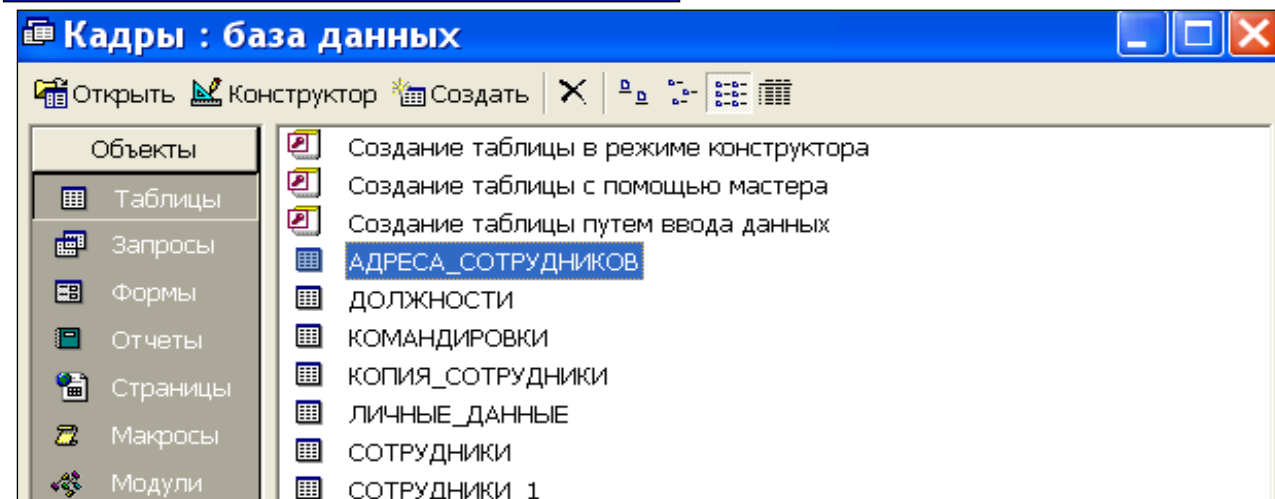
СОТРУДНИКИ

КодСотрудника: 1  
Фамилия: Алексеев  
Имя: Иван  
Отчество: Иванович  
Должность: инженер  
Оклад: 4 400,00р.  
ДатаНазначения: 11.12.2005

Запись: 1 из 10

Рис. 16. Пример выполнения запроса к базе данных

*Формы* используются для ввода новых данных в таблицы базы данных, а также для просмотра имеющихся данных. На рис. 17 показана форма для таблицы



СОТРУДНИКИ.

Рис. 17. Пример формы базы данных

*Отчеты* предназначены для вывода данных на экран или на принтер. В них предусмотрены средства для вывода специальных элементов оформления, характерных для печатных документов: верхний и нижний колонтитулы, номера страниц и т.п. (рис. 18).

Рис. 18. Пример простейшего отчета

<i>Сотрудники1</i>				
<i>Должность</i>	<i>Фамилия</i>	<i>Имя</i>	<i>Отчество</i>	<i>Оклад</i>
<i>Бухгалтер</i>				
	Мельникова	Анастасия	Сергеевна	5 000,00р.
	Ким	Ирина	Станиславовна	5 000,00р.
	Иванов	Иван	Иванович	5 000,00р.
<i>Инженер</i>				
	Прохорова	Ирина	Александровна	4 000,00р.
	Сидоров	Александр	Александрович	4 000,00р.
	Алифанова	Алёна	Анатольевна	4 000,00р.

*Страницы доступа к данным* служат для обеспечения удаленного доступа к данным, содержащимся в базе данных. Их можно встроить в любой Web-документ. Страницы доступа выполняют связующую функцию между базой данных и Web-страницами Интернета.

*Макросы* предназначены для автоматизации выполнения повторяющихся операций при работе с СУБД. Несколько команд управления базой данных могут быть сохранены в виде макроса, чтобы в дальнейшем можно было воспроизводить эти команды, запуская сохраненный макрос.

*Модули* создаются средствами языка программирования Visual Basic for Applications. С их помощью разработчик базы данных может заложить в нее нестандартные функциональные возможности.

Работа с любыми объектами базы данных выполняется в окне База данных (рис. 19). На левой панели данного окна находятся элементы управления для вызова всех типов объектов.

#### Работа с таблицами

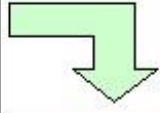
Структура таблицы определяется набором и свойствами полей (столбцов таблицы), а информация хранится в виде записей (строк таблицы). Далее перечислены некоторые свойства полей:

- *Имя поля* – определяет, как следует обращаться к данным этого поля при операциях с базой данных (должно быть уникальным в рамках таблицы);
- Наименование поля вводится в столбце имя поля. При задании наименований полей вы должны следовать следующим правилам:

**СОТРУДНИКИ : таблица**

КодСотрудника	Фамилия	Имя	Отчество	Должность	Оклад	ДатаНазначения
1	Алексеев	Иван	Иванович	инженер	4 400,00р.	11.12.2005
2	Алекин	Игорь	Иванович	инженер	8 600,00р.	11.12.2005
9	Булкин	Алексей	Игоревич	инженер	3 000,00р.	21.11.2006
4	Зотова	Инна	Андреевна	инженер	3 300,00р.	13.01.2005
5	Иванов	Иван	Иванович	бухгалтер	4 400,00р.	26.09.2006
6	Петров	Петр	Петрович	бухгалтер	11 000,00р.	31.12.2005
15	Пирогов	Александр	Алексеевич	инженер	7 000,00р.	01.01.2003
7	Сафронов	Игорь	Трофимович	инженер	3 600,00р.	01.01.2006
8	Сидоров	Иван	Олегович	менеджер	5 500,00р.	02.06.2006
10	Хлебов	Олег	Олегович	инженер	4 500,00р.	30.03.2004

Базовая таблица



Запрос на выборку

**Запрос21 : запрос на выборку**

СОРТУ...

КодСотрудн  
Фамилия  
Имя  
Отчество  
Должность

Поле: КодСотрудника  
Имя таблицы: СОТРУДНИКИ  
Сортировка: по возрастанию  
Вывод на экран:  
Условие отбора:

КодСотрудника	Фамилия	Имя	Отчество	Должность
СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ

**Инженер : запрос на выборку**

КодСотрудни	Фамилия	Имя	Отчество	Должность
1	Алексеев	Иван	Иванович	инженер
2	Алекин	Игорь	Иванович	инженер
4	Зотова	Инна	Андреевна	инженер
7	Сафронов	Игорь	Трофимович	инженер
9	Булкин	Алексей	Игоревич	инженер
10	Хлебов	Олег	Олегович	инженер
15	Пирогов	Александр	Алексеевич	инженер

Результирующая таблица

1. Наименование поля может содержать до 6 символов.
  2. Наименование поля может содержать буквы, цифры, пробелы и специальные знаки за исключением точки, прямых скобок и управления символов с кодами ASCLL0-31.
  3. Наименование поля не может начинаться с пробела.
  4. 2 поля в одной таблице не могут иметь одинаковых наименований.
- Тип поля – определяет тип данных, которые могут содержаться в данном поле ;
  - Размер поля – определяет предельную длину данных, которые могут размещаться в данном поле;
  - Формат поля – определяет способ форматирования данных в ячейках таблицы, принадлежащих полю (например, значение даты 19 января 2004 года может быть представлено в разных форматах: 19 января 2004 г., 19.06.2004, 19-январь-2004);
  - Подпись – определяет заголовок столбца таблицы для данного поля (если подпись не указана, то в качестве заголовка столбца используется Имя поля);
  - Значение по умолчанию – то значение, которое вводится в ячейки поля автоматически при формировании очередной записи таблицы;
  - Условие на значение – ограничение, используемое для проверки правильности ввода данных;
- Свойство Тип поля может принимать одно из следующих значений:
- Текстовый – используется для хранения текста ограниченного размера (до 255 символов);  
При вводе имени поля по умолчанию M.S.Access присваивает ему текстовый тип данных шириной поля равной пятьдесят. Для изменения ширины поля нужно в строке размер поля, в разделе свойства поля задать число, определяющее ширину поля.
  - Поле Мемо – специальный тип для хранения больших объемов текста (до 65 535 символов); Текстовые поля произвольной длины могут содержать те же типы данных, что и простые текстовые поля. Отличие между этими полями заключается в том, что размер поля Мемо не ограничивается 255 символами, а может содержать до 65535 символов.
  - Числовой – используется для хранения чисел;  
Возникает вопрос с какой целью вводится числовой тип полей? В качестве основных аргументов по использованию числовых полей приведём следующие: При вводе данных числового типа автоматически производится проверка данных. Если данные содержат текстовые или специальные символы M.S.Access выдаёт предупреждение и ошибочные данные не будут введены. Выполнение математических операций возможно только над числовыми полями. Для указания количества десятичных знаков используется свойство поле число десятичных знаков. Оно может принимать значение от нуля до пятнадцати. Атрибут автор данного свойства служит для автоматической установки количества знаков после запятой.
  - Дата/время – используется для хранения календарных дат и текущего времени;  
Использование для хранения даты времени полей в которых данные представлены в специальном формате выдвигает

нам ряд преимуществ:

1. M.S.Access позволяет вводить и отображать даты в нескольких форматах. Конкретно варианты отображения даты/времени устанавливаются в свойстве формат поля окна конструктора таблиц. При хранении данных эти форматы автоматически преобразуются во внутреннее представление данных.

2. при вводе дат M.S.Access осуществляет проверку формата данных и допускает ввод только правильных дат. В поле данного типа могут создаваться любые дата и время от 1 января 100 года до 31 декабря 9999 года.

- *Денежный* – используется для хранения денежных сумм.

Данное поле аналогично числовому полю. В отличие от числового поля для денежного поля свойство формат поля устанавливается автоматически, а значение денежной в свойстве число десятичных знаков принимает значение равное двум знакам после запятой, которые при желании можно изменить.

- *Счетчик* – содержит натуральные числа 1, 2, 3, ... (при этом значения *вводятся в поле автоматически* при создании новой записи);

Поля типа счетчик предназначены для хранения данных, значение которых не редактируется, а устанавливается автоматически при добавлении каждой новой записи в таблицу. Их значения являются уникальными последственно возникающими при добавлении каждой новой записи. Их значения могут быть и случайными числами.

- *Логический* – используется для хранения логических данных (Истина или Ложь);

Логическое поле используется для хранения данных, которые могут принимать одно из двух возможных значений. Свойство формат поля логического поля позволяет использовать специальный формат или один из трех встроенных : истина- ложь, да- нет, включить- выключить, при этом значение истина, да и включить являются эквивалентами логического значения true, а значения ложь, нет и выключить – эквивалентами логического значения false. Если вы выбрали встроенный формат, а затем будете вводить эквивалентное логическое значение, то введенное значение будет отображаться в выбранном формате.

- *Поле объекта OLE* – используется для хранения графических, аудио- и видеоданных, фрагментов электронных таблиц и др. Реально такие данные в таблице не хранятся. Они хранятся в другой части файла базы данных, а в таблице хранятся только указатели на них.

- *Гиперссылка* – специальное поле для хранения адресов Web-страниц Интернета. При щелчке на ссылке автоматически происходит запуск программы-браузера и воспроизведение страницы в его окне.

### Создание таблиц

Создание таблицы выполняется после выбора объекта Таблицы в окне База данных (рис. 19). Чтобы создать таблицу, следует использовать значок Создание таблицы в режиме конструктора или выбрать пункт Конструктор в управляющем меню окна. Окно Конструктора таблиц представлено на рис. 20. В столбце Имя поля вводятся имена полей будущей таблицы. Тип каждого поля выбирается из раскрывающегося списка. Нижняя часть окна содержит список свойств поля, выделенного в верхней части.

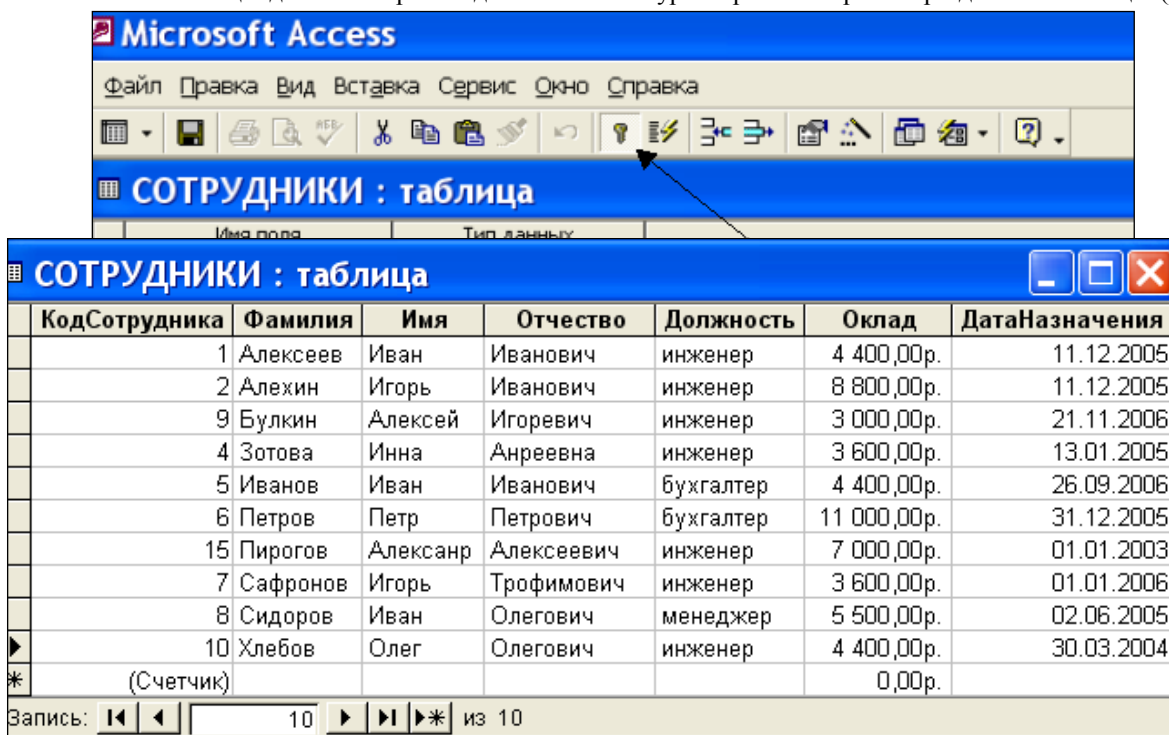
Рис.20. Таблица в режиме Конструктор

При создании таблицы следует определить ключевое поле, являющееся первичным ключом таблицы. Для этого нужно щелкнуть на имени поля правой кнопкой мыши и выбрать пункт Ключевое поле из контекстного меню или щелкнуть инструмент Ключевое поле на панели инструментов в окне программы (рис. 21). Закончив создание структуры таблицы, нужно закрыть окно Конструктора, сохранив созданную таблицу. После сохранения значок таблицы появляется на вкладке Таблицы в окне База данных.

Рис. 21. Задание ключевого поля таблицы

Ввод данных в таблицу

Заполнение таблицы данными производится с клавиатуры в режиме просмотра данных таблицы (рис. 22). В



нижней части таблицы расположена Панель кнопок перехода, с помощью инструментов которой также можно выполнить перемещение к нужной строке таблицы. Ширину столбцов можно изменять методом перетаскивания их границ с помощью мыши. *Сохранение вводимых в таблицу данных выполняется автоматически в режиме реального времени.*

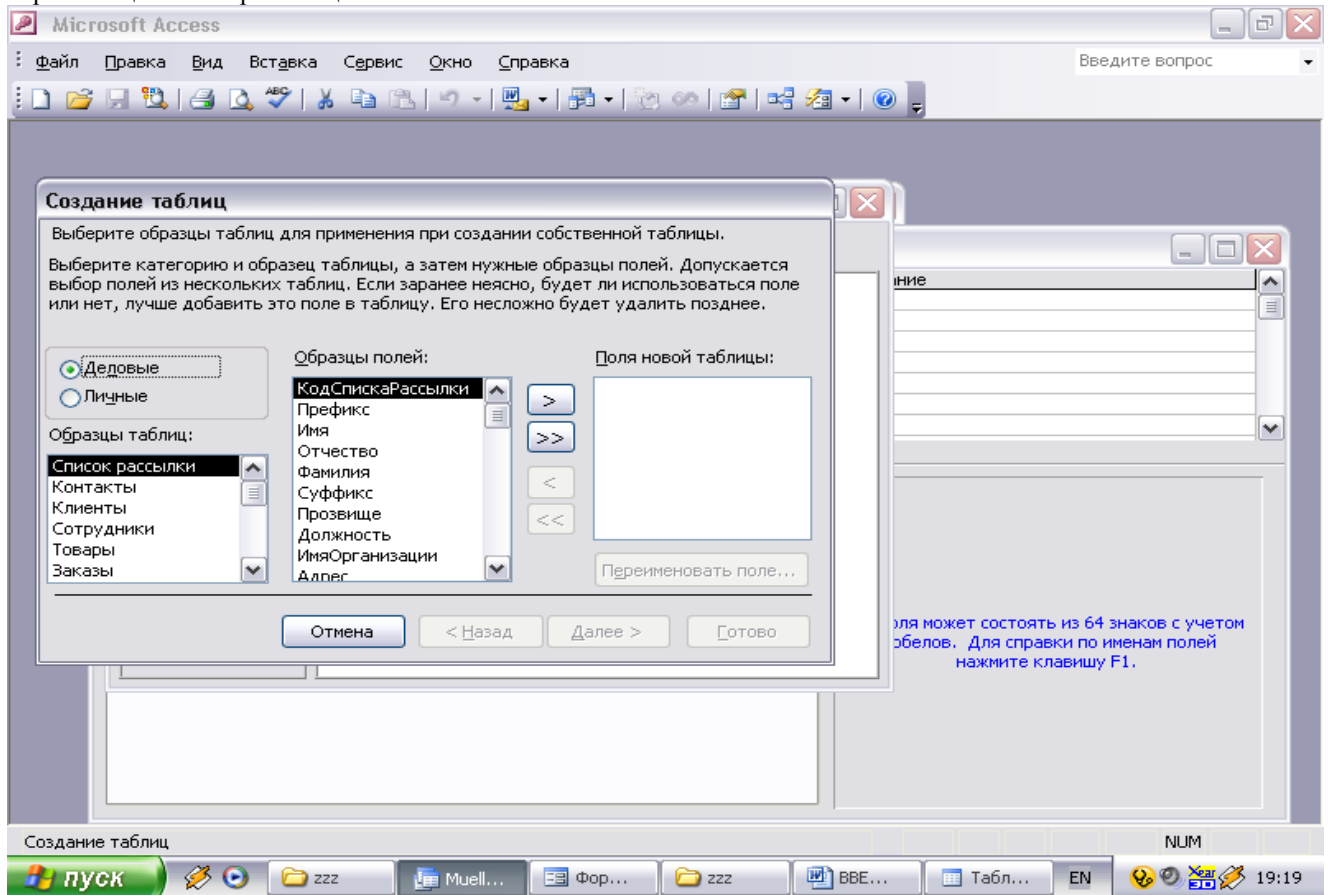
Рис. 22. Таблица в режиме просмотра данных

Если потребуется изменить структуру таблицы, это можно сделать в режиме Конструктор. Для этого нужно выделить таблицу в окне База данных (рис. 19) и щелкнуть на кнопке Конструктор. Создание таблиц в окне конструктора предполагает наиболее широкие возможности по определению параметров созданных таблиц. Для того чтобы перейти в конструктор таблиц выбираем из списка возможных вариантов в окне диалога нашей таблицы опцию конструктор и нажимаем кнопку ОК. В результате открывается окно конструктора таблицы.

В верхней части окна диалога находится таблица, которая содержит следующие атрибуты созданной таблицы: наименование поля, тип данных и описание. Кроме этих основных атрибутов каждое поле таблицы обладает дополнительными свойствами, отображенными в нижней части конструктора и определенными условиями ввода данных. Наименование каждого из полей выбирается произвольно, но таким образом чтобы отразить характер информации, которая будет храниться в данном поле. Тип поля определяется типом данных.

#### **Создание таблицы с помощью мастера.**

M.S.Access содержит ряд таблиц которые мы можем использовать в качестве прототипов требуемых нашей таблицей. При использовании мастера мы можем не только сэкономить время, но и обеспечить стандартные имена и типы данных полей таблиц. Для того чтобы вызвать мастера для создания таблицы в окне диалога новая таблица выбираем опцию мастер таблиц.



Откроется окно диалога, в его левой части находится список образцов таблиц, а рядом список образцов полей, содержащих предлагаемые образцы полей для выбранной таблицы. Сначала нам необходимо из списка образцы таблиц выбрать прототип таблицы, которая похожа на создаваемую нами таблицу. Затем из списка образцы полей выбираем поля таблицы и размещаем их в списке полей новой таблицы. Для выбора полей используем кнопки со стрелками, которые расположены правее списка образцы полей. Используя кнопку переменное поле можно в случае необходимости изменить наименование полей.

[>] – Выделенные в списке образцы полей поле добавляется в список полей создаваемой таблицы.

[>>] – Все поля из списка образцы полей добавляются в список полей созданной таблицы.

[<] – Удаление из списка полей созданной таблицы поля выделенного в списке полей новой таблицы.

[<<] – Удаляем все элементы из списка полей новой таблицы.

На следующем шаге создание таблицы мы задаём имя таблицы и определяем ключевые положения.

#### **Создание межтабличных связей**

Эта операция выполняется с помощью мыши в окне Схема данных кнопкой на панели инструментов в окне базы данных или командой



(рис.24), которое открывается Сервис – Схема данных.

Одновременно с окном Схема данных открывается окно Добавление таблицы, в котором нужно выбрать таблицу для

включения их в структуру межтабличных связей. Если окно не открылось, его можно открыть командой Добавить таблицу из контекстного меню после щелчка правой кнопкой мыши внутри окна Схема данных. Связь между двумя таблицами устанавливается перетаскиванием поля одной таблицы на одноименное поле другой таблицы.

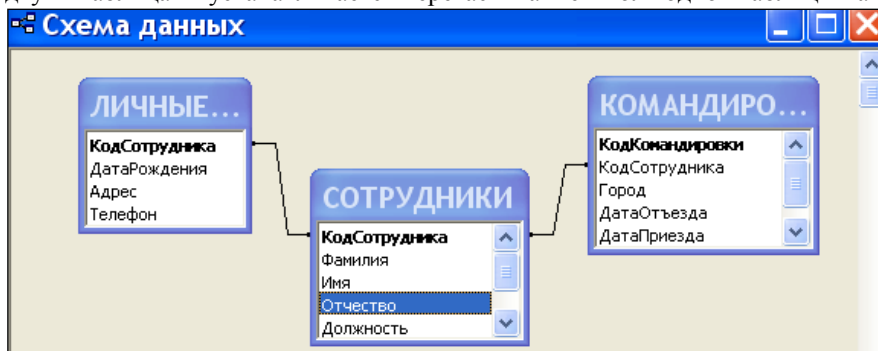
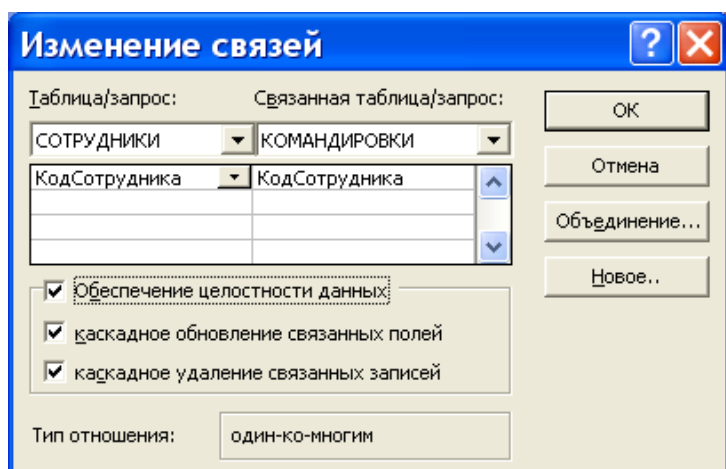


Рис. 24. Связанные таблицы в окне Схема данных

Образовавшаяся связь отображается в виде линии, соединяющей два поля разных таблиц. При этом одна из таблиц считается главной, а другая – связанной. Главная – это та таблица, которая участвует в связи своим ключевым полем (название этого поля на схеме данных отображается жирным шрифтом). Например, в отношении СОТРУДНИКИ – КОМАНДИРОВКИ главной является таблица СОТРУДНИКИ.

При создании связей между таблицами следует позаботиться об обеспечении целостности данных. Примером нарушения целостности данных может служить следующая ситуация. Если в таблице СОТРУДНИКИ удалить запись

для одного из сотрудников, но не сделать этого в таблице КОМАНДИРОВКИ, то получится, что согласно таблице КОМАНДИРОВКИ некий сотрудник, имеющий только код, ездил в командировки. Узнать по коду, кто этот сотрудник, будет невозможно – произошло нарушение целостности данных.



Настройка связи с целью защиты целостности данных выполняется в окне Изменение связей (рис. 25). Если установлен только флажок Обеспечение целостности данных, то удалять данные из ключевого поля главной таблицы нельзя. Если вместе с ним включены флажки Каскадное обновление связанных полей и Каскадное удаление связанных записей, то операции редактирования и удаления данных в ключевом поле главной таблицы

разрешены, но сопровождаются автоматическими изменениями в связанной таблице.

Рис. 25. Диалоговое окно Изменение связей

Окно Изменение связей открывается, если в окне Схема данных выделить линию, соединяющую поля двух таблиц, щелкнуть на ней правой кнопкой мыши и открыть контекстное меню связи (рис. 26), после чего выбрать в нем пункт Изменить связь. Здесь же можно выбрать команду Удалить для удаления существующей связи.

Рис. 26. Контекстное меню связи в окне Схема данных



### Модификация структуры таблицы.

Если мы описали всю структуру таблицы, не сделав при этом ни одной ошибки, нам необходимо сохранить структуру созданной таблицы. Однако вполне вероятно, что при создании структуры мы могли допустить ошибки. MS Access представляет нам средство для исправления ошибок:

- Изменение наименования поля и его типа.
- Вставка пропущенного поля.
- Удаление ошибочно вставленного поля.

Изменение порядка следования полей в таблице.

Для модификации структуры таблиц, входящих, в базу данных, устанавливаем в окне базы данных указатель на

модифицируемую таблицу и нажимаем кнопку конструктор. На экране откроется окно конструктора, содержащее структуру выбранной таблицы.

### ***Изменение порядка следования полей.***

Предположим, что мы, хотели бы перенести одно из полей на несколько полей выше или наоборот ниже, чем оно находится сейчас. Для этого выполняем следующие действия:

1. Слева от имен полей находится область выбора полей. Переходим на строку с описанием поля, которое хотим перевести и нажимаем на эту область. При этом вся строка будет выделена.
2. Теперь нажимаем на область выбора поля и не отпускаем кнопку мыши. При этом под курсором мыши появится пунктирный прямоугольник.
3. Переносим всю строку так, чтобы она оказалась прямо над тем полем, перед которым мы хотим расположить её.
4. Отпускаем кнопку мыши.

Поле теперь будет находиться на новом месте.

### ***Прибавление нового поля.***

Если мы забыли создать поле, выполняем команду вставка строки. Новая строка будет вставлена над текущей строкой, то есть той строкой, на которой мы сейчас находимся. Вводим в этой строке имя поля и типы данных.

### ***Индексы.***

Одним из основных требований является возможность быстрого поиска требуемых записей среди большого объёма информации. Индексы представляют собой наиболее эффективные средства, которое позволяет ускорить поиск данных в таблицах, по сравнению с таблицами, не содержащими индексы. В зависимости от количества полей использованных в индексе различают простые и составные индексы. В M.S.Access допускается создание произвольного количества индексов. Индексы создаются при сохранении макета таблицы и автоматически обновляются при вводе и изменении записи. Мы можем в любое время добавить новые или же удалить ненужные индексы в окне конструктора таблиц. Важной особенностью индексов является то, что мы можем использовать индексы для создания первичного ключа. Первичный ключ индексируется автоматически. В этом случае индексы должны быть уникальными.

#### ***Создание индекса для поля.***

Для создания простого индекса в столбце свойства поля выбираем команду индекс поля позволяющую ускорить выполнение поиска и сортировку записей по одному полю таблицы. Индексированное поле может содержать, как уникальное, так и повторяющиеся значение. Например, в таблице телефонно-адресная книга M.S.Access автоматически создала индекс по полю, которое содержит уникальное значение код. Однако может создать индекс по полю фамилия, который может содержать повторяющиеся значения. Создание простого индекса предлагает выполнение следующих действий:

В окне конструктора таблиц выбираем в верхней половине окна поле, для которого создаётся индекс. В нижней половине окна, в столбце свойства выбираем команду индексное поле. Затем выбираем одно из следующих значений Да (допускаются совпадения) или Да (совпадения не допускаются). Значение Да (совпадения не допускаются) обеспечивает уникальность каждого значения данного поля.

Примечание : не допускается создание индексов для образования полей MEMO или OLE.

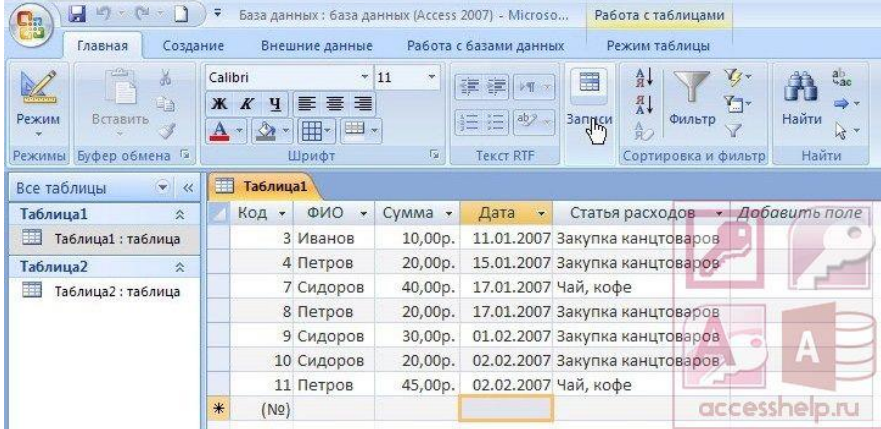
#### ***Создание составного индекса.***

Для этого:

1. В окне конструктора открываем таблицу, для которой создаётся составной индекс, для этого в окне базы данных устанавливаем указатель на данную таблицу и нажимаем кнопку конструктора.
2. Нажимаем кнопку индекса на панели инструментов. На экране открывается окно диалога.
3. В окне диалога вводим имя индекса в первом пустом поле столбца индекс. В качестве имени индекса можно использовать имя одного из полей, включенных в индекс или любое допустимое имя.
4. В столбце имя поля в той же строке нажимаем кнопку раскрытие списка и выбираем первое имя индекса.
5. В столбце имя поля в следующей строке выбираем следующее имя поля индекса. Определяем таким образом остальные поля индекса. Индекс может включать до десяти полей.
6. Закончив выбор полей для индекса нажимаем кнопку закрытие окна, расположенную в заголовке окно диалога. По умолчанию задаётся порядок сортировки по возрастанию. Для сортировки конкретного поля по убыванию выбираем в столбце порядок сортировки строки с выбранным полем значением по убыванию.

## **Фильтрация, сортировка и поиск данных в Microsoft Access**

Когда записей в таблице становится достаточно много, большое значение приобретает возможность просматривать их выборочно и в определенном порядке. С помощью инструментов страницы «ленты» «Главная» можно скрыть некоторые столбцы и записи таблицы, а также отсортировать записи по одному или нескольким полям. Для того чтобы спрятать на время выделенный столбец или группу столбцов, раскройте раздел «ленты» «Записи». Нажмите кнопку «Дополнительно» и выберите в открывшемся меню команду «Скрыть столбцы» Для того чтобы снова показать скрытые столбцы, раскройте раздел «ленты» «Записи», нажмите кнопку «Дополнительно» и выберите в меню команду «Отобразить столбцы...».

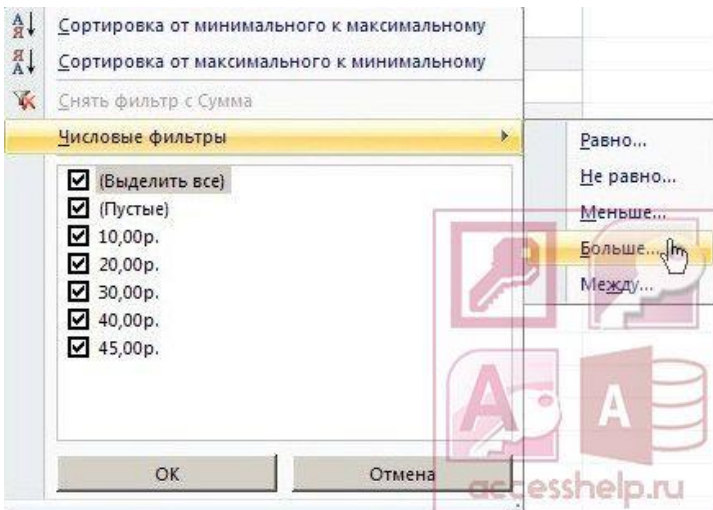


В открывшемся окне диалога необходимо отметить галочками те столбцы, которые должны отображаться в представлении данных таблицы. Сделанные в окне изменения сразу отображаются в представлении. Отметьте галочкой скрытый столбец и нажмите кнопку «Заккрыть».

Для того чтобы отображать только записи таблицы, отобранные по определенному признаку, можно использовать фильтр, то есть условие,

задаваемое для поля. После применения фильтра записи, не удовлетворяющие заданному условию, будут скрыты. Для назначения фильтра выделите один из столбцов, например, столбец «Сумма» и нажмите кнопку «Фильтр» в разделе «Ленты» «Сортировка и фильтр».

В списке значений поля уберите значения, которые не должны отображаться путем снятия галочек напротив каждого из значений. Для применения фильтра нажмите кнопку «ОК».

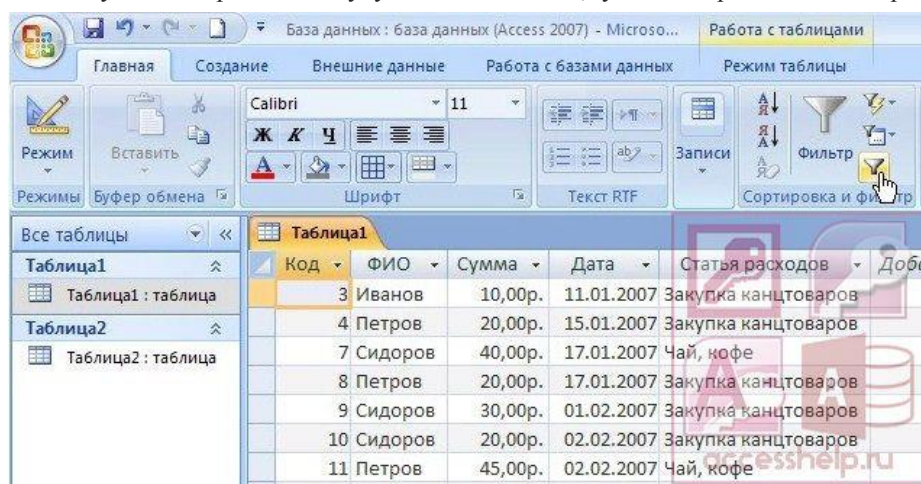


После применения нового фильтра представление данных обновляется. Поле с установленным фильтром маркируется специальным значком в заголовке. Для временного снятия фильтра с выделенного поля нажмите кнопку «Удалить фильтр». При снятии фильтра представление тоже обновляется. Повторно нажмите на эту же кнопку, чтобы включить фильтр снова.

Для окончательной очистки фильтра щелкните по стрелке в заголовке столбца таблицы и выберите команду «Снять фильтр с Сумма». Чтобы задать условие для фильтра в виде логического выражения, вызовите снова окно настройки сортировки и фильтров.

доступных логических операторов будет зависеть от типа данных поля. Раскройте список «Числовые фильтры». Выберите в списке фильтр «Больше...» и, задав величину для сравнения, нажмите кнопку «ОК».

Теперь на экране будут отражаться только записи, содержащие в поле «Сумма» значения не менее двадцати. К отдельному полю можно применить только один простой фильтр, однако для каждого поля можно определить свой. В этом случае отображаться будут только записи, удовлетворяющие одновременно всем условиям. Более сложные



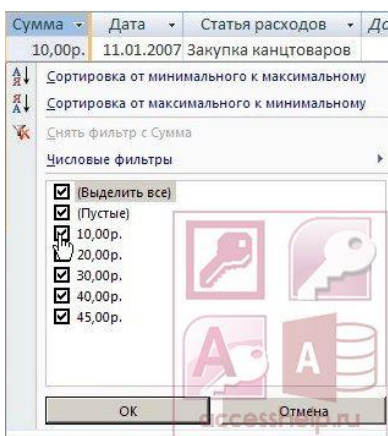
фильтры, которые могут включать значения нескольких полей одновременно, задаются с помощью команды «Параметры расширенного фильтра».

Параметры фильтра действуют до закрытия таблицы и не теряются при переключении режима отображения.

Если сохранить таблицу, примененный фильтр будет доступен и при следующем её открытии. Включать и выключать сохраненный фильтр можно с помощью кнопки панели навигации «Без фильтра» — «С фильтром», надпись на кнопке отражает текущее

состояние поля. Нажмите на кнопку с надписью «С фильтром», чтобы отключить фильтр.

Для упорядочивания записей в представлении данных таблицы применяется сортировка по текстовым, числовым значениям или по дате. Сортировка позволяет пользователям находить нужные данные, не просматривая все данные подряд. Записи можно сортировать по одному или нескольким полям. Для установки порядка сортировки по возрастанию значений поля нажмите кнопку «По возрастанию». Для сортировки по убыванию значений нажмите кнопку «По убыванию». Для отмены сортировки по всем полям сразу нажмите кнопку «Очистить все сортировки». В Microsoft Office Access записи сортируются по возрастанию или по убыванию без учета регистра. Порядок сортировки чисел, текста и специальных знаков зависит от выбранных параметров языка и стандартов



для данного компьютера. При сортировке по нескольким полям сразу результат будет зависеть от порядка применения режима сортировки к этим полям. В случае, когда надо найти одну или несколько записей по фрагменту текста, входящему в одно из их полей, можно воспользоваться стандартным средством поиска Microsoft Office Access. Нажмите кнопку «Найти» на странице «ленты» «Главная».

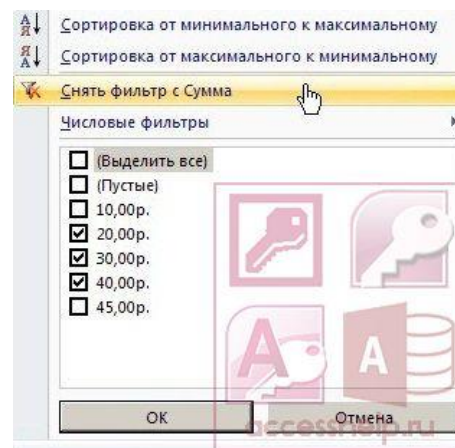
В окне диалога «Поиск и замена» необходимо задать образец для поиска и выбрать нужные параметры поиска, например, задать область совпадения образца с любой частью поля, просмотр по всем записям. Для перехода на следующую найденную запись нажмите кнопку «Найти далее».

После завершения поиска закройте окно. Предположим, у нас есть форма, содержащая подчиненную форму, в которой отображается много записей, и вы хотели бы интерактивно задать в главной форме какое-нибудь условие для отбора интересующих записей в подчиненной форме, чтобы упростить их визуальный анализ.

Это несложно сделать с помощью применения фильтра для источника данных формы (в данном случае — подчиненной). Фильтр можно применить двумя способами:

- с помощью интерфейса Access (соответствующих кнопок на панели инструментов Режим формы (Form View), команд стандартного меню и контекстного меню поля формы);
  - с помощью изменения значений свойств формы.
- Первый способ применения фильтра в форме во многом аналогичен подобным действиям в режиме Таблицы

Код	ФИО	Сумма	Дата	Статья расходов
4	Петров	20,00р.	15.01.2007	Закупка канцтоваров
7	Сидоров	40,00р.	17.01.2007	Чай, кофе
8	Петров	20,00р.	17.01.2007	Закупка канцтоваров
9	Сидоров	30,00р.	01.02.2007	Закупка канцтоваров
10	Сидоров	20,00р.	02.02.2007	Закупка канцтоваров
*	(№)			



## Работа с формами

### Использование форм для ввода и редактирования данных.

Формы являются мощным и гибким средством представления информации. В M.S.Access существует две формы отображения содержимого в таблице:

1. В виде таблицы.
2. В виде формы.

В M.S.Access формат отображаемых в виде формы называется формой.

### Простейший способ создания формы.

Для создания простейшей формы в M.S.Access достаточно выполнить следующие действия:

1. Открываем окно базы данных;
2. В окне базы данных переходим на вкладку таблица;
3. Устанавливаем указатель на таблицу, для которой хотим создать экранную форму;
4. В меню вставка выбираем команду авто форма или нажимаем кнопку **новый объект** на панели инструментов и из раскрывшегося списка выбираем опцию автоформа. В результате на экране появится, готовая к использованию форма.

### Кнопки перемещения по записям.

В форме для каждой записи в таблице отведена своя страница. Чтобы увидеть следующую запись, надо нажать кнопку Page Down, чтобы вернуться на предыдущую запись Page Up. Мы можем перейти как в соседние, так и в отдалённые записи, используя кнопки перемещения, расположенные в самом низу формы:

- Переход на первую запись.
- Переход на одну запись назад.
- Переход на одну запись вперёд.
- Переход на последнюю запись.
- Переход на чистую страницу, где можно ввести новую запись.

Для перехода из формы в режим таблицы в меню вид выбираем команду режим таблицы. Для перехода из формы в режим таблицы также можем нажать кнопку вид на панели инструментов и из раскрытого списка выбираем пункт режим таблицы.

### Альтернативные способы создания формы.

Кроме просмотра простейшего способа создания формы, в M.S.Access имеется ряд альтернативных способов.

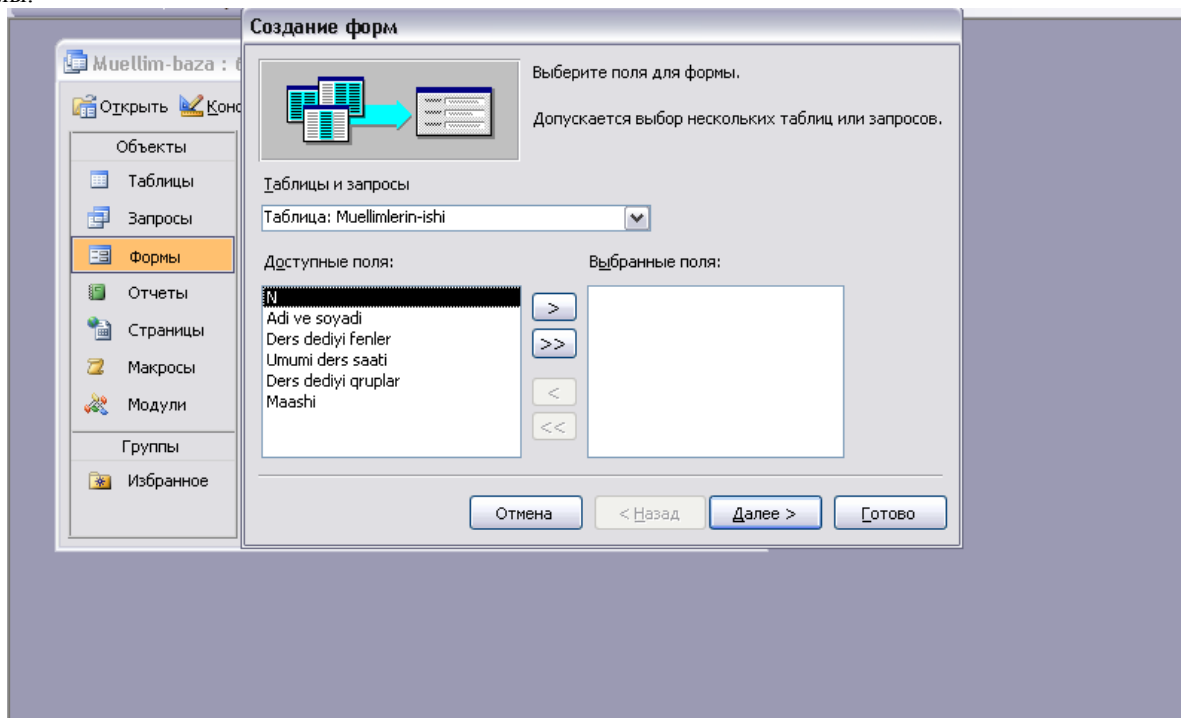
- Конструктор формы – в нём самой разрабатываемой собственные формы с заданными свойствами для просмотра, ввода и редактирования данных.
- Мастер формы – позволяет достаточно быстро создать форму на основе выбранных полей.
- Автоформа столбец – позволяет автоматически создать формы с полями, расположенными в один или несколько столбцов.
- Ленточные автоформы – позволяют автоматически создать ленточные формы.

- **Табличные автоформы** – позволяют автоматически создать табличные формы.
- **Диаграмма** – позволяет создать форму с диаграммой.
- **Свободная таблица** – позволяет создать форму со свободной таблицей M.S.Access.

### Ленточные автоформы.

#### **Использование мастера для создания формы.**

Для запуска мастера форм в окне диалога новая форма. Выбираем из списка варианты создание формы мастер форм.



После того как мы запустим мастер форм достаточно ответить на вопросы текущего экрана мастера и нажать кнопку далее для перехода к следующему экрану. Если мы допустим ошибку или решим изменить ответы предыдущего экрана нажимаем кнопку назад, для отказа от продолжения – кнопку отмена. Нажимаем кнопку раскрытие списка таблицы запросов, из списка выбираем таблицу, для которой создаётся форма. При этом в списке допустимые поля появляется перечень всех полей выбранной таблицы. Из данного перечня переносим в список выбранные поля, поле которое хотим поместить в создаваемую форму. Мастер форм позволяет создавать форму, используя имя не только одной таблицы, но и из нескольких связанных таблиц. В этом случае после выбора полей из одной таблицы, выбираем из списка таблиц базы данных вторую таблицу и переносим требуемые поля в список выбранные поля. Завершив, нажимаем далее. Появляется диалоговое окно, где можно изменить внешний вид формы. Выбираем нужную опцию, нажимаем далее, потом выбираем стиль нашей формы и ОК.

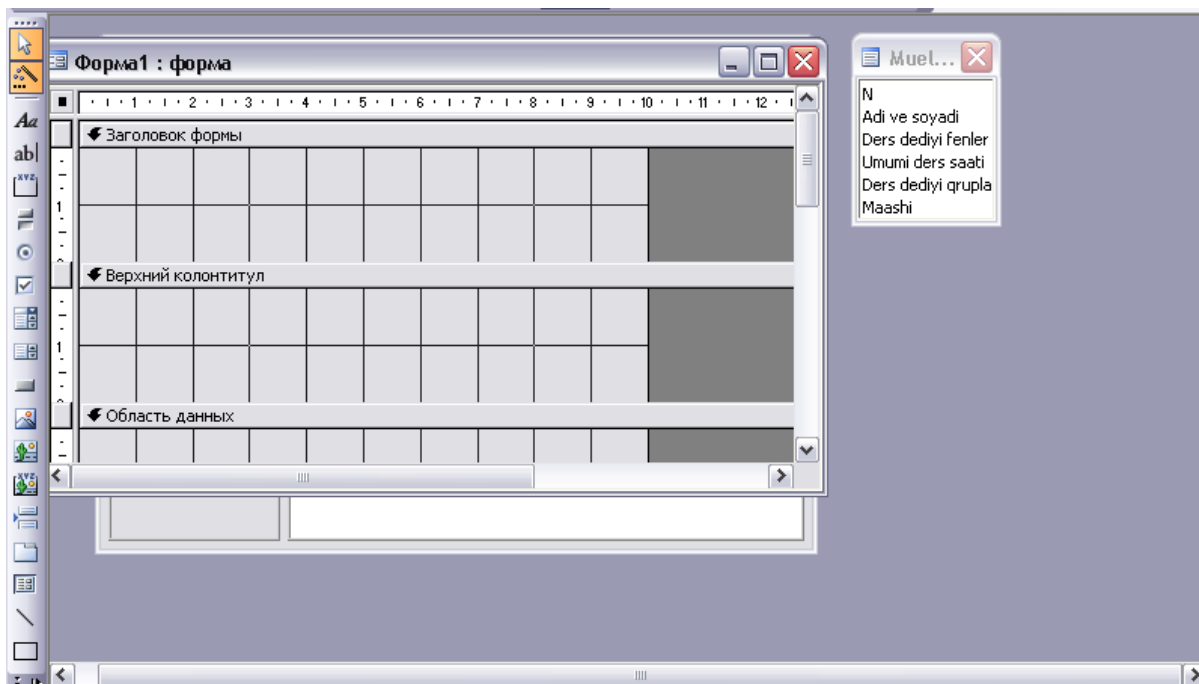
#### **Создание формы в конструкторе форм.**

Мы рассмотрели создание и использование автоформ. Они позволяют быстро создать форму для ввода и просмотра данных, но для создания сложных и более удобных для конечного пользования форм, этих средств не достаточно. Любая форма в M.S.Access состоит из объектов форма, которая имеет характерные для них свойства. Процесс создания формы состоит из размещения объектов в форме и определяет для них свойства связанных с ними событий и выполняемых действий.

#### **Окно конструктора форм.**

Для открытия окна конструктора форм, при создание новой формы, выполняем следующие действия:

- В окне базы данных переходим на вкладку форма;
- Нажимаем кнопку создать;
- В открывшемся окне диалога новая форма, в списке способов создания форм, выбираем конструктор;
- На экране открывается окно конструктора форм;
- Некоторые кнопки на панели инструментов в верхней части экрана кажутся знакомыми. Но кнопки расположенные в середине панели инструментов являются новыми;
- Вторая панель инструментов расположена под первой предназначенной для обработки текста. Эта панель формат с помощью, которой редактируется шрифт. Ниже двух панелей инструментов находится горизонтальная линия, в левой части экрана есть вертикальная линия.
- В правой части экрана находится перемещаемая панель элементов. Мы можем перемещать ее.

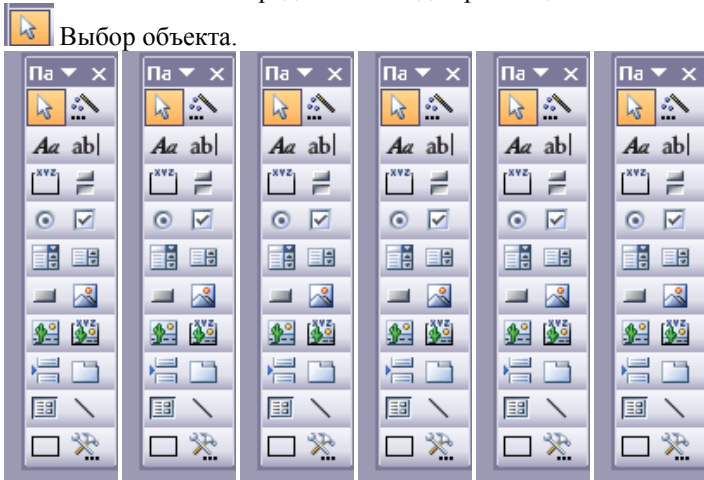


### Область форма.

Пре открытия окна конструктор содержит одну область – область данных. Помимо этого форма может содержать область заголовка, примечание, нижний и верхний колонтитул. Для добавления этих областей в столбце вид выбираем команды заголовок - примечание - формы и вид колонтитулов и получаем нужный вид.

### Панель элементов.

Панель элементов не предназначена для размещения объекта форма:



Выбор объекта.

– Построитель элемента.



– Надпись позволяет размещать в форме текст.



– Поле осуществляет: отображение, ввод, изменение данных; вывод результата вычислений; приём данных при их вводе.

– Группа параметры используется для размещения флажков.



– Выключатель



– Переключатель



– Флажок.

– Поле со списком.

М – Список.

М – Кнопка позволяет делать разнообразные -----

М – Рисунок.

М – Свободная рамка объекта.

М – ----- рамка объекта.

М – Конец страницы.

М – Набор вкладок используемых для создания нескольких вкладок.

М – Подчинённая форма.

### ОБЛАСТЬ ФОРМЫ

При открытии окна конструктора формы мы можем увидеть одну область – область данных. Помимо этого форма может содержать область заголовка, примечания, верхний и нижние колонтитулы. Для добавления этих областей используется команда ВИД→ЗАГОЛОВОК (или ПРИМЕЧАНИЯ, КОЛОНТИТУЛ). Для изменения размеров какой либо области устанавливаем указатель мыши на (нижнюю) верхнюю часть границы между областями. Когда указатель мыши принимает вид двухсторонней стрелки, нажимаем кнопку мыши и перемещаем границу.

### РАЗМЕЩЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

Размещение текста в экранной форме осуществляется с помощью инструмента НАДПИСЬ, которая находится на панели элементов. Для размещения текста в форме выполняем следующие действия:

1. Выбираем инструмент НАДПИСЬ на панели элементов. Если эта панель отсутствует то выполняем следующую команду ВИД→ ПАНЕЛЬ ИНСТРУМЕНТОВ →ПАНЕЛЬ ЭЛЕМЕНТОВ
2. Устанавливаем указатель мыши на место предполагаемое для размещения текстового объекта.
3. Закончив ввод текста нажимаем клавишу Enter.
4. Выделяем созданный объект.
5. Используя панель инструментов форматирования задаём для него ТИП ШРИФТА, РАЗМЕР ШРИФТА и т.д

### СОЗДАНИЕ РАСКРЫВАЮЩЕГОСЯ СПИСКА.

Поля со списком широко используется в MS ACCESS и представляет нам возможность выбрать из списка допустимых значений и непосредственно ввести в поле значения, которые отсутствуют в списке. Список значений хранится в свернутом виде до тех пор, пока мы не раскроем его, который расположен в правой части объекта. Предположим, что большинство клиентов информация, о которых содержится в таблице «клиенты» проживают в Баку, Гянджа, Шеки. Однако мы не можем пренебрегать потенциальными заказчиками, проживающих в других городах. Для решения данной задачи оптимальным средством являются поле со списком. Для этого:

1. Создаем форму альтернативными методами (мастер, конструктор) для таблицы «клиенты». Размещаем поля.
2. Выделяем поле «город» и нажимаем кнопку мыши. На экране откроется контекстное меню поля.
3. Выбираем ПРЕОБРАЗОВАТЬ ЭЛЕМЕНТ В, а за тем ПОЛЕ СО СПИСКОМ. MS ACCESS выполнит преобразование о чем мы можем судить по кнопке РАСКРЫТИЯ СПИСКА, появился в правой части поля.
4. Теперь нам предстоит определить элементы списка. Для этого вызываем еще раз меню поля «город» и выбираем СВОЙСТВО. На экране появится окно СВОЙСТВА ПОЛЯ «ГОРОД».
5. Переходим на вкладку ДАННЫЕ.
6. Устанавливаем для свойства типа источника строк значение СПИСОК ЗНАЧЕНИЙ.
7. Задаем с помощью свойства источника строк предопределенных значений. Для этого вводим в поле ЭЛЕМЕНТЫ списка используя в качестве разделителя «;» .
8. Устанавливаем для свойства ОГРАНИЧИТЬСЯ СПИСКОМ значение НЕТ.
9. Закрываем окно свойство.

### Создание форм в конструкторе форм.

Процесс по созданию форм может включать в себя все или часть из приведённых ниже процедур.

1. Размещение текста.
2. Размещение полей.
3. Создание управляющих кнопок.
4. Размещение линий, прямоугольников и рисунков.
5. Установка цвета объекта формы.
6. Помещение объекта формы.

При создании формы учитываем следующее, если пользователи привыкли использовать стандартные бланки, формы должны выглядеть так же как эти бланки. Для группировки элементов управления используются линии и прямоугольники. При этом пользователь будет вводить близкие по смыслу данные. Пояснительный текст должен быть так максимально информирован и иметь минимальную длину. Используйте макет ввода для ввода стандартной информации.

Если вам известно точное количество символов, необходимых для ввода номеров телефона используйте маску ввода.

### Постройка формы.

После того как мы рассмотрели общие рекомендации для разработки формы, можно приступить к ее созданию.

Формы позволяют пользователям вводить данные в таблицы базы данных, просматривать и редактировать данные без непосредственного доступа к самим таблицам. Автоформа всегда создается на базе только одной таблицы и содержит все поля этой таблицы. Существует три вида автоформ: в столбец, ленточные и табличные (рис. 27) Для

создания автоформы следует открыть панель Формы в окне База данных и воспользоваться командной кнопкой Создать. В открывшемся диалоговом окне Новая форма выбирается тип автоформы и таблица, для которой эта форма создается. После щелчка на кнопке <ОК> автоформа формируется автоматически.

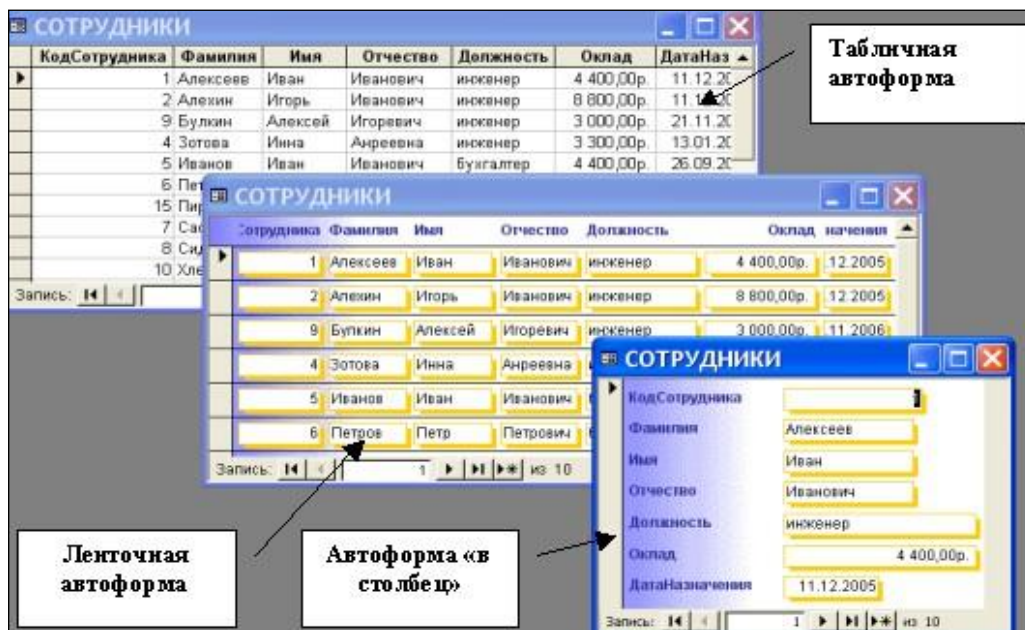


Рис. 27. Три вида автоформ

*Мастер форм* – специальное средство, создающее структуру формы в режиме диалога. В режиме Мастера форм можно создать форму, базирующуюся как на одной, так и на нескольких таблицах. Мастер форм можно запустить из окна База данных щелчком на значке Создание формы с помощью мастера на панели Формы. Создание формы выполняется в несколько этапов. На первом этапе работы Мастера форм выбирают таблицы и поля, которые войдут в будущую форму (рис. 28, а). На втором этапе выбирается внешний вид формы (рис. 28, б). На третьем этапе выбирается стиль оформления формы (рис. 28, в). На последнем этапе выполняется сохранение формы (рис. 28, г).

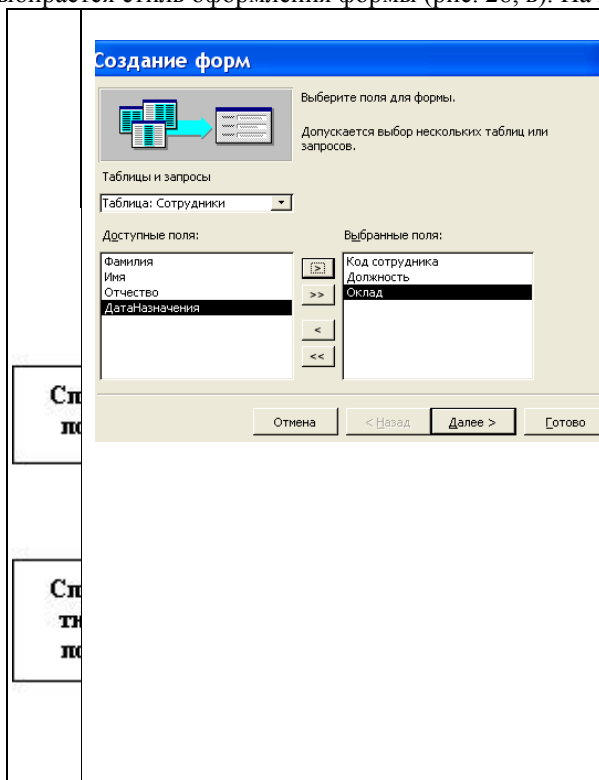


Рис. 28, а. Создание формы в режиме Мастер форм, шаг 1

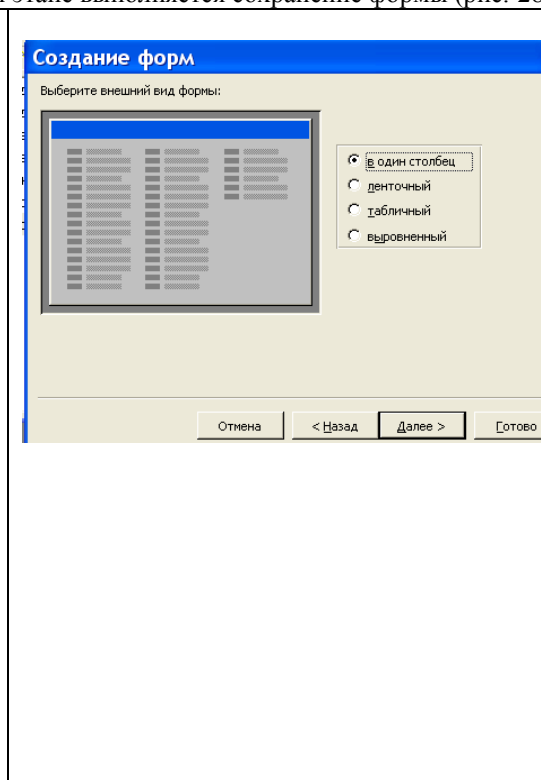


Рис. 28, б. Создание формы в режиме Мастер форм, шаг 2

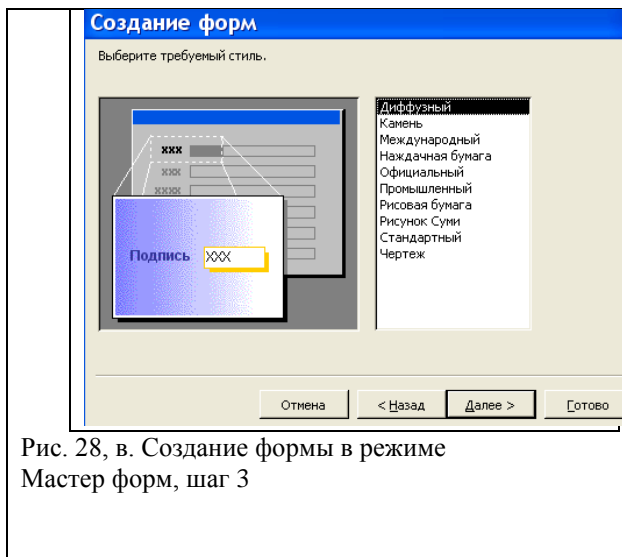


Рис. 28, в. Создание формы в режиме Мастер форм, шаг 3

Рис. 28, г. Создание формы в режиме Мастер форм, шаг 4

*Подчиненная форма* – это форма, находящаяся внутри другой формы. При этом первичная форма называется главной формой. Форма с подчиненной формой создается для таблиц, связанных отношением «один-ко-многим». Например, можно создать форму с подчиненной формой для таблиц СОТРУДНИКИ и КОМАНДИРОВКИ (рис. 29). Главная форма и подчиненная форма связаны таким образом, что *в подчиненной форме выводятся только те записи, которые связаны с текущей записью в главной форме*. Например, когда главная форма отображает данные о конкретном сотруднике, подчиненная форма отображает только те командировки, которые относятся к данному сотруднику.

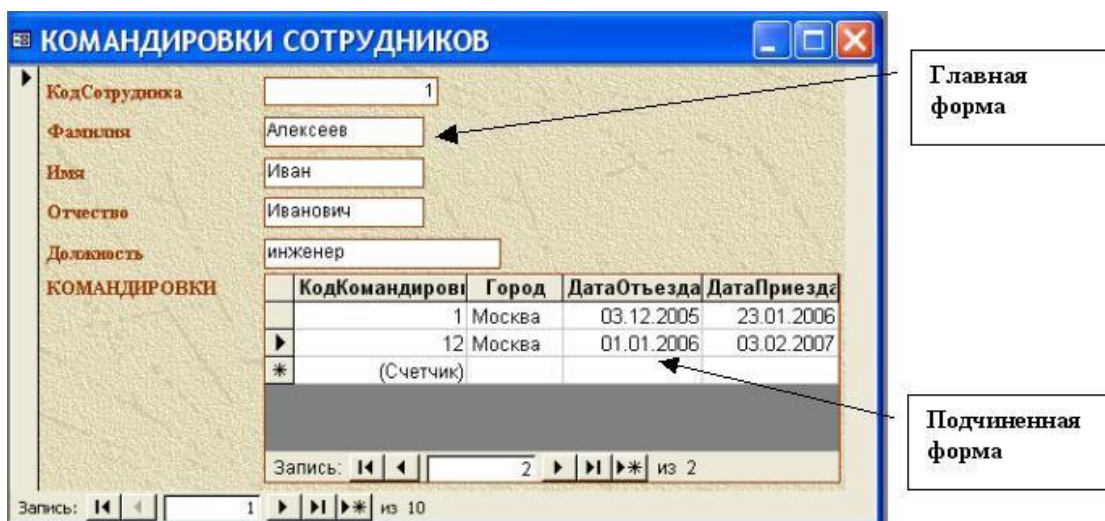
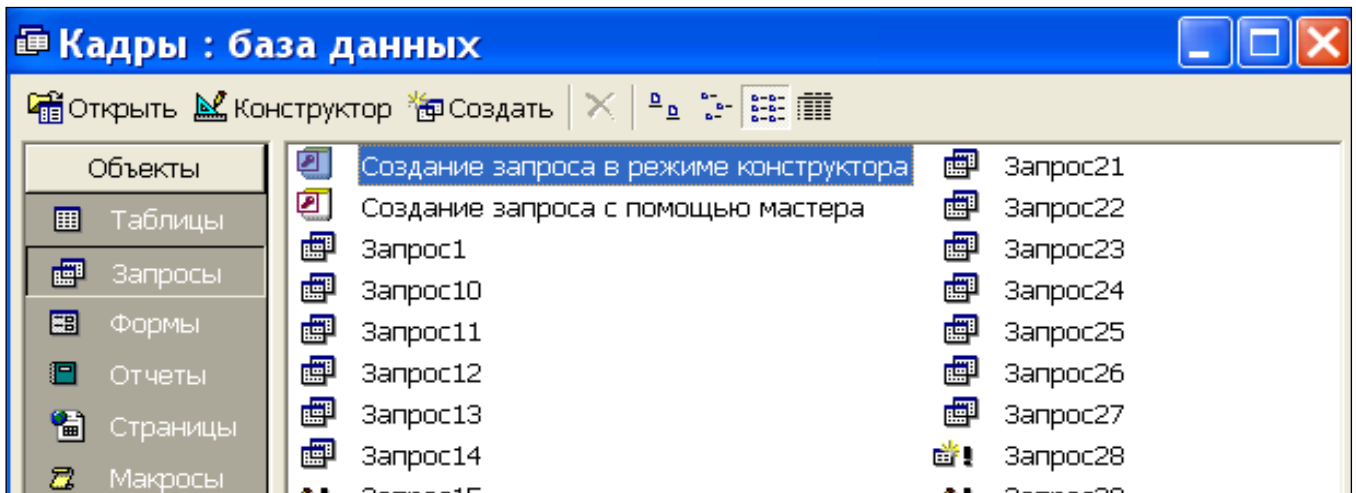


Рис. 29. Пример формы, содержащей подчиненную форму

## Работа с запросами и отчетами

### Запросы на выборку



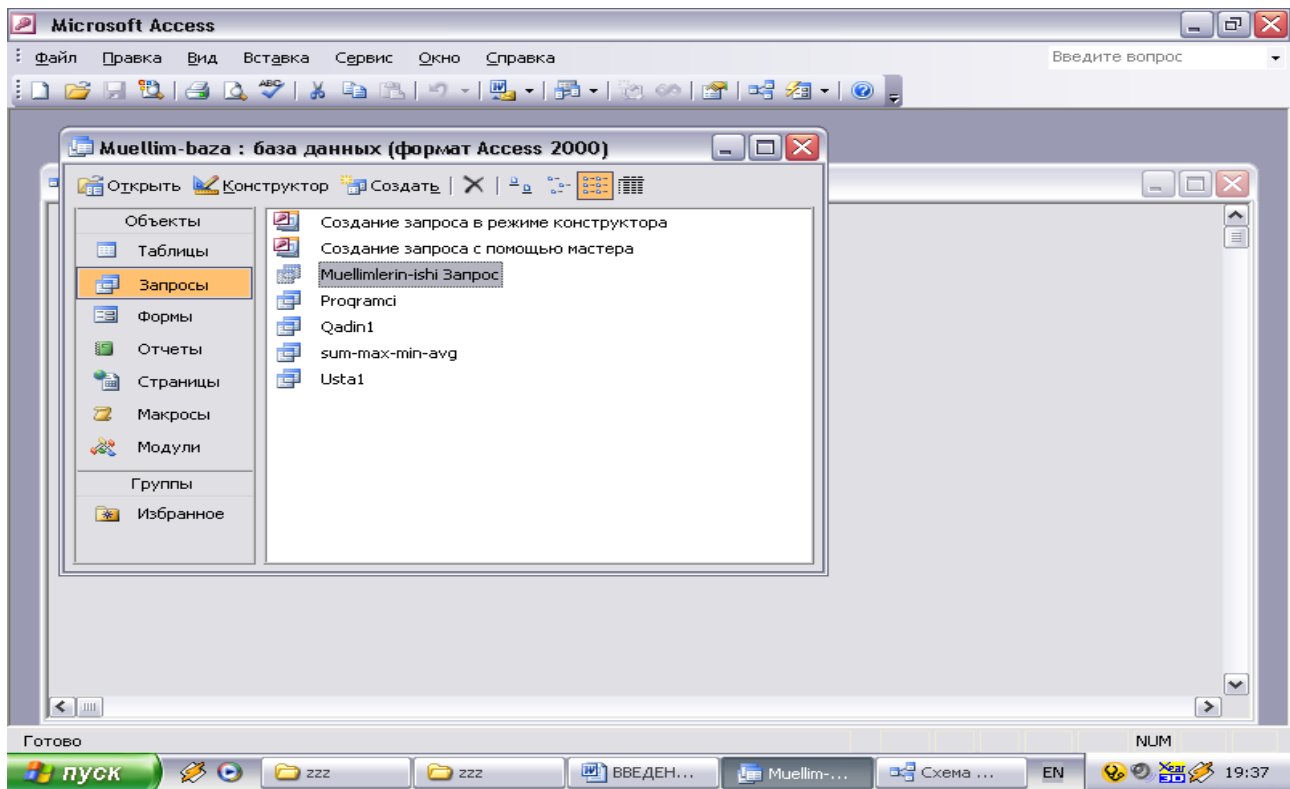
Для получения данных из таблиц базы данных, пользователь должен использовать запросы. Запрос извлекает данные из базовых таблиц и создает на их основе временную результирующую таблицу. При сохранении запроса сохраняется не результирующая таблица, а только алгоритм ее получения. При следующем выполнении запроса мы получим результирующую таблицу, отражающую состояние базы данных на момент выполнения запроса. Если запрос подготовлен и сохранен в базе данных, то для выполнения запроса следует открыть панель Запросы в окне База данных (рис. 31), выбрать запрос и открыть его двойным щелчком на значке запроса. Запрос будет выполнен и откроется результирующая таблица, соответствующая данному запросу.

Запросы можно создавать в режиме Конструктор. Значок Создание запроса в режиме конструктора в окне База данных (рис. 31) открывает окно, называемое *бланком запроса по образцу* (рис. 32). В верхней части бланка отображается структура таблиц, к которым применяется запрос, а нижняя часть разбита на столбцы, соответствующие полям будущей результирующей таблицы запроса. Для включения поля базовой таблицы в результирующую таблицу запроса следует дважды щелкнуть соответствующее поле в верхней части бланка. При этом автоматически заполняется столбец в нижней части бланка. Для вызова конструктора запроса переходим в окне базы данных на вкладку запроса и нажимаем кнопку создать. В окне диалога запроса выбираем опцию конструктор и нажимаем кнопку ОК.

MS Access предложит нам выбрать таблицу или запрос, на основе которой будет осуществляться выбор. Выбираем таблицу, нажимаем кнопку добавить и закрываем окно диалога. Для формирования запроса в окне конструктора запроса необходимо выбрать таблицы, из которых осуществляется выборка и поля результата запроса, указать критерии для выборки группировки и упорядочивания данных.

- Выполняет запрос.
- Добавляет новую таблицу к запросу.
- Выбирает тип запроса.
- Добавляет в бланк запроса строку группа операторов.
- Открывает окно свойства запроса.

В верхней части окна конструктора запросов находится схема данных запроса. Эта схема напоминает схему данных базы данных, в отличие от неё данная схема содержит список таблиц включаемый в запросе и отображающую связи между ними. В нижней части окна располагается бланк запроса. Каждая страница этого бланка выполняет определённую функцию.



- **Поле** – в этой странице помещаются те поля, которые мы используем для создания запроса, каждая в своей ячейке таблицы.
- **Имя таблицы** – эта страница показывает нам, из какой таблицы выбрано данное поле.
- **Сортировка** – в этой странице мы указываем тип сортировки информации по возрастанию или по убыванию.
- **Вывод на экран** – если мы хотим чтобы M.S.Access показал информацию найденную в поле, отмечаем эту ячейку, чтобы установить флажок просмотра поля, если же поле использовать только для задания условия выбора данных которые возвращают нам наш запрос, оставляем его пустым.
- **Условия отбора** – это наиболее важная часть бланка запроса. В этой строке мы вводим ограничения поиска. Задавая определённые условия, которые называются критерием поиска.

### ***Выбор данных из таблиц с помощью запросов.***

#### ***Создание простого запроса.***

Предположим, что мы решили позвонить всем клиентам и сообщить им о презентации новых товаров, номера их телефонов находятся в одном из столбцов таблицы, содержащей всю информацию о наших клиентах. Для создания запроса выполняем следующие действия:

1. В окне базы данных переходим на вкладку запросы и нажимаем кнопку создать.
2. Открывается окно диалога добавить таблицу, в котором выбираем таблицу и нажимаем кнопку добавить запрос в окне диалога.

#### ***Добавление полей в бланк запроса.***

Для выбора полей, в которых должны присутствовать результирующие таблицы нам необходимо отобразить их в бланк запроса. Для добавления в таблицу отдельных полей мы можем выбрать поле таблицы на схеме данных и дважды нажать кнопку мыши. Выбранное поле будет вставлено в следующий доступный столбец в строке поле бланка запроса. В строке имя таблицы сразу же появляется имя таблицы, а позиция выводимая на экран будет помечена.

#### ***Удаление полей из бланка запроса.***

Если мы по ошибке перемещаем в бланк запроса несколько лишних полей и хотим начать всё с начала, то в столбце правка выбираем команду очистить бланк, которая удаляет все поля из бланка запроса. Для удаления данного поля из запроса нажимаем на область выбора столбца, затем нажимаем кнопку DELETE и поле исчезает.

#### ***Изменение порядка панели.***

Порядок в панели бланка запроса определяет параметры появления их в результирующей таблице. Для того чтобы изменить расположение поля в этом списке выполняем следующие действия:

1. Устанавливаем указатель мыши на область выбора столбца, который располагается прямо над названием поля тогда указатель изменяет вид на стрелку, нажимаем кнопку, чтобы выделить столбец.
2. Нажимаем и удерживаем кнопку мыши в этом положении. На кнопке указателя появляется прямоугольник.
3. Перетаскиваем столбец в требуемом направлении. Толстая вертикальная линия покажет нам текущее положение.
4. Отпускаем кнопку, когда толстая вертикальная линия окажется в требуемом месте. Поле будет перемещено в нужное место.

#### ***Сортирование результатов выбора.***

Сортирование результатов выбора в M.S.Access выполняется достаточно просто. Мы можем сортировать выборки по одному или нескольким полям, так как мы это делали при сортировке таблиц. Порядок сортировки записей

результатирующей таблицы определяется использованием следования полей в бланке записи или критерием упорядочивания отдельных полей. Чтобы отсортировать данные следующего поля переходим на строку сортирование требуемого поля и из раскрывшегося списка выбираем значение по возрастающей или по убывающей. Если мы решили отказаться от сортировки поля, выбираем значение отсутствия. Для сортировки по нескольким полям, переходим на сортирование полей и устанавливаем для них критерии сортировки.

Одним из основных назначений базы данных является быстрое получение информации по интересующему нас вопросу. Вопросы, формулируемые по отношению к базе данных, называются **запросами**. Что же такое запрос по образцу? **Запрос по образцу** это интерактивное средство для выбора данных из одной или нескольких таблиц. При форматировании запроса нам необходимо указать критерии выборки записей в исходной таблице. При этом вместо того, чтобы печатать предложение на специальном языке, мы должны просто заполнить бланк запроса, который располагается в окне конструктора запроса.

Для создания простейших запросов мы можем использовать мастера запросов, который у нас последственно запрашивает наименование таблиц, используемых в запросе. Перечень полей таблиц и некоторые дополнительные параметры.

#### Создание запросов с помощью мастера.

Для создания такого запроса в окне базы данных переходим на вкладку запросы и нажимаем кнопку создать открывается окно диалога, там можем выбрать опцию создание запроса с помощью конструктора запросов или один из мастеров создания запроса. Для создания с помощью мастера выбираем опцию простой запрос. В этом случае нам будет предложено указать список таблиц и их полей. И на основании этих данных будем создавать требуемый запрос. Выбираем опцию произвести и нажимаем кнопку ОК. Открывается окно мастера, в котором можем выбрать требуемые нам поля из одной или нескольких таблиц. Если мы выбрали более одной таблицы мастер автоматически связывает таблицы на основе сходства данных. После выбора полей нажимаем кнопку далее.

#### Упорядочение записей в результирующей таблице

Если необходимо, чтобы данные, отобранные в результате работы запроса, были упорядочены по какому-либо полю, применяют сортировку. В нижней части бланка конструктора запроса имеется специальная строка Сортировка. При щелчке на этой строке в любом столбце таблицы появляется раскрывающийся список, в котором можно выбрать метод сортировки: по возрастанию или по убыванию. На рис. 33 задана сортировка по полю Фамилия.

Рис. 33. Задание условий сортировки записей и показа полей в результирующей таблице запроса

Поле:	Фамилия	Имя	Отчество	Город	ДатаОтъезда
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ	КОМАНДИРОВКИ	КОМАНДИРОВКИ
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				"Москва"	
или:					

#### Управление отображением данных в результирующей таблице

В нижней части бланка запроса по образцу имеется строка Вывод на экран. Возможна ситуация, когда какое-то поле необходимо включить в запрос, но при этом запретить вывод на экран его содержимого. В таких случаях нужно сбросить флажок Вывод на экран в соответствующем поле. На рис. 33 поле Город в результирующей таблице не будет показываться, хотя оно содержится в этой таблице. Необходимость включения этого поля в запрос обусловлена тем, что по нему задано условие отбора данных.

#### Условия в Запросах

Запросы позволяют получать результирующие таблицы, поля которых удовлетворяют определённым условиям (критериям). Эти условия задают в бланке запроса в строке *Условия отбора*. Условиями отбора являются логические выражения, состоящие из операторов и операндов. Используются операторы сравнения =, <, >, <> (не равно), Between, In, Like и логические операторы And, Or, Not. Допускается применять шаблоны с подстановочными символами. Если точное значение не известно или необходимо вводить значение не полностью, то удобно использовать **шаблон (образец)** с подстановочными символами (знаками). Примеры подстановочных символов:

- «#» - диез - применяется для замены любой одиночной цифры.
- «[]» - соответствует одному любому символу, заключенному в скобки.
- «!» - соответствует одному любому символу, не заключенному в скобки.
- «-» - соответствует любому символу из диапазона. Диапазон необходимо указать по возрастающей.

\* - соответствует любому количеству любых символов. Пример: 77\* - для нахождения всех телефонов с номерами, начинающимися на 77.

? - соответствует одному текстовому символу. Пример: 77-4?-0? - для нахождения всех телефонов с номерами, содержащими четыре указанные цифры.

Шаблоны используются совместно с оператором **Like**. Этот оператор позволяет создавать шаблоны, использующие подстановочные символы при поиске в текстовых полях. Например, фамилия сотрудника известна неточно. Это может

быть Петров, Петровский, Пеотровский и т.п. Тогда следует использовать для выборки в строке *Условие* запись Like "Пе\*". Известно, что имя состоит из 4-х букв. Тогда подойдет запись Like "????".

Оператор *Between* задаёт интервал значений. Например, Between 1 And 5 (указанные края интервалов в выборку включаются).

### **Операторы OR и AND (&).**

При вводе условия можно использовать операторы OR и AND (&) которые позволяют нам форматировать в одной строке сложные условия выбора. Например, при поиске записей об участниках из Баку и Шеки, можем поместить или выражение в одной строке результирующей таблицы будут содержать те же записи.

Оператор *In* выполняет проверку на равенство любому значению из списка, заданному в круглых скобках. Например, In("ручка";"духи").

Логические операции И, ИЛИ могут быть заданы явно в выражении условия с помощью операторов *AND* и *OR*. Например, «духи» OR «карандаш».

В качестве *операндов* в запросах могут использоваться литералы, константы, идентификаторы (ссылки).

**Литералами** являются конкретные значения, воспринимаемые системой так, как они записаны. Литералом может быть число, дата, строка. Например, 1146, #31.01.02 #, "Липецк".

**Константами** являются постоянные значения, которые определены в Access. Например, True, False, Null, Да, Нет.

**Идентификатор** осуществляет ссылку на поле, элемент управления или свойство. Идентификаторами могут быть имена полей, таблиц, форм, и так далее. Они заключаются в квадратные скобки. Ссылка на конкретное значение должна указывать на его местоположение в иерархии объектов в БД. Ссылка на поле в таблице имеет вид [Имя таблицы]![Имя Поля]. Например, [Сотрудники]![Фамилия].

Условие отбора целесообразно формировать с помощью построителя выражений. Для этого следует открыть окно **Построитель выражений**, щёлкнув кнопку **Построить** на панели инструментов или выбрав команду **Построить** в контекстном меню. Предварительно необходимо установить курсор мыши в ячейке ввода условия.

**Условие точного несовпадения значений одного из полей.** Если в таблице надо найти записи, значения полей которых не удовлетворяют определенному условию, то используется оператор *Not*. Оператор Not или <> вводится перед сравниваемым значением. Пример. Отобразить все записи таблицы TYPE, за исключением записей «карандаш» в поле *Товар*. Для этого в бланке запроса в столбце поля *Товар* в строке *Условие отбора* вводится Not "карандаш".

**Условие неточного совпадения.** Выбор записей по условию неточного совпадения значений можно осуществить, используя оператор Like. Этот оператор позволяет найти требуемые записи, зная лишь приблизительное написание текстовой величины. В операторе Like можно использовать шаблоны с подстановочными символами, что расширяет возможности поиска записей при неточном задании условий. Пример условия отбора: Like "[д-к]\*". Здесь - (минус) соответствует любому символу из диапазона. Диапазон необходимо указывать по возрастанию (д-к, но не к-д).

**Выбор записей по диапазону значений.** Для задания диапазона значений в окне конструктора запросов используются операторы >, <, Between, Like. Их можно применять с текстовыми, числовыми полями и полями типа даты. Примеры: в строке *Условие отбора* возможно ввести: >100.00 AND < 500.00; Between # 01.01.97 # AND #31.03.97#; Like "[М-Т]\*". Напомним, что символ # применяется для данных типа «дата/время».

### **Простые запросы на выборку**

Такие запросы используются для отбора записей из одной или нескольких взаимосвязанных таблиц в соответствии с заданными *критериями отбора*. Критерии отбора записей записываются в строке *Условие отбора* бланка запроса по образцу и расположенных под ней строках. Условия отбора записываются по определенным правилам. В табл. 1 приведены примеры записи условий отбора.

Условия отбора для нескольких полей могут быть записаны в одной строке или в разных строках. В первом случае при выполнении запроса будут выбираться те записи из таблиц, для которых выполнены все условия отбора. Если условия отбора находятся на разных строках бланка, то будут отбираться те записи, для которых выполняется хотя бы одно из условий.

Рис. 34. Пример записи условий отбора в запросе

Поле:	КодСотрудника	Фамилия	Имя	Отчество	Оклад
Имя таблицы:	Сотрудники	Сотрудники	Сотрудники	Сотрудники	Сотрудники
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	>4		<3000		

Например, в случае записи условий отбора так, как показано на рис. 34, будут отбираться те записи из таблицы СОТРУДНИКИ, для которых значение в поле КодСотрудника больше 4 и при этом значение в поле Оклад меньше 3000.

Критерий отбора записей	Запись условия отбора
Значение в текстовом поле начинается с символа "А", остальные символы могут быть любыми	A*
Значение в числовом поле >= 60	>= 60
Значение в текстовом поле НЕ равно "Инженер"	NOT "Инженер"
Значения в поле типа дата в диапазоне от 1.12.96 до 15.12.96	BETWEEN 1.12.96 AND 15.12.96
Значение в числовом поле в диапазоне от 10 до 100	BETWEEN 10 AND 100
Значение в текстовом поле равно "инженер" или "бухгалтер"	"инженер" OR "бухгалтер"
Значение в поле логического типа равно TRUE (истина)	TRUE

Таблица 1. Примеры записи условий отбора записей в запросе

#### **Запуск запроса.**

Мы закончили создание простого запроса, и наступило время запустить его на выполнение, нажимаем кнопку [!] запуск на панели инструментов или выполняем команду запрос запуск.

#### **Сохранение запроса.**

Созданный запрос можно использовать в дальнейшем, для этого мы должны присвоить ему имя и сохранить его. Для сохранения запроса в столбце файл выбираем команду сохранить как. По умолчанию M.S.Access предложит нам имя запроса в поле ввода новое имя, но лучше подобрать, что ни будь более значимое.

#### **Изменение внешнего вида результирующей таблицы.**

Мы можем легко изменить внешний вид результирующей таблицы, используя те же средства что и для обычных таблиц. Например, мы можем делать наш невидимый, можем зафиксировать их, изменить шрифт, размер строк и столбцов. Для выполнения этих функций переходим в режим таблицы и выбираем соответствующую команду из меню формат.

#### **Точное несовпадение значения одного из полей.**

Предположим, что нам требуется найти записи, значение которых не удовлетворяет определенным условиям. Для устранения таких условий используется оператор NOT, который печатается перед сравниваемым значением. Кроме NOT можно использовать и < >.

#### **Условие неточного совпадения.**

Требования точного задания чисел или последующих символов в качестве критерия поиска является жестким ограничением. Время от времени возникает неприятная ситуация, при которой мы не помним точного написания условия поиска. В подобных случаях, мы можем осуществить выбор записей по условию не точного совпадения значений непереключающих операторов LIKE. Данный оператор позволяет найти требуемые записи, зная лишь приблизительные написания величины. Кроме того, оператор LIKE используется с подстановочными символами:

? – Любой один знак.

\* – Нуль или более знаков.

# – Любая одна цифра.

[список знаков] – Любой один знак в списке знаков.

! – Любой один знак не входящий в список знаков.

Группа из одного или более символов заключается в квадратные скобки. Выражение может содержать любые символы за исключением следующих: [, ], ?, \*, знак числа. Помимо простого списка символов заключенных в квадратные скобки, список знаков позволяет задать диапазон символов. Мы можем указать знак « - » для разделения верхней границы диапазона до нижней.

Допустим нам надо выбрать всех студентов которые начинаются на А или D. Для этого в строке отбора поля фамилия, имя, отчество вводим следующие условия: LIKE "[A D]\*".

#### **Выбор записей по диапазону значений.**

Предположим, что чей кредит превышает сто долларов в другом случае нам нужно получить сведения о товарах, заказанных с 1 января по 31 декабря 2005 года. В обоих случаях необходимо выбрать из таблицы записи данные, которые попадают в диапазон значений. Для задания диапазона значений в окне конструктора запросов используем: >, >=, <, <=, и between. Рассмотрим пример использования оператора диапазона. В качестве примера используем список клиентов, чей кредит превышает сто долларов.

Выполняем следующие действия:

1. Открываем окно конструктора запросов для нужной таблицы.
2. Выделяем поля, которое хотим отобразить в запросе, и переносим их в область запроса.
3. Для задания условий отбора, выбираем поле кредит и в строке «условия отбора поля» печатаем > 100.
4. Нажимаем кнопку запуск и на экране появляется результирующая таблица, содержащая записи о товарах стоимостью свыше ста долларов.

В рассмотренном примере мы задавали только одну нижнюю границу диапазона. Для получения сведений обо всех заказах с 1 января по 31 декабря нам придется задать обе границы. Информация, необходимая для решения этой задачи находится в нужной таблице. Создадим запрос для этой таблицы. Для этого выполняем следующие действия:

1. Открываем окно конструктора запросов для нужной таблицы и переносим в бланк запроса в нужные поля.
2. Для задания условий отбора из списка полей таблицы выбираем поле дата заказа. Переходим на строку условия отбора и нажимаем кнопку построить . В списке операторов сравнения выбираем оператор BETWEEN. В области ввода выражения появится шаблон для ввода параметров данного оператора: BETWEEN “выражения” AND “выражения”.
3. Вставляем вместо первого параметра начальные данные, в качестве второго конечные данные.
4. Нажимаем [!] и на экране появляется результирующая таблица, содержащая информацию обо всех продажах за интересующий нас промежуток времени.

В заключение рассмотрим пример заданий диапазона строковых данных. В этом случае кроме операторов сравнения мы можем использовать и оператор LIKE. Например, для получения списка клиентов, фамилии которых начинаются от М до Я. На странице условия отбора в поле фамилия вводим условия отбора LIKE “[М - Я]\*”.

**Объединение критерий нескольких полей.**

В предыдущих примерах мы вводили условия запроса для одного из полей таблицы. Довольно часто возникает ситуация когда нам необходимо использовать сложные критерии выборки, в которых задаются условия для нескольких полей таблицы, или же несколько условий для одного поля. Для задания и выражения мы должны просто задать условия в строке условия отбора для каждого из полей образующихся критерий. При задании или выражение каждого из условий выбора, образующих критерий должна располагаться на отдельной строке бланка запроса. Например для выбора списка участников из Баку и Шеки нужно просто расположить первое условие в строке условие отбора, а второе в строке или.

В случае записи условий отбора так, как показано на рис. 35, будут отбираться записи из таблицы СОТРУДНИКИ, для которых значение в поле КодСотрудника больше 4, или записи, для которых значение в поле Оклад меньше 3000, а также записи, для которых справедливы оба условия отбора.

Рис. 35. Пример записи условий отбора в запросе

Поле:	КодСотрудника	Фамилия	Имя	Отчество	Оклад	
Имя таблицы:	Сотрудники	Сотрудники	Сотрудники	Сотрудники	Сотрудники	
Сортировка:	по возрастанию					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Условие отбора:	>4					
или:					<3000	

#### Вычисления в запросах на выборку

Результирующие таблицы запросов на выборку могут включать не только поля базовых таблиц, но и новые поля, значения в которых получаются как результат вычислений с использованием данных из базовых таблиц. Поле, содержимое которого получается как результат расчета по содержимому других полей, называется *вычисляемым полем*. Для создания вычисляемого поля вместо имени поля записывается формула в следующем формате:

Имя поля: выражение

Выражение может содержать: функции, имена полей, знаки действий, константы. При этом *имена полей заключаются в квадратные скобки*. Действия могут быть:

- арифметические*: + (сложение), - (вычитание), \* (умножение), / (деление), ^ (возведение в степень);
- логические*: And (логическое умножение), Or (логическое сложение), Not (логическое отрицание);
- сравнения*: > (больше), >= (больше либо равно), < (меньше), <= (меньше либо равно), = (равно), <> (не равно);
- соединения текстовых цепочек*: & (например: [Фамилия] & " " & [Имя]).

На рис. 36 показан пример запроса с вычисляемым полем Налог. При конструировании запроса в заголовке поля записывается формула Налог: [Оклад]\*0,13. В результирующей таблице появится поле с именем Налог, значения в котором будут получены как результат умножения соответствующих значений из поля Оклад на число 0,13. Чтобы вводить формулу было удобнее, можно нажать на клавиатуре <Shift>+<F2>. Открывается диалоговое окно Область ввода для ввода формулы (рис. 37).

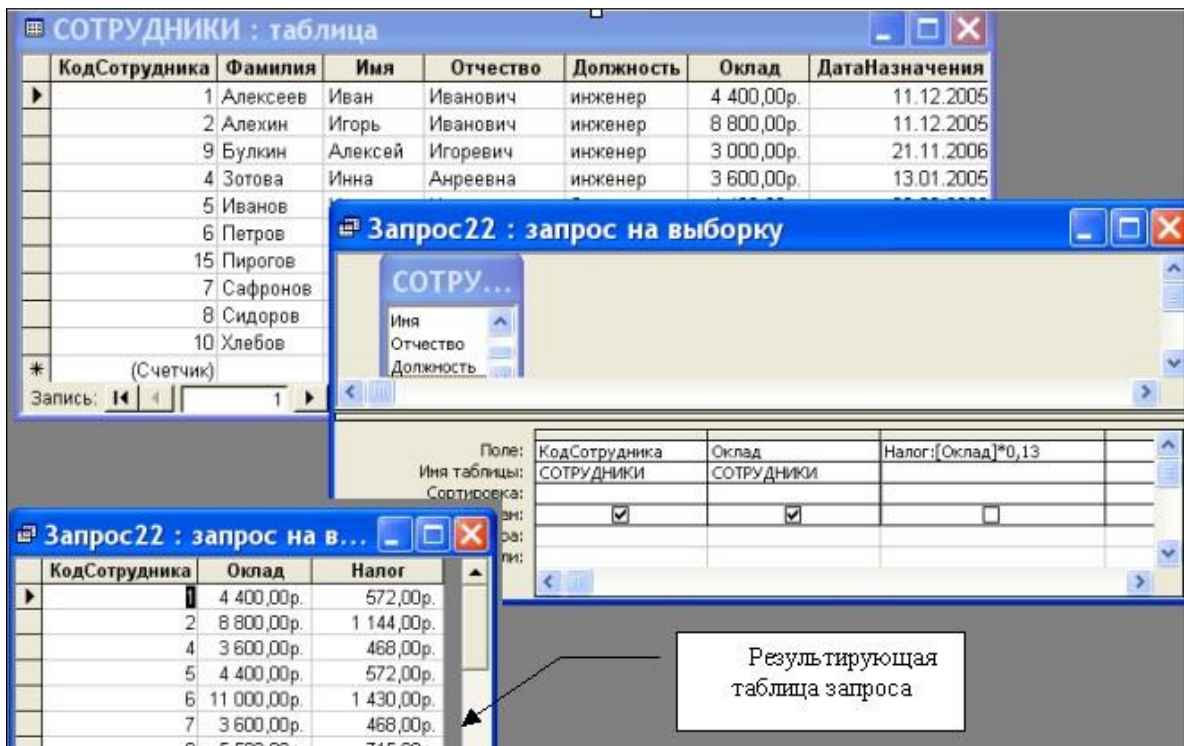


Рис. 36. Запрос с вычисляемым полем

Рис. 37. Фрагмент диалогового окна Область ввода

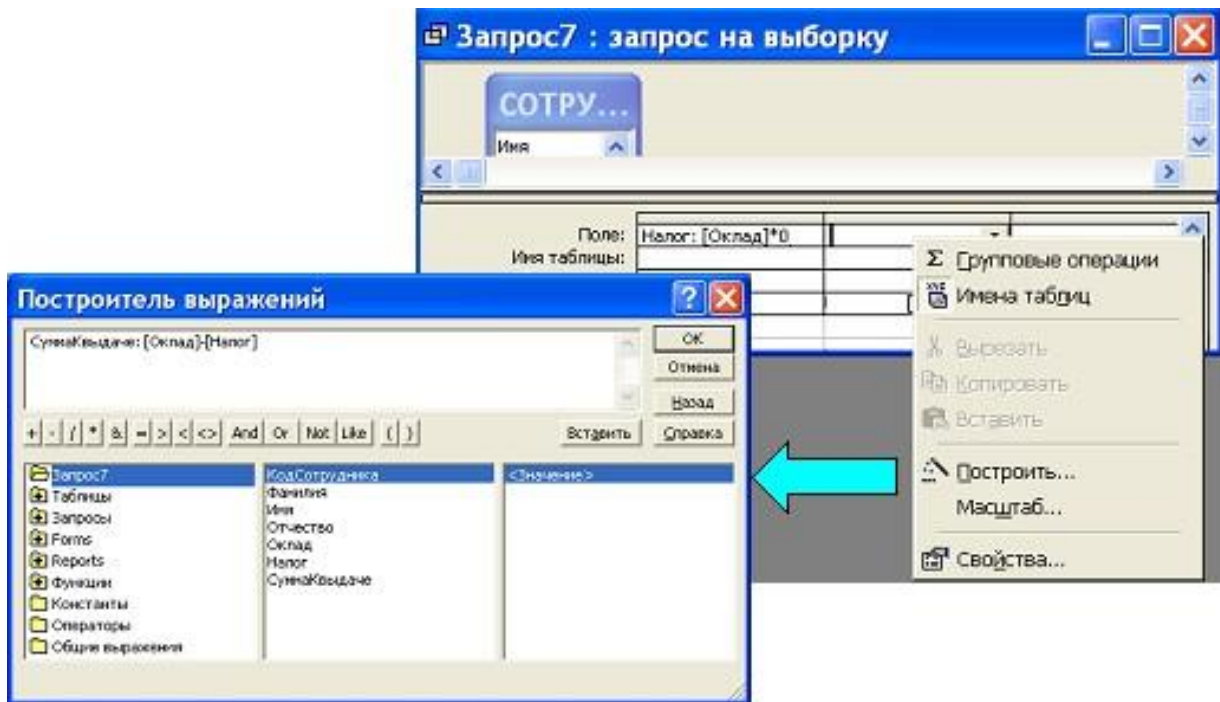


Рис. 38. Окно Построитель выражений

Удобно пользоваться для ввода формул Построителем выражений (рис. 38), которое открывается командой Построить из контекстного меню для поля ввода формулы.

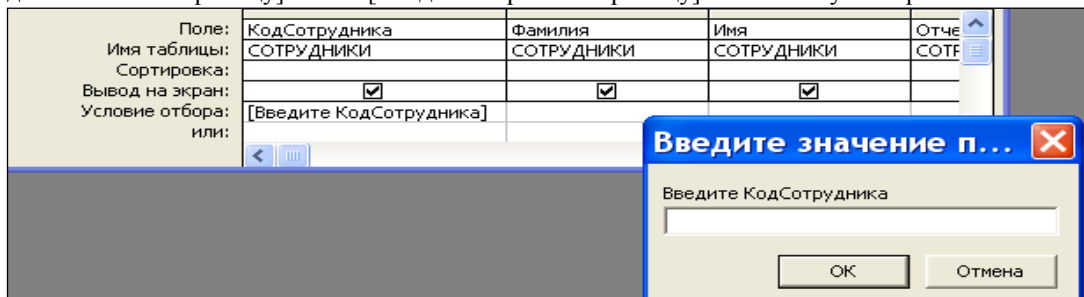
### Запросы с параметром

Запросы с параметром позволяют пользователю самому ввести критерий отбора данных на этапе выполнения запроса. При конструировании запроса с параметром в строке Условие отбора в квадратных скобках указывается текст, который будет выведен в диалоговом окне при выполнении этого запроса (рис. 39). Пользователь должен будет ввести с клавиатуры значение запрашиваемого параметра. Это значение будет использовано для отбора записей из базовых таблиц и включения их в результирующую таблицу запроса.

Рис. 39. Запрос с параметром-значением

Если в качестве параметров отбора нужно указать верхнюю и нижнюю границы какого-либо диапазона, то следует ввести условие отбора в виде:

BETWEEN [Введите нижнюю границу] AND [Введите верхнюю границу]. В этом случае при выполнении запроса



будут поочередно запрашиваться значения нижней и верхней границы параметра-диапазона, которые будут использованы для отбора записей из таблиц.


### Операторы OR и AND ( & ).

При вводе условия можно использовать операторы OR и AND (&) которые позволяют нам форматировать в одной строке сложные условия выбора. Например, при поиске записей об участниках из Баку и Шеки, можем поместить или выражение в одной строке результирующей таблицы будут содержать те же записи.

### Итоговые запросы и запросы на изменение данных

Итоговый запрос – это запрос, в котором выводятся результаты статистических расчетов по какой-либо группе записей из одной или нескольких таблиц. Можно находить сумму (функция Sum), среднее значение (функция Avg), наибольшее значение (функция Max) или наименьшее значение (функция Min), количество записей в группе (функция Count).

В итоговом запросе записи объединяются в группы по совпадающим значениям в каком-либо поле таблицы.

При создании запроса нужно щелкнуть по кнопке Групповые операции  на панели инструментов. В бланке запроса по образцу появляется дополнительная строка Групповая операция. В тех полях, по которым проводится группировка, следует установить функцию Группировка. В тех полях, где проводится итоговые операции, нужно в строке Групповая операция раскрыть список и выбрать одну из функций (Sum, Avg, Max, Min, Count и т. д.)

СОТРУДНИКИ : таблица							
	КодСотрудника	Фамилия	Имя	Отчество	Должность	Оклад	
+	1	Алексеев	Иван	Иванович	инженер	4 400,00р.	
+	2	Алехин	Игорь	Иванович	инженер	8 800,00р.	
+	9	Булкин	Алексей	Игоревич	инженер	3 000,00р.	
+	4	Зотова	Инна	Андреевна	инженер	3 600,00р.	
+	5	Иванов	Иван	Иванович	бухгалтер	4 400,00р.	
+	6	Петров	Петр	Петрович	бухгалтер	11 000,00р.	
+	15	Пирогов	Александр	Алексеевич	инженер	7 000,00р.	
+	7	Сафронов	Игорь	Трофимович	инженер	3 600,00р.	
+	8	Сидоров	Иван	Олегович	менеджер	5 500,00р.	

Рис. 41. Таблица СОТРУДНИКИ

Пример 4.2. Таблица СОТРУДНИКИ содержит данные о должностях и размерах окладов (рис. 41). Можно создать запрос для определения среднего, наибольшего, наименьшего оклада для каждой должности.

Группировка выполняется по полю Должность, в поле Оклад, которое включается в бланк запроса трижды (рис. 42), выбираются из раскрывающегося списка функции для вычисления среднего, минимального и максимального значения. Результатом выполнения запроса будет таблица, показанная на рис. 43.

Рис. 42. Создание итогового запроса

Рис. 43. Результат выполнения итогового запроса

**СОТРУДНИКИ**

КодСотрудника

Фамилия

Имя

Отчество

Поле:	Должность	Оклад	Оклад	Оклад
Имя таблицы:	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ	СОТРУДНИКИ
Групповая операция:	Группировка	Avg	Min	Max
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				

**Запрос12 : запрос на выборку**

Должность	Avg-Оклад	Min-Оклад	Max-Оклад
бухгалтер	7 700,00р.	4 400,00р.	11 000,00р.
инженер	4 971,43р.	3 000,00р.	8 800,00р.
менеджер	5 500,00р.	5 500,00р.	5 500,00р.

Запрос на изменение данных – это запрос, который за одну операцию вносит изменения в несколько записей таблицы. Существует четыре типа запросов на изменение данных: на удаление записей, на обновление записей, на добавление записей, на создании таблицы.

Запрос на удаление записей удаляет группу записей из одной или нескольких связанных таблиц.

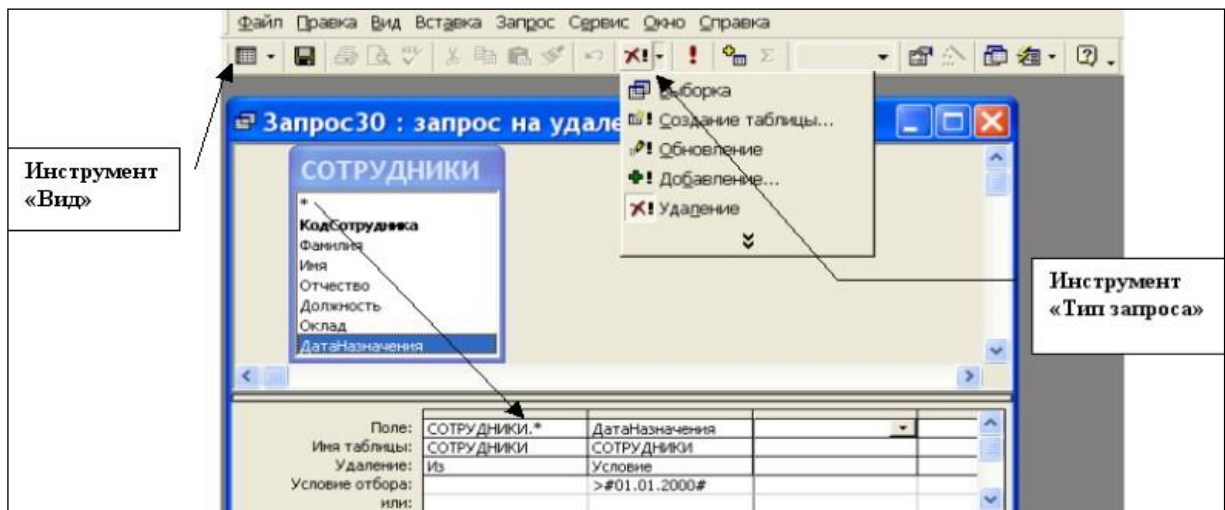


Рис. 44. Создание запроса на удаление записей таблицы

Пример 4.3. Требуется удалить из таблицы СОТРУДНИКИ все записи о сотрудниках, принятых на работу после 01.01.2000.

При заполнении бланка запроса перетаскиваем символ «\*» в строку Поле первого столбца, включаем в бланк также поле Дата назначения. Для поля Дата назначения в строке Условие отбора вводим условие: >01.01.2000 (рис. 44). В результате выполнения этого запроса из таблицы СОТРУДНИКИ будут удалены те записи таблицы, для которых значение в поле Дата назначения больше 01.01.2000.

Запрос на создание таблицы создает новую базовую таблицу (имена базовых таблиц указаны на вкладке Таблицы в окне базы данных) на основе всех или части данных из одной или нескольких таблиц

Запрос на обновление записей вносит общие изменения в группу записей одной или нескольких базовых таблиц базы данных.

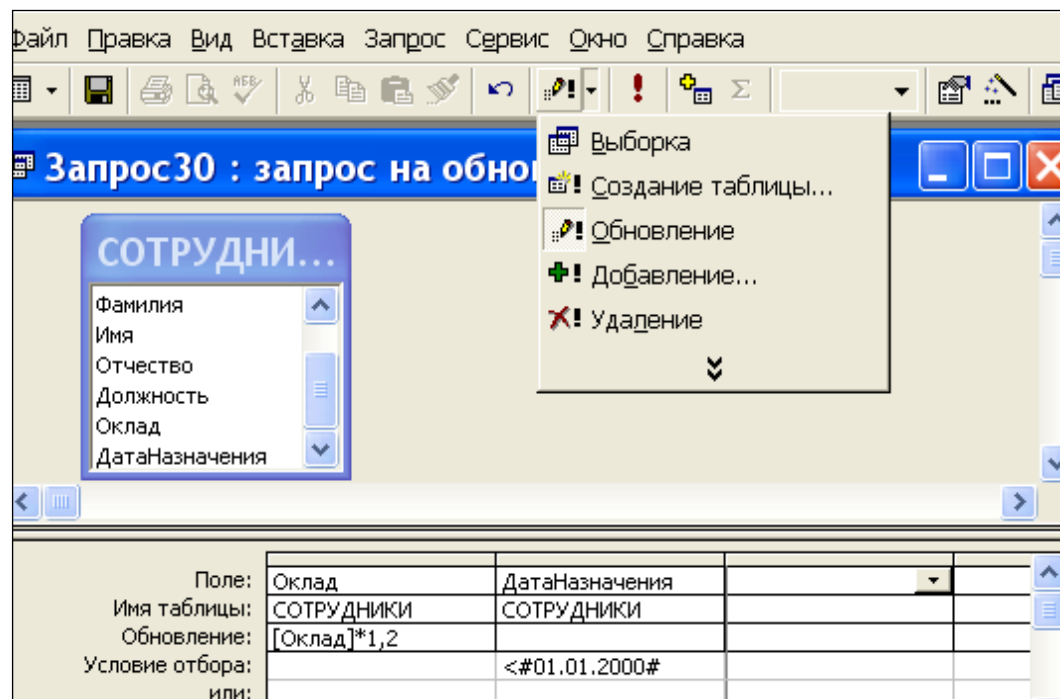


Рис. 45. Создание запроса на обновление записей

Пример 4.4. Требуется создать запрос на обновление, после выполнения которого в таблице СОТРУДНИКИ будут увеличены на 20% оклады сотрудников, принятых на работу до 01.01.2000 г.

При заполнении бланка запроса включаем в него поля Оклад и Дата назначения из таблицы СОТРУДНИКИ (рис. 45). Для поля Оклад в строке Обновление вводим правило обновления: [Оклад] \* 1,2. Для поля Дата назначения в строке Условие отбора вводим условие: < 01.01.2000. В результате выполнения этого запроса в таблице СОТРУДНИКИ будут изменены значения в поле Оклад в тех записях таблицы, для которых значение в поле Дата назначения меньше 01.01.2000.

### Работа с отчетами

Отчеты служат для вывода данных на печатающие устройства и должны учитывать параметры принтера и параметры используемой бумаги. Средства автоматического проектирования отчетов реализованы автоотчетами (База данных – Создать – Новый отчет – Автоотчет в столбце). Кроме автоотчетов в столбец существуют ленточные автоотчеты (рис. 47, 48). В режиме Автоотчет можно создать только отчет на базе одной таблицы или запроса, в котором выводятся все поля и все записи из базовой таблицы или запроса.

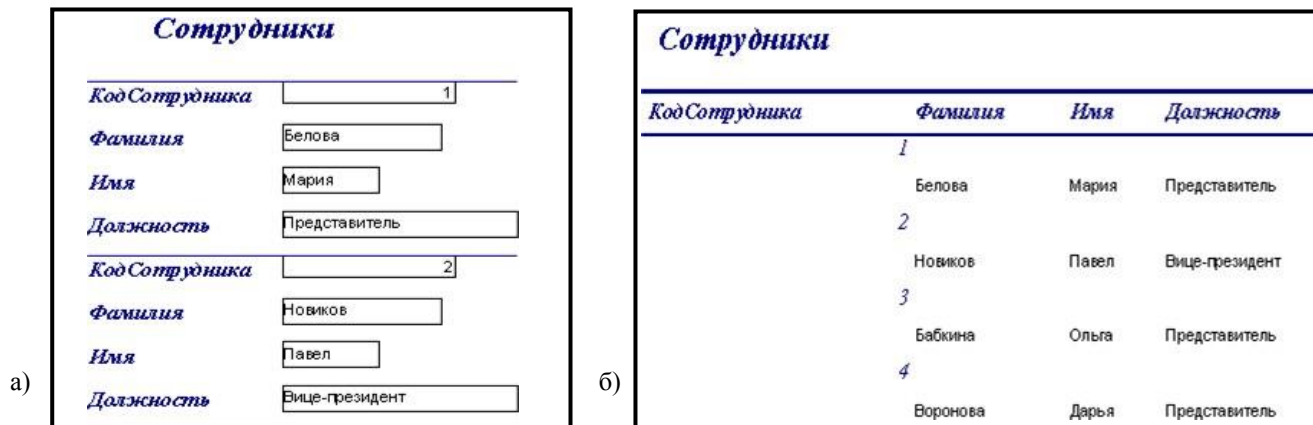


Рис. 48. Автоотчеты: в столбец (а) и ленточный (б)

Средством автоматизированного создания отчетов является Мастер отчетов (запускается на вкладке Отчеты в окне База данных двойным щелчком на значке Создание отчета с помощью мастера)

Под отчетами понимается форматированное представление данных, выводимое на экран, принтер или файл. Таблица отчетов представляет собой напечатанную таблицу, в которой данные упорядочены по столбцам и строкам. Каждый из столбцов отчета содержит поле исходной таблицы или вычисляемое поле. Строки представляют собой запись. Отчеты в свободной форме позволяют установить ограничения, свойственные табличным отчетам. При получении отчетов свободной формы можно использовать стандартный формат, автоматически создаваемый M.S.Access для каждой таблицы. В этом формате поля исходной таблицы расположены вертикально, но с помощью конструктора отчетов можно разработать специальный формат, где поля исходной таблицы будут располагаться в требуемых местах отчета.

#### Простейший способ создания отчета.

Для создания самого простого отчета нужно в столбце вставка выбрать команду автоотчет.

#### Другие способы создания отчетов.

В M.S.Access для создания отчетов используются следующие средства:

- Конструктор отчетов.
- Мастер отчетов.
- Автоотчет в столбце.
- Автоотчет ленточный.

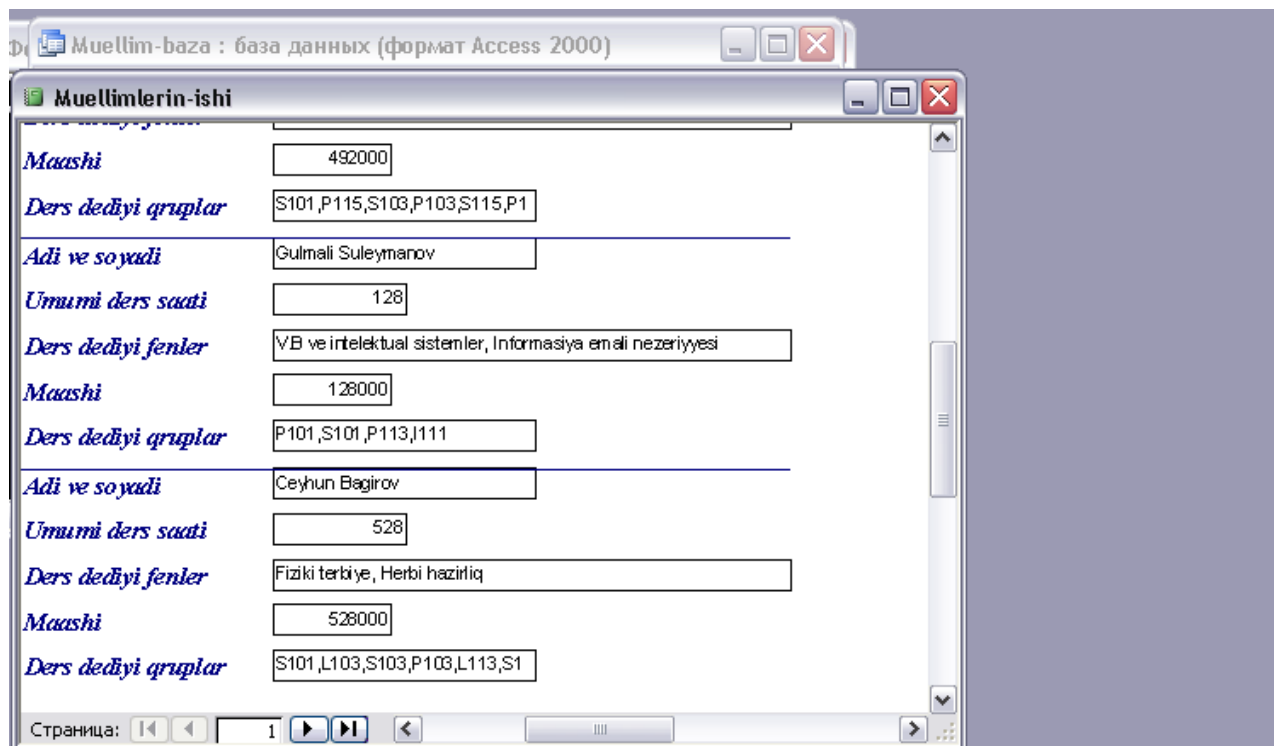


Диаграмма.

Почтовые наклейки.

### Окно конструктора отчетов .

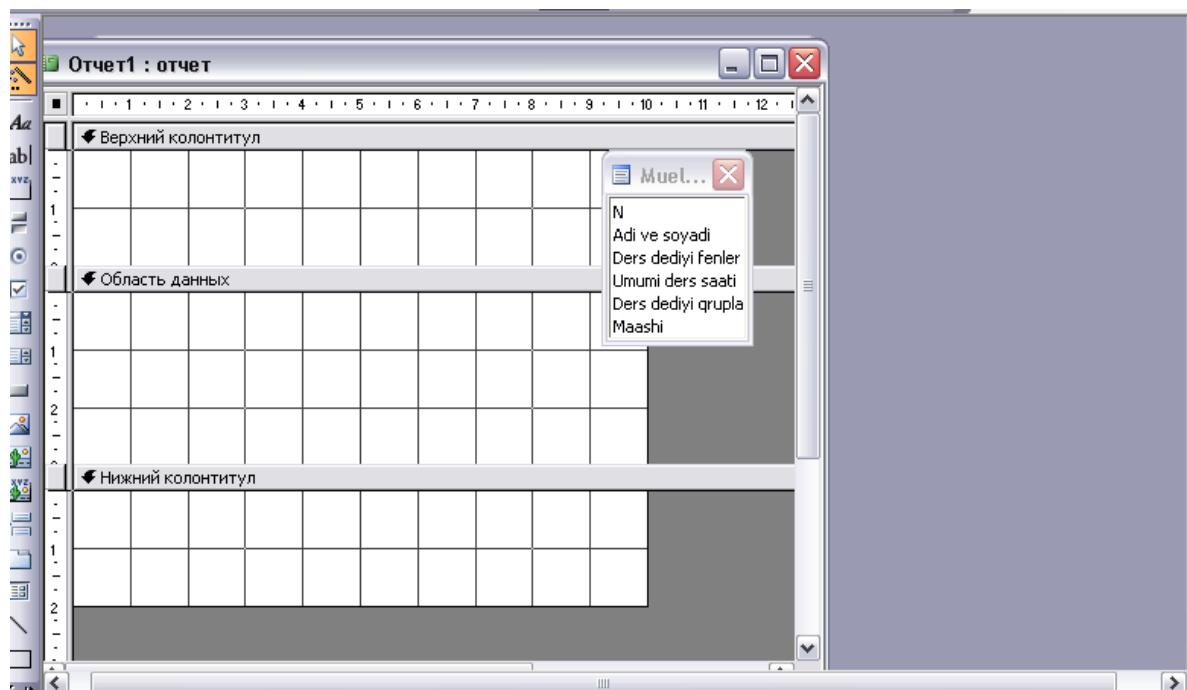
Основными объектами является пояснительный текст и поля отчета.

При создании и модификации отчетов конструктор отчетов позволяет удалять , добавлять, перемещать области вместе с расположенными в них объектами .Мы можем устанавить цвет и т.д.

Для открытия окна конструктора выполняем следующие:

1. Переходим на вкладку "Отчет".
2. Нажимаем кнопку **Создать** .
3. Из списка способов создания выбираем **Конструктор** и нажимаем кнопку **ОК** .

Создание отчета аналогично созданию формы.

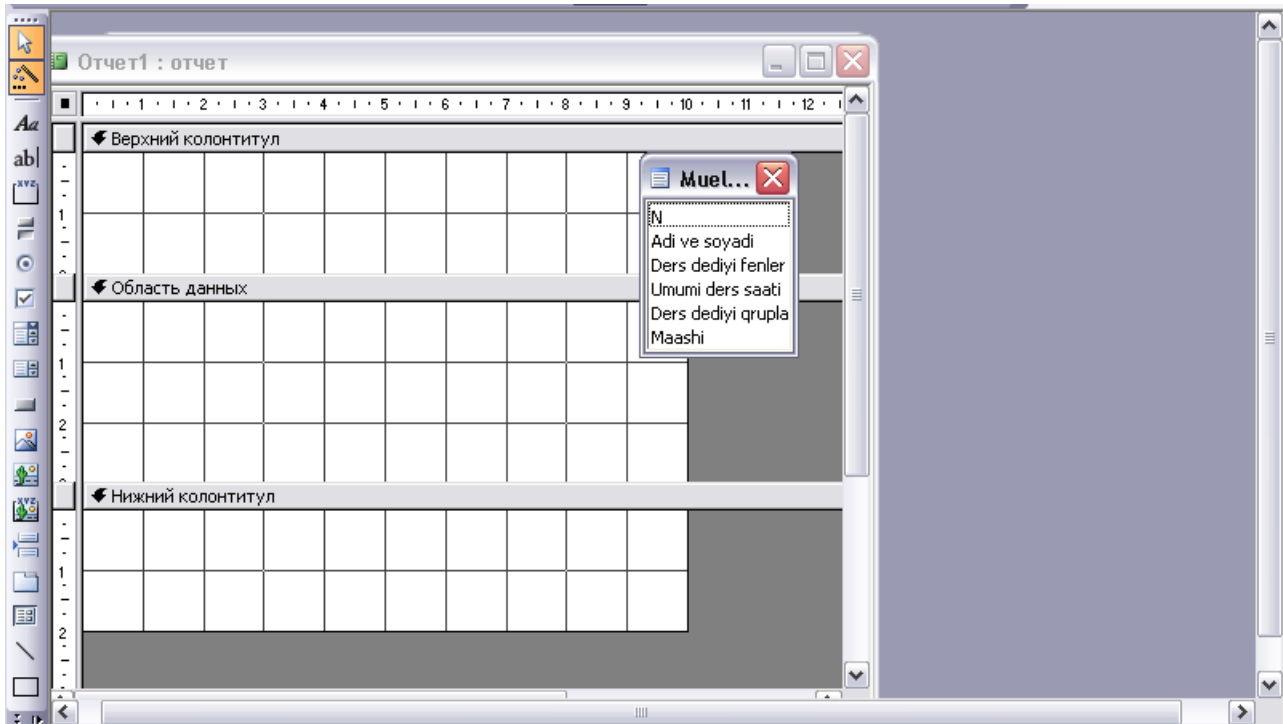


**Области отчетов**

Окно конструктора отчетов содержит объекты и сетку аналогично окну конструктора форм.

- @ Заголовок отчета
- @ Верхний колонтитул

- @ Область заголовка группы
- @ Область данных
- @ Область примечание группы
- @ Нижний колонтитул
- @ Область примечаний
- @



#### Размещение даты печати отчета.

Обычно в заголовке отчета или в нижнем колонтитуле размещается дата печати отчета:

1. для создания данного элемента выбираем инструмент, поле, устанавливаем кнопку мыши на место в области верхнего колонтитула, в котором хотим разместить поле дата. В отчете появится связный объект из поля ввода и надпись к нему.
2. устанавливаем надпись связанного поля и удаляем ее.
3. выделяем поле ввода и открываем для него окно свойств.
4. переходим на вкладку данные и нажимаем кнопку свойства данных.
5. на экране построение выражения открывается раздел встроенная функция. Используя функцию NOW задаем выражение для этого поля.
6. нажимаем кнопку раскрытие списка и в столбце формат включаем, может, и выбираем нужный нам формат.

#### Отображение промежуток суммы в конце каждой строки отчета.

M.S.Access не позволяет использование суммы в нижнем колонтитуле. Рассмотрим отчет для таблицы в нижнем колонтитуле которой требуется разместить сумму. Для этого:

1. в любой другой области создаем элемент управления, подсчитывающий значение этой суммы.
2. устанавливаем свойство вывод на экран значения нет. Кроме этого, по желанию выбираем для него свойства суммы с накоплением значений для группы или всех.
3. создаем в нижнем колонтитуле не связанное поле. В свойстве данные этого поля вводим имя элемента управления, который подсчитывает сумму и содержит выражение равное сумме. Он будет отображать промежуточную сумму, вычисленную элементом управления сумма.