

4. Базы данных

4.1. Базы данных и модели данных. Основы проектирования реляционных баз 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. **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.

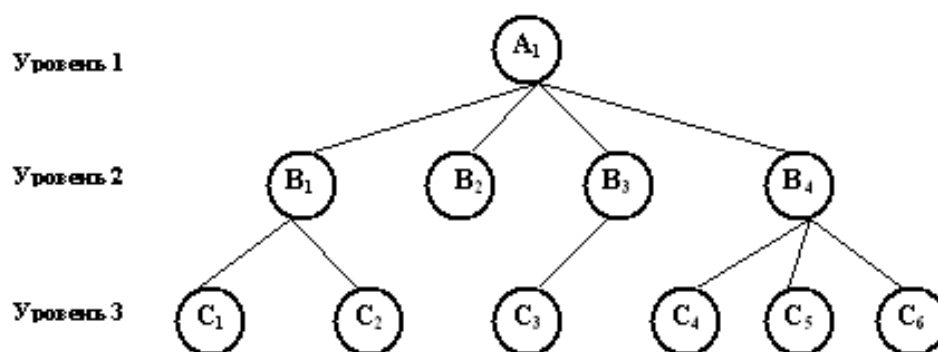


Рис. 1. Графическое изображение иерархической структуры данных

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.

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

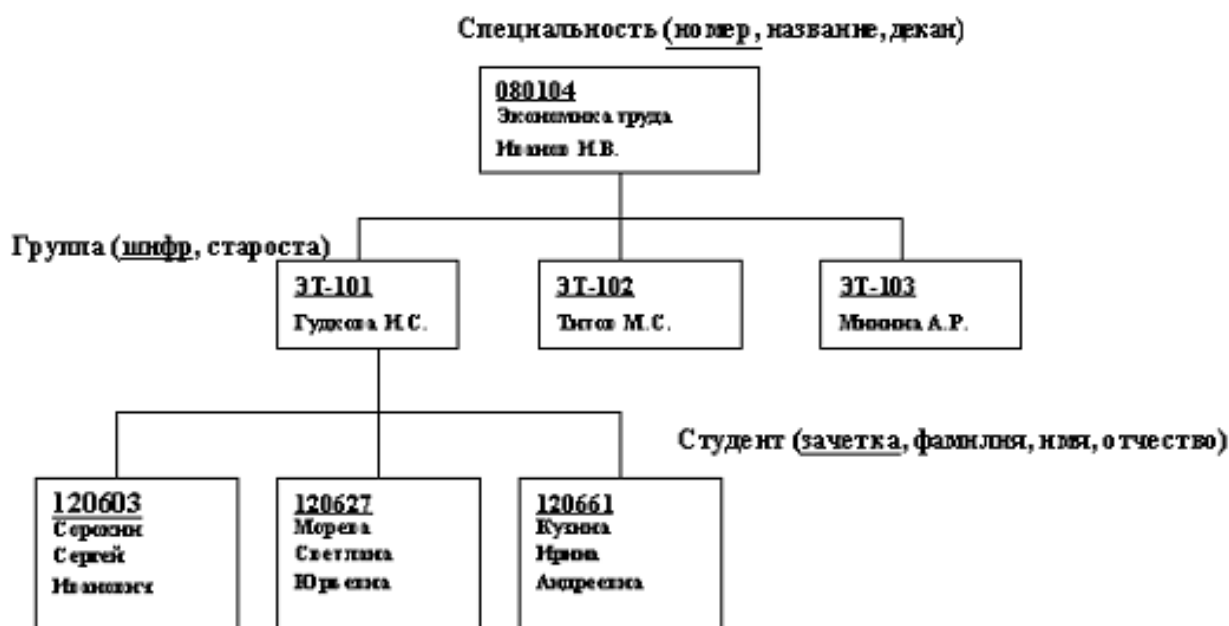


Рис. 2. Пример иерархической структуры данных

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.

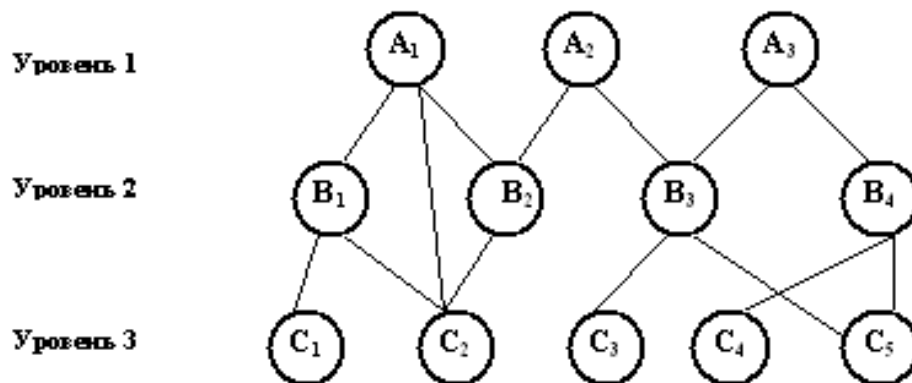


Рис. 3. Графическое изображение сетевой структуры данных

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.

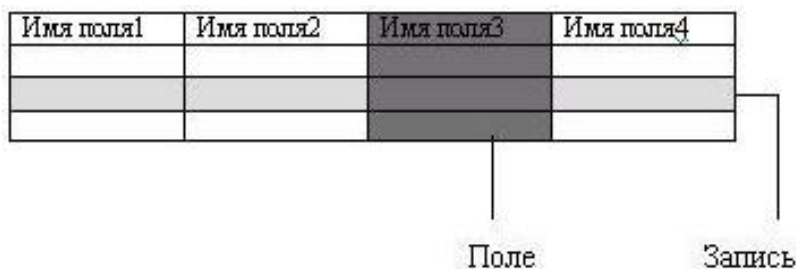


Рис. 5. Основные структурные элементы реляционной таблицы

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

| 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 |

Рис. 6. Пример реляционной таблицы

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.

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. 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).

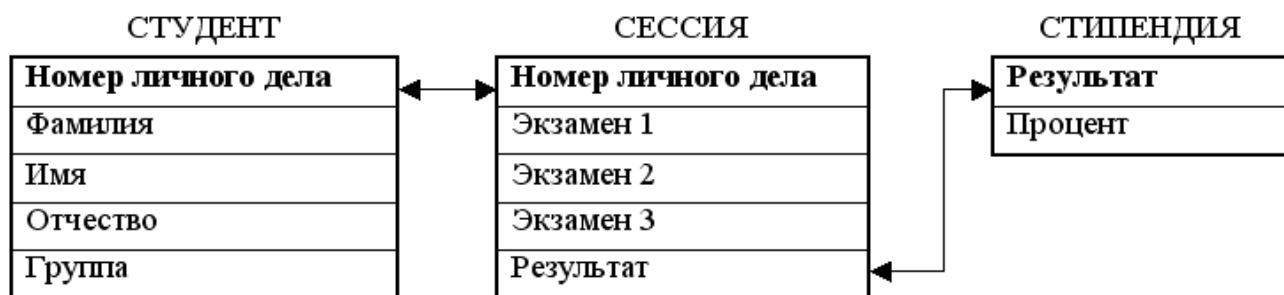


Рис. 8. Пример реляционной структуры данных

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 *A* связана с записью таблицы *B*, если в обеих таблицах эти записи содержат одно и то же значение в поле, по которому установлена связь между таблицами.

Между таблицами *A* и *B* установлена **связь «один-к-одному»**, если каждая запись в таблице *A* может иметь не более одной связанной с ней записи в таблице *B* и наоборот, каждая запись в таблице *B* может иметь не более одной связанной с ней записи в таблице *A*. 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 *A* и *B* установлена **связь «один-ко-многим»**, если каждая запись в таблице *A* может быть связана с несколькими записями таблицы *B*, но каждая запись в таблице *B* не может быть связана более чем с одной записью таблицы *A*. Таблица *A* в этом случае называется **главной** таблицей, а таблица *B* – **подчиненной**. В этом случае для связи используется поле, которое

является первичным ключом таблицы, находящейся на стороне отношения «один» (таблица *A*), и являющееся внешним ключом в таблице, находящейся на стороне отношения «многие» (таблица *B*).

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**.

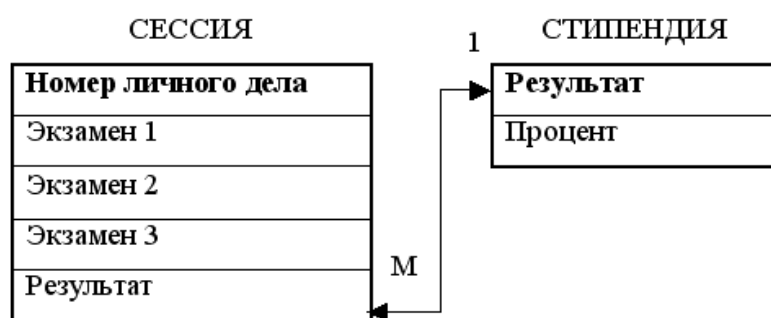


Рис. 10. Пример таблиц, находящихся в отношении «один-ко-многим»

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

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»

Этапы проектирования реляционной базы данных

В процессе разработки базы данных можно выделить несколько этапов.

Анализ предметной области. На этом этапе формируется задание по созданию базы данных. В нем подробно описывается состав базы данных, назначение и цели ее создания, а также перечисляется, какие виды работ предполагается осуществлять в этой базе данных (отбор, изменение данных, печать или вывод отчета и т. д.).

Разработка схемы данных БД. На этом этапе рассматривается, из каких информационных объектов должна состоять база данных, какими атрибутами описывается каждый объект. Затем определяется структура реляционных таблиц с указанием свойств полей и связей между таблицами.

Синтез компьютерной модели объекта, предполагающий выполнение следующих типовых операций:

- 1) создание файла базы данных;
- 2) создание базовых таблиц;

- 3) создание экранных форм для ввода, редактирования и просмотра данных в таблицах;
- 4) заполнение таблиц данными;
- 5) работа с созданной базой данных:
 - сортировка, фильтрация и поиск записей в таблицах;
 - отбор данных из таблиц в соответствии с заданными критериями отбора;
 - выполнение обработки данных (удаление, добавление, изменение данных, выполнение вычислений);
 - подготовка отчетов.

Рассмотрим подробно *этап разработки схемы данных*. Разработку схемы данных обычно выполняют в следующей последовательности:

- 1) составляется общий список полей, отражающий атрибуты таблиц базы данных;
- 2) поля общего списка распределяются по базовым таблицам;
- 3) в соответствии со свойствами данных определяются свойства каждого поля;
- 4) в каждой таблице выделяется ключевое поле;
- 5) определяются связи между таблицами.

На рис. 12 показан пример установки связей между таблицами, составляющими одну базу данных. Схемы таблиц с указанием связей между ними образуют *схему данных* проектируемой базы данных.

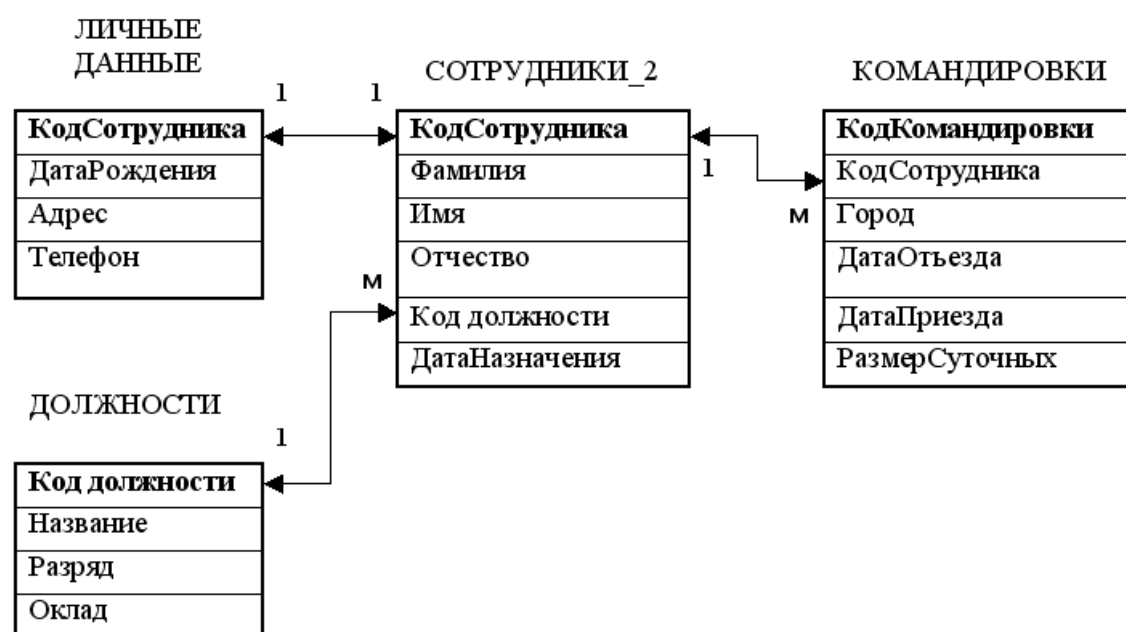


Рис. 12. Пример схемы данных

Критерии распределения полей по таблицам могут быть различными. Например, сведения о сотрудниках организации могут храниться в одной таблице, или могут быть созданы две таблицы, одна из которых (СОТРУДНИКИ) хранит информацию служебного назначения, а другая (ЛИЧНЫЕ ДАННЫЕ) – информацию личного плана (рис. 13).

СОТРУДНИКИ_1 : таблица

| КодСотрудн | Фамилия | Имя | Отчество | ДатаРождения | Адрес | Телефон | Должность | Оклад | ДатаНазн |
|------------|----------|-----------|------------|--------------|----------------|----------|-----------|-------------|----------|
| 1 | Алексеев | Иван | Иванович | 12.12.1976 | ул. Мурысева | 24-68-86 | инженер | 4 400,00р. | 11.12. |
| 2 | Алехин | Игорь | Иванович | 12.12.1976 | ул. Ярославск | 24-68-87 | инженер | 8 800,00р. | 11.12. |
| 4 | Зотова | Инна | Андреевна | 12.12.1976 | ул. Лесная 34 | 24-68-88 | инженер | 3 300,00р. | 13.01. |
| 5 | Иванов | Иван | Иванович | 12.12.1976 | ул. Ярославск | 24-68-89 | бухгалтер | 4 400,00р. | 26.09. |
| 6 | Петров | Петр | Петрович | 12.12.1976 | ул. Гидрострое | 24-68-90 | бухгалтер | 11 000,00р. | 31.12. |
| 7 | Сафронов | Игорь | Трофимович | 12.09.1976 | ул. Ярославск | 24-68-91 | инженер | 3 600,00р. | 01.01. |
| 8 | Сидоров | Иван | Олегович | 15.12.1976 | ул. Матросова | 24-68-92 | менеджер | 5 500,00р. | 02.06. |
| 9 | Булкин | Алексей | Игоревич | 15.12.1976 | ул. Матросова | 24-68-93 | инженер | 3 000,00р. | 21.11. |
| 10 | Хлебов | Олег | Олегович | 15.12.1976 | ул. Зеленая 54 | 24-68-94 | инженер | 4 500,00р. | 30.03. |
| 15 | Пирогов | Александр | Алексеевич | 19.01.1976 | ул. Ярославск | 24-68-95 | инженер | 7 000,00р. | 01.01. |

СОТРУДНИКИ : таблица

| КодСотрудн | 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. | 21.11.2005 |
| 4 | Зотова | Инна | Андреевна | инженер | 3 300,00р. | 13.01.2005 |
| 5 | Иванов | Иван | Иванович | бухгалтер | 4 400,00р. | 26.09.2005 |
| 6 | Петров | Петр | Петрович | бухгалтер | 11 000,00р. | 31.12.2005 |
| 15 | Пирогов | Александр | Алексеевич | инженер | 7 000,00р. | 01.01.2005 |
| 7 | Сафронов | Игорь | Трофимович | инженер | 3 600,00р. | 01.01.2005 |
| 8 | Сидоров | Иван | Олегович | менеджер | 5 500,00р. | 02.06.2005 |
| 10 | Хлебов | Олег | Олегович | инженер | 4 500,00р. | 30.03.2005 |

ЛИЧНЫЕ ДАННЫЕ : таблица

| КодСотрудн | ДатаРождения | Адрес | Телефон |
|------------|--------------|----------------------------|----------|
| 1 | 01.02.1980 | ул. Мурысева 45 - 59 | 24-68-86 |
| 2 | 12.12.1981 | ул. Ярославская 56 - 98 | 24-68-87 |
| 4 | 12.12.1976 | ул. Лесная 34 - 21 | 24-68-88 |
| 5 | 12.12.1982 | ул. Ярославская 56 - 98 | 24-68-89 |
| 6 | 12.12.1976 | ул. Гидростроевская 1 - 78 | 24-68-90 |
| 7 | 12.09.1987 | ул. Ярославская 56 - 98 | 24-68-91 |
| 8 | 15.12.1986 | ул. Матросова 76 - 43 | 24-68-92 |

Рис. 13. Пример разделения одной таблицы на две по признаку разделения доступа к данным

Часто критерием выделения новой таблицы является факт повторения данных в исходной таблице. Например, в таблице СОТРУДНИКИ многократно повторяются названия должностей и оклады. Можно создать таблицу ДОЛЖНОСТИ, в которой хранить названия и характеристики должностей (рис.14). Тогда данные таблицы СОТРУДНИКИ будут храниться в двух таблицах: СОТРУДНИКИ_2 и ДОЛЖНОСТИ, между которыми устанавливается связь.

Разработкой схемы данных заканчивается этап проектирования базы данных. Далее следует этап компьютерной реализации базы данных, который в значительной мере определяется выбором конкретной СУБД.

СОТРУДНИКИ : таблица

| | КодСотрудника | Фамилия | Имя | Отчество | Должность | Оклад | ДатаНазначен |
|---|---------------|----------|-----------|------------|-----------|-------------|--------------|
| + | 1 | Алексеев | Иван | Иванович | инженер | 4 400,00р. | 11.12.20 |
| + | 2 | Алехин | Игорь | Иванович | инженер | 8 800,00р. | 11.12.20 |
| + | 9 | Булкин | Алексей | Игоревич | инженер | 3 000,00р. | 21.11.20 |
| + | 4 | Зотова | Инна | Андреевна | инженер | 3 600,00р. | 13.01.20 |
| + | 5 | Иванов | Иван | Иванович | бухгалтер | 4 400,00р. | 26.09.20 |
| + | 6 | Петров | Петр | Петрович | бухгалтер | 11 000,00р. | 31.12.20 |
| + | 15 | Пирогов | Александр | Алексеевич | инженер | 7 000,00р. | 01.01.20 |
| + | 7 | Сафронов | Игорь | Трофимович | инженер | 3 600,00р. | 01.01.20 |
| + | 8 | Сидоров | Иван | Олегович | менеджер | 5 500,00р. | 02.06.20 |
| + | 10 | Ушаков | Олег | Олегович | инженер | 4 400,00р. | 30.03.20 |

Запись: 1 из 10

СОТРУДНИКИ_2 : таблица

| | КодСотрудника | Фамилия | Имя | Отчество | Код должности | ДатаНазначен |
|---|---------------|----------|-----------|------------|---------------|--------------|
| + | 1 | Алексеев | Иван | Иванович | 100 | 11.12.200 |
| + | 2 | Алехин | Игорь | Иванович | 101 | 11.12.200 |
| + | 9 | Булкин | Алексей | Игоревич | 102 | 21.11.200 |
| + | 4 | Зотова | Инна | Андреевна | 103 | 13.01.200 |
| + | 5 | Иванов | Иван | Иванович | 301 | 26.09.200 |
| + | 6 | Петров | Петр | Петрович | 303 | 31.12.200 |
| + | 15 | Пирогов | Александр | Алексеевич | 104 | 01.01.200 |
| + | 7 | Сафронов | Игорь | Трофимович | 103 | 01.01.200 |
| + | 8 | Сидоров | Иван | Олегович | 201 | 02.06.200 |

Запись: 9 из 10

ДОЛЖНОСТИ : таблица

| | Код должности | Название | Оклад |
|---|---------------|-----------|-------------|
| + | 100 | инженер | 4 400,00р. |
| + | 101 | инженер | 8 800,00р. |
| + | 102 | инженер | 3 000,00р. |
| + | 103 | инженер | 3 600,00р. |
| + | 104 | инженер | 7 000,00р. |
| + | 201 | менеджер | 5 500,00р. |
| + | 301 | бухгалтер | 4 400,00р. |
| + | 303 | бухгалтер | 11 000,00р. |

Запись: 1 из 8

Рис. 14. Пример разделения одной таблицы на две по признаку устранения повторяемости данных

? Вопросы и упражнения для самоконтроля

1. Что такое база данных (БД), система управления базами данных (СУБД)?
2. Что такое информационный объект?
3. Что такое модель данных?

4. Укажите основные понятия и характеристики иерархической и сетевой моделей данных.
5. Укажите характеристики и объекты обработки реляционной модели данных.
6. Что такое первичный ключ реляционной таблицы базы данных?
7. Что такое внешний ключ реляционной таблицы базы данных?
8. Какого типа связи (отношения) могут быть установлены между таблицами реляционной базы данных? Охарактеризуйте каждый тип связи.
9. Что такое «схема данных» реляционной базы данных?
10. Перечислите и охарактеризуйте основные этапы проектирования и разработки реляционной базы данных.

4.2. Основы работы с СУБД Microsoft Access. Основные объекты базы данных. Работа с таблицами

Объекты базы данных Microsoft 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.

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

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

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).

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.



| | 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

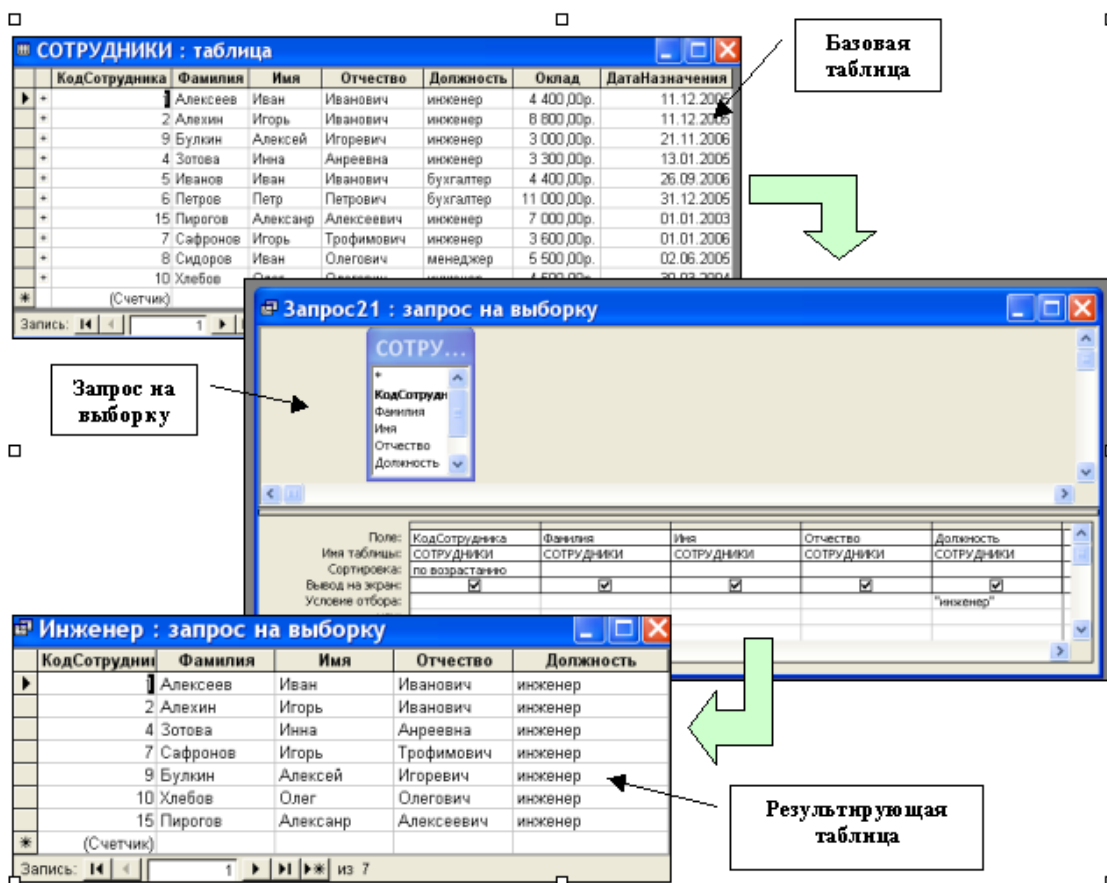


Рис. 16. Пример выполнения запроса к базе данных

СОТРУДНИКИ

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

Запись: 1 из 10

Рис. 17. Пример формы базы данных

| <i>Сотрудники1</i> | | | | |
|--------------------|------------|-----------|---------------|------------|
| Должность | Фамилия | Имя | Отчество | Оклад |
| <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. |

Рис. 18. Пример простейшего отчета

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.

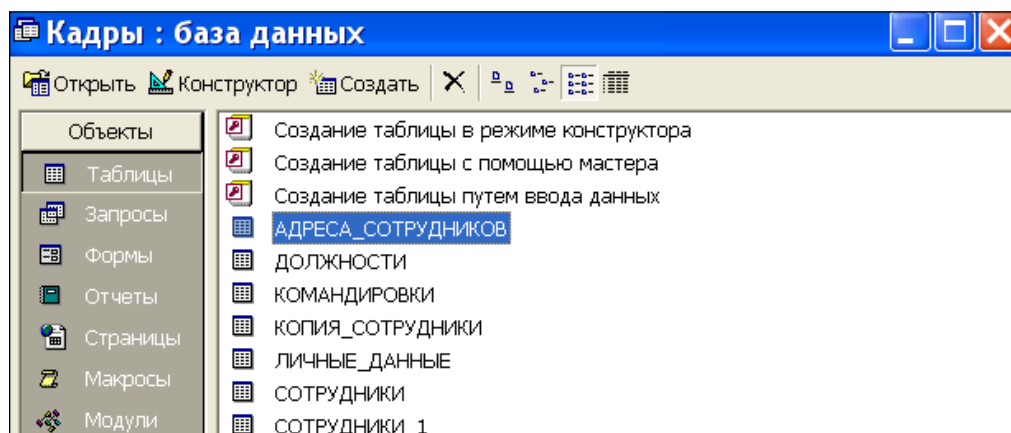


Рис. 19. Окно **База данных**

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);
- **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-января-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);
- **Поле Мемо** – специальный тип для хранения больших объемов текста (до 65 535 символов);
- **Числовой** – используется для хранения чисел;
- **Дата/время** – используется для хранения календарных дат и текущего времени;
- **Денежный** – используется для хранения денежных сумм.
- **Счетчик** – содержит натуральные числа 1, 2, 3, ... (при этом значения вводятся в поле автоматически при создании новой записи);
- **Логический** – используется для хранения логических данных (**Истина** или **Ложь**);
- **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.

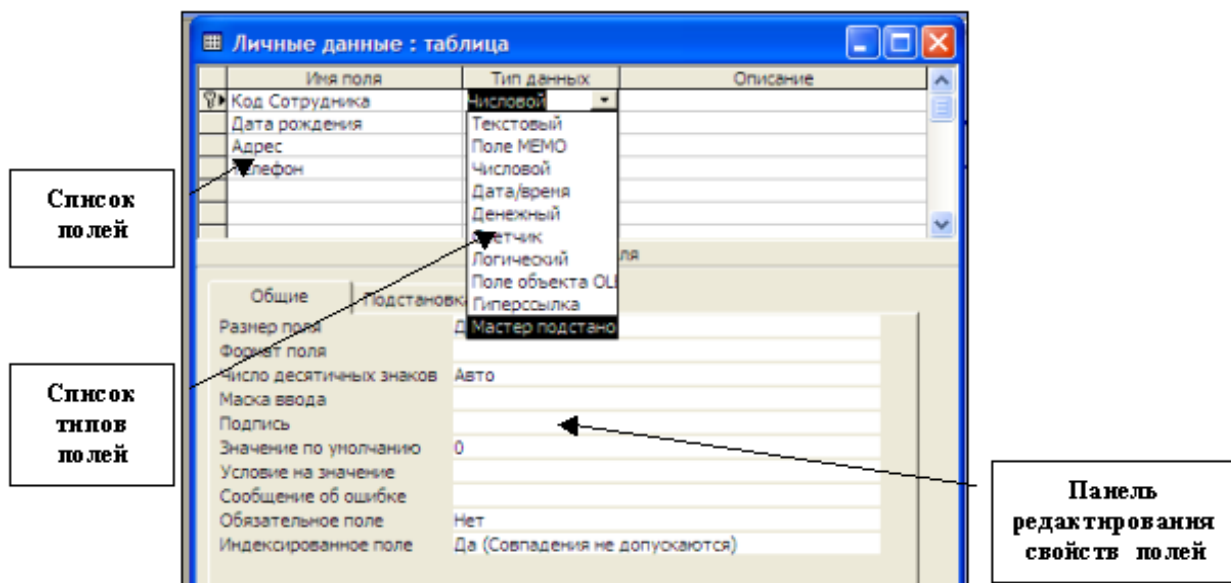


Рис.20. Таблица в режиме **Конструктор**

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**.

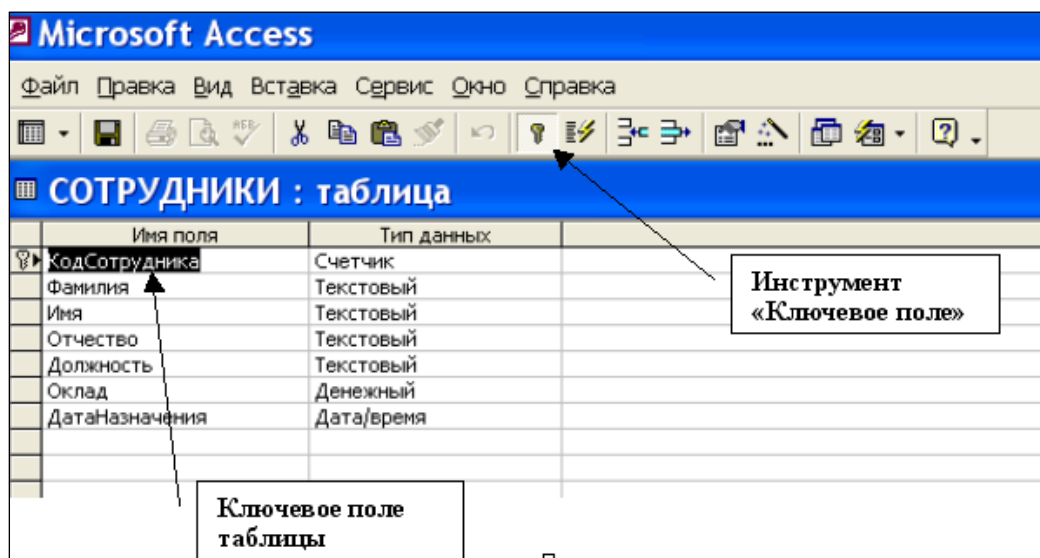


Рис. 21. Задание ключевого поля таблицы

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.*

| СОТРУДНИКИ : таблица | | | | | | |
|--------------------------------|-----------|----------|------------|-----------|-------------|----------------|
| 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 | 3 600,00р. | 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 | Алексанр | 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. | 02.06.2005 |
| 10 | Хлебов | Олег | Олегович | инженер | 4 400,00р. | 30.03.2004 |
| * | (Счетчик) | | | | 0,00р. | |
| Запись: << << 10 >> >> * из 10 | | | | | | |

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 межтабличных связей

Эта операция выполняется с помощью мыши в окне **Схема данных** (рис. 24), которое открывается кнопкой  на панели инструментов в окне базы данных или командой **Сервис – Схема данных**. Одновременно с окном **Схема данных** открывается окно **Добавление таблицы**, в котором нужно выбрать таблицы 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.

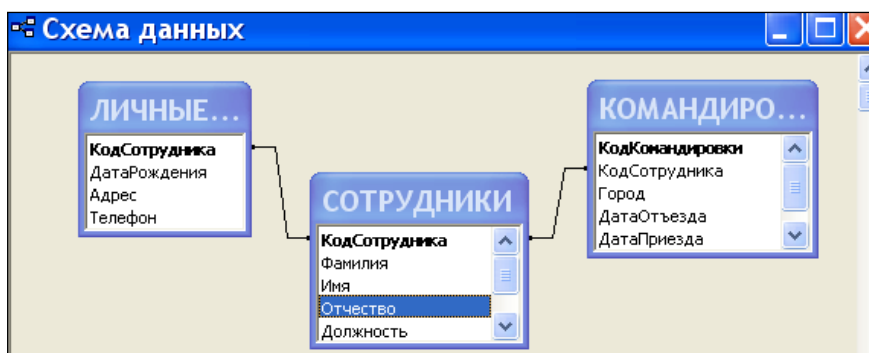


Рис. 24. Связанные таблицы в окне **Схема данных**

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.

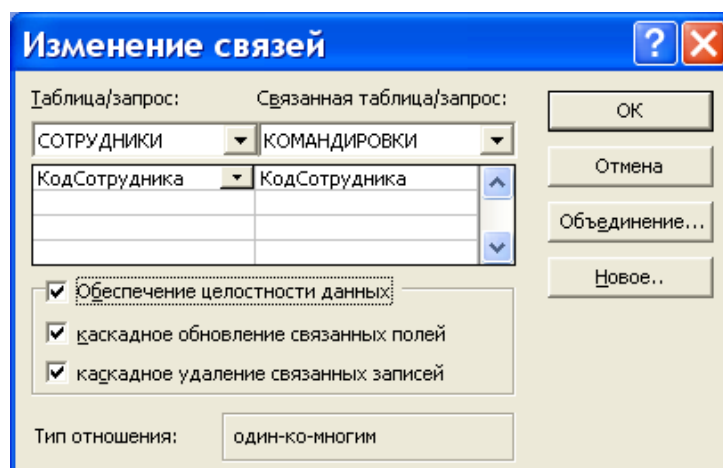


Рис. 25. Диалоговое окно **Изменение связей**

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.

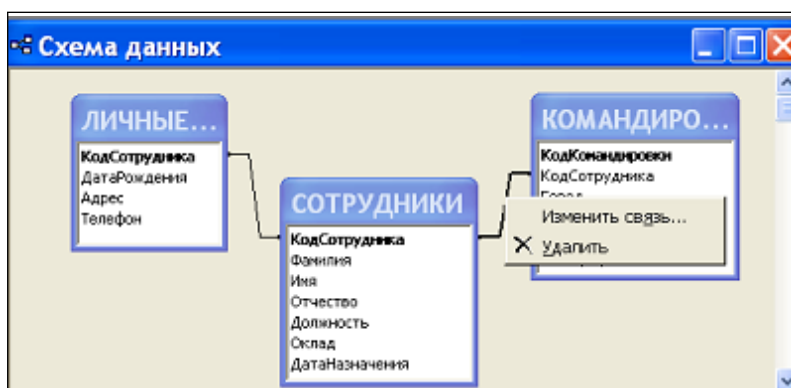


Рис. 26. Контекстное меню связи в окне **Схема данных**

Работа с формами

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.

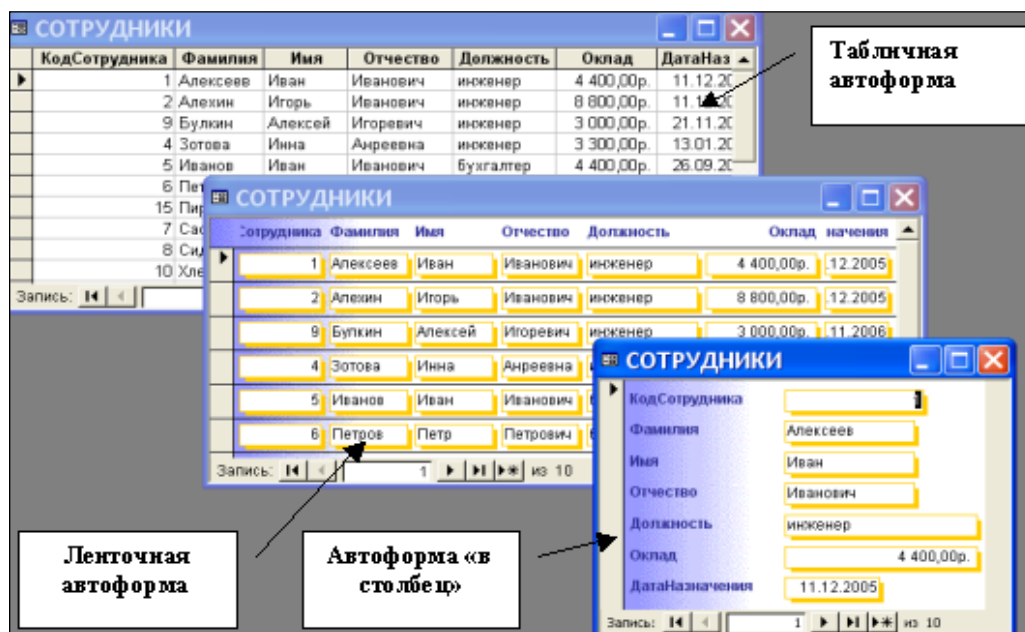


Рис. 27. Три вида автоформ

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).

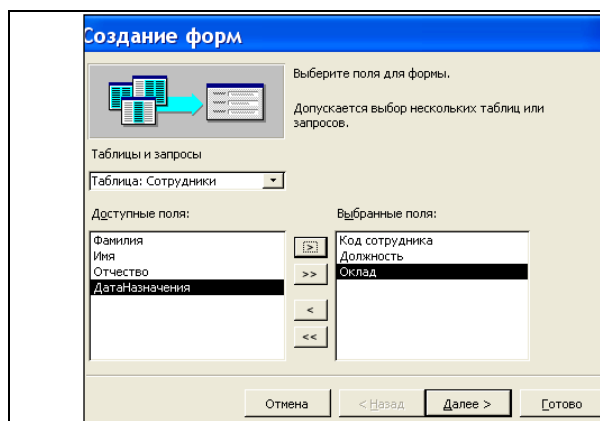


Рис. 28, а. Создание формы в режиме
Мастер форм, шаг 1

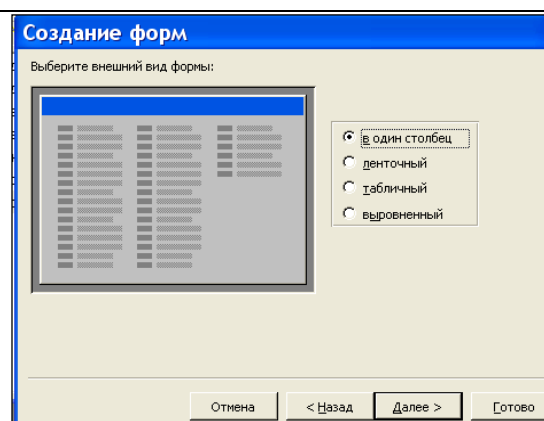


Рис. 28, б. Создание формы в режиме
Мастер форм, шаг 2

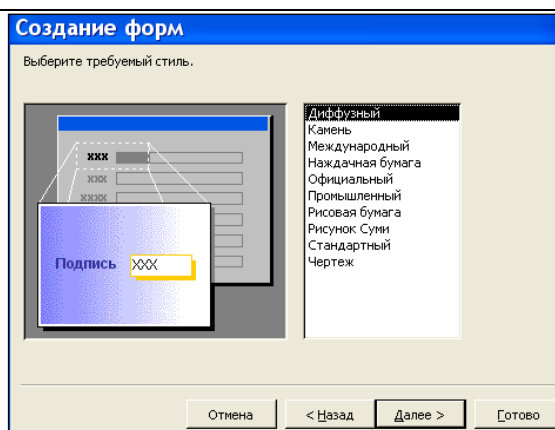


Рис. 28, в. Создание формы в режиме
Мастер форм, шаг 3

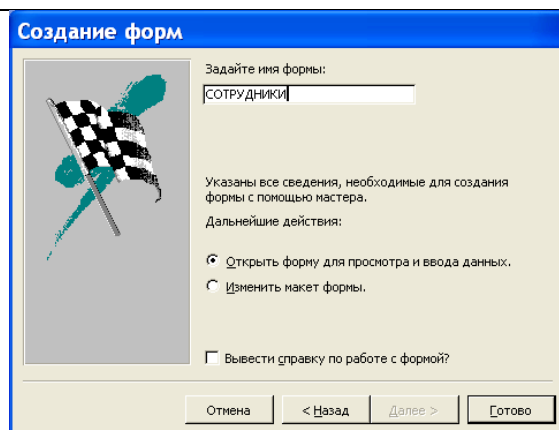


Рис. 28, г. Создание формы в режиме
Мастер форм, шаг 4

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.

КОМАНДИРОВКИ СОТРУДНИКОВ

КодСотрудника: 1

Фамилия: Алексеев

Имя: Иван

Отчество: Иванович

Должность: инженер

| КодКомандировки | Город | ДатаОтъезда | ДатаПриезда |
|-----------------|-----------|-------------|-------------|
| 1 | Москва | 03.12.2005 | 23.01.2006 |
| 12 | Москва | 01.01.2006 | 03.02.2007 |
| * | (Счетчик) | | |

Запись: 2 из 2

Рис. 29. Пример формы, содержащей подчиненную форму

? Вопросы и упражнения для самоконтроля

1. Какие объекты базы данных Microsoft Access используются для хранения данных?
2. Какие объекты базы данных Microsoft Access используются для отбора данных из базовых таблиц в соответствии с какими-либо критериями отбора?
3. Чем результирующая таблица запроса отличается от базовой таблицы базы данных Microsoft Access?
4. Каково назначение экранных форм базы данных Microsoft Access?
5. Какие объекты базы данных Microsoft Access используются для вывода данных из таблиц или запросов на печать?
6. Каково назначение страниц доступа к данным?
7. Для чего используются макросы и модули базы данных Microsoft Access?
8. Укажите основные свойства полей таблицы базы данных Microsoft Access.
9. Какого типа данные может содержать таблица БД Microsoft Access? Охарактеризуйте основные типы данных.
10. В каком режиме работы с таблицей может быть изменена структура таблицы?
11. Для чего используется окно **Схема данных** при работе с таблицами базы данных Microsoft Access?
12. Укажите типы автоформ, которые можно создать для таблицы базы данных Microsoft Access.

4.3. Основы работы с СУБД Microsoft Access. 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.

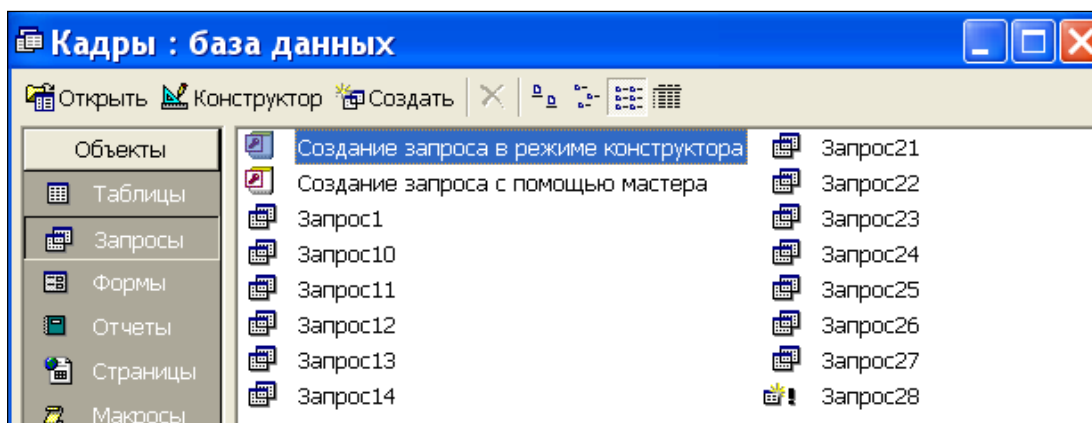


Рис. 31. Запросы в окне базы данных

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.

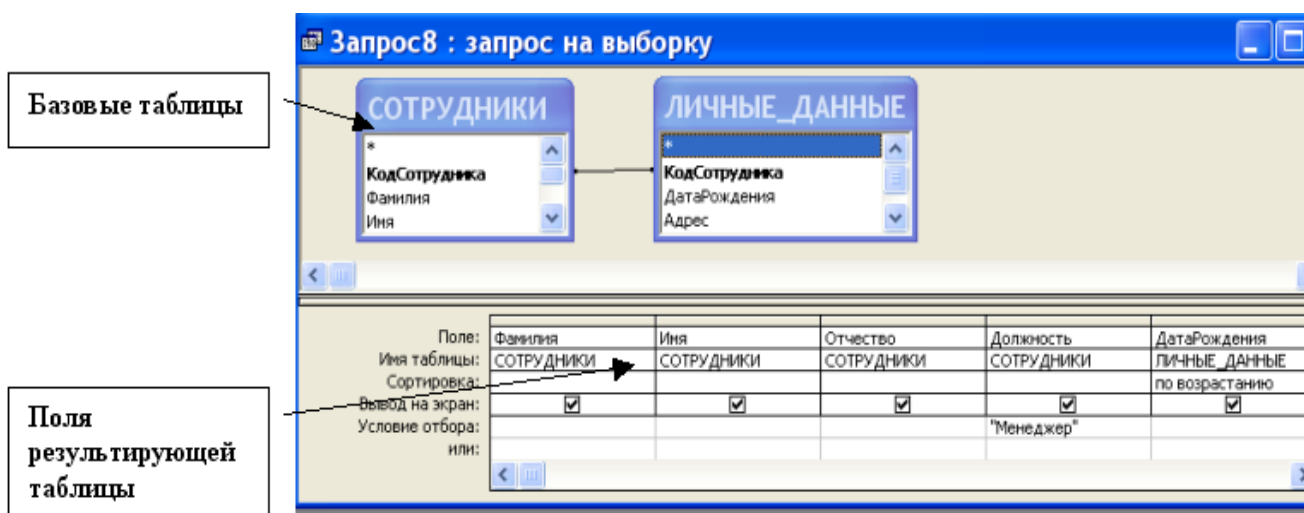


Рис. 32. Запрос в режиме Конструктор

Упорядочение записей в результирующей таблице

Если необходимо, чтобы данные, отображенные 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**.

| | | | | | |
|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|
| 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: | | | | | |

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

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.

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.

Таблица 1. Примеры записи условий отбора записей в запросе

| 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 |

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*.

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.

| | | | | | |
|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 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 |
| или: | | | | | |

Рис. 34. Пример записи условий отбора в запросе

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.

| | | | | | |
|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 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 |

Рис. 35. Пример записи условий отбора в запросе

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).

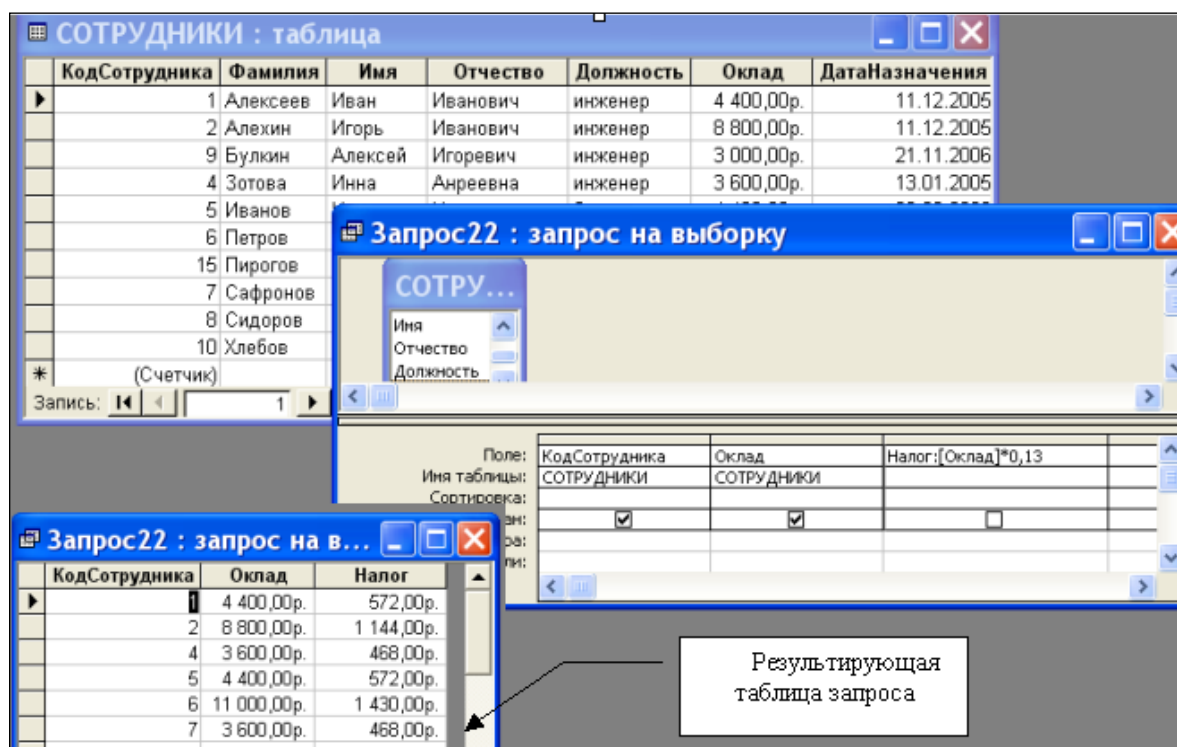


Рис. 36. Запрос с вычисляемым полем

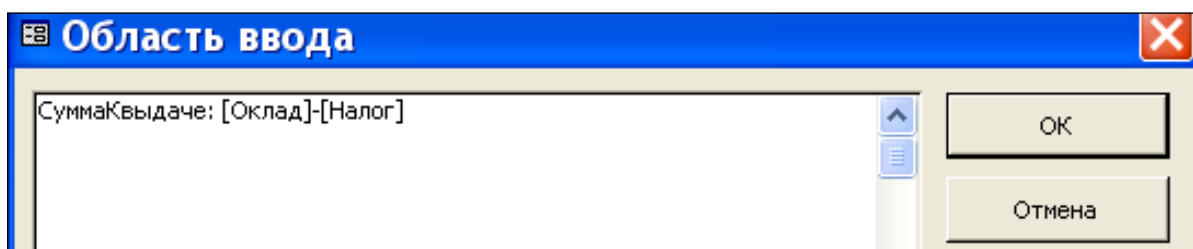


Рис. 37. Фрагмент диалогового окна **Область ввода**

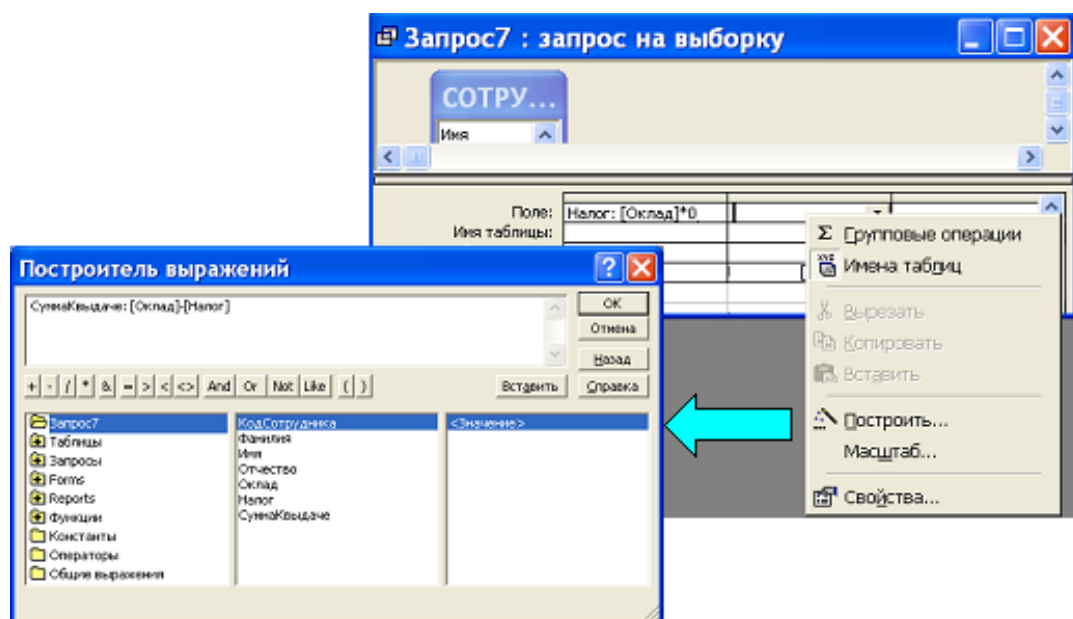


Рис. 38. Окно **Построитель выражений**

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.

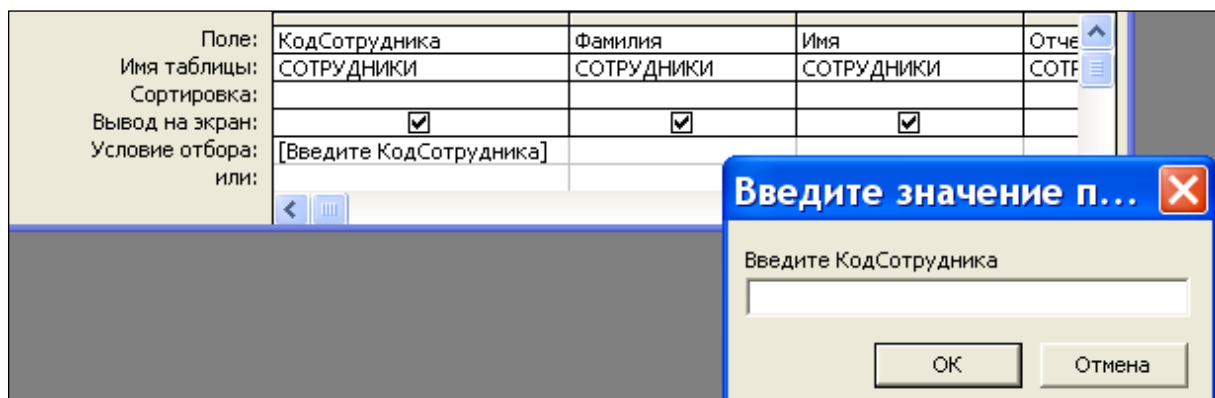


Рис. 39. Запрос с параметром-значением

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.

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.)

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

| 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

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.

| 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: | | | | |

Рис. 42. Создание итогового запроса

| 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. |

Рис. 43. Результат выполнения итогового запроса

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.

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**.

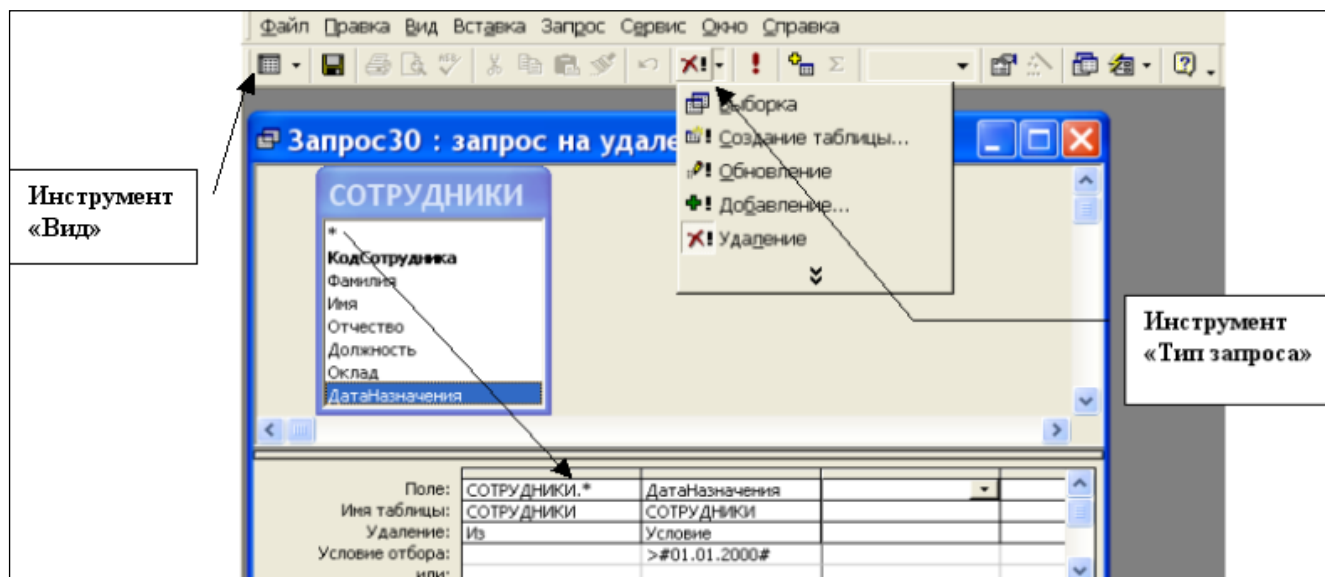


Рис. 44. Создание запроса на удаление записей таблицы

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

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**.

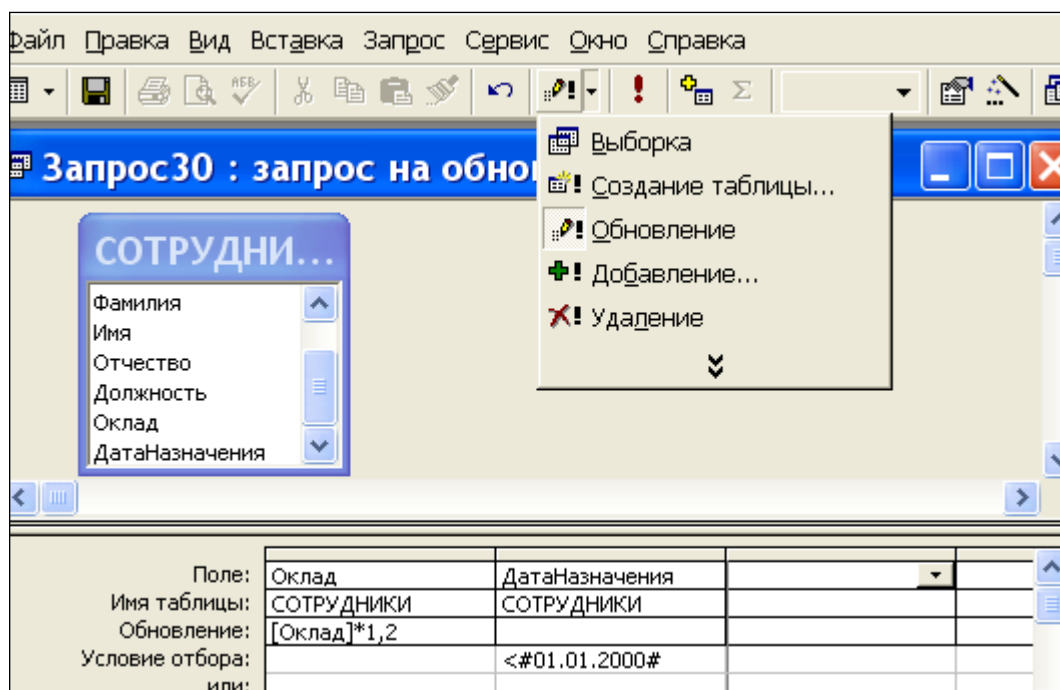


Рис. 45. Создание запроса на обновление записей

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.

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 столбец**). 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*.

Сотрудники

| | |
|----------------|---------------|
| Код Сотрудника | 1 |
| Фамилия | Белова |
| Имя | Мария |
| Должность | Представитель |

а)

Сотрудники

| Код Сотрудника | Фамилия | Имя | Должность |
|----------------|----------|-------|----------------|
| 1 | Белова | Мария | Представитель |
| 2 | Новиков | Павел | Вице-президент |
| 3 | Бабкина | Ольга | Представитель |
| 4 | Воронова | Дарья | Представитель |

б)

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**).

? Вопросы и упражнения для самоконтроля

1. Какого типа запросы используются для отбора записей из одной или нескольких взаимосвязанных таблиц БД Microsoft Access?
2. Приведите примеры записи условий отбора в запросах на выборку.
3. В чем заключается отличие запросов с параметром от простых запросов на выборку?
4. Укажите назначение вычисляемых полей запроса. Как включить в запрос вычисляемое поле?
5. Каково назначение итоговых запросов БД Microsoft Access?
6. Какие функции могут быть использованы для обработки сгруппированных данных в итоговых запросах?
7. Укажите типы запросов на изменение записей. Поясните назначение каждого из типов.
8. Что является результатом выполнения запроса на создание базовой таблицы?
9. В каком режиме можно создать отчет, содержащий данные из нескольких взаимосвязанных таблиц?