



**Современный  
Гуманитарный  
Университет**

**Дистанционное образование**

---

Рабочий учебник

Фамилия, имя, отчество \_\_\_\_\_

Факультет \_\_\_\_\_

Номер контракта \_\_\_\_\_

## **ИНФОРМАТИКА. РАСШИРЕННЫЙ КУРС**

### **ЮНИТА 1**

**БАЗЫ ДАННЫХ НА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ  
И ИХ ИСПОЛЬЗОВАНИЕ**

**МОСКВА 1999**

Разработано И.Б. Глазыриной, А.Д. Супониным, Т.Н. Худенко

Рекомендовано Министерством общего  
и профессионального образования  
Российской Федерации в качестве  
учебного пособия для студентов  
высших учебных заведений

## **КУРС: ИНФОРМАТИКА. РАСШИРЕННЫЙ КУРС**

Юнита 1. Базы данных на персональных компьютерах и их использование.  
Юнита 2. Аппаратные средства и основы управления персональными  
компьютерами.

### **ЮНИТА 1**

Рассмотрены основные понятия, связанные с базами данных, принципы их проектирования и функционирования. На примере ИПС «Кодекс» раскрыты особенности работы с текстовыми базами данных. Описаны возможности и принципы использования СУБД Access и dBase III.

Для студентов Современного Гуманитарного Университета

Юнита соответствует профессиональной образовательной программе № 1

## ОГЛАВЛЕНИЕ

ПРОГРАММА КУРСА .....	5
ЛИТЕРАТУРА .....	6
ПЕРЕЧЕНЬ УМЕНИЙ .....	7
ТЕМАТИЧЕСКИЙ ОБЗОР .....	9
1. Основные понятия теории баз данных .....	9
1.1. Введение .....	9
1.2. Классификация баз данных .....	10
1.3. Структурные элементы базы данных .....	12
1.4. Модели данных .....	13
1.4.1. Иерархическая модель данных .....	14
1.4.2. Сетевая модель данных .....	14
1.4.3. Реляционная модель данных .....	16
1.5. Системы управления базами данных .....	18
1.5.1. СУБД в многопользовательских системах .....	19
1.5.2. Свойства СУБД и базы данных .....	20
1.5.3. Технология использования СУБД .....	20
1.5.4. Обзор СУБД .....	22
2. Проектирование баз данных .....	24
2.1. Понятие предметной области .....	24
2.2. Технология анализа предметной области .....	24
2.2.1. Анализ концептуальных требований и информационных потребностей .....	24
2.2.2. Выявление информационных объектов и связей между ними .....	26
2.2.3. Построение концептуальной модели предметной области .....	29
2.3. Логическое проектирование .....	32
3. СУБД Microsoft Access .....	34
3.1. Связи между таблицами .....	34
3.2. Ключи .....	35
3.3. Макросы, модули и объекты .....	35
3.4. Полезные помощники .....	35
3.4.1. Карточки подсказки .....	36
3.4.2. Мастера .....	36
3.4.3. Конструкторы .....	36
3.5. Запросы и выборки .....	36
3.5.1. Основные типы запросов .....	36
3.5.2. Два способа формирования запросов .....	38
3.6. Формы данных .....	38
3.6.1. Создание формы .....	38
3.6.2. Форма в один столбец .....	39
3.6.3. Табличная форма .....	39
3.6.4. Форма – диаграмма .....	40
3.6.5. Составная форма .....	41
3.6.6. Простая форма .....	41
3.7. Обработка данных с помощью отчетов .....	42
3.7.1. Отчет в один столбец .....	43

3.7.2. Группировка данных и вычисление итогов . . . . .	43
3.7.3. Итоговый отчет . . . . .	43
3.7.4. Табличный отчет . . . . .	43
3.7.5. Почтовые наклейки . . . . .	46
3.7.6. Слияние с MS Word . . . . .	46
3.7.7. Простой отчет . . . . .	46
4. Особенности работы с текстовыми базами данных . . . . .	48
4.1. Структура базы данных . . . . .	49
4.2. Поиск информации в базе данных . . . . .	51
4.2.1. Тематический (проблемный) поиск . . . . .	51
4.2.2. Приблизительный контекстный поиск . . . . .	52
4.2.3. Контекстный поиск . . . . .	53
4.2.4. Универсальный атрибутный поиск . . . . .	54
5. СУБД DBASE III . . . . .	55
ТРЕНИНГ УМЕНИЙ . . . . .	60
ГЛОССАРИЙ*	

---

\* Глоссарий расположен в середине учебного пособия и предназначен для самостоятельного заучивания новых понятий.

## **ТЕМАТИЧЕСКИЙ ПЛАН**

Основные понятия теории баз данных. Введение. Классификация баз данных. Структурные элементы базы данных. Модели данных. Системы управления базами данных.

Проектирование баз данных. Понятие предметной области. Технология анализа предметной области. Анализ концептуальных требований и информационных потребностей. Выявление информационных объектов и связей между ними. Построение концептуальной модели предметной области. Логическое проектирование.

СУБД Microsoft Access. Связи между таблицами. Ключи. Макросы, модули и объекты. Полезные помощники. Запросы и выборки. Формы данных. Обработка данных с помощью отчетов.

Особенности работы с текстовыми базами данных. Структура базы данных. Поиск информации в базе данных. Тематический (проблемный) поиск. Приблизительный контекстный поиск. Контекстный поиск. Универсальный атрибутный поиск.

СУБД dBASE III.

## **ЛИТЕРАТУРА**

### **Базовая**

1. Информатика: Учебник для вузов / Под. ред. проф Н.В. Макаровой. М., 1997.
2. Экономическая информатика: Учебник для вузов / Под. ред. проф. В.В. Евдокимова. СПб., 1997.

### **Дополнительная**

3. Каратыгин С., Тихонов А., Долголаптев В. Базы данных. В 2 т. М., 1995.
4. Кагаловский М.Р. Технология баз данных на персональных ЭВМ. М., 1992.
5. Диго С.М. Проектирование и использование баз данных. М., 1995.
6. Аладьев В.З. и др. Основы информатики. М., 1998.
7. Мейер Д. Теория реляционных баз данных. Пер. с англ. М., 1987.
8. Мартин Дж. Организация баз данных в вычислительных системах. М., 1980.
5. Microsoft Office для Windows 95. М., 1997.
9. Джонс Э., Саттон Д. Библия пользователя Microsoft Office Professional для Windows 95. К., 1996.
10. Информационные системы в экономике / Под ред. проф. В.В. Дика. М., 1996.
11. Компьютерные технологии обработки информации: Учеб. пособие для вузов / под ред. С.В. Назарова. М., 1995.
12. Якубайтис Э.А. Информационные сети и системы. Справочная книга. М., 1996.
13. Першиков В.И., Савинков В.М. Толковый словарь по информатике. М., 1995.

### ПЕРЕЧЕНЬ УМЕНИЙ

№ п/п	Умение	Алгоритмы
1	Описание логической структуры записи файла	<ol style="list-style-type: none"> <li>1. Определить последовательность расположения полей записи.</li> <li>2. Присвоить каждому полю уникальное имя.</li> <li>3. Определить тип поля.</li> <li>4. Определить длину (размер) поля.</li> <li>5. Указать точность поля.</li> </ol>
2	Построение иерархической схемы базы данных	<ol style="list-style-type: none"> <li>1. Изучить правила организации связи между объектами иерархической модели данных.</li> <li>2. Определить узлы (объекты) модели.</li> <li>3. Выделить главный (корневой) узел.</li> <li>4. Задать отношение подчиненности между узлами.</li> </ol>
3	Построение таблицы реляционной модели базы данных	<ol style="list-style-type: none"> <li>1. Изучить правила создания реляционных таблиц.</li> <li>2. Определить объекты модели.</li> <li>3. Описать структуру объекта.</li> <li>4. Сформировать столбцы таблицы.</li> <li>5. Присвоить уникальное имя каждому столбцу.</li> <li>6. Выделить ключ.</li> <li>7. Определить тип и длину элемента столбца.</li> </ol>
4	Создание БД в системе dBASE III	<ol style="list-style-type: none"> <li>1. Определить структуру БД.</li> <li>1. В главном меню системы выбрать пункт «Создать».</li> <li>2. В раскрывшемся подменю выбрать пункт «файл БД».</li> <li>3. Указать дисковод и каталог для размещения файла.</li> <li>4. В ответ на запрос СУБД ввести имя файла.</li> <li>5. В появившуюся на экране таблицу ввести информацию о структуре файла.</li> <li>6. Выйти из режима создания, нажав клавиши [Ctrl] + [End].</li> </ol>

№ п/п	Умение	Алгоритмы
5	Использование «Помощника» в информационной системе «Кодекс» для создания запроса с использованием логических операторов	<ol style="list-style-type: none"> <li>1. Нажать кнопку [fx] для раскрытия окна-диалога «Помощника».</li> <li>2. В поле ввода запроса ввести первую фразу поискового запроса, заключив ее в кавычки.</li> <li>3. Нажать кнопку оператора «ИЛИ» (вертикальная черта) или оператора «И» (&amp;).</li> <li>4. Ввести вторую фразу поискового запроса, заключив ее в кавычки.</li> <li>5. Нажать кнопку команды «ОК».</li> <li>6. Нажать кнопку «Поиск» в раскрывшемся окне «Новая выборка».</li> <li>7. Нажать кнопку «Список» для вывода выборки в информационное окно.</li> </ol>
6	Создание запроса-выборки в СУБД ACCESS	<ol style="list-style-type: none"> <li>1. В окне БД нажать кнопку «Запрос».</li> <li>2. Нажать кнопку «Создать».</li> <li>3. В диалоге «Создание запроса» нажать кнопку «Новый запрос».</li> <li>4. В диалоге «Добавление таблицы» указать таблицу для выборки.</li> <li>5. Указать поля, по которым будет проводиться поиск, перетаскив их из списка полей выбранной таблицы в ячейки строки «Поле» окна «Запроса».</li> <li>6. В ячейки строки «Условие отбора» поместить значения условий.</li> <li>7. Выполнить запрос, щелкнув на кнопке пиктографического меню «Выполнить».</li> </ol>

## ТЕМАТИЧЕСКИЙ ОБЗОР\*

### 1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ БАЗ ДАННЫХ

#### 1.1. Введение

Современные информационные системы, основанные на концепции интеграции данных, характеризуются огромными объемами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

**Информационная система** - система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

*Цель любой информационной системы* – обработка данных об объектах реального мира. Основой информационной системы является база данных. В широком смысле слова база данных – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под **предметной областью** принято понимать часть реального мира, подлежащего изучению для организации управления и в конечном счете автоматизации, например, предприятие, вуз и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. Большое значение при этом приобретает структурирование данных.

**Структурирование данных** – это введение соглашений о способах представления данных.

*Неструктурированными* называют данные, записанные, например, в текстовом файле.

Ниже приведен пример неструктурированных и структурированных данных, содержащих сведения о студентах (номер личного дела, фамилию, имя, отчество и год рождения).

#### **Неструктурированные данные:**

---

Личное дело № 16493. Сергеев Петр Михайлович, дата рождения 1 января 1976 г.; Л/д № 16593, Петрова Анна Владимировна, дата рожд. 15 марта 1975 г.; № личн. дела 16693, д.р. 14.04.76, Анохин Андрей Борисович.

---

Легко убедиться, что сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде.

---

\* Жирным шрифтом выделены новые понятия, которые необходимо усвоить. Знание этих понятий будет проверяться при тестировании.

#### Структурированные данные:

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75
16693	Анохин	Андрей	Борисович	14.04.76

Чтобы автоматизировать поиск и систематизировать эти данные, необходимо выработать определенные соглашения о способах представления данных, т.е. дату рождения нужно записывать одинаково для каждого студента, она должна иметь одинаковую длину и определенное место среди остальной информации. Эти же замечания справедливы и для остальных данных (номер личного дела, фамилия, имя, отчество).

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария – системы управления базами данных.

**База данных (БД)** – это поименованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

**Объектом** называется элемент предметной области, информацию о котором мы сохраняем.

Объект может быть реальным (например, человек, изделие или населенный пункт) и абстрактным (например, событие, счет покупателя или изучаемый студентами курс). Так, в области продажи автомобилей примерами объектов могут служить МОДЕЛЬ АВТОМОБИЛЯ, КЛИЕНТ и СЧЕТ. На товарном складе – это ПОСТАВЩИК, ТОВАР, ОТПРАВЛЕНИЕ и т. д.

**Система управления базами данных (СУБД)** – это комплекс программных и языковых средств, предназначенных для создания, ведения и совместного применения баз данных многими пользователями.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данными, хранимыми в базе.

## 1.2. Классификация баз данных

По технологии обработки данных базы данных подразделяются на централизованные и распределенные.

**Централизованная база данных** хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом

сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

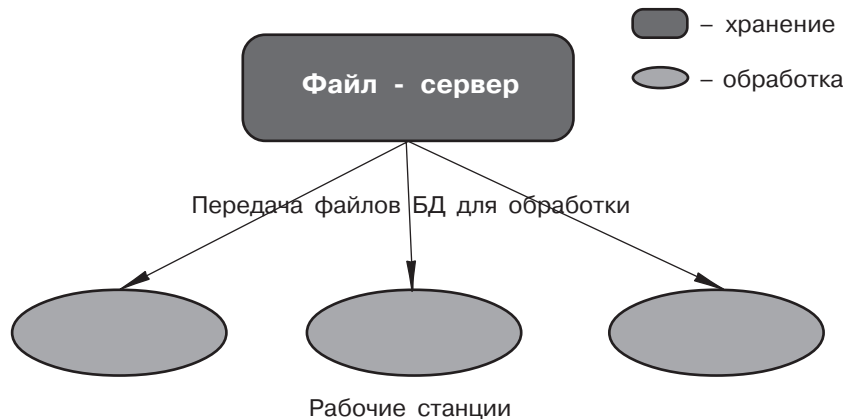
**Распределенная база данных** состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По способу доступа к данным базы данных разделяются на базы данных с локальным доступом и базы данных с удаленным (сетевым) доступом.

Системы централизованных баз данных с сетевым доступом предполагают различные **архитектуры** подобных систем:

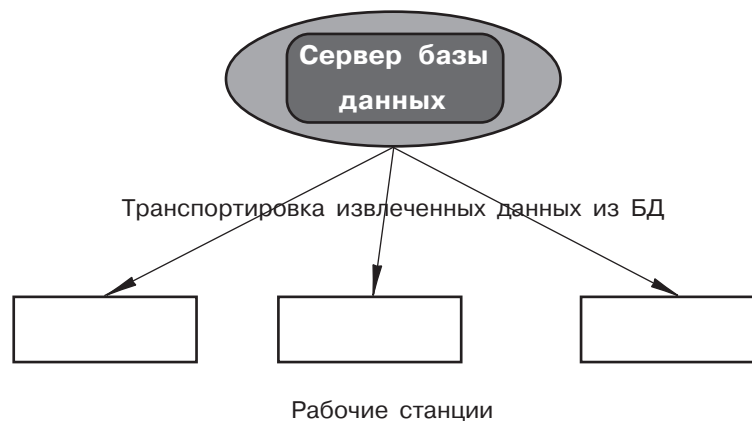
- файл-сервер;
- клиент-сервер.

**Файл-сервер.** Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монополично. Концепция файл-сервер условно отображена на рис.1.1.



**Рис. 1.1. Схема обработки информации в БД по принципу файл-сервер**

**Клиент-сервер.** В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL. Концепция клиент-сервер условно изображена на рис. 1.2.



**Рис. 1.2. Схема обработки информации в БД по принципу клиент-сервер**

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица) (рис. 1.3).

Имя поля 1	Имя поля 2	Имя поля 3	Имя поля 4

Поле

Запись

**Рис. 1.3. Основные структурные элементы БД**

### 1.3. Структурные элементы базы данных

**Поле** – элементарная единица логической организации данных, которая соответствует неделимой единице информации – реквизиту. Для описания поля используются следующие характеристики:

*имя*, например, Фамилия, Имя, Отчество, Дата рождения;

*тип*, например, символьный, числовой, календарный;

*длина*, например, 15 байт, причем будет определяться максимально возможным количеством символов;

*точность* для числовых данных, например два десятичных знака для отображения дробной части числа.

**Запись** – совокупность логически связанных полей.

**Экземпляр записи** – отдельная реализация записи, содержащая конкретные значения ее полей.

**Файл (таблица)** – совокупность экземпляров записей одной структуры. Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики, как это показано на рис. 1.4 и 1.5.

Имя файла					
Поле		Признак ключа	Формат поля		
Имя (обозначение)	Полное наименование		Тип	Длина	Точность (для чисел)
имя 1					
...					
имя п					

**Рис. 1.4. Описание логической структуры записи файла**

В структуре записи файла указываются поля, значения которых являются ключами: первичными (ПК) и вторичными (ВК).

**Первичный ключ (ПК)** – это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей – составным ключом.

**Вторичный ключ (ВК)** - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение вторичного ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по вторичному – несколько.

Имя файла: СТУДЕНТ					
Поле		Признак ключа	Формат поля		
Обозначение	Наименование		Тип	Длина	Точность
Номер	№ личного дела	*	Симв	5	
Фамилия	Фамилия студента		Симв	15	
Имя	Имя студента		Симв	10	
Отчество	Отчество студента		Симв	15	
Дата	Дата рождения		Дата	8	

**Рис. 1.5. Описание логической структуры записи файла СТУДЕНТ**

## 1.4. Модели данных

Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

**Модель данных** – совокупность структур данных и операций их обработки.

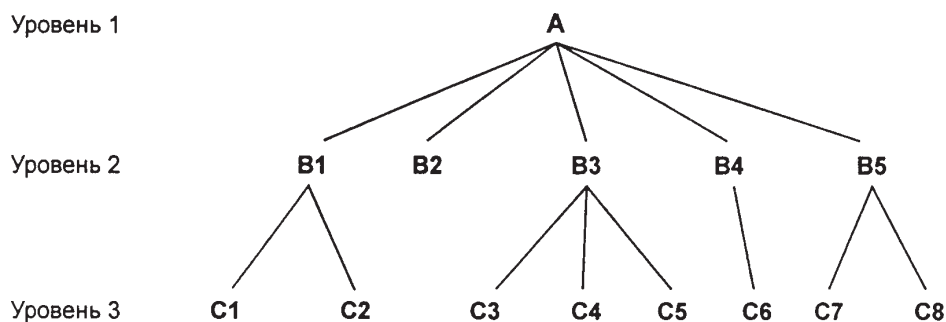
СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную.

#### 1.4.1. Иерархическая модель данных

**Иерархическая модель** организует данные в виде древовидной структуры.

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. Дерево представляет собой иерархию элементов, называемых узлами. Узел – это совокупность атрибутов данных, описывающих некоторый объект. На самом верхнем уровне иерархии имеется один и только один узел – корень. Каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемым исходным для данного узла. Ни один элемент не имеет более одного исходного. Каждый элемент может быть связан с одним или несколькими элементами на более низком уровне. Они называются порожденными (рис 1.6).



**Рис.1.6. Графическое изображение иерархической структуры БД**

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 1.6, для записи C4 путь проходит через записи A и B3.

Пример, представленный на рис. 1.7, иллюстрирует использование иерархической модели базы данных. Для рассматриваемого примера иерархическая структура правомерна, так как каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) институту.

#### 1.4.2. Сетевая модель данных

**Сетевая модель** организует данные в виде сетевой структуры.

Структура называется сетевой, если в отношениях между данными порожденный элемент имеет более одного исходного.



Рис 1.7. Пример иерархической структуры БД

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

На рис. 1.8 изображена сетевая структура базы данных в виде графа.

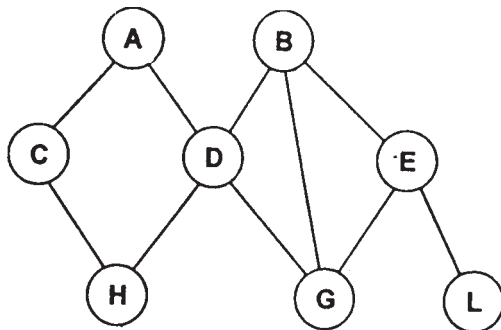


Рис. 1.8. Графическое изображение сетевой структуры

Примером сложной сетевой структуры может служить структура базы данных, содержащей сведения о студентах, участвующих в научно-исследовательских работах (НИРС). Возможно участие одного студента в нескольких НИРС, а также участие нескольких студентов в разработке одной НИРС. Графическое изображение описанной в примере сетевой структуры, состоящей только из двух типов записей, показано на рис. 1.9. Единственное

отношение представляет собой сложную связь между записями в обоих направлениях.



Рис. 1.9. Пример сетевой структуры БД

#### 1.4.3. Реляционная модель данных

Понятие “реляционный” (англ. relation – отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Например, реляционной таблицей можно представить информацию о студентах, обучающихся в вузе (рис. 1.10).

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Анохин	Андрей	Борисович	14.04.76	111

Рис. 1.10. Пример реляционной таблицы

**Реляционная модель** данных является совокупностью взаимосвязанных двумерных таблиц – объектов модели.

Связи между двумя логически связанными таблицами в реляционной модели устанавливаются по равенству значений одинаковых атрибутов этих таблиц.

Каждая реляционная таблица представляет собой двумерный массив и обладает следующими *свойствами*:

- каждый элемент таблицы – один элемент данных;
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;

- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

При описании реляционной модели часто используют следующие термины: отношение, кортеж, домен.

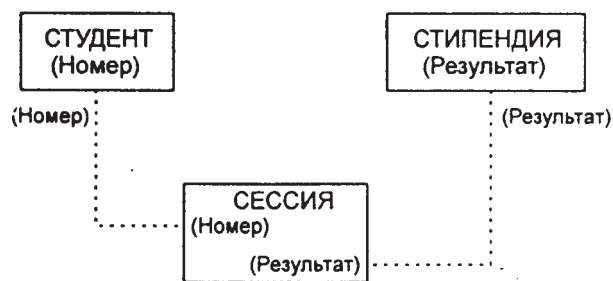
Отношения представлены в виде таблиц, строки которых соответствуют записям (кортежам), а столбцы – полям, атрибутам отношений (доменам).

Поле, каждое значение которого однозначно определяет соответствующую запись, называется простым ключом (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет составной ключ.

В примере, показанном на рис. 1.10, ключевым полем таблицы является “№ личного дела”.

Между двумя реляционными таблицами могут быть сформированы связи. Различные таблицы, могут быть связаны между собой через общее поле данных.

На рис. 1.11 показан пример реляционной модели, построенной на основе отношений: СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ.



**Рис 1.11. Пример реляционной модели**

Таблица СТУДЕНТ имеет поля: Номер, Фамилия, Имя, Отчество, Дата рождения, Группа;

СЕССИЯ - Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат;

СТИПЕНДИЯ - Результат, Процент.

Таблицы СТУДЕНТ И СЕССИЯ имеют совпадающие ключи (Номер), что дает возможность легко организовать связь между ними. Таблица СЕССИЯ имеет первичный ключ Номер и содержит внешний ключ Результат, который обеспечивает ее связь с таблицей СТИПЕНДИЯ.

Благодаря имеющимся связям достигаются следующие преимущества:

1. Удастся избежать дублирования информации. Все необходимые данные можно хранить только в одной таблице. Так, например, нет необходимости в таблице СЕССИЯ хранить номер группы каждого студента, сдающего экзамены, достаточно задать связь с таблицей СТУДЕНТ.

2. В реляционных базах данных легко производить изменения. Если в таблице СЕССИЯ изменить какие-нибудь значения, то правильная информация автоматически будет связана с другими таблицами, ссылающимися на первую (например, таблица СТИПЕНДИЯ).

3. В нереляционных базах данных сложно передать все имеющиеся зависимости, т.е. связать друг с другом данные из различных таблиц. Реляционная база данных выполняет все эти действия автоматически.

4. В реляционных базах данных удается легко избежать установления ошибочных связей между различными таблицами данных, а необходимый объем памяти сокращен до минимума.

### 1.5. Системы управления базами данных

Система управления базами данных представляет собой пакет программ, посредством которого реализуется централизованное управление базой данных и обеспечивается доступ к данным. СУБД выступает в качестве интерфейса между пользователями и БД.

**СУБД обеспечивает** программные средства для создания, загрузки, запроса и обновления данных, контролирует действия, связанные с вводом-выводом данных, решает вопросы совместного их использования и защиты.

**СУБД служит** для поддержания базы данных в актуальном состоянии и обеспечивает эффективный доступ пользователей к содержащимся в ней данным в рамках предоставленных пользователям полномочий.

По степени универсальности различают два класса СУБД:

- 1) системы общего назначения;
- 2) специализированные системы.

СУБД общего назначения не ориентированы на какую-либо предметную область или на информационные потребности какой-либо группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной системе, и поставляется многим пользователям как коммерческое изделие. Такие СУБД обладают средствами настройки на работу с конкретной базой данных. Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, основанных на технологии баз данных, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности и даже определенная функциональная избыточность.

Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего назначения.

**СУБД общего назначения** – это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

- Они позволяют определять структуру создаваемой базы, инициализировать ее и производить начальную загрузку данных.
- Они управляют полномочиями пользователей на доступ к БД, организуют параллельный доступ к ней нескольким пользователям.
- Они обеспечивают защиту логической и физической целостности данных - защиту от разрушений.

*СУБД поддерживают один из возможных типов моделей данных – сетевую, иерархическую или реляционную, которые являются одним из важнейших признаков классификации СУБД.*

СУБД обеспечивают многоцелевой характер использования базы данных, защиту и восстановление данных. Наличие развитых диалоговых средств и языка запросов высокого уровня делает СУБД удобным средством для конечного пользователя.

**Основными средствами СУБД являются:**

- средства задания (описания) структуры базы данных;
- средства конструирования экранных форм, предназначенных для ввода данных, просмотра и их обработки в диалоговом режиме;

средства создания запросов для выборки данных при заданных условиях, а также выполнения операций по их обработке;

средства создания отчетов из базы данных для вывода на печать результатов обработки в удобном для пользователя виде;

языковые средства – макросы, встроенный алгоритмический язык (Dbase, Visual Basic или другой), язык запросов (QBE – Query By Example, SQL) и т.п., которые используются для реализации нестандартных алгоритмов обработки данных, а также процедур обработки событий в задачах пользователя;

средства создания приложений пользователя (генераторы приложений, средства создания меню и панелей управления приложениями), позволяющие объединить различные операции работы с базой данных в единый технологический процесс.

#### 1.5.1. СУБД в многопользовательских системах

База данных, как правило, содержит данные, необходимые многим пользователям. Получение одновременного доступа нескольких пользователей к общей базе данных возможно при установке СУБД в локальной сети персональных компьютеров и создании **многопользовательской базы данных** (рис. 1.12).

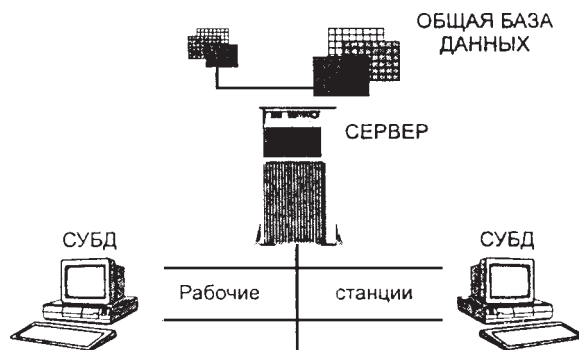


Рис. 1.12. СУБД в многопользовательской системе

В сети СУБД следит за разграничением доступа разных пользователей к общей базе данных и обеспечивает защиту данных при одновременной работе пользователей с общими данными. Автоматически обеспечивается защита данных от одновременной их корректировки несколькими пользователями-клиентами.

В сети с файловым сервером база данных может размещаться на сервере. При этом СУБД загружается и осуществляет обработку данных базы на рабочих станциях пользователей. Концепция файлового сервера в локальной сети обеспечивается рядом сетевых операционных систем. Наиболее популярными являются Microsoft Windows NT и NetWare Novell.

В сети, поддерживающей концепцию «клиент-сервер», используется сервер баз данных, который располагается на мощной машине, выполняет обработку данных, размещенных на сервере, и отвечает за их целостность и сохранность. Для управления базой данных на сервере используется язык структурированных запросов SQL (Structured Queries Language). На рабочих

станциях-клиентах работает СУБД-клиент. Пользователи могут взаимодействовать не только со своими локальными базами, но и с данными, расположенными на сервере. СУБД-клиент, в которой поддерживается SQL, в полном объеме может посылать на сервер запросы SQL, получать необходимые данные, а также посылать обновленные данные.

При этом с общей базой данных могут работать СУБД разного типа, установленные на рабочих станциях, если в них поддерживается SQL.

Подключение из СУБД к серверам баз данных SQL может быть осуществлено с помощью драйверов ODBC. ODBC (Open Database Connectivity – открытый стандарт доступа к базам данных) поддерживает стандартный протокол для серверов баз данных SQL.

### 1.5.2. Свойства СУБД и базы данных

К основным свойствам СУБД и базы данных можно отнести:

- отсутствие дублирования данных в различных объектах модели, обеспечивающее однократный ввод данных и простоту их корректировки;
- непротиворечивость данных;
- целостность БД;
- возможность многоаспектного доступа;
- всевозможные выборки данных и их использование различными задачами и приложениями пользователя;
- защиту и восстановление данных при аварийных ситуациях, аппаратных и программных сбоях, ошибках пользователя;
- защиту данных от несанкционированного доступа средствами разграничения доступа для различных пользователей;
- возможность модификации структуры базы данных без повторной загрузки данных;
- обеспечение независимости программ от данных, позволяющее сохранить программы при модификации структуры базы данных;
- реорганизацию размещения данных базы на машинном носителе для улучшения объемно-временных характеристик БД;
- наличие языка запросов высокого уровня, ориентированного на конечного пользователя, который обеспечивает вывод информации из базы данных по любому запросу и предоставление ее в виде соответствующих отчетных форм, удобных для пользователя.

### 1.5.3. Технология использования СУБД

СУБД является основой создания практических приложений пользователя для различных предметных областей.

**Критерии выбора СУБД пользователем.** Выбор СУБД для практических приложений пользователем определяется многими факторами, к которым относятся:

- имеющееся техническое и базовое программное обеспечение, их конфигурация, оперативная и дисковая память;
- потребности разрабатываемых приложений пользователя;
- тип поддерживаемой модели данных, специфика предметной области, топология информационно-логической модели;
- требования к производительности при обработке данных;
- наличие в СУБД необходимых функциональных средств;
- наличие русифицированной версии СУБД;
- уровень квалификации пользователей и наличие в СУБД диалоговых средств разработки и взаимодействия с БД.

**Установка СУБД.** СУБД является программным продуктом, поставляемым в виде пакета прикладных программ, который должен быть установлен (инсталлирован) на компьютер с учетом его конфигурации, ресурсов и операционной системы, а также требований к набору функций.

**Процесс поэтапного внедрения.** После установки СУБД можно осуществлять создание БД, в том числе задавать структуру БД, производить ввод данных, а также выполнять любые действия, предусмотренные функциональными возможностями СУБД. Следует заметить, что современные СУБД для ПК обладают достаточной гибкостью. Это позволяет на самых ранних этапах разработки приложений пользователя приступать к созданию отдельных частей БД. Такая БД по мере углубления разработки может легко расширяться и модифицироваться. Таким образом, облегчается ускоренное освоение персоналом технологии работы с БД, изучение возможностей СУБД и поэтапное внедрение.

**Разработка структуры базы данных.** Разработка приложений на основе СУБД предполагает подготовку решений по структуре БД. Эти решения непосредственно связаны с внемашиной сферой – с описанием внемашиной информационной базы, ее документов, содержащих необходимую информацию, а также с постановкой и алгоритмизацией задач по обработке этой информации.

На начальном этапе разработки структуры БД целесообразно построение информационно-логической модели, отражающей логическую структуру информации предметной области. Такая модель, отвечающая требованиям нормализации данных, является основой создания реляционных баз данных.

**Создание базы данных средствами СУБД.** В соответствии с разработанной структурой базы данных осуществляется ее создание средствами СУБД на машинном носителе и ввод в эксплуатацию. Для обеспечения процессов создания БД и ее эксплуатации необходимо знание возможностей инструментальных средств СУБД. При этом следует руководствоваться рекомендациями по технологии использования средств СУБД. Такая технология должна определять все необходимые процессы, включая первоначальный ввод, загрузку БД и контроль данных, выполнение операций по внесению изменений, реализацию запросов для получения нужных справок, восстановление БД и т.п. Одним из важнейших этапов этой технологии является подготовка экранных форм ввода-вывода для загрузки информации с документов внемашиной сферы в базу данных, корректировки данных и их просмотра.

**Обработка данных средствами СУБД.** Добавление, удаление, изменение и выборка данных производится при помощи языка запросов, встроенного алгоритмического языка и других средств СУБД. Реализация запросов обеспечивается диалоговой системой команд с меню или запросами по примеру QBE (Query By Example). В первом случае отдельный запрос выполняется одной или несколькими командами языка СУБД. Последовательность команд языка СУБД образует программу – командный файл (СУБД Dbase). Во втором – для выполнения запроса пользователь выбирает последовательно один или несколько пунктов меню или указывает в запросе пример (образец), по которому составляется запрос, а также при необходимости указывает условия выбора и операции вычисления, которые необходимо выполнять с данными (СУБД Paradox, Access). Последовательность команд меню и запросов может быть запомнена в программе-макросе и в дальнейшем выполнена так же, как командный файл.

СУБД может иметь включающий или базовый язык программирования. В СУБД с включающим языком используется один из универсальных алгоритмических языков (C, Pascal и т. п.). Прикладная программа, написанная

на включающем языке, может инициировать команды СУБД. В СУБД с базовым языком применяется собственный алгоритмический язык, позволяющий кроме операций манипулирования данными выполнять различные вычисления и обработку данных.

**Стандартным реляционным языком запросов** является язык структурированных запросов SQL (Structured Queries Language).

Язык запросов SQL реализован в целом ряде популярных СУБД для различных типов ЭВМ либо как базовый, либо как альтернативный. В силу своего широкого использования является международным стандартом языка запросов. Язык SQL предоставляет развитые возможности как конечным пользователям, так и специалистам в области обработки данных.

Совместимость с SQL-системами играет большую роль, когда предполагается проведение работы с корпоративными данными. Access и Paradox for Windows работают с источниками SQL-данных, совместимых с системой ODBC (Open Database Connectivity – открытое соединение баз данных). FoxPro (for DOS и for Windows) поставляются с дополнительными библиотеками, которые обеспечивают доступ к SQL-базам данных, способным работать совместно с системой ODBC.

Можно напрямую управлять базами данных Access с помощью языка SQL и передавать сквозные SQL-запросы совместимым со спецификацией ODBC SQL-базам данных, таким, как MS SQL Server и Oracle.

#### 1.5.4. Обзор СУБД

Рынок программного обеспечения ПК располагает большим числом разнообразных по своим функциональным возможностям коммерческих систем управления базами данных общего назначения, а также средствами их окружения практически для всех массовых моделей машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и безопасности, что дает возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил на низкоуровневое программирование.

Продукты, функционирующие в среде WINDOWS, выгодно отличаются удобством пользовательского интерфейса и встроенными средствами повышения производительности.

Наибольшую популярность завоевали следующие СУБД: dBASE, FoxPro, Paradox, Access, Oracle.

**Производительность СУБД.** Производительность СУБД оценивается:

- временем выполнения запросов;
- скоростью поиска информации;
- временем выполнения операций импортирования базы данных из других форматов;
- скоростью создания индексов и выполнения таких массовых операций, как обновление, вставка, удаление данных;
- максимальным числом параллельных обращений к данным в многопользовательском режиме;
- временем генерации отчета.

Достаточно быстрой СУБД является FoxPro 2.6, однако она не обладает средствами соблюдения целостности данных в отличие от более медленной СУБД Access 2.0.

**Обеспечение целостности данных на уровне базы данных.** Эта характеристика подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной. Должны быть установлены правила целостности, и они должны храниться вместе с базой данных и соблюдаться на глобальном уровне.

Access и Paradox for Windows лучше других СУБД обеспечивают надежность сохранения целостности данных на уровне базы данных; правила хранятся вместе с базой данных и автоматически соблюдаются.

**Обеспечение безопасности.** Некоторые СУБД предусматривают средства обеспечения безопасности данных. Такие средства обеспечивают выполнение следующих операций:

- шифрование прикладных программ;
- шифрование данных;
- защиту паролем;
- ограничение уровня доступа (к базе данных, к таблице, к словарю, для пользователя).

Самый высокий уровень безопасности данных реализован в СУБД dBASE IV. Администратор может назначить системе различные права доступа на уровне файла, поля, а также организовать автоматическое шифрование данных.

Хорошими характеристиками обеспечения безопасности отличается Access 2.0. Он предусматривает назначение паролей для индивидуальных пользователей или групп пользователей и присвоение различных прав доступа отдельно таблицам, запросам, отчетам, макрокомандам или новым объектам на уровне пользователя или группы.

**Работа в многопользовательских средах.** Практически все рассматриваемые СУБД предназначены для работы в многопользовательских средах, но обладают для этого различными возможностями.

Обработка данных в многопользовательских средах предполагает выполнение программным продуктом следующих функций:

- блокировку базы данных, файла, записи, поля;
- идентификацию станции, установившей блокировку;
- обновление информации после модификации;
- контроль за временем и повторение обращения;
- обработку транзакций (транзакция – последовательность операций пользователя над базой данных, которая сохраняет ее логическую целостность);
- работу с сетевыми системами (LAN Manager, NetWare, Unix).

Хорошими возможностями для работы в многопользовательских средах обладают СУБД Paradox for DOS 4.5, Access 2.0 и dBASE IV.

**Импорт-экспорт.** Эта характеристика отражает:

- возможность обработки СУБД информации, подготовленной другими программными средствами;
- возможность использования другими программами данных, сформированных средствами рассматриваемой СУБД.

Особый интерес представляют следующие форматы файлов: ASCII-файлы, .DBF, .WK\*, .XLS.

Все рассматриваемые здесь СУБД обладают хорошими возможностями импорта-экспорта данных.

## 2. ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

### 2.1. Понятие предметной области

Каждая информационная система в зависимости от ее назначения имеет дело с частью реального мира, которую принято называть предметной областью (ПО) системы. ПО может относиться к любому типу организаций: банк, университет, завод, магазин и т.д.

Предметная область информационной системы - это совокупность реальных объектов (сущностей), которые представляют интерес для пользователей.

Объект (сущность) - предмет, процесс или явление, о котором собирается информация, необходимая для решения задачи. Объектом может быть человек, предмет, событие.

Каждый объект характеризуется рядом основных свойств - атрибутов. **Атрибутом** называется поименованная характеристика объекта. Атрибут показывает, какая информация должна быть собрана об объекте.

Например, объект - клиент банка.

Атрибуты - номер счета, адрес, сумма вклада.

### 2.2. Технология анализа предметной области

Первым этапом проектирования БД любого типа является анализ предметной области, который заканчивается построением информационной структуры (концептуальной схемы). На данном этапе анализируются запросы пользователей, выбираются информационные объекты и их характеристики, которые определяют содержание проектируемой БД. На основе проведенного анализа структурируется предметная область. Анализ предметной области не зависит от программной и технической сред, в которых будет реализовываться БД.

*Анализ предметной области целесообразно разбить на три фазы:*

- 1) анализ концептуальных требований и информационных потребностей;
- 2) выявление информационных объектов и связей между ними;
- 3) построение концептуальной модели предметной области и проектирование концептуальной схемы БД.

#### 2.2.1. Анализ концептуальных требований и информационных потребностей

Требования пользователей к разрабатываемой БД представляют собой список запросов с указанием их интенсивности и объемов данных. Эти сведения разработчики БД получают в диалоге с ее будущими пользователями. Здесь же выясняются требования к вводу, обновлению и корректировке информации. Требования пользователей уточняются и дополняются при анализе имеющихся и перспективных задач.

Рассмотрим примерный состав вопросника при анализе различных предметных областей.

*Пример 1.* Предлагается разработать БД для учета студентов вуза.

Анализ предметной области:

1. Сколько студентов учится в вузе?
2. Сколько факультетов и отделений в вузе?
3. Как распределены студенты по факультетам отделений и курсам?
4. Сколько дисциплин читается на каждом курсе по каждой специальности?

5. Как часто обновляется информация в БД?
6. Сколько преподавателей в вузе?
7. Сколько иногородних студентов живет в общежитии, на частных квартирах?
8. Сколько лекционных аудиторий и аудиторий для проведения практических занятий, лабораторий?
9. Какая преемственность существует между читаемыми курсами?
10. Как информация, представленная в п.п. 1–9, используется в настоящее время (расписание занятий, экзаменов, зачетов и т.д.) и как ее собираются использовать?
11. Сколько раз в день, сколько человек и кто пользуются БД?

*Пример 2.* Разработать требования к локальной БД “Аэропорт”.

Вопрос 1. Для каких типов задач (приложений) проектируется БД?

Ответ. Для трех типов задач:

Задача 1. Информация об обслуживающем персонале.

Задача 2. Информация о полетных средствах.

Задача 3. Информация о графике движения самолетов.

Вопрос 2. Какими информационными объектами характеризуются эти задачи?

Ответ. Задача 1 характеризуется тремя информационными объектами: летный состав, диспетчеры, технический персонал.

Задача 2 характеризуется двумя информационными объектами: самолет, взлетное поле.

Задача 3 характеризуется одним информационным объектом – рейсы.

Вопрос 3. Каким текущим запросам должны удовлетворять данные информационные объекты?

Ответ.

1. ФИО, звание, должность членов экипажа самолета.
2. Списочный состав диспетчеров.
3. Состав смены технического персонала.
4. Тип самолета, который может обслуживать тот или иной пилот.
5. Номер самолета, который обслуживает данный пилот, данная смена диспетчеров и технического персонала.
6. Номер личного дела сотрудника аэропорта.
7. Номер смены диспетчеров и технического персонала, обслуживающего аэропорт в заданном интервале времени.
8. Готовность самолета с таким-то номером к полету.
9. Количество часов налета такого-то самолета.
10. Готовность данной взлетной полосы в настоящее время.
11. Длина данной полосы.
12. Номер (номера) рейса до данного пункта назначения.
13. Какие промежуточные посадки совершает рейс №... ?
14. Время вылета и расчетное время прибытия рейса №... .
15. Время и место регистрации рейса №... .
16. Время посадки на рейс №... .
17. До какого времени задерживается рейс №... ?
18. Какие типы самолетов обслуживают рейс №... ?
19. Какой номер самолета обслуживает рейс №... ?

Вопрос 4. Каким перспективным запросам должны удовлетворять информационные объекты в БД “Аэропорт”?

1. С какого года используется самолет с №... в аэропорту, тип самолета?
2. Какое количество часов полета у члена экипажа, ФИО?
3. Расчетное время отпуска члена экипажа, диспетчера, технического работника.

*Пример 3. Разработать БД "Видеомагнитофоны".*

Вопрос 1. На кого рассчитана эта БД?

Ответ. На покупателя видеосистем.

Вопрос 2. Что интересует покупателя?

Ответ. Покупателя интересуют технические характеристики системы, ее цена, фирма-изготовитель, технические характеристики и цена видеокассет, фирма-изготовитель кассет.

Далее проектировщик выбирает по технической документации параметры видеосистем, разрабатывает перечень запросов и уточняет его с будущим пользователем БД. Однако БД пока нет.

Есть только предложения проектировщика и одобрение будущего пользователя. Пользователю кажется, что все проблемы позади, а проектировщика еще ждет очень большая работа.

Выявленные запросы представлены следующим перечнем:

1. Выдать данные на определенную модель системы.
2. Какова цена той или иной системы?
3. Выдать системы определенной страны-изготовителя, цены которых не превышают заданную.
4. Выдать последние модели определенной фирмы.
5. Выдать модели определенной фирмы, габаритные размеры которых не превышают заданные.
6. Выдать № моделей определенной страны-изготовителя, цены которых не превышают заданные.
7. Какова цена последних моделей определенной фирмы?
8. Выдать № модели и фирму-изготовителя самого дешевого видеоплеера, выпущенного в определенном году.
9. Выдать характеристики видеомагнитофонов, выпущенных в определенном году.
10. Выдать наименование модели и фирму-изготовителя видеокассет с наибольшим временем проигрывания.
11. Выдать данные на самую дешевую видеокассету.
12. Выдать данные на самую дорогую видеокассету.
13. Сколько стоит видеокассета определенного наименования и определенной фирмы?
14. Выдать данные на видеокассету, имеющую определенное время проигрывания.

### **2.2.2. Выявление информационных объектов и связей между ними**

Вторая фаза анализа предметной области состоит в выборе информационных объектов, задании необходимых свойств для каждого объекта, выявлении связей между объектами, определении ограничений, накладываемых на информационные объекты, типы связей между ними, характеристики информационных объектов. Проанализируем предметную область на примере БД "Видеомагнитофоны".

При выборе информационных объектов постараемся ответить на ряд вопросов:

На какие классы можно разбить данные, подлежащие хранению в БД?

Какое имя можно присвоить каждому классу данных?

Какие наиболее интересные характеристики (с точки зрения пользователя) каждого класса данных можно выделить?

Какие имена можно присвоить выбранным наборам характеристик?

*Пример.* Продолжим создание БД “Видеомагнитофоны”, рассчитанной на пользователей, которые хотят приобрести данный вид техники.

После беседы с различными пользователями и просмотра каталогов было выяснено, что интерес представляют три информационных объекта: видеомагнитофон, видеоплейер, видеокассета. Рассмотрим наиболее существенные характеристики каждого информационного объекта.

Объект - ВИДЕОМАГНИТОФОН.

Атрибуты - страна-изготовитель, фирма-изготовитель, № модели, телевизионные системы, число кассетных гнезд, ресурс непрерывной работы, система автопоиска, напряжение в сети, наличие таймера, число программ, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОПЛЕЙЕР.

Атрибуты - страна-изготовитель, фирма-изготовитель, N° модели, телевизионные системы, число воспроизводящих головок, ресурс непрерывной работы, напряжение в сети, наличие таймера, габаритные размеры, масса, цена в долларах, год выпуска.

Объект - ВИДЕОКАССЕТА.

Атрибуты - наименование, страна-изготовитель, фирма-изготовитель, тип кассеты, время проигрывания, цена в долларах.

Далее выделим связи между информационными объектами. В ходе этого процесса постараемся ответить на следующие вопросы:

Какие типы связей между информационными объектами?

Какое имя можно присвоить каждому типу связей?

Каковы возможные типы связей, которые могут быть использованы впоследствии?

Имеют ли смысл какие-нибудь комбинации типов связей?

Попытаемся задать ограничения на объекты и их характеристики.

Под **ограничением целостности** обычно понимают логические ограничения, накладываемые на данные. Ограничение целостности – это такое свойство, которое мы задаем для некоторого информационного объекта или его характеристики и которое должно сохраняться для каждого их состояния.

Введем следующие ограничения:

1. Значение атрибута “число кассетных гнезд” изменяется от 1 до 2.
2. Значение атрибута “ресурс непрерывной работы” изменяется от 4 до 24.
3. Значение атрибута “напряжение в сети” изменяется от 110 до 240 В.
4. Значение атрибута “число программ” изменяется от 1 до 20 и т.д.

**Типы связей.** Все информационные объекты предметной области связаны между собой.

Соответствия, отношения, возникающие между объектами предметной области называются **связями**. Различаются связи нескольких типов, для которых введены следующие обозначения:

- а) один к одному (1:1);
- б) один ко многим (1:M);

в) многие ко многим (M:M).

Рассмотрим эти типы связей на примере.

*Пример.* Дана совокупность информационных объектов, отражающих учебный процесс в вузе:

СТУДЕНТ (Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа)

СЕССИЯ (Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат)

СТИПЕНДИЯ (Результат, Процент)

ПРЕПОДАВАТЕЛЬ (Код преподавателя, Фамилия, Имя, Отчество)

**Связь один к одному (1:1)** предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует не более одного экземпляра информационного объекта В и наоборот.

Рис. 2.1 иллюстрирует указанный тип отношений.



Рис. 2.1. Графическое изображение реального отношения 1:1

Примером связи 1:1 может служить связь между информационными объектами СТУДЕНТ и СЕССИЯ:

СТУДЕНТ <—> СЕССИЯ

Каждый студент имеет определенный набор экзаменационных оценок в сессию.

При связи **один ко многим (1:M)** одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В, но каждый экземпляр объекта В связан не более чем с 1 экземпляром объекта А. Графически данное соответствие имеет вид, представленный на рис. 2.2.

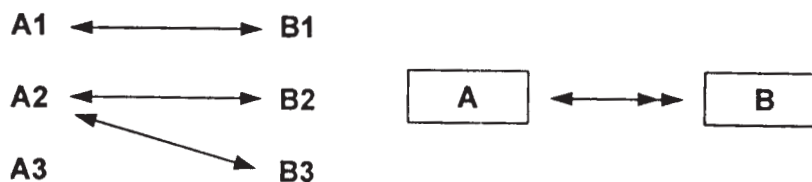


Рис. 2.2. Графическое изображение реального отношения 1:M

Примером связи 1:M служит связь между информационными объектами СТИПЕНДИЯ и СЕССИЯ:

СТИПЕНДИЯ <—>> СЕССИЯ

Установленный размер стипендии по результатам сдачи сессии может повторяться многократно для различных студентов.

Связь **многие ко многим** (М:М) предполагает, что в каждый момент времени одному экземпляру информационного объекта А соответствует 0, 1 или более экземпляров объекта В и наоборот. На рис. 2.3 графически представлено указанное соответствие.

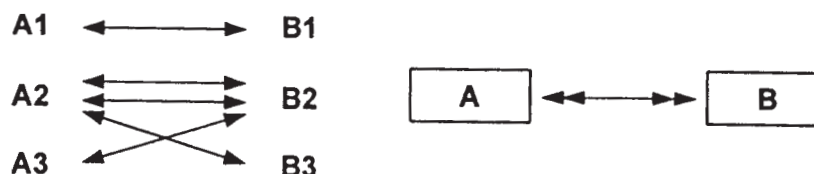


Рис. 2.3. Графическое изображение реального отношения М:М

Примером данного отношения служит связь между информационными объектами СТУДЕНТ и ПРЕПОДАВАТЕЛЬ:

СТУДЕНТ <<—> ПРЕПОДАВАТЕЛЬ

Один студент обучается у многих преподавателей, один преподаватель обучает многих студентов.

### 2.2.3. Построение концептуальной модели предметной области

Заключительная фаза анализа предметной области состоит в проектировании ее информационной структуры или концептуальной модели.

**Концептуальная модель** включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области (ПО) и выявляемых в результате анализа данных.

Концептуальная модель применяется для структурирования предметной области с учетом информационных интересов пользователей системы. Она дает возможность систематизировать информационное содержание предметной области, позволяет как бы “подняться вверх” над ПО и увидеть ее отдельные элементы. При этом, уровень детализации зависит от выбранной модели.

Концептуальная модель является представлением точки зрения пользователя на предметную область и не зависит ни от программного обеспечения СУБД, ни от технических решений.

Концептуальная модель должна быть стабильной. Могут меняться прикладные программы, обрабатывающие данные, может меняться организация их физического хранения, концептуальная модель остается неизменной или увеличивается с целью включения дополнительных данных.

Одной из распространенных моделей концептуальной схемы является модель «сущность - связь». Основными конструкциями данной модели являются сущности и связи.

Под **сущностью** понимают основное содержание объекта ПО, о котором собирают информацию. В качестве сущности могут выступать место, вещь, личность, явление.

Экземпляр **сущности** - конкретный объект.

Например:

сущность (объект) – служащий

экземпляр сущности – Иванов А.В.;

сущность (объект) – институт  
экземпляр сущности – СГУ.

Сущность принято определять атрибутами - поименованными характеристиками. Например:

сущность – служащий

атрибуты: ФИО, год рождения, адрес, образование и т.д.

Чтобы задать атрибут в модели, ему надо присвоить имя и определить область допустимых значений. Одно из назначений атрибута - идентифицировать сущность.

Связь определяет отношения между сущностями. Типы связей: один к одному, один ко многим, многие ко многим.

При построении модели «сущность - связь» используют графические диаграммы. При этом обозначают:

сущности - прямоугольниками,

атрибуты - овалами,

связи - ромбами.

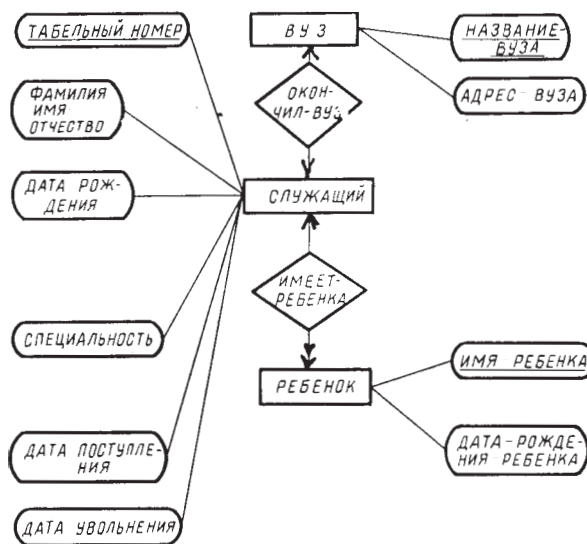


Рис. 2.4. Пример модели «сущность – связь»

На практике приходится строить несколько вариантов моделей, из которых выбирается одна, наиболее полно отображающая предметную область.

*Пример.* Спроектировать БД «Сессия». База данных должна выдавать оперативную информацию об успеваемости студентов на факультетах в семестре. Результатами сессии считать только экзамены.

По сути дела в БД исходя из формулировки задания можно выделить лишь одно приложение. Речь идет об успеваемости студентов разных факультетов по тем или иным дисциплинам. Более конкретно речь идет о выдаче справок по результатам сессии каждого студента, учебной группы, курса, факультета, а также об автоматизированном составлении ведомости.

Выберем следующие сущности:

ИНСТИТУТ, ФАКУЛЬТЕТ, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ, ДИСЦИПЛИНА.

В данном примере можно выделить сущность ЭКЗАМЕН или ВЕДОМОСТЬ, но можно не выделять, а сформировать ведомость из имеющихся данных по средствам связей.

Зададим каждую сущность набором атрибутов:

ИНСТИТУТ (название, подчиненность, адрес, телефон, ФИО ректора).

ФАКУЛЬТЕТ (название, код специальности, данные о кафедрах, число выпускников, декан).

СТУДЕНТ (ФИО, группа, курс, номер текущего семестра, пол).

ПРЕПОДАВАТЕЛЬ (ФИО, должность, звание, кафедра, стаж).

ДИСЦИПЛИНА (название, число часов, код дисциплины, виды занятий, число читаемых семестров, номера текущих семестров, на каких курсах преподается)

В каждом наборе атрибутов, характеризующих сущность, необходимо выбрать ключевые атрибуты, т.е. атрибуты, делающие сущность уникальной. При задании атрибутов ключевые атрибуты подчеркивались.

Определим связи между сущностями.

#### Название связи

#### Связи между сущностями

учится  
изучает  
имеет  
работает  
преподает  
экзамен

студент, факультет  
студент, дисциплина  
институт, факультет  
преподаватель, факультет  
преподаватель, дисциплина  
студент, дисциплина, преподаватель

После выбора сущностей, задания атрибутов и анализа связей можно перейти к проектированию информационной (концептуальной) схемы БД.

Концептуальная схема БД «Успеваемость» представлена на рис. 2.5 (атрибуты сущностей на диаграмме не показаны).

Рассмотрим некоторые ограничения в рассматриваемом примере:

1. Значение атрибута «телефон» (сущность – ИНСТИТУТ) задается целым положительным шестизначным числом.
2. Значение атрибута «код факультета» (сущность – ФАКУЛЬТЕТ) лежит в интервале 1–10.
3. Значение атрибута «курс» (сущность – СТУДЕНТ) лежит в интервале 1–6.
4. Значение атрибута «семестр» (сущность – СТУДЕНТ, ДИСЦИПЛИНА) лежит в интервале 1–12.
5. Значение атрибута «число часов» (сущность – ДИСЦИПЛИНА) лежит в интервале 1–300.
6. Одному студенту может быть приписана только одна группа.
7. Один студент может учиться только на одном факультете.
8. Один студент в семестре сдает от 3 до 5 дисциплин.
9. Один студент изучает в семестре от 6 до 12 дисциплин.
10. Одному преподавателю приписывается только одна кафедра.
11. Один студент может пересдавать одну дисциплину не более трех раз.
12. Ключи: название института, название факультета, ФИО и группа студента, ФИО и кафедра преподавателя, название дисциплины.

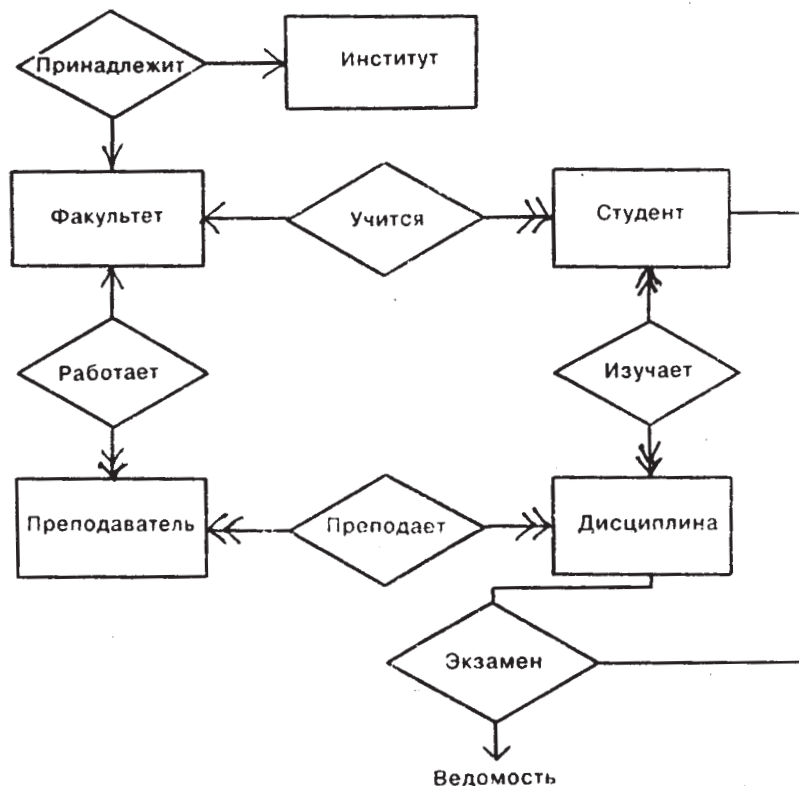


рис 2.5. Концептуальная схема БД «Успеваемость»

### 2.3. Логическое проектирование

Логическое проектирование представляет собой необходимый этап при создании БД. **Основной задачей логического проектирования** является разработка логической схемы, ориентированной на выбранную систему управления базами данных (СУБД). Этап логического проектирования в отличие от концептуального проектирования полностью ориентирован на инструментальные средства компьютера.

Процесс логического проектирования состоит из следующих этапов:

1. Выбор конкретной СУБД.
2. Отображение концептуальной схемы на логическую схему.
3. Выбор ключей.
4. Описание языка запросов.

Одним из основных критериев выбора СУБД является оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему. Существующие СУБД делятся по типам моделей данных на реляционные, иерархические и сетевые. СУБД, ориентированные на персональные компьютеры, как правило, поддерживают реляционную модель данных. Подавляющее большинство современных СУБД

– реляционные. Если выбрана реляционная система, то концептуальную схему БД предстоит отображать на реляционную.

При отображении концептуальной схемы на реляционную модель данных каждый прямоугольник схемы отображается в таблицу. При этом следует учитывать ограничения на размеры таблиц, которые накладывает выбранная СУБД.

Отобразим концептуальную схему, изображенную на рис. 2.5, на реляционную модель. Каждый прямоугольник (сущность) этой схемы будет представлен в виде таблицы. Каждый столбец таблицы предназначен для записи одного атрибута и имеет свое уникальное имя.

Представим сущность ПРЕПОДАВАТЕЛЬ (ФИО, должность, звание, кафедра, стаж) в виде таблицы.

#### ПРЕПОДАВАТЕЛЬ

<u>ФИО</u>	<u>Должность</u>	<u>Звание</u>	<u>Кафедра</u>	<u>Стаж</u>

Определим структуру каждой таблицы, то есть зададим типы и размеры полей.

Признак ключа	Поле	Тип поля	Размер поля
ключ	ФИО	символьный	21
	Должность	символьный	15
	Звание	символьный	10
ключ	Кафедра	символьный	20
	Стаж	числовой	2

Можно произвести оценку требуемого объема памяти на хранение данной таблицы.

На хранение одной записи необходимо  $21+15+10+20+2=68$  байт. Всего предусмотрено 100 записей, то есть на хранение таблицы потребуется 6800 байт.

Аналогично поступают со всеми остальными сущностями (объектами) концептуальной схемы.

**Выводы.** Концептуальная модель представляет объекты предметной области и их взаимосвязи без указания способов их физического хранения.

Таким образом, концептуальная модель является, по существу, моделью предметной области. При проектировании концептуальной модели все усилия разработчика должны быть направлены в основном на структуризацию данных и выявление взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки. Проектирование концептуальной модели основано на анализе решаемых на этом предприятии задач по обработке данных.

Концептуальная модель включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных.

Концептуальная модель транспонируется затем в модель данных, совместимую с выбранной СУБД. Версия концептуальной модели, которая может быть обеспечена конкретной СУБД, называется логической моделью.

**Логическая модель** отражает логические связи между элементами данных вне зависимости от их содержания и среды хранения.

Логическая модель данных может быть реляционной, иерархической или сетевой. Логическая модель отображается в физическую память, такую, как диск, лента или какой-либо другой носитель информации.

**Физическая модель**, определяющая размещение данных, методы доступа и технику индексирования, называется внутренней моделью системы.

В современных СУБД выполнение задач физического проектирования автоматизировано.

### 3. СУБД MICROSOFT ACCESS

Microsoft Access представляет собой реляционную базу данных.

Информация в базе данных Access представляется в виде отдельных таблиц. При этом каждый столбец таблицы соответствует полю данных, а каждая строка – записи данных.

Запись данных состоит из нескольких полей. При этом действует следующее правило: запись данных представляет собой группу взаимосвязанных полей, рассматриваемых как единое целое. Например, запись данных может содержать информацию о конкретном клиенте или продукте. Каждая запись данных в таблице содержит одинаковое число полей. Каждое поле содержит один и тот же тип информации. Например, в каждой записи данных некоторой таблицы поле ИМЯ КЛИЕНТА будет предназначено для имени клиента, т.е. будет содержать однотипную информацию.

Существуют различные типы данных. Тип данных для конкретного поля данных выбирается в зависимости от того, какая информация будет располагаться в этом поле. Размер поля данных определяется в зависимости от выбранного для него типа.

Вот некоторые примеры типов данных.

**Текстовые поля (TEXT):** могут содержать отдельные слова (например, имена), сочетания слов и чисел (например, почтовый адрес), или числа, которые не используются для математических расчетов (например, номера телефонов), а также специальные знаки (например «/» или «-»).

**Числовые поля (NUMBER):** содержат числа, используемые для различных расчетов (например, цены продуктов).

**Поля даты и времени (DATE/TIME):** содержат дату и/или время.

**Поля денежных сумм (CURRENCY):** содержат числовые величины в формате денежных сумм с обозначением денежных единиц – р. (рубль), \$ (доллар), DM (немецкая марка) и т.п.

**Логические поля (YES/NO):** содержат логические данные со значениями ДА/НЕТ (YES/NO) или ИСТИНА/ЛОЖЬ (TRUE/FALSE).

**Поля текстовых примечаний (MEMO):** содержат большие участки текста (до 32 000 знаков).

#### 3.1. Связи между таблицами

В базе данных Access между отдельными таблицами могут быть сформированы связи, соединяющие их в единую базу. Связи осуществляются через общие поля данных.

В качестве примера представим себе базу данных о клиентах, которая состоит из таблицы клиентов и таблицы заказов. Пусть в таблице клиентов

поля содержат разнообразную информацию о клиентах: фамилию, адрес, телефон и т.д., в том числе, и поле «Код клиента». В таблице заказов одним из прочих полей является «Код заказчика», имеющий такой же тип данных и размер, что и «Код клиента» в первой таблице. Если необходимо внести в таблицу заказов несколько конкретных заказов, поступивших от клиента Петрова, следует внести в записи об этих заказах только соответствующий «Код заказчика» и установить связь таблиц по этим полям. Access свяжет адресные данные клиента с заказом, и пользователь легко сможет выяснить, по какому адресу должен быть доставлен заказ, хотя описание заказа и адреса заказчиков хранятся в разных таблицах.

### **3.2. Ключи**

Часто необходимо осуществлять поиск данных, хранящихся в различных таблицах.

Предпосылкой и условием успешного поиска и установления связей служит правильное определение так называемых ключей. При помощи ключа Access может различать записи таблицы и упорядочивать их соответствующим образом. Каждая таблица должна содержать, как минимум, один ключ. Ключ представляет собой поле в соответствующей таблице. Он однозначно характеризует каждую запись, содержащуюся внутри таблицы.

Идеальным первичным ключом является такая информация, как номер клиента, номер счета и т.д. При помощи такого ключа легко определить соответствующую запись. Для первичного ключа не годятся поля, содержащие имя клиента. Всегда возможна ситуация, когда несколько человек имеют одно и то же имя.

### **3.3. Макросы, модули и объекты**

При помощи макросов можно автоматизировать отдельные действия, не погружаясь в программирование. Запись макросов происходит по принципу магнитофона: отдельные действия, необходимые для определенной операции, записываются в нужной последовательности, а затем воспроизводятся любое количество раз простым нажатием кнопки. Например, для перехода к форме при открытии базы данных можно использовать специальный макрос. Тогда при открытии соответствующей базы данных на экране будет появляться форма.

Макросы могут не вполне удовлетворять потребностям пользователя. В этом случае можно воспользоваться языком программирования, имеющимся в Access. Он называется Access-Basic. При помощи этого языка автоматизируются сложные действия, например, сложные вычисления в отчете. Можно запрограммировать функцию вычислений и использовать ее, когда это необходимо. Процедуры, написанные на Access-Basic, запоминаются в программном модуле.

Все элементы, которые формируются и используются в Access называются объектами Access. К ним относятся таблицы, формы, запросы, отчеты, макросы, модули, графики, диаграммы, диалоговые окна, а также некоторые системные объекты, например, системный буфер обмена (Clipboard).

### **3.4. Полезные помощники**

В Access имеется целый ряд полезных помощников. Все они предназначены для облегчения работы с базой данных.

Эти помощники разделены на три основные группы.

#### 3.4.1. Карточки подсказки

Они могут дать совет при выполнении самых различных действий, например, при создании базы данных. Карточки подсказки содержат подробные пошаговые инструкции, помогающие выполнить все необходимые действия. Карточки подсказки могут помочь графическими примерами и прямыми указаниями.

#### 3.4.2. Мастера

В то время как Карточки подсказки лишь подают нужные советы во время работы, Мастера сами выполняют нужные действия. При вызове Мастера открывается соответствующая последовательность диалоговых окон. Получив ответы на заданные вопросы, Мастер выполняет все необходимые действия. Предположим, вы хотите сформировать отчет. Мастер отчетов спросит вас, как должен выглядеть отчет и какие данные он должен содержать. После получения ответов отчет будет полностью готов. С помощью других Мастеров можно формировать таблицы баз данных, запросы и формы.

Работа с Мастерами проста, надежна и эффективна, и позволяет использовать все богатство возможностей Microsoft Access.

#### 3.4.3. Конструкторы

**Конструктор (Builder)** – это инструмент Access, который облегчает выполнение конкретного задания. Крайне полезным является Конструктор выражений – он позволяет быстро сформировать сложное выражение. С помощью Конструктора запросов легко формируются запросы, которые используются для получения выборок данных для формы или отчета. Помимо перечисленных, в Access имеются и другие конструкторы. Это – макро-конструктор, с помощью которого формируются различные макросы, а также конструкторы меню, полей, цветов, кодов и другие.

### 3.5. Запросы и выборки

Запросы используются для выбора из базы данных интересующей пользователя информации. Например, пользователя может заинтересовать, сколько клиентов его фирмы живет в Москве или каковы суммы выплаченной зарплаты сотрудникам его фирмы по подразделениям. Результат такого запроса называется выборкой. Под **выборкой** мы будем понимать динамическую таблицу с записями данных, которые удовлетворяют определенным условиям запроса.

**Динамическая таблица запроса (Dynaset)**, в дальнейшем, выборка, – это таблица, формируемая всякий раз заново на основе реальных таблиц базы данных, содержимое которой удовлетворяет условиям запроса. С выборкой можно обращаться как с реальной таблицей – например, редактировать ее записи. Внесенные изменения будут отражены в записях реальных таблиц, стоящих за этой выборкой.

#### 3.5.1. Основные типы запросов

**Стандартные запросы.** Чаще всего используются **стандартные запросы выбора**, используемые для того, чтобы отобрать и представить в виде удобной динамической таблицы интересующие пользователя данные из таблиц базы данных.

**Запросы на выполнение действия.** Наряду со стандартными запросами выбора, можно выделить также **запросы на выполнение действия**, которые используются для создания новых реальных таблиц данных, в дальнейшем существующих уже независимо от тех таблиц базы, которые были использованы для их построения. Эти запросы позволяют также изменять таблицы базы данных: обновлять их, дополнять новыми записями или удалять некоторые записи.

С помощью запросов на выполнение действия можно перемещать или изменять табличные данные, обновлять, добавлять или удалять группы записей. Можно сформировать на основе выборки новую реальную таблицу данных, которая продолжит свое независимое существование и после закрытия запроса. Различают четыре вида запросов на выполнение действия.

*Запрос на добавление.* Можно добавлять отобранные записи из таблицы или запроса текущей базы данных в конец другой таблицы. Дополняемая таблица может находиться как в той же самой, так и в другой базе данных.

*Запрос на удаление.* С помощью запроса на удаление можно удалить группу записей данных, удовлетворяющих заданным условиям.

*Запрос на обновление.* Можно изменить группу записей данных, удовлетворяющих определенному условию.

*Запрос на создание новой таблицы.* Из выборки, являющейся динамической таблицей, сформированной при выполнении запроса и существующей только до окончания работы с результатами запроса, можно создать новую таблицу базы данных. Динамическая таблица является представлением данных других таблиц базы данных, и изменения, внесенные в выборку, будут отражены в записях таблиц, на основании которых эта выборка построена. Если же создать новую таблицу, она начнет свое независимое существование. Изменения в этой новой таблице не будут затрагивать использованные для ее построения исходные таблицы базы данных.

**Параметрические запросы.** Часто пользователю приходится иметь дело с запросами, устроенными в принципе одинаково, но имеющими некоторые различия в поставленных условиях отбора. В таком случае, чтобы каждый раз заново не создавать отдельный запрос, следует сформировать запрос с параметрами.

При выполнении **запроса с параметрами** Access запрашивает у пользователя значение параметров для определения условий выборки. Речь может идти, например, о выборе телефонов клиентов из определенного региона, причем интересующий пользователя регион изменяется от одного запроса к другому. В данном случае система запрашивает название региона.

**Перекрестные запросы.** **Перекрестные запросы** позволяют представить данные в весьма наглядной, компактной форме сводных кросс-таблиц, осуществляя при этом разнообразную группировку записей и групповые вычисления при обработке данных.

В качестве заголовков строк и столбцов используются значения заданных полей или выражения.

К примеру, нужно представить информацию о ежемесячных объемах продажи по всем наименованиям товаров. Если просто сгруппировать записи по месяцам и наименованиям товаров, то каждое наименование товара появится в выборке двенадцать раз – по разу для каждого месяца. Если же представить данные в виде сводной кросс-таблицы, в строках которой данные группируются по наименованиям товаров, а в столбцах – по месяцам, то результирующая выборка будет намного компактнее и нагляднее.

### 3.5.2. Два способа формирования запросов

По способу формирования запросы можно разделить на два вида:

**Запросы по образцу или QBE-запросы**, для определения которых пользователь должен указать параметры запроса в окне конструирования, задавая образцы для поиска информации.

**Структурированные, или SQL-запросы**, для определения которых пользователь должен описать запрос с помощью особого языка запросов, используя специальные команды и функции.

Практически любой запрос может быть сформулирован как в виде запроса по образцу, так и в виде SQL-запроса. Однако многие пользователи предпочитают работать с запросами по образцу, так как этот способ более наглядный, его легче освоить, не требуется учить команды и функции структурированного языка запросов.

При определении запроса по образцу Access автоматически формирует соответствующий SQL-запрос.

Основной тип запросов по образцу – это запросы на выбор. Пользователь может формировать запрос самостоятельно или обратиться к Мастерам запросов.

## 3.6. Формы данных

В Access существует несколько способов ввода, редактирования и просмотра хранящейся в базе данных информации. Простейшим из них является следующий: открыть таблицу базы данных или запрос в табличном режиме – и можно вводить новые записи, удалять или редактировать старые, просматривать данные, относящиеся к этой таблице или запросу.

Однако есть и другой способ – подготовить формы данных, которые специально предназначены именно для наглядного представления информации из базы данных и могут существенно облегчить как ввод данных, так и восприятие хранящейся в базе данных информации.

### 3.6.1. Создание формы

Пользователь может создать простейшую форму на основе уже существующей таблицы Access. Может выбрать, какие поля и в какой последовательности должны быть представлены в форме, разбить их на логически связанные группы и задать их расположение на экране.

Access позволяет создать несколько форм для различного вида информации, например, формы счетов, заказов, адресов покупателей (клиентов) и другие. Данные одной таблицы могут быть представлены в нескольких различных формах, в то же время в одной форме может быть представлена информация из различных таблиц базы данных. Формы могут содержать иллюстрации, а также графическое представление информации из базы данных в виде диаграмм.

Как и при разработке таблиц и запросов, при разработке форм можно воспользоваться помощью соответствующего Мастера. При создании формы с помощью Мастера пользователю предстоит «дать интервью», ответив на его вопросы в серии диалоговых окон. Результатом такого диалога с Мастером форм будет служить готовая к употреблению форма, которую при желании можно будет подвергнуть дальнейшему усовершенствованию.

Существуют пять видов форм, которые можно разработать с помощью Мастера форм:

В один столбец (SINGLE COLUMN)  
Табличная форма (TABULAR)  
Диаграмма (GRAPH)  
Составная форма (MAIN/SUBFORM)  
Простая форма (AUTOFORM).

### 3.6.2. Форма в один столбец

Каждая запись данных представлена на отдельной странице формы. Для просмотра записей данных их необходимо пролистывать. Каждое поле данных находится в собственной строке. Поэтому имя и фамилия или индекс и место жительства представлены в отдельных строках, так как в данном случае речь идет об отдельных полях.

Подписи полей в форме соответствуют именам полей таблицы или их подписям, если они были заданы при определении свойств полей таблицы (в нижней части окна конструирования). Это, пожалуй, самый распространенный вид формы (рис. 3.1).

Модели автомобилей (1)

Код модели: 12416

Модель: Classic Combi

Мощность двигателя: 60 / 90

Коробка передач: Автоматическая

Прочее оснащение: Навесной багажник, раздвижная крыша, электроподъемник окон

Заводская цена: 46 200,00 р.

Record: 1 of 5

Рис. 3.1. Форма в один столбец

### 3.6.3. Табличная форма

Этот вид формы напоминает обыкновенное табличное представление запроса или таблицы базы данных. В табличной форме на одном листе одновременно представлено несколько записей данных. При этом каждая запись данных занимает отдельную строку. Каждое поле данных находится в отдельном столбце, а название поля располагается в шапке столбца. Внешний вид такой формы может быть очень похож на стандартное представление таблицы базы данных (рис. 3.2).

Клиенты и заказы					
Клиенты и заказы					
Код модели	Номер заказа	Обращение	Имя	Отчество	
12572	123	Господин	Геннадий	Иванович	
12580	130	Господин	Сергей	Петрович	
12653	135	Господин	Геннадий	Иванович	
12651	136	Господин	Остап	Ибрагимович	
12410	138	Господин	Павел	Петрович	
12410	140	Господин	Михаил	Илларионович	
12410	141	Господин	Александр	Васильевич	
12410	146	Господин	Петр	Алексеевич	

Рис. 3.2. Табличная форма

### 3.6.4. Форма-диаграмма

**Форма-диаграмма** - это форма со встроенной диаграммой, отображающей информацию из базы данных в графическом виде.

*Пример.* В базе данных AUTO, в таблице «Модели автомобилей», хранятся технические и экономические характеристики автомобилей – например, заводская цена. Эта информация может быть отображена в форме в виде диаграммы (рис.3.3).

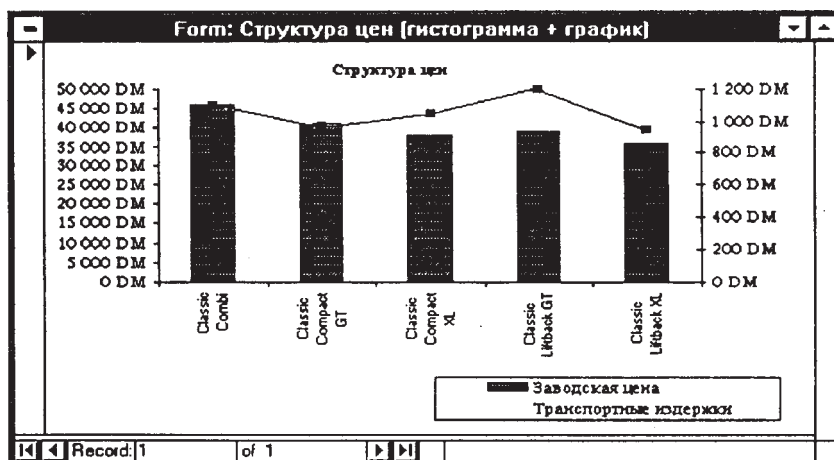


Рис. 3.3. Форма-диаграмма

### 3.6.5. Составная форма

**Составная форма** – это форма, содержащая информацию из двух различных таблиц базы данных, между которыми установлена связь. Одна из этих таблиц является главной, другая – подчиненной таблицей, с отношением связи между ними «один ко многим». При этом подчиненная таблица встроена в форму главной таблицы.

*Пример.* В верхней части формы представлены данные о моделях автомобилей (в виде формы в один столбец). В нижней части формы находится табличное представление всех заказов на данную модель. Главная таблица моделей связана с таблицей заказов отношением «один ко многим» (рис. 3.4).

Модели и заказы

Модель: Classic Lillback GT

Мощность двигателя: 69 / 90

Коробка передач: Автоматическ

Прочие оснащение: Автоматизированная крыша, покрытие "металлик"

Заводская цена: 39 200,00 р.

Фамилия:	Имя:	Отчество:	Город:	Место жительства:	Ски:
Петров	Сергей	Петрович	С.Петербург	Невский пр., д. 16	

Record: 1 of 1

Record: 3 of 5

Рис. 3.4. Составная форма

### 3.6.6. Простая форма

Эта форма – частный случай формы в один столбец. Возможно очень быстро, без лишних переговоров с Мастером, создать форму, которая представляет в виде формы в один столбец все без исключения поля записи данных (рис.3.5).

При этом каждое поле данных представлено в виде отдельной строки. Слева расположено название поля. Получившуюся форму можно впоследствии отредактировать, в том числе и удалить лишние поля.

The screenshot shows a database form window titled "Клиенты и заказы". The form contains the following fields and values:

Field Name	Value
Код модели:	12411
Номер заказа:	141
Обращение:	Господин
Имя:	Александр
Отчество:	Васильевич
Фамилия:	Сухов
Индекс:	190000
Город:	С.Петербург
Место жительства:	Садовая ул.

At the bottom of the form, there is a status bar showing "Record: 7 of 8" and navigation icons.

Рис. 3.5. Простая форма

### 3.7. Обработка данных с помощью отчетов

В отчете пользователь может наглядно представить извлеченную из базы данных информацию, дополнив ее результатами анализа и вычислений. Пользователь свободен в выборе макета отчета и его оформления. В MS Access предусмотрены некоторые стандартные компоненты структуры отчета (шапки, примечания, колонтитулы, основная часть), однако каким именно будет макет отчета – решает пользователь.

Сконструировать макет отчета можно двумя способами: с помощью Мастера отчетов или самостоятельно. Начинаящим пользователям Мастер отчетов помогает избежать ошибок. Во втором случае пользователь обладает более широкими возможностями. Однако возможен и промежуточный вариант: можно использовать помощь Мастера для создания «заготовки» отчета, а затем самостоятельно доработать этот отчет, внося необходимые дополнения и изменения. Это наиболее рациональный способ.

Рассмотрим подробнее конструирование макета отчета с помощью Мастера.

Мастер отчетов помогает спроектировать отчет, ставя вопросы, касающиеся структуры, содержания и оформления отчета. Проведя с помощью серии диалоговых окон своеобразное «интервью», предлагая при этом на выбор возможные варианты ответов и демонстрируя возможные результаты выбора того или иного варианта ответа, Мастер формирует макет отчета.

Существует семь различных видов отчетов, которые можно создать с помощью мастера:

В один столбец (SINGLE COLUMN)  
Группировка данных и вычисление итогов (GROUPS/TOTALS)  
Итоговый отчет (SUMMARY)  
Табличный отчет (TABULAR)  
Почтовые наклейки (MAILING LABEL)  
Слияние с MS Word (MS WORD MAIL MERGE)  
Простой отчет (AUTOREPORT).

#### **3.7.1. Отчет в один столбец**

В отчете этого вида каждое отображаемое поле записи расположено в отдельной строке. Слева в каждой строке находится подпись к полю. Записи данных расположены на странице строго друг под другом. Количество записей данных, помещающихся на одну страницу, определяется размером страницы.

У пользователя есть выбор между тремя различными стилями оформления отчета:

1. РАБОЧИЙ (EXECUTIVE) (предлагается использовать по умолчанию). В этом отчете используются двойные разделительные линии между записями. Подписи к полям выделяются полужирным шрифтом и отделены от полей несколькими пробелами.

2. ПРЕЗЕНТАЦИОННЫЙ (PRESENTATION). В нем записи отделены друг от друга ординарными линиями. Подписи к полям выделяются полужирным шрифтом и подчеркиванием. Расположение полей определяется графически, что обеспечивает строгое соблюдение выравнивания полей.

3. РАЗГРАФЛЕННЫЙ (LEDGER). В нем подписи к полям и сами поля располагаются в расчерченной таблице. Это облегчает сопоставление значений в отчете.

На рис 3.6 представлен отчет в один столбец (рабочий вариант).

#### **3.7.2. Группировка данных и вычисление итогов**

В этом отчете поля записи данных располагаются в строчку, в табличной форме. Подписи полей располагаются у верхнего края страницы. Access производит группировку записей и вычисляет промежуточные итоги по группам и общий итог для всех групп (рис. 3.7).

#### **3.7.3. Итоговый отчет**

В таком отчете производятся группировка данных и вычисление промежуточных и общих итогов. Выводимые поля располагаются в строчку – в табличной форме. Отличие от предыдущего отчета состоит в том, что итоговый отчет не содержит детальных записей. В этом отчете выводятся только строки итогов, и результат выглядит более компактным (рис. 3.8).

#### **3.7.4. Табличный отчет**

В этом отчете поля записей данных располагаются, в строчку, в табличной форме. Подписи полей располагаются у верхнего края страницы. Каждая строка содержит новую запись данных. Вид этого отчета соответствует виду данных таблиц базы данных и запросов, представленных в табличной форме (рис. 3.9).

Microsoft Access - [Report: Report1]

File Edit View Format Window Help

Заказы

12-Dec-94

Торговый агент: Грачев

Наименование: Вержишель

Количество: 250

Цена: 1 150,00 р.

Дата: 3.01.94

Торговый агент: Грачев

Наименование: Арахис жареный

Количество: 200

Page 1

Print Preview

Рис. 3.6. Отчет в один столбец

Microsoft Access - [Report: Стоимость заказов (группировка)]

File Edit View Format Window Help

заказов (группировка)

Дата	Категория	Наименование	Количество	Цена
			250	
<b>Jan 1994</b>				
3.01.94	3	Тушенка 525 г.		3 100,00 р.
5.01.94	3	Ветчина кус.		2 640,00 р.
25.01.94	3	Тушенка 525 г.		3 100,00 р.
<b>Jan 1994</b>				<b>8 840,00 р.</b>
				<b>4,89%</b>
<b>Feb 1994</b>				
9.02.94	1	Мед		5 000,00 р.

Page 1

Ready

Рис. 3.7. Отчет с группировкой данных и вычислением итогов

Microsoft Access - [Report: Стоимость заказов (итого)]

File Edit View Format Window Help

Стоимость заказов (Вычисления)

12-Dec-94

Торговый агент	Дата	Количество	Цена
Грачев	Янв 1994	650	8 240,00 р.
		6,05%	4,89%
	Фев 1994	2650	28 260,00 р.
		24,65%	15,63%
	Мая 1994	950	26 710,00 р.
		8,84%	14,78%
Грачев		4250	63 810,00 р.
		39,53%	35,30%
Журавлев	Янв 1994	300	2 640,00 р.

Page: 1

Ready

Рис. 3.8. Итоговый отчет

Microsoft Access - [Report: Report2]

File Edit View Format Window Help

Клиенты и заказы

12-Dec-94

Код заказа	Фамилия	Город	Место жительства	Дата заказа
12410	Романов	С.Петербург	Исакиевская пл., д. 5	23.05
12410	Куров	С.Петербург	ул. Плеханова, д. 11	20.04
12410	Сухов	С.Петербург	Садовая ул., д. 9	21.05
12579	Иванов	Гатчина	пр. Павла Первого, д. 5	15.01
12580	Романов	Гатчина	Красная ул., д. 13	15.04
12580	Петров	С.Петербург	Невский пр., д. 100	19.02
12651	Бендер	С.Петербург	пр. Ветеранов, д. 7	21.03

Page: 1

Print Preview

Рис. 3.9. Табличный отчет

### 3.7.5. Почтовые наклейки

**Почтовые наклейки** - вид отчетов, используемый для создания стандартных почтовых наклеек на конверты при рассылке серийных писем.

Для формирования такого отчета в качестве источника данных обычно используется не таблица, а запрос к таблице с адресными данными. Условия запроса служат для выбора из общего списка тех, кому должна быть направлена корреспонденция. По результатам запроса можно отпечатать необходимые почтовые наклейки (рис. 3.10).

### 3.7.6. Слияние с MS Word

**Слияние с MS Word** - вид отчетов, используемый для подготовки серийных документов, что позволяет объединить данные таблицы или запроса MS Access с документом MS Word. Такое объединение документов и таблиц дает возможность быстро сформировать и отпечатать комплект схожих писем или других документов, подготовленных в MS Word и отличающихся друг от друга лишь некоторыми элементами, например, адресом или обращением к клиенту.

Эти элементы заполняются данными, взятыми из таблицы или запроса MS Access. Это означает следующее. Вместо конкретных данных об адресе клиента в документ MS Word вставляются специальные поля, соответствующие полям таблицы или запроса Access. При выводе документа на печать содержимое этих полей заполняется на основе конкретной записи данных из Access (рис. 3.11).

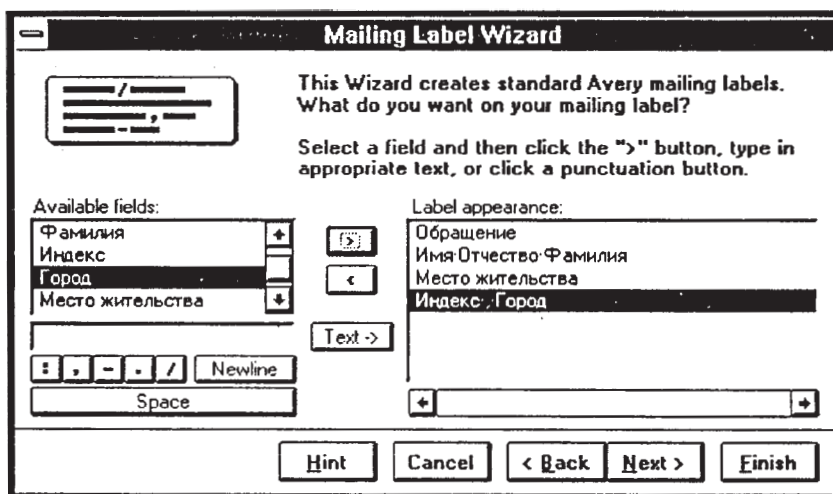


Рис. 3.10. Почтовые наклейки в режиме предварительного просмотра

### 3.7.7. Простой отчет

Наконец, последний вид отчетов, создаваемых с помощью Мастера, – это простой отчет. Если требуется быстро сформировать отчет в один столбец, в котором будут представлены все поля записи данных конкретной таблицы или запроса, следует выбрать этот вид отчета.

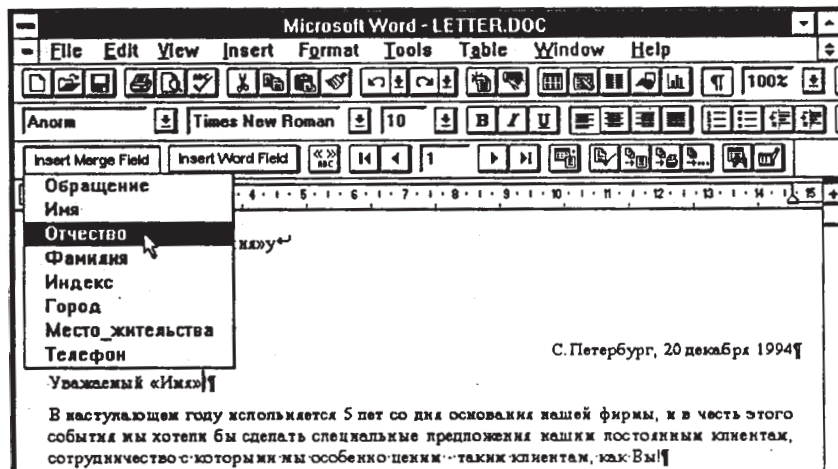


Рис. 3.11. Раскрывающийся список доступных для вставки полей

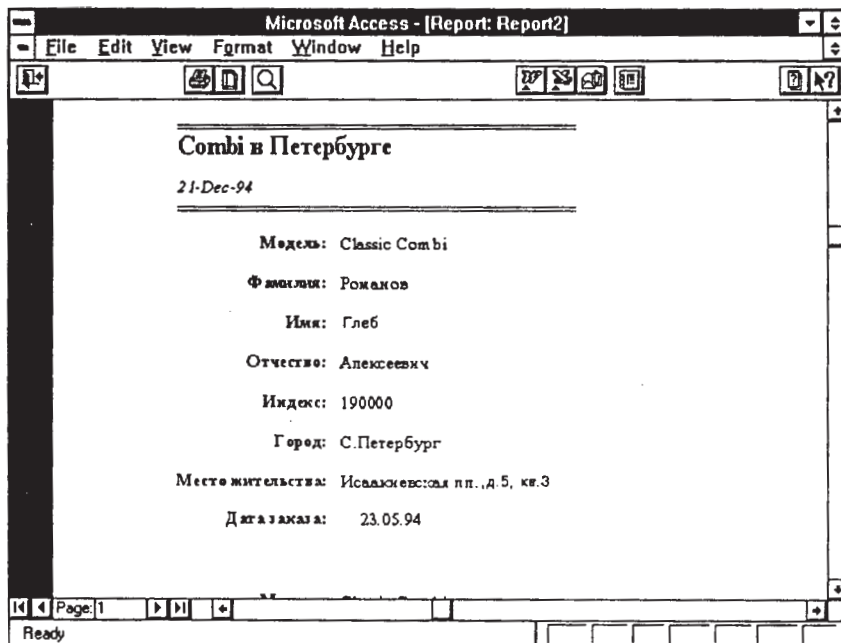


Рис. 3.12. Простой отчет

Создать простой отчет можно с помощью единственного щелчка мышью на кнопке инструмента ПРОСТОЙ ОТЧЕТ. Эта кнопка располагается на панели инструментов базы данных или открытой таблицы и запроса. При этом таблица или запрос не обязательно должны быть открыты. Достаточно выделить нужную таблицу или запрос на вкладке таблиц или запросов окна базы данных и нажать соответствующую кнопку.

Access автоматически сформирует простой отчет на основе открытой или выделенной таблицы или запроса, не задавая вопросов (рис. 3.12). Впоследствии можно, если нужно, отредактировать отчет. Для этого надо будет открыть отчет в режиме конструирования и внести изменения в его макет.

#### 4. ОСОБЕННОСТИ РАБОТЫ С ТЕКСТОВЫМИ БАЗАМИ ДАННЫХ

**Текстовыми базами данных** называются базы данных, объектами хранения в которых являются тексты. Под текстом понимаются неструктурированные данные, построенные из строк.

Проблемно ориентированные текстовые БД используются в разных областях медицины, юриспруденции, научно-технической информации, делопроизводства и др. Они хранят и обрабатывают огромные массивы постоянно увеличивающейся текстовой информации.

Системы обработки текстовой информации отличаются от систем обработки структурированных данных, имеющих фиксированный формат записи, в первую очередь типами хранимых данных.

Текстовые БД могут содержать различные тексты и документы, например тексты законов, библиографические описания, книги, рефераты, статьи и др.

Пакеты прикладных программ, предназначенные для ввода, обработки, поиска и обновления текстов, называют информационно-поисковой системой (ИПС).

В отличие от систем обработки структурированных данных ИПС, предназначенные для работы с текстом, имеют более гибкую стратегию поиска, т.е. поиск может проводиться по тексту либо по заданному описанию интересующей нас темы, а найденные документы должны частично либо полностью соответствовать сформулированному запросу. При поиске же структурированной информации, осуществляемом СУБД, запрос, принимаемый к исполнению, может быть сформулирован только в терминах определенного формального языка, а данные, выдаваемые на запрос при структурированном поиске, полностью соответствуют сформулированному запросу.

Перечень задач концептуального проектирования текстовой БД:

1. Анализ информационных интересов пользователей к данной предметной области.

2. Определение источников формирования БД.

3. Выбор архитектуры БД.

4. Разработка языка описания документов.

К анализу информационных интересов пользователей относится изучение информационных потребностей, информационных запросов, собственно потребителей информации. При анализе информационных интересов используются следующие методы:

- Социологические. Основываются на анкетировании и интервьюировании.
- Наукометрические. Основываются на анализе информационных потоков, которые представляют собой формальные каналы научных коммуникаций: отчеты по НИР, журналы, книги, каталоги, реферативные

журналы и обзоры, связь с автоматизированными информационными системами и т.д.

- Математико-статистические. К этим методам относятся методы ранговых распределений, методы построения выборочных совокупностей, корреляционный анализ, факторный анализ.

Все методы могут быть взаимосвязаны в единую методику, создание которой определяется предметной областью и интеллектом проектировщика. В конечном итоге вырабатывается механизм тематического распределения информации.

На основе анализа запросов и анализа информационных потоков принимается решение *об источниках формирования БД*, при этом рассматриваются следующие варианты:

1. Проектирование политематической текстовой БД из имеющихся БД.
2. Проектирование текстовой БД из первичных документов.

*Архитектура БД* выбирается исходя из интересов пользователей, вида документа, объема памяти ЭВМ.

*Язык описания документа* выбирается исходя из удобства интерфейса пользователя и приемлемой стоимости поиска.

На примере информационной системы «Кодекс» рассмотрим устройство и особенности работы с текстовыми базами данных.

База данных системы «Кодекс» представляет собой электронную юридическую библиотеку, содержащую полные тексты законодательных и нормативных актов Российской Федерации, документов региональных органов государственной власти, международных соглашений и других правовых и информационных материалов.

#### **4.1. Структура базы данных**

На физическом уровне база данных «Кодекс» представляет собой совокупность томов, подключенных к программному комплексу для работы. Том - это специальный файл, который содержит в себе объекты текстовой базы данных (разделы, документы, ссылки, элементы классификаторов, комментарии и т.п.). Каждый том базы данных характеризуется набором атрибутов, среди которых: наименование тома, номер обновления, описание структуры и др.

На логическом уровне (с точки зрения представления пользователя) базу данных «Кодекс» можно рассматривать как совокупность информационных (тематических) разделов.

Разделы могут иметь различную структуру данных, но все они логически связаны между собой, что позволяет говорить о едином информационном пространстве при работе с базой данных. Это выражается в наличии сквозного поиска по всем разделам и возможности быстрого перехода из раздела в раздел по гипертекстовым ссылкам.

В каждом тематическом разделе можно выделить две информационных области: область тематического справочника и область текстовых объектов.

Особой частью базы данных является область папок пользователя. В папках содержатся необходимые выборки материалов (документов, тематик, справок, консультаций и т.п.) и закладки в текстах материалов базы данных.

На рисунке 4.1 приведена примерная схема логической структуры базы данных «Кодекс».

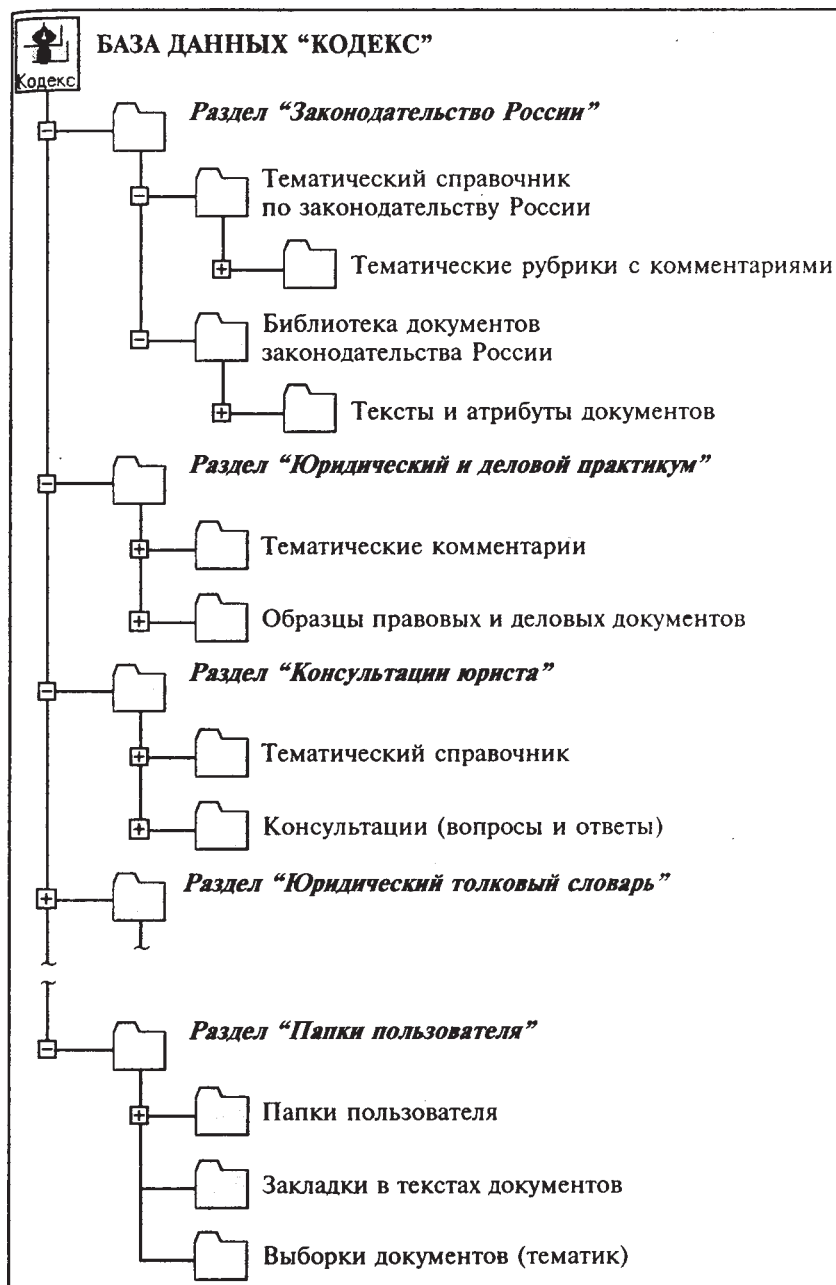


Рис. 4.1. Структура базы данных «Кодекс»

На схеме в качестве примера приведены несколько информационных разделов различной структуры.

Рассмотрим подробнее структуру информационного раздела “Законодательство”.

Область тематического справочника раздела представляет собой расширенный тематический классификатор.

Область текстовых объектов такого раздела представляет собой библиотеку документов, каждый из которых характеризуется следующим набором атрибутов:

- вид документа;
- дата принятия;
- номер;
- название документа;
- принявший орган;
- номер регистрации Минюста;
- дата регистрации Минюста;
- действующий/недействующий;
- дата начала действия;
- дата окончания действия;
- опубликование;
- дата редакции;
- дата изменения;
- дата включения в БД;
- тематика;
- тип документа.

Некоторые атрибуты могут иметь два и более значений (тематика, принявший орган, вид документа), но могут быть и “пустыми”, не иметь значения вовсе (например, атрибуты “N рег. Минюста” или “Дата рег. Минюста”).

Все атрибуты являются поисковыми в системе.

Список данных атрибутов появляется в окне-диалоге универсального атрибутного поиска системы «Кодекс», которое открывается специальными командами. Например командами НОВЫЙ ПОИСК, ПОИСК/ВЫБОРКА и др.

## **4.2. Поиск информации в базе данных**

Информационная система “Кодекс” содержит многообразные возможности поиска. К любой информации, находящейся в базе данных, существует несколько способов доступа. В зависимости от ситуации и от стоящей проблемы можно выбрать наиболее оптимальный путь поиска.

Рассмотрим более подробно некоторые возможности поиска.

### **4.2.1. Тематический (проблемный) поиск**

Тематический поиск рекомендуется в следующих случаях:

- пользователю необходимо получить список документов определенной тематической направленности;
- пользователь нечетко представляет проблему или ее границы и нужно полистать и просмотреть наименования тематических рубрик, а потом выбрать соответствующий список документов;
- пользователю необходимо просмотреть тематический комментарий по какой-либо теме.

Работа с тематическими комментариями очень проста: пользователь, передвигаясь по наименованиям тематических рубрик и, выбрав нужную, раскрывает её, просматривая дальше подразделы (подрубрики) выбранной тематики и т.д.

Перед началом тематического поиска следует выбрать группу разделов базы данных, в которых будет производиться поиск, загрузить страничку (информационное окно с перечнем разделов) данной группы и войти в какой-либо раздел.

В зависимости от состава базы данных, установленной на компьютере, меню Поиск будет содержать соответствующие команды выбора групп разделов (например, ЗАКОНОДАТЕЛЬСТВО, ПОМОЩНИК БУХГАЛТЕРА, КОНСУЛЬТАЦИИ..., СЛОВАРИ).

#### 4.2.2. Приблизительный контекстный поиск

Приблизительный контекстный поиск может быть рекомендован в следующих случаях:

- пользователь затрудняется точно сформулировать запрос;
- пользователь неточно помнит термины, которыми характеризуется интересующая его проблема;
- пользователь хочет получить в результате поиска максимум документов (материалов), пусть даже отдаленно относящихся к интересующей его проблеме.

По результатам поиска система выстраивает найденные документы по степени их близости к контексту заданного запроса.

Каждому документу присваивается “вес”. Документы с наибольшим “весом” имеют большую вероятность содержать полезную информацию и будут располагаться вверху списка. По мере движения вниз “вес” документов и соответственно вероятность содержания полезной информации – убывает. Работая со списком документов, полученных по результатам приблизительного поиска, надо отдавать отчет в том, что список не является абсолютно правильным: среди важных документов в нем будут представлены несущественные и случайные документы. Но даже несущественный документ может вывести на основные (главные) документы, соответствующие запросу (например, при переходе по гипертекстовой ссылке).

**Особенности формирования запросов при приблизительном контекстном поиске.** Запрос должен быть направлен на смысловое содержание документа, а не содержать вопрос о самом документе. Например, запрос «документы, принятые налоговой инспекцией по Москве» составлен неверно. Пример правильного запроса «льготы по налогу на прибыль для малых предприятий»

Составляя и вводя поисковую фразу (запрос), не надо задумываться о том, является ли какое-либо слово в данной фразе “шумовым” **Шумовые слова** - это часто встречающиеся, неинформативные слова (предлоги, местоимения, частицы, наиболее часто употребляемые сокращения). В целях уменьшения объема базы данных они не включаются в индексы, и контекстный поиск по ним невозможен. Система автоматически отбросит шумовые слова и те, которые не встречаются в базе данных. В запросе можно не использовать знаки препинания. Например, если пользователя интересует вопрос: “Как определяется выслуга лет для офицеров Вооруженных сил, увольняющихся в запас”, следует ввести в строке запроса поисковую фразу “выслуга лет офицеров Вооруженных сил увольняющихся в запас”.

#### 4.2.3. Контекстный поиск

**Контекстный поиск** - поиск по текстам документов (комментариев). Производится при выборе условия поиска «Поиск по тексту». Это самый мощный поиск в системе.

При вводе поисковой фразы система дает предупреждение о «шумовых словах», используемых в запросе. Такие слова выделяются цветом, появляется предупреждающее сообщение, и при поиске они будут отброшены. Кроме этого при вводе запроса автоматически подключается словарь (тезаурус) словоформ, что, с одной стороны, позволяет упростить процесс составления (ввода) поисковой фразы (пользователь выбирает предлагаемые системой слова, заведомо содержащиеся в базе данных), а с другой стороны, система сразу выделит во введенной поисковой фразе слова, которых нет в текстах базы данных.

Алгоритм контекстного поиска следующий:

- в слове (словах поисковой цепочки) выделяется основа (в каждом слове отбрасываются окончания);
- порядок слов в поисковой фразе не важен, слова фразы объединяются по принципу логического “И”;
- система воспринимает группу слов, заключенных в кавычки, как задание найти тексты, в которых искомые слова находятся рядом (составляют фразу);
- в поисковой фразе не пропускаются “шумовые слова”. В поисковой фразе контекстного поиска возможно использование чисел и буквенно-цифровых последовательностей (номера документов, химические формулы и т.п.). Допустимыми символами для таких последовательностей являются буквы русского и латинского языка, дефис, наклонная и обратная черта (“\” и “/”).

*Приведем пример задания логических выражений для условий контекстного поиска.*

Предположим, нас интересуют тексты, содержащие термины “малые предприятия” и “льготы”, и одновременно с этим либо термин “налог на прибыль”, либо “налог на добавленную стоимость”.

Переведем данную фразу на язык формального запроса. Как уже говорилось ранее, система воспринимает группу слов, заключенных в кавычки, как задание найти тексты, которые содержат данные слова подряд. Поэтому следует вводить термины как они есть, заключив их в кавычки (если термин из одного слова, его, очевидно, можно ввести без кавычек), а связки “и” и “либо - либо” заменить соответствующими логическими операторами: “И” (обозначается символом - &) и “ИЛИ” (обозначается символом - |).

В результате получим:

**“малые предприятия” & льготы & “налог на прибыль” | “налог на добавленную стоимость”**

При задании логического условия рекомендуется следовать следующим правилам:

1. Задавать максимально простые запросы, их всегда можно усложнить, если результирующий список слишком большой.
2. Если получен слишком большой список, следует просмотреть несколько документов, чтобы понять, почему выборка слишком большая, а лишь затем уточнять условие.

3. Помнить, что обычно у терминов есть синонимы. Следует указать в запросе через “ИЛИ” все подходящие синонимы.

В системе «Кодекс» для составления сложного контекстного запроса с использованием логических операторов используется так называемый Помощник, при вызове которого появляется окно-диалог. Окно имеет поле для ввода запроса, а также кнопки логических операторов, операторов расстояния и специальных символов, которые могут помочь пользователю в составлении запроса.

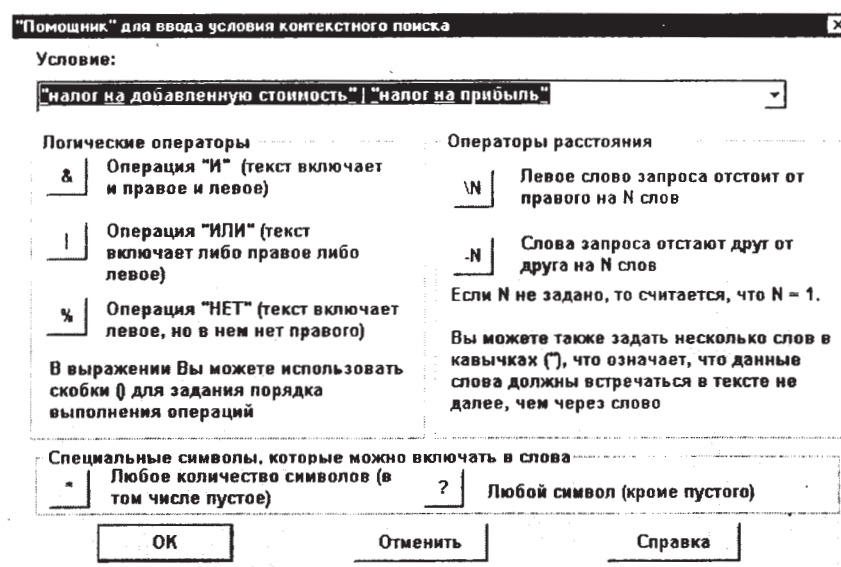


Рис. 4.2. Окно-диалог «Помощник» для ввода условия контекстного поиска

#### 4.2.4. Универсальный атрибутный поиск

Универсальный атрибутный поиск рекомендуется применять в случаях, когда пользователь более или менее точно знает, какие документы или материалы ему нужны, т.е. он знает значения их атрибутов или достаточно определенно может воспроизвести контекст, или когда он хочет осуществить поиск по комбинации различных признаков.

**Выбор поискового атрибута (условия поиска) в окне-диалоге системы «Новая выборка».** Список атрибутов, по которым можно осуществлять поиск, был приведен ранее.

Главная особенность данного вида поиска – возможность гибко комбинировать его условия.

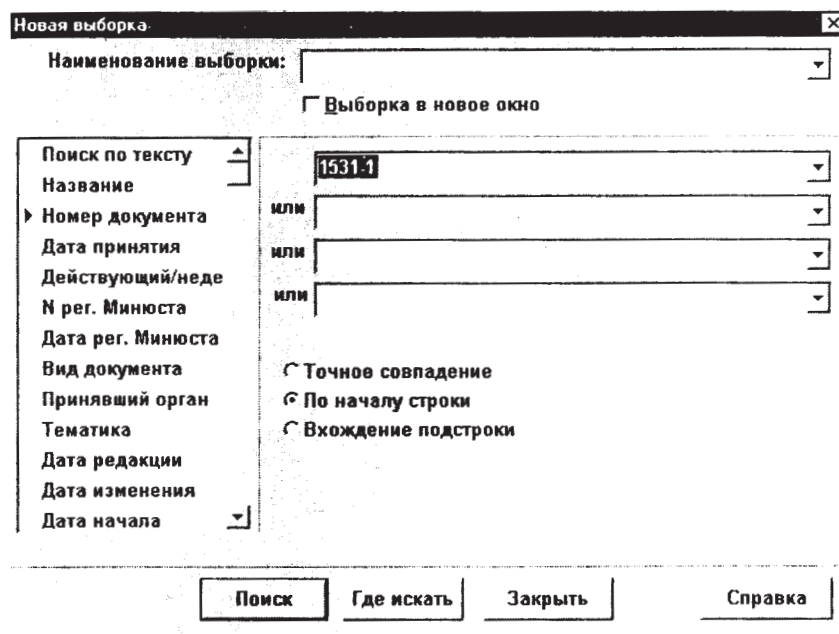


Рис 4.3 Пример поиска по атрибуту «Номер документа»

## 5. СУБД DBASE III

СУБД dBASE III – это одна из популярных и универсальных программ управления базами данных при работе на персональных ЭВМ, являющаяся системой управления реляционной базой данных. С ее помощью можно создать базу данных, а затем войти в нее и обращаться к информации, представленной там в различных форматах, используя для этого сравнительно небольшое количество команд. СУБД dBASE III можно использовать как средство управления базами данных самого разнообразного назначения – от систем автоматизированного проектирования и производства до систем информационного обслуживания (сбора, хранения, модификации и воспроизведения научно-технической, планово-экономической, учетно-статистической и другой информации).

СУБД dBASE III использует порядка 13 типов файлов, из которых основной - файл данных, имеющий расширение .dbf. Остальные можно рассматривать как приложения к файлу БД. Таким образом, информация, хранящаяся в файлах на диске, характеризуется определенными расширениями, которые описывают соответствующие типы файлов (табл. 5.1).

Накапливаемые данные помещаются в файл, имеющий специальную организацию и расширение .dbf.

**Поля файла данных.** *Символьное поле (Character)* предназначено для ввода символьной информации, такой, например, как фамилия. Длина символьного поля не превышает 254 байта.

Расширение	Файл
.dbf	Файл данных
.dbt	Файл примечаний (файл MEMO)
.ndx	Индексный файл
.fmt	Форматный файл, содержащий конкретную форму экрана
.scr	Экранный файл (содержит информацию о форматных файлах)
.txt	Текстовый файл
.qry	Файл условий фильтрации записей в запросах
.frm	Файл отчета
.lbl	Файл почтовых наклеек
.vue	Файл выборки (содержит информацию о файлах данных, их полях и связях)
.cat	Каталог файлов, используемых программой
.prg	Командный или процедурный файл
.mem	Файл временных переменных

*Числовое поле (Numeric)* может быть длиной до 19 байт. Это обеспечивает достаточно большую точность вычислений (до 15 значащих цифр).

*Логическое поле (Logical)* используется для хранения информации логического характера. Его размер – 1 байт. Особенно удобны эти поля для хранения результатов различных вопросов, анкет и т.д., поскольку анкеты можно составить так, чтобы на определенные вопросы в них вводили ответы “Да” (Yes) или “Нет” (No).

*Поле даты (Date)* имеет длину 8 байт. С данным полем можно выполнить некоторые операции, например посчитать разность между датами. Кроме того, в СУБД предусмотрены специальные функции для перевода полей типа «дата» в символьные поля и наоборот.

*Поле примечаний (Memo).* Используется когда необходимо хранить достаточно большие тестовые массивы, например характеристики сотрудников.

Система dBASE III предоставляет пользователю возможность работы в нескольких режимах.

*Первый способ – работа в режиме «ассистент» (assist),* который связан с использованием набора меню. Он обеспечивает решение основных задач управления реляционной базой данных без использования (и даже знания) команд языка программирования dBASE. Основным недостаток данного режима работы – необходимость повторять ответы на многочисленные вопросы системы в тех случаях, когда приходится выполнить подряд несколько однородных запросов.

*Второй способ – работа в командном режиме СУБД dBASE III,* позволяющая выполнять любые операции. В тех случаях, когда пользователь затрудняется в правильном написании команд, он может воспользоваться «помощью», нажав клавишу F1. При этом на экран дисплея выводятся формат любой команды и ее краткое описание. Командный язык удобен и быстр, однако при необходимости продублировать операцию приходится повторять ввод одной или нескольких команд. Работа на командном языке имеет и другой недостаток – невозможность повторять команду или группу команд, пока не выполнится некоторое заранее заданное условие.

Поэтому пользователю *предоставляется возможность писать целые программы на процедурном языке dBASE.* Это в итоге позволяет создавать целые пакеты программ под конкретные задачи. Однако для этого требуются некоторые познания в английском языке (для чтения сообщений операционной системы при отладке программы), а также знание основ работы с системой dBASE.

Рассмотрим основы работы с системой – в режиме «ассистент».

После входа в среду dBASE набираем команду *assist* в командной строке либо, что гораздо проще, – нажимаем функциональную клавишу [F2]. С помощью навигационных клавиш [->] и [<-] можно выбрать курсором любой пункт главного меню. Под выбранным пунктом главного меню находится большой прямоугольник, обведенный двойной линией, который содержит несколько подменю. Работа с пунктами главного меню и подменю традиционная. Используя навигационные клавиши [↑] и [↓], можно открыть любой выбранный пункт подменю, нажав [Enter]. После этого система начнет выполнять выбранный пункт или выдаст новое подменю, которое будет уже меню следующего уровня.

Поставим задачу создать БД, в которой содержится информация о сотрудниках некоего предприятия. При этом структура файла БД определяет такую последовательность набора полей: табельный номер служащего; Фамилия И. О.; дата рождения; семейное положение; номер отдела, где работает сотрудник; зарплата.

Для создания БД необходимо определить имя для файла БД, установить размер полей, назвать и определить их тип.

Назовем файл БД «kadru», а все остальные характеристики, произвольно выбранные нами для определения структуры файла БД «kadru», приведем в табл 5.2.

Таблица 5.2.

№	Имя поля	Тип поля	Длина поля	Точность	Значение поля
1	TAB	Символьное	5		Табельный номер
2	FAM	Символьное	15		Фамилия И. О.
3	DTR	Дата	8		Дата рождения
4	SEM	Логическое	1		Семейное положение
5	OTD	Числовое	2		Номер отдела
6	ZRP	Числовое	7	2	Заработная плата

В главном меню выберем пункт «Создать», а в соответствующем ему подменю – пункт «файл БД» и нажмем [Enter]. В ответ СУБД предложит выбрать для создания БД любой диск, начиная от A: и кончая E:. Укажем курсором диск C: и нажмем [Enter]. На запрос СУБД «Введите имя файла:» наберем на клавиатуре слово «kadru» – имя БД и нажмем [Enter].

При этом, если обратить внимание на строки внизу экрана – строку состояния, подсказки и помощи, то можно убедиться, что все действия отражаются в виде команд dBase.

В данном случае команда примет вид CREATE C:KADRU. Отметим, что, в любой момент можно воспользоваться [F1], вызвав «помощь». В этом случае на экран дисплея выводится краткое описание и формат команды, инициируемой с помощью соответствующих пунктов меню «assist».

Как только ввели имя файла БД «kadru» (расширение .dbf система присваивает сама), на экране дисплея реализуется специальный режим работы команды, представляемый в виде таблицы, в которую вводится информация о структуре файла БД. В верхней части экрана выводится также описание дополнительных команд редактирования.

Используя данные табл.5.2, последовательно заполним строки выводимой на экран дисплея таблицы именами полей и их характеристиками. Переход от имени поля к его характеристикам осуществляется нажатием [Enter], а тип полей выбирается нажатием клавиши «Пробел». Длины логического поля и поля даты устанавливаются автоматически самой системой.

После того, как структура файла БД будет определена и примет вид, приводимый на рис. 5.1, следует выйти из режима команды CREATE, нажав [Ctrl] + [End] (которые реализуют выход с сохранением), и подтвердить свой выход нажатием [Enter]. В ответ система запросит: «Будете вводить данные? (Y/N)». Если да - нажмем [Y] (Yes).

Команда CREATE сменится специальным режимом команды APPEND, которая используется для ввода информации и добавления записей в конец активной БД. Для более удобного ввода информации команда APPEND работает в специальном режиме, называемом «полноэкранный ввод данных». Это значит, что при вводе данные размещаются на целом экране и появляется возможность вводить и исправлять информацию в любом поле. На рис. 5.2 представлена первая запись:

Адамов А. К., 20 января 1924 года рождения, семейный, из 1-го отдела, с окладом 1200 руб.

Доступно байт: 3962

КУРСОР <-- --> Симв: Слово: Home End Кадр:	ВСТАВКА Симв: Ins Поле: ^M Помощь: F1	УДАЛЕНИЕ Симв: Del Слово: ^Y Поле: ^U	Поле вверх: Поле вниз: Выход/Сохранение: ^End Отмена: Esc
---	--	--	--

Имя Поля	Тип	Длина	Точ
1 TAB	Символ	5	
2 FAM	Символ	15	
3 DTR	Дата	8	
4 SEM	Логич	1	
5 OTD	Число	2	8
6 ZRP	Число	7	2

CREATE	<C>	KADRU	Поле: 6/6
--------	-----	-------	-----------

Введите имя поля.  
Имена полей начинаются с буквы и могут состоять из букв, цифр и подчеркиваний.

Рис. 5.1. Режим команды создания файла БД

КУРСОР <-- --> Символ: Слово: Home End	ВВЕРХ ВНИЗ Поля: Экран: PgUp PgDn Помощь: F1	УДАЛЕНИЕ Символ: Del Поля: ^Y Записи: ^U	Вставка: Ins Выход/Сохранение: ^End Отмена: Esc Примечание: ^Home
--	---	---	--

TAB	1
FAM	Адамов А.К.
DTR	81/28/24
SEM	1
OTD	1
ZRP	1288.88

CREATE	<C>	KADRU	Зап: 1/1
--------	-----	-------	----------

Рис. 5.2. Режим «полноэкранный ввод данных»

Для перехода к новому полю необходимо нажать [Enter]. После ввода информации в БД вернемся в главное меню. Для этого выйдем из режима ввода информации команды APPEND, для чего воспользуемся нажатием [Ctrl] + [End] (которые реализуют выход с сохранением), и нажмем [Enter] для подтверждения своего выхода.

## ТРЕНИНГ УМЕНИЙ

### 1. Пример выполнения упражнения тренинга на умение 1

#### Задание

Опишите логическую структуру записи файла “Сотрудники фирмы”.

#### Решение

Предварительно заполните таблицу, подобрав к каждому алгоритму конкретное соответствие из данного задания.

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму
1	Определить последовательность расположения полей записи	1. Табельный номер 2. Ф.И.О. 3. Год рождения 4. Адрес 5. Образование 6. Стаж работы
2	Присвоить каждому полю уникальное имя	1. Табельный номер 2. Ф.И.О. 3. Год рождения 4. Адрес 5. Образование 6. Стаж
3	Определить тип поля	1. Табельный номер – Символьное 2. Ф.И.О. – Символьное 3. Год рождения – Дата 4. Адрес – Символьное 5. Образование – Символьное 6. Стаж работы – Числовое
4	Определить длину (размер) поля	1. Табельный номер – 6 2. Ф.И.О. – 30 3. Год рождения – 8 4. Адрес – 40 5. Образование – 20 6. Стаж работы – 4
5	Указать точность поля	Точность указывается только для числовых полей. Стаж работы – один знак после запятой. Точность – 1.

Логическая структура записи файла “Сотрудники фирмы”

ПОЛЕ	ИМЯ	ТИП	ДЛИНА	ТОЧНОСТЬ
Табельный номер	Таб. номер	Символьное	6	
Ф.И.О.	Ф.И.О.	Символьное	30	
Год рождения	Год рождения	Дата	8	
Адрес	Адрес	Символьное	40	
Образование	Образование	Символьное	20	
Стаж работы	Стаж	Числовое	4	1

*Выполните самостоятельно:*

*Задание 1*

Опишите логическую структуру записи файла “Студенты”.

*Задание 2*

Опишите логическую структуру записи файла “Компьютеры”.

## 2. Пример выполнения упражнения тренинга на умение 2

### Задание

Составьте иерархическую схему базы данных “Транспорт”.

### Решение

Предварительно заполните таблицу, подобрав к каждому алгоритму конкретное соответствие из данного задания.

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму
1	Изучить правила организации связи между объектами иерархической модели данных	
2	Определить узлы (объекты) модели	Транспорт. Воздушный транспорт. Наземный транспорт. Водный транспорт. Железнодорожный транспорт. Автотранспорт. Легковой транспорт. Грузовой транспорт. Автобусный транспорт.
3	Выделить главный (корневой) узел	Транспорт.
4	Задать отношение подчиненности между узлами	Транспорт – воздушный, наземный, водный. Наземный транспорт – железнодорожный, автотранспорт. Автотранспорт – легковой, грузовой, автобусный.



*Выполните самостоятельно:*

*Задание 1*

Постройте иерархическую схему базы данных “Университет”

*Задание 2*

Постройте иерархическую схему базы данных “Растительный мир”

### 3. Пример выполнения упражнения тренинга на умение 3

#### Задание

Постройте таблицу СОТРУДНИК ФИРМЫ реляционной модели базы данных.

#### Решение

Предварительно заполните таблицу, подобрав к каждому алгоритму конкретное соответствие из данного задания.

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму																					
1	Изучить правила создания реляционных таблиц																						
2	Определить объекты модели	В соответствии с заданием объект – СОТРУДНИК ФИРМЫ.																					
3	Описать структуру объекта	Пользователя интересует следующая информация о СОТРУДНИКЕ ФИРМЫ: Фамилия, имя, отчество. Табельный номер. Возраст. Место проживания. Образование и стаж.																					
4	Сформировать столбцы таблицы	Табельный номер. Фамилия, имя, отчество. Год рождения. Адрес. Образование. Стаж.																					
5	Присвоить уникальное имя каждому столбцу	Табельный номер Ф.И.О. Год рождения Адрес Образование Стаж																					
6	Выделить ключ	Табельный номер																					
7	Определить тип и длину элемента столбца	<table> <tr> <th>№ столбца</th><th>Тип</th><th>Длина</th></tr> <tr> <td>1.</td><td>Символьное</td><td>6</td></tr> <tr> <td>2.</td><td>Символьное</td><td>30</td></tr> <tr> <td>3.</td><td>Дата</td><td>8</td></tr> <tr> <td>4.</td><td>Символьное</td><td>40</td></tr> <tr> <td>5.</td><td>Символьное</td><td>20</td></tr> <tr> <td>6.</td><td>Числовое</td><td>4</td></tr> </table>	№ столбца	Тип	Длина	1.	Символьное	6	2.	Символьное	30	3.	Дата	8	4.	Символьное	40	5.	Символьное	20	6.	Числовое	4
№ столбца	Тип	Длина																					
1.	Символьное	6																					
2.	Символьное	30																					
3.	Дата	8																					
4.	Символьное	40																					
5.	Символьное	20																					
6.	Числовое	4																					

Таблица “СОТРУДНИК ФИРМЫ” реляционной модели базы данных  
Ключ выделен серым цветом.

Таб. номер	Ф.И.О.	Год рождения	Адрес	Образование	Стаж
003453	Королев Ф.В.	1968	Москва, пр-т Мира	среднее	12
002356	Ларин П.О.	1953	Москва, пр-т Мира	высшее	22
002308	Митин С.А.	1947	Москва, пр-т Мира	средне- техническое	31

ИМЯ	ТИП	ДЛИНА
Табельный номер	Символьное	6
Ф.И.О	Символьное	30
Год рождения	Дата	8
Адрес	Символьное	40
Образование	Символьное	20
Стаж	Числовое	4

*Выполните самостоятельно:*

### *Задание 1*

Постройте таблицу СТУДЕНТ реляционной модели базы данных.

## Задание 2

Постройте таблицу КОМПЬЮТЕР реляционной модели базы данных.

### 4. Пример выполнения упражнения тренинга на умение 4

#### Задание

Создайте БД в системе dBASE III, содержащую данные о сотрудниках фирмы. Файл БД должен располагаться на диске "С" в каталоге DB3. Имя файла БД определите самостоятельно.

#### Решение

Предварительно заполните таблицу, подобрав к каждому алгоритму конкретное соответствие из данного задания.

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму
1	Определить структуру БД	Воспользуемся структурой записи «Сотрудники фирмы» из примера 1.
2	В главном меню системы выбрать пункт «Создать»	Подведем курсор к пункту главного меню «Создать» и нажмем [Enter].
3	В раскрывшемся подменю выбрать пункт «файл БД»	Подведем курсор к пункту «файл БД» и нажмем [Enter].
4	Указать дисковод и каталог для размещения файла	C:\DB3
5	В ответ на запрос СУБД ввести имя файла	Назовем файл БД «Employee».

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму
6	В появившуюся на экране таблицу ввести информацию о структуре файла	Последовательно заполним строки появившейся на экране дисплея таблицы. Сначала введем имя поля, а затем его характеристики. Переход от имени поля к его характеристикам осуществляется нажатием [Enter], а тип полей выбирается нажатием клавиши «Пробел».
7	Выйти из режима создания, нажав клавиши [Ctrl] + [End]	Нажимаем клавиши [Ctrl] + [End]

Структура базы данных “Сотрудники фирмы”

№ п/п	ПОЛЕ	ТИП	ДЛИНА	ТОЧНОСТЬ
1	Табельный номер	Символьное	6	
2	Ф.И.О.	Символьное	30	
3	Год рождения	Дата	8	
4	Адрес	Символьное	40	
5	Образование	Символьное	20	
6	Стаж работы	Числовое	4	1

Структура файла БД “Сотрудники фирмы” в системе dBASE III.

Имя поля	Тип	Длина	Точность
1 TAB	Символ	6	
2 FIO	Символ	30	
3 GDR	Дата	8	
4 ADR	Символ	40	
5 OBR	Символ	20	
6 STG	Число	4	1

*Выполните самостоятельно:*

### *Задание 1*

Создайте БД в системе dBASE III, содержащую данные о студентах. Файл БД должен располагаться на диске "С" в каталоге UNIVERSITY. Имя файла БД определите самостоятельно.

### *Задание 2*

Создайте БД в системе dBASE III, содержащую данные о компьютерах. Файл БД должен располагаться на диске "А" в каталоге COMPUTERS. Имя файла БД определите самостоятельно.

## 5. Пример выполнения упражнения тренинга на умение 5

(использование “Помощника” в информационной системе “Кодекс” для создания запроса с использованием логических операторов)

### Задание

Создайте запрос для выборки документов , содержащих термины “налог на добавленную стоимость” либо “налог на прибыль”.

### Решение

Предварительно заполните таблицу, подобрав к каждому алгоритму конкретное соответствие из данного задания.

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму
1	Нажать кнопку [fx] для раскрытия окна-диалога «Помощника»	Щелкнуть левой кнопкой мыши по кнопке [fx].
2	В поле ввода запроса ввести первую фразу поискового запроса, заключив ее в кавычки	«налог на добавленную стоимость»
3	Нажать кнопку оператора «ИЛИ» (вертикальная черта) или оператора «И» (&)	Нажать кнопку оператора "ИЛИ" ( ).
4	Ввести вторую фразу поискового запроса, заключив ее в кавычки	«налог на прибыль»
5	Нажать кнопку команды «ОК»	Щелкнуть левой кнопкой мыши по кнопке «ОК».
6	Нажать кнопку «Поиск» в раскрывшемся окне «Новая выборка»	Щелкнуть левой кнопкой мыши по кнопке «Поиск».
7	Нажать кнопку «Список» для вывода выборки в информационное окно	Щелкнуть левой кнопкой мыши по кнопке «Список».

Поисковая фраза:

**“налог на добавленную стоимость” п “налог на прибыль”**

*Выполните самостоятельно:*

### *Задание 1*

Создайте запрос для выборки документов , содержащих термины “налог на имущество” и “физические лица”.

### *Задание 2*

Создайте запрос для выборки документов, содержащих термины “отставка президента”, либо термины “ропуск Государственной Думы”.

## **6. Пример выполнения упражнения тренинга на умение 6**

Пример создания запроса-выборки в СУБД Access

### **Задание**

Создайте запрос-выборку, позволяющий получить из таблицы “Сотрудники фирмы” список сотрудников, чей стаж превышает 20 лет.  
Воспользуйтесь таблицей “Сотрудник фирмы” из примера 3.

### **Решение**

Предварительно заполните таблицу, подобрав к каждому алгоритму конкретное соответствие из данного задания.

№ п/п	Алгоритмы	Конкретное соответствие данной ситуации предложенному алгоритму
1	В окне базы данных нажать кнопку «Запрос»	Щелкнуть левой кнопкой мыши по кнопке «Запрос».
2	Нажать кнопку «Создать»	Щелкнуть левой кнопкой мыши по кнопке «Создать».
3	В диалоге «Создание запроса» нажать кнопку «Новый запрос»	Щелкнуть левой кнопкой мыши по кнопке «Новый запрос».
4	В диалоге «Добавление таблицы» указать таблицу для выборки	«Сотрудник фирмы»
5	Указать поля, по которым будет проводиться поиск, перетаскив их из списка полей выбранной таблицы в ячейки строки «Поле» окна «Запроса»	Из списка полей таблицы «Сотрудник фирмы» в первую ячейку строки «Поле» окна «Запроса» перетащим поле «Таб. номер», во вторую - «Ф.И.О.», в третью - «Стаж».
6	В ячейки строки «Условие отбора» поместить значения условий	В третью ячейку строки «Условие отбора» поместим выражение: <b>&gt;20</b>
7	Выполнить запрос, щелкнув на кнопке пиктографического меню «Выполнить»	Щелкнуть левой кнопкой мыши по кнопке «Выполнить».

ПОЛЕ	Таб. номер	Ф.И.О.	Стаж
<b>УСЛОВИЕ ОТБОРА</b>			<b>&gt;20</b>

#### ВЫБОРКА.

Таб. номер	Ф.И.О.	Стаж
002356	Ларин П.О.	22
002308	Митин С.А.	31

*Выполните самостоятельно:*

### *Задание 1*

Создайте запрос-выборку, позволяющий получить из таблицы “Студент” список студентов, обучающихся на юридическом факультете.

Воспользуйтесь таблицей “Студент” из примера 3, задание 1.

### *Задание 2*

Создайте запрос-выборку, позволяющий получить из таблицы “Компьютер” данные о компьютерах (его модели, процессоре, тактовой частоте), объем оперативной памяти (RAM) которых превышает 16 мб, а емкость жесткого диска - более 5 гб.

Воспользуйтесь таблицей “Компьютер” из примера 3 задания 2.

## **ИНФОРМАТИКА. РАСШИРЕННЫЙ КУРС**

### **ЮНИТА 1**

#### **БАЗЫ ДАННЫХ НА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ И ИХ ИСПОЛЬЗОВАНИЕ**

Редактор Л.С. Лебедева  
Оператор компьютерной верстки Д.В. Федотов

---

Изд. лиц. ЛР № 071765 от 07.12.1998  
НОУ “Современный Гуманитарный Институт”  
Тираж

Сдано в печать  
Заказ

---

Современный Гуманитарный Университет