

## БАЗЫ ДАННЫХ (БД). СИСТЕМЫ УПРАВЛЕНИЯ БД

### Общие положения

Цель любой информационной системы - обработка данных об объектах реального мира. 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 извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы.

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

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

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 структурированные данных, относящихся к определенной предметной области.

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

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

### Классификация баз данных

1) 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).

2) 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 монопольно. Концепция файл-сервер условно отображена на рис. 15.3.

**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.15.4

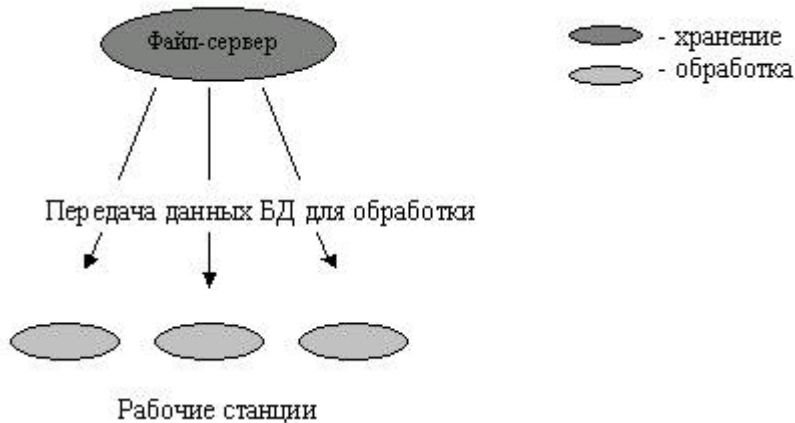


Схема обработки информации в БД по принципу файл-сервер

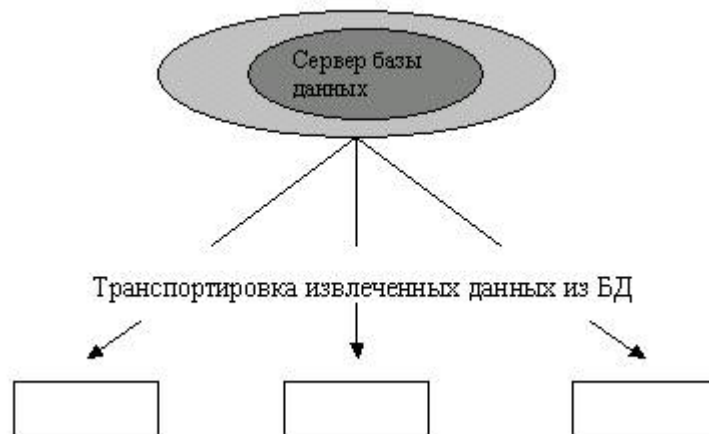


Схема обработки информации в БД по принципу клиент-сервер

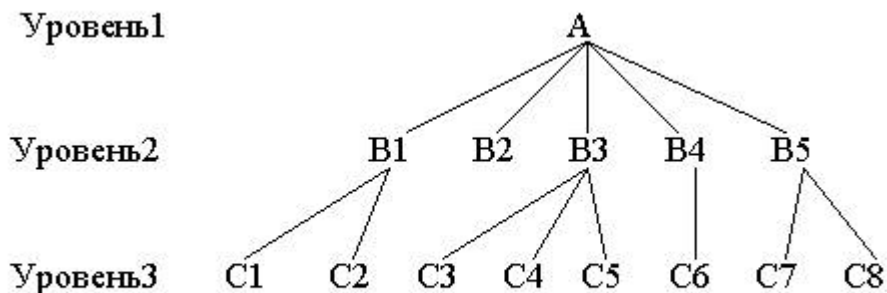
### Структурные элементы базы данных

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица) (рис. 15.5).



объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи. Например, как видно из рис., для записи С4 путь проходит через записи А и В3.



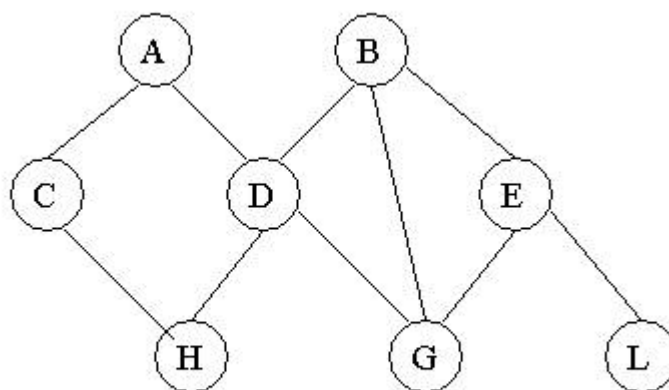
### Графическое изображение иерархической структуры БД

### 2) Сетевая модель данных

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

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

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. Единственное отношение представляет собой сложную связь между записями в обоих направлениях.



### Графическое изображение сетевой структуры

## 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**. 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.

Реляционной таблицей можно представить информацию о студентах, обучающихся в вузе.

### Пример реляционной таблицы

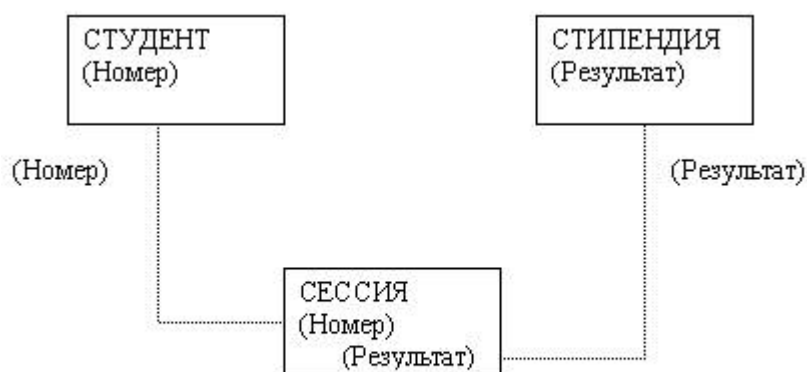
№ дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Анохин	Андрей	Борисович	14.04.76	111

Отношения представлены в виде **таблиц**, строки которых соответствуют кортежам или **записям**, а столбцы - атрибутам отношений, доменам, **полям**.

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 в табл., ключевым полем таблицы является "№ личного дела".

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно ввести в структуру первой таблицы **внешний ключ** - ключ второй таблицы.

пример реляционной модели, построенной на основе отношений: **СТУДЕНТ, СЕССИЯ, СТИПЕНДИЯ**.



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.

## СУБД ACCESS

Большие массивы данных в определенном формате, но и обрабатывать их, представляя в удобном для пользователей виде. Access дает возможность также автоматизировать часто выполняемые операции (например, расчет заработной платы, учет материальных ценностей и т.п.). С помощью Access можно не только разрабатывать удобные формы ввода и просмотра данных, но и составлять сложные отчеты.

Access является приложением Windows, а поскольку и Windows и Access разработаны одной фирмой (Microsoft), они очень хорошо взаимодействуют друг с другом. СУБД Access работает под управлением Windows; таким образом, все преимущества Windows доступны в Access, например, вы можете вырезать, копировать и вставлять данные из любого приложения Windows в приложение Access и наоборот.

Access - это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных. Применение реляционной СУБД помогает упростить структуру данных и таким образом облегчить выполнение работы. Таблицу Access можно связать с данными, хранящимися на другом компьютере или на сервере, а также использовать таблицу, созданную в СУБД Paradox или Dbase. Данные Access очень просто комбинировать с данными Excel.

В СУБД Access предусмотрено много дополнительных сервисных возможностей. *Мастера* помогут вам создать таблицы, формы или отчеты из имеющихся заготовок, сделав за вас основную черновую работу. *Выражения* используются в Access, например, для проверки допустимости введенного значения. *Макросы* позволяют автоматизировать многие процессы без программирования, тогда как встроенный в Access язык *VBA* (Visual Basic for Applications) - специально разработанный компанией Microsoft диалект языка Basic для использования в приложениях Microsoft Office - дает возможность опытному пользователю программировать сложные процедуры обработки данных. Просматривая свою форму или отчет, вы сможете представить, как они будут выглядеть в распечатанном виде. И наконец, используя такие возможности языка программирования C, как функции и обращения к Windows API (Application Programming Interface - интерфейс прикладных программ Windows), можно написать подпрограмму для взаимодействия Access с другими приложениями - источниками данных.

В Microsoft Access добавлено множество новых средств, разработанных для облегчения работы в Интернет и создания приложений для Web. Для доступа к сети Интернет и использования преимуществ новых средств необходимы средства просмотра Web, например Microsoft Internet Explorer, а также модем. Пользователь имеет возможность непосредственно подключаться к узлам Microsoft Web из программ Office (в том числе и из Access) с помощью команды **Microsoft на Web** из пункта меню ?. При этом можно, например, получить доступ к техническим ресурсам и загрузить общедоступные программы, не прерывая работу с Access.

Система Access содержит набор инструментов для управления базами данных, включающий конструкторы таблиц, форм, запросов и отчетов. Кроме того, Access можно рассматривать и как среду для разработки приложений. Используя макросы для автоматизации задач, вы можете создавать такие же мощные, ориентированные на пользователя приложения, как и приложения, созданные с помощью "полноценных" языков программирования, дополнять их кнопками, меню и диалоговыми окнами. Программируя на VBA, можно создавать программы, по мощности не уступающие самой Access. Более того, многие средства Access, например *мастера* и *конструкторы*, написаны на VBA. Мощность и гибкость системы Access делают ее сегодня одной из лучших программ для управления базами данных.

Пример проектирования БД в Access показан на рис.

