

Базы данных. Системы управления базами данных

Лекция 3



Содержание лекции:

1. *История появления и развития БД*
2. *Предметная область, основные понятия*
3. *Классификация СУБД*
4. *Состав и структурные элементы СУБД*
5. *Модель данных*
6. *Основные функции СУБД*
7. *Функциональные возможности СУБД*
8. *Подходы к проектированию БД*



История появления и развития БД

На современном этапе развития информационных технологий достигнуты большие успехи в получении, обработке, хранении, управлении, анализе и визуализации данных. Совместно эти задачи называются **управлением данными**.

Для хранения и обработки данных можно использовать либо **электронные таблицы**, либо **системы управления базами данных**. Использование систем управления базами данных требует более высокого уровня подготовки для создания таблицы и описания обработки ее данных.

История появления и развития БД

В истории вычислительной техники можно проследить развитие двух основных областей ее использования.

Первая область — применение вычислительной техники для выполнения численных расчетов.

Вторая область — это использование средств вычислительной техники в автоматических или автоматизированных информационных системах.



История появления и развития БД Информационные системы

Информационные системы имеют дело с большими объемами информации, имеющей достаточно сложную структуру (банковские системы, автоматизированные системы управления предприятиями, системы резервирования авиационных и железнодорожных билетов, мест в гостиницах и т.д.)

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



История появления и развития БД Информационные системы

Разработчики информационных систем предложили новый подход к управлению информацией.

Этот подход был реализован в рамках новых программных систем, названных впоследствии **Системами Управления Базами Данных (СУБД)**, а сами хранилища информации, которые разработаны под управлением данных систем, назывались **базами или банками данных (БД и БнД)**.

Основные требования, предъявляемые к банкам данных: многократное использование данных, простота, легкость использования, гибкость использования, быстрая обработка запросов на данные, язык взаимодействия.

Основой информационной системы является база данных.

Целью любой информационной системы является обработка данных об объектах реального мира.

История появления и развития БД

Теория баз данных — сравнительно молодая область знаний. Возраст ее составляет немногим более 45 лет.

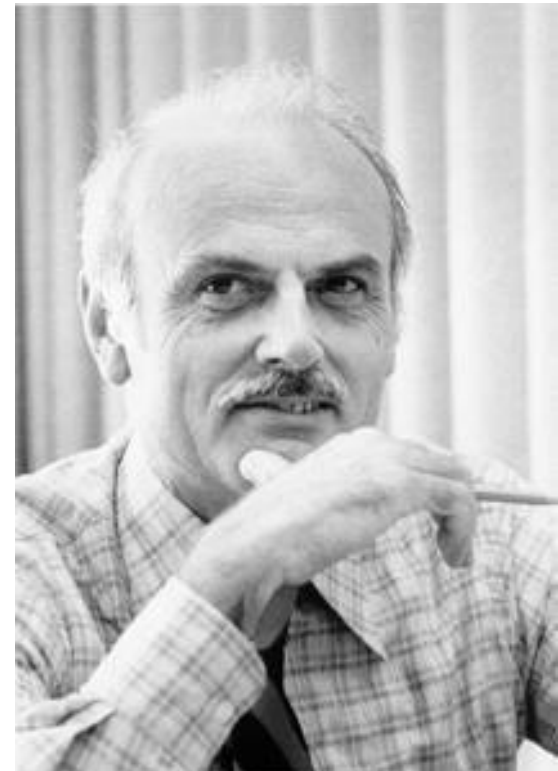
В 1968 году была введена в эксплуатацию первая промышленная СУБД система IMS фирмы IBM.

В 1975 году появился первый стандарт ассоциации по языкам систем обработки данных — **Conference of Data System Languages (CODASYL)**, который определил ряд фундаментальных понятий в теории систем баз данных, которые и до сих пор являются основополагающими для сетевой модели данных.

История появления и развития БД

В дальнейшее развитие теории баз данных большой вклад был сделан американским математиком **Э.Ф.Коддом**, который является создателем реляционной модели данных.

В 1981 году Э.Ф.Кодд получил за создание реляционной модели и реляционной алгебры престижную премию Тьюринга Американской ассоциации по вычислительной технике.



Эдгар Кодд

История появления и развития БД

Появление мощных рабочих станций и сетей ЭВМ повлияло также и на развитие технологии баз данных.

Можно выделить **четыре этапа** в развитии данного направления в обработке данных.



История появления и развития БД

Первый этап развития БД связан с организацией баз данных на больших ЭВМ типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation — DEC), разных моделях HP (фирмы Hewlett Packard).

Характеристика БД первого этапа:

- Базы данных хранились во внешней памяти центральной ЭВМ.
- Программы доступа к БД писались на различных языках.
- Доступ к БД поддерживался от многих пользователей-задач (распределенный доступ).

История появления и развития БД

Второй этап - появление персональных компьютеров

Появились первые программы - **системы управления базами данных**, которые позволяли хранить значительные объемы информации, они имели удобный интерфейс для заполнения данных, встроенные средства для генерации различных отчетов. Появились настольные СУБД. Эти программы позволяли автоматизировать многие **учетные** функции, которые раньше велись вручную.

Характеристика БД второго этапа:

- Все СУБД были рассчитаны на создание БД в основном с монопольным доступом.
- В настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.
- Сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД.

История появления и развития БД

Третий этап - распределенные базы данных

Появившиеся *распределенные базы данных*, сохраняют все преимущества настольных СУБД и в тоже время позволяют организовать параллельную обработку информации и поддержку целостности БД

Характеристика БД третьего этапа

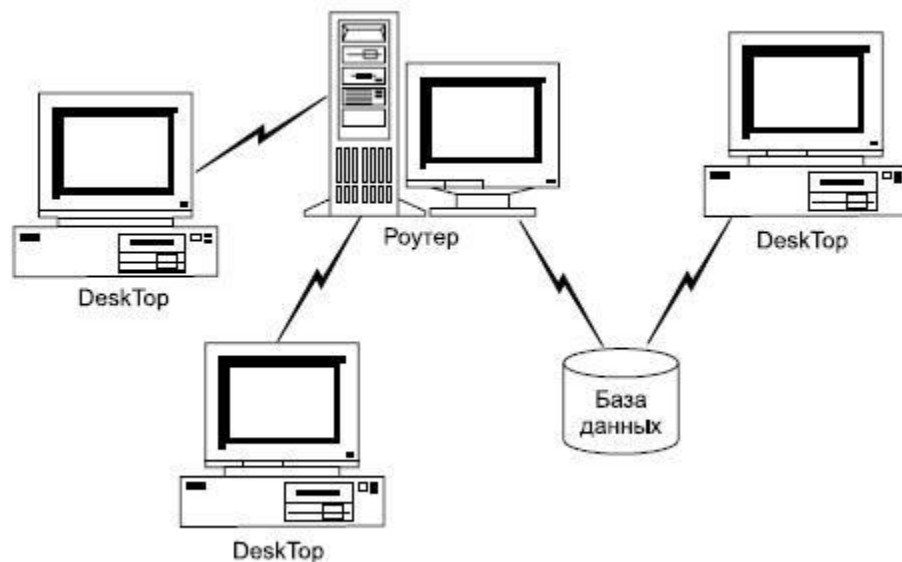
- Практически все современные СУБД обеспечивают поддержку полной реляционной модели.
- Большинство современных СУБД рассчитаны на многоплатформенную архитектуру.
- Необходимость поддержки многопользовательской работы с базой данных и возможность децентрализованного хранения данных потребовали развития средств администрирования БД с реализацией общей концепции средств защиты данных.

История появления и развития БД

Четвертый этап – сетевой доступ к системе управления базами данных

Этот этап характеризуется появлением новой технологии доступа к данным — интранет. Основное отличие этого подхода от технологии клиент-сервер состоит в том, что отпадает необходимость использования специализированного клиентского программного обеспечения.

Для работы с удаленной базой данных используется стандартное программное приложение - браузер Интернета и для конечного пользователя процесс обращения к данным происходит аналогично работе с ресурсами в сети Интернет.

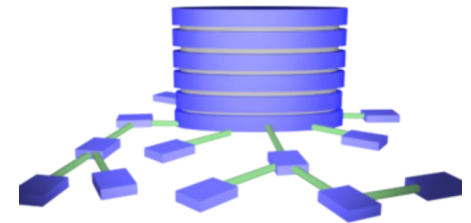


История появления и развития БД: поколения СУБД

К **СУБД первого поколения** относят СУБД на основе *сетевой модели данных* (их иногда называют CODASYL-системы) и системы на основе иерархических подходов.

CODASYL (англ. COⁿference on DA^Ta SY^Stems Language — Конференция по языкам систем обработки данных) — организация (название произносится «кодасил»), принимавшая активное участие в эволюции информационных технологий в 60-80-е годы XX века.

СУБД второго поколения – *реляционные*.



СУБД третьего поколения – *объектно-реляционные и объектно-ориентированные*.

ОСНОВНЫЕ ПОНЯТИЯ

База данных (БД) - это именованная совокупность структурированных данных, отображающая состояние объектов и их отношения в рассматриваемой предметной области.

Под **предметной областью** принято понимать часть реального мира, подлежащего изучению.

Структурирование - введение соглашений о способах представления данных.

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

Пользователей (СУБД) можно разделить на две основные категории: *конечные пользователи*; *администраторы баз данных*.

Классификация СУБД

В силу многогранности баз данных и СУБД (комплекса технических и программных средств, для хранения, поиска, защиты и использования данных) имеется множество классификационных признаков:

По степени универсальности (сфере применения) :

- СУБД общего назначения (СУБД ОН) и специализированные СУБД (СпСУБД).

По используемой модели данных

- иерархические, сетевые, реляционные; объектно-ориентированные СУБД.

По методам организации хранения и обработки данных :

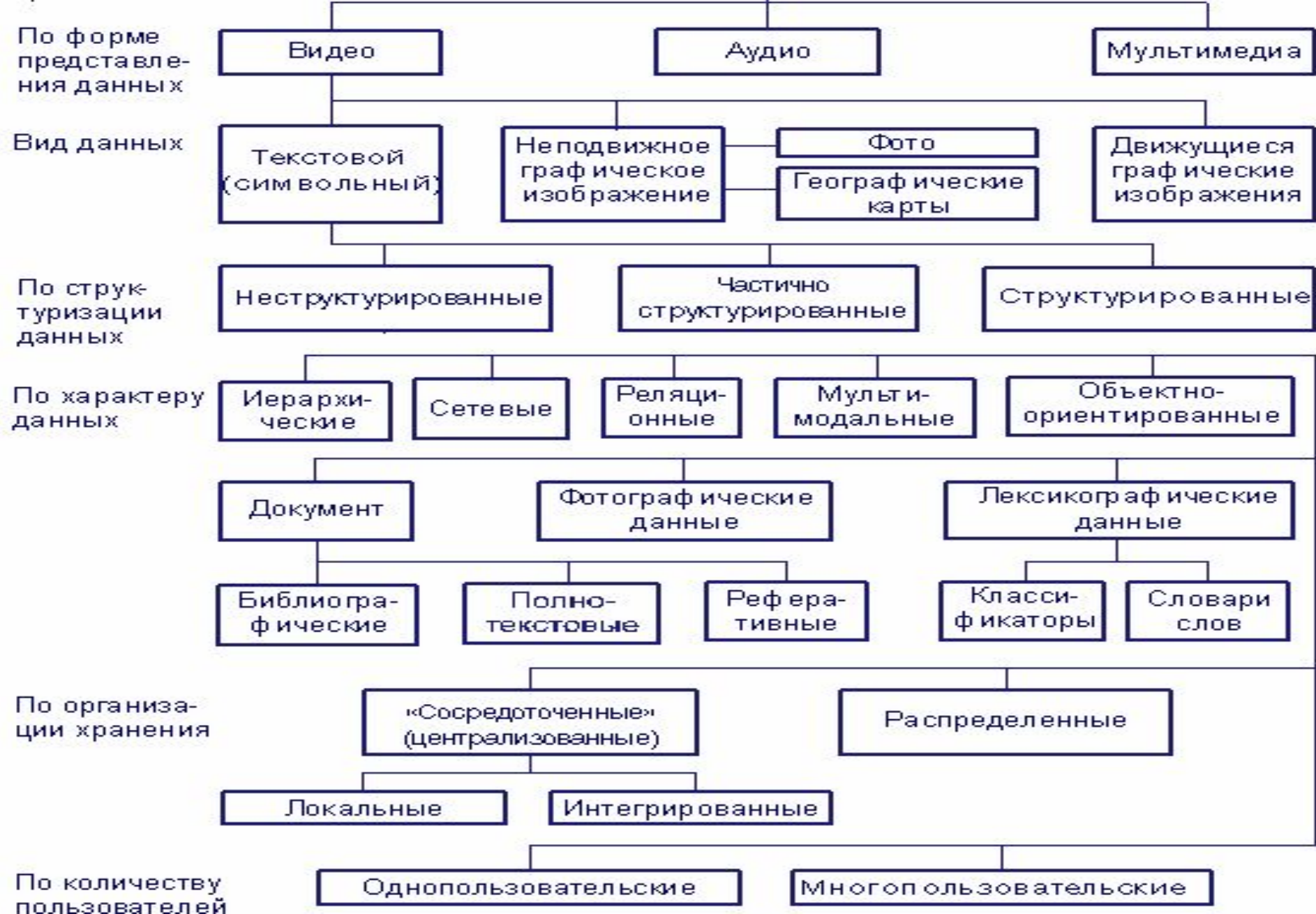
- централизованные (локальные, файл – серверные, клиент-серверные) и распределённые СУБД.

По сфере применения

- справочные системы и системы обработки данных.

Классификация по масштабу систем:

- персональные; уровня группы, отдела, предприятия; корпоративные; географически распределенные.



Классификация БД по основным признакам

Классификация СУБД

По степени универсальности различают два класса СУБД:

- **системы общего назначения (СУБД ОН) ;**
- **специализированные системы (СпСУБД).**

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. Им присущи развитые функциональные возможности, функциональная избыточность.

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

Классификация СУБД

По технологии обработки

Централизованные

(хранятся в памяти одной вычислительной системы, возможен распределенный доступ к данным)



Распределенные

(состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети)



Классификация СУБД

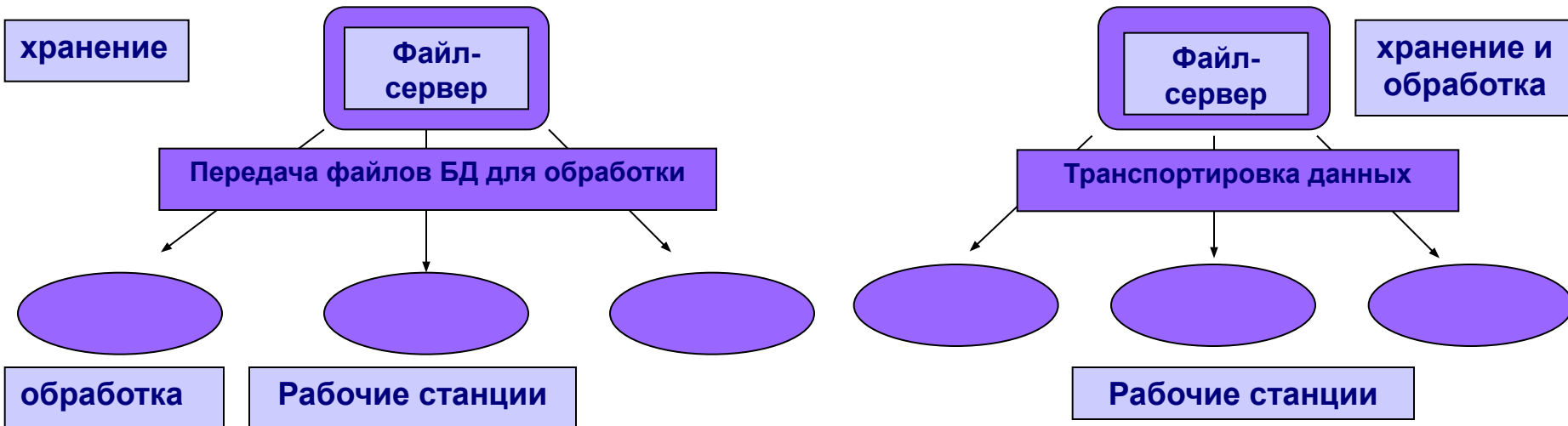
По способу доступа к данным

с локальным доступом

с удаленным (сетевым) доступом

Архитектура файл-сервер

Архитектура клиент-сервер



Классификация СУБД

Архитектура систем БД с сетевым доступом (**файл-сервер**) предполагает выделение одной из машин сети в качестве центральной (сервер файлов).

На ней хранится совместно используемая централизованная БД. Все другие машины сети являются рабочими станциями.

Файлы БД в соответствии с пользовательскими запросами передаются на рабочие станции, где и производится обработка.

При большой интенсивности доступа к одним и тем же данным производительность системы падает.

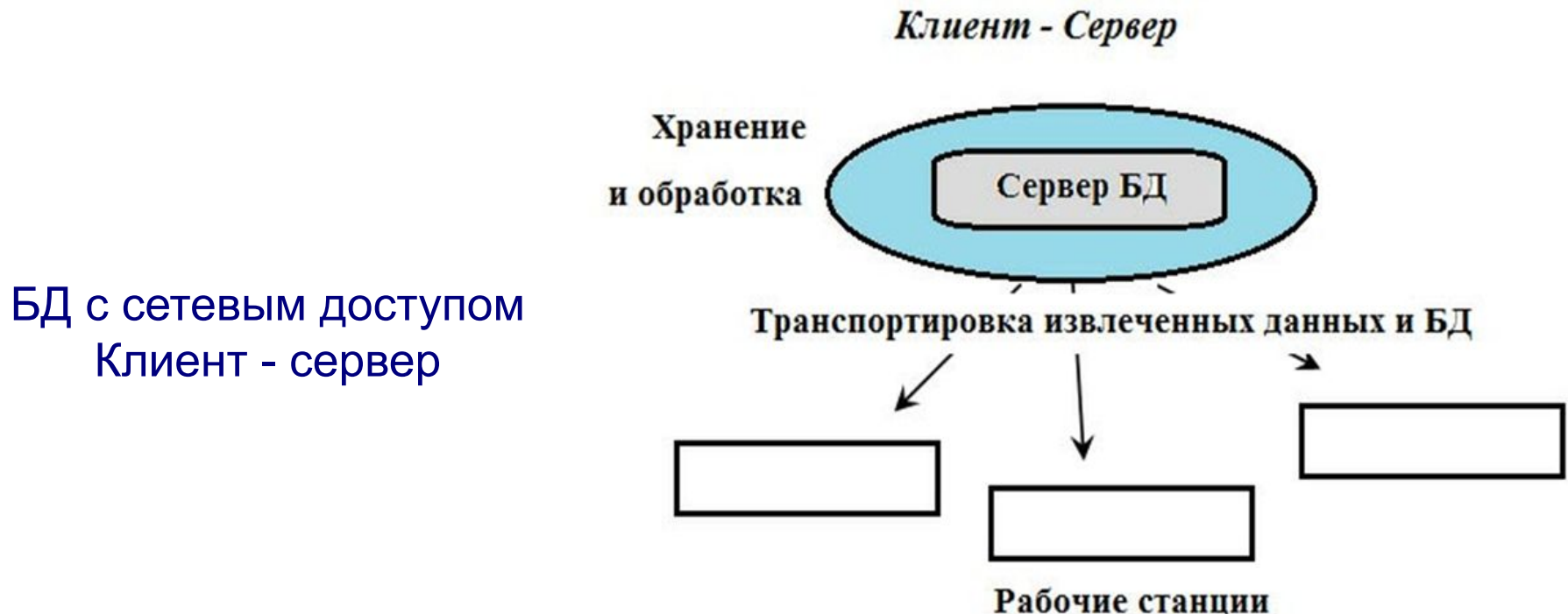
БД с сетевым доступом
(Файл-сервер)



Классификация СУБД

В архитектуре **Клиент-сервер** - помимо хранения централизованной БД центральная машина (**сервер базы данных**) должна обеспечивать выполнение основного объёма обработки данных.

Запрос на данные клиента, порождает поиск и извлечение данных на сервере. Извлечённые данные (но не файлы) транспортируются по сети от сервера к клиенту.

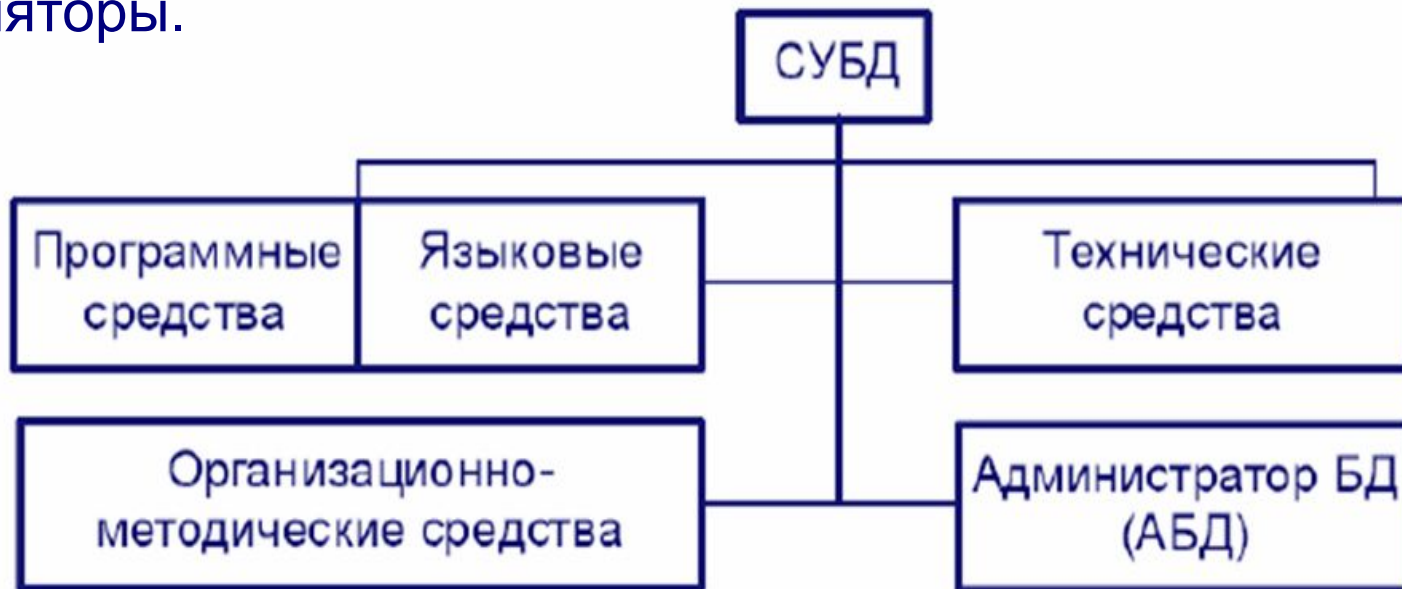


Состав СУБД

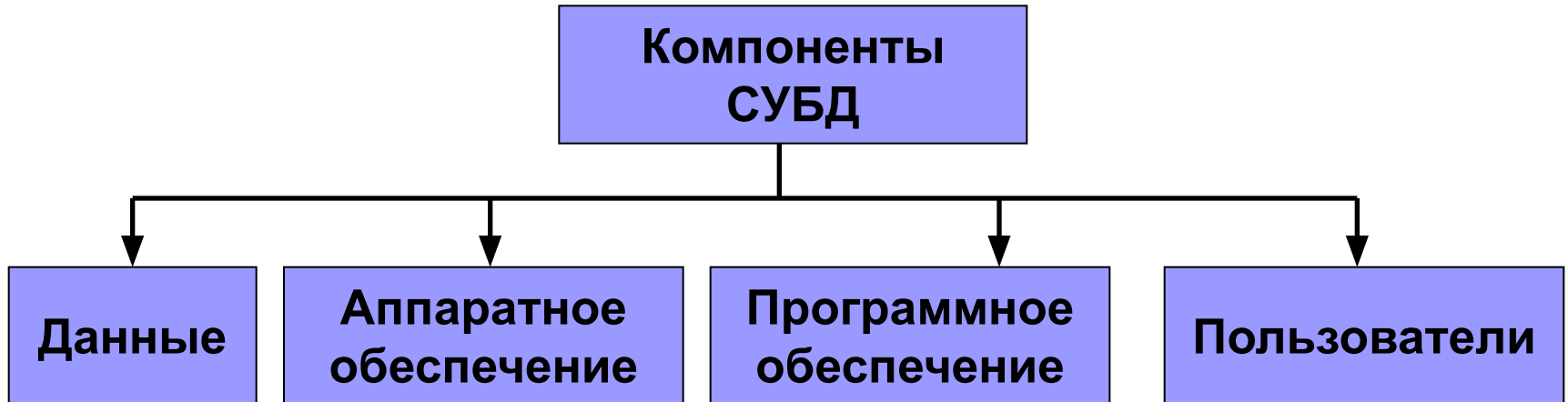
СУБД представляет собой оболочку, с помощью которой при организации структуры таблиц и заполнения их данными получается та или иная база данных.

В связи с этим в составе СУБД различают систему *программно-технических, организационных и "человеческих" составляющих.*

Программные средства включают систему управления, обеспечивающую ввод-вывод, обработку и хранение информации, создание, модификацию и тестирование БД, трансляторы.



Основные компоненты СУБД и их состав

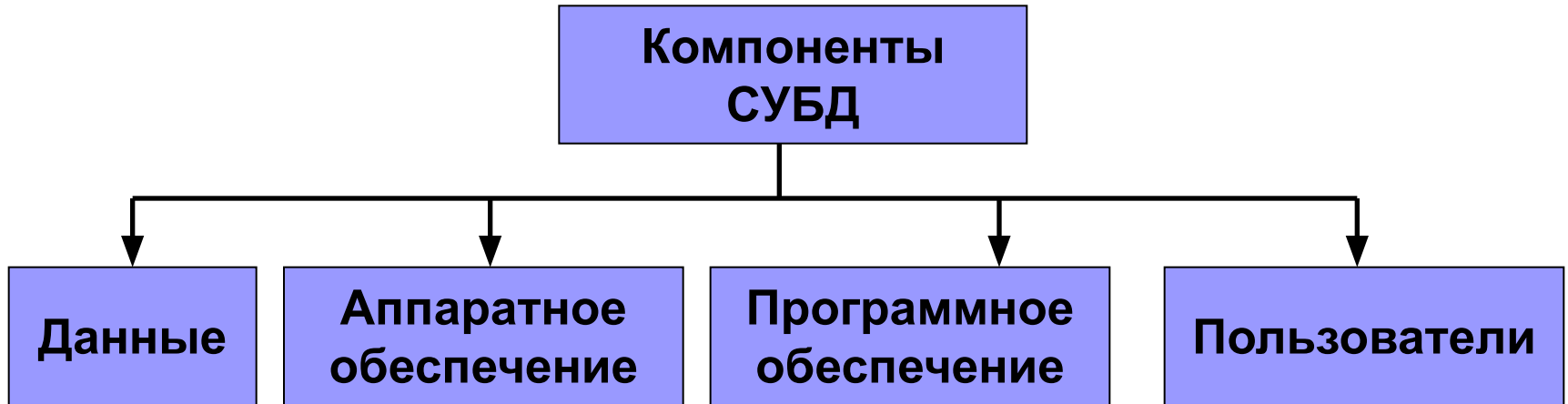


Данные должны быть интегрированными и общими.

Интегрирование – возможность представлять базу данных как объединение нескольких отдельных файлов данных полностью или частично не перекрывающихся.

Общие – возможность использования отдельных областей данных в БД несколькими различными пользователями, причем даже в одно и тоже время (одновременный доступ).

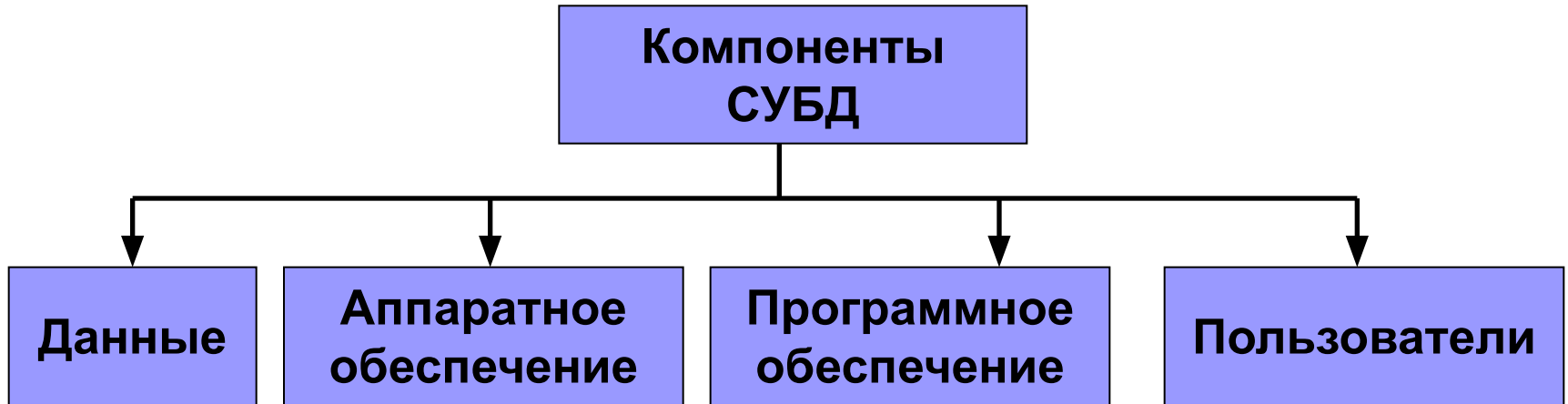
Основные компоненты СУБД и их состав



Накопители для хранения информации (обычно диски с перемещаемыми головками) вместе с подсоединенными устройствами ввода-вывода, контроллерами устройств, каналами ввода-вывода и т.д.

Процессор или процессоры вместе с основной памятью, которая используется для поддержки работы программного обеспечения системы

Основные компоненты СУБД и их состав

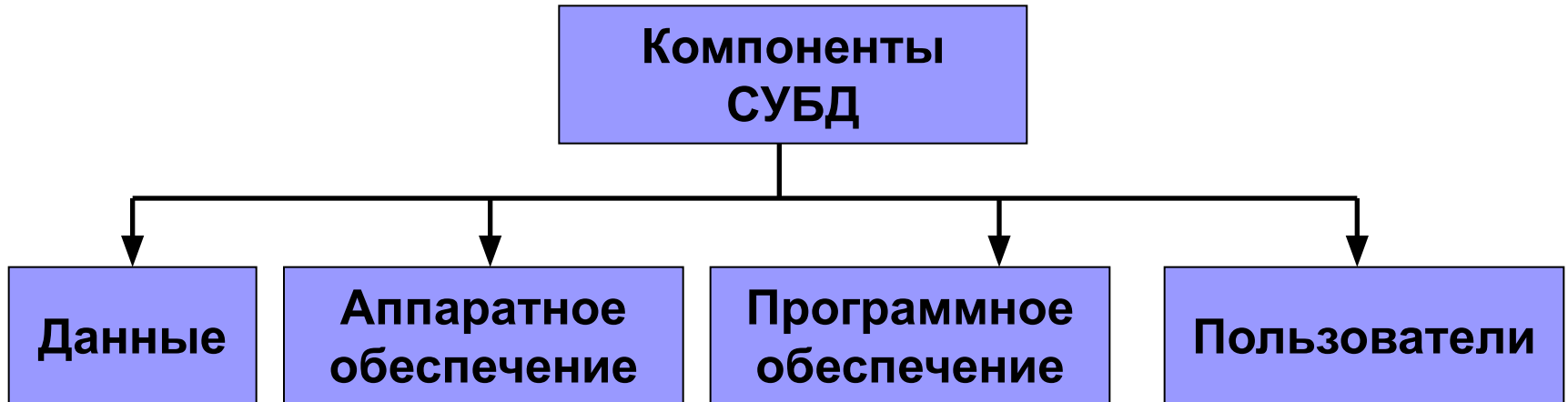


Диспетчер базы данных (database manager), или система управления базами данных СУБД (database management system (DBMS)).

СУБД предоставляет пользователю возможность рассматривать БД как объект более высокого уровня по сравнению с аппаратным обеспечением, а также поддерживает выражаемые в терминах высокого уровня пользовательские запросы (SQL).

Кроме СУБД, в программном обеспечении – утилиты, средства разработки приложений, средства проектирования, генераторы отчетов и другие.

Основные компоненты СУБД и их состав



Работающие с базами данных пользователи обладают различными знаниями, навыками и сталкиваются с решением различных задач:

- конечные пользователи;*
- разработчики баз данных;*
- разработчики приложений;*
- администраторы баз данных.*

Основные компоненты СУБД и их состав

Базовыми внутренними языками программирования являются языки четвертого поколения. В качестве базовых языков могут использоваться **C, C++, Pascal, Object Pascal**.

Исторически для системы управления базой данных сложились *три языка*:

1. **Язык описания данных (ЯОД)**, называемый также языком описания схем, - для построения структуры ("шапки") таблиц БД;

2. **Язык манипулирования данными (ЯМД)** - для заполнения БД данными и операций обновления (*запись, удаление, модификация*);

3. **Язык запросов** - язык поиска наборов величин в файле в соответствии с заданной совокупностью критериев поиска и выдачи затребованных данных без изменения содержимого файлов и БД (язык преобразования критериев в систему команд).

В настоящее время функции всех трех языков выполняет **язык SQL**, относящийся к классу языков, базирующихся на исчислении кортежей (кортеж - единица информации), языки СУБД FoxPro, Visual Basic for Application (СУБД Access) и т.д.

Структурные элементы БД

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, определяется 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.

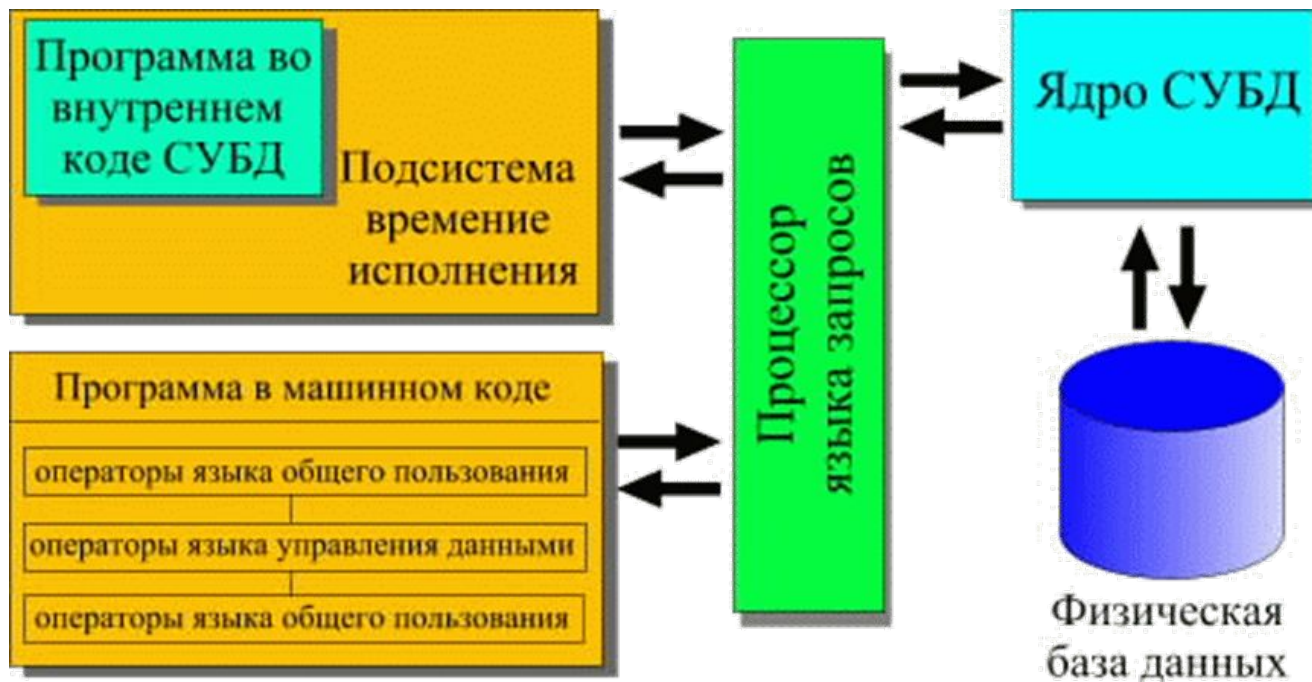
Таблица (файл) – совокупность экземпляров записей одной структуры.

Модель данных

Ядром любой базы данных является **модель данных**.

С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

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



Определение понятия «модель»

Модель - это материальный или мысленно представляемый объект, который в процессе познания (созерцания, анализа и синтеза) замещает объект-оригинал.

Модель — это упрощенное представление реального устройства, процесса, явления.

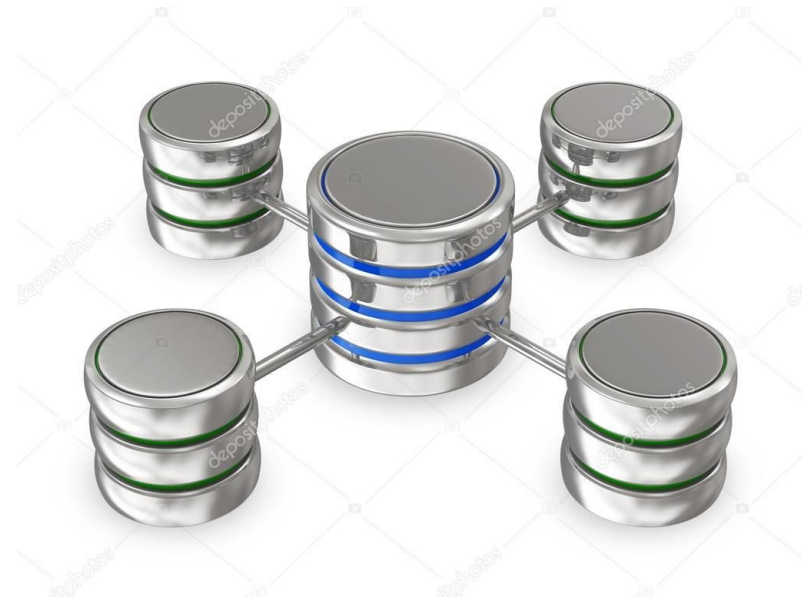
*Процесс построения и исследования моделей называется **моделированием**, облегчает изучение имеющихся в реальном устройстве (процессе, явлении) свойств и закономерностей.*



Определение понятия «модель данных»

Модель данных — это некоторая интерпретация данных, связанная с этапом проектирования БД, которая трактуется как сведения, имеющие определенную структуру.

Модель данных – это логическое определение объектов, связанное с этапом проектирования БД.



Модели данных

Инфологические модели

Даталогические модели

Физические модели

Диаграммы Бахмана

Модель «сущность-связь» (ER-модель)

Основанные на файловых структурах

Основанные на странично-сегментной организации

Документальные

Фактографические

Ориентированные на формат документа

Дескрипторные

Тезаурусные

Теоретико-графовые

Иерархическая

Сетевая

Теоретико-множественные

Реляционная

Бинарных ассоциаций

Объектно-ориентированные

Модели данных

Инфологические модели

Модель «сущность-связь» (ER-модель)

Даталогические модели

Иерархическая

Сетевая

Реляционная

Объектно-ориентированная

Физические модели

Инфологическое моделирование связано со 2-м этапом проектирования БД: созданием формализованного описания предметной области

Логическое (или даталогическое) моделирование осуществляется после этапа выбора СУБД. Этот тип модели полностью зависит от типа модели, поддерживаемой выбранной системой.

Физическое моделирование заключается в выборе эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы.

Предметная область

(часть реального мира отображаемая в базе данных)



ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Формализованное, обобщенное, не привязанное к каким-либо СУБД описание предметной области



ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Описание на языке конкретной СУБД



ФИЗИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Описание хранимых данных на уровне операционной системы



БАЗА ДАННЫХ

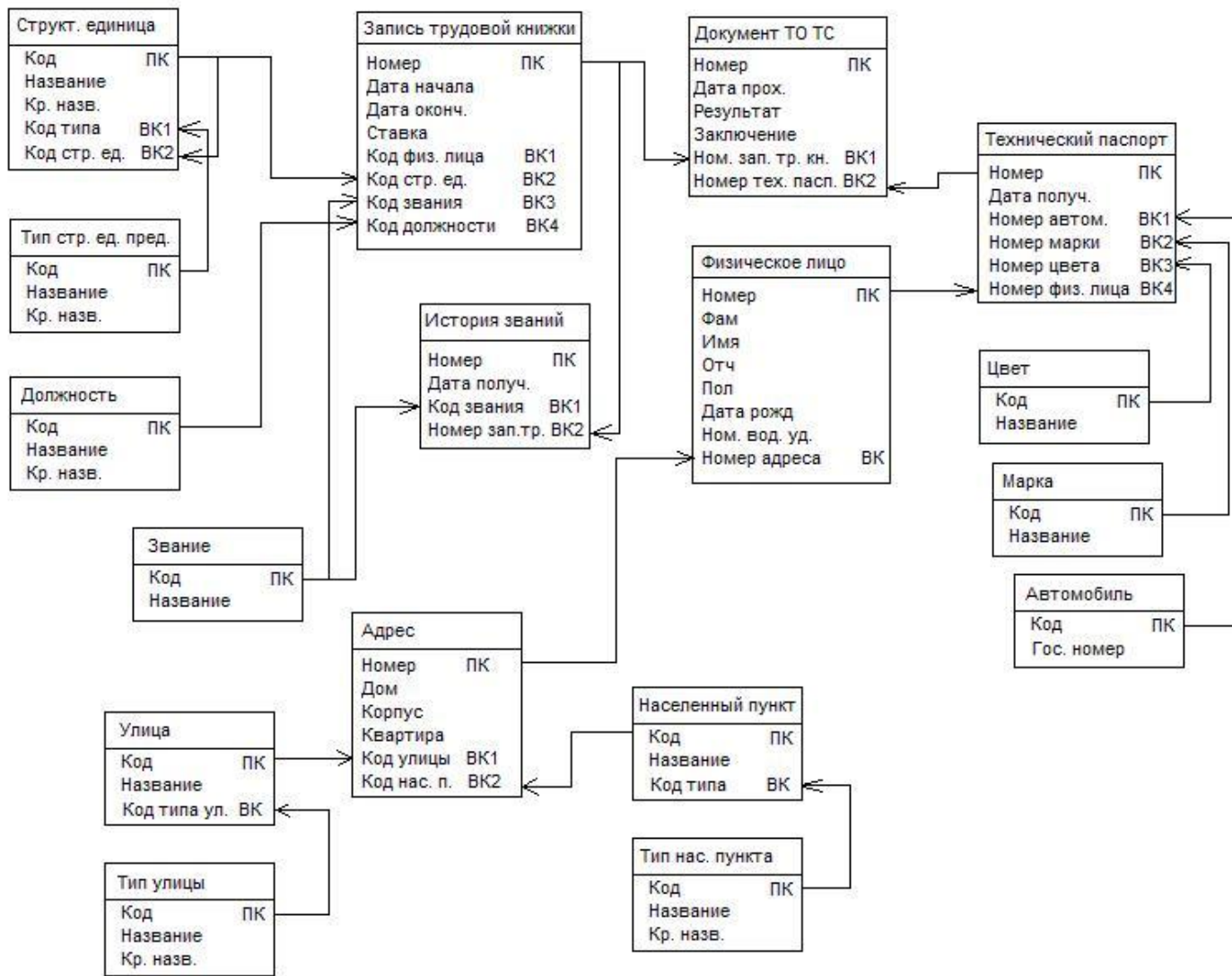
Модели описания, используемые СУБД

Виды моделей данных

Сетевая модель данных

В Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа.





Реляционная модель данных

Основные термины реляционной базы данных:

Отношение – информация об объектах одного типа. Отношение обычно хранится в виде *таблицы*.
Свойства таблиц.

Атрибут – определенная часть информации о некотором объекте. Атрибут обычно хранится в виде *столбца* или *поля таблицы*.

Связь – способ, которым связана информация в одной таблице с информацией в другой таблице.
Типы связей

Объединение – процесс объединения таблиц или запросов на основе совпадающих значений определенных атрибутов.



Свойства реляционных таблиц

- каждый элемент таблицы - один элемент данных;
- все записи в столбцах таблицы 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.



Реляционная база данных

Все объекты в реляционной базе данных связаны между собой.

Различают связи нескольких типов:

Один к одному (1:1)

любая запись в первой таблице связана только с одной записью во второй таблице и наоборот (создается между ключевыми полями таблиц)

Один ко многим (1:M)

любая запись в первой таблице может быть связана с несколькими записями во второй, но в то же время любая запись второй таблицы связана только с одной записью первой

Многие ко многим (M:M)

многие записи одной таблицы соответствуют многим записям из другой таблицы

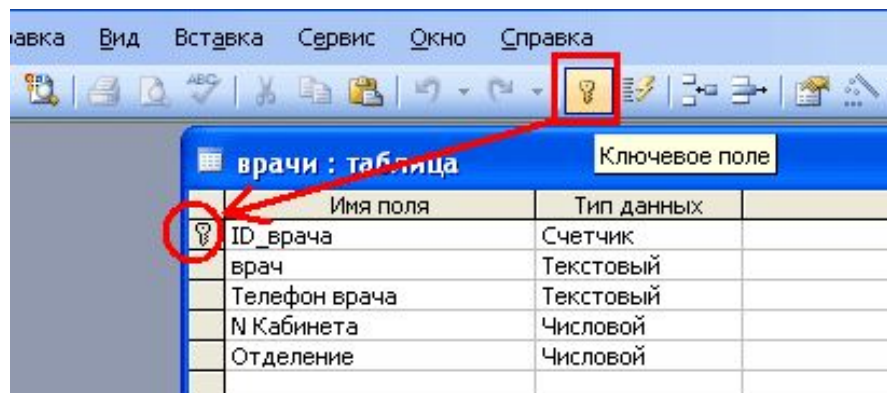


Понятие ключевого поля

Поле, каждое значение которого однозначно определяет соответствующую запись, называется **простым ключом (ключевым полем)**.

Если записи однозначно определяются значениями нескольких полей, то такая таблица БД имеет **составной ключ**.

Чтобы связать две реляционные таблицы, необходимо **ключ первой таблицы ввести в состав ключа второй таблицы (связь 1:1)**.



Имя поля	Тип данных
ID_врача	Счетчик
врач	Текстовый
Телефон врача	Текстовый
N Кабинета	Числовой
Отделение	Числовой



Основные функции СУБД

К числу *функций СУБД* принято относить следующие:

1. Непосредственное управление данными во внешней памяти

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей.

2. Управление буферами оперативной памяти

СУБД обычно работают с БД значительного размера. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти.

3. Управление транзакциями

Транзакция - это последовательность операций над БД, рассматриваемых СУБД как единое целое.

Основные функции СУБД

4. Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под *надежностью хранения* понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя.

Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью, в которую поступают записи обо всех изменениях основной части БД.

5. Поддержка языков БД

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных.

Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык запросов **SQL** (Structured Query Language).

Функциональные возможности СУБД

Любая СУБД позволяет выполнить четыре *простейшие операции с данными*:

Добавлять в таблицу одну или несколько записей;

Удалять из таблицы одну или несколько записей;

Обновлять значения некоторых полей в одной или нескольких записях;

Находить одну или несколько записей, удовлетворяющих заданному условию.

И еще одна функция СУБД - это **управление данными**.

Под **управлением данными** обычно понимают защиту данных от несанкционированного доступа, поддержку многопользовательского режима работы с данными и обеспечение целостности и согласованности данных.

Функциональные возможности СУБД

Производительность СУБД оценивается:

- *временем выполнения запросов;*
- *скоростью поиска информации;*
- *временем выполнения операций импортирования данных из других форматов;*
 - *скоростью выполнения таких операций как обновления, вставка, удаление данных;*
 - *максимальным числом параллельных обращений к данным в многопользовательском режиме;*
 - *временем генерации отчёта.*

На производительность СУБД оказывают влияния 2 фактора:

- *правильное проектирование;*
- *технология построения БД.*

Функциональные возможности СУБД

Обеспечение целостности и согласованности данных подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной.

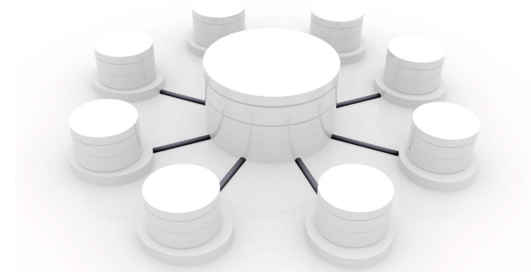
Эти средства не дают выполнять такие изменения, после которых данные могут оказаться несогласованными.

Например, когда две таблицы связаны отношением «один-ко-многим», нельзя внести запись в таблицу на стороне «многие», если в таблице на стороне «один» отсутствует соответствующая запись.

Функциональные возможности СУБД

СУБД предусматривают следующие *средства обеспечения безопасности данных*:

- *шифрование прикладных программ;*
- *шифрование данных;*
- *защиту паролем;*
- *ограничение уровня доступа.*

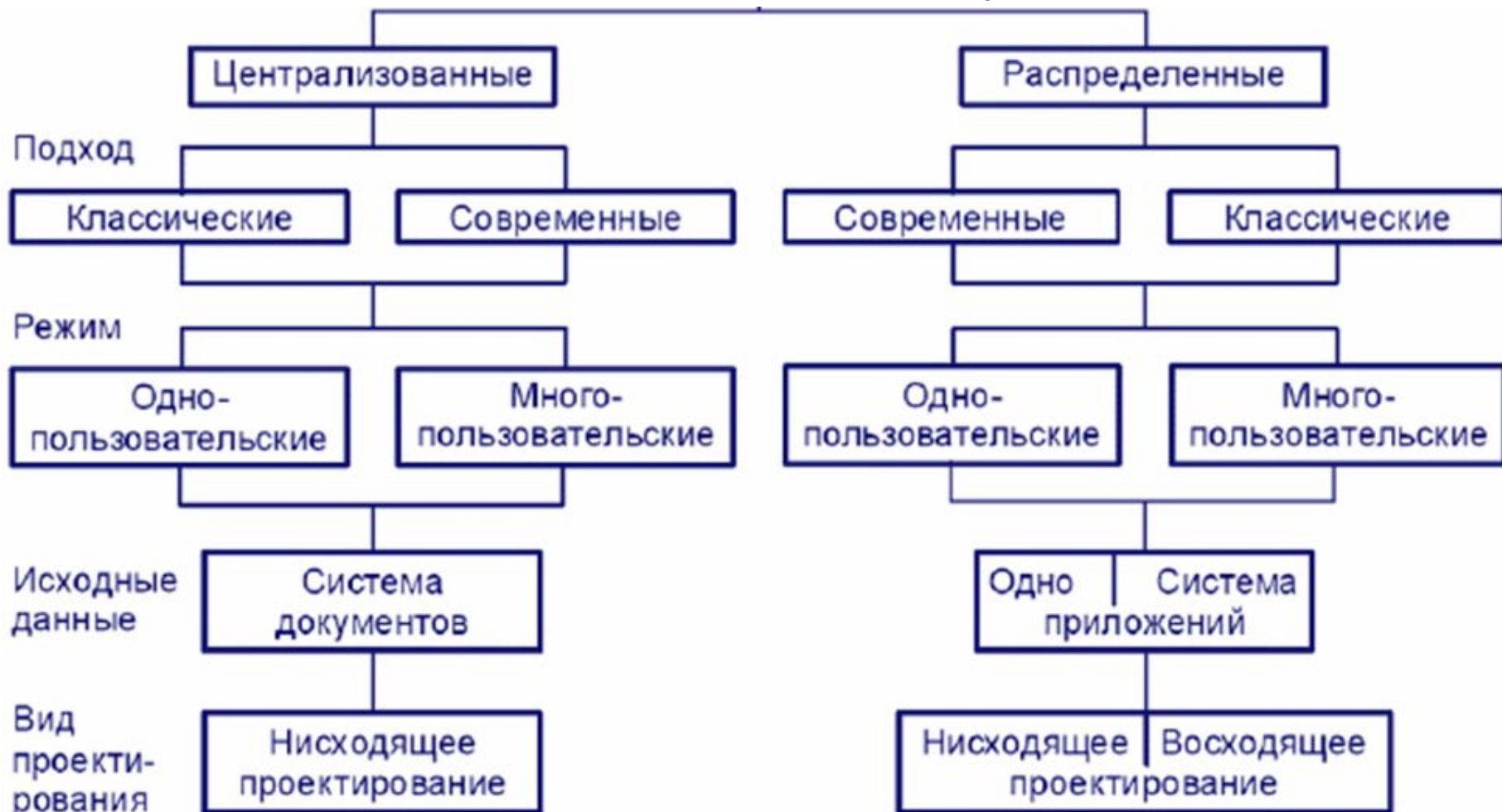


Практически все СУБД предназначены для работы в *многопользовательских средах*, что предполагает выполнение программным продуктом следующих функций:

- *блокировку базы данных, файла, записи, поля;*
- *идентификацию станции, установившей блокировку;*
- *обновление информации после модификации;*
- *контроль за временем и повторение обращения*
- *работу с сетевыми системами.*

Подходы к проектированию БД

Последнее десятилетие характеризуется появлением *распределенных и объектно-ориентированных баз данных*, характеристики которых определяются приложениями средств автоматизации проектирования и интеллектуализации БД.

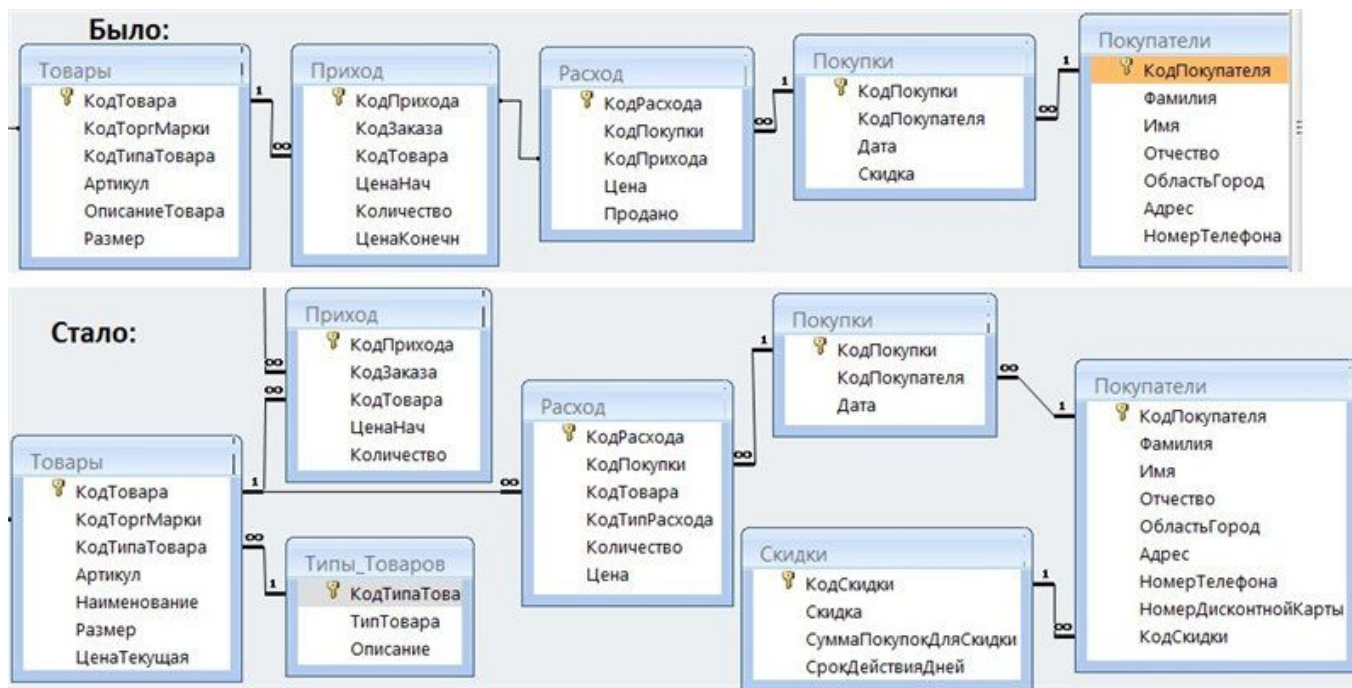


Подходы к проектированию БД

Используют **восходящее и нисходящее проектирование БД**.

Восходящее проектирование БД применяют в распределенных БД при интеграции спроектированных локальных баз данных, которые могут быть выполнены с использованием различных моделей данных.

Нисходящее проектирование более характерно для разработки централизованных БД.



Основные этапы разработки БД



Этап 1. Уточнение задач

Этап 2. Последовательность выполнения задач

Этап 3. Анализ данных

Этап 4. Определение структуры данных

Этап 5. Разработка макета приложения и пользовательского интерфейса

Этап 6. Создание приложения

Этап 7. Тестирование и усовершенствование



Спасибо за внимание!