**СУБД**

**Лекция №1**

**Раздел 1: Общая теория реляционных баз данных**

**Тема: БД, СУБД, типы БД**

БД – совокупность данных организованных по законам, принципам, правилам, предусматривающее хранение инф. обработку данных, и выдачу необходимых сведений.

СУБД – комплекс программных систем позвол-й обрабатывать данные, и выдавать пользователю.

Существуют распределенные БД и централизованные БД.

Распределенные БД хранится в разных БД в которых нет связи между БД.

Централизованные БД хранится в одном комп. другие подкл. к этому компьютеру.

*Технологии обработки данных*

Технологии обработки данных:

1. Файл-сервер (толстый клиент) предусматривает что сервер хранит данные, а все операции по обработке данных выполняет клиент
2. Клиент- сервер (тонкий клиент) предусматривает то сервер хранит данные и обрабатывает данные

При ФС клиент обращается к серверу, сервер выдает ему данные, клиент обрабатывает их и результат перенаправляет ее к серверу, сервер записывает вместо старых.

Для КС характерно, что клиент выполняет запрос серверу, сервер обрабатывает данные, клиент видит результат.

|  |  |
| --- | --- |
| ФС | КС |
| Высокие требования к железу клиента | Высокое требование к железу клиента |
|  | Безопасность данных при передаче данных |
|  |  |
|  |  |

По таблице видно подавляющее превосходство тонкого клиента, поэтому во многих инф сервисах необходимо использовать технологию клиент сервера. Толстый клиент используется тогда, когда решен проблемы:

Создана надежная сеть, обеспечивающая хорошую скорость, защиту, надежность;

Клиент стоит на комп. С высокими достаточно высокой производительностью, а также данными может редактировать только один клиент

Тема: Модели БД

Сущ-ет 4 модели БД:

1. Иерархическая
2. Сетевая
3. Объектно-ориентированная
4. Реляционная

**Иерархическая модель** предполагает строгое одних данных на других, то есть одни не могут сущ-ть без других, данные выстраиваются в виде дерева, где сущ-ют предки и потомки. У потомка только один предок, у одного предка несколько потомков. Связь между потомками осущ-ся через их общего предка.

**Сетевая модель** яв-ся расширением иерархической в отличие от между 2 данными может сущ-ть несколько связей в каждой из которых иерархия определяется по-разному.

ОО модель предполагают работу с объектами: электр.таблица, фото, файл и т.д.

**Реляционная** наиболее гибкая модель. Данные хранятся в таблицах, между которыми есть отношения. Отсутствует всякая иерархия.

Рассмотрим пример иерархической модели состоящая из отделов, рассмотрим схему отдела. Это тип дерева Отдел, которые имеют предка Отдел потомков Начальник и Сотрудники. Есть один тип предка и несколько типов потомков. Обход иерархии осущ-ся в виде сверху-вниз-слева-направо. В нашем случае Отдел яв-ся предком Начальник старший потомок Сотрудник младший потомок. Сначала создается Отдел, назначается начальник, начальник собирает сотрудников. Каждое дерево внутри иерархической модели представляет тип дерева. Внутри типа дерева имеется тип предка и наследника. Если рассматривать конкретное дерево, то вводится понятие экземпляр типа дерева. Например: конкретный отдел - «Отдел 310» внутри дерева он экземпляр типа предка, Иванов экземпляр типа потомка Начальник.

Виды манипулирования данных в иерархической модели:

………

Рассмотрим сетевую модель в сетевой модели рассматривается каждая связь между объектами, каждый раз иерархия определяется отдельно. Связь идет от предка к потомку. без предка не может быть связи

В сетевой модели существуют следующие виды манипулировании данных:

………..

**Тема: Модель «Сущность связь»**

Сущностью (Entity) – называется любой объект реальной действительности, которую мы изучающий для создания БД. Связь – некоторая ассоциация между сущностями.

ER-model – это модель сущность-связь. Каждая сущность представ. со своими атрибутами. Несколько сущностей, объединённых общим свойством наз-ся набором сущностей (Entity Set). Связь имеет свой тип и кардинальность.

OneToOne – В этой связи одной сущности слева может соответствовать не более одного справа.

Схема модели: сущности в прямоугольники, связи в ромбик. Схематически изображаем тип и кардинальность. Тип отвечает на вопрос – сколько?, кардинальность – обязательно ли?.

Для определения типа и кардинальности отвечаем на 4 вопроса:

1. Для одной сущности слева в этой связи сколько сущностей справа соответствует? Один студент яв-ся старостой в скольких группах? В одной (1, ?: ?, ?)
2. Для одной сущности справа в этой связи сколько сущностей слева соответствует?

В одной группе яв-ся старостой сколько студентов? Один (1, 1: ?, ?)

1. Обязательно ли сущность слева должна быть в связи сущностью справа?

Обязательно ли студент яв-ся старостой группы? Нет (1, 1: 0, ?). Группа имеет кардинальность 0.

1. Обязательно ли сущность справа должна быть в связи сущностью слева?

Обязательно ли в группе яв-ся старостой студент? Да (1, 1: 0, 1). Студент кардинальность 1.

ManyToOne

Студент учится в группе

(1, n: 1, 1)

OneToMany

Группа состоит из студентов

(1, 1: 1, n)

ManyToMany

Студент участвует во многих сорев.

(1, n: 0, n)

Студент имеет ИНН

Студент участвует в олимпиаде