

# **Понятие баз данных и их моделей.**

# План:

- Понятие баз данных (БД)
- Классификация баз данных
- Классификация БД по моделям данных
  - Иерархическая модель
  - Сетевая модель
  - Реляционная модель
- Элементы табличной (реляционной) структуры данных.

**База данных (БД) — это организованная структура, предназначенная для хранения информации, обновления и использования.**

**СУБД** – система **у**правления  
**б**азами **д**анных – комплекс  
языковых и программных  
средств, предназначенных  
для создания, ведения и  
использования БД многими  
пользователями

По способу доступа к данным БД делят на:

- БД локального доступа;
- БД удаленного доступа.

**Локальный доступ.** СУБД обрабатывает базу данных, которая хранится на той же вычислительной системе;

**Удаленный доступ.** Обращение к БД, которая хранится на одной из систем. Удаленный доступ может быть выполнен по принципу **файл-сервер** или **клиент-сервер**.

# Классификация БД по моделям данных

Ядром любой БД является  
***модель представления  
данных.***

Известны три основные модели организации данных и связей между ними:

- 1. Иерархическая (в виде дерева) модель***
- 2. Сетевая модель***
- 3. Реляционная модель***



# Иерархическая модель

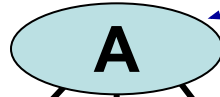
позволяет строить базы данных с древовидной структурой.

## К основным понятиям иерархической модели относятся:

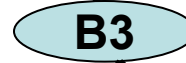
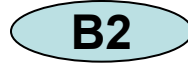
- **узел** – набор атрибутов данных, описывающих объект;
- **связь** – линия, связывающая узлы нижнего уровня с одним узлом вышележащего уровня.
- **Уровень** – номер слоя узлов, отсчитанный от корня.

Главный элемент, узел («корень»)

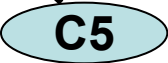
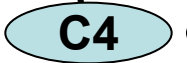
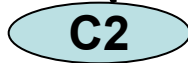
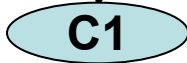
Уровень 1



Уровень 2



Уровень 3



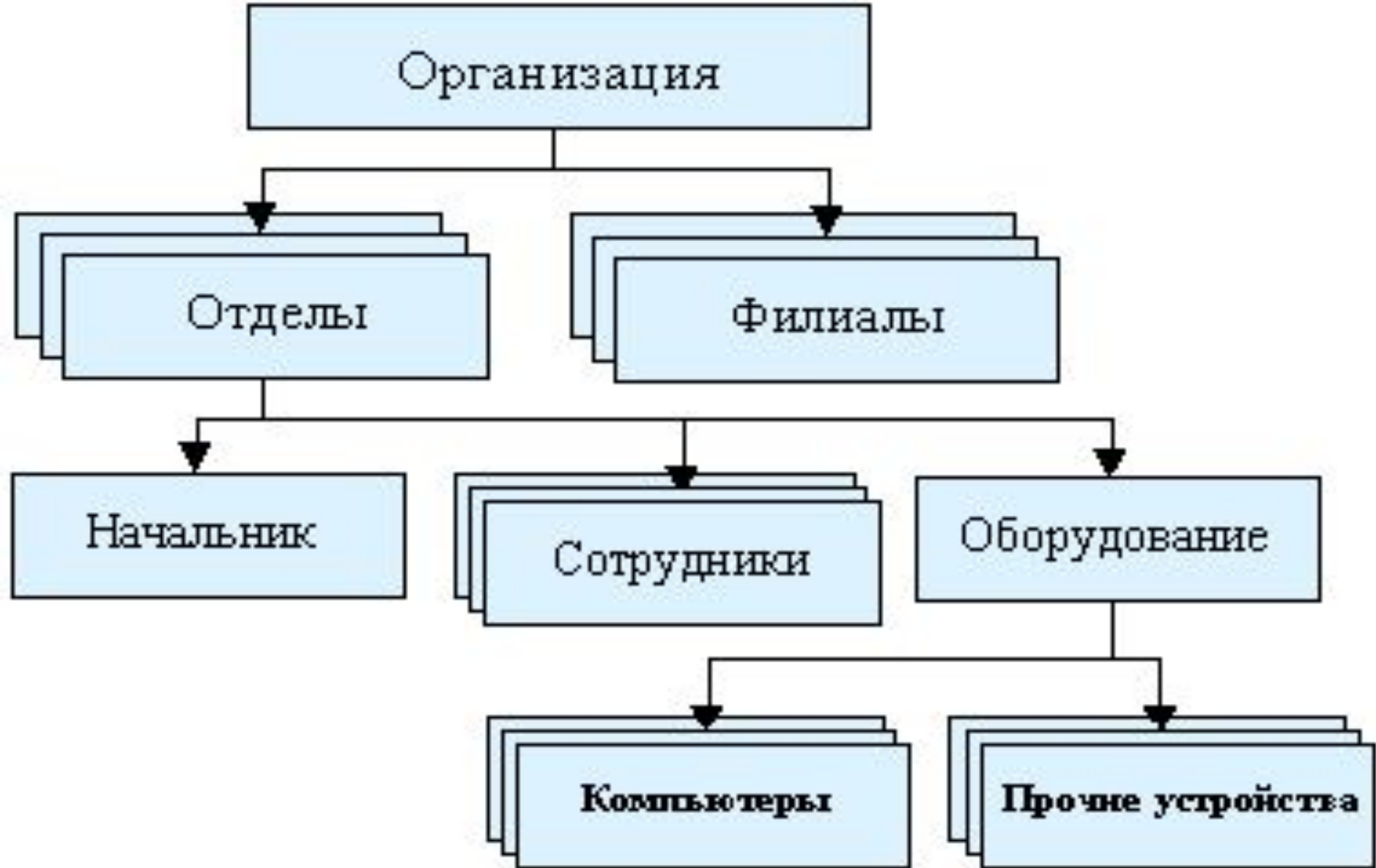
**Рис.1 Иерархическая модель данных**

Узел B1 – предок для узлов C1 и C2, а узлы C1, C2 – потомки узла B1

Каждый узел может иметь только одного предка и множество «потомков», т. е. вершин на нижележащем уровне. Такой принцип связи называют

**"ОДИН-КО-МНОГИМ".**

Поиск данных в иерархической системе всегда начинается с корня. Затем производится спуск с одного уровня на другой пока не будет достигнут искомый уровень. Перемещения по системе от одной записи к другой осуществляются **с помощью ссылок.**



*Рис.2 Иерархическая древовидная структура модели **БД***

- Примеры иерархических структур в жизни

# Сетевая модель

Имеет те же составляющие, что и иерархическая, но каждый узел может быть связан с любым другим узлом.

В сетевых структурах потомок может иметь любое число предков.



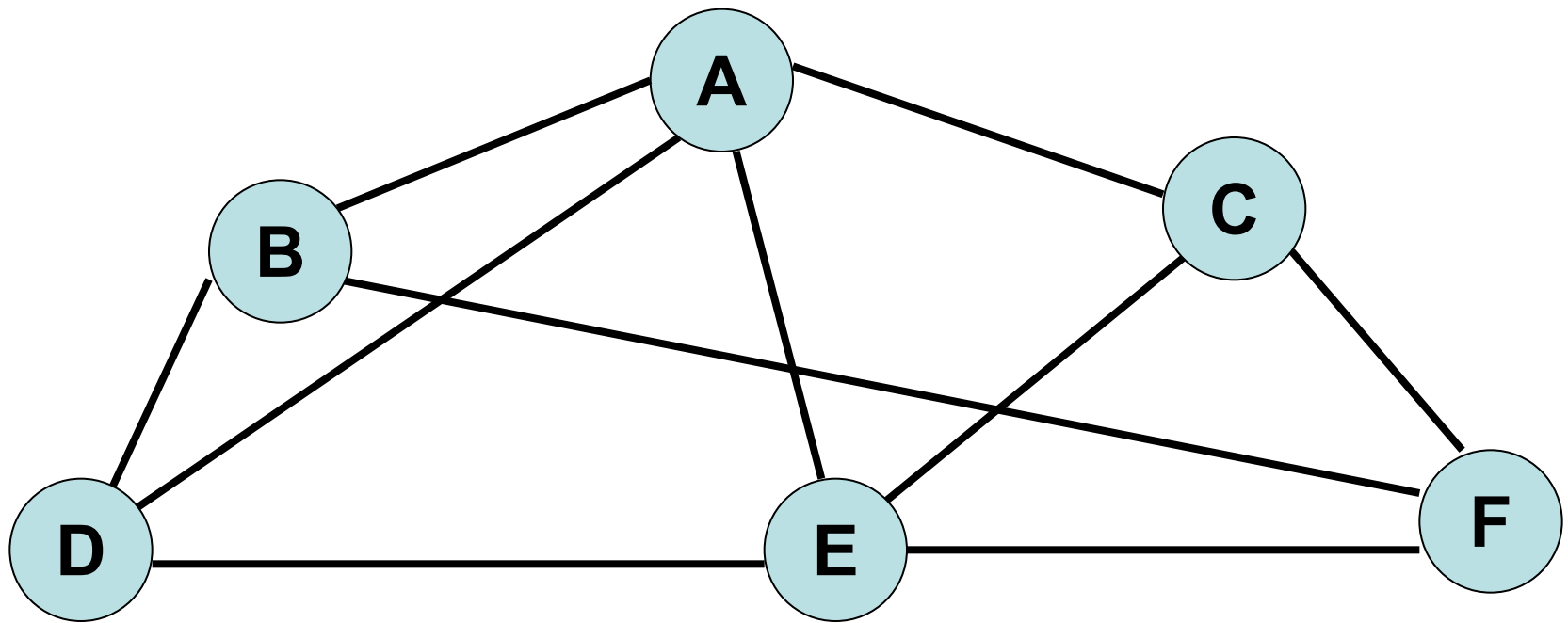
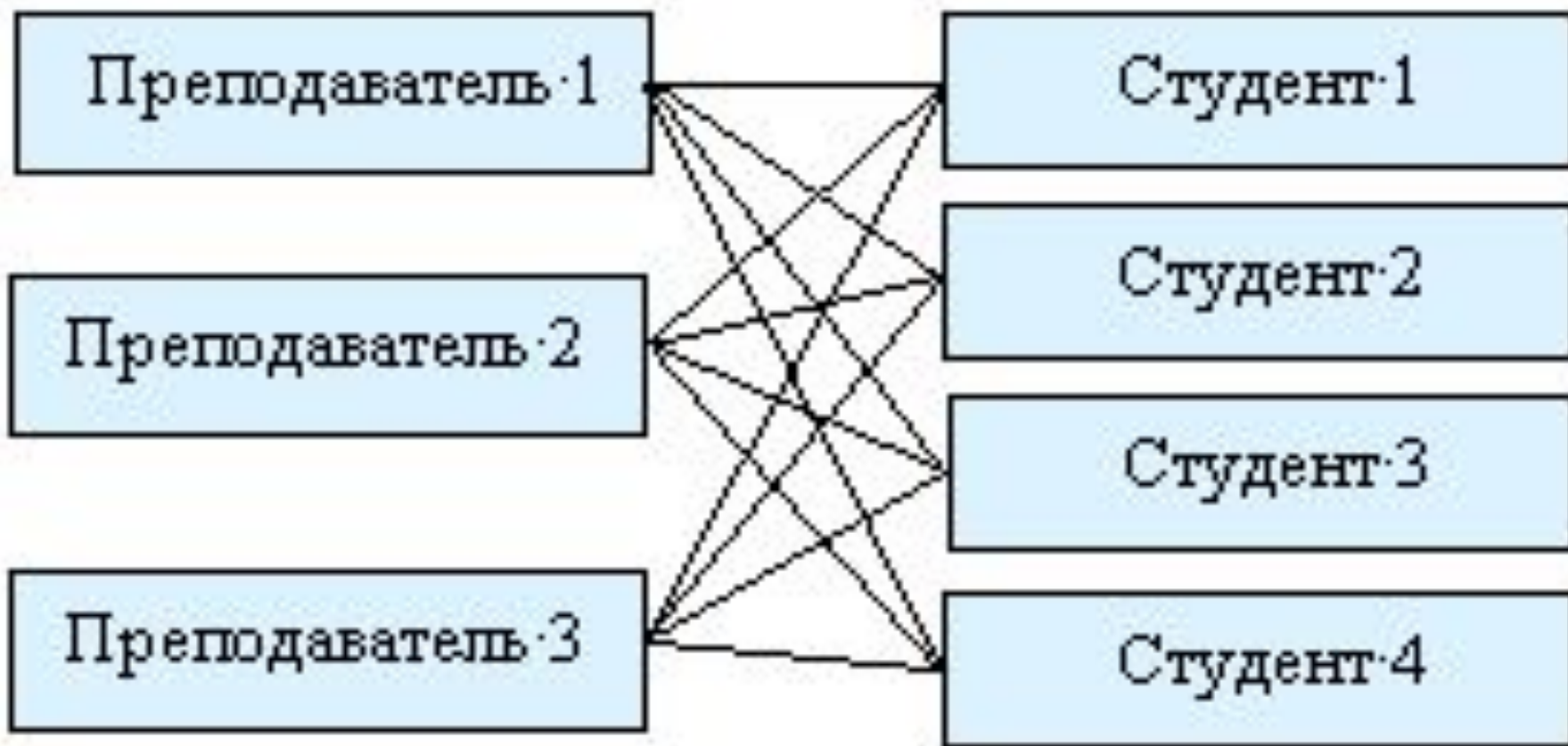


Рис.3 Сетевая модель данных

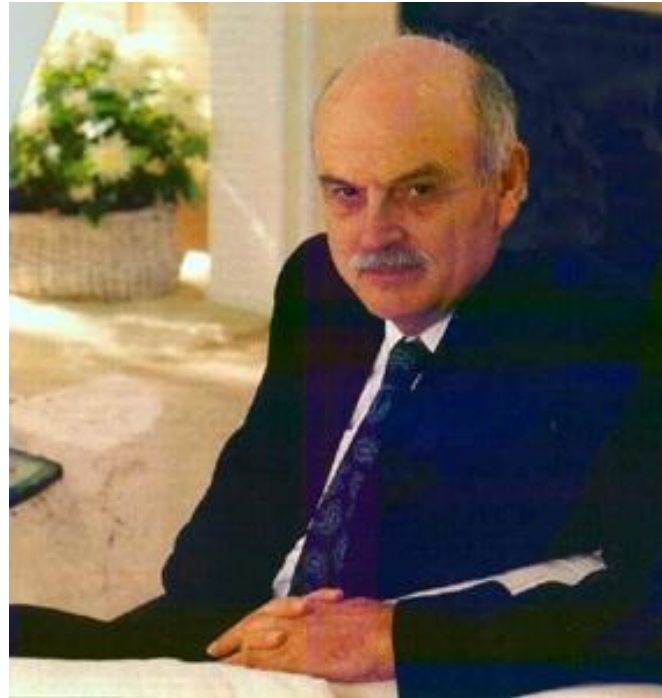
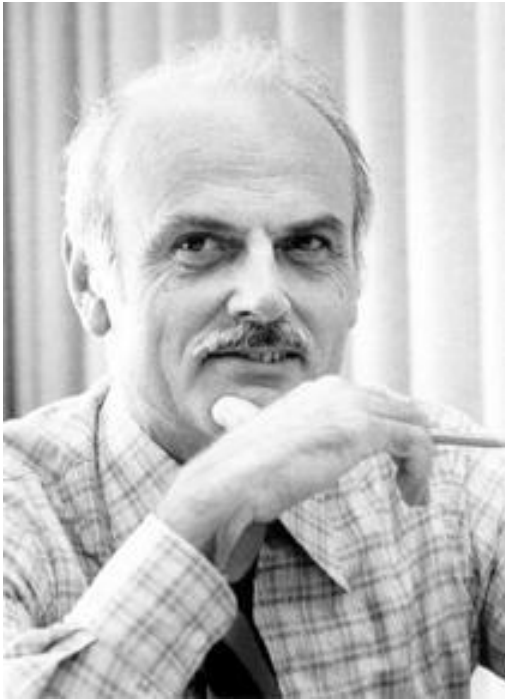
Реализована связь «**многие-ко-многим**»



***Рис.4*** ***Сетевая структура модели БД***  
**"МНОГИЕ-КО-МНОГИМ"**

# Реляционная модель

Реляционными базами данных (БД) (от англ. слова *relation* — "отношение") называются БД, содержащие информацию, организованную в виде двумерных таблиц.



**Edgar Frank Codd (Эдгар Френк Кодд)  
[23.08.1923 - 18.04.2003]**

создал и описал концепцию реляционных баз данных и реляционную алгебру [1970г.]; для проектирования БД предложил аппарат нормализации отношений

Основная идея реляционной модели заключается в следующем. БД состоит из ряда неупорядоченных таблиц. Таблицами можно манипулировать посредством непроцедурных (декларативных) операций — **запросов**, *результатами которых также являются таблицы*

# Преимущества реляционной модели

Легко реализуются связи

**«многие-ко-многим»** и

**«один-ко-многим».**

Структурно простая модель.

Одним из основных преимуществ реляционной модели является ее однородность. Все данные хранятся в плоских таблицах

СУБД реляционного типа  
— **dBase, Clipper,**  
**Paradox, FoxPro,**  
**Clarion, Access**

# Элементы табличной структуры данных

- **запись базы данных** — это строка таблицы. Запись характеризует *один объект* БД и содержит набор значений, размещенных в полях БД;
- **поле** — столбец таблицы. Поле характеризует *один из параметров* каждого объекта базы данных.



Каждый столбец таблицы имеет уникальное для своей таблицы имя.

Строки не имеют имен, порядок их следования в таблице не определен, а количество - логически не ограничено. Так как строки в таблице не упорядочены, невозможно выбрать строку по ее позиции.

Хотя в таблице БД у каждой строки имеется номер, он не характеризует строку. Его значение изменяется при удалении строк из таблицы. Логически среди строк не существует “первой” и “последней” .

Таблица отражает тип  
объекта реального мира -  
**сущность**, а каждая ее  
строка один конкретный  
объект - **экземпляр**  
**сущности.**

Например, если нам требуется хранить информацию о людях, включающую **фамилию** каждого, **имя, отчество, ИНН, страну проживания и дату рождения**, то сущностью является именно человек, а указанные данные — атрибутами. Сама сущность естественным образом становится названием таблицы.

# Таблица «Человек»

ИНН	Фамилия	Имя	Отчество	Дата рождения

Столбец или совокупность столбцов, значения которых однозначно идентифицируют каждую строку таблицы **называется первичным ключом таблицы.**

Если таблица удовлетворяет требованию уникальности первичного ключа, она называется отношением. В реляционной модели все таблицы должны быть преобразованы в отношения.

Отношения реляционной модели связаны между собой. Связи поддерживаются внешними ключами.

**Внешний ключ** - это столбец (совокупность столбцов), значение которого однозначно характеризует значения первичного ключа другого отношения



Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором та же совокупность столбцов является первичным ключом.

Таблица 1 СОТРУДНИК ← *Название таблицы*

№ ПРОПУСКА	Фамилия	Должность	НАЗВАНИЕ ОТДЕЛА	Телефон

*Первичный ключ  
Таблицы 1*

*Внешний ключ Таблицы 1*

Таблица 2 ОТДЕЛ ← *Название таблицы*

НАЗВАНИЕ ОТДЕЛА	Расположение отдела	Назначение отдела

*Первичный ключ  
Таблицы 2*

*Организация ссылки от одной таблицы к другой*