

Иерархическая,  
сетевая и  
реляционная модели  
базы данных

# ТИПЫ ЛОГИЧЕСКИХ МОДЕЛЕЙ БАЗ ДАННЫХ

Ядром любой базы данных является модель данных.

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

По способу установления связей между данными различают:

- **Иерархическую;**
- **Сетевую;**
- **Реляционную модель.**

# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ



# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

это модель данных, где используется представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

Позволяет строить базы данных с древовидной структурой. В них каждый узел содержит свой тип данных (сущность).

На верхнем уровне дерева в этой модели имеется один узел – «корень», на следующем уровне располагаются узлы, связанные с этим корнем, затем узлы, связанные с узлами предыдущего уровня и т.д., причем каждый узел может иметь только одного предка

# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

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.

# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

Основные достоинства иерархической модели - простота описания иерархических структур реального мира и быстрое выполнение запросов, соответствующих структуре данных, однако, они часто содержат избыточные данные. Кроме того, не всегда удобно каждый раз начинать поиск нужных данных с корня, а другого способа перемещения по базе в иерархических структурах нет.

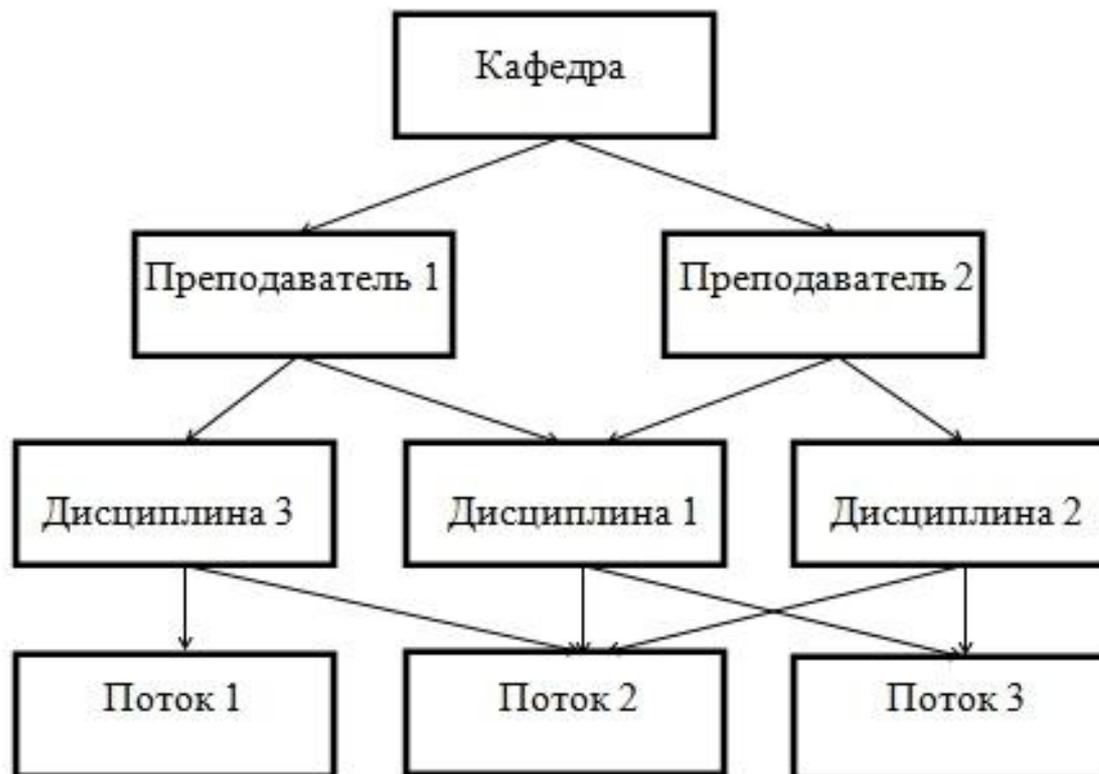
Указанный недостаток снят в сетевой модели, где, по крайней мере, теоретически возможны связи «всех информационных объектов со всеми».

# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

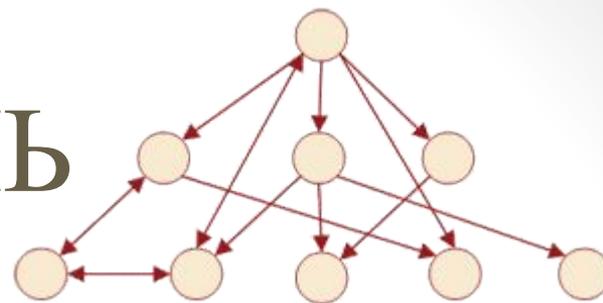
Примерами баз данных с иерархической моделью являются:

- Типичным представителем (наиболее известным и распространенным) является Information Management System (IMS) фирмы IBM (1966-1968 г.).
- Time-Shared Date Management System (TDMS) компании System Development Corporation;
- Mark IV MultiAccess Retrieval System компании Control Data Corporation;
- System 2000 разработки SAS Institute;
- InterSystems Caché.

# СЕТЕВАЯ МОДЕЛЬ

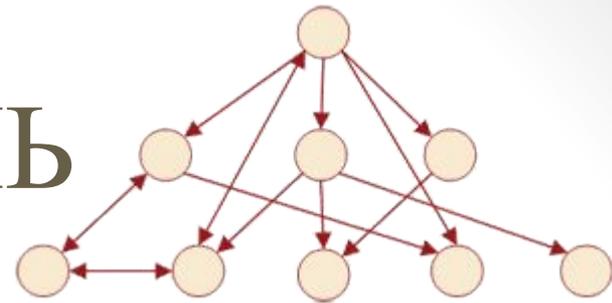


# СЕТЕВАЯ МОДЕЛЬ



Сетевая модель данных — логическая модель данных, являющаяся расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных.

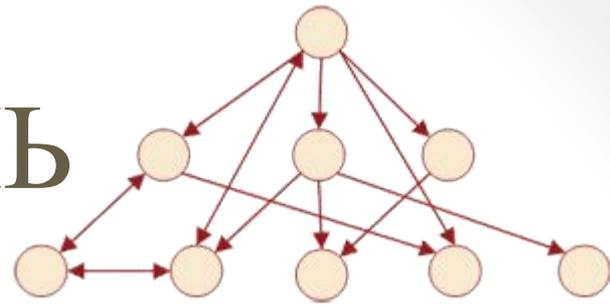
# СЕТЕВАЯ МОДЕЛЬ



Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков.

Сетевая БД состоит из набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между этими записями.

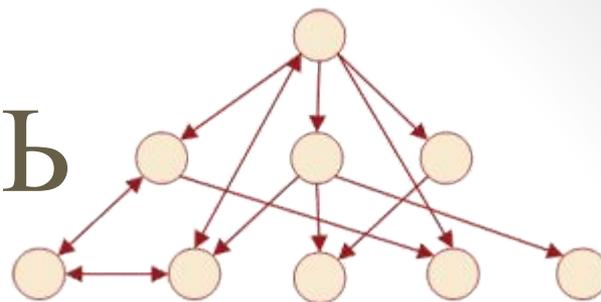
# СЕТЕВАЯ МОДЕЛЬ



Тип связи определяется для двух типов записи: предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка. Для данного типа связи  $L$  с типом записи предка  $P$  и типом записи потомка  $C$  должны выполняться следующие два условия:

- каждый экземпляр типа записи  $P$  является предком только в одном экземпляре типа связи  $L$ ;
- каждый экземпляр типа записи  $C$  является потомком не более чем в одном экземпляре типа связи  $L$ .

# СЕТЕВАЯ МОДЕЛЬ



## Достоинства:

- Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности.

## Недостатки:

- Недостатком сетевой модели данных являются высокая сложность и жесткость схемы БД, построенной на её основе. Поскольку логика процедуры выборки данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение.

# СЕТЕВЫЕ СУБД

Сетевая СУБД — СУБД, построенная на основе сетевой модели данных.

К основным понятиям сетевой модели базы данных относятся:

- уровень,
- элемент (узел),
- связь.

Узел — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. В сетевой структуре каждый элемент может быть связан с любым другим элементом.

# СЕТЕВЫЕ СУБД

Сетевые базы данных подобны иерархическим, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.

Несмотря на то, что эта модель решает некоторые проблемы, связанные с иерархической моделью, выполнение простых запросов остается достаточно сложным процессом.

Также, поскольку логика процедуры выборки данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение.

# СЕТЕВЫЕ СУБД

## **Список самых значимых сетевых СУБД на 1978 год:**

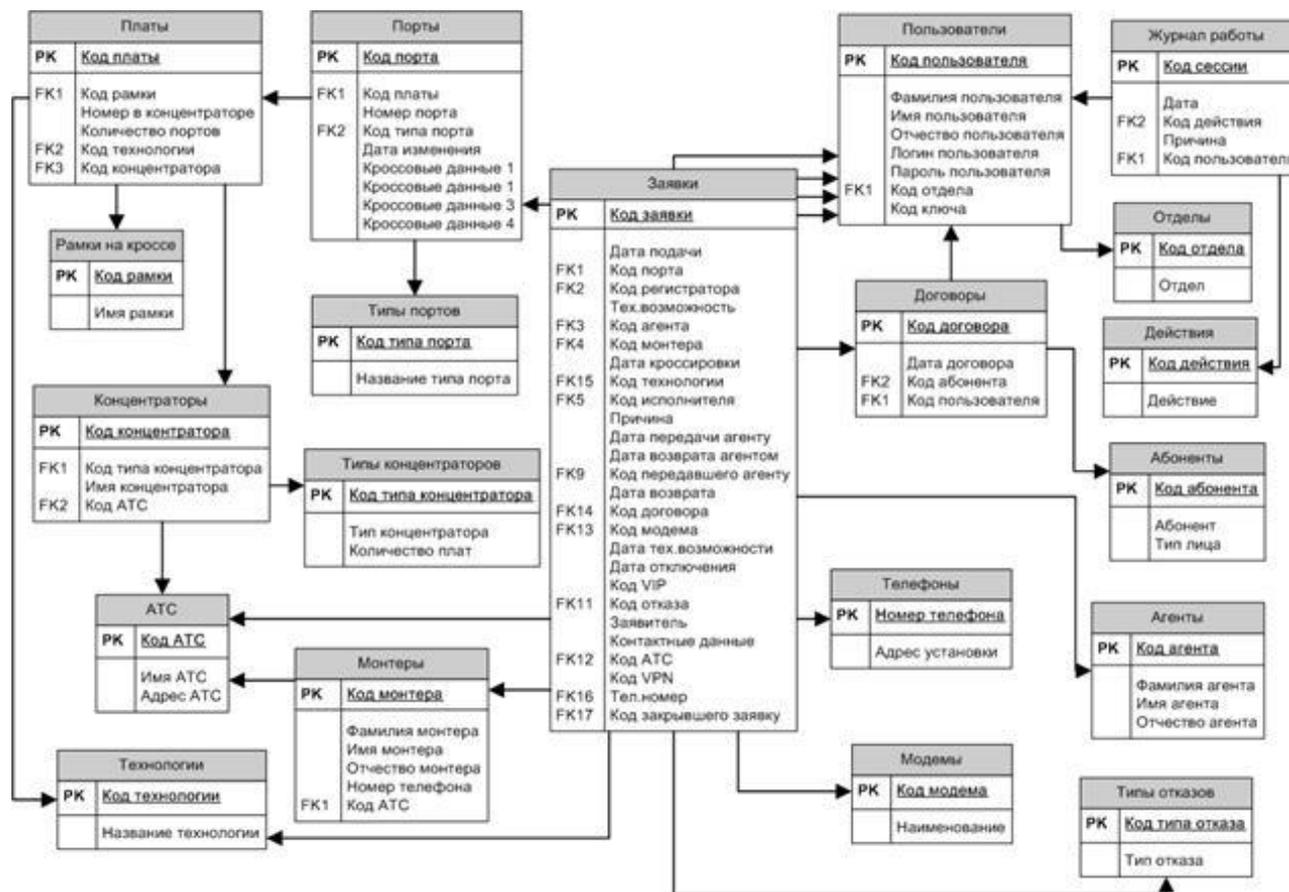
- IDS (Integrated Data Store) компании General Electric — самая первая сетевая СУБД, разработанная Чарльзом Бахманом в 1960 г.
- IDS/2 или IDS/II) компании Honeywell, купившей IDS у General Electric, позднее — компании Bull
- Integrated Database Management System (IDMS) компании Cullinet, развитие IDS на основе её исходных кодов
- DMS-1100 (для мейнфреймов UNIVAC 1100) и DMS-90 (для миникомпьютеров, первый релиз — ноябрь 1974) компании UNIVAC
- DBMS-10 компании DEC для Decsystem-10 и Decsystem-20
- CDC DMS-170
- Burroughs Data Management System (DMS-2). Продукт представлен на рынке в октябре 1974 года.

# СЕТЕВЫЕ СУБД

## **Другие подобные СУБД:**

- IMAGE/3000 компании Hewlett-Packard (1974 г.)
- Norsk-Data SYBAS
- NCR IDM-9000
- Cincom TOTAL
- dbVista
- Universal Datenbank System (UDS) от Siemens
- СООБЗ Cerebrum
- ИСУБД «CronosPRO»
- Caché
- GT.M

# РЕЛЯЦИОННАЯ МОДЕЛЬ



# РЕЛЯЦИОННАЯ МОДЕЛЬ

От англ. **Relation** – **отношение**.

Была разработана в начале 70-х годов Коддом. Простота и гибкость модели привлекли к ней внимание разработчиков. В 80-х годах она получила широкое распространение и реляционные СУБД оказались промышленным стандартом.

Все операции сводятся к манипуляциям с таблицами.

В реляционной модели информация представляется в виде прямоугольных таблиц. Каждая таблица состоит из строк и столбцов и имеет имя, уникальное внутри базы данных.

# РЕЛЯЦИОННАЯ МОДЕЛЬ

Основными понятиями реляционных баз данных являются:

- Тип данных
- Домен
- Атрибут
- Кортеж
- Первичный ключ
- Отношение

# Типы данных

Целые числа

Строки символов

Деньги

## Домены

Номера пропусков

Имена

Размеры выплат

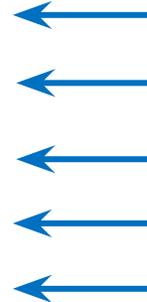
Номера отделов

Первичный ключ

Отношение «Сотрудники»

Сотр_номер	Сотр_фамил	Сотр_зарплата	Сотр_отдел_номер
2934	Иванов	112,000	310
2935	Петров	144,000	310
2936	Сидоров	92,000	313
2937	Федоров	110,000	310
2938	Иванова	112,000	315

Атрибуты



Кортежи

## Типы данных



Первичный  
ключ

Отношение  
«Сотрудники»

Понятие *тип данных* в реляционной модели данных полностью адекватно понятию типа данных в языках программирования. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как "деньги"), а также специальных "темпоральных" данных (дата, время, временной интервал). Достаточно активно развивается подход к расширению возможностей реляционных систем абстрактными типами данных (соответствующими возможностями обладают, например, системы семейства Ingres/Postgres). В нашем примере мы имеем дело с данными трех типов: строки символов, целые числа и "деньги".

В самом общем виде домен определяется заданием некоторого базового типа данных, к которому относятся элементы домена, и произвольного логического выражения, применяемого к элементу типа данных. Если вычисление этого логического выражения дает результат "истина", то элемент данных является элементом домена.

Наиболее правильной интуитивной трактовкой понятия домена является понимание домена как допустимого потенциального множества значений данного типа. Например, домен "Имена" в нашем примере определен на базовом типе строк символов, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).



Схема отношения - это именованное множество пар {имя атрибута, имя домена (или типа, если понятие домена не поддерживается)}. Степень или "арность" схемы отношения - мощность этого множества. Степень отношения СОТРУДНИКИ равна четырем, то есть оно является 4-арным. Если все атрибуты одного отношения определены на разных доменах, осмысленно использовать для именования атрибутов имена соответствующих доменов (не забывая, конечно, о том, что это является всего лишь удобным способом именования и не устраняет различия между понятиями домена и атрибута).

Схема БД (в с



## Типы данных

Кортеж, соответствующий данной схеме отношения, - это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. "Значение" является допустимым значением домена данного атрибута (или типа данных, если понятие домена не поддерживается). Тем самым, степень или "арность" кортежа, т.е. число элементов в нем, совпадает с "арностью" соответствующей схемы отношения. Попросту говоря, кортеж - это набор именованных значений заданного

Пер  
типа.  
ключ

Отношение  
«Сотрудники»

Сотр_номер	Сотр_фамил	Сотр_зарпл	Сотр_отд_номер
2934	Иванов	112,000	310
2935	Петров	144,000	310
2936	Сидоров	92,000	313
2937	Федоров	110,000	310
2938	Иванова	112,000	315

Атрибуты

Кортежи

## Типы данных

**Первичный ключ** — это атрибут или набор из минимального числа атрибутов, который однозначно идентифицирует конкретный кортеж и не содержит дополнительных атрибутов.

Подразумевается, что все атрибуты в первичном ключе должны быть необходимыми и достаточными для идентификации конкретного кортежа, и исключение любого из атрибутов в ключе сделает его недостаточным для

Сотр_номер	Сотр_фамил	Сотр_зарпл	Сотр_отд_номер
2934	Иванов	112,000	310
2935	Петров	144,000	310
2936	Сидоров	92,000	313
2937	Федоров	110,000	310
2938	Иванова	112,000	315

Отношение - это множество кортежей, соответствующих одной схеме отношения. Иногда, чтобы не путаться, говорят "отношение-схема" и "отношение-экземпляр", иногда схему отношения называют заголовком отношения, а отношение как набор кортежей - телом отношения. На самом деле, понятие схемы отношения ближе всего к понятию структурного типа данных в языках программирования. Было бы вполне логично разрешать отдельно определять схему отношения, а затем одно или несколько отношений с данной схемой.

Однако в реляционных базах данных это не принято. Имя схемы отношения в таких базах данных всегда совпадает с именем соответствующего отношения-экземпляра. В классических реляционных базах данных после определения схемы базы данных изменяются только отношения-экземпляры. В них могут появляться новые и удаляться или модифицироваться существующие кортежи. Однако во многих реализациях допускается и изменение схемы базы данных: определение новых и изменение существующих схем отношения. Это принято называть *эволюцией схемы базы данных*.

Сотр_номер	Сотр_фамил	Сотр_зарпл	Сотр_отд_номер
2934	Иванов	112,000	310
2935	Петров	144,000	310
2936	Сидоров	92,000	313
2937	Федоров	110,000	310
2938	Иванова	112,000	315