



university

Тюменский  
индустриальный  
университет

**Тема №2. Взаимосвязи в моделях и реляционный  
подход к построению моделей**

**Лекция №4-5. Логическая и физическая независимость  
данных.  
Типы моделей данных. Реляционная модель  
данных.**

# Основные понятия моделей данных

**База данных (БД)** — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.



# Основные понятия моделей данных

**Данные** — это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы.

**Модель данных** — это некоторая абстракция, которая, будучи применима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

# Уровни моделей данных

Инфологическая модель данных

Обобщенное, не привязанное к каким либо ЭВМ и СУБД, описание предметной области (набор данных, их типов, длин, связей и т.д.)



Даталогическая модель данных

Описание на языке конкретной СУБД



Физические модели данных

Описание хранимых данных



**БАЗА ДАННЫХ**





# Основные

## ПОНЯТИЯ

1 **Атрибуты** – это значения, описывающие свойства сущности

2 **Объекты (сущности)** – вершины математического графа

3 **Связи** – это соединения между двумя или более множествами сущностей

4 **Связи** – дуги графа

5 **Сущности** – это члены множества сущностей



# Инфологические модели

**Инфологические модели данных** используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения.

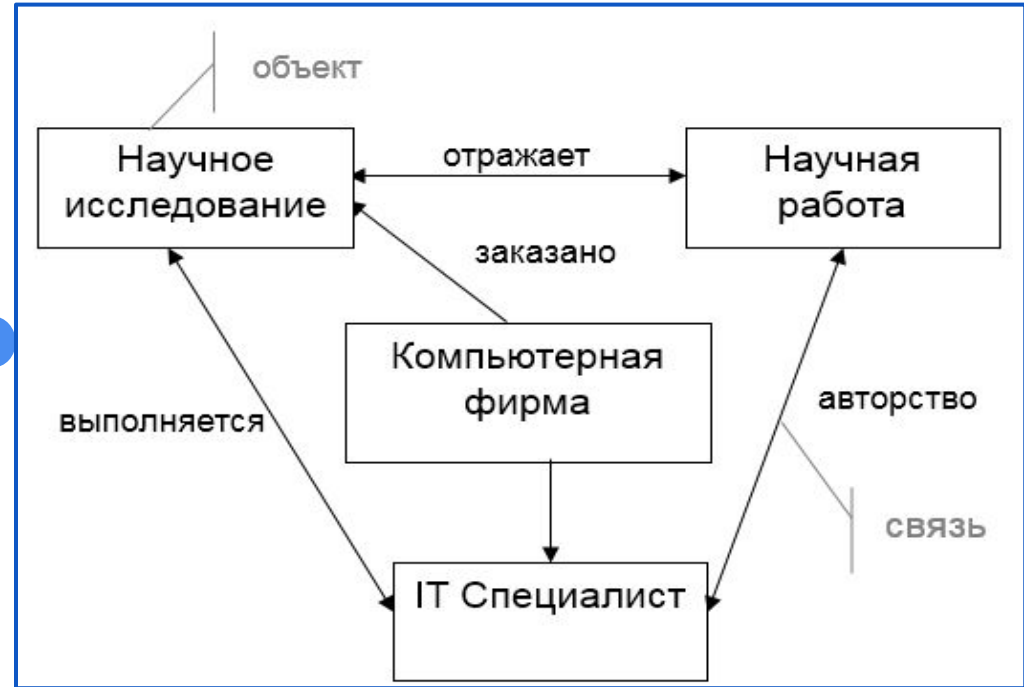


**Цель инфологического моделирования** – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных.

# Диаграммы Бахмана

## Минус

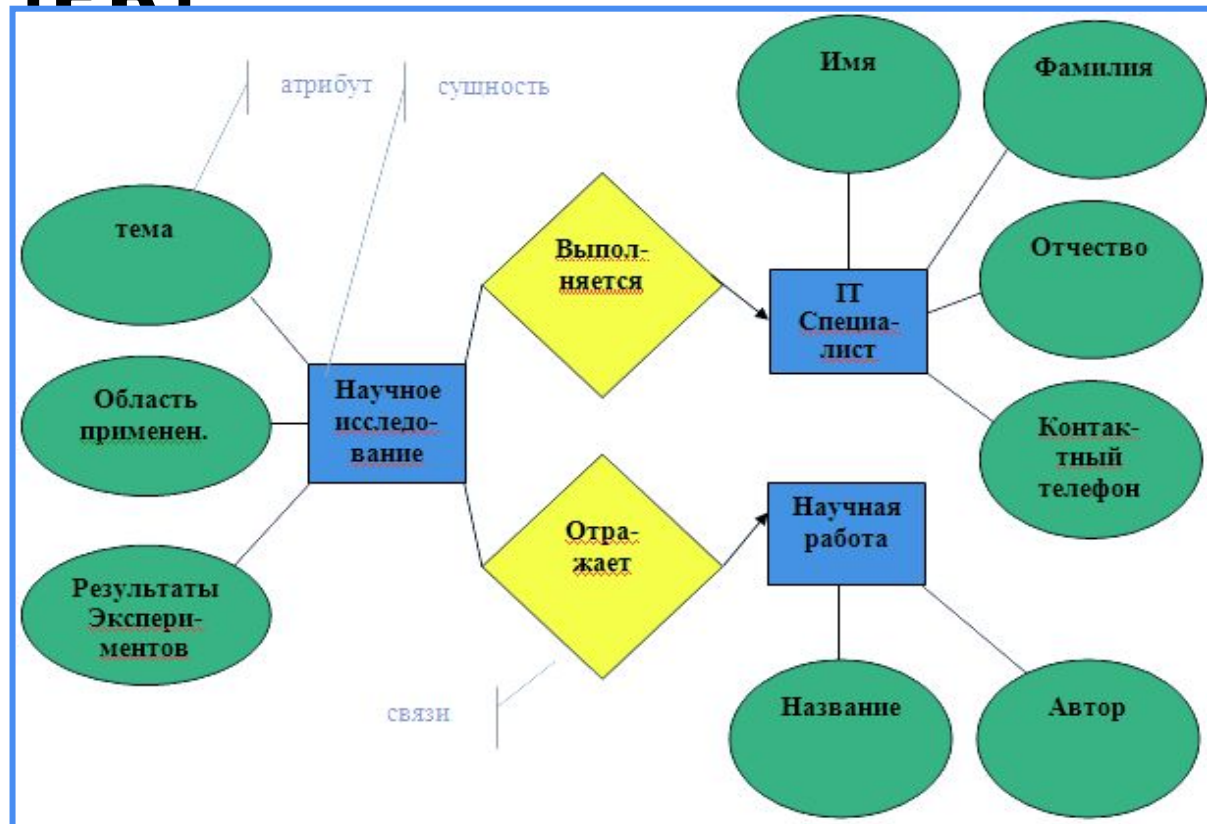
Статичность, не позволяющая наглядно отображать процессы, в которые вовлечены сущности и к которым подвержены отношения.



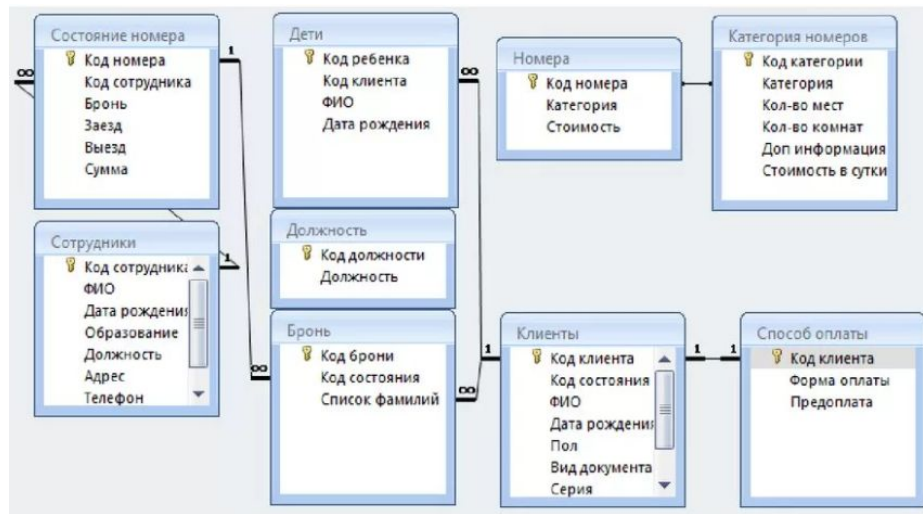
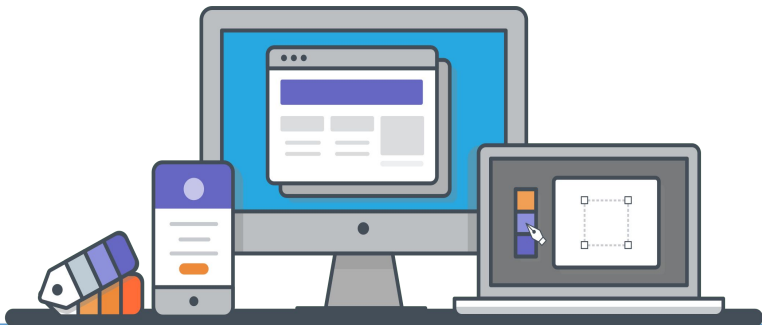


# Модель «сущность-связь» (ER)

Пример  
диаграммы



# Даталогические модели



**Даталогическая модель** – это модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации.

# Документальные модели

**Документальная модель** соответствует представлению о слабоструктурированной информации, ориентированной на свободные форматы документов, текстов на естественном языке.

**Документальная  
модель**

**Тезаурусные  
модели**

**Ориентированные на  
формат документа**

**Дескрипторные  
модели**

# Ориентированные на формат документа



Модели основаны на языках разметки документов и связаны прежде всего со стандартным общим языком разметки – SGML.

**SGML** – это язык предназначенный для создания других языков разметки, он определяет допустимый набор тегов (ссылок), их атрибуты и внутреннюю структуру документа.

# Ориентированные на формат документа

Пример XML, описывающий данные о IT Специалисте:

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="IT Специалист" type="IT Специалист"/>
  <xs:complexType name="IT Специалист">
    <xs:sequence>
      <xs:element name="Фамилия" type="xs:string"/>
      <xs:element name="Имя" type="xs:string"/>
      <xs:element name="Отчество" type="xs:string"/>
      <xs:element name="Контактный телефон" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

# Дескрипторные модели

**Дескрипторные модели** — используются на ранних стадиях использования документальных баз данных.

**Дескриптор** — описатель

Он имеет жесткую структуру и описывает документ в соответствии с теми характеристиками, которые требуются для работы с документами в разрабатываемой документальной БД.



# Дескрипторные модели



Пример дескрипторной классификации — УДК  
— универсальная десятичная классификация.

Применяется, например, так: 142.214.53 +  
213434/(431) имеет значение «Справочники по  
радиоприемникам и телевизорам».

# Тезаурусные модели

Тезаурусные модели основаны на принципе организации словарей, содержат определенные языковые конструкции и принципы их взаимодействия в заданной грамматике.

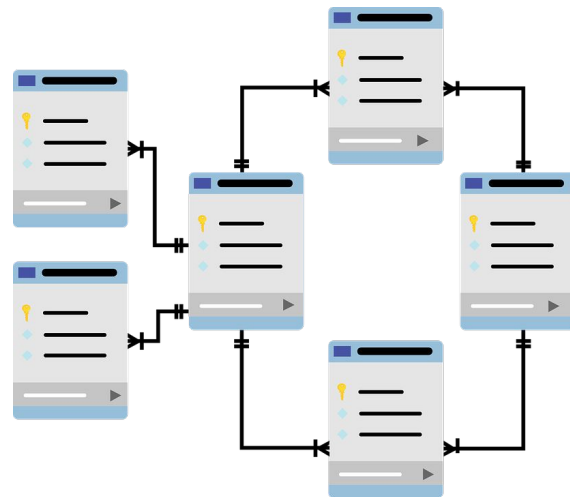
**Тезаурус** – это словарь, в явном виде фиксирующий семантические отношения между составляющими его единицами.



# Фактографические модели

**Фактографические модели** – соответствуют представлению о четко структурированной информации, формализованных данных (дерево, сеть, таблица)

Примерами фактографических моделей могут служить справочные таблицы значений физических величин, спецификации деталей и сборок, используемых в промышленных изделиях, и т.п.



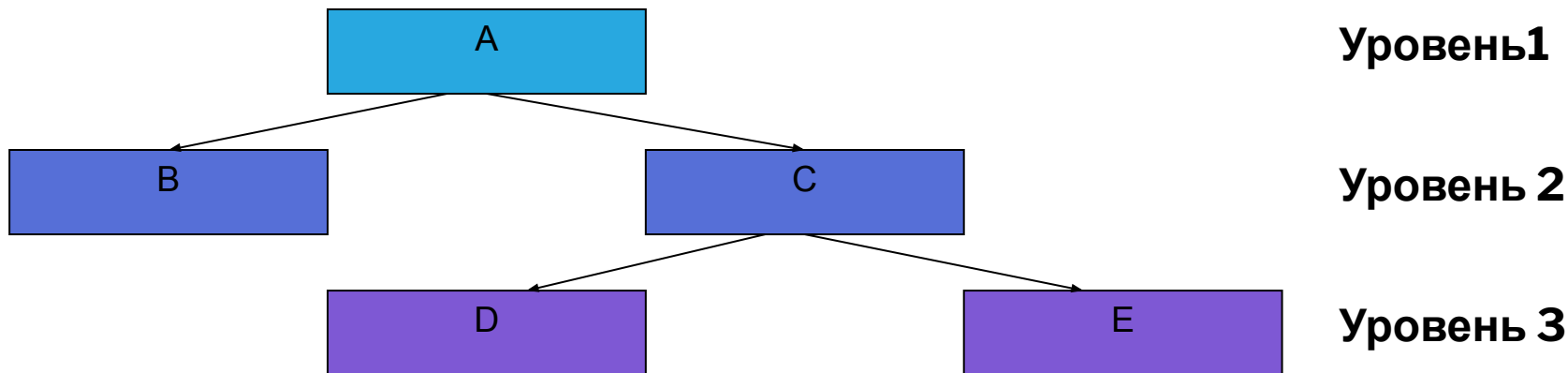
# Теоретико-графовые модели

**Теоретико-графовая модель** – это совокупность объектов реального мира в виде графа взаимосвязанных информационных объектов.



# Иерархическая модель

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



# Иерархическая модель

Основные информационные единицы в иерархической модели: база данных, поле данных, сегмент данных.

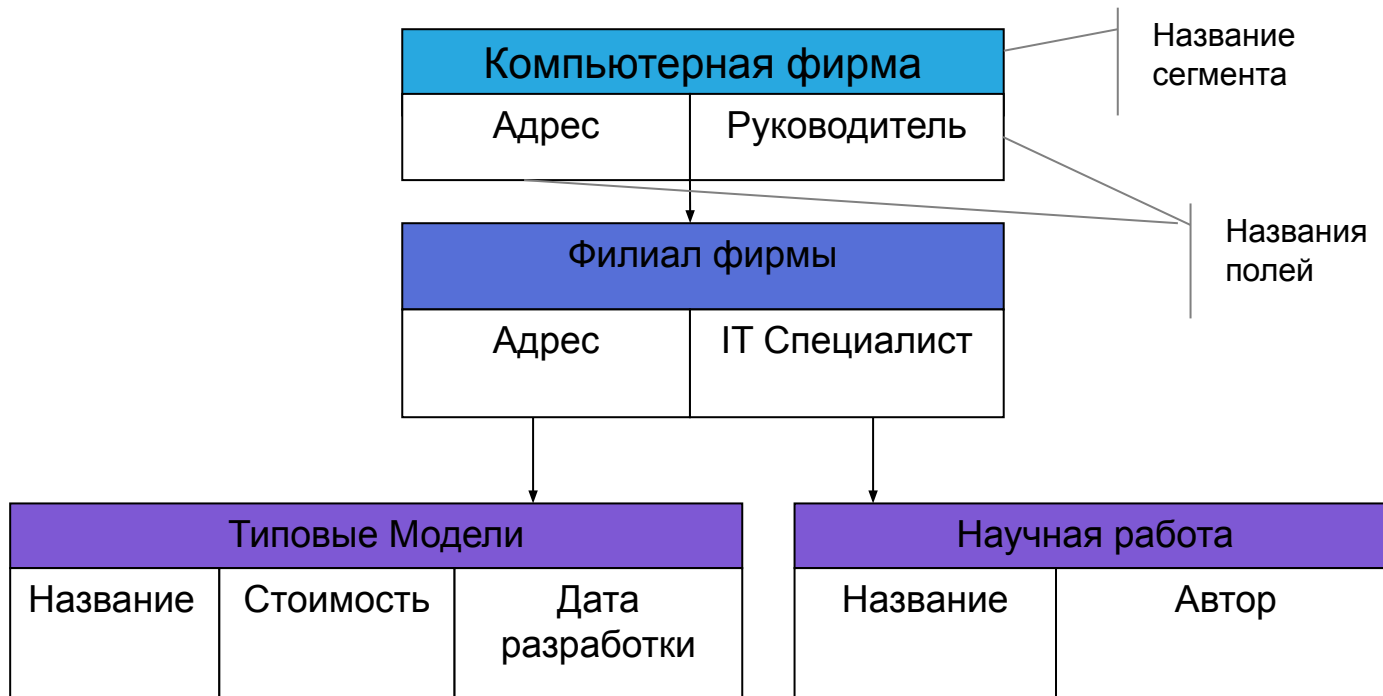


**Поле данных** – это минимальная, неделимая единица данных, доступная пользователю с помощью СУБД.

**Сегмент (запись) данных** – более высокий уровень абстракции, объединение полей данных.

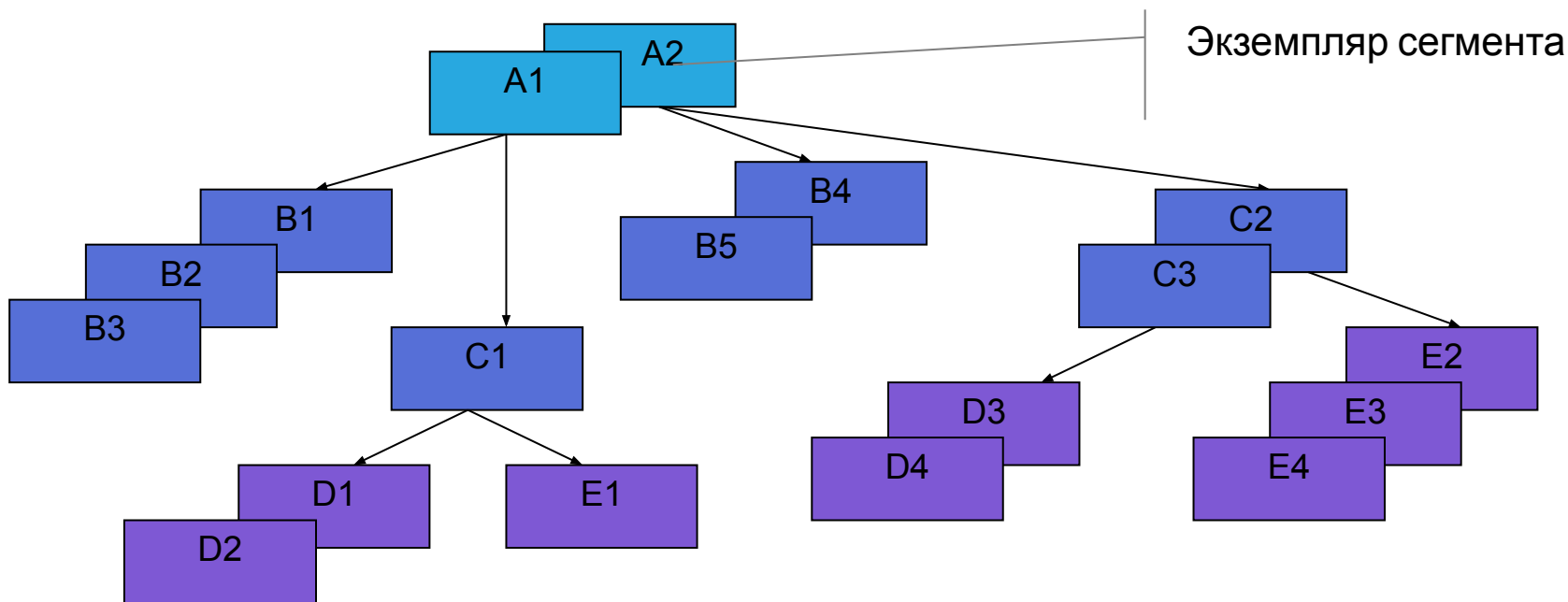
# Иерархическая модель

Пример Иерархической модели данных:



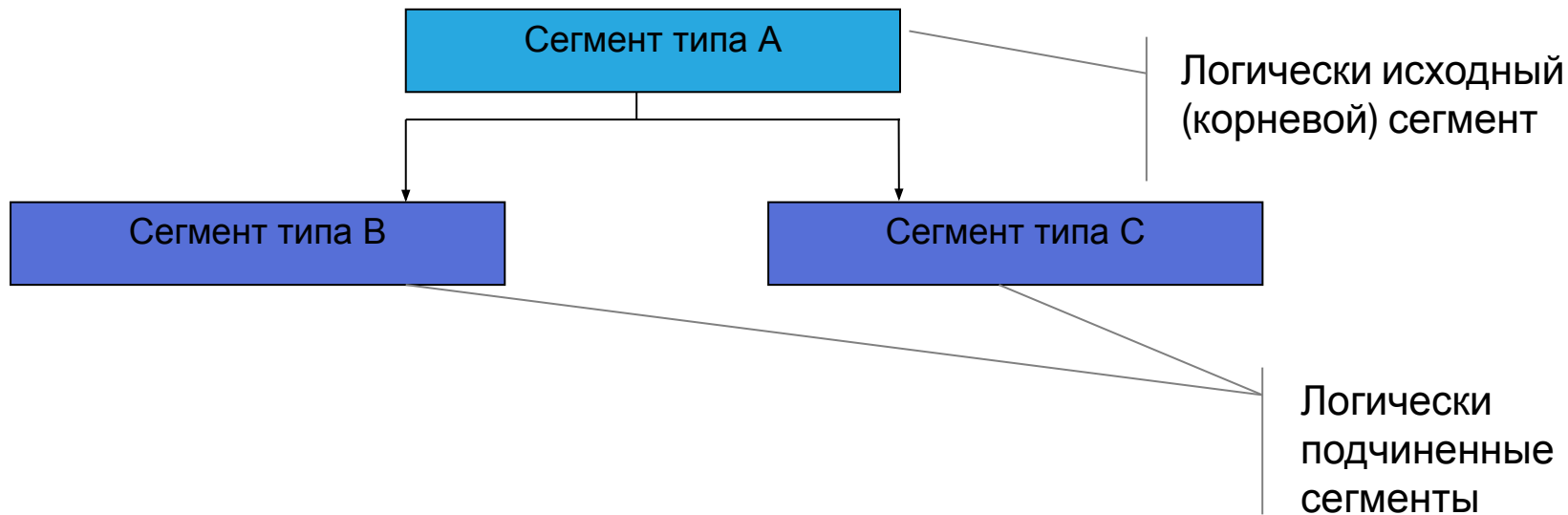
# Иерархическая модель

Пример иерархии экземпляров дерева



# Иерархическая модель

Приме  
р



# Иерархическая модель

**Физическая БД удовлетворяет иерархическим ограничениям**

В каждой физической БД существует один корневой сегмент, у которого нет логически исходного (родительского) типа сегмента.

Каждый логически исходный сегмент может быть связан с произвольным числом логически подчиненных сегментов

Каждый логически подчиненный сегмент может быть связан только с одним логически исходным (родительским) сегментом



# Иерархическая модель

Логически исходный  
сегмент

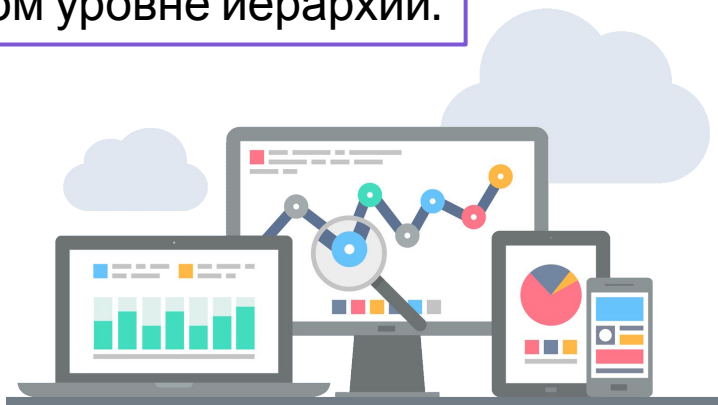


Тип сегмента,  
находящийся на более  
высоком уровне иерархии.

Логически  
подчиненный сегмент



Данные соединенные  
направленными  
иерархическими ребрами  
с логически исходными  
сегментами.



# Иерархическая модель



**Экземпляр сегмента** образуется из конкретных значений полей или элементов данных, в него входящих.

**Тип сегмента** — это поименованная совокупность типов элементов данных (полей), которые его образуют.

# Иерархическая модель

## Преимущества

- Простота понимания и использования, быстрота доступа к данным
- Простота оценок операционных характеристик.



## Недостатки

- Трудность реализации взаимосвязей «многие-ко-многим»
- Из-за строгой иерархической упорядоченности объектов модели значительно усложняются операции включения и удаления
- Язык манипулирования данными в иерархической модели поддерживает в явном виде навигационные операции

# Сетевая модель

Агрегат типа вектор соответствует линейному набору элементов данных:

Адрес				
Индекс	Город	Улица	Дом	Квартира

Зарплата	
месяц	сумма



## Базовые объекты модели

**Элемент данных** — минимальная информационная единица, доступная пользователю с использованием СУБД.

**Агрегат данных** — агрегат типа вектор и агрегат типа повторяющаяся группа.

**Запись** — совокупность агрегатов или элементов данных, моделирующая некоторый класс объектов реального мира. Различают типа записи и экземпляра записи.

**Набор данных** — двухуровневый граф, связывающий отношением «один-ко-многим» два типа записи.

# Теоретико-множественные модели

Появление теоретико-множественных моделей в системах баз данных было предопределено настоятельной потребностью пользователей в переходе от работы с элементами данных к работе с некоторыми макрообъектами.



# Реляционная модель

**Реляционная база данных** – это совокупность отношений, содержащих всю информацию, которая должна храниться в базе данных.

**Отношение** – подмножество декартова произведения одного или более доменов.



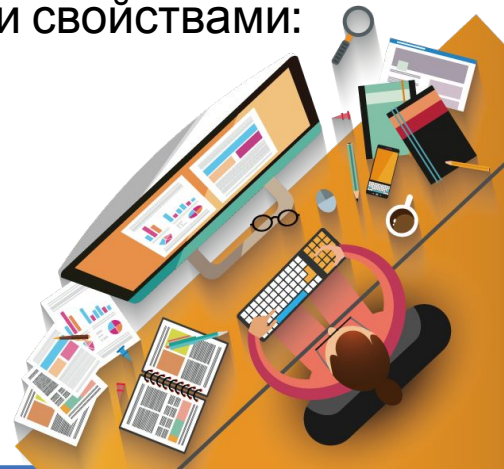
# Реляционная модель

## Примеры отношений

● **Бинарные отношения.** В математике большую роль играют бинарные отношения, т.е. отношения, заданные на декартовом произведении двух множеств

● **Отношение эквивалентности.** Отношение  $R$  на множестве называется отношением эквивалентности, если оно обладает следующими свойствами:

- 1)  $(x,x) \in R$  для всех  $x \in A$  (рефлексивность)
- 2) Если  $(x,y) \in R$ , то  $(y,x) \in R$  (симметричность)
- 3) Если  $(x,y) \in R$  и  $(y,z) \in R$ , то  $(x,z) \in R$  (транзитивность)





# Реляционная модель

## Примеры отношений

- Отношения порядка. Отношение  $R$  на множестве  $A$  называется отношением порядка, если оно обладает следующими свойствами:

- 1)  $(x,x) \in R$  для всех  $x \in A$  (рефлексивность)
- 2) Если  $(x,y) \in R$  и  $(y,x) \in R$ , то  $x=y$  (антисимметричность)
- 3) Если  $(x,y) \in R$  и  $(y,z) \in R$ , то  $(x,z) \in R$  (транзитивность)



# Реляционная модель

**Домен** – множество возможных значений конкретного атрибута.

**Атрибут** – свойство объекта, явления или процесса. Примеры атрибутов: фамилия, имя, отчество, дата рождения.

**Кортеж** – элемент отношения, это отображение имен атрибутов в значения, взятые из соответствующих доменов.

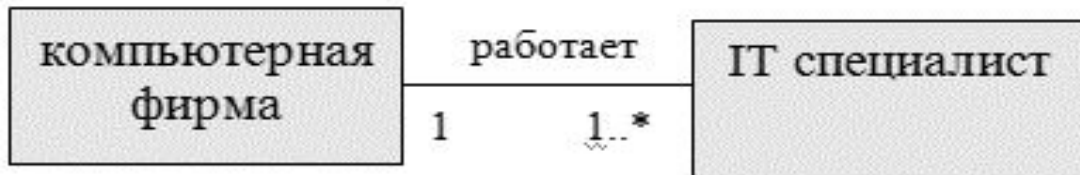
# Модель бинарных ассоциаций

**Бинарная ассоциация** – это ассоциация между ровно двумя классами.

**Роль** – это неотделимая часть ассоциации, описывающая некоторые свойства её соединения с классом (роль класса в данной ассоциации).

## Свойства роли

- Имя роли
- Навигация
- Множественность
- Квалификатор
- Агрегирование



# Модель бинарных ассоциаций

**Агрегирование** (aggregation) – это отношение между классами типа целое/часть.



**Композиция** указывает на то, что данный класс может являться частью только одного класса.

У ассоциации может быть атрибут под названием **квалификатор** (qualifier), который содержит один или несколько атрибутов класса, прикрепленного к другому концу ассоциации.

# Модель бинарных ассоциаций

**Множественность** (multiplicity) показывает возможное количество объектов, которые могут быть связаны в соответствии с этой ассоциацией.

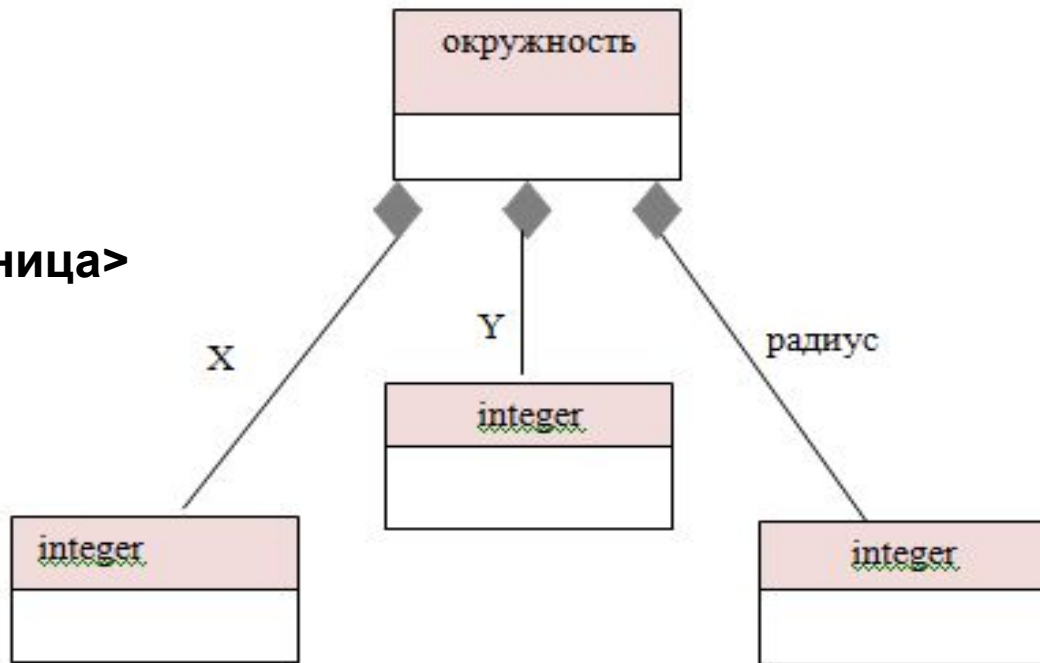
Множественность указывается

для ролей ассоциации и

имеет следующий

формат:

**<нижняя граница>...<верхняя граница>**



# Объектно-ориентированные модели

**Объекты**, обладающие внутренней структурой и однозначно идентифицируемые уникальным внутрисистемным ключом.

**Классы**, являющиеся по сути типами объектов.

**Операции** над объектами одного или разных типов, называемые «методами».

# Объектно-ориентированные модели

**Компьютерная фирма**

родительский класс

Название, Адрес, Директор, Научная работа, Инженерная работа, Научная работа

Название, Срок выполнения, Исполнитель, Проверяющий, Результаты экспериментов, Возможное применение, Тип

**Научно-исследовательская**

**Научно-методическая**

**Научно-консалтинговая**

**Инженерная работа**

дочерний класс

Название, Задачи, Цели, Результаты

Тип

**Векторные модели**

**3d макетирование**

**3d MAX  
(растровое и векторное)**

## Достоинства объектно-ориентированной модели:

- возможность для пользователя системы определять сложные типы данных;
- наследуемость свойств объектов;
- повторное использование программного описания типов объектов при обращении к другим типам, на них ссылающимся.

## Недостатки объектно-ориентированной модели:

- отсутствие строгих определений;
- отсутствие общеупотребимых стандартов.

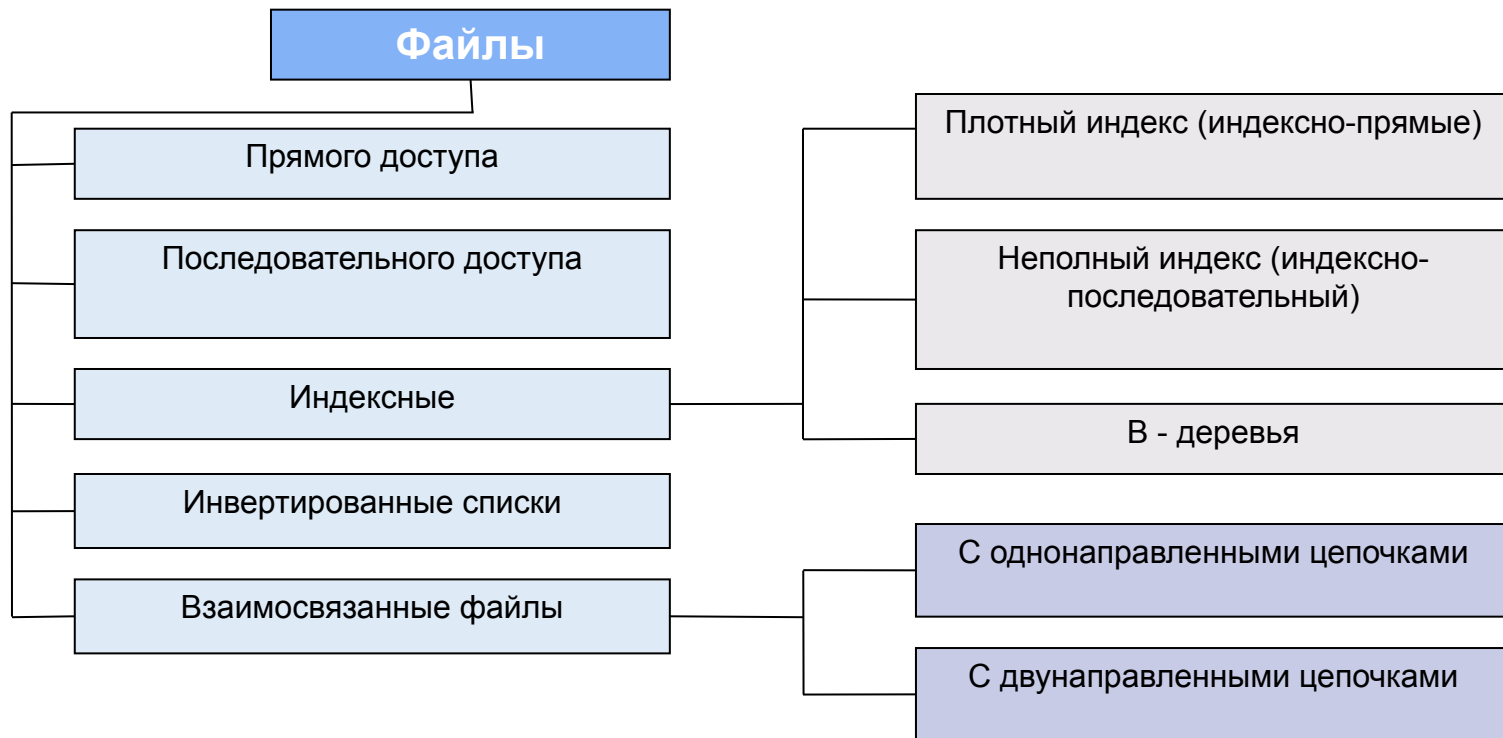


# Физические модели

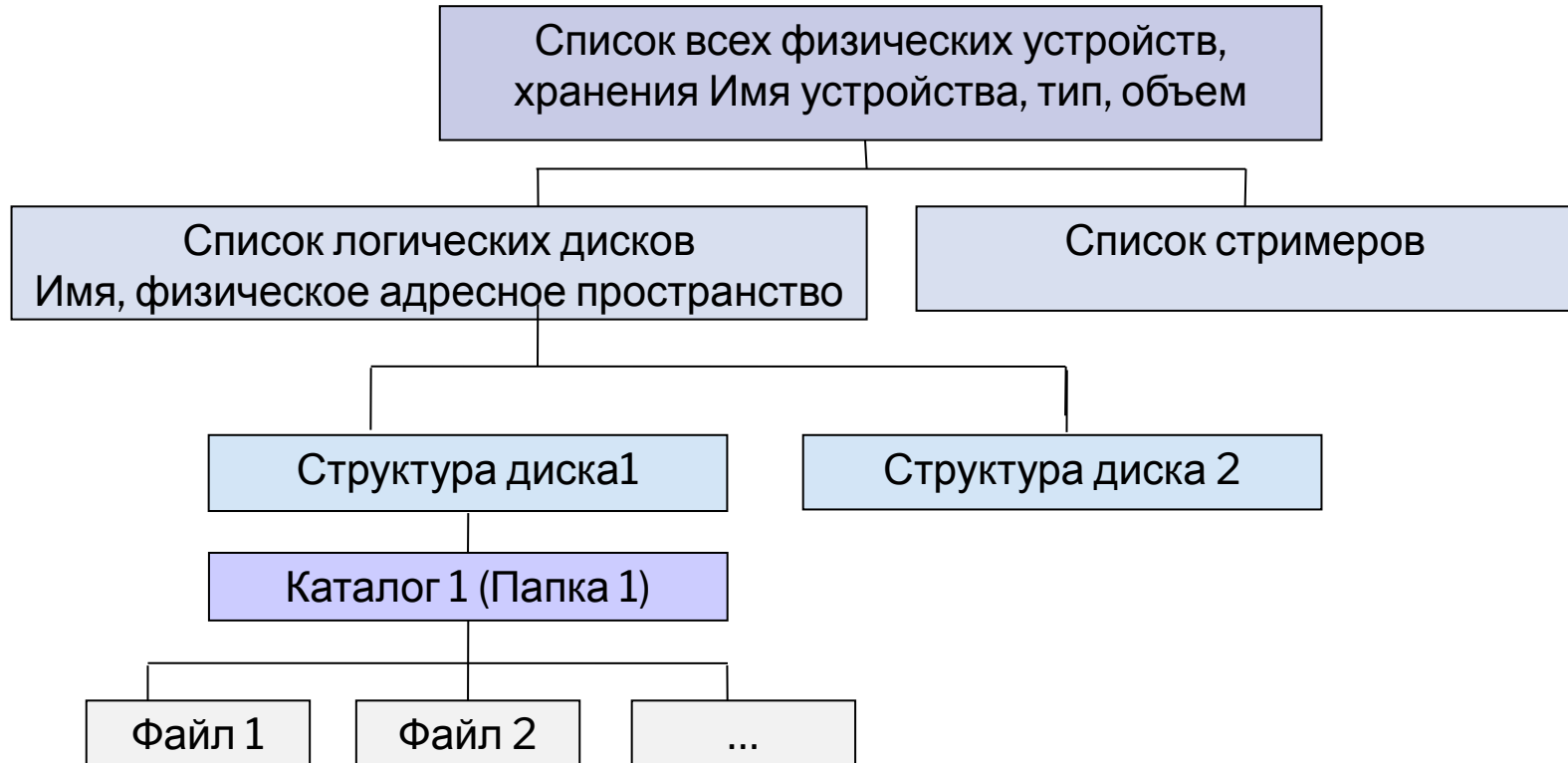
Физические модели баз данных определяют способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне.



# Модели, основанные на файловых структурах



# Модели, основанные на файловых структурах



# Модели, основанные на файловых структурах

Для каждого файла в системе хранится следующая информация

- имя файла
- тип файла (например, расширение или другие характеристики)
- размер записи
- количество занятых физических блоков
- базовый начальный адрес
- ссылка на сегмент расширения
- способ доступа (код защиты)



# Модели, основанные на странично- сегментной организации

Для каждого сегмента поддерживается таблица страниц.

Номер записи в таблице страниц соответствует номеру виртуальной страницы.

Размер записи колеблется от системы к системе, но чаще всего он составляет 32 бита. Из этой записи в таблице страниц находится номер кадра для данной виртуальной страницы, затем прибавляется смещение и формируется физический адрес.

# Сетевая модель

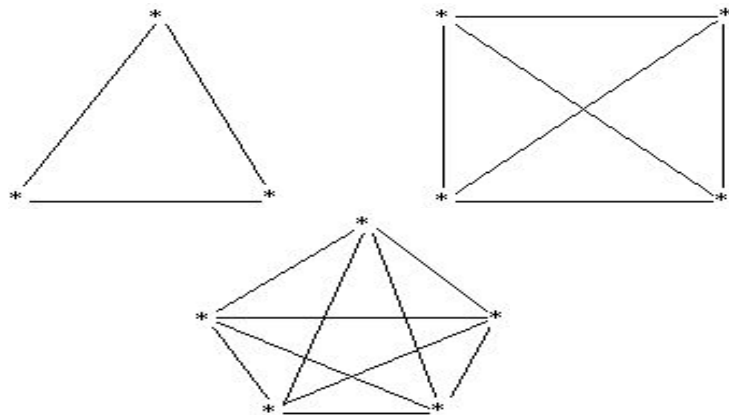
Запись в таблице страниц содержит информацию об атрибутах страницы. Это биты присутствия и защиты (0 – read/write, 1 – read only...).

- Бит модификации, который устанавливается, если содержимое страницы модифицировано
- Бит ссылки, который помогает выделить малоиспользуемые страницы;
- Бит, разрешающий кэширование, и другие управляющие биты.

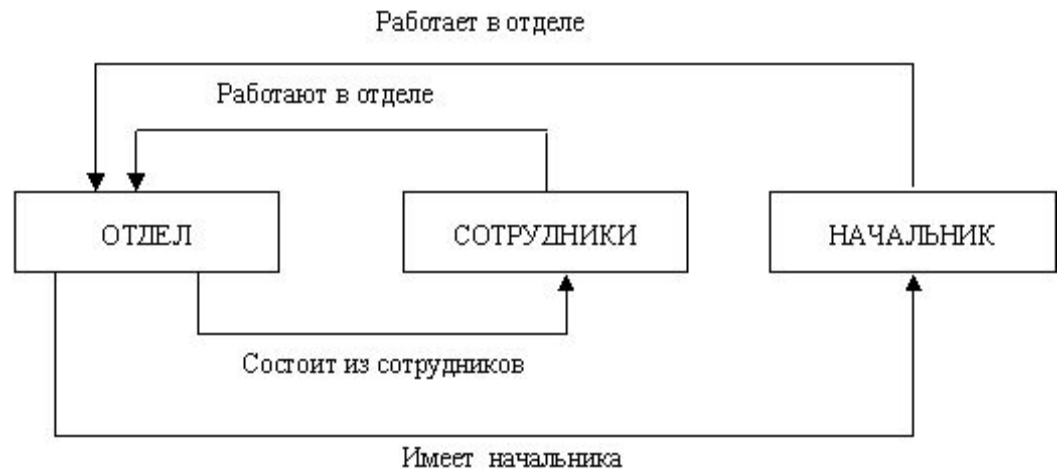
**Адреса страниц на диске не являются частью таблицы страниц.**

# Сетевая модель

**Сетевая модель** – это модель данных, хранящаяся в базе данных и описывающая взаимосвязи в виде неупорядоченного графа (сети).



**Представление связей в сетевой модели данных**



**Пример сетевой модели данных**

# Сетевая модель: операции





# Сетевая модель, базовые объекты



# Сетевая модель, элементы данных

Элементы данных определяют характеристики типов

Простые

Составные

- Векторы – повторяющиеся однотипные элементы
- Повторяющиеся группы – набор групп, включающих несколько разнотипных элементов
- неповторяющиеся группы – адрес – (город, улица, дом квартира)



# Сетевая модель преимущества

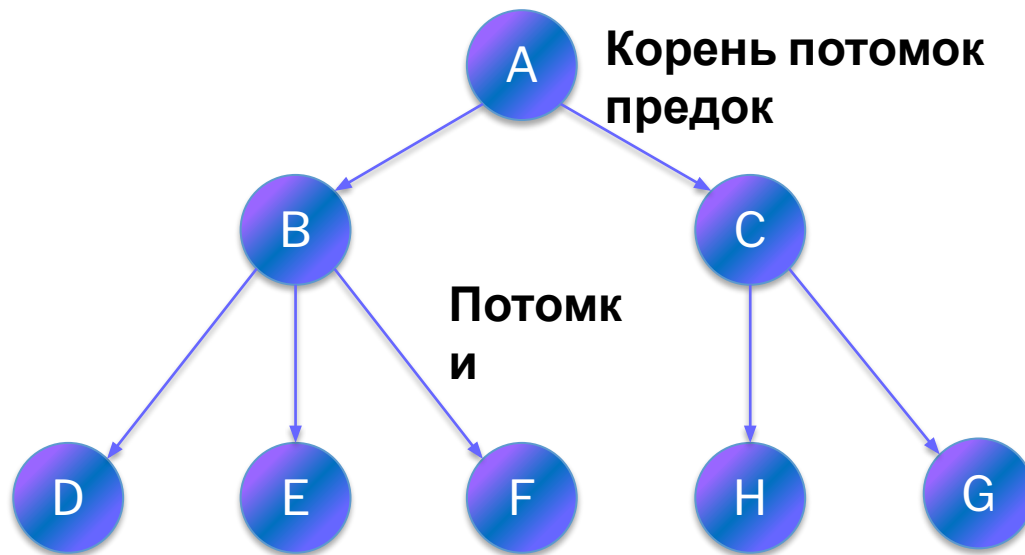
- + Возможность эффективной реализации по показателям затрат памяти и оперативности.
- + Большие возможности допустимости образования произвольных связей.
- + Поддержка сложные соотношения между типами данных, что делает их пригодными в различных приложениях.

# Сетевая модель недостатки

- Высокая сложность и жесткость схемы БД.
- Сложность для понимания и выполнения обработки информации в БД обычным пользователем.
- Ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.

# Иерархическая МОДЕЛЬ

Иерархическая модель базы данных представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое дерево (граф).



# Элементы графа

**Корень** - это элемент, который имеет подчиненные элементы и сам не является таковым.

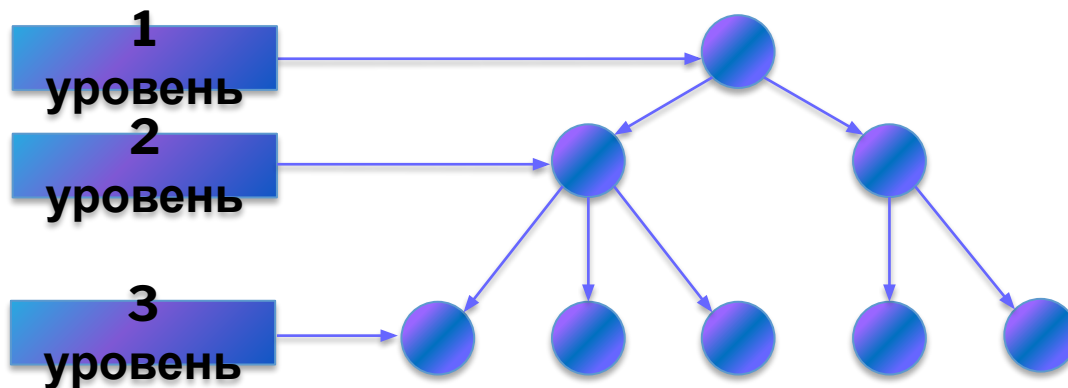
**Потомок** - это подчиненный элемент по отношению к элементу, который выступает для него в роли предка (родителя).

**Близнецы** - это потомки одного и того же родителя по отношению друг к другу.

# Элементы графа

Принцип работы: несколько узлов более низкого уровня соединяется при помощи связи с одним узлом более высокого уровня.

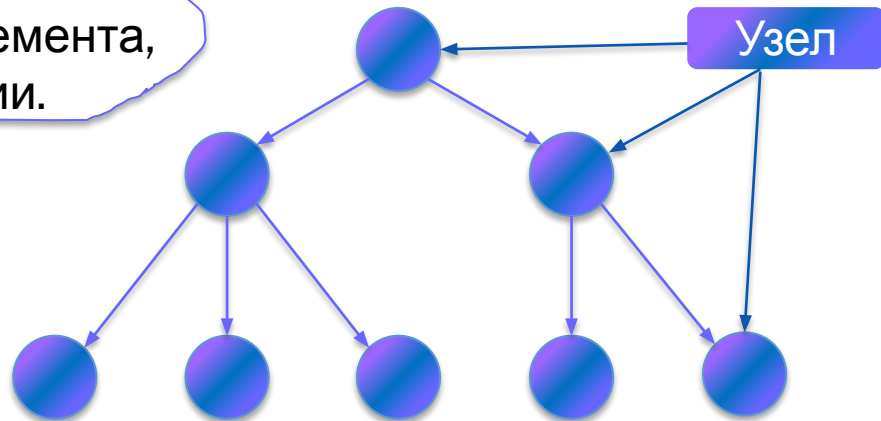
Данная модель характеризуется такими параметрами, как уровни, узлы, связи.



Верхний уровень (корень) – занимает один объект. Второй – объекты второго уровня и т.д.

# Элементы графа

Узел — информационная модель элемента, находящегося на данном уровне иерархии.



Между узлами существуют связи

**Отношение предка** – если узел связан с узлом более близким к корню

**Отношение потомка** – если узел связан с узлом более низкого уровня

**Отношение близнецы** – узлы имеют общего предка



# Свойства иерархической модели базы данных



Несколько узлов низшего уровня связано только с одним узлом высшего уровня.

Иерархическое дерево имеет только одну вершину (корень), неподчиненный никакой другой вершине.

Каждый узел имеет свое имя (идентификатор).

# Достоинства и недостатки

## Достоинства

- Эффективное использование памяти ЭВМ.
- Неплохие показатели времени выполнения основных операций над данными.
- Удобна для работы с иерархически упорядоченной информацией.

## Недостатки

- Громоздкость для обработки информации с достаточно сложными логическими связями.
- Сложность понимания для обычного пользователя.



# Правила преобразования к табличному виду

Количество уровней равно количеству столбцов. Количество элементов на последнем уровне равно количеству строк. Заполнять таблицу необходимо с левого нижнего элемента.



Фамилии	Специальность	Учеб.заведение	Университет
Иванов	ИС	МПК	ТИУ
Петров	ПКС	МПК	ТИУ
Сидоров	ПГС	СТР	ТИУ

# Реляционная модель данных (РМД)

В 1970 г. американский математик Э.Ф.Кодд опубликовал статью, с которой отсчитывается начало существования РМД.

РМД основана на теории множеств.

**Домен, D** – множество значений, которые может принимать элемент данных.

**Декартово произведение доменов** – множество всех возможных комбинаций значений доменов:

$$D_1 \times D_2 \times \dots \times D_n = \{(d_{1i}, d_{2i}, \dots, d_{ni})\}, \text{ где } d_{ki} \in D_k$$

Пример:  $D_1 = (1, 2)$ ,  $D_2 = (a, b, c)$ .

$$D_1 \times D_2 = \{(1,a), (1,b), (1,c), (2,a), (2,b), (2,c)\}$$

**Отношение** – подмножество декартова произведения доменов.



# Пример декартова произведения



Должность
директор
инженер
экономист

ФИО
Белов С.Ю.
Рогов А.И.
Панина А.А.
Волкова Н.М.



Оклад
40000
75000

ФИО	Должность	Оклад
Белов С.Ю.	директор	40000
Белов С.Ю.	директор	75000
<b>Белов С.Ю.</b>	<b>инженер</b>	<b>40000</b>
Белов С.Ю.	инженер	75000
Белов С.Ю.	экономист	40000
Белов С.Ю.	экономист	75000
Рогов А.И.	директор	40000
<b>Рогов А.И.</b>	<b>директор</b>	<b>75000</b>
Рогов А.И.	инженер	40000
Рогов А.И.	инженер	75000
Рогов А.И.	экономист	40000
...	...	...
<b>Волкова Н.М.</b>	<b>экономист</b>	<b>40000</b>

Полужирным шрифтом выделены записи, имеющие соответствие в предметной области.

# Пример таблицы реляционной БД

Табельный номер	ФИО сотрудника	Должность	Оклад	Год рождения	<u>Отдел</u>
023	Волкова Елена Павловна	секретарь	26000	1985	2
113	Белов Сергей Юрьевич	инженер	39800	1980	1
101	Рогов Сергей Михайлович	директор	62000	1972	2
056	Панина Анна Алексеевна	инженер-программист	41800	1978	1
...	...	...	...	...	...
098	Фролов Юрий Вадимович	начальник отдела	49200	1971	9

# Термины, Свойства отношения

Первичный ключ

Отношение, таблица

Столбец

Табельный номер	ФИО сотрудника	Должность	Оклад	Год рождения
023	Волкова Елена Павловна	секретарь	26000	1985
113	Белов Сергей Юрьевич	инженер	39800	1980

Описание (схема отношения)

Строка, запись, кортеж

**Отношение обладает двумя основными свойствами**

- 1 В отношении не должно быть одинаковых кортежей, т.к. это множество.
- 2 Порядок кортежей в отношении несущественен.

# Ключи отношения

**Ключ** – атрибут (группа атрибутов), которые позволяют классифицировать кортеж (запись таблицы).

**Внешний ключ** – служит для организации связей между таблицами.

**Уникальный ключ** – атрибут (группа атрибутов), которые позволяют идентифицировать кортеж (запись таблицы).

**Первичный ключ** – обязательный уникальный ключ. Для каждой таблицы может быть определен только один первичный ключ.



# Пример отношения ж/д расписание



# Организация связей между таблицами

Связь один-ко-многим: Отделы – Сотрудники

Таблица «Отделы»

Таблица «Сотрудники»

<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Отдел</i>
023	Волкова Елена Павловна	2
113	Белов Сергей Юрьевич	1
101	Рогов Сергей Михайлович	2
056	Панина Анна Алексеевна	1
...	...	...
098	Фролов Юрий Вадимович	9

<i>Номер отдела</i>	<i>Название отдела</i>
1	Информационный отдел
2	Администрация
3	Отдел кадров
...	...
9	Проектный отдел

«Номер отдела» - первичный ключ в таблице «Отделы»

«Отдел» – внешний ключ в таблице «Сотрудники»

# Организация связей между таблицами

Связь многие-ко-многим: Проекты – Сотрудники

<i>ФИО</i>	<i>Номер</i>
Волкова Е.П.	023
Белов С.Ю.	113
Рогов С.М.	101
Панина А.А.	056
Фролов Ю.В.	098
...	...

Таблица «Сотрудники»

Таблица «Участие»


<u>Участник</u>	<i>Роль</i>	<u>Проект</u>
113	исполнитель	23/Н
101	руководитель	18-К
056	исполнитель	18-К
101	консультант	09/Р
098	руководитель	23/Н
...	...	...

<i>Шифр</i>	<i>Название проекта</i>
23/Н	АИС "Налог"-2
18-К	ИПС "Жители"
09/Р	ГИС "Город"
...	...

Таблица «Проекты»

В таблице «Участие»: «Участник» – внешний ключ к таблице «Сотрудники», «Проект» – внешний ключ к таблице «Проекты»

# Пример связи внутри таблицы

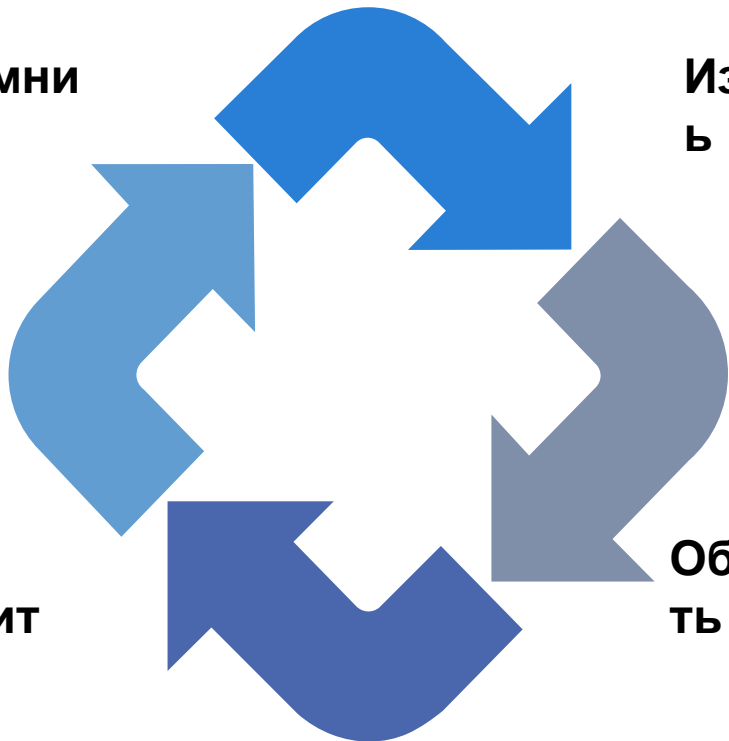


<i>Табельный номер</i>	<i>ФИО сотрудника</i>	<i>Должность</i>	<i>Оклад</i>	<i><u>Начальник</u></i>
023	Волкова Елена Павловна	секретарь	26000	101
113	Белов Сергей Юрьевич	инженер	39800	205
101	Рогов Сергей Михайлович	директор	62000	NULL
205	Махова Ольга Алексеевна	начальник отдела	51300	101
...	...	...	...	...

# Операции над данными в РМД

Операции применяются к кортежам отношений:

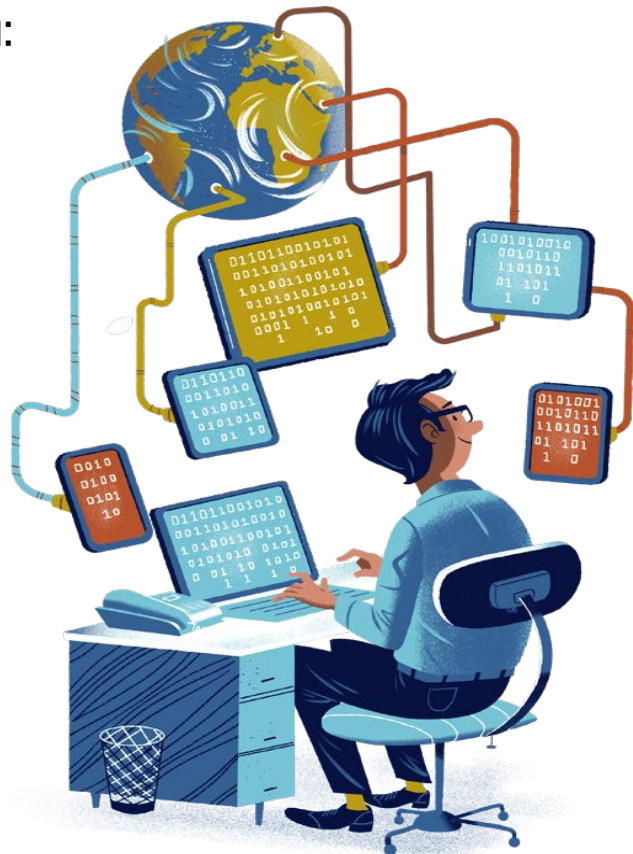
**Запомни  
ть**



**Извлеч  
ь**

**Удалит  
ь**

**Обнови  
ть**



# Преимущества и недостатки РМД

## Преимущества

- Наличие теоретического базиса.
- Максимальная степень независимости данных от программ.
- Наличие декларативного языка запросов.

## Недостатки

- Низкая эффективность выполнения запросов.
- Отсутствие однозначного соответствия между сущностями предметной области и таблицами реляционной базы данных.

