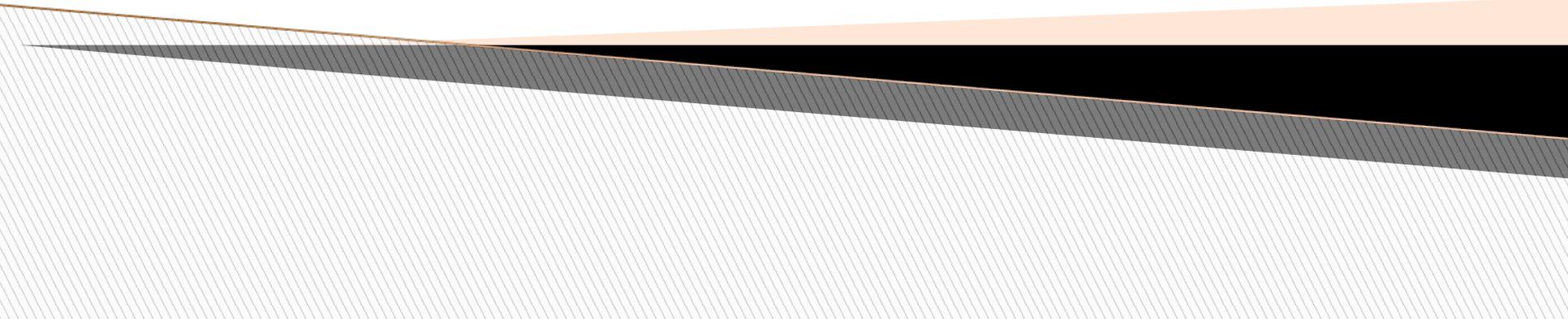


Теоретико-графовые модели данных

Учебные вопросы:

- 1. Иерархическая модель.**
- 2. Сетевая модель.**
- 3. Реляционная и постреляционная модели.**

Информационные модели. Графы.



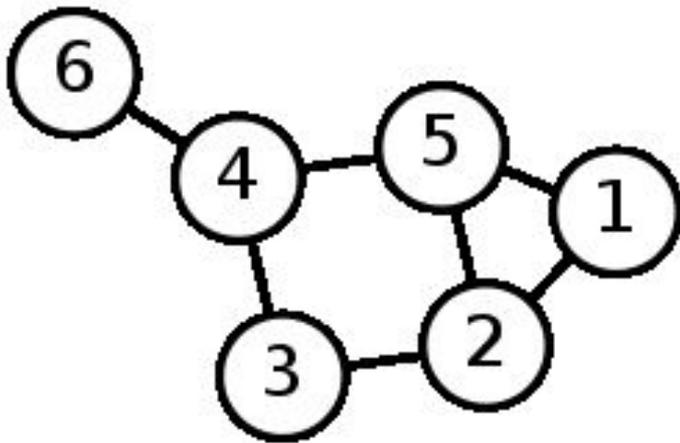


- Впервые основы теории графов появились в работах **Леонарда Эйлера** (1707-1783; *швейцарский, немецкий и российский математик*), в которых он описывал решение головоломок и математических развлекательных задач.
- Теория графов началась с решения Эйлером **задачи о семи мостах Кёнигсберга**.

ГРАФЫ

Граф – это информационная модель, представленная в графической форме.

Граф - множество вершин (узлов), соединённых рёбрами.



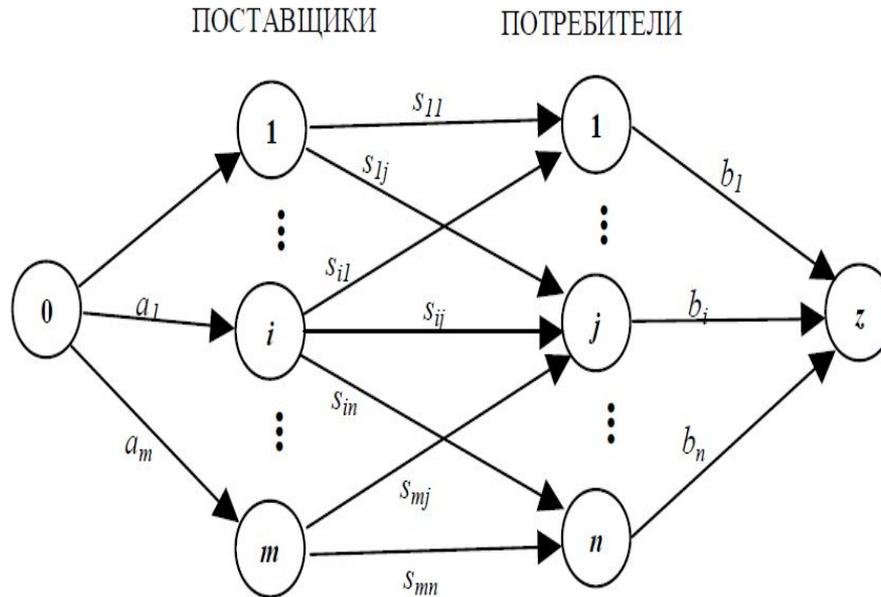
Вершины называют смежными, если их соединяет ребро.

Граф с шестью вершинами и семью рёбрами.

Ориентированные графы - орграфы

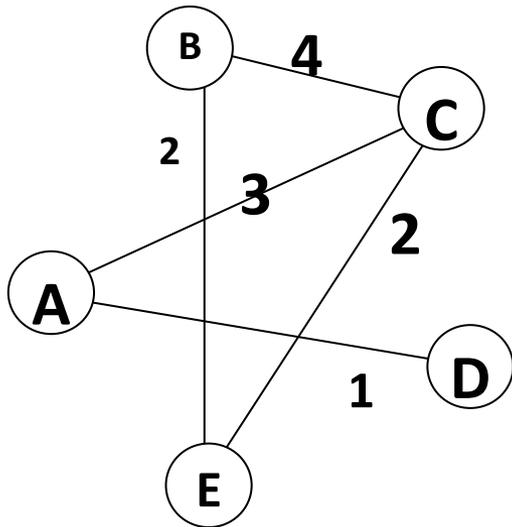
- Каждое ребро имеет одно направление.
- Такие ребра называются **дугами**.

Ориентированный граф



Взвешенный граф

- Это граф, рёбрам или дугам которого поставлены в соответствие числовые величины (они могут обозначать, например, расстояние между городами или стоимость перевозки).
- Вес графа равен сумме весов его рёбер.



	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

Таблице (она называется **весовой матрицей**) соответствует граф.

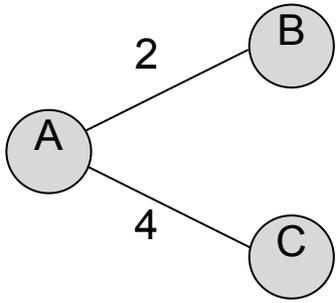
Задача

- Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет). **Определите длину кратчайшего пути между пунктами А и F** (при условии, что передвигаться можно только по построенным дорогам).

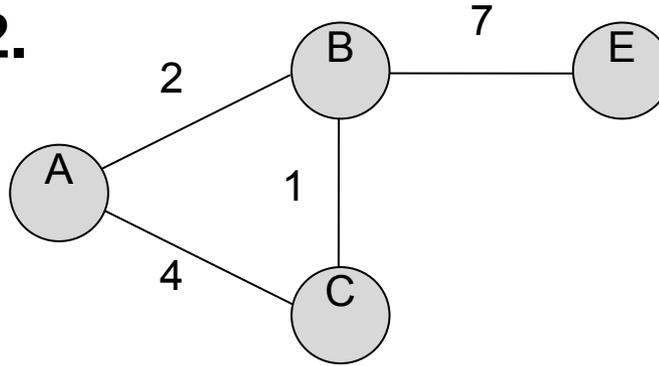
	A	B	C	D	E	F
A		2	4			
B	2		1		7	
C	4	1		3	4	
D			3		3	
E		7	4	3		2
F					2	

1) 9 2) 10 3) 11 4) 12

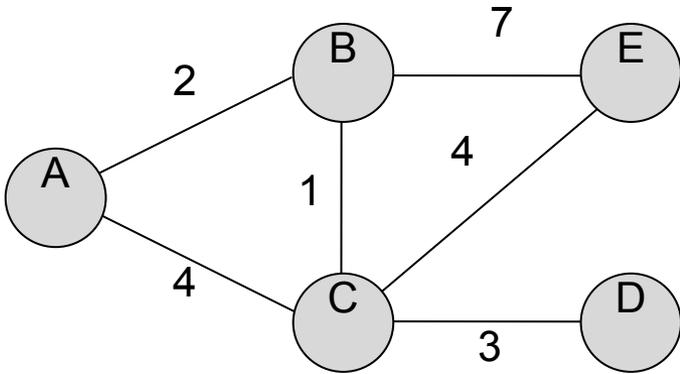
1.



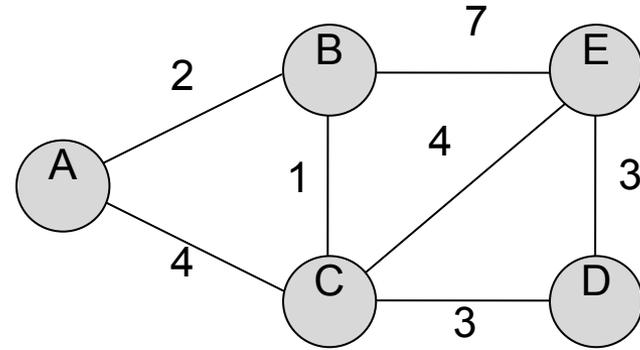
2.



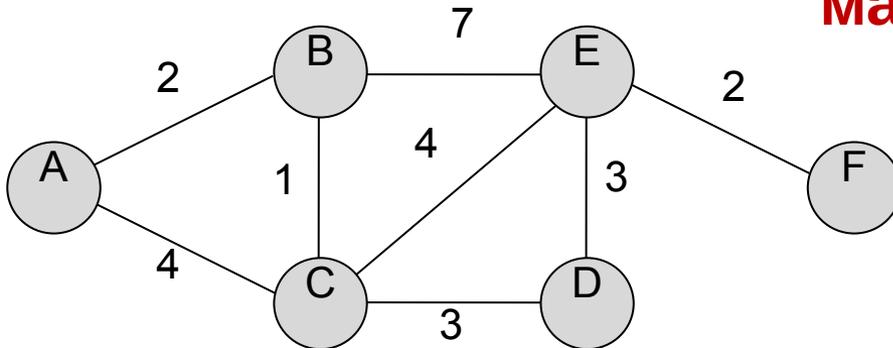
3.



4.



5.



Длина кратчайшего маршрута А-В-С-Е-Г равна 9

Задача

- Таблица стоимости перевозок устроена следующим образом: **числа, стоящие на пересечениях строк и столбцов таблиц, означают стоимость проезда между соответствующими соседними станциями.** Если пересечение строки и столбца пусто, то станции не являются соседними. **Укажите таблицу, для которой выполняется условие: «Минимальная стоимость проезда из А в В не больше 6».** Стоимость проезда по маршруту складывается из стоимостей проезда между соответствующими соседними станциями.

	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

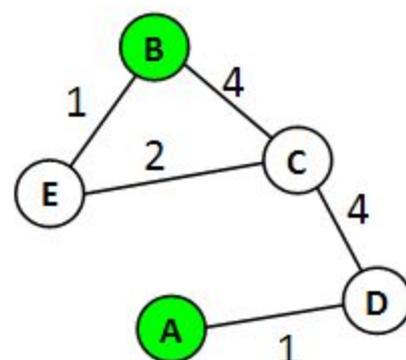
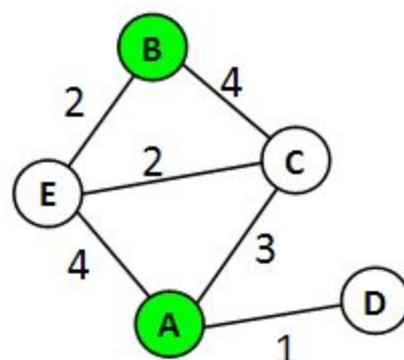
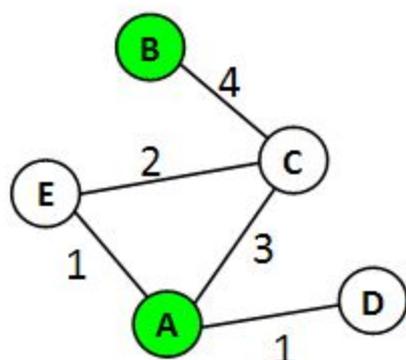
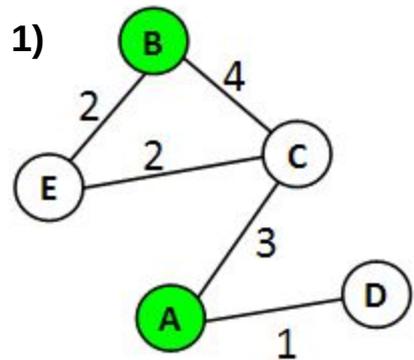
	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

	A	B	C	D	E
A			3	1	1
B			4		
C	3	4			2
D	1				
E	1		2		

	A	B	C	D	E
A			3	1	4
B			4		2
C	3	4			2
D	1				
E	4	2	2		

	A	B	C	D	E
A				1	
B			4		1
C		4		4	2
D	1		4		
E		1	2		



2) теперь по схемам определяем кратчайшие маршруты для каждой таблицы:

1: $A \xrightarrow{3} C \xrightarrow{4} B$ или $A \xrightarrow{3} C \xrightarrow{2} E \xrightarrow{2} B$, стоимость 7

2: $A \xrightarrow{3} C \xrightarrow{4} B$ или $A \xrightarrow{1} E \xrightarrow{2} C \xrightarrow{4} B$, стоимость 7

3: $A \xrightarrow{4} E \xrightarrow{2} B$, стоимость 6

4: $A \xrightarrow{1} D \xrightarrow{4} C \xrightarrow{2} E \xrightarrow{1} B$, стоимость 8

правильный ответ – 3.

Данные – это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

Для хранения данных используются **базы и хранилища данных.**

Знания – это закономерности, принципы, связи, законы предметной области, полученные в результате практической деятельности и профессионального опыта.

Для хранения знаний используются **базы знаний.**

Модель представления данных – множество элементов (объектов, типов данных) и связей (отношений) между ними, а также ограничений операций над типами данных и отношениями.

Основные модели представления данных в базах данных:

1. Иерархическая

2. Сетевая

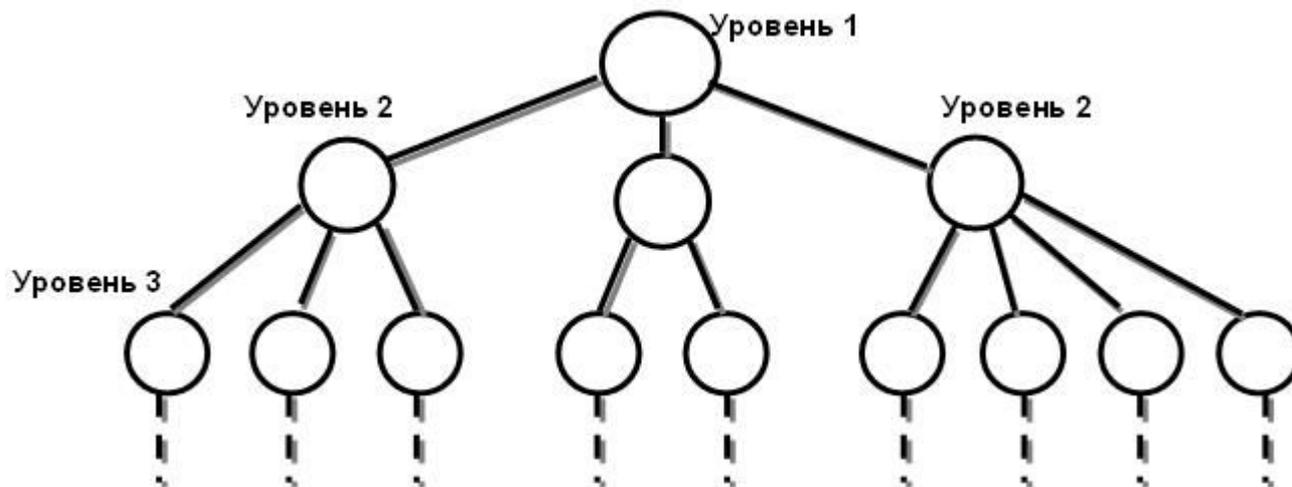
3. Реляционная

4. Многомерная

5. Объектно-ориентированная

1. Иерархическая модель

Иерархическая модель представляет собой ориентированный граф (перевернутое дерево) объектов, связанных иерархическими отношениями



К основным понятиям иерархической модели относятся: **уровень**, **элемент (узел)**, **связь**

Пример иерархической модели базы данных



Достоинства иерархической модели данных

→ Простота понимания

→ Простота оценки операционных характеристик

→ Хорошие временные показатели выполнения операций над данными

Недостатки иерархической модели данных

→ Структура данных задается на этапе проектирования БД и не может быть изменена при организации доступа к данным

→ Громоздкость модели для обработки информации со сложными логическими связями

→ Отношения М : М могут быть реализованы только искусственно

→ Возможны избыточные данные

→ Удаление исходных объектов ведет к удалению порожденных объектов

→ Доступ к любому порожденному узлу возможен только через корневой узел

→ Ограниченный набор структур запроса

Иерархическая БД

Иерархическая БД – это набор данных в виде многоуровневой структуры.

Прайс-лист:

Продавец (уровень 1)

Кей

Товар (уровень 2)

Мониторы

Принтеры

Изготовитель (уровень 3)

Sony

Phillips

Samsung

Модель (уровень 4)

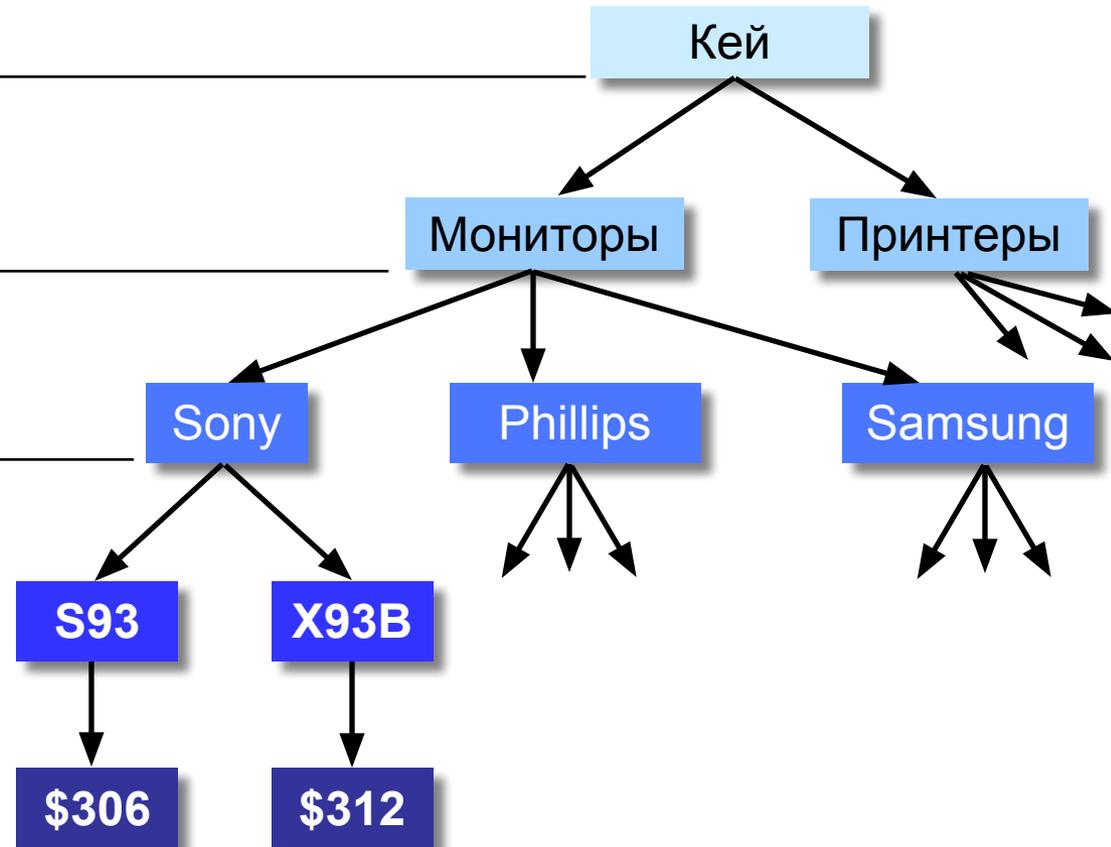
S93

X93B

Цена (уровень 5)

\$306

\$312



Приведение к табличной форме:

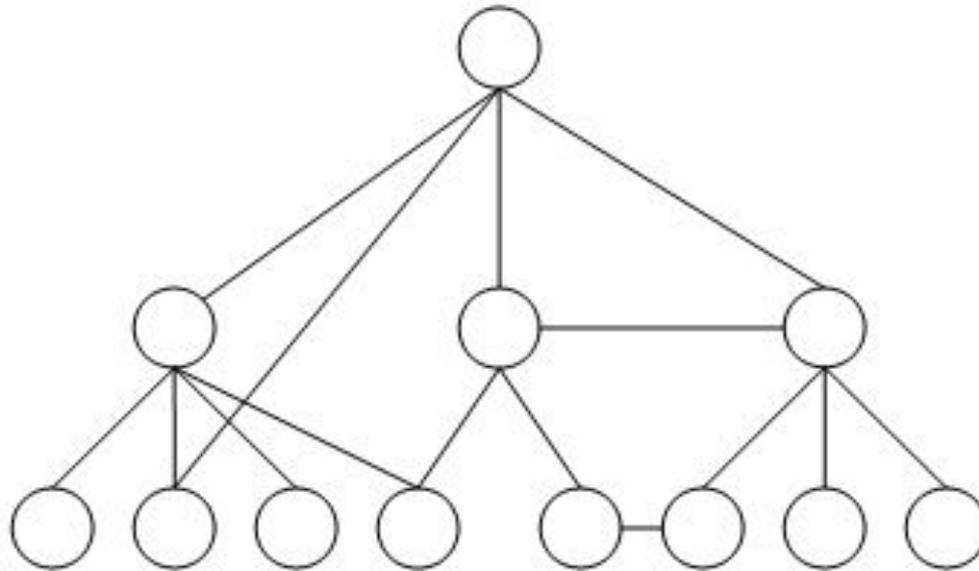
Продавец	Товар	Изготовитель	Модель	Цена
Кей	Монитор	Sony	S93	\$306
Кей	Монитор	Sony	X93B	\$312
Key	Монитор	Phillips	190 B5 CG	\$318
Кей	Монитор	Samsung	SyncMaster 193P	\$452
...				

-  1) дублирование данных
- 2) при изменении адреса фирмы надо менять его во всех строках
- 3) нет защиты от ошибок ввода оператора
(Кей – Key), лучше было бы выбирать из списка

СУБД, основанные на иерархической модели данных:
IMS, PC/Focus, Team-Up, Data Edge, Ока, ИНЭС, МИРИС.

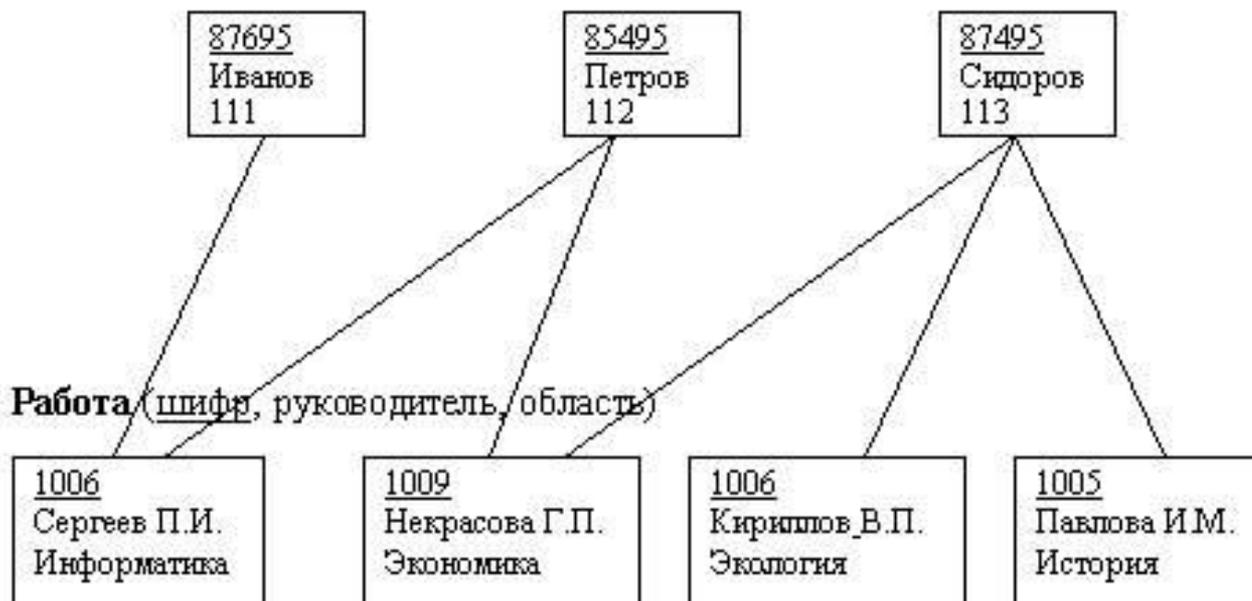
2. Сетевая модель

В **сетевой модели** при тех же основных понятиях (**уровень, узел, связь**) каждый элемент может быть связан с любым другим элементом, связанных иерархическими отношениями



Пример сетевой модели базы данных

Студент (номер зачетной книжки, фамилия, группа)



Достоинства сетевой модели данных:

→ Возможность эффективной реализации по показателям затрат памяти и оперативности

→ Сохранение информации при уничтожении владельца

→ Более богатая, чем в иерархической модели данных, структура запросов

Недостатки сетевой модели данных:

→ Структура данных задается на этапе проектирования БД и не может быть изменена при организации доступа к данным

→ Жесткость схемы базы данных, построенной на ее основе

→ Сложность структуры (для навигации в наборах и записях прикладной программист должен детально знать логическую структуру базы данных)

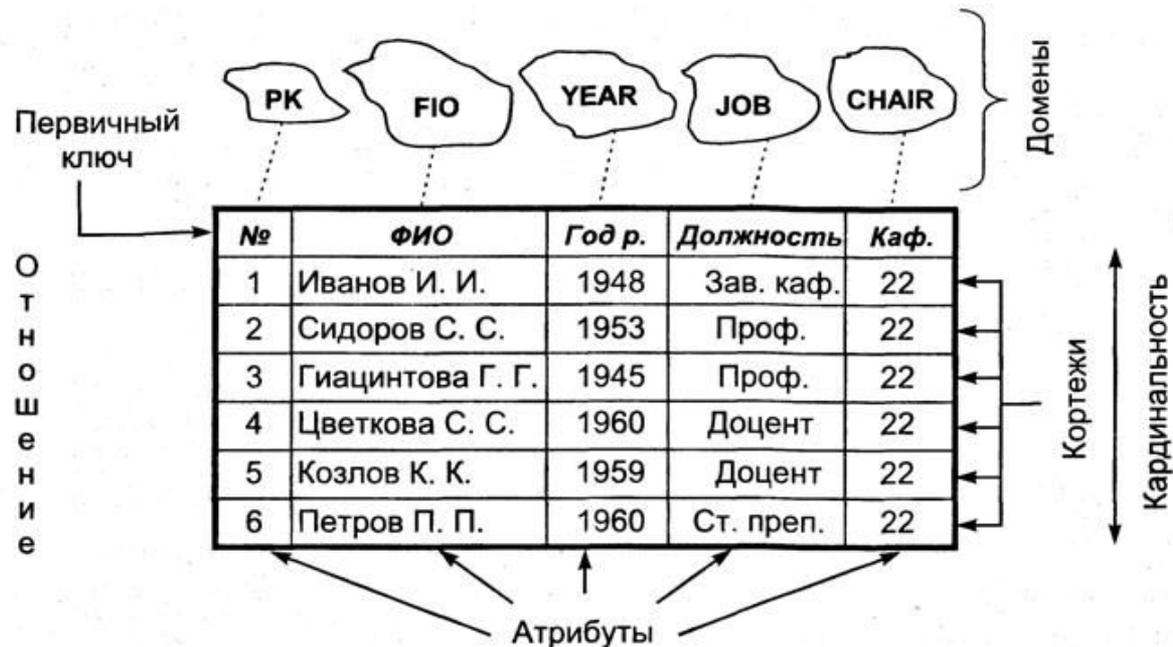
→ Возможна потеря независимости данных при реорганизации БД

→ Представление в прикладной программе сложнее, чем в иерархической модели данных

СУБД, основанные на сетевой модели данных:
IDMS, db_Vista III, СЕТЬ, СЕТОР, КОМПАС

3. Реляционная модель

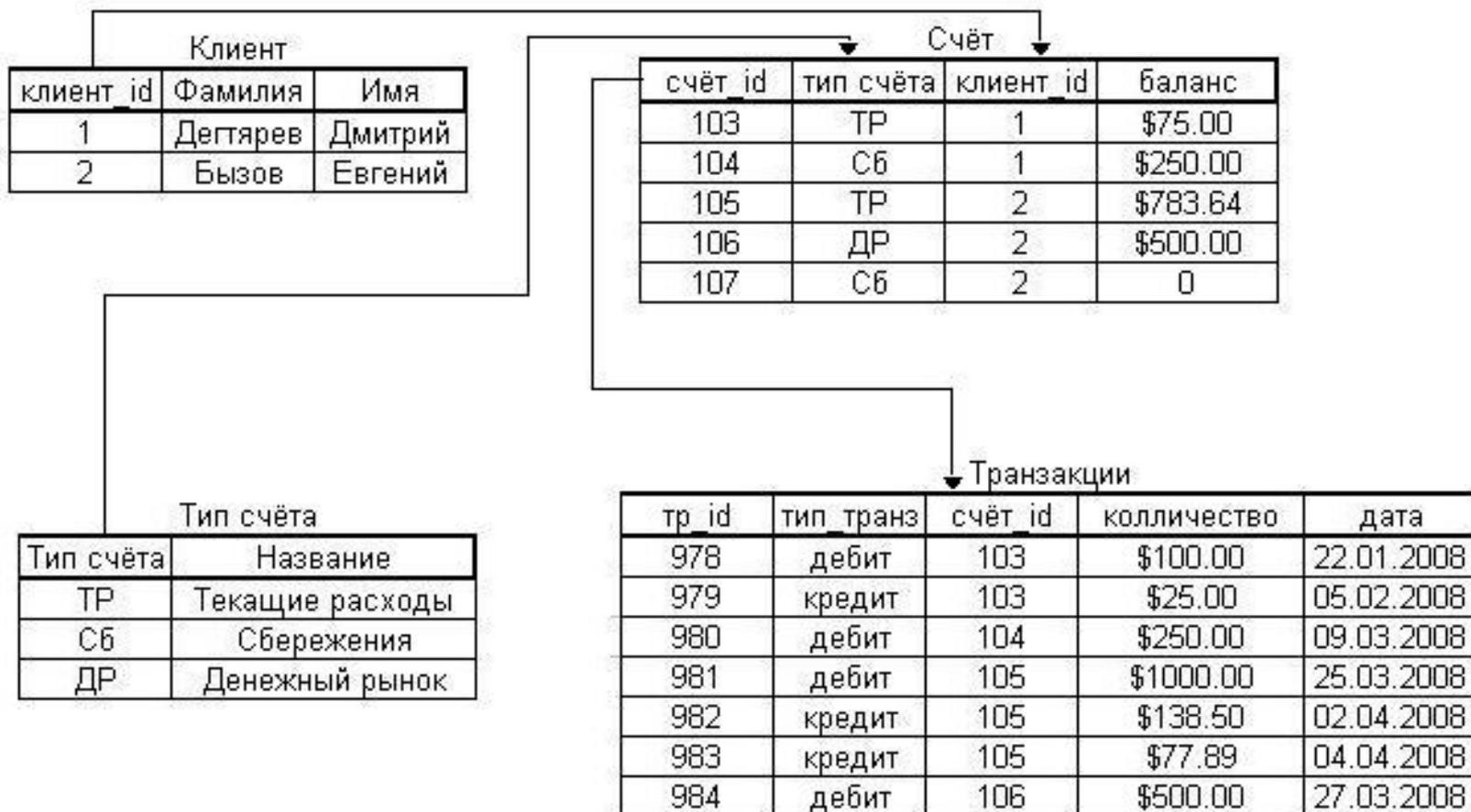
Реляционная модель представляет собой совокупность двумерных таблиц, связанных отношениями.



Элементы
реляционной
модели

К основным понятиям иерархической модели относятся: **тип данных, домен, атрибут, кортеж, отношение, первичный ключ**

Пример реляционной модели



Достоинства реляционной модели данных

- Простота работы и отражение представлений пользователя
- Гибкость (соединение, разделение файлов)
- Простота внедрения плоских файлов
- Отделение от физической реализации (независимость)
- Произвольная структура запросов
- Хорошее теоретическое обоснование

Недостатки реляционной модели данных

- Сложность структуры, вызванная процессом нормализации
- Низкая производительность из-за поиска по ключу (что в 3-5 раз увеличивает количество операций доступа)
- Ограниченный набор типов данных (например, отсутствуют форматы мультимедиа, геоинформации и т.д.)
- Недостаточное естественное представление данных (в виде плоских двумерных таблиц, а не таблиц со сложной структурой, как в сетевой модели данных)
- Невозможность рассмотрения данных послойно, на разных уровнях абстракции
- Невозможность определить набор операторов (методов), связанных с определенным типом данных (приходится задавать операции в конкретном приложении)

СУБД, основанные на реляционной модели данных:
Clipper, dBase, Paradox, FoxPro, Access, Oracle

Хранилище данных: предметно-ориентированный, интегрированный, неизменяемый и поддерживающий хронологию набор данных, предназначенный для обеспечения принятия управленческих решений.

Основные модели представления данных в хранилищах данных:

→ **1. Реляционная**

→ **2. Многомерная**

→ **3. Гибридная**

→ **4. Виртуальная**

Реляционная модель хранилищ данных

В основе реляционных хранилищ данных лежит разделение данных на две группы – **измерения** и **факты**.

Измерения – это категориальные атрибуты, наименования и свойства объектов, участвующих в некотором бизнес-процессе.

Примеры измерений: наименования товаров, названия фирм-поставщиков и покупателей, ФИО людей, названия городов и т. д.

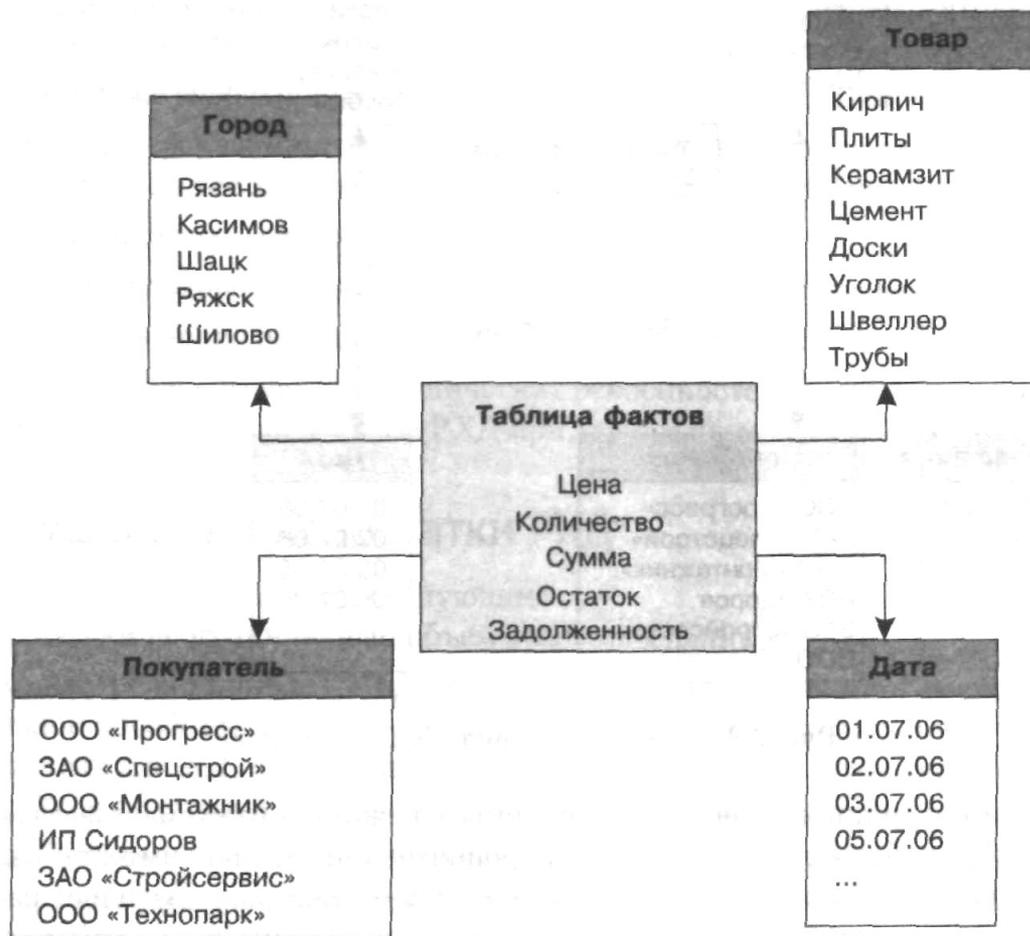
Измерения качественно описывают исследуемый бизнес-процесс.

Факты – это непрерывные по своему характеру данные (могут принимать бесконечное множество значений).

Примеры фактов: цена товара или изделия, их количество, сумма продаж или закупок, зарплата сотрудников, сумма кредита и т. д.

Факты количественно описывают бизнес-процесс.

Схема построения реляционного хранилища данных «звезда»



Центральной является **таблица фактов** (Fact table), с которой связаны **таблицы измерений** (Dimension tables).

Преимущества схемы «звезда»:

→ **простота и логическая прозрачность модели**

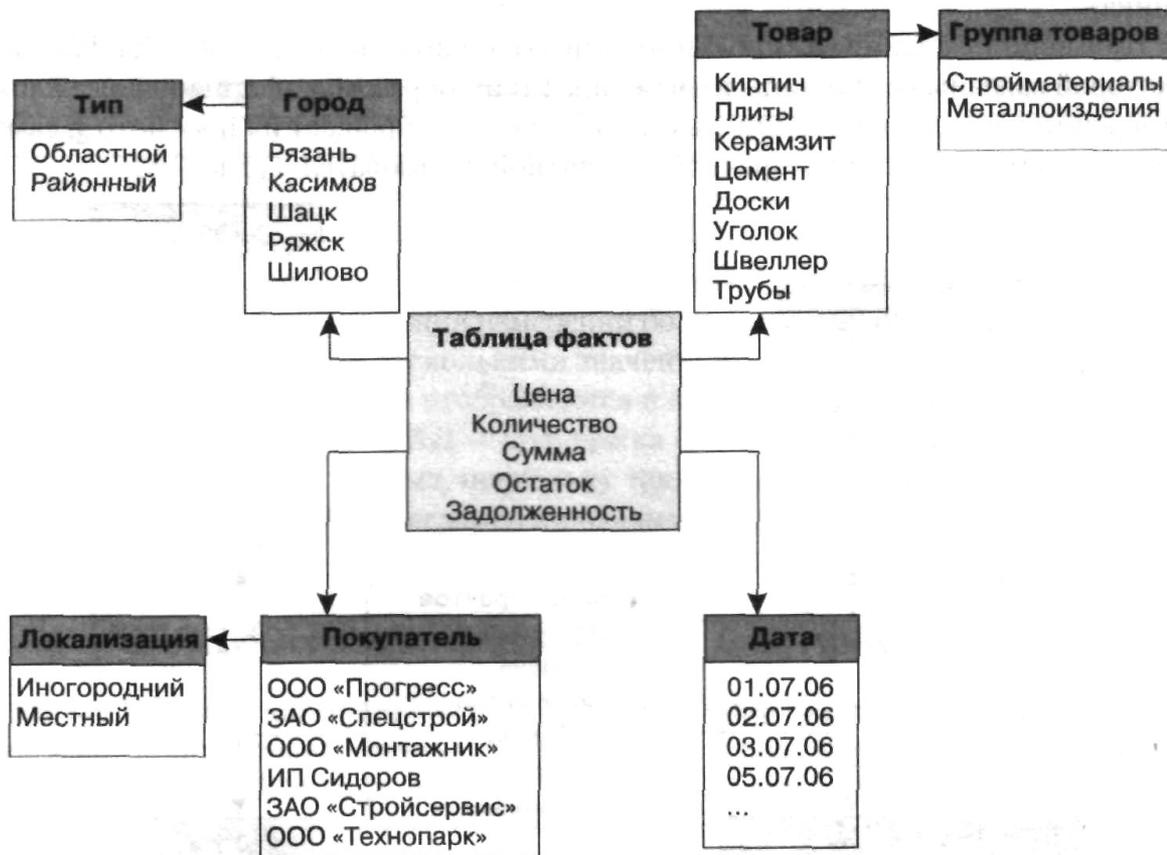
→ **более простая процедура пополнения измерений, поскольку приходится работать только с одной таблицей**

Недостатки схемы «звезда»:

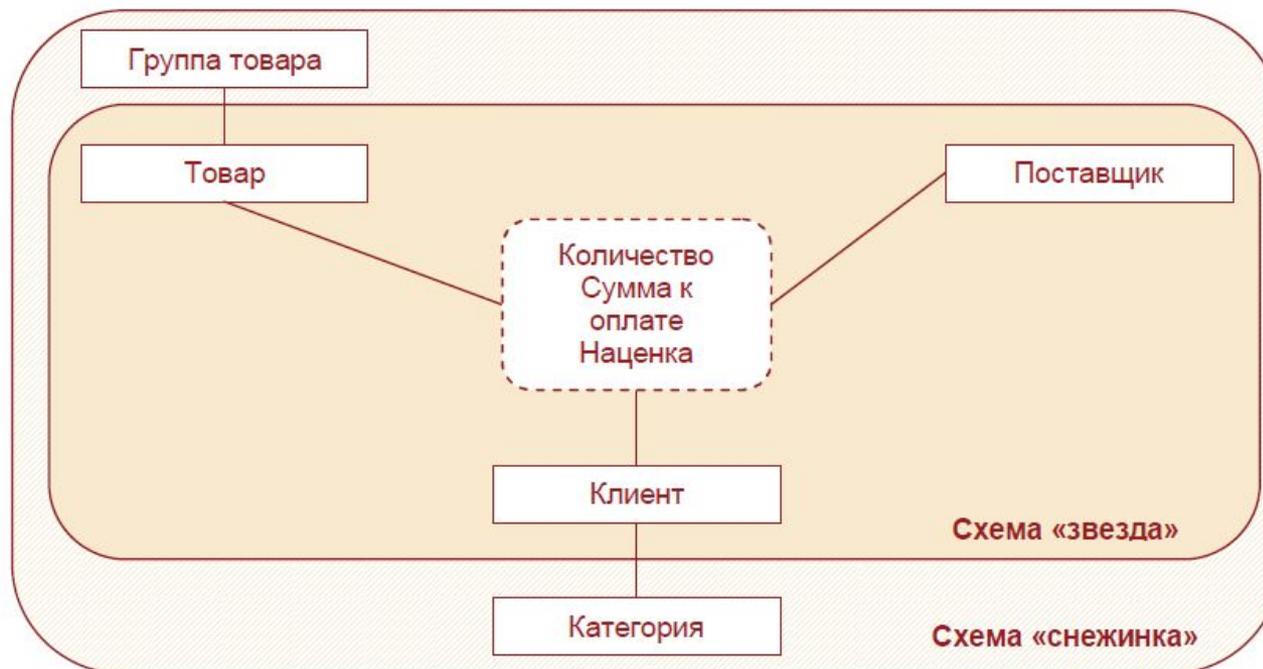
→ **медленная обработка измерений, поскольку одни и те же значения измерений могут встречаться несколько раз в одной и той же таблице**

→ **высокая вероятность возникновения несоответствий в данных (в частности, противоречий), например, из-за ошибок ввода**

Схема построения реляционного хранилища данных «снежинка» (модификация схемы «звезда»)



Основное функциональное отличие схемы «снежинка» от схемы «звезда» — это возможность работы с иерархическими уровнями, определяющими степень детализации данных.



Преимущества схемы «снежинка»:

→ она ближе к представлению данных в многомерной модели

→ процедура загрузки из РХД в многомерные структуры более эффективна и проста, поскольку загрузка производится из отдельных таблиц

→ намного ниже вероятность появления ошибок, несоответствия данных

→ большая, по сравнению со схемой «звезда», компактность представления данных, поскольку все значения измерений упоминаются только один раз

Недостатки схемы «снежинка»:

→ достаточно сложная для реализации и понимания структура данных

→ усложненная процедура добавления значений измерений

Преимущества реляционных хранилищ данных:

→ **Практически неограниченный объем хранимых данных**

→ **Поскольку реляционные СУБД лежат в основе построения многих систем оперативной обработки (OLTP), которые обычно являются главными источниками данных для ХД, использование реляционной модели позволяет упростить процедуру загрузки и интеграции данных в хранилище**

→ **При добавлении новых измерений данных нет необходимости выполнять сложную физическую реорганизацию хранилища, в отличие, например, от многомерных ХД**

→ **Обеспечиваются высокий уровень защиты данных и широкие возможности разграничения прав доступа**

Главный недостаток реляционных хранилищ данных:

→ **При использовании высокого уровня обобщения данных и иерархичности измерений в таких хранилищах начинают «размножаться» таблицы агрегатов. В результате скорость выполнения запросов реляционным хранилищем замедляется**

4. Многомерная модель

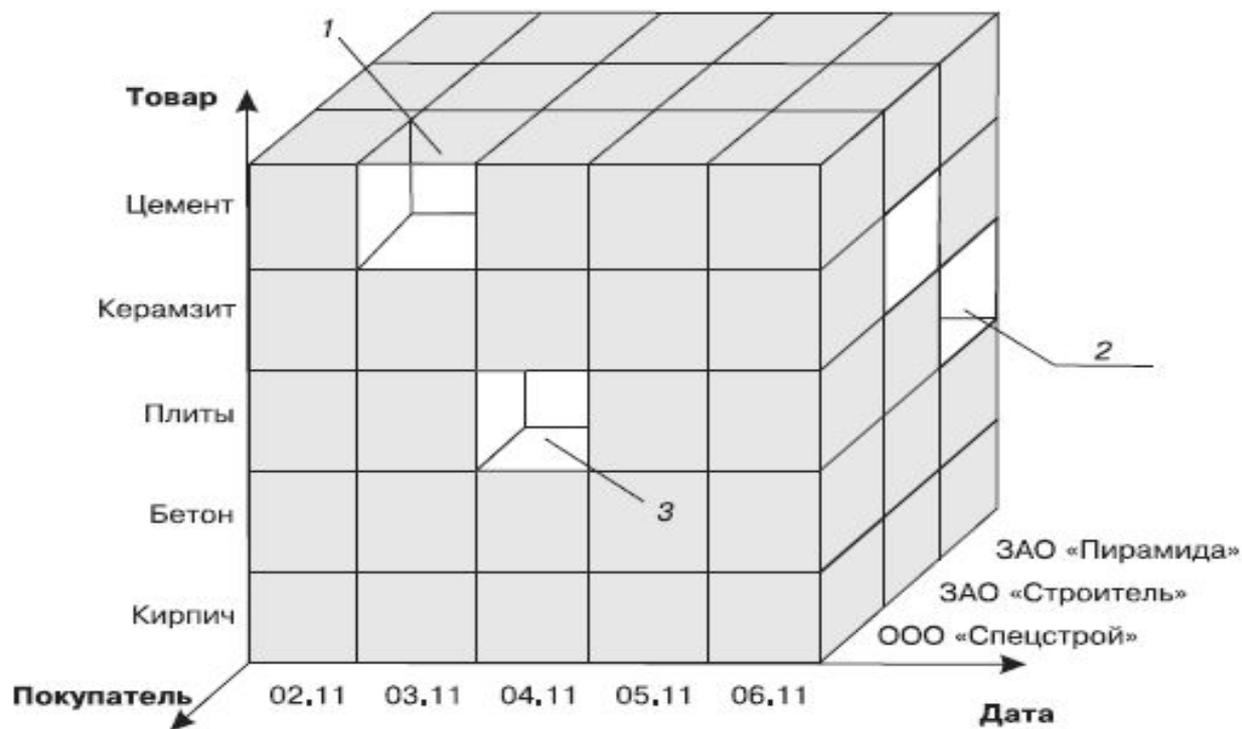
В основе многомерного представления данных лежит их разделение на две группы – **измерения** и **факты**

Многомерная модель данных реализуется с помощью **многомерных кубов**

Измерения – это категориальные атрибуты, наименования и свойства объектов, участвующих в некотором бизнес-процессе (**наименования товаров, названия фирм-поставщиков и покупателей, ФИО людей, названия городов** и т. д.)

Факты – это данные, количественно описывающие бизнес-процесс, непрерывные по своему характеру, то есть они могут принимать бесконечное множество значений (**цена товара или изделия, их количество, сумма продаж или закупок, зарплата сотрудников, сумма кредита, страховое вознаграждение** и т. д.)

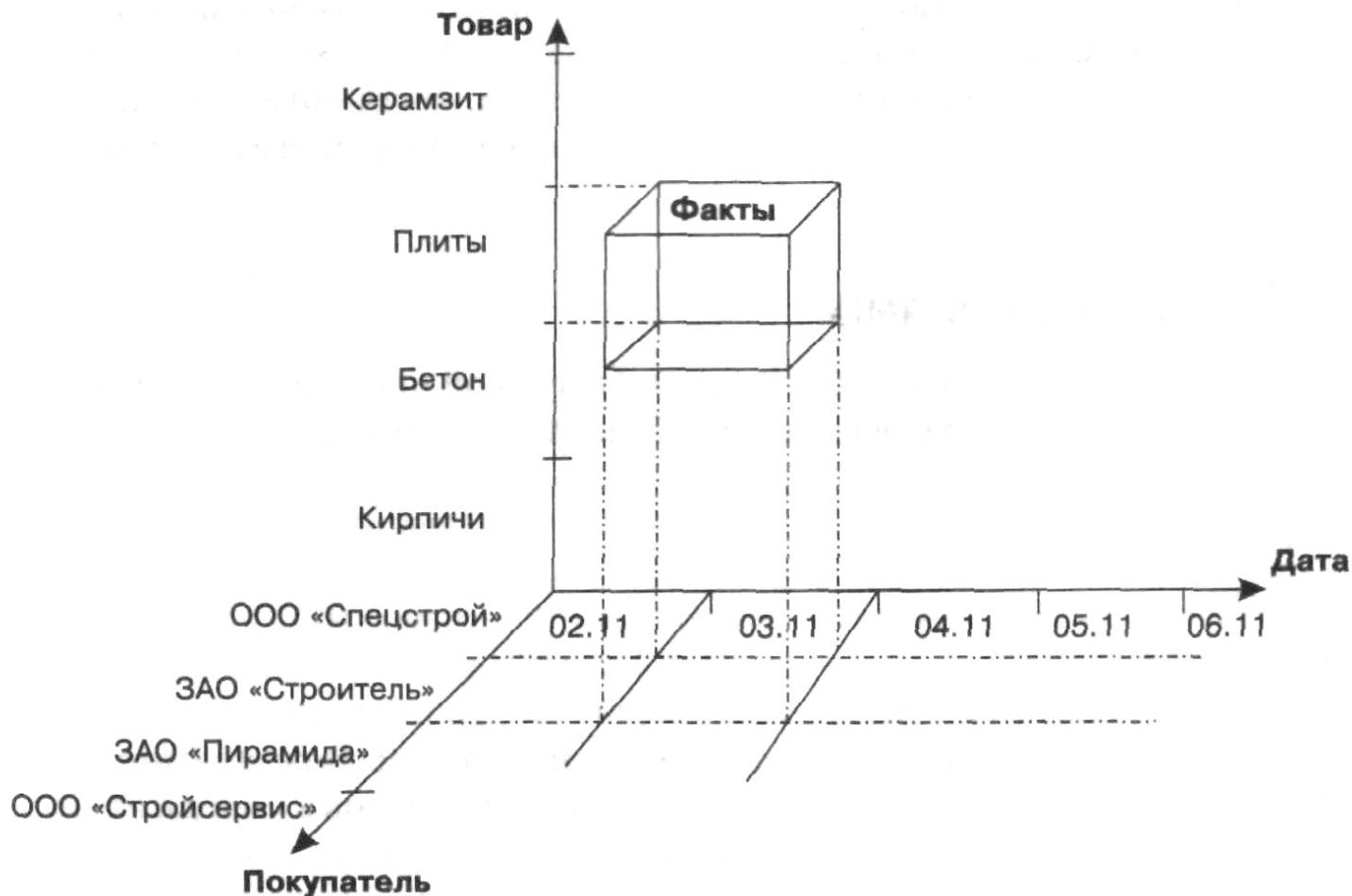
Многомерный куб можно рассматривать как систему координат, осями которой являются измерения (например, Дата, Товар, Покупатель). По осям будут откладываться значения измерений



В ячейке 1 будут располагаться факты, относящиеся к продаже цемента ООО «Спецстрой» 3 ноября, в ячейке 2 – к продаже плит ЗАО «Пирамида» 6 ноября, а в ячейке 3 – к продаже плит ООО «Спецстрой» 4 ноября.

Многомерный взгляд на измерения *Дата*, *Товар* и *Покупатель*

Выделенный сегмент будет содержать информацию о том, сколько плит, на какую сумму и по какой цене приобрела фирма ЗАО «Строитель» 3 ноября.





Из многомерного куба может быть составлен обычный плоский отчёт. По столбикам и строчкам отчёта будут бизнес-категории (границы куба), а в ячейках показатели.

		Январь	Февраль	Март	
Москва	Яблоки	20 000	21 000	22 000	63 000
	Груши	25 000	27 000	29 000	81 000
	Киви	30 000	33 000	36 000	99 000
	Итого	75 000	81 000	87 000	243 000
Пермь	Яблоки	18 000	19 000	20 000	57 000
	Груши	20 000	22 000	24 000	66 000
	Киви	14 000	17 000	20 000	51 000
	Итого	52 000	58 000	64 000	174 000
Уфа	Яблоки	12 000	13 000	14 000	39 000
	Груши	18 000	20 000	22 000	60 000
	Киви	22 000	25 000	28 000	75 000
	Итого	52 000	58 000	64 000	174 000
Итого		179 000	197 000	215 000	591 000

Преимущества многомерного подхода:

→ Представление данных в виде многомерных кубов более наглядно, чем совокупность нормализованных таблиц реляционной модели, структуру которой представляет только администратор БД

→ Возможности построения аналитических запросов к системе, использующей МХД, более широки

→ В некоторых случаях использование многомерной модели позволяет значительно уменьшить продолжительность поиска в МХД, обеспечивая выполнение аналитических запросов практически в режиме реального времени

Недостатки использования многомерной модели:

→ Для ее реализации требуется большой объем памяти

→ Многомерная структура труднее поддается модификации

Системы, поддерживающие многомерную модель данных:

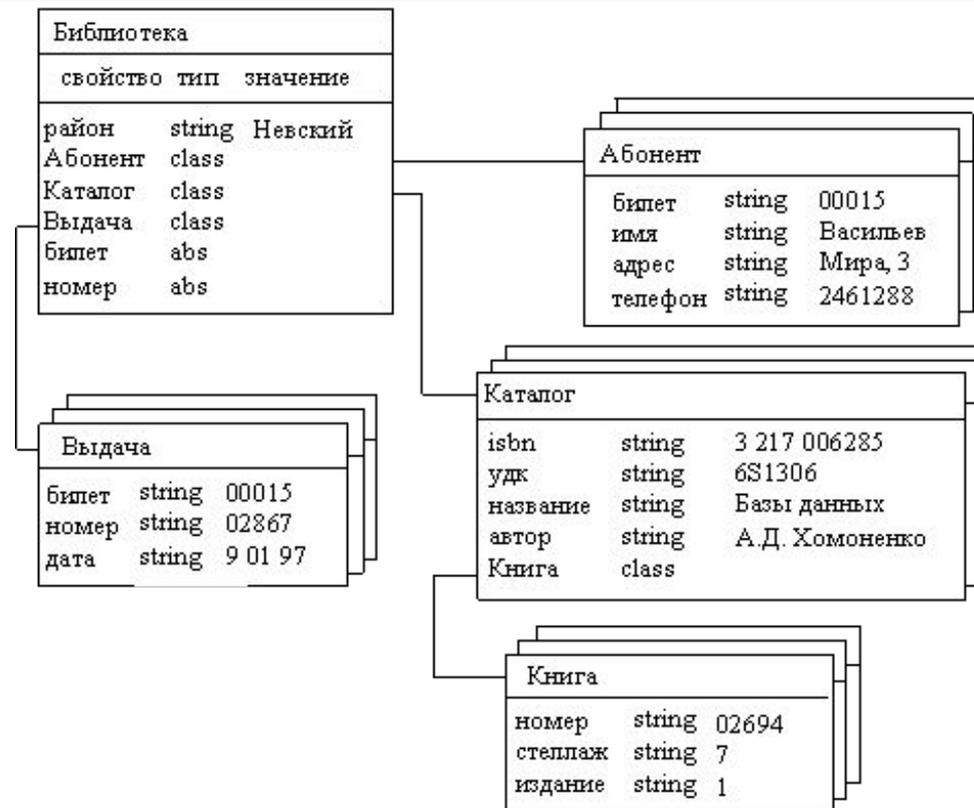
Essbase, Media Multi-matrix, Oracle Express Server, Cache.

Многие программные продукты позволяют одновременно работать с многомерными и с реляционными БД.

5. Объектно-ориентированная модель

ООМ графически представима в виде дерева, узлами которого являются **объекты**. Свойства объектов описываются некоторым стандартным типом или типом, конструируемым пользователем (определяется как *Class*).

К основным понятиям объектно-ориентированной модели относятся: **объект, линии поведения, сообщения, класс, отношения.**



Объекты являются моделями, очень близкими по своим свойствам и характеристикам объектам реального мира.

Объекты характеризуются свойствами, определяющими их состояние, и методами, определяющими их поведение. Объекты взаимодействуют друг с другом путем передачи сообщений, активизирующих их линии поведения.

Компоненты объектно-ориентированной модели:

- ❑ **Объект** – любая сущность реального мира. Объекты характеризуются свойствами, определяющими их состояние, и методами, определяющими их поведение. Объекты взаимодействуют друг с другом путем передачи сообщений.
- ❑ **Линии поведения** – это методы, или операции, которые объект может реализовать.
- ❑ **Сообщения** – это действие одного объекта, запускающее определенное поведение другого объекта.
- ❑ **Класс** – это способ группирования объектов, имеющих одинаковые наборы атрибутов и линии поведения, в шаблон. Объекты определенного класса называются экземплярами этого класса.
- ❑ **Отношения** описывают то, как объекты ассоциированы друг с другом.

Реализация объектно-ориентированного подхода характеризуется следующими ключевыми свойствами объектов:

→ **Инкапсуляция** ограничивает область видимости имени свойства пределами того объекта, в котором оно определено.

→ **Наследование**, наоборот, распространяет область видимости свойства на всех потомков объекта.

→ **Полиморфизм** означает способность одного и того же программного кода работать с разнотипными данными. Другими словами, он означает допустимость в объектах разных типов иметь методы (процедуры или функции) с одинаковыми именами. Во время выполнения объектной программы одни и те же методы оперируют с разными объектами в зависимости от типа аргумента.

Достоинства объектно-ориентированной модели данных:

→ В сравнении с реляционной у этой модели есть возможность отображения информации о сложных взаимосвязях объектов.

→ Позволяет идентифицировать отдельную запись базы данных и определять функции их обработки

Недостатки объектно-ориентированной модели данных:

→ Высокая понятийная сложность

→ Неудобство обработки данных

→ Низкая скорость выполнения запросов

СУБД, основанные на объектно-ориентированной модели данных:
POET, Jasmine, Versant, Iris, Orion, Postgres.