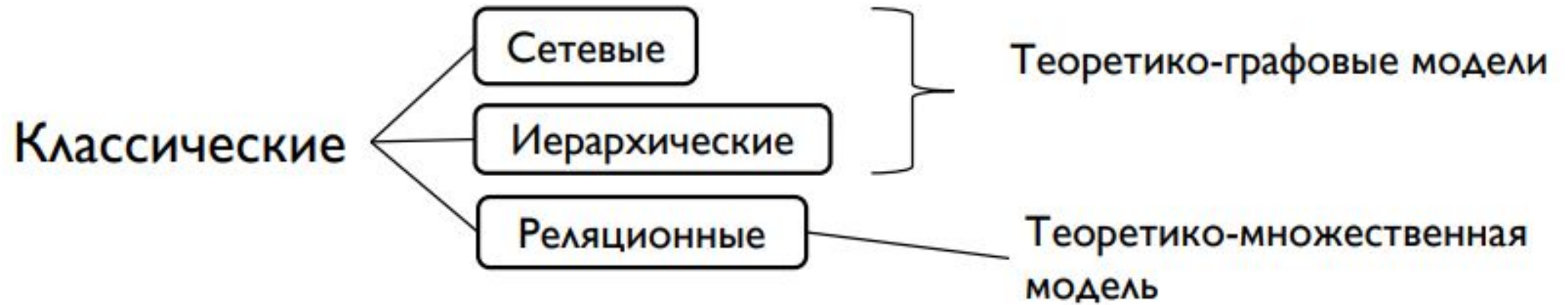


ТИПЫ МОДЕЛЕЙ ДАННЫХ

Модели данных

- Иерархическая модель данных (ИМД).
- Сетевая модель данных (СМД).
- Реляционная модель данных (РМД).
- Объектно-реляционная модель данных (ОРМД).
Стандарт SQL-3 (SQL-2003). Oracle (с версии 8.0), DB2, Informix, PostgreSQL, SQL Server 2008 и др.)
- Объектно-ориентированная модель данных (ООМД). O2, GemStone, Iris и др.
Стандарт ODMG 3.0 (Object Database Management Group).
- Многомерные базы данных.
- Поточковые базы данных.
- ...

Классические модели данных



Модель данных – это совокупность правил порождения структур данных в базе данных, операций над ними, а также ограничений целостности, определяющих допустимые связи и значения данных, последовательность их изменения [ГОСТ 20886-85].

Модель данных состоит из трёх частей:

1. Набор типов структур данных

Здесь можно провести аналогию с языками программирования, в которых тоже есть predetermined типы структур данных, такие как скалярные данные, векторы, массивы, структуры (например, тип *struct* в языке Си) и т.д.

2. Набор операторов или правил вывода, которые могут быть применены к любым правильным примерам типов данных, перечисленных в (1), чтобы находить, выводить или преобразовывать информацию, содержащуюся в любых частях этих структур в любых комбинациях.

3. Набор общих правил целостности, которые прямо или косвенно определяют множество непротиворечивых состояний базы данных и/или множество изменений её состояния.

Иерархическая модель данных

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

Между объектами существуют связи. Каждый объект может включать в себя несколько объектов низкого уровня. Такие объекты находятся в отношении **предка** (объект более близкий к корню) к **потомку** (объект более низкого уровня). При этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда, как у объекта-потомка обязательно только один предок.

Иерархическая модель данных

(ИМД)

В 1968 году была введена в эксплуатацию первая промышленная СУБД система IMS фирмы IBM. Это была первая иерархически база данных благодаря которой определили ряд фундаментальных понятий в теории систем баз данных.

Иерархическая модель позволяет строить БД с иерархической древовидной структурой. В основе ИМД лежит понятие дерева.

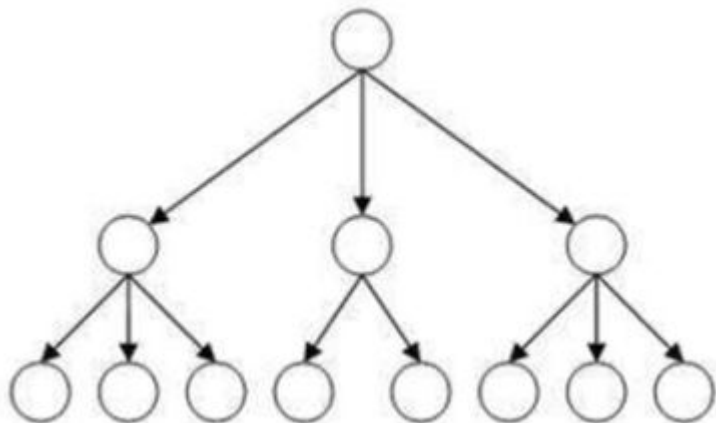
Дерево – это связный неориентированный граф, который не содержит циклов. При работе с деревом выделяют какую-то конкретную вершину, определяют её как корень дерева и рассматривают особо – в эту вершину не заходит ни одно ребро. В этом случае дерево становится ориентированным, ориентация определяется от корня. Дерево как ориентированный граф определяется так:

- имеется единственная особая вершина, называемая корнем, в которую не заходит ни одно ребро;
- во все остальные вершины заходит только одно ребро, а исходит произвольное количество ребер;
- граф не содержит циклов.

Конечные вершины, то есть вершины, из которых не выходит ни одной дуги, называются *листьями* дерева.

Количество вершин на пути от корня к листьям в разных ветвях дерева может быть различным.

Графическая диаграмма концептуальной схемы базы данных называется *деревом определения*.



Пример иерархической базы данных:



Иерархическая структура должна удовлетворять следующим требованиям:

- каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне;
- существует только один корневой узел на самом верхнем уровне, не подчиненный никакому другому узлу;
- к каждому узлу существует ровно один путь от корневого узла.

Достоинства ИМД

- Простота
- Минимальный расход памяти и неплохие показатели временных затрат на выполнение операций над данными

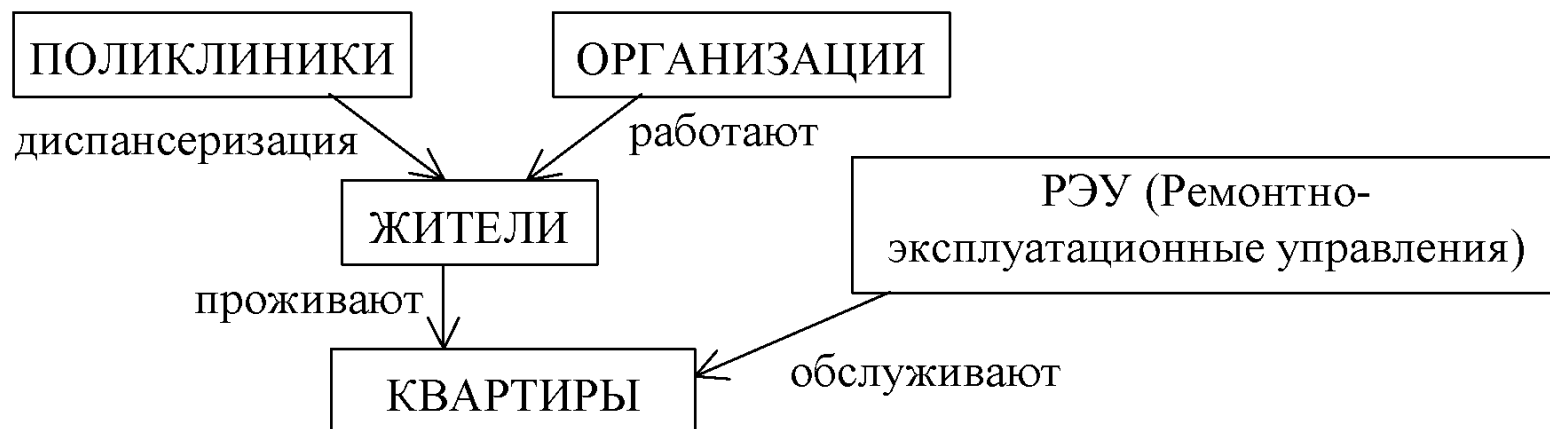
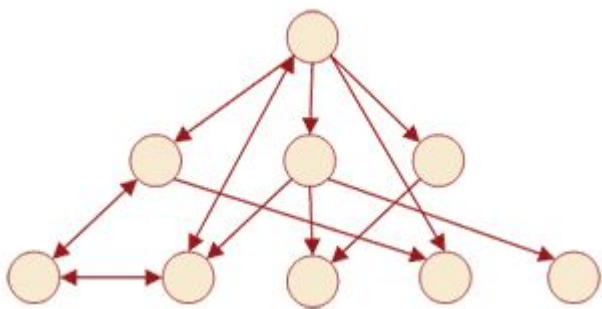
Недостатки ИМД

- Громоздкость модели для обработки информации со сложными логическими связями
- Отношение М:М может быть реализовано только искусственно
- Возможны избыточные данные
- Структура данных задается на этапе проектирования БД и не может быть изменена при организации доступа к данным
- Отсутствие универсальности: не всякую информацию можно выразить в ИМД
- Исключительно навигационный принцип доступа к данным
- Доступ к данным только через корневой узел
- Сложность внесения изменений
- Удаление исходных объектов ведет к удалению порожденных объектов
- Ограниченный набор структур запроса

Сетевая модель данных (СМД)

Сетевая модель (произвольный граф) позволяет организовывать БД, структура которых представляется графом общего вида. Это структура, у которой любой элемент может быть связан с любым другим.

Связи между записями в СМД выполняются в виде указателей, т.е. каждая запись хранит ссылку на другую однотипную запись (или признак конца списка) и ссылки на списки подчинённых записей, связанных с ней групповыми отношениями.



Цель разработчиков сетевой модели – создание модели, позволяющей описывать связи M:N, чтобы одна запись могла участвовать в нескольких отношениях предок/потомок.

Структура сетевой базы данных основана на следующих правилах.

1. База данных содержит любое количество типов записей и типов наборов.
2. Между двумя типами записей может быть определено любое количество типов наборов.
3. Тип записи может быть владельцем и одновременно членом нескольких типов наборов.

На рисунке представлены три типа записей: «Отдел», «Служащие» и «Руководитель» и три типа связей (три набора): «Состоит из служащих», «Имеет руководителя» и «Является служащим».



Рис. 1.20. Записи и связи в сетевой модели

Пример схемы сетевой базы данных приведен на рис. 1.21.



Достоинства и недостатки сетевой модели данных

СМД является **наиболее полной** с точки зрения реализации различных типов связей и ограничений целостности, но она является достаточно **сложной** для проектирования и поддержки. В этой модели **не обеспечивается физическая независимость данных**, т.к. наборы организованы с помощью физических ссылок. Также в СМД **не обеспечивается независимость данных от программ**. Из-за этих недостатков эта модель не получила широкого распространения.

Реляционная модель данных

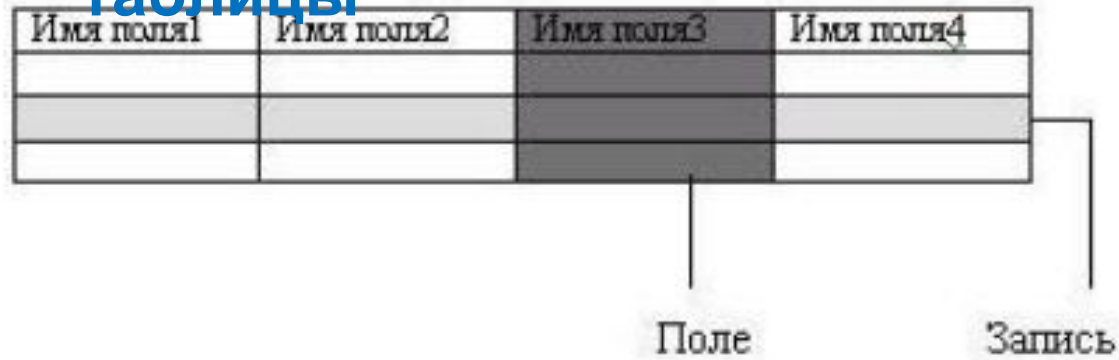
Реляционная модель данных использует организацию данных в виде двумерных таблиц. Каждая такая таблица, называемая реляционной таблицей или отношением, представляет собой двумерный массив и обладает следующими свойствами:

- все столбцы в таблице однородные, т.е. все элементы в одном столбце имеют одинаковый тип и максимально допустимый размер;
 - каждый столбец имеет уникальное имя;
 - одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись. Поле (столбец реляционной таблицы) – элементарная единица логической организации данных, которая соответствует конкретному атрибуту информационного объекта.

Запись (строка реляционной таблицы) – совокупность логически связанных полей, соответствующая конкретному экземпляру информационного объекта.

Основные структурные элементы таблицы



Например, в виде реляционной таблицы можно представить информацию о сотрудниках.

Таб номер	ФИО	Пол	Дата рождения	Должность
003	Иванов И.И.	М	04.12.1989	прораб
123	Петров П.П.	М	14.05.1986	бухгалтер
563	Сидорова С.С.	Ж	23.02.1974	гл. бухгалтер
432	Антонова А.А.	Ж	17.06.1955	директор
111	Федоров Ф.Ф.	М	22.04.1964	зам. директора

Реляционная модель предложена Э. Коддом (E. Kodd, математик, сотрудник IBM, 1970 г., статья «A Relational Model of Data for Large Shared Data Banks») на основе теории отношений (relation) и опирается на систему понятий, важнейшими из которых являются тип данных, домен, атрибут, кортеж, первичный и внешний ключ. Принципы, используемые в реляционной модели, вытекают из понятия n -арного отношения, представляющего собой подмножество декартового произведения. Факторы, обеспечившие быстрое распространение реляционной модели: с прагматической точки зрения база данных представляется в виде двумерных таблиц (отношений), обработка в которых не зависит от организации хранения данных в памяти; с математической точки зрения реляционная база данных – конечный набор отношений различной арности, являющихся областью приложений математической логики, теории множеств и общей алгебры; замкнутость реляционной модели (операции над отношениями дают отношение) обеспечивает основу для интерпретации выводимости, избыточности и непротиворечивости данных; наличие небольшого набора абстракций, которые позволяют сравнительно просто моделировать большую часть распространенных предметных областей и допускают точные формальные определения.

Достоинства и недостатки реляционной модели

Достоинства РМД

- Эта модель данных отображает информацию в наиболее простой для пользователя форме.
- Основана на развитом математическом аппарате, который позволяет достаточно лаконично описать основные операции над данными.
- Позволяет создавать языки манипулирования данными не процедурного типа.
- Манипулирование данными на уровне выходной БД и возможность изменения.

Недостатки РМД

- Далеко не всегда предметная область может быть представлена в виде "таблиц"
- В результате логического проектирования появляется множество "таблиц". Это приводит к трудности понимания структуры данных.
- БД занимает относительно много внешней памяти.
- Относительно низкая скорость доступа к данным.
- Не допускает представления объектов со сложной структурой, поскольку в ее рамках возможно моделирование лишь с помощью двумерных таблиц.

Объектно-ориентированная модель

данных

Объектно-ориентированная модель данных – представляет структуру, которую можно изобразить графически в виде дерева, узлами которого являются объекты. Поддерживается следующими СУБД: Cache, Oracle Express Server, Media Multi-matrix и Essbase.

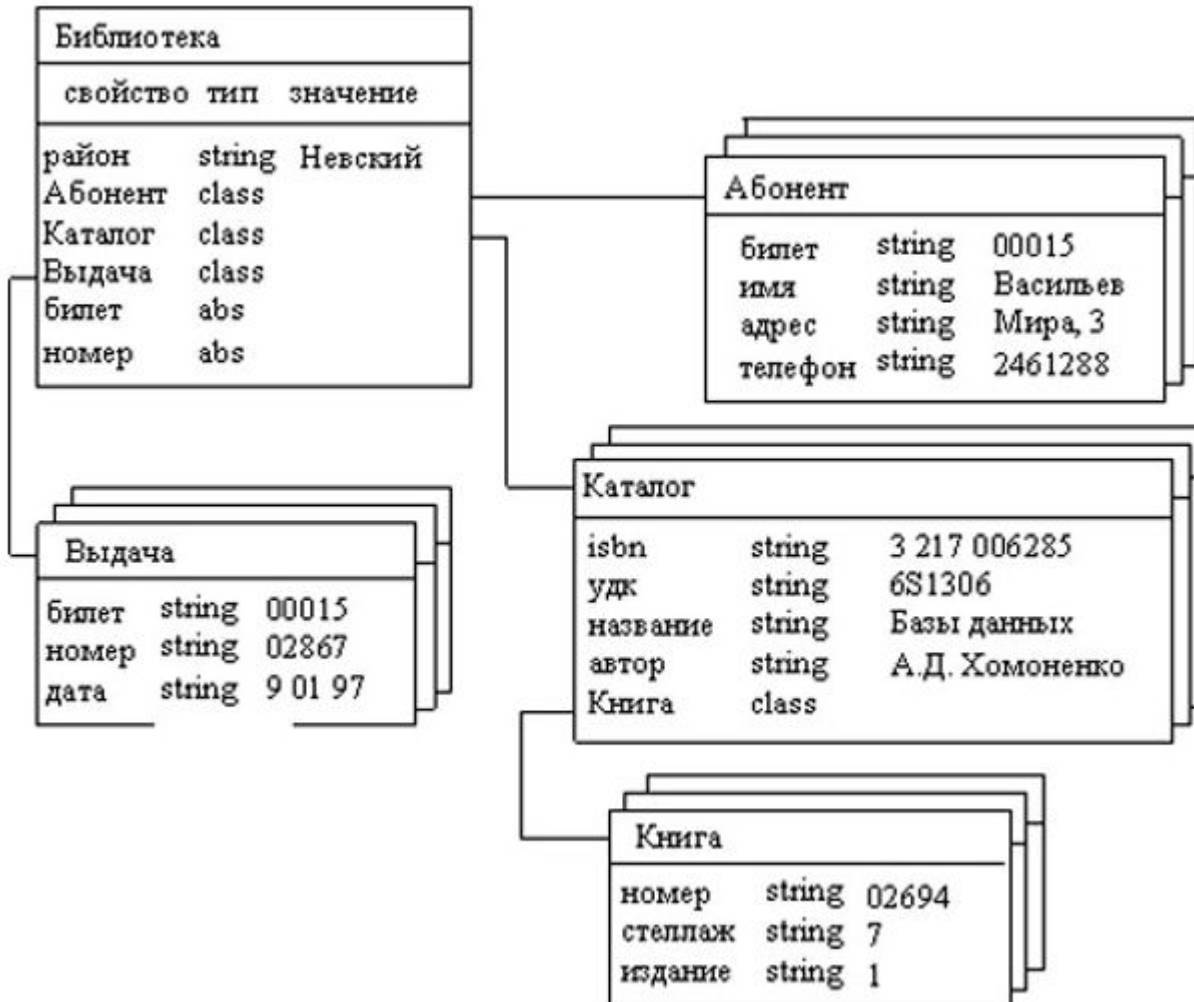
Объектно-реляционные модели данных (ОРМД) объединяют в себе черты реляционной и объектной моделей.

ОРМД реализована с помощью реляционных таблиц, но включает объекты, аналогичного понятию объекта в объектно-ориентированном программировании. Возникновение объектно-реляционной модели данных объясняется тем, что реляционные базы данных хорошо работают со встроенными типами данных и гораздо хуже с пользовательскими, нестандартными. Перестройка СУБД с целью включения в нее поддержки нового типа данных – не лучший выход из положения. Вместо этого объектно-реляционная СУБД позволяет загружать код, предназначенный для обработки «нетипичных» данных. Таким образом, база данных сохраняет свою табличную структуру, но способ обработки некоторых полей таблиц определяется извне, т. е. программистом

В объектно-реляционной модели данных используются такие объектно-ориентированные компоненты, как пользовательские типы данных, инкапсуляция, полиморфизм, наследование, переопределение методов и т.п. Категория объектно-реляционных СУБД позволяет: поддерживать сложные типы данных; вводить новые типы данных; наследовать объекты: типы данных, таблицы и пр.

Объектно-реляционные свойства имеют все широко известные СУБД, в том числе Oracle Database, Informix, DB2, PostgreSQL. К сожалению, до настоящего времени разработчики не пришли к единому мнению о том, что должна обеспечивать объектно-реляционная модель данных. В 1999 г. был принят стандарт SQL-99, а в 2003 г. вышел второй релиз этого стандарта, получивший название SQL-3, который определяет основные характеристики объектно-реляционной модели данных. Но до сих пор объектно-реляционные модели, поддерживаемые различными производителями СУБД, существенно отличаются друг от друга.

Стандартный тип (например, строковый – string) или тип, созданный пользователем (class), описывает свойства объектов.



На рисунке объект БИБЛИОТЕКА является родителем для объектов-экземпляров классов КАТАЛОГ, АБОНЕНТ и ВЫДАЧА. У разных объектов типа КНИГА может быть один или разные родители. У объектов типа КНИГА, которые имеют одного и того же родителя, должны быть по крайней мере разные инвентарные номера (уникальные для каждого экземпляра книги), но одинаковые значения свойств автор, название, удк и isbn. Логические структуры объектно-ориентированной и иерархической базы данных внешне похожи. Отличаются они в основном методами манипулирования данными.

Преимущества и недостатки объектно-ориентированной модели

- Основное **преимущество** объектно-ориентированной модели данных в отличие от реляционной модели состоит в возможности отображения информации о сложных взаимосвязях объектов. Рассматриваемая модель данных позволяет определять отдельную запись БД и функции ее обработки.
- Нерациональное использование дискового пространства.
- Недостаточная точность предоставления данных.

К **недостаткам** объектно-ориентированной модели относят

- высокую понятийную сложность,
- неудобную обработку данных
- низкую скорость выполнения запросов.

На сегодняшний день СУБД, поддерживающие ООМД достаточно широко распространены.

К ним относятся СУБД: Postgres, Orion, Iris, ODBJupiter, Versant, Objectivity /DB, ObjectStore, Statice, GemStone, G-Base.

Многомерная модель данных

Основное назначение многомерных хранилищ данных (МХД) – поддержка систем, ориентированных на аналитическую обработку данных, поскольку такие хранилища лучше справляются с выполнением сложных нерегламентированных запросов.

Многомерная модель данных, лежащая в основе построения многомерных хранилищ данных, опирается на концепцию многомерных кубов, или гиперкубов. Они представляют собой упорядоченные многомерные массивы, которые также часто называют OLAP-кубами.

Многомерные модели рассматривают данные либо как факты с соответствующими числовыми параметрами, либо как текстовые измерения, которые характеризуют эти факты.

В розничной торговле, например, покупка – это факт, объем покупки и стоимость – параметры, а тип приобретенного продукта, время и место покупки – измерения.

Запросы агрегируют значения параметров по всему диапазону измерения, и в итоге получают такие величины, как общий месячный объем продаж данного продукта.

Представление данных в виде гиперкуба

Куб – многомерная структура данных (многомерный массив, содержащий данные).

Измерение – одна из координат куба, параметр, по которому строится один из индексов многомерного массива.

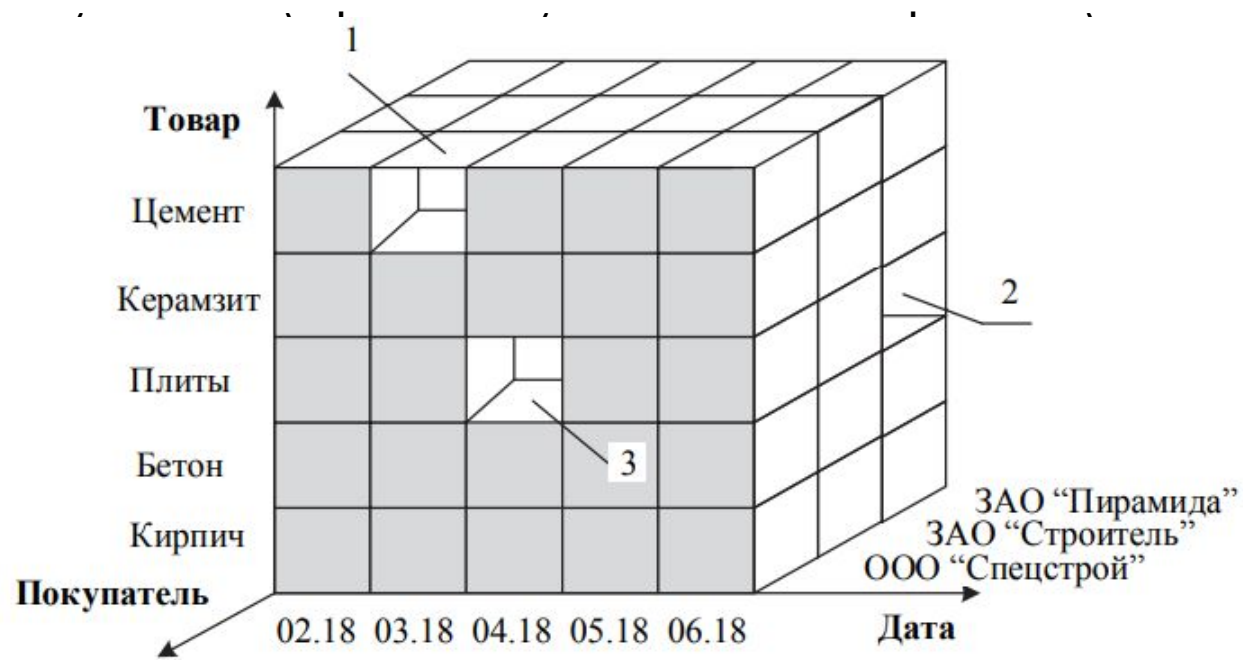
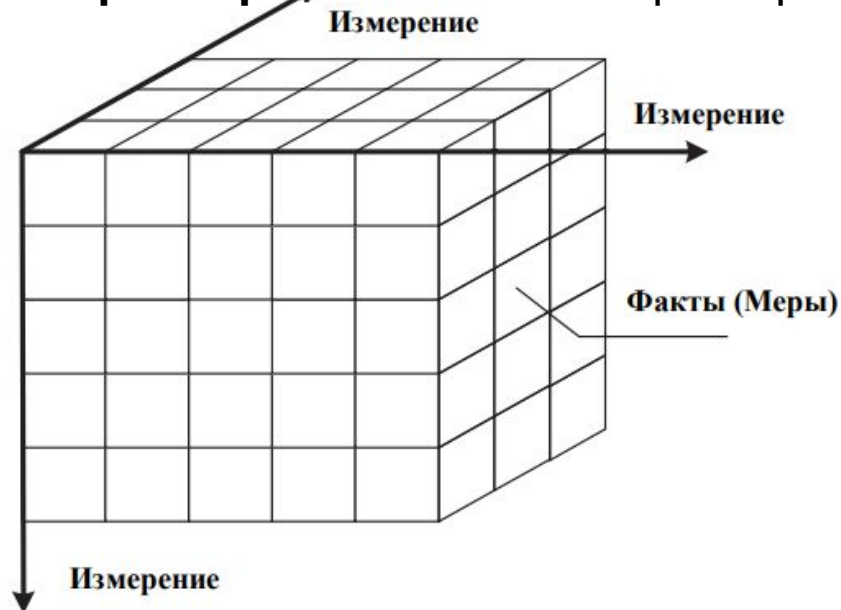
Мера – конкретное значение, находящееся в ячейке куба.

Факт – событие, которое надо проанализировать. Факты определяются комбинацией значений измерений. Факт существует, когда ячейка, соответствующая сочетанию измерений не пуста.

Виды фактов: - события (events) реального мира. Например, продажи, движение товара на складе и т.д.
- мгновенные снимки состояния объекта в данный момент (н-р, количество товара на складе, число пользователей сайта).

- совокупные мгновенные снимки – состояние объекта на протяжении времени (н-р, совокупный объем продаж)

Параметры – численная характеристика факта



ПРОСТЕЙШАЯ МНОГОМЕРНАЯ ТАБЛИЦА

	2010	2011	2012
Москва	900 000	850 000	810 000
Омск	78 000	85 000	73 000
Чита	5 500	4 700	4 100

Заголовки строк – города,
Заголовки столбцов – годы,
Содержимое (значения) таблицы – число зачисленных
абитуриентов в том или ином городе в тот или иной год

ТРЕХМЕРНАЯ ТАБЛИЦА

Заголовки первой оси – города,

Заголовки второй оси – годы,

Заголовки третьей оси (слои) – специальности,

Содержимое (значения) таблицы – число зачисленных абитуриентов в тот или иной год в том или ином городе на некоторую специальность

МО	2010	2011	2012
ФУ	2010	2011	2012
ФК	2010	2011	2012
Москва	9300	8530	10010
Омск	1900	1490	1120
Чита	60	47	61

Пример:

Куб: база, описывающая контингент;

Измерения: год, город, специальность,

Мера: количество зачисленных.

ФК	2010	2011	2012
Москва	9300	8530	10010
Омск	1900	1490	1120
Чита	60	47	61

ТРЕХМЕРНАЯ ТАБЛИЦА

Заголовки первой оси – города,

Заголовки второй оси – годы,

Заголовки третьей оси (слои) – специальности,

Содержимое (значения) таблицы – число зачисленных абитуриентов в тот или иной год в том или ином городе на некоторую специальность

	2010	2011	2012
ФК	9300	8530	10010
Москва	1900	1490	1120
Омск	60	47	61
Чита			

		C			
		Модель	Апрель	Май	Июнь
B		ВАЗ Калина	9	11	5
		Модель	Апрель	Май	Июнь
		ВАЗ Калина	12	8	3
A		Модель	Апрель	Май	Июнь
		ВАЗ Калина	10	15	4
		Ford focus	10	N	18
		Renoult duster	N	N	20
		Scoda rapid	5	N	N

Трехмерная модель данных

Достоинства и недостатки многомерной модели данных

Основным **достоинством** многомерной модели данных является удобство и эффективность аналитической обработки больших объемов данных, связанных со временем. При организации обработки аналогичных данных на основе реляционной модели происходит нелинейный рост трудоемкости операций в зависимости от размерности БД и существенное увеличение затрат оперативной памяти на индексацию.

Недостатком многомерной модели данных является ее громоздкость для простейших задач обычной оперативной обработки информации.