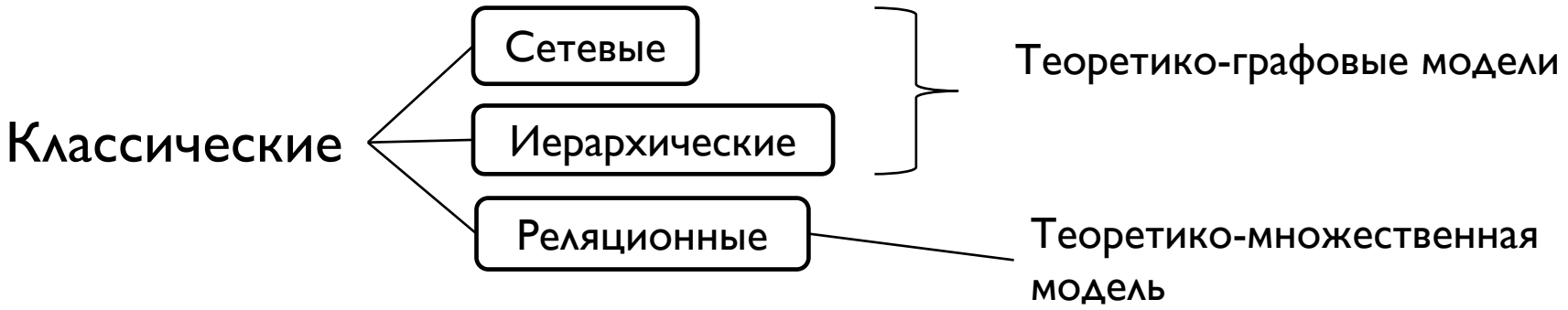
Лекция 4

Модели данных. Реляционная модель

Модель данных



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

Реляционная модель данных

Реляционная модель данных была предложена в 1970 году сотрудником фирмы IBM Эдгаром Коддом и основывается на понятии отношения (*relation*).

Реляционная модель данных, как любая другая модель, состоит из трех частей: структурной, целостной и манипуляционной.

В реляционных базах данных (далее – РБД) *единственной структурой данных* является ***нормализованное n-арное отношение***.

Отношение называется **нормализованным**, если значения всех его атрибутов (аргументов) являются атомарными, т.е. имеют простой тип и не включают множества значений.

Реляционная модель данных

В математике отношение *R* определяется следующим образом:

Отношение *R*, заданное на *n* множествах *S1, S2, ... Sn* ,
есть набор кортежей вида $< s_1^i, s_2^i, \dots s_n^i >$,
таких что $s_1^i \in S1, s_2^i \in S2, \dots s_n^i \in Sn$.

Заметим, что не обязательно всем множествам быть различными.

В зависимости от *n* отношения называются бинарными, тернарными и т.п

Самый простой способ представления n-арного отношения – это таблица, включающая *n* столбцов. Поэтому в Реляц. БД отношения называют еще и **таблицами**.

A	*B*	…	*Z*
…	…	…	…

Компоненты реляционной модели данных

Структурная компонента

Основная структура данных – отношение – двумерная таблица, в которой каждая строка имеет одинаковое число полей, все поля определены и атомарны (недопустимы неопределенные и составные значения).

Манипуляционная компонента

Целостная компонента

Реляционная модель данных

Основные понятия

Основными понятиями РБД являются *тип данных, домен, атрибут, кортеж, отношение, первичный ключ*.

Тип данных – как и в традиционном программировании – это множество значений и совокупность операций над ними.

Типами данных в РБД являются числа (целые и вещественные), строки (символы), логические значения, даты и другие специальные типы, как, например, деньги.

Домен – это подмножество значений данного типа или подтип данного типа, полученный из него путем накладывания ограничений на его значения (в виде логических выражений или перечислением конкретных значений).

Реляционная модель данных

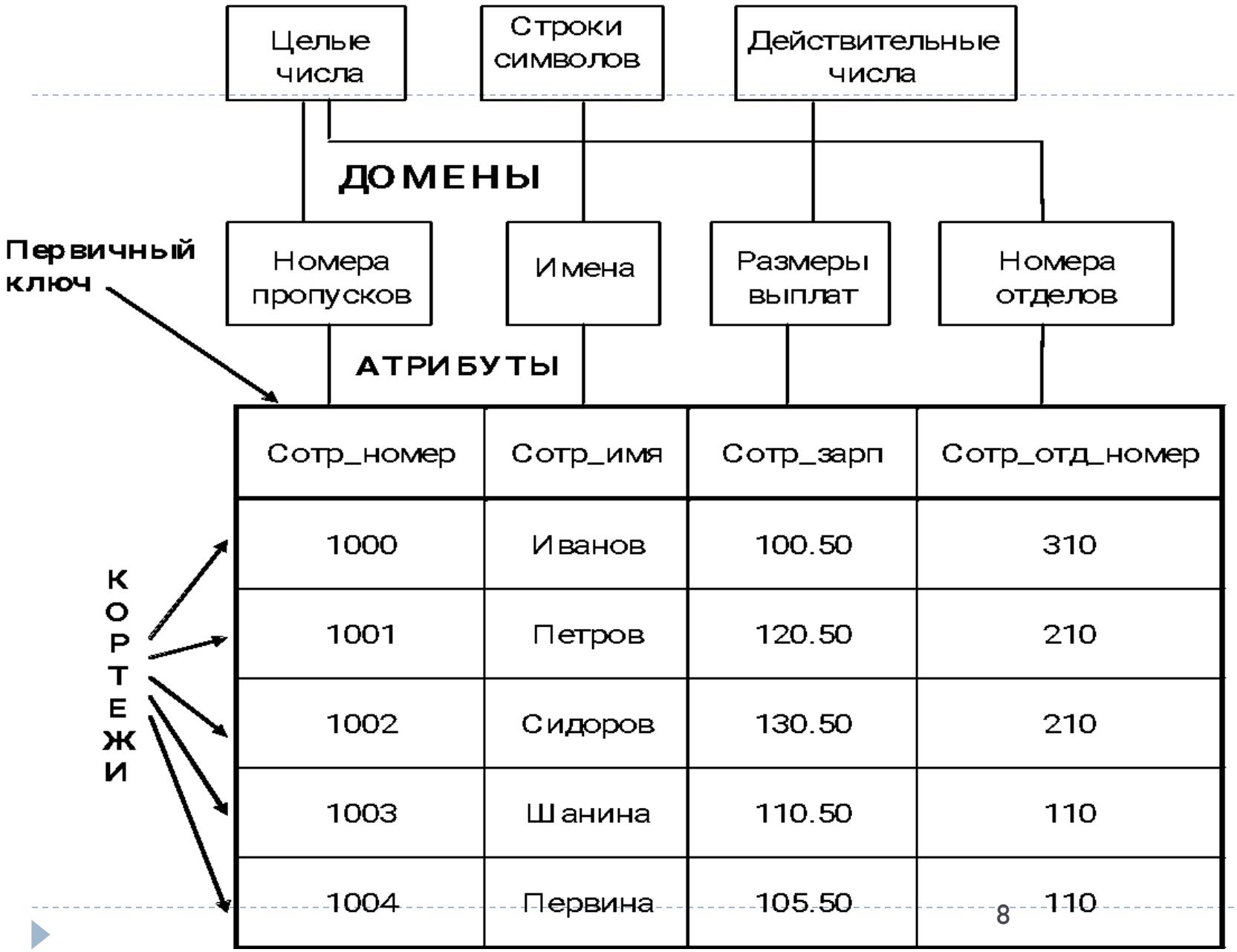
Например, домен "Размеры выплат" получен из типа данных "Действительные числа" путем накладывания на его значения ограничения вида $V \leq 1000$.

Домен "Номера отделов" является подмножеством значений целых чисел и может быть задан перечислением вида {110, 210, 310}.

Домен "Имена" определен на базовом типе "Строки символов", но включает только те строки, которые могут изображать имя.

Смысловая нагрузка домена состоит в том, что данные сравнимы, если они относятся к одному и тому же домену. Это вводит дополнительный контроль за правильным использованием данных и манипулированием ими.

ТИПЫ ДАННЫХ



Сотр_номер	Сотр_имя	Сотр_зарп	Сотр_отд_номер
1000	Иванов	100.50	310
1001	Петров	120.50	210
1002	Сидоров	130.50	210
1003	Шанина	110.50	110
1004	Первина	105.50	110

Реляционная модель данных

Схема отношения (заголовок отношения) – это именованное множество пар вида ***<Имя атрибута, Имя домена>*** или ***<Имя атрибута, Имя типа>***.

Степень или "арность" схемы отношения – мощность множества таких пар.
По определению требуется, чтобы все имена *атрибутов* в *заголовке отношения* были различны

Схема 4-арного отношения *СОТРУДНИКИ*, приведенного на рисунке, имеет вид:

СОТРУДНИКИ
{ <Сотр_номер, Номера_пропусков>,
<Сотр_имя, Имена>,
<Сотр_зарп, Размеры_выплат>,
<Сотр_отд_номер, Номера_отделов> }

Сокращенный способ описания схемы отношения:

СОТРУДНИКИ
(Сотр_ном, Сотр_имя, Сотр_зарп, Сотр_отд_номер).

Реляционная модель данных

Схема БД – это совокупность схем-отношений.

Кортеж – это последовательность упорядоченных пар ***<Имя атрибута, Значение>***, содержащее только одно вхождение каждого ***имени атрибута***, принадлежащего схеме отношения.
Значение является допустимым значением домена данного атрибута.
Арность кортежа совпадает с арностью схемы отношения. На практике имя атрибута в кортеже опускается, тогда его роль играет номер позиции, которую атрибут занимает в схеме отношения.
Проще говоря, ***кортеж*** – это набор именованных значений заданного типа.

Отношение – это множество кортежей, соответствующих одной схеме отношения.

В общем случае *порядок кортежей в отношении*, как и в любом множестве, не определен. Однако в реляционных СУБД для удобства кортежи все же упорядочивают. Чаще всего для этого выбирают некоторый атрибут, по которому система автоматически сортирует кортежи по возрастанию или убыванию. Если пользователь не назначает атрибута упорядочения, система автоматически присваивает номер кортежам в порядке их ввода.

Реляционная модель данных

Обычным представлением отношения является плоская ***двумерная таблица***, заголовком которой является *схема отношения*, а строками – *кортежи отношения*. В этом случае *атрибуты* именуют *столбцы* этой таблицы.

Реляционная БД – это набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

В отличие от теоретико-графовых моделей в реляционной модели **связи между отношениями** поддерживаются неявным образом. В этой модели, так же как и в остальных, поддерживаются иерархические связи между отношениями. В каждой связи одно отношение может выступать как основное, а другое отношение выступает в роли подчиненного. Это означает, что один кортеж основного отношения может быть связан с несколькими кортежами подчиненного отношения. Для поддержки этих связей оба отношения должны содержать наборы атрибутов, по которым они связаны.

Реляционная модель данных

Ключ – это атрибут или набор атрибутов, используемых для идентификации кортежа.

Первичный или *основной* ***ключ*** – это атрибут или набор атрибутов, значения которых однозначно определяют кортеж отношения.

Ключ состоящий из одного атрибута называют простым, а состоящий из нескольких атрибутов- составным (сложным).

В подчиненном отношении для моделирования связи должен присутствовать набор атрибутов, соответствующий первичному ключу основного отношения. Данный набор атрибутов в подчиненном отношении принято называть **внешним ключом**

Реляционная модель данных

В схеме отношения атрибуты, составляющие первичный ключ, как правило, подчеркиваются:

СОТРУДНИКИ (Сотр номер, Сотр_имя, Сотр_зарп, Сотр_отд_номер).

Например, атрибут *Сотр_отд_номер* может использоваться для поиска всех сотрудников из отдела номер 310.
Возможно использование нескольких внешних ключей.

СТУДЕНТ (номер_ личного_ дела , ФИО, Группа, Специальность) и ***ПРЕДМЕТ (Назв_Пр., Часы),*** которые связаны с отношением ***СТУДЕНТ_ПРЕДМЕТ (ФИО, . Назв.Предмета, Оценка).***
В отношении СТУДЕНТ_ПРЕДМЕТ атрибуты ФИО и Назв.Предмета образуют внешний составной ключ.
Атрибуты ФИО и Назв.Предмета первичные ключи в отношениях СТУДЕНТ и ПРЕД-
МЕТ, соответственно.

Реляционная модель данных

Важным требованием к первичному ключу является его *минимальность* или *отсутствие избыточности*: никакой атрибут нельзя удалить из ключа, не нарушая при этом свойства однозначной идентификации кортежей.

В некоторых СУБД допускается создавать отношения, не определяя ключи.

Реляционная модель данных

Возможны случаи, когда отношение имеет несколько комбинаций атрибутов, каждая из которых однозначно определяет все кортежи отношения. Все эти комбинации атрибутов являются **возможными (*потенциальными*) ключами** отношения.

Любой из возможных ключей может быть выбран как первичный.
Пусть *K* — множество атрибутов переменной-отношения R. Множество *K* будет потенциальным ключом переменной-отношения R тогда и только тогда, когда оно обладает следующими свойствами,
а) Уникальность. Никакие допустимые значения переменной-отношения *R* не содержат двух различных кортежей с одинаковыми значениями атрибутов множества *K*.
б) Неизбыточность. Никакое из собственных подмножеств множества K не обладает свойством уникальности.

Обычно используют тот из них, который имеет минимальное число атрибутов.

Базовые свойства отношений

Из приведенных определений следуют следующие свойства отношений:

1) **Отсутствие кортежей дубликатов**

Это следует из определения отношения как множества кортежей
(в классической теории множеств каждое множество состоит из различных элементов).
Из этого свойства вытекает *наличие у каждого отношения первичного ключа*. (По крайней мере, полный набор атрибутов является первичным ключом.)

2) **Отсутствие упорядоченности кортежей**

Это свойство также вытекает из свойств множества.
Отсутствие требования порядка дает дополнительную гибкость СУБД при хранении данных во внешней памяти и обработке запросов к БД.

Базовые свойства отношений

3) Отсутствие упорядоченности атрибутов

Атрибуты не упорядочены, так как по определению схема отношения есть множество пар вида *<Имя атрибута, Значение>* (для ссылки на значение атрибута всегда используется его имя). Это свойство позволяет модифицировать схемы существующих отношений не только путем добавления новых атрибутов, но и удалением существующих. Позволяет также соединять отношения, выдавать атрибуты в любом порядке, хотя часто на практике используется неявный порядок атрибутов.

4) **Атомарность значений атрибутов**

Каждый кортеж содержит ровно одно значение для каждого атрибута.
Это свойство следует из определения домена как потенциального множества значений простого типа данных.

Ограничения целостности

В любой реляционной СУБД должны удовлетворяться два базовых требования целостности:

1) требование целостности сущностей;

2) требование целостности по ссылкам.

Ограничения целостности

1. Целостность сущностей

Сущности, или *объекту* реального мира в РБД соответствуют кортежи отношений. Любой кортеж любого отношения должен быть отличим от любого другого кортежа этого же отношения.
Другими словами, *любое отношение должно обладать первичным ключом.*

Это требование автоматически удовлетворяется, если в системе не нарушаются *базовые свойства отношений*.

Ограничения целостности

2. Целостность по ссылкам

При соблюдении нормализованности отношений сложные сущности реального мира представляются в РБД в виде нескольких кортежей нескольких отношений. Например, сущность *ОТДЕЛ* имеет следующие характеристики: *номер отдела, количество сотрудников* и *список сотрудников отдела*. При правильном проектировании эта сущность представляется в виде двух отношений:

ОТДЕЛЫ (Отд_номер, Отд_кол).
СОТРУДНИКИ (Сотр_номер, Сотр_имя, Сотр_зарп,
Сотр_отд_номер).

В отношении *СОТРУДНИКИ* атрибут *Сотр_отд_номер* является не только собственным свойством сотрудника, но и обеспечивает при необходимости возможность восстановления полной сущности *ОТДЕЛ*.

Ограничения целостности

ОТДЕЛЫ (Отд_номер, Отд_кол).
СОТРУДНИКИ (Сотр_номер, Сотр_имя, Сотр_зарп, Сотр_отд_номер).

Значение атрибута *Сотр_отд_номер* в любом кортеже отношения *СОТРУДНИКИ* должно соответствовать значению атрибута *Отд_номер* в некотором кортеже отношения *ОТДЕЛЫ.*

Такой атрибут, как *Сотр_отд_номер* называется ***внешним ключом***, поскольку его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения
(т.е. задают его *первичный ключ*).

Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут (набор атрибутов) является первичным ключом.

Ограничения целостности

С учетом сказанного выше, *требование целостности по ссылкам* можно сформулировать как *требование внешнего ключа*:
для каждого значения внешнего ключа из ссылающегося отношения в отношении, на которое ведется ссылка, должен найтись кортеж с таким же значением первичного ключа. В противном случае значение внешнего ключа должно быть *неопределенным.*

В нашем примере: если для сотрудника указан номер отдела, то такой отдел должен существовать.

Ограничения целостности

Ограничение целостности по ссылкам автоматически поддерживается следующим образом:

1) **при добавлении новых кортежей** достаточно следить за тем, чтобы не появлялись некорректные значения внешнего ключа;

2) **при удалении кортежей** из отношения, на которое ведет ссылка, имеется три способа поддержания целостности:

а) запрещается удалять кортеж, на который существует ссылка (сначала нужно разобраться со ссылающимися кортежами);

б) при удалении кортежа, на который есть ссылка, во всех ссылающихся кортежах значение внешнего ключа устанавливается неопределенным;

в) при удалении такого кортежа удаляются все ссылающиеся на него кортежи (каскадное удаление).

Базисные средства манипулирования реляционными данными

Определены два базовых механизма манипулирования данными в РБД:
1) *реляционная алгебра* (алгебра отношений);
2) *реляционное исчисление* (исчисление отношений).

Реляционная алгебра (далее – РА) основана на теории множеств. теоретико-множественные операции над данными: объединение, пересечение, разность и декартово произведение;

Реляционное исчисление (далее – РИ) базируется на математической логике, точнее, на исчислении предикатов 1-го порядка. Селекция (выборка), проекция, Соединение и деление;

Оба механизма обладают одним важным свойством: они **замкнуты относительно понятия *отношение***. Это означает, что выражения РА и формулы РИ определяются над *отношениями* РБД и результатом вычисления также являются *отношения*.
В результате любое выражение и любая формула могут интерпретироваться как отношение, что позволяет использовать их в других выражениях и формулах.

Базисные средства манипулирования реляционными данными

Алгебра и исчисление обладают большой выразительной мощностью: очень сложные запросы к базе данных могут быть выражены с помощью одного выражения реляционной алгебры или одной формулы реляционного исчисления. Именно по этой причине именно эти механизмы включены в реляционную модель данных.

В реализациях конкретных реляционных СУБД сейчас не используется в чистом виде ни реляционная алгебра, ни реляционное исчисление. Фактическим стандартом доступа к реляционным данным стал язык SQL (Structured Query Language).

Язык SQL представляет собой смесь операторов реляционной алгебры и выражений реляционного исчисления, использующий синтаксис, близкий к фразам английского языка и расширенный дополнительными возможностями, отсутствующими в реляционной алгебре и реляционном исчислении

Компоненты реляционной модели данных

Структурная компонента

Основная структура данных – отношение – двумерная таблица, в которой каждая строка имеет одинаковое число полей, все поля определены и атомарны (недопустимы неопределенные и составные значения).

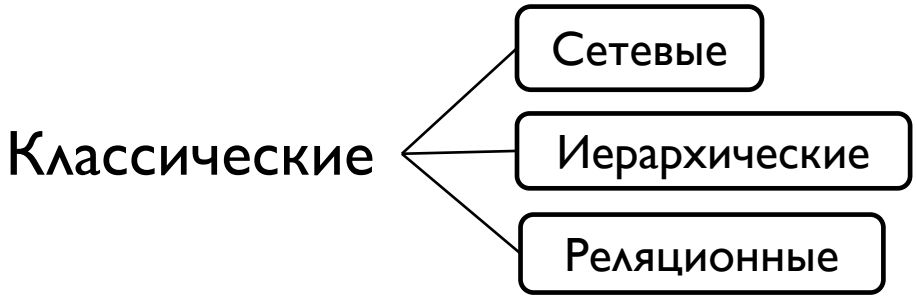
Манипуляционная компонента

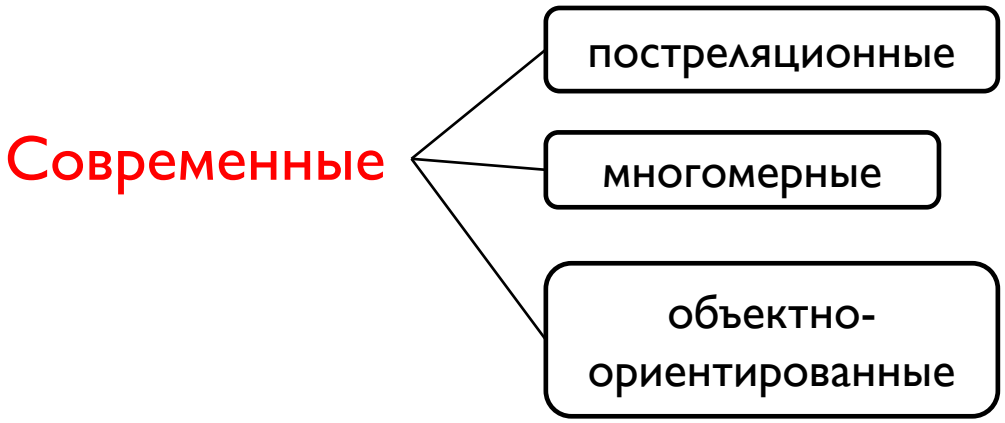
Представлена двумя способами манипулирования данными: **реляционная алгебра**, включающая в себя теоретико-множественные операции над отношениями, и **реляционное исчисление**. Первый механизм базируется в основном на классической теории множеств, а второй - на математической логике.

Целостная компонента

1) **требование целостности сущностей;**
любое отношение должно обладать первичным ключо
2) **требование целостности по ссылкам;**
Например, если удаляется какая-то запись, то должен быть удален и соответствующий экземпляр типа связи.

Типы моделей данных





Современные модели данных являются расширением и обобщением классических. Многие эксплуатируемые в настоящее время СУБД поддерживают несколько типов моделей данных

Постреляционная модель

- Классическая реляционная модель накладывает жесткие ограничения на целостность таблиц. Именно эти ограничения обеспечивают удобство обновления оперативных баз данных в информационных системах.
- Однако практика эксплуатации реляционных СУБД показала, что в ряде случаев эти ограничения чрезвычайно мешают эффективной обработке данных
- Постреляционная модель данных расширяет реляционную модель, снимая некоторые ограничения целостности.

Постреляционная модель

- Так, в постреляционной модели допускаются:
  - неопределенные значения полей
  - составные и многозначные поля. Например, составное поле может быть представлено вектором значений, а набор значений многозначных полей – таблицей, встроенной в основную таблицу,
  - избыточное дублирование данных
- Кроме того, на длину полей и количество полей в записях таблицы не накладывается требований постоянства. Все это обеспечивает постреляционной модели большую гибкость по сравнению с классической реляционной моделью

Постреляционная модель

a) INVOICES

INV_N	CUST_N
0373	8723
8374	8232
7364	8723

INVOICE.ITEMS

INV_N	GOODS	QTY
0373	Сыр	3
0373	Рыба	2
8374	Лимонад	1
8374	Сок	6
8374	Печенье	2
7364	Йогурт	1

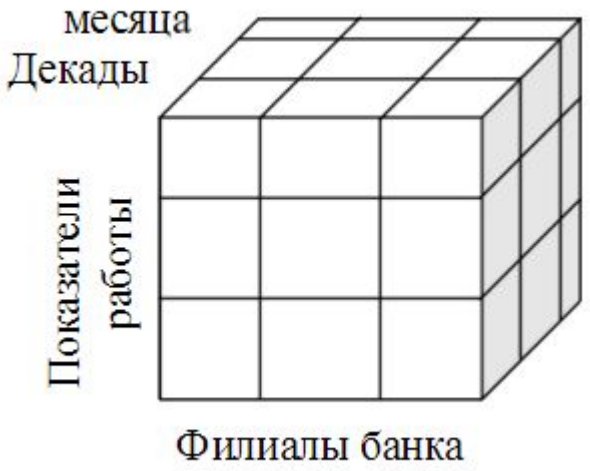
INV_N	CUST_N	GOODS	QTY
0373	8723	Сыр	3
0373		Рыба	2
8374	8232	Лимонад	1
8374		Сок	6
8374		Печенье	2
7364	8723	Йогурт	1

Постреляционная модель

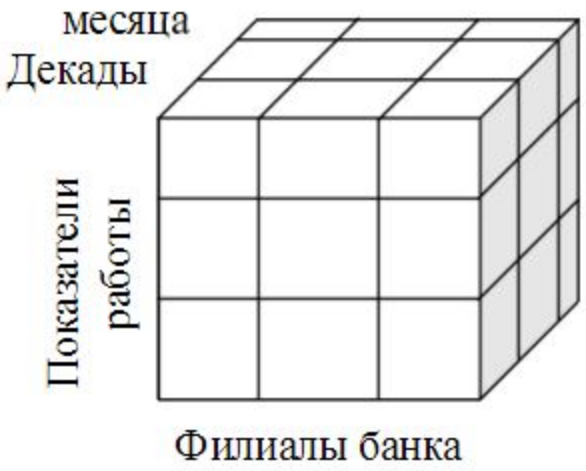
- Достоинством постреляционной модели является возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей. Это обеспечивает высокую наглядность представления информации и повышение эффективности ее обработки.
- Поскольку постреляционная модель допускает избыточное дублирование данных (ненормализованность таблиц), то возникает проблема обеспечения целостности базы данных. В этом состоит недостаток данной модели. В настоящее время проблема целостности решается путем включения в СУБД дополнительных механизмов – хранимых процедур и триггеров, что ведет к усложнению и удорожанию СУБД. Современные СУБД в большинстве своем являются реляционными и постреляционными

Многомерная модель

- Многомерная модель данных – обобщение классической реляционной модели. Многомерный подход к представлению данных появился практически одновременно с реляционным. Однако реально работающих многомерных СУБД
- Основная структура данных – многомерная таблица. Как правило, одной из координат выступает время. Многомерная база данных – совокупность гиперкубов различной размерности. Пользователь по-прежнему может работать с двумерными таблицами, которые являются срезами многомерной таблицы. Многомерную базу данных принято называть хранилищем данных



Многомерная модель



- Многомерные СУБД являются узкоспециализированными СУБД, предназначенными для интерактивной аналитической обработки информации.

Основные понятия, используемые в этих СУБД: агрегируемость, историчность и прогнозируемость данных.

- Агрегируемостъ данных означает рассмотрение информации на различных уровнях ее обобщения. В информационных системах степень детальности представления информации для пользователя зависит от его уровня: аналитик, пользователь-оператор, управляющий, руководитель.
- Историчность данных предполагает обеспечение высокого уровня статичности (неизменности) собственно данных и их взаимосвязей, а также обязательность привязки данных ко времени.
- Прогнозируемость данных подразумевает задание функций прогнозирования и применение их к различным временным интервалам.

Многомерная модель

Реляционная модель

Модель	Месяц	Объем
"Калина"	апрель	*10*
" Калина "	май	*15*
" Калина "	июнь	*4*
"Ford"	апрель	*10*
"Ford"	июнь	*18*
"Renault"	июнь	*20*

многомерная модель

Модель	апрель	май	июнь
" Калина "	*10*	*15*	*4*
" Ford "	*10*	*18*	*N*
" Renault"	*N*	*20*	*N*

Многомерная модель

- Многомерность модели данных означает не многомерность визуализации цифровых данных, а многомерное логическое представление структуры информации при описании и в операциях манипулирования данными. С формальной точки зрения мы имеем дело с n-арным отношением
- $\sigma, \sigma \subseteq A_1 \times A_2 \times .. \times A_n$ Ai – конечное множество, характеризующее г

- **Манипуляционная компонента:**
  - Помимо средств манипулирования данными реляционной модели имеются также специальные операции: формирование среза, замена координат и др., вращение (изменение порядка измерений)

- Достоинство: удобство и эффективность аналитической обработки больших объемов информации, связанных со временем.
- Недостатки: узкая специализация модели, сложность контроля целостности хранилища данных

Объектно-ориентированная модель

- В объектно-ориентированной модели данных реализуются две концепции:
  - объектно-ориентированного программирования и
  - концепции баз данных и СУБД.
- Такая модель данных была впервые описана в рекомендациях стандарта ODMG-93 (Object Database Management group)
- Между тем эти рекомендации в полном объеме до сих пор пока не реализованы в промышленных СУБД. Многие полнофункциональные современные СУБД содержат лишь некоторые механизмы объектно-ориентированного программирования

Объектно-ориентированная модель

- Данные структурируются подобно иерархической модели данных. Однако вершинами дерева выступают не записи, а объекты. Между объектами и функциями их обработки устанавливаются взаимосвязи. Это позволяет идентифицировать отдельные объекты базы данных

- **Манипуляционная компонента:**

  Логические операции, усиленные механизмами инкапсуляции, наследования и полиморфизма. Для создания базы данных применяются операции, подобные командам языка SQL

- Достоинствами объектно-ориентированной модели данных в сравнении с реляционной моделью является возможность: отображения информации о сложных взаимосвязях объектов, идентификации отдельных объектов и определения для них специфических функций обработки.
- Недостатки: высокая понятийная сложность, низкая скорость обработки запросов

Выбор средств реализации базы данных

- Выбор средств реализации выполняется на третьем этапе процесса проектирования базы данных.
- Вначале рекомендуется выбрать модель данных, а затем – программную платформу.
- Такой порядок действий имеет следующее объяснение: выбор СУБД до инфологического моделирования предметной области можно сравнить с выбором инструмента до того, как мастер узнает, что ему предстоит делать; отдать обоснованное предпочтение той или иной модели данных (с ее структурами данных, операциями и ограничения целостности) можно лишь после того, как построена ER -диаграмма предметной области и выявлены возможные запросы к базе данных. Ведь с умом выбрать «тару» можно только тогда, когда ясно, что подлежит хранению и какие операции надо будет выполнять над хранимыми объектами;