

Севастопольский государственный университет
Кафедра «Информационные системы»

Управление данными

курс лекций

лектор:
ст. преподаватель кафедры ИС Абрамович А.Ю.



Лекция 3

Модели представления данных

(сетевая модель, иерархическая модель, реляционная модель, постреляционная модель, многомерная модель, объектно-ориентированная модель).

Одними из основополагающих в концепции баз данных являются обобщенные категории **«данные»** и **«модель данных»**.

«Иванов Иван Иванович 24»

«Управление данными 36 36»

«Печенье Темная сторона 30 80»

Данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознаёт их смысловое содержание. Поэтому **центральным понятием в области баз данных является понятие модели данных.**

ФИО студента	Осталось лабораторных сдать, шт.
Иванов Иван Иванович	24

Дисциплина	Лекции, ч.	ЛЗ, ч.
Управление данными	36	36

Товар	Поставщик	Количество, коробки шт.	Цена за ед., руб.
Печенье	Темная сторона	30	80

МОДЕЛИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Модель данных - интегрированный набор понятий для описания и обработки данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Модель данных - схема (порядок, совокупность принципов, система) организации данных в единое целое для создания, накопления, обработки и управления. Это **некоторая абстракция**, которая будучи приложена к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию.

Модель данных - совокупность структур данных и операций по их обработке.

Структурная часть

- набор правил, по которым может быть построена база данных

Управляющая часть

- определяющая типы допустимых операций с данными

Ограничения

- набор (необязательный) ограничений поддержки целостности данных, гарантирующих корректность используемых данных.

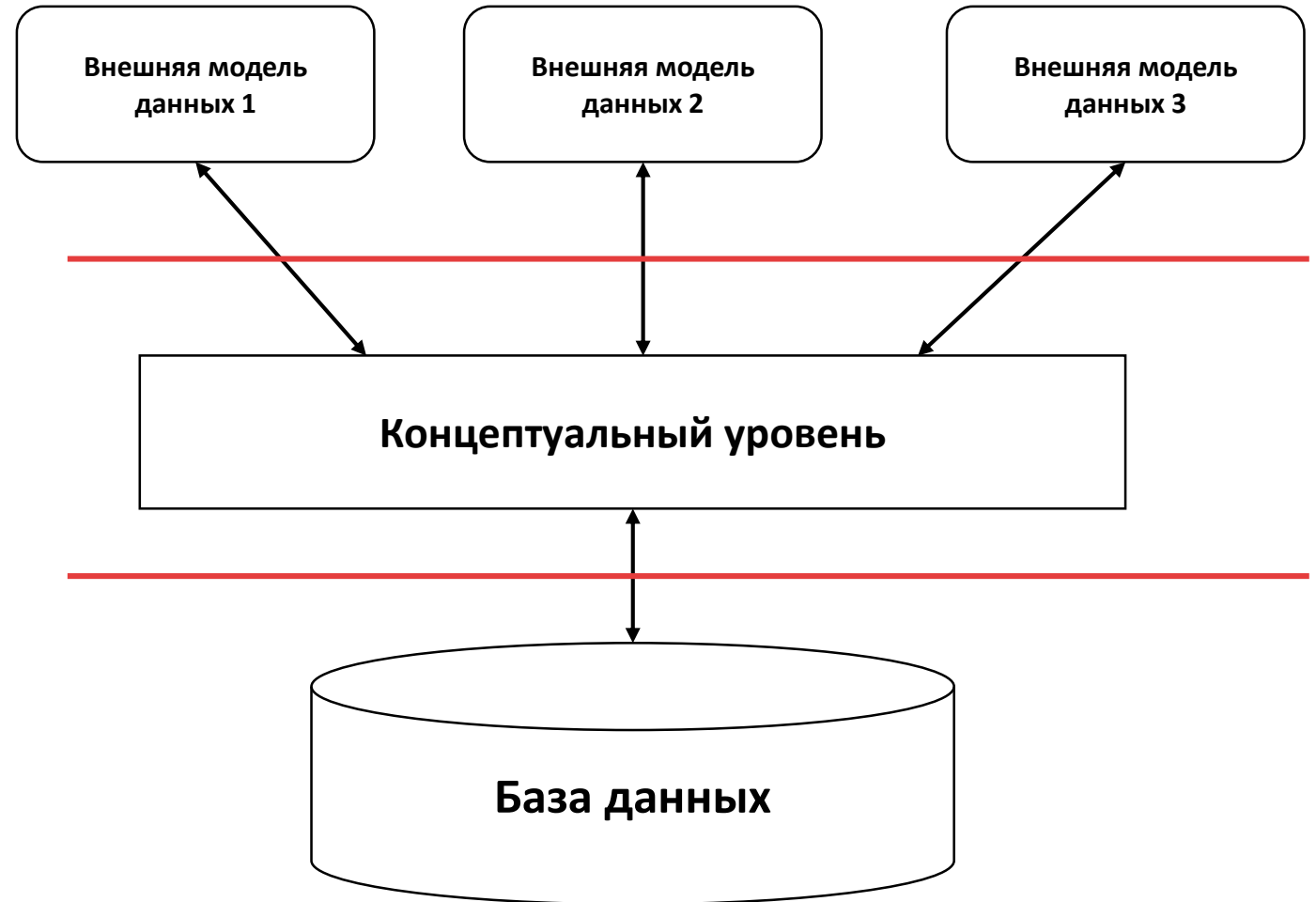
АРХИТЕКТУРА БАЗЫ ДАННЫХ

Трехуровневая система организации БД

Уровень внешних моделей — является самым верхним уровнем или уровнем пользователя. Это совокупность внешних представлений данных, которые обрабатывают приложения и какими их видит пользователь на экране.

Концептуальный уровень — центральное управляющее звено, здесь база данных представлена в наиболее общем виде, объединяет данные. Фактически отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась БД.

Физический уровень — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.



КЛАССИФИКАЦИЯ МОДЕЛЕЙ ДАННЫХ



КЛАССИФИКАЦИЯ МОДЕЛЕЙ ДАННЫХ

Инфологические модели данных используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения. Среди разновидностей инфологических моделей наибольшее распространение получили *модели сущность-связь*.

Даталогические модели поддерживаются конкретной СУБД.

Документальные модели данных соответствуют представлению о **слабоструктурированной информации**, ориентированной в основном на свободные форматы документов, текстов на естественном языке. **Фактографические модели** содержат информацию, относящуюся непосредственно к предметной области

Физические модели описывают структуры и принципы их хранения во внешней памяти, а также доступа к ним в зависимости от используемых аппаратных средств и программного обеспечения низкого уровня.

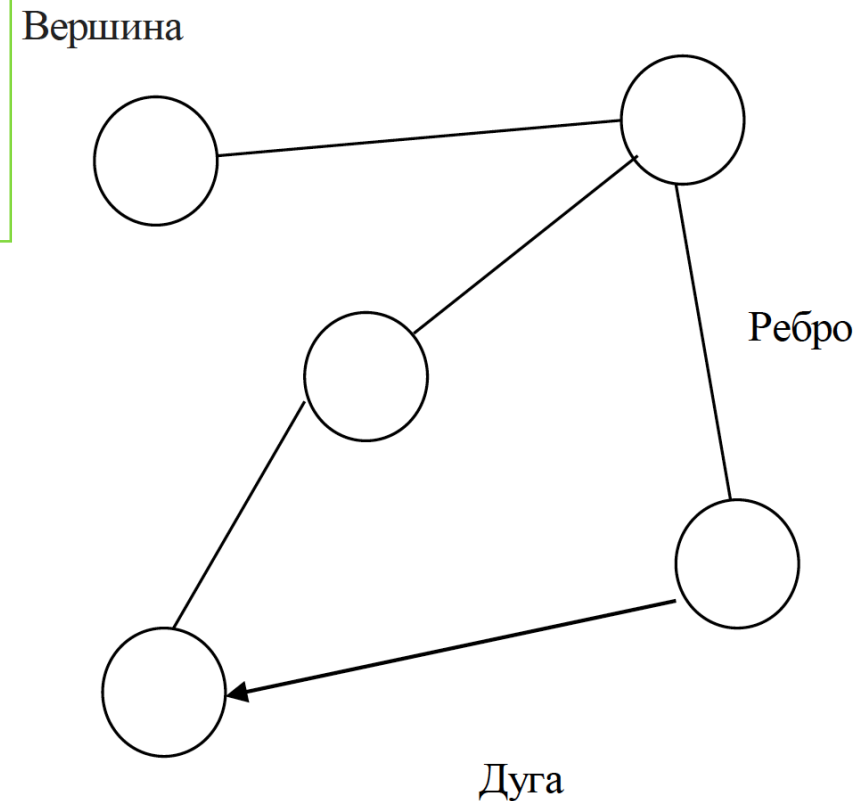
ТЕОРЕТИКО-ГРАФОВЫЕ МОДЕЛИ ДАННЫХ

Модели отражают совокупность объектов реального мира в виде графа взаимосвязанных информационных объектов.

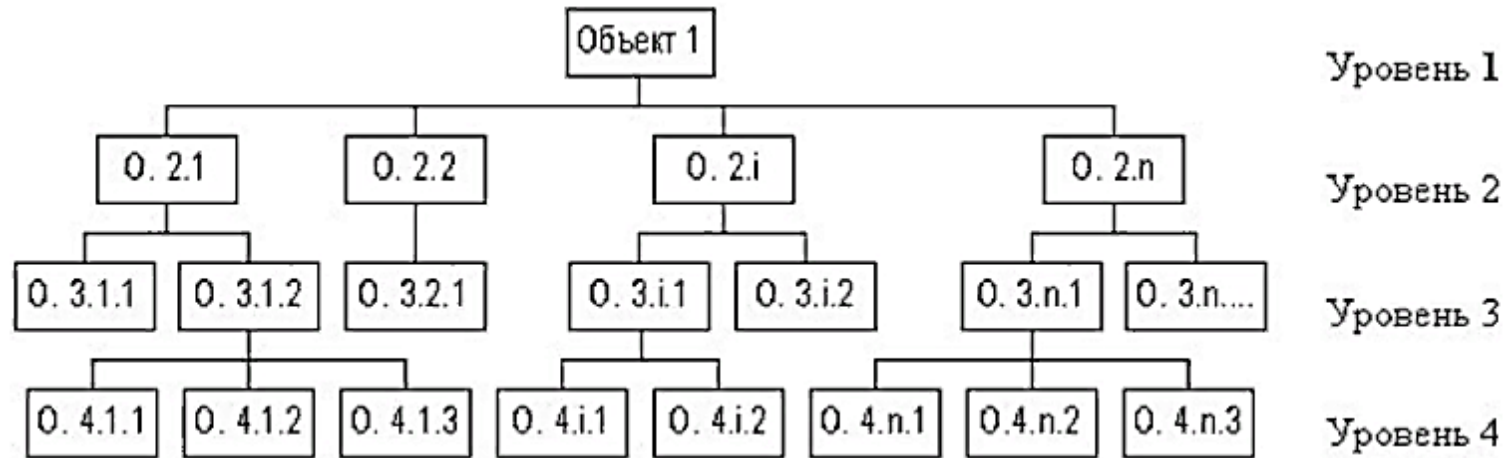
К теоретико-графовым моделям относятся две разновидности:

- сетевые модели;
- иерархические модели.

В таких моделях данных предусматриваются характерные для подобного рода структур операции навигации и манипулирования данными. Принципиальное значение при этом имеет то обстоятельство, что предусматривается одновременная обработка только одиночных объектов данных из БД — записей, сегментов или полей записей. Интерактивный доступ к БД поддерживается только путем создания соответствующих прикладных программ с собственным интерфейсом. В связи с этим системы, построенные на этих моделях данных, называют навигационными.



ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ



В иерархической модели связи между данными можно описать с помощью упорядоченного графа (или дерева).

Основными информационными единицами в иерархической модели данных являются **сегменты поля**.

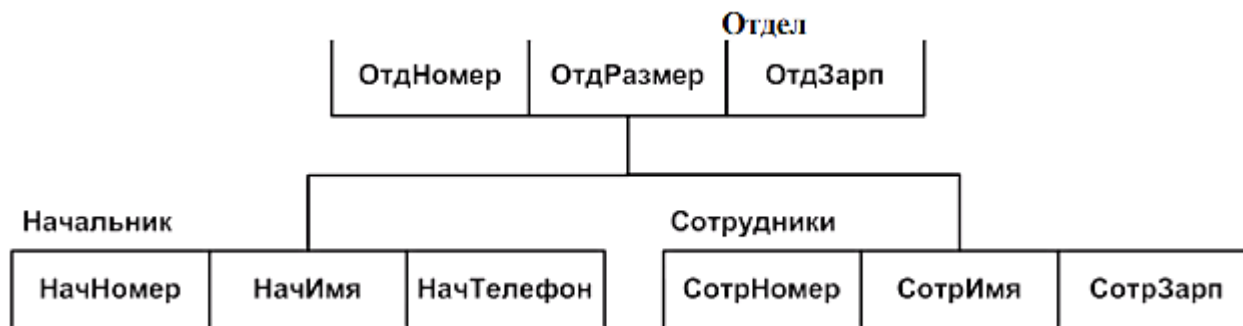
Поле данных — определяется как наименьшая неделимая единица данных доступная пользователю.

Для сегмента определяется **тип** и **экземпляр сегмента**. **Экземпляр сегмента** образуется из конкретных значений полей данных. **Тип сегмента** — именованная совокупность входящих в него типов данных.

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

В иерархической модели данных **вершине графа соответствует тип сегмента или просто сегмент, а дугам – типы связи (предок-потомок)**. В данных **моделях сегмент, являющийся потомком должен иметь в точности одного предка**.

Здесь **Отдел** является предком для **Начальник** и **Сотрудники**, а **Начальник** и **Сотрудники** - **потомки Отдел**. Между типами записи поддерживаются связи.



Каждый из типов «дерево» состоит из одного «корневого» типа и упорядоченного набора (возможно пустого) подчиненных типов. **Каждый из элементарных типов, включенных в тип «дерево», является простым или составным типом «запись»**. Простая запись состоит из одного типа, например числового, а составная «запись» объединяет некоторую совокупность типов, например целое, строку символов и указатель (ссылку).

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Иерархическая БД представляет собой упорядоченную совокупность экземпляров данных типа «дерево» (деревьев), содержащих экземпляры типа «запись» (записи). Часто отношения родства между типами переносят на отношения между самими записями. Поля записей хранят собственно числовые или символьные значения, составляющие основное содержание БД. **Обход всех элементов иерархической БД обычно производится сверху вниз слева направо.**



Пример данных в БД (один экземпляр) по схеме ИМД

Для организации физического размещения иерархических данных в памяти ЭВМ могут использоваться следующие группы методов:

- представление **линейным списком с последовательным распределением памяти** (адресная арифметика, левосписковые структуры);
- представление **связными линейными списками** (методы, использующие указатели и справочники).

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Между предками и потомками автоматически поддерживается контроль целостности связей. **Основное правило контроля целостности звучит следующим образом: потомок не может существовать без родителя, а у некоторых родителей может не быть потомков.** Механизмы поддержания целостности связей между записями различных деревьев отсутствуют.

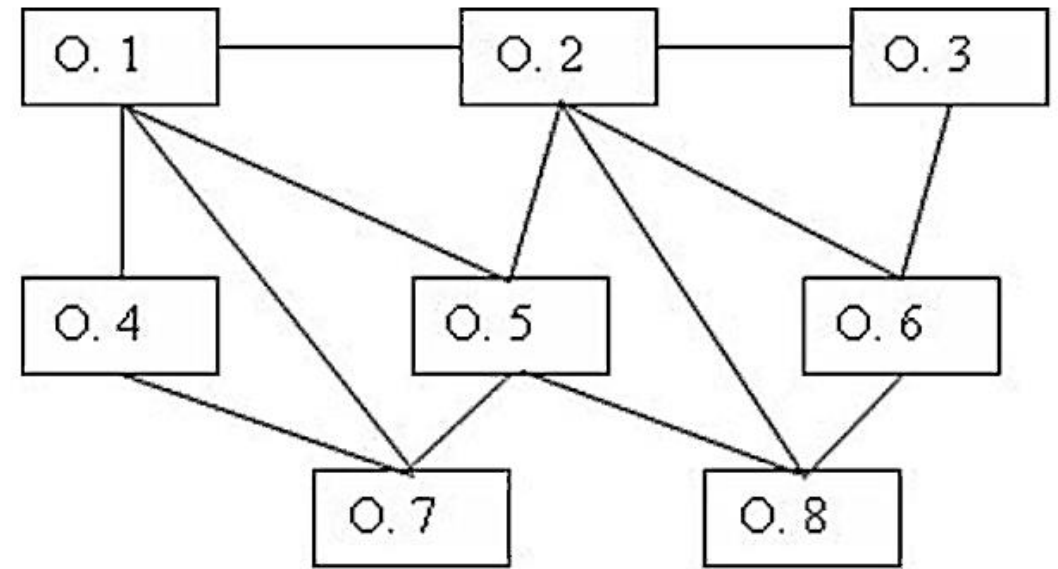
К достоинствам иерархической модели данных относятся **эффективное использование памяти ЭВМ** и неплохие показатели **времени выполнения основных операций** над данными. Иерархическая модель удобна для работы с иерархически упорядоченной информацией.

Недостатком иерархической модели является частичное **дублирование информации** (не предусмотрена поддержка соответствия между парными записями), **громоздкость для обработки информации** с достаточно сложными логическими связями, а также **сложность понимания** для обычного пользователя.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Сетевая модель данных позволяет отображать разнообразные взаимосвязи элементов данных **в виде произвольного графа**, обобщая тем самым иерархическую модель данных.

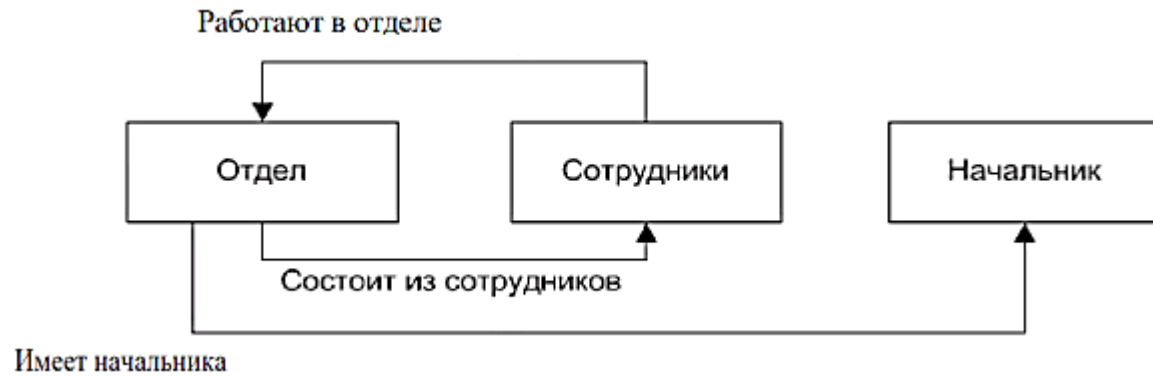
Для описания **схемы сетевой БД** используется две группы типов: «**запись**» и «**связь**». Тип «связь» определяется для двух типов «запись»: предка и потомка. Переменные типа «связь» являются экземплярами связей.



Сетевая БД состоит из набора записей и набора соответствующих связей. На формирование связи особых ограничений не накладывается. Если в иерархических структурах запись-потомок могла иметь только одну запись-предка, то **в сетевой модели данных запись-потомок может иметь произвольное число записей-предков.**

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Физическое размещение данных в базах сетевого типа может быть организовано практически теми же методами, что и в иерархических БД.



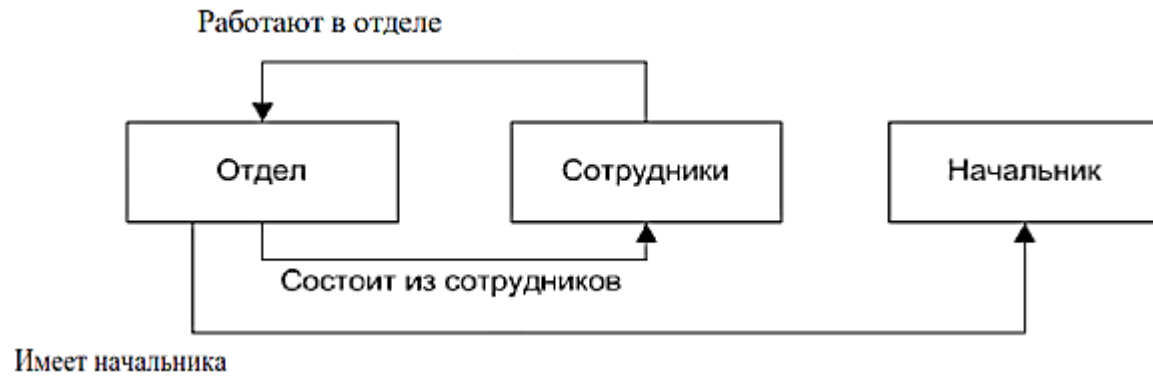
Пример схемы простейшей сетевой БД. Типы связей здесь обозначены надписями на соединяющих типы записей линиях.

Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. Сетевая модель предоставляет большие возможности в смысле **допустимости образования произвольных связей**.

Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также **сложность для понимания и выполнения обработки информации в БД обычным пользователем**. В сетевой модели данных **ослаблен контроль целостности** связей вследствие допустимости установления произвольных связей между записями.

СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Физическое размещение данных в базах сетевого типа может быть организовано практически теми же методами, что и в иерархических БД.

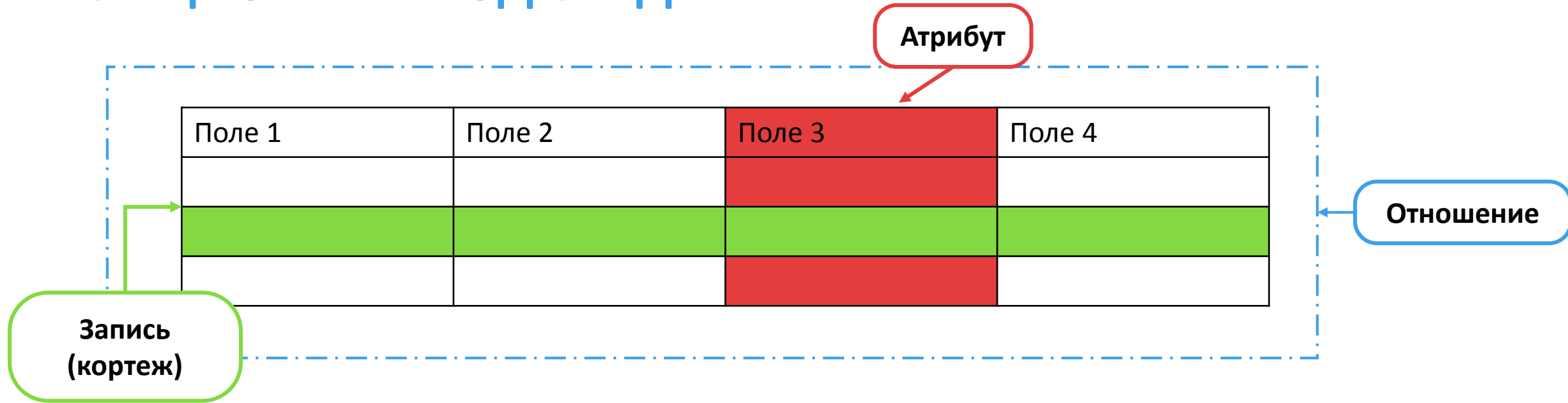


Пример схемы простейшей сетевой БД. Типы связей здесь обозначены надписями на соединяющих типы записей линиях.

Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. Сетевая модель предоставляет большие возможности в смысле **допустимости образования произвольных связей**.

Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также **сложность для понимания и выполнения обработки информации в БД обычным пользователем**. В сетевой модели данных **ослаблен контроль целостности** связей вследствие допустимости установления произвольных связей между записями.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ



Реляционная модель данных предложена сотрудником фирмы IBM Эдгаром Коддом и **основывается на понятии отношение (relation).**

Реляционная модель данных (РМД) - это способ рассмотрения данных, при котором данные воспринимаются пользователем как **взаимосвязанные таблицы и в распоряжении пользователя** имеются некоторые операторы, которые генерируют новые таблицы из старых.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Реляционная модель данных представляет собой совокупность данных, содержащихся в двумерных таблицах, соединённых между собой отношениями. Любые данные можно преобразовать в простую таблицу. Такое представление является наиболее удобным и для пользователя, для и машины.

Под таблицами в РМД понимается **структура данных, состоящая из строк и столбцов**. В этой структуре каждый столбец содержит данные только одного типа, каждая строка состоит из набора значений составляющих ее столбцов.

Под операторами понимаются операции **выборки, группировки, соединения и некоторые другие**, результатом которых являются новые таблицы, полученные на основании старых.

Основными **понятиями реляционных баз данных являются**: тип данных, домен, атрибут, кортеж, отношение, первичный ключ.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

РЕЛЯЦИОННАЯ МОДЕЛЬ СОСТОИТ ИЗ ТРЕХ ЧАСТЕЙ:

Структурная часть

- описывает, какие объекты рассматриваются реляционной моделью.

Целостная часть

- описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. **Это целостность сущностей и целостность внешних ключей.**

Манипуляционная часть

- описывает два эквивалентных способа манипулирования реляционными данными - **реляционную алгебру и реляционное исчисление.**

Единственные **структуры данных, используемые в реляционной модели, являются нормализованные n-арные отношения.**

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Достоинства реляционной модели:

- **простота и доступность** для понимания пользователем. Единственной используемой информационной конструкцией является «таблица»;
- **строгие правила проектирования**, базирующиеся на математическом аппарате;
- **полная независимость данных**. Изменения в прикладной программе при изменении реляционной БД минимальны;
- для организации запросов и написания прикладного ПО **нет необходимости знать конкретную организацию БД** во внешней памяти.

Недостатки реляционной модели:

- **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.

ПОСТРЕЛЯЦИОННАЯ МОДЕЛЬ

Постреляционная модель данных представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных, хранящихся в записях таблиц. Постреляционная модель данных допускает многозначные поля — поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу.

РЕЛЯЦИОННАЯ МОДЕЛЬ

INVOICES

INVNO	CUSTNO
373	8723
8374	8232
7464	8723

INVOICE ITEMS

INVNO	GOODS	QTY
373	Сыр	3
373	Рыба	2
8374	Лимонад	1
8374	Сок	6
8374	Печенье	2
7364	Йогурт	1

ПОСТРЕЛЯЦИОННАЯ МОДЕЛЬ

INVOICES


INVNO	CUSTNO	GOODS	QTY
373	8723	Сыр	3
		Рыба	2
8374	8232	Лимонад	1
		Сок	6
		Печенье	2
7364	8723	Йогурт	1


Постреляционная модель поддерживает **ассоциированные многозначные поля (множественные группы)**. Совокупность ассоциированных полей называется **ассоциацией**. При этом в строке первое значение одного столбца ассоциации соответствует первым значениям всех других столбцов ассоциации. Аналогичным образом связаны все вторые значения столбцов и т.д.

ПОСТРЕЛЯЦИОННАЯ МОДЕЛЬ

Поскольку постреляционная модель допускает хранение в таблицах ненормализованных данных, **возникает проблема обеспечения целостности и непротиворечивости данных**. Эта проблема решается включением в СУБД механизмов, подобных хранимым процедурам в клиент-серверных системах.

Для описания функций контроля значений в полях **имеется возможность создавать процедуры** (коды конверсии и коды корреляции), **автоматически вызываемые до или после обращения к данным**. Коды корреляции выполняются сразу после чтения данных, перед их обработкой. Коды конверсии, наоборот, выполняются после обработки данных.

 **Достоинства:** возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей, что обеспечивает высокую наглядность представления информации и повышение эффективности ее обработки.

 **Недостатки:** сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

МНОГОМЕРНАЯ МОДЕЛЬ

Многомерные системы позволяют оперативно обрабатывать информацию для проведения анализа и принятия решения. Многомерные СУБД являются узкоспециализированными СУБД, предназначенными для интерактивной аналитической обработки информации. Основные свойства, присущие к этим СУБД: **агрегируемость, историчность и прогнозируемость данных.**

- **Агрегируемость данных** означает рассмотрение информации на различных уровнях ее обобщения.
- **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.

РЕЛЯЦИОННАЯ МОДЕЛЬ

Дисциплина	Месяц	Должники
Управление данными	Сентябрь	15
Высшая математика	Сентябрь	13
Веб-технологии	Сентябрь	12
Управление данными	Октябрь	7
Управление данными	Ноябрь	4
Веб-технологии	Октябрь	6

МНОГОМЕРНАЯ МОДЕЛЬ

Дисциплина	Сентябрь	Октябрь	Ноябрь
Управление данными	15	7	4
Высшая математика	13	NULL	NULL
Веб-технологии	12	6	NULL

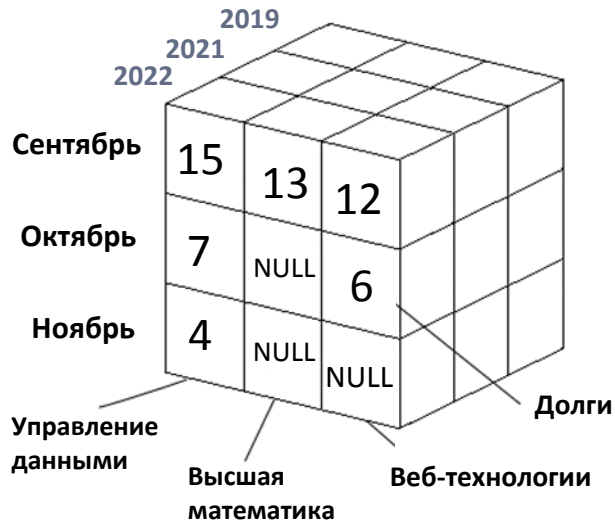
Многомерность модели данных означает не многомерность визуализации цифровых данных, а многомерное логическое представление структуры информации при описании и в операциях манипулирования данными.

По сравнению с реляционной моделью многомерная организация данных обладает более высокой наглядностью и информативностью.

МНОГОМЕРНАЯ МОДЕЛЬ

Если речь идет о многомерной модели с мерностью больше двух, то визуально информация представляется в виде многомерных объектов (трех-, четырех- и более мерных гиперкубов).

В существующих МСУБД используются два основных варианта (схемы) организации данных: гиперкубическая и поликубическая.



В поликубической схеме предполагается, что в БД может быть определено несколько гиперкубов с различной размерностью и с различными измерениями в качестве граней.

В случае гиперкубической схемы предполагается, что все показатели определяются одним и тем же набором измерений. Это означает, что при наличии нескольких гиперкубов БД все они имеют одинаковую размерность и совпадающие измерения.

+ Достоинства: удобство и эффективность аналитической обработки больших объемов данных, связанных со временем.

□ Недостатки: громоздкость для простейших задач обычной оперативной обработки информации

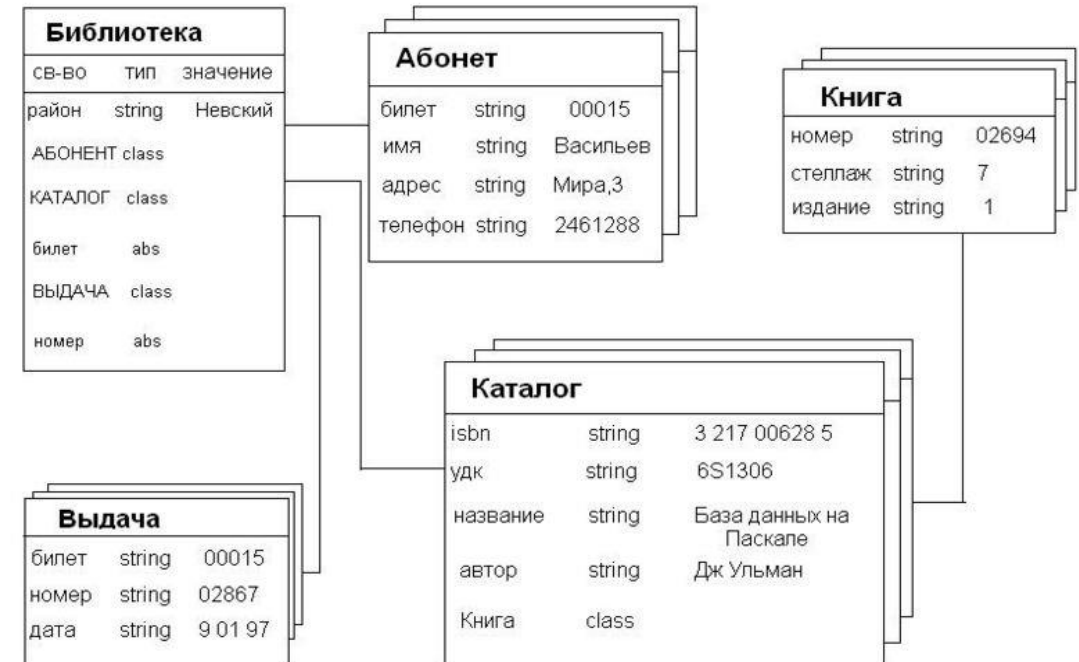
ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

В объектно-ориентированной модели при представлении данных имеется возможность идентифицировать отдельные записи базы. Между записями базы данных и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектно-ориентированных языках программирования.

Стандартизованная объектно-ориентированной модель описана в рекомендациях стандарта ODMG (Object Database Management Group — группа управления объектно-ориентированными базами данных).

Логическая структура объектно-ориентированной БД внешне похожа на структуру иерархической БД. Основное отличие между ними состоит в методах манипулирования данными.


Структура объектно-ориентированной БД графически представима в виде дерева, узлами которого являются объекты. Свойства объектов описываются некоторым стандартным типом (например, строковым — string) или типом, конструируемым пользователем (определяется как class).




ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ

Для выполнения действий над данными в рассматриваемой модели БД применяются логические операции, **усиленные объектно-ориентированными механизмами инкапсуляции, наследования и полиморфизма**. Ограниченно могут применяться операции, подобные командам SQL (например, для создания БД).

Поиск в объектно-ориентированной БД состоит в выяснении сходства между объектом, задаваемым пользователем, и объектами, хранящимися в БД. Определяемый пользователем объект, называемый объектом-целью (свойство объекта имеет *тип goal*) в общем случае может представлять собой подмножество всей хранимой в БД иерархии объектов. Объект-цель, а также результат выполнения запроса могут храниться в самой базе.

 **Достоинства:** возможность отображения информации о сложных взаимосвязях объектов. Объектно-ориентированная модель данных позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.

 **Недостатки:** высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов.