

# Лекция 2-2020\_2021

- На прошлой лекции
- Модели представления данных
  - Иерархическая
  - Сетевая
  - Реляционная
  - Постреляционная
- OLTP и OLAP системы
  - Многомерная модель данных

# На прошлой лекции

- Содержание курса. Самостоятельные лабораторные работы. Зачет. Курсовая работа. Экзамен. Инструменты, применяемые для практики.
- Примеры баз данных.
- История возникновения и развития баз данных (60-е годы XX столетия – разработка проекта полета корабля Apollo на Луну. Результат – создание системы IMS (Information Management System).
- Основные понятия и определения. Базы данных (БД) – набор логически связанных данных и их описаний.
-

# На прошлой лекции

- Система управления базой данных (СУБД) – программное обеспечение, которое позволяет пользователям работать с БД и обеспечивает контролируемый доступ к БД
- Типы информации в БД : данные пользователей, метаданные, индексы.
- Функции СУБД: управление данными во внешней памяти, управление буферами оперативной памяти, управление транзакциями, журнализация, поддержка языков БД.
- Классификация СУБД: полнофункциональные, серверы БД.

# Модели представления данных

СУБД основаны на моделях баз данных — определённых структурах для обработки данных. Каждая СУБД реализует одну из моделей баз данных для логической структуризации используемых данных. Эти модели являются главным критерием того, как будет работать и управлять информацией приложение.

Решений, реализующих различные модели баз данных, очень много. Периодически некоторые из них становятся очень популярными и используются на протяжении многих лет.

# Модели представления данных

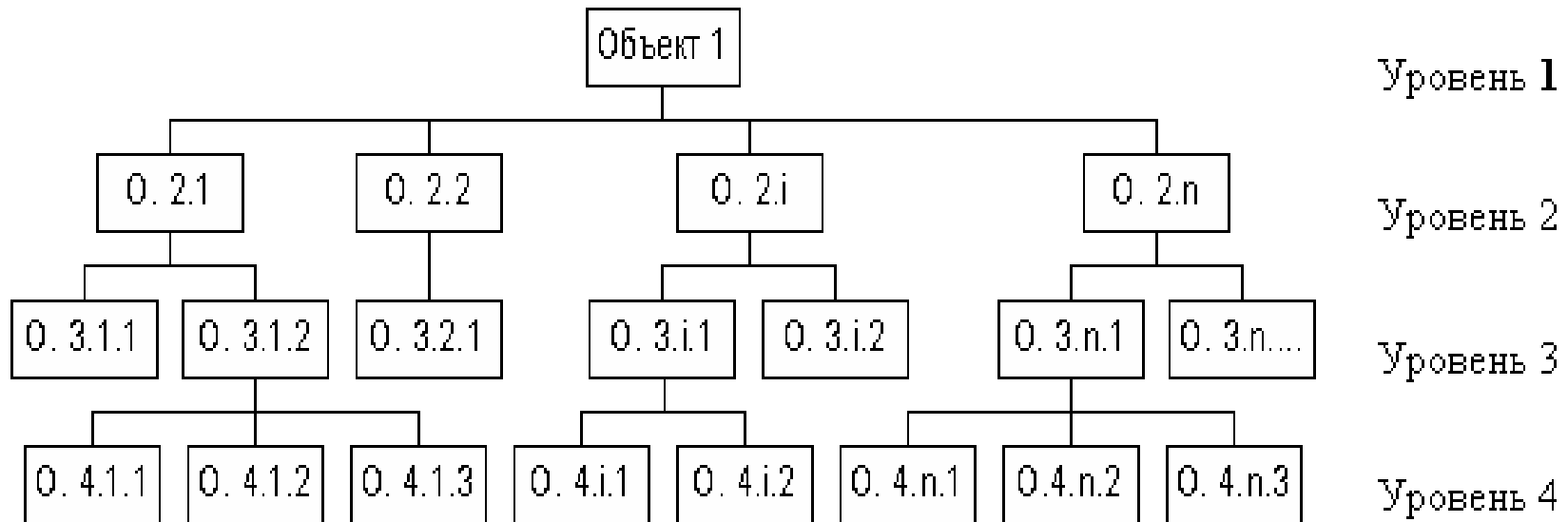
Основные модели представления данных:

- иерархическая,
- сетевая,
- реляционная,
- постреляционная,
- многомерная,
- объектно-реляционная,
- объектно-ориентированная,
- NoSQL,
- NewSQL.

# Иерархическая модель

Иерархическая БД состоит из упорядоченного набора нескольких экземпляров одного типа дерева.

Основное правило: запись-потомок должна иметь в точности [одного предка](#).



*Иерархические СУБД: IMS, PC/Focus, Team-Up, Data Edge, ИНЭС, МИРИС.*

## **Задача: предприятие-отделы-контракты-заказчики**

Предприятие состоит из отделов, в которых работают сотрудники. В каждом отделе может работать несколько сотрудников, но сотрудник не может работать более чем в одном отделе. Предприятие в лице своих сотрудников выполняет работы по контракту с юридическими или физическими лицами (заказчиками).

## Пример иерархической модели данных

Отношение (а) **отдел - сотрудники**, состоящее из родительской записи **ОТДЕЛ** (*НАИМЕНОВАНИЕ\_ОТДЕЛА*, *ЧИСЛО\_РАБОТНИКОВ*) и дочерней записи **СОТРУДНИК** (*ФАМИЛИЯ*, *ДОЛЖНОСТЬ*, *ОКЛАД*). Структура экземпляра дерева:

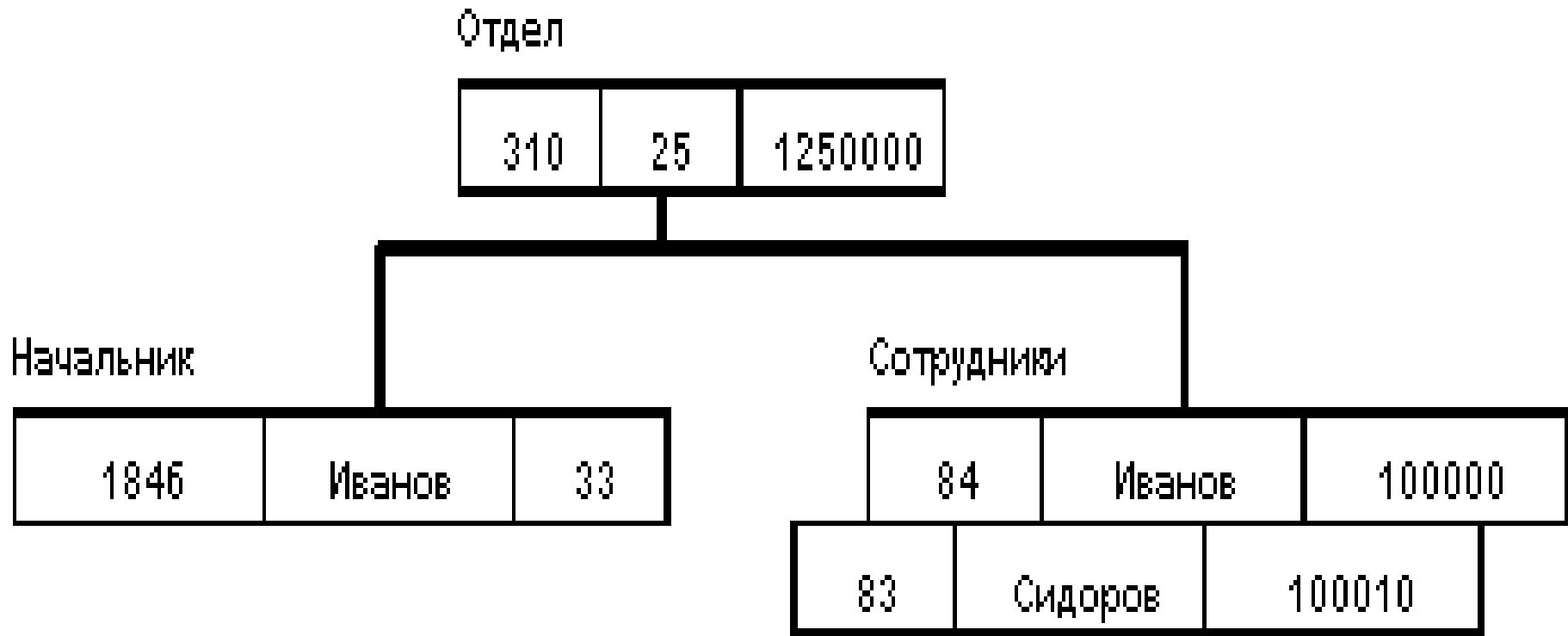


(а)

Для простоты полагаем, что имеются только две дочерние записи.



# Пример иерархической модели данных



Пример экземпляра дерева с видоизмененной структурой ( в отдел добавлено поле ФОНД, для сотрудника добавлен табельный номер). Деревьев столько, сколько отделов.

# Пример иерархической модели данных

Операции:

1. Найти указанное дерево БД (например, отдел 310);
2. Перейти от одного дерева к другому;
3. Перейти от одной записи к другой внутри дерева (например, от отдела - к первому сотруднику);
4. Перейти от одной записи к другой в порядке обхода иерархии;
5. Вставить новую запись в указанную позицию;
6. Удалить текущую запись.

# Пример иерархической модели данных

Отношение (б):

заказчик — контракты — исполнители.

ЗАКАЗЧИК

(НАИМЕНОВАНИЕ\_ЗАКАЗЧИКА, АДРЕС),

КОНТРАКТ (НОМЕР, ДАТА, СУММА),

ИСПОЛНИТЕЛЬ  
(ФАМИЛИЯ, ДОЛЖНОСТЬ, НАИМЕНОВАНИЕ\_ОТДЕЛА)  
)



(b)

# Пример иерархической модели данных

Если исполнитель принимает участие более чем в одном контракте (связь типа М:N).

Отношение (с)

исполнитель – контракт:

ИСПОЛНИТЕЛЬ (ФАМИЛИЯ,  
ДОЛЖНОСТЬ,  
НАИМЕНОВАНИЕ\_ОТДЕЛА)

КОНТРАКТ (НОМЕР,  
ДАТА, СУММА)



(с)

# Пример иерархической модели данных



(a)



(b)



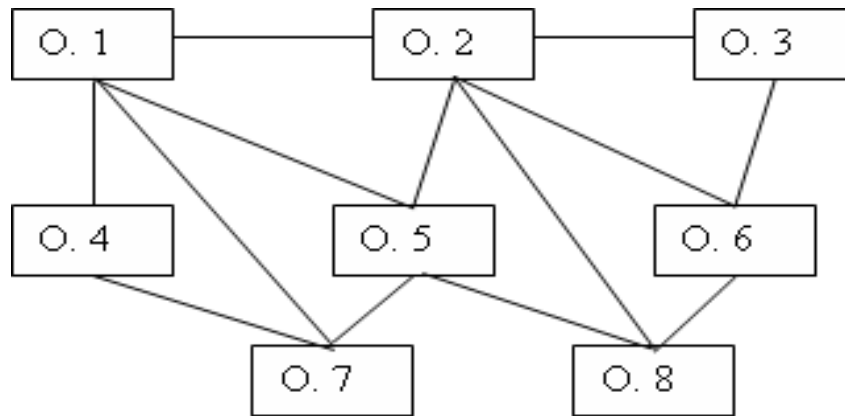
(c)

## Недостатки иерархической модели для этой БД

- Частично дублируется информация между записями СОТРУДНИК и ИСПОЛНИТЕЛЬ (парные записи). В иерархической модели данных не поддерживается соответствие между парными записями.
- Иерархическая модель реализует отношение между исходной и дочерней записью по схеме 1:N, то есть одной родительской записи может соответствовать любое число дочерних. Если присутствует отношение (с ) – опять дублирование информации.
- Затруднения при выполнении операций включения и удаления.

# Сетевая модель

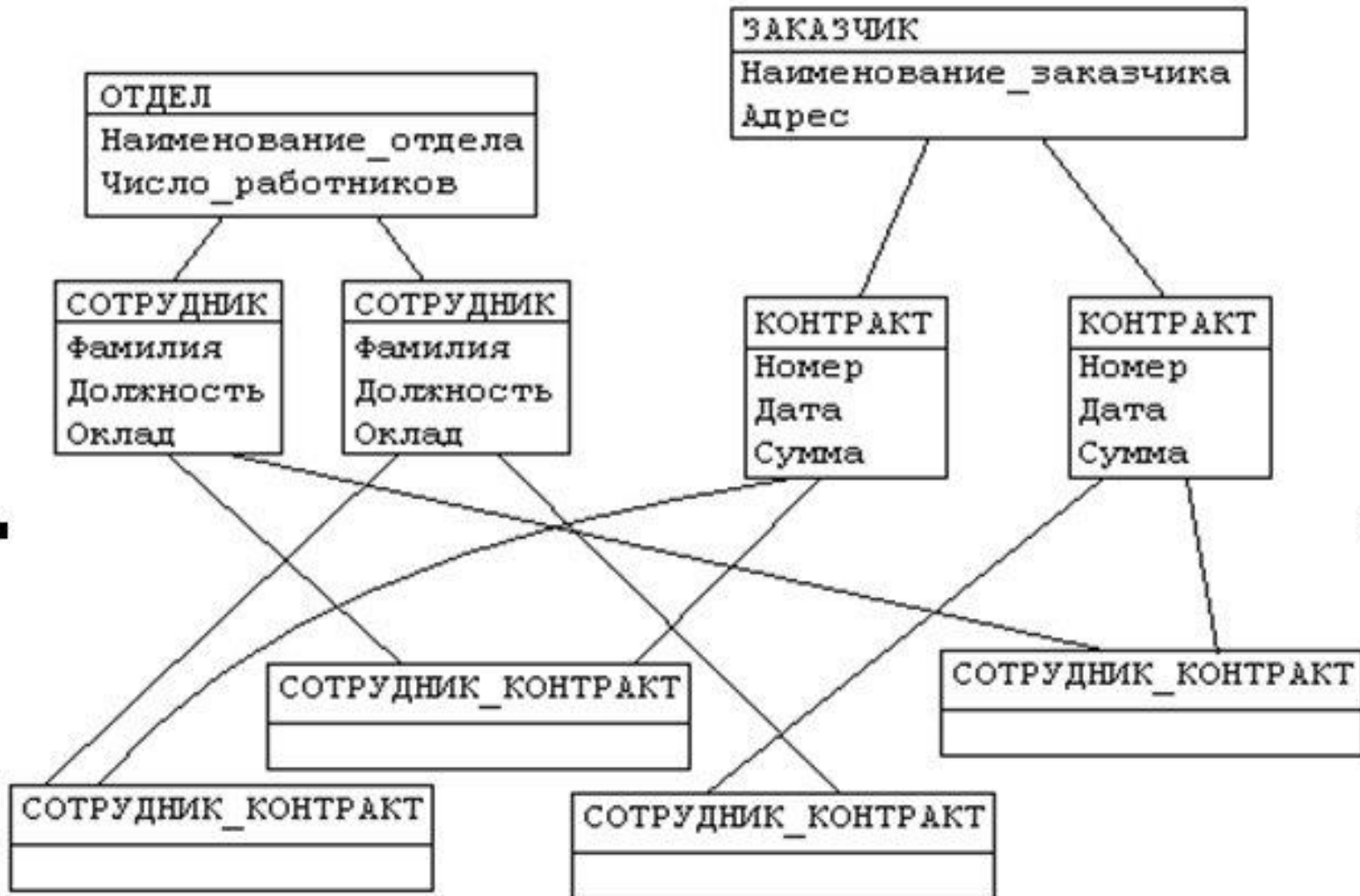
Середина 60-х — система IDS (Integrated Data Store), Чарльз Бачман, премия Тьюринга а работу «Программист как навигатор» . Развитие IDS — создание сетевых СУБД. Основное правило: любой объект может быть одновременно **и главным, и подчиненным**, и может участвовать в образовании любого числа взаимосвязей с другими объектами.



*Сетевые СУБД: IDS, db\_VistaIII, СЕТЬ, СЕТОР, КОМПАС.*

# Преобразование иерархической структуры примера в сетевую

Запись *СОТРУДНИК\_КОНТРАКТ* не имеет полей, служит для связи записей *КОНТРАКТ* и *СОТРУДНИК*.





# Недостатки иерархической и сетевой моделей

- Основными недостатками сетевой модели данных являются: сложная структура памяти и необходимость понижать сложность сетевой модели (исключать имеющиеся циклы).
  - Для выполнения даже простых запросов с использованием переходов и доступом к определенным записям необходимо создавать достаточно **сложные программы**.
  - **Независимость от данных** существует лишь в минимальной степени.
  - Отсутствие общепризнанных **теоретических основ**.

# Реляционная модель

В 1970 году Эдгар Кодд (IBM) предложил реляционную модель данных – данные представляются в виде двумерных таблиц (отношений). Основа – теория множеств и логика предикатов. Достоинство – простота, понятность и удобство физической реализации на ЭВМ.

Проект **System R** (IBM). Цель проекта – доказать практичность реляционной модели. Результат проекта – **разработан структурированный язык запросов SQL**, который стал стандартным языком реляционных СУБД. Реляционные СУБД относятся к СУБД **второго поколения**.

Реляционные СУБД предназначены для **оперативной** обработки информации.

*PCСУБД: dBaseIII Plus и dBase IV, DB2, R:BASE, FoxPro, Visual FoxProe, Paradox, dBase for Windows, Access, Clarion, Ingres, Oracle (до 8 версии), ПАЛЬМА, HyTech.*

# Для того, чтобы таблица была отношением

- Все строки таблицы должны быть уникальны. Уникальность достигается за счет наличия ключей.
- Все строки таблицы должны иметь одну и ту же структуру, т.е. одно и то же количество столбцов с соответственно совпадающими именами.
- Имена столбцов таблицы должны быть различны, а данные в одном столбце должны быть однотипными.
- Значения атрибутов должны быть атомарными (скалярными), т.е. отношения не могут иметь в качестве компонент другие отношения или массивы.
- Порядок следования строк в таблице несущественен, так как влияет лишь на скорость доступа к строке.
- Порядок следования столбцов в таблице несущественен.

# Реляционная модель

special...

doctor : таблица

		id_spec	name
	+	1	терапевт
	+	2	кардиолог
	+	3	хирург
	+	4	окулист
	+	5	лор
▶		нетчик)	

Запись:

6

		id_d	name	sname	patronymic	start_work	id_spec	adress	phone
	+	1	Петр	Петров	Петрович	12.12.2000	1	адрес1	7778866
	+	2	Иван	Иванов	Иванович	30.09.1998	2	адрес2	4569842
	+	3	Степан	Степанов	Степанович	25.05.2006	3	адрес3	7894561
	+	4	Генадий	Генадиев	Генадиевич	20.10.1976	4	адрес4	1538975
	+	5	Василий	Васильев	Васильевич	10.10.2005	5	адрес5	1122334
▶		чик)					0		0

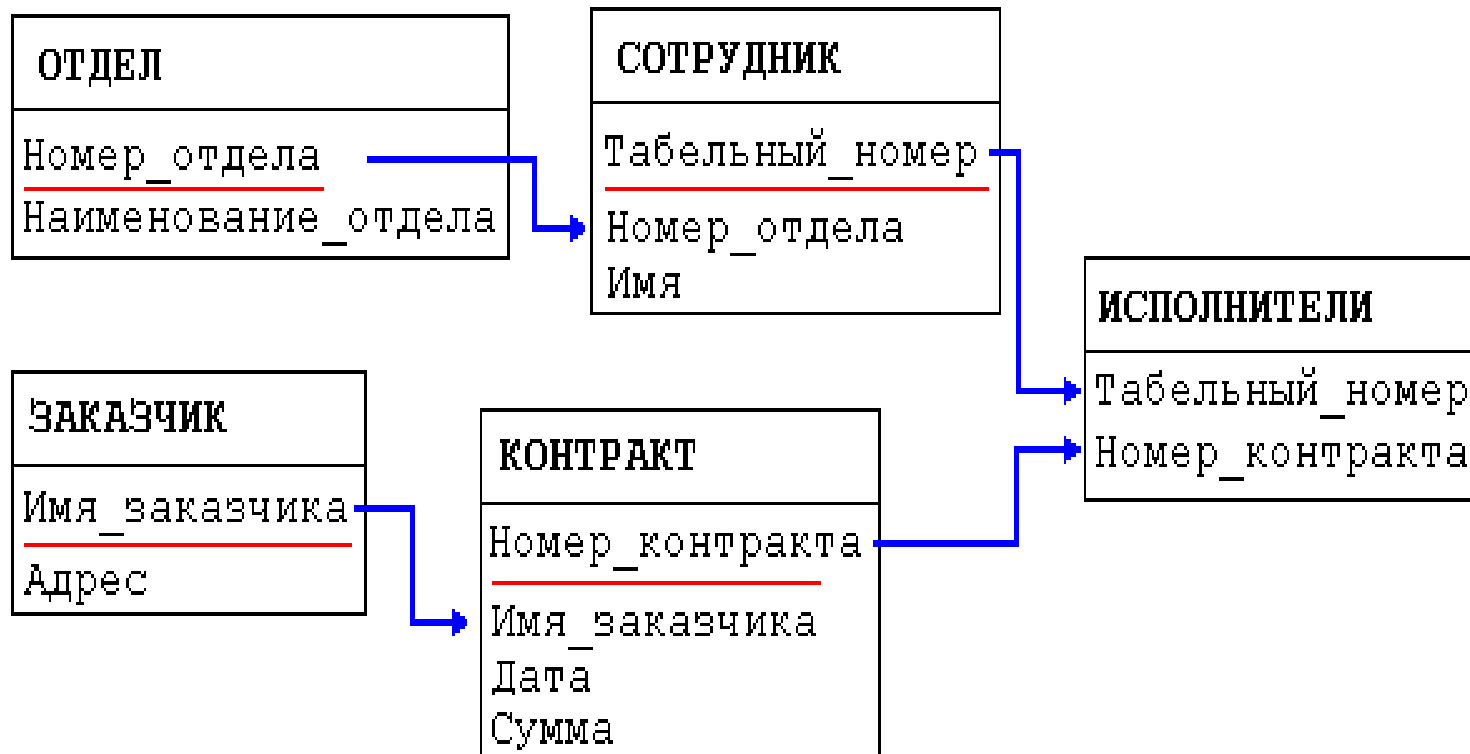
Запись:

6

из 6

# Реляционная модель данных к задаче о предприятии, сотрудниках и контрактах

Для связей между строками разных отношений используется дублирование их ключей. Например, связь между ОТДЕЛ и СОТРУДНИК создается путем копирования первичного ключа "Номер\_отдела" из первого отношения во второе.



## Постреляционная модель данных

Постреляционная модель данных — расширенная реляционная модель. Постреляционная модель допускает **многозначные** поля, значения которых состоят из **подзначений**. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную таблицу.

*Примеры ПРСУБД: uniVers, Bubba, Dasdb.*

# Задача о накладных и товарах. Реляционная модель данных.

Накладные

N накладной	Покупатель
03	Сельпо
83	Дигма
73	Таргет

Накладные-товары

N накладной	Товар	Количество
03	Сыр	3
03	Рыба	2
83	Лимонад	1
83	Сок	6
83	Печенье	2
73	Йогурт	1

# Задача о накладных и товарах. Постреляционная модель данных.

Накладные

N накладной	Покупатель	Товар	Количество
03	Таргет	Сыр	3
		Рыба	2
83	Дигма	Лимонад	1
		Сок	6
		Печенье	2
73	Сельпо	Йогурт	1



# Достоинства и недостатки постреляционной модели

*Достоинство* – возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей. Это обеспечивает высокую наглядность представления информации и повышение эффективности ее обработки.

*Недостаток* – сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

# Системы OLTP

Обработка транзакций в реальном времени OLTP – Online Transaction Processing.

Режим оперативной обработки транзакций OLTP применяется в информационных системах организационного управления для отражения актуального состояния предметной области в любой момент времени. Классическими примерами OLTP-систем являются ввод заказов, розничные продажи и системы финансовых транзакций.

**Реляционные и постреляционные БД служат для оперативной обработки. Это OLTP системы.**

# Системы OLTP

Основные характеристики среды OLTP:

- Короткое время отклика
- Малые транзакции
- Операции по обслуживанию данных
- Большие группы пользователей
- Высокая параллельность
- Большие объемы данных
- Высокая доступность
- Использование данных, связанных с жизненным циклом

# Системы OLTP

Преимущества : высокая надёжность и достоверность данных, как следствие транзакционного подхода. При любом исходе выполнения транзакции целостность данных не нарушается.

Недостатки: OLTP-системы ориентированы на небольшие дискретные транзакции. Обычно аналитические возможности OLTP-систем сильно ограничены. Получение комплексной, аналитической информации порождает сложные соединения таблиц и сложные запросы. Тем не менее это в OLTP-системах реализуется и на первый план выходит проблема оптимизации таких запросов.

# Системы OLAP

Сложность структуры таблиц и большие объемы накопленных данных приводят к снижению скорости выполнения сложных запросов на извлечение данных в системах OLTP.

Для решения задач, диктуемых бизнес-аналитиками, разработан новый подход OLAP (Online Analytical Processing) – технология оперативной аналитической обработки данных, использующая методы и средства для сбора, хранения и анализа многомерных данных в целях поддержки процессов принятия решений. Цель таких систем – проверка гипотез пользователя-аналитика.

# Системы OLAP

В 1993 году Э. Кодд опубликовал статью «Обеспечение OLAP (оперативной аналитической обработки) для пользователей-аналитиков», в которой были изложены 12 законов, заложившие основу концепции аналитической обработки данных в реальном времени. В 1995 году эти правила были дополнены (18).

Некоторые основные правила:

1. Многомерное концептуальное представление данных. Это основа технологии OLAP.
2. Интуитивное манипулирование данными.
3. Пакетное извлечение против интерпретации. Эффективный доступ к собственному хранилищу и к внешним данным.
4. Модели анализа OLAP. Требуется, чтобы OLAP-системы поддерживали формирование настраиваемых отчетов, формирование разрезов и группировок данных

# Системы OLAP

5. Архитектура "клиент-сервер". Требуется также, чтобы различные клиенты могли подключаться с минимумом усилий и программирования.

6. Многопользовательская поддержка. Инструменты OLAP должны обеспечивать одновременный доступ (чтение и запись), интеграцию и конфиденциальность.

7. Сохранение результатов OLAP: хранение их отдельно от исходных данных.

8. Универсальность измерений. Все измерения должны быть равноправны.

9. Неограниченное число измерений и уровней агрегации. Кодд предлагает, что в случае принятия некоторого максимума, он должен обеспечивать хотя бы 15 измерений, а предпочтительнее – 20.

# Многомерная модель

Многомерные базы данных – разновидность реляционной модели, которая использует многомерные структуры для организации данных и выражают отношения между данными.

Многомерные модели данных имеют три важных области применения, связанных с проблематикой анализа данных.

1. Хранилища данных интегрируют для анализа информации из нескольких источников.

2. Системы оперативной аналитической обработки позволяют оперативно получить ответы на запросы, охватывающие большие объемы данных в поисках общих тенденций.

3. Приложения добычи данных служат для выявления знаний за счет полуавтоматического поиска ранее неизвестных шаблонов и связей в базах данных.



# Многомерные СУБД

Многомерные СУБД предназначены для интерактивной аналитической обработки информации в системах поддержки принятия решений.

Основные свойства:

1. Агрегируемость данных – рассмотрение информации на различных уровнях ее обобщения. Например, для уровня аналитик, пользователь, управляющий, руководитель.
2. Историчность данных. Данные привязаны ко времени.
3. Прогнозируемость данных – задание функций прогнозирования и применение их к различным временным интервалам.

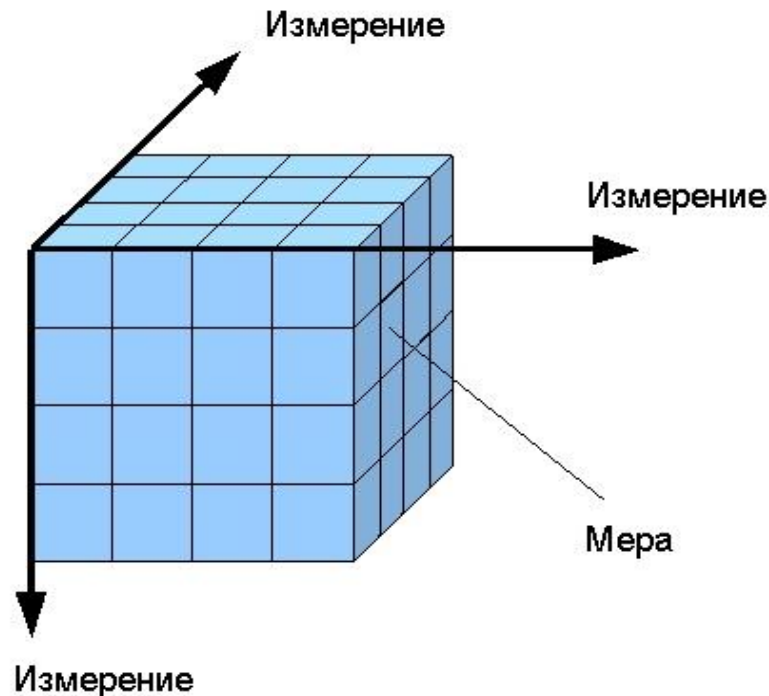
*Примеры систем, поддерживающих многомерное представление данных: Essbase, Media Multi-matrix, Oracle Express Server, Cache.*

# Многомерные СУБД

Основные понятия:

**Измерения** — множество однотипных данных, образующих одну из граней гиперкуба.

**Факты** (меры) — это данные, которые количественно описывают процесс. Тип поля чаще всего определен как цифровой (переменная либо формула).



# Задача о вычислении объема продажи автомобилей.

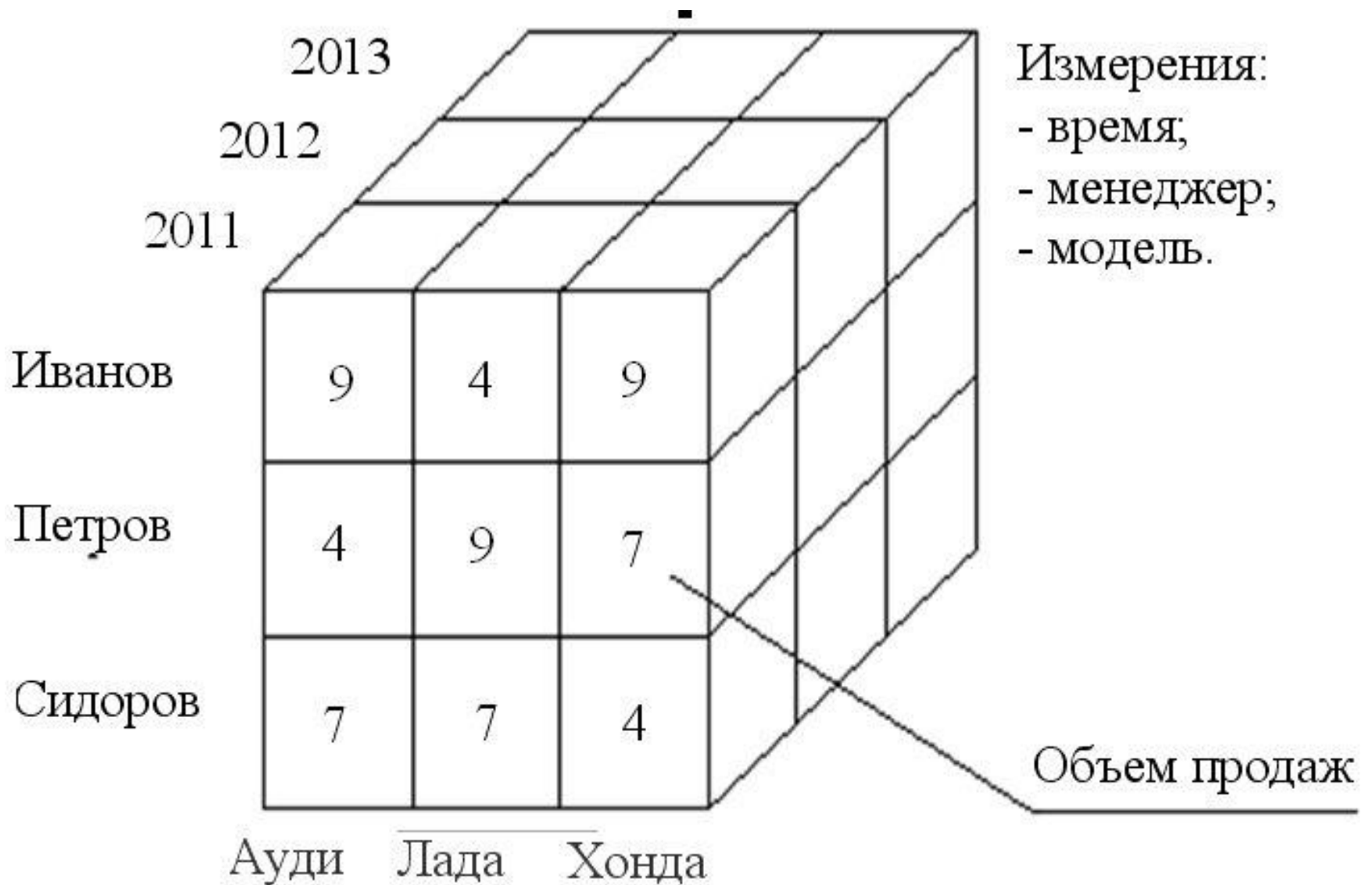
Реляционная модель

Модель	Месяц	Объем
Ауди	июнь	12
Ауди	июль	24
Ауди	август	5
Лада	июнь	2
Лада	июль	18
Хонда	июль	19

Многомерная модель  
(двухмерное представление)

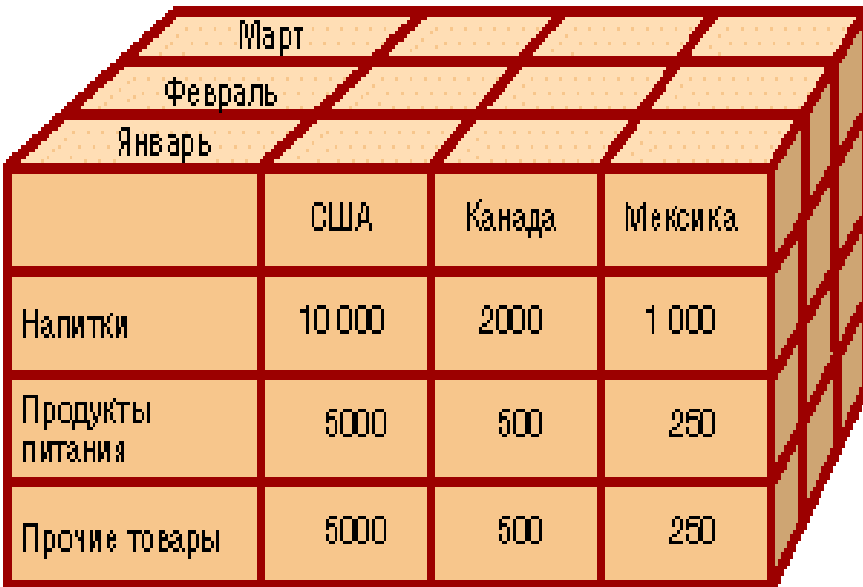
Модель	Июнь	Июль	Август
Ауди	12	24	5
Лада	2	18	No
Хонда	No	19	No

## Пример трехмерной модели .



# Многомерные СУБД

Двумерное представление куба можно получить, "разрезав" его поперек одной или нескольких осей (измерений): мы фиксируем значения всех измерений, кроме двух, - и получаем обычную двумерную таблицу.



A 3D cube diagram representing a multi-dimensional database. The top face of the cube is labeled with the months 'Март' (March), 'Февраль' (February), and 'Январь' (January). The front face is labeled with 'США' (USA), 'Канада' (Canada), and 'Мексика' (Mexico). The side face is labeled with 'Напитки' (Beverages), 'Продукты питания' (Food products), and 'Прочие товары' (Other goods). The cube is divided into a 3x3x3 grid of smaller cells.

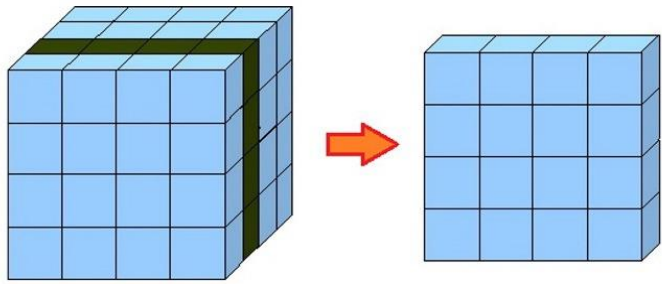
	США	Канада	Мексика
Напитки	10 000	2000	1 000
Продукты питания	5000	500	250
Прочие товары	5000	500	250

Пример двумерного среза куба для одной меры «Продано» и двух "неразрезанных" измерений – «Страна» и «Месяц».

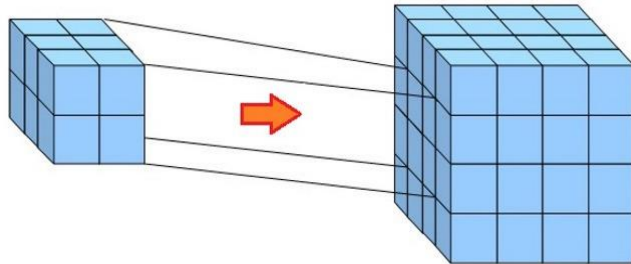
	США	Канада	Мексика
Январь	20 000	4000	2000
Февраль	30 000	6000	3000
Март	50 000	10 000	5000

# Многомерные СУБД

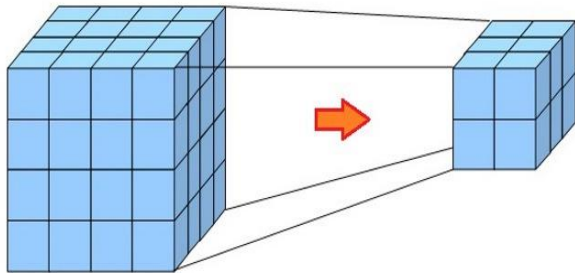
Примеры возможных запросов:



Срез куба

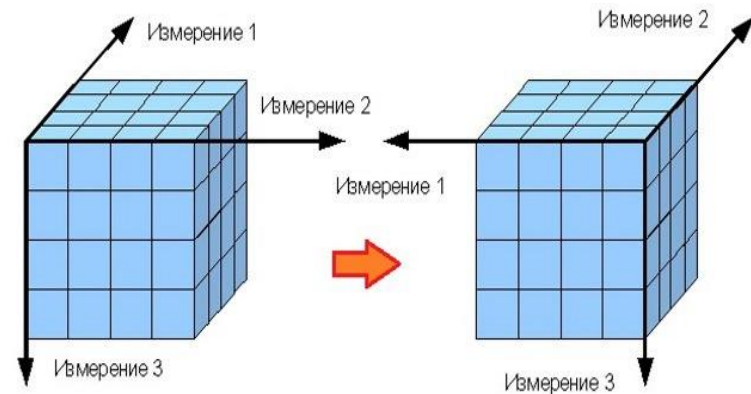


Детализация куба



Консолидация

- комбинирование кубов, которые имеют одно или несколько общих измерений. С точки зрения реляционной алгебры – это join.
- возврат только тех ячеек, которые появляются в верхней или нижней части упорядоченного определенным образом списка.



Поворот, чтобы увидеть данные,  
других измерений