

# Технологии баз данных

<https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads>



<https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>

# Технологии баз данных

Или скачайте бесплатный  
специализированный выпуск



## Developer

SQL Server 2022 Developer — это бесплатный выпуск с полным набором функций, лицензируемый для использования в качестве базы данных для разработки и тестирования и не предназначенный для применения в рабочей среде.



## Express

SQL Server 2022 Express является бесплатным выпуском SQL Server, который идеально подходит для разработки приложений для использования на настольных компьютерах, веб-серверах и других небольших серверах.



# Технологии баз

## данных

Редакции MS SQL Server 2019

**Express** является бесплатной для использования редакцией. Функционал довольно ограничен, самое ощутимое ограничение экспресс версии — максимальный размер базы 10 ГБ. Эта редакция подойдет для небольших проектов, например, студенческих работ или для обучения SQL/T-SQL.

**Standard** это полноценная платная редакция, но многих функций всё еще нет. Максимальный объем оперативной памяти, который сможет использовать SQL Server – 128 ГБ, также отсутствуют группы доступности AlwaysOn и другие компоненты. Standard предназначен для приложений в небольших организациях.

**Enterprise** включает в себя все возможные функции и компоненты, никаких ограничений нет. Корпоративная редакция обычно используется крупными корпорациями или компаниями, которым необходим функционал этой версии.

**Developer** редакция так же как и Enterprise не имеет никаких ограничений и её можно использовать бесплатно, но она может использоваться только для разработки и тестирования приложений.

**Web** редакция почти ничем не отличается от standard, кроме как более сильными ограничениями в функционале и соответственно более низкой стоимости лицензирования;

**Evaluation** — ознакомительная редакция SQL Server, которая предоставляет полный функционал Enterprise и работает в течении 180 дней (может быть [обновлена до полноценной версии](#)).

<https://info-comp.ru/install-ms-sql-server-2019-express>

<https://winitpro.ru/index.php/2020/01/15/guide-ustanovka-ms-sql-server-best-practices/>

# Технологии баз данных

## Литература:

1. К. Дж. Дэйт , Введение в системы баз данных
2. Карпова Т.С. , Базы данных: модели, разработка, реализация
3. Мартин Фаулер, Прамодкумар Дж. Садаладж  
NoSQL. Новая методология разработки нереляционных баз  
данных
4. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных
5. Мартин Грабер, Введение в SQL
6. Власова, О. В., Системы управления базами данных
7. Власова, О. В., SQL
8. Власова, О. В., Transact-SQL
9. Посполит А.В. Visual Studio .NET: разработка приложений баз  
данных

# Технологии баз

# данных

## Тренажеры SQL(TSQL)

[www.sql-ex.ru](http://www.sql-ex.ru)

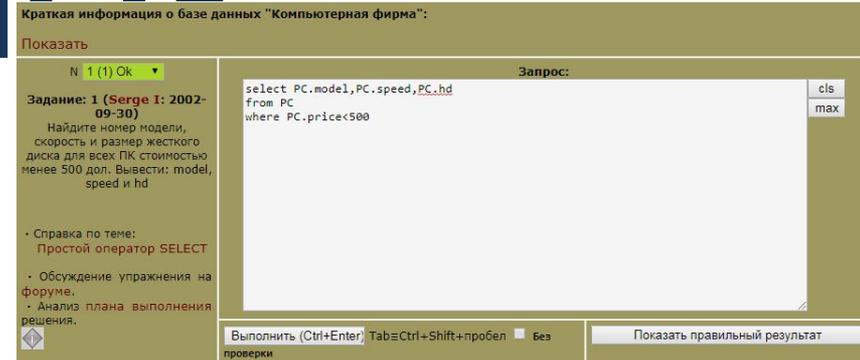
<http://sql-tutorial.ru/content.html>

<https://stepik.org/course/63054/syllabus>

<http://sqlcontest.uniyar.ac.ru/>

## SQL-ИНЪЕКЦИИ

<http://sql.training.hackerdom.ru/>



Интерактивный тренажер  
по SQL

# Технологии баз данных

1. Основные понятия и определения теории баз данных
2. Структура СУБД и основные функции
3. Классификация моделей данных
4. Метод “сущность-связь” проектирования баз данных
5. Реляционная модель
6. Процесс нормализации отношений.

# **Технологии баз данных**

**Тема 1. Основные понятия и  
определения теории баз данных**

## **Автоматизированная**

**информационная система** – это функционирующий на основе ЭВМ программно-аппаратный комплекс, обеспечивающий сбор, хранение, актуализацию и обработку информации в целях поддержки какого-либо вида деятельности.

**Предметная область** – часть реального мира подлежащая изучению с целью организации управления и, в конечном счете, автоматизации.

## *Классификация автоматизированных информационных систем по типу хранимой и обрабатываемой информации:*

### **□ Документальные системы**

Служат для работы с текстами на естественном языке

### **□ Фактографические системы**

Оперируют фактическими данными, представленными в виде специальным образом организованных совокупностей записей.

### **□ Лексикографические системы**

Словари, классификаторы.

## *Недостатки файловой организации данных на внешних носителях.*

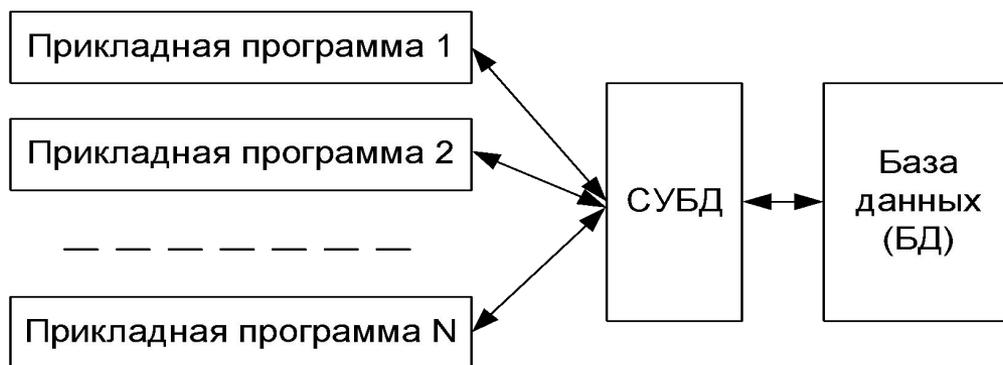
- ✓ *Зависимость программ от данных.*
- ✓ *Одновременный доступ к данным.*
- ✓ *Разграничение прав доступа к данным.*
- ✓ *Дублирование данных.*

# ПРИЧИНЫ ВОЗНИКНОВЕНИЯ СИСТЕМ БАЗ ДАННЫХ

---

## *Решение:*

Было предложено отделить процесс хранения данных от процесса их обработки, выделив специальную программу-посредник – **систему управления базами данных (СУБД)**, а сами данные организовать и хранить в виде определенной структуры – **базы данных**.



# ПРИЧИНЫ ВОЗНИКНОВЕНИЯ СИСТЕМ БАЗ ДАННЫХ

---

**Банк данных** – это система специальным образом организованных данных (баз данных), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

В англоязычной литературе понятие по сути близкое к банку данных, обозначается термином **система баз данных (database system)**

# ПРИЧИНЫ ВОЗНИКНОВЕНИЯ СИСТЕМ БАЗ ДАННЫХ

---

**База данных** – именованная совокупность данных, отражающая состояние объектов и их отношений в заданной предметной области.



## Основные компоненты системы баз данных

### 1. Данные

1. *Постоянные*
2. *Транзитные (промежуточные результаты, входные, выходные)*
3. *Метаинформация*

*SINGL-USER SYSTEM*

*MULTI-USER SYSTEM*

*Данные должны быть интегрированными и общими.*

### 2. Аппаратное обеспечение

1. *Устройства хранения данных*
2. *Устройства обработки данных*

## **Основные компоненты системы баз данных**

### **3. Программное обеспечение**

*Система управления базами данных (СУБД) – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.*

### **4. Пользователи**

- 1. Прикладные программисты*
- 2. Конечные пользователи*
- 3. Администраторы данных и администраторы БД.*

# Этапы развития СУБД

---

В 1968 году была введена в эксплуатацию первая промышленная СУБД система *IMS (Information Management System)* фирмы *IBM*.

В 1975 году появился первый стандарт ассоциации по языкам систем обработки данных — *Conference of Data System Languages (CODASYL)*, который определил ряд фундаментальных понятий в теории систем баз данных, которые и до сих пор являются основополагающими для *сетевой модели данных*.

В дальнейшее развитие теории БД большой вклад был сделан американским математиком Эдгаром Франком Коддом (*Edgar Frank Codd, 1923-2003*), который является создателем реляционной модели данных.

В 1970 г. он опубликовал статью «*A Relation Model of Data for Large Shared Data Banks*», в которой описал основные идеи реляционного подхода.

В 1981 году Э. Ф. Кодд получил за создание **реляционной модели** и **реляционной алгебры** престижную премию Тьюринга Американской ассоциации по вычислительной технике.

# Этапы развития СУБД

---

**Первый этап** развития СУБД связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11, разных моделях HP.

**Особенности этого этапа развития выражаются в следующем:**

Все СУБД базируются на мощных мультипрограммных операционных системах (MVS, SVM, RTE, OSRV, RSX, UNIX), поэтому в основном поддерживается работа с централизованной базой данных в режиме распределенного доступа.

Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС).

Поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным.

Значительная роль отводится администрированию данных.

Проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и была создана первая система (System R), реализующая идеологию реляционной модели данных.

Проводятся теоретические работы по оптимизации запросов и управлению распределенным доступом к централизованной БД, было введено понятие транзакции. Результаты научных исследований открыто обсуждаются в печати, идет мощный поток общедоступных публикаций, касающихся всех аспектов теории и практики баз данных, и результаты теоретических исследований активно внедряются в коммерческие СУБД. 17

# Этапы развития СУБД

---

## Эпоха персональных компьютеров (Настольные СУБД)

### *Особенности этого этапа следующие:*

Все СУБД были рассчитаны на создание БД в основном с монопольным доступом.

Большинство СУБД имели развитый и удобный пользовательский интерфейс. В большинстве существовал *интерактивный режим* работы с БД как в рамках описания БД, так и в рамках проектирования запросов.

Инструментальная среда состояла из готовых элементов приложения в виде шаблонов экранных форм, отчетов, этикеток (Labels), графических конструкторов запросов, которые достаточно просто могли быть собраны в единый комплекс.

При наличии высокоуровневых языков манипулирования данными типа *реляционной алгебры* и SQL в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.

В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных.

Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД и в связи с этим — к отсутствию инструментальных средств администрирования БД.

Сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД.

Яркие представители этого семейства — очень широко использовавшиеся до недавнего

СУБД *Phase (Phase III, Phase IV), FoxPro, Clipper, Paradox*

# Этапы развития СУБД

---

## Распределенные базы данных (Клиент-серверная технология)

*Особенности данного этапа:*

Практически все современные СУБД обеспечивают поддержку **полной реляционной модели**, а именно:

**структурной целостности** — допустимыми являются только данные, представленные в виде отношений реляционной модели;

**языковой целостности**, то есть языков манипулирования данными высокого уровня (в основном SQL);

**ссылочной целостности**, контроля за соблюдением ссылочной целостности в течение всего времени функционирования системы, и гарантий невозможности со стороны СУБД нарушить эти ограничения.

Большинство современных СУБД рассчитаны на многоплатформенную архитектуру.

Необходимость поддержки многопользовательской работы с базой данных и возможность децентрализованного хранения данных потребовали развития средств администрирования БД с реализацией общей концепции средств защиты данных.

Потребность в новых реализациях вызвала создание серьезных теоретических трудов по оптимизации реализаций распределенных БД и работе с

# Этапы развития СУБД

---

## Распределенные базы данных

### *Особенности данного этапа:*

Для того чтобы не потерять клиентов, которые ранее работали на настольных СУБД, практически все современные СУБД имеют средства подключения клиентских приложений, разработанных с использованием настольных СУБД, и средства экспорта данных из форматов настольных СУБД второго этапа развития. Разработка ряда стандартов в рамках языков описания и манипулирования данными начиная с SQL89, SQL92, SQL99 и технологий по обмену данными между различными СУБД, к которым можно отнести и протокол ODBC (Open DataBase Connectivity), предложенный фирмой Microsoft.

Начало работ, связанных с концепцией объектно-ориентированных БД — ООБД.

Яркие представители этого семейства

### **MS Access, Oracle, MS SQL, MY SQL, PostgreSQL**

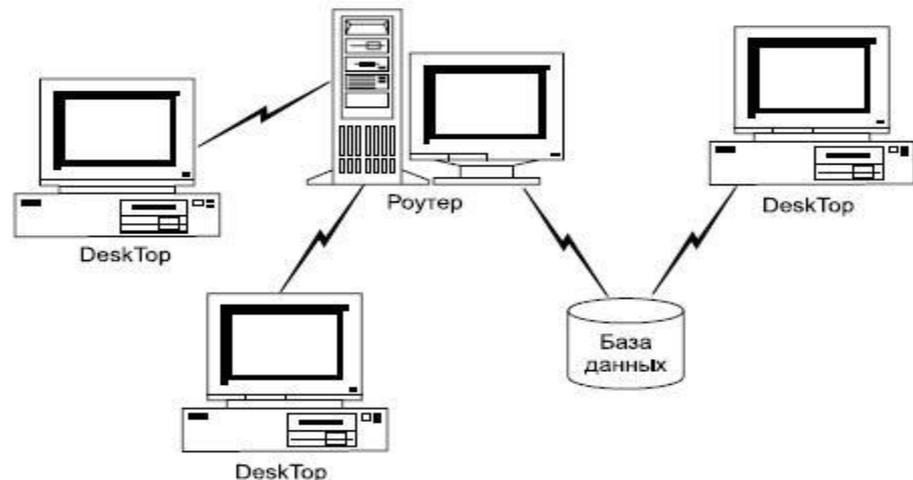
и другие современные серверы баз данных, которых в настоящий момент насчитывается несколько десятков.

# Этапы развития СУБД

---

**Интранет** - отпадает необходимость использования специализированного клиентского программного обеспечения. Для работы с удаленной базой данных используется стандартный браузер Интернета.

Встроенный в загружаемые пользователем HTML-страницы код, написанный обычно на языке *Java*, *Java-script*, *Perl* и других, отслеживает все действия пользователя и транслирует их в низкоуровневые SQL-запросы к базе данных



# Этапы развития СУБД

---



Термин «**NoSQL**» имеет абсолютно стихийное происхождение и не имеет общепризнанного определения или научного учреждения за спиной. Это название скорее характеризует вектор развития ИТ в сторону от реляционных баз данных.

Расшифровывается как **Not Only SQL**, хотя есть сторонники и прямого определения No SQL.

## NoSQL базы в-основном open-source и созданы в 21 столетии.



NoSQL движение набирает популярность гигантскими темпами. Однако это не означает, что реляционные базы данных становятся рудиментом или чем-то архаичным. Скорее всего они будут использоваться и использоваться по-прежнему активно, но все больше в симбиозе с ними будут выступать NoSQL базы. Мы вступаем в эру polyglot persistence — эру, когда для различных потребностей используются разные хранилища данных. Теперь нет монополизма реляционных баз данных, как безальтернативного источника данных. Все чаще архитекторы выбирают хранилище исходя из природы самих данных и того, как мы ими хотим манипулировать, какие объемы информации ожидаются.

# Категории NoSQL баз

## Хранилища, построенные как «ключ/значение»

(примеры — [Memcached](#), [Redis](#), [Oracle NoSQL Database](#));

## Колоночные (или разреженные табличные) хранилища

(примеры — [Google BigTable](#), [Cassandra](#), [Amazon SimpleDB](#), [Hbase](#));

## Документо-ориентированные

(примеры — [CouchDB](#) и [MongoDB](#), есть попытка создания и спецификации общего языка такого рода — [UnQL](#));

## Графовые хранилища

(очень распространены в социальных сетях, примеры: [Neo4J](#), [Infinite Graph](#), [Bigdata](#));

## XML-хранилища

(примеры: [Mark Logic Server](#), [EMC Documentum](#), [eXist](#)).



## ***Жизненный цикл системы обработки данных состоит из нескольких этапов:***

- 1) проектирование базы данных;*
- 2) проектирование приложений;*
- 3) реализация базы данных, включающая начальную загрузку и запуск в эксплуатацию;*
- 4) разработка специальных средств администрирования базы данных;*
- 5) эксплуатация базы данных (анализ функционирования, модификация и адаптация).*

# ТРЕХУРОВНЕВАЯ АРХИТЕКТУРА БАЗЫ ДАННЫХ

---

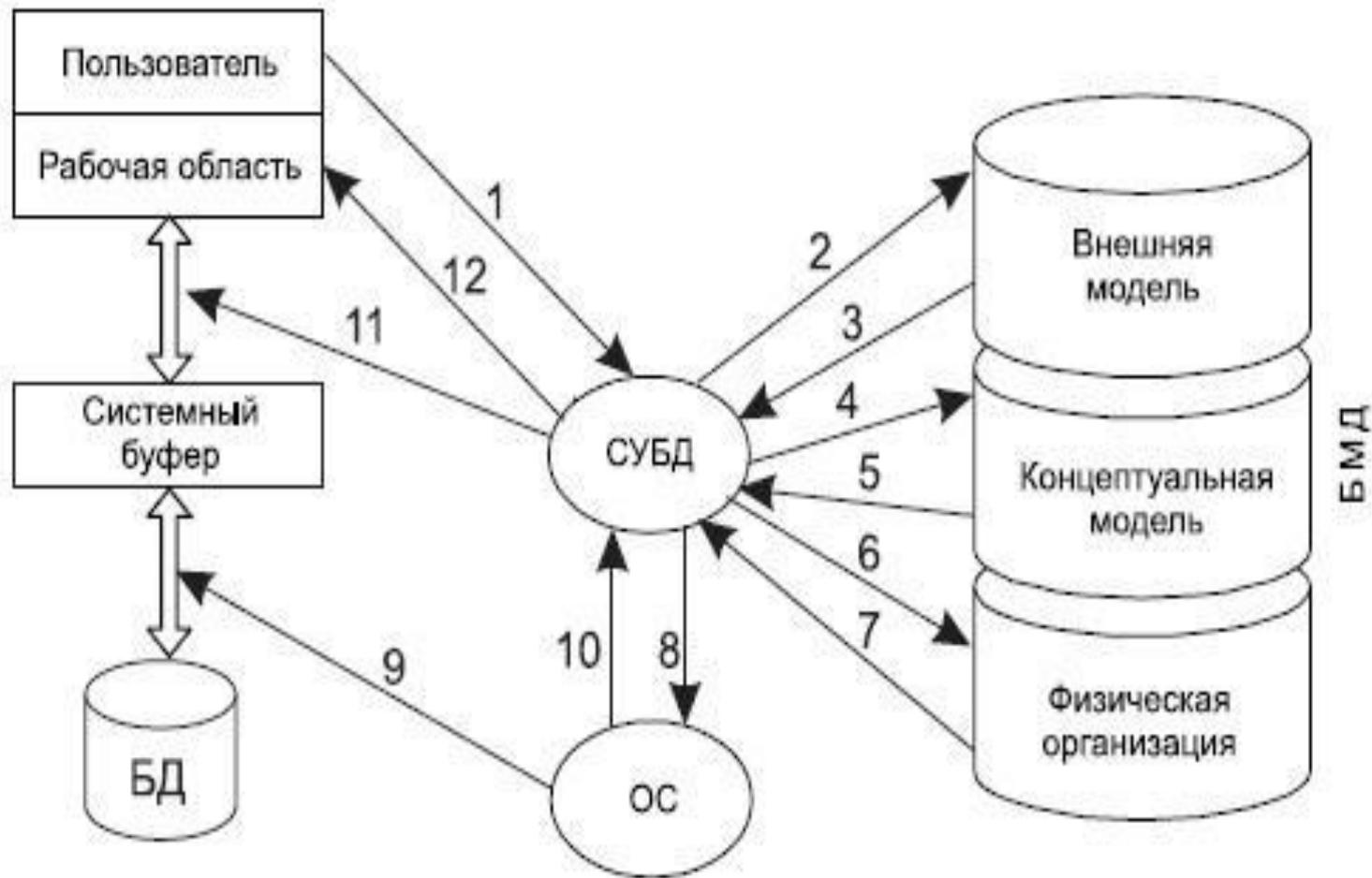
**Архитектура системы БД - это совокупность её функциональных компонентов, а также средств обеспечения их взаимодействия друг с другом, с пользователями и системным персоналом.**  
**Архитектура ANSI/SPARC 1975 г.**



# ТРЕХУРОВНЕВАЯ АРХИТЕКТУРА БАЗЫ ДАННЫХ



# ТРЕХУРОВНЕВАЯ АРХИТЕКТУРА БАЗЫ ДАННЫХ



# Технологии баз данных

Тема 2. Структура СУБД и основные  
функции

# СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

---

**Система управления базами данных** представляет собой совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

## **Основные функции СУБД:**

- ✓ управление данными во внешней памяти;
- ✓ управление буферами оперативной памяти;
- ✓ управление транзакциями;
- ✓ журнализация и восстановление БД после сбоев.
- ✓ поддержка языков БД;

# ЧАСТИ СУБД

---

**Ядро СУБД (Data Base Engine )** – это набор программных модулей, необходимый и достаточный для создания и поддержания БД, то есть универсальная часть, решающая стандартные задачи по информационному обслуживанию пользователей.

**Процессор языка БД** обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода

**Подсистему поддержки времени исполнения** -которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД

**Сервисные программы(утилиты)** предоставляют пользователям ряд дополнительных возможностей и услуг, зависящих от описываемой предметной области и потребностей конкретного пользователя.

# **Технологии баз данных**

## **Тема 3. Классификация моделей данных**

# Классификация моделей данных

---

Понятие "**данные**" в концепции баз данных — это набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы.

*Пример данных: Петров Николай Степанович, \$30*

**Модель данных** – это некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, т. е. как сведения, содержащие не только данные, но и взаимосвязи между ними.

# Классификация моделей данных



# Иерархическая модель данных (СУБД IMS (IBM))

---

Основными информационными единицами в иерархической модели являются: **база данных (БД), сегмент и поле.**

**Поле данных** - как минимальная, неделимая единица данных, доступная пользователю с помощью СУБД.

**Сегмент** в терминологии Американской Ассоциации по базам данных DBTG (Data Base Task Group) называется *записью*, при этом в рамках иерархической модели определяются два понятия: *тип сегмента* или *тип записи* и *экземпляр сегмента* или *экземпляр записи*.

**Тип сегмента** — это поименованная совокупность типов элементов данных, в него входящих. **Экземпляр сегмента** образуется из конкретных значений полей или элементов данных, в него входящих. Каждый тип сегмента в рамках иерархической модели образует некоторый набор однородных записей.

# Иерархическая модель данных

---

В иерархической модели сегменты объединяются в ориентированный древовидный граф.

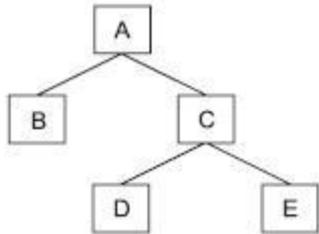


Схема иерархической БД представляет собой совокупность отдельных деревьев, каждое дерево в рамках модели называется **физической базой данных**.

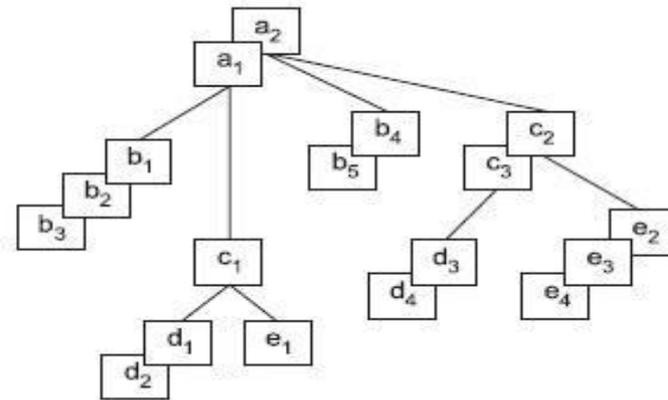
Каждая физическая БД удовлетворяет следующим иерархическим ограничениям:

- в каждой физической БД существует один корневой сегмент, то есть сегмент, у которого нет логически исходного (родительского) типа сегмента;
- каждый логически исходный сегмент может быть связан с произвольным числом логически подчиненных сегментов;
- каждый логически подчиненный сегмент может быть связан только с одним логически исходным (родительским) сегментом.

# Иерархическая модель данных



Пример структуры иерархического дерева



Пример двух экземпляров данного дерева



# Сетевая модель данных (СУБД IDS (General

**Electric))**  
IDMS(Integrated Database Management System) (с 1970 г.)

**Базовыми объектами модели являются:**

**Элемент данных** — минимальная информационная *единица*, доступная пользователю СУБД.

**Агрегат данных** соответствует следующему уровню обобщения в модели.(именованная совокупность элементов)

В модели определены агрегаты двух типов: агрегат типа *вектор* и агрегат типа *повторяющаяся группа*.

Адрес			
Город	Улица	Дом	Квартира

Зарплата	
Месяц	Сумма
...	...

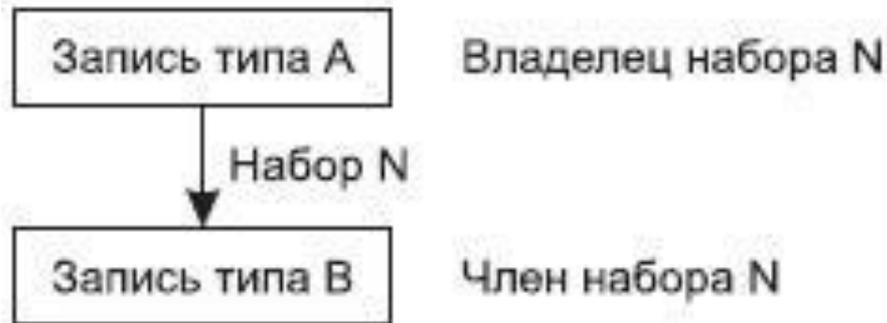
**Записью** называется совокупность агрегатов или элементов данных, моделирующая некоторый *класс* объектов реального мира.

# Сетевая модель данных

---

**Базовыми объектами модели являются:**

**Набором** называется двухуровневый *граф*, связывающий отношением «*один-ко-многим*» два типа записи.

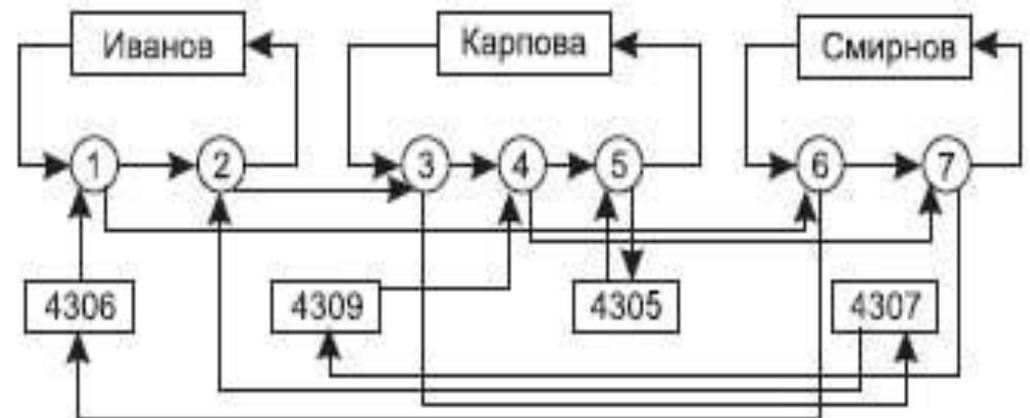
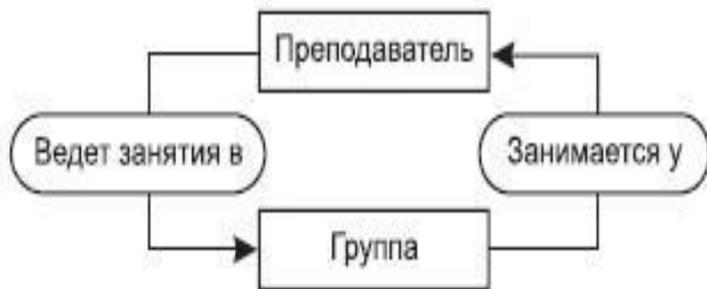


Для любых двух типов записей может быть задано любое количество наборов, которые их связывают. Отношение «**многие-ко-многим**»

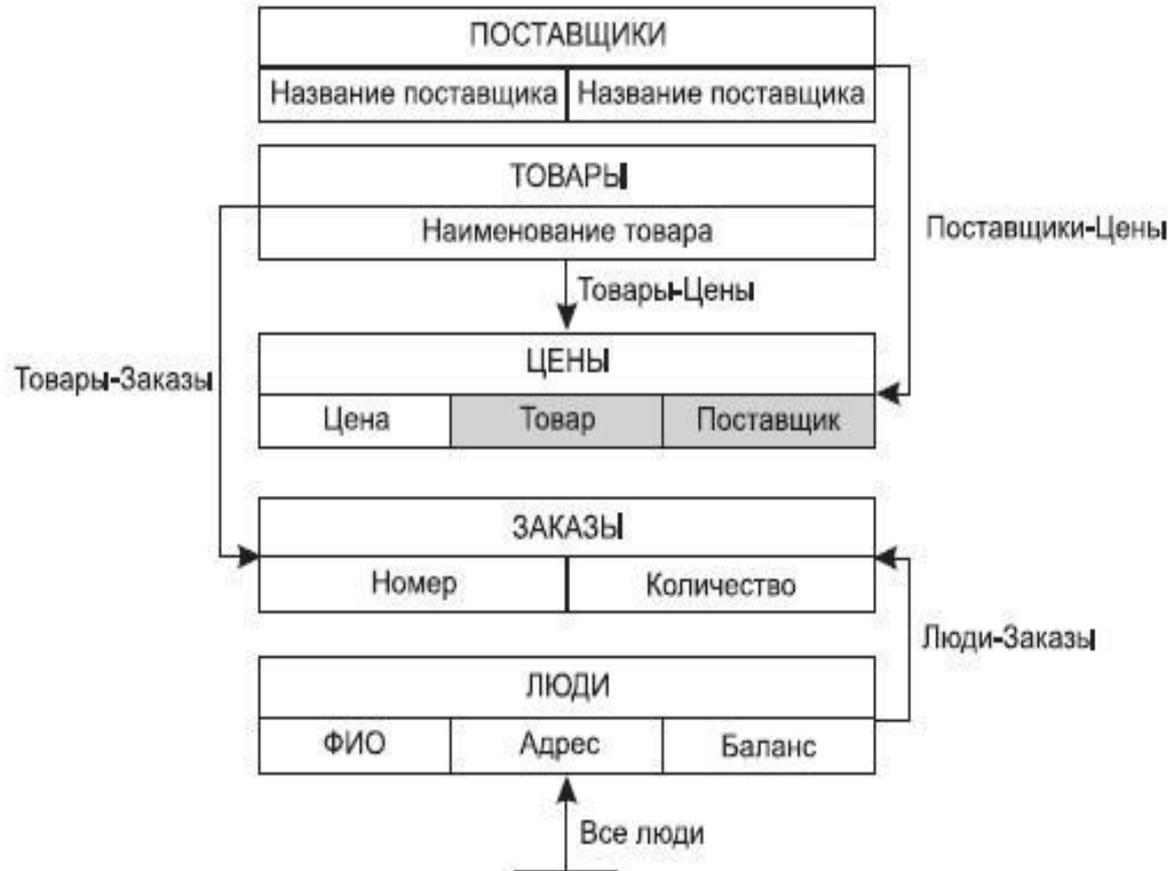


# Сетевая модель данных

Преподаватель	Группа	День недели	№ пары	Аудитория	Дисциплина
Иванов	4306	Понедельник	1	22-13	КИД
Иванов	4307	Понедельник	2	22-13	КИД
Карпова	4307	Вторник	2	22-14	БЗ и ЭС
Карпова	4309	Вторник	4	22-14	БЗ и ЭС
Карпова	4305	Вторник	1	22-14	БД
Смирнов	4306	Вторник	3	23-07	ГВП
Смирнов	4309	Вторник	4	23-07	ГВП



# Сетевая модель данных



# Реляционная модель данных

---

Фамилия	Дисциплина	Оценка
Иванов	Теория автоматов	4
Иванов	Базы данных	3
Крылов	Теория автоматов	5
Степанов	Теория автоматов	5
Степанов	Базы данных	4

Каждое отношение представляется в виде таблицы и обладает рядом специфических свойств:

В таблице нет двух одинаковых строк.

Порядок строк в таблице произвольный.

Каждый атрибут в отношении имеет уникальное имя.

Значение атрибутов атомарны.

# Постреляционная модель данных

---

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.

Модель	Месяц	Объём
Mark II	Май	10
	Июнь	25
Chaser	Май	5
	Июнь	15
Cresta	Май	10
	Июнь	12

# Многомерная модель данных

---

Многомерная модель представляет собой *узкоспециализированную модель* предназначенной *для интерактивной аналитической обработки информации*.

**OLAP** (OnLine Analytical Processing — оперативная аналитическая обработка) технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу.

**OLTP** (Online Transaction Processing — транзакционная система — обработка транзакций в реальном времени) Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика.

# Многомерная модель данных

---

Модель	Месяц	объём
Mark II	Май	10
Mark II	Июнь	25
Chaser	Май	5
Chaser	Июнь	15
Cresta	Май	10
Cresta	Июнь	12

Модель	Май	Июнь
Mark II	10	25
Chaser	5	15
Cresta	10	12

# Многомерная модель данных

	2004	2005	2006
Петров	9	4	8
Иванов	6	5	11
Сидоров	3	18	20

MarkII      Chaser      Cresta

# Многомерная модель данных

---

Ячейка или показатель это понятия значение которых однозначно определяется фиксированным набором измерений

Измерение это множество однотонных данных, образующих одну из данных гипер куба.

Историчность

Срез это подмножество гипер куба полученное в результате фиксации одного или нескольких измерений

Вращение

Агрегируемость

Прогнозирование данных

# Технологии баз данных

Тема 4. Метод “сущность-связь”  
проектирования баз данных

# Метод "Сущность – Связь" (Entity–Relation,

## ER–method)

Модель “**Сущность-Связь**” была предложена в **1976** году **Питером Пин-Шен Ченом** (англ. *Peter Pin-Shen Chen*), американским профессором компьютерных наук в университете штата Луизиана.

**Сущность** – это любой различимый объект ПО, о котором в системе будет накапливаться информация.

Сущности бывают как **физически существующие** (например, КЛИЕНТ или АВТОМОБИЛЬ), так и **абстрактные** (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают **тип сущности** и **экземпляр**. Тип характеризуется именем и списком <sup>Товарищ</sup> **свойств**, а экземпляр – конкретными значениями свойств.

<u>КЛИЕНТ</u>
Фамилия
Дата рождения
Телефон
Место жительства

Экземпляр сущности

Иванов
21.05.87
43-90-78
Нижний Новгород

Типы сущностей можно классифицировать как **сильные** и **слабые**.

Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных.

# Метод "Сущность – Связь" (Entity–Relation,

## ER–method)

*Информация о сущности представляется совокупностью*

*атрибутов.*

**Атрибут** – поименованное свойство (характеристика) сущности.

**Атрибут** представляет информационное описание **количественных** или **качественных** свойств сущности, описывает **состояние** сущности, позволяет **идентифицировать** сущность.

**Спецификация атрибута** состоит из его **названия**, указания **типа данных** и **описания ограничений целостности** – допустимого множества значений (или **домена**), которые может принимать данный атрибут.

# Метод "Сущность – Связь" (Entity–Relation,

ER–method)  
**Различают:**

## **Идентифицирующие и описательные атрибуты.**

Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). Остальные атрибуты называются описательными и заключают в себе интересующие свойства сущности.

## **Составные и простые атрибуты**

Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

## **Однозначные и многозначные атрибуты**

(могут иметь соответственно одно или много значений для каждого экземпляра сущности).

## **Основные и производные атрибуты**

Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения<sup>51</sup> и текущей даты).

## Метод "Сущность – Связь" (Entity–Relation, ER–method)

---

**Связь** - ассоциирование двух или более сущностей. Каждая связь характеризуется **именем, степенью, типом и обязательностью**.

Связь является **обязательной**, если в данной связи должен участвовать каждый экземпляр сущности, **факультативной** — если не каждый экземпляр сущности должен участвовать в данной связи. При этом связь может быть **обязательной** с одной стороны и **факультативной** с другой стороны.

**Число типов сущностей**, участвующих в связи, называется **степенью связи**  $n = 2, 3, \dots$

# Метод "Сущность – Связь" (Entity–Relation, ER–method)

---

## **Типы связей:**

**Связь 1:1.** Каждый экземпляр сущности *A* связан с не более чем одним экземпляром сущности *B* и наоборот.

Примером является связь между типами сущностей СПЕЦИАЛЬНОСТЬ и УЧЕБНЫЙ ПЛАН ПО СПЕЦИАЛЬНОСТИ (каждой специальности соответствует свой учебный план по специальности).

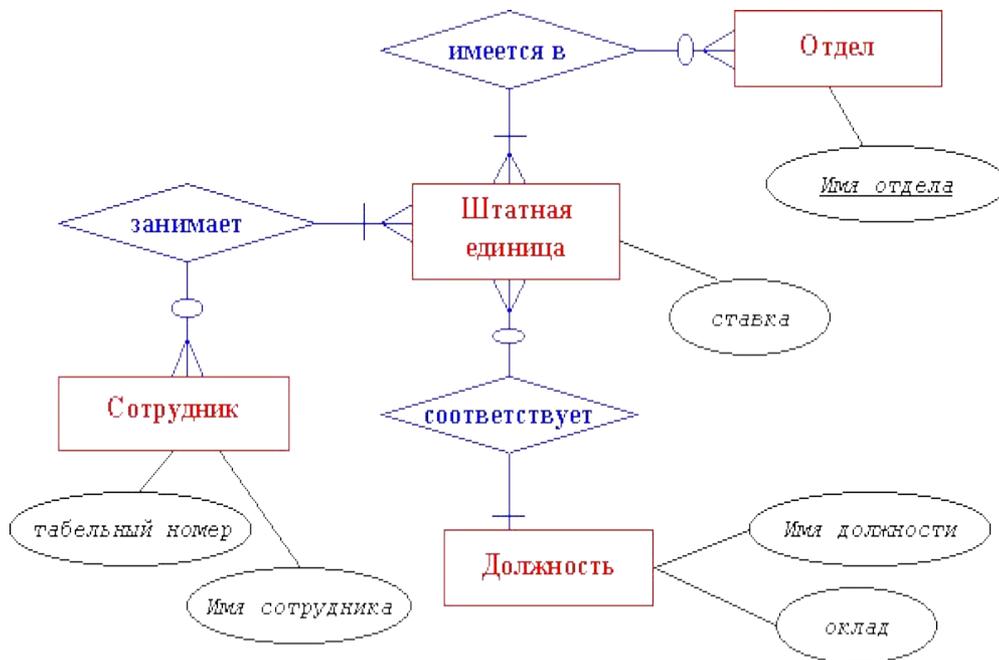
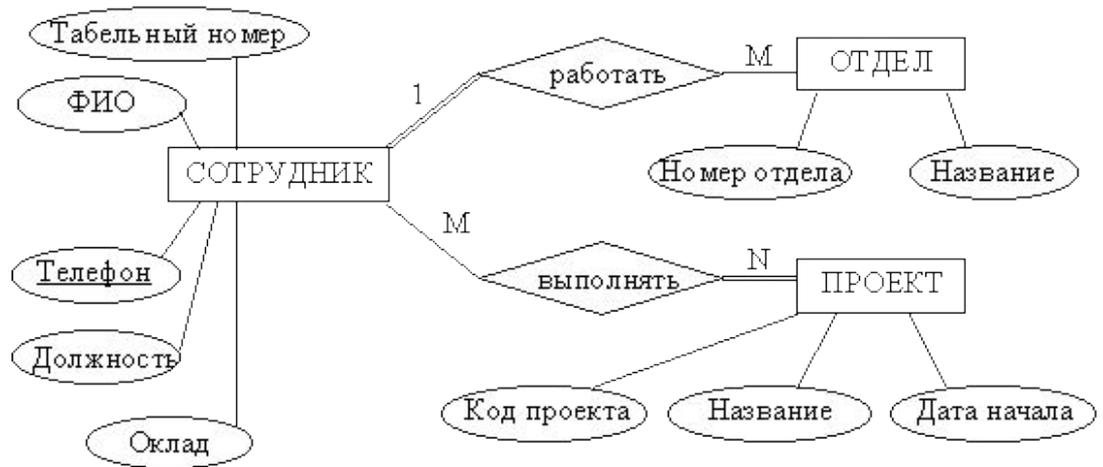
**Связь 1:M.** Каждый экземпляр сущности *A* связан со многими экземплярами сущности *B*, но каждый экземпляр сущности *B* связан с не более чем одним экземпляром сущности *A*.

Примером является связь между типами сущностей СПЕЦИАЛЬНОСТЬ и СТУДЕНТ (на одной специальности учатся много студентов, а каждый студент учится на одной специальности).

**Связь M:N.** Несколько экземпляров сущности *A* связаны с несколькими экземплярами сущности *B* и наоборот. Примером является связь между типами сущностей ФАКУЛЬТЕТ и СПЕЦИАЛЬНОСТЬ (на факультете может быть несколько специальностей и одна и та же специальность может быть на нескольких факультетах).

# Метод "Сущность – Связь" (Entity–Relation, ER–method)

## ER - диаграмма



# Метод "Сущность – Связь" (Entity–Relation, ER–method)

## ER - диаграмма



# Метод "Сущность – Связь" (Entity–Relation, ER–method)

---

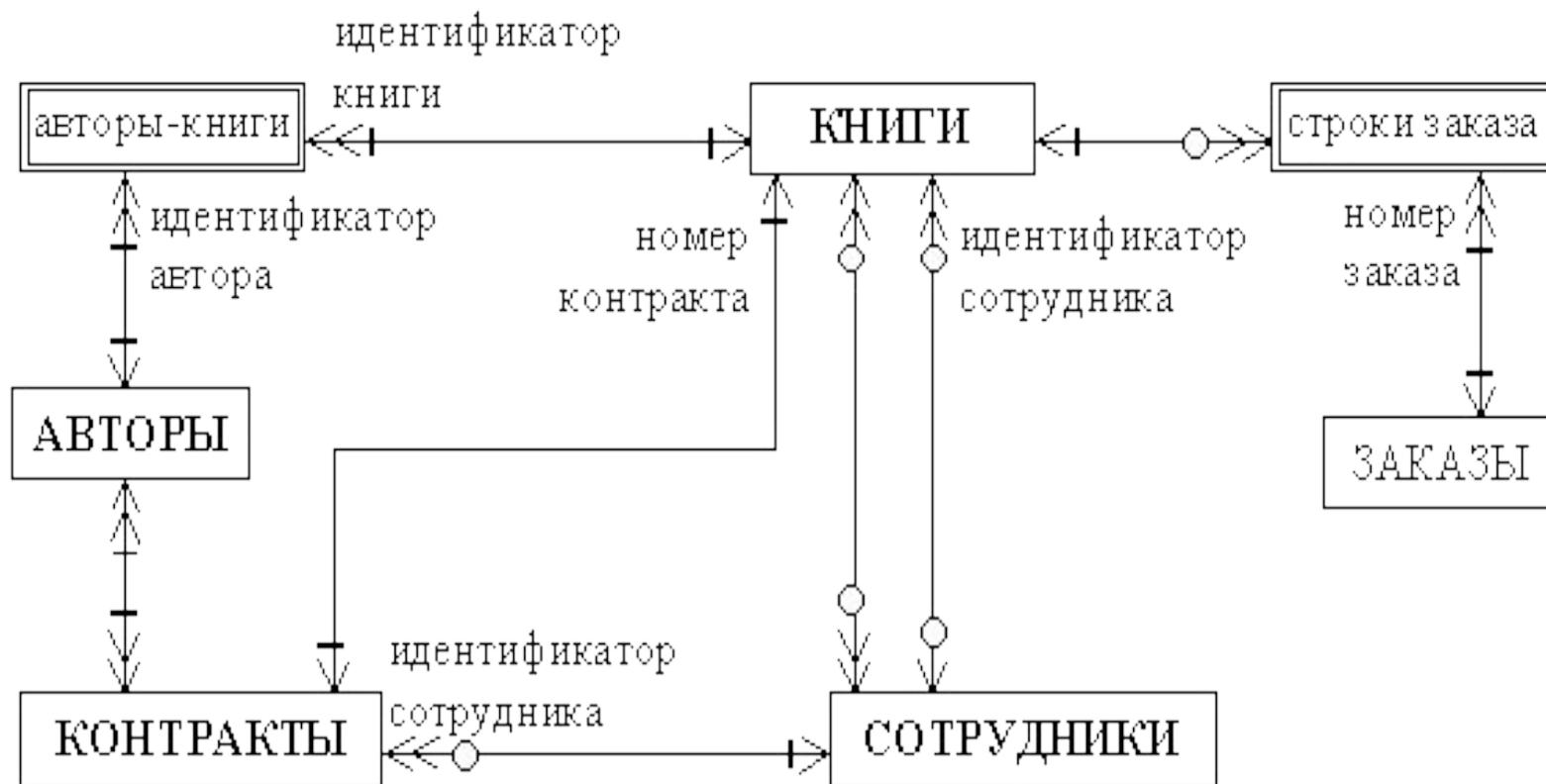
## *Пример компании, которая занимается издательской деятельностью :*

База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников компании. БД должна содержать данные о сотрудниках компании, книгах, авторах, финансовом состоянии компании и предоставлять возможность получать разнообразные отчёты.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- каждая книга издаётся в рамках контракта;
- книга может быть написана несколькими авторами;
- контракт подписывается одним менеджером и всеми авторами книги;
- каждый автор может написать несколько книг (по разным контрактам);
- порядок, в котором авторы указаны на обложке, влияет на размер гонорара;
- если сотрудник является редактором, то он может работать одновременно над несколькими книгами;
- у каждой книги может быть несколько редакторов, один из них – ответственный редактор;
- каждый заказ оформляется на одного заказчика;
- в заказе на покупку может быть перечислено несколько книг.

# Метод "Сущность – Связь" (Entity–Relation, ER–method)



# Метод "Сущность – Связь" (Entity–Relation, ER–method)

---

**Пример проектной организации. Основной вид деятельности такой организации – выполнение проектов по договорам с заказчиками:**

База данных создаётся для информационного обслуживания руководства организации, руководителей проектов и участников проектов. БД должна содержать данные об отделах организации, сотрудниках и проектах. В соответствии с предметной областью система строится с учётом следующих особенностей:

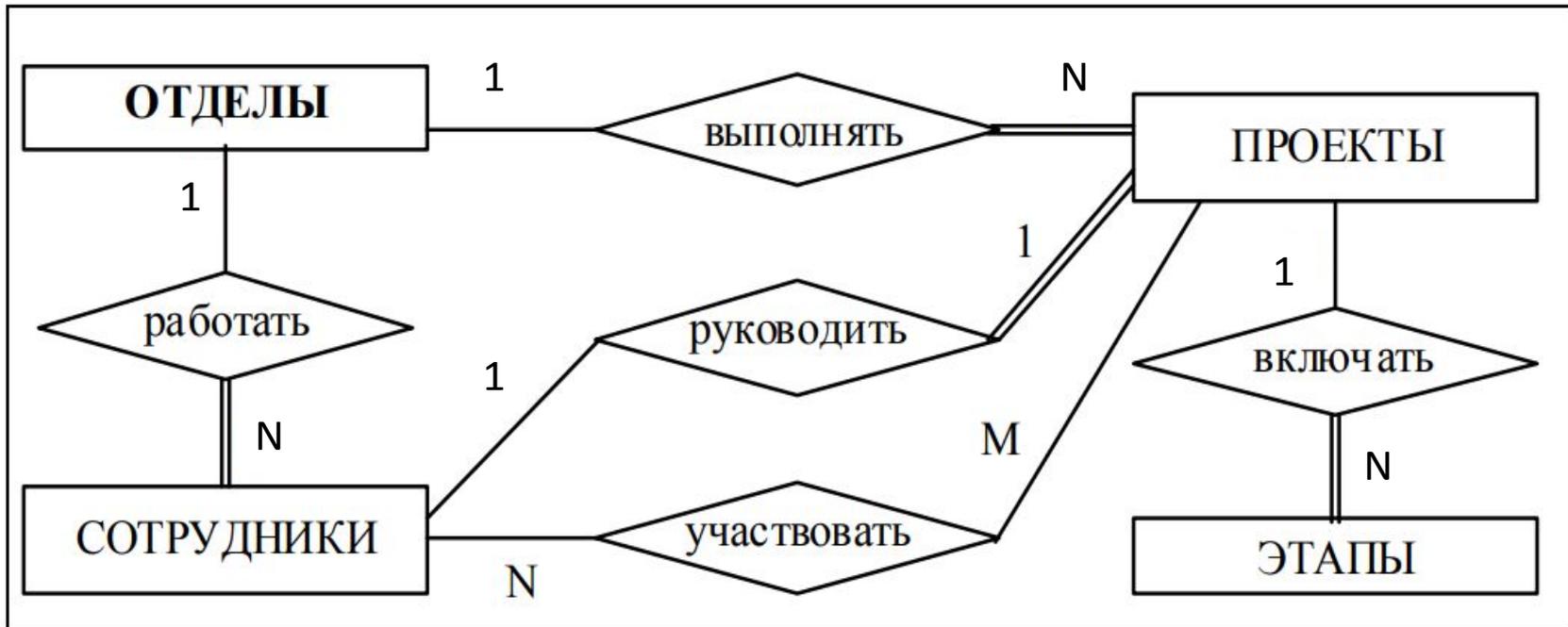
- Каждый сотрудник работает в определённом отделе, в каждом отделе могут работать несколько сотрудников.
- Каждый проект относится к определённому отделу, каждый отдел может отвечать за выполнение нескольких проектов.
- Каждый сотрудник может принимать участие в выполнении нескольких проектов, над каждым проектом может трудиться несколько сотрудников.
- Для каждого проекта назначается руководитель из числа сотрудников того отдела, к которому относится проект.
- Каждый проект должен быть выполнен в заданные сроки, каждый проект может состоять из нескольких этапов. Если проект состоит из одного этапа, то сроки его выполнения должны совпадать со сроками выполнения проекта в целом.

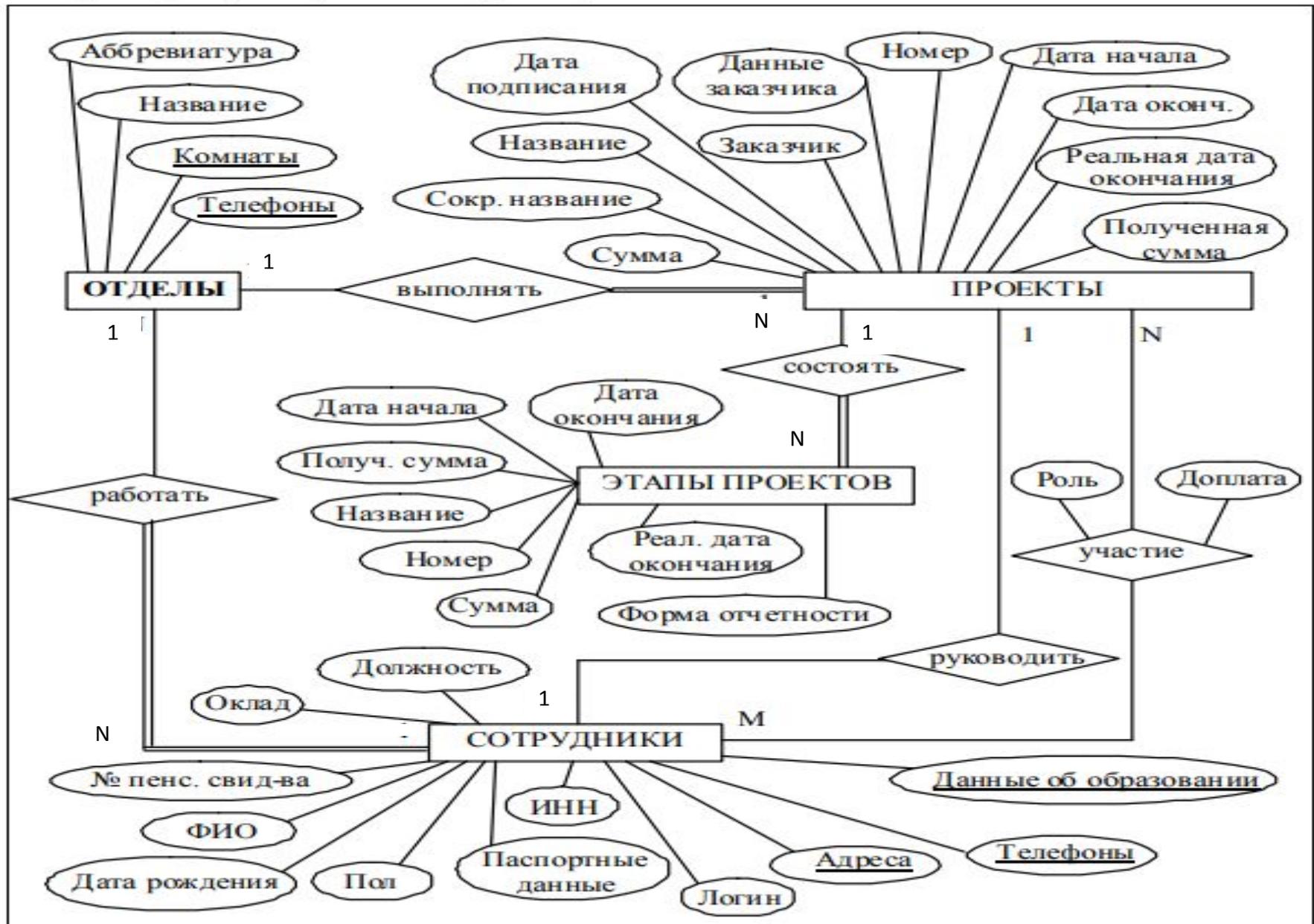
# Метод "Сущность – Связь" (Entity–Relation, ER–method)

---

- Оклад сотрудника зависит от занимаемой должности, за участие в проектах сотрудник получает дополнительное вознаграждение.
- Виды участия сотрудников в проектах: руководитель, консультант, исполнитель.
- Каждый отдел занимает одно или несколько помещений (комнат), в каждом помещении может быть один или несколько стационарных телефонов.

# Метод "Сущность – Связь" (Entity–Relation, ER–method)





# Технологии баз данных

## Тема 5. Реляционная модель

# Реляционная модель данных

---

Теоретической основой этой модели стала теория отношений, основу которой заложили два логика — американец **Чарльз Содерс Пирс** (1839-1914) и немец **Эрнст Шредер** (1841-1902).

***n*-арным отношением**  $R$  называют подмножество декартова произведения множеств  $D_1, D_2, \dots, D_n$  ( $n > 1$ ), несформированное в виде  $R \subseteq D_1 \times D_2 \times \dots \times D_n$ .

Исходные множества  $D_1, D_2, \dots, D_n$  называют в реляционной модели **доменами**.

**Домен** – скалярный тип данных + (опционально) некоторое дополнительное ограничение диапазона значений указанного типа

# Реляционная модель данных

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

$D_1 = \{\text{Иванов, Крылов, Степанов}\};$

$D_2 = \{\text{Теория автоматов, Базы данных}\};$

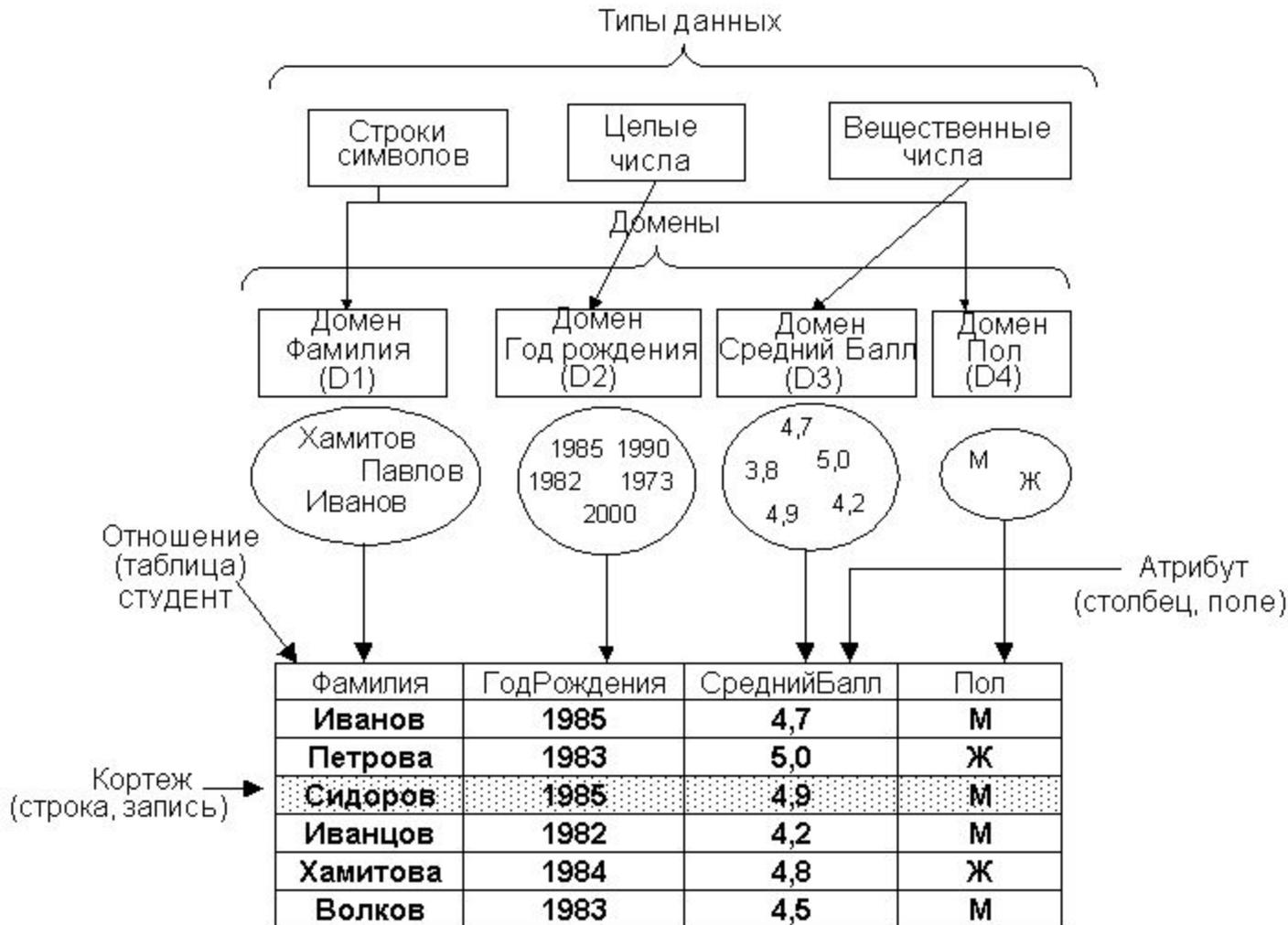
$D_3 = \{3, 4, 5\}$

$\langle \text{Иванов, Теория автоматов, 4} \rangle;$   
 $\langle \text{Крылов, Теория автоматов, 5} \rangle;$   
 $\langle \text{Степанов, Теория автоматов, 5} \rangle;$   
 $\langle \text{Иванов, Базы данных, 3} \rangle;$   
 $\langle \text{Степанов, Базы данных, 4} \rangle;$

$\langle \text{Иванов, Теория автоматов, 3} \rangle;$   
 $\langle \text{Иванов, Теория автоматов, 4} \rangle;$   
 $\langle \text{Иванов, Теория автоматов, 5} \rangle;$   
 $\langle \text{Крылов, Теория автоматов, 3} \rangle;$   
 $\langle \text{Крылов, Теория автоматов, 4} \rangle;$   
 $\langle \text{Крылов, Теория автоматов, 5} \rangle;$   
 $\langle \text{Степанов, Теория автоматов, 3} \rangle;$   
 $\langle \text{Степанов, Теория автоматов, 4} \rangle;$   
 $\langle \text{Степанов, Теория автоматов, 5} \rangle;$   
 $\langle \text{Иванов, Базы данных, 3} \rangle;$   
 $\langle \text{Иванов, Базы данных, 4} \rangle;$   
 $\langle \text{Иванов, Базы данных, 5} \rangle;$   
 $\langle \text{Крылов, Базы данных, 3} \rangle;$   
 $\langle \text{Крылов, Базы данных, 4} \rangle;$   
 $\langle \text{Крылов, Базы данных, 5} \rangle;$   
 $\langle \text{Степанов, Базы данных, 3} \rangle;$   
 $\langle \text{Степанов, Базы данных, 4} \rangle;$   
 $\langle \text{Степанов, Базы данных, 5} \rangle;$

Фамилия	Дисциплина	Оценка
Иванов	Теория автоматов	4
Иванов	Базы данных	3
Крылов	Теория автоматов	5
Степанов	Теория автоматов	5

# Реляционная модель данных



Количество атрибутов в отношении называется **степенью**, или **рангом**, отношения.

**Мощность** – число кортежей в таблице (проще говоря, число записей)

# Реляционная модель данных

Фамилия	Дисциплина	Оценка
Иванов	Теория автоматов	4
Иванов	Базы данных	3
Крылов	Теория автоматов	5
Степанов	Теория автоматов	5
Степанов	Базы данных	4

Каждое отношение представленное в виде таблицы и обладает рядом специфических свойств:

- Отсутствие кортежей дубликатов.
- Отсутствие упорядоченности кортежей.
- Отсутствие упорядоченности атрибутов.
- Атомарность значений атрибутов.

**Ключом отношения**  $R$  называется минимальное подмножество  $K = \{A_{i_1}, A_{i_2}, \dots, A_{i_m}\} \subseteq \{A_1, A_2, \dots, A_n\}$ , где  $\{i_1, i_2, \dots, i_m\} \subseteq \{1, 2, \dots, n\}$ , такое, что любые два различных кортежа  $t_1, t_2 \in R$  ( $t_1 \neq t_2$ ) не совпадают по значениям множества  $K = \{A_{i_1}, A_{i_2}, \dots, A_{i_m}\}$ .

# Реляционная модель данных

---

**Схемой отношения  $R$**  называется перечень имен атрибутов данного отношения с указанием домена, к которому они принадлежат:  $S_R = (A_1, A_2, \dots, A_n)$ , где  $A_i \in D_i$

Если атрибуты принимают значения из одного и того же домена, то они называются -  $\Theta$  **сравнимыми**, где  $\Theta$  — множество допустимых операций сравнения, заданных для данного домена.

$$S_{R1} \sim S_{R2} \Leftrightarrow \begin{cases} 1. & n=m \\ 2. & A_j, B_{ij} \in D_j \end{cases}$$

Совокупность схем отношений, используемых для представления концептуальной модели, называется **схемой реляционной базы данных**, а текущие значения соответствующих отношений – **реляционной базой данных**.

# Правила преобразования ER-модели в реляционную

---

- 1. Каждой сущности ставится в соответствие отношение реляционной модели данных.*
- 2. Каждый атрибут сущности становится атрибутом соответствующего отношения. Для каждого атрибута задается конкретный допустимый в СУБД тип данных и обязательность или необязательность данного атрибута (то есть допустимость или недопустимость **NULL** значений для него).*
- 3. Первичный ключ сущности становится **PRIMARY KEY** соответствующего отношения. Атрибуты, входящие в первичный ключ отношения, автоматически получают свойство обязательности (непустоты).*

# Правила преобразования ER-модели в

## реляционную

### *Преобразование связей 1:1*

- 1. Следует объединить данные сущностей в одно отношение.**

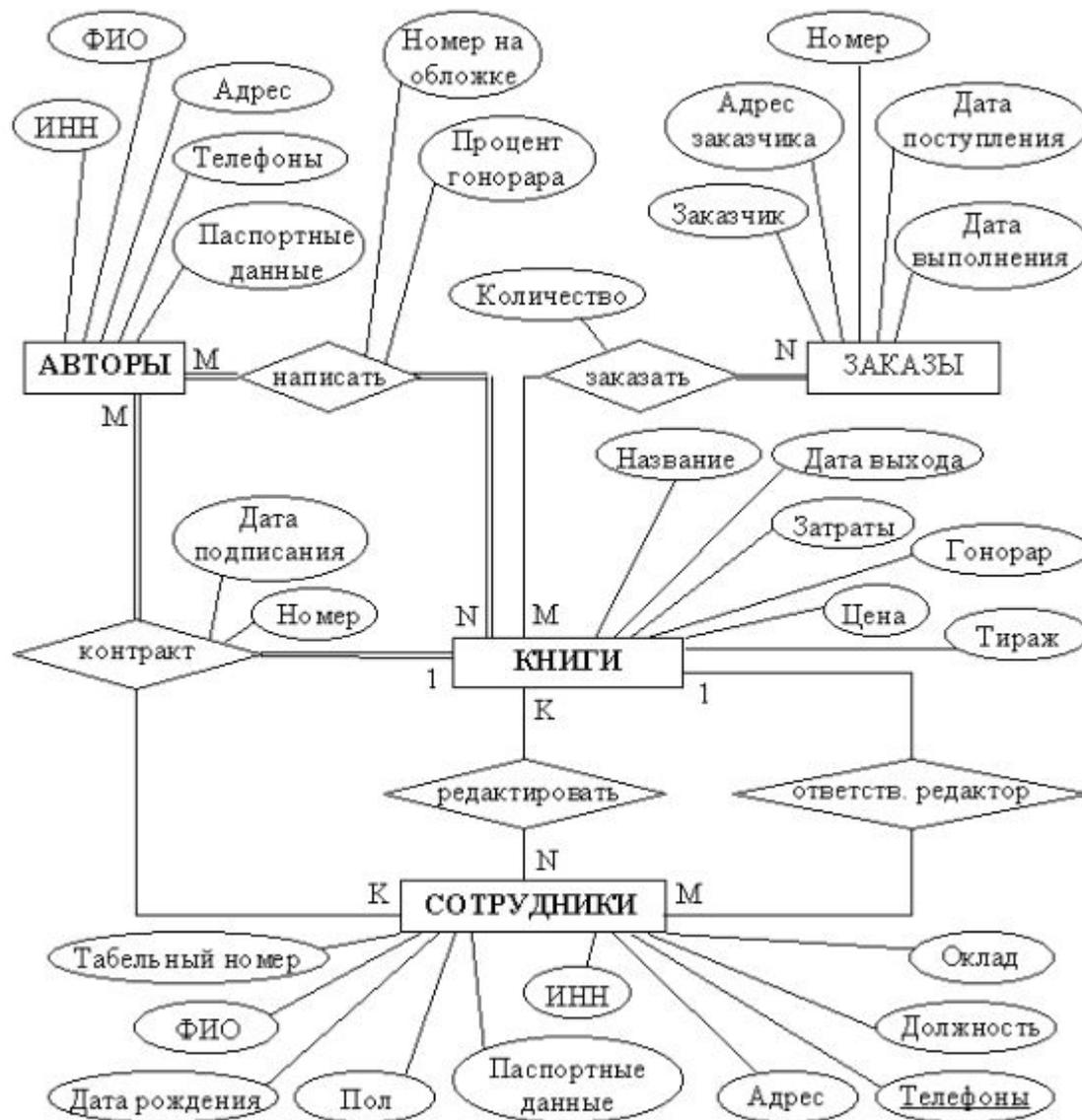
### *Преобразование связей 1:M*

- 1. В каждое отношение, соответствующее подчиненной сущности, добавляется набор атрибутов основной сущности, являющейся первичным ключом основной сущности. В отношении, соответствующем подчиненной сущности, этот набор атрибутов становится внешним ключом (*FOREING KEY*).**
- 2. Для моделирования необязательного типа связи на физическом уровне у атрибутов, соответствующих внешнему ключу, устанавливается свойство допустимости неопределенных значений (признак *NULL*). При обязательном типе связи атрибуты получают свойство отсутствия неопределенных значений (признак *NOT NULL*)**

### *Преобразование связей N:M и не бинарных связей*

- 1. Для каждой связи добавляется новое отношение, атрибутами которого становятся первичные ключи сущностей, участвующих в этой связи. Все вместе они образуют первичный ключ нового отношения, а каждый по отдельности внешний ключ для связи.**

# Правила преобразования ER-модели в ре.



# **Технологии баз данных**

**Тема 6. Процесс нормализации  
отношений**

# Процесс нормализации отношений.

---

При представлении **концептуальной схемы** в виде **реляционной модели** возможны **различные варианты выбора схем отношений**. От **правильного выбора схем отношений**, представляющих концептуальную схему, в значительной степени будет зависеть **эффективность функционирования БД**.

СТУДЕНТ (Код студента, Фамилия, Название факультета, Название специальности).

**Эта схема отношений обуславливает следующие недостатки соответствующей БД:**

1. **Дублирование информации (избыточность).**  
У студентов, обучающихся на одном факультете, будет повторяться название факультета.
2. **Потенциальная противоречивость (аномалии обновления).**  
Изменится название специальности, то изменяя её в одном кортеже (у одного студента), необходимо изменять и во всех других кортежах, где она присутствует.
3. **Потенциальная возможность потери сведений (аномалии удаления).**  
При удалении информации обо всех студентах, поступающих на определенную специальность, мы теряем все сведения об этой специальности.
4. **Потенциальная возможность невключения информации в БД**

# Процесс нормализации отношений.

---

*В теории реляционных БД существует **формальный метод** построения реляционной модели БД, в которой отсутствует избыточность и аномалии обновления, удаления и включения. Это метод называется **НОРМАЛИЗАЦИЯ**.*

- *первая нормальная форма (1NF);*
- *вторая нормальная форма (2NF);*
- *третья нормальная форма (3NF);*
- *нормальная форма Бойса—Кодда (BCNF);*
- *четвертая нормальная форма (4NF);*
- *пятая нормальная форма, или форма проекции-соединения (5NF или PJNF).*

## Зачем нормализовывать базу данных?

- Точка зрения архитектора:
  - Устранение аномалий
  - Нормализуем базу данных до предела
  
- Точка зрения программиста:
  - Повышение производительности
  - Удобство работы с базой при помощи SQL-запросов
  - Нормализуем базу до тех пор, пока это приносит заметные улучшения

## Суть первых трёх нормальных форм

1. Все неключевые столбцы определяются ключом
2. Полным ключом!
3. И ничем, кроме ключа!!!

# Процесс нормализации отношений.

---

## Основные свойства нормальных форм:

*каждая следующая нормальная форма в некотором смысле улучшает свойства предыдущей;*

*при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.*

*Нормализация схемы отношения выполняется путём декомпозиции схемы.*

**Декомпозицией** **схемы** **отношения** **R** называется замена её совокупностью схем отношений  $A_i$  таких, что ,  
$$R = \text{Join } A_i$$

*и не требуется, чтобы отношения  $A_i$  были непересекающимися.*

# Процесс нормализации отношений.

---

## **Функциональная зависимость.**

В отношении  $R$  набор атрибутов  $Y$  функционально зависит от набора атрибутов  $X$  или набор атрибутов  $X$  функционально определяет набор атрибутов  $Y$  ( обозначается  $R.X \square R.Y$  ) тогда и только тогда, когда каждому значению  $X$  соответствует в точности одно значение  $Y$ .

## **Полная функциональная зависимость.**

Функциональная зависимость называется полной, если набор атрибутов  $Y$  функционально зависит от  $X$  и не зависит функционально от любого точного подмножества  $X$ .

## **Транзитивная зависимость.**

Функциональная зависимость  $R.X \square R.Y$  называется транзитивной, если существует набор атрибутов  $Z$  такой, что:  $Z$  не является подмножеством  $X$ ;  $Z$  не включает в себя  $Y$ ;

существует функциональная зависимость  $R.X \square R.Z$ ; существует функциональная зависимость  $R.Z \square R.Y$  и не существует функциональной зависимости  $R.Z \square R.X$ .

Замечание: Транзитивная зависимость появляется, когда неключевой атрибут имеет связь с другим неключевым атрибутом.

# Процесс нормализации отношений.

## Первая нормальная форма (1NF).

Отношение находится в первой нормальной форме(1NF), если все атрибуты отношения принимают простые значения (атомарные или неделимые), не являющиеся множеством или кортежем из элементарных составляющих.

<u>Код студента</u>	Фамилия	<u>Код экзамена</u>	Предмет и дата	Оценка
1	Сергеев	1	Математика 5.06.10	4
2	Иванов	1	Математика 5.06.10	5
1	Сергеев	2	Физика 9.06.10	5
2	Иванов	2	Физика 9.06.10	5

<u>Код студента</u>	Фамилия	<u>Код экзамена</u>	Предмет	Дата	Оценка
1	Сергеев	1	Математика	5.06.10	4
2	Иванов	1	Математика	5.06.10	5
1	Сергеев	2	Физика	9.06.10	5
2	Иванов	2	Физика	9.06.10	5

# Процесс нормализации отношений.

---

## **Вторая нормальная форма (2NF).**

*Отношение находится во второй нормальной форме (2NF), если оно находится в 1NF, и каждый неключевой атрибут функционально полно зависит от всего первичного ключа (не зависит от части ключа).*

<u>Код студента</u>	Фамилия	<u>Код экзамена</u>	Предмет	Дата	Оценка
1	Сергеев	1	Математика	5.06.10	4
2	Иванов	1	Математика	5.06.10	5
1	Сергеев	2	Физика	9.06.10	5
2	Иванов	2	Физика	9.06.10	5

### **Полная функциональная зависимость**

Код студента, Код экзамена  $\square$  Оценка

### **Неполная функциональная зависимость**

Код студента  $\square$  Фамилия

Код экзамена  $\square$  Предмет

Код экзамена  $\square$  Дата

# Процесс нормализации отношений.

---

## **Вторая нормальная форма (2NF).**

*Отношение находится во второй нормальной форме (2NF), если оно находится в 1NF, и каждый неключевой атрибут функционально полно зависит от всего первичного ключа (не зависит от части ключа).*

<u>Код студента</u>	<u>Код экзамена</u>	Оценка
1	1	4
2	1	5
1	2	5
2	2	5

<u>Код студента</u>	Фамилия
1	Сергеев
2	Иванов

<u>Код экзамена</u>	Предмет	Дата
1	Математика	5.06.10
2	Физика	9.06.10

# Процесс нормализации отношений.

---

## *Третья нормальная форма (3NF).*

*Отношение находится в 3NF, если оно находится в 2NF, и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.*

R (№ студ. билета, ФИО, Группа, Специальность, Факультет)

№ студ. билета  ФИО

№ студ. билета.  Группа

№ студ. билета  Факультет

№ студ. билета

Специальность

Группа  Факультет

Группа  Специальность

(№ студ. билета, ФИО, Группа)

(Группа, Специальность, Факультет)

# Процесс нормализации отношений.

## Нормальная форма Бойса-Кодда (BCNF) - усиленная 3NF

**Пример 1.** Пусть требуется хранить данные о поставках деталей некоторыми поставщиками. Предположим, что наименования поставщиков являются уникальными. Кроме того, каждый поставщик имеет свой уникальный номер. Данные о поставках можно хранить в следующем отношении:

Номер поставщика PNUM	Наименование поставщика PNAME	Номер детали DNUM	Поставляемое количество VOLUME
1	Фирма 1	1	100
1	Фирма 1	2	200
1	Фирма 1	3	300
2	Фирма 2	1	150
2	Фирма 2	2	250
3	Фирма 3	1	1000

Таблица 1 Отношение "Поставки"

Два потенциальных ключа -  $\{PNUM, DNUM\}$  и  $\{PNAME, DNUM\}$

$PNUM \square PNAME$

$\{PNUM, DNUM\} \square PNAME$

$PNAME \square PNUM$

$\{PNUM, DNUM\} \square VOLUME$

$\{PNAME, DNUM\} \square VOLUME$

$\{PNAME, DNUM\} \square PNUM$

# Процесс нормализации отношений.

## Нормальная форма Бойса-Кодда (BCNF) - усиленная 3NF

**Детерминант** – это любой атрибут, от которого функционально полно зависит некоторый другой атрибут.

Отношение находится в **нормальной форме Бойса-Кодда (BCNF)** тогда и только тогда, когда **детерминанты всех функциональных зависимостей являются потенциальными ключами**.

**Пример 1.** Пусть требуется хранить данные о поставках деталей некоторыми поставщиками. Предположим, что наименования поставщиков являются уникальными. Кроме того, каждый поставщик имеет свой уникальный номер. Данные о поставках можно хранить в следующем отношении:

Номер поставщика PNUM	Наименование поставщика PNAME	Номер детали DNUM	Поставляемое количество VOLUME
1	Фирма 1	1	100
1	Фирма 1	2	200
1	Фирма 1	3	300
2	Фирма 2	1	150
2	Фирма 2	2	250
3	Фирма 3	1	1000

Таблица 1 Отношение "Поставки"

# Процесс нормализации отношений.

---

## Нормальная форма Бойса-Кодда (BCNF) - усиленная 3NF

Номер поставщика PNUM	Наименование поставщика PNAME
1	Фирма 1
2	Фирма 2
3	Фирма 3

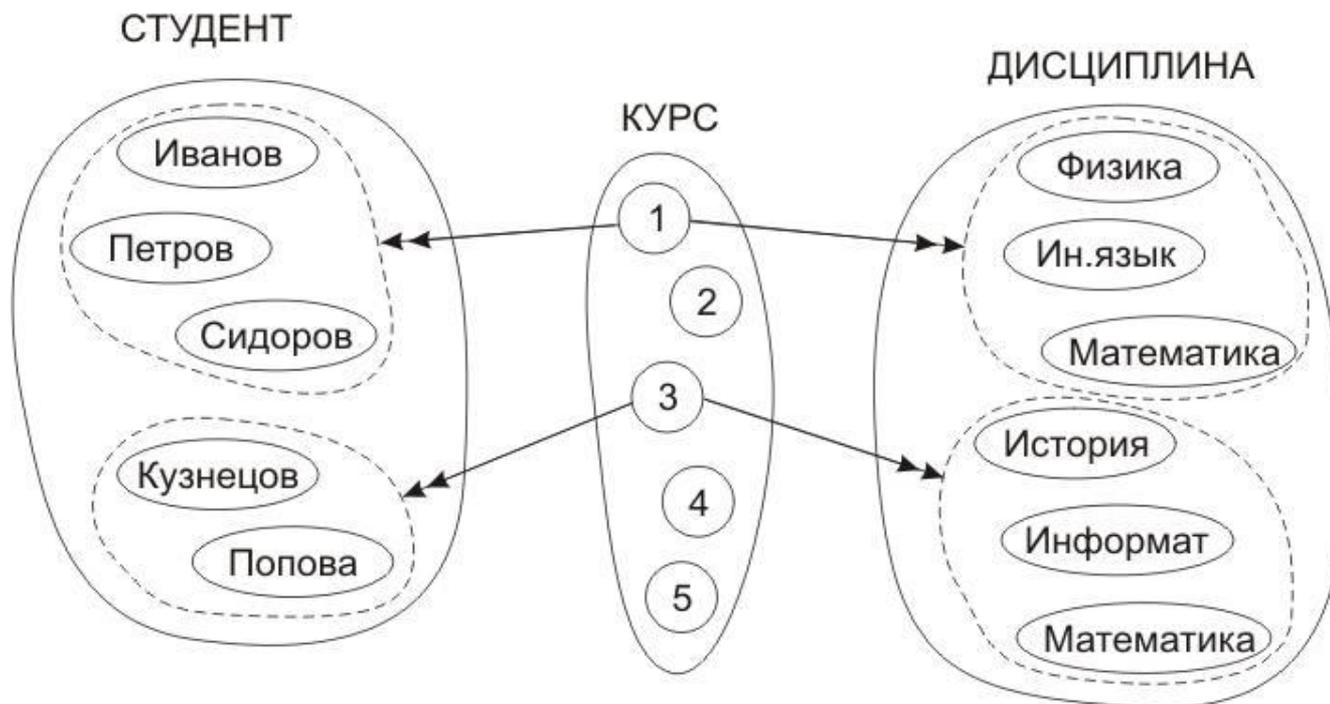
Таблица 2 Отношение "Поставщики"

Номер поставщика PNUM	Номер детали DNUM	Поставляемое количество VOLUME
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	1	1000

Таблица 3 Отношение "Поставки-2"

# Процесс нормализации отношений.

## Четвертая нормальная форма (4NF).



### **Многозначная зависимость.**

*В отношении  $R(A, B, C)$  существует многозначная зависимость  $R.A \twoheadrightarrow R.B$  в том*

*и только том случае, если множество значений  $B$ , соответствующее паре значений  $A$  и  $C$ , зависит только от  $A$  и не зависит от  $C$*

# Процесс нормализации отношений.

---

## *Четвертая нормальная форма (4NF).*

### *Учебные дисциплины*

Дисциплина	Книга	Лектор
МатАн	Кудрявцев	Иванов А.
МатАн	Фихтенгольц	Петров Б.
МатАн	Кудрявцев	Петров Б.
МатАн	Фихтенгольц	Иванов А.
МатАн	Кудрявцев	Смирнов В.
МатАн	Фихтенгольц	Смирнов В.
ВМ	Кудрявцев	Иванов А.
ВМ	Кудрявцев	Петров Б.

# Процесс нормализации отношений.

## Четвертая нормальная форма (4NF).

*Под проецированием без потерь понимается такой способ декомпозиции отношения, при котором исходное отношение полностью и без избыточности восстанавливается путем естественного соединения своих проекций*

*R*

<i>a</i>	<i>b</i>	<i>c</i>
Москва	Россия	столица
Томск	Россия	не столица
Берлин	Германия	столица

Декомпозиция  $R_1 = \{a\}$ ,  $R_2 = \{b, c\}$  имеет вид:

*R*<sub>1</sub>

<i>a</i>	<i>b</i>	<i>c</i>
Москва	Россия	столица
Томск	Россия	не столица
Берлин	Германия	столица

*R*<sub>2</sub>

Результат операции соединения этих отношений:

$R' = R_1 \text{ NATURAL JOIN } R_2$

<i>a</i>	<i>b</i>	<i>c</i>
Москва	Россия	столица
Москва	Россия	не столица
Москва	Германия	столица
Томск	Россия	столица
Томск	Россия	не столица
Томск	Германия	столица
Берлин	Россия	столица
Берлин	Россия	не столица
Берлин	Германия	столица

# Процесс нормализации отношений.

## Четвертая нормальная форма (4NF).

*Под проецированием без потерь понимается такой способ декомпозиции отношения, при котором исходное отношение полностью и без избыточности восстанавливается путем естественного соединения своих проекций*

Рассмотрим теперь декомпозицию  $R_1 = \{a, b\}$ ,  $R_2 = \{a, c\}$ :

$a$	$b$	$c$
Москва	Россия	столица
Томск	Россия	не столица
Берлин	Германия	столица

$a$	$b$	$a$	$c$
Москва	Россия	Москва	столица
Томск	Россия	Томск	не столица
Берлин	Германия	Берлин	столица

# Процесс нормализации отношений.

---

## Четвертая нормальная форма (4NF).

### Теорема Фейджина

Отношение  $R(A, B, C)$  можно спроецировать без потерь

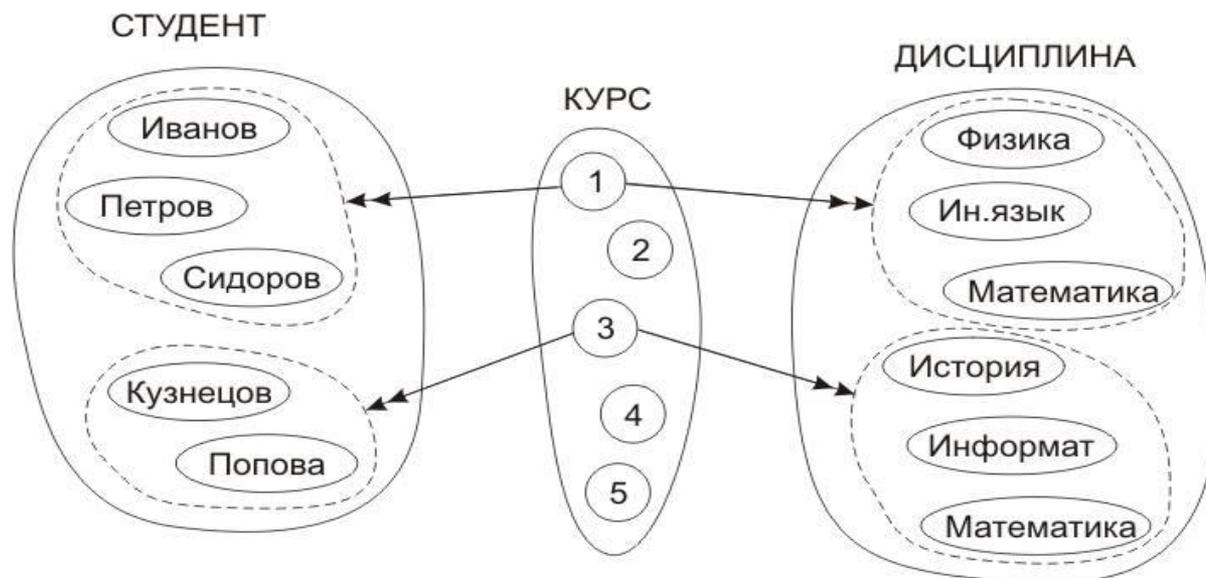
в отношения  $R1(A, B)$  и  $R2(A, C)$

тогда и только тогда когда существует  $R.A \twoheadrightarrow R.B$   
( $R.A \twoheadrightarrow R.C$ )

Отношение  $R$  находится в 4NF в том и только том случае, если в случае существования многозначной зависимости  $R.A \twoheadrightarrow R.B$  все остальные атрибуты  $R$  функционально зависят от  $A$ .

# Процесс нормализации отношений.

## Четвертая нормальная форма (4NF).



КУРС  $\twoheadrightarrow$  СТУДЕНТ | ДИСЦИПЛИНА

СТУД\_КУРС

КУРС\_ДИСЦ

СТУДЕНТ	КУРС
Иванов	1
Петров	1
Сидоров	1
Кузнецов	3
Попова	3

КУРС	ДИСЦИПЛИНА
1	Математика
1	Физика
1	Иностр.язык
3	Информатика
3	История
3	Математика

# Процесс нормализации отношений.

## Пятая нормальная форма (5NF).

Ассортимент (продавцы, фирмы,  
товары)

Продавец	Фирма	Товар
Иванов	Рога и Копыта	Пылесос
Иванов	Рога и Копыта	Хлебница
Петров	Безенчук&Ко	Сучкорез
Петров	Безенчук&Ко	Пылесос
Петров	Безенчук&Ко	Хлебница
Петров	Безенчук&Ко	Зонт
Сидоров	Безенчук&Ко	Пылесос
Сидоров	Безенчук&Ко	Телескоп
Сидоров	Рога и Копыта	Пылесос
Сидоров	Рога и Копыта	Лампа
Сидоров	Геркулес	Вешалка

Предположим, что нужно учесть следующее ограничение:  
**каждый продавец имеет в своём ассортименте ограниченный список фирм и ограниченный список типов товаров и предлагает товары из списка товаров, производимые фирмами из списка фирм.**

# Процесс нормализации отношений.

---

## *Пятая нормальная форма (5NF).*

### **Зависимость соединения**

Пусть  $R$  — отношение, а  $A, B, \dots, Z$  — некоторые подмножества множества его атрибутов.

Если декомпозиция отношения  $R$  на отношения, состоящие из множеств атрибутов  $A, B, \dots, Z$ , является декомпозицией без потерь, говорят, что переменная отношения  $R$  удовлетворяет зависимости соединения  $\{A, B, \dots, Z\}$

*Отношение находится в 5NF (нормальной форме проекции-соединения) тогда и только тогда, когда каждая зависимость соединения в нём определяется потенциальным ключом (ключами) этого отношения*

# Процесс нормализации отношений.

## Пятая нормальная форма (5NF).

Товары продавцов		Фирмы продавцов		Товары фирм	
<u>Продавец</u>	<u>Товар</u>	<u>Продавец</u>	<u>Фирма</u>	<u>Фирма</u>	<u>Товар</u>
Иванов	Пылесос	Иванов	Рога и Копыта	Рога и Копыта	Пылесос
Иванов	Хлебница	Петров	Безенчук&Ко	Рога и Копыта	Хлебница
Петров	Сучкорез	Сидоров	Безенчук&Ко	Рога и Копыта	Лампа
Петров	Пылесос	Сидоров	Рога и Копыта	Безенчук&Ко	Сучкорез
Петров	Хлебница	Сидоров	Геркулес	Безенчук&Ко	Пылесос
Петров	Зонт			Безенчук&Ко	Хлебница
Сидоров	Телескоп			Безенчук&Ко	Зонт
Сидоров	Пылесос			Безенчук&Ко	Телескоп
Сидоров	Лампа			Геркулес	Вешалка
Сидоров	Вешалка				

Ассортимент (продавцы, фирмы, товары)

<u>Продавец</u>	<u>Фирма</u>	<u>Товар</u>
Иванов	Рога и Копыта	Пылесос
Иванов	Рога и Копыта	Хлебница
Петров	Безенчук&Ко	Сучкорез
Петров	Безенчук&Ко	Пылесос
Петров	Безенчук&Ко	Хлебница
Петров	Безенчук&Ко	Зонт
Сидоров	Безенчук&Ко	Пылесос
Сидоров	Безенчук&Ко	Телескоп
Сидоров	Рога и Копыта	Пылесос
Сидоров	Рога и Копыта	Лампа
Сидоров	Геркулес	Вешалка

## В какой нормальной форме находится эта таблица?

- Название звезды – ключевой столбец
  - все звёзды уникальны
- Название созвездия – неключевой столбец
  - звезда принадлежит одному из созвездий

 Звезда	Созвездие
Сириус	Большой Пёс
Полярная звезда	Малая Медведица
Антарес	Скорпион
Кохаб	Малая Медведица
Шаула	Скорпион
Саргас	Скорпион

**Что не так с этой таблицей?**

## С этими таблицами всё в порядке?

 Звезда	Созвездие
Сириус	Большой Пёс
Полярная звезда	Малая Медведица
Антарес	Скорпион
Кохаб	Малая Медведица
Шаула	Скорпион
Саргас	Скорпион

 Город	Страна
Ленинград	СССР
Сталинград	СССР
Пномпень	Камбоджа
Рангун	Бирма
Бомбей	Индия
Калькутта	Индия

## А так лучше?

 № страны	Страна
1	СССР
2	Камбоджа
3	Бирма
4	Индия

 Город	№ страны
Ленинград	1
Сталинград	1
Пномпень	2
Рангун	3
Бомбей	4
Калькутта	4