

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ

## Раздел: Технологии Баз Данных

д.т.н. Мокрова Наталия Владиславовна

Кафедра информационных технологий

## ОСНОВНАЯ ЛИТЕРАТУРА

1. Информационные технологии управления: Учебник / Б.В. Черников. – 2-е изд., перераб. и доп. – М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. – 368 с.
2. Информационные системы в экономике: учеб. пособие для вузов / Е. В. Варфоломеева [и др.] ; под ред. Д.В. Чистова. – гриф УМО. – М. : ИНФРА-М, 2013. – 234 с.
3. Учебно-практические пособия кафедры информационных технологий.

## ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Карпузова В. И. Информационные технологии в менеджменте: Учебное пособие / В.И. Карпузова, Э.Н. Скрипченко, К.В. Чернышева, Н.В. Карпузова. – 2-е изд., доп. – М.: Вузовский учебник: НИЦ ИНФРА-М, 2014. – 301 с. +
2. Саак А.Э., Пахомов Е.В., Тюшняков В.Н. Информационные технологии управления: учебник для вузов. 2-е изд. – СПб.: Питер, 2008.
3. Данчул А.Н. Информационные технологии управления. Компьютерный практикум. М: Изд-во РАГС, 2006 – 206 с.
4. Титоренко Г.А. Информационные технологии управления: учеб. Пособие для вузов. 2-е изд., доп. М.: 2008.

### **Рекомендуемая литература:**

Microsoft Access. MS Visio Шаг за шагом. MS Project 2003. Шаг за шагом



## **Этапы и составляющие обработки данных:**

- архив, письменность;
- язык описания данных, средствами программируемой обработки записей (ПО 1955),
- операции над базами данных (оперативные сетевые БД 1960);
- концепция схем баз данных,
- технологии реляционной модели данных в тесной связи с математикой и логикой.

**Единая формулировка** БД отсутствует, используются отличительные признаки:

*БД хранится и обрабатывается в вычислительной системе (внекомпьютерные хранилища БД не являются).*

*Данные в БД логически структурированы (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки, с выделением составных частей (элементов), связей между ними, а также типизацию элементов и связей (с типом элемента или связи соотносится определённая семантика и допустимые операции).*

*БД включает метаданные, описывающие логическую структуру БД в формальном виде (в соответствии с некоторой метамоделью).*

*Файловые архивы, Интернет-порталы, электронные таблицы, обладают признаками БД.*

# ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

**База данных** – представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и т.п.), систематизированных так, чтобы эти материалы могли быть найдены и обработаны с помощью ЭВМ (Гражданский кодекс РФ, ст. 1260).

**База данных** – организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей (*Когаловский М.Р. Энциклопедия технологий баз данных – М.: Финансы и статистика, 2002. – 800 с.*)

**База данных** – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных (ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными (аналог ISO/IEC TR 10032:2003))

**База данных** – некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия (Дейт К. Дж. Введение в системы баз данных. – 8-е издание.: Пер. с англ. – М.: Изд. дом «Вильямс», 2005. - 1328 с.)

**База данных** – совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации (Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. 1436 с.)

# Организация БД

Развитие ИТ – концепция интегральных данных (централизованное хранение структур данных сложной организации, использование многими потребителями, имеющими различные требования к информации). Создание систем управления БД (СУБД).

Концепция БД – данные должны адекватно отражать предметную область (часть реальной среды, описываемая БД) и обеспечивать информационные нужды многих пользователей.

## Свойства БД

1. Многократное использование информации БД.
2. Простота обновления с минимальными затратами.
3. Быстрый поиск и получение необходимой информации по запросу.
4. Уменьшение избыточности – получение данных из существующей БД, а не путем повторного ввода.
5. Защита от несанкционированного доступа.
6. Максимальная независимость от прикладных программ – изменение в структуре БД не должно приводить к изменениям в прикладных программах.
7. Защита от уничтожения и искажения информации.

## Основные требования к БД

1. Адекватность отражения предметной области:
  - а. полнота данных;
  - б. динамичность информационной модели;
  - в. актуальность информации в данный момент времени.
2. Возможность взаимодействия с пользователями различных категорий.
3. Обеспечение секретности и целостности БД.
4. Совместимость компонентов БД.
5. Простота изменений логической и физической структуры БД.
6. Способность к расширению и модификации.



# ВИДЫ БАЗ ДАННЫХ

Существует много разновидностей БД, отличающихся по различным критериям, можно выделить свыше 50 видов БД.

## **Классификация по модели данных:**

Иерархическая                      Сетевая  
Реляционная                      Объектная и объектно-ориентированная  
Объектно-реляционная      Функциональная.

## **Классификация по среде постоянного хранения:**

Во вторичной памяти (традиционная)  
В оперативной памяти              В третичной памяти.

## **Классификация по содержанию:**

Географическая              Историческая  
Научная                      Мультимедийная.

## **Классификация по степени распределённости:**

Централизованная              Распределённая  
Неоднородная                      Однородная  
Фрагментированная (секционированная)      Тиражированная.

## **Другие виды БД:**

Пространственная              Временная (темпоральная)  
Пространственно-временная      Циклическая.



# СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

## Основные функции СУБД:

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными).

## Современная СУБД содержит **компоненты**:

- ядро – отвечает за управление данными во внешней и оперативной памяти, и журнализацию,
  - процессор языка базы данных – обеспечивает 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 (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.
- 

# СУБД MICROSOFT ACCESS

В СУБД размещены не только данные, но и методы взаимодействия с потребителем или с другими программно-аппаратными комплексами (*хранятся данные и информация*).

СУБД – создание структуры базы, наполнение содержимым, редактирование содержимого и визуализация информации (отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи).

СУБД *Microsoft Access* в обобщенных понятиях и методах различия между конкретными версиями программ второстепенны.

Пустая база (без данных) содержит информацию о структуре БД (методы занесения данных и хранения их в базе).

Простейшая БД имеет хотя бы одну таблицу.

	Компонент	Модель	Основной параметр	Цена	Добави
+	Клавиатура	HP	30	30	
+	Монитор	HP	20	200	
+	Мышь	Ljgitech	10	10	
+	Оперативная память	HP	40	300	
+	Системный блок	HP	60	710	
*			0	0	

Записи базы

Поля базы

# СВОЙСТВА ПОЛЕЙ БД

**Имя поля** – как обращаться к данным поля при автоматических операциях с базой (по умолчанию – заголовки столбцов таблиц).

**Тип поля** – тип данных, которые могут содержаться в поле.

**Размер поля** – предельная длину (в символах) данных.

**Формат поля** – способ форматирования данных в ячейках.

**Маска ввода** – форма, в которой вводятся данные в поле (средство автоматизации ввода).

**Подпись** – заголовок столбца таблицы для данного поля (если не указана, то используется свойство *Имя поля*).

**Значение по умолчанию** – значение, которое вводится в ячейки поля автоматически (средство автоматизации ввода данных).

**Условие на значение** – ограничение для проверки правильности ввода данных (средство автоматизации ввода, например, числовой, денежный типы или тип даты).

**Сообщение об ошибке** – текстовое сообщение, выдается автоматически при попытке ввода ошибочных данных (автоматически, если задано *Условие на значение*).

**Обязательное поле** – свойство, обязательности заполнения поля при наполнении базы.

**Пустые строки** – свойство, разрешающее ввод пустых строковых данных (относится не ко всем типам данных, например к текстовым).

**Индексированное поле** – свойство ускоряющее операции поиска или сортировки записей по значению, хранящемуся в данном поле (можно использовать для контроля на наличие повторов данных).

# ТИПЫ ДАННЫХ

Кроме текстовых, числовых и формул таблицы БД допускают работу с большим количеством разных типов данных.

**Текстовый** – тип данных для хранения неформатированного текста (до 255 символов).

**Поле Мемо** – спец. тип данных для хранения больших объемов информации (физически в поле хранится указатель на место хранения).

**Числовой** – тип данных для хранения действительных чисел.

**Дата/время** – тип данных для хранения календарных дат и текущего времени.

**Денежный** – тип данных для хранения денежных сумм (особенности правил округления отличны от числового типа).

**Счетчик** – спец. тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием (для порядковой нумерации записей).

**Логический** – тип для хранения логических данных (значения Да или Нет).

**Поле объекта OLE** – спец. тип данных, для хранения объектов *OLE*, например мультимедийных (в таблице хранятся указатели).

**Гиперссылка** – спец. поле для хранения адресов *URL* Web-объектов.

**Мастер подстановок** – объект, для автоматизации ввод в данных поле.



# БЕЗОПАСНОСТЬ БД

Для баз данных предъявляются **особые требования** с точки зрения безопасности, поэтому в них реализован другой подход к сохранению данных.

Обычно при работе для сохранения данных мы выдаем команду операционной системе.

Целостность содержимого базы не зависит от конкретных пользователей и от перебоев в электросети.

В СУБД для сохранения информации используется **двойной подход**.

Операции изменения структуры базы данных, создания новых таблиц или иных объектов происходят при сохранении файла базы данных (СУБД предупреждает пользователя). Их никогда не проводят с базой данных, находящейся в коммерческой эксплуатации – только с ее копией.

Операции по изменению содержания данных максимально автоматизированы и выполняются **без предупреждения** (изменения сохраняются немедленно и автоматически).

Редактируя таблицы БД, создавая новые записи и удаляя старые, **работаем с жестким диском** напрямую, минуя операционную систему. Попытка закрыть базу «без сохранения» ничего не даст, так как все уже сохранено.

Нельзя заниматься учебными экспериментами на БД, находящихся в эксплуатации. Следует создавать специальные учебные базы или выполнять копии структуры реальных баз (без фактического наполнения данными).

# РЕЖИМЫ РАБОТЫ С БД

В БД различают категории исполнителей.

*Проектировщики* – разрабатывают структуру таблиц БД и согласуют ее с заказчиком (не наполняют базу данными, исключение составляет экспериментальное наполнение модельными данными на этапе отладки объектов базы).

*Пользователи* получают БД от проектировщиков, наполняют ее данными и обслуживают (часто не имеют средств доступа к управлению структурой базы, а только к данным и только на конкретном рабочем месте).

Система управления БД имеет режимы работы: *проектировочный* и *пользовательский*.

## Основные типы объектов СУБД Microsoft Access:

**Таблицы**

**Запросы**

**Формы**

**Отчеты**

**Страницы**

**Макросы**

**Модули**



# Концепция баз данных

Основная идея БД – принцип «**независимости программ и данных**».

Принцип реализуется разделением логического и физического уровней представления и хранения данных.

В базах данных:

- на физическом уровне данные хранятся в файлах определённого формата; физический уровень полностью скрыт для прикладных программ;
- логическая структура данных описывается отдельно с использованием языка описания данных (ЯОД);
- обработка запросов к БД реализуется через специальный интерфейс манипулирования данными (например, язык SQL);
- прикладные программы обработки данных используют программный интерфейс СУБД (библиотеку специальных процедур) логического уровня.

«Мельчайшей порцией» описания предметной области является отдельно взятое свойство – *элемент данных*.

Элемент данных
<имя>, <тип>, <значение>

Описание сложного объекта можно получить как совокупность элементов данных, связанных между собой тем или иным способом. Конфигурация этих связей определяет *логическую структуру* данных.



# ПОНЯТИЕ МОДЕЛИ ДАННЫХ

**Модель данных** – формальная теория представления и обработки данных в СУБД, включающая:

- 1) структуру – методы описания типов и логических структур данных в БД;
- 2) манипуляции – методы манипулирования данными (способы перехода между состояниями базы данных или способы модификации данных и способы извлечения данных из базы данных);
- 3) целостность – методы описания и поддержки целостности базы данных (средства описаний корректных состояний базы данных).

**Модель данных** есть их логическая структура, она отражает объективно существующие взаимосвязи в предметной области и не зависит от физического представления данных или программной реализации процедур.

# ИНФОРМАЦИОННАЯ МОДЕЛЬ

**Свойство** – существенная характеристика объекта, имеющая количественное значение, выражаемое числом, или качественное значение, выражаемое текстом. Свойство описывается набором из трёх признаков: 1) имя, 2) тип, 3) значение.

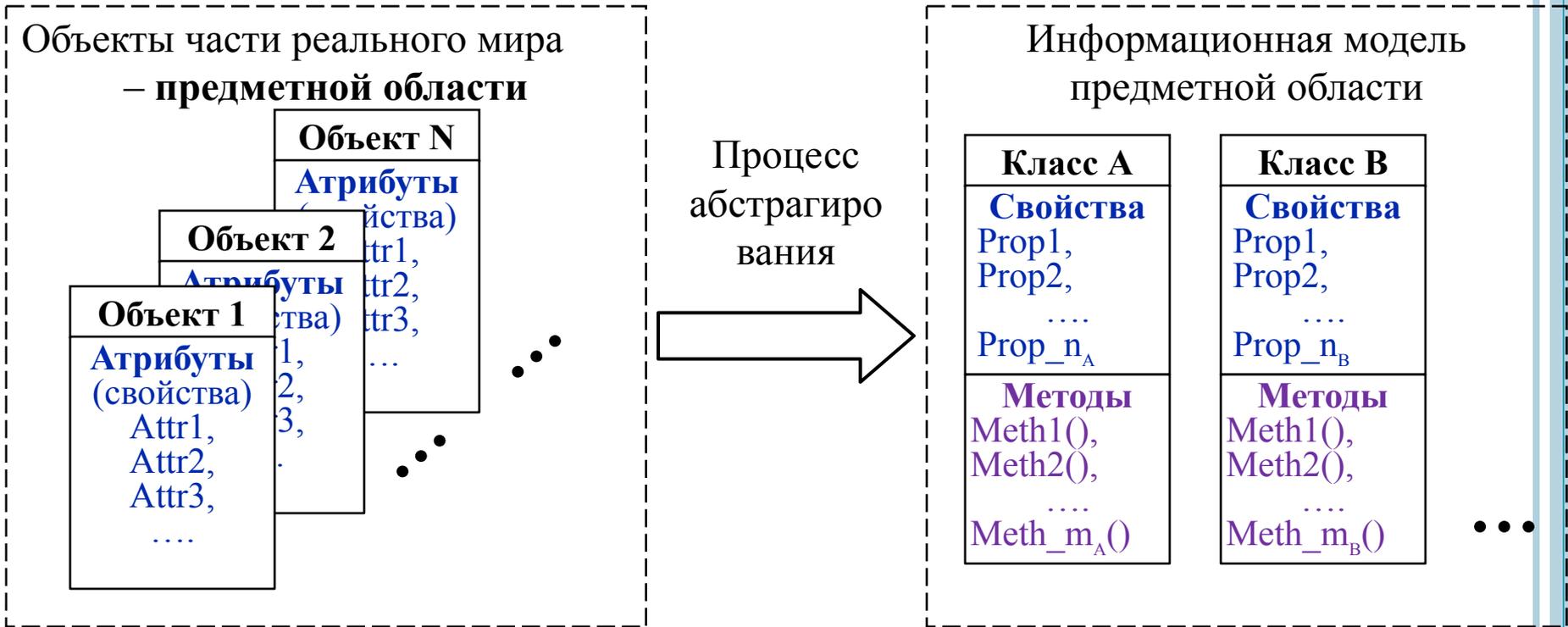
**Имя** однозначно идентифицирует свойство.

**Тип** соответствует множеству допустимых значений свойства, которое называется **доменом**. Тип и домен взаимно однозначно определяют друг друга.

**Значение** – одно из значений, принадлежащее соответствующему домену. Иногда свойство может иметь особое значение NULL, что означает отсутствие значения.

# МОДЕЛЬ ДАННЫХ ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД

Объектно-ориентированная база данных (ООБД) – данные моделируются в виде объектов, их атрибутов, методов и классов



## Отношение A

(Prop1, Type1), (Prop2, Type2), ..., (Prop\_n\_A, Type\_n\_A)

## Кортежи отношения A

(Prop1, Value11), (Prop2, Value12), ..., (Prop\_n\_A, Value1n\_A),

(Prop1, Value21), (Prop2, Value22), ..., (Prop\_n\_A, Value2n\_A),

...



# ПРИМЕР ИНФОРМАЦИОННОЙ МОДЕЛИ

**Класс** – абстракция общих свойств родственных (сходных) объектов. В класс помещаются также общие процедуры обработки объектов – **методы**.

Например, имеются «объекты предметной области»:

Иванов Пётр Андреевич, 1990 года рожд., холост, студент гр. У0523;  
Петров Андрей Сергеевич, 1980 года рожд., женат, инженер отд. «Х»; ...

Общее – люди.

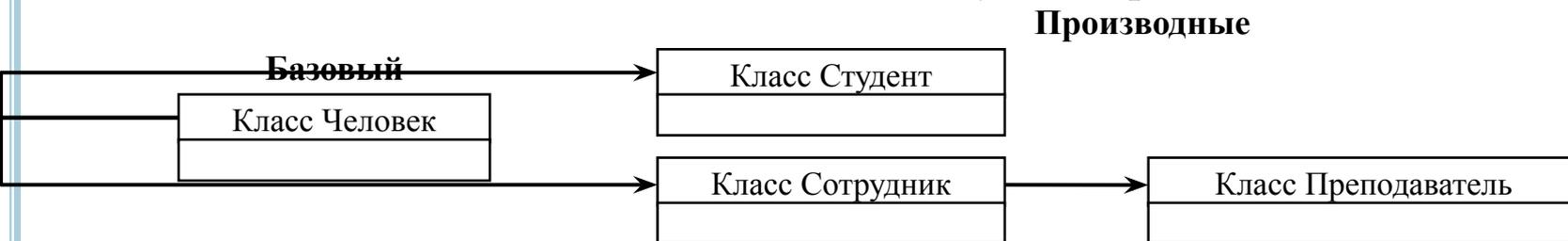
Класс характеризуется набором свойств:

```
класс Человек {  
    текстовый    Фамилия;  
    текстовый    Имя;  
    текстовый    Отчество;  
    целый        ГодРождения;  
    текстовый    СемейноеПоложение;  
    ...  
}
```

Все перечисленные люди являются студентами или сотрудниками некоторого учебного заведения. Для них можно задать, например, следующие методы (процедуры обработки):

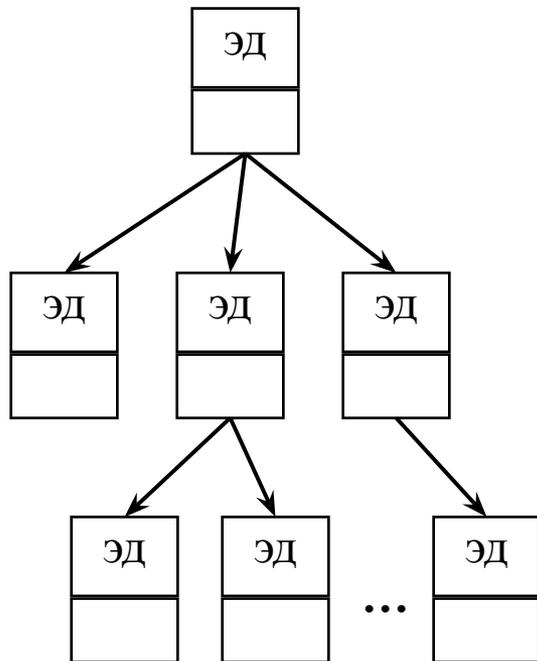
```
ПринятьЗачислить(); //на работу или на обучение  
Перевести();        //на другую должность или в другую группу  
УволитьОтчислить(); ...
```

Новый, производный класс может быть образован от другого, базового класса. При этом свойства и методы базового класса наследуются производным классом.

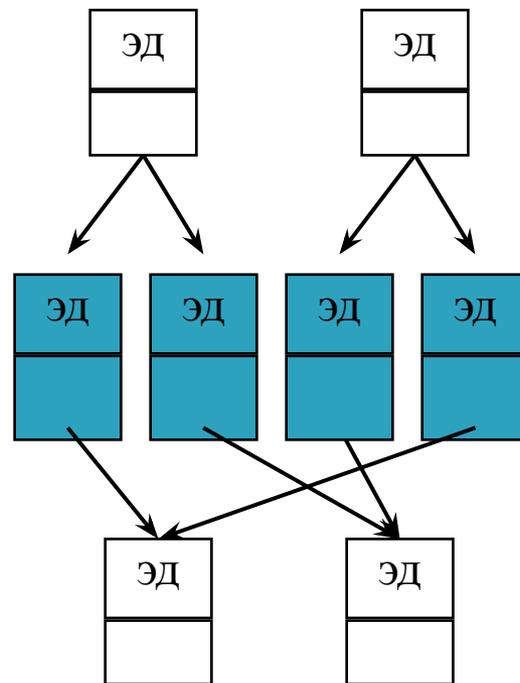


# ОСНОВНЫЕ МОДЕЛИ ДАННЫХ

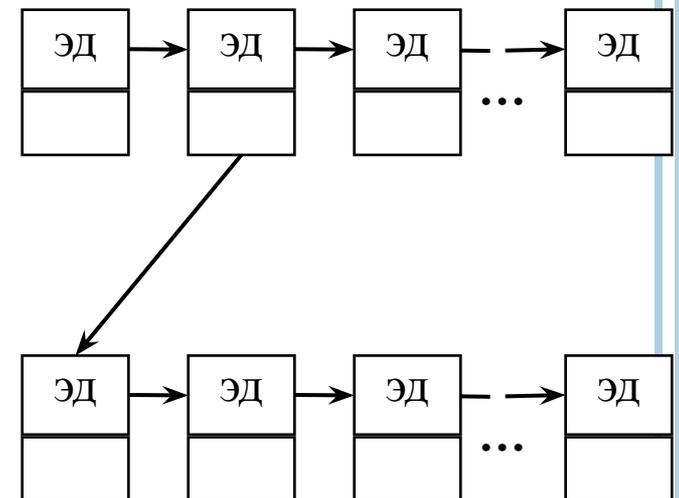
Иерархическая



Сетевая



Реляционная



# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

*Иерархическая модель данных* отражает древовидную логическую структуру, часто встречается в практике. Модель адекватно отражает отношения подчинения, отношения части и целого и т.п.

- Имеется один главный узел, корень дерева иерархии.
- Каждый из остальных, подчинённых, узлов имеет один родительский узел.
- Каждый узел может иметь один или несколько подчинённых узлов или не иметь их вовсе.
- Узлы, не имеющие подчинённых, называются терминальными (листья дерева).

Между узлом и подчинёнными узлами по условию модели существует связь «один-ко-многим».

**Основные недостатки иерархической модели:**

- сложность отображения связи "многие-ко-многим" ;
- сложность операции включения новых объектов и удаления устаревших объектов непосредственно в базе данных (в особенности обновление и удаление связей).



# СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Сетевая структура отчасти похожа на древовидную, но здесь нарушаются принципы организации дерева:

- может существовать несколько корневых узлов;
- узел может иметь более одного родительского узла (обычно два, что называют бинарной связью типа «многие-к-одному»).

В *сетевой модели* элементы данных объединяются в наборы двух видов: наборы-данные и наборы-связи (на схеме выделены цветом).

Основные недостатки *сетевой модели*:

- сложность самой модели данных;
- сложность операций манипулирования данными (в особенности сложность отслеживания цепочек связанных групповыми отношениями данных при операциях вставки, обновления и удаления).

## РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

- элементы данных связаны в цепочки;
- каждая цепочка представляет один объект, экземпляр класса.
- связь между двумя экземплярами разных классов задаётся отдельным элементом одного экземпляра как ссылка на ключевой элемент другого экземпляра.

**Достоинства:**

- 1) однородность представления данных в модели, которая обуславливает простоту восприятия её конструкций пользователями базы данных,
- 2) наличие развитой математической теории, реляционной алгебры, которая обуславливает корректность её применения.



# РЕЛЯЦИОННАЯ МОДЕЛЬ

Реляционные модели (англ. relation – отношение, Эдгар Кодд, 1970 ) – характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц.

Каждая реляционная таблица представляет собой двумерный массив и обладает свойствами:

- каждый элемент таблицы – один элемент данных (в таблице нет двух одинаковых строк);
- все ячейки в столбце таблицы однородные, т.е. элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.);
- таблица имеет столбцы, соответствующие атрибутам отношения, каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Базовые понятия реляционных БД: отношение; атрибут; кортеж .

В состав РМД может входить несколько отношений (таблиц).

Одна таблица представляет один объект и состояние столбцов и строк.

**Основной недостаток** – избыточность.



# СООТВЕТСТВИЕ ПОНЯТИЙ

Реляционная модель – 60-е годы Объектная модель – 80-е годы XX века.

Обе модели описывают одну и ту же реальность с разных позиций, установим соответствие (неполное) основных понятий.

Понятие реляционной модели данных	Понятие объектной информационной модели предметной области
атрибут	свойство
домен	тип
отношение	совокупность класса и множества его экземпляров
заголовок отношения	класс
тело отношения	множество экземпляров класса
кортеж отношения	экземпляр класса

Заголовок отношения Человек

(КодПерсоны, счётчик), (Фамилия, текстовый), (Имя, текстовый), ...

Тело отношения Человек (множество кортежей)

(2305, Иванов, Пётр, Андреевич, 1990, холост), (2403, Петров, Андрей, Сергеевич, 1980, женат),

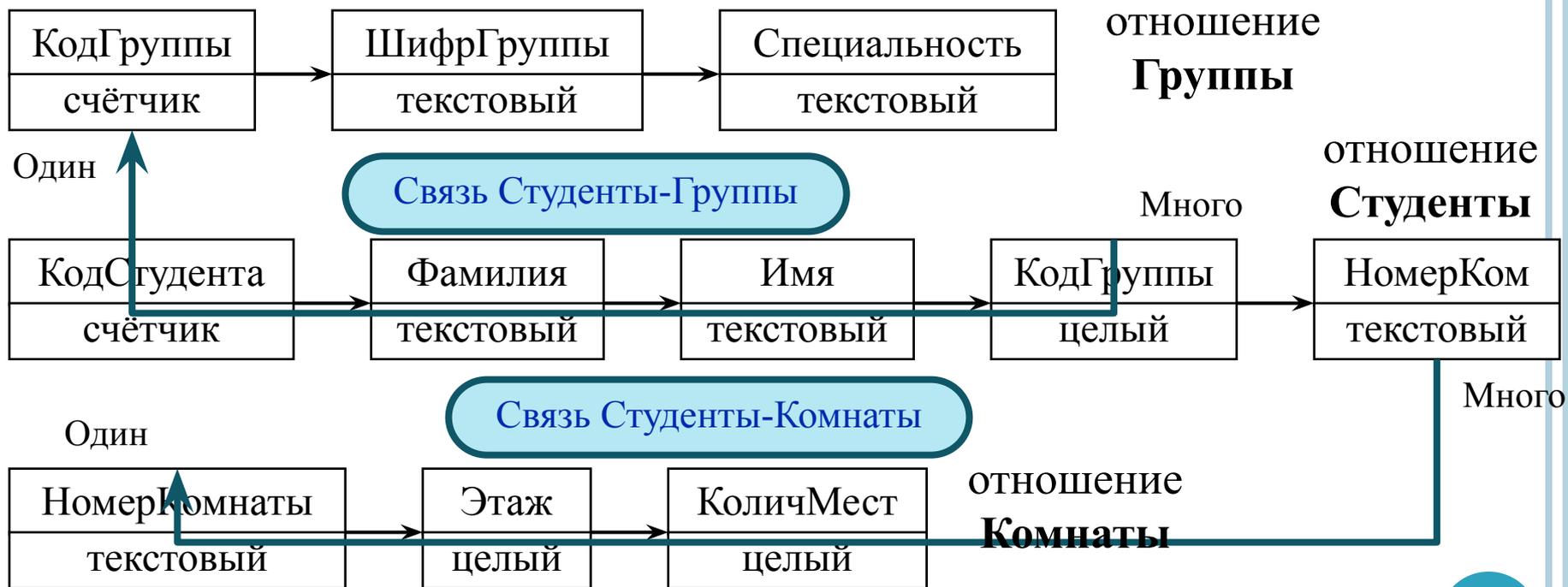


# СВЯЗАННЫЕ ОТНОШЕНИЯ

Рассмотрим предметную область: студенты обучаются в группах по специальностям, часть студентов проживает в общежитии по комнатам.

Выделены три класса объектов, соответственно – три отношения:

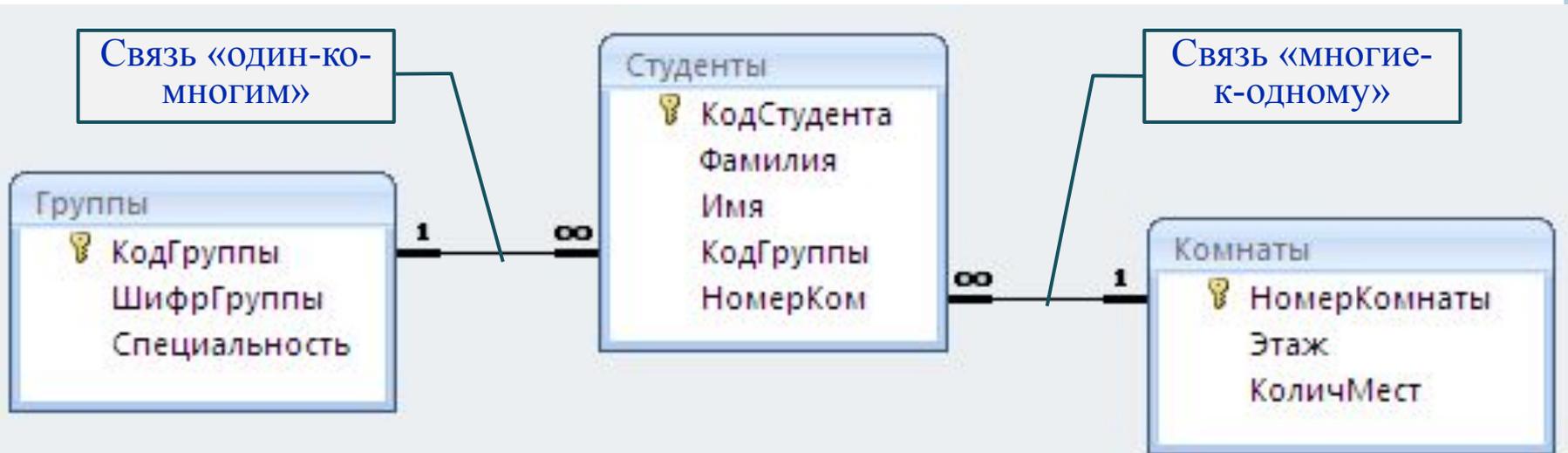
**Студенты, Группы, Комнаты.**



Связь отношения Группы с отношением Студенты называется – «**один-ко-многим**».  
Связь отношения Студенты с отношением Группы – «**многие-к-одному**».

# ОТОБРАЖЕНИЕ СВЯЗАННЫХ ОТНОШЕНИЙ

Наиболее распространённый способ изображения связанных отношений — диаграммы «сущность-связь» (ER-диаграммы, от англ. **Entity-Relation**).



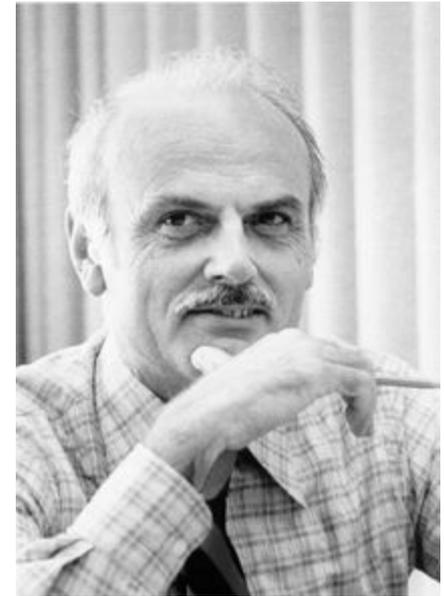
Эту диаграмму называют также логической схемой базы данных.

**Первичный ключ** — уникален для каждого кортежа отношений; служит для обеспечения целостности данных.

**Транзакция** — группа последовательных операций, представляющая логически неделимую единицу работы с данными. Транзакция может быть либо выполнена успешно целиком с гарантией целостности данных, либо не выполнена вообще. СУБД, поддерживающие транзакции, автоматически ведут журнал транзакций, который обеспечивает возможность отката, т.е. возврата БД в исходное состояние при неуспешном завершении транзакции.

# РЕЛЯЦИОННАЯ АЛГЕБРА КОДДА

Эдгар Франк Кодд (1923 - 2003)  
реляционной модели данных



- Теоретико-множественные операции:
- объединение отношений;
  - пересечение отношений;
  - взятие разности отношений;
  - расширенное прямое произведение отношений.

- Специальные реляционные операции:
- ограничение отношения;
  - проекция отношения;
  - соединение отношений;
  - деление отношений.

Все операции бинарные. Первые три операции требуют *совместности отношений по объединению*.

Два отношения называются *совместными*, если они имеют один и тот же набор атрибутов и одноимённые атрибуты определены на одном и том же домене.

«Обычное» прямое (декартово) произведение двух множеств – это множество, составленное из всех возможных пар элементов множеств-операндов. Если применить эту операцию к отношениям, то результат не будет отношением.

При взятии расширенного прямого произведения двух отношений заголовком результирующего отношения становится *конкатенация* заголовков исходных отношений, а кортежи представляют собой *конкатенацию* упомянутых пар кортежей отношений-операндов.

# РЕЛЯЦИОННАЯ АЛГЕБРА

Пусть  $Q_a, Q_b, Q_c$  – множество кортежей совместных отношений  $A, B, C$  соответственно.

**Объединение отношений**

**C = A UNION B**

$$Q_c = Q_a \cup Q_b = \{Q_a + Q_b, \text{отличные от } Q_a\}.$$

**Пересечение отношений**

**C = A INTERSECT B**

$$Q_c = Q_a \cap Q_b = \{Q_a, \text{совпадающие с } Q_b\}$$

**Разность отношений**

**C = A MINUS B**

$$Q_c = Q_a - Q_b = \{Q_a, \text{отличные от } Q_b\}.$$

**Расширенное прямое  
произведение отношений**

**C = A TIMES B**

$$Q_c = Q_a \times Q_b.$$



# Язык реляционных БД – SQL

Большинство современных реляционных БД используют стандартный язык манипулирования данными SQL (англ. **S**tructured **Q**uery **L**anguage – структурированный язык запросов), основанный на реляционной алгебре. Многие конструкции SQL очень похожи на выражения реляционной алгебры (оператор *SELECT*). Однако, возможности языка выходят за её рамки.

Некоторые из дополнительных средств:

- Операторы создания и изменения отношений и индексов (*CREATE TABLE*, *ALTER TABLE*, *CREATE INDEX*, *DROP INDEX*).
- Операторы вставки, удаления и изменения кортежей (*INSERT*, *DELETE*, *UPDATE*).
- Арифметические выражения: в SQL, помимо логических, можно использовать и арифметические выражения, а также набор математических функций.
- Итоговые функции: такие как *Sum()*, *Count()*, *Avg()*, *Max()* и др. могут применяться к столбцам отношения для получения обобщающей величины.

# ЭТАПЫ ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ БД

ИТ-проекты по созданию базы данных включают в себя следующие этапы:

- определение стратегии построения системы,
- анализ требований к базе данных,
- проектирование базы данных,
- реализация базы данных,
- тестирование и внедрение базы данных.

*Проектирование базы данных* – это поиск способов удовлетворения функциональных требований средствами имеющейся компьютерной технологии с учётом заданных ограничений.

Проектирование базы данных, является плохо структурированной задачей. Оно начинается с момента принятия стратегических решений и продолжается на этапах реализации и тестирования.

# ТЕНДЕНЦИИ РАЗВИТИЯ СУБД

Дальнейшее развитие СУБД предусматривает:

- расширение спецсредств (мастера, конструкторы, эксперты);
- наличие собственных языковых средств;
- применение объектной технологии типа OEL;
- возможность работы в локальном режиме и в режиме клиента;
- улучшение многоволоконного графического интерфейса;
- улучшение параллельной обработки информации на многопроцессорных вычислительных схемах;
- развитие БД, СУБД и Internet происходит в направлении расширения их интеллекта за счет развития и применения интеллектуальных программ и языковых средств, к созданию баз знаний в соответствующих предметных областях и глобальных баз знаний.

