

Системы управления базами данных

Модели данных

Основы проектирования баз данных

# Определение базы данных

- **База данных (БД)** — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).
- **Выделяются следующие признаки баз данных:**
  - БД хранится и обрабатывается в вычислительной системе.
  - Данные в БД логически структурированы (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе.
  - БД включает схему, или метаданные, описывающие логическую структуру БД в формальном виде (в соответствии с некоторой метамоделью).

# Система управления базами данных

- *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).

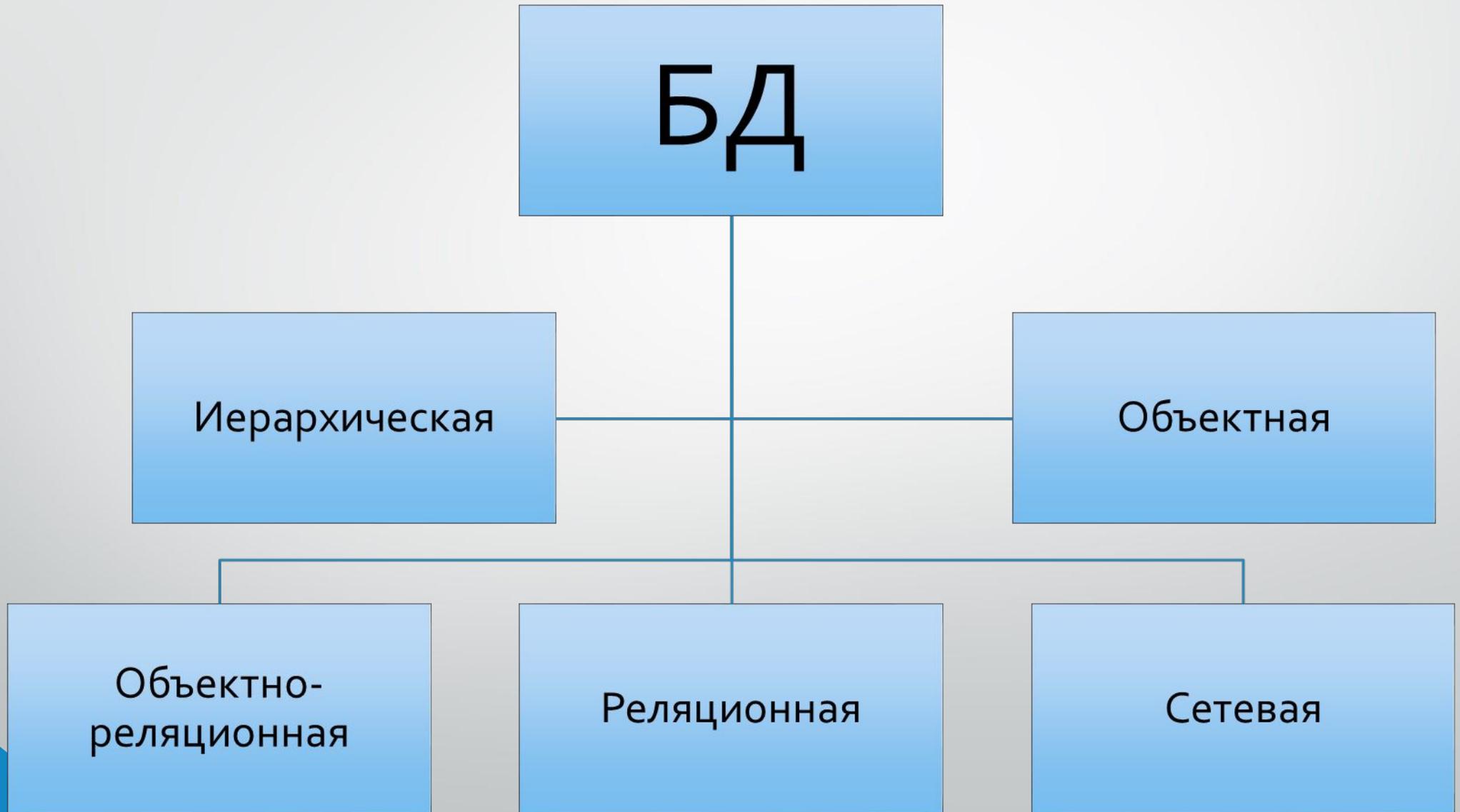
# Система управления базами данных

- **Обычно современная СУБД содержит следующие компоненты:**
  - **ядро**, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию,
  - **процессор языка базы данных**, 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
  - а также **сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

# Метаданные

- **Метаданные**, в общем случае, это информация о базе данных:
- **1 - Метаданные** — это субканальная информация об используемых данных[1].
- **2 - Структурированные данные**, представляющие собой характеристики описываемых сущностей для целей их идентификации, поиска, оценки, управления ими.
  - набор допустимых структурированных описаний, которые доступны в явном виде и предназначение которых может помочь найти объект.
- **3 - Данные из более общей формальной системы**, заданную с описывающей свойства системы данных.
- **4 - Информация о содержащейся на веб-странице свойства информации** (создателе и т. п.). Пример: Имя автора правки в тексте.

# Виды моделей данных



# Виды моделей данных

<b>Внешняя</b>	Для пользователей, абстрагируется от особенностей реализации. Данные описываются в терминах исследуемой области.
<b>Концептуальная</b>	Способ логического упорядочения данных. иерархическая модель данных сетевая модель данных реляционная модель данных
<b>Логическая (внутренняя)</b>	Методы доступа к данным, логическая структура файлов.
<b>Физическая</b>	Поддержка ОС и аппаратными средствами устройств хранения данных.

# Виды моделей данных

- В классической теории баз данных, **модель данных** есть формальная теория представления и обработки данных в системе управления базами данных (СУБД), которая включает, по меньшей мере, **три аспекта**:
  - **аспект структуры**: методы описания типов и логических структур данных в базе данных;
  - **аспект манипуляции**: методы манипулирования данными;
  - **аспект целостности**: методы описания и поддержки целостности базы данных.
- Аспект структуры определяет, что из себя логически представляет база данных, аспект манипуляции определяет способы перехода между состояниями базы данных (то есть способы модификации данных) и способы извлечения данных из базы данных, аспект целостности определяет средства описаний корректных состояний базы данных.
- **Модель данных** — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь.

# Виды моделей данных

- **Иерархическая модель данных** — это модель данных, где используется представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.
- Между объектами существуют **связи**, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении **предка** (объект более близкий к корню) к **потомку** (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок.
- Объекты, имеющие общего предка, называются **близнецами** (в программировании применительно к структуре данных дерево устоялось название **братья**).

# Виды моделей данных

- **Объектные базы данных** (также объектно-ориентированные системы управления базами данных) являются системой управления базами данных, в которых информация представлена в виде объектов, используется в объектно-ориентированном программировании.
- Объектные базы данных отличаются от реляционных баз данных, являющихся таблично-ориентированными. Объектно-реляционные базы данных являются гибридом обоих подходов.
- Объектно-ориентированные системы управления базами данных (ООСУБД) объединяют возможности базы данных с возможностями языка объектно-ориентированного программирования

# Виды моделей данных

- **Объектно-реляционная СУБД** (ОРСУБД) — реляционная СУБД (РСУБД), поддерживающая некоторые технологии, реализующие объектно-ориентированный подход: объекты, классы и наследование реализованы в структуре баз данных и языке запросов.
- Объектно-реляционными СУБД являются, например, широко известные Oracle Database, Informix, DB2, PostgreSQL.
- **Реляционная модель данных** (РМД) — логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики, как теория множеств и логика первого порядка.

# Виды моделей данных

- На реляционной модели данных строятся **реляционные базы данных**.
- **Реляционная база данных** — база данных, основанная на реляционной модели данных. Слово «реляционный» происходит от англ. *relation* («отношение», «зависимость», «связь»). Для работы с реляционными БД применяют реляционные СУБД.
- **Реляционная модель данных включает следующие компоненты:**
- **Структурный аспект** (составляющая) — данные в базе данных представляют собой набор отношений.
- **Аспект (составляющая) целостности** — отношения (таблицы) отвечают определенным условиям целостности. РМД поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных.
- **Аспект (составляющая) обработки** (манипулирования) — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление).

Кроме того, в состав реляционной модели данных включают теорию **нормализации**.

# Виды моделей данных

- **Нормальная форма** — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять **отношение**.
- Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам, называется **нормализацией**.
- **Отношение** — фундаментальное понятие реляционной модели данных.
- **Целостность базы данных** (database integrity) — соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.

# Виды моделей данных

- **Сетевая модель данных** — логическая модель данных, являющаяся расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных.
- Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков.
- **Сетевая БД состоит из** набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между этими записями.
- **Тип связи определяется для двух типов записи:** предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка.
- Для данного типа связи  $L$  с типом записи предка  $P$  и типом записи потомка  $C$  должны выполняться следующие **два условия**:
  - каждый экземпляр типа записи  $P$  является предком только в одном экземпляре типа связи  $L$ ;
  - каждый экземпляр типа записи  $C$  является потомком не более чем в одном экземпляре типа связи  $L$ .

# Сущность

- **Сущность** - это что-то такое, о чем нужно хранить информацию в базе данных.
- При проектировании баз данных достаточно описать происходящую ситуацию - и большинство существительных и часть глаголов будут кандидатами на сущности.
- **Например:** "Покупатели покупают товары. Сотрудники продают товары покупателям. Поставщики поставляют товары" - покупатели, товары, сотрудники и поставщики - это **сущности**. Глаголы "покупать" и "продавать" - тоже **сущности** (хотя могут быть и одной сущностью, разной с точки зрения покупателя и продавца).
- При проектировании БД **главный источник информации о сущностях** - это беседа с заказчиком в целях уяснения его бизнес-процессов.
- Кроме того, анализируются стандартные документы, используемые в бизнес-процессах: бланки, отчеты, инструкции и т.п. После получения такого списка необходимо проверить его на полноту и связность, а также выявить **дубли** - одинаковые сущности, которые называются разными словами, и сущности, которые на самом деле отличаются, но описываются один и ем же термином.

Сущности могут моделировать **конкретные понятия** (клиенты, товары, звонки) и **абстрактные** (агент отвечает за клиента, студент записан на курс).

# Атрибуты

- Записи об определенных параметрах каждой из сущностей называются *атрибутами*.
- *Например*, для сущности "заказчик", видимо, будет храниться информация об его наименовании, представителях, адресе и т.п.
- **Выбор нужного комплекта атрибутов** - одна из самых больших проблем при проектировании баз данных.
- Очень часто в реальной базе данных нужный комплект атрибутов в итоге не хранится - просто по той причине, что пользователи не смогли сообщить в процессе сбора информации, что он действительно нужен.
- Иногда в базе, наоборот, попадают лишние атрибуты, заполнение которых требует дополнительного времени.

Очень часто возникает проблема с форматом вводимых данных, например, на какие части делить адрес и что делать с нестандартными случаями.

# Ключи (идентификаторы)

- **Ключ или потенциальный ключ** – это минимальный набор атрибутов, по значениям которых можно однозначно выбрать требуемый экземпляр сущности.
- Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждая сущность должна но не обязана обладать хотя бы одним возможным ключом.
- **Ключи можно разделить по признаку общности как:**
  - **Простой** – сформирован из значений единственного поля, которые однозначно определяют каждую запись
  - **Составной** – сформирован из значений нескольких полей, применяется
- **По способу возникновения:**
  - **Естественный** – основан на уже существующем поле. Например поле фамилия.
  - **Суррогатный** – основан на добавленном искусственным путем отдельном поле для однозначной идентификации.
  - **Интеллектуальный** ключ – основан на естественном ключе путем добавления

# ER-модель данных

- **Модель сущность-связь** (ER-модель) (англ. entity-relationship model, ERM) — модель данных, позволяющая описывать **концептуальные схемы предметной области**.
- ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить **ключевые сущности** и обозначить **связи**, которые могут устанавливаться между этими сущностями.
- Во время проектирования баз данных происходит преобразование ER-модели в конкретную **схему базы данных** на основе **выбранной модели данных** (реляционной, объектной, сетевой или др.).
- ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации.
- В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена **диаграмма сущность-связь** (ER-диаграмма) (англ. entity-relationship diagram, ERD)