

# Управление данными

Лекция 1

Основные понятия и определения

- **Банк данных** это автоматизированная система специальным образом организованных данных — **баз данных**, программных, технических, языковых, организационно-методических средств и персонала, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных. (архив)
- **Предметная область (ПрО)** – область применения конкретного банка данных (совокупность описаний объектов, представляющих интерес для пользователя).

➤ **Данные** — это информация, фиксированная в определенной форме для последующей обработки, передачи и хранения.

***Различают два аспекта данных:***

инфологический и даталогический.

- **Инфологический аспект** связан со смысловым содержанием данных без учета их представления в памяти машины.
- **Даталогический аспект** связан с представлением данных в памяти машины.

- **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 многих пользователей.
- **База данных (БД)** — это совокупность взаимосвязанных, хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их оптимальное использование для одного или нескольких приложений.
- **База данных** — совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей пользователей.
- Данные организуются так, чтобы они были независимы от программных средств. Добавление (модификация) данных должна осуществляться общим управляемым способом.

➤ **Система управления базой данных (СУБД)** – это программное обеспечение для использования и модификации БД одним или несколькими лицами.

➤ **Система управления базами данных (СУБД)** — программный комплекс, обеспечивающий обслуживание баз данных.

# Функции СУБД:

- 1) Главная: обеспечить пользователя инструментарием, позволяющим оперировать данными в абстрактной форме, не связанным со способами хранения. СУБД представляет собой интерпретатор языка высокого уровня.
- 2) Обеспечение секретности.
- 3) Защита целостности данных (СУБД может осуществлять проверку на ограничение и не противоречивость данных).
- 4) Синхронизация (защита от одновременного доступа несколькими пользователями к одному элементу данных).
- 5) Защита от отказов и восстановление (обеспечение создания копий для восстановления БД после устранения причин сбоя).

# Банк данных как АИС (Автоматизированная Информационная Система)

Основные компоненты банка данных: БД, СУБД, администратор БД (АБД), словарь данных, обслуживающий персонал, вычислительная система.

- 1) **БД** – даталогическое представление информационной модели предметной области.
- 2) **СУБД** – программа, полностью управляющая доступом к БД. СУБД включает трансляторы (интерпретаторы) с языка описания данных (ЯОД) и языка манипулирования данными (ЯМД).

***ЯОД*** – язык высокого уровня, предназначенный для задания схемы БД, то есть описывающий типы данных, их структуру и взаимосвязь между ними. Это декларативный язык.

***ЯМД*** – язык запросов – система команд, осуществляющая манипулирование данными.

- 3) **Словарь данных** - специальная система, предназначенная для хранения единообразной и централизованной информации обо всех ресурсах конкретного БнД.
- 4) **Администратор базы данных (АБД)** – лицо или группа лиц, реализующих управление БД.

# Требования к организации БД

Пользовательские требования :

- 1) Многократное использование данных. (Пользователи, которые по-разному понимают данные, могут их по-разному использовать).
- 2) Сохранение затрат умственного труда. (Существующие программы и структуры данных не изменяются при внесении изменений в БД).
- 3) Низкие затраты. (Низкая стоимость хранения и использования данных, минимизация затрат на внесение изменений в БД).
- 4) Уменьшение избыточности данных. (Требование новых приложений удовлетворяется за счёт уже имеющихся данных, а не путем создания новых файлов).



- 5) Производительность. (Запросы на данные удовлетворяются с такой скоростью, которая требуется для использования данных).
- 6) Простота. (Пользователи могут удалить и понять какие данные имеются в их распоряжении).
- 7) Гибкость использования. (Обращение к данным или их использование осуществляется с применением различных методов доступа).
- 8) Быстрая обработка непредсказуемых запросов. (Случайные запросы на данные обрабатываются с помощью языка высокого уровня запросов или генератора отсчетов, а не прикладными программами, написанными программистами).

9) Достоверность данных и соответствие их одному уровню обновления.  
(Система запрещает наличие одной и той же версии элементов данных).

10) Секретность. (Несанкционированный доступ к данным невозможен).  
Одни и те же данные получают пользователи различных уровней различными путями.

11) Защита от уничтожения и искажения. (Данные защищают от сбоя, вмешательств и т.д).

12) Готовность. (Пользователь быстро получает данные, когда ему это необходимо).

# Функциональные требования:

1. Физическая независимость данных.
2. Логическая независимость данных.
3. Контролируемая избыточность.
4. Обеспечение требуемой скорости доступа.
5. Обеспечение требуемой скорости поиска.
6. Стандартизация данных в пределах учреждения.
7. Словарь данных.
8. Наличие интерфейса высокого уровня для связи с пользователем.
9. Наличие языка взаимодействия пользователя с системой.
10. Контроль целостности данных.
11. Восстановление данных после сбоев.
12. Вспомогательные средства для проектирования и управления.
13. Автоматическая реорганизация или перемещение.

# Модели представления 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.

Эти объекты позволяют моделировать структуру данных, а операторы — поведение данных.

Каждая БД и СУБД строится на основе некоторой явной или неявной модели данных.

## Типы моделей представления данных:

1. Иерархическая
2. Сетевая
3. Реляционная
4. Постреляционная
5. Объектная

# Иерархические системы

Основаны на иерархической модели.

Типичным представителем (наиболее известным и распространенным) является Information Management System (IMS) фирмы IBM.

Первая версия появилась в 1968 г. До сих пор поддерживается много баз данных, что создает существенные проблемы с переходом как на новую технологию БД, так и на новую технику.

## Иерархические структуры данных

Иерархическая БД состоит из *упорядоченного набора деревьев*;

более точно, из упорядоченного набора нескольких экземпляров одного типа дерева.

Тип дерева состоит из одного "корневого" типа записи и упорядоченного набора из нуля или более типов поддеревьев (каждое из которых является некоторым типом дерева).

Тип дерева в целом представляет собой иерархически организованный набор типов записи.

Пример типа дерева (схемы иерархической БД):

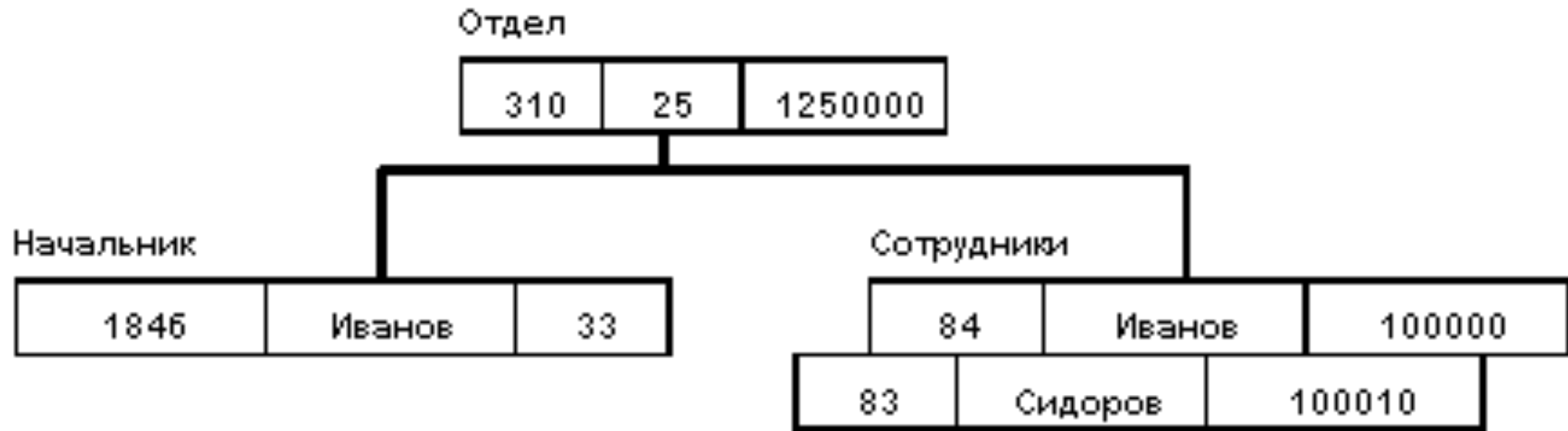


Здесь Отдел является предком для Начальник и Сотрудники, а Начальник и Сотрудники - потомки Отдел.

Между типами записи поддерживаются связи.



База данных с такой схемой могла бы выглядеть следующим образом (мы показываем один экземпляр дерева):



Все экземпляры данного типа потомка с общим экземпляром типа предка называются близнецами. Для БД определен полный порядок обхода - сверху-вниз, слева-направо.

В IMS использовалась оригинальная и нестандартная терминология: "сегмент" вместо "запись", а под "записью БД" понималось все дерево сегментов.

## **Манипулирование данными**

Примерами типичных операторов манипулирования иерархически организованными данными могут быть следующие:

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

## Ограничения целостности

- Автоматически поддерживается целостность ссылок между предками и потомками.

Основное правило: никакой потомок не может существовать без своего родителя.

Заметим, что аналогичное поддержание целостности по ссылкам между записями, не входящими в одну иерархию, не поддерживается.

- В иерархических системах поддерживалась некоторая форма представлений БД на основе ограничения иерархии. Примером представления приведенной выше БД может быть иерархия



# Сетевые системы

Типичным представителем является Integrated Database Management System (IDMS) компании Cullinet Software, Inc., предназначенная для использования на машинах основного класса фирмы IBM под управлением большинства операционных систем.

Архитектура системы основана на предложениях Data Base Task Group (DBTG) Комитета по языкам программирования Conference on Data Systems Languages (CODASYL), организации, ответственной за определение языка программирования Кобол.

Отчет DBTG был опубликован в 1971 г., а в 70-х годах появилось несколько систем, среди которых IDMS.

# Сетевые структуры данных

- Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков.
- Сетевая БД состоит из набора записей и набора связей между этими записями, а если говорить более точно, из набора экземпляров каждого типа из заданного в схеме БД набора типов записи и набора экземпляров каждого типа из заданного набора типов связи.
- 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.

Простой пример сетевой схемы БД:



## **Манипулирование данными**

Примерный набор операций может быть следующим:

- Найти конкретную запись в наборе однотипных записей (инженера Гусева);
- Перейти от предка к первому потомку по некоторой связи (к первому сотруднику отдела 310);
- Перейти к следующему потомку в некоторой связи (от Гусева к Рыбину);
- Перейти от потомка к предку по некоторой связи (найти отдел Гусева);
- Создать новую запись;
- Уничтожить запись;
- Модифицировать запись;
- Включить в связь;
- Исключить из связи;
- Переставить в другую связь и т.д.

## **Ограничения целостности**

В общем их поддержание не требуется, но иногда требуют целостности по ссылкам (как в иерархической модели).



# Достоинства и недостатки

## **Сильные места ранних СУБД:**

- Развитые средства управления данными во внешней памяти на низком уровне;
- Возможность построения вручную эффективных прикладных систем;
- Возможность экономии памяти за счет разделения подобъектов (в сетевых системах).

## **Недостатки:**

- Слишком сложно пользоваться;
- Фактически необходимы знания о физической организации;
- Прикладные системы зависят от этой организации;
- Их логика перегружена деталями организации доступа к БД.

# Реляционная модель

## Реляционная модель данных: кем, когда и для чего создана

**Реляционная модель данных** - созданная Эдгаром Коддом.

Логическая модель данных, описывающая:

- структуры данных в виде (изменяющихся во времени) наборов отношений;
- теоретико-множественные операции над данными: объединение, пересечение разность и декартово произведение;
- специальные реляционные операции: селекция, проекция, соединение и деление;
- специальные правила, обеспечивающие целостность данных.

**Реляционная модель данных** - это способ рассмотрения данных, то есть предписание для способа представления данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов).

Она связана с тремя аспектами данных: структурой (объекты), целостностью и обработкой данных (операторы).

В 2002 журнал Forbes поместил реляционную модель данных в список важнейших инноваций последних 85 лет.

**Цели создания реляционной модели данных:**

- обеспечение более высокой степени независимости от данных;
- создание прочного фундамента для решения семантических вопросов и проблем непротиворечивости и избыточности данных;
- расширение языков управления данными за счёт включения операций над множествами.

# Структура данных в реляционной модели данных

- Реляционная модель данных предусматривает структуру данных, обязательными объектами которой являются:
- отношение;
- атрибут;
- домен;
- кортеж;
- степень;
- кардинальность;
- первичный ключ.

**Отношение** - это плоская (двумерная) таблица, состоящая из столбцов и строк:

Например:

идент	фио	пол	должность	год рожден
123	Гусев В. В.	М	Директор	28.09.1987
124	Рыбин В. Н.	М	Зам. директора	19.08.1989
125	Цветкова А. И.	Ж	Бухгалтер	23.03.1994
126	Пирогов М. А.	М	Завхоз	21.04.1977
127	Булкина Н. А.	Ж	Экономист	05.02.1985

**Отношение** (формальный термин) - таблица (неформальный термин);

**Атрибут** - это поименованный столбец отношения (столбец);

**Домен** - это набор допустимых значений для одного или нескольких атрибутов.

(общая совокупность допустимых значений)

**Кортеж** - это строка отношения (строка или запись);

**Степень** определяется количеством атрибутов, которое оно содержит (количество столбцов)

**Кардинальность** - это количество кортежей, которое содержит отношение

(количество строк)

**Первичный ключ** - это уникальный идентификатор для таблицы (уникальный идентификатор)

# Отношения и их реализация в реляционной модели данных

**Отношение  $R$**  на множестве доменов  $D_1, D_2, \dots, D_n$  - это подмножество декартова произведения этих доменов:  $R \subseteq D_1 \times D_2 \times \dots \times D_n$

*Пример 1.* Определены домены:  $D_1$  - множество фамилий преподавателей,  $D_2$  - множество аудиторий,  $D_3$  - множество учебных групп,  $D_4$  - множество учебных дисциплин. Записать отношения:  
1) закрепление преподавателей за учебными курсами; 2) расписание занятий в группах.

Решение.

1) закрепление преподавателей за учебными курсами:  $R \subseteq D_1 \times D_4$ .

Это отношение определяет множество преподавателей, ведущих множество учебных дисциплин.

2) расписание занятий в группах:  $R \subseteq D_2 \times D_3 \times D_4$ .

Это отношение определяет множество аудиторий, в которых проводятся занятия по множеству учебных дисциплин для множества учебных групп.



**Пример 2.** Пусть домен "Свет"  $D_1 = \{\text{красный, желтый, зеленый}\}$ , а домен действие  $D_2 = \{\text{идти, стоять}\}$ , тогда  $D_1 * D_2 = \{(\text{красный, идти}), (\text{красный, стоять}), (\text{желтый, идти}), (\text{желтый, стоять}), (\text{зеленый, идти}), (\text{зеленый, стоять})\}$ .

Из приведенного выше декартова произведения выберем подмножество, которое образует отношение "светофор".

Светофор - имя отношения.

Свет	Действие
красный	стоять
желтый	стоять

кортеж

атрибуты

## Свойства отношений:

- уникальное (информативное) имя отношения;
- уникальное (информативное) имя атрибута;
- нет одинаковых кортежей;
- кортежи не упорядочены сверху вниз;
- атрибуты не упорядочены слева направо;
- все значения атрибутов атомарные (нормализованное отношение).

Таким образом, **реляционная база данных** - это набор нормализованных отношений.

**Переменная отношения** - это именованный объект, значение которого может изменяться с течением времени.

Переменная отношения в разное время - это различные таблицы базы данных, у которых разные строки, но одинаковые столбцы.

## Достоинства реляционной модели

- Традиционность
- Простота построения
- Простота понимания
- Эффективная организация хранения данных
- Проработанные алгоритмы нормализации
- Основывается на мощном, хорошо проработанном математическом аппарате (реляционной алгебре)
- И др.

**СПАСИБО ЗА ВНИМАНИЕ!**