



# **СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ**

# ОГЛАВЛЕНИЕ

1. Основные понятия баз данных
2. Определение СУБД. Ее классификация.
3. Модель данных СУБД MS ACCESS
4. Предназначение СУБД Access. Схемы СУБД.
5. Список используемой литературы



# 1. *ОСНОВНЫЕ ПОНЯТИЯ БД И СУБД*

Всякая прикладная программа является отображением какой-то части реального мира и поэтому содержит его описание в виде **данных**. Крупные массивы данных размещают, как правило, отдельно от исполняемой программы, и организуют в виде **БД**.

**База данных** - это совокупность предназначенных для машинной обработки данных, которая служит для удовлетворения нужд большого количества пользователей.

**База данных** — поименованная, целостная, единая система данных, организованная по определённым правилам, которые предусматривают общие принципы описания, хранения и обработки данных.



## Система управления базами данных (СУБД) - это

комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями.



База знаний – это формализованная система сведений о некоторой предметной области, содержащая данные о свойствах объектов, закономерностях процессов и явлений и правила использования в задаваемых ситуациях этих данных для принятия новых решений.

Автоматизированный банк данных – это организационно-техническая система, представляющая собой совокупность баз данных пользователей, технических и программных средств формирования и ведения этих баз и персонала, обеспечивающих функционирование системы



# БАНКОВ ДАННЫХ ПОЗВОЛЯЕТ ОБЕСПЕЧИТЬ:

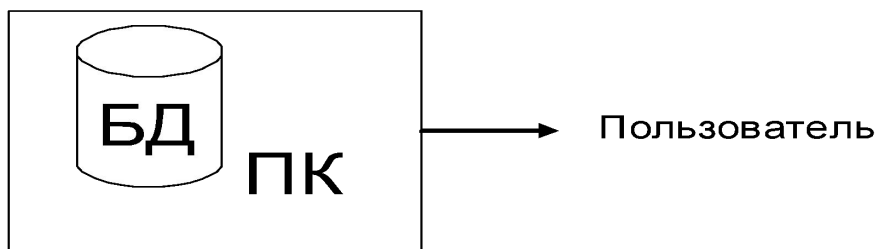
- Многоаспектный доступ к совокупности взаимосвязанных данных
- Достаточно высокую степень независимости прикладных программ от изменений логической и физической организации данных
- Интеграцию и централизацию управления данными
- Устранение излишней избыточности данных

## Составные части банка данных:

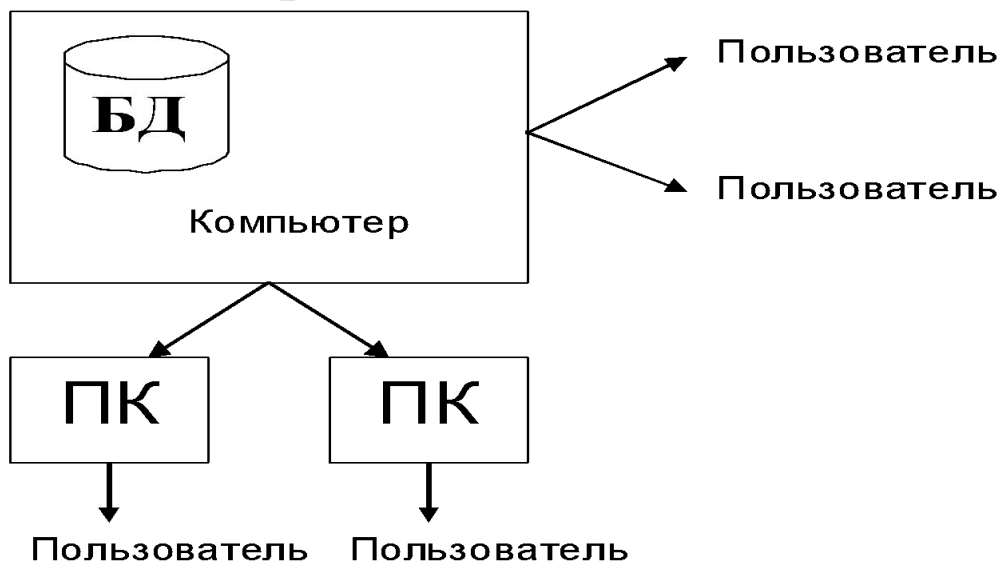


# КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

## □ Централизованная база данных

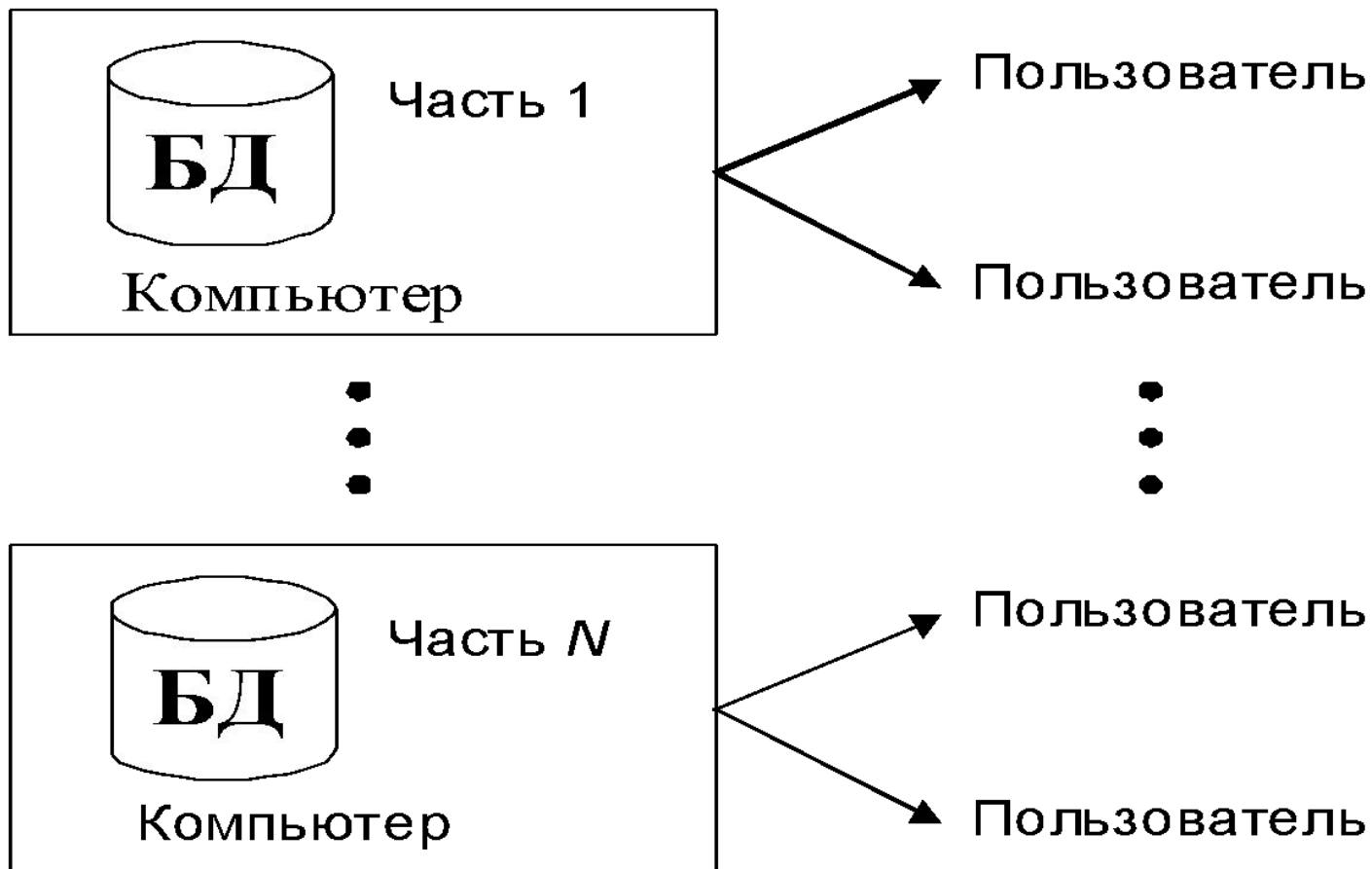


## □ Сетевая централизованная база данных



# КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

## ▣ Сетевая распределённая база данных



# КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

- БД первого поколения
  - 📧 иерархические;
  - 📧 сетевые;
- БД второго поколения
  - реляционные
- БД третьего поколения
  - 📧 объектно-ориентированные;
  - 📧 объектно-реляционные.





# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

## □ Описание предметной области

**Предметная область** – совокупность объектов, их свойств и взаимосвязей, которые представляют определённую ценность для пользователя и должны фиксироваться в БД.

**Объект** – это всё то, что существует вне нас и независимо от нашего сознания, влияния внешнего мира и материальной действительности.

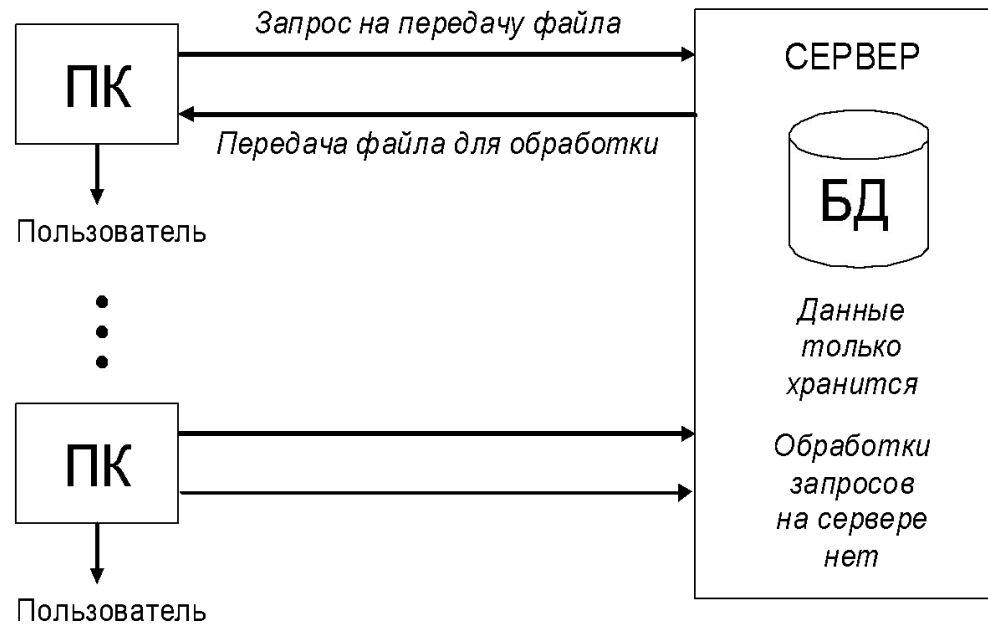
**Предмет** – это объект, ставший носителем определённой совокупности свойств и входящий в различные взаимоотношения, которые представляют интерес для потребителей.

Совокупность объектов, информация о которых представляет интерес для пользователей, образует **объектное ядро предметной области**.



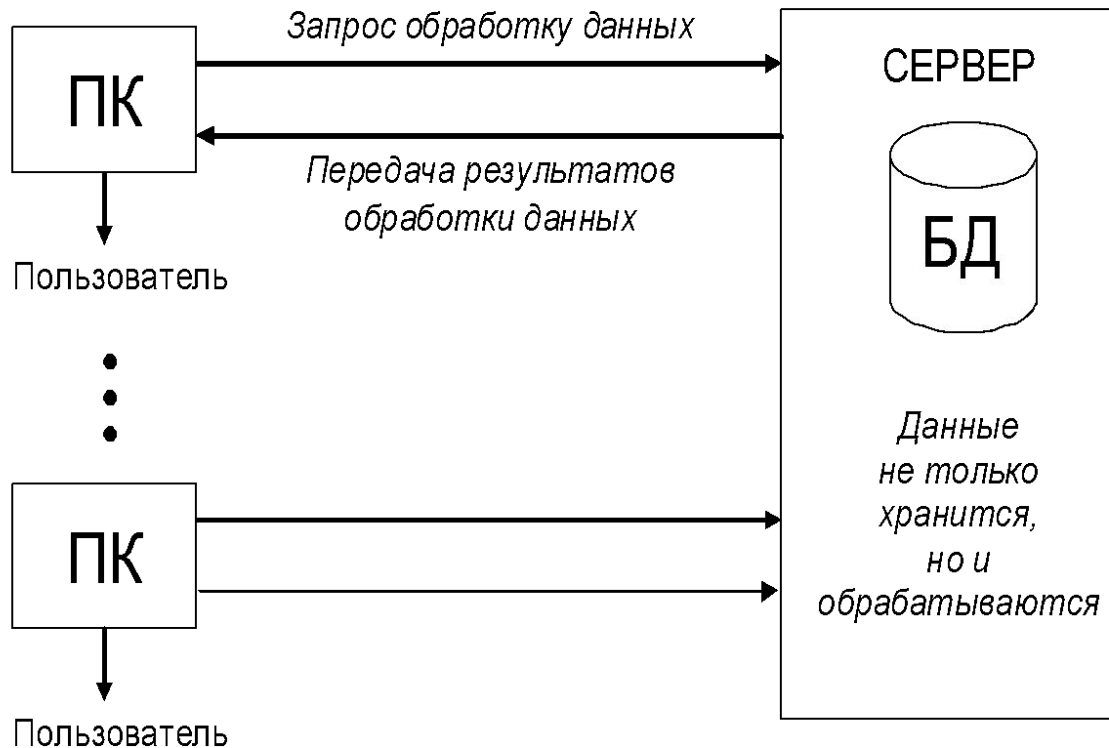
# АРХИТЕКТУРА ФАЙЛ-СЕРВЕР

- ❑ Недостатки архитектуры «файл-сервер»:
- ❑ высокая загрузка сети, так как обмен данными идет на уровне файлов;
- ❑ низкий уровень защиты данных, так как доступ к файлам базы данных осуществляется общими средствами операционной системы сервера;
- ❑ низкий уровень управления целостностью и непротиворечивостью данных, так как правила функциональной обработки, сосредоточенные на клиентской части, могут быть противоречивыми и несогласованными.



Программа управления данными, которая выполняется на машине-клиенте, должна сначала осуществить запрос каждой записи базы данных, после чего она может определить, удовлетворяет ли запись поисковым условиям, и только после этого передать запись для обработки. Для архитектуры «файл-сервер» характерно большое суммарное время обработки данных.

# АРХИТЕКТУРА КЛИЕНТ-СЕРВЕР



В архитектуре «клиент-сервер» на рабочей станции работает клиентское приложение, которое реализует интерфейс с пользователем и формирует запросы к базе данных на обработку данных, но сами запросы выполняет СУБД на сервере. Клиентское приложение создается либо в среде СУБД рабочей станции, либо без использования СУБД с помощью языков программирования.

# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

- Концептуальное проектирование (инфологическое моделирование предметной области)

**Тип** – это понятие, объединяющее все объекты данного типа.

Каждый тип имеет уникальное имя. Например, ПРЕПОДАВАТЕЛЬ, СТУДЕНТ, ЛАБОРАНТ и т.д.

**Концептуальная модель (схема предметной области)**

– множество типов данной предметной области, снабженное некоторой структурой.



# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

- Модель сущность – атрибут – связь (ER)

**Сущность** – это реальный или воображаемый объект, информация о котором представляет интерес.

**Связь** – это графически изображаемая ассоциация, устанавливаемая между двумя сущностями



# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

- Модель сущность – атрибут-связь (ER)

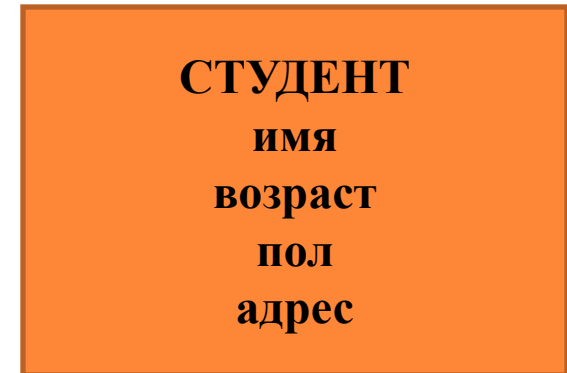
**Атрибутом** – является любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности

## **Типы связей:**

«один-к-одному» 1:1

«один-ко-многим» 1:M

«многие-ко-многим» M:M



# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

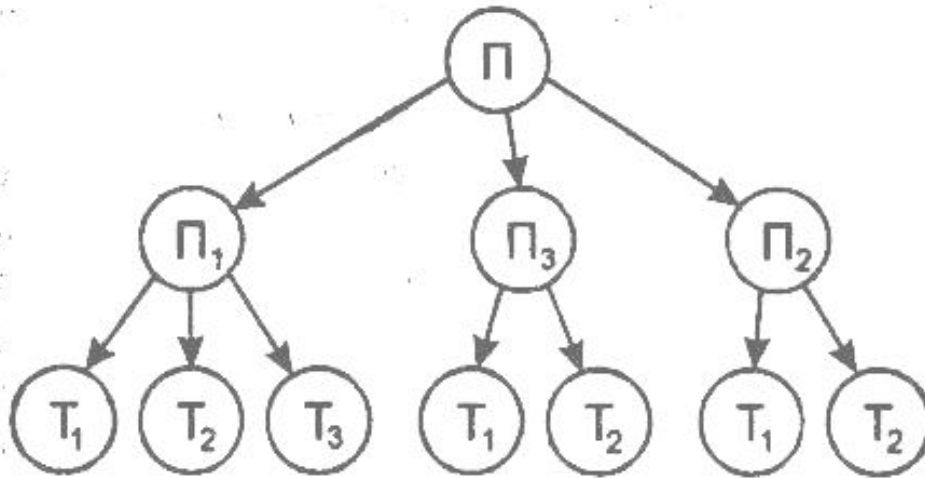
## □ Модели данных



# ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Типичным представителем (наиболее известным и распространенным) является СУБД IMS (Information Management System) компании IBM. Первая версия системы появилась в 1968 г.

*Иерархическая модель* представляется в виде древовидного графа, в котором объекты выделяются по уровням соподчиненности (иерархии) объектов



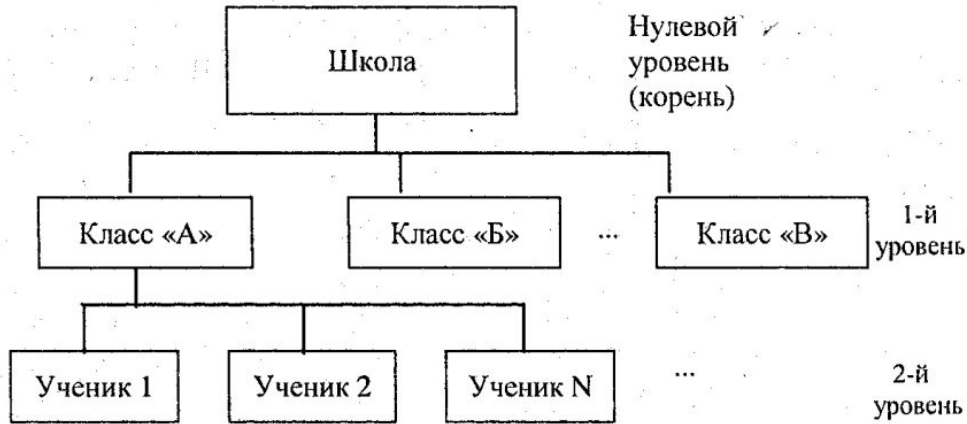
В основании дерева – один элемент (корень), связанный с несколькими элементами (стволами), которые связаны еще с несколькими (ветками) и т.д. до последних элементов (листьев).

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





# ДАННЫХ



## Манипулирование данными

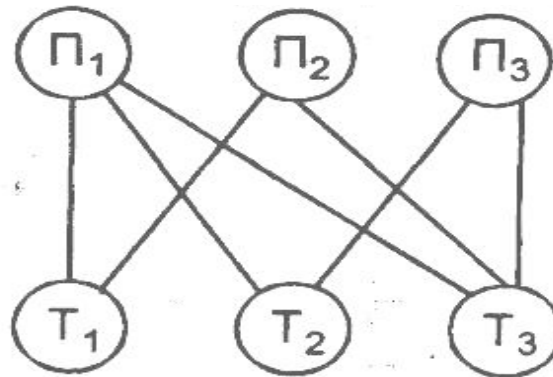
- найти указанный экземпляр типа дерева БД (например, отдел 310);
- перейти от одного экземпляра типа дерева к другому;
- перейти от экземпляра одного типа записи к экземпляру другого типа записи внутри дерева (например, перейти от отдела к первому сотруднику);
- перейти от одной записи к другой в порядке обхода иерархии;
- вставить новую запись в указанную позицию;
- удалить текущую запись.



# СЕТЕВАЯ МОДЕЛЬ БАЗЫ ДАННЫХ

Типичным представителем систем, основанных на сетевой модели данных, является СУБД IDMS (Integrated Database Management System), разработанная компанией Cullinet Software, Inc.

*Сетевая модель* базы данных для поставленной задачи представлена в виде диаграммы связей. Сетевая БД состоит из набора записей и набора связей между этими записями

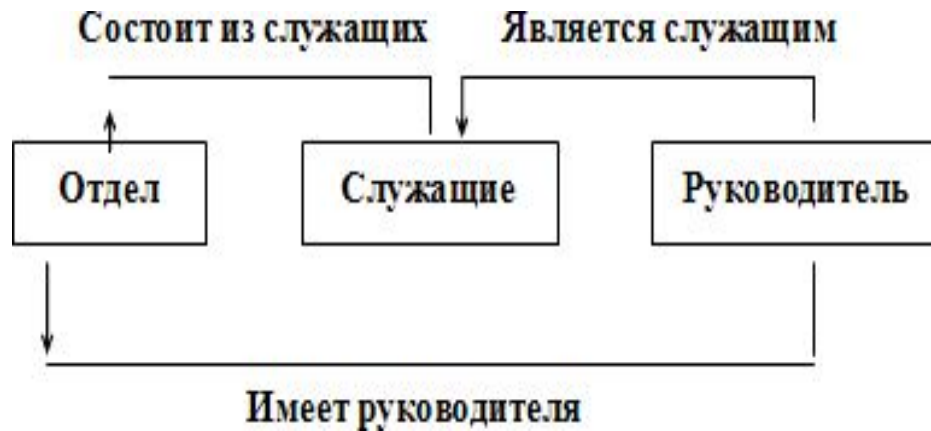
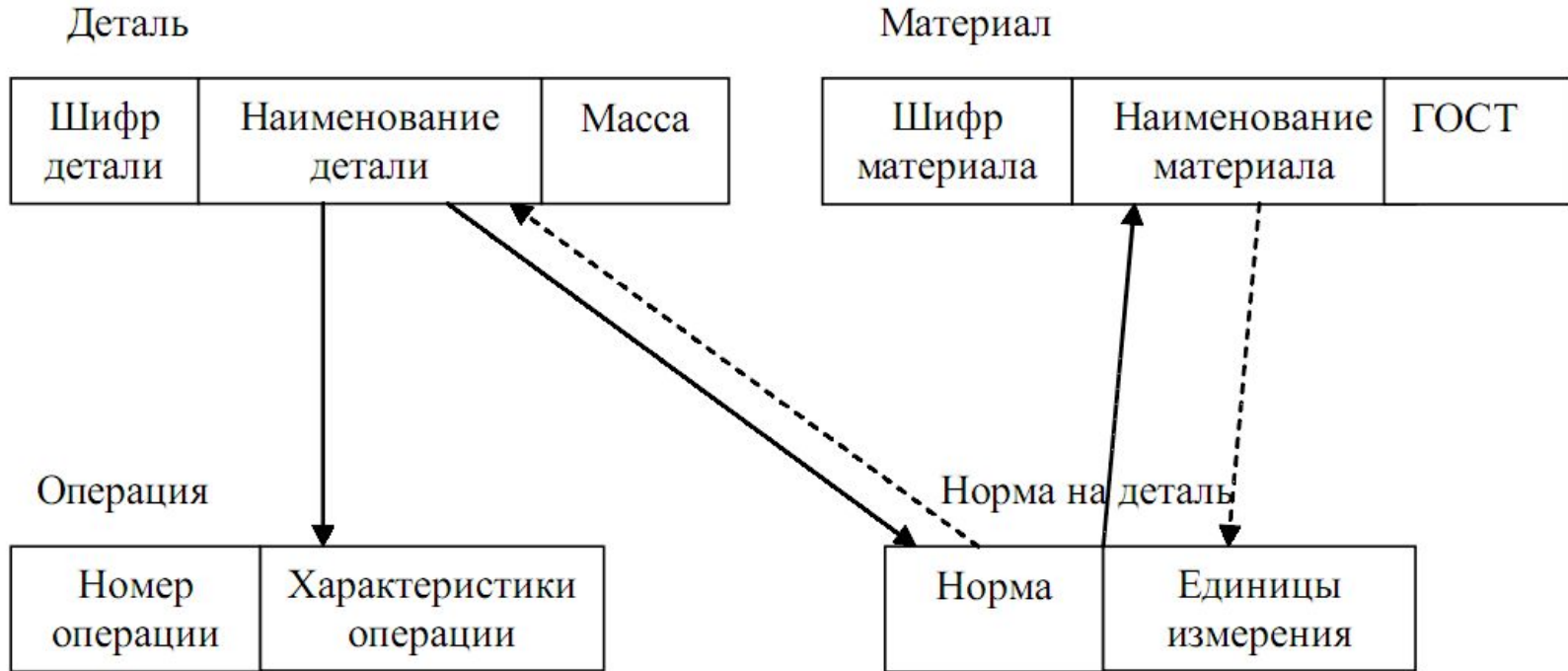


Тип связи определяется для двух типов записи: предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка.

В сетевой модели допустимы любые виды связей между записями и отсутствует ограничение на число обратных связей. Но должно соблюдаться одно правило: связь включает основную и зависимую записи



# ПРИМЕР СЕТЕВОЙ БАЗЫ ДАННЫХ



# МАНИПУЛИРОВАНИЕ ДАННЫМИ В СЕТЕВОЙ БД

- найти конкретную запись в наборе однотипных записей (например, служащего с именем Иванов);
- перейти от предка к первому потомку по некоторой связи (например, к первому служащему отдела 625);
- перейти к следующему потомку в некоторой связи (например, от Иванова к Сидорову);
- перейти от потомка к предку по некоторой связи (например, найти отдел, в котором работает Сидоров);
- создать новую запись;
- уничтожить запись;
- модифицировать запись;
- включить в связь;
- исключить из связи;
- переставить в другую связь и т.д.



# РЕЛЯЦИОННАЯ МОДЕЛЬ

К отечественным СУБД реляционного типа относятся системы: ПАЛЬМА (ИКАН УССР ), а также система НуTech ( МИФИ ).

R1 (ПОСТАВЩИКИ)

Номер поставщика
П1
П2
П3

R2 (ТОВАРЫ)

Номер товара
Т1
Т2
Т3

R3 (ПОСТАВКАТОВАРОВ)

Номер поставщика	Номер детали
П1	Т1
П1	Т2
П1	Т3
П2	Т1
П2	Т3
П3	Т2
П3	Т3

В реляционной модели базы данных взаимосвязи между элементами данных представляются в виде двумерных таблиц, называемых отношениями.

**Отношения обладают следующими свойствами:**

- ✓ каждый элемент таблицы представляет собой один элемент данных (повторяющиеся группы отсутствуют);
- ✓ элементы столбца имеют одинаковую природу, и столбцам однозначно присвоены имена;
- ✓ в таблице нет двух одинаковых строк;
- ✓ строки и столбцы могут просматриваться в любом порядке вне зависимости от их информационного содержания.

# ПРИМЕР РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

R1 (ПОСТАВЩИКИ)

Номер поставщика	Фамилия	Рейтинг	Город
П1	Иванов	20	Москва
П2	Петров	10	Курск
П3	Сидоров	30	Краснодар

R2 (ТОВАРЫ)

Номер товара	Название	Масса	Цвет
T1	Гайка	12	Красный
T2	Болт	17	Зеленый
T3	Шайба	5	Голубой

R3 (ПОСТАВКА ТОВАРОВ)

Номер поставщика	Номер детали	Количество
П1	T1	300
П1	T2	200
П1	T3	400
П2	T1	300
П2	T3	400
П3	T2	200
П3	T3	300

Операции реляционной модели данных дают возможность произвольно манипулировать отношениями, позволяя обновлять БД, а также выбирать подмножества хранимых данных и представлять их в нужном виде.



# ДОСТОИНСТВА И НЕДОСТАТКИ ИЕРАРХИЧЕСКОЙ, СЕТЕВОЙ И РЕЛЯЦИОННОЙ МОДЕЛЕЙ:

Модель БД	достоинства	недостатки
иерархическая	<ul style="list-style-type: none"><li>□ позволяет описать структуру данных как на логическом, так и на физическом уровне</li></ul>	<ul style="list-style-type: none"><li>□ жесткая фиксированность взаимосвязей между элементами данных, вследствие чего любые изменения связей требуют изменения структуры,</li><li>□ жесткая зависимость физической и логической организации данных</li></ul>
сетевая	<ul style="list-style-type: none"><li>□ большая информационная гибкость по сравнению с иерархической моделью</li></ul>	<ul style="list-style-type: none"><li>□ достаточно жесткая структура, что препятствует развитию информационной базы системы управления.</li></ul>
реляционная	<ul style="list-style-type: none"><li>□ простота логической модели (таблицы привычны для представления информации);</li><li>□ гибкость системы защиты (для каждого отношения может быть задана правомерность доступа);</li><li>□ Независимость данных;</li><li>□ возможность построения простого языка манипулирования данными с помощью математически строгой теории реляционной алгебры (алгебры отношений).</li><li>□ наибольшее распространение и перспективность в современных информационных технологиях.</li></ul>	<ul style="list-style-type: none"><li>□ отсутствие стандартных средств идентификации отдельных записей</li><li>□ сложность описания иерархических и сетевых связей.</li></ul>



# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

## □ Логическое проектирование

**Логическое проектирование БД** – это процесс конструирования общей информационной модели на основе отдельных моделей данных пользователей, которая является независимой от особенностей реально используемой СУБД и других физических условий.

На данном этапе выполняют следующие действия:

1. Удаление связей типа М:М;
2. Удаление сложных связей;
3. Удаление рекурсивных связей;
4. Удаление связей с атрибутами;
5. Удаление множественных атрибутов;
6. Перепроверка связей типа 1:1;
7. Удаление избыточных связей.





# МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ

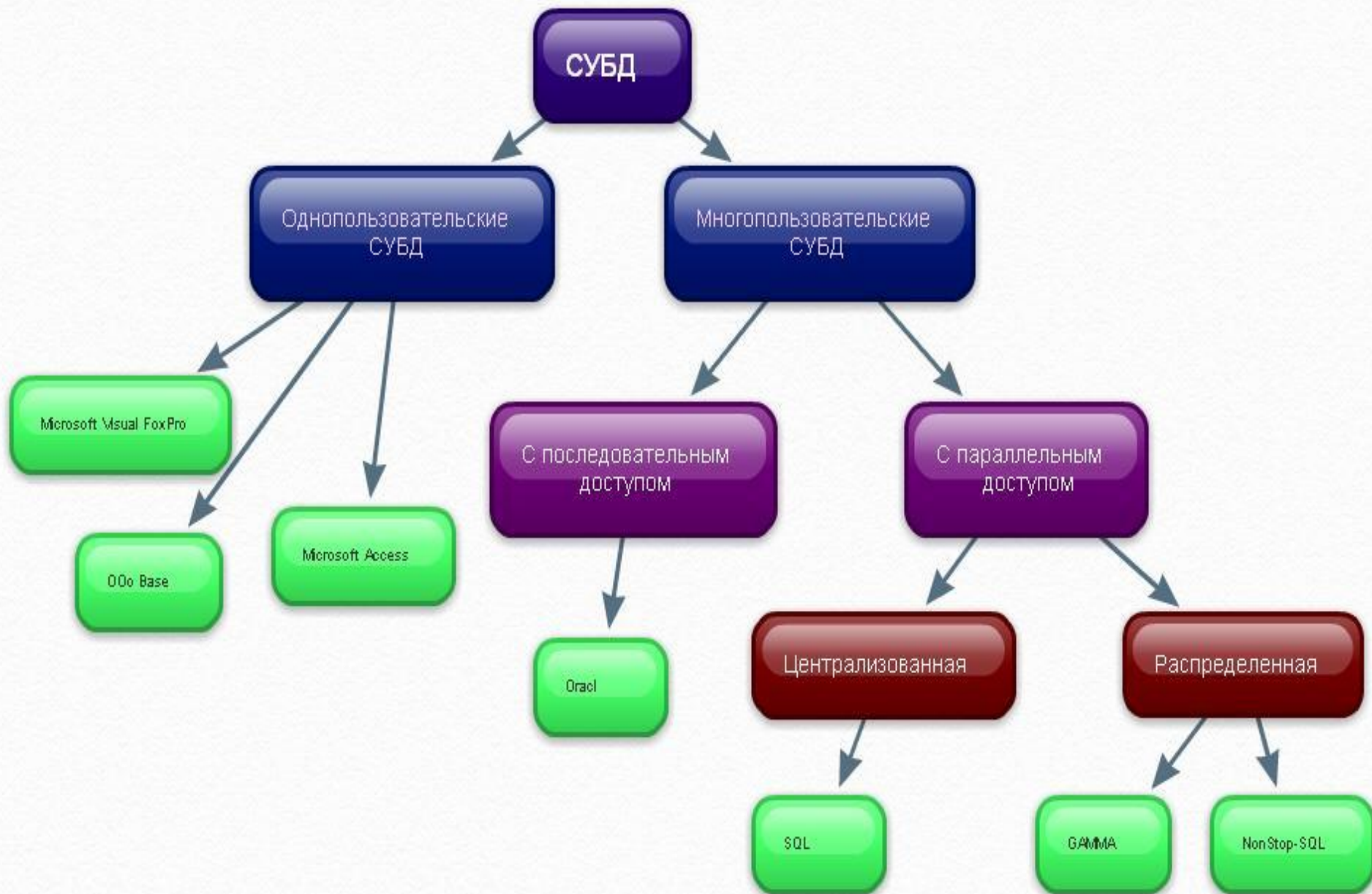
## ▣ Физическое проектирование

Этапы физического проектирования:

1. Разработка таблиц БД и установка необходимых ограничений целостности данных.
2. Выбор схемы хранения данных и определение методов доступа к таблицам БД.
3. Проектирование системы защиты БД от несанкционированного доступа.
4. Организация процессов мониторинга созданной системы, задача которого состоит в выявлении и устранении любых проблем, связанных с производительностью приложений и вытекающих из особенностей реализации проекта. Здесь же осуществляется реализация новых и изменяющихся требований.



# КЛАССИФИКАЦИЯ СУБД



# В ЦЕЛЕМ ТАКИЕ СИСТЕМЫ МОЖНО ОХАРАКТЕРИЗОВАТЬ СЛЕДУЮЩИМ ОБРАЗОМ:

Эти системы активно использовались в течение многих лет, задолго до появления работоспособных реляционных СУБД. Одной из актуальных проблем информационных систем является использование этих систем совместно с современными.

Все ранние системы не основывались на каких-либо абстрактных моделях. Абстрактные представления ранних систем появились позже на основе анализа и выявления общих признаков у различных конкретных систем.

В ранних системах доступ к БД производится на уровне записей. Интерактивный доступ к БД поддерживается только путем создания соответствующих прикладных программ с собственным интерфейсом.

Навигационная природа ранних систем и доступ к данным на уровне записей заставляли пользователей самих производить всю оптимизацию доступа к БД, без какой-либо поддержки системы.

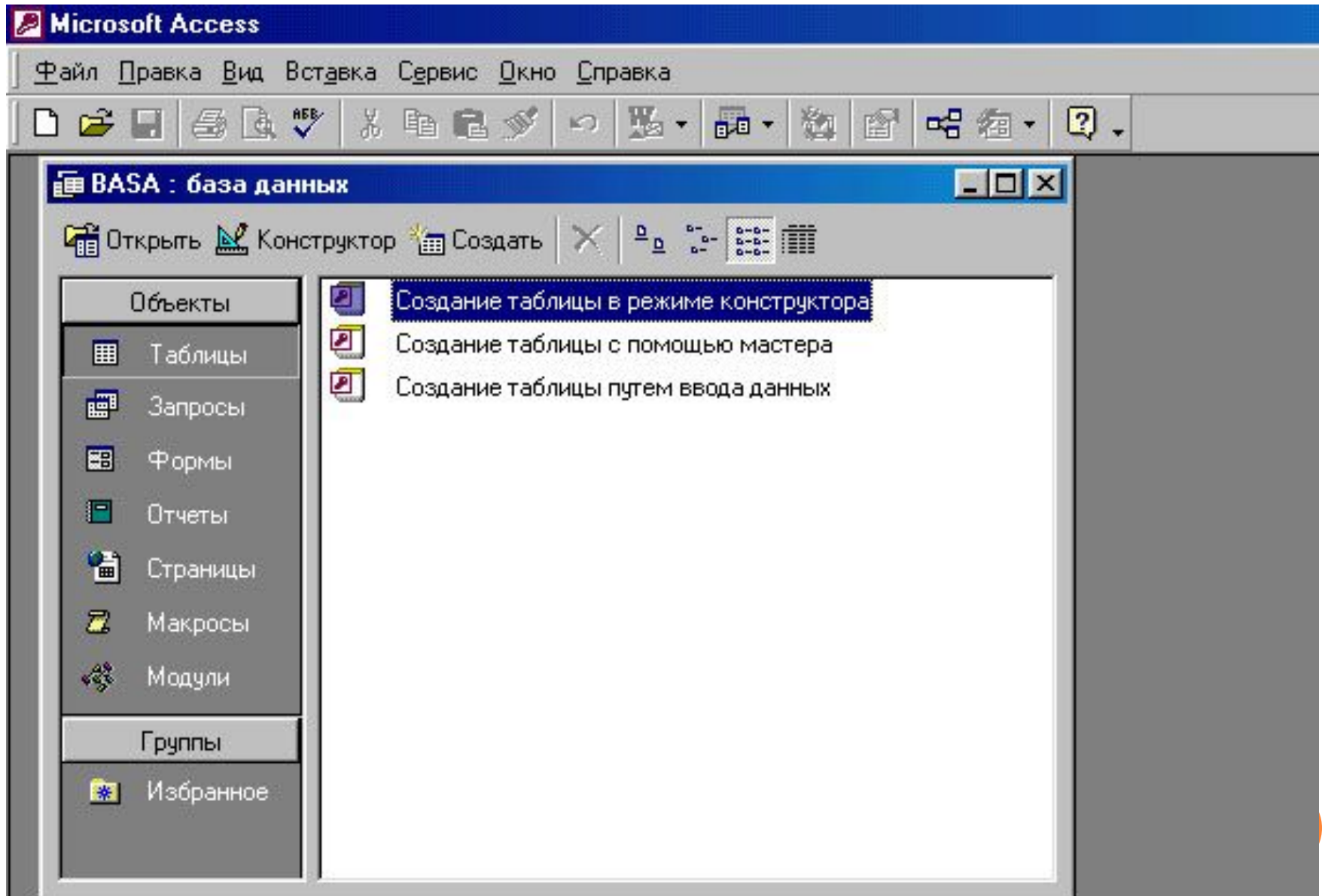
После появления реляционных систем большинство ранних систем было оснащено «реляционными» интерфейсами. Однако в большинстве случаев это не сделало их по-настоящему реляционными системами, поскольку оставалась возможность манипулировать данными в естественном для них режиме.

### 3. МОДЕЛЬ ДАННЫХ СУБД MS ACCESS

*Microsoft Access является одной из популярных систем проектирования и сопровождения БД, она представляет собой полнофункциональную СУБД, в которую входят таблицы данных, экранные формы для ввода данных в эти таблицы, запросы и отчеты для получения новой информации по данным из таблиц, макросы и модули для дополнительного программирования.*



# МОДЕЛЬ СУБД MS ACCESS

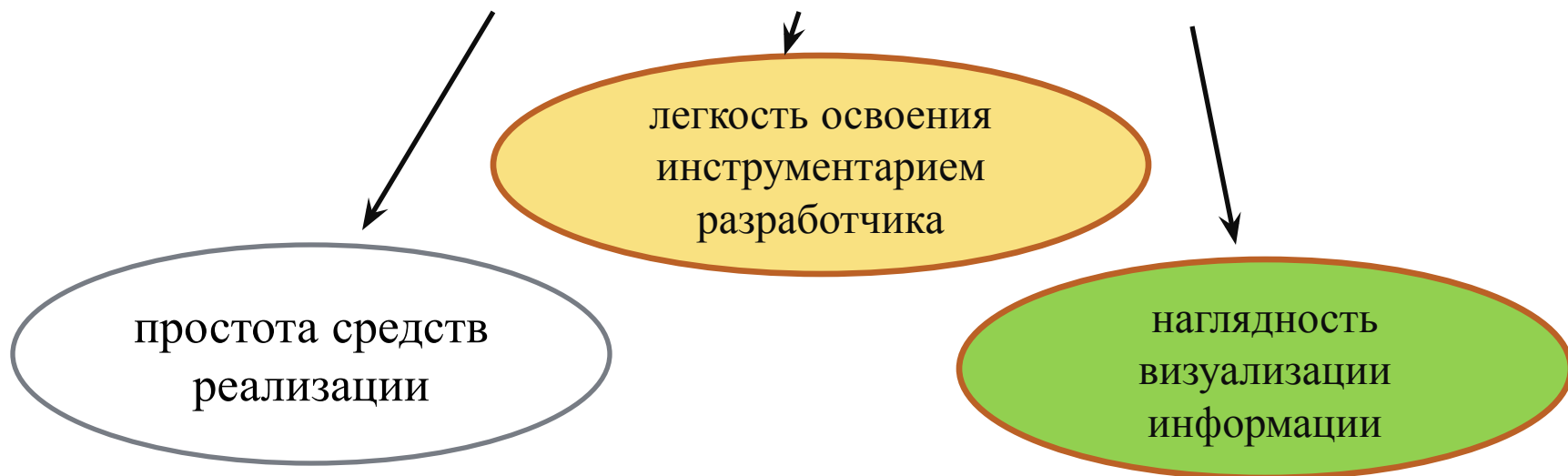




## 4. ПРЕДНАЗНАЧЕНИЕ СУБД ACCESS

- ❑ СУБД Access предназначена для разработки баз данных реляционного типа для локального их использования на персональных компьютерах и для работы с этими базами.
- ❑ При проектировании базы данных, в первую очередь, необходимо определить, что именно нужно хранить.

**СУБД была выбрана по следующим причинам:**



## СХЕМА КАК СТРУКТУРА БАЗЫ ДАННЫХ

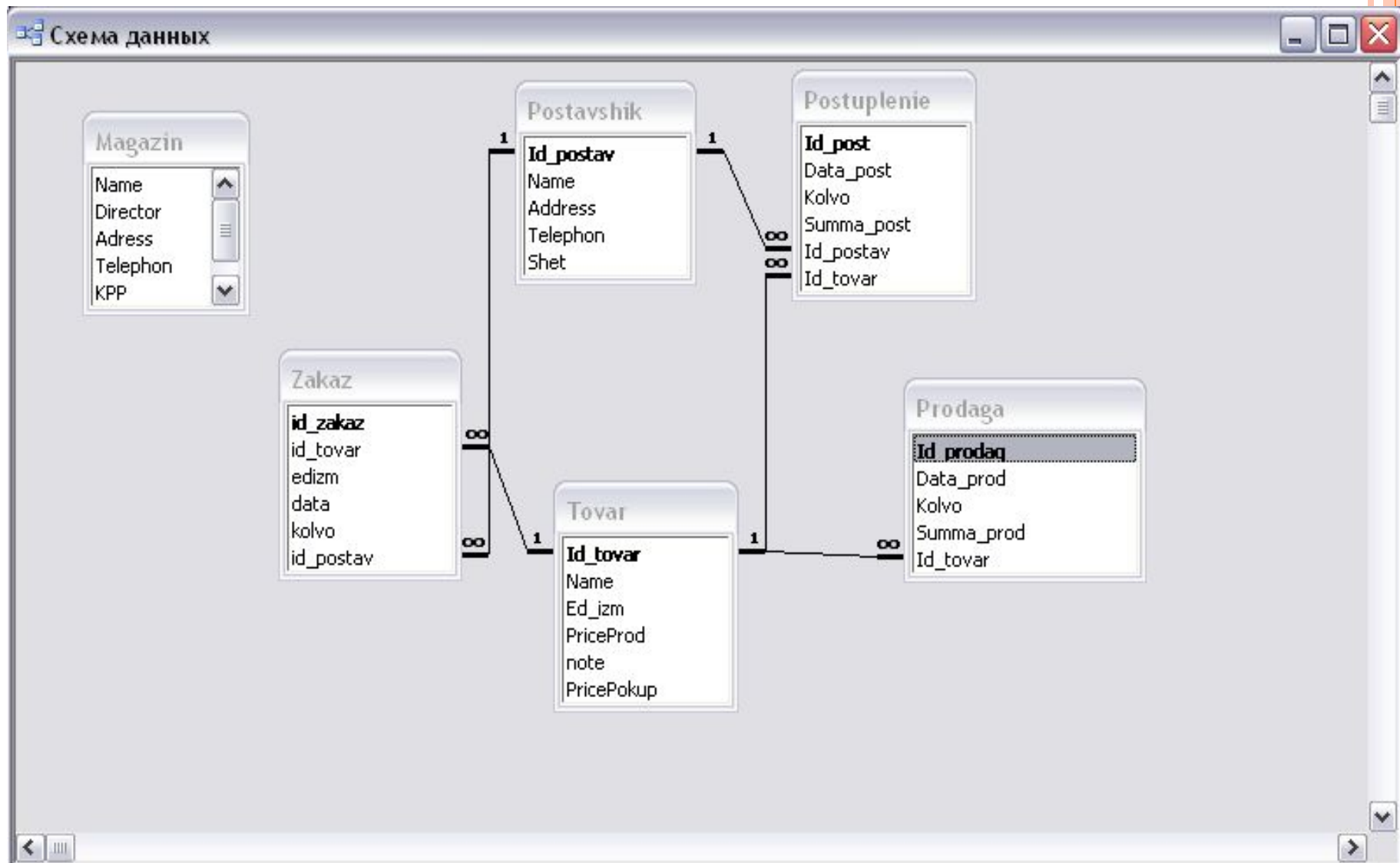
Схема СУБД (от англ. *Database scheme*) - ее структура, описанная на формальном языке, поддерживаемом СУБД. В *реляционных базах данных* схема определяет *таблицы, поля* в каждой таблице, а также отношения между полями и таблицами.

Схемы в общем случае хранятся в *словаре данных*. Хотя схема определена на языке базы данных в виде текста, термин часто используется для обозначения графического представления структуры базы данных.

*Основными объектами схемы являются таблицы и связи.*



# Пример схемы СУБД.





# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ:

1. А.Д. Хоменко «Основы современных компьютерных технологий».
2. *Информатика. Базовый курс* / Под ред. С.В. Симоновича.— СПб.: Питер, 2001.
3. *Бакаревич Ю.Б., Пушкина Н.В.* Самоучитель Microsoft Access 2002. – СПб.: БХВ-Петербург, 2002
4. <http://www.oraclub.ru>
5. <http://studentbank.ru>

