

# БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

1. Информационные системы
2. Базы данных (БД)
3. Реляционные БД

# **БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

## **Тема 1. Информационные системы**

# Определения

---

**База данных (БД)** – это хранилище данных о некоторой предметной области, организованное в виде специальной структуры.

**Важно:**

- данные о некоторой области (не обо всем)
- упорядоченные

**Система управления базой данных (СУБД)** – это программное обеспечение для работы с БД.

**Функции:**

- поиск информации в БД
- выполнение несложных расчетов
- вывод отчетов на печать
- редактирование БД



**Информационная система = БД + СУБД!**

- **локальные ИС**

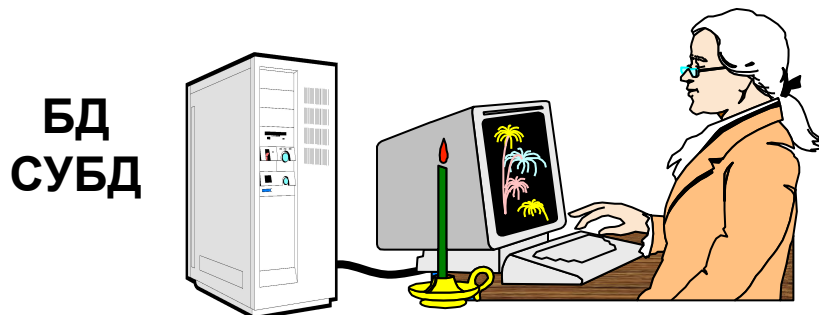
БД и СУБД находятся на одном компьютере.

- **файл-серверные**

БД находится на сервере сети (файловом сервере), а СУБД на компьютере пользователя.

- **клиент-серверные**

БД и основная СУБД находятся на сервере, СУБД на рабочей станции посылает запрос и выводит на экран результат.



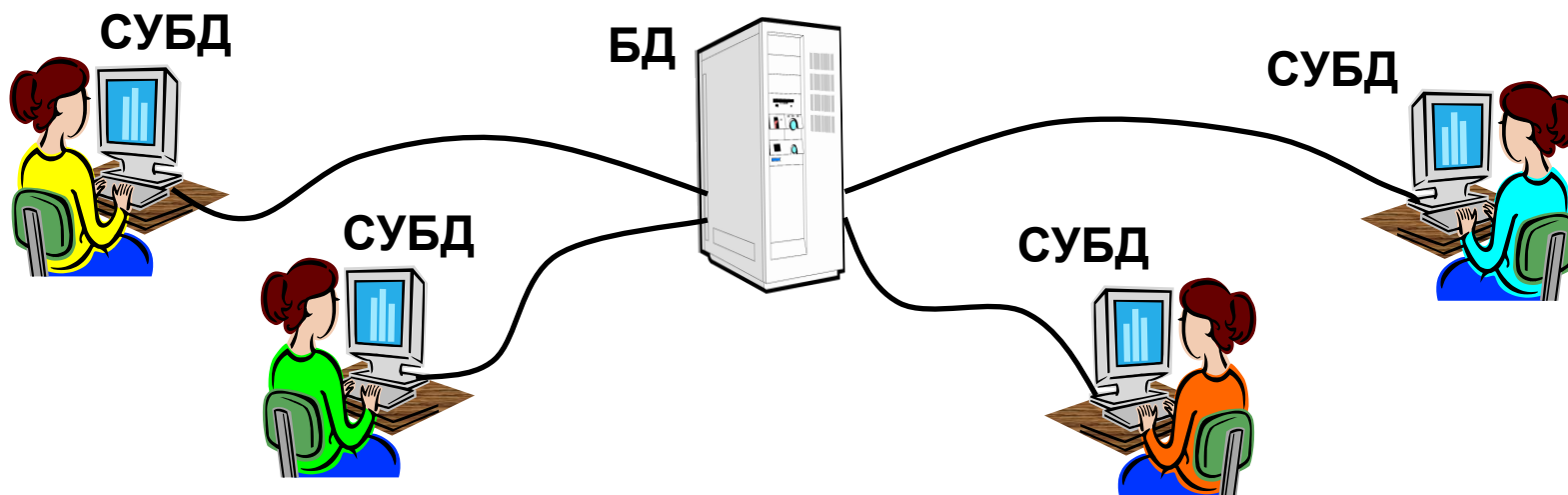
БД  
СУБД



- автономность  
(независимость)



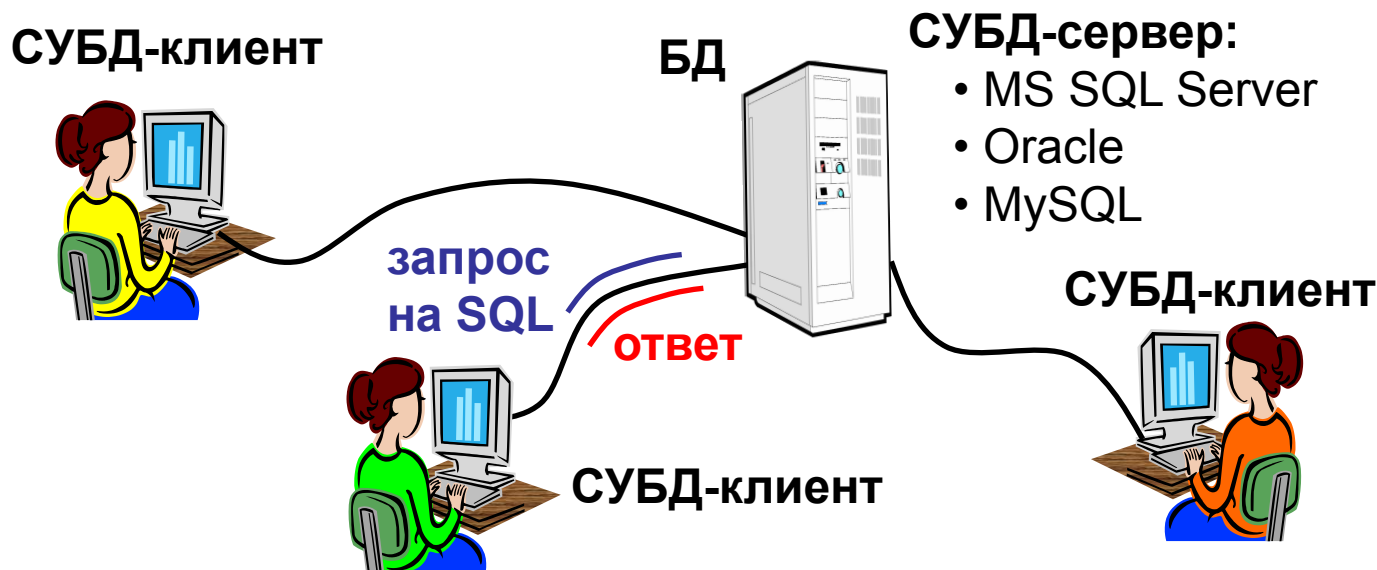
- с БД работает только один человек
- сложно обновлять при большом количестве пользователей
- практически невозможно «стыковать» изменения, вносимые несколькими пользователями



- несколько человек работают с одной базой



- основную работу выполняют рабочие станции (РС), они должны быть мощными
- для поиска строки на РС копируется вся БД – нагрузка на сеть
- слабая защита от взлома (только на РС)
- проблемы при одновременном изменении с разных РС



**SQL (*Structured Query Language*)** – язык структурных запросов



- основную работу выполняет сервер
- проще модернизация (только сервер)
- по сети идут только нужные данные
- защита на сервере (сложнее взломать)
- разделение доступа (очередь заданий)



- сложность настройки
- высокая стоимость ПО (тысячи \$)

# **БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

## **Тема 2. Базы данных**



- **табличные БД**

данные в виде одной таблицы

- **сетевые БД**

набор узлов, в котором каждый может быть связан с каждым.

- **иерархические БД**

в виде многоуровневой структуры

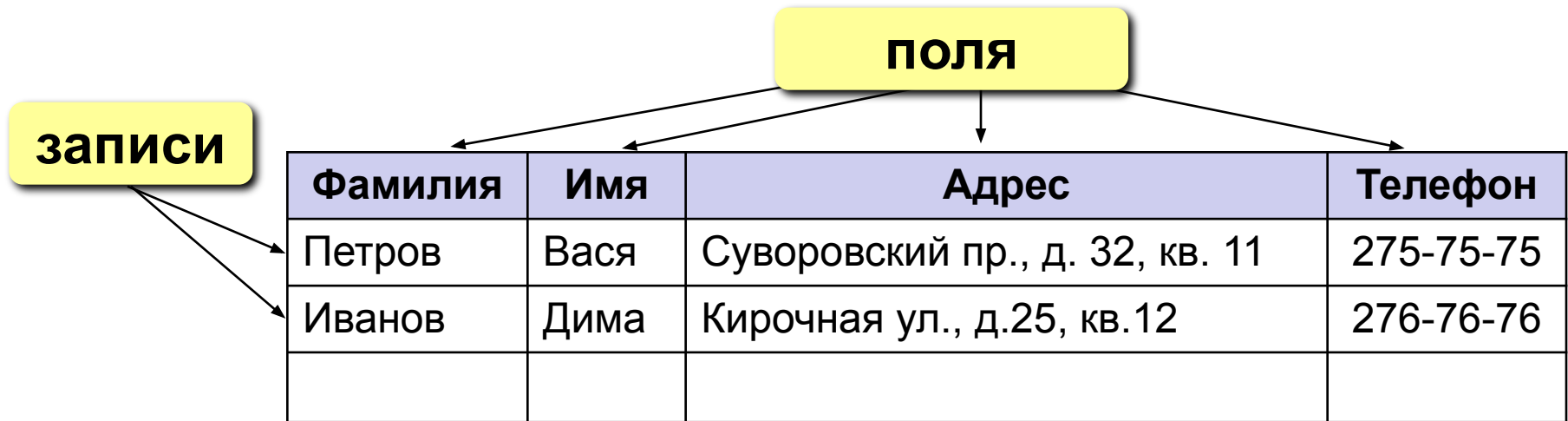
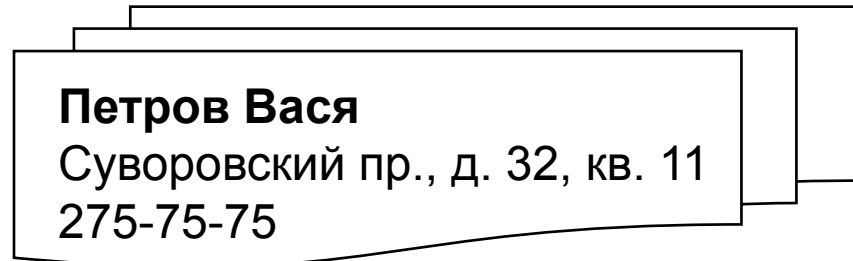
- **реляционные БД (99,9%)**

набор взаимосвязанных таблиц

**Модель** – картотека

**Примеры:**

- записная книжка
- каталог в библиотеке



- 1) самая простая структура
- 2) все другие типы БД используют таблицы



во многих случаях – дублирование данных:

А.С. Пушкин	Сказка о царе Салтане	20 стр.
А.С. Пушкин	Сказка о золотом петушке	12 стр.

- 1. Количество полей определяется разработчиком и не может изменяться пользователем.**
- 2. Любое поле должно иметь уникальное имя.**
- 3. Поля могут иметь различный тип:**
  - строка символов (длиной до 255 символов)
  - вещественное число (с дробной частью)
  - целое число
  - денежная сумма
  - дата, время, дата и время
  - логическое поле (истина или ложь, да или нет)
  - многострочный текст (МЕМО)
  - рисунок, звук или другой объект (объект OLE)
- 4. Поля могут быть обязательными для заполнения или нет.**
- 5. Таблица может содержать сколько угодно записей (это количество ограничено только объемом диска); записи можно добавлять, удалять, редактировать, сортировать, искать.**

# Ключевое поле (ключ таблицы)

---

**Ключевое поле (ключ)** – это поле (или комбинация полей), которое однозначно определяет запись.

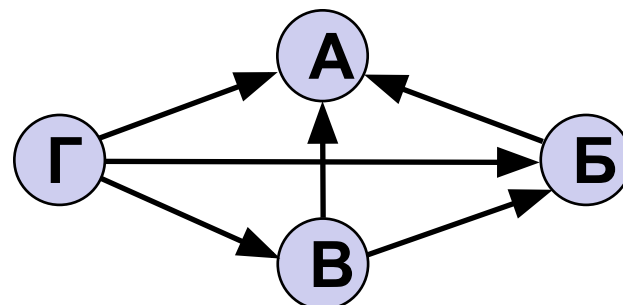
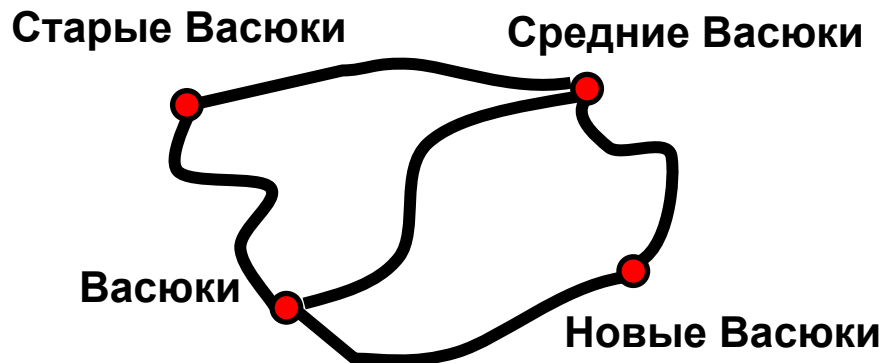
В таблице не может быть двух записей с одинаковым значением ключа.

## Могут ли эти данные быть ключом?

- ~~фамилия~~
- ~~имя~~
- номер паспорта
- ~~номер дома~~
- регистрационный номер автомобиля
- ~~город проживания~~
- ~~дата выполнения работы~~
- марка стиральной машины



**Сетевая БД** – это набор узлов, в которых каждый может быть связан с каждым (схема дорог).



- лучше всего отражает структуру некоторых задач (сетевое планирование в экономике)



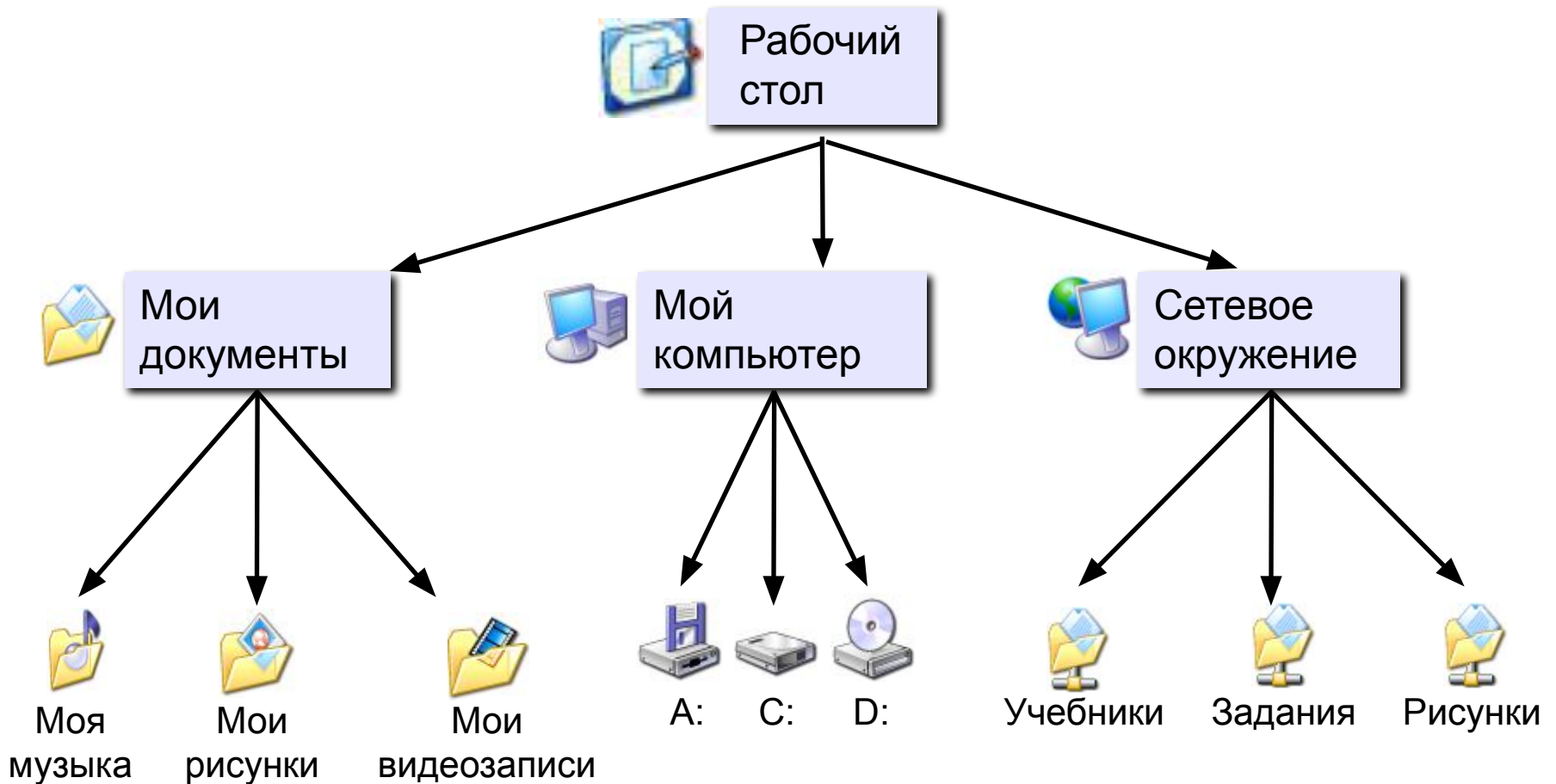
- сложно хранить информацию о всех связях
- запутанность структуры



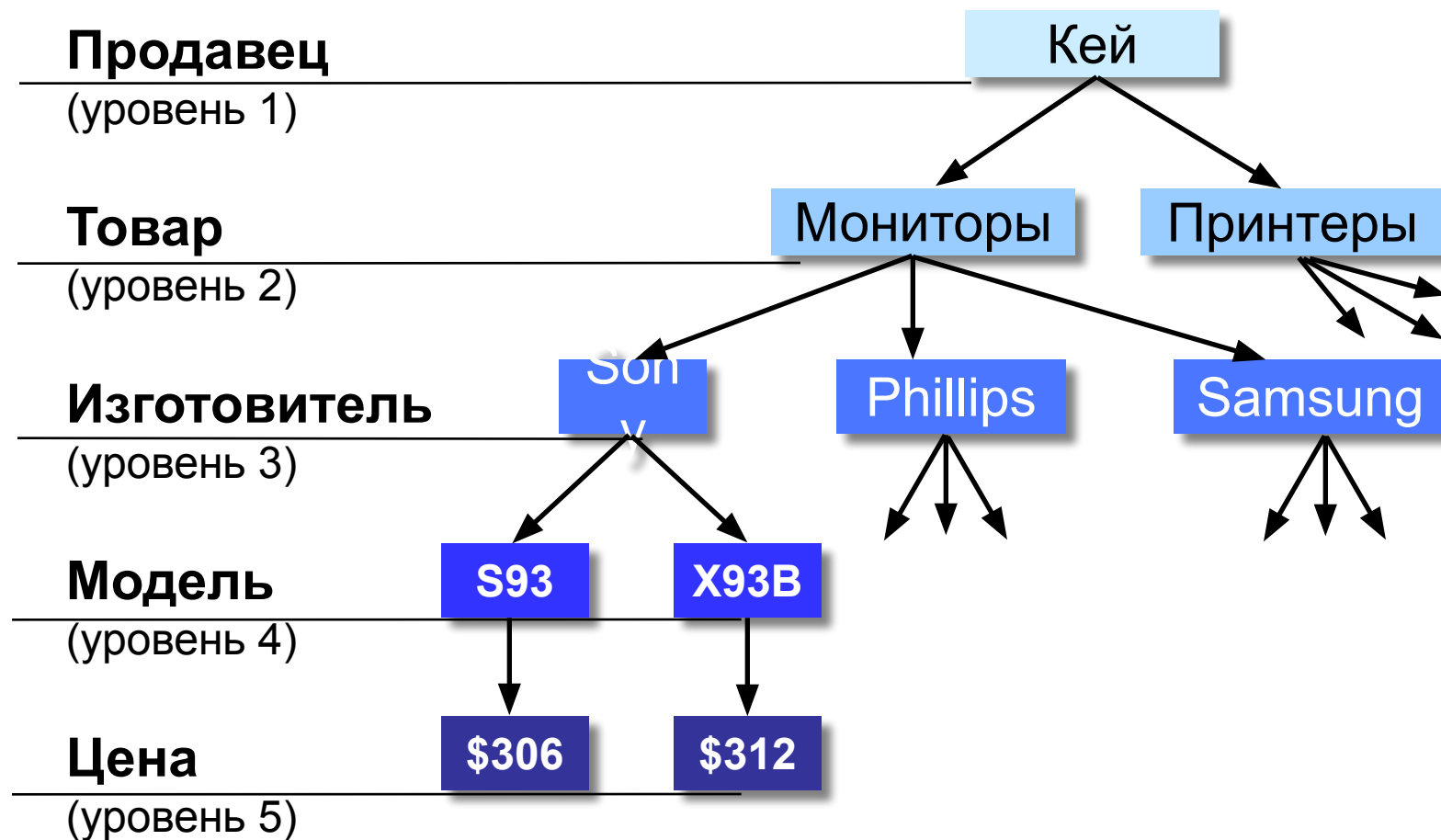
**Можно хранить в виде таблицы, но с дублированием данных!**

# Иерархические БД

**Иерархическая БД** – это набор данных в виде многоуровневой структуры (дерева).




## Прайс-лист:



# Иерархические БД

## Приведение к табличной форме:

Продавец	Товар	Изготовитель	Модель	Цена
Кей	Монитор	Sony	S93	\$306
Кей	Монитор	Sony	X93B	\$312
Key	Монитор	Phillips	190 B5 CG	\$318
Кей	Монитор	Samsung	SyncMaster 193P	\$452
...				

-  дублирование данных
- при изменении адреса фирмы надо менять его во всех строках
- нет защиты от ошибок ввода оператора (*Кей* – *Key*), лучше было бы выбирать из списка



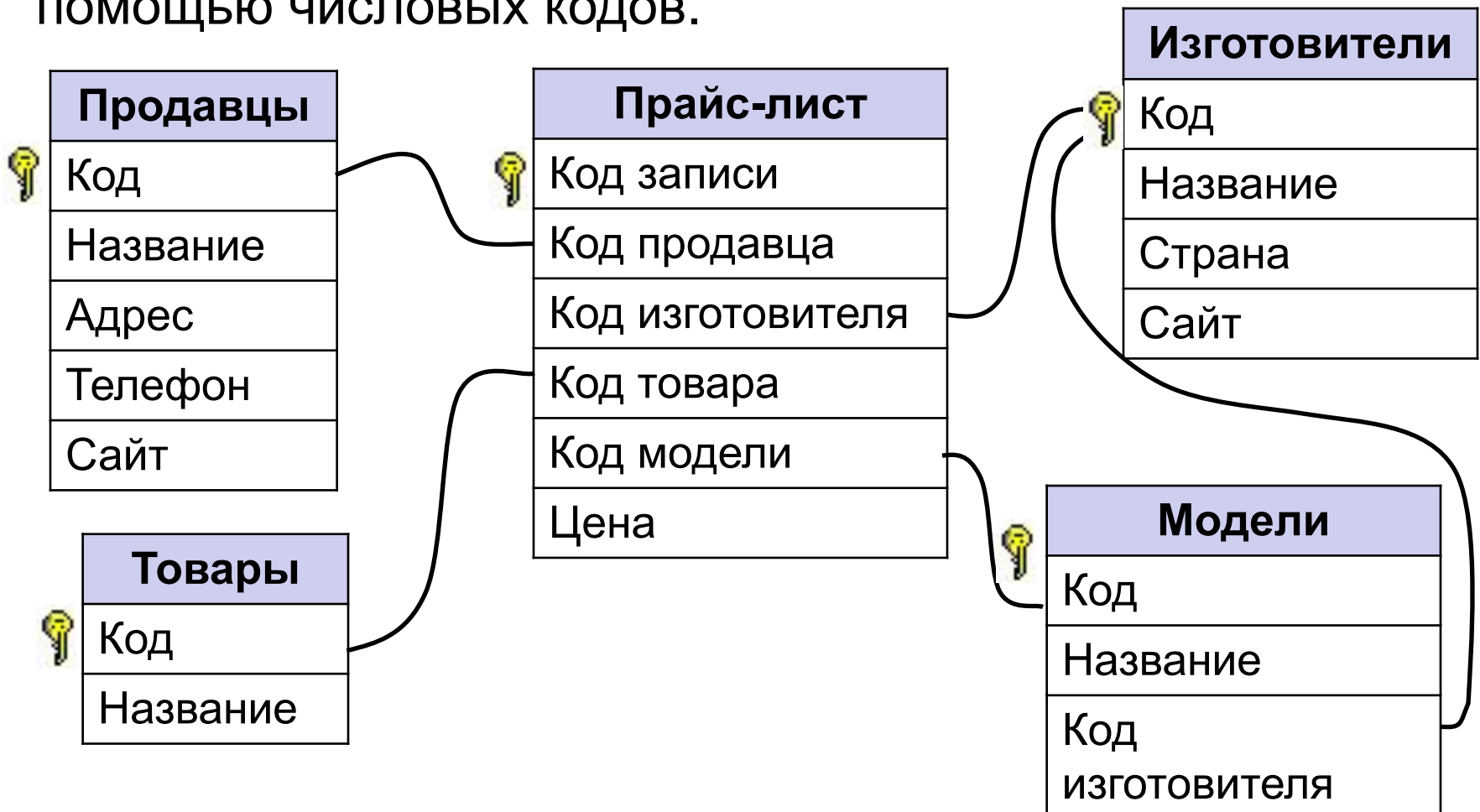
# **БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

## **Тема 3. Реляционные базы данных**

# Реляционные БД



1970-е гг. Э. Кодд, англ. *relation* – отношение.

**Реляционная база данных** – это набор простых таблиц, между которыми установлены связи (отношения) с помощью числовых кодов.



# Реляционные БД

---

- 
  - нет дублирования информации;
  - при изменении адреса фирмы достаточно изменить его только в таблице **Продавцы**;
  - защита от неправильного ввода: можно выбрать только фирму, которая есть в таблице **Продавцы**;
  - механизм **транзакций**: любые изменения вносятся в базу только тогда, когда они полностью завершены.
- 
  - сложность структуры (не более 40-50 таблиц);
  - при поиске надо обращаться к нескольким таблицам;
  - нужно поддерживать **целостность**: при удалении фирмы-продавца надо удалять все связанные записи (автоматически, **каскадное удаление**).

# Связи между таблицами

**Один к одному («1-1»)** – одной записи в первой таблице соответствует ровно одна запись во второй.

Применение: выделение часто используемых данных.

Код	Фамилия	Имя
1	Иванов	Кузьма
2	Петров	Василий
...		

Код	Год рожд.	Адрес
1	1992	Суворовский, д.20, кв. 6
2	1993	Кирочная, д. 30, кв 18
...		

**Один ко многим («1-∞»)** – одной записи в первой таблице соответствует сколько угодно записей во второй.

Код	Название
1	Монитор
2	Винчестер
...	

Код	Код товара	Цена
123	1	10 999
345	1	11 999
...		

# Связи между таблицами

**Многие ко многим (« $\infty$  -  $\infty$ »)** – одной записи в первой таблице соответствует сколько угодно записей во второй, и наоборот.

учителя

Код	Фамилия
1	Иванов
2	Петров
...	

$\infty$   $\infty$

предметы

Код	Название
1	История
2	География
3	Биология
...	

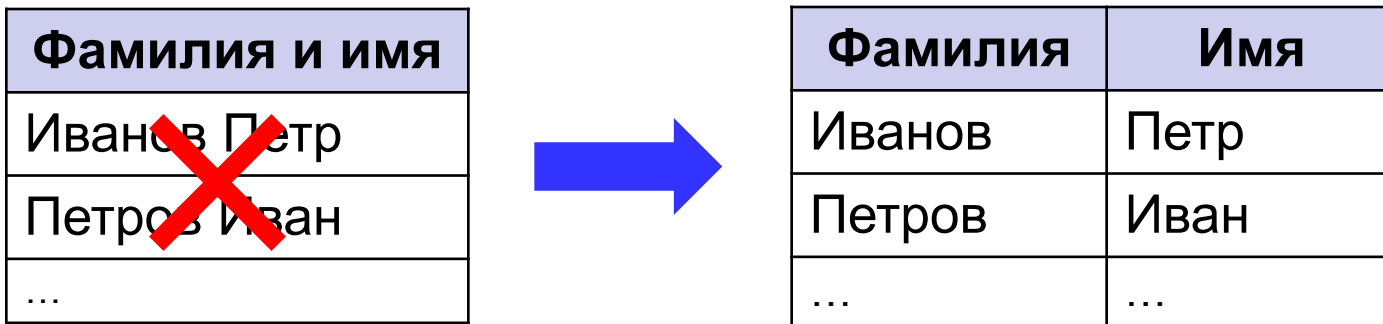
**Реализация** – через третью таблицу и две связи «1- $\infty$ ».



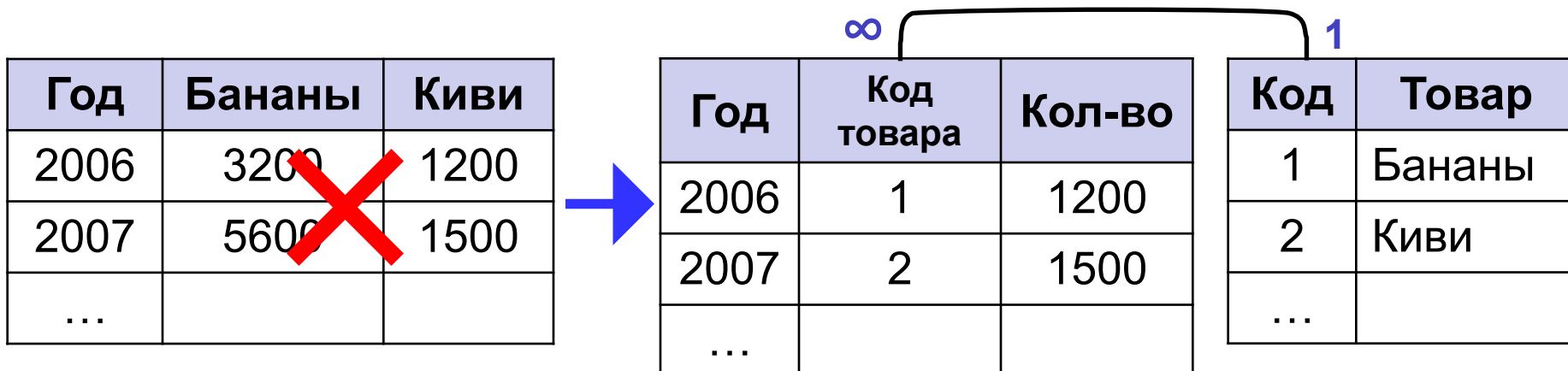
# Нормализация базы данных

**Нормализация** – это разработка такой структуры БД, в которой нет избыточных данных и связей.

- Любое поле должно быть **неделимым**.



- Не должно быть полей, которые обозначают различные виды одного и того же, например, товаров.



# Нормализация базы данных

- Любое поле должно зависеть только от ключа (**ключ** – это поле или комбинация полей, однозначно определяющая запись).

товары

Код	Название	Цена
1	Монитор	<del>9 000 р.</del>
2	Винчестер	<del>11 000 р.</del>
...		

зависит не только от названия товара!




прайс-лист

- Не должно быть полей, которые могут быть найдены с помощью остальных.

Код	Товар	Цена за тонну	Кол-во, тонн	Стоимость
1	Бананы	1200	10	<del>12 000</del>
2	Киви	1500	20	<del>30 000</del>
...				

# Поиск в базах данных

**Линейный поиск** – это перебор всех записей до тех пор, пока не будет найдена нужная.



Код	Фамилия
1	Сидоров
2	Ветров
...	
1024	Померанцев

Иванов?

**1024  
сравнения!**

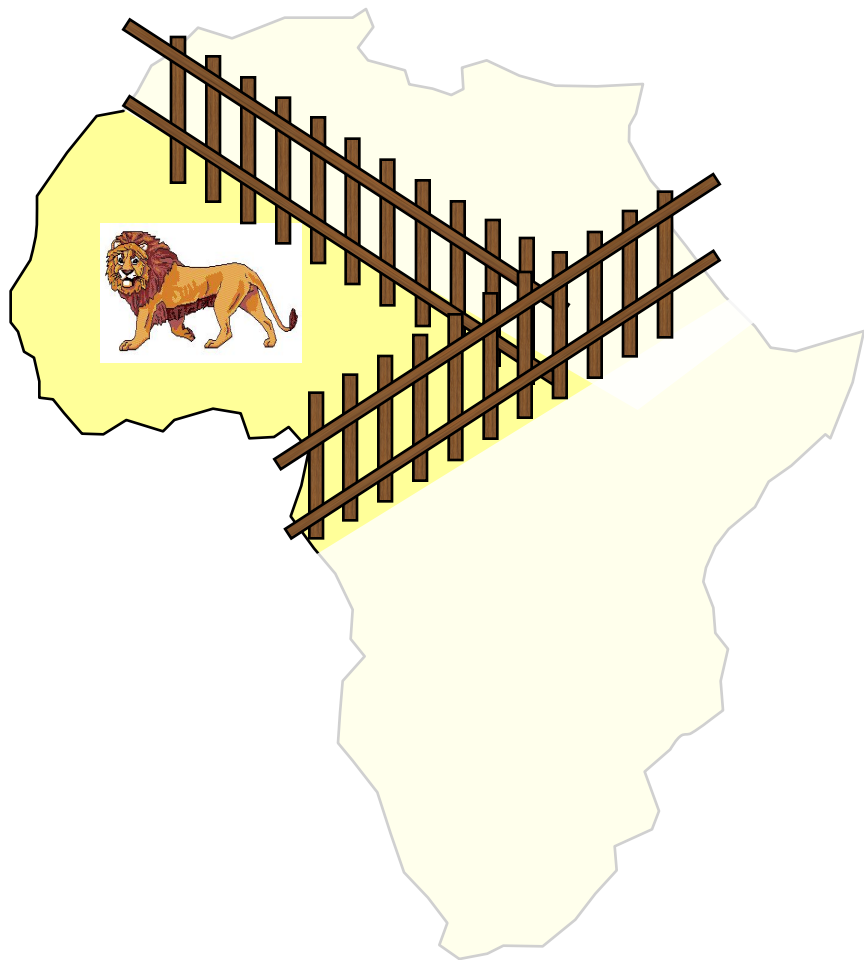


▪ данные не надо предварительно готовить



▪ низкая скорость поиска





1. Разделить область поиска на две равные части.
2. Определить, в какой половине находится нужный объект.
3. Перейти к шагу 1 для этой половины.
4. Повторять шаги 1-3 пока объект не будет «пойман».

# Поиск в базах данных

**Двоичный поиск в БД** – требует предварительной сортировки. **Иванов?**

1	Андреев
2	Барсуков
...	
<b>512</b>	<b>Ковалев</b>
...	
1023	Юрьев
1024	Яшин

1	Андреев
...	
<b>255</b>	<b>Журов</b>
...	
512	Ковалев
...	
1024	Яшин

...	
255	Журов
...	
<b>383</b>	<b>Игнатъев</b>
...	
512	Ковалев
...	



**Сколько сравнений?**

**11  
сравнений!**



■ быстрый поиск



- записи надо отсортировать по нужному полю
- можно использовать только для одного поля.

# Поиск по индексам

**Индекс** – это вспомогательная таблица, которая предназначена для быстрого поиска в основной таблице по выбранному столбцу.

## Таблица

Номер	Дата	Товар	Количество
1	02.02.2006	Киви	6
2	01.11.2006	Бананы	3
3	12.04.2006	Апельсины	10

## Индексы:

### по дате

Номер	Дата
1	02.02.2006
3	12.04.2006
2	01.11.2006

### по товару

Номер	Товар
3	Апельсины
2	Бананы
1	Киви

### по количеству

Номер	Количество
2	3
1	6
3	10

# Поиск по индексам

---

## Алгоритм:

- 1) **двоичный поиск по индексу** – найти номера нужных записей;
- 2) выбрать эти записи по номерам из основной таблицы.



- двоичный поиск по всем столбцам, для которых построены индексы



- индексы занимают **место на диске**;
- при изменении таблицы надо перестраивать все индексы (в СУБД – автоматически).

