

БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

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

БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Тема 1. Информационные системы

Базы данных и информационные системы

Информационная система – это аппаратные и программные средства, предназначенные для того, чтобы своевременно обеспечить пользователей нужной информацией.

Задачи ИС – обеспечить

-хранение данных;

-доступ к данным, т.е. возможность искать и изменять данные

Базы данных и информационные системы

Информационная система

База данных(БД) – это организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ПК и постоянного применения

Фактографические
– краткие сведения в строго определенном формате

- Каталог библиотеки
- База данных кадров
- Справочная картотека
-

Документальные –
документы в
различном формате

- Текстовая информация
- Графические объекты
- Звуковая информация
- Объекты мультимедиа
-

Система управления базами данных(СУБД) – комплекс программных средств для создания баз данных, хранения и поиска в них необходимой информации

Распределение БД – хранение различных частей одной БД на множестве компьютеров, объединенных между собой сетью

- **локальные ИС**

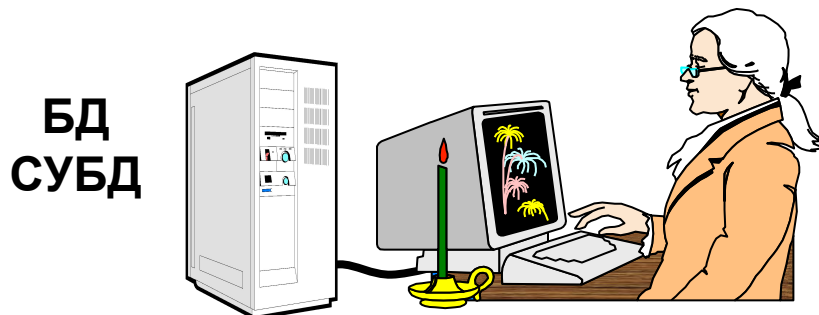
БД и СУБД находятся на одном компьютере.

- **файл-серверные**

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

- **клиент-серверные**

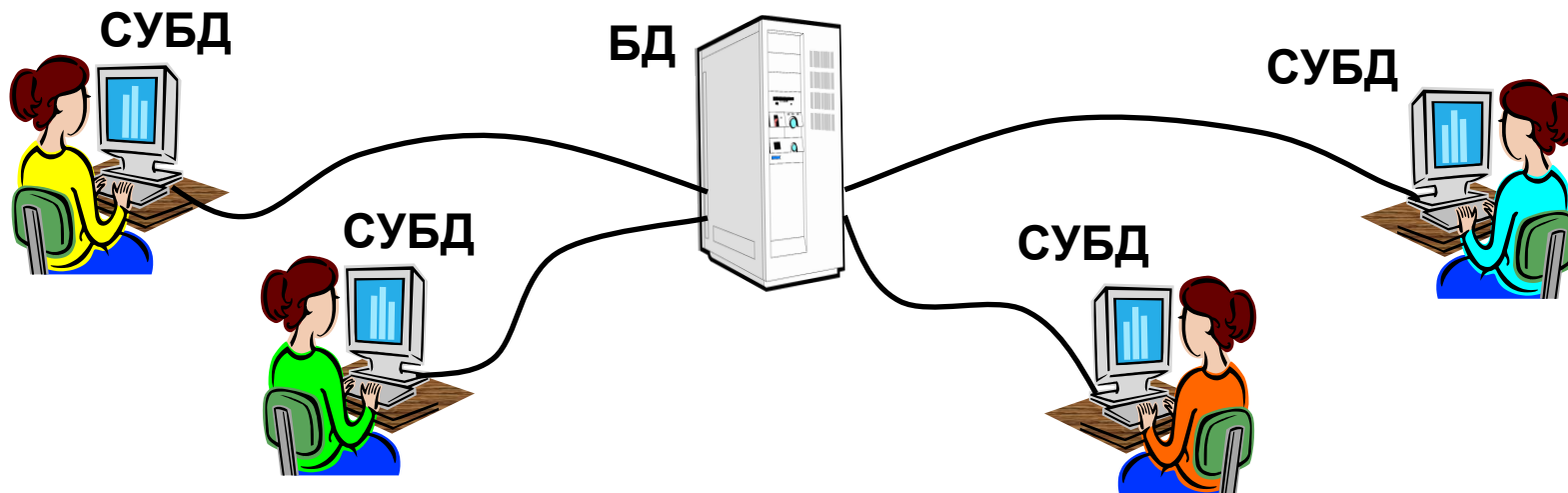
БД и основная СУБД находятся на сервере, СУБД на рабочей станции посылает запрос и выводит на экран результат.



- автономность
(независимость)



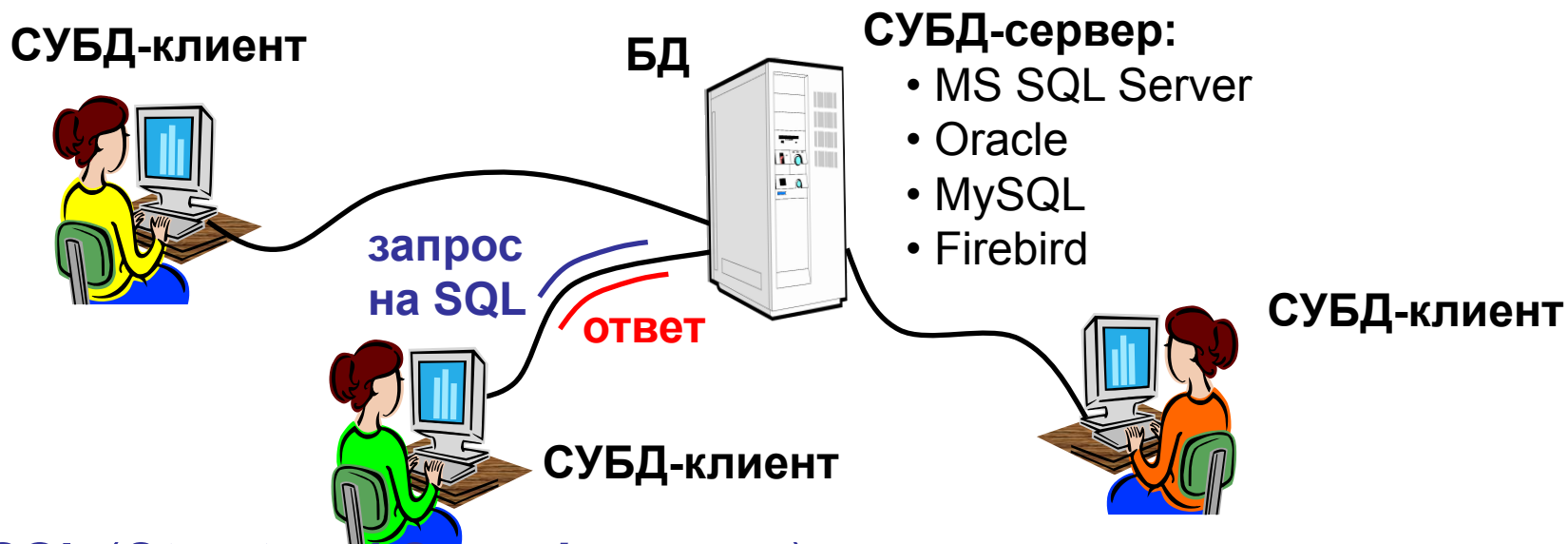
- с БД работает только один человек
- сложно обновлять при большом количестве пользователей
- практически невозможно «стыковать» изменения, вносимые несколькими пользователями



- 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

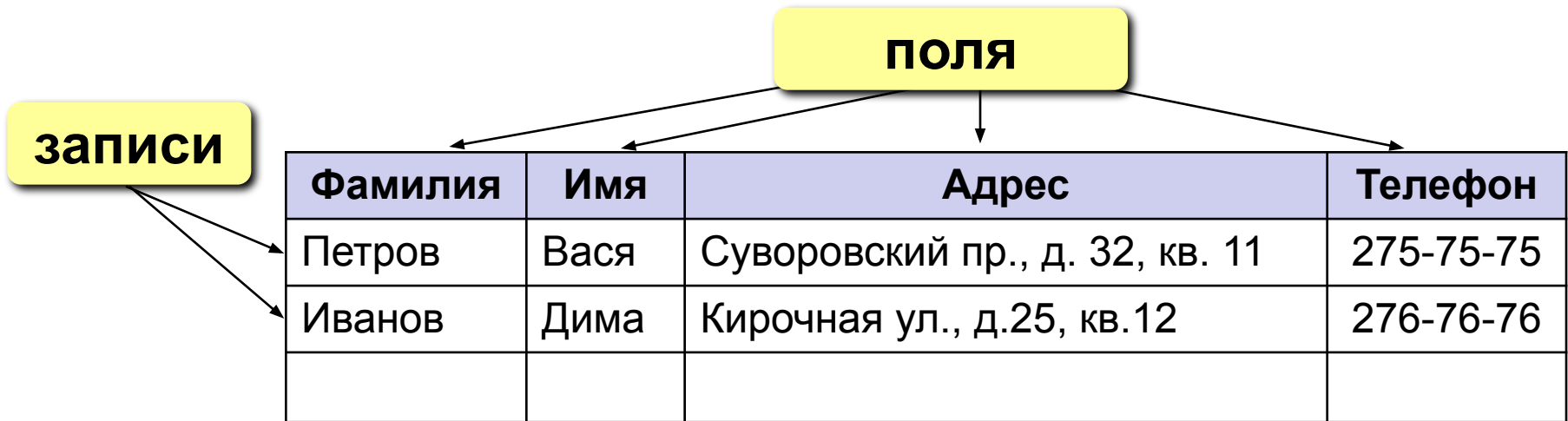
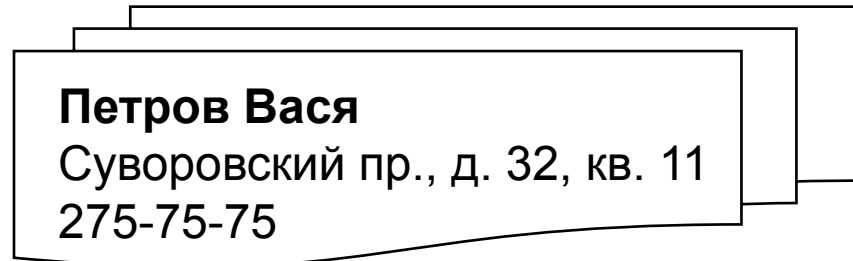
Типы баз данных

- **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. |

Табличные БД

- 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



Простой ключ



| Номер | Автор | Название | Год | Полка |
|-------|-------------|------------|------|-------|
| 001 | Беляев А.Р. | Звезда КЭЦ | 1990 | 3 |
| 002 | Олеша Ю.К. | Избранное | 1987 | 5 |
| 003 | Беляев А.Р. | Избранное | 1994 | 1 |

В БД «Домашняя библиотека» у разных книг могут совпадать значения полей,
но инвентарный номер у каждой книги свой

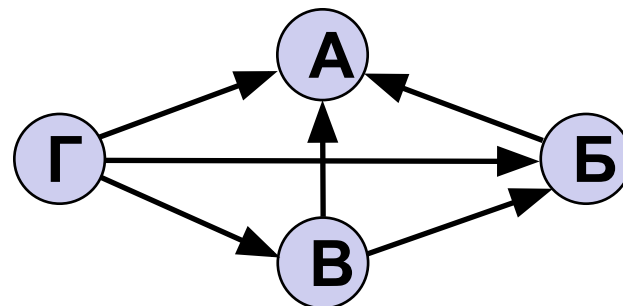
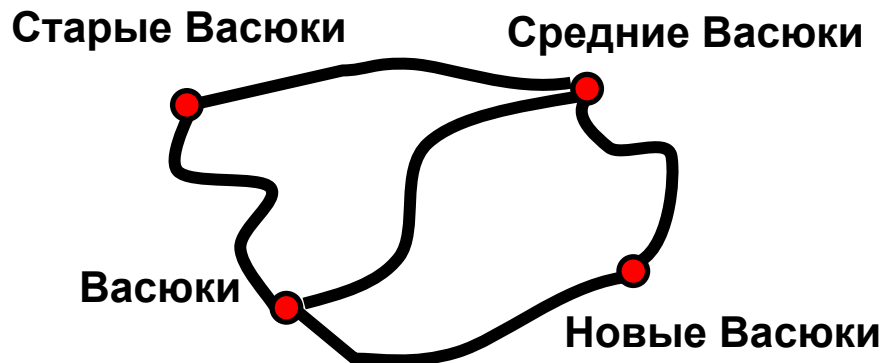
Составной ключ



| Город | № школы | Директор | Адрес | Телефон |
|----------|---------|-------------|------------|---------|
| Крюков | 1 | Иванов А.П. | Пушкина, 5 | 12-35 |
| Шадринск | 1 | Строев С.С. | Лесная, 14 | 4-33-11 |
| Шадринск | 2 | Иванов А.П. | Мира, 34 | 4-23-24 |

В БД «Школы области» у разных записей
одновременно не могут совпасть
только сочетание двух полей:
город и номер школы (это составной ключ)

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)



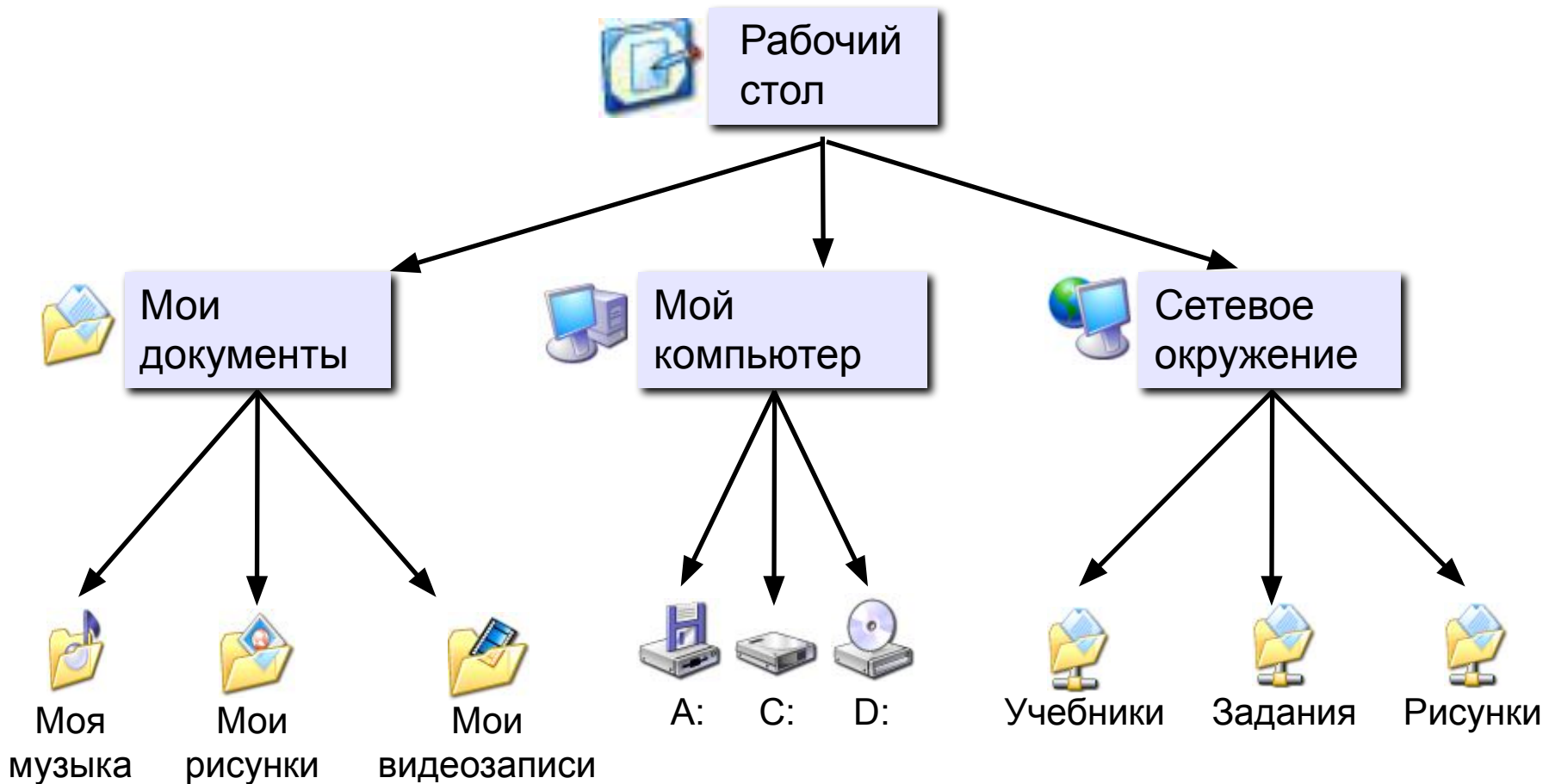
- сложно хранить информацию о всех связях
- запутанность структуры



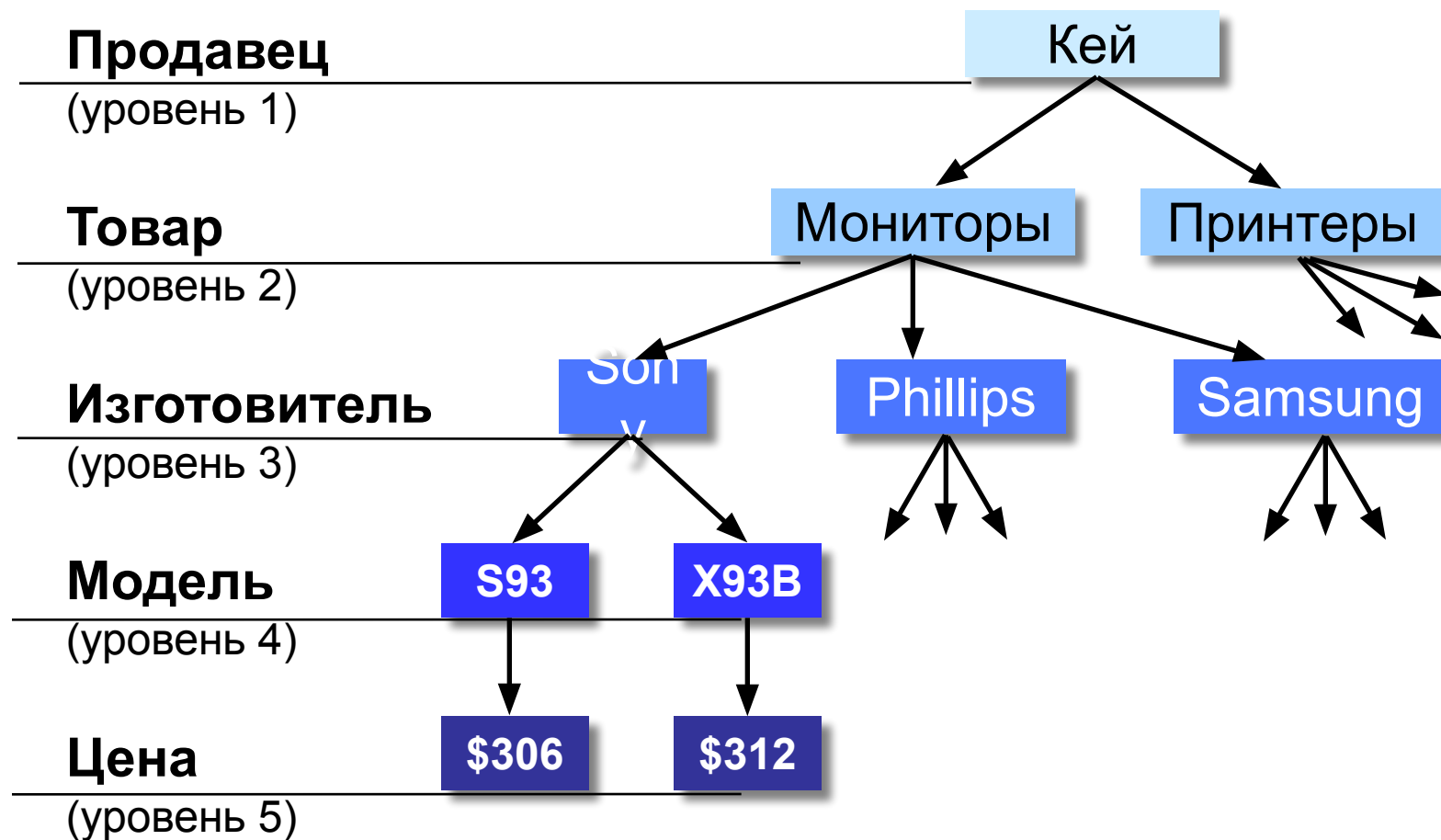
Можно хранить в виде таблицы, но с дублированием данных!

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

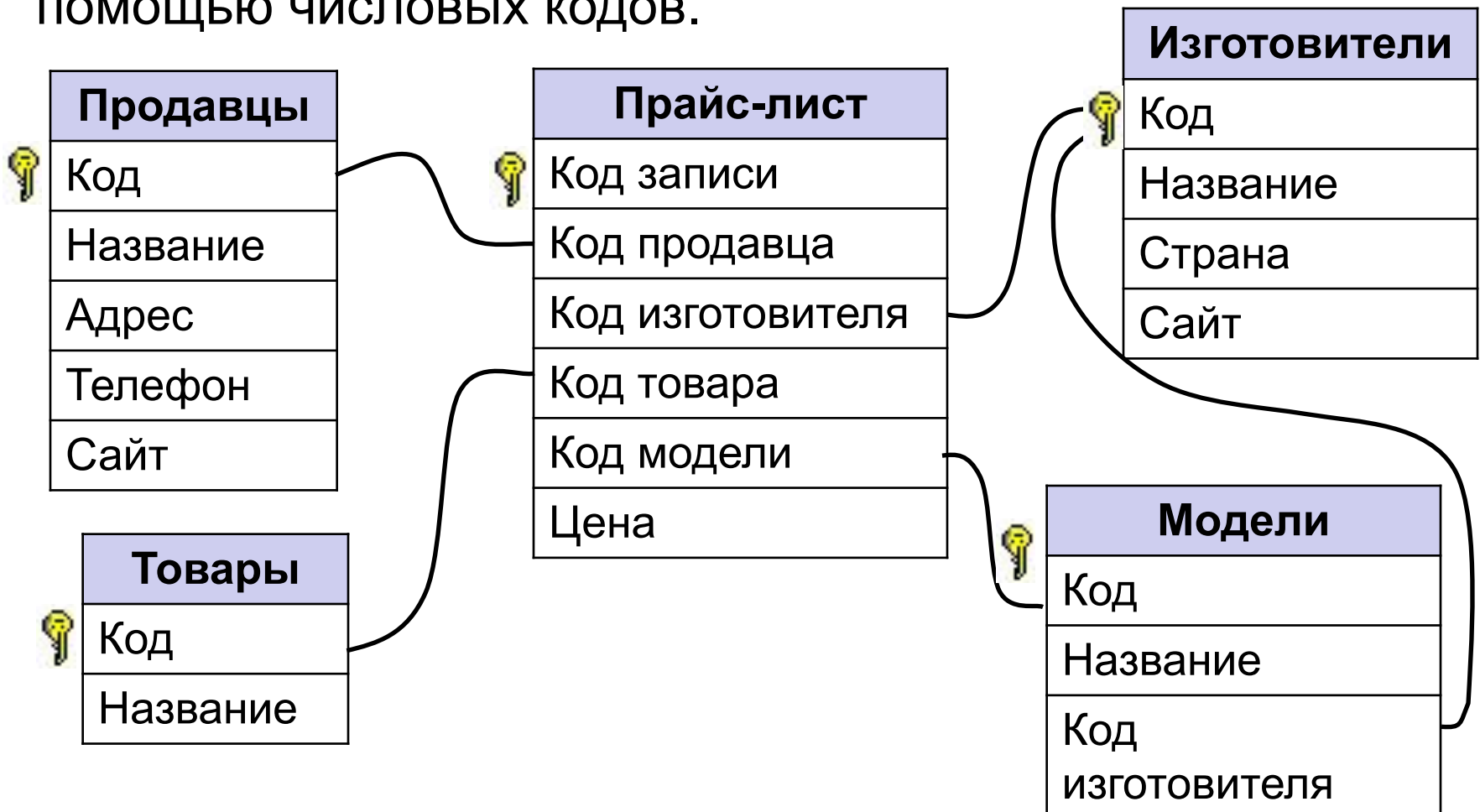
БАЗЫ ДАННЫХ. ИНФОРМАЦИОННЫЕ СИСТЕМЫ

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

| Фамилия | Кафедра | Дата_рождения | Медаль | Балл1 | Балл2 | Балл3 |
|---------------|---------|---------------|--------|-------|-------|-------|
| Иванов И.И. | МОиПЭВМ | 15.09.84 | Есть | 5 | 5 | 5 |
| Алексеев А.А. | ИНОУп | 24.01.84 | Есть | 5 | 5 | 5 |
| Сергеев С.С. | ИНОУп | 12.11.84 | Нет | 5 | 4 | 4 |
| Борисов Н.Г | ВТ | 29.08.84 | Есть | 5 | 5 | 5 |
| Николаев А.И. | МОиПЭВМ | 10.03.84 | Есть | 5 | 5 | 5 |
| Агеев Ю.И. | ВТ | 31.12.83 | Нет | 4 | 4 | 4 |
| Володин Н.И. | МОиПЭВМ | 03.12.80 | Нет | 4 | 4 | 3 |

Реляционные БД



Реляционная модель базы данных имеет следующие свойства:

- Каждый элемент таблицы — один элемент данных.
- Все столбцы в таблице являются однородными, т. е. имеют один тип (числа, текст, дата и т. д.).
- Каждый столбец (поле) имеет уникальное имя.
- Одинаковые строки в таблице отсутствуют.
- Порядок следования строк в таблице может быть произвольным и может характеризоваться количеством полей, количеством записей, типом данных.

Над этой моделью базы данных удобно производить следующие действия:

- сортировку данных (например, по алфавиту);
- выборку данных по группам (например, по датам рождения или по фамилиям);
- поиск записей (например, по фамилиям) и т. д.

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**).

Реляционные БД

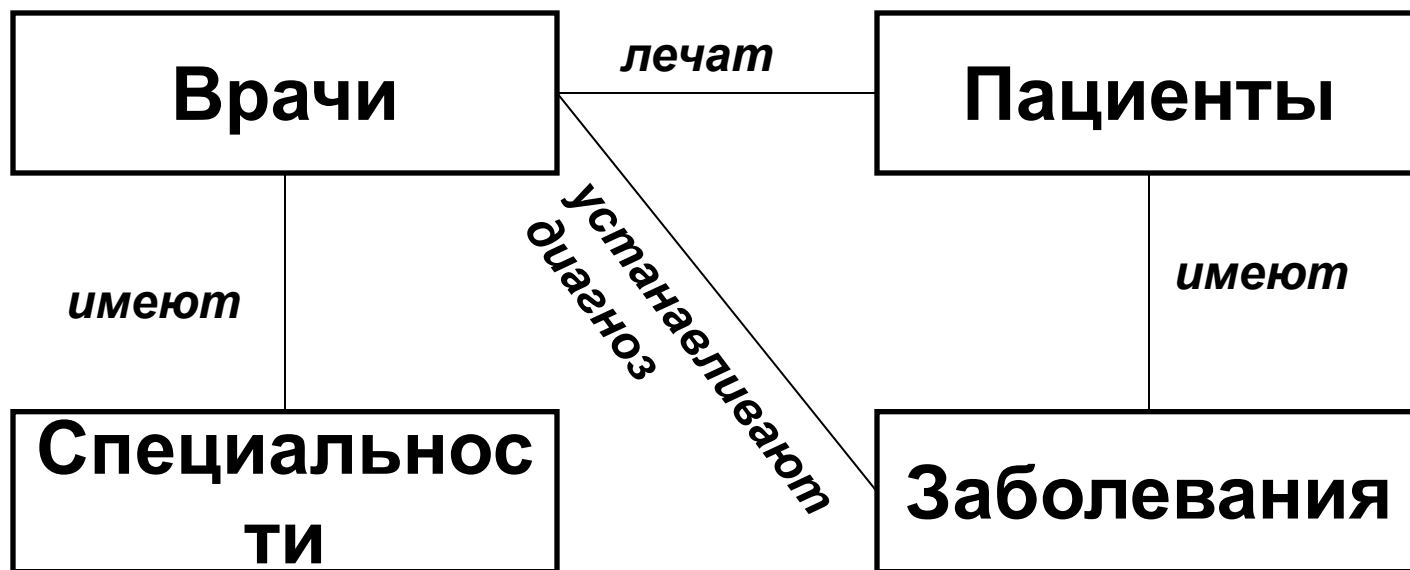
Первым этапом в работе над базой данных должно стать составление инфологической модели, которая содержит данные конкретной предметной области.

Предметную область образует совокупность предметов, которые находятся между собой в определенных отношениях и связях.

Реляционные БД

Пример:

В предметной области Поликлиника можно выделить врачей, пациентов, диагнозы, специальности врачей и т.д.:



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 (« ∞ - ∞ ») – одной записи в первой таблице соответствует сколько угодно записей во второй, и наоборот.

учителя

| Код | Фамилия |
|-----|---------|
| 1 | Иванов |
| 2 | Петров |
| ... | |

∞ ∞

предметы

| Код | Название |
|-----|-----------|
| 1 | История |
| 2 | География |
| 3 | Биология |
| ... | |

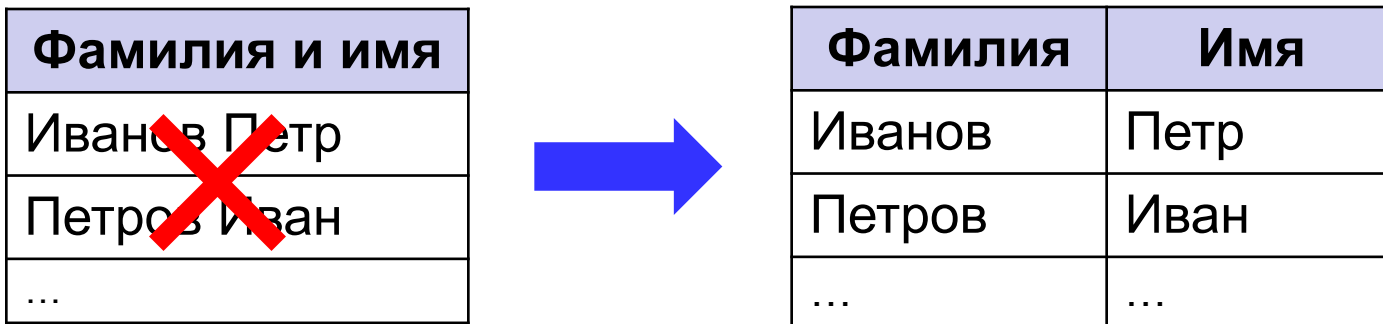
Реализация – через третью таблицу и две связи «1- ∞ ».



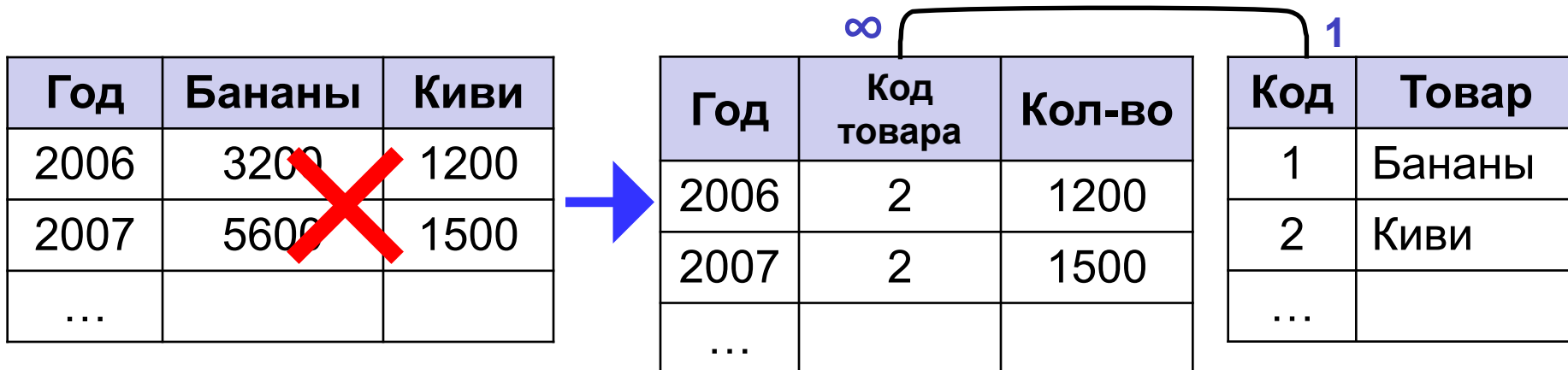
Нормализация базы данных

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 | 9 000 р. |
| 2 | Винчестер | 11 000 р. |
| ... | | |

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 | 12 000 |
| 2 | Киви | 1500 | 20 | 30 000 |
| ... | | | | |

Поиск в базах данных



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

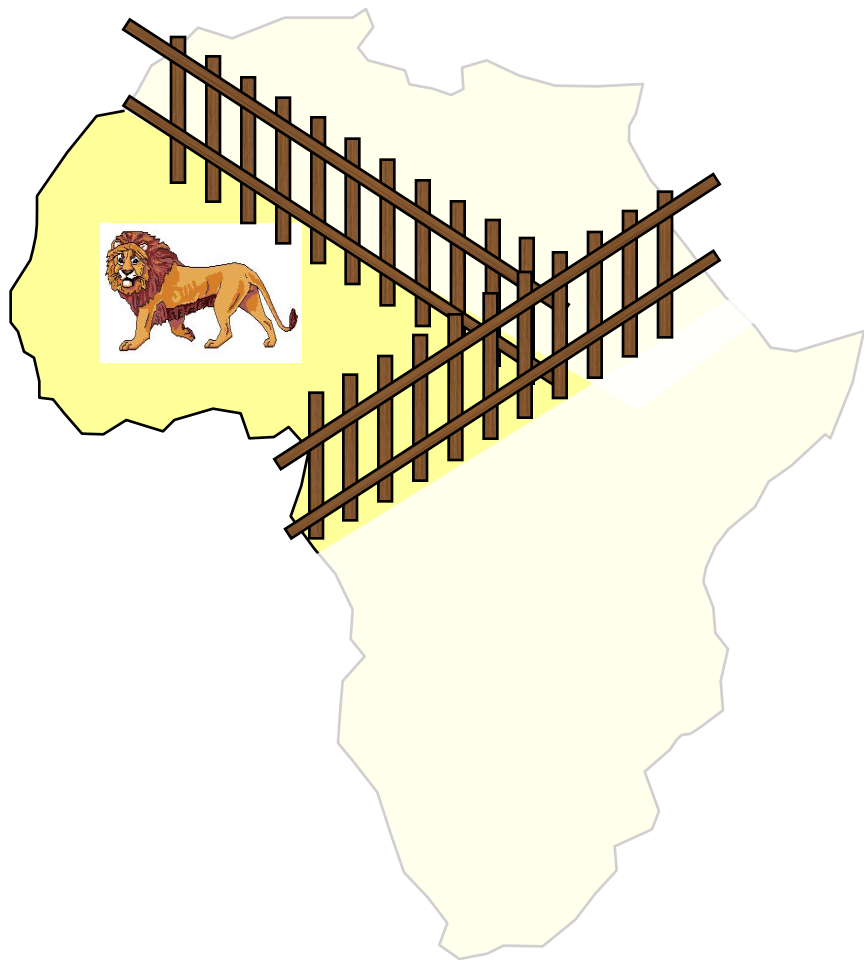


| Код | Фамилия |
|------|------------|
| 1 | Сидоров |
| 2 | Ветров |
| ... | |
| 1024 | Померанцев |

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 |
| ... | |
| 512 | Ковалев |
| ... | |
| 1023 | Юрьев |
| 1024 | Яшин |

| | |
|------------|--------------|
| 1 | Андреев |
| ... | |
| 255 | Журов |
| ... | |
| 512 | Ковалев |
| ... | |
| 1024 | Яшин |

| | |
|------------|-----------------|
| ... | |
| 255 | Журов |
| ... | |
| 383 | Игнатъев |
| ... | |
| 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).

Целостность базы данных

Целостность БД означает, что БД содержит полную и непротиворечивую информацию и удовлетворяет всем заданным ограничениям

- Физическая целостность БД (защита от разрушения в случае отказа оборудования)
- Логическая целостность – непротиворечивость данных

Логическая целостность:

- каждое поле имеет свой тип;
- некоторые поля объявляются обязательными;
- для полей, значение которых не может повторяться, строятся уникальные индексы;
- вводятся условия для значений отдельных полей;
- для сложных данных используются шаблоны (маски);
- вводятся условия, которые должны выполняться для нескольких полей каждой записи.