

Лекция 2.

**Задачи СУБД. Классификация и
выбор СУБД**

Повторяем термины

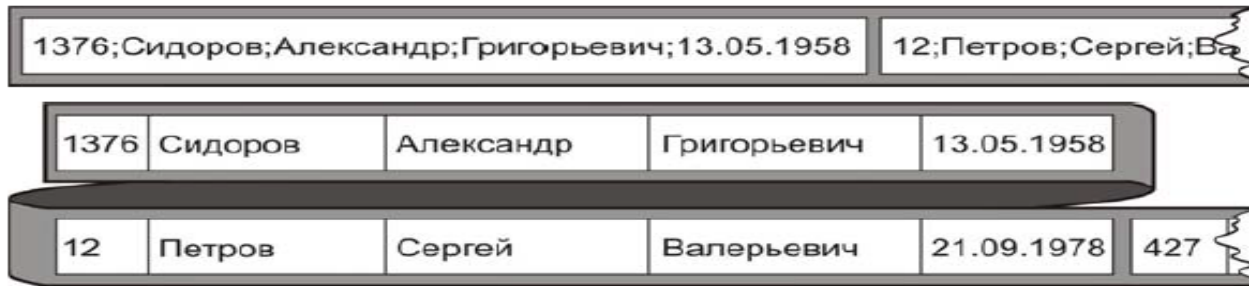
- Файл
- Структурированный файл
- Структурированный файл с метаданными
- База данных

Повторяем термины

- **Файл** – поток байтов, хранящихся на внешнем носителе

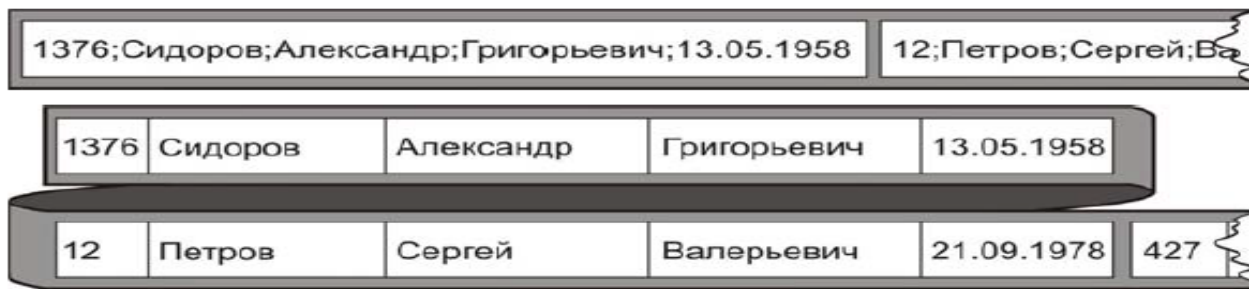
Повторяем термины

- **Файл** – поток байтов, хранящихся на внешнем носителе
- **Структурированный файл** – файл с внутренней структурой

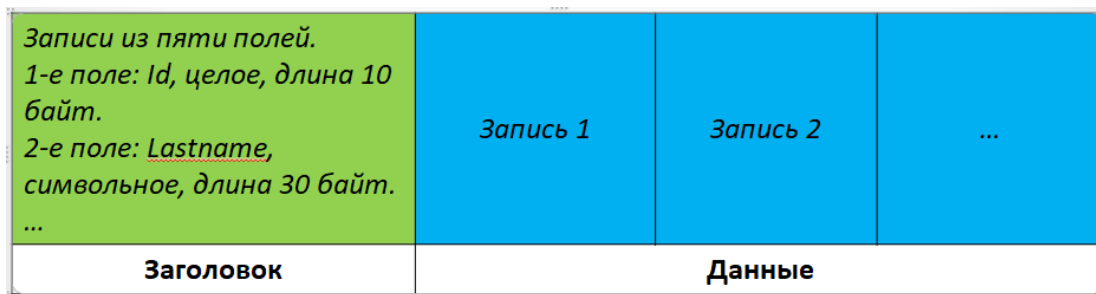


Повторяем термины

- **Файл** – поток байтов, хранящихся на внешнем носителе
- **Структурированный файл** – файл с внутренней структурой



- **Структурированный файл с метаданными** – файл, содержащий в себе информацию о своей структуре

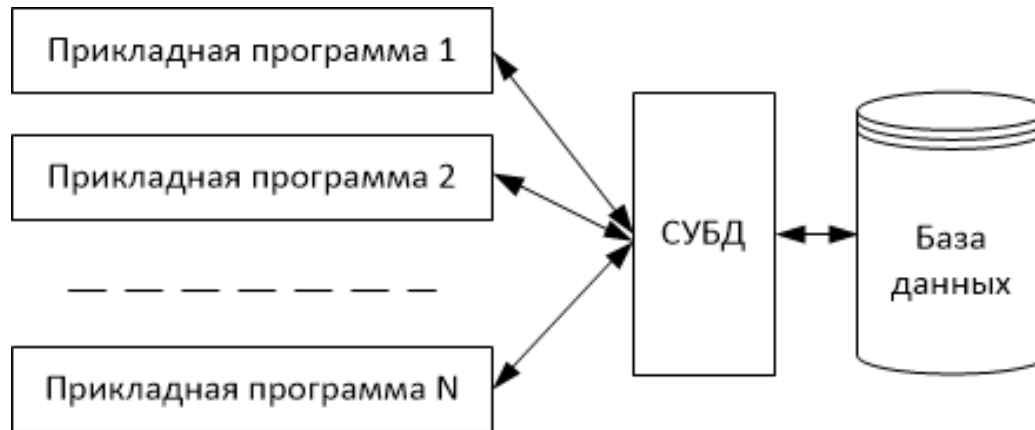


Повторяем термины

База данных - совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения.

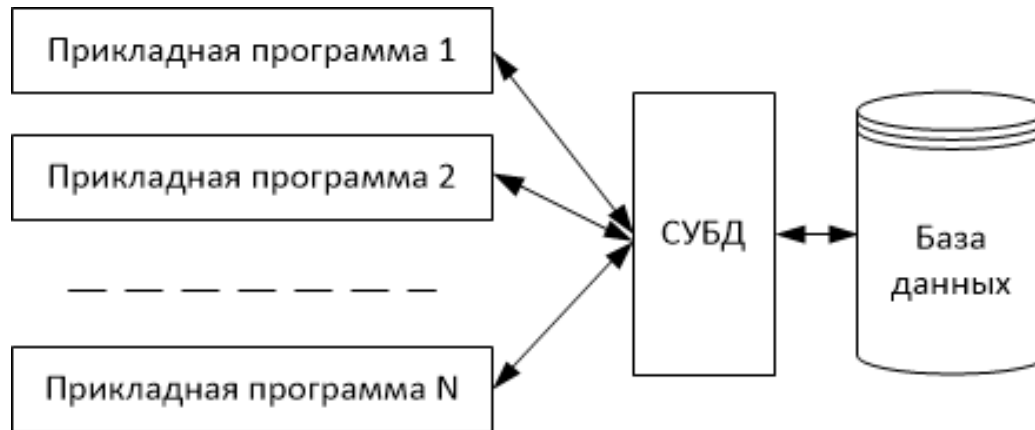
Отделение данных от программ

СУБД – программный комплекс поддержки общедоступных данных, предназначенный для создания, ведения и использования базы данных многими пользователями (прикладными программами).



Отделение данных от программ

СУБД – программный комплекс поддержки общедоступных данных, предназначенный для создания, ведения и использования базы данных многими пользователями (прикладными программами).



Логическая независимость данных – общая структура данных может быть изменена без изменения прикладных программ.

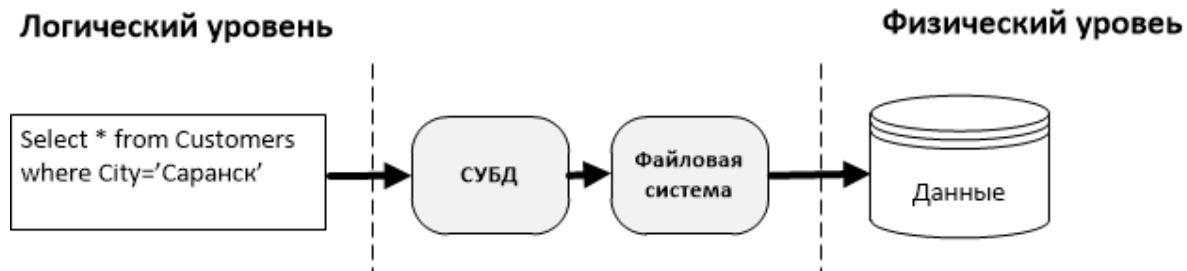
Физическая независимость данных – изменение способов организации базы данных на внешнем носителе не должно влиять на логическое представление данных и прикладные программы.

- Файловая система на логическом уровне унифицирует запись и считывание данных.
- СУБД на логическом уровне унифицирует операции с данными, учитывающие их структуру и содержание данных.

Работа с файлом



Работа с базой данных



Основные функции СУБД

1. Обеспечение логической и физической независимости прикладных программ и данных.

Основные функции СУБД

2. Определение структуры создаваемой базы данных, ее инициализация и проведение начальной загрузки.

```
CREATE TABLE `vendor` (  
  `id` INTEGER PRIMARY KEY,  
  `name` VARCHAR(30) NOT NULL,  
  `city` VARCHAR(30) NOT NULL,  
  `percent` INTEGER NOT NULL CHECK (`percent`>0)  
);
```

Основные функции СУБД

3. Обеспечение возможности манипулирования данными (выборка данных, вычисления, разработка интерфейса ввода/вывода, визуализация).

Специальный язык, CLI или GUI.

SQL

```
INSERT INTO `vendor` (`id`, `name`, `city`, `percent`) VALUES
(1001, 'Иванов', 'Саранск', 12),
(1002, 'Петров', 'Москва', 13),
(1003, 'Андреев', 'Кострома', 10),
(1004, 'Сидоров', 'Саранск', 10)
```

```
SELECT * FROM vendor WHERE name LIKE "Пе%"
```

NoSQL (MongoDB)

```
db.vendor.insert({name: 'Потапкин', city: 'Саранск', percent: '14'})
db.vendor.insert({name: 'Петров', city: 'Москва', percent: '15'})
db.vendor.insert({name: 'Андреев', city: 'Кострома', percent: '12'})
db.vendor.insert({name: 'Сидоров', city: 'Саранск', percent: '12'})
db.vendor.insert({name: 'Козлов', city: 'Саранск', percent: '26'})
```

```
db.vendor.find({city: 'Саранск', percent: {$gt: '14'}}).sort({name: 1})
```

Основные функции СУБД

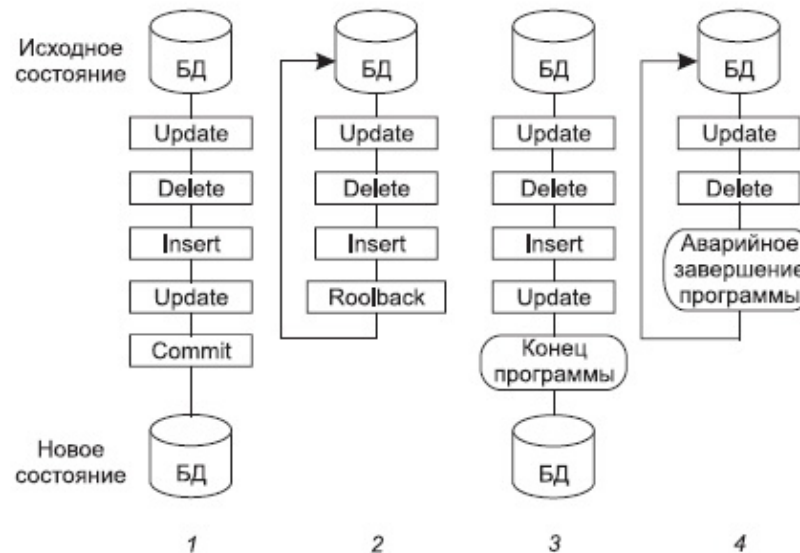
4. Защита логической целостности базы данных.

```
CREATE TABLE `vendor` (  
  `id` INTEGER PRIMARY KEY,  
  `name` VARCHAR(30) NOT NULL,  
  `city` VARCHAR(30) NOT NULL,  
  `percent` INTEGER NOT NULL CHECK (`percent` > 0)  
)
```

```
CREATE TABLE `customer` (  
  `id` INTEGER PRIMARY KEY,  
  `name` VARCHAR(30) NOT NULL,  
  `city` VARCHAR(30) NOT NULL,  
  `rating` INTEGER NOT NULL CHECK (`rating` > 0),  
  `vendor_id` INTEGER NULL,  
  FOREIGNKEY (`vendor_id`) REFERENCES `vendor` (`id`)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE  
)
```

Основные функции СУБД

5. Защита физической целостности. Поддержка транзакций и блокировок.



Основные функции СУБД

6. Управление полномочиями пользователей на доступ к базе данных.
7. Синхронизация работы нескольких пользователей.

Классификация СУБД

1. По архитектуре:

- Централизованная (встроенная)
- Файл-сервер
- Клиент-сервер

2. По масштабу:

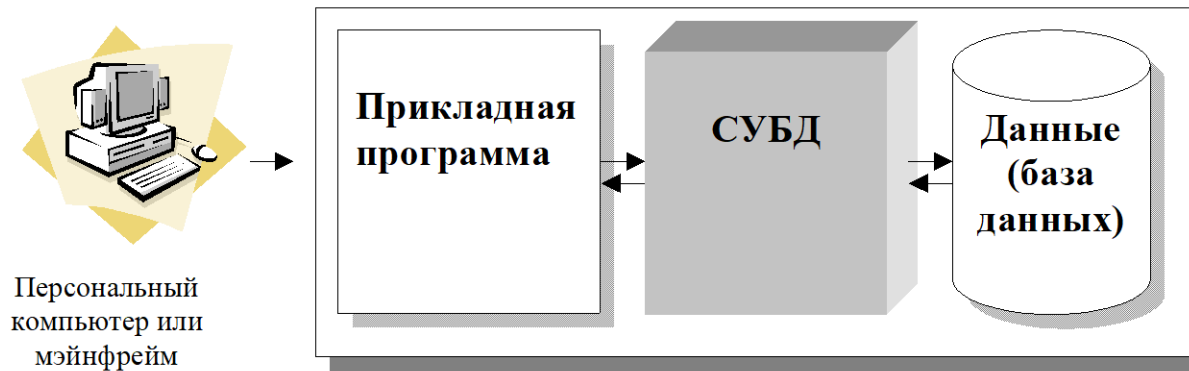
- Персональные (десктопные и облачные)
- Серверные

3. По способу структурирования данных:

- Реляционные
- Хранилища «ключ-значение»
- Столбцовые
- Документарные
- Графовые

Централизованная архитектура. Встраиваемые СУБД

- База данных – набор файлов, находящихся на жестком диске компьютера.
- СУБД и приложение для работы с базой данных установлены на той же машине.
- Простейший случай (встроенная СУБД): БД — один файл, СУБД — библиотека, подгружаемая к прикладной программе.



Ранние СУБД для мэйнфреймов (IDMS, ...) и персоналок (dBase, ...).

Централизованная архитектура. Встраиваемые СУБД

Microsoft Access,
LibreOffice Base, Firebird,
SQLite



Преимущества:

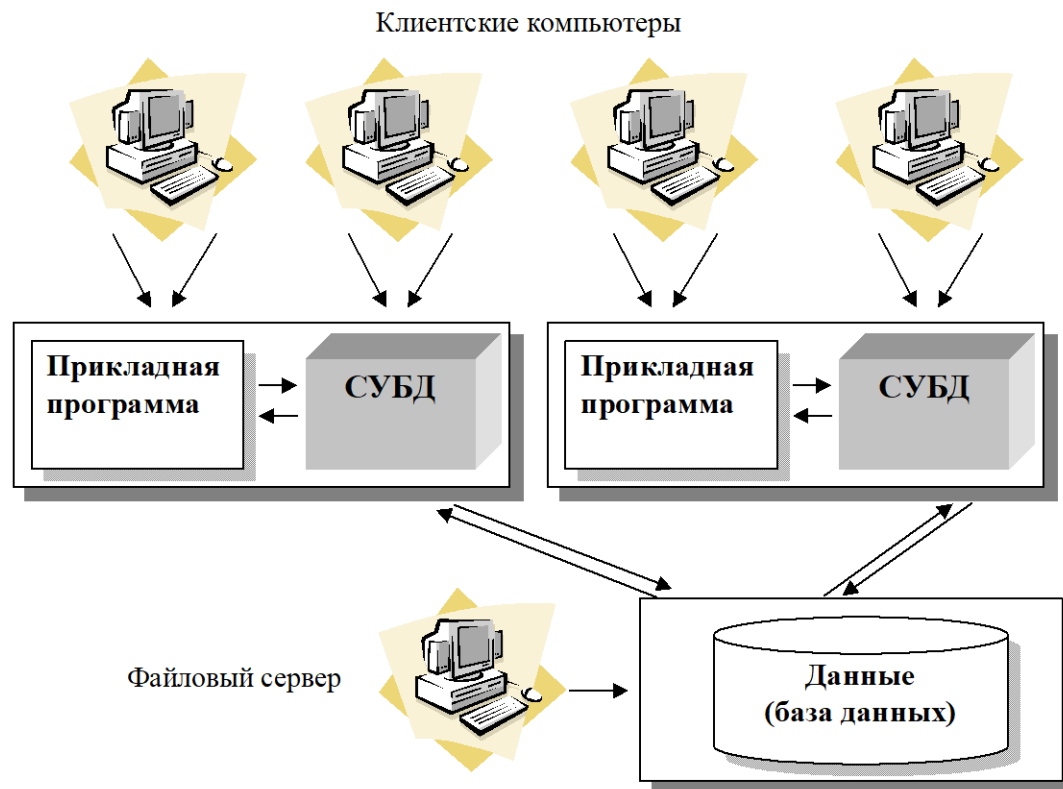
- Простота разработки и развертывания приложений.
- Простота обслуживания локальной БД.
- Высокое быстродействие на простых операциях считывания и модификации одиночных записей.

Недостатки:

- Высокий риск потери или повреждения данных.
- Невозможность распределения вычислительной нагрузки.

Архитектура «файл-сервер»

- База данных – набор файлов на выделенном сервере.
- СУБД устанавливается на клиентах.
- Центральный сервер выполняет в основном только роль хранилища файлов, не участвуя в обработке самих данных.



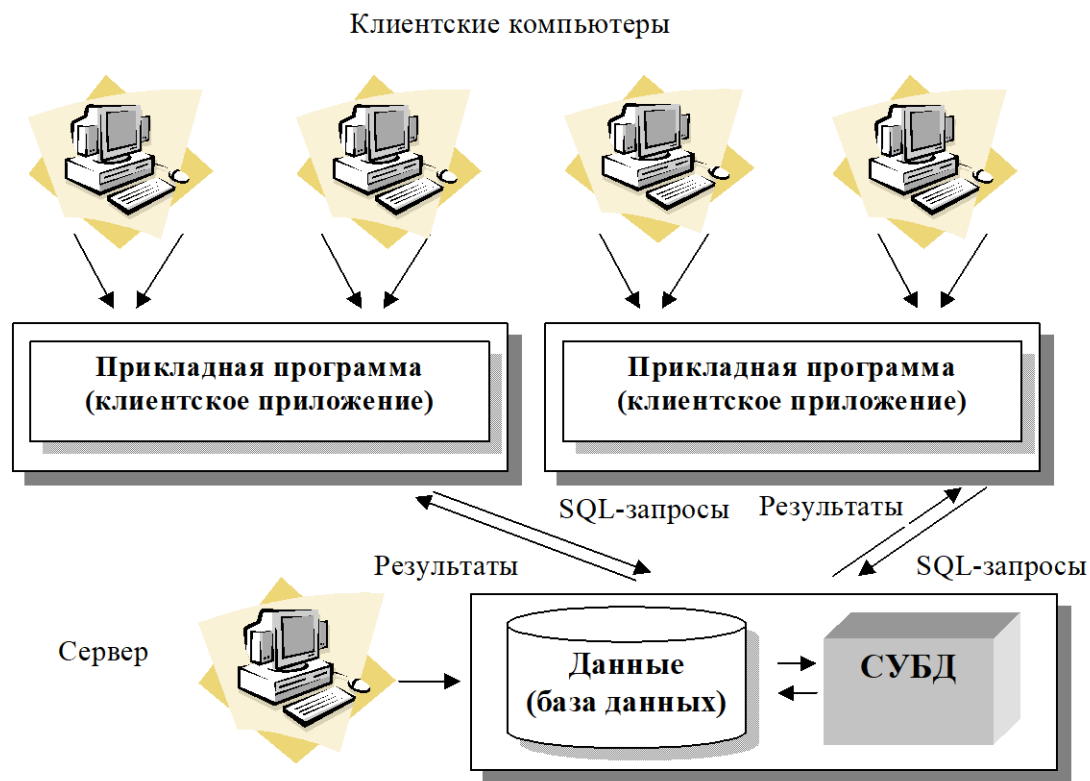
Архитектура «файл-сервер»

Недостатки

- Резкое падение производительности при обращении многих пользователей к одним и тем же данным (блокировка файла на время выполнения медленных сетевых операций).
- Данные обрабатываются на клиентских машинах – большой сетевой трафик (файлы с данными полностью копируются на клиентов) и загрузка мощностей клиентов.
- Низкий уровень безопасности (только на уровне файловой системы на сервере). *Несанкционированный доступ, внесение ошибочных изменений.*

Архитектура «клиент-сервер»

- База данных и СУБД (интеллектуальная) на выделенном сервере в сети или облачном ресурсе (DBaaS).
- На клиентах стоят приложения. К серверу посылаются только текст запросов на специальном языке.



MySQL, Microsoft SQL Server, PostgreSQL, MongoDB ...

Разделение функций

Клиент

- Формирование и посылка запросов серверу (на специальном языке или через API).
- Интерпретация результатов запросов, полученных от сервера (*зависит от языка программирования*).
- Пользовательский интерфейс.

Сервер

- Прием запросов от клиентов, их интерпретация, оптимизация и выполнение.
- Отправка результатов приложению-клиенту.
- Обеспечение системы безопасности и разграничение доступа.
- Управление целостностью БД.
- Реализация стабильности многопользовательского режима работы

Преимущества «клиент-серверной» архитектуры

1. Уменьшается сетевой трафик.
2. Повышается целостность и безопасность БД.
3. Уменьшается сложность клиентских приложений.

Преимущества «клиент-серверной» архитектуры

1. Уменьшается сетевой трафик.
2. Повышается целостность и безопасность БД.
3. Уменьшается сложность клиентских приложений.

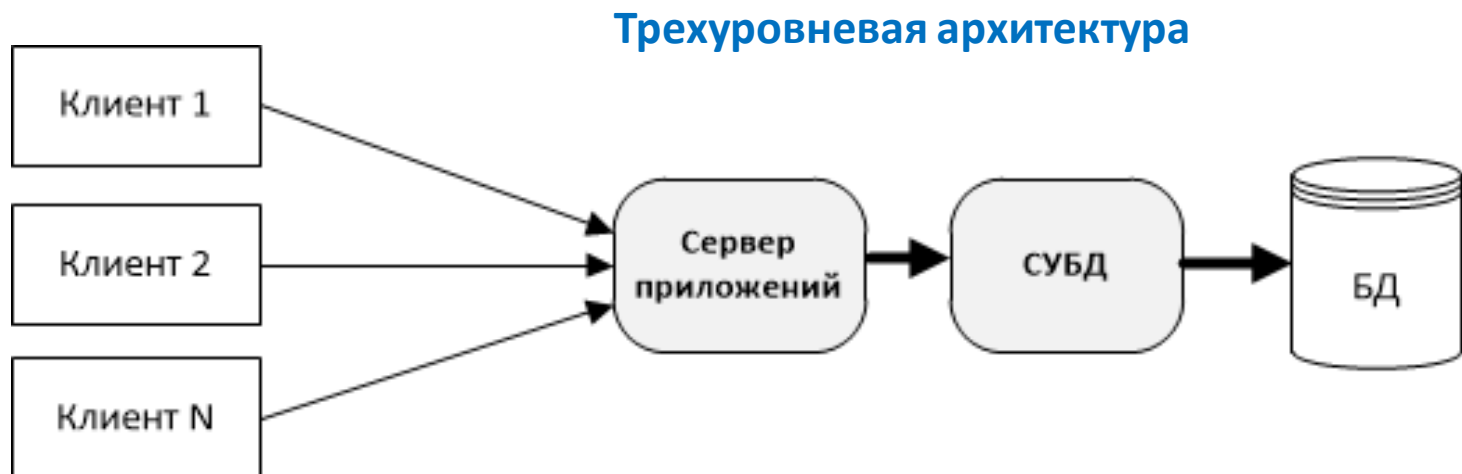
Проблема: Могут быть трудности с обновлением ПО на клиентах.

Преимущества «клиент-серверной» архитектуры

1. Уменьшается сетевой трафик.
2. Повышается целостность и безопасность БД.
3. Уменьшается сложность клиентских приложений.

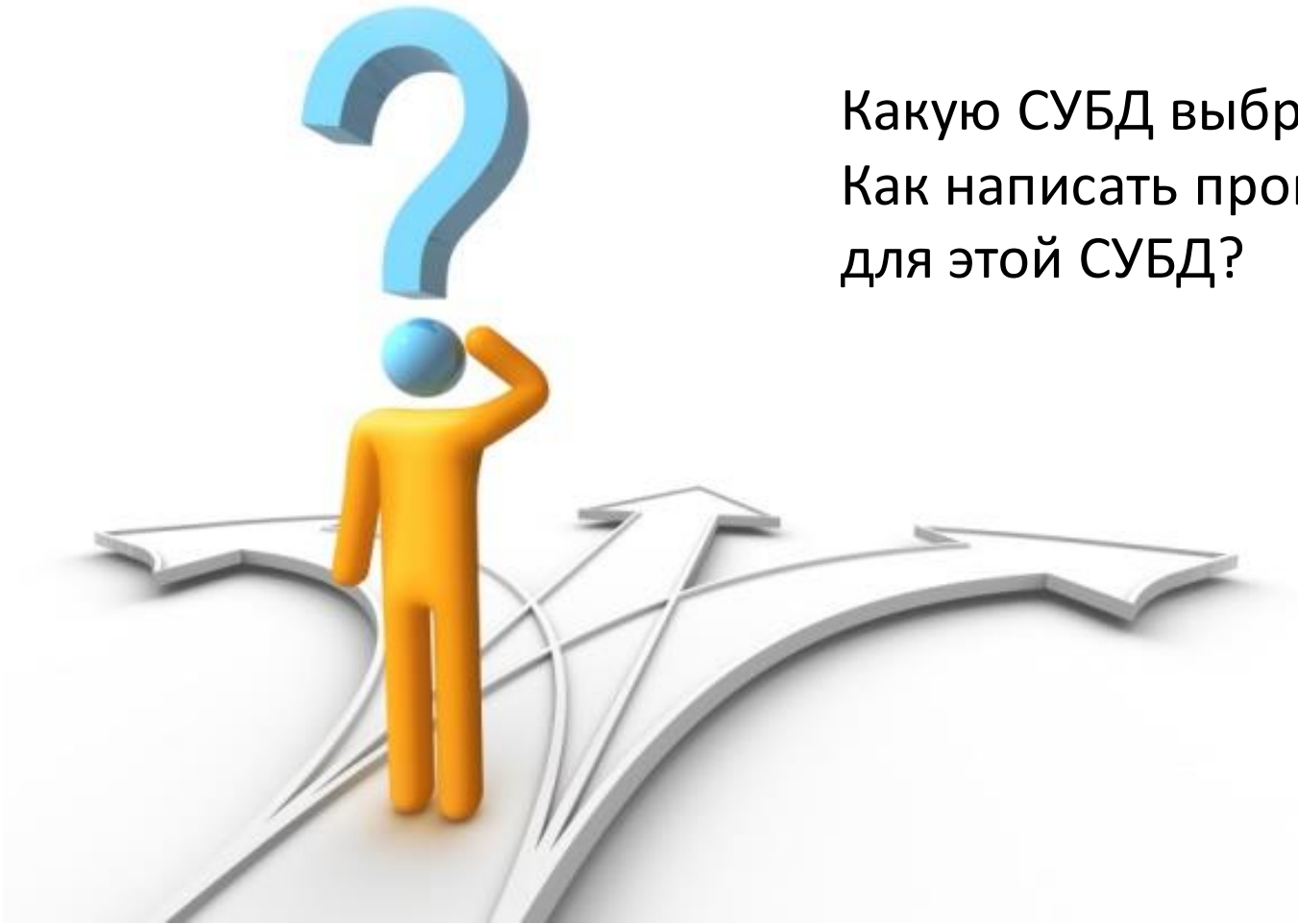
Проблема: Могут быть трудности с обновлением ПО на клиентах.

Решение: Клиент должен быть «тощим» (в идеале – только веб-браузер для отображения интерфейса пользователя). Бизнес-логика приложений выносится на отдельный сервер.



Есть задача с данными

Какую СУБД выбрать?
Как написать программу
для этой СУБД?



Выбор СУБД по масштабу задачи

Персональные (desktopные или облачные):

- Для небольших простых задач.

Серверные:

- Для средних и крупных задач, где важна производительность и надежность (либо кроссплатформенность).

Персональные СУБД

- Для небольших простых задач.
- Упрощенная архитектура и функциональность.
- Простые и удобные визуальные средства для быстрой разработки приложений (даже без программирования).

Электронные таблицы с элементами СУБД

- Microsoft Excel (*Windows, платная*)
- Google таблицы (*Облачная, бесплатная*)

fx =QUERY(Sales!\$A:\$D, "SELECT A, SUM(C) GROUP BY A PIVOT B ORDER BY A")

	A	B	C	D
1	Sales By Manager Monthly			
2	Manager	201601	201602	201603
3	Greg Lestrage	£4,144.00	£6,397.00	£3,344.00
4	John Watson	£3,500.00	£2,678.00	
5	Mrs Hudson	£6,654.00	£377.00	
6	Mycroft Holmes	£14,210.00	£88.00	
7	Sherlock Holmes	£1,000.00	£5,000.00	

В7 Нектарин

A	B	C
1	Форма ввода данных о про...	
2		
3	Дата	08.09.2016 11:20
4		
5	Клиент	Дубинин
6		
7	Товар	Нектарин
8		
9	Количество	3
10		
11	Цена	90
12		
13	Стоимость	270
14		
15		
16		
17		
18		

Проверка вводимых значений

Параметры Сообщение для ввода Сообщение об ошибке

Условие проверки

Тип данных: Игнорировать пустые ячейки

Значение: Список допустимых значений

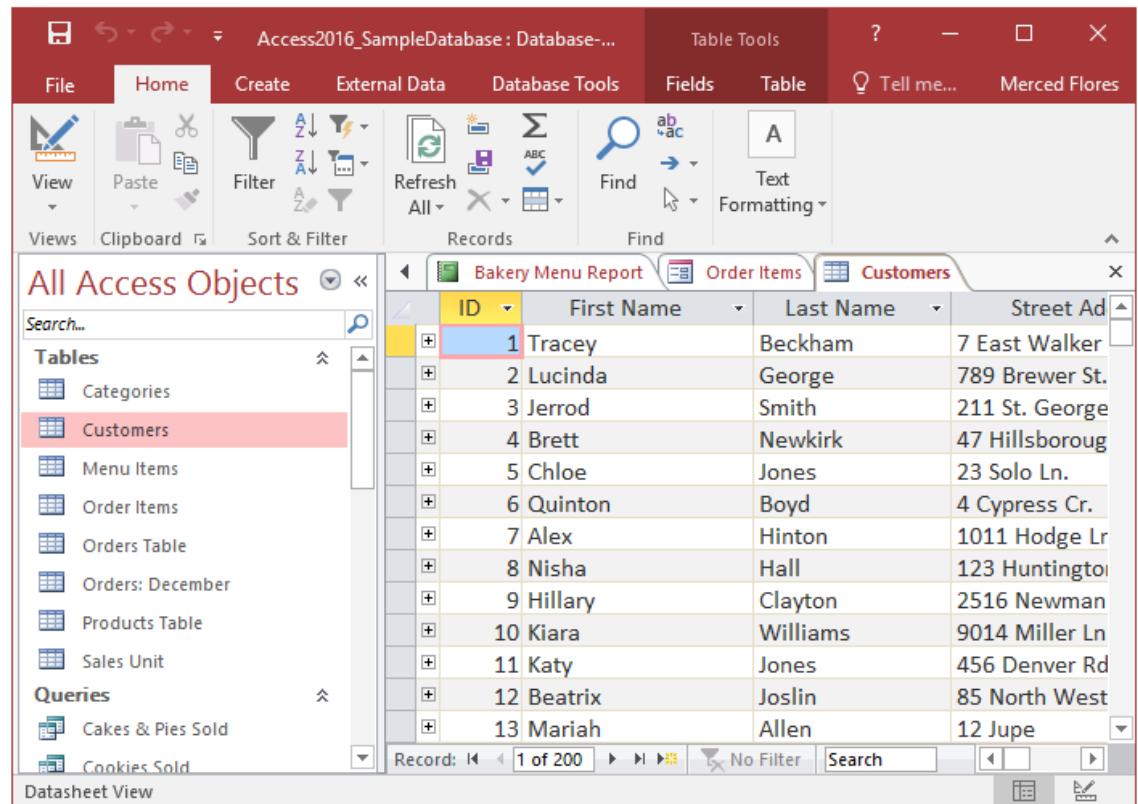
Источник:

Распространить изменения на другие ячейки с тем же условием

Очистить все

Десктопные СУБД

- Microsoft Access (*Windows, платная*)
- LibreOffice Base (*Windows, Mac, Linux, бесплатная*)
- KEXI (*Windows, Mac OS, Linux, бесплатная*)
- Filemaker (*Mac OS, платная*)

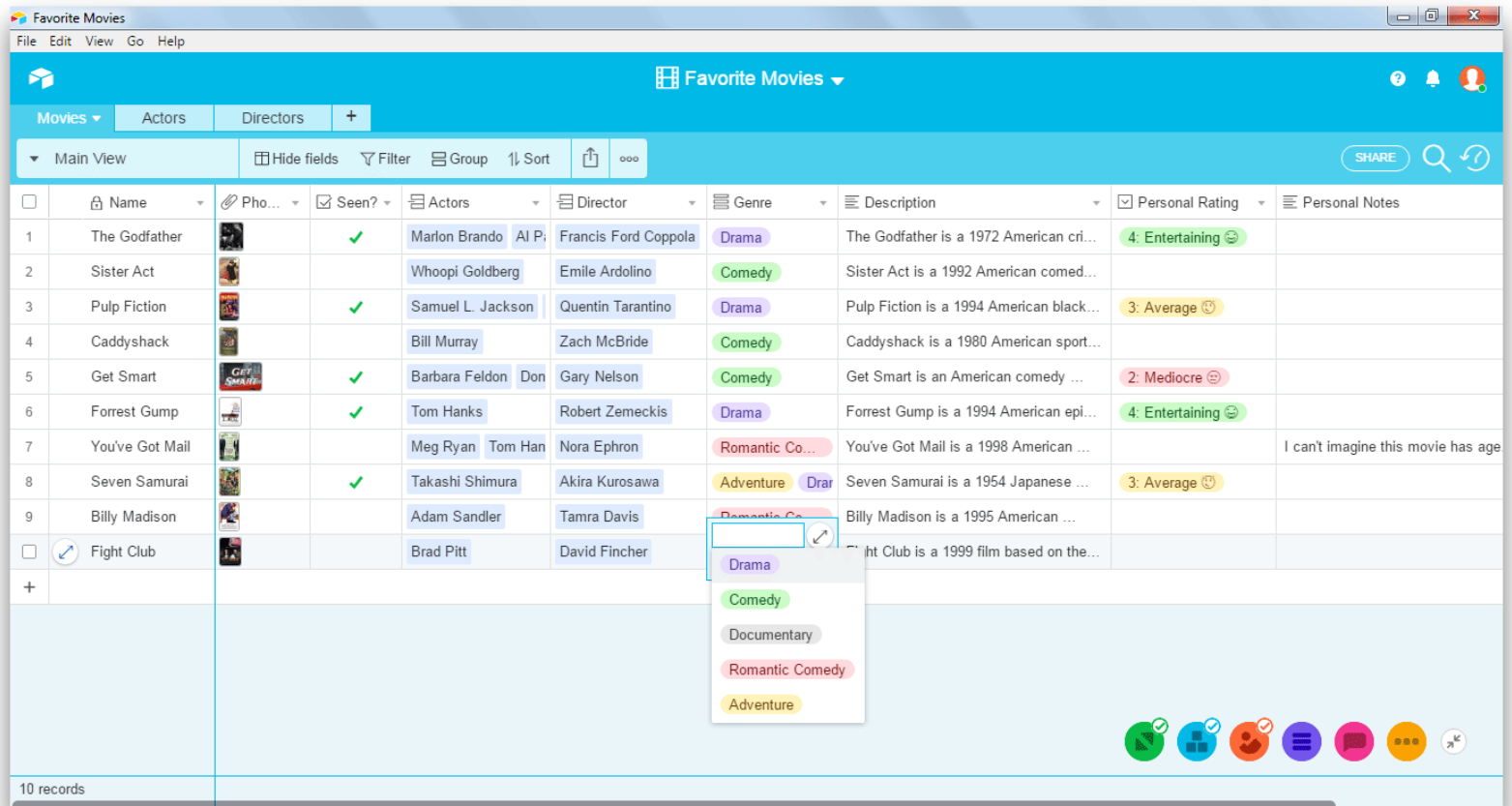


The screenshot shows the Microsoft Access 2016 interface. The title bar indicates the database is 'Access2016_SampleDatabase : Database...'. The ribbon is set to 'Table Tools' with the 'Table' tab selected. The 'All Access Objects' pane on the left shows a list of tables, with 'Customers' selected. The main window displays a datasheet view of the 'Customers' table with 13 records. The status bar at the bottom shows 'Record: 1 of 200' and 'No Filter'.

ID	First Name	Last Name	Street Ad
1	Tracey	Beckham	7 East Walker
2	Lucinda	George	789 Brewer St.
3	Jerrold	Smith	211 St. George
4	Brett	Newkirk	47 Hillsboroug
5	Chloe	Jones	23 Solo Ln.
6	Quinton	Boyd	4 Cypress Cr.
7	Alex	Hinton	1011 Hodge Lr
8	Nisha	Hall	123 Huntingto
9	Hillary	Clayton	2516 Newman
10	Kiara	Williams	9014 Miller Ln
11	Katy	Jones	456 Denver Rd
12	Beatrix	Joslin	85 North West
13	Mariah	Allen	12 Jupe

Облачные СУБД

- NuBuilder Forte
- Memento Database
- Airtable (гибрид электронной таблицы и СУБД)



The screenshot displays the Airtable interface for a table named "Favorite Movies". The table contains 10 records, each representing a movie. The columns are: Name, Photo, Seen?, Actors, Director, Genre, Description, Personal Rating, and Personal Notes. The "Genre" column is currently being edited for the "Fight Club" record, with a dropdown menu showing options like Drama, Comedy, Documentary, Romantic Comedy, and Adventure.

	Name	Photo	Seen?	Actors	Director	Genre	Description	Personal Rating	Personal Notes
1	The Godfather		✓	Marlon Brando Al P...	Francis Ford Coppola	Drama	The Godfather is a 1972 American cri...	4: Entertaining 😊	
2	Sister Act			Whoopi Goldberg	Emile Ardolino	Comedy	Sister Act is a 1992 American comed...		
3	Pulp Fiction		✓	Samuel L. Jackson	Quentin Tarantino	Drama	Pulp Fiction is a 1994 American black...	3: Average 😐	
4	Caddyshack			Bill Murray	Zach McBride	Comedy	Caddyshack is a 1980 American sport...		
5	Get Smart		✓	Barbara Feldon Don	Gary Nelson	Comedy	Get Smart is an American comedy ...	2: Mediocre 😞	
6	Forrest Gump		✓	Tom Hanks	Robert Zemeckis	Drama	Forrest Gump is a 1994 American epi...	4: Entertaining 😊	
7	You've Got Mail			Meg Ryan Tom Han	Nora Ephron	Romantic Co...	You've Got Mail is a 1998 American ...		I can't imagine this movie has age
8	Seven Samurai		✓	Takashi Shimura	Akira Kurosawa	Adventure Dram	Seven Samurai is a 1954 Japanese ...	3: Average 😐	
9	Billy Madison			Adam Sandler	Tamra Davis	Romantic Co...	Billy Madison is a 1995 American ...		
10	Fight Club			Brad Pitt	David Fincher	Drama	ht Club is a 1999 film based on the...		

Серверные СУБД

- Для средних и крупных задач, где важна производительность, надежность, кроссплатформенность.
- Для создания приложений требуется более высокая квалификация.

Стек технологий

Как написать простое CRUD веб-приложение, работающее с SQLite/MySQL/PostgreSQL ?

Стек технологий

Как написать простое CRUD веб-приложение, работающее с SQLite/MySQL/PostgreSQL ?

Frontend

HTML, CSS, JavaScript — обязательно.

- Фреймворки типа Vue.js, React — желательно.

Backend

- SQL — обязательно
- PHP или PHP-фреймворк (Laravel, Symfony)
- Или Django (Python), Node.js (JavaScript), Ruby on Rails (Ruby), ASP.NET (C#) и т.д.

Классификация СУБД по способу структурирования данных

1. Реляционные

2. Нереляционные (NoSQL)

Классификация СУБД по способу структурирования данных

1. Реляционные

Самые распространенные и развитые. База данных – совокупность связанных друг с другом двумерных таблиц (строки=записи, столбцы=поля). Каждая запись в таблице имеет одинаковую структуру (набор полей). Стандартный язык SQL для работы с данными.

MySQL, PostgreSQL, Microsoft SQL Server

2. Нереляционные (NoSQL)

Классификация СУБД по способу структурирования данных

1. Реляционные

Самые распространенные и развитые. База данных – совокупность связанных друг с другом двумерных таблиц (строки=записи, столбцы=поля). Каждая запись в таблице имеет одинаковую структуру (набор полей). Стандартный язык SQL для работы с данными.

MySQL, PostgreSQL, Microsoft SQL Server

2. Нереляционные (NoSQL)

- **Хранилища ключей и значений.** Простейшая модель. Данные можно легко распределять в кластере. Для сложных запросов не подходят. Redis, Riak.
- **Столбцовые.** Данные хранятся не по строкам, а по столбцам. Хорошо подходят для BigData. HBase, ClickHouse.
- **Документарные.** Документ – объект, обладающий произвольным набором атрибутов (полей). Коллекция – набор документов (документы могут иметь разную структуру). База данных – совокупность коллекций. MongoDB.
- **Графовые.** Упор на установление произвольных связей между данными. Neo4j.