



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

Базы данных

Лекция 1 Введение в БД

Агафонов Антон Александрович
к.т.н., доцент кафедры ГИИБ

Самара



- Реляционные базы данных
- Целостность реляционных данных
- Проектирование баз данных
- Манипулирование данными средствами SQL
- Определение данных средствами SQL
- Процедурный SQL
- Управление транзакциями
- Работа с пространственными и слабоструктурированными данными
- Работа с документо-ориентированными базами данных





- Понятие базы данных (БД)
- Модели баз данных
- Понятие системы управления базами данных (СУБД)
- Функции СУБД
- Компоненты СУБД





OpenStreetMap –
~1.5 Тб несжатых
данных



~42 Тб данных
(по данным 2015
года)



~500 Тб
климатических
данных





Понятие базы данных

База данных — совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь.

База данных — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины.

База данных — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, которая поддерживает одну или более областей применения.

База данных — совместно используемый набор логически связанных данных и описание этих данных, предназначенный для удовлетворения информационных потребностей организации.





- БД хранится и обрабатывается в вычислительной системе.
- Данные в БД логически структурированы с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе.
- БД включает схему, или метаданные, описывающие логическую структуру БД в формальном виде.
- Потребителем данных может выступать не только человек, но и программно-аппаратные комплексы.

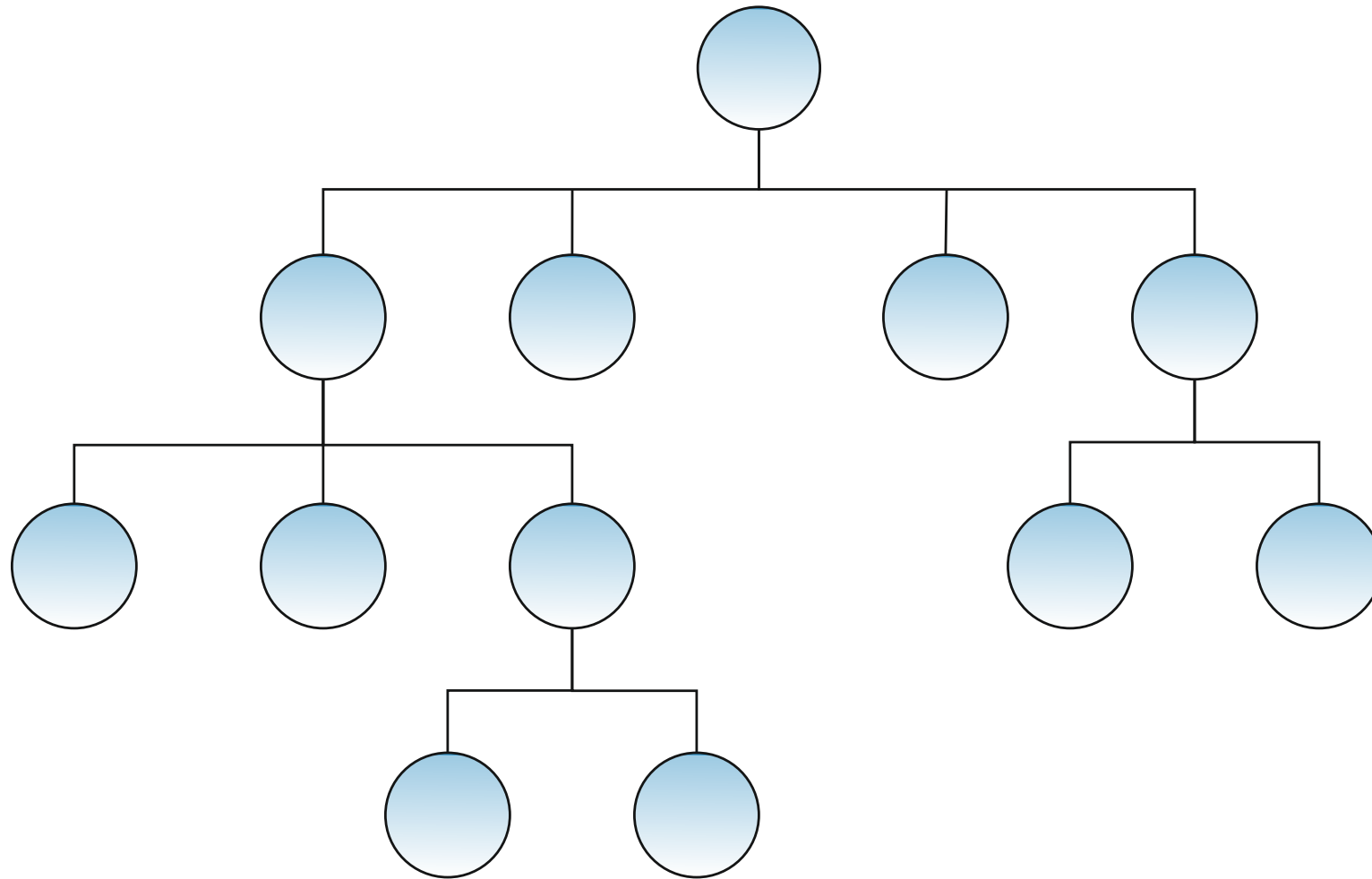




Эволюция моделей БД

Электронные картотеки (до 1965 г.)	<ul style="list-style-type: none">• Системы, основанные на файлах
Первые модели БД (1965-1975 гг.)	<ul style="list-style-type: none">• Иерархическая модель• Сетевая модель
Реляционная модель (с 1970 г. по наши дни)	<ul style="list-style-type: none">• Реляционная модель• Концептуальная модель «сущность-связь»
Развитие реляционной модели (с 1986 г. по наши дни)	<ul style="list-style-type: none">• Улучшенная реляционная модель• Объектно-ориентированная модель
Слабоструктурированные данные (с 1998 г. по наши дни)	<ul style="list-style-type: none">• XML• JSON
Документно-ориентированная модель (с 2008 г. по наши дни)	<ul style="list-style-type: none">• Системы NoSQL





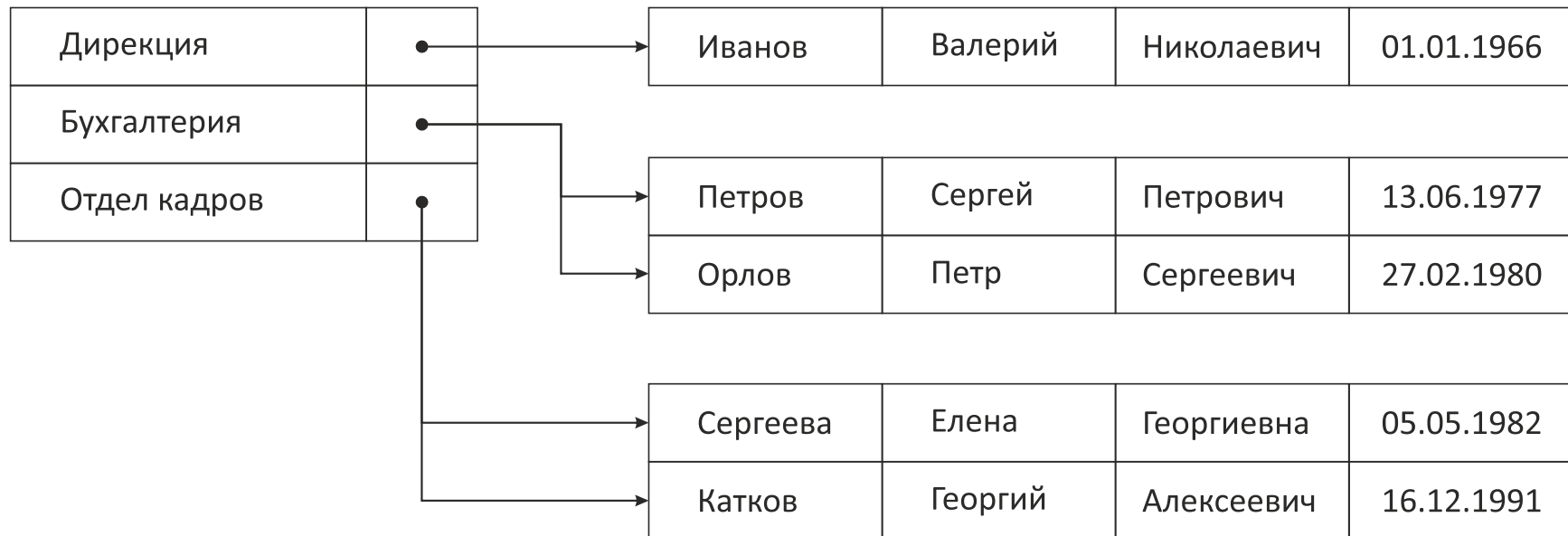


Иерархическая модель БД

Модель



Пример



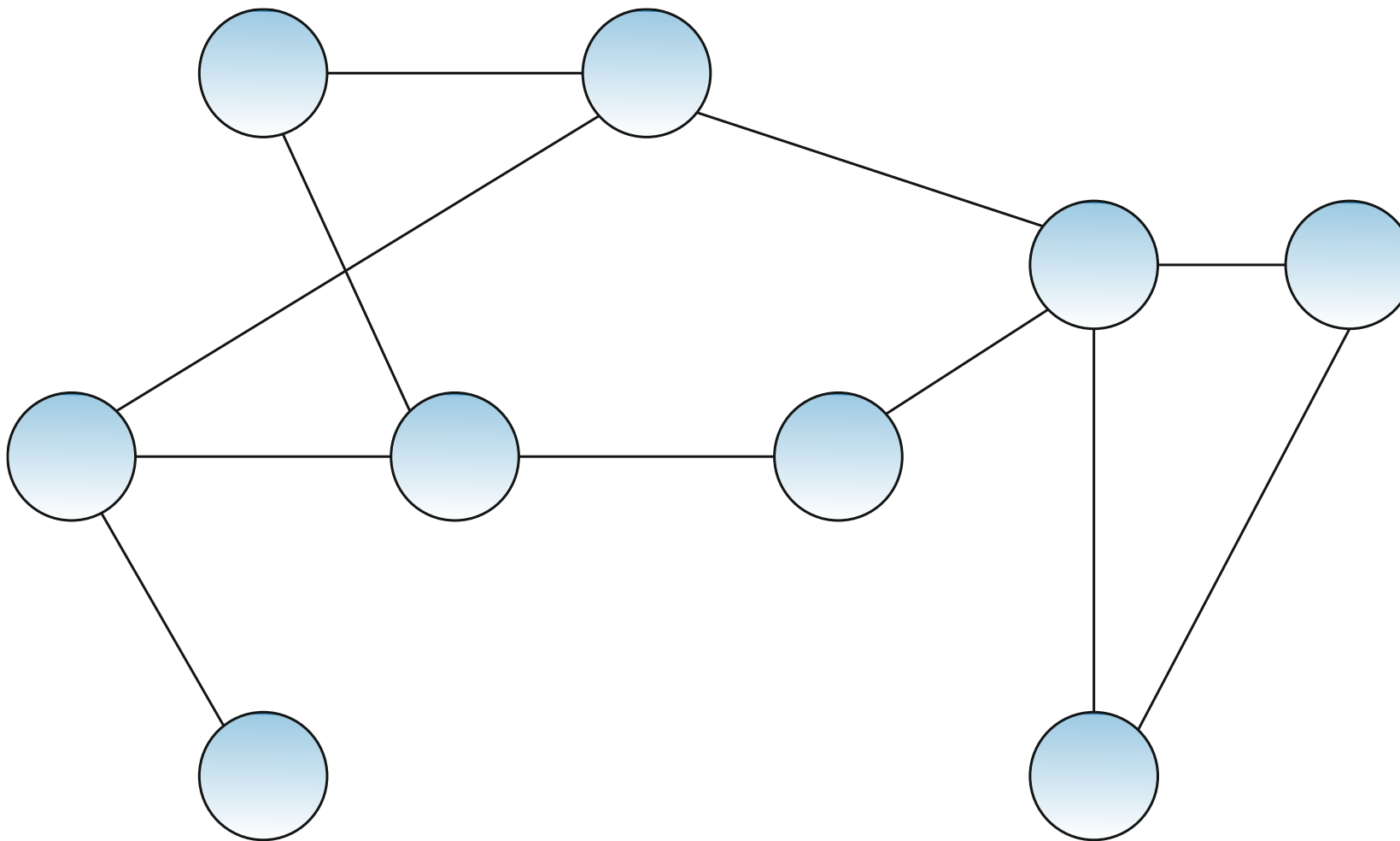


Достоинства:

- ▲ Простота понимания структуры данных
- ▲ Целостность данных
- ▲ Независимость данных
- ▲ Безопасность данных
- ▲ Высокая эффективность алгоритмов поиска

Недостатки:

- ▼ Ограничения в организации отношений между сущностями
- ▼ Структурная зависимость
- ▼ Высокая стоимость операций модификации
- ▼ Отсутствие стандартизации





Достоинства:

- ▲ Произвольное количество связей между узлами
- ▲ Простое удаление записи

Недостатки:

- ▼ Высокая сложность схемы БД
- ▼ Сложность разработки программного обеспечения
- ▼ Отсутствие стандартизации



Реляционная модель данных – логическая модель, основанная на множестве взаимосвязанных именованных отношений.

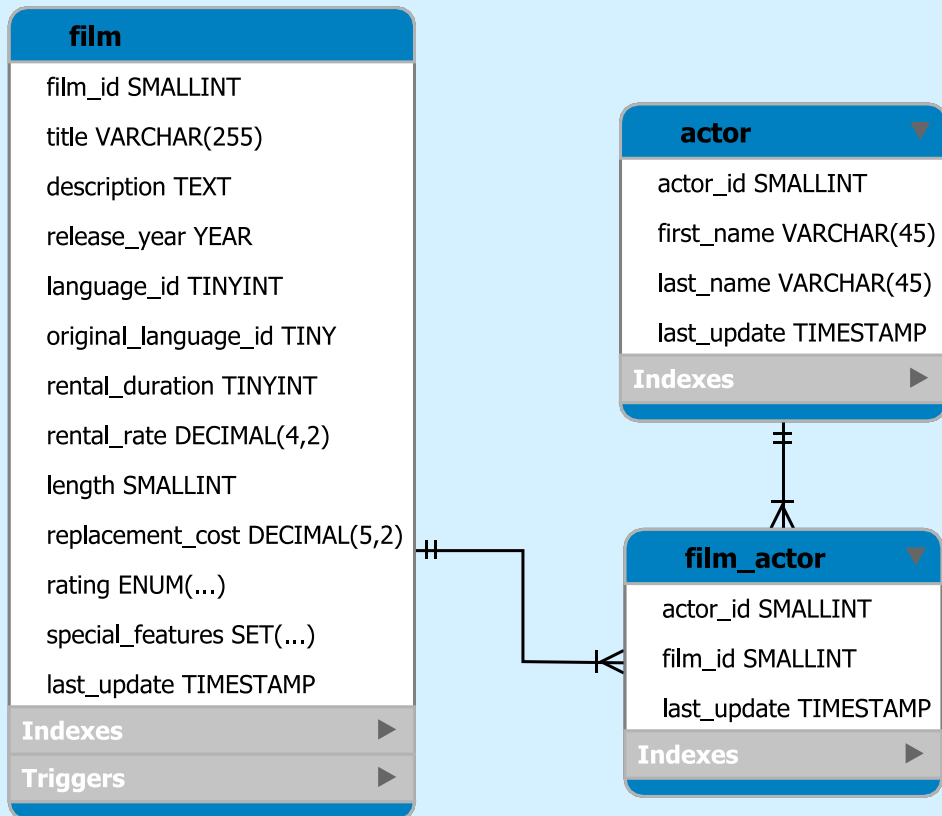
Отношение – это информационная модель реального объекта («сущности») предметной области, формально представленная множеством однотипных кортежей.

Кортеж отношения представляет экземпляр моделируемого объекта, свойства которого определяются значениями соответствующих атрибутов («полей») кортежа.

Связи между кортежами отношений (при их наличии) реализуются через простой механизм «**внешних ключей**», являющихся, по существу, ссылками на атрибуты связываемых кортежей отношений.



БД фильмов



film

film_id	title	description
①	ACADEMY DINOSAUR	A Epic Drama of...
②	ACE GOLDFINGER	A Astounding Epistle of ...
3	ADAPTATION HOLES	A Astounding Reflection of ...
4	AFFAIR PREJUDICE	A Fanciful Documentary of ...
...

film_actor

actor_id	film_id
①	①
1	23
...	...
19	②
85	②
...	...

actor

actor_id	first_name	last_name
①	PENELOPE	GUINESS
2	NICK	WAHLBERG
3	ED	CHASE
...





Реляционная модель данных включает следующие компоненты:

- **Структурный аспект** — данные в базе данных представляют собой набор отношений.
- **Аспект целостности** — отношения отвечают определённым условиям целостности. Реляционная модель данных поддерживает декларативные ограничения целостности уровня типа данных, уровня отношения и уровня базы данных.
- **Аспект обработки (манипулирования)** — реляционная модель данных поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление).





Достоинства:

- ▲ Простота представления структуры БД
- ▲ Высокая гибкость управления данными
- ▲ Высокая эффективность выполнения модифицирующих операций
- ▲ Стандарт SQL

Недостатки:

- ▼ Временные затраты на получение сводных отношений
- ▼ Особенности организации отношения «многие ко многим»



Объектные и объектно-реляционные модели основаны на хранении в базе данных объектов, определяемых в рамках объектной модели языка программирования, их атрибутов, методов и классов.

Достоинства:

- ▲ Способность адекватно описать информационные структуры любой сложности
- ▲ Высокая эффективность доступа к данным

Недостатки:

- ▼ Не удалось создать высокоуровневый декларативный язык запросов
- ▼ Низкая производительность при массовой выборке данных



Слабоструктурированные данные — это форма данных, не соответствующая строгой структуре таблиц и отношений в моделях реляционных баз данных, содержащая теги и другие маркеры для отделения семантических элементов и для обеспечения иерархической структуры записей и полей в наборе данных.

- Широко используются форматы данных XML, JSON
- Тип данных является бессхемным (schemaless), а структура — самоописываемой
- Данные могут храниться как значения атрибутов отношений в объектно-реляционных БД
- Можно сочетать реляционные и слабоструктурированные языки запросов

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Это комментарий-->
<example>
  <line>Пример данных в формате XML</line>
</example>
```

```
{
  "example": {
    "line": "Пример данных в формате JSON"
  }
}
```





Модель ключ-значение основана на хранении пар ключ-значение с использованием ассоциативного массива или словаря.

Достоинства:

▲ Очень высокая производительность

Недостатки:

▼ Очень сильная ограниченность реализуемых функций СУБД

▼ Не поддерживают сложные запросы и агрегирование





Документо-ориентированная БД — база данных, предназначенная для хранения иерархических структур данных (документов) и обычно реализуемая с помощью подхода NoSQL («Not-only SQL»).

Пример записи из БД
MongoDB с информацией о
фильме

```
{
  "_id": {
    "$oid": "573a139bf29313caabcf4dd0"
  },
  "fullplot": "A ticking-time-bomb insomniac and a slippery soap salesman...",
  "imdb": {
    "rating": 8.9,
    "votes": 1191784,
    "id": 137523
  },
  "year": 1999,
  "plot": "An insomniac office worker, looking for a way to change his life...",
  "genres": [ "Drama" ],
  "rated": "R",
  "metacritic": 66,
  "title": "Fight Club",
  "lastupdated": "2015-09-02 00:16:15.833000000",
  "languages": [ "English" ],
  "writers": [ "Chuck Palahniuk (novel)", "Jim Uhls (screenplay)" ],
  "type": "movie",
  "awards": {
    "wins": 11,
    "nominations": 22,
    "text": "Nominated for 1 Oscar. Another 10 wins & 22 nominations."
  },
  "countries": [ "USA", "Germany" ],
  "cast": [ "Edward Norton", "Brad Pitt", "Helena Bonham Carter", "Meat Loaf" ],
  "directors": [ "David Fincher" ]
}
```





Достоинства:

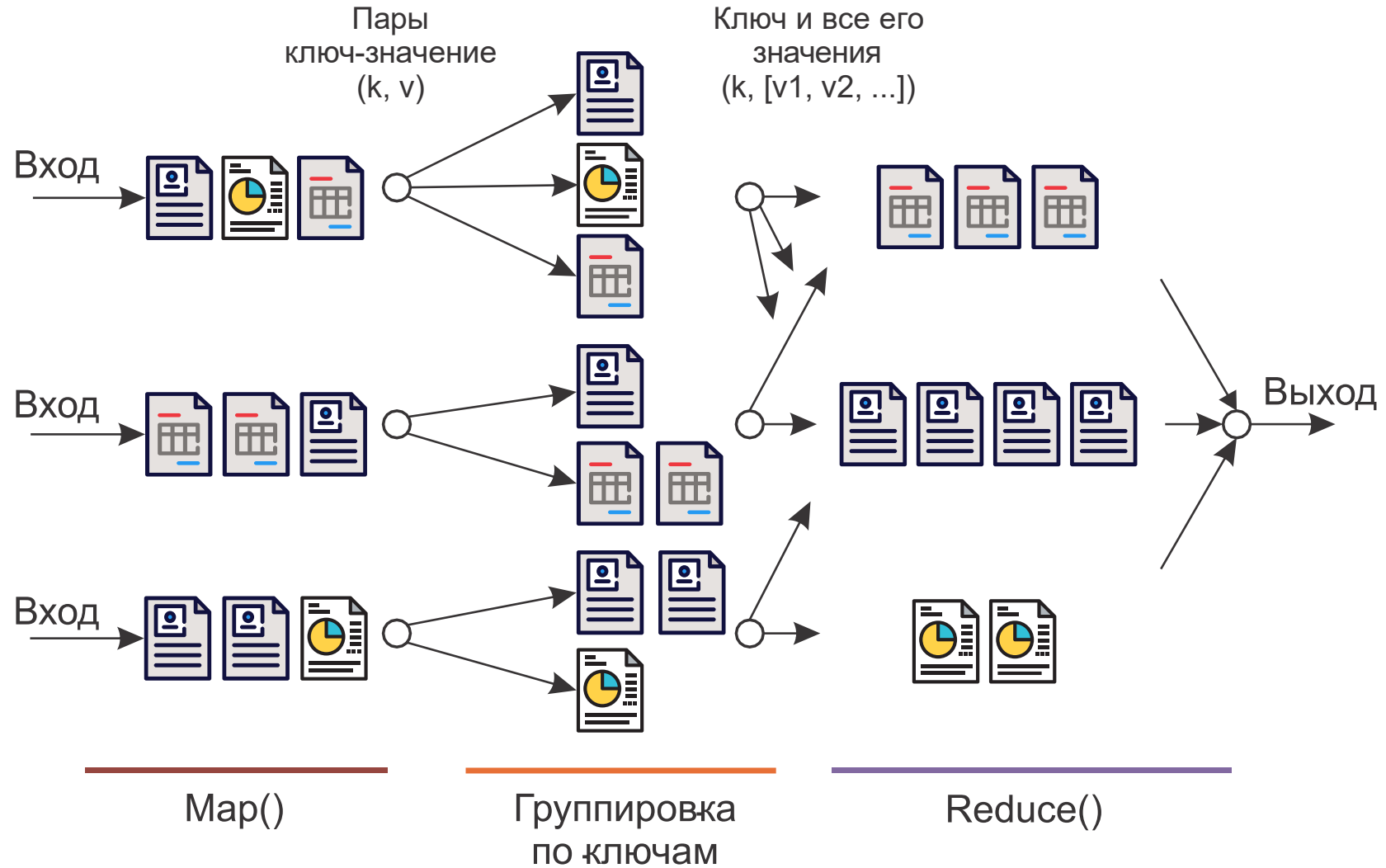
- ▲ Отсутствие затрат на получение сводных отношений
- ▲ Ориентированы на работу с большими массивами данных, находящимися в распределенных системах

Недостатки:

- ▼ Ограниченность синтаксических конструкций языка NoSQL
- ▼ Отсутствие транзакционности в терминах реляционных БД



Распределенная обработка MapReduce





Система управления базами данных (Database Management System, DBMS) – это комплекс программных средств, позволяющих создать базу данных и манипулировать данными. СУБД обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства администрирования доступа к схемам данных и данным.

Основные функции СУБД:

- Управление базами данных
- Предотвращение несанкционированного доступа к данным
- Контроль многопользовательского доступа
- Поддержка целостности данных
- Восстановление данных
- Резервное копирование данных





1. Доступность данных

Предоставление пользователю возможности вставлять, редактировать, удалять и извлекать данные из БД.

В том числе:

- a. Управление данными во внешней памяти*
- b. Управление данными в оперативной памяти*

2. Метаописание данных

- a. Описание хранимых в БД данных
- b. Описание связей между данными
- c. Ограничения целостности данных
- d. Регистрационные данные пользователей и другая служебная информация

3. Управление параллельностью

Реализация механизма одновременного многопользовательского доступа к обрабатываемым данным с гарантией корректного обновления этих данных.



4. Обработка данных в рамках транзакции

СУБД гарантирует, что БД будет всегда находиться в непротиворечивом состоянии вне зависимости от любых сбоев при проведении операций обновления данных. Для этого операции с данными (в первую очередь вставки, редактирования и удаления) объединяются в единый блок, называемый транзакцией.

5. Обеспечение целостности данных

Все содержащиеся в БД данные должны быть корректны и непротиворечивы. В самом общем случае можно говорить о существовании трех правил поддержания целостности данных: целостность доменов, целостность отношений, целостность связей между отношениями.

6. Журнализация изменений, резервное копирование и восстановление базы данных после сбоев

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя.



7. Обмен данными

СУБД обязана поддерживать современные сетевые технологии и предоставлять доступ к БД удаленным персональным компьютерам.

8. Контроль за доступом к данным.

Доступ к данным могут осуществлять только зарегистрированные в СУБД пользователи в соответствии с назначенными администратором СУБД им правами.

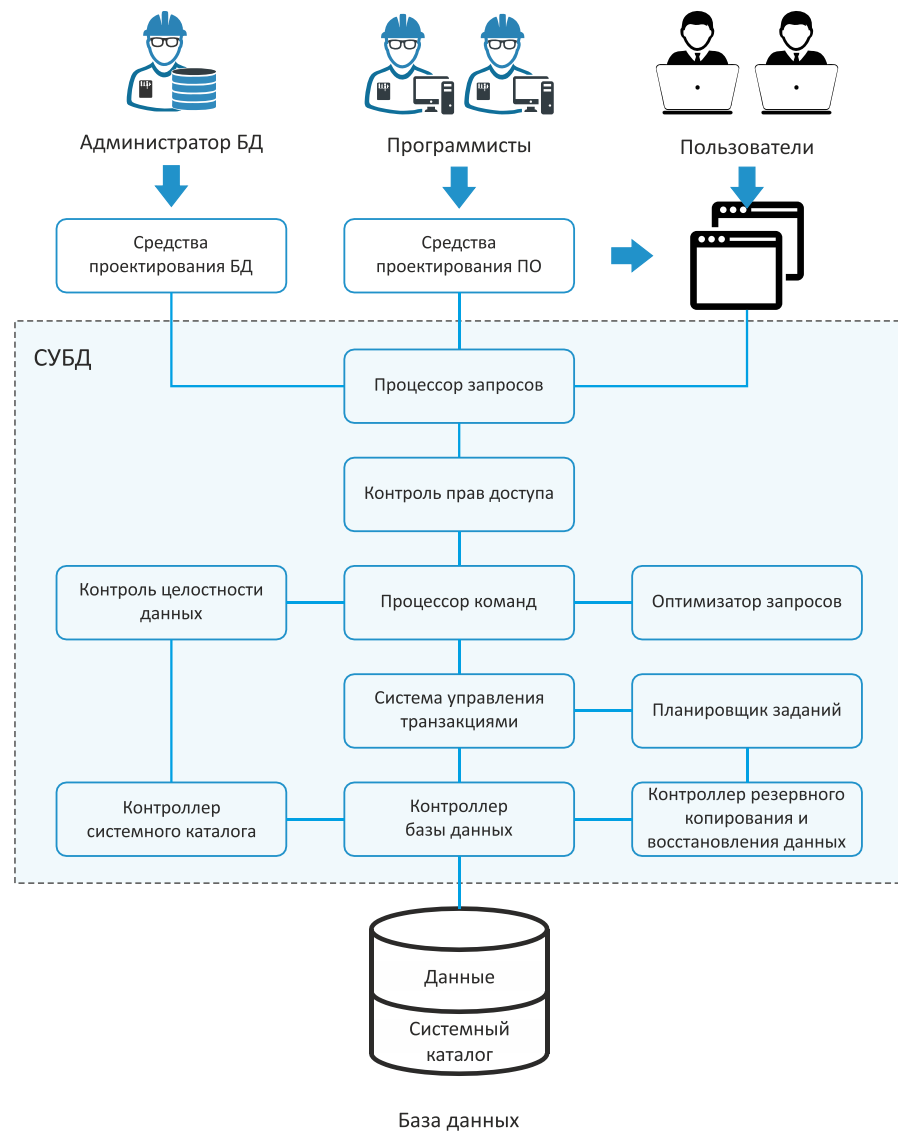
9. Поддержка языков БД

Стандартным языком реляционных СУБД является язык SQL (Structured Query Language) - декларативный язык программирования, применяемый для создания, модификации и управления данными.

- Язык определения данных DDL
- Язык манипулирования данными DML
- Язык контроля данных DCL



Компоненты СУБД





По модели данных:

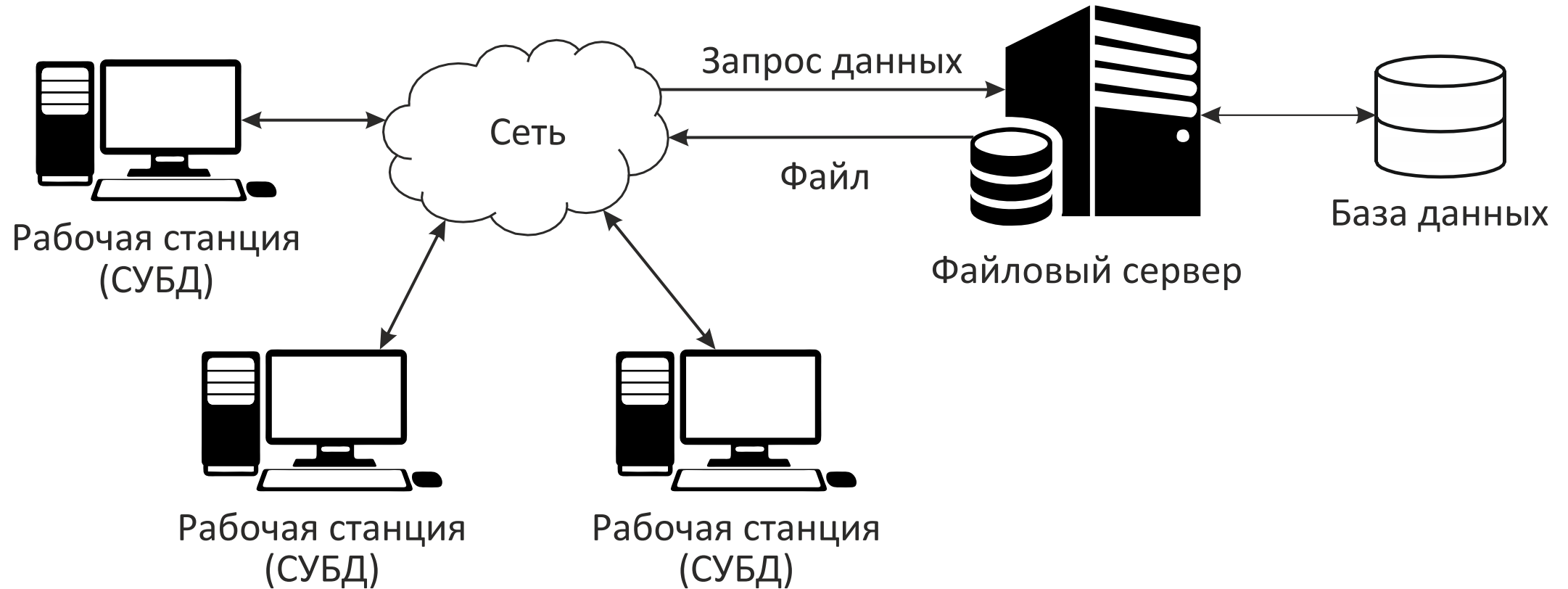
- Иерархические
- Сетевые
- Реляционные
- Объектно-ориентированные
- Объектно-реляционные
- Документо-ориентированные и т.д.

По способу доступа к БД:

- Файл-серверные
- Клиент-серверные
- Встраиваемые

По степени распределенности:

- Локальные
- Распределенные



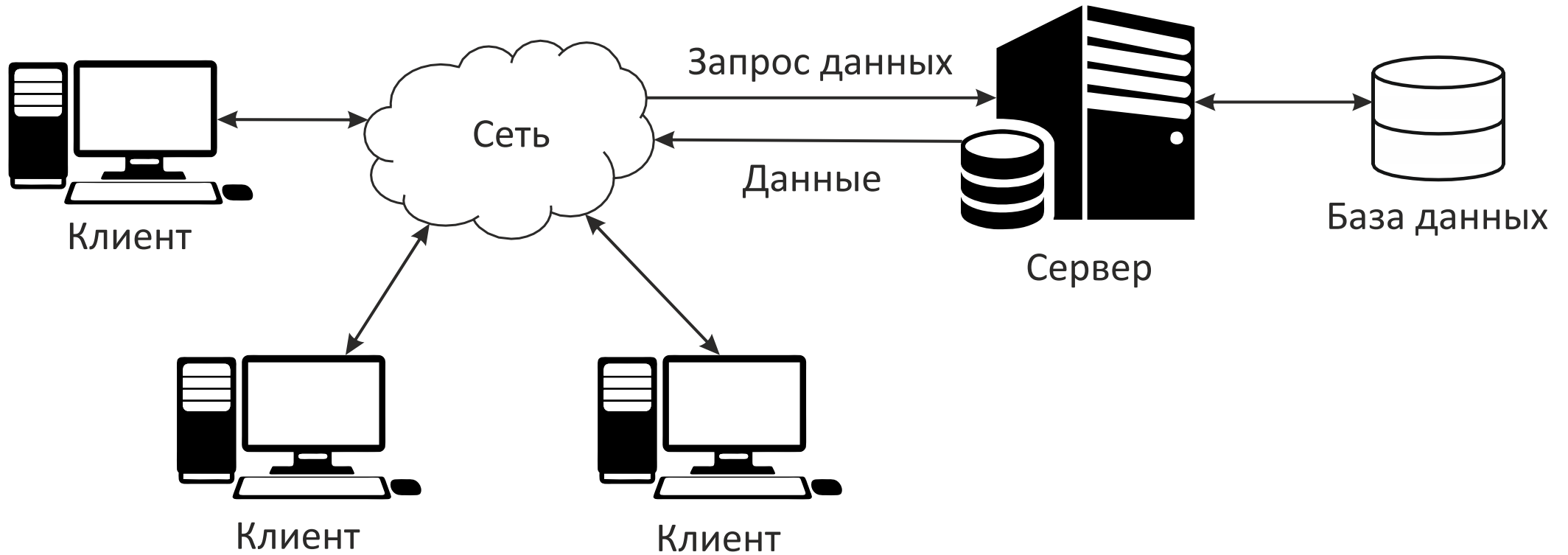


Достоинства:

- ▲ Простота реализации
- ▲ Невысокие требования к файловому серверу

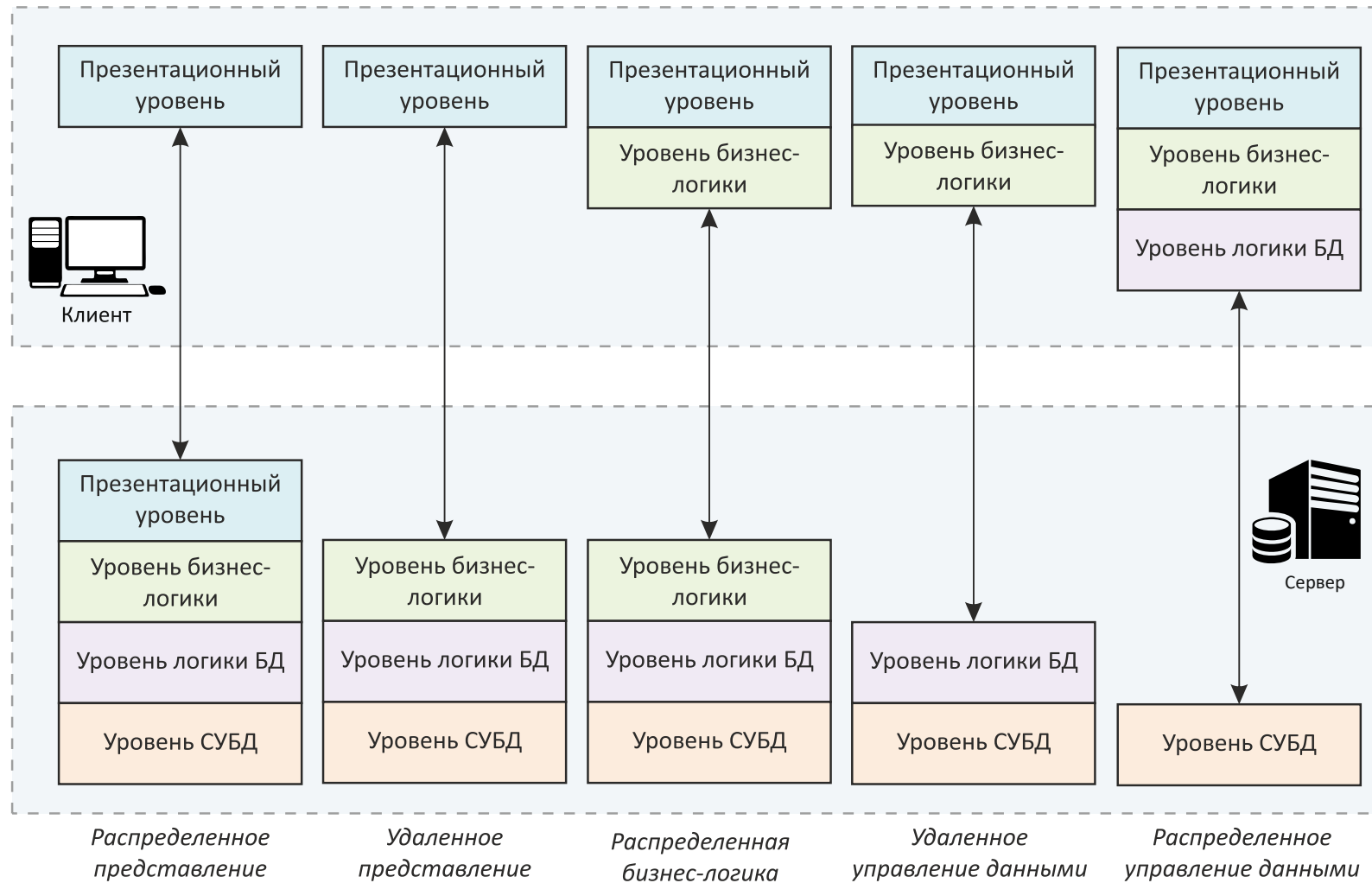
Недостатки:

- ▼ На каждой рабочей станции необходимо устанавливать полную копию СУБД
- ▼ Потенциально высокая загрузка локальной сети
- ▼ Проседание производительности при одновременной работе многих клиентов с одними и теми же данными
- ▼ Высокие требования к клиентским рабочим местам
- ▼ Затруднённость или невозможность централизованного управления
- ▼ Низкий уровень обеспечения целостности и сохранности данных





Клиент-серверные СУБД





Достоинства:

- ▲ Доступность БД
- ▲ Параллельная многопользовательская обработка данных
- ▲ Поддержка целостности данных
- ▲ Потенциально более низкая загрузка локальной сети
- ▲ Удобство обеспечения безопасности
- ▲ Стандарт SQL
- ▲ Удобство обслуживания и администрирования БД

Недостатки:

- ▼ Повышенные требования к производительности сервера



- Является составной частью программного продукта
- Предназначена для локального хранения данных своего приложения
- Чаще всего реализована в виде подключаемой библиотеки
- Высокая скорость и малый расход памяти
- Как правило, нет пользовательских прав
- Как правило, простейшая изоляция транзакций
- Как правило, нет архивации и репликации



- Однопользовательские СУБД:
 - SQLite
 - Microsoft Access
- Многопользовательские реляционные СУБД:
 - Oracle
 - Microsoft SQL Server
 - MySQL
 - PostgreSQL
- СУБД типа «ключ-значение»:
 - Amazon DynamoDB
 - Apache Cassandra
- Документо-ориентированные СУБД:
 - MongoDB
 - CouchDB



САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

**БЛАГОДАРЮ
ЗА ВНИМАНИЕ**

Агафонов А.А.
к.т.н., доцент кафедры ГИИБ