

Лекция 1 - Основы систем управления базами данных (СУБД)

СУБД — это программное обеспечение для создания, управления и манипуляции данными. Оно позволяет пользователю взаимодействовать с базой данных, выполняя запросы и обновления.

Исторический экскурс: от простых систем файлового хранения к современным СУБД. Обсуждение первых СУБД, таких как IBM System R и Oracle.

Виды СУБД:

1. Реляционные СУБД (RDBMS) используют структурированные таблицы и язык SQL для управления данными. Примеры: Oracle, MySQL, PostgreSQL.
2. Нереляционные СУБД (NoSQL) поддерживают более гибкие схемы данных. Примеры: MongoDB (документо-ориентированные), Cassandra (широкие столбцы), Redis (ключ-значение).

Основные компоненты и принципы

Архитектура СУБД:

1. Описание клиент-серверной модели: клиенты выполняют запросы к серверу, который управляет базой данных.
2. Многоуровневая архитектура: представление, логика и хранение данных.

Модели данных:

1. Реляционная модель: таблицы со строками и столбцами, ключи, нормализация данных для избежания дублирования и обеспечения целостности.
2. Нереляционные модели: гибкость в хранении данных, примеры документо-ориентированных и графовых баз данных.

Язык запросов SQL:

1. Базовые операции SQL: SELECT для извлечения, INSERT для вставки, UPDATE для обновления, DELETE для удаления.
2. Расширенные возможности SQL: объединение (JOIN), подзапросы, агрегатные функции.

Расширенные темы

Транзакции и управление параллелизмом:

1. Концепция транзакций в СУБД, свойства ACID (атомарность, согласованность, изоляция, долговечность).
2. Механизмы управления параллелизмом, такие как блокировки и многоверсионное управление параллелизмом (MVCC).

Оптимизация запросов и производительность:

1. Важность индексов в оптимизации запросов.
2. Инструменты и методы для мониторинга и улучшения производительности СУБД.

Безопасность и резервное копирование:

1. Основы безопасности в СУБД: управление доступом, шифрование данных.
2. Стратегии резервного копирования и восстановления данных.

Примеры из реальной практики

Использование СУБД в крупных проектах:

1. Анализ кейсов использования СУБД в крупных IT-компаниях, обсуждение различных сценариев применения.
2. Как выбрать подходящую СУБД в зависимости от требований проекта: производительность, масштабируемость, надежность.

Тренды и будущее СУБД:

Влияние облачных технологий на СУБД, обзор облачных баз данных как сервиса (DBaaS).

Изучение новых тенденций и инноваций в области СУБД, таких как распределенные базы данных и машинное обучение.

Лекция 1.2. Технологии работы с базами данных

План:

1. Централизованная архитектура
2. Архитектура "файл-сервер"
3. Технология "клиент – сервер"
4. Трехзвенная (многозвенная) архитектура "клиент – сервер"

Понятие базы данных изначально предполагало возможность решения многих задач несколькими пользователями. В связи с этим, важнейшей характеристикой современных СУБД является наличие

многопользовательской технологии работы. Разная реализация таких технологий в разное время была связана как с основными свойствами вычислительной техники, так и с развитием программного обеспечения.

Централизованная архитектура

При использовании этой технологии база данных, СУБД и прикладная программа (приложение) располагаются на одном компьютере (рисунок 1). Для такого способа организации не требуется поддержки сети и все сводится к автономной работе.

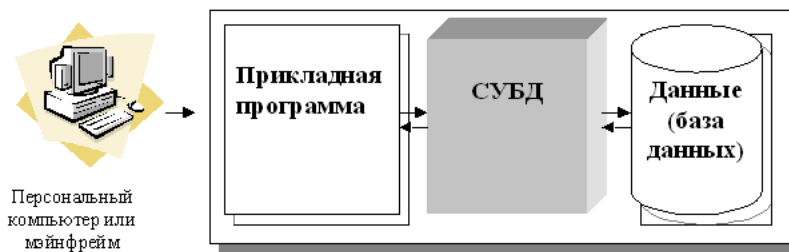


Рисунок 1 - Централизованная архитектура

Многопользовательская технология работы обеспечивалась либо режимом мультипрограммирования, либо режимом разделения времени.

Архитектура "файл-сервер"

Увеличение сложности задач, появление персональных компьютеров и локальных вычислительных сетей явились предпосылками появления новой архитектуры *файл-сервер*. Эта архитектура баз данных с сетевым доступом предполагает назначение одного из компьютеров сети в качестве выделенного сервера, на котором будут храниться файлы базы данных. В соответствии с запросами пользователей файлы с *файл-сервера* передаются на рабочие станции пользователей, где и осуществляется основная часть обработки данных. Центральный сервер выполняет в основном только роль хранилища файлов, не участвуя в обработке самих данных (рисунок 2).

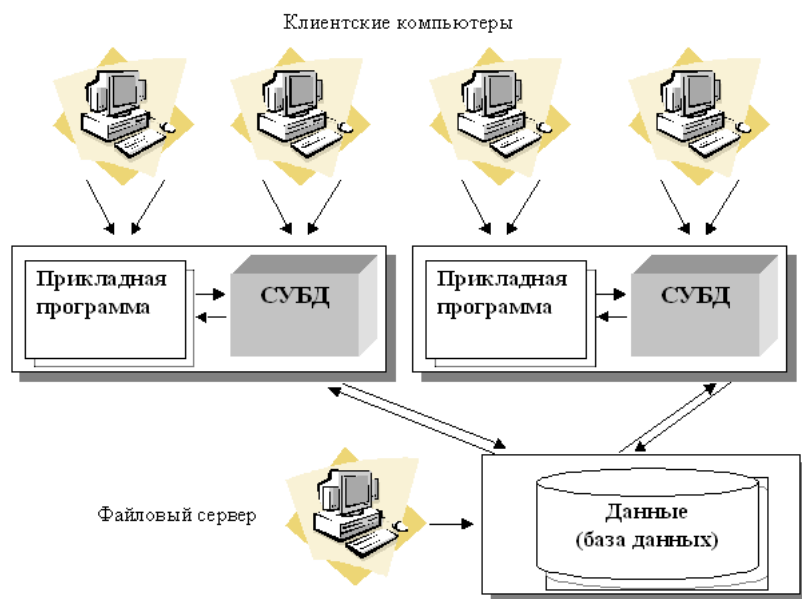


Рисунок 2 - Архитектура "файл-сервер"

Технология "клиент – сервер"

Использование технологии "*клиент – сервер*" предполагает наличие некоторого количества компьютеров, объединенных в сеть, один из которых выполняет особые управляющие функции (является сервером сети).

Так, архитектура "*клиент – сервер*" разделяет функции приложения пользователя (называемого клиентом) и сервера. Приложение-клиент формирует запрос к серверу, на котором расположена БД, на структурном языке запросов SQL (Structured Query Language), являющемся промышленным стандартом в мире реляционных БД. Удаленный сервер принимает запрос и переадресует его SQL-серверу БД.

SQL-сервер – специальная программа, управляющая удаленной базой данных. SQL-сервер обеспечивает интерпретацию запроса, его выполнение в базе данных, формирование результата выполнения запроса и выдачу его приложению-клиенту. При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса; клиентский компьютер лишь отправляет запрос к серверной БД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю. Так

как клиентскому приложению посылается результат выполнения запроса, по сети "путешествуют" только те данные, которые необходимы клиенту. В итоге снижается нагрузка на сеть. Поскольку выполнение запроса происходит там же, где хранятся данные (на сервере), нет необходимости в пересылке больших пакетов данных. Кроме того, SQL-сервер, если это возможно, оптимизирует полученный запрос таким образом, чтобы он был выполнен в минимальное время с наименьшими накладными расходами.

Архитектура системы представлена на рисунке 3.

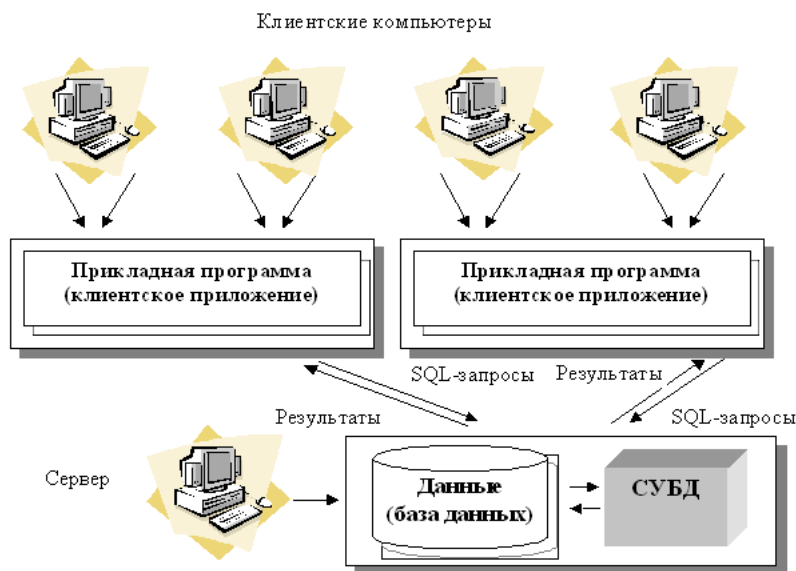


Рисунок 3 - Архитектура "клиент – сервер"

Трехзвенная (многозвенная) архитектура "клиент – сервер"

Трехзвенная (в некоторых случаях *многозвенная*) **архитектура** (N-tier или multi-tier). представляет собой дальнейшее совершенствование технологии "клиент – сервер". Рассмотрев архитектуру "клиент – сервер", можно заключить, что она является 2-звенной: первое звено – клиентское приложение, второе звено – сервер БД + сама БД. В *трехзвенной архитектуре* вся бизнес-логика (деловая логика), ранее входившая в клиентские приложения, выделяется в отдельное звено, называемое сервером приложений. При этом клиентским приложениям остается лишь пользовательский интерфейс.

Схематически такую *архитектуру* можно представить, как показано на рисунке 4.

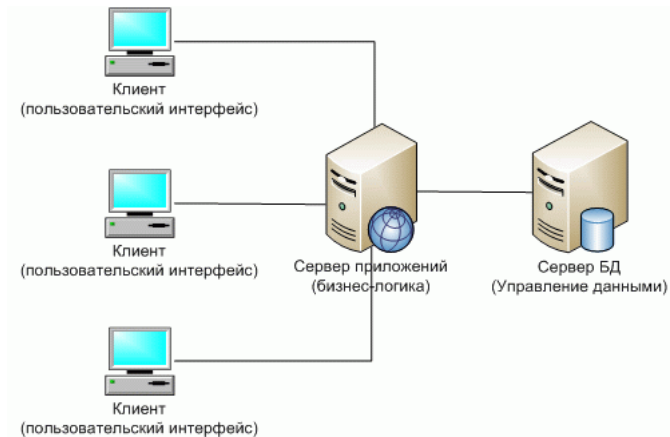


Рисунок 4 - Представление многоуровневой архитектуры "клиент-сервер"

Вопросы для самоконтроля:

1. Назовите достоинства и недостатки существующих многопользовательских технологий с базами данных.