**ЗМІСТ**

[**1 ЗАГАЛЬНИЙ РОЗДІЛ** 2](#_Toc113794069)

[**1.1 Клиент серверная архитектура** 2](#_Toc113794070)

[**1.2 Базы данных, их типы и использование** 5](#_Toc113794071)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 6](#_Toc113794072)

# **1 ЗАГАЛЬНИЙ РОЗДІЛ**

# **1.1 Клиент серверная архитектура**

Клиент-серверная архитектура, архитектура компьютерной сети, в которой множество клиентов (удаленных процессоров) запрашивают и получают услуги от централизованного сервера (хост-компьютера).

Клиент – локальный компьютер на стороне виртуального пользователя, который выполняет отправку запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

Клиентские компьютеры предоставляют интерфейс, позволяющий пользователю компьютера запрашивать услуги сервера и отображать результаты, возвращаемые сервером.

Сервер – компьютер или специальное системное оборудование. Он обслуживает запросы клиента, предоставляет доступ к тем или иным данным, изменяет или добавляет данные в базу данных.

Особенности такой модели заключаются в том, что пользователь отправляет определенный запрос на сервер, где тот системно обрабатывается и конечный результат отсылается клиенту. В возможности сервера входит одновременное обслуживание сразу нескольких клиентов.

Если одновременно поступает более одного запроса, то такие запросы устанавливаются в определенную очередь и сервером выполняются по очереди. Порой запросы могут иметь свои собственные приоритеты. Часть запросов с более высокими приоритетами будут постоянно выполняться в первоочередном порядке!

Процесс распределения операций может существенным образом снизить нагрузку на используемый сервер.

Параметры, которые могут реализоваться на стороне сервера:

* хранение, защита и доступ к данным;
* работа с поступающими клиентскими запросами;
* процесс отправки ответа клиенту.

Параметры, которые могут реализоваться на стороне клиента:

* площадка по предоставлению пользовательского графического интерфейса;
* формулировка запроса к серверу и его последующая отправка;
* получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

Архитектура системы клиент-сервер формулирует принципы виртуального общения между локальными компьютерами, а все правила и принципы взаимодействия находятся внутри протокола.

Сетевой протокол – это особый набор правил, на основании которого выполняется точное взаимодействие между компьютерами внутри виртуальной сети.

Виды сетевых протоколов:

* TCP/IP – это особое обозначение всей сети, которая функционирует на основе протоколов TCP, а также IP.
* TCP – вид протокола, который является связующим звеном для установки качественного соединения между 2 устройствами, передачи данных и верификации их получения;
* IP – протокол, в функции которого входит корректность доставки сообщений по выбранному адресу. При этом информация делится на пакеты, которые могут поставляться по-разному;
* MAC – вид протокола, на основании которого происходит процесс верификации сетевых устройств. Все устройства, которые подключены к сети Интернет, содержат свой оригинальный MAC-адрес;
* ICMP – протокол, который ответственен за обмен данными, но не используется для процесса передачи информации;
* UDP – протокол, управляющий передачей данных, но данные не проходят верификацию при получении. Этот протокол функционирует быстрее, чем протокол TCP;
* HTTP – протокол для передачи информации (гипертекста), на базе которого функционируют все сегодняшние сайты. В его возможности входит процесс запрашивания необходимых данных у виртуально удаленной системы (файлы, веб-страницы и прочее);
* FTP – протокол передачи информации из особого файлового сервера на ПК конечного пользователя;
* POP3 – классический протокол простого почтового соединения, который ответственен за передачу почты;
* SMTP – вид протокола, который может устанавливать правила для передачи виртуальной почты. Он ответственен за передачу и верификацию доставки, а также оповещения о возможных ошибках.

Есть сразу 2 вида клиент-серверных архитектур:

* двухуровневая
* трехуровневая

Двухуровневая состоит из 2 узлов:

* сервер, который ответственен за получение входящих запросов и отправку ответа пользователю, применяя при этом собственные ресурсы системы;
* клиент, который может предоставлять пользовательский графический интерфейс.

Особенности работы заключаются в том, что на сервер приходит определенный запрос, потом его обрабатывают и дают напрямую, без дополнительного применения группы внешних ресурсов.

Трехуровневая система состоит из использования таких компонентов:

* предоставление информации – графический пользовательский интерфейс, прикладной объект в виде сервера приложения;
* менеджмент ресурсов – сервер БД, который может предоставлять данные.

Особенность работы состоит в том, что сразу несколько серверов могут обрабатывать клиентские запросы. Процесс распределения операций может существенным образом снизить нагрузку на используемый сервер.

Трехуровневая архитектура может трансформироваться до многоуровневой, установкой группы дополнительных серверов. Подобная виртуальная архитектура позволяет существенным образом повысить эффективность функционирования информационных систем, а также выполнить оптимизированное распределение части ее программно-аппаратных ресурсов.

# **1.2 Базы данных, их типы и использование**

База данных – организованный набор структурированной информации (данных), предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов. Используется в том числе в клиент-серверной архитектуре. База данных управляются системой управления базами данных.

Система управления базами данных (СУБД) – это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов, формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

Structured Query Language (SQL) – язык структурированных запросов, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Типы баз данных:

* реляционные (табличные) – данные организованы в виде таблиц, состоящих из столбцов и строк. Реляционная СУБД обеспечивает быстрый и эффективный доступ к структурированной информации.;
* иерархические (древовидные) – каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок имеет несколько потомков;
* сетевые (графоподобные) – хранит данные в контексте сущностей и связей между ними;
* объектно-ориентированные – данные моделируются в виде объектов, их атрибутов, методов и классов;
* NoSQL – вся информация хранится без четко установленной структуры и явных связей между всеми данными.

Сегодня предприятия имеют доступ к большему количеству данных, чем когда-либо прежде. Организации используют базы данных, чтобы выйти за рамки базового хранения данных для анализа в огромных объемах из нескольких систем. Используя базу данных, организации теперь могут более эффективно обрабатывать данные, повышая гибкость и масштабируемость данных.

Наиболее распространенными СУБД являются:

* MySQL/MariaDB;
* PostgreSQL;
* MongoDB;
* Oracle;
* Microsoft SQL Server.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Какойто список литературы потом придумаю я не знаю что тут писать по этому наспамлю ссылками все равно это не окончательный вариант
2. <https://testmatick.com/ru/osnovnye-ponyatiya-i-osobennosti-klient-servernoj-arhitektury/>
3. <https://www.britannica.com/technology/client-server-architecture>
4. <https://www.oracle.com/database/what-is-database/>

всем спасибо всем пока жду свою пятор очку а пока я могу пойти там не знаю поспать штоле пака