DBMS

Overview -

DBMS (Database Management System) is a software system that manages the storage, organization, and retrieval of data stored in a database. The main purpose of a DBMS is to provide a centralized and organized approach to managing and manipulating data, to ensure data consistency and integrity, and to provide a seamless interaction between the users and the data. Some popular examples of DBMS are MySQL, Oracle, MS SQL Server, and MongoDB.

Database -

A database is an organized collection of structured data stored and accessed electronically. It can be thought of as an electronic filing system for storing, managing, and retrieving large amounts of information. Databases are used in a wide range of applications, including record-keeping, financial management, online shopping, and social media. They are designed to be flexible, scalable, and reliable, and to provide users with fast access to the information they need.

There are several types of databases:

1. Relational databases: Store data in tables with relationships between them, e.g., MySQL, Oracle
2. NoSQL databases: Store data in different ways other than tables, e.g., MongoDB, Cassandra
3. Graph databases: Store data as nodes and edges representing relationships, e.g., Neo4j
4. Time-series databases: Optimized for storing time-stamped data, e.g., InfluxDB
5. Key-value databases: Store data as collections of key-value pairs, e.g., Redis, Riak
6. Object-oriented databases: Store data as objects, e.g., MongoDB, Apache Cassandra.

Each type of database has its strengths and weaknesses, and the choice of a database depends on the specific requirements of the application.

RDBMS -

RDBMS stands for Relational Database Management System. It is a type of database management system that is based on the relational model, where data is stored in tables and can be easily related to each other through a unique identifier, such as a primary key. RDBMSs provide users with an easy way to manage large amounts of structured data, and are widely used for a variety of applications, including enterprise resource planning, customer relationship management, and data warehousing. Examples of RDBMSs include MySQL, Oracle, and Microsoft SQL Server.

Acid Properties -

ACID stands for Atomicity, Consistency, Isolation, and Durability. These are the properties that ensure the integrity and reliability of a database system:

1. Atomicity: This property ensures that a transaction is treated as a single, indivisible unit of work. Either all changes made within the transaction are committed, or none are.

2. Consistency: This property ensures that transactions bring the database from one valid state to another. The database must satisfy all constraints and business rules.

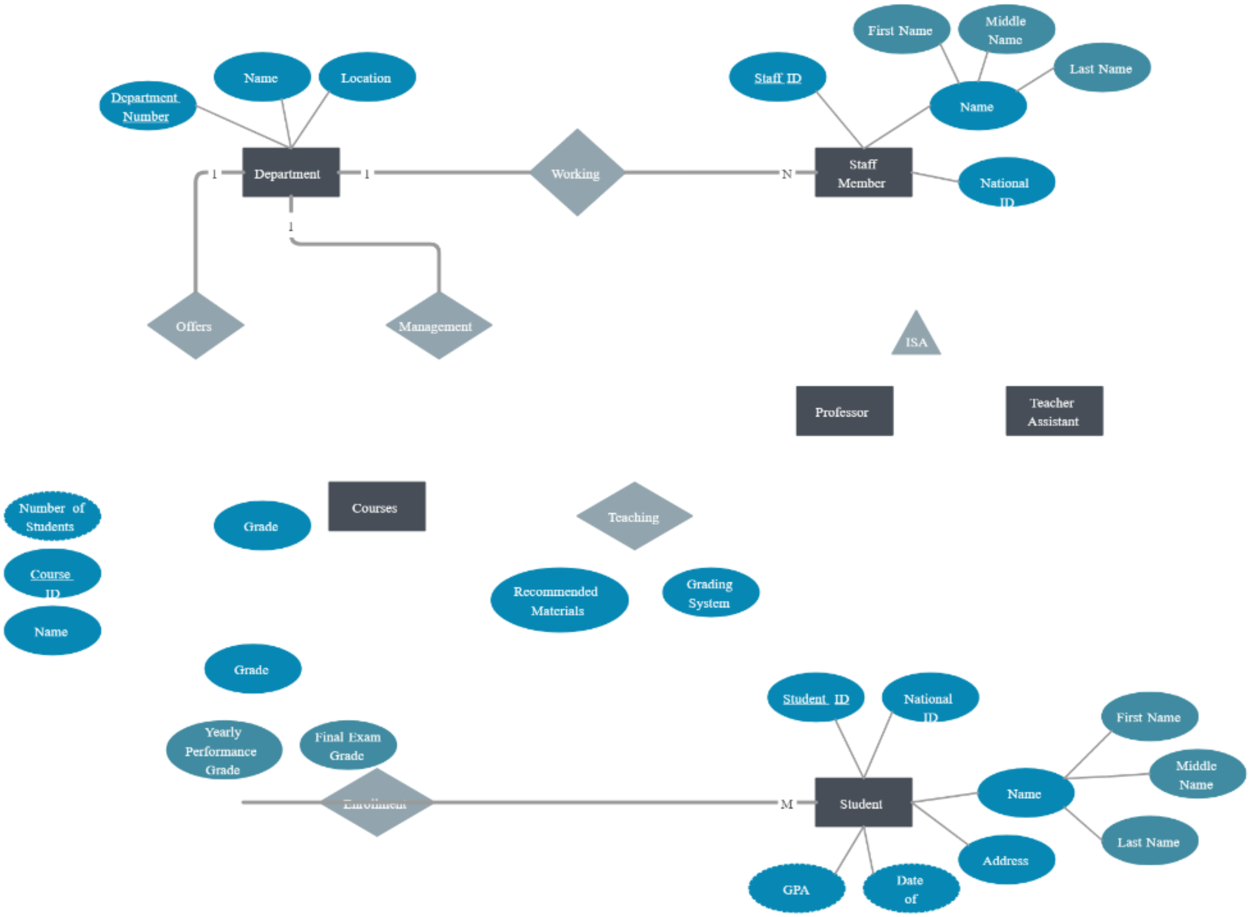
3. Isolation: This property ensures that transactions are executed as if they are running in isolation from each other, even if multiple transactions are executing concurrently. Each transaction should be unaware of the others.

4. Durability: This property ensures that the changes made by a transaction are permanent and survive subsequent system failures. They are stored in a durable manner, such as in a non-volatile memory or on disk.

ER Diagram -

An ER (Entity-Relationship) diagram is a graphical representation of entities and their relationships to each other, used in database design to illustrate an organization's data and information structure. An ER diagram shows the relationship between entities as an entity set, the attributes of an entity, relationships, and the constraints that must be satisfied by the data in the database. It helps in visualizing and clarifying the relationships between entities and the flow of data within an organization's database.

Eg-



In this example, the entities are "Student," "Department," "Faculty," "Course," and "Enrollment." The relationships between the entities are shown by lines connecting them.

- The "Student" entity is related to the "Enrollment" entity through a many-to-many relationship, indicated by the diamond shape. This means that a student can enroll in multiple courses, and a course can have multiple students.
- The "Department" entity is related to the "Faculty" entity through a one-to-many relationship, indicated by the arrow pointing from "Department" to "Faculty." This

means that each department can have multiple faculty members, but each faculty member can only belong to one department.

- The "Faculty" entity is related to the "Course" entity through a many-to-many relationship, indicated by the diamond shape. This means that a faculty member can teach multiple courses, and a course can be taught by multiple faculty members.
- The "Department" entity is related to the "Course" entity through a one-to-many relationship, indicated by the arrow pointing from "Department" to "Course." This means that each department can offer multiple courses, but each course can only be offered by one department.

Normalisation -

Normalization is a process in database design that aims to minimize data redundancy and improve data consistency by organizing data into separate tables based on their logical relationships. The objective of normalization is to eliminate data duplication and make data management more efficient by creating a set of tables with well-defined relationships between them. Normalization is usually done in a series of steps, each step being a higher level of normalization. The most commonly used levels of normalization are first normal form (1NF), second normal form (2NF), third normal form (3NF), and so on.

SQL -

SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases. It is used to create, modify and query databases, as well as perform data manipulation and aggregation tasks. SQL statements can be used to perform operations like inserting, updating and deleting records in a database, retrieving data from a database, and creating tables, indexes, and other database objects.

Keys -

Keys are an important concept in SQL, as they are used to establish relationships between tables. There are several types of keys in SQL:

1. Primary Key: A primary key is a unique identifier for each record in a table. There can only be one primary key per table, and it cannot contain null values.

2. Foreign Key: A foreign key is a field in one table that is linked to the primary key of another table. It is used to establish a relationship between the two tables.
3. Candidate Key: A candidate key is a set of one or more fields that can serve as a primary key. There can be multiple candidate keys in a table.
4. Composite Key: A composite key is a primary key that is made up of two or more columns.
5. Surrogate Key: A surrogate key is a unique identifier that is generated by the database, and is used as the primary key for a table.
6. Natural Key: A natural key is a key that is made up of one or more columns that naturally identify a record in a table, such as a Social Security number or a passport number.

Joins -

Joins in databases are operations that combine rows from two or more tables based on a related column between them. The result of a join is a single table that includes columns from all joined tables, with each row representing a combination of related rows from each table. There are different types of joins, including inner join, left join, right join, and full outer join, each with different behaviour for combining rows and handling unmatched rows from one or both tables. Joins are used to extract meaningful information from multiple tables and are an important part of relational database design and querying.

Joins are used to combine data from two or more tables based on a common column between them. In relational databases, there are mainly four types of joins:

1. INNER JOIN: An inner join returns only the rows that have matching values in both tables. It is the most common type of join and it returns only the matching records from both tables.
2. LEFT JOIN (or LEFT OUTER JOIN): A left join returns all the rows from the left table and the matching rows from the right table. If there is no match, NULL values are returned for the right table's columns.
3. RIGHT JOIN (or RIGHT OUTER JOIN): A right join is similar to a left join, but it returns all the rows from the right table and the matching rows from the left table. If there is no match, NULL values are returned for the left table's columns.

4. FULL JOIN (or FULL OUTER JOIN): A full join returns all the rows from both tables, regardless of whether there is a match between the tables. If there is no match, NULL values are returned for the missing columns.

Each type of join has its own use cases, and the choice of join type depends on the data and the desired outcome.

Important Questions -

Some popular and important questions asked on DBMS in technical interviews are:

1. Explain the difference between RDBMS and NoSQL databases.
2. What is normalization, and why is it important?
3. What are ACID properties and explain each of the properties?
4. Explain the SQL join operation, and discuss the different types of joins.
5. What is indexing and why is it important?
6. What is transaction management and explain the transaction states?
7. What are referential integrity constraints and explain how they ensure data consistency in a database?
8. What are some commonly used SQL commands such as SELECT, INSERT, UPDATE and DELETE?
9. Explain how you would optimize a slow SQL query.
10. What is a relational database management system, and what are its benefits?

These are just a few of the important questions related to DBMS that are commonly asked in technical interviews.

Operating System

Overview -

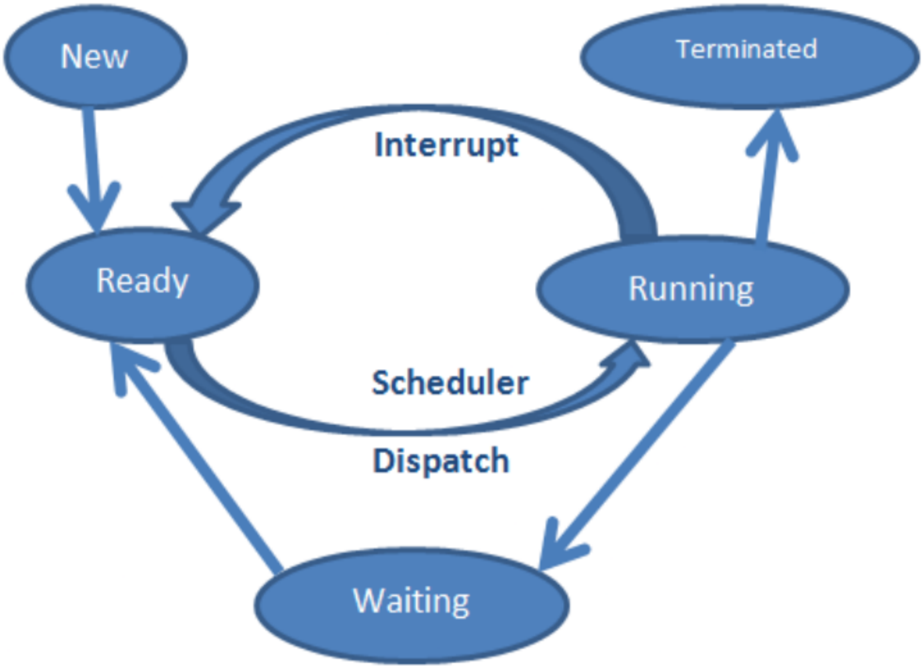
OS stands for Operating System. It is a software program that manages the hardware resources of a computer and provides a platform for the execution of application software. It acts as an intermediary between the computer's hardware and the user or applications. Some of the common functions of an operating system include managing memory and processing, controlling input and output, scheduling tasks, and providing a file system. Examples of popular operating systems include Microsoft Windows, Apple macOS, and Linux.

Processes and Threads -

A process is a program in execution, a dynamic instance of a computer program. It is an independent program that, while running, interacts with the system and other processes. Each process has its own memory space and system resources, and runs in isolation from other processes.

A thread, on the other hand, is a lightweight and independent execution unit within a process. It shares the same memory space and system resources as the process, but can be scheduled by the operating system independently. Threads are used to increase the efficiency of a program by allowing multiple tasks to run concurrently.

In terms of states, a process can be in one of the following states: New, Running, Waiting, Ready, and Terminated.



Deadlocks -

Deadlocks are a situation in operating systems where two or more processes are blocked indefinitely, waiting for each other to release a resource. In this situation, none of the processes can proceed and the system is stuck in a state of deadlock. This can happen when multiple processes hold a resource that another process needs and each process is waiting for the other to release the resource. Deadlocks are a major issue in multi-process systems and can be prevented through careful design and programming.

Deadlocks can be prevented or resolved using several algorithms such as:

1. Prevention algorithms:
    - Mutual Exclusion: Ensuring that only one process accesses a shared resource at a time.
    - Hold and Wait: Limiting processes to hold only those resources that they need, and then request others.
    - No Preemption: Preventing the removal of a held resource from a process.

- Circular Wait: Imposing a circular waiting chain of processes.

2. Detection and Recovery algorithms:
    - Wait-for-Graph: This method uses a graph to represent processes and resources.
    - Resource Allocation Graph: This method uses a graph to represent the allocation of resources to processes.
    - Timeout: In this method, a timer is associated with each process and the lock is automatically released if the process does not complete within a specified time.
3. Prevention and Recovery algorithms:
    - Banker’s Algorithm: This algorithm checks for deadlocks in a more comprehensive manner.
    - Chandy-Misra-Haas Algorithm: This algorithm uses distributed algorithms to detect and recover from deadlocks.

Race Condition -

A race condition is a situation in computer programming where the behavior of a program or system depends on the timing of events or the order in which they occur. In a multithreaded environment, race conditions occur when two or more threads access shared resources simultaneously, and the outcome of the program depends on which thread completes the task first. If a race condition occurs, the program may produce unpredictable results, leading to bugs and system crashes. To prevent race conditions, proper synchronization mechanisms, such as mutexes or semaphores, must be used to control access to shared resources and ensure that only one thread can access a shared resource at a time.

Synchronisation -

Synchronization is the coordination of activities in a computer system such that only one activity can be executed at a time. It is a technique used to avoid race conditions and ensure that shared resources are accessed in a controlled and predictable manner. In a multi-threaded or multi-process environment, synchronization is used to protect shared resources from being accessed by multiple activities at the same time, leading to data corruption or other unintended consequences. The goal of synchronization is to ensure that each resource is used in a consistent and well-defined manner, to ensure the correctness of the system. The methods used for synchronization include mutual exclusion, semaphores, monitors, and others, depending on the requirements of the system.

Memory Management -

Memory management is the process of managing the computer's memory usage. It involves allocating memory space to different processes and ensuring that the memory is used efficiently and effectively. Memory management is an important aspect of operating systems, and is responsible for ensuring that processes have sufficient memory to run, while also preventing processes from interfering with each other. This is achieved by implementing various algorithms, such as paging, segmentation, and virtual memory, which determine how memory is allocated, shared, and reused among different processes.

Scheduling -

Scheduling is the process of deciding when a computer program or process should be executed, and allocating the necessary resources to execute it. In operating systems, scheduling is the task of assigning the available CPU time to different processes or threads so that they can be executed efficiently. The objective of scheduling is to ensure that the processes or threads are executed in an order that optimizes the use of system resources such as the CPU, memory, and I/O devices. Scheduling algorithms take into account the priority, responsiveness, and resource requirements of each process or thread to make decisions about which process or thread should run next.

Scheduling algorithms are used to determine which process or thread should be executed next and how much time should be allocated for it. They play a crucial role in the efficient functioning of an operating system. Some of the common scheduling algorithms are:

1. First-Come, First-Served (FCFS) Scheduling: The process that arrives first is executed first.
2. Shortest-Job-First (SJF) Scheduling: The process with the shortest execution time is executed first.
3. Priority Scheduling: Each process is assigned a priority, and the process with the highest priority is executed first.

4. Round-Robin (RR) Scheduling: Each process is executed for a fixed time slice and is then rotated to the end of the queue.
5. Multilevel Queue Scheduling: A system can have multiple priority levels and each level can have its own scheduling algorithm.
6. Multilevel Feedback Queue Scheduling: A process can move between multiple priority levels based on its behavior, e.g. CPU utilization.

Paging -

Paging is a memory management technique in Operating Systems where the process's address space is divided into pages. Each page is a fixed-sized block of memory that holds a part of the process's code, data, or stack. When a process requires memory, it is allocated a page, and when it no longer requires the page, it is freed and can be allocated to another process. Paging helps reduce fragmentation and simplifies memory allocation and deallocation, as well as enabling virtual memory, which allows processes to run in memory larger than the physical memory of the computer.

Important Questions -

Here are some common Operating System (OS) interview questions that you may encounter:

1. What is an Operating System (OS)?
2. Explain the functions of an OS.
3. What are the different types of OS?
4. What is a process in OS?
5. Explain process management in OS.
6. What is a thread in OS?
7. Explain memory management in OS.
8. What is file management in OS?
9. What is paging and segmentation in memory management?
10. What is virtual memory in OS?
11. Explain scheduling algorithms in OS.

12. What is a deadlock in OS and how does it occur?
13. What is synchronization in OS?
14. Explain the role of I/O management in OS.
15. What are system calls in OS?
16. What is an interrupt in OS?
17. What is an interrupt handler in OS?
18. What is multitasking in OS?
19. What is multi-threading in OS?
20. What is multiprocessing in OS?

These are some of the important OS interview questions that you may face during an interview.

If you are preparing for OS right now, we recommend you take a 1-1 session with a matched mentor from FAANG or Top startups on Mentro who can help you out with related concepts and problems and help you crack your interview.

Computer Networks

Overview -

CN stands for Computer Network, a network of interconnected computers, allows communication and sharing of resources among connected devices. CNs can be connected through various technologies like Ethernet, Wi-Fi, Cellular, etc. They can be used to connect a local area network (LAN) or wide area network (WAN) in organizations or to connect devices across the internet for global communication.

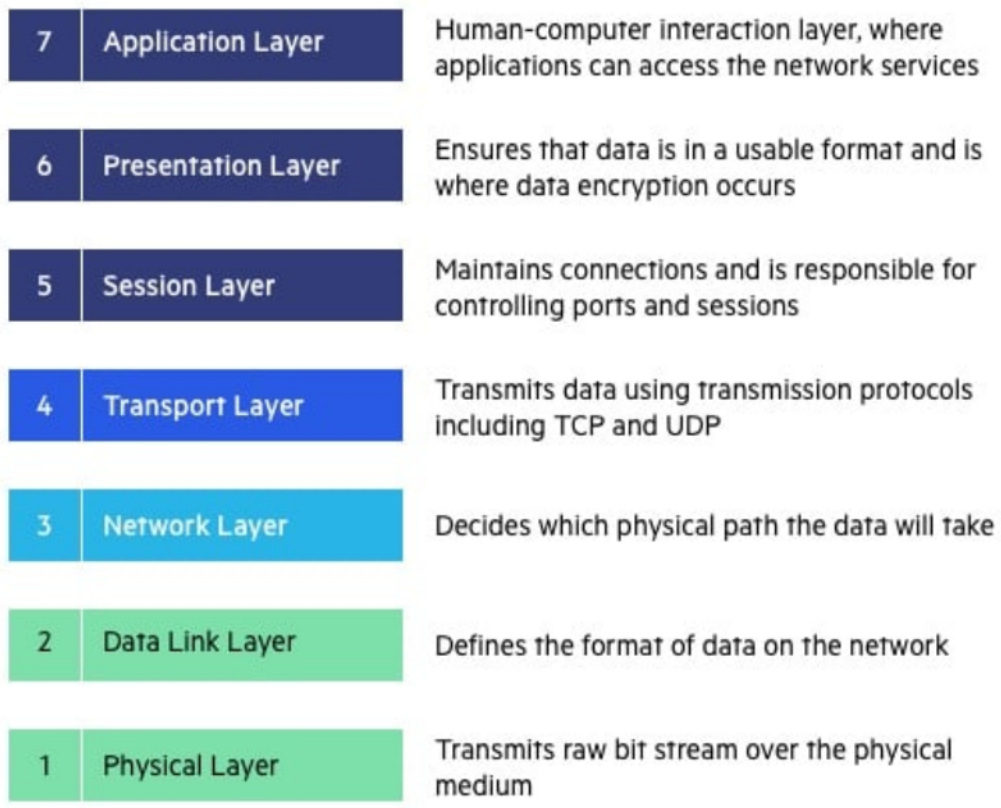
OSI Model -

The OSI Model (Open Systems Interconnection Model) is a theoretical framework that defines the seven layers of a computer network architecture. It was developed by the International Organization for Standardization (ISO) in the 1980s to promote standardization and interoperability between different computer networks.

The 7 layers of the OSI model are:

1. Physical Layer: Defines the electrical, mechanical, and functional specifications for the physical transmission of data.
2. Data Link Layer: Provides error-free delivery of data frames between adjacent nodes in a network.
3. Network Layer: Responsible for routing data from one network to another, ensuring that the data arrives at its destination without errors.
4. Transport Layer: Provides reliable and ordered data transmission, flow control, and error checking.
5. Session Layer: Coordinates and manages the establishment, maintenance, and termination of communication sessions between applications.
6. Presentation Layer: Performs data formatting, encryption, and compression to make sure the data is usable by the application.
7. Application Layer: Provides a user interface and a set of services to support applications, such as email, file transfer, and virtual terminal services.

Each layer of the OSI model performs specific functions and communicates with the layer above and below it, allowing different computer networks to communicate with each other.



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

TCP/IP Model -

The Transmission Control Protocol (TCP) Model is a layered network architecture that provides reliable communication between applications running on different hosts over a network. It is often referred to as the TCP/IP Model, as it works in conjunction with the Internet Protocol (IP) to provide end-to-end communication over the internet.

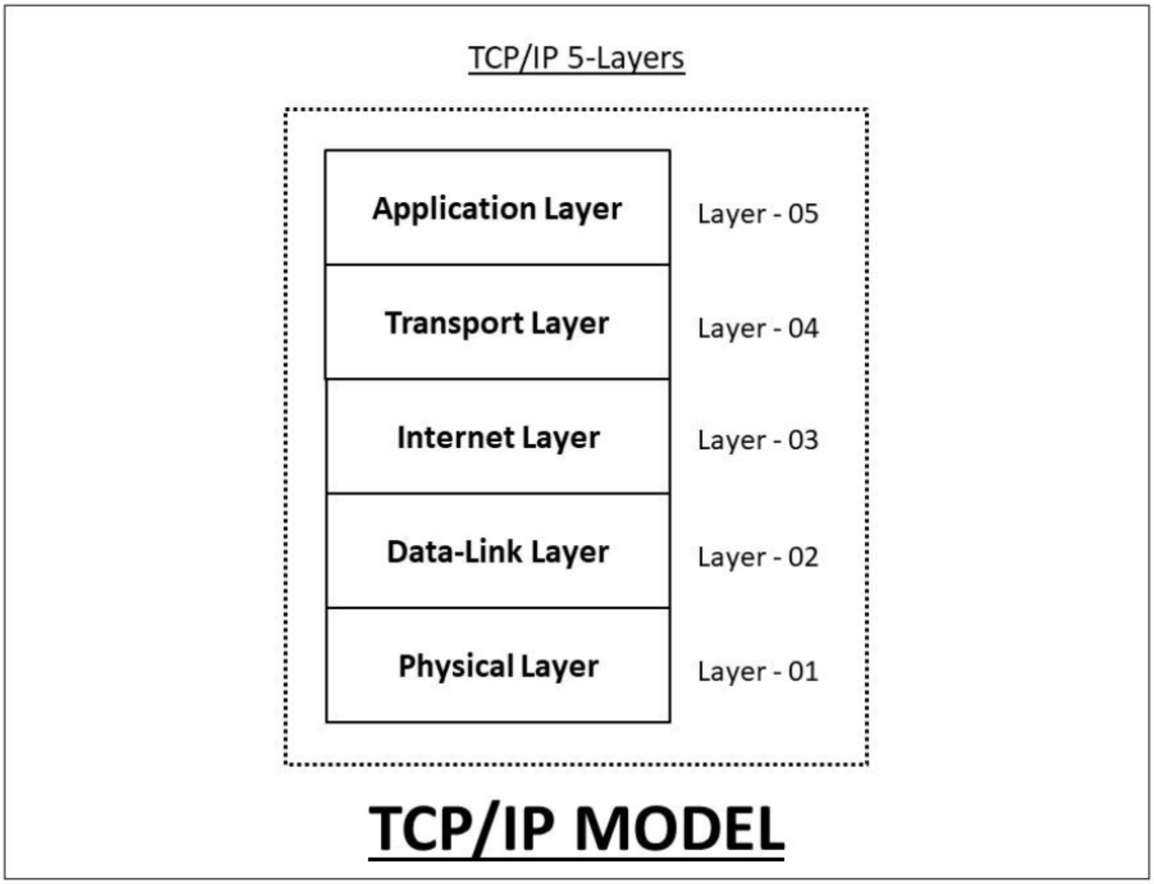
The TCP Model consists of four layers:

1. Application layer: This layer interacts directly with the user applications and provides services such as email, file transfer, and remote access.
2. Transport layer: This layer is responsible for providing reliable data transport services to the applications and maintaining the flow of data between the

communicating parties. It includes the TCP and User Datagram Protocol (UDP).

3. Network layer: This layer is responsible for routing the data across the network and providing logical addressing (IP addresses) to the devices.
4. Link layer: This layer provides low-level services such as physical addressing (MAC addresses) and error detection.

Each layer of the TCP Model is designed to provide specific functionality, and the layers interact to provide a complete communication solution.



Topologies -

Network topologies refer to the physical or logical layout of a network, which describes how different nodes or devices in a network are connected to each other and how data flows between them. The most common network topologies are:

1. Bus Topology: A single cable or backbone runs through the entire network and devices are connected to it with taps or terminators.
2. Star Topology: All devices are connected to a central device, typically a switch or hub, which acts as a hub of data flow.
3. Ring Topology: Devices are connected in a closed loop, with data flowing in a single direction.
4. Mesh Topology: Every device is connected to every other device in the network, allowing for multiple paths for data to flow.

Ethernet -

Ethernet is a type of local area network (LAN) technology used to physically connect devices together in a network. It was standardized in the 1980s and has since become one of the most widely used LAN technologies in the world. Ethernet uses a bus or star topology and supports data transfer rates of 10 Mbps, 100 Mbps, and 1 Gbps. Ethernet also uses the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol to manage network access and prevent data collisions.

IPv4 and IPv6 -

The Internet Protocol (IP) is the main protocol used for transmitting data over the internet. IP is responsible for routing data packets from the source to the destination across multiple network hops.

There are two versions of IP: IP version 4 (IPv4) and IP version 6 (IPv6).

IPv4 is the older version of IP and is still widely used. It uses a 32-bit address space, allowing for approximately 4 billion unique IP addresses. However, the rapid growth of the internet has led to the depletion of the available IPv4 addresses.

IPv6, on the other hand, is the newer version of IP and uses a 128-bit address space, providing for a virtually unlimited number of unique IP addresses. IPv6 also provides improvements in routing, security, and network autoconfiguration compared to IPv4.

Due to the limited number of available IPv4 addresses, many organizations are gradually transitioning to IPv6.

Media Access Control -

Media Access Control (MAC) is a sublayer of the Data Link Layer in the OSI Model. It provides the necessary controls for data transmission over a network, and enables devices to access the physical layer of a network.

The MAC address, also known as a hardware address or physical address, is a unique identifier assigned to each device on a network. It is used to identify devices at the data link layer and is required for communication between devices.

The MAC address is a 48-bit number, expressed as 12 hexadecimal digits, and is unique to each network adapter. The first half of the address identifies the manufacturer, while the second half is unique to the specific device. The MAC address is used by the network to send data to the correct device and is also used for the ARP (Address Resolution Protocol) to translate IP addresses to physical addresses.

Address Resolution Protocol -

Address Resolution Protocol (ARP) is a networking protocol used to map an IP address to a physical (MAC) address on a local network. ARP operates at the Data Link Layer of the OSI Model, and is used to convert the IP addresses of devices into their corresponding MAC addresses, which are unique and used to identify devices on a network. When a device wants to send a network packet to another device on the same network, it uses ARP to look up the MAC address of the destination device based on its IP address. ARP maintains a cache (also known as ARP table) of mappings between IP addresses and MAC addresses to speed up subsequent lookups and reduce the amount of ARP traffic on the network.

Firewall -

Firewalls: A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It acts as a barrier between a trusted internal network and an untrusted external network, such as the Internet.

TCP Ports -

TCP Ports: Transmission Control Protocol (TCP) is a network communication protocol that uses port numbers to identify applications or services. A TCP port number is 16-bit number assigned to a specific process or service. The most commonly used TCP ports are between 0 and 1023.

UDP Ports -

UDP Ports: User Datagram Protocol (UDP) is another network communication protocol that uses port numbers to identify applications or services. Like TCP, UDP port numbers are 16-bit numbers assigned to a specific process or service. Unlike TCP, UDP is connectionless, meaning that data is sent without establishing a reliable connection between the sender and receiver.

DNS -

DNS (Domain Name System) is a distributed, hierarchical naming system used to translate domain names (such as **www.example.com**) into IP addresses (such as 192.0.2.1) in order to access the internet. The DNS maps the domain names to IP addresses and provides a unified global system to access the internet. It operates at the application layer of the OSI model. The DNS server is used to look up the IP address associated with a domain name, so that requests for internet resources can be routed to the correct server.

Important Questions -

Some of the popular and important questions asked on computer networks (CN) in technical interviews are:

1. What is the OSI Model and how does it work?
2. What is the difference between TCP and UDP?
3. What is the Address Resolution Protocol (ARP)?
4. What is the Internet Protocol (IP) and what are IP version 4 (IPv4) and IP version 6 (IPv6)?
5. What is the Transmission Control Protocol (TCP) and how does it work?
6. What is the User Datagram Protocol (UDP) and how does it work?
7. What is the Domain Name System (DNS) and how does it work?
8. What is a firewall and how does it work?
9. What is the Media Access Control (MAC) and how does it work?
10. What is network topology and what are the different types of network topologies?