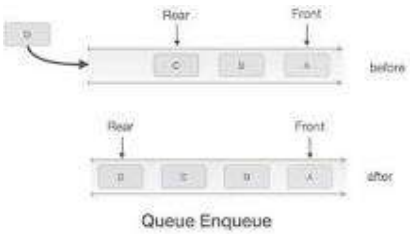
Data structures

01.	**Data Structures**	A **data structure** is a particular way of organizing data in a computer so that it can be used effectively.
02.	**Linked list**	Like arrays, Linked List is a linear data structure. Unlike arrays, linked list elements are not stored at a contiguous location; the elements are linked using pointers.
03.	**Advantages of linked list**	**1)** Dynamic size **2)** Ease of insertion/deletion
04.	**Singly linked list**	A singly linked list is a type of linked list that is **unidirectional**, that is, it can be traversed in only one direction from head to the last node (tail). ... A single node contains data and a pointer to the next node which helps in maintaining the structure of the list.
05.	**Doubly linked list**	a doubly linked list is a linked data structure that **consists of a set of sequentially linked records called nodes**. Each node contains three fields: two link fields (references to the previous and to the next node in the sequence of nodes) and one data field.
06.	**Circular linked list**	Circular Linked List is a **variation of Linked list** in which the first element points to the last element and the last element points to the first element. Both Singly Linked List and Doubly Linked List can be made into a circular linked list.
07.	**Stack**	A stack is **an abstract data type that holds an ordered, linear sequence of items**. In contrast to a queue, a stack is a last in, first out (LIFO) structure. A real-life example is a stack of plates: you can only take a plate from the top of the stack, and you can only add a plate to the top of the stack.
08.	**Stack operation**	Push (insertion) Pop (deletion)

09.	**queue**	 Queue Enqueue Queue is **an abstract data structure**, somewhat similar to Stacks. Unlike stacks, a queue is open at both its ends. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue). ... A real-world example of queue can be a single-lane one-way road, where the vehicle enters first, exits first. FIFO
10.	**Priority queue**	a priority queue is an abstract data type similar to a regular queue or stack data structure in which **each element additionally has a "priority" associated with it**. In a priority queue, an element with high priority is served before an element with low priority.
11.	**Tree**	A tree is **a hierarchical data structure defined as a collection of nodes**. Nodes represent value and nodes are connected by edges. ... Each node has one parent only but can have multiple children. Each node is connected to its children via edge.
12.	**graphs**	A graph is a non-linear data structure, which consists of vertices(or nodes) connected by edges(or arcs) where edges may be directed or undirected. In Computer science graphs are **used to represent the flow of computation**.
13.	**set**	Set is a **data structure that is used as a collection of objects**. ... set can not contain duplicate elements . Java Set allow at most one null value. Set interface contains only methods inherited from Collection.

14.	**Map**	A map data type represents **an unordered collection of key-value pair elements**. ... A map element is a key and value pair that maps one thing to another.
15.	**Searching algorithms**	Linear Search. Binary Search. Sequential search
16.	**Binary search**	Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise, narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.
17.	**Sequential search**	a sequential search is **made over all items one by one**. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.
18.	**Sorting techniques**	bubble sort, insertion sort, selection sort, bucket sort, heap sort, quick sort, radix sort
19.	**Bubble sort**	Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order. **Worst and Average Case Time Complexity:** O(n*n). Worst case occurs when array is reverse sorted. **Best Case Time Complexity:** O(n). Best case occurs when array is already sorted. **Auxiliary Space:** O(1)
20.	**Merge sort**	Merge Sort is a Divide and Conquer algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted

		halves. **The merge() function** is used for merging two halves. Time complexity of Merge Sort is θ(nLogn) in all 3 cases (worst, average and best) as merge sort always divides the array into two halves and takes linear time to merge two halves. **Auxiliary Space:** O(n)
21.	**Quick sort**	QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways. 1. Always pick first element as pivot. 2. Always pick last element as pivot (implemented below) 3. Pick a random element as pivot. 4. Pick median as pivot. time complexity (nLogn)
	DBMS	
01.	**Super key**	We can define a super key as a set of those keys that identify a row or a tuple uniquely. The word super denotes the superiority of a key. Thus, a super key is the superset of a key known as a **Candidate key** (discussed in the next section). It means a candidate key is obtained from a super key only. The role of the super key is simply to identify the tuples of the specified table in the database.
02.	**Primary key**	In the relational model of databases, a primary key is **a specific choice of a minimal set of attributes (columns) that uniquely specify a tuple (row) in a relation** (table). Informally, a primary key is "which attributes identify a record" For example, students are routinely assigned unique identification (ID) numbers

03.	**Foreign key**	A foreign key is **a column or group of columns in a** relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table. For example: In the below example the Stu_Id column in Course_enrollment table is a foreign key as it points to the primary key of the Student table.
04.	**Join stmnts**	A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are: • INNER JOIN • LEFT JOIN • RIGHT JOIN • FULL JOIN
05.	**Inner join**	The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be same.
06.	**Left join**	This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain *null*.
07.	**Right join**	RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain *null*. RIGHT JOIN is also known as RIGHT OUTER JOIN
08.	**Full join**	FULL JOIN creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables. The rows

		for which there is no matching, the result-set will contain *NULL* values.
09.	**Types of SQL commands**	1. DDL – Data Definition Language 2. DQl – Data Query Language 3. DML – Data Manipulation Language 4. DCL – Data Control Language
10.	**DDl**	DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.
11.	**DDL example**	• **CREATE** – is used to create the database or its objects (like table, index, function, views, store procedure and triggers). • **DROP** – is used to delete objects from the database. • **ALTER**-is used to alter the structure of the database. • **TRUNCATE**–is used to remove all records from a table, including all spaces allocated for the records are removed. • **COMMENT** –is used to add comments to the data dictionary. • **RENAME** –is used to rename an object existing in the database.
12.	**DQL**	statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it.
13.	**DQL example**	• **SELECT** – is used to retrieve data from the database.

14.	**DML**	The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.
15.	**DML example**	• **INSERT** – is used to insert data into a table. • **UPDATE** – is used to update existing data within a table. • **DELETE** – is used to delete records from a database table.
16.	**DCL**	DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions and other controls of the database system.
17.	**DCL example**	• **GRANT**-gives users access privileges to the database. • **REVOKE**-withdraw user's access privileges given by using the GRANT command.
18.	**TCL(transaction Control Language)**	TCL commands deal with the transaction within the database.
19.	**TCL example**	• **COMMIT**– commits a Transaction. • **ROLLBACK**– rollbacks a transaction in case of any error occurs. • **SAVEPOINT**–sets a savepoint within a transaction. • **SET TRANSACTION**–specify characteristics for the transaction.
20.	**PL/SQL**	PL/SQL is a combination of SQL along with the procedural features of programming languages. PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.
21.	**Callable statement**	CallableStatement interface is used to call the **stored procedures and functions**.

		We can have business logic on the database by the use of stored procedures and functions that will make the performance better because these are precompiled.
22.	**Prepared statement**	The PreparedStatement interface is a subinterface of Statement. It is used to execute parameterized query. **Improves performance**: The performance of the application will be faster if you use PreparedStatement interface because query is compiled only once.
	Technology Based	
01.	**Artificial intelligence**	Artificial intelligence (AI) refers to **the simulation of human intelligence in machines** that are programmed to think like humans and mimic their actions.
02.	**Types of AI**	narrow or weak AI, general or strong AI, and artificial superintelligence.
03.	**Narrow AI (ANI)**	• Artificial narrow intelligence (ANI or narrow AI) refers to a computer's ability to perform a single task extremely well, such as crawling a webpage or playing chess.
04.	**General AI (AGI)**	• Artificial general intelligence (AGI) is when a computer program can perform any intellectual task that a human could. • **Self-driving cars**
05.	**Super AI (ASI)**	Artificial super intelligence (ASI) is an AI that surpasses human intellect. **Siri** by Apple, Alexa by Amazon IBM's Watson and other virtual assistants

01.	**Machine Learning**	• It is a branch of **artificial intelligence** based on the idea that systems can learn from data. • It is the study of computer algorithm that improve automatically through experience and by the use of data. • In Ml, the input is given to the machine learning model and it will generate the output according to the algorithms we have given. If the output is correct it will finalize the result. Else it will return the feedback to the model until it learns.
02.	**Labelled data**	Labelled data is a data which associated with the features. For example, if we labelled the data as 3grm then the feature would be one rupee coin.
03.	**Supervised learning**	In supervised learning the machine learns from the labelled data. Machine learning model knew the features of the object and also the labelles associated with those features.
04.	**Unsupervised learning**	In Unsupervised learning the machine learns from the labelled data. The machine learning model will predict which result belongs to which category by learning the data sets.
05.	**Reinforcement Learning**	It is a reward based learning. It works in the principle of feedback. If the machine learning model produced cat as output while the input given is dog. Then, it will send the negative feedback to the model. The machine starts learning from the feedbacks.
06.	**Cloud computing**	Cloud computing is the on demand services through the internet. In Cloud computing, Computing services performs over the cloud. Cloud is shared resources include tools and applications like data storage, servers, networking, database and software.

		Cloud services follows the principle “Pay as you Go”. For example the default storage for Google drive is 15GB, if we want some more space then we need to pay according the space. The major cloud services are Saas, Paas and Iaas. Some familiar cloud providers are AWS, Google cloud, Microsoft azure.
07.	**SAAS**	SAAS provides the software. We can use the softwares from the cloud no need to install the applications and those softwares are platform independent and can be used by multiple end users. These softwares are accessible by any web browsers. Ex for SAAS : G-mail, google sheets, CRM services like Salesforce.
08.	**PAAS**	Platform as a service is made up of a programming language execution environment, an operating system, a web server and database. Developers can use these services. Ex of PAAS: google app engine, heroku , ms windows cloud azure.
09.	**IAAS**	This services offers the computing architecture & Infrastructure, all computing resources but in a virtual environment so that multiple users can access them. Resources include, data storage, virtualization, servers & networking. SysAdmins can use these services. Ex for IAAS: Amazon EC2, GOGRID
