DBMS

1. What is DBMS ? Mention advantages.

   Database Management System (DBMS) is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.
   Advantages:
   - Improved data sharing
   - Improved data security
   - Better data integration
   - Minimised data inconsistency
   - Improved data access
   - Improved decision making
   - Improved end-user productivity

2. What is Database and a database system?

   A database is an organised collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just a database.

3. What is RDBMS ? Properties.

   A Relational Database Management system (RDBMS) is a database management system that is based on the relational model. It has the following major components: Table, Record/Tuple/Row, Field, and Column/Attribute. Examples of the most popular RDBMS are MYSQL, Oracle, IBM DB2, and Microsoft SQL Server database.

   Relational databases have the following properties:
   - Values are atomic.
   - All of the values in a column have the same data type.
   - Each row is unique.
   - The sequence of columns is insignificant.
   - The sequence of rows is insignificant.
   - Each column has a unique name.
   - Integrity constraints maintain data consistency across multiple tables.

4. Types of database languages

1. Data Definition Language

   DDL stands for Data Definition Language. It is used to define database structure or pattern.
   It is used to create schema, tables, indexes, constraints, etc. in the database.
   Using the DDL statements, you can create the skeleton of the database.
   Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.
   Here are some tasks that come under DDL:

   - Create: It is used to create objects in the database.
   - Alter: It is used to alter the structure of the database.
   - Drop: It is used to delete objects from the database.
   - Truncate: It is used to remove all records from a table.
   - Rename: It is used to rename an object.
   - Comment: It is used to comment on the data dictionary.

   These commands are used to update the database schema that's why they come under Data definition language.

2. Data Manipulation Language

   DML stands for Data Manipulation Language. It is used for accessing and manipulating data in a database. It handles user requests.

   Here are some tasks that come under DML:

   - Select: It is used to retrieve data from a database.
   - Insert: It is used to insert data into a table.
   - Update: It is used to update existing data within a table.
   - Delete: It is used to delete all records from a table.
   - Merge: It performs UPSERT operation, i.e., insert or update operations.
   - Call: It is used to call a structured query language or a Java subprogram.
   - Explain Plan: It has the parameter of explaining data.
   - Lock Table: It controls concurrency.

3. Data Control Language

   DCL stands for Data Control Language. It is used to retrieve the stored or saved data.
   The DCL execution is transactional. It also has rollback parameters.
   (But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Here are some tasks that come under DCL:

- Grant: It is used to give user access privileges to a database.
- Revoke: It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:

CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

4. Transaction Control Language

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

Here are some tasks that come under TCL:

- Commit: It is used to save the transaction on the database.
- Rollback: It is used to restore the database to originality since the last commit.

5. ACID properties (VVVVV IMP)

To ensure the consistency of the database, certain properties are followed by all the transactions occurring in the system.
These properties are called ACID Properties of a transaction.

1. Atomicity
- This property ensures that either the transaction occurs completely or it does not occur at all.
- In other words, it ensures that no transaction occurs partially.

2. Consistency
- This property ensures that integrity constraints are maintained.
- In other words, it ensures that the database remains consistent before and after the transaction.

3. Isolation
- This property ensures that multiple transactions can occur simultaneously without causing any inconsistency.
- The resultant state of the system after executing all the transactions is the same as the state that would be achieved if the transactions were executed serially one after the other.

4. Durability
- This property ensures that all the changes made by a transaction after its successful executions are written successfully to the disk.
- It also ensures that these changes exist permanently and are never lost even if there occurs a failure of any kind.

6. Difference between vertical and horizontal scaling

Scaling alters the size of a system. In the scaling process, we either compress or expand the system to meet the expected needs. The scaling operation can be achieved by adding resources to meet the smaller expectation in the current system, or by adding a new system in the existing one, or both.

Vertical scaling keeps your existing infrastructure but adds computing power. Your existing pool of code does not need to change — you simply need to run the same code on machines with better specs. By scaling up, you increase the capacity of a single machine and increase its throughput. Vertical scaling allows data to live on a single node, and scaling spreads the load through CPU and RAM resources for your machines.

Horizontal scaling simply adds more instances of machines without first implementing improvements to existing specifications. By scaling out, you share the processing power and load balancing across multiple machines.

Horizontal Scaling	Vertical Scaling
When new server racks are added in the existing system to meet the higher expectation, it is known as horizontal scaling.	When new resources are added in the existing system to meet the expectation, it is known as vertical scaling
It expands the size of the existing system horizontally.	It expands the size of the existing system vertically.
It is difficult to implement	It is easy to implement
It is costlier, as new server racks comprises of a lot of resources	It is cheaper as we need to just add new resources
It takes more time to be done	It takes less time to be done

7. What is sharding

Sharding is a method of splitting and storing a single logical dataset in multiple databases. By distributing the data among multiple machines, a cluster of database systems can store larger dataset and handle additional requests. Sharding is necessary if a dataset is too large to be stored in a single database. Moreover, many sharding strategies allow additional machines to be added. Sharding allows a database cluster to scale along with its data and traffic growth.
https://www.mongodb.com/features/database-sharding-explained
https://youtu.be/5faMjKuB9bc

8. Keys in DBMS

Keys:
A key is a set of attributes that can identify each tuple uniquely in the given relation.

Types of Keys:
● Super Key - A superkey is a set of attributes that can identify each tuple uniquely in the given relation. A super key may consist of any number of attributes.

● Candidate Key - A set of minimal attribute(s) that can identify each tuple uniquely in the given relation is called a candidate key.

● Primary Key - A primary key is a candidate key that the database designer selects while designing the database. Primary Keys are unique and NOT NULL.

● Alternate Key - Candidate keys that are left unimplemented or unused after implementing the primary key are called alternate keys.

● Foreign Key - An attribute ‘X’ is called as a foreign key to some other attribute ‘Y’ when its values are dependent on the values of attribute ‘Y’. The relation in which attribute ‘Y’ is present is called the referenced relation. The relation in which attribute ‘X’ is present is called the referencing relation.

● Composite Key - A primary key composed of multiple attributes and not just a single attribute is called a composite key.

● Unique Key - It is unique for all the records of the table. Once assigned, its value cannot be changed i.e. it is non-updatable. It may have a NULL value.

9. Types of relationship

Relationship:
A relationship is defined as an association among several entities.

- Unary Relationship Set - Unary relationship set is a relationship set where only one entity set participates in a relationship set.

- Binary Relationship Set - Binary relationship set is a relationship set where two entity sets participate in a relationship set.

- Ternary Relationship Set - Ternary relationship set is a relationship set where three entity sets participate in a relationship set.

- N-ary Relationship Set - N-ary relationship set is a relationship set where ‘n’ entity sets participate in a relationship set.

10. Data abstraction in DBMS, three levels of it

Data Abstraction is a process of hiding unwanted or irrelevant details from the end user. It provides a different view and helps in achieving data independence which is used to enhance the security of data.

Database systems include complex data-structures. In terms of retrieval of data, reduce complexity in terms of usability of users and in order to make the system efficient, developers use levels of abstraction that hide irrelevant details from the users. Levels of abstraction simplify database design.

Mainly there are three levels of abstraction for DBMS, which are as follows:

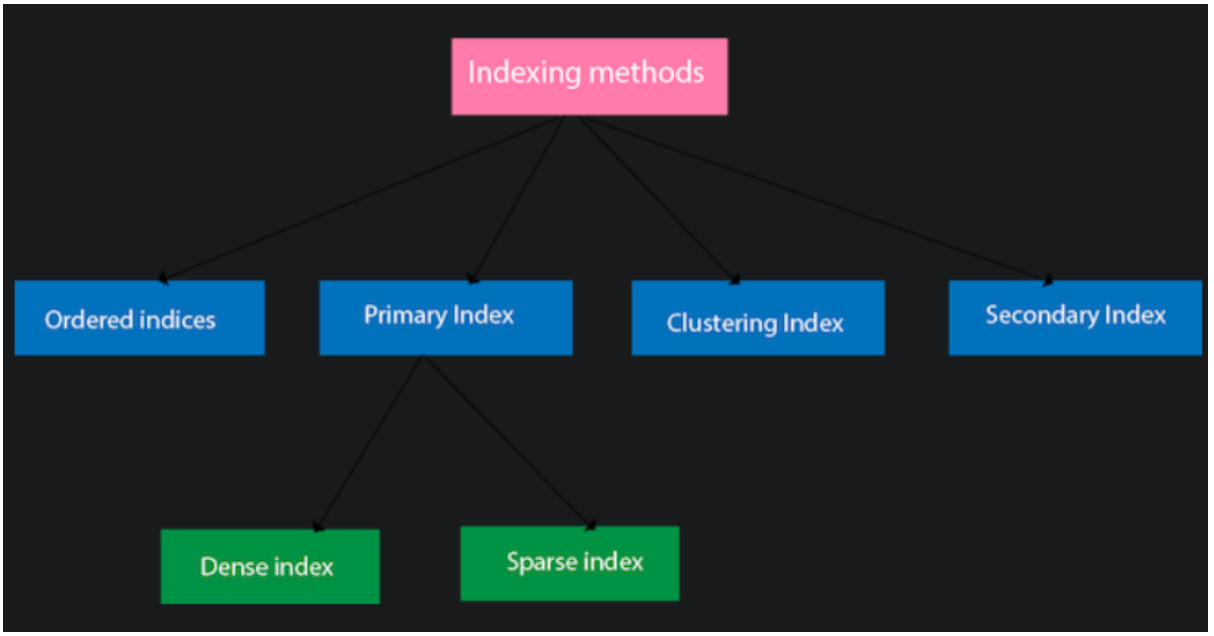
1. Physical or Internal Level: It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database.

2. Logical or Conceptual Level: Logical level is the intermediate level or next higher level. It describes what data is stored in the database and what relationship exists among those data.

3. View or External Level: It is the highest level. In view level, there are different levels of views and every view only defines a part of the entire data

11. Indexing in DBMS

Indexing is a way to optimise the performance of a database by minimising the number of disk accesses required when a query is processed. It is a data structure technique which is used to quickly locate and access the data in a database.



12. What is DDL (Data Definition Language)
Discussed earlier

13. What is DML (Data Manipulation Language)
Discussed earlier

14. What is normalisation ? Types of them *(Refer Javatpoint for all)*

Normalisation:
In DBMS, database normalisation is a process of making the database consistent by-

- Reducing the redundancies
- Ensuring the integrity of data through lossless decomposition

Normal Forms:

1. First Normal Form (1NF) - A given relation is called in First Normal Form (1NF) if and only if
    - There are **only Single Valued** Attributes.
    - Attribute Domain does not change.
    - There is a unique name for every Attribute/Column.
    - The order in which data is stored does not matter.

2. Second Normal Form (2NF) - A given relation is called in Second Normal Form (2NF) if and only if
    - Relation already exists in 1NF.
    - **No partial dependency** exists in the relation.
      No non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

3. Third Normal Form (3NF) - A given relation is called in Third Normal Form (3NF) if and only if
    - Relation already exists in 2NF.
    - **No transitive dependency** exists for non-prime attributes.
      if it holds at least one of the following conditions for every non-trivial functional dependency X → Y.
        - X is a super key.
        - Y is a prime attribute, i.e., each element of Y is part of some candidate key.

4. Boyce-Codd Normal Form - A given relation is called in BCNF if and only if
    - Relation already exists in 3NF.
    - For every non-trivial functional dependency ‘A → B’, A is a **super key** of the relation.

5. Fourth Normal Form (4NF) - A relation will be in 4NF if it is
    - in Boyce Codd normal form

- **No multivalued dependency.**
  For a dependency A → B, if for a single value of A, multiple values of B exists, then the relation will be a multivalued dependency.

6. Fifth Normal Form (5NF) - A relation is in 5NF if it is
   - in 4NF
   - does not contain any join dependency
   - joining should be lossless

15. What is denormalization ?

https://www.geeksforgeeks.org/denormalization-in-databases/

Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that denormalization does not mean not doing normalisation. It is an optimization technique that is applied after doing normalisation.

16. What is functional dependency ?

A functional dependency is a constraint that specifies the relationship between two sets of attributes where one set can accurately determine the value of other sets. It is denoted as X → Y, where X is a set of attributes that is capable of determining the value of Y. The attribute set on the left side of the arrow, X is called Determinant, while on the right side, Y is called the Dependent.

17. E-R Model ?

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
In ER modelling, the database structure is portrayed as a diagram called an entity-relationship diagram.

18. Conflict Serializability in DBMS

Lec-77: Conflict Equivalent Schedules with Example | Transaction concurrency …

Serializability is a concept that helps us to check which schedules are serializable. A serializable schedule is the one that always leaves the database in consistent state.

A schedule is called conflict serializability if after swapping of non-conflicting operations, it can transform into a serial schedule.
The schedule will be a conflict serializable if it is conflict equivalent to a serial schedule.
Conflicting Operations

The two operations become conflicting if all conditions satisfy:

a) Both belong to separate transactions.
b) They have the same data item.
c) They contain at least one write operation

19. Explain Normal forms in DBMS

Discussed earlier

20. What is CCP ? (Concurrency Control Protocols)

Concurrency Control is the management procedure that is required for controlling concurrent execution of the operations that take place on a database

The concurrency control protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions.

Therefore, these protocols are categorised as:

1. Lock Based Concurrency Control Protocol
2. Timestamp Concurrency Control Protocol
3. Validation Based Concurrency Control Protocol

21. Entity, Entity Type, Entity Set, Weak Entity Set.

Entity in DBMS can be a real-world object with an existence, For example, in a College database, the entities can be Professor, Students, Courses, etc.

The entity type is a collection of the entity having similar attributes.

Types of Entity type

1. Strong Entity Type
   Strong entities are those entity types which have a key attribute. The primary key helps in identifying each entity uniquely. It is represented by a rectangle.

2. Weak Entity Type
   Weak entity type doesn't have a key attribute. Weak entity types can't be identified on their own. It depends upon some other strong entity for its distinct identity.

Entity Set:
Entity Set is a collection of entities of the same entity type. We can say that entity type is a superset of the entity set as all the entities are included in the entity type.

https://www.geeksforgeeks.org/weak-entity-set-in-er-diagrams/

22. What are SQL commands ? Types of them.

https://www.javatpoint.com/dbms-sql-command

23. Nested Queries in SQL ?

https://www.tutorialspoint.com/explain-about-nested-queries-in-dbms

24. What is JOIN ? Explain types of JOINs

https://www.javatpoint.com/dbms-sql-joins

25. Inner and Outer Join

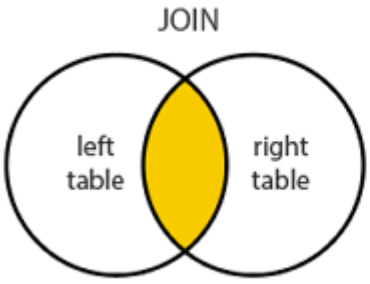
Different Types of SQL JOINs
Here are the different types of the JOINs in SQL:

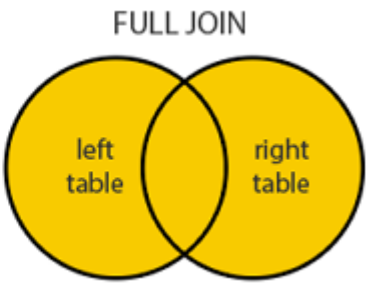
1. (INNER) JOIN:
   Returns records that have matching values in both tables

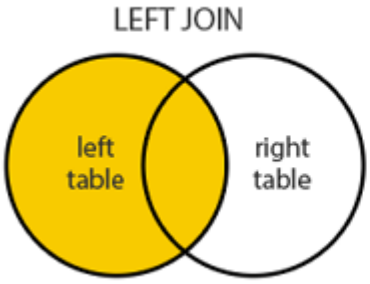
2. LEFT (OUTER) JOIN:
   Returns all records from the left table, and the matched records from the right table

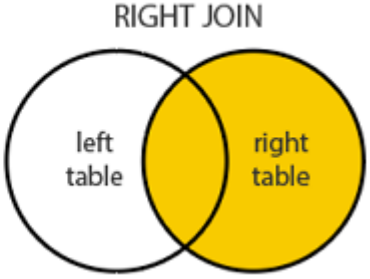
3. RIGHT (OUTER) JOIN:
   Returns all records from the right table, and the matched records from the left table

4. FULL (OUTER) JOIN:
   Returns all records when there is a match in either left or right table









26. Practice sql queries from leetcode

27. Diff between 2 tier and 3 tier architecture

    Difference Between Two-Tier And Three-Tier Database Architecture - GeeksforGeeks

28. Diff between TRUNCATE and DELETE command ..

Comparison Basis	DELETE	TRUNCATE
Definition	The delete statement is used to remove single or multiple records from an existing table depending on the specified condition.	The truncate command removes the complete data from an existing table but not the table itself. It preserves the table structure or schema.
Language	It is a DML (Data Manipulation Language) command.	It is a DDL (Data Definition Language) command.
WHERE	It can use the WHERE clause to filter any specific row or data from the table.	It does not use the WHERE clause to filter records from the table.
Permission	We need to have DELETE permission to use this command.	We need to have ALTER permission to use this command.
Working	This command eliminates records one by one.	This command deletes the entire data page containing the records.

29. Difference between Intension and Extension in a DataBase

    Intension is the permanent part of the relation and comprises two things: relation schema and the integrity constraints. Relation schema defines the name and attributes of the relation, and integrity constraints define key constraints, referential constraints. etc.

    As it corresponds to the schema of the relation, it provides definition to all the extensions of the relation and is time independent.

    For example : Intention of student :
    Student (RollNo Number(4) Not NULL, Name Char(20), Age Number(2), Course Char(15))

Extension is the snapshot of the system at a particular time. It displays values for tuples in a relation at the particular instance of time . It is dependent on time and it keeps on changing as the tuples are added, deleted, edited.

For example, extension of student at time t1 when two tuples were added

RollNo	Name	Age	Course
101	Amit	21	B.Tech.
102	Annanya	20	BCA

extension of student at time t2 when one more tuple was added and one tuple was updated

RollNo	Name	Age	Course
101	Amit	21	B.Tech.
102	Annanya	20	BSc
103	Ankit	24	MCA

30. Difference between share lock and exclusive lock, definition of lock

A lock is a data variable which is associated with a data item. This lock signifies operations that can be performed on the data item. Locks in DBMS help synchronise access to the database items by concurrent transactions. All lock requests are made to the concurrency-control manager

Difference between Shared Lock and Exclusive Lock :

S.No.	Shared Lock	Exclusive Lock
1.	Lock mode is read only operation.	Lock mode is read as well as write operation.
2.	Shared lock can be placed on objects that do not have an exclusive lock already placed on them.	Exclusive lock can only be placed on objects that do no have any other kind of lock.
3.	Prevents others from updating the data.	Prevents others from reading or updating the data.
4.	Issued when transaction wants to read item that do not have an exclusive lock.	Issued when transaction wants to update unlocked item.
5.	Any number of transaction can hold shared lock on an item.	Exclusive lock can be hold by only one transaction.
6.	S-lock is requested using lock-S instruction.	X-lock is requested using lock-X instruction.