Q1. What are the differences between a DBMS and RDBMS?

DBMS	RDBMS
Provides an organized way of managing, retrieving, and storing from a collection of logically related information	Provides the same as that of DBMS, but it provides relational integrity

Q2. Explain the terms database and DBMS. Also, mention the different types of DBMS.

A software application that interacts with databases, applications, and users to capture and analyze the required data. The data stored in the database can be retrieved, deleted and modified based on the client's requirement.

The different types of DBMS are as follows:

- Relational DBMS (RDBMS): This type of DBMS, uses a structure which allows the users to access data in relation to another piece of data in a database. In this type of DBMS, data is stored in the form of tables.
- Hierarchical DBMS: As the name suggests, this type of DBMS has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.
- Network DBMS: This type of DBMS supports many-to-many relations wherein multiple member records can be linked.
- Object-oriented DBMS: Uses small individual software called object to store pieces of data and the instructions for the actions to be done with the data.

Q3. What are the advantages of DBMS?

The advantages of DBMS are as follows:

- Sharing of Data: Multiple users can use data from the same database simultaneously.
- Integrity constraints: These constraints allow the data to be stored in a database in a refined manner.
- Redundancy control: Supports a mechanism to control the redundancy of data by integrating all the data into a single database.
- Data Independence: Allows to change the structure of the data without affecting the structure of any of the running application programs.
- Provide backup and recovery facility: Provides a feature of 'backup and recovery' to automatically create the data backup and restore the data as and when required.

Q4. Mention the different languages present in DBMS

The different languages present in DBMS are as follows:

- DDL(Data Definition Language) – Consists of commands which are used to define the database.

- DML(Data Manipulation Language) – Consists of commands which are used to manipulate the data present in the database.
- DCL(Data Control Language) – Consists of commands which deal with the user permissions and controls of the database system.
- TCL(Transaction Control Language) – Consist of commands which deal with the transaction of the database.

Q5. What do you understand by query optimization?

Query optimization is the phase that identifies a plan for evaluation query that has the least estimated cost. This phase comes into the picture when there are a lot of algorithms and methods to execute the same task.

The advantages of query optimization are as follows:

- The output is provided faster
- A larger number of queries can be executed in less time
- Reduces time and space complexity

Q6. Do we consider NULL values the same as that of blank space or zero?

A NULL value is not at all same as that of zero or a blank space. The NULL value represents a value which is unavailable, unknown, assigned or not applicable whereas zero is a number and blank space is a character.

Q7. What do you understand by aggregation and atomicity?

Aggregation	Atomicity
This is a feature of the E-R model which allows a relationship set to participate in another relationship set.	This property states that a database modification must either follow all the rules or nothing at all. So, if one part of the transaction fails, then the entire transaction fails.

Q8. What are the different levels of abstraction in the DBMS?

There are three levels of data abstraction in DBMS. They are:

- Physical Level: It is the lowest level of abstraction and describes how the data is stored.
- Logical Level: This is the next level of abstraction after the Physical level. This layer determines what data is stored in the database, and what is the relationship between the data points.
- View Level: The View Level is the highest level of abstraction and it describes only a part of the entire database.

Q9. What is an entity-relationship model?

It is a diagrammatic approach to database design, where you represent real-world objects as entities and mention relationships between them. This approach helps the team of DBAs' to understand the schema easily.

Q10. What do you understand by the terms Entity, Entity Type, and Entity Set in DBMS?

- **Entity: An entity is a real-world object having attributes, which are nothing but characteristics of that particular object. For example, an employee can be an entity. This particular entity can have attributes such as empid, empname, etc.**
- **Entity Type: Entity type is nothing but a collection of entities, having the same attributes. Generally, an entity type refers to one or more related tables in a particular database. So, you can understand, entity type as a characteristic which uniquely identifies the entity. For example, An employee can have attributes such as empid, empname, department, etc.**
- **Entity Set: An entity set is the collection of all the entities of a particular entity type in a database. For example, a set of employees, a set of companies, and a set of people can come under an entity set.**

Q11. What are relationships and mention different types of relationships in the DBMS

A relationship in DBMS is the scenario where two entities are related to each other. In such a scenario, the table consisting of foreign key references to that of a primary key of the other table.

The different types of relationships in DBMS are as follows:

- **One-to-One Relationship – Used when a single row in Table A is related to a single row in Table B.**
- **One-to-Many Relationship – Used when a single row in Table A is related to many rows in table B.**
- **Many-to-Many Relationship – Used when many rows in table A can be related to many rows in table B.**
- **Self -Referencing Relationship – Used when a record in table A is related to the same table itself.**

Q12. What is concurrency control?

This is a process of managing simultaneous operations in a database so that database integrity is not compromised. The following are the two approaches involved in concurrency control:

- **Optimistic approach – Involves versioning**
- **Pessimistic approach – Involves locking**

Q13. What are the ACID properties in DBMS?

ACID stands for Atomicity, Consistency, Isolation, Durability. It is used to ensure that the data transactions are processed reliably in a database system.

- **Atomicity: Atomicity refers to those transactions which are completely successful or failed. Here each transaction refers to a single logical operation of a data. So, even if one part of any transaction fails, the entire transaction fails and the database state is left unchanged.**
- **Consistency: Consistency ensures that the data must meet all the validation rules. In simple words, you can say that your transaction never leaves the database without completing its state.**
- **Isolation: The main goal of isolation is concurrency control.**
- **Durability: Durability means that if a transaction has been committed, it will occur whatever may be the scenario.**

Q14. What is normalization and what are the different types of normalization?

The process of organizing data to avoid any duplication of data and redundancy is known as Normalization. There are many successive levels of normalization which are known as normal forms. Each consecutive normal form depends on the previous one. The following are the first three normal forms. Apart from these, you have higher normal forms such as BCNF.

- **First Normal Form (1NF) – No repeating groups within rows**
- **Second Normal Form (2NF) – Every non-key (supporting) column value is dependent on the whole primary key.**
- **Third Normal Form (3NF) – Dependent solely on the primary key and no other non-key (supporting) column value.**

Q15. What are the different types of keys in the database?

There are mainly 7 types of Keys, that can be considered in a database. I am going to consider the below tables to explain to you the various keys.

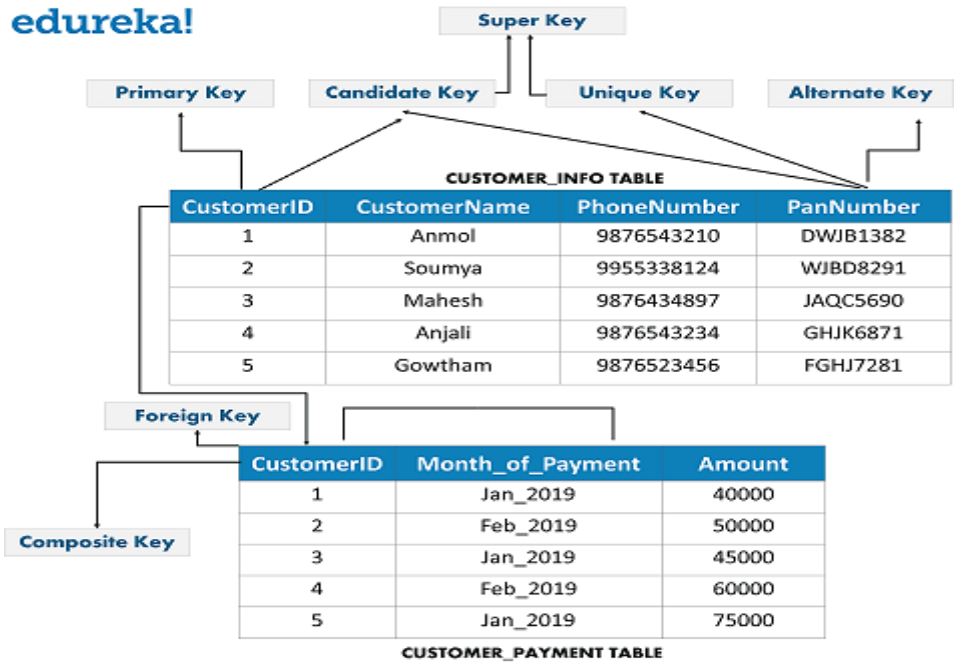


Fig 1: Different Types of Keys in Database – DBMS Interview Questions

- **Candidate Key –** This is a set of attributes which can uniquely identify a table. Each table can have more than a candidate key. Apart from this, out of all the candidate keys, one key can be chosen as the Primary key. In the above example, since CustomerID and PanNumber can uniquely identify every tuple, they would be considered as a Candidate Key.
- **Super Key –** This is a set of attributes which can uniquely identify a tuple. So, a candidate key, primary key, and a unique key is a superkey, but vice-versa isn't true.
- **Primary Key –** This is a set of attributes which are used to uniquely identify every tuple. In the above example, since CustomerID and PanNumber are candidate keys, any one of them can be chosen as a Primary Key. Here CustomerID is chosen as the primary key.
- **Unique Key –** The unique key is similar to the primary key, but allows NULL values in the column. Here the PanNumber can be considered as a unique key.
- **Alternate Key –** Alternate Keys are the candidate keys, which are not chosen as a Primary key. From the above example, the alternate key is PanNumber
- **Foreign Key –** An attribute that can only take the values present as the values of some other attribute, is the foreign key to the attribute to which it refers. in the above example, the CustomerID from the Customers Table is referred to the CustomerID from the Customer_Payment Table.
- **Composite Key –** A composite key is a combination of two or more columns that identify each tuple uniquely. Here, the CustomerID and Date_of_Payment can be grouped together to uniquely identify every tuple in the table.

Q16. What do you understand by correlated subqueries in DBMS?

A correlated subquery is also a sort of subquery reliant on another query. So, when subqueries are executed for each of the rows of outer queries, then they are termed as correlated subqueries. Each subquery is executed a single time for every row of the outer query.

You can also understand correlated subqueries as those queries, which are used for row-by-row processing by the parent statement. Here, the parent statement can be SELECT, UPDATE or DELETE statement.

Q17. Explain Database partitioning and its importance.

Data partitioning is the process of dividing a logical database into independent units for the betterment of availability, performance, and manageability.

The importance of database partitioning is as follows:

- Enables you to access large parts of a specific partition
- Cheap and slower storage can be used to store data
- Improves query performance

Q18. What do you understand by functional dependency and transitive dependency in DBMS?

Functional Dependency: A functional dependency is a constraint that is used in describing the relationship among different attributes in a relation.

Example: **Consider a relation "A1" having attributes X and Y. The functional dependency among these two attributes will be X -> Y, this implies that Y is functionally dependent on X.**

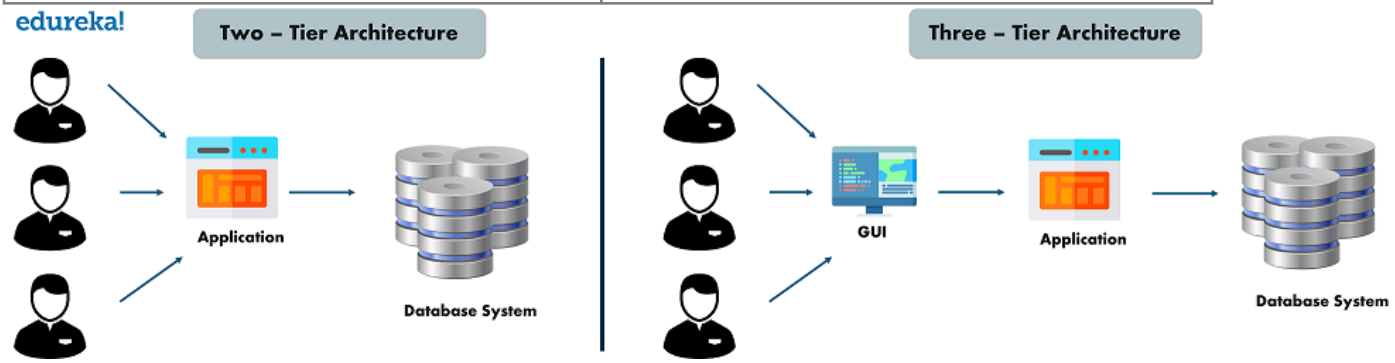
Transitive Dependency: A transitive dependency is a constraint that can only occur in a relation of three or more attributes.

Example: **Consider a relation "A1" having attributes X, Y and Z. Now, X->Z is said to hold transitive dependency, only if the following functional dependencies holds true:**

- **X -> Y**
- **Y doesn't ->X**
- **Y -> Z**

Q19. What is the difference between two and three-tier architectures?

Two-tier architecture	Three-tier architecture
This is similar to the client-server architecture.	This architecture contains an extra layer between the client and the server.
Clients directly communicate with the database at the server-side	Clients communicate with an application(GUI) on the server-side, that makes the system more secure and accessible. This application thereafter communicates with the database system.



Q20. Mention the differences between Unique Key and Primary Key

Unique Key	Primary Key
Unique Key can have a NULL value	The primary key cannot have a NULL value

Each table can have more than one unique key	Each table can have only one primary key

Q21. What is a checkpoint in DBMS and when does it occur?

A checkpoint is a mechanism where all the previous logs are removed from the system and are permanently stored on the storage disk. So, basically, checkpoints are those points from where the transaction log record can be used to recover all the committed data up to the point of crash.

Q22. Mention the differences between Trigger and Stored Procedures

Triggers	Stored Procedures
A special kind of stored procedure that is not called directly by a user. In fact, a trigger is created and is programmed to fire when a specific event occurs.	A group of SQL statements which can be reused again and again. These statements are created and stored in the database.
A trigger cannot be called or execute directly by a user. Only when the corresponding events are fired, triggers are created.	Can execute stored procedures by using the exec command, whenever we want.
You cannot schedule a trigger.	You can schedule a job to execute the stored procedure on a pre-defined time.
Cannot directly call another trigger within a trigger.	Call a stored procedure from another stored procedure.
Parameters cannot be passed as input	Parameters can be passed as input
Cannot return values.	Can return zero or n values.
Transactions are not allowed within a trigger.	You can use transactions within a stored procedure.

Q23. What are the differences between Hash join, Merge join and Nested loops?

Hash join	Merge join	Nested loops

The hash join is used when you have to join large tables.	Merge join is used when projections of the joined tables are sorted on the join columns.	The nested loop consists of an outer loop and an inner loop.

Q24. What do you understand by Proactive, Retroactive and Simultaneous Update?

- Proactive Update: These updates are applied to the database before it becomes effective in the real-world environment.
- Retroactive Update: These retroactive updates are applied to a database after it becomes effective in the real-world environment.
- Simultaneous Update: These updates are applied to the database at the same instance of time as it becomes effective in a real-world environment.

Q25. What are indexes? Mention the differences between the clustered and non-clustered index.

Indexes are data structures responsible for improving the speed of data retrieval operations on a table. This data structure uses more storage space to maintain extra copies of data by using additional writes. So, indexes are mainly used for searching algorithms, where you wish to retrieve data in a quick manner.

The differences between clustered and non-clustered index are as follows:

Clustered Index	Non-clustered Index
A clustered index is faster	Non clustered index is relatively slower
Alters the way records are stored in a database as it sorts out rows by the column which is set to be clustered index	Does not alter the way it was stored but it creates a separate object within a table which points back to the original table rows after searching
One table can only have one clustered index	One table can only have many non clustered indexes