

1) What is DBMS?

DBMS is a collection of programs that facilitates users to create and maintain a database. In other words, DBMS provides us an interface or tool for performing different operations such as the creation of a database, inserting data into it, deleting data from it, updating the data, etc. DBMS is a software in which data is stored in a more secure way as compared to the file-based system. Using DBMS, we can overcome many problems such as- data redundancy, data inconsistency, easy access, more organized and understandable, and so on. There is the name of some popular Database Management System- MySQL, Oracle, SQL Server, Amazon simple DB (Cloud-based), etc.

2) What is a database?

A Database is a logical, consistent and organized collection of data that it can easily be accessed, managed and updated. Databases, also known as electronic databases are structured to provide the facility of creation, insertion, updating of the data efficiently and are stored in the form of a file or set of files, on the magnetic disk, tapes and another sort of secondary devices. Database mostly consists of the objects (tables), and tables include of the records and fields. Fields are the basic units of data storage, which contain the information about a particular aspect or attribute of the entity described by the database. DBMS is used for extraction of data from the database in the form of the queries.

3) What is a database system?

The collection of database and DBMS software together is known as a database system. Through the database system, we can perform many activities such as-

The data can be stored in the database with ease, and there are no issues of data redundancy and data inconsistency.

The data will be extracted from the database using DBMS software whenever required. So, the combination of database and DBMS software enables one to store, retrieve and access data with considerate accuracy and security.

4) What are the advantages of DBMS?

- Redundancy control
- Restriction for unauthorized access
- Provides multiple user interfaces
- Provides backup and recovery
- Enforces integrity constraints
- Ensure data consistency

- Easy accessibility
- Easy data extraction and data processing due to the use of queries

5) What is a checkpoint in DBMS?

The Checkpoint is a type of mechanism where all the previous logs are removed from the system and permanently stored in the storage disk.

There are two ways which can help the DBMS in recovering and maintaining the ACID properties, and they are- maintaining the log of each transaction and maintaining shadow pages. So, when it comes to log based recovery system, checkpoints come into existence. Checkpoints are those points to which the database engine can recover after a crash as a specified minimal point from where the transaction log record can be used to recover all the committed data up to the point of the crash.

6) When does checkpoint occur in DBMS?

A checkpoint is like a snapshot of the DBMS state. Using checkpoints, the DBMS can reduce the amount of work to be done during a restart in the event of subsequent crashes. Checkpoints are used for the recovery of the database after the system crash. Checkpoints are used in the log-based recovery system. When due to a system crash we need to restart the system then at that point we use checkpoints. So that, we don't have to perform the transactions from the very starting.

7) What do you mean by transparent DBMS?

The transparent DBMS is a type of DBMS which keeps its physical structure hidden from users. Physical structure or physical storage structure implies to the memory manager of the DBMS, and it describes how the data stored on disk.

8) What are the unary operations in Relational Algebra?

PROJECTION and SELECTION are the unary operations in relational algebra. Unary operations are those operations which use single operands. Unary operations are SELECTION, PROJECTION, and RENAME.

As in SELECTION relational operators are used for example - $=$, $<=$, $>=$, etc.

9) What is RDBMS?

RDBMS stands for Relational Database Management Systems. It is used to maintain the data records and indices in tables. RDBMS is the form of DBMS which uses the structure to identify and access data concerning the other piece of data in the database. RDBMS is the system that enables you to perform different operations such as- update, insert, delete, manipulate and administer a relational database with minimal difficulties. Most of the time RDBMS use SQL language because it is easily understandable and is used for often.

10) How many types of database languages are?

There are four types of database languages:

- **Data Definition Language (DDL)** e.g., CREATE, ALTER, DROP, TRUNCATE, RENAME, etc. All these commands are used for updating the data that's why they are known as Data Definition Language.
- **Data Manipulation Language (DML)** e.g., SELECT, UPDATE, INSERT, DELETE, etc. These commands are used for the manipulation of already updated data that's why they are the part of Data Manipulation Language.
- **DATA Control Language (DCL)** e.g., GRANT and REVOKE. These commands are used for giving and removing the user access on the database. So, they are the part of Data Control Language.
- **Transaction Control Language (TCL)** e.g., COMMIT, ROLLBACK, and SAVEPOINT. These are the commands used for managing transactions in the database. TCL is used for managing the changes made by DML.

Database language implies the queries that are used for the update, modify and manipulate the data.

11) What do you understand by Data Model?

The Data model is specified as a collection of conceptual tools for describing data, data relationships, data semantics and constraints. These models are used to describe the relationship between the entities and their attributes.

There is the number of data models:

- Hierarchical data model

- network model
 - relational model
 - Entity-Relationship model and so on.
-

12) Define a Relation Schema and a Relation.

A Relation Schema is specified as a set of attributes. It is also known as table schema. It defines what the name of the table is. Relation schema is known as the blueprint with the help of which we can explain that how the data is organized into tables. This blueprint contains no data.

A relation is specified as a set of tuples. A relation is the set of related attributes with identifying key attributes

See this example:

Let r be the relation which contains set tuples $(t_1, t_2, t_3, \dots, t_n)$. Each tuple is an ordered list of n -values $t=(v_1, v_2, \dots, v_n)$.

13) What is a degree of Relation?

The degree of relation is a number of attribute of its relation schema. A degree of relation is also known as Cardinality it is defined as the number of occurrence of one entity which is connected to the number of occurrence of other entity. There are three degree of relation they are one-to-one(1:1), one-to-many(1:M), many-to-one(M:M).

14) What is the Relationship?

The Relationship is defined as an association among two or more entities. There are three type of relationships in DBMS-

One-To-One: Here one record of any object can be related to one record of another object.

One-To-Many (many-to-one): Here one record of any object can be related to many records of other object and vice versa.

Many-to-many: Here more than one records of an object can be related to n number of records of another object.

15) What are the disadvantages of file processing systems?

- Inconsistent
 - Not secure
 - Data redundancy
 - Difficult in accessing data
 - Data isolation
 - Data integrity
 - Concurrent access is not possible
 - Limited data sharing
 - Atomicity problem
-

16) What is data abstraction in DBMS?

Data abstraction in DBMS is a process of hiding irrelevant details from users. Because database systems are made of complex data structures so, it makes accessible the user interaction with the database.

For example: We know that most of the users prefer those systems which have a simple GUI that means no complex processing. So, to keep the user tuned and for making the access to the data easy, it is necessary to do data abstraction. In addition to it, data abstraction divides the system in different layers to make the work specified and well defined.

17) What are the three levels of data abstraction?

Following are three levels of data abstraction:

Physical level: It is the lowest level of abstraction. It describes how data are stored.

Logical level: It is the next higher level of abstraction. It describes what data are stored in the database and what the relationship among those data is.

View level: It is the highest level of data abstraction. It describes only part of the entire database.

For example- User interacts with the system using the GUI and fill the required details, but the user doesn't have any idea how the data is being used. So, the abstraction level is entirely high in VIEW LEVEL.

Then, the next level is for PROGRAMMERS as in this level the fields and records are visible and the programmers have the knowledge of this layer. So, the level of abstraction here is a little low in VIEW LEVEL.

And lastly, physical level in which storage blocks are described.

18) What is DDL (Data Definition Language)?

Data Definition Language (DDL) is a standard for commands which defines the different structures in a database. Most commonly DDL statements are CREATE, ALTER, and DROP. These commands are used for updating data into the database.

19) What is DML (Data Manipulation Language)?

Data Manipulation Language (DML) is a language that enables the user to access or manipulate data as organized by the appropriate data model. For example- SELECT, UPDATE, INSERT, DELETE.

There is two type of DML:

Procedural DML or Low level DML: It requires a user to specify what data are needed and how to get those data.

Non-Procedural DML or High level DML: It requires a user to specify what data are needed without specifying how to get those data.

20) Explain the functionality of DML Compiler.

The DML Compiler translates DML statements in a query language that the query evaluation engine can understand. DML Compiler is required because the DML is the family of syntax element which is very similar to the other programming language which requires compilation. So, it is essential to compile the code in the language which query evaluation engine can understand and then work on those queries with proper output.

21) What is Relational Algebra?

Relational Algebra is a Procedural Query Language which contains a set of operations that take one or two relations as input and produce a new relationship. Relational algebra is the basic set of operations for the relational model. The decisive point of relational algebra is that it is similar to the algebra which operates on the number.

There are few fundamental operations of relational algebra:

- select
 - project
 - set difference
 - union
 - rename, etc.
-

22) What is Relational Calculus?

Relational Calculus is a Non-procedural Query Language which uses mathematical predicate calculus instead of algebra. Relational calculus doesn't work on mathematics fundamentals such as algebra, differential, integration, etc. That's why it is also known as predicate calculus.

There is two type of relational calculus:

- Tuple relational calculus
 - Domain relational calculus
-

23) What do you understand by query optimization?

The term query optimization specifies an efficient execution plan for evaluating a query that has the least estimated cost. The concept of query optimization came into the frame when there were a number of methods, and algorithms existed for the same task then the question arose that which one is more efficient and the process of determining the efficient way is known as query optimization.

There are many benefits of query optimization:

- It reduces the time and space complexity.

- More queries can be performed as due to optimization every query comparatively takes less time.
 - User satisfaction as it will provide output fast
-

24) What do you mean by durability in DBMS?

Once the DBMS informs the user that a transaction has completed successfully, its effect should persist even if the system crashes before all its changes are reflected on disk. This property is called durability. Durability ensures that once the transaction is committed into the database, it will be stored in the non-volatile memory and after that system failure cannot affect that data anymore.

25) What is normalization?

Normalization is a process of analysing the given relation schemas according to their functional dependencies. It is used to minimize redundancy and also used to minimize insertion, deletion and update distractions. Normalization is considered as an essential process as it is used to avoid data redundancy, insertion anomaly, updation anomaly, deletion anomaly.

There most commonly used normal forms are:

- First Normal Form(1NF)
 - Second Normal Form(2NF)
 - Third Normal Form(3NF)
 - Boyce & Codd Normal Form(BCNF)
-

26) What is Denormalization?

Denormalization is the process of boosting up database performance and adding of redundant data which helps to get rid of complex data. Denormalization is a part of database optimization technique. This process is used to avoid the use of complex and costly joins. Denormalization doesn't refer to the thought of not to normalize instead of that denormalization takes place after normalization. In this process, firstly the redundancy of the data will be removed using normalization process than through denormalization process we will add redundant data as per the requirement so that we can easily avoid the costly joins.

27) What is functional Dependency?

Functional Dependency is the starting point of normalization. It exists when a relation between two attributes allow you to determine the corresponding attribute's value uniquely. The functional dependency is also known as database dependency and defines as the relationship which occurs when one attribute in a relation uniquely determines another attribute. It is written as $A \rightarrow B$ which means B is functionally dependent on A.

28) What is the E-R model?

E-R model is a short name for the Entity-Relationship model. This model is based on the real world. It contains necessary objects (known as entities) and the relationship among these objects. Here the primary objects are the entity, attribute of that entity, relationship set, an attribute of that relationship set can be mapped in the form of E-R diagram.

In E-R diagram, entities are represented by rectangles, relationships are represented by diamonds, attributes are the characteristics of entities and represented by ellipses, and data flow is represented through a straight line.

29) What is an entity?

The Entity is a set of attributes in a database. An entity can be a real-world object which physically exists in this world. All the entities have their attribute which in the real world considered as the characteristics of the object.

For example: In the employee database of a company, the employee, department, and the designation can be considered as the entities. These entities have some characteristics which will be the attributes of the corresponding entity.

30) What is an Entity type?

An entity type is specified as a collection of entities, having the same attributes. Entity type typically corresponds to one or several related tables in the database. A characteristic or trait which defines or uniquely identifies the entity is called entity type.

For example, a student has student_id, department, and course as its characteristics.

31) What is an Entity set?

The entity set specifies the collection of all entities of a particular entity type in the database. An entity set is known as the set of all the entities which share the same properties.

For example, a set of people, a set of students, a set of companies, etc.

32) What is an Extension of entity type?

An extension of an entity type is specified as a collection of entities of a particular entity type that are grouped into an entity set.

33) What is Weak Entity set?

An entity set that doesn't have sufficient attributes to form a primary key is referred to as a weak entity set. The member of a weak entity set is known as a subordinate entity. Weak entity set does not have a primary key, but we need a mean to differentiate among all those entries in the entity set that depend on one particular strong entity set.

34) What is an attribute?

An attribute refers to a database component. It is used to describe the property of an entity. An attribute can be defined as the characteristics of the entity. Entities can be uniquely identified using the attributes. Attributes represent the instances in the row of the database.

For example: If a student is an entity in the table then age will be the attribute of that student.

35) What are the integrity rules in DBMS?

Data integrity is one significant aspect while maintaining the database. So, data integrity is enforced in the database system by imposing a series of rules. Those set of integrity is known as the integrity rules.

There are two integrity rules in DBMS:

Entity Integrity : It specifies that "Primary key cannot have a NULL value."

Referential Integrity: It specifies that "Foreign Key can be either a NULL value or should be the Primary Key value of other relation"

36) What do you mean by extension and intension?

Extension: The Extension is the number of tuples present in a table at any instance. It changes as the tuples are created, updated and destroyed. The actual data in the database change quite frequently. So, the data in the database at a particular moment in time is known as extension or database state or snapshot. It is time dependent.

Intension: Intension is also known as Data Schema and defined as the description of the database, which is specified during database design and is expected to remain unchanged. The Intension is a constant value that gives the name, structure of tables and the constraints laid on it.

37) What is System R? How many of its two major subsystems?

System R was designed and developed from 1974 to 1979 at IBM San Jose Research Centre. System R is the first implementation of SQL, which is the standard relational data query language, and it was also the first to demonstrate that RDBMS could provide better transaction processing performance. It is a prototype which is formed to show that it is possible to build a Relational System that can be used in a real-life environment to solve real-life problems.

Following are two major subsystems of System R:

- Research Storage
 - System Relational Data System
-

38) What is Data Independence?

Data independence specifies that "the application is independent of the storage structure and access strategy of data." It makes you able to modify the schema definition at one level without altering the schema definition in the next higher level.

It makes you able to modify the schema definition in one level should not affect the schema definition in the next higher level.

There are two types of Data Independence:

Physical Data Independence: Physical data is the data stored in the database. It is in the bit-format. Modification in physical level should not affect the logical level.

For example: If we want to manipulate the data inside any table that should not change the format of the table.

Logical Data Independence: Logical data is the data about the database. It basically defines the structure. Such as tables stored in the database. Modification in logical level should not affect the view level.

For example: If we need to modify the format of any table, that modification should not affect the data inside it.

NOTE: Logical Data Independence is more difficult to achieve.

39) What are the three levels of data abstraction?

Following are three levels of data abstraction:

Physical level: It is the lowest level of abstraction. It describes how data are stored.

Logical level: It is the next higher level of abstraction. It describes what data are stored in the database and what relationship among those data.

View level: It is the highest level of data abstraction. It describes only part of the entire database.

For example- User interact with the system using the GUI and fill the required details, but the user doesn't have any idea how the data is being used. So, the abstraction level is absolutely high in VIEW LEVEL.

Then, the next level is for PROGRAMMERS as in this level the fields and records are visible and the programmer has the knowledge of this layer. So, the level of abstraction here is a little low in VIEW LEVEL.

And lastly, physical level in which storage blocks are described.

40) What is Join?

The Join operation is one of the most useful activities in relational algebra. It is most commonly used way to combine information from two or more relations. A Join is always performed on the basis of the same or related column. Most complex queries of SQL involve JOIN command.

There are following types of join:

- Inner joins: Inner join is of 3 categories. They are:
 - Theta join
 - Natural join
 - Equi join
 - Outer joins: Outer join have three types. They are:
 - Left outer join
 - Right outer join
 - Full outer join
-

41) What is 1NF?

1NF is the **First Normal Form**. It is the simplest type of normalization that you can implement in a database. The primary objectives of 1NF are to:

- Every column must have atomic (single value)
 - To Remove duplicate columns from the same table
 - Create separate tables for each group of related data and identify each row with a unique column
-

42) What is 2NF?

2NF is the **Second Normal Form**. A table is said to be 2NF if it follows the following conditions:

- The table is in 1NF, i.e., firstly it is necessary that the table should follow the rules of 1NF.
 - Every non-prime attribute is fully functionally dependent on the primary key, i.e., every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted, then even the non_key element will still be saved in the database.
-

43) What is 3NF?

3NF stands for **Third Normal Form**. A database is called in 3NF if it satisfies the following conditions:

- It is in second normal form.
- There is no transitive functional dependency.
- For example: $X \rightarrow Z$

Where:

$X \rightarrow Y$

Y does not \rightarrow X

$Y \rightarrow Z$ so, $X \rightarrow Z$

44) What is BCNF?

BCMF stands for **Boyce-Codd Normal Form**. It is an advanced version of 3NF, so it is also referred to as 3.5NF. BCNF is stricter than 3NF.

A table complies with BCNF if it satisfies the following conditions:

- It is in 3NF.
 - For every functional dependency $X \rightarrow Y$, X should be the super key of the table. It merely means that X cannot be a non-prime attribute if Y is a prime attribute.
-

45) Explain ACID properties

ACID properties are some basic rules, which has to be satisfied by every transaction to preserve the integrity. These properties and rules are:

ATOMICITY: Atomicity is more generally known as 'all or nothing rule.' Which implies all are considered as one unit, and they either run to completion or not executed at all.

CONSISTENCY: This property refers to the uniformity of the data. Consistency implies that the database is consistent before and after the transaction.

ISOLATION: This property states that the number of the transaction can be executed concurrently without leading to the inconsistency of the database state.

DURABILITY: This property ensures that once the transaction is committed it will be stored in the non-volatile memory and system crash can also not affect it anymore.

46) What is stored procedure?

A stored procedure is a group of SQL statements that have been created and stored in the database. The stored procedure increases the reusability as here the code or the procedure is stored into the system and used again and again that makes the work easy, takes less time in processing and decreases the complexity of the system. So, if you have a code which you need to use again and again then save that code and call that code whenever it is required.

47) What is the difference between a DELETE command and TRUNCATE command?

DELETE command: DELETE command is used to delete rows from a table based on the condition that we provide in a WHERE clause.

- DELETE command delete only those rows which are specified with the WHERE clause.
- DELETE command can be rolled back.
- DELETE command maintain a log, that's why it is slow.
- DELETE use row lock while performing DELETE function.

TRUNCATE command: TRUNCATE command is used to remove all rows (complete data) from a table. It is similar to the DELETE command with no WHERE clause.

- The TRUNCATE command removes all the rows from the table.

- The TRUNCATE command cannot be rolled back.
 - The TRUNCATE command doesn't maintain a log. That's why it is fast.
 - TRUNCATE use table log while performing the TRUNCATE function.
-

48) What is 2-Tier architecture?

The **2-Tier architecture** is the same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side.

49) What is the 3-Tier architecture?

The 3-Tier architecture contains another layer between the client and server. Introduction of 3-tier architecture is for the ease of the users as it provides the GUI, which, make the system secure and much more accessible. In this architecture, the application on the client-end interacts with an application on the server which further communicates with the database system.

50) How do you communicate with an RDBMS?

You have to use Structured Query Language (SQL) to communicate with the RDBMS. Using queries of SQL, we can give the input to the database and then after processing of the queries database will provide us the required output.

51) What is the difference between a shared lock and exclusive lock?

Shared lock: Shared lock is required for reading a data item. In the shared lock, many transactions may hold a lock on the same data item. When more than one transaction is allowed to read the data items then that is known as the shared lock.

Exclusive lock: When any transaction is about to perform the write operation, then the lock on the data item is an exclusive lock. Because, if we allow more than one transaction then that will lead to the inconsistency in the database.

52) Describe the types of keys?

There are following types of keys:

Primary key: The Primary key is an attribute in a table that can uniquely identify each record in a table. It is compulsory for every table.

Candidate key: The Candidate key is an attribute or set of an attribute which can uniquely identify a tuple. The Primary key can be selected from these attributes.

Super key: The Super key is a set of attributes which can uniquely identify a tuple. Super key is a superset of the candidate key.

Foreign key: The Foreign key is a primary key from one table, which has a relationship with another table. It acts as a cross-reference between tables.

What are super, primary, candidate and foreign keys?

Ans: A **super key** is a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent. No two rows can have the same value of super key attributes. A **Candidate key** is minimal superkey, i.e., no proper subset of Candidate key attributes can be a superkey. A **Primary Key** is one of the candidate keys. One of the candidate keys is selected as most important and becomes the primary key. There cannot be more than one primary keys in a table. A **Foreign key** is a field (or collection of fields) in one table that uniquely identifies a row of another table. See [this](#) for an example.

What is the difference between having and where clause?

Ans: HAVING is used to specify a condition for a group or an aggregate function used in select statement. The WHERE clause selects before grouping. The HAVING clause selects rows after grouping. Unlike HAVING clause, the WHERE clause cannot contain aggregate functions.

What is Identity?

Ans: Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

What is a view in SQL? How to create one

Ans: A [view](#) is a virtual table based on the result-set of an SQL statement. We can create using create view syntax.

```
CREATE VIEW view_name AS
```

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE condition
```

What are the uses of view?

1. Views can represent a subset of the data contained in a table; consequently, a view can limit the degree of exposure of the underlying tables to the outer world: a given user may have permission to query the view, while denied access to the rest of the base table.
 2. Views can join and simplify multiple tables into a single virtual table
 3. Views can act as aggregated tables, where the database engine aggregates data (sum, average etc.) and presents the calculated results as part of the data
 4. Views can hide the complexity of data; for example a view could appear as Sales2000 or Sales2001, transparently partitioning the actual underlying table
 5. Views take very little space to store; the database contains only the definition of a view, not a copy of all the data which it presents.
 6. Depending on the SQL engine used, views can provide extra security
- Source:

What is a Trigger?

Ans: A [Trigger](#) is a code that associated with insert, update or delete operations. The code is executed automatically whenever the associated query is executed on a table. Triggers can be useful to maintain integrity in database.

What is a stored procedure?

Ans: A [stored procedure](#) is like a function that contains a set of operations compiled together. It contains a set of operations that are commonly used in an application to do some common database tasks.

What is the difference between Trigger and Stored Procedure?

Ans: Unlike Stored Procedures, Triggers cannot be called directly. They can only be associated with queries.

What is a transaction? What are ACID properties?

Ans: A [Database Transaction](#) is a set of database operations that must be treated as whole, means either all operations are executed or none of them. An example can be bank transaction from one account to another account. Either both debit and credit operations must be executed or none of them. [ACID](#) (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably.

What are indexes?

Ans: A [database index](#) is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data. Data can be stored only in one order on disk. To support faster access according to different values, faster search like binary search for different values is desired, For this purpose, indexes are created on tables. These indexes need extra space on disk, but they allow faster search according to different frequently searched values.

What are clustered and non-clustered Indexes?

Ans: Clustered indexes is the index according to which data is physically stored on disk. Therefore, only one clustered index can be created on a given database table.

Non-clustered indexes don't define physical ordering of data, but logical ordering. Typically, a tree is created whose leaf point to disk records. [B-Tree](#) or [B+ tree](#) are used for this purpos

What's the difference between materialized and dynamic view?

Materialized views

- Disk based and are updated periodically based upon the query definition.
- A materialized table is created or updated infrequently and it must be synchronized with its associated base tables.

Dynamic views

- Virtual only and run the query definition each time they are accessed.
- A dynamic view may be created every time that a specific view is requested by the user.

Q. What is embedded and dynamic SQL?

Static or Embedded SQL

- SQL statements in an application that do not change at runtime and, therefore, can be hard-coded into the application.

Dynamic SQL

- SQL statements that are constructed at runtime; for example, the application may allow users to enter their own queries.
- Dynamic SQL is a programming technique that enables you to build SQL statements dynamically at runtime. You can create more general purpose, flexible applications by using dynamic SQL because the full text of a SQL statement may be unknown at compilation.

Q. What is the difference between CHAR and VARCHAR?

- CHAR and VARCHAR differ in storage and retrieval.
- CHAR column length is fixed while VARCHAR length is variable.
- The maximum no. of character CHAR data type can hold is 255 character while VARCHAR can hold up to 4000 character.
- CHAR is 50% faster than VARCHAR.
- CHAR uses static memory allocation while VARCHAR uses dynamic memory allocation.

What is the concept of sub-query in terms of SQL?

Answer: Sub-query is basically the query which is included inside some other query and can also be called as an inner query which is found inside the outer query.

What is the main difference between UNION and UNION ALL?

Answer: UNION and UNION ALL are used to join the data from 2 or more tables but UNION removes duplicate rows and picks the rows which are distinct after combining the data from the tables whereas UNION ALL does not remove the duplicate rows, it just picks all the data from the tables.

Explain Entity, Entity Type, and Entity Set in DBMS?

Answer:

Entity is an object, place or thing which has its independent existence in the real world and about which data can be stored in a database. **For Example,** any person, book, etc.

Entity Type is a collection of entities that have the same attributes. **For Example,** the STUDENT table contains rows in which each row is an entity holding the attributes like name, age, and id of the students, hence STUDENT is an Entity Type which holds the entities having the same attributes.

Entity Set is a collection of entities of the same type. **For Example,** A collection of the employees of a firm.

What are different types of joins in SQL?

Answer: There are 4 types of SQL Joins:

- **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.

- **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
- **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
- **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

Q.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.

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.

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.

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

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.

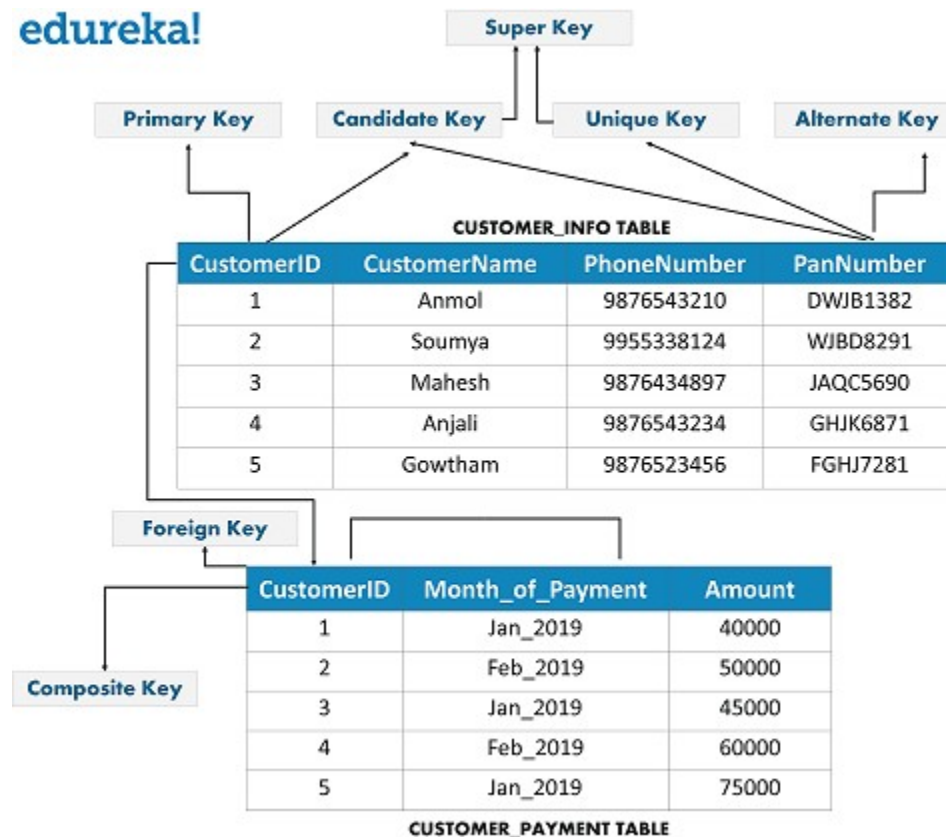


Fig 1: Different Types of Keys in Database – DBMS Interview Questions

- **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.

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.

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

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.

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.

What do you understand by cursor? Mention the different types of cursor

A cursor is a database object which helps in manipulating data, row by row and represents a result set.

The types of cursor are as follows:

- **Implicit cursor:** This type of cursor is declared automatically as soon as the execution of [SQL](#) takes place. Here, the user is not indicated about the declaration of the cursor.
- **Explicit cursor:** This type of cursor is defined by the PL/ SQL, as it handles a query in more than a single row.

Q Explain the terms specialization and generalization

- **Specialization:** Specialization is a process of defining a set of subclasses of the entity type. Here, each subclass will contain all the attributes and relationships of the parent entity. Apart from this, the subclasses may contain additional attributes and relationships specific to itself.
- **Generalization:** Generalization is a process of finding relations, common attributes for a particular set of entities; and finally defining a common superclass for them.

What does Fill Factor concept mean with respect to indexes?

Fill Factor is used to mention the percentage of space left on every leaf-level page, which is packed with data. Usually, the default value is 100.

Q32. What is Index hunting and how does it help in improving query performance?

The process of boosting a collection of indexes is known as Index hunting. This is done as indexes improve the query performance and the speed at which they are processed.

It helps in improving query performance in the following way:

- The best queries are suggested using the query optimizer.
- Index, query distribution and their performance are used as metrics to check the effect
- Databases are tuned into a small collection of problem queries.

Q33. What are the differences between network and hierarchical database model?

Network Database Model	Hierarchical Database Model
Each parent node can have multiple children nodes and vice versa.	A top-down structure where each parent node can have many child nodes. But, a child node can have only a single parent node.
Supports one-to-one, one-to-many, and	Supports one-to-one and one-to-many

many-to-many relationships	relationships
----------------------------	---------------

Define Data Warehousing.

Storage and access of data from the central location in order to take some strategic decision is called Data Warehousing. Enterprise management is used for managing the information whose framework is known as Data Warehousing.

Define B-trees.

A data structure in the form of tree which stores sorted data and searches, insertions, sequential access and deletions are allowed in logarithmic time.

Define Fragmentation.

Fragmentation can be defined as a database feature of server that promotes control on data which is stored at table level by the user.

What is Database partitioning?

Division of logical database into independent complete units for improving its management, availability and performance is called Database partitioning.

Define DDL Interpreter.

DDL statements are interpreted and recorded in tables called metadata.

Enlist the various transaction phases.

The various transaction phases are:

- Analysis Phase.
- Redo Phase
- Undo Phase