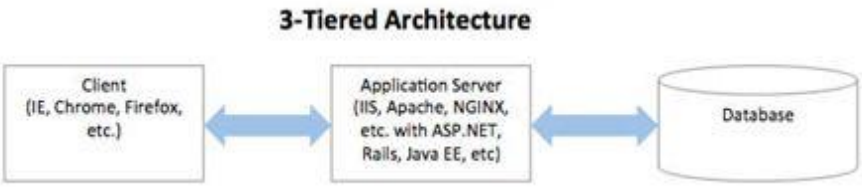
DBMS Interview Questions & Answers

IMPORTANT QUESTIONS IN DBMS ?



1. What are the differences between a DBMS and RDBMS?
2. Explain the terms database and DBMS. Also, mention the different types of DBMS.
3. What are the advantages of DBMS?
4. Mention the different languages present in DBMS
5. What do you understand by query optimization?
6. Do we consider NULL values the same as that of blank space or zero?
7. What do you understand by aggregation and atomicity?
8. What are the different levels of abstraction in the DBMS?
9. What is an entity-relationship model?
10. What do you understand by the terms Entity, Entity Type, and Entity Set in DBMS?

Generic DBMS Interview Questions

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

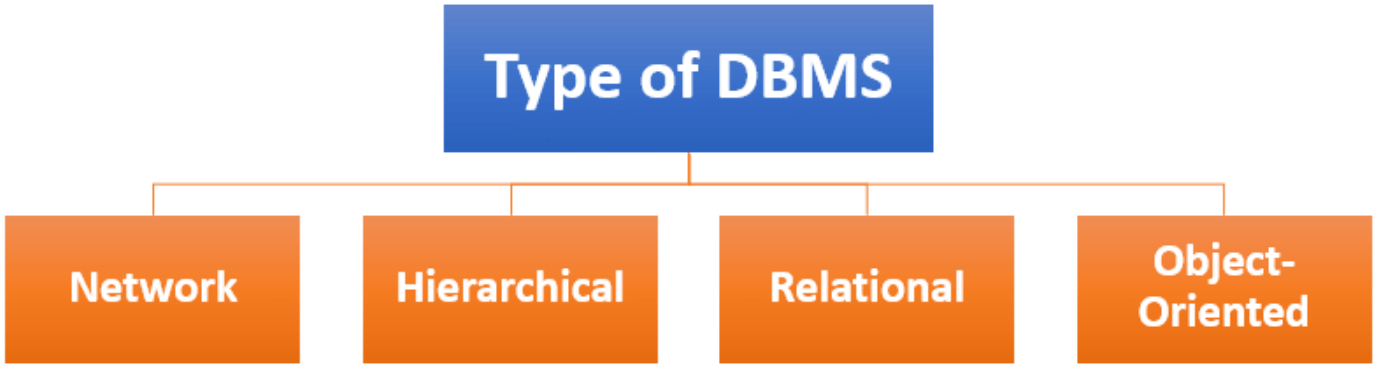
Q2. Explain the terms database and DBMS. Also, mention the different types of DBMS.

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.
- **)** – 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.

DBMS Interview Questions

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.

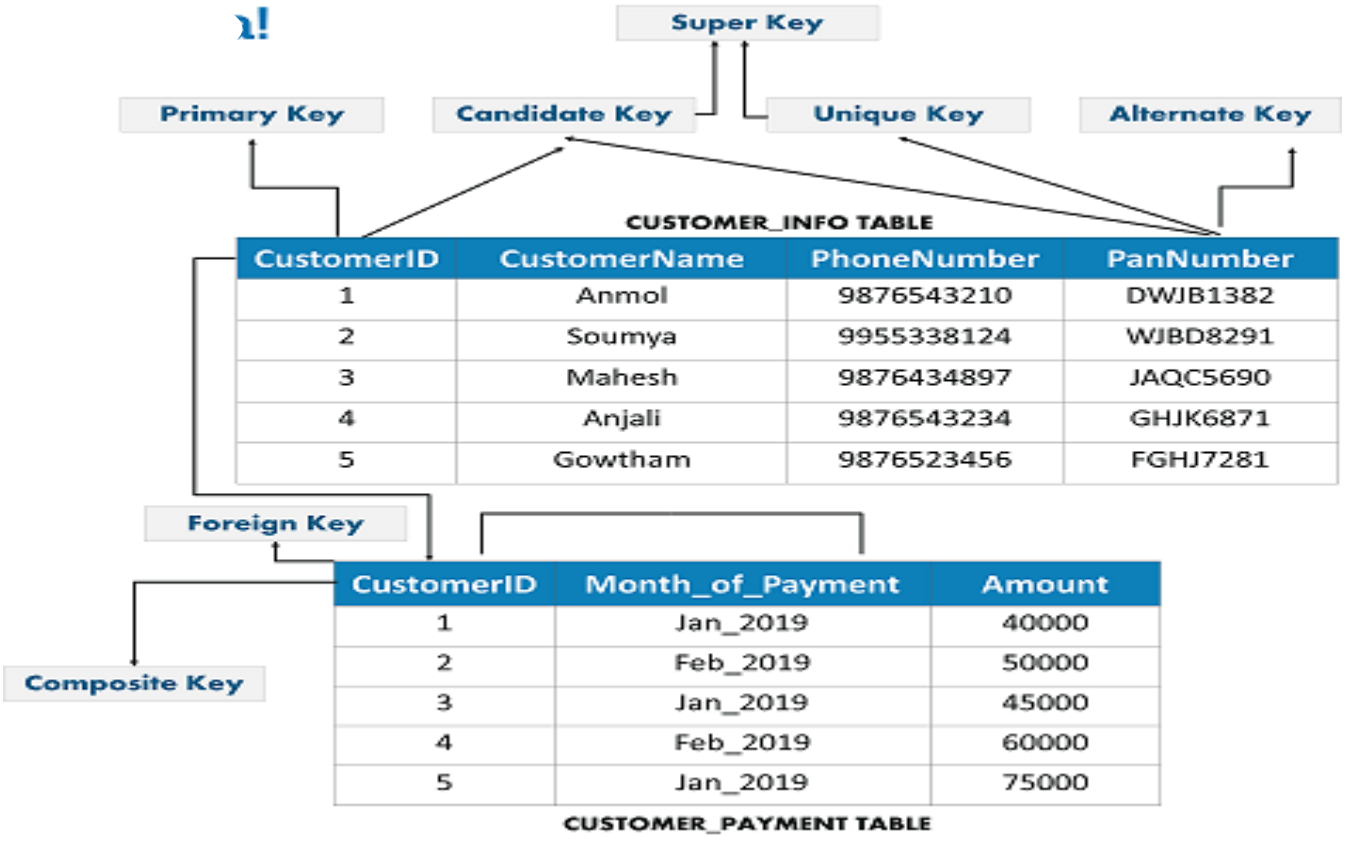


Fig 1: Different Types of Keys in Database – DBMS Interview Questions

- **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

DBMS Interview Questions

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.

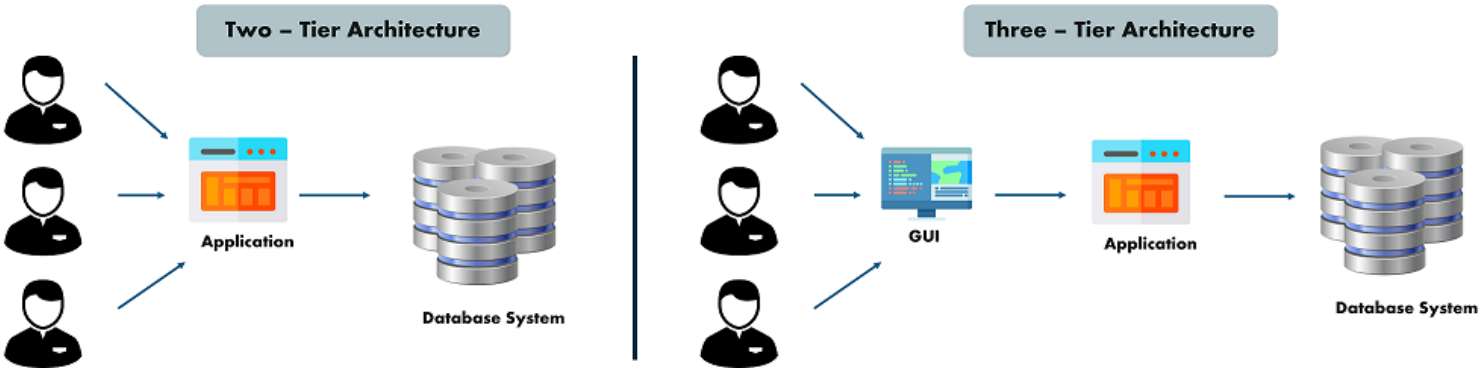


Fig 2: Two-Tier vs Three-Tier Architecture – DBMS Interview Questions

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.

Next, le us discuss one of the most commonly asked DBMS interview questions, that is:

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

Q26. What do you understand by intension and extension?

Intension: Intension or most commonly known as Database schema defines the description of the database. This is specified during the database design and mostly remains unchanged.

Extension: Extension is the number of tuples available in the database at any instance of time. This value keeps changing as and when the tuples are created, updated and destroyed. So, the data present in the database at a specific instance of time is known as the extension of the database or most commonly known as the snapshot of the database.

Q27. 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.

Q28. 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.

Q29. What do you understand by Data Independence?

When you say an application has data independence, it implies that the application is independent of the storage structure and data access strategies of data.

Q30. What are the different integrity rules present in the DBMS?

The different integrity rules present in DBMS are as follows:

- **Entity Integrity:** This rule states that the value of the primary key can never be NULL. So, all the tuples in the column identified as the primary key should have a value.
- **Referential Integrity:** This rule states that either the value of the foreign key is NULL or it should be the primary key of any other relation.

DBMS Interview Questions

Q31. 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 many-to-many relationships	Supports one-tone and one-to-many relationships

Q34. Explain what is a deadlock and mention how it can be resolved?

Deadlock is a situation which occurs when two transactions wait on a resource which is locked or other transaction holds. Deadlocks can be prevented by making all the transactions acquire all the locks at the same instance of time. So, once deadlock occurs, the only way to cure is to abort one of the transactions and remove the partially completed work.

Q35. What are the differences between an exclusive lock and a shared lock?

Exclusive Lock	Shared Lock
An exclusive lock is a lock on a data item when a transaction is about to perform the write operation.	A shared lock allows more than one transaction to read the data items.

Next, in this article on DBMS interview questions,

SQL Interview Questions

Q1. What are the differences between DROP, TRUNCATE and DELETE commands?

DROP	TRUNCATE	DELETE
Used to delete a database, table or a view	Used to delete all rows from a table	Used to delete a row in the table
Data cannot be rollbacked	Data cannot be rollbacked	Data can be rollbacked
A DDL command	A DDL command	A DML command.
Slower than TRUNCATE	Faster than DROP and DELETE	Slower than TRUNCATE
Deletes the full structure of the table	Preserves the structure of the table	Deletes the structure of the row from a table

Q2. What is SQL and where is it used?

SQL aka Structured Query Language is the core of the relational database which is used for accessing and managing the databases. This language is used to manipulate and retrieve data from a structured data format in the form of tables and holds relationships between those tables. So, in layman terms, you can use SQL to communicate with the database.

Q3. What do you understand by sub-queries in SQL?

A subquery is a query inside another query where a query is defined to retrieve data or information back from the database. In a subquery, the outer query is called as the main query whereas the inner query is called subquery. Subqueries are always executed first and the result of the subquery is passed on to the main query. It can be nested inside a SELECT, UPDATE or any other query. A subquery can also use any comparison operators such as >,< or =.

Q**4. Mention the differences between UNION and UNION ALL**

UNION	UNION ALL
Combines the result of two or more SELECT statements consisting of distinct values	Combines the result set of two or more SELECT statements consisting of duplicate values
Syntax: UNION	**Syntax:** UNION ALL
Has low performance than UNION ALL, as duplicate rows need to be removed.	Has better performance than UNION, as duplicate rows need not have to be removed.

Q5. What do you understand by CLAUSE in SQL?

CLAUSE in SQL is used to limit the result set by mentioning a condition to the query. So, you can use a CLAUSE to filter rows from the entire set of records.

Example: WHERE HAVING clause.

Q6. Mention the differences between HAVING and WHERE clause?

HAVING	WHERE
Used only with SELECT statement	Used in a GROUP BY clause
Used with the GROUP BY function in a query	Applied to each row before they are a part of the GROUP BY function in a query

Q7. How can you perform pattern matching in SQL?
You can perform pattern matching in SQL by using the LIKE operator. With the LIKE operator, you can use the following symbols:

1. **%**(Percentage sign) – To match zero or more characters.
2. _ **(**Underscore) –To match exactly one character.

Example:

SELECT * FROM Customers WHERE CustomerName LIKE 's%'

SELECT * FROM Customers WHERE CustomerName like 'xyz_'

Q8. Mention few case manipulation functions in SQL
There are three case manipulation functions in SQL, namely:

LOWER: This function returns the string in lowercase. It takes a string as an argument and returns it by converting it into lower case.

Syntax: LOWER('string')

UPPER: This function returns the string in uppercase. It takes a string as an argument and returns it by converting it into uppercase.

```
Syntax:  UPPER('string')
```

INITCAP: This function returns the string with the first letter in uppercase and the rest of the letters in lowercase.

Syntax: INITCAP('string')

Q**9. What are joins in SQL and what are the different types of joins?**
A JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there. There are 4 joins in SQL namely:

- Inner Join
- Right Join
- Left Join
- Full Join

Q10. What do you understand by the view and mention the steps to create, update and drop a view?
A view in SQL is a single table, which is derived from other tables. So, a view contains rows and columns similar to a real table and has fields from one or more table.

To create a view, use the following syntax:

```
CREATE VIEW ViewName AS
SELECT Column1, Column2, ..., ColumnN
FROM TableName
WHERE Condition;
```

To update a view, use the following syntax:

```
CREATE VIEW OR REPLACE ViewName AS
SELECT Column1, Column2, ..., ColumnN
FROM TableName
WHERE Condition;
```

To drop a view, use the following syntax:

```
DROP VIEW ViewName;
```

Next, in this article on DBMS interview questions, let us discuss the most frequently asked queries about SQL.

SQL Query Based Interview Questions

Q1. Write a query to create a duplicate table with and without data present?

Consider you have a table named Customers, having details such as CustomerID, CustomerName and so on. Now, if you want to create a duplicate table named 'DuplicateCustomer' with the data present in it, you can mention the following query:

```
CREATE TABLE DuplicateCustomer AS SELECT * FROM Customers;
```

Similarly, if you want to create a duplicate table without the data present, mention the following query:

```
CREATE TABLE DuplicateCustomer AS SELECT * FROM Customers WHERE 1=2;
```

Q2. Mention a query to calculate the even and odd records from a table

To write a query to calculate the even and odd records from a table, you can write two different queries by using the MOD function.

So, if you want to retrieve the even records from a table, you can write a query as follows:

```
SELECT CustomerID FROM (SELECT rowno, CustomerID from Customers) where mod(rowno,2)=0;
```

Similarly, if you want to retrieve the odd records from a table, you can write a query as follows:

```
SELECT CustomerID FROM (SELECT rowno, CustomerID from Customers) where mod(rowno,2)=1;
```

Q3. Write a query to remove duplicate rows from a table?

To remove duplicate rows from a table, you have to initially select the duplicate rows from the table without using the DISTINCT keyword. So, to select the duplicate rows from the table, you can write a query as follows:

```
SELECT CustomerNumber FROM Customers WHERE ROWID (SELECT MAX (rowid) FROM Customers C
WHERE CustomerNumber = C.CustomerNumber);
```

Now, to delete the duplicate records from the Customers table, mention the following query:

```
1       DELETE FROM Customers WHERE ROWID(SELECT MAX (rowid) FROM
          Customers C WHERE CustomerNumber = C.CustomerNumber);
```

Q4. Mention a query to add email validation to your database

Well, there are multiple ways to add email validation to your database, but one out the lot is as

```
1 SELECT Email FROM Customers WHERE NOT REGEXP_LIKE(Email, &lsquo;[A-Z0-9._%+-]+@[A-Z0-9.-
                        ]+.[A-Z]{2,4}&rsquo;, &lsquo;i&rsquo;);
```

Q5. Write a query to retrieve the last day of next month in Oracle.

To write a query to retrieve the last day of the next month in Oracle, you can write a query as follows:

```
1                SELECT LAST_DAY (ADD_MONTHS (SYSDATE,1)) from dual;
```

CREATED BY – ATUL KUMAR (LINKEDIN)

https://www.linkedin.com/in/atul3

DBMS Interview and Aptitude Questions with Answers

1. What is a DBMS?

A DBMS is software that allows users to create, manage, and manipulate databases. It provides an interface between the data and the users or applications.

2. What are the types of DBMS?

- Hierarchical DBMS
- Network DBMS
- Relational DBMS (RDBMS)
- Object-Oriented DBMS

3. What is the difference between DBMS and RDBMS?

DBMS stores data in files without relationships, whereas RDBMS stores data in tabular form and supports relationships using keys.

4. What is SQL?

SQL (Structured Query Language) is the standard language for accessing and manipulating relational databases.

5. Types of SQL commands?

- DDL: CREATE, ALTER, DROP
- DML: SELECT, INSERT, UPDATE, DELETE
- DCL: GRANT, REVOKE
- TCL: COMMIT, ROLLBACK

6. What is normalization?

Normalization is the process of organizing data to reduce redundancy and improve data integrity.

7. Explain different normal forms.

- 1NF: No repeating groups

- 2NF: No partial dependency

- 3NF: No transitive dependency

- BCNF: Stronger version of 3NF

8. What is a primary key?

A primary key uniquely identifies each row in a table and cannot contain NULL or duplicate values.

9. What is a foreign key?

A foreign key in one table refers to the primary key in another table, creating a relationship between the two.

10. What is indexing in DBMS?

Indexing improves the speed of data retrieval operations by creating a data structure (index) on specified columns.

11. What is a view?

A view is a virtual table based on a SQL query. It doesn't store data physically but presents data from one or more tables.

12. What is ACID in DBMS?

Atomicity, Consistency, Isolation, Durability - properties ensuring reliable processing of database transactions.

13. What is a transaction?

A transaction is a logical unit of work that includes one or more SQL operations executed as a single unit.

14. What is a trigger?

A trigger is a stored procedure that automatically executes in response to certain events on a table (e.g., INSERT, UPDATE).

15. Difference between WHERE and HAVING?

WHERE filters rows before grouping; HAVING filters groups after aggregation.

16. What is a candidate key?

A candidate key is a column or group of columns that can uniquely identify a row. One is chosen as the primary key.

17. What is the difference between INNER JOIN and OUTER JOIN?

- INNER JOIN: Only matching rows

- OUTER JOIN: Includes non-matching rows from one or both tables

18. What is denormalization?

Denormalization is the process of introducing redundancy to improve read performance.

19. What is a schema?

A schema is the logical structure of the database defined by tables, views, relationships, etc.

20. What is a cursor?

A cursor is a pointer used to retrieve rows one-by-one from a query result set.

Aptitude Sample Q1

Which of the following is a valid DDL command?

A) SELECT

B) INSERT

C) DELETE

D) CREATE

Answer: D) CREATE

Aptitude Sample Q2

In which normal form are transitive dependencies removed?

Answer: 3NF

Aptitude Sample Q3

Which SQL clause is used to filter grouped records?

Answer: HAVING

Aptitude Sample Q4

Which property of a transaction ensures that it can be rolled back on failure?

Answer: Atomicity