

Example SQL tester interview questions

Here are some common SQL tester interview questions:

1. What is SQL?

“Structured Query Language is an American National Standards Institute database language used as a communication tool to create and access databases. SQL supports software applications by performing tasks such as retrieving, updating, inserting and deleting data. SQL makes it possible to execute queries, insert and update records, create and delete databases and tables, and more.”

[Upload your resume on Indeed](#)

Let employers find you when you create an Indeed Resume

2. What is a DBMS?

“DBMS stands for 'database management system,' which is a program used to help maintain, monitor, create, deploy and control the use of a database. It also acts as a file manager for the data kept in a database, no matter the size or complexity of the database.”

3. What are tables in SQL?

“The table is a collection of records organized into a set of rows and columns to form a model. There are no precise upper limits to the number of records that can be defined by rows in the table.”

4. What are the different clauses used in SQL?

“An SQL clause is defined to limit the queried results to certain specified conditions.

GROUP BY: used in aggregation to arrange identical data into groups, the GROUP BY clause follows the WHERE clause in a SELECT statement and is followed by the ORDER BY clause

HAVING: used to specify a search condition in a GROUP BY clause, HAVING can be used in the absence of a GROUP BY clause by using a WHERE clause

ORDER BY: sorts the result set in ascending (default) or descending (using DESC keyword) order

WHERE: used to define the condition of the records to be extracted”

5. What are the factors to check during database testing?

“Every functionality should be tested, including field size validation, constraints, indexing, stored procedures, data validity, data integrity, triggers and functions, and database performance, but checking the accuracy of added and deleted records should be monitored more closely.”

6. What is data-driven testing?

“Data-driven testing is an automation framework within which test data is stored in a table or spreadsheet, where input values are read from data files and stored in various test scripts. This is useful to avoid the need to create individual tests for each set of data, which can be a cumbersome process. The framework keeps the data quarantined, and the same test script can be used to generate results for multiple combinations of input test data. Input data can be stored as XLS, XML, CSV and database formats. There are four types of data-driven testing:

- *Key-driven: Dynamic test data is submitted via keyboard to retest an application with new input values to validate the original calculation.*
- *Flat files (.txt, .doc): Retesting using the data in a flat-file format.*
- *Front-end object: Automation scripts are created by considering values like list, menu, table, data window and ocx.*
- *Excel: This test runs the script for multiple inputs, which are stored in an Excel sheet.”*

7. What are JOINS in SQL, and what types of JOINS are there?

“A JOIN is a keyword used to retrieve data from multiple tables based upon the relationship between the fields of the tables and present the results as a single set. The various types of JOINS commonly include INNER JOIN, OUTER JOIN, LEFT JOIN and RIGHT JOIN.”

8. What is DML?

“DML stands for 'data manipulation language.' DML is a computer programming language that facilitates the selection, insertion, deletion and updating of data in a database. DML parallels SQL in that it allows users to retrieve and use data in a relational database.”

9. What are DCL commands?

“DCL stands for 'Data Control Language' and is a component of SQL that controls access to data stored in a database. The two types of DCL commands are:

Grant: grants users access to the database

Revoke: stops permitting users to access the database”

10. Does QTP support SQL queries?

“QTP doesn't have built-in functionality for database connectivity, but VBScript language can be used to connect and interact with databases using ADODB objects, divided into four methods:

- Connection: forms a connection with a database
- Command: executes an SQL command
- Fields: retrieves the correct column from a record
- Recordset: retrieves data from a database”

11. How is data-load testing performed?

“You’ll need to have access to the source and destination databases.

1. First, check the compatibility between the source database and destination database via the DTS package.
2. Then, open the corresponding DTS package in SQL Enterprise Manager and run the DTS package.
3. Compare the Source and the Target column data.
4. Check the number to rows of Source and Target.
5. Update the Source database, then check to make sure the changes have been reflected on the destination database.
6. Check for NULLs.
7. Finally, check the volume and response time of the database server in processing queries.”

12. What is retesting, and how is it different from data-driven testing?

“Retesting involves executing the same test with different input values after previously testing, finding the problem and repairing it. Retesting differs from data-driven testing in that retesting is a manual process where testing is done with a whole new data set, while data-driven testing is an automated process that tests the application with multiple sets of test data.”

13. What is performance testing?

“Performance testing is a highly specialized software testing technique that helps to determine how a system performs in terms of speed, sensitivity and stability under a heavy workload.”

14. What are SQL constraints?

“Constraints are statements used to establish the rules for all records in the table. If any action violates a constraint, that action will be aborted. Constraints are defined while creating the database itself with CREATE TABLE statement, or after the table is created, by using ALTER TABLE statement. The five major constraints used in SQL are:

- NOT NULL: indicates that the column is required to have some value; it cannot be left null
- UNIQUE: ensures that each row and column has unique value; no value is being repeated in any other row or column
- PRIMARY KEY: used in association with NOT NULL and UNIQUE constraints to identify a particular unique record

- **FOREIGN KEY:** used to ensure the referential integrity of data in the table and also matches the value in one table with another using PRIMARY KEY
- **CHECK:** used to ensure the value in columns obeys specified conditions”

Q #1) What is SQL?

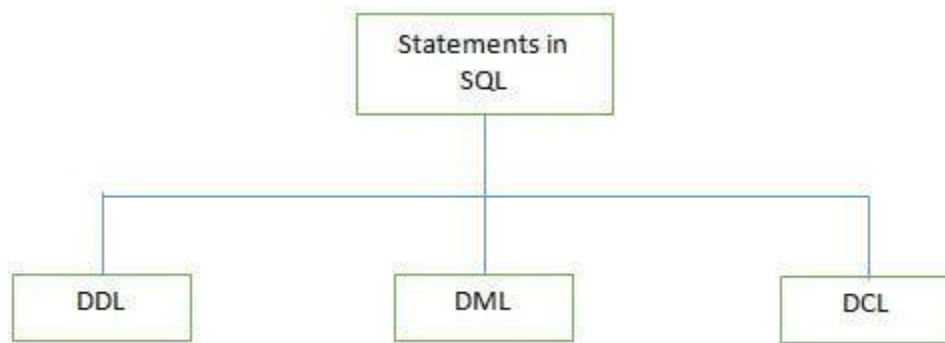
Answer: Structured Query Language SQL is a database tool that is used to create and access the database to support software applications.

Q #2) What are tables in SQL?

Answer: The table is a collection of record and its information at a single view.

Q #3) What are the different types of statements supported by SQL?

Answer:



There are 3 types of SQL statements:

a) DDL (Data Definition Language): It is used to define the database structure such as tables. It includes three statements such as CREATE, ALTER, and DROP.

Also read ==> [MySQL Create Table Tutorial](#)

Some of the DDL Commands are listed below:

CREATE: It is used for creating the table.

```

CREATE TABLE table_name
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
  
```

ALTER: The ALTER table is used for modifying the existing table object in the database.

```
ALTER TABLE table_name  
  ADD column_name datatype
```

OR

```
ALTER TABLE table_name  
  DROP COLUMN column_name
```

b) DML (Data Manipulation Language): These statements are used to manipulate the data in records. Commonly used DML statements are INSERT, UPDATE, and DELETE.

The SELECT statement is used as a partial DML statement, used to select all or relevant records in the table.

c) DCL (Data Control Language): These statements are used to set privileges such as GRANT and REVOKE database access permission to the specific user.

Q #4) How do we use the DISTINCT statement? What is its use?

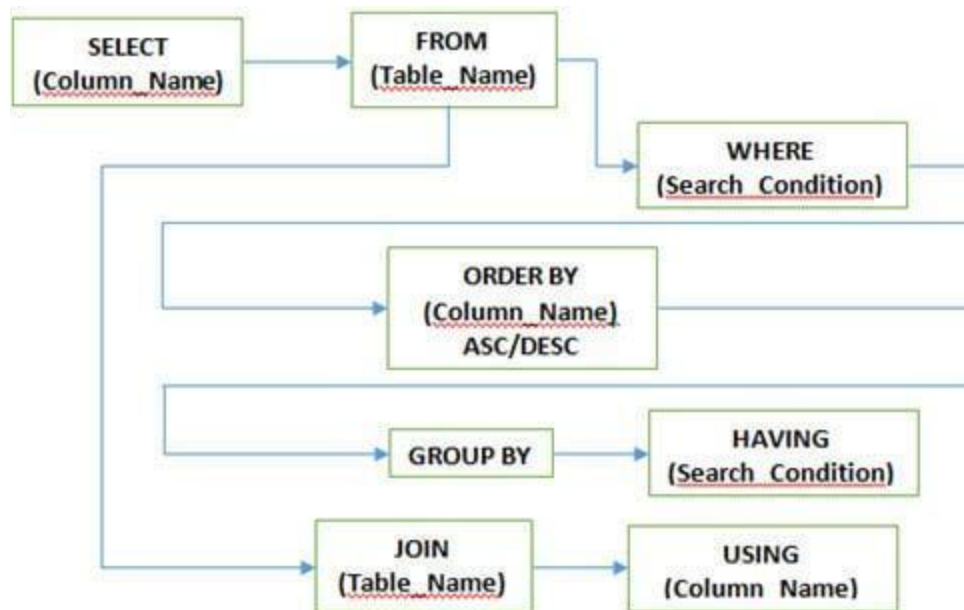
Answer: The DISTINCT statement is used with the SELECT statement. If the record contains duplicate values then the DISTINCT statement is used to select different values among duplicate records.

Syntax:

```
SELECT DISTINCT column_name(s)  
  FROM table_name;
```

Q #5) What are the different Clauses used in SQL?

Answer:



WHERE Clause: This clause is used to define the condition, extract and display only those records which fulfill the given condition.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition;
```

GROUP BY Clause: It is used with SELECT statement to group the result of the executed query using the value specified in it. It matches the value with the column name in tables and groups the end result accordingly.

Further reading => [MySQL Group By Tutorial](#)

Syntax:

```
SELECT column_name(s)
FROM table_name
GROUP BY column_name;
```

HAVING clause: This clause is used in association with the GROUP BY clause. It is applied to each group of results or the entire result as a single group. It is much similar as WHERE clause but the only difference is you cannot use it without GROUP BY clause

Syntax:

```
SELECT column_name(s)
```

```
FROM table_name  
GROUP BY column_name  
HAVING condition;
```

ORDER BY clause: This clause is used to define the order of the query output either in ascending (ASC) or in descending (DESC). Ascending (ASC) is set as the default one but descending (DESC) is set explicitly.

Syntax:

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
ORDER BY column_name ASC|DESC;
```

USING clause: USING clause comes in use while working with SQL JOIN. It is used to check equality based on columns when tables are joined. It can be used instead of the ON clause in JOIN.

Syntax:

```
SELECT column_name(s)  
FROM table_name  
JOIN table_name  
USING (column_name);
```

Q #6) Why do we use SQL constraints? Which constraints we can use while creating a database in SQL?

Answer: Constraints are used to set the rules for all records in the table. If any constraints get violated then it can abort the action that caused it.

Constraints are defined while creating the database itself with the CREATE TABLE statement or even after the table is created once with the ALTER TABLE statement.

There are 5 major constraints are used in SQL, such as

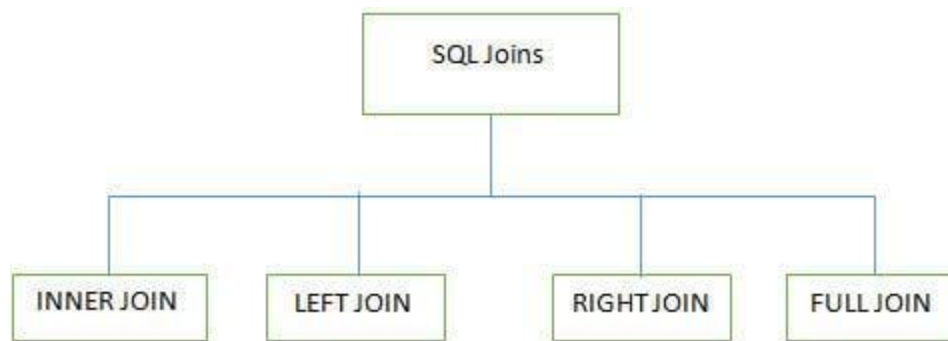
- **NOT NULL:** That indicates that the column must have some value and cannot be left NULL.
- **UNIQUE:** This constraint is used to ensure that each row and column has a unique value and no value is being repeated in any other row or column.
- **PRIMARY KEY:** This constraint is used in association with NOT NULL and UNIQUE constraints such as on one or the combination of more than one column to identify the particular record with a unique identity.
- **FOREIGN KEY:** It is used to ensure the referential integrity of data in the table. It matches the value in one table with another using the PRIMARY KEY.

- **CHECK:** It ensures whether the value in columns fulfills the specified condition.

Suggested Reading => [Boost Your Career with these SQL Certifications](#)

Q #7) What are different JOINS used in SQL?

Answer:



4 major types of Joins are used while working on multiple tables in SQL databases:

INNER JOIN: It is also known as SIMPLE JOIN which returns all rows from BOTH tables when it has at least one matching column.

Syntax:

```

SELECT column_name(s)
FROM table_name1 
INNER JOIN table_name2
ON column_name1=column_name2;
  
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
INNER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

There will be 4 records selected. **Results are:**

Emp_Id	Joining_Date
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

Employee and **Orders** tables have a matching *customer_id* value.

LEFT JOIN (LEFT OUTER JOIN): This join returns all rows from the LEFT table and its matched rows from a RIGHT table.

Syntax:

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01
NULL	NULL	NULL	2016/03/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
LEFT OUTER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

There will be 4 records selected. **You will see the following results:**

Emp_Id	Joining_Date
NULL	NULL
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

RIGHT JOIN (RIGHT OUTER JOIN): This joins returns all rows from the RIGHT table and its matched rows from the LEFT table.

Syntax:

```
SELECT column_name(s)
```

```
FROM table_name1
RIGHT JOIN table_name2
ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01
NULL	NULL	NULL	2016/03/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date FROM Employee
RIGHT JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

Output:

Emp_id Joining_Date

E0012 2016/04/18

E0013 2016/04/19

E0014 2016/05/01

FULL JOIN (FULL OUTER JOIN): This joins returns all results when there is a match either in the RIGHT table or in the LEFT table.

Syntax:

```
SELECT column_name(s)
FROM table_name1
FULL OUTER JOIN table_name2
ON column_name1=column_name2;
```

For Example,

In this example, we have a table **Employee** with the following data:

Emp_Id	Last_Name	First_Name	Job_Role
E0011	Verma	Akhil	Administration
E0012	Samson	Nikita	Asst. Manager
E0013	Jordan	Nil	In charge
E0014	Smith	Joe	Technician

The second table's name is **Joining**.

Emp_Id	Last_Name	First_Name	Joining_Date
E0012	Verma	Akhil	2016/04/18
E0013	Samson	Nikita	2016/04/19
E0014	Jordan	Nil	2016/05/01
NULL	NULL	NULL	2016/03/01

Enter the following SQL statement:

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
FULL OUTER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

There will be 8 records selected. **These are the results that you should see.**

Emp_Id	Joining_Date
NULL	NULL
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01
NULL	2016/03/01
E0012	2016/04/18
E0013	2016/04/19
E0014	2016/05/01

Also Read => [MySQL Join Tutorial](#)

Q #8) What are transactions and their controls?

Answer: A transaction can be defined as the sequence task that is performed on databases in a logical manner to gain certain results. Operations like Creating, updating, deleting records performed in the database come from transactions.

In simple words, we can say that a transaction means a group of SQL queries executed on database records.

There are 4 transaction controls such as

- **COMMIT:** It is used to save all changes made through the transaction.
- **ROLLBACK:** It is used to roll back the transaction. All changes made by the transaction are reverted back and the database remains as before.
- **SET TRANSACTION:** Set the name of the transaction.
- **SAVEPOINT:** It is used to set the point where the transaction is to be rolled back.

Q #9) What are the properties of the transaction?

Answer: Properties of the transaction are known as ACID properties. These are:

- **Atomicity:** Ensures the completeness of all transactions performed. Checks whether every transaction is completed successfully or not. If not, then the transaction is aborted at the failure point and the previous transaction is rolled back to its initial state as changes are undone.

- **Consistency:** Ensures that all changes made through successful transactions are reflected properly on the database.
- **Isolation:** Ensures that all transactions are performed independently and changes made by one transaction are not reflected on others.
- **Durability:** Ensures that the changes made in the database with committed transactions persist as it is even after a system failure.

Q #10) How many Aggregate functions are available in SQL?

Answer: SQL Aggregate functions determine and calculate values from multiple columns in a table and return a single value.

There are 7 aggregate functions in SQL:

- **AVG():** Returns the average value from specified columns.
- **COUNT():** Returns number of table rows.
- **MAX():** Returns the largest value among the records.
- **MIN():** Returns smallest value among the records.
- **SUM():** Returns the sum of specified column values.
- **FIRST():** Returns the first value.
- **LAST():** Returns last value.

Q #11) What are Scalar functions in SQL?

Answer: Scalar functions are used to return a single value based on the input values.

Scalar Functions are as follows:

- **UCASE():** Converts the specified field in the upper case.
- **LCASE():** Converts the specified field in lower case.
- **MID():** Extracts and returns character from the text field.
- **FORMAT():** Specifies the display format.
- **LEN():** Specifies the length of the text field.
- **ROUND():** Rounds up the decimal field value to a number.

Q #12) What are triggers?

Answer: Triggers in SQL is kind of stored procedures used to create a response to a specific action performed on the table such as INSERT, UPDATE or DELETE. You can invoke triggers explicitly on the table in the database.

Action and Event are two main components of SQL triggers. When certain actions are performed, the event occurs in response to that action.

Syntax:

```
CREATE TRIGGER name {BEFORE|AFTER} (event [OR...])  
ON table_name [FOR [EACH] {ROW|STATEMENT}]  
EXECUTE PROCEDURE functionname {arguments}
```

Q #13) What is View in SQL?

Answer: A View can be defined as a virtual table that contains rows and columns with fields from one or more tables.

Syntax:

```
CREATE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

Q #14) How we can update the view?

Answer: SQL CREATE and REPLACE can be used for updating the view.

Execute the below query to update the created view.

Syntax:

```
CREATE OR REPLACE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

Q #15) Explain the working of SQL Privileges?

Answer: SQL GRANT and REVOKE commands are used to implement privileges in SQL multiple user environments. The administrator of the database can grant or revoke privileges to or from users of database objects by using commands like SELECT, INSERT, UPDATE, DELETE, ALL, etc.

GRANT Command: This command is used to provide database access to users other than the administrator.

Syntax:

```
GRANT privilege_name  
ON object_name  
TO {user_name|PUBLIC|role_name}  
[WITH GRANT OPTION];
```

In the above syntax, the GRANT option indicates that the user can grant access to another user too.

REVOKE command: This command is used to provide database deny or remove access to database objects.

Syntax:

```
REVOKE privilege_name  
ON object_name  
FROM {user_name|PUBLIC|role_name};
```

Q #16) How many types of Privileges are available in SQL?

Answer: There are two types of privileges used in SQL, such as

- **System privilege:** System privilege deals with the object of a particular type and provides users the right to perform one or more actions on it. These actions include performing administrative tasks, ALTER ANY INDEX, ALTER ANY CACHE GROUP, CREATE/ALTER/DELETE TABLE, CREATE/ALTER/DELETE VIEW etc.
- **Object privilege:** This allows to perform actions on an object or object of another user(s) viz. table, view, indexes etc. Some of the object privileges are EXECUTE, INSERT, UPDATE, DELETE, SELECT, FLUSH, LOAD, INDEX, REFERENCES etc.

Q #17) What is SQL Injection?

Answer: SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database in a way that once it is executed, the database is exposed to an attacker for the attack. This technique is usually used for attacking data-driven applications to have access to sensitive data and perform administrative tasks on databases.

For Example,

```
SELECT column_name(s) FROM table_name WHERE condition;
```

Q #18) What is SQL Sandbox in SQL Server?

Answer: SQL Sandbox is a safe place in the SQL server environment where untrusted scripts are executed. There are 3 types of SQL sandbox:

- **Safe Access Sandbox:** Here a user can perform SQL operations such as creating stored procedures, triggers etc. but cannot have access to the memory as well as cannot create files.
- **External Access Sandbox:** Users can access files without having the right to manipulate the memory allocation.

- **Unsafe Access Sandbox:** This contains untrusted codes where a user can have access to memory.

Q #19) What is the difference between SQL and PL/SQL?

Answer: SQL is a Structured Query Language to create and access databases whereas PL/SQL comes with procedural concepts of programming languages.

Q #20) What is the difference between SQL and MySQL?

Answer: SQL is a Structured Query Language that is used for manipulating and accessing the relational database. On the other hand, MySQL itself is a relational database that uses SQL as the standard database language.

Q #21) What is the use of the NVL function?

Answer: NVL function is used to convert the null value to its actual value.

Q #22) What is the Cartesian product of the table?

Answer: The output of Cross Join is called a Cartesian product. It returns rows combining each row from the first table with each row of the second table. **For Example,** if we join two tables having 15 and 20 columns the Cartesian product of two tables will be $15 \times 20 = 300$ rows.

Q #23) What do you mean by Subquery?

Answer: Query within another query is called as Subquery. A subquery is called inner query which returns output that is to be used by another query.

Q #24) How many row comparison operators are used while working with a subquery?

Answer: There are 3-row comparison operators that are used in subqueries such as IN, ANY and ALL.

Q #25) What is the difference between clustered and non-clustered indexes?

Answer: The differences between the two are as follows:

- One table can have only one clustered index but multiple non-clustered indexes.
- Clustered indexes can be read rapidly rather than non-clustered indexes.
- Clustered indexes store data physically in the table or view whereas, non-clustered indexes do not store data in the table as it has separate structure from the data row.

Q #26) What is the difference between DELETE and TRUNCATE?

Answer: The differences are:

- The basic difference in both is DELETE command is DML command and the TRUNCATE command is DDL.
- DELETE command is used to delete a specific row from the table whereas the TRUNCATE command is used to remove all rows from the table.
- We can use the DELETE command with WHERE clause but cannot use the TRUNCATE command with it.

Q #27) What is the difference between DROP and TRUNCATE?

Answer: TRUNCATE removes all rows from the table which cannot be retrieved back, DROP removes the entire table from the database and it also cannot be retrieved back.

Q #28) How to write a query to show the details of a student from Students table whose name start with K?

Answer: Query:

```
SELECT * FROM Student WHERE Student Name like 'K%';
```

Here 'like' operator is used to perform pattern matching.

Q #29) What is the difference between Nested Subquery and Correlated Subquery?

Answer: Subquery within another subquery is called Nested Subquery. If the output of a subquery depends on column values of the parent query table then the query is called Correlated Subquery.

```
SELECT adminid(SELECT Firstname+' '+Lastname&nbsp;&nbsp; FROM Employee WHERE
empid=emp. adminid)AS EmpAdminId FROM Employee;
```

The result of the query is the details of an employee from the Employee table.

Q #30) What is Normalization? How many Normalization forms are there?

Answer: Normalization is used to organize the data in such a manner that data redundancy will never occur in the database and avoid insert, update and delete anomalies.

There are 5 forms of Normalization:

- **First Normal Form (1NF):** It removes all duplicate columns from the table. It creates a table for related data and identifies unique column values.
- **First Normal Form (2NF):** Follows 1NF and creates and places data subsets in an individual table and defines the relationship between tables using the primary key.
- **Third Normal Form (3NF):** Follows 2NF and removes those columns which are not related through the primary key.

- **Fourth Normal Form (4NF):** Follows 3NF and does not define multi-valued dependencies. 4NF is also known as BCNF.

Q #31) What is a Relationship? How many types of Relationships are there?

Answer: The relationship can be defined as the connection between more than one table in the database.

There are 4 types of relationships:

- One to One Relationship
- Many to One Relationship
- Many to Many Relationship
- One to Many Relationship

Q #32) What do you mean by Stored Procedures? How do we use it?

Answer: A stored procedure is a collection of SQL statements that can be used as a function to access the database. We can create these stored procedures earlier before using it and can execute them wherever required by applying some conditional logic to it. Stored procedures are also used to reduce network traffic and improve performance.

Syntax:

```
CREATE Procedure Procedure_Name
(
//Parameters
)
AS
BEGIN
SQL statements in stored procedures to update/retrieve records
END
```

Q #33) State some properties of Relational databases?

Answer: Properties are as follows:

- In relational databases, each column should have a unique name.
- The sequence of rows and columns in relational databases is insignificant.
- All values are atomic and each row is unique.

Q #34) What are Nested Triggers?

Answer: Triggers may implement data modification logic by using INSERT, UPDATE, and DELETE statements. These triggers that contain data modification logic and find other triggers for data modification are called Nested Triggers.

Q #35) What is a Cursor?

Answer: A cursor is a database object which is used to manipulate data in a row-to-row manner.

Cursor follows steps as given below:

- Declare Cursor
- Open Cursor
- Retrieve row from the Cursor
- Process the row
- Close Cursor
- Deallocate Cursor

Q #36) What is Collation?

Answer: Collation is a set of rules that check how the data is sorted by comparing it. Such as character data is stored using correct character sequence along with case sensitivity, type, and accent.

Q #37) What do we need to check in Database Testing?

Answer: In Database testing, the following thing is required to be tested:

- Database connectivity
- Constraint check
- Required application field and its size
- Data Retrieval and processing with DML operations
- Stored Procedures
- Functional flow

Q #38) What is Database White Box Testing?

Answer: Database White Box testing involves:

- Database Consistency and ACID properties
- Database triggers and logical views
- Decision Coverage, Condition Coverage, and Statement Coverage
- Database Tables, Data Model, and Database Schema
- Referential integrity rules

Q #39) What is Database Black Box Testing?

Answer: Database Black Box testing involves:

- Data Mapping
- Data stored and retrieved

- Use of Black Box testing techniques such as Equivalence Partitioning and Boundary Value Analysis (BVA)

Q #40) What are Indexes in SQL?

Answer: The index can be defined as the way to retrieve the data more quickly. We can define indexes using CREATE statements.

Syntax:

```
CREATE INDEX index_name  
ON table_name (column_name)
```

Further, we can also create a Unique Index using the following syntax:

```
CREATE UNIQUE INDEX index_name  
ON table_name (column_name)
```

UPDATE: We have added few more short questions for practice.

Q #41) What does SQL stand for?

Answer: SQL stands for [Structured Query Language](#).

Q #42) How to select all records from the table?

Answer: To select all the records from the table we need to use the following syntax:

```
Select * from table_name;
```

Q #43) Define join and name different types of joins?

Answer: Join keyword is used to fetch data from two or more related tables. It returns rows where there is at least one match in both the tables included in the join. [Read more here](#).

Type of joins are:

1. Right join
2. Outer join
3. Full join
4. Cross join
5. Self join.

Q #44) What is the syntax to add a record to a table?

Answer: To add a record in a table INSERT syntax is used.

For Example,

```
INSERT into table_name VALUES (value1, value2..);
```

Q #45) How do you add a column to a table?

Answer: To add another column in the table, use the following command:

```
ALTER TABLE table_name ADD (column_name);
```

Recommended reading ==>> How to [add a column to a table](#) in MySQL

Q #46) Define the SQL DELETE statement.

Answer: DELETE is used to delete a row or rows from a table based on the specified condition.
The basic syntax is as follows:

```
DELETE FROM table_name  
WHERE <Condition>;
```

Q #47) Define COMMIT?

Answer: COMMIT saves all changes made by DML statements.

Q #48) What is the Primary key?

Answer: A Primary key is a column whose values uniquely identify every row in a table. Primary key values can never be reused.

Q #49) What are Foreign keys?

Answer: When a table's primary key field is added to related tables in order to create the common field which relates the two tables, it called a foreign key in other tables. Foreign key constraints enforce referential integrity.

Q #50) What is CHECK Constraint?

Answer: A CHECK constraint is used to limit the values or type of data that can be stored in a column. They are used to enforce domain integrity.

Q #51) Is it possible for a table to have more than one foreign key?

Answer: Yes, a table can have many foreign keys but only one primary key.

Q #52) What are the possible values for the BOOLEAN data field?

Answer: For a BOOLEAN data field, two values are possible: -1(true) and 0(false).

Q #53) What is a stored procedure?

Answer: A stored procedure is a set of SQL queries that can take input and send back output.

Q #54) What is identity in SQL?

Answer: An identity column is where SQL automatically generates numeric values. We can define a start and increment value of the identity column.

Q #55) What is Normalization?

Answer: The process of table design to minimize the data redundancy is called normalization. We need to divide a database into two or more tables and define the relationship between them.

Q #56) What is a Trigger?

Answer: The Trigger allows us to execute a batch of SQL code when a table event occurs (INSERT, UPDATE or DELETE commands are executed against a specific table).

Q #57) How to select random rows from a table?

Answer: Using a SAMPLE clause we can select random rows.

For Example,

```
SELECT * FROM table_name SAMPLE(10);
```

Q #58) Which TCP/IP port does SQL Server run?

Answer: By default SQL Server runs on port 1433.

Q #59) Write a SQL SELECT query that only returns each name only once from a table?

Answer: To get the result as each name only once, we need to use the DISTINCT keyword.

```
SELECT DISTINCT name FROM table_name;
```

Q #60) Explain DML and DDL?

Answer: DML stands for Data Manipulation Language. INSERT, UPDATE and DELETE are DML statements.

DDL stands for Data Definition Language. CREATE, ALTER, DROP, RENAME are DDL statements.

Q #61) Can we rename a column in the output of the SQL query?

Answer: Yes, using the following syntax we can do this.

```
SELECT column_name AS new_name FROM table_name;
```

Q #62) Give the order of SQL SELECT?

Answer: Order of SQL SELECT clauses is: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY. Only the SELECT and FROM clauses are mandatory.

Q #63) Suppose a Student column has two columns, Name and Marks. How to get names and marks of the top three students.

Answer: SELECT Name, Marks FROM Student s1 where 3 <= (SELECT COUNT(*) FROM Students s2 WHERE s1.marks = s2.marks)

Q #64) What is SQL comments?

Answer: SQL comments can be inserted by adding two consecutive hyphens (–).

Q #65) Difference between TRUNCATE, DELETE and DROP commands?

Answer:

- **DELETE** removes some or all rows from a table based on the condition. It can be rolled back.
- **TRUNCATE** removes ALL rows from a table by de-allocating the memory pages. The operation cannot be rolled back
- **DROP** command removes a table from the database completely.

Q #66) What are the properties of a transaction?

Answer: Generally, these properties are referred to as ACID properties. They are:

1. Atomicity
2. Consistency
3. Isolation
4. Durability.

Q #67) What do you mean by ROWID?

Answer: It's an 18 character long pseudo column attached with each row of a table.

Q #68) Define UNION, MINUS, UNION ALL, INTERSECT?

Answer:

- **MINUS** – returns all distinct rows selected by the first query but not by the second.
- **UNION** – returns all distinct rows selected by either query
- **UNION ALL** – returns all rows selected by either query, including all duplicates.
- **INTERSECT** – returns all distinct rows selected by both queries.

Q #69) What is a transaction?

Answer: A transaction is a sequence of code that runs against a database. It takes the database from one consistent state to another.

Q #70) What is the difference between UNIQUE and PRIMARY KEY constraints?

Answer: The differences are as follows:

- A table can have only one PRIMARY KEY whereas there can be any number of UNIQUE keys.
- The primary key cannot contain Null values whereas the Unique key can contain Null values.

Q #71) What is a composite primary key?

Answer: The primary key created on more than one column is called composite primary key.

Q #72) What is an Index?

Answer: An Index is a special structure associated with a table to speed up the performance of queries. The index can be created on one or more columns of a table.

Q #73) What is the Subquery?

Answer: A Subquery is a subset of select statements whose return values are used in filtering conditions of the main query.

Q #74) What do you mean by query optimization?

Answer: Query optimization is a process in which a database system compares different query strategies and select the query with the least cost.

Q #75) What is Collation?

Answer: Set of rules that define how data is stored, how case-sensitivity and Kana character can be treated etc.

Q #76) What is Referential Integrity?

Answer: Set of rules that restrict the values of one or more columns of the tables based on the values of the primary key or unique key of the referenced table.

Q #77) What is the Case function?

Answer: Case facilitates if-then-else type of logic in SQL. It evaluates a list of conditions and returns one of the multiple possible result expressions.

Q #78) Define a temp table?

Answer: A temp table is a temporary storage structure to store the data temporarily.

Q #79) How can we avoid duplicating records in a query?

Answer: By using the DISTINCT keyword, duplication of records in a query can be avoided.

Q #80) Explain the difference between Rename and Alias?

Answer: Rename is a permanent name given to a table or column whereas Alias is a temporary name given to a table or column.

Q #81) What is a View?

Answer: A view is a virtual table that contains data from one or more tables. Views restrict data access of the table by selecting only required values and make complex queries easy.

Q #82) What are the advantages of Views?

Answer: Advantages of Views are:

- Views restrict access to the data because the view can display selective columns from the table.
- Views can be used to make simple queries to retrieve the results of complicated queries. **For Example,** views can be used to query information from multiple tables without the user knowing.

Q #83) List the various privileges that a user can grant to another user?

Answer: SELECT, CONNECT, RESOURCES.

Q #84) What is schema?

Answer: A schema is a collection of database objects of a User.

Q #85) What is a Table?

Answer: A table is the basic unit of data storage in the database management system. Table data is stored in rows and columns.

Q #86) Does View contain Data?

Answer: No, Views are virtual structures.

Q #87) Can a View based on another View?

Answer: Yes, A View is based on another View.

Q #88) What is the difference between the HAVING clause and WHERE clause?

Answer: Both specify a search condition but Having clause is used only with the SELECT statement and typically used with GROUP BY clause.

If GROUP BY clause is not used then Having behaved like WHERE clause only.

Q #89) What is the difference between Local and Global temporary tables?

Answer: If defined inside a compound statement a local temporary table exists only for the duration of that statement but a global temporary table exists permanently in the DB but its rows disappear when the connection is closed.

Q #90) What is CTE?

Answer: A CTE or common table expression is an expression that contains temporary result set which is defined in a SQL statement.

EmployeeInfo Table:

EmpID	EmpFname	EmpLname	Department	Project	Address	DOB	Gender
1	Sanjay	Mehra	HR	P1	Hyderabad(HYD)	01/12/1976	M
2	Ananya	Mishra	Admin	P2	Delhi(DEL)	02/05/1968	F
3	Rohan	Diwan	Account	P3	Mumbai(BOM)	01/01/1980	M
4	Sonia	Kulkarni	HR	P1	Hyderabad(HYD)	02/05/1992	F
5	Ankit	Kapoor	Admin	P2	Delhi(DEL)	03/07/1994	M

EmployeePosition Table:

EmpID	EmpPosition	DateOfJoining	Salary
-------	-------------	---------------	--------

1	Manager	01/05/2022	500000
2	Executive	02/05/2022	75000
3	Manager	01/05/2022	90000
2	Lead	02/05/2022	85000
1	Executive	01/05/2022	300000

Let us start by taking a look at some of the most frequently asked SQL Query interview questions,

- [Write a query to fetch the EmpFname from the EmployeeInfo table in the upper case and use the ALIAS name as EmpName.](#)
- [Write a query to fetch the number of employees working in the department 'HR'.](#)
- [Write a query to get the current date.](#)
- [Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.](#)
- [Write a query to fetch only the place name\(string before brackets\) from the Address column of EmployeeInfo table.](#)
- [Write a query to create a new table that consists of data and structure copied from the other table.](#)
- [Write a query to find all the employees whose salary is between 50000 to 100000.](#)
- [Write a query to find the names of employees that begin with 'S'.](#)
- [Write a query to fetch top N records.](#)
- [Write a query to retrieve the EmpFname and EmpLname in a single column as "FullName". The first name and the last name must be separated with space.](#)

Q1. Write a query to fetch the EmpFname from the EmployeeInfo table in upper case and use the ALIAS name as EmpName.

```
1SELECT UPPER(EmpFname) AS EmpName FROM EmployeeInfo;
```

Q2. Write a query to fetch the number of employees working in the department 'HR'.

```
1SELECT COUNT(*) FROM EmployeeInfo WHERE Department = 'HR';
```

Q3. Write a query to get the current date.

You can write a query as follows in SQL Server:

```
1SELECT GETDATE();
```

You can write a query as follows in [MySQL](#):

Databases Training

```
1SELECT SYSDATE();
```

Q4. Write a query to retrieve the first four characters of EmpLname from the EmployeeInfo table.

```
1SELECT SUBSTRING(EmpLname, 1, 4) FROM EmployeeInfo;
```

Q5. Write a query to fetch only the place name(string before brackets) from the Address column of EmployeeInfo table.

Using the MID function in [MySQL](#)

```
1SELECT MID(Address, 0, LOCATE('(',Address)) FROM EmployeeInfo;
```

Using SUBSTRING

```
1SELECT SUBSTRING(Address, 1, CHARINDEX('(',Address)) FROM EmployeeInfo;
```

Q6. Write a query to create a new table which consists of data and structure copied from the other table.

Using the SELECT INTO command:

```
1SELECT * INTO NewTable FROM EmployeeInfo WHERE 1 = 0;
```

Using the [CREATE command](#) in MySQL:



[MySQL DBA Certification Training](#)

[Weekday / Weekend Batches](#)

```
1CREATE TABLE NewTable AS SELECT * FROM EmployeeInfo;
```

Q7. Write a query to find all the employees whose salary is between 50000 to 100000.

```
1SELECT * FROM EmployeePosition WHERE Salary BETWEEN '50000' AND '100000';
```

Q8. Write a query to find the names of employees that begin with ‘S’

```
1SELECT * FROM EmployeeInfo WHERE EmpFname LIKE 'S%';
```

Q9. Write a query to fetch top N records.

By using the TOP command in SQL Server:

```
1SELECT TOP N * FROM EmployeePosition ORDER BY Salary DESC;
```

By using the LIMIT command in MySQL:

```
1SELECT * FROM EmpPosition ORDER BY Salary DESC LIMIT N;
```

Q10. Write a query to retrieve the EmpFname and EmpLname in a single column as “FullName”. The first name and the last name must be separated with space.

```
1SELECT CONCAT(EmpFname, ' ', EmpLname) AS 'FullName' FROM EmployeeInfo;
```

Q11. Write a query find number of employees whose DOB is between 02/05/1970 to 31/12/1975 and are grouped according to gender

```
1SELECT COUNT(*), Gender FROM EmployeeInfo WHERE DOB BETWEEN '02/05/1970 ' AND  
'31/12/1975' GROUP BY Gender;
```

Q12. Write a query to fetch all the records from the EmployeeInfo table ordered by EmpLname in descending order and Department in the ascending order.

To order the records in ascending and descending order, you have to use the [ORDER BY statement in SQL](#).

```
1SELECT * FROM EmployeeInfo ORDER BY EmpFname desc, Department asc;
```

Q13. Write a query to fetch details of employees whose EmpLname ends with an alphabet ‘A’ and contains five alphabets.

To fetch details matching a certain value, you have to use the [LIKE operator in SQL](#).

```
1SELECT * FROM EmployeeInfo WHERE EmpLname LIKE '____a';
```

Q14. Write a query to fetch details of all employees excluding the employees with first names, “Sanjay” and “Sonia” from the EmployeeInfo table.

```
1SELECT * FROM EmployeeInfo WHERE EmpFname NOT IN ('Sanjay','Sonia');
```

Want to upskill yourself to get ahead in your career? Check out this video

Top 10 Technologies to Learn in 2022 | Edureka



Q15. Write a query to fetch details of employees with the address as “DELHI(DEL)”.

```
1SELECT * FROM EmployeeInfo WHERE Address LIKE 'DELHI(DEL)%';
```

Q16. Write a query to fetch all employees who also hold the managerial position.

```
1SELECT E.EmpFname, E.EmpLname, P.EmpPosition  
2FROM EmployeeInfo E INNER JOIN EmployeePosition P ON  
3E.EmpID = P.EmpID AND P.EmpPosition IN ('Manager');
```

Q17. Write a query to fetch the department-wise count of employees sorted by department's count in ascending order.

```
1SELECT Department, count(EmpID) AS EmpDeptCount  
2FROM EmployeeInfo GROUP BY Department  
3ORDER BY EmpDeptCount ASC;
```

Q18. Write a query to calculate the even and odd records from a table.

To retrieve the even records from a table, you have to use the MOD() function as follows:

```
1SELECT EmpID FROM (SELECT rowno, EmpID from EmployeeInfo) WHERE MOD(rowno,2)=0;
```

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

```
1SELECT EmpID FROM (SELECT rowno, EmpID from EmployeeInfo) WHERE MOD(rowno,2)=1;
```

Q19. Write a SQL query to retrieve employee details from EmployeeInfo table who have a date of joining in the EmployeePosition table.

```
1SELECT * FROM EmployeeInfo E
2WHERE EXISTS
3(SELECT * FROM EmployeePosition P WHERE E.EmpId = P.EmpId);
```

Q20. Write a query to retrieve two minimum and maximum salaries from the EmployeePosition table.

To retrieve two minimum salaries, you can write a query as below:

```
1SELECT DISTINCT Salary FROM EmployeePosition E1
2 WHERE 2 >= (SELECTCOUNT(DISTINCT Salary)FROM EmployeePosition E2
3  WHERE E1.Salary >= E2.Salary) ORDER BY E1.Salary DESC;
```

To retrieve two maximum salaries, you can write a query as below:

```
1SELECT DISTINCT Salary FROM EmployeePosition E1
2 WHERE 2 >= (SELECTCOUNT(DISTINCT Salary) FROM EmployeePosition E2
3  WHERE E1.Salary <= E2.Salary) ORDER BY E1.Salary DESC;
```

Q21. Write a query to find the Nth highest salary from the table without using TOP/limit keyword.

```
1SELECT Salary
2FROM EmployeePosition E1
3WHERE N-1 = (
4    SELECT COUNT( DISTINCT ( E2.Salary ) )
5    FROM EmployeePosition E2
6    WHERE E2.Salary > E1.Salary );
```


Q22. Write a query to retrieve duplicate records from a table.

```
1SELECT EmpID, EmpFname, Department COUNT(*)
2FROM EmployeeInfo GROUP BY EmpID, EmpFname, Department
3HAVING COUNT(*) > 1;
```

Q23. Write a query to retrieve the list of employees working in the same department.

```
1Select DISTINCT E.EmpID, E.EmpFname, E.Department
2FROM EmployeeInfo E, Employee E1
3WHERE E.Department = E1.Department AND E.EmpID != E1.EmpID;
```

Q24. Write a query to retrieve the last 3 records from the EmployeeInfo table.

```
1SELECT * FROM EmployeeInfo WHERE
2EmpID <=3 UNION SELECT * FROM
3(SELECT * FROM EmployeeInfo E ORDER BY E.EmpID DESC)
4AS E1 WHERE E1.EmpID <=3;
```

Q25. Write a query to find the third-highest salary from the EmpPosition table.

```
1SELECT TOP 1 salary
2FROM (
3SELECT TOP 3 salary
4FROM employee_table
5ORDER BY salary DESC) AS emp
6ORDER BY salary ASC;
```

Q26. Write a query to display the first and the last record from the EmployeeInfo table.

To display the first record from the EmployeeInfo table, you can write a query as follows:

```
1SELECT * FROM EmployeeInfo WHERE EmpID = (SELECT MIN(EmpID) FROM EmployeeInfo);
```

To display the last record from the EmployeeInfo table, you can write a query as follows:

```
1SELECT * FROM EmployeeInfo WHERE EmpID = (SELECT MAX(EmpID) FROM EmployeeInfo);
```

Q27. Write a query to add email validation to your database

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

Q28. Write a query to retrieve Departments who have less than 2 employees working in it.

```
1SELECT DEPARTMENT, COUNT(EmpID) as 'EmpNo' FROM EmployeeInfo GROUP BY DEPARTMENT  
1 HAVING COUNT(EmpID) < 2;
```

Q29. Write a query to retrieve EmpPosition along with total salaries paid for each of them.

```
1SELECT EmpPosition, SUM(Salary) from EmployeePosition GROUP BY EmpPosition;
```

Q30. Write a query to fetch 50% records from the EmployeeInfo table.

```
1SELECT *  
  
2FROM EmployeeInfo WHERE  
  
3EmpID <= (SELECT COUNT(EmpID)/2 from EmployeeInfo);
```