

SQL IQ:

1. What is a Primary key?

PK is a column in the Table that uniquely identifies each record and the value is never duplicated in the same table. PK cannot contain NULL Values

OR

A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A primary key's main features are: It must contain a unique value for each row of data. It cannot contain null values.

2. What is a Foreign key?

Is existence of PK in another table

Is a key used to link two tables together

It can accept Null Values

We can have more than one Foreign Key in a table.

OR

In the context of relational databases, a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table or the same table. In simpler words, the foreign key is defined in a second table, but it refers to the primary key or a unique key in the first table.

3. How would you avoid duplicate records in a SELECT query?

DISTINCT identifier eliminates duplicate records.

```
SELECT DISTINCT DEPARTMENT_ID  
FROM EMPLOYEES  
ORDER BY DEPARTMENT_ID;
```

4. When and why do we use ORDER BY Clause?

The ORDER BY keyword is used to sort the result-set in ascending or descending order. The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

```
SELECT EMPLOYEE_ID, FIRST_NAME
```

```
FROM EMPLOYEES  
ORDER BY 1 DESC;
```

5. What is the difference between AND and OR operators?

The AND and OR operators are used to filter records based on more than one condition:
The AND operator displays a record if all the conditions separated by AND is TRUE.
The OR operator displays a record if any of the conditions separated by OR is TRUE.

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90 AND LAST_NAME='King';
```

```
SELECT FIRST_NAME, LAST_NAME, SALARY  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID = 90 OR LAST_NAME='King';
```

6. What is the difference between BETWEEN and IN operators?

Both of these operators are used to find out the multiple values from the table.
Difference between these operators is that the **BETWEEN** operator is used to select a range of data between two values while The **IN** operator allows you to specify multiple values.

```
SELECT * FROM EMPLOYEES  
WHERE SALARY IN (6000, 8000, 10000);
```

```
SELECT * FROM EMPLOYEES  
WHERE SALARY BETWEEN 6000 AND 10000;
```

7. Let's say there are some null values in my table. How would you test those values?

A field with a NULL value is a field with no value. NULL value cannot be compared with other NULL values. Hence, It is not possible to test for NULL values with comparison operators, such as =, <, or >. For this, we have to use the IS NULL operator.

```
SELECT * FROM EMPLOYEES  
WHERE COMMISSION_PCT IS NULL;
```

```
SELECT * FROM EMPLOYEES  
WHERE COMMISSION_PCT IS NOT NULL;
```

8. When and why do we use GROUP BY Clause?

Group by clause is used to group the results of a SELECT query based on one or more columns. The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

```
SELECT COUNT(*), DEPARTMENT_ID  
FROM EMPLOYEES  
WHERE DEPARTMENT_ID IN(10,50,80,100,110,120,150)  
GROUP BY DEPARTMENT_ID  
HAVING COUNT(*)>5;
```

9. Are you familiar with functions in SQL? What type?

LOWER, UPPER, and INITCAP MAX, MIN, AVG

```
SELECT UPPER(COUNTRY_NAME) from COUNTRIES  
WHERE LENGTH(COUNTRY_NAME)>6;
```

```
SELECT * FROM EMPLOYEES  
WHERE HIRE_DATE=(SELECT MIN(HIRE_DATE) FROM EMPLOYEES)  
UNION  
SELECT * FROM EMPLOYEES  
WHERE HIRE_DATE=(SELECT MAX(HIRE_DATE) FROM EMPLOYEES);
```

10. What is the difference between the WHERE and HAVING clauses?

WHERE clause is used for filtering rows and it applies on each and every row, while **HAVING** clause is used to filter groups in SQL.

One syntax level difference between WHERE and HAVING clause is that, WHERE is used **before** GROUP BY clause, while HAVING is used **after** GROUP BY clause.

```
SELECT COUNT(*), DEPARTMENT_ID  
FROM EMPLOYEES
```

```
WHERE DEPARTMENT_ID IN(50,80,100)
GROUP BY DEPARTMENT_ID
HAVING COUNT(*)>5;
```

11. What is a Subquery?

A subquery is a query within another query, also known as a nested query. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

```
SELECT * FROM EMPLOYEES
WHERE SALARY>(SELECT AVG(SALARY) FROM EMPLOYEES);
```

```
SELECT * FROM EMPLOYEES
WHERE DEPARTMENT_ID IN
(SELECT DEPARTMENT_ID FROM DEPARTMENTS
WHERE DEPARTMENT_NAME LIKE 'A%');
```

12. How would you retrieve data that is located in 2 different tables?

Using join of two table base on condition we can retrieve data from two tables.

```
SELECT          COUNT(EMPLOYEE_ID),          E.DEPARTMENT_ID,
DEPARTMENT_NAME
FROM EMPLOYEES E INNER JOIN DEPARTMENTS D
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID
GROUP BY E.DEPARTMENT_ID, DEPARTMENT_NAME;
```

13. What is the difference between an inner and outer join?

An **inner** join returns rows when there is at least some matching data between two (or more) tables that are being compared.

An **outer** join returns rows from both tables that include the records that are unmatched from one or both the tables.

```
SELECT COUNT(EMPLOYEE_ID), E.DEPARTMENT_ID,
DEPARTMENT_NAME
FROM EMPLOYEES E INNER JOIN DEPARTMENTS D
```

```
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
GROUP BY E.DEPARTMENT_ID, DEPARTMENT_NAME;
```

```
SELECT          COUNT(EMPLOYEE_ID),          E.DEPARTMENT_ID,  
DEPARTMENT_NAME  
FROM EMPLOYEES E FULL OUTER JOIN DEPARTMENTS D  
ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
GROUP BY E.DEPARTMENT_ID, DEPARTMENT_NAME;
```

14. What is Self Join?

A self JOIN is a regular join but the table is joined with itself.

```
SELECT  
E.FIRST_NAME AS EMPLOYEE_NAME,  
E.LAST_NAME AS EMPLOYEE_LASTNAME,  
M.FIRST_NAME AS MANAGER_NAME,  
M.LAST_NAME AS MANAGER_LASTNAME, M.MANAGER_ID  
FROM EMPLOYEES E JOIN EMPLOYEES M  
ON E.MANAGER_ID=M.EMPLOYEE_ID;
```

15. How would you find a second highest salary from the table? Third highest?

```
SELECT MAX(SALARY)  
FROM EMPLOYEES  
WHERE SALARY <  
(SELECT MAX(SALARY) FROM EMPLOYEES);
```

```
SELECT MAX(SALARY)  
FROM EMPLOYEES  
WHERE SALARY <  
(SELECT MAX (SALARY) FROM EMPLOYEES  
WHERE SALARY <  
(SELECT MAX(SALARY) FROM EMPLOYEES));
```

16. What is the difference between UNION and UNION ALL?

UNION merges the contents of two structurally-compatible tables into a single combined table.

UNION will omit duplicate records whereas and it will also sort the results where is UNION ALL will include duplicate records.

The performance of UNION ALL will typically be better than UNION, since UNION requires the server to do the additional work of removing any duplicates.

```
SELECT FIRST_NAME, LAST_NAME, EMAIL FROM EMPLOYEES  
UNION ALL  
SELECT FIRST_NAME, LAST_NAME, EMAIL FROM MY_TABLE;
```

```
SELECT FIRST_NAME FROM EMPLOYEES  
UNION  
SELECT LAST_NAME FROM EMPLOYEES;
```

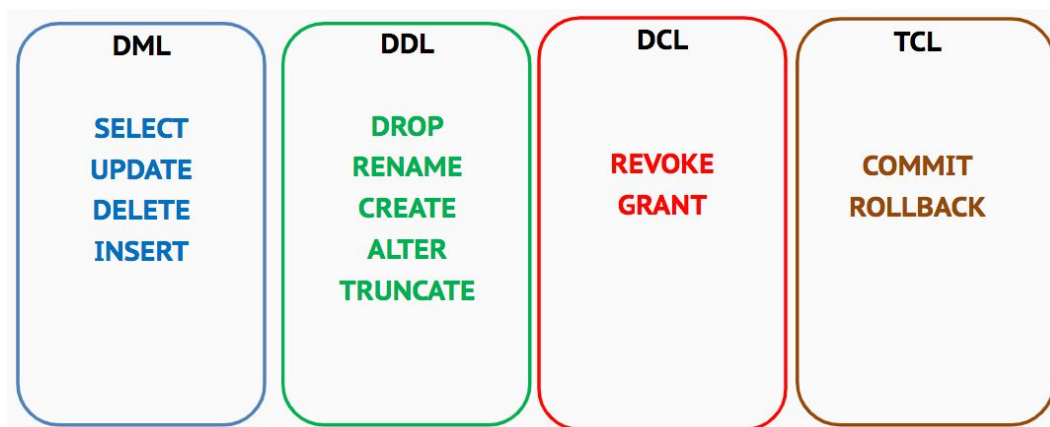
17. What is the difference between DML & DDL Commands? DCL?

Data Definition Language (**DDL**) - defining and dealing with database schemas and descriptions of how the data should exist in database.

Data Manipulation Language (**DML**) - deals with data manipulation (used to retrieve, store, modify, delete and update data in database).

Data Control Language (**DCL**) - deals with rights, permissions and other controls of the database system.

Transaction Control Language (**TCL**) – deals with a transaction within a database.



18. What is the difference between DELETE and TRUNCATE?

DELETE: delete selected rows from the table, WHERE clause can be used for conditional parameters. Deleted records can be rolled back or committed.

TRUNCATE: delete ALL rows from the table >> auto-commits, i.e. cannot be rolled back.

**DELETE FROM INSURANCE
WHERE POLICY_NUMBER=0002;**

TRUNCATE TABLE JOB_HISTORY;

19. What is the difference between Commit and Rollback?

Both ROLLBACK and COMMIT Commands are TCC (Transaction Control Commands), they have differences:

COMMIT is used to save the changes permanently on the server **ROLLBACK** is used to undo the changes and restore previous state.

20. Have you perform Database Testing?

Yes, currently in my project I am working with Relational Database which is Oracle that is stored in AWS.

21. How do you perform Database Testing? How do you connect to Database in Automation?

Manually: I use SQL Developer to execute SQL query against our DataBase

Automation: I use JDBC (Java DataBase Connectivity API) which is Java based library that allows to interact with different DataBases.

First I establish a **Connection** to our database using DatabaseURL, username and password

Once connection is established I will prepare **Statement** Object that will send SQL queries to the Database

And then results from Database will be stored inside **ResultSet** Object.

```
public class JdbcDemo {  
    String dbUrl = "jdbc:oracle:thin:@syntaxdatabase.cdjflmucstpo.us-east-1.rds.amazonaws.com:1521:orcl";  
    String dbUsername = "Syntax";  
    String dbPassword = "syntax123";  
    public void jdbcTest() throws SQLException {  
        Connection connection = DriverManager.getConnection(dbUrl, dbUsername, dbPassword);  
        Statement statement = connection.createStatement();  
        ResultSet rs = statement.executeQuery("Select First_Name, Last_Name from Employees Order by First_Name");  
        while (rs.next()) {  
            System.out.println(rs.getString("First_Name"));  
        }  
    }  
}
```

22. What type of driver you use to connect to Database?

We are using thin driver which is fully java driver. It converts JDBC calls directly into the vendor-specific database protocol.

23. What is metadata?

The metadata means data about data i.e. we can get further information from the data.

JDBC Metadata supports:

- DatabaseMetaData
- ResultSetMetaData

DatabaseMetaData - will give information about Database itself, such as name or version

```
Connection conn = DriverManager.getConnection(dbUrl, dbUname, dbPassword);
```

```
DatabaseMetaData dbMetaData = conn.getMetaData();
```

```
// get database name
```

```
String dbName = dbMetaData.getDatabaseProductName();
```

```
// get database version
```

```
String dbVersion = dbMetaData.getDatabaseProductVersion();
```

ResultSetMetaData - will give an information about our ResultSet


```
Connection conn = DriverManager.getConnection(dbUrl, dbUsername, dbPassword);
Statement st = conn.createStatement();
ResultSet rs = st.executeQuery("SELECT * FROM EMPLOYEES));

ResultSetMetaData rsMetaData = rs.getMetaData();
//get total # or columns
int colNum = rsMetaData.getColumnCount();
//get name of specified column
String colName=rsMetaData.getColumnName(2);
```

24. How do you process results from Database?

Result from Database comes back in ResultSet Object and now we need one of the Java data structures to process results. In my project I am storing data from ResultSet object into List<Map>.

```
String dbUrl =
jdbc:oracle:thin:@syntaxdatabase.cdjflmucstpo.us-east-1.rds.amazonaws.com:1521
:orcl";
String dbUsername = "Syntax";
String dbPassword = "syntax123";

//@Test
public void getResultSetData() throws SQLException {

Connection conn =
DriverManager.getConnection(dbUrl, dbUsername, dbPassword);
Statement st = conn.createStatement();
ResultSet rs =
st.executeQuery("SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEES
WHERE JOB_ID='IT_PROG'");

List<Map<String, String>> rsList = new ArrayList<>();

while (rs.next()) {
    Map<String, String> rsMap = new HashMap<>();
    rsMap.put("FIRST_NAME", rs.getString("FIRST_NAME"));
```

```
        rsMap.put("Last_name", rs.getString("Last_name"));
        rsList.add(rsMap);
    }

    rs.close();
    st.close();
    conn.close();
}
```

To handle any Database related operations I have created a Utility class in which I developed reusable functions to Open Connection, process resultset data and close connections.