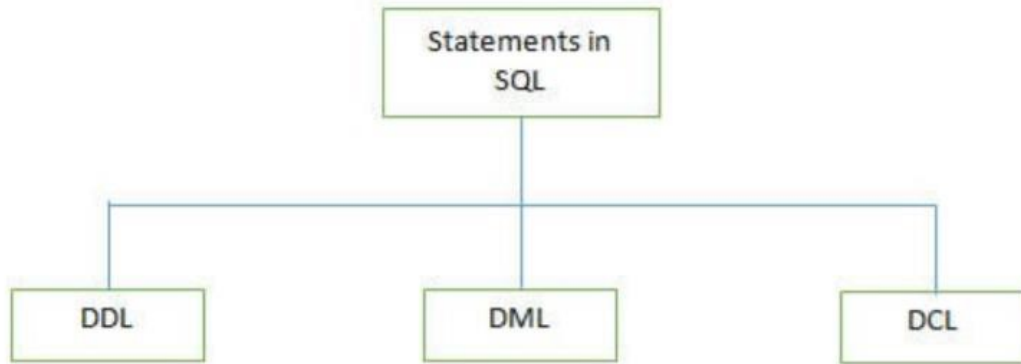


# SQL Interview Questions

## 1. What is SQL?

Structured Query Language is a database tool which is used to create and access database to support software application.

## 2. What are different types of statements supported by SQL?



*There are 3 types of SQL statements*

1) **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.

Some of the DDL Commands are listed below

- **CREATE**: It is used for creating the table.

```
1 CREATE TABLE &table_name  
2 column_name1 data_type(size),  
3 column_name2 data_type(size),  
4 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
```

2) **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 partial DML statement that is used to select all or relevant records in the table.

3) **DCL (Data Control Language)**: These statements are used to set privileges such as Grant and Revoke database access permission to the specific user.

## 3. What is DBMS?

A Database Management System (DBMS) is a program that controls creation, maintenance and use of a database.

DBMS can be termed as File Manager that manages data in a database rather than saving it in file systems.

#### **4. What is RDBMS?**

RDBMS stands for Relational Database Management System. RDBMS store the data into the collection of tables, which is related by common fields between the columns of the table. It also provides relational operators to manipulate the data stored into the tables.

Example: SQL Server.

#### **5. Why do we use SQL constraints? Which constraints we can use while creating database in SQL?**

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 CREATE TABLE statement or even after the table is created once with 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

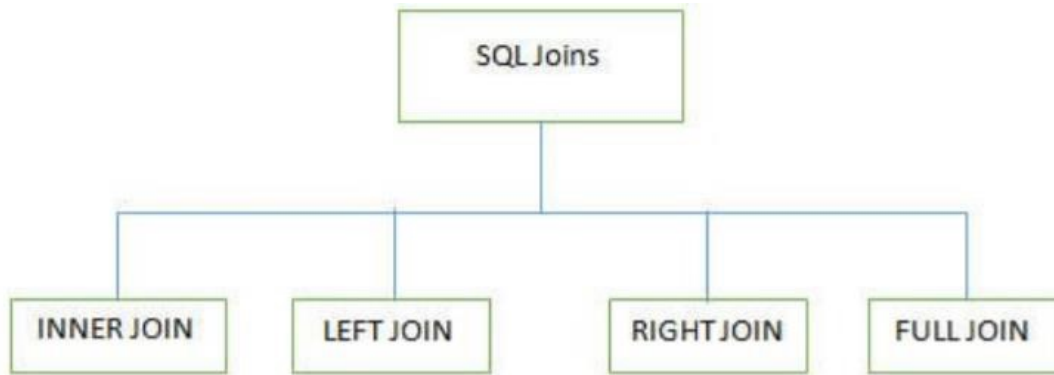
UNIQUE: This constraint is used to ensure that each row and column has unique value and no value is being repeated in any other row or column

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 and also matches the value in one table with another using Primary Key

CHECK: It is used to ensure whether the value in columns fulfills the specified condition

#### **6. What are different JOINS used in SQL?**



There are 4 major types of joins made to use 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 column matched

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

**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 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

```
1 SELECT Employee.Emp_id, Joining.Joining_Date
2 FROM Employee
3 INNER JOIN Joining
4 ON Employee.Emp_id = Joining.Emp_id
5 ORDER BY Employee.Emp_id;
```

There will be 4 records selected. These are the results that you should see

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

*Employee* and *orders* tables where there is a matching *customer\_id* value in both the *Employee* and *orders* tables

- **LEFT JOIN (LEFT OUTER JOIN):** This join returns all rows from a 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;*

### 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

Second Table 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

```
1 SELECT Employee.Emp_id, Joining.Joining_Date
2 FROM Employee
3 LEFT OUTER JOIN Joining
4 ON Employee.Emp_id = Joining.Emp_id
5 ORDER BY Employee.Emp_id;
```

There will be 4 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

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

**Syntax:** *SELECT column\_name(s)*  
*FROM table\_name1*  
*RIGHT JOIN table\_name2*  
*ON column\_name1=column\_name2;*

### 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 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

```
1 SELECT Employee.Emp_id, Joining.Joining_Date
2 FROM Employee
3 LEFT OUTER JOIN Joining
4 ON Employee.Emp_id = Joining.Emp_id
5 ORDER BY Employee.Emp_id;
```

There will be 4 records selected. These are the results that you should see

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

- **FULL JOIN (FULL OUTER JOIN):** This joins returns all 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;`

#### 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

Second Table 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:

```
1 SELECT Employee.Emp_id, Joining.Joining_Date
2 FROM Employee
3 FULL OUTER JOIN Joining
4 ON Employee.Emp_id = Joining.Emp_id
5 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

## **7. What is normalization?**

Normalization is the process of minimizing redundancy and dependency by organizing fields and table of a database. The main aim of Normalization is to add, delete or modify field that can be made in a single table.

## **8. What are all the different normalizations?**

The normal forms can be divided into 4 forms, and they are explained below -.

1. First Normal Form (1NF): This should remove all the duplicate columns from the table. Creation of tables for the related data and identification of unique columns.
2. Second Normal Form (2NF): Meeting all requirements of the first normal form. Placing the subsets of data in separate tables and Creation of relationships between the tables using primary keys.
3. Third Normal Form (3NF): This should meet all requirements of 2NF. Removing the columns which are not dependent on primary key constraints.
4. Fourth Normal Form (4NF): Meeting all the requirements of third normal form and it should not have multivalued dependencies.

## **9. How many Aggregate Functions are available there in SQL?**

SQL Aggregate Functions calculates values from multiple columns in a table and returns a single value.

There are 7 aggregate functions we use in SQL

AVG(): Returns the average value from specified columns

COUNT(): Returns number of table rows

MAX(): Returns 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

### **10. What is an Index? What are all the different types of indexes?**

An index is performance tuning method of allowing faster retrieval of records from the table. An index creates an entry for each value and it will be faster to retrieve data. This indexing does not allow the field to have duplicate values if the column is unique indexed. Unique index can be applied automatically when primary key is defined.

1. Clustered Index: This type of index reorders the physical order of the table and search based on the key values. Each table can have only one clustered index.
2. Non-Clustered Index: Non-Clustered Index does not alter the physical order of the table and maintains logical order of data. Each table can have 999 non clustered indexes.

### **11. What is SQL Injection?**

SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database such that once it is executed the database is opened for an attacker. This technique is usually used for attacking Data-Driven Applications to have an access to sensitive data and perform administrative tasks on databases.

For Example: `SELECT column_name(s) FROM table_name WHERE condition;`

### **12. What is the difference between "Primary Key" and "Unique Key"?**

1. We can have only one Primary Key in a table whereas we can have more than one Unique Key in a table.
2. The Primary Key cannot have a NULL value whereas a Unique Key may have only one null value.
3. By default, a Primary Key is a Clustered Index whereas by default, a Unique Key is a unique non-clustered index.
4. A Primary Key supports an Auto Increment value whereas a Unique Key doesn't support an Auto Increment value.