# SQL Interview Questions

## Q1. What is a database?

A database is an organized collection of data, stored and retrieved digitally from a remote or local computer system. Databases can be vast and complex, and such databases are developed using fixed design and modelling approaches.

## Q2. What is SQL?

SQL stands for Structured Query Language. It is the standard language for relational database management systems. It is especially useful in handling organized data comprised of entities (variables) and relations between different entities of the data.

## Q3. What is the difference between SQL and MySQL?

SQL is a standard language for retrieving and manipulating structured databases. On the contrary, MySQL is a relational database management system, like SQL Server, Oracle or IBM DB2, that is used to manage SQL databases.

## Q4. What is a Query?

A query is a request for data or information from a database table or combination of tables. A database query can be either a select query or an action query.
Example-

```
SELECT fname, lname            /* select query */
FROM myDb.students
WHERE student_id = 1;
```

## Q5. What is a Subquery? What are its types?

A subquery is a query within another query, also known as nested query or inner query. It is used to restrict or enhance the data to be queried by the main query, thus restricting or enhancing the output of the main query, respectively.

For example, here we fetch the contact information for students who have enrolled for the maths subject:

```
SELECT name, email, mob, address
FROM myDb.contacts
WHERE roll_no IN (
        SELECT roll_no
        FROM myDb.students
        WHERE subject = 'Maths');
```

There are two types of subquery - Correlated and Non-Correlated.

- A **correlated** subquery cannot be considered as an independent query, but it can refer the column in a table listed in the FROM of the main query.
- A **non-correlated** subquery can be considered as an independent query and the output of subquery is substituted in the main query.

## Q6. What is the SELECT statement?

SELECT operator in SQL is used to select data from a database. The data returned is stored in a result table, called the result-set. Example-

```
SELECT * FROM myDB.students;
```

## Q7. What are some common clauses used with SELECT query in SQL?

Some common SQL clauses used in conjuction with a SELECT query are as follows:

- **WHERE** clause in SQL is used to filter records that are necessary, based on specific conditions.
- **ORDER BY** clause in SQL is used to sort the records based on some field(s) in ascending (ASC) or descending order (DESC).

**Example –**

```
SELECT *
FROM myDB.students
WHERE graduation_year = 2019
ORDER BY studentID DESC;
```

- **GROUP BY** clause in SQL is used to group records with identical data and can be used in conjuction with some aggregation functions to produce summarized results from the database.
- **HAVING** clause in SQL is used to filter records in combination with the GROUP BY clause. It is different from WHERE, since WHERE clause cannot filter aggregated records.

**Example –**

```
SELECT COUNT(studentId), country

FROM myDB.students

WHERE country != "INDIA"

GROUP BY country

HAVING COUNT(studentID) > 5;
```

## Q8. What are the TRUNCATE, DELETE and DROP statements?

**DELETE** statement is used to delete rows from a table.

```
DELETE FROM Candidates

WHERE CandidateId > 1000;
```

**TRUNCATE** command is used to delete all the rows from the table and free the space containing the table.

```
TRUNCATE TABLE Candidates;
```

**DROP** command is used to remove an object from the database. If you drop a table, all the rows in the table is deleted and the table structure is removed from the database.

```
DROP TABLE Candidates;
```

## Q9. What are Aggregate functions?

An aggregate function performs operations on a collection of values to return a single scalar value. Aggregate functions are often used with the GROUP BY and HAVING clauses of the SELECT statement. Following are the widely used SQL aggregate functions:

**AVG() -** Calculates the mean of a collection of values.

**COUNT() -** Counts the total number of records in a specific table or view.

**MIN() -** Calculates the minimum of a collection of values.

**MAX() -** Calculates the maximum of a collection of values.

**SUM() -** Calculates the sum of a collection of values.

**FIRST() -** Fetches the first element in a collection of values.

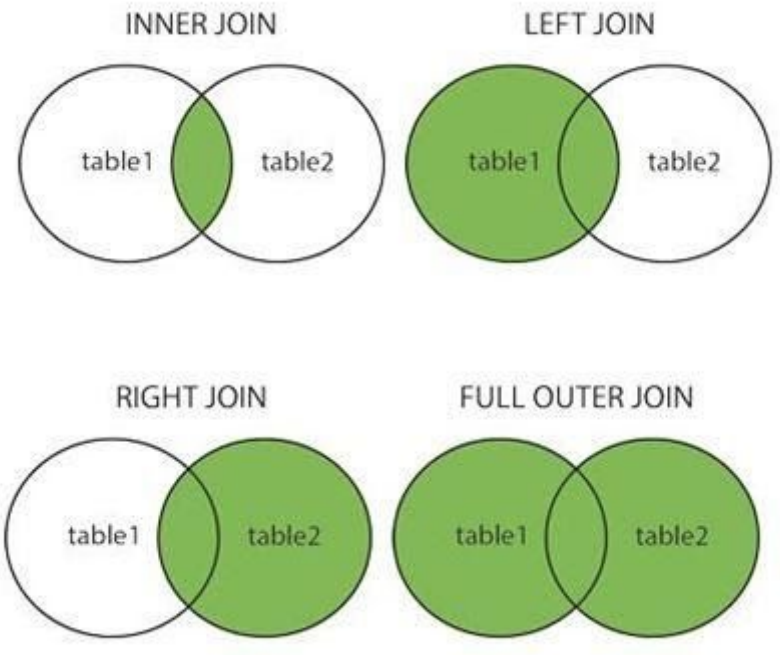
**LAST() -** Fetches the last element in a collection of values.

Note: All aggregate functions described above ignore NULL values except for the COUNT function.

## Q10. What is a Join? List its different types.

The SQL Join clause is used to combine records (rows) from two or more tables in a SQL database based on a related column between the two.

There are four different types of JOINs in SQL:




**(INNER) JOIN:** Retrieves records that have matching values in both tables involved in the join. This is the widely used join for queries.

```
SELECT *

FROM Table_A
```

```
JOIN Table_B;


SELECT *
FROM Table_A
INNER JOIN Table_B;
```

**LEFT (OUTER) JOIN:** Retrieves all the records/rows from the left and the matched records/rows from the right table.

```
SELECT *
FROM Table_A A
LEFT JOIN Table_B B
ON A.col = B.col;
```

**RIGHT (OUTER) JOIN:** Retrieves all the records/rows from the right and the matched records/rows from the left table.

```
SELECT *
FROM Table_A A
RIGHT JOIN Table_B B
ON A.col = B.col;
```

**FULL (OUTER) JOIN:** Retrieves all the records where there is a match in either the left or right table.

```
SELECT *
FROM Table_A A
FULL JOIN Table_B B
ON A.col = B.col;
```

## Q11. What is a Self-Join?

A self-JOIN is a case of regular join where a table is joined to itself based on some relation between its own column(s). Self-join uses the INNER JOIN or LEFT JOIN clause, and a table alias is used to assign different names to the table within the query.


SELECT A.emp_id AS "Emp_ID",A.emp_name AS "Employee",

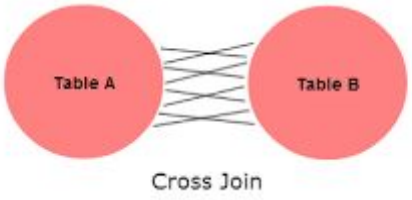
B.emp_id AS "Sup_ID",B.emp_name AS "Supervisor"

FROM employee A, employee B

WHERE A.emp_sup = B.emp_id;

## Q12. What is a Cross-Join?

Cross join can be defined as a cartesian product of the two tables included in the join. The table after join contains the same number of rows as in the cross-product of number of rows in the two tables. If a WHERE clause is used in cross join, then the query will work like an INNER JOIN.




SELECT stu.name, sub.subject

FROM students AS stu

CROSS JOIN subjects AS sub;

## Q13. What is the difference between BETWEEN and IN condition operators?

The BETWEEN operator is used to display rows based on a range of values. The values can be numbers, text, and dates as well. BETWEEN operator gives us the count of all the values occurs between a particular range.

The IN-condition operator is used to check for values contained in a specific set of values. IN operator is used when we have more than one value to choose.

## Q14. What is the difference between NULL value, zero and blank space?

A NULL value is not the same as zero or a blank space. A NULL value is a value which is 'unavailable, unassigned, unknown or not applicable.'

On the other hand, zero is a number, and a blank space is treated as a character.

The NULL value can be treated as unknown and missing value as well, but zero and blank spaces are different from the NULL value.

## Q15. What is the difference between ‘HAVING’ CLAUSE and a ‘WHERE’ CLAUSE?

HAVING clause can be used only with SELECT statement. It is usually used in a GROUP BY clause and whenever GROUP BY is not used, HAVING behaves like a WHERE clause.

Having Clause is only used with the GROUP BY function in a query whereas WHERE Clause is applied to each row before they are a part of the GROUP BY function in a query.