**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”*

### ****What are the different types of SQL commands?****

SQL commands are segregated into the following types:

* DDL – Data Definition Language
* DML – Data Manipulation Language
* DQL – Data Query Language
* DCL – Data Control Language
* TCL – Transaction Control Language

**What are the different DDL commands in SQL?**

DDL commands are used to define or alter the structure of the database.

* CREATE: To create databases and database objects
* ALTER: To alter existing database objects
* DROP: To drop databases and databases objects
* TRUNCATE: To remove all records from a table but not its database structure
* RENAME: To rename database objects

**What are the different DML commands in SQL?**

DML commands are used for managing data present in the database.

* SELECT: To select specific data from a database
* INSERT: To insert new records into a table
* UPDATE: To update existing records
* DELETE: To delete existing records from a table

**What are the different DCL commands in SQL?**

DCL commands are used to create roles, grant permission, and control access to the database objects.

* GRANT: To provide user access
* DENY: To deny permissions to users
* REVOKE: To remove user access

**What are the different TCL commands in SQL?**

TCL commands are used to manage the changes made by DML statements.

* COMMIT: To write and store the changes to the database
* ROLLBACK: To restore the database since the last commit

**What is a View in SQL?**

A view is like a subset of a table which is stored logically in a database. A view is a virtual table. It contains rows and columns similar to a real table. The fields in the view are fields from one or more real tables. Views do not contain data of their own. They are used to restrict access to the database or to hide data complexity.



|  |  |
| --- | --- |
| 1 | CREATE VIEW view\_name AS SELECT column\_name1, column\_name2 FROM table\_name WHERE CONDITION; |

**What are the advantages of Views?**

Some of the advantages of Views are

1. Views occupy no space
2. Views are used to simply retrieve the results of complicated queries that need to be executed often.
3. Views are used to restrict access to the database or to hide data complexity.

### ****How to avoid duplicate records in a query?****

The SQL SELECT DISTINCT query is used to return only unique values. It eliminates all the duplicated values.

**What is the difference between Rename and Alias?**

‘Rename’ is a permanent name given to a table or column  
‘Alias’ is a temporary name given to a table or column.

**What is a Join?**

Join is a query, which retrieves related columns or rows from multiple tables.

**What are the different types of joins?**

Types of Joins are as follows:

* INNER JOIN
* LEFT JOIN
* RIGHT JOIN
* OUTER JOIN

### ****What are SQL constraints?****

SQL constraints are the set of rules that enforced some restriction while inserting, deleting or updating of data in the databases.

### ****What are the constraints available in SQL?****

Some of the constraints in SQL are – Primary Key, Foreign Key, Unique Key, SQL Not Null, Default, Check and Index constraint.

### ****What is a Unique constraint?****

A unique constraint is used to ensure that there are no duplication values in the field/column.

### ****What is a Primary Key?****

A PRIMARY KEY constraint uniquely identifies each record in a database table. All columns participating in a primary key constraint must not contain NULL values.

### ****Can a table contain multiple****PRIMARY *****KEY’s*?****

The short answer is no, a table is not allowed to contain multiple primary keys but it allows to have one composite primary key consisting of two or more columns.

### ****What is a Composite****PRIMARY KEY****?****

Composite PRIMARY KEY is a primary key created on more than one column (combination of multiple fields) in a table.

### ****What is a****FOREIGN KEY****?****

A FOREIGN KEY is a key used to link two tables together. A FOREIGN KEY in a table is linked with the PRIMARY KEY of another table.

### ****Can a table contain multiple****FOREIGN KEY’s****?****

A table can have many FOREIGN KEY’s.

### ****What is the difference between****UNIQUE****and****PRIMARY KEY****constraints?****

There should be only one PRIMARY KEY in a table whereas there can be any number of UNIQUE Keys.  
PRIMARY KEY doesn’t allow NULL values whereas Unique key allows NULL values.

### ****What is a****NULL****value?****

A field with a NULL value is a field with no value. A NULL value is different from a zero value or a field that contains spaces. A field with a NULL value is one that has been left blank during record creation. Assume, there is a field in a table is optional and it is possible to insert a record without adding a value to the optional field then the field will be saved with a NULL value.

### ****What is the difference between NULL value, Zero, and Blank space?****

As I mentioned earlier, Null value is field with no value which is different from zero value and blank space.  
Null value is a field with no value.  
Zero is a number  
Blank space is the value we provide. The ASCII value of space is CHAR(32).

### ****How to Test for****NULL****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 and IS NOT NULL operators.



|  |  |
| --- | --- |
| 1 | SELECT column\_names FROM table\_name WHERE column\_name IS NULL; |



|  |  |
| --- | --- |
| 1 | SELECT column\_names FROM table\_name WHERE column\_name IS NOT NULL; |

### ****What is SQL****NOT NULL****constraint?****

NOT NULL constraint is used to ensure that the value in the filed cannot be a NULL

### ****What is a****CHECK ****constraint?****

A CHECK constraint is used to limit the value that is accepted by one or more columns.

E.g. ‘Age’ field should contain only the value greater than 18.



|  |  |
| --- | --- |
| 1 | CREATE TABLE EMP\_DETAILS(EmpID int NOT NULL, NAME VARCHAR (30) NOT NULL, Age INT CHECK (AGE &gt; 18), PRIMARY KEY (EmpID)); |

### ****What is a****DEFAULT ****constraint?****

DEFAULT constraint is used to include a default value in a column when no value is supplied at the time of inserting a record.

### ****What is the difference between Delete and Truncate?****

The difference between the Delete, and Truncate are

| **DELETE** | **TRUNCATE** |
| --- | --- |
| Delete statement is used to delete rows from a table. It can be rolled back. | Truncate statement is used to delete all the rows from the table and free the space containing the table. It cant be rolled back. |
| We can use WHERE condition in DELETE statement and can delete required rows | We cant use WHERE condition in TRUNCATE statement. So we cant delete required rows alone |
| We can delete specific rows using DELETE | We can only delete all the rows at a time using TRUNCATE |
| Delete is a DML command | Truncate is a DDL command |
| Delete maintains log and performance is slower than Truncate | Truncate maintains minimal log and performance wise faster |
| We need DELETE permission on Table to use DELETE command | We need at least ALTER permission on the table to use TRUNCATE command |

### ****What is the difference between Union and Union All command?****

This is one of the tricky SQL Interview Questions. Interviewer may ask you this question in another way as what are the advantages of Union All over Union.

Both Union and Union All concatenate the result of two tables but the way these two queries handle duplicates are different.

**Union:**It omits duplicate records and returns only distinct result set of two or more select statements.  
**Union All:** It returns all the rows including duplicates in the result set of different select statements.

Performance wise Union All is faster than Union, Since Union All doesn’t remove duplicates. Union query checks the duplicate values which consumes some time to remove the duplicate records.

Assume: Table1 has 10 records, Table2 has 10 records. Last record from both the tables are same.

If you run Union query.



|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM Table1  UNION  SELECT \* FROM Table2 |

**Output:** Total 19 records

If you run Union query.



|  |  |
| --- | --- |
| 1  2  3 | SELECT \* FROM Table1  UNION ALL  SELECT \* FROM Table2 |

**Output:** Total 20 records

Data type of all the columns in the two tables should be same.

**What are aggregate functions in SQL?**

SQL aggregate functions return a single value, calculated from values in a column. Some of the aggregate functions in SQL are as follows

* AVG() – This function returns the average value
* COUNT() – This function returns the number of rows
* MAX() – This function returns the largest value
* MIN() – This function returns the smallest value
* ROUND() – This function rounds a numeric field to the number of decimals specified
* SUM() – This function returns the sum

### ****How to add new Employee details in an Employee\_Details table with the following details**** ****Employee\_Name: John, Salary: 5500, Age: 29?****

### INSERT into Employee\_Details (Employee\_Name, Salary, Age) VALUES (‘John’, 5500 , 29);

### ****How to add a column ‘Salary’ to a table Employee\_Details?****

### ALTER TABLE Employee\_Details ADD (Salary);

### ****How to change a value of the field ‘Salary’ as 7500 for an Employee\_Name ‘John’ in a table Employee\_Details?****

UPDATE Employee\_Details set Salary = 7500 where Employee\_Name = ‘John’;

### ****Write an SQL Query to select all records from the table?****

Select \* from table\_name;

### ****Define SQL Delete statement.****

The SQL Delete statement is used to delete records from a table.

DELETE FROM table\_name WHERE some\_column=some\_value;

### ****Write the command to remove all Players named Sachin from the Players table.****

DELETE from Players WHERE Player\_Name = ‘Sachin’;

### ****How to get each name only once from an employee table?****

By using the DISTINCT keyword, we could get each name only once.

SELECT DISTINCT employee\_name FROM employee\_table;

### ****How to rename a column in the output of SQL query?****

By using SQL AS keyword

SELECT column\_name AS new\_name FROM table\_name;

### ****What is the order of SQL SELECT?****

Order of SQL SELECT statement is as follows

SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY.

### ****How to display the current date in SQL?****

In SQL, there is a built-in function called GetDate() which helps to return the current date.

SELECT GetDate();

### ****Write an SQL Query to find an Employee\_Name whose Salary is equal or greater than 5000 from the below table Employee\_Details.****

### SELECT Employee\_Name FROM Employee\_Details WHERE Salary>=5000;

### ****Write an SQL Query to find list of Employee\_Name start with ‘E’ from the below table****

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | SELECT \* FROM Employee\_Details WHERE Employee\_Name like 'E%'; ****Write SQL SELECT query that returns the FirstName and LastName from Employee\_Details table.****  |  | | --- | | SELECT FirstName, LastName FROM Employee\_Details; |  ****How to select all the even number records from a table?**** To select all the even number records from a table: Select \* from table where id % 2 = 0; **What are different JOINS used in SQL?**  **Answer:**  [SQL Joins](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/SQL-Joins.jpg)  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&nbsp;   INNER JOIN table\_name2   ON column\_name1=column\_name2; |   **For Example,**  In this example, we have a table **Employee** with the following data:  [Employee table](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)  The second table’s name is**Joining.**  [joining](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining.jpg)  **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:**  [result of innerjoin](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/result-of-innerjoin.jpg)  **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:  [Employee table](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)  The second table’s name is**Joining.**  [joining 1](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining-1.jpg)  **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:**  [result of LEFT OUTER JOIN](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/result-of-LEFT-OUTER-JOIN.jpg)  **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:  [Employee table](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)  The second table’s name is**Joining.**  [joining table](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining-1.jpg)  **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:  [Employee table](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)  The second table’s name is**Joining.**  [joining 1](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining-1.jpg)  **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.**  [result of FULL OUTER JOIN](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/result-of-FULL-OUTER-JOIN.jpg)  **Write an SQL query to fetch the EmpId and FullName of all the employees working under Manager with id – ‘986’.** Ans. We can use the EmployeeDetails table to fetch the employee details with a where clause for the manager-  SELECT EmpId, FullName  FROM EmployeeDetails  WHERE ManagerId = 986; ****Write an SQL query to fetch the different projects available from the EmployeeSalary table.****  SELECT DISTINCT(Project)  FROM EmployeeSalary;  **Write an SQL query to fetch the count of employees working in project ‘P1’.** Here, we would be using aggregate function count() with the SQL **where** clause-  SELECT COUNT(\*)  FROM EmployeeSalary  WHERE Project = 'P1';  **Write an SQL query to find the maximum, minimum, and average salary of the employees.** Ans. We can use the aggregate function of SQL to fetch the max, min, and average values-  SELECT Max(Salary),  Min(Salary),  AVG(Salary)  FROM EmployeeSalary;  **Write an SQL query to find the employee id whose salary lies in the range of 9000 and 15000.** Ans. Here, we can use the ‘Between’ operator with a where clause.  SELECT EmpId, Salary  FROM EmployeeSalary  WHERE Salary BETWEEN 9000 AND 15000;  **Write an SQL query to** f**etch all the employees who either live in California or work under a manager with ManagerId – 321.** Ans. This interview question requires us to satisfy either of the conditions – employees living in ‘California’ and working under Manager with ManagerId ‘321’. So, we will use the OR operator here-  SELECT EmpId, City, ManagerId  FROM EmployeeDetails  WHERE City='California' OR ManagerId='321';  **Write an SQL query to fetch all those employees who work on Project other than P1.** Ans. Here, we can use the NOT operator to fetch the rows which are not satisfying the given condition.  SELECT EmpId  FROM EmployeeSalary  WHERE NOT Project='P1';  Or using the not equal to operator-  SELECT EmpId  FROM EmployeeSalary  WHERE Project <> 'P1';  **Write an SQL query to fetch the EmpIds that are present in both the tables –   ‘EmployeeDetails’ and ‘EmployeeSalary.** Ans. Using sub query-  SELECT EmpId FROM  EmployeeDetails  where EmpId IN  (SELECT EmpId FROM EmployeeSalary);  **Write an SQL query to fetch the EmpIds that are present in EmployeeDetails but not in EmployeeSalary.** Ans. Using sub query-  SELECT EmpId FROM  EmployeeDetails  where EmpId Not IN  (SELECT EmpId FROM EmployeeSalary); |