1.Scalar functions are functions that take one or more input values and return a single value. Scalar functions can be used in SELECT statements, WHERE clauses, and other parts of SQL queries to manipulate data or perform calculations on individual values.

Example of a scalar function in SQL that returns the length of a string:

**SELECT LEN('Hello, world!') AS StringLength;**

2.Joins are used to combine data from two or more tables based on a common column or set of columns. There are several types of joins in SQL, including inner join, left join, right join, and full outer join.

Example of an inner join in SQL:

**SELECT emp.Ename, dept.Dname, dept.loc**

**FROM EMP**

**INNER JOIN orders**

**ON EMP.DEPTNO= DEPT.DEPTNO**

3.To rename a column , we can use the ALTER TABLE statement along with the RENAME COLUMN clause.

SYNTAX:

**ALTER TABLE table\_name RENAME COLUMN old\_column\_name TO new\_column\_name;**

4.To find duplicate records ,we can use the GROUP BY clause along with the HAVING clause to filter the results.

SYNTAX:

**SELECT column1, column2, COUNT(\*) as count**

**FROM table\_name**

**GROUP BY column1, column2**

**HAVING COUNT(\*) > 1;**

5.The DISTINCT keyword is used to eliminate duplicate rows from the result set of a query. When you use the DISTINCT keyword in a SELECT statement, it instructs the database to return only unique values for the specified columns, removing any duplicates that may be present in the data.

SYNTAX:

**SELECT DISTINCT column1, column2**

**FROM table\_name;**

6.To remove duplicate rows from a table, we can use the DISTINCT keyword along with the INSERT INTO statement to create a new table with the distinct rows, or we can use the DELETE statement to remove the duplicate rows from the original table.

SYNTAX:

**CREATE TABLE employees\_distinct AS**

**SELECT DISTINCT \* FROM employees;**

7.**SELECT MAX(salary) AS max\_salary**

**FROM employees**

**WHERE department = 'Sales';**

8.**Comparison Operators:** Comparison operators are used to compare values and return a Boolean value (TRUE or FALSE) based on the comparison result. The comparison operators in SQL are:

= (equal to)

<> or != (not equal to)

< (less than)

> (greater than)

<= (less than or equal to)

>= (greater than or equal to)

Example: **SELECT \* FROM employees WHERE salary > 50000;**

**Logical Operators:** Logical operators are used to combine two or more Boolean expressions to create a more complex expression. The logical operators in SQL are:

AND (returns TRUE if all expressions are TRUE)

OR (returns TRUE if any of the expressions is TRUE)

NOT (returns the opposite Boolean value of the expression)

Example: **SELECT \* FROM employees WHERE department = 'Sales' AND salary > 50000;**

**Arithmetic Operators:** Arithmetic operators are used to perform mathematical operations on numeric values. The arithmetic operators in SQL are:

+ (addition)

- (subtraction)

\* (multiplication)

/ (division)

% (modulo)

Example: **SELECT salary \* 1.05 AS increased\_salary FROM employees;**

9.To display the first 5 records from an Employee table in SQL, we can use the SELECT statement along with the LIMIT clause to limit the number of rows returned by the query.

Example:

**SELECT \* FROM Employee LIMIT 5;**

10.To display the last 5 records from an Employee table in SQL, you can use the SELECT statement along with the ORDER BY and LIMIT clauses.

Example:

**SELECT \* FROM Employee ORDER BY EmployeeID DESC LIMIT 5;**

11.To fetch the 3rd highest salary using the RANK() function in SQL,

**SELECT salary**

**FROM (**

**SELECT salary, RANK() OVER (ORDER BY salary DESC) AS ranking**

**FROM employee) AS ranking**

**WHERE rank = 3;**

12.To create a table with the same structure and data as the Employee table , we can use the CREATE TABLE statement with the AS clause.

Example:

**CREATE TABLE EmployeeCopy AS**

**SELECT \* FROM Employee;**

13.

**SELECT Employee.\***

**FROM Employee**

**LEFT JOIN Department**

**ON Employee.DepartmentID = Department.DepartmentID**

**WHERE Department.DepartmentID IS NULL;**