SQL

1. Scalar functions are functions in SQL that take one or more input parameters and return a single scalar value. They can be used in SQL queries to perform calculations or manipulate data.

CREATE FUNCTION square (@num FLOAT)

RETURNS FLOAT

AS

BEGIN

RETURN @num \* @num

END

1. A join is used to combine rows from two or more tables based on a related column between them. The most used types of joins are INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

Example:

INNER JOIN to combine the "customers" and "orders" tables based on the "customer\_id" column.

SELECT \*

FROM customers

INNER JOIN orders

ON customers.customer\_id = orders.customer\_id;

1. The ALTER TABLE statement with the RENAME COLUMN clause.

ALTER TABLE table\_name

RENAME COLUMN old\_column\_name TO new\_column\_name;

1. To find duplicate records in SQL the GROUP BY clause is used with the HAVING clause.

SELECT email, COUNT(\*)

FROM customers

GROUP BY email

HAVING COUNT(\*) > 1;

1. The DISTINCT keyword is used to retrieve unique values from a table. It is used in the SELECT statement to eliminate duplicate rows.

SELECT DISTINCT column\_name

FROM table\_name;

1. Remove duplicate from table:

DELETE FROM table\_name

WHERE column\_name NOT IN (

SELECT MIN(column\_name)

FROM table\_name

GROUP BY column\_name

);

1. Max salary

SELECT MAX(salary)

FROM employee

WHERE department = 'department\_name';

1. Operators in sql
2. Comparison operators: Used to compare two values or expressions. The comparison operators include:

* = (Equal to)
* <> or != (Not equal to)
* < (Less than)
* <= (Less than or equal to)
* (Greater than)
* = (Greater than or equal to)

1. Logical operators: Used to combine multiple conditions in a query. The logical operators include:

* AND (Returns true if all conditions are true)
* OR (Returns true if any of the conditions are true)
* NOT (Negates a condition)

1. Arithmetic operators: Used to perform mathematical calculations on data. The arithmetic operators include:

* (Addition)
* (Subtraction)
* (Multiplication)
* / (Division)
* % (Modulo)

Example:

SELECT \*

FROM employees

WHERE department = 'sales' OR salary > 50000;

SELECT \*

FROM Employee

LIMIT 5;

SELECT \*

FROM Employee

ORDER BY Employee\_ID DESC

LIMIT 5;



SELECT salary

FROM (

SELECT salary, RANK() OVER (ORDER BY salary DESC) AS salary\_rank

FROM employee

) AS ranked\_salaries

WHERE salary\_rank = 3;



CREATE TABLE new\_employee

AS SELECT \*

FROM employee;



SELECT e.\*

FROM Employee e

LEFT JOIN Department d ON e.Department\_ID = d.Department\_ID

WHERE d.Department\_ID IS NULL;