Name : Deepankar Sharma

Course: BCA

University Roll No: 2092014

Student Id : 20041299

Semester: 4

Date: April 2, 2022

**Practical 04: Operators in SQL**

**Objective:**  Implementation of different types of operators in SQL

**1-ARITHMETIC OPERATORS-**They perform mathematical operations.

**ADDITION**

**SYNTAX-** SELECT operand1+operand 2;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select salary+200 as new\_salary from workers;

+------------+

| new\_salary |

+------------+

| 500 |

| 400 |

| 1400 |

| 212 |

| 1542 |

| 1442 |

+------------+

6 rows in set (0.01 sec)

**SUBTRACTION**

**SYNTAX-** SELECT operand1- operand 2;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select salary-100 as new\_salary from workers;

+------------+

| new\_salary |

+------------+

| 200 |

| 100 |

| 1100 |

| -88 |

| 1242 |

| 1142 |

+------------+

6 rows in set (0.00 sec)

**MULTIPLICATION:**

**SYNTAX:**SELECT operand1\* operand 2;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select id\*salary as new\_salary from workers;

+------------+

| new\_salary |

+------------+

| 3600 |

| 2600 |

| 16800 |

| 180 |

| 45628 |

| 290628 |

+------------+

6 rows in set (0.00 sec)

**DIVISION:**

**SYNTAX:**SELECT operand1 / operand 2;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select salary/id as new\_salary from workers;

+------------+

| new\_salary |

+------------+

| 25.0000 |

| 15.3846 |

| 85.7143 |

| 0.8000 |

| 39.4706 |

| 5.3077 |

+------------+

6 rows in set (0.00 sec)

**MODULUS:**

**SYNTAX:** SELECT operand1 % operand 2;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select salary%id as new\_salary from workers;

+------------+

| new\_salary |

+------------+

| 0 |

| 5 |

| 10 |

| 12 |

| 16 |

| 72 |

+------------+

6 rows in set (0.00 sec)

**LOGICAL OPERATORS:**

**DEFINITION**:These operators perform the boolean operations,which give two results true and false.

**OBJECTIVE**:Implementation of all types of logical operator as mentioned below.

**1-ALL**

**SYNTAX**-select column\_name1.... from table\_name where column comparison\_operator all(select column from table\_name);

**DEFINITION:** compares the value of a column from the database

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select id from workers where salary>all(select emp\_id from workers where salary>1200);

ERROR 1054 (42S22): Unknown column 'emp\_id' in 'field list'

mysql> select id from workers where salary>all(select id from workers where salary>1200);

+-----+

| id |

+-----+

| 12 |

| 14 |

| 34 |

| 234 |

+-----+

4 rows in set (0.01 sec)

**2-AND :**

**DEFINITION:** Used to filter records based on one or more condition.

**SYNTAX:** select \* from table\_name

Where condition 1 and condition 2

**EXAMPLE:**

mysql> select \* from emp\_details;

+--------+----------------+---------------+------------+-------------+

| emp\_no | emp\_first\_name | emp\_last\_name | emp\_salary | emp\_address |

+--------+----------------+---------------+------------+-------------+

| 123 | suraj | singh | 1000 | NULL |

| 124 | suresh | kumar | 1000 | NULL |

| 237 | ravi | pandey | 1000 | NULL |

| 1237 | rajkumar | shetty | 1000 | NULL |

| 1337 | rajkumar | rao | 1000 | NULL |

| 3437 | ramcharan | shetty | 1000 | NULL |

| 3537 | ramcharan | singhaniya | 1000 | NULL |

+--------+----------------+---------------+------------+-------------+

7 rows in set (0.00 sec)

mysql> select \* from emp\_details where emp\_last\_name='shetty' and emp\_first\_name='rajkumar';

+--------+----------------+---------------+------------+-------------+

| emp\_no | emp\_first\_name | emp\_last\_name | emp\_salary | emp\_address |

+--------+----------------+---------------+------------+-------------+

| 1237 | rajkumar | shetty | 1000 | NULL |

+--------+----------------+---------------+------------+-------------+

1 row in set (0.00 sec)

**3-OR :**

**DEFINITION:**Used to filter records based on one or more condition.

**SYNTAX:** select \* from table\_name

Where condition 1 and condition 2

**EXAMPLE:**

mysql> select \* from emp\_details where emp\_first\_name='ramcharan' OR emp\_last\_name=' shetty';

+--------+----------------+---------------+------------+-------------+

| emp\_no | emp\_first\_name | emp\_last\_name | emp\_salary | emp\_address |

+--------+----------------+---------------+------------+-------------+

| 3437 | ramcharan | shetty | 1000 | NULL |

| 3537 | ramcharan | singhaniya | 1000 | NULL |

+--------+----------------+---------------+------------+-------------+

2 rows in set (0.00 sec)

**4-LIKE:**

**DEFINITION:**Like operator is used in where clause to search for a specified pattern in a column.

**SYNTAX:**Select column1,column2,……..

From table\_name

Where column like pattern;

**EXAMPLE:**

mysql> select \* from emp\_details;

+--------+----------------+---------------+------------+-------------+

| emp\_no | emp\_first\_name | emp\_last\_name | emp\_salary | emp\_address |

+--------+----------------+---------------+------------+-------------+

| 123 | suraj | singh | 1000 | NULL |

| 124 | suresh | kumar | 1000 | NULL |

| 237 | ravi | pandey | 1000 | NULL |

| 1237 | rajkumar | shetty | 1000 | NULL |

| 1337 | rajkumar | rao | 1000 | NULL |

| 3437 | ramcharan | shetty | 1000 | NULL |

| 3537 | ramcharan | singhaniya | 1000 | NULL |

+--------+----------------+---------------+------------+-------------+

7 rows in set (0.00 sec)

mysql> select \* from emp\_details where emp\_first\_name like 'r%';

+--------+----------------+---------------+------------+-------------+

| emp\_no | emp\_first\_name | emp\_last\_name | emp\_salary | emp\_address |

+--------+----------------+---------------+------------+-------------+

| 237 | ravi | pandey | 1000 | NULL |

| 1237 | rajkumar | shetty | 1000 | NULL |

| 1337 | rajkumar | rao | 1000 | NULL |

| 3437 | ramcharan | shetty | 1000 | NULL |

| 3537 | ramcharan | singhaniya | 1000 | NULL |

+--------+----------------+---------------+------------+-------------+

5 rows in set (0.01 sec)

**5-BETWEEN**

**DEFINITION:**Shows the record mentioned in the query

**SYNTAX:**select column\_name.... from table\_name where column\_name between value1 and value2;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select \* from workers where salary between 100 and 1000;

+----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

+----+------------+-----------+--------+

2 rows in set (0.00 sec)

**6-IN**

**DEFINITION:**it allows database users to specify two or more values in a where clause.

**SYNTAX:**select column\_name.... from table\_name where column\_name in(list\_of\_values);

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select first\_name from workers where salary in (1200,300);

+------------+

| first\_name |

+------------+

| ram |

| aman |

+------------+

2 rows in set (0.00 sec)

**7-NOT**

**DEFINITION:**shows the record from the table if the condition evaluates to false.

**SYNTAX:**select column\_name.... from table\_name where column\_name where not condition;

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select first\_name from workers where not salary<500;

+------------+

| first\_name |

+------------+

| aman |

| vikas |

| bhagya |

+------------+

3 rows in set (0.01 sec)

**8-ANY**

**DEFINITION:**shows the record when any of the values returned by the sub-query meet the condition.

**SYNTAX:**select column\_name.... from table\_name where column\_name

comparison\_operator any(select column from table\_name where condition(s));

**EXAMPLE:**

mysql> select \* from workers;

+-----+------------+-----------+--------+

| id | first\_name | last\_name | salary |

+-----+------------+-----------+--------+

| 12 | ram | kumar | 300 |

| 13 | dhani | rao | 200 |

| 14 | aman | sharma | 1200 |

| 15 | deepankar | sharma | 12 |

| 34 | vikas | bhatia | 1342 |

| 234 | bhagya | sri | 1242 |

+-----+------------+-----------+--------+

6 rows in set (0.00 sec)

mysql> select id from workers where salary> any(select id from workers where salary>100);

+-----+

| id |

+-----+

| 12 |

| 13 |

| 14 |

| 34 |

| 234 |

+-----+

5 rows in set (0.00 sec)