Experiment No: 4

Title : Implementation of different types of operators in SQL.

- Arithmetic Operator
- Logical Operator
- Comparison Operator
- Special Operator
- Set Operator

Objective:

To learn different types of operators.

Theory:

ARITHMETIC OPERATORS:

(+) : Addition - Adds values on either side of the operator .

(-):Subtraction - Subtracts right hand operand from left hand operand .

(*):Multiplication - Multiplies values on either side of the operator .

(/):Division - Divides left hand operand by right hand operand .

(^):Power- raise to power of .

(%):Modulus - Divides left hand operand by right hand operand and returns remainder.

LOGICAL OPERATORS:

AND : The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.

OR: The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.

NOT: The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. **This is a negated operator.**

COMPARISON OPERATORS:

(=): Checks if the values of two operands are equal or not, if yes then condition becomes true.

(!=): Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.

(< >): Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.

(>): Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true

(<): Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.

(>=): Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.

(<=): Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.

SPECIAL OPERATOR:

BETWEEN: The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.

IS NULL: The NULL operator is used to compare a value with a NULL attributevalue.
ALL: The ALL operator is used to compare a value to all values in another value set.

ANY: The ANY operator is used to compare a value to any applicable value in the list according to the condition.

LIKE: The LIKE operator is used to compare a value to similar values using wildcard operators.It allows to use percent sign(%) and underscore (_) to match a given string pattern.

IN: The IN operator is used to compare a value to a list of literal values that have been specified.

EXIST: The EXISTS operator is used to search for the presence of a row in a specified table that meets certain criteria.

SET OPERATORS:

The Set operator combines the result of 2 queries into a single result. The following are the operators:

- Union
- Union all
- Intersect
- Minus

Union: Returns all distinct rows selected by both the queries

Union all: Returns all rows selected by either query including the duplicates.

Intersect: Returns rows selected that are common to both queries.

Minus: Returns all distinct rows selected by the first query and are not by the second

LAB PRACTICE ASSIGNMENT:

1. Display all the dept numbers available with the dept and emp tables avoiding duplicates.

```
mysql> SELECT department_number
    -> FROM department
    ->
    -> UNION
    ->
    -> SELECT department_number
    -> FROM employee;
+-------------------+
| department_number |
+-------------------+
|               101 |
|               102 |
|               103 |
|               104 |
|               105 |
|               201 |
|               202 |
|               203 |
|               204 |
|               205 |
|               106 |
|               107 |
+-------------------+
12 rows in set (0.00 sec)
```

2. Display all the dept numbers available with the dept and emp tables.

```
mysql> SELECT department_number
    -> FROM department
    ->
    -> UNION ALL
    ->
    -> SELECT department_number
    -> FROM employee;
+-------------------+
| department_number |
+-------------------+
|               101 |
|               102 |
|               103 |
|               104 |
|               105 |
|               201 |
|               202 |
|               203 |
|               204 |
|               205 |
|               101 |
|               102 |
|               103 |
|               104 |
|               105 |
|               103 |
|               106 |
|               102 |
|               101 |
|               107 |
```

3. Display all the dept numbers available in emp and not in dept tables and vice versa.

```
mysql> SELECT DISTINCT d.department_number
    -> FROM department d
    -> LEFT JOIN employee e ON d.department_number = e.department_number
    -> WHERE e.department_number IS NULL;
+-------------------+
| department_number |
+-------------------+
|               201 |
|               202 |
|               203 |
|               204 |
|               205 |
+-------------------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT e.department_number
    -> FROM employee e
    -> LEFT JOIN department d ON e.department_number = d.department_number
    -> WHERE d.department_number IS NULL;
+-------------------+
| department_number |
+-------------------+
|               106 |
|               107 |
+-------------------+
2 rows in set (0.00 sec)
```