SQL logic Operations and Operators (AND, OR, NOT, BETWEEN, LIKE, IN, HAVING commands)

The Logical Operator is nothing but which returns the result in one form, i.e., either it will display the query is true, or the query is false. The results displayed to combine or merge more than one true or false data. So, SQL logical operators are used to test for the truth of the condition.

The Logical Operators in SQL are as follows:

Operator	Meaning
AND	TRUE if both Boolean expressions are TRUE.
IN	TRUE if the operand is equal to one of a list of expressions.
NOT	Reverses the value of any other Boolean operator.
OR	TRUE if either Boolean expression is TRUE.
LIKE	TRUE if the operand matches a pattern.
BETWEEN	TRUE if the operand is within a range.
ALL	TRUE if all of a set of comparisons are TRUE.
ANY	TRUE if any one of a set of comparisons is TRUE.
EXISTS	TRUE if a subquery contains any rows.
SOME	TRUE if some of a set of comparisons are TRUE.

Let us Discuss one by One. To move ahead let us take the help of two tables created earlier in **Experiment No. 10.**

Let us see the recods of the table STUDENT and TEACHER.

Statement: **SELECT * FROM student;**

Output:

```
mysql> select * from student;
+---------+----------+---------+--------+----------+
| roll_no | fname    | lname   | sex    | state    |
+---------+----------+---------+--------+----------+
|       1 | Robin    | Das     | Male   | Assam    |
|       2 | Dwipen   | Laskar  | Male   | Assam    |
|       3 | Asangla  | Sema    | Female | Nagaland |
|       4 | Praneeta | Rabha   | Female | Manipur  |
|       5 | Rosy     | Kalita  | Female | Tripura  |
|       6 | Naznin   | Akhtara | Female | Assam    |
|       7 | Karabi   | Bora    | Female | Manipur  |
|       8 | Anil     | Gogoi   | Male   | Manipur  |
+---------+----------+---------+--------+----------+
8 rows in set (0.45 sec)
```

Let us see the recods of the table TEACHER by using SELECT * statement as below:

Statement: **SELECT * FROM teacher;**

Output:

```
mysql> select * from teacher;
+-----+----------+-----------+--------+-------------+
| tid | t_fname  | t_lname   | sex    | address     |
+-----+----------+-----------+--------+-------------+
|   1 | Monoj    | Pathak    | Male   | Assam       |
|   2 | K.V.     | Kanimozhi | Female | Kerala      |
|   3 | Atul     | Agnihotri | Male   | Maharastra  |
|   4 | Praneeta | Bodo      | Female | Assam       |
|   5 | Priyanka | Bsumatari | Male   | Mizoram     |
|   6 | Bikash   | Punchal   | Male   | West Bangal |
+-----+----------+-----------+--------+-------------+
6 rows in set (0.14 sec)
```

1) AND Operator

The SQL AND operator is used with the where clause in the SQL Query. AND operator in SQL returns only those records which satisfy both the conditions in the SQL query.

Example: Suppose, we want to to retrieve only those records of students from the student table who are ‘male’ and belonging to the state of ‘Assam’

Statement:

SELECT * FROM student WHERE sex = "Male" AND state = "Assam";

Output:

```
mysql> SELECT * FROM student WHERE sex = "Male" AND state = "Assam";
+---------+--------+--------+------+-------+
| roll_no | fname  | lname  | sex  | state |
+---------+--------+--------+------+-------+
|       1 | Robin  | Das    | Male | Assam |
|       2 | Dwipen | Laskar | Male | Assam |
+---------+--------+--------+------+-------+
2 rows in set (1.17 sec)
```

2) OR operator

The SQL OR operator is used with the where clause in an SQL Query. OR operator in SQL returns only those records that satisfy any of the conditions in the SQL query.

Example: Suppose, we want to to retrieve only those records of students from the student table who are either ‘male’ or belonging to the state of ‘Assam’

Statement:

SELECT * FROM student WHERE sex = "Male" OR state = "Assam";

Output:

```
mysql> SELECT * FROM student WHERE sex = "Male" OR state = "Assam";
+---------+--------+---------+--------+---------+
| roll_no | fname  | lname   | sex    | state   |
+---------+--------+---------+--------+---------+
|       1 | Robin  | Das     | Male   | Assam   |
|       2 | Dwipen | Laskar  | Male   | Assam   |
|       6 | Naznin | Akhtara | Female | Assam   |
|       8 | Anil   | Gogoi   | Male   | Manipur |
+---------+--------+---------+--------+---------+
4 rows in set (0.05 sec)
```

3) IN Operator

When we want to check for one or more than one value in a single SQL query, we use IN operator with the WHERE clause in a SELECT query.

Example Suppose, we want to to retrieve only those records of students from the student table who belonging to the any of the states {'Assam', Manipur, 'Mizoram');

Statement:

SELECT * FROM student WHERE state IN ("Assam", "Manipur", "Mizoram");

Output:

```
mysql> SELECT * FROM student WHERE state IN ("Assam", "Manipur", "Mizoram");
+---------+----------+---------+--------+---------+
| roll_no | fname    | lname   | sex    | state   |
+---------+----------+---------+--------+---------+
|       1 | Robin    | Das     | Male   | Assam   |
|       2 | Dwipen   | Laskar  | Male   | Assam   |
|       4 | Praneeta | Rabha   | Female | Manipur |
|       6 | Naznin   | Akhtara | Female | Assam   |
|       7 | Karabi   | Bora    | Female | Manipur |
|       8 | Anil     | Gogoi   | Male   | Manipur |
+---------+----------+---------+--------+---------+
6 rows in set (0.04 sec)
```

4) NOT Operator

NOT operator in SQL shows those records from the table where the criteria is not met. NOT operator is used with where clause in a SELECT query.

Example-1: Suppose, we want to to retrieve only those records of students from the student table who are not beloing to the state of 'Assam';

Statement:

SELECT * FROM student WHERE NOT state="Assam";

Output:

```
mysql> SELECT * FROM student WHERE NOT state="Assam";
+---------+----------+--------+--------+----------+
| roll_no | fname    | lname  | sex    | state    |
+---------+----------+--------+--------+----------+
|       3 | Asangla  | Sema   | Female | Nagaland |
|       4 | Praneeta | Rabha  | Female | Manipur  |
|       5 | Rosy     | Kalita | Female | Tripura  |
|       7 | Karabi   | Bora   | Female | Manipur  |
|       8 | Anil     | Gogoi  | Male   | Manipur  |
+---------+----------+--------+--------+----------+
5 rows in set (0.06 sec)
```

Example-2: Suppose, we want to to retrieve only those records of students from the student table who are not beloing to the any state of { ‘Assam’, ‘Nagaland’};

Statement:

SELECT * FROM student WHERE state NOT IN ('Assam','Nagaland');

Output:

```
mysql> SELECT * FROM student WHERE state NOT IN ('Assam','Nagaland');
+---------+----------+--------+--------+---------+
| roll_no | fname    | lname  | sex    | state   |
+---------+----------+--------+--------+---------+
|       4 | Praneeta | Rabha  | Female | Manipur |
|       5 | Rosy     | Kalita | Female | Tripura |
|       7 | Karabi   | Bora   | Female | Manipur |
|       8 | Anil     | Gogoi  | Male   | Manipur |
+---------+----------+--------+--------+---------+
```

5) LIKE Operator

LIKE Operator in SQL displays only those data from the table which matches the pattern specified in the query. Percentage (%) and underscore (_) are the two wildcard operators used with LIKE Operator to perform pattern matching tasks.

Example-1: Suppose, we want to to retrieve only those records of students from the student table whos first name start with the letter ‘R’;

Statement:

SELECT * FROM student WHERE fname LIKE "R%";

Output:

```
mysql> SELECT * FROM student WHERE fname LIKE "R%";
+---------+-------+--------+--------+---------+
| roll_no | fname | lname  | sex    | state   |
+---------+-------+--------+--------+---------+
|       1 | Robin | Das    | Male   | Assam   |
|       5 | Rosy  | Kalita | Female | Tripura |
+---------+-------+--------+--------+---------+
2 rows in set (0.13 sec)
```

Example-2: Suppose, we want to to retrieve only those records of students from the student table whos first name has any letter but end with ‘obin’;

Statement:

SELECT * FROM student WHERE fname LIKE "_obin";

Output:

```
mysql> SELECT * FROM student WHERE fname LIKE "_obin";
+---------+-------+-------+------+-------+
| roll_no | fname | lname | sex  | state |
+---------+-------+-------+------+-------+
|       1 | Robin | Das   | Male | Assam |
+---------+-------+-------+------+-------+
1 row in set (0.00 sec)
```

6) BETWEEN operator

This operator displays the records which fall between the given ranges in the SQL query. The results of the BETWEEN operator include begin and end values of the given range.

Example: Suppose, we want to to retrieve only those records of students from the student table whos roll numbers lie between the range of 4 to 6.;

Statement:

SELECT * FROM student WHERE roll_no BETWEEN 4 AND 6;

Output:

```
mysql> SELECT * FROM student WHERE roll_no BETWEEN 4 AND 6;
+---------+----------+---------+--------+---------+
| roll_no | fname    | lname   | sex    | state   |
+---------+----------+---------+--------+---------+
|       4 | Praneeta | Rabha   | Female | Manipur |
|       5 | Rosy     | Kalita  | Female | Tripura |
|       6 | Naznin   | Akhtara | Female | Assam   |
+---------+----------+---------+--------+---------+
3 rows in set (0.03 sec)
```

7) ALL operator

The ALL keyword is a MySQL operator that returns the Boolean value TRUE if the comparison is TRUE for ALL of the subquery condition.

Example: Suppose, we want to to retrieve only those records of students from the student table whos roll numbers are greater than all the tid that are greater than 3 teacher table.

Statement:

SELECT * FROM student WHERE roll_no >ALL (SELECT tid FROM teacher WHERE tid>3);

Output:

```
mysql> SELECT * FROM student WHERE roll_no >ALL (SELECT tid FROM teacher WHERE tid>3);
+---------+--------+-------+--------+---------+
| roll_no | fname  | lname | sex    | state   |
+---------+--------+-------+--------+---------+
|       7 | Karabi | Bora  | Female | Manipur |
|       8 | Anil   | Gogoi | Male   | Manipur |
+---------+--------+-------+--------+---------+
2 rows in set (0.00 sec)
```

8) ANY operator

The ANY keyword is a MySQL operator that returns the Boolean value TRUE if the comparison is TRUE for ANY of the subquery condition.

Example: Suppose, we want to to retrieve only those records of students from the student table whos roll numbers are greather than any of the teacher id greater than 3 in teacher table.

Statement:

SELECT * FROM student WHERE roll_no >ALL (SELECT tid FROM teacher WHERE tid>3);

Output:

```
mysql> SELECT * FROM student WHERE roll_no >ANY (SELECT tid FROM teacher WHERE tid>3);
+---------+--------+---------+--------+---------+
| roll_no | fname  | lname   | sex    | state   |
+---------+--------+---------+--------+---------+
|       5 | Rosy   | Kalita  | Female | Tripura |
|       6 | Naznin | Akhtara | Female | Assam   |
|       7 | Karabi | Bora    | Female | Manipur |
|       8 | Anil   | Gogoi   | Male   | Manipur |
+---------+--------+---------+--------+---------+
4 rows in set (0.04 sec)
```

9) EXIST operator

The EXISTS operator in MySQL is a type of Boolean operator which returns the true or false result. It is used in combination with a subquery and checks the existence of data in a subquery. It means if a subquery returns any record, this operator returns true. Otherwise, it will return false. The true value is always represented numeric value 1, and the false value represents 0. We can use it with SELECT, UPDATE, DELETE, INSERT statement.

Example: Suppose we are going to use EXISTS operator to find the *roll*_no,*fname*,*lname* and *state* of the students who are belonging to any of the state belongin by any of the teacher:

Statement:

SELECT roll_no, fname, lname, state FROM student WHERE EXISTS (SELECT address FROM teacher where student.state=teacher.address);

Output:

```
mysql> SELECT roll_no, fname, lname, state FROM student  WHERE EXISTS (SELECT address
    -> FROM teacher where student.state=teacher.address);
+---------+--------+---------+-------+
| roll_no | fname  | lname   | state |
+---------+--------+---------+-------+
|       1 | Robin  | Das     | Assam |
|       2 | Dwipen | Laskar  | Assam |
|       6 | Naznin | Akhtara | Assam |
+---------+--------+---------+-------+
3 rows in set (0.05 sec)
```

10) SOME operator

SOME operator evaluates the condition between the outer and inner tables and evaluates to true if the final result returns any one row. If not, then it evaluates to false.

Example: Suppose, we want to to retrieve those records of students from the student table whos roll numbers is greater than some of the roll numbers of the students belonging to state of *Assam*.

Statement:

SELECT * FROM students WHERE roll_no > SOME (SELECT roll_no FROM student WHERE state=’Assam’);

Output:

```
mysql> SELECT * FROM student WHERE roll_no > SOME (SELECT roll_no FROM student WHERE state='Manipur');
+---------+--------+---------+--------+---------+
| roll_no | fname  | lname   | sex    | state   |
+---------+--------+---------+--------+---------+
|       5 | Rosy   | Kalita  | Female | Tripura |
|       6 | Naznin | Akhtara | Female | Assam   |
|       7 | Karabi | Bora    | Female | Manipur |
|       8 | Anil   | Gogoi   | Male   | Manipur |
+---------+--------+---------+--------+---------+
4 rows in set (0.00 sec)
```