

MySQL Conditions

MySQL Conditions:

MySQL AND

MySQL OR

MySQL AND OR

MySQL Boolean

MySQL LIKE

MySQL IN

MySQL ANY

MySQL Exists

MySQL NOT

MySQL Not Equal

MySQL IS NULL

MySQL IS NOT NULL

MySQL BETWEEN

1. MySQL AND

The AND operator is used to combine multiple conditions in a query. All conditions must be true for the row to be included in the result set.

I.M.P => if all condition is true then and only then give Result data.

Syntax:

SELECT column1, column2, ...

FROM table_name

WHERE condition1 AND condition2 AND condition3 ...;

Example:

SELECT * FROM employees

WHERE department = 'Sales' AND salary > 50000;

=> This query selects all employees from the Sales department with a salary greater than 50,000.

2. MySQL OR

The OR operator is used to combine multiple conditions, where at least one condition must be true for the row to be included in the result set.

I.M.P => if any one condition is true then also give Result data.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

Example:

```
SELECT * FROM employees
WHERE department = 'Sales' OR department = 'HR';
```

=> This query selects all employees from either the Sales or HR departments.

3. MySQL AND OR :

You can combine AND and OR operators to form complex conditions.

Parentheses () can be used to specify the order of evaluation.

Example:

```
SELECT * FROM employees
```

```
WHERE (department = 'Sales' OR department = 'HR') AND  
salary > 50000;
```

=> This query selects employees who either work in Sales or HR and have a salary greater than 50,000.

4. MySQL LIKE

The LIKE operator is used for pattern matching in string comparisons. It can include wildcard characters:

=> "%" represents zero, one, or multiple characters.

=> "_" represents a single character.

Syntax:

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE column LIKE pattern;
```

Example:

```
SELECT * FROM customers  
WHERE name LIKE 'A%';
```

=> For Better Understanding ...

1. "%" can match zero, one, or multiple characters:

a% could match "apple," "ant," "arctic," "a," etc.

%er could match "water," "player," "super," etc.

%is% could match "this," "wisdom," "list," etc.

2. "_" matches exactly one character:

_a could match "ha," "ca," "ba," etc.

b_t could match "bat," "bet," "bit," etc.

c_t% could match "cat," "cut," "cart," "cute," etc.

5. MySQL Boolean :

MySQL supports Boolean data types, which are usually represented as TINYINT with values 0 (false) and 1 (true).

=> Example:

```
SELECT * FROM products  
WHERE available = 1;
```

here , -- where available is a Boolean column

=> This query selects all products that are available (where available is true).

6. MySQL IN :

The IN operator is used to specify multiple values in a WHERE clause.

=> Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE column IN (value1, value2, ...);
```

=> Example:

```
SELECT * FROM employees
WHERE department IN ('Sales', 'HR', 'Finance');
```

=> This query selects employees who work in Sales, HR, or Finance.

7. MySQL ANY :

The ANY operator is used to compare a value to any value in a list or subquery.

=> Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE column operator ANY (subquery);
```

=> Example:

```
SELECT * FROM employees
WHERE salary > ANY (SELECT salary FROM managers);
```

=> This query selects employees whose salary is greater than any salary in the managers' salaries.

8. MySQL EXISTS :

The EXISTS operator is used to test for the existence of any record in a subquery.

It returns true if the subquery returns one or more records.

=> Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE EXISTS (subquery);
```

=> Example:

```
SELECT * FROM employees e
WHERE EXISTS (SELECT 1 FROM managers m WHERE e.id
= m.employee_id);
```

=> This query selects all employees who are also listed in the managers' table.

9. MySQL NOT :

The NOT operator negates a condition, returning true if the condition is false.

=> NOT is used to negate conditions. It is commonly used with operators like IN, LIKE, or with other logical conditions, but not directly with single value comparisons.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

Example:

```
SELECT * FROM employees
WHERE NOT department = 'HR';
```

=> This query selects all employees who are not in the HR department.

10. MySQL Not Equal :

The != or <> operator is used to test if two values are not equal.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE column != value;
```

Example:

```
SELECT * FROM products  
WHERE price != 100;
```

=> This query selects all products with a price not equal to 100.

11. MySQL IS NULL

=> The IS NULL condition is used to check for null values.

Syntax:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column IS NULL;
```

Example:

```
SELECT * FROM employees  
WHERE manager_id IS NULL;
```

=> This query selects employees who do not have a manager.

12. MySQL IS NOT NULL :

=> The IS NOT NULL condition is used to check for non-null values.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE column IS NOT NULL;
```

Example:

```
SELECT * FROM employees
WHERE manager_id IS NOT NULL;
```

=> This query selects employees who have a manager.

13. MySQL BETWEEN :

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

Syntax:

SELECT column1, column2, ...

FROM table_name

WHERE column BETWEEN value1 AND value2;

Example:

SELECT * FROM orders

WHERE order_date BETWEEN '2023-01-01' AND '2023-12-31';

=> This query selects orders made between January 1, 2023, and December 31, 2023.

👉 Practice Questions :

1. Retrieve all records from the students table where the grade is 'A' and the attendance is greater than 90.

2. Select all employees from the employees table who work in the 'Finance' department or have a job title of 'Manager'.

3. Find all products from the products table where the name starts with 'S' and has a length of 5 characters.
4. Select all users from the users table where the active column is true.
5. Retrieve all items from the items table where the description contains the word 'laptop'.
6. Select all customers from the customers table where the city is either 'Chicago', 'Dallas', or 'San Francisco'.
7. Find all orders from the orders table where the order_date is within the last 30 days.
8. Retrieve all employees from the employees table where their salary is not equal to 60000.
9. Select all records from the sales table where the total is between 100 and 500.
10. Find all records in the projects table where the end_date is NULL.

11. Retrieve all transactions from the transactions table where the amount is not NULL.
12. Select all products from the products table where the category is not 'Electronics'.
13. Find all orders from the orders table where the order_status is 'Pending' and the customer_id is in the list of VIP customers (use a subquery).
14. Retrieve all records from the logs table where the error_code matches any code in a list of known issues (use a subquery with ANY).

👉 ANS

1. Retrieve all records from the students table where the grade is 'A' and the attendance is greater than 90.
=> `SELECT * FROM students
WHERE grade = 'a' AND attendance > 90`

2. Select all employees from the employees table who work in the 'Finance' department or have a job title of 'Manager'.

=> SELECT * FROM employees

WHERE department = 'Finance' OR title = 'manager'

3. Find all products from the products table where the name starts with 'S' and has a length of 5 characters.

=> SELECT * FROM products

WHERE name LIKE 's_____'

4. Select all users from the users table where the active column is true.

=> SELECT * FROM users WHERE active = 1

5. Retrieve all items from the items table where the description contains the word 'laptop'.

=> SELECT * FROM items

WHERE description LIKE '%laptop%'

6. Select all customers from the customers table where the city is either 'Chicago', 'Dallas', or 'San Francisco'.

=> SELECT * FROM customers

WHERE city IN ('Chicago', 'Dallas', 'San Francisco')

7. Find all orders from the orders table where the order_date is within the last 30 days.

=> SELECT * FROM orders

WHERE date > CURDATE() - INTERVAL 30 DAY

8. Retrieve all employees from the employees table where their salary is not equal to 60000.

=> SELECT * FROM employees

WHERE salary != 60000

9. Select all records from the sales table where the total is between 100 and 500.

=> SELECT * FROM sales

WHERE total BETWEEN 100 AND 500

10. Find all records in the projects table where the end_date is NULL.

=> SELECT * FROM projects

WHERE end_date IS null

11. Retrieve all transactions from the transactions table where the amount is not NULL.
=> SELECT * FROM transactions
WHERE amount IS NOT NULL

12. Select all products from the products table where the category is not 'Electronics'.
=> SELECT * FROM products
WHERE category != 'Electronics'

13. Find all orders from the orders table where the order_status is 'Pending' and the customer_id is in the list of VIP customers (use a subquery).
=> SELECT * FROM orders
WHERE order_status = 'Pending' AND customer_id IN
(SELECT customer_id FROM vip_customers)

14. Retrieve all records from the logs table where the error_code matches any code in a list of known issues (use a subquery with ANY).
=> SELECT * FROM logs
WHERE error_code IN (SELECT known_issue_code FROM known_issues)