Logical Operators | WHERE Clause

Logical Operators within the WHERE Clause allow you to combine multiple conditions to create more complex filters for your SQL queries.

Here are the main logical operators and how to use them:

---

1. AND

- **Usage:** Combines two or more conditions; all conditions must be `TRUE` for the record to be selected.
- **Example:**

SQL

```sql
SELECT *
FROM employees
WHERE department = 'HR'
  AND salary > 50000;

```

Explanation: Fetches employees who are in the HR department **and** earn a salary greater than 50,000.

---

2. OR

- **Usage:** Combines two or more conditions; at least one condition must be `TRUE` for the record to be selected.
- **Example:**

SQL

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

```

Explanation: Fetches employees who are in either the HR **or** IT department.

---

3. NOT

- **Usage:** Negates a condition, selecting records where the condition is `FALSE`.
- **Example:**

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

```

Explanation: Fetches employees who are **not** in the HR department.

---

4. Combining AND and OR

- **Usage:** Use parentheses to group conditions and define the order of evaluation.
- **Example:**

```sql
SELECT *
FROM employees
WHERE (department = 'HR' OR department = 'IT')
  AND salary > 50000;

```

Explanation: Fetches employees who are in either the HR **or** IT department **and** earn a salary greater than 50,000.

---

5. BETWEEN with AND

- **Usage:** Used to filter values within a range.
- **Example:**

```sql
SELECT *
FROM orders
WHERE order_date BETWEEN '2023-01-01' AND '2023-12-
31';
```

```
4
```

Explanation: Fetches orders placed between January 1, 2023, and December 31, 2023.

---

6. IN with OR

- **Usage:** Acts as a shorthand for multiple `OR` conditions.
- **Example:**

SQL

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

```

Explanation: Fetches employees who are in the HR, IT, or Finance departments.
Equivalent to:

SQL

```sql
WHERE department = 'HR'
   OR department = 'IT'
   OR department = 'Finance';

```

---

7. Combining NOT with Other Operators

- **Example with NOT IN:**

SQL

```sql
SELECT *
FROM employees
WHERE department NOT IN ('HR', 'IT');

```

Explanation: Fetches employees who are not in the HR or IT department.

- **Example with NOT LIKE:**

SQL

```sql
SELECT *
FROM employees
WHERE name NOT LIKE 'A%';

```

Explanation: Fetches employees whose names do not start with "A".

---

8. IS NULL with NOT

- **Usage:** Checks for null or non-null values.
- **Example:**

SQL

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

```

Explanation: Fetches employees who have a manager assigned (non-null values).

Order of Evaluation (Precedence)

1. Parentheses `( )`
2. NOT
3. AND
4. OR

Example with Precedence:

SQL

```sql
SELECT *
FROM employees
WHERE NOT department = 'HR'
    AND (salary > 50000 OR age < 30);

```

Explanation:

- First, the `NOT department = 'HR'` condition is evaluated.
- Then, the `salary > 50000 OR age < 30` condition is evaluated.
- Finally, both are combined using `AND`.

Tips:

- Use parentheses to clarify logic, especially when combining `AND` and `OR`.
- Test queries with subsets of conditions to ensure each part behaves as expected.

Would you like example data to practice these concepts?