SQL: WHERE

The `WHERE` clause in SQL is used to filter records in a database table, allowing you to retrieve only those rows that meet a specified condition or set of conditions. This is essential for narrowing down query results to only the data that matches certain criteria, making the `WHERE` clause one of the most powerful tools in SQL for data selection. The general syntaxt writes:

```
SELECT column1, column2, ...  FROM table_name WHERE conditi
on;
```

So the condition can take really any conditions:

- Specific value

  ```
  SELECT * FROM table_name WHERE column='value'; -- the va
  lue can be int, string
  ```

- With a threshold

  ```
  SELECT * FROM table_name WHERE column < 5; -- for instan
  ce.
  ```

Not that similarly to programming, we can put a condition that will take the opposite by using the `NOT` keyword:

```
SELECT column1, column2, ... FROM table_name WHERE NOT cond
ition;
```

For the comparison operators we have:

```
10 > 20 -- false
10 < 20 -- true
10 <= 20 -- false
10 >= 9 -- true
0 = 0 -- true
1 !=0  -- true or we can write 1 <> 0
```

```
'abc' > 'ace' -- true because in the alphabet 'ab' comes be
fore 'ac'
```

The `WHERE` clause filters the results of a `SELECT` query to include only those rows that satisfy the condition specified. Conditions in the `WHERE` clause can be simple or complex and can use a variety of operators, such as:

- **Comparison Operators**: `=`, `!=` (or `<>`), `<`, `>`, `<=`, `>=`
- **Logical Operators**: `AND`, `OR`, `NOT`
- **Other Operators**: `BETWEEN`, `IN`, `LIKE`, `IS NULL`

Examples of Using `WHERE`

Let's go through some examples to understand how the `WHERE` clause works:

Example Table: Employees

employee_id	first_name	last_name	department	salary
1	John	Doe	Sales	50000
2	Jane	Smith	Marketing	60000
3	Alice	Johnson	Sales	55000
4	Bob	Brown	IT	70000

1. **Simple `WHERE` Clause**:

   If you want to select all employees in the Sales department:

   ```
   SELECT employee_id, first_name, last_name
   FROM Employees
   WHERE department = 'Sales';
   ```

   Result:

employee_id	first_name	last_name
1	John	Doe
3	Alice	Johnson

2. **Using `WHERE` with Comparison Operators**:

To find all employees with a salary greater than 60,000:

```
SELECT first_name, last_name, salary
FROM Employees
WHERE salary > 60000;
```

Result:

first_name	last_name	salary
Bob	Brown	70000

3. **Combining Conditions with `AND` and `OR`**:

   To find all employees in the Sales department with a salary above 50,000:

   ```
   SELECT first_name, last_name, salary
   FROM Employees
   WHERE department = 'Sales' AND salary > 50000;
   ```

   Result:

first_name	last_name	salary
Alice	Johnson	55000

   To find employees in either the Sales or IT departments:

   ```
   SELECT first_name, last_name, department
   FROM Employees
   WHERE department = 'Sales' OR department = 'IT';
   ```

   Result:

first_name	last_name	department
John	Doe	Sales
Alice	Johnson	Sales
Bob	Brown	IT

4. **Using `WHERE` with `LIKE` for Pattern Matching**:

   To find all employees whose last name starts with 'J':

```
SELECT first_name, last_name
FROM Employees
WHERE last_name LIKE 'J%';
```

Result:

first_name	last_name
Alice	Johnson

Key Points

- The `WHERE` clause is essential for filtering data based on specific criteria.
- You can use a wide range of operators to define conditions in the `WHERE` clause.
- Conditions can be combined using logical operators like `AND`, `OR`, and `NOT`.
- The `WHERE` clause can be used with `SELECT`, `UPDATE`, `DELETE`, and other SQL statements to narrow down the data affected by the query.

The `WHERE` clause gives you precise control over which rows are included in your query results, making it a crucial part of almost any SQL operation that requires conditional logic.