

SQL Where Clause

In the previous article we have seen about how to write effective SELECT statement to get data from database.

The SQL WHERE clause is powerful & one of the most commonly used clause use in day to day life. Basically It allows you to filter out the result set and your get limited data based on condition. The WHERE clause can be used alongwith the **SQL SELECT statement, SQL INSERT statement,, SQL UPDATE statement, or SQL DELETE statement.**

SQL WHERE clause syntax

It is the most frequently used SQL command and following is the general syntax of SELECT command with WHERE clause to fetch data from table:

```
1 SELECT column_name(s)
```

```
2 FROM table_name
```

```
3 WHERE column_name operator value
```

- **[SELECT column_name(s)]** – The result set retrieve all entered columns data. It may contain multiple columns. If pass '*' then it results all columns from selected table.
- **[FROM table_name]** – This is a table name from which result set is retrieved.
- **[WHERE]** – This Clause used to filter out the result data set based on "condition". You can specify any condition using WHERE clause. It supports **AND** or **OR** operators to specify more than one condition in single query.

The **WHERE** clause works similar to IF condition used in the coding language. This clause is used to compare given value with the field value available in MySQL table. If given value from outside is equal to the available field value in MySQL table then it returns that row.

Let's take example A=100 and B=200 and see how operators works with this example:

Operator	Description	Example
=	This operator checks the values of both fields are same.	[A = B] is False
!=	This operator checks the values of both fields are not same.	[A != B] is True.

<	This checks the value of left operand is less than the value of right operand. If this condition satisfies then result is True.	[A < B] is True.
>	This checks the value of left operand is greater than the value of right operand. If this condition satisfies then result is True.	[A > B] is not True.
<=	This checks the value of left operand is less than equal to the value of right operand. If this condition satisfies then result is True.	[A <= B] is True.
>=	This checks the value of left operand is greater than equal to the value of right operand. If this condition satisfies then result is True.	[A >= B] is not True.

Examples:

Let's see upon executing following query on Employee table resulting following list:

```
1 SELECT * FROM Employee
```

Following is the result after executing the above SQL query:

Employee_Id	Full_Name	Gender	City	Phone
1	Tanya Smith	Female	Stavanger	9123456789
2	Janet Tampi	Female	Chicago	9876543210
3	Nilsen Phil	Male	<i>NULL</i>	9112233445
4	Nilsen Jones	Male	Atlantic	9988776655
5	Peter Willams	Male	Sandnes	9944552299

Let's now look at a **practical example**.

```
1 SELECT * FROM Employee
```

```
2 WHERE Employee_Id = 2
```

Following is the filtered result after executing the above SQL query:

Employee_Id	Full_Name	Gender	City	Phone
2	Janet Tampi	Female	Chicago	9876543210

AND logical operator:

```
1 SELECT * FROM Employee
```

```
2 WHERE Full_Name = 'Nilsen Jones'
```

```
3 AND Gender = 'Male'
```

Following is the filtered result after executing the above SQL query using **AND Logical Operator**:

Employee_Id	Full_Name	Gender	City	Phone
4	Nilsen Jones	Male	Atlantic	9988776655

OR logical operator:

```
1 SELECT * FROM Employee
```

```
2 WHERE Employee_Id = 1 OR Employee_Id = 2
```

Following is the filtered result after executing the above SQL query using **OR Logical Operator**:

Employee_Id	Full_Name	Gender	City	Phone
1	Tanya Smith	Female	Stavanger	9123456789
2	Janet Tampi	Female	Chicago	9876543210

IN logical operator:

```
1 SELECT * FROM Employee
```

```
2 WHERE City IN ('Chicago', 'Atlantic')
```

Following is the filtered result after executing the above SQL query using **IN Logical Operator**:

Employee_Id	Full_Name	Gender	City	Phone
2	Janet Tampi	Female	Chicago	9876543210
4	Nilsen Jones	Male	Atlantic	9988776655

NOT IN logical operator:

```
1 SELECT * FROM Employee
```

```
2 WHERE City <strong>NOT</strong> IN ('Chicago', 'Atlantic')
```

Following is the filtered result after executing the above SQL query using **NOT IN Logical Operator**:

Employee_Id	Full_Name	Gender	City	Phone
1	Tanya Smith	Female	Stavanger	9123456789

3	Nilsen Phil	Male	NULL	9112233445
5	Peter Willams	Male	Sandnes	9944552299

BETWEEN logical operator:

```
1 SELECT * FROM Employee
```

```
2 WHERE Employee_Id BETWEEN 2 AND 4
```

Following is the filtered result after executing the above SQL query using

Employee_Id	Full_Name	Gender	City	Phone
2	Janet Tampi	Female	Chicago	9876543210
3	Nilsen Phil	Male	NULL	9112233445
4	Nilsen Jones	Male	Atlantic	9988776655

LIKE logical operator:

Ending Wildcard:

```
1 SELECT * FROM Employee
```

```
2 WHERE Full_Name LIKE 'N%'
```

Following is the filtered result after executing the above SQL query using **LIKE Logical Operator with Ending Wildcard:**

Employee_Id	Full_Name	Gender	City	Phone
--------------------	------------------	---------------	-------------	--------------

3	Nilsen Phil	Male	NULL	9112233445
4	Nilsen Jones	Male	Atlantic	9988776655

Leading Wildcard:

```
1 SELECT * FROM Employee
```

```
2 WHERE Full_Name LIKE '%s'
```

Following is the filtered result after executing the above SQL query using **LIKE Logical Operator with Leading Wildcard:**

Employee_Id	Full_Name	Gender	City	Phone
4	Nilsen Jones	Male	Atlantic	9988776655
5	Peter Willams	Male	Sandnes	9944552299

Multiple Wildcards:

Find any record which has anywhere, the letter 'a'.

```
1 SELECT * FROM Employee
```

```
2 WHERE Full_Name LIKE '%a%'
```

Following is the filtered result after executing the above SQL query using **LIKE Logical Operator with Middle Wildcard:**

Employee_Id	Full_Name	Gender	City	Phone
1	Tanya Smith	Female	Stavanger	9123456789

2	Janet Tampi	Female	Chicago	9876543210
5	Peter Willams	Male	Sandnes	9944552299

Single Character Wildcard:

Find the record which has Name started with 'J' and has 'a' character anywhere in record.

```
1 SELECT * FROM Employee
```

```
2 WHERE Full_Name LIKE 'J%a%'
```

Following is the filtered result after executing the above SQL query using **LIKE Logical Operator with Single Character Wildcard**:

Employee_Id	Full_Name	Gender	City	Phone
2	Janet Tampi	Female	Chicago	9876543210

Summary:

- The SQL WHERE clause is used to filter the number of rows in result set based on condition.
- The WHERE clause can be used with a SELECT, UPDATE or DELETE query.
- In the WHERE clause allows you to write the condition using logical operators like AND, OR, LIKE, BETWEEN, IN, NOT IN etc.
- Using AND operator means the all condition should satisfies.
- Using OR operator means the any condition satisfies.
- Using IN keyword is basically used to choose rows matching a list of values.

Source:

<http://www.softwaretestingclass.com/sql-where-clause/>