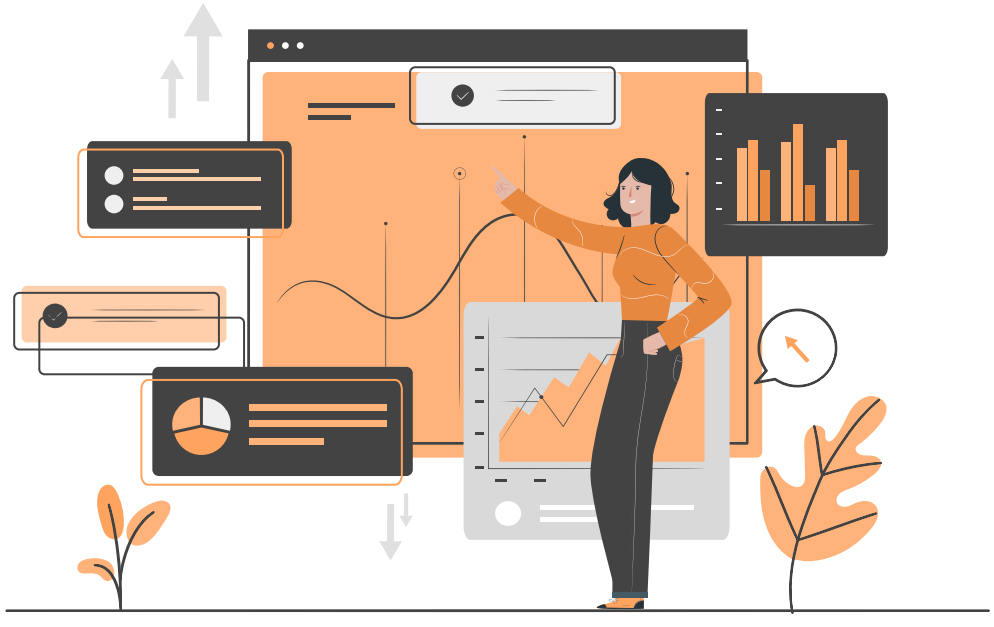
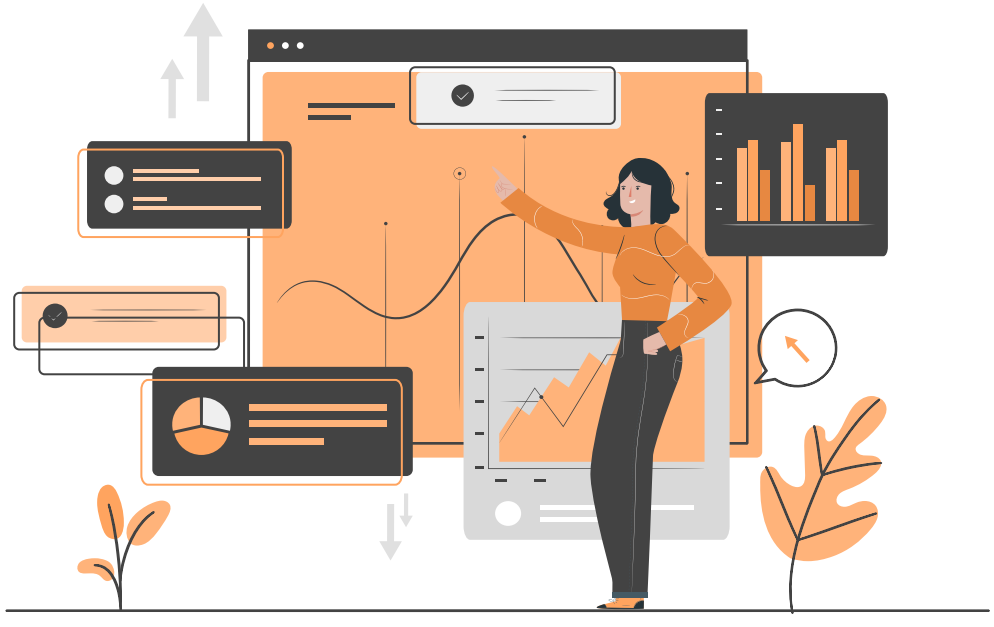


Master In SQL

By Fireblaze AI School

Index



Master In SQL

By Fireblaze AI School

Displaying Data from Multiple Tables Using Joins

- Types of JOINS and its syntax
- Natural join:
  - USING clause
  - ON clause
- Self-join
- Nonequijoins
- OUTER join
  - LEFT OUTER
  - RIGHT OUTER
  - FULL OUTER
- Cartesian product
  - Cross join

Creating Natural Joins

- The NATURAL JOIN clause is based on all the columns in the two tables that have the same name.
- It selects rows from the two tables that have equal values in all matched columns.
- If the columns having the same names have different data types, an error is returned.

Syntax:- Select Column_name, Column_name
From Table_name1
Natural Join Table_name2;

Creating Joins with the USING Clause

- If multiple columns have the same names but the data types do not match, use the **USING clause** to specify the columns for the equijoin.
- Use the USING clause to match only one column when more than one column matches.

Syntax:- Select Column_name,Column_name
From Table_name1 Join Table_name2
Using(Column_name);

Note:- Use the column_name in using which common between two.

Using Table Aliases with the USING Clause

- Do not qualify a column that is used in the USING clause.
- If the same column is used elsewhere in the SQL statement, do not alias it.

Syntax:- Select T1.Column_name,T2.Column_name
From Table1 T1 Join Table2 T2
using(Column_name);

Creating Joins with the ON Clause

- The join condition for the natural join is basically an equijoin of all columns with the same name.
- Use the ON clause to specify arbitrary conditions or specify columns to join.
- The join condition is separated from other search conditions.
- The ON clause makes code easy to understand.

Syntax:- Select T1.Column_name,T2.Column_name
From Table1 T1 Join Table2 T2
On(T1.Column_name=T2.Column_name)

Self Join

- Self Join is used when you apply join inside the same table.
- For if there are multiple columns in single table and we want data which is depends on one column from same table at that time we use Self Join.

Syntax:- Syntax is same as the on clause difference is here we use same table to perform join.

```
Select T1.Column_name,T1.Column_name
From Table1 T1 Join Table1 T1
On(T1.Column_name=T1.Column_name)
```

Nonequijoins

A nonequijoin is a join condition containing something other than an equality operator.

Syntax:- Select T1.Column_name,T2.Column_name
From Table1 T1 Join Table2 T2
On(T1.Column_nam Between T2.Column_name
And T2.Column_name);

Outer Join

If a row does not satisfy a join condition, the row does not appear in the query result.

Outer Join Returns the records with no direct match.

LEFT OUTER JOIN

Left outer join is used to retrieve all the information from left side table and match rows from right side table.

Syntax:- Select T1.Column_name,T2.Column_name
From Table1 T1 Left Outer Join Table2 T2
On(T1.Column_name=T2.Column_name)

Right Outer Join

Right outer join is used to retrieve all the information from Right side table and match rows from left side table.

Syntax:- Select T1.Column_name,T2.Column_name
From Table1 T1 Right Outer Join Table2 T2
On(T1.Column_name=T2.Column_name)

Full Outer Join

Full outer join is used to retrieve all the information from Right side table and left side table even there is unmatched rows present.

Syntax:- Select T1.Column_name,T2.Column_name
From Table1 T1 Full Outer Join Table2 T2
On(T1.Column_name=T2.Column_name)

Cartesian Products

A Cartesian product is formed when:

- A join condition is omitted.
- A join condition is invalid.
- All rows in the first table are joined to all rows in the second table.
- Always include a valid join condition if you want to avoid a Cartesian product.

The CROSS JOIN clause produces the cross-product of two tables.

Syntax:-

```
SELECT Column_name,Column_name
FROM Table_name
CROSS JOIN Table_name ;
```

Summery

In this lesson, We have learned how to use joins to display data from multiple tables by using:

- Equijoins
- Nonequijoins
- OUTER joins
- Self-joins
- Cross joins
- Natural joins
- Full (or two-sided) OUTER joins