

The JOINS command in SQL

Typically, the JOINS statement in SQL is used to connect information in different tables; the JOINS condition is part of an SQL query to retrieve rows from two or more tables.

The JOINS condition in SQL is used in SELECT, UPDATE, and DELETE statements with **the WHERE clause in SQL** .



The JOINS command in SQL

The syntax for the JOINS command in SQL to link two tables is:

SELECT col1, col2, col3.

FROM table_name1, table_name2

WHERE table_name1.col2 = table_name2.col1;

If the JOINS condition in SQL is omitted or is invalid, it will return the Cartesian product of the set of records from two or more connected tables. The Cartesian product returns the row number equivalent to the product of all rows in the connected tables.

For example, if the first table has 20 rows and the second table has 10 rows, the result would be $20 * 10$, which is equivalent to 200 rows. This query takes a long time to execute.

Example of the JOINS command in SQL

Use SQL JOINS to connect the two tables below:

Product database table :

product_id	product_name	supplier_name	unit_price
100	Camera	Nikon	300
101	Television	Onida	100
102	Refrigerator	Vediocon	150
103	Ipod	Apple	75
104	Mobile	Nokia	50

The order_items database table :

order_id	product_id	total_units	customer
5100	104	30	Infosys
5101	102	5	Satyam
5102	103	25	Wipro
5103	101	10	TCS

SQL JOINS are divided into two types: **Equi join** and **Non-Equi join** .

SQL Equi join

The condition of the JOINS statement in SQL uses the equals sign (=) as the comparison operator. Equi joins are divided into two types: **SQL Outer join** and **SQL Inner join** .

For example, you can retrieve information about customers who purchased products and the quantity purchased.

SQL Non-equivalent join

This condition uses other comparison operators, such as >, <, >=, <=

1. Learn about SQL Equi Join

An equi-join is classified into two types:

- **SQL Inner Join**
- **SQL Outer Join**

1.1. SQL Inner Join

All rows returned by the SQL query satisfy the specified SQL connection condition.

Example of SQL Inner Join:

If you want to display product information for each order, the query would look like this. Since you are retrieving data from two tables, you need to identify the common column between the two tables, which is product_id.

The SQL query has the following form:

```
SELECT order_id, product_name, unit_price, supplier_name, total_units  
FROM product, order_items  
WHERE order_items.product_id = product.product_id;
```

The columns must be referenced by the table name in the Join condition, because product_id is a column present in both tables and needs to be identified in the same way. This is to avoid using columns in the SELECT statement in SQL.

The number of the Join condition is (n-1) if there are more than 2 tables joined in a query, where n is the number of related tables. The rule must be true to avoid Cartesian product.

Alternatively, you can also use aliases to reference column names; the query will look like this:

```
SELECT o.order_id, p.product_name, p.unit_price, p.supplier_name, o.total_units  
FROM product p, order_items o  
WHERE o.product_id = p.product_id;
```

1.2. SQL Outer Join

The Join condition returns all rows from both tables that meet the join condition, along with rows that do not meet the condition from one of the tables. The SQL Outer Join operator in Oracle is (+) and is only used on one side of the Join condition.

Different RDBMS implementation syntaxes vary. Very few of these syntaxes represent join conditions such as "sql left outer join", "sql right outer join".

If you want to display all data in the product table along with the data in the order items table, with null values ?? displayed for orders if a product is not in stock, the SQL Outer Join query would look like this:

```
SELECT p.product_id, p.product_name, o.order_id, o.total_units  
FROM order_items o, product p  
WHERE o.product_id (+) = p.product_id;
```

Note: If the (+) operator is used on the left side of the Join condition, it is equivalent to a left outer join. If used on the right side of the Join condition, it is equivalent to a right outer join.

SQL Self Join

SQL Self Join is a type of JOIN in SQL, used to join two separate tables, especially when the table has a FOREIGN KEY referencing a PRIMARY KEY. It is essential to ensure that the JOIN statement specifies aliases

for both copies of the table.

The following is an example of a SQL Self Join query:

```
SELECT a.sales_person_id, a.name, a.manager_id, b.sales_person_id, b.name  
FROM sales_person a, sales_person b  
WHERE a.manager_id = b.sales_person_id;
```

2. SQL Non-Equivalent Join

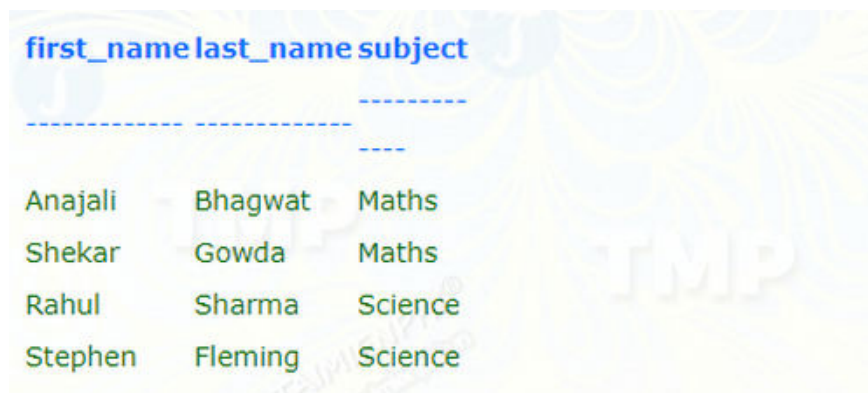
Non-Equivalent Join is a conditional SQL JOIN statement that uses all comparison operators except the equals operator (=). Examples include >=, =, >, <, <=.

Example of SQL Non-Equivalent Join:

If you want to find the names of students who are not enrolled in economics, the SQL query would look like this (using the student information table that TipsMake has introduced in previous articles):

```
SELECT first_name, last_name, subject  
FROM student_details  
WHERE subject != 'Economics'
```

The output will look like this:



first_name	last_name	subject
Anajali	Bhagwat	Maths
Shekar	Gowda	Maths
Rahul	Sharma	Science
Stephen	Fleming	Science

Above is detailed information and examples of the JOINS command in SQL. Basically, the JOINS command in SQL is used to connect information in different tables. Additionally, you can also learn more about **the UPDATE command in SQL** here. If you have any questions or need clarification, please leave a comment in the section below the article.

You finished reading the article "**The JOINS command in SQL**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
