**James Bailey Module9Discussion Left/Right Joins(Due July 7, 2022)**

Left [Outer] joins and Right [Outer] joins.

<https://www.javatpoint.com/mysql-left-join>

Left [Outer] Join, return all the records from the first (left-side) table, if no matching records are found from the second (right side) table it returns null.

The Left Join clause can also be called a Left Outer Join clause. So, Outer is an optional keyword to use with Left Join.

1. **SELECT** columns
2. **FROM** table1
3. LEFT [OUTER] JOIN table2
4. **ON** Join\_Condition;

Sample tables:

**Table: customers**

Graphical user interface, application, table

Description automatically generated

**Table: orders**

Graphical user interface, application, table

Description automatically generated

Select records from both tables.

1. **SELECT** customers.customer\_id, cust\_name, price, **date**
2. **FROM** customers
3. LEFT JOIN orders **ON** customers.customer\_id = orders.customer\_id;

Text, table

Description automatically generated

The table customers and orders have the same column name, which is customer\_id. In that case, MySQL Left Join can also be used with the USING clause to access the records.

1. **SELECT** customer\_id, cust\_name, occupation, price, **date**
2. **FROM** customers
3. LEFT JOIN orders USING(customer\_id);

A screenshot of a computer

Description automatically generated

1. **SELECT** customers.customer\_id, cust\_name, qualification, price, **date**
2. **FROM** customers
3. LEFT JOIN orders **ON** customers.customer\_id = orders.customer\_id
4. **GROUP** **BY** price;

Table

Description automatically generated

<https://www.javatpoint.com/mysql-right-join>

**Right [Outer] join**

Starting tables:

**Table: customers**

Graphical user interface, application, table

Description automatically generated

**Table: orders**

Graphical user interface, application, table

Description automatically generated

1. **SELECT** customers.customer\_id, cust\_name, price, **date**
2. **FROM** customers
3. RIGHT JOIN orders **ON** customers.customer\_id = orders.customer\_id
4. **ORDER** **BY**  customer\_id;

OR,

1. **SELECT** customers.customer\_id, cust\_name, price, **date**
2. **FROM** customers
3. RIGHT JOIN orders USING(customer\_id)
4. **ORDER** **BY**  customer\_id;

Graphical user interface, table

Description automatically generated

1. **SELECT** \* **FROM** customers
2. RIGHT JOIN orders USING(customer\_id)
3. **WHERE** price>2500 AND price<5000;

Table

Description automatically generated

Created new table

Table

Description automatically generated

1. **SELECT** customers.customer\_id, cust\_name, order\_id, price, cellphone
2. **FROM** customers
3. RIGHT JOIN contacts **ON** customer\_id = contact\_id
4. RIGHT JOIN orders **ON** customers.customer\_id = orders.customer\_id **ORDER** **BY** order\_id;

Graphical user interface, table

Description automatically generated

The Right Join clause is also useful in such a case when we want to get records in the table that does not contain any matching rows of data from another table.

1. **SELECT** customer\_id, cust\_name, cellphone, homephone
2. **FROM** customers
3. RIGHT JOIN contacts **ON** customer\_id = contact\_id
4. **WHERE** cellphone **IS** NULL
5. **ORDER** **BY** cellphone;

Table, Teams

Description automatically generated with medium confidence

**Works Cited:**

<https://www.javatpoint.com/mysql-right-join>

<https://www.javatpoint.com/mysql-left-join>