MySQLTUTORIAL

MySQL RIGHT JOIN

Summary: in this tutorial, you will learn how to use the MySQL `RIGHT JOIN` to query data from two tables.

Introduction to MySQL RIGHT JOIN clause

MySQL `RIGHT JOIN` is similar to `LEFT JOIN (https://www.mysqltutorial.org/mysql-left-join.aspx)` , except that the treatment of the joined tables is reversed.

Here's the syntax of the `RIGHT JOIN` of two tables `t1` and `t2` :

```
SELECT
    select_list
FROM t1
RIGHT JOIN t2 ON
    join_condition;
```

In this syntax:

- The `t1` is the left table and `t2` is the right table.
- The `join_condition` specifies the rule for matching rows from both tables.

If the `join_condition` uses the equal operator (`=`) and the joined columns of both tables have the same name, and you can use the `USING` syntax like this:

```
SELECT
    select_list
FROM t1
RIGHT JOIN t2 USING(column_name);
```

Therefore, the following join conditions are equivalent:

```
ON t1.column_name = t2.column_name
```

and

```
USING (column_name);
```

How the `RIGHT JOIN` works.

The `RIGHT JOIN` starts selecting data from the right table (`t2`). It matches each row from the right table with every row from the left table. If both rows cause the join condition to evaluate to `TRUE` , the `RIGHT JOIN` combines columns of these rows into a new row and includes this new row in the result set.

If a row from the right table does not have a matching row from the left table, the `RIGHT JOIN` combines columns of rows from the right table with `NULL` values for all columns of the right table into a new row and includes this row in the result set.

In other words, the `RIGHT JOIN` returns all rows from the right table regardless of having matching rows from the left table or not.

It's important to emphasize that `RIGHT JOIN` and `LEFT JOIN` clauses are functionally equivalent, and they can replace each other as long as the table order is reversed.

Notice that the `RIGHT OUTER JOIN` is a synonym for `RIGHT JOIN` . Therefore, you can use them interchangeably.

MySQL RIGHT JOIN clause examples

We'll use the tables `employees` and `customers` from the sample database (https://www.mysqltutorial.org/mysql-sample-database.aspx) for the demonstration:

The column `salesRepEmployeeNumber` in the table `customers` links to the column `employeeNumber` in the `employees` table.

A sales representative, or an employee, may be in charge of zero or more customers. And each customer is taken care of by zero or one sales representative.

If the value in the column `salesRepEmployeeNumber` is NULL, which means the customer does not have any sales representative.

1) Simple MySQL RIGHT JOIN example

This statement uses the `RIGHT JOIN` clause join the table `customers` with the table `employees`.

```
SELECT
    employeeNumber,
    customerNumber
FROM
    customers
RIGHT JOIN employees
    ON salesRepEmployeeNumber = employeeNumber
ORDER BY
        employeeNumber;
```

In this example:

- The `RIGHT JOIN` returns all rows from the table `employees` whether rows in the table `employees` have matching values in the column `salesRepEmployeeNumber` of the table `customers`.
- If a row from the table `employees` has no matching row from the table `customers`, the `RIGHT JOIN` uses `NULL` for the `customerNumber` column.

2) Using MySQL RIGHT JOIN to find unmatching rows

The following statement uses the `RIGHT JOIN` clause to find employees who do not in charge of any customers:

```
SELECT
    employeeNumber,
    customerNumber
FROM
    customers
RIGHT JOIN employees ON
        salesRepEmployeeNumber = employeeNumber
WHERE customerNumber is NULL
ORDER BY employeeNumber;
```

Summary

- MySQL `RIGHT JOIN` allows you to query data from two or more related tables.
- The `RIGHT JOIN` starts selecting rows from the right table. It always returns rows from the right table whether or not there's match rows in the left table.
- The `RIGHT OUTER JOIN` is the synonym of the `RIGHT JOIN`.