MySQLTUTORIAL

MySQL Self Join

Summary: in this tutorial, you will learn how to use **MySQL self join** that joins a table to itself using the inner join or left join.

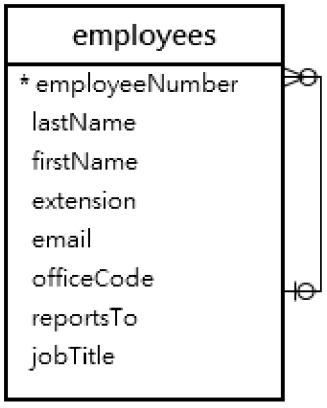
In the previous tutorials, you have learned how to join a table to the other tables using `INNER JOIN` (https://www.mysqltutorial.org/mysql-inner-join.aspx), `LEFT JOIN` (https://www.mysqltutorial.org/mysql-left-join.aspx), `RIGHT JOIN` (https://www.mysqltutorial.org/mysql-right-join/), or `CROSS JOIN` (https://www.mysqltutorial.org/mysql-cross-join/) clause. However, there is a special case that you need to join a table to itself, which is known as a self join.

The self join is often used to query hierarchical data or to compare a row with other rows within the same table.

To perform a self join, you must use table aliases (https://www.mysqltutorial.org/mysql-alias/) to not repeat the same table name twice in a single query. Note that referencing a table twice or more in a query without using table aliases will cause an error.

MySQL self join examples

Let's take a look at the `employees` table in the sample database (https://www.mysqltutorial.org/mysql-sample-database.aspx).



The table `employees` stores not only employees data but also the organization structure data. The `reportsto` column is used to determine the manager id of an employee.

1) MySQL self join using INNER JOIN clause

To get the whole organization structure, you can join the `employees` table to itself using the `employeeNumber` and `reportsTo` columns. The table `employees` has two roles: one is the *Manager* and the other is *Direct Reports*.

```
SELECT
    CONCAT(m.lastName, ', ', m.firstName) AS Manager,
    CONCAT(e.lastName, ', ', e.firstName) AS 'Direct report'
FROM
    employees e
INNER JOIN employees m ON
    m.employeeNumber = e.reportsTo
ORDER BY
    Manager;
```

Try It Out ›

The output only shows the employees who have a manager. However, you don't see the President because his name is filtered out due to the `INNER JOIN` clause.

2) MySQL self join using LEFT JOIN clause

The President is the employee who does not have any manager or value in the `reportsTo` column is `NULL` .

The following statement uses the `LEFT JOIN` clause instead of `INNER JOIN` to include the President:

```
SELECT
    IFNULL(CONCAT(m.lastname, ', ', m.firstname),
            'Top Manager') AS 'Manager',
    CONCAT(e.lastname, ', ', e.firstname) AS 'Direct report'
FROM
    employees e
LEFT JOIN employees m ON
    m.employeeNumber = e.reportsto
ORDER BY
    manager DESC;
```

Try It Out ›

3) Using MySQL self join to compare successive rows

By using the MySQL self join, you can display a list of customers who locate in the same city by joining the `customers` table to itself.

```
SELECT
    c1.city,
    c1.customerName,
    c2.customerName
FROM
    customers c1
```

```
INNER JOIN customers c2 ON
    c1.city = c2.city
    AND c1.customername > c2.customerName
ORDER BY
    c1.city;
```

Try It Out ›

In this example, the table `customers` is joined to itself using the following join conditions:

- `c1.city = c2.city` makes sure that both customers have the same city.
- `c.customerName > c2.customerName` ensures that no same customer is included.

In this tutorial, you have learned how to the MySQL self join that to join a table to itself using the `INNER JOIN` or `LEFT JOIN` clauses.