SQL Standard

SQL “Using” Query

7 months ago • by John Otieno

I can bet anything that if you have ever worked with a relational database, you have probably come across the term SQL JOINS. Even if you haven’t worked with JOINS in particular, you might have heard of the concept.

JOINS are extremely powerful and very common in relational databases since that is a major foundation of SQL databases.

One of the common and useful clauses that work well with SQL JOINS is the USING keyword. We combine this keyword in conjunction with the JOIN statement to specify the columns that we should use to join two or more tables.

You can think of it as a feature that provides a more concise and readable way of performing joins instead of using the typical ON clause, especially when we are dealing with similarly named columns from both tables.

With that out of the way, join us in this post as we explore what, how, and why we use the USING clause in the context of SQL join.

Requirements:

Before we dive into the technicalities and actual implementation, we need to talk about requirements.

For this one, we only need a database that can provide with a good dataset and structure to perform joins. For our case, we choose the Sakila sample database.

SQL USING Clause

As we mentioned, the USING clause in SQL is a feature that provides us with a more readable and concise way of joining the tables when we have two columns with similar names from the tables.

Using the USING clause, instead of specifying the join conditions using the ON clause and explicitly setting the table and column names, we can indicate which columns from the involved tables are part of the JOIN.

Syntax:

The following code snippet demonstrates the syntax of the USING clause in SQL:

```
SELECT column1, column2, ...
FROM table1
JOIN table2
USING (column_name);
```

In the given syntax, we start with a SELECT statement which allows us to specify the columns that we wish to retrieve from the resulting JOIN.

Next, we specify the first table that we wish to join followed by the second table that we wish to join.

Lastly, we use the USING clause to specify the column(s) with identical names in both tables that we wish to use for the JOIN.

Example 1: INNER JOIN Using the “USING” Clause

To better understand how this clause works, let us look at an example of an INNER JOIN that utilizes it.

Let us take the Sakila sample database. Suppose we want to retrieve a list of customers along with their corresponding rental details.

We can utilize an INNER JOIN based on the “customer_id” column which should exists in both the customer and rental tables.

An example query is as follows:

```
SELECT
    customer.customer_id,
    first_name,
    last_name,
    rental_date
FROM
    customer
JOIN rental
        USING (customer_id);
```

In the given query, we select the “customer_id”, “first_name”, “last_name”, and “rental_date” columns from the tables.

We then specify the JOIN condition using the USING (customer_id) clause to indicate that we want to use the “customer_id” that is common in both tables.

But what does the previous query without the USING clause looks like? Take a look at the following example:

```
SELECT
    customer.customer_id,
    first_name,
    last_name,
    rental_date
FROM
    customer
JOIN rental
    ON customer.customer_id = rental.customer_id;
```

As you will notice, we need to explicitly specify the JOIN condition in this case.

Example 2: Self-Join Using the “USING” Clause

Yes, we can use it even in a self-join. Take for example a case where we need to find all the employees and respective managers.

We can use the self-join on the “employees” table based on the “manager_id” column as the JOIN condition.

```
SELECT
    e.employee_id,
    e.first_name AS employee_first_name,
    e.last_name AS employee_last_name,
    m.first_name AS manager_first_name,
    m.last_name AS manager_last_name
FROM
    employees e
LEFT JOIN employees m
        USING (manager_id);
```

In this case, we perform a self-join on the “employees” table by setting two aliases for it. One is “e” for the employee and “m” for the managers.

We then use the USING (manager) to join the employees with the corresponding manager based on that column.

Conclusion

In this tutorial, we learned about the USING clause in SQL that allows us to simplify and make the SQL joins more readable and concise.

ABOUT THE AUTHOR

John Otieno

My name is John and am a fellow geek like you. I am passionate about all things computers from Hardware, Operating systems to Programming. My dream is to share my knowledge with the world and help out fellow geeks. Follow my content by subscribing to LinuxHint mailing list

View all posts

RELATED LINUX HINT POSTS

- Add a Column to the Table in SQL
- Compare Two Tables in SQL
- Combine Two Columns in SQL
- SQL Having Clause
- SQL Subtract
- SQL Select All Except
- SQL Outer Join

Linux Hint LLC, editor@linuxhint.com
1210 Kelly Park Circle, Morgan Hill, CA 95037