

JOINS en MySQL para consultas de bases de datos relacionales

Federico León Gonçalves Pereira

Tecnólogo en Análisis y Desarrollo de Sistemas

Universidade Federal Sul-riograndense – Universidad Tecnológica del Uruguay

Campus Santana do Livramento

Santana do Livramento – RS – Brasil

`federico.goncalvez@estudiantes.utec.edu.uy`

Abstract. *In this article, we will discuss how to combine tables in relational databases in MySQL using INNER JOIN, RIGHT JOIN, and LEFT JOIN. As we know, in SQL we have multiple tables that store related information in a database, and these JOINS allow us to efficiently combine this information in a single query. We will see how these three types of JOINS work and how they can be used to perform more complex queries.*

Resumen. En este artículo, vamos a hablar sobre cómo combinar tablas de bases de datos relacionales en MySQL utilizando INNER JOIN, RIGHT JOIN y LEFT JOIN. Como sabemos, en SQL tenemos varias tablas que almacenan información relacionada entre sí en una base de datos y estos JOINS nos permiten combinar esa información de manera eficiente en una sola consulta.

1. Introducción

En bases de datos relacionales para evitar la redundancia es común tener varias tablas que almacenan información que está relacionada entre sí. Por ejemplo, una base de datos de una tienda en línea podría tener una tabla para productos, una tabla para clientes y una tabla para órdenes. Como cabe esperar podemos relacionar estas tablas con identificadores específicos para obtener información relevante. En MySQL, podemos usar INNER JOIN, RIGHT JOIN y LEFT JOIN para juntar estas tablas de manera eficiente en una sola consulta. En este artículo, vamos a ver cómo funcionan estos tres tipos de JOINS.

Hay otros tipos de JOINS menos comunes, como FULL JOIN y CROSS JOIN, pero INNER JOIN, LEFT JOIN y RIGHT JOIN son los más utilizados en la práctica.

2. ¿Qué son los JOINS?

Los JOINS son operadores utilizados en SQL para combinar tablas de bases de datos relacionales. Hay varios tipos de JOINS, cada uno con un propósito y comportamiento diferentes, estos son a los que nos vamos a referir en el artículo y vamos a dar un ejemplo de cada uno:

INNER JOIN: combina dos tablas y solo muestra filas que coinciden en ambas.

LEFT JOIN: combina dos tablas y muestra todas las filas de la tabla de la izquierda, incluso si no hay coincidencias en la tabla de la derecha.

RIGHT JOIN: combina dos tablas y muestra todas las filas de la tabla de la derecha, incluso si no hay coincidencias en la tabla de la izquierda.

2.1. Ejemplos en MySQL

INNER JOIN combina dos tablas y muestra solo filas que coinciden en ambas. En este ejemplo, se están seleccionando el nombre del cliente y el total de la orden y filtrando solo las órdenes realizadas por clientes existentes:

```
SELECT c.name, o.total  
FROM customers c  
INNER JOIN orders o  
ON c.id = o.customer_id
```

LEFT JOIN: combina dos tablas y muestra todas las filas de la tabla de la izquierda, incluso si no hay coincidencias en la tabla de la derecha. En este ejemplo estamos mostrando todos los clientes, incluso si no han realizado ninguna orden:

```
SELECT c.name, o.total  
FROM customers c  
LEFT JOIN orders o  
ON c.id = o.customer_id
```

RIGHT JOIN combina dos tablas y muestra todas las filas de la tabla de la derecha, incluso si no hay coincidencias en la tabla de la izquierda. En este ejemplo, se muestran todas las órdenes, incluso si no están asociadas a ningún cliente existente:

```
SELECT c.name, o.total  
FROM customers c  
RIGHT JOIN orders o  
ON c.id = o.customer_id
```

3. Conclusión

Como vimos los tipos más comunes de cláusulas JOINS son INNER JOIN, LEFT JOIN y RIGHT JOIN. INNER JOIN nos sirve para obtener información que compartan un identificador común entre ambas tablas, LEFT JOIN muestra todas las filas de la tabla de la izquierda, aunque no haya coincidencias en la tabla de la derecha, y RIGHT JOIN muestra todas las filas de la tabla de la derecha, incluso si no hay coincidencias en la tabla de la izquierda, también se pueden combinar entre sí en una misma consulta. Utilizar estos JOINS de manera adecuada puede ayudar a realizar consultas más complejas y obtener información más relevante al mismo tiempo que ahorramos costes computacionales.

4. Referencias

MySQL. (2022). JOIN Clause. Recuperado de <https://dev.mysql.com/doc/refman/8.0/en/join.html> (accesado el 3/11/2022).