

## **JOINS (inner, left, right)**

**Kaiane Gabriela Cardozo Rodrigues**

Tecnólogo en Análisis y Desarrollo de Sistemas – Universidad IFSul Sant'Ana do  
Livramento.

Cardozo.kaiane12@gmail.com /  
kaianerodrigues.sl008@academico.ifsul.edu.br

***Resumen.*** Investigación sobre los tres tipos de join: inner join, left join y right join.

## 1. Introducción

Uno de los factores más importantes en cualquier base de datos relacional es la capacidad de relacionar dos o más tablas cualesquiera para que el usuario logre publicar datos de una tabla con otra tabla. Este factor son las combinaciones, es muy útil cuando se necesita consultar datos que están en más de una tabla y además de ello también se desea recuperarlos. Los JOIN son una operación que puede hacer coincidir filas de una tabla con otras de manera que las columnas de cada tabla son colocadas lado a lado, obteniendo como resultado visual una sola tabla. Por ello, en la presente investigación se estarán abordando puntos importantes, como concepto y sintaxis, acerca de varios tipos de JOIN. En concreto, tres tipos de JOIN: INNER JOIN, LEFT JOIN, RIGHT JOIN.

## 2. Desarrollo.

### 2.1. Inner Join.

La INNER JOIN es la operación más común entre los tipos de JOIN. Aunque, al crear el comando, se escribe "INNER JOIN", también es posible utilizarlo sin la necesidad de escribir el "INNER JOIN". Si solamente se utiliza la palabra "JOIN" por sí sola, el programa asume que es una INNER JOIN. Este JOIN una operación condición, es decir, devuelve solamente aquellas filas que coinciden con la condición definidas en la instrucción del SELECT. Las columnas especificadas en el SELECT no tienen que tener necesariamente el mismo nombre. Su sintaxis es la siguiente.

```
SELECT columna1, columna2, columna3... FROM tabla1 INNER JOIN tabla2 ON  
tabla1.coulumna_ID= tabla2.columna_ID;
```

- Explicación de la sintaxis.

Luego del SELECT se deben especificar las columnas que necesitan ser devueltas al usuario. Se le añade la clausula FROM y seguido, el nombre de la primera tabla con la que se desea realizar la operación, el orden de las tablas dependerá del problema ha resolver. Agrega el INNER JOIN, el nombre de la segunda tabla y finalmente la sentencia ON, la cual se encargará de relacionar las dos tablas. Como puede ver en la sintaxis: "tabla1.coulumna\_ID= tabla2.columna\_ID". Se recomienda relacionar las dos tablas con la columna a la cual pertenece el ID de cada tabla. Esto para facilitar la operación.

### 2.2. Outer join.

Antes de profundizar sobre LEFT JOIN y RIGHT JOIN, es necesario aclarar que estas dos operaciones provienen de otra, OUTER JOIN. La cual es una operación que puede devolver, como resultado de un comando, todas las filas que coinciden y alguna más, o todas las filas que no son coincidentes de las tablas que se requieren relacionar. El resultado dependerá del tipo de OUTER JOIN que se utilice.

Existen tres tipos de OUTER JOIN: LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN.

Como no es el objetivo tratar sobre FULL OUTER JOIN, solamente se lo definirá. Este tipo de OUTER JOIN arroja todas las filas que coinciden, así como también todas las filas que no coinciden. Se podría decir que FULL OUTER JOIN es la combinación de los otros dos tipos de OUTER JOIN.

### **2.3 Left Join.**

LEFT JOIN es un tipo de OUTER JOIN que, como resultado, devuelve todas las filas coincidentes y todas las filas no coincidentes de la tabla izquierda. Se considera al a tabla izquierda como dominante. Esta devolverá absolutamente todos los datos guardados en la tabla izquierda y todos los datos que se relacionen de la tabla derecha. Cuando hay resultados NULL, significa que hay datos que no se relacionan.

La sintaxis es la siguiente:

```
SELECT columna1, columna2, columna3... FROM tabla1 LEFT JOIN tabla2 ON  
      tabla1.columna_id = tabla2.columna_id;
```

- Explicación de la sintaxis:

La sintaxis es parecida a la de INNER JOIN, los cambios reales se verán reflejados en el resultado de este comando. Para empezar, se escribe la sentencia SELECT, seguido de las columnas que necesitan ser devueltas, luego se agrega la sentencia FROM, el nombre de la tabla1, la cual será la tabla izquierda, por ende, la tabla dominante. A continuación, se escribe LEFT JOIN, nombre de la tabla2 y finaliza con la sentencia ON, la cual es la condición para que las dos tablas sean relacionadas.

### **2.4. Right Join.**

La operación RIGHT JOIN, es similar al LEFT JOIN y es un tipo de OUTER JOIN. Esta operación, a diferencia del LEFT JOIN, que devolvía todos los datos de la tabla izquierda y algunos datos coincidentes de la tabla derecha, funcionará del mismo modo, con la excepción de que, en vez de mostrar todos los datos de la tabla izquierda, el resultado devolverá todos los datos pertenecientes a la tabla derecha y los datos que son coincidentes de la tabla izquierda.

La sintaxis es la siguiente:

```
SELECT columna1, columna2, columna3... FROM tabla1 RIGHT JOIN tabla2 ON  
      tabla1.columna_id = tabla2.columna_id;
```

- Explicación de la sintaxis:

La sintaxis es muy similar a la del LEFT JOIN. Las diferencias reales se verán reflejadas en el resultado al hacer la operación. El comando puede ser escrito del mismo

modo que el LEFT JOIN, con la diferencia de que en vez de “LEFT”, es necesario escribir “RIGHT” antes del JOIN.

### **3. Conclusión.**

Existen muchos tipos de JOIN, profundizar y aprender sobre ellos es muy beneficioso para el futuro. Al indagar sobre los tipos de JOIN, no solo se obtuvo conocimiento, también se obtuvo experiencia, en consecuencia, al momento de trabajar con estos operadores no habrá dificultades en sus usos. El hecho de empezar a averiguar sobre estos operadores puede impulsar al lector, tanto como a mí, a investigar más sobre otros JOIN. Tener conocimiento y conciencia sobre ello, será de utilidad y facilitará el trabajo a la hora de manipular bancos de datos.

### **Referencias.**

Fundamentos de SQL, (2006). Tercera edición. Escrito por Andy Oppel y Robert Sheldon. Traducido por Carlos Fabián Jiménez Castillo.

Artículo: OUTER JOIN: definición, tipos y ejemplos (2019).

SQL – Structured Query Language (Linguagem de consulta estruturada) Introdução. Profesor Silvio Normey Gómez.

### **Enlaces.**

[https://pedrobeltrancanessa-biblioteca.weebly.com/uploads/1/2/4/0/12405072/fundamentos\\_de\\_sql\\_3edi\\_oppel.pdf](https://pedrobeltrancanessa-biblioteca.weebly.com/uploads/1/2/4/0/12405072/fundamentos_de_sql_3edi_oppel.pdf)

<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/sql-outer-join/>

[https://apnp.ifsul.edu.br/pluginfile.php/400415/mod\\_resource/content/1/SQL\\_Introducao.pdf](https://apnp.ifsul.edu.br/pluginfile.php/400415/mod_resource/content/1/SQL_Introducao.pdf)