Contents

[**Tipos de JOIN** 1](#_Toc66969493)

[1. **Inner Join** 1](#_Toc66969494)

[2. **Left Join** 2](#_Toc66969495)

[3. **Right Join** 3](#_Toc66969496)

[4. **Outer Join** 3](#_Toc66969497)

[5. **Left excluding join** 4](#_Toc66969498)

[6. **Right Excluding join** 5](#_Toc66969499)

[7. **Outer excluding join** 5](#_Toc66969500)

[**Algunos comandos** 6](#_Toc66969501)

# **Tipos de JOIN**

Existen diferentes formas en las que se pueden unir las tablas en nuestras consultas y de acuerdo con esta unión se va a mostrar información, y es importante siempre tener clara esta relación. En esta clase te voy a mostrar gráficamente 7 diferentes tipos de uniones que puedes realizar.

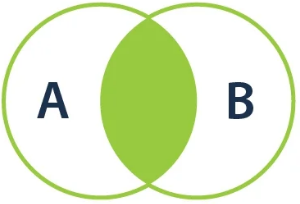
Usar correctamente estas uniones puede reducir el tiempo de ejecución de tus consultas y mejorar el rendimiento de tus aplicaciones.

Como yo lo veo cuando hacemos uniones en las consultas para seleccionar información, estamos trabajando con tablas, estas tablas podemos verlas como conjuntos de información, de forma que podemos asimilar los joins entre tablas como uniones e intersecciones entre conjuntos.

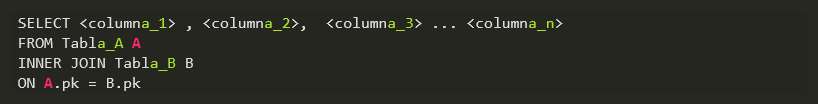
Supongamos que contamos con dos conjuntos, el conjunto A y el conjunto B, o, la tabla A y la tabla B. Sobre estos conjuntos veamos cuál es el resultado si aplicamos diferentes tipos de join.

## **Inner Join**

Esta es la forma más fácil de seleccionar información de diferentes tablas, es tal vez la que más usas a diario en tu trabajo con bases de datos. Esta union retorna **todas las filas de la tabla A** que **coinciden** en la **tabla B**. Es decir, aquellas que están en la **tabla A** Y en la **tabla B**, si lo vemos en conjuntos la intersección entre la tabla A y la B.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.

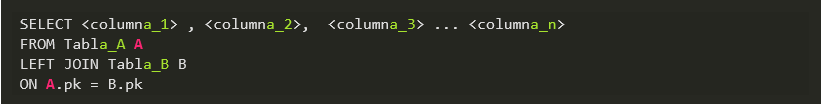


## **Left Join**

Esta consulta retorna **todas las filas** que están en la **tabla A** y además si hay coincidencias de filas en la **tabla B** también va a traer esas filas.

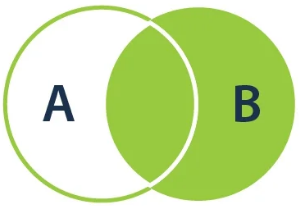


Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.

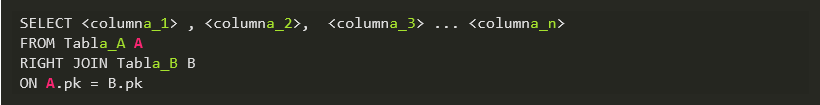


## **Right Join**

Esta consulta retorna todas las filas de la **tabla B** y además si hay filas en la **tabla A** que **coinciden** también va a traer estas filas de la **tabla A**.

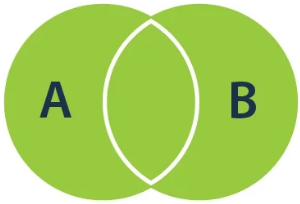


Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.

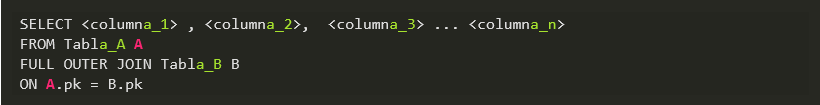


## **Outer Join**

Este join retorna **TODAS** las filas de las **dos tablas**. Hace la union entre las filas que **coinciden** entre la **tabla A** y la **tabla B**.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.

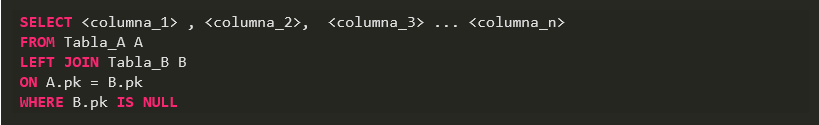


## **Left excluding join**

Esta consulta retorna **todas las filas** de la **tabla de la izquierda**, es decir, la **tabla A** que **no** tienen **ninguna coincidencia** con la tabla de la derecha, es decir la **tabla B**.

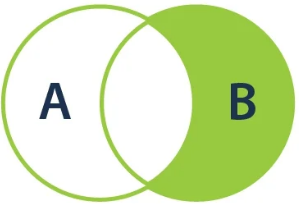


Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.

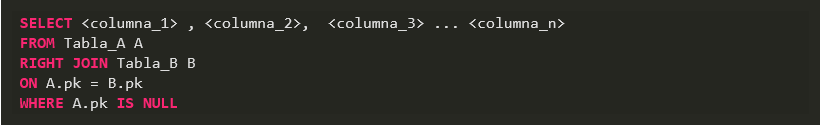


## **Right Excluding join**

Esta consulta retorna **todas las filas** de la tabla de la **derecha**, es decir, la **tabla B** que **no** **tienen** **coincidencias** en la tabla de la izquierda, es decir, la **tabla A**.

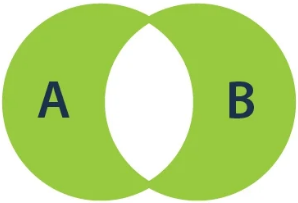


Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.

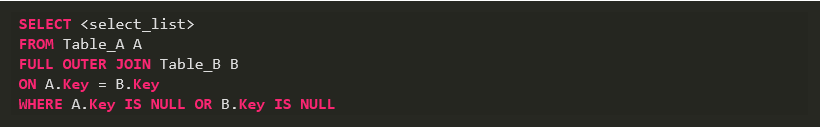


## **Outer excluding join**

Esta consulta retorna **todas las filas** de la tabla de la **izquierda**, **tabla A**, y **todas las filas** de la tabla de la **derecha**, **tabla B** que no coinciden.



Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas.



# **Algunos comandos**

|  |  |
| --- | --- |
| Comandos | Descripción |
| SHOW databases | Muestra las bases de datos existentes. |
| USE database\_name | Selecciona una base de datos específica. |
| SHOW tables | Muestra las tablas de la base de datos. |
| SELECT database() | Muestra el nombre de la base de datos seleccionada. |
| CREATE database database\_name | Crea una nueva base de datos. |
| CREATE DATABASE IF NOT EXISTS database\_name | Crea una base de datos si no existe. |
| SHOW warnings | Muestra las advertencias. |
| DROP table table\_name | Elimina permanentemente una tabla. |
| DESCRIBE table\_name | Nos indica las columnas que tenemos en una tabla. |
| SHOW FULL COLUMNS FROM table\_name | Es parecido al comando **DESCRIBE** pero muestra más datos. |
| INSERT INTO table\_name(columns) VALUES (values) | Inserta datos en una tupla. |
| ON DUPLICATE KEY IGNORE ALL | Ignora las resticciones al insertar una tupla con un valor repetido y que está restringido en una columna con UNIQUE (**Nota**: No es una buena práctica utilizarlo utilizarlo). |
| ON DUPLICATE KEY UPDATE column = VALUES (value) | Al insertar una tupla con un campo duplicado actualiza el valor de un campo específico con un nuevo valor tomado de los datos insertados. |
| SELECT \* FROM table\_name WHERE column\_value = 1\G | En lugar de cerrar la sentencia con **;** se utiliza **\G**, lo cual muestra los datos de una manera más legible. |
| mysql -u root -p < all\_schema.sql | Podemos ejecutar un script SQL inmediatamente después de acceder a la base de datos. |
| mysql -u root -p -D database\_name < all\_schema.sql | Es parecido al anterior sólo que con la bandera **-D** indicamos el nombre de la base de datos sobre la que queremos ejecutar el script. |
| SELECT YEAR(NOW()) | Muestra el año de la fecha actual utilizando las funciones **YEAR()** y **NOW()**. |
| SELECT \* FROM table\_name WHERE column\_value like ‘%value%’ | Nos muestra las tuplas que en un campo específico contengan un valor, el wildcard **%** indica que no nos importa que valor existan **antes** o **después** del dato que especificamos. |
| SELECT COUNT(\*) FROM table\_name | Devuelve el número de tuplas de una tabla. |
| SELECT \* FROM table\_name WHERE column\_value BETWEEN value AND value | Nos devuelve las tuplas que se encuentren en medio de los valores indicados. |
| DELETE FROM table\_name WHERE column\_value = value | Elimina una tupla de una tabla. |
| UPDATE table\_name SET [column\_value = value, …] WHERE column\_value = value | Actualiza una tupla de una tabla. |
| TRUNCATE table\_name | Borra todo el contenido de una tabla. |
| mysqldump -u user -p database\_name > esquema.sql | Guarda el **esquema** de una base de datos con todo y **datos** en un  archivo sql. |
| mysqldump -u user -p -d database\_name | Es parecido al anterior sólo que aquí **no** se guardan los **datos**. |