

Nunca pares de aprender sobre la seguridad de tus datos . Aquí puedes ver nuestros [Términos de Uso](#) y actualizaciones de nuestras [políticas de privacidad](#)



1.145 pts ▾

Menú



Curso de SQL y MySQL

Artículo

Tipos de JOIN

**Alberto Alcocer** ⌚ 2 de Mayo de 2018

En la clase anterior estuvimos hablando de dos tipos de joins que podemos usar cuando estemos trabajando con consultas a nuestras bases de datos.

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.

1. Inner Join



5 casos de negocio

Esta es la forma mas fácil de seleccionar información de diferentes tablas, es tal vez la que mas 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:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
INNER JOIN Tabla_B B
ON A.pk = B.pk
```

2. Left Join

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



5 casos de negocio



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

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>  
FROM Tabla_A A  
LEFT JOIN Tabla_B B  
ON A.pk = B.pk
```

3. 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.



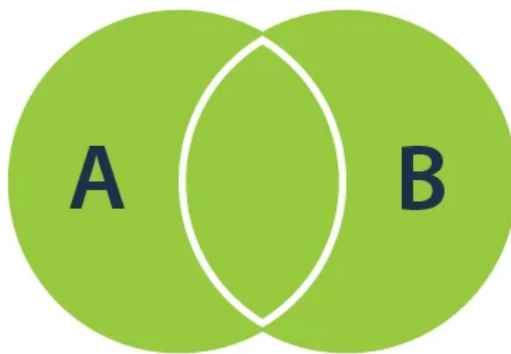
5 casos de negocio

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

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>  
FROM Tabla_A A  
RIGHT JOIN Tabla_B B  
ON A.pk = B.pk
```

4. 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:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>  
FROM Tabla_A A  
FULL OUTER JOIN Tabla_B B  
ON A.pk = B.pk
```

5. Left excluding join



5 casos de negocio

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:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>
FROM Tabla_A A
LEFT JOIN Tabla_B B
ON A.pk = B.pk
WHERE B.pk IS NULL
```

6. 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.



5 casos de negocio

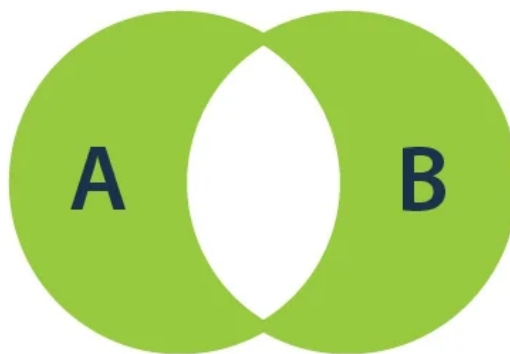


Esto lo podemos implementar de esta forma cuando estemos escribiendo las consultas:

```
SELECT <columna_1> , <columna_2>, <columna_3> ... <columna_n>  
FROM Tabla_A A  
RIGHT JOIN Tabla_B B  
ON A.pk = B.pk  
WHERE A.pk IS NULL
```

7. 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.



5 casos de negocio

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

```
SELECT <select_list>
FROM Table_A A
FULL OUTER JOIN Table_B B
ON A.Key = B.Key
WHERE A.Key IS NULL OR B.Key IS NULL
```



Escribe aquí tu pregunta

+2



jahirmorenoa Estudiante • hace 3 días

Oh perfecto, justo el tipo de "join" que buscaba, "outer excluding join" = "diferencia simetrica". 😊



2



lenielluzardo Estudiante • hace 5 días

Excelente explicacion, con esto y el curso de Fundamentos de Bases de datos queda todo completamente claro.



2



Jsobrino Estudiante • hace 6 días

Alguien podría ayudarme para el **FULL OUTER JOIN** me aparece error de sintaxis al compilar y no logro detectar el error:

```
mysql> select a.author_id, a.name, a.nationality, b.title
-> from authors as a
-> full outer join books as b
-> on a.author_id = b.author_id
-> where a.author_id is null or b.author_id is null;
ERROR 1064 (42000): You have an error in your SQL syntax;
in books as b
on a.author_id = b.author_id
where a.author_id is nul' at line 3
```



1



davidandreszapataflores Estudiante • hace 7 días



5 casos de negocio