

**INSTITUTO TECNOLÓGICO**  
**JOSÉ MARIO MOLINA**  
**PASQUEL Y HENRÍQUEZ**  
**LAGOS DE MORENO**

Administración de Base de Datos

Axel Aarón De Anda Pérez

Pedro Espinosa Esparza

Práctica JOINS

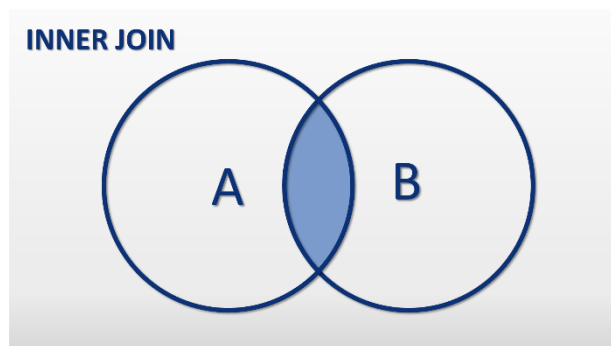
28-02-2020

## Descripción de la Actividad

Esta actividad consiste en crear un reporte acerca de los diferentes tipos de joins y anexar los ejercicios realizados en clase.

### INNER JOIN

El inner join ofrece a los usuarios un conjunto de datos. Este retorna los registros cuando al menos un dato se repite en 2 tablas esto nos ayuda a evitar la ambigüedad. Consiste en combinar cada fila de una tabla con cada fila de la otra tabla, seleccionando aquellas filas que cumplan una determinada condición.



Para esta práctica se creó una base de datos llamada ejemplos join.

```
create database ejemplosjoin;
```

Se necesitaron crear 2 tablas:

Tabla 1

```
create table tabla1(id int, value varchar(10));
```

Datos insertados

```
insert into tabla1(id, value)
select 1, 'First'
union all
select 2, 'Secound'
union all
select 3, 'Third'
union all
select 4, 'Fourth'
union all
select 5, 'Fifth';
```

Tabla 2

```
create table tabla2( id int, value varchar(10));
```

Datos insertados

```
insert into tabla2
(id,value)
values
(1, 'First'),
(2, 'Secound'),
(3, 'Third'),
(6, 'Sixth'),
(7, 'Seventh'),
(8, 'Eighth');
```

Consultas para ejemplo de inner joins

Consulta 1

```
#region join
select * from tabla1, tabla2 where tabla1.id = tabla2.id;
```

Guion Ejecutado

| id | value   | id1 | value1  |
|----|---------|-----|---------|
| 1  | First   | 1   | First   |
| 2  | Secound | 2   | Secou ▾ |
| 3  | Third   | 3   | Third   |

Consulta 2

```
select t1.id, t1.value from tabla1 t1, tabla2 t2
where t1.id = t2.id and t1.value = 'Secound';
```

Guion Ejecutado

| id | value   |
|----|---------|
| 2  | Secound |

### Consulta 3

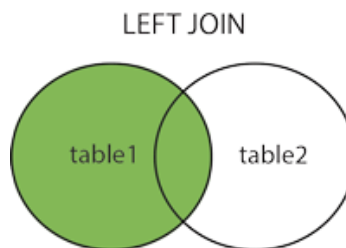
```
select t1.* , t2.* from tabla1 t1 inner join tabla2 t2
on t1.id = t2.id where t1.value = 'Secound';
```

### Guion Ejecutado

| id | value   | id1 | value1  |
|----|---------|-----|---------|
| 2  | Secound | 2   | Secound |

## LEFT JOIN

Esta consulta sirve para añadir a los resultados de la primera tabla, los valores de una segunda tabla. La sentencia LEFT JOIN combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de la primera tabla, incluso aunque no cumplan la condición.



### Consultas para ejemplo de LEFT JOIN

#### Consulta 1

```
#region Left join
select t1.* , t2.* from tabla1 t1
left join tabla2 t2 on t1.id = t2.id;
```

### Guion Ejecutado

| id | value   | id1    | value1  |
|----|---------|--------|---------|
| 1  | First   | 1      | First   |
| 2  | Secound | 2      | Secound |
| 3  | Third   | 3      | Third   |
| 4  | Fourth  | {null} | {null}  |
| 5  | Fifth   | {null} | {null}  |

## Consulta 2

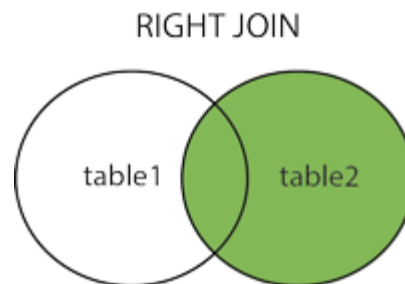
```
select t1.* , t2.* from tabla1 t1 left join  
tabla2 t2 on t1.id = t2.id where t2.id is null;
```

## Guion Ejecutado

| id | value  | id1    | value1 |
|----|--------|--------|--------|
| 4  | Fourth | {null} | {null} |
| 5  | Fifth  | {null} | {null} |

## RIGHT JOIN

Retorna los valores de la tabla derecha más los valores que se cruzan en la tabla izquierda. La sentencia RIGHT JOIN combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de la segunda tabla, incluso aunque no cumplan la condición.



## Consultas para ejemplo de RIGHT JOIN

### Consulta 1

```
select t1.* , t2.* from tabla1 t1 right join  
tabla2 t2 on t1.id = t2.id;
```

## Guion Ejecutado

| id     | value   | id1 | value1   |
|--------|---------|-----|----------|
| 1      | First   | 1   | First    |
| 2      | Secound | 2   | Secound  |
| 3      | Third   | 3   | Third    |
| {null} | {null}  | 6   | Sixth    |
| {null} | {null}  | 7   | Seven! ▾ |
| {null} | {null}  | 8   | Eighth   |

## Consulta 2

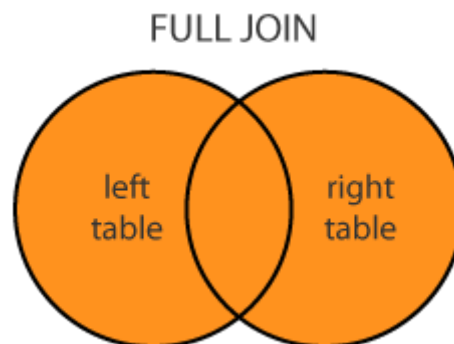
```
select t1.* , t2.* from tabla1 t1 right join  
tabla2 t2 on t1.id = t2.id where t1.id is null;
```

## Guion Ejecutado

| id     | value  | id1 | value1   |
|--------|--------|-----|----------|
| {null} | {null} | 6   | Sixth    |
| {null} | {null} | 7   | Seventh  |
| {null} | {null} | 8   | Eighth ▾ |

## FULL JOIN

La sentencia FULL JOIN combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de las dos tablas, aunque no cumplan la condición. Este combina los registros de las tablas de la izquierda con los de la derecha. Cuando las condiciones no se cumplen retorna valores nulos. MySql no soporta esta sintaxis, por lo tanto se debe hacer un unión de la tabla izquierda con la tabla derecha.



## Consultas para ejemplo de FULL JOIN

### Consulta 1

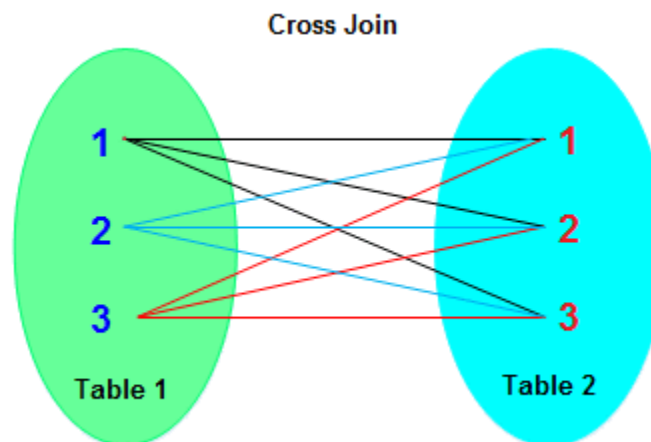
```
select t1.* , t2.* from tabla1 t1 left join  
tabla2 t2 on t1.id = t2.id union  
select t1.* , t2.* from tabla1 t1 right  
join tabla2 t2 on t1.id = t2.id;
```

## Guion Ejecutado

| id     | value   | id1    | value1   |
|--------|---------|--------|----------|
| 1      | First   | 1      | First    |
| 2      | Secound | 2      | Secound  |
| 3      | Third   | 3      | Third    |
| 4      | Fourth  | {null} | {null}   |
| 5      | Fifth   | {null} | {null}   |
| {null} | {null}  | 6      | Sixth    |
| {null} | {null}  | 7      | Seventh  |
| {null} | {null}  | 8      | Eighth ▾ |

## CROSS JOIN

Retorna todos los registros de todas las tablas implicadas en la unión, devuelve el producto cartesiano. No es muy utilizado. Este no necesita una condición para usar el JOIN.



## Consulta para ejemplo de FULL JOIN

```
select t1.*, t2.* from tabla1 t1 cross join tabla2 t2;
```

## Guion Ejecutado

| id | value   | id1 | value1  |
|----|---------|-----|---------|
| 1  | First   | 1   | First   |
| 2  | Secound | 1   | First   |
| 3  | Third   | 1   | First   |
| 4  | Fourth  | 1   | First   |
| 5  | Fifth   | 1   | First   |
| 1  | First   | 2   | Secound |
| 2  | Secound | 2   | Secound |
| 3  | Third   | 2   | Secound |
| 4  | Fourth  | 2   | Secound |
| 5  | Fifth   | 2   | Secound |
| 1  | First   | 3   | Third   |
| 2  | Secound | 3   | Third   |
| 3  | Third   | 3   | Third   |
| 4  | Fourth  | 3   | Third   |
| 5  | Fifth   | 3   | Third   |
| 1  | First   | 6   | Sixth   |
| 2  | Secound | 6   | Sixth   |