

## Combinaciones externas

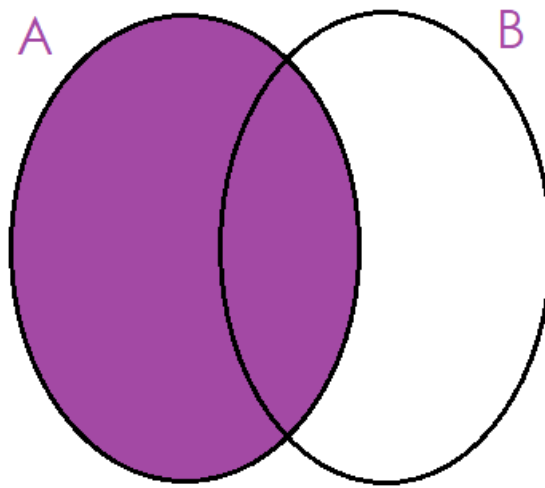
Las combinaciones externas se realizan mediante la instrucción **OUTER JOIN** y nos permiten seleccionar algunas filas de una tabla aunque éstas no tengan correspondencia con las filas de la otra tabla con la que se combina.

### LEFT JOIN

Se utiliza una combinación externa izquierda para mostrar todos los registros de la tabla de la izquierda aunque no tengan correspondencia con la tabla de la derecha.

Si no encuentra coincidencia con la tabla de la derecha, el registro muestra los campos de la segunda tabla a *NULL*

En un **LEFT JOIN** es importante la posición en que se colocan las tablas, ya que la tabla de la izquierda es la que se usa para localizar registros en la tabla de la derecha.



### Ejemplo

# idGrupo	Nombre	idDiscografica	# idDiscografica	Nombre
1	Beethoven	1	1	Clásicos reunidos
			2	Ritmo y poesía
			3	Metal pesado
2	Eminem	2		
3	Mozart	1		
4	Chanel	NULL		

Están fuera de la intersección

```
select *
from grupo left join discografica on
grupo.idDiscografica=discografica.idDiscografica;
```

El resultado del **LEFT JOIN** de estas tablas será:

# idGrupo	Nombre	idDiscografica	idDiscografica	Nombre
1	Beethoven	1	1	Clásicos reunidos
2	Eminem	2	2	Ritmo y poesía
3	Mozart	1	1	Clásicos reunidos
4	Chanel	NULL	NULL	NULL

Donde vemos que están todos los elementos de la tabla de la izquierda, estén o no en la intersección de ambas tablas.

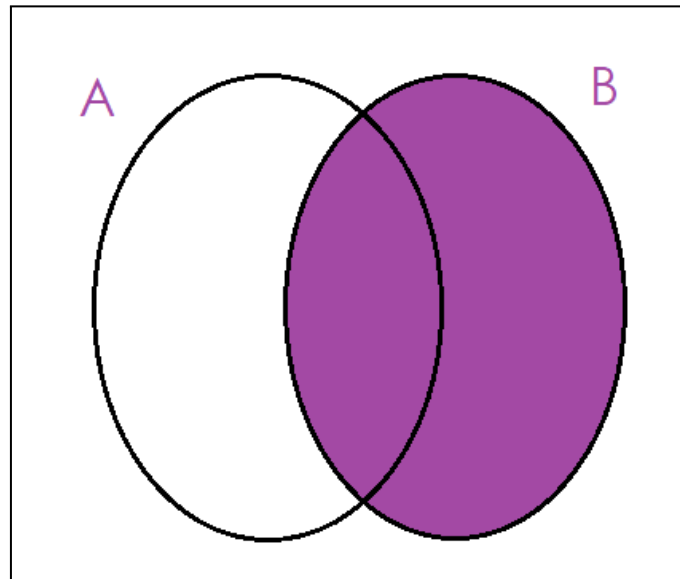
En cambio, si ponemos a la izquierda la tabla *discográfica* el resultado no será el mismo:

```
select *
from discografica left join grupo on
discografica.idDiscografica=grupo.idDiscografica;
```

# idDiscografica	Nombre	idGrupo	Nombre	idDiscografica
1	Clásicos reunidos	1	Beethoven	1
1	Clásicos reunidos	3	Mozart	1
2	Ritmo y poesía	2	Eminem	2
3	Metal pesado	NULL	NULL	NULL

## RIGHT JOIN

De manera análoga, usando **RIGHT JOIN** obtenemos todas las filas de la tabla de la derecha, aunque no tengan correspondencia en la tabla de la izquierda.



### Ejemplo

# idGrupo	Nombre	idDiscografica	# idDiscografica	Nombre
1	Beethoven	1	1	Clásicos reunidos
			2	Ritmo y poesía
			3	Metal pesado
2	Eminem	2		
3	Mozart	1		
4	Chanel	NULL		

Están fuera de la intersección

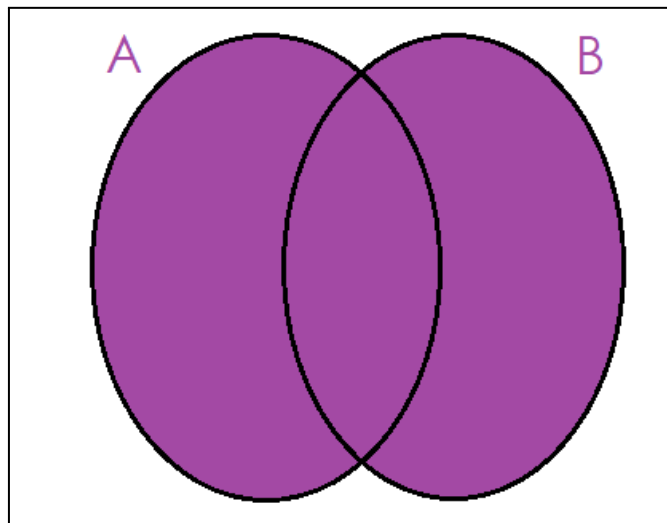
```
select *
from grupo right join discografica on
grupo.idDiscografica=discografica.idDiscografica;
```

El resultado de la consulta es:

# idGrupo	Nombre	idDiscografica	idDiscografica	Nombre
1	Beethoven	1	1	Clásicos reunidos
3	Mozart	1	1	Clásicos reunidos
2	Eminem	2	2	Ritmo y poesía
NULL	NULL	NULL	3	Metal pesado

### FULL OUTER JOIN

Obtenemos todas las filas de ambas tablas, aunque no haya correspondencia entre ellas rellenando con nulos los campos que falten.



#### Ejemplo

Siguiendo con el mismo ejemplo, haríamos

```
select *
from grupo full join discografica on
grupo.idDiscografica=discografica.idDiscografica;
```

```
select *
from grupo full outer join discografica on
grupo.idDiscografica=discografica.idDiscografica;
```

La combinación **FULL OUTER JOIN** no está implementada en MySQL, por tanto para poder simular esta operación utilizaremos el operador **UNION**, que realiza la unión del resultado de dos consultas:

El resultado esperado de una combinación de tipo **FULL OUTER JOIN** es obtener la intersección de las dos tablas junto con las filas de ambas tablas que no se puedan combinar. Dicho de otro modo, el resultado sería el equivalente a realizar la unión de una consulta de tipo **LEFT JOIN** y una consultas de tipo **RIGHT JOIN** sobre las mismas tablas:

```

select *
from grupo right join discografica on
grupo.idDiscografica=discografica.idDiscografica

union

select *
from grupo left join discografica on
grupo.idDiscografica=discografica.idDiscografica;

```

El resultado de esta consulta sería:

# idGrupo	Nombre	idDiscografica	idDiscografica	Nombre
1	Beethoven	1	1	Clásicos reunidos
3	Mozart	1	1	Clásicos reunidos
2	Eminem	2	2	Ritmo y poesía
NULL	NULL	NULL	3	Metal pesado
4	Chanel	NULL	NULL	NULL

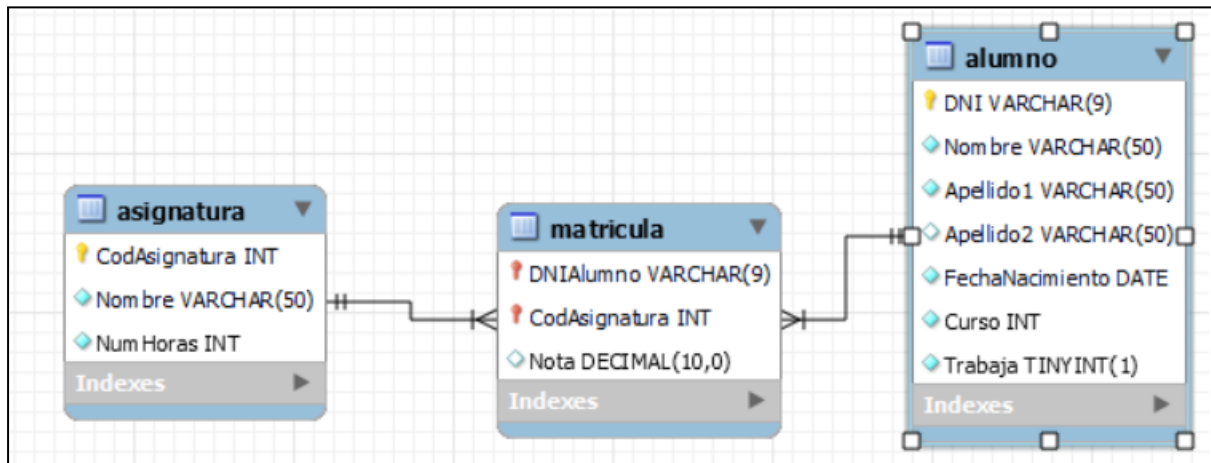
## Uso de alias en las tablas

Por comodidad podemos darles un alias a las tablas:

<pre>select * from alumno inner join casa on alumno.idCasa=casa.idCasa;</pre>	<pre>select * from alumno A inner join casa C on A.idCasa=C.idCasa;</pre> <pre>select * from alumno as A inner join casa as C on A.idCasa=C.idCasa;</pre>
---	---

## Consultas con varias tablas

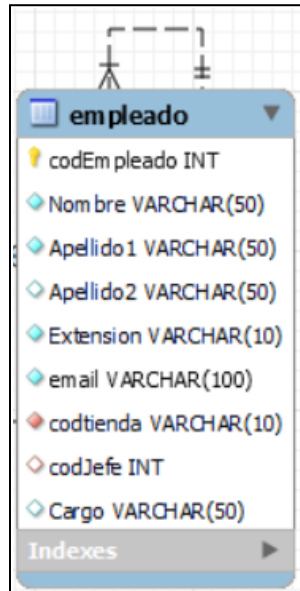
También podemos combinar más de dos tablas para realizar consultas:



```
select *
from alumno AL inner join matricula M on AL.DNI=M.DNIALumno
inner join asignatura A on M.CodAsignatura=A.CodAsignatura;
```

## Combinar una tabla con ella misma

Para poder hacer una operación de **INNER JOIN** sobre la misma tabla es necesario utilizar un alias para la tabla.




```
SELECT empleado.nombre, empleado.apellido1, empleado.apellido2,
jefe.nombre, jefe.apellido1, jefe.apellido2
FROM empleado INNER JOIN empleado AS jefe ON empleado.codigo_jefe =
jefe.codigo_empleado
```

```
SELECT empleado.nombre, empleado.apellido1, empleado.apellido2,
jefe.nombre, jefe.apellido1, jefe.apellido2
FROM empleado INNER JOIN empleado jefe ON empleado.codigo_jefe =
jefe.codigo_empleado
```

## Errores comunes

Cuando usemos LEFT JOIN o RIGHT JOIN no debemos tener varias condiciones en la cláusula ON:

<pre>SELECT * FROM proveedor LEFT JOIN producto ON proveedor.CodProveedor=producto.CodProveedor AND producto.CodProveedor IS NOT NULL ORDER BY proveedor.CodProveedor;</pre>	
<pre>SELECT * FROM proveedor LEFT JOIN producto ON proveedor.CodProveedor=producto.CodProveedor WHERE producto.CodProveedor IS NOT NULL ORDER BY proveedor.CodProveedor;</pre>	