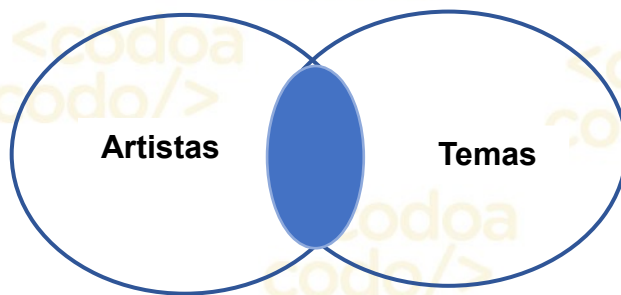


1. INNER JOIN selecciona los registros que tienen valores coincidentes en ambas tablas. Los registros de ambas tablas serán seleccionados siempre que haya **una coincidencia entre las columnas**. Si hay registros en la tabla "Artistas" que no tienen coincidencias en la tabla "Temas", los registros de la tabla "Artistas" no se mostrarán. Es el tipo de JOIN más común.



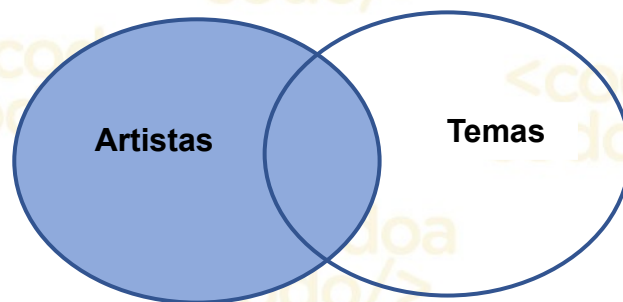
```
SELECT artistas.nombre, artistas.id_artista, temas.id_artista, temas.nombre_tema  
FROM `artistas`  
INNER JOIN `temas`  
ON artistas.id_artista = temas.id_artista;
```

| Nombre | id_artista | id_artista | nombre_tema |
|-----------------|------------|------------|-------------|
| Beyonce | 1 | 1 | Partition |
| Tina Turner | 2 | 2 | The Best |
| Michael Jackson | 3 | 3 | Thriller |

Otra sintaxis para realizar INNER JOIN (JOIN Implícito):

```
SELECT artistas.nombre, artistas.id_artista, temas.id_artista, temas.nombre_tema  
FROM `artistas`, `temas`  
WHERE artistas.id_artista = temas.id_artista;
```

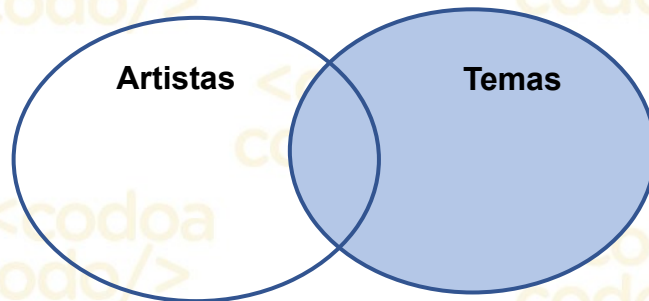
2. LEFT JOIN mantiene **todos los registros de la tabla izquierda** (la tabla “Artistas”). Los registros de la tabla derecha se mostrarán sólo si hay una coincidencia con los de la izquierda. Si existen valores en la tabla izquierda no coincidentes con los de la tabla derecha, estos últimos se mostrarán como **null**.



```
SELECT artistas.nombre, artistas.id_artista, temas.id_artista, temas.nombre_tema  
FROM `artistas`  
LEFT JOIN `temas`  
ON artistas.id_artista = temas.id_artista;
```

| Nombre | id_artista | id_artista | nombre_tema |
|-----------------|------------|------------|-------------|
| Beyonce | 1 | 1 | Partition |
| Tina Turner | 2 | 2 | The Best |
| Michael Jackson | 3 | 3 | Thriller |
| Luis Fonsi | 4 | (null) | (null) |

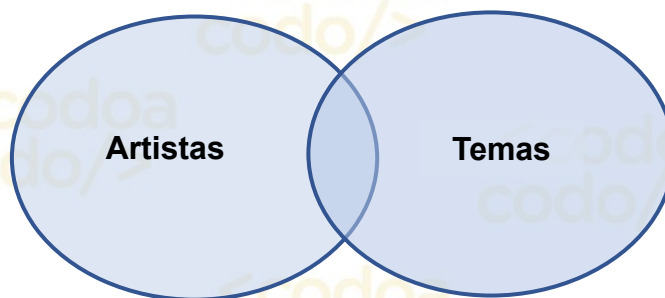
3. RIGHT JOIN. En este caso **se mantienen todos los registros de la tabla derecha** (la tabla “Temas”). Los registros de la tabla izquierda se mostrarán si hay una coincidencia con los de la derecha. Si existen valores en la tabla derecha no coincidentes con los de la tabla izquierda, estos últimos se mostrarán como **null**.



```
SELECT artistas.nombre, artistas.id_tema, temas.id_tema, temas.nombre_tema  
FROM `artistas`  
RIGHT JOIN `temas`  
ON temas.id_tema = artistas.id_tema;
```

| Nombre | id_tema | id_tema | nombre_tema |
|-----------------|---------|---------|-------------|
| Beyonce | 1 | 1 | Partition |
| Tina Turner | 2 | 2 | The Best |
| Michael Jackson | 3 | 3 | Thriller |
| (null) | (null) | 4 | Despacito |

4. FULL JOIN o FULL OUTER JOIN o LEFT JOIN UNION RIGHT JOIN devuelve todas las filas de la tabla izquierda (“Artistas”) y de la tabla derecha (“Temas”). Combina el resultado de los JOINS **LEFT** y **RIGHT**. Aparecerá **null** en cada una de las tablas alternativamente cuando no haya una coincidencia.



```
SELECT artistas.nombre, artistas.id_artista, temas.id_artista, temas.nombre_tema
FROM `artistas` LEFT JOIN `temas` ON artistas.id_artista = temas.id_artista
UNION
SELECT artistas.nombre, artistas.id_artista, temas.id_artista, temas.nombre_tema
FROM `artistas` RIGHT JOIN `temas` ON artistas.id_artista = temas.id_artista;
```

| Nombre | id_artista | id_artista | nombre_tema |
|-----------------|------------|------------|-------------|
| Beyonce | 1 | 1 | Partition |
| Tina Turner | 2 | 2 | The Best |
| Michael Jackson | 3 | 3 | Thriller |
| Luis Fonsi | 4 | 4 | Despacito |

Fuente: https://www.w3schools.com/sql/sql_join_inner.asp
Nicolás Basualdo Cornet - Instructor Codo a Codo 4.0 – Big Data