

TIPOS DE JOIN EN MySQL

En nuestro ejemplo, lo usaremos para ver cómo sería la estructura de la base de datos de los alumnos que se apuntan a cursos. Para ello, creamos una base de datos que se llame Maestralia y dos tablas: una de cursos y otra de alumnos que se relacionan entre sí por la columna idCurso.

Empezamos por crear la base de datos:

```
mysql> CREATE DATABASE maestralia;
```

Seleccionamos la base de datos que vamos a utilizar:

```
mysql> USE maestralia;
```

Creamos dos tablas:

Tabla cursos

```
mysql> CREATE TABLE cursos (  
  
->     idCurso INT AUTO_INCREMENT,  
  
->     curso VARCHAR(100),  
  
->     PRIMARY KEY(idCurso)  
  
-> );
```

Insertamos varios cursos de prueba:

```
mysql> INSERT INTO cursos (curso)  
  
-> VALUES ('Auxiliar de Enfermería'),  
  
-> ('Doblaje'),  
  
-> ('Pastelería');
```

```
SELECT * FROM cursos;
```

```
+-----+-----+  
  
| idCurso | curso                |
```

```

+-----+-----+
|      1 | Auxiliar de Enfermería |
|      2 | Doblaje                  |
|      3 | Pastelería               |
+-----+-----+

```

Tabla alumnos

```

mysql> CREATE TABLE alumnos (

->     idAlumno INT AUTO_INCREMENT,

->     nombre VARCHAR(100),

->     apellidos VARCHAR(100),

->     idCurso INT,

->     FOREIGN KEY (idCurso) REFERENCES cursos(idCurso),

->     PRIMARY KEY(idAlumno)

-> );

```

Insertamos datos de alumnos de prueba:

```

INSERT INTO alumnos (nombre, apellidos, idCurso)

-> VALUES('Pilar', 'Mirás', '2'),('Antonio', 'Cambados', '2'),

-> ('Pedro', 'Fraile', '3'),('Miguel', 'Yañez', '3'),

-> ('Carlos', 'Sanchidrián', NULL), ('Alejandro', 'Rey', NULL);

```

```

mysql> SELECT * FROM alumnos;

```

```

+-----+-----+-----+-----+
| idAlumno | nombre  | apellidos | idCurso |
+-----+-----+-----+-----+
|      7 | Pilar   | Mirás     |      2 |

```

8	Antonio	Cambados	2
9	Pedro	Fraile	3
10	Miguel	Yañez	3
11	Carlos	Sanchidrián	NULL
12	Alejandro	Rey	NULL

```

+-----+-----+-----+-----+

```

Una vez que tenemos los datos cargados, empezamos a revisar las distintas opciones de unión.

INNER JOIN

La cláusula de INNER JOIN une dos tablas en función de una columna común comparando cada fila de la primera tabla con cada fila de la segunda tabla.

Si los valores de ambas filas coinciden, la cláusula de INNER JOIN crea una nueva fila cuya columna contiene todas las columnas de las dos filas de ambas tablas e incluye esta nueva fila en el conjunto de resultados.

La cláusula de INNER JOIN solo incluye filas coincidentes de ambas tablas como se puede ver en el siguiente ejemplo:

```

mysql> SELECT *
      -> FROM alumnos a
      -> INNER JOIN cursos c
      -> ON a.idCurso = c.idCurso;

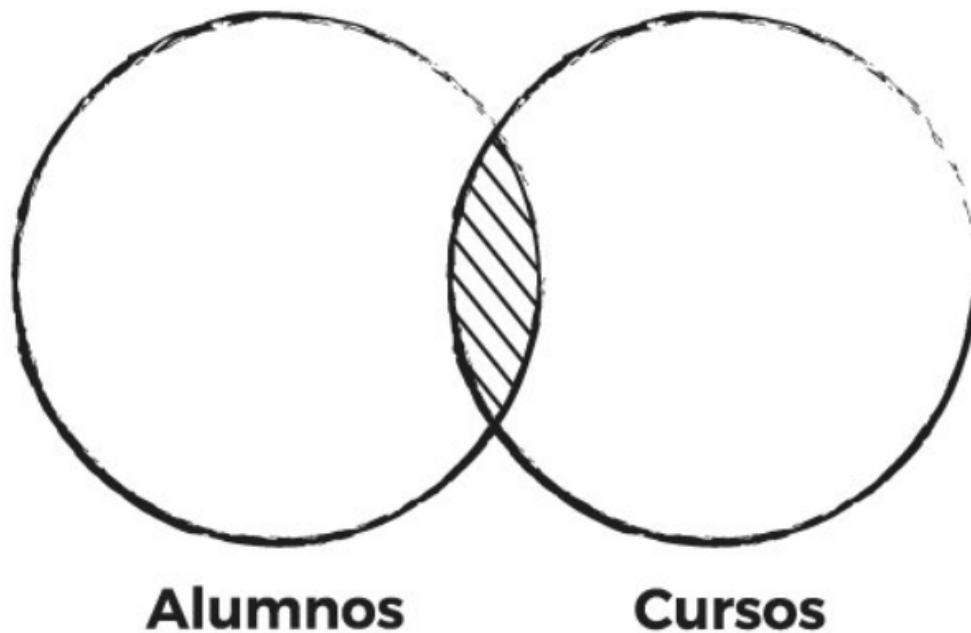
```

idAlumno	nombre	apellidos	idCurso	idCurso	curso
7	Pilar	Mirás	2	2	Doblaje
8	Antonio	Cambados	2	2	Doblaje
9	Pedro	Fraile	3	3	Pastelería

| 10 | Miguel | Yañez | 3 | 3 | Pastelería |

+-----+-----+-----+-----+-----+-----+

El siguiente diagrama de Venn ilustra el comportamiento del INNER JOIN:



LEFT JOIN

El LEFT JOIN es similar al INNER JOIN, solo que al seleccionarlo muestra todos los valores de la tabla de la izquierda aunque no tenga correspondencia en la tabla de la derecha. Para cada fila de la tabla de la izquierda el LEFT JOIN compara cada fila de la tabla de la derecha.

Si el valor de las dos filas cumple la condición del JOIN, la cláusula de LEFT JOIN crea una nueva fila que contiene todas las columnas de las dos tablas incluyendo aquellas que no cumplen la condición de la unión.

Si el valor no coincide, el LEFT JOIN crea igualmente una fila que contiene todos los datos de la tabla de la izquierda y la correspondencia con la tabla de la derecha se completa con NULL.

La cláusula de LEFT JOIN incluye todas las filas de la tabla de la izquierda como se puede ver en el siguiente ejemplo:

```
mysql> SELECT *
```

```
-> FROM alumnos a
```

```
-> LEFT JOIN cursos c
```

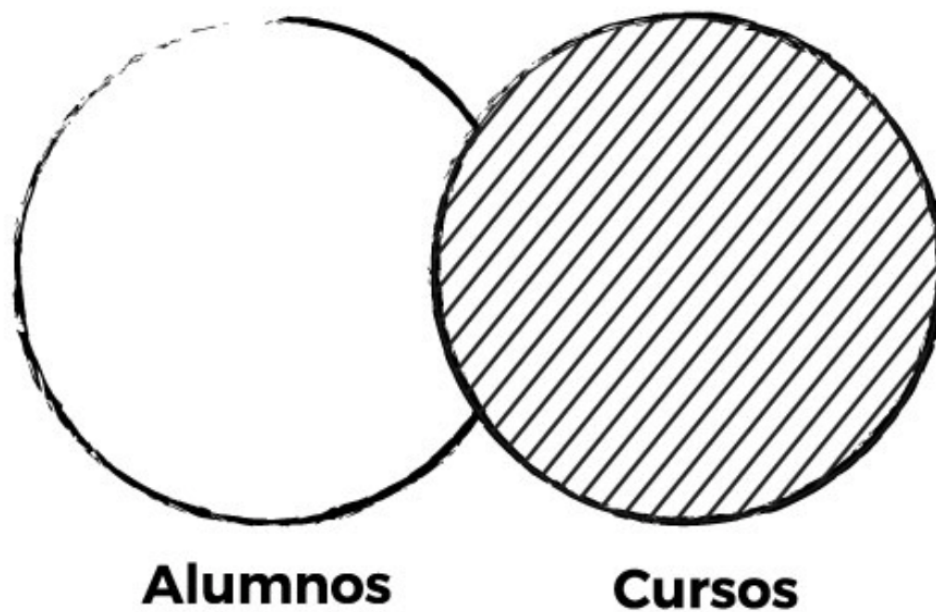
```
-> ON a.idCurso = c.idCurso;
```

idAlumno	nombre	apellidos	idCurso	idCurso	curso
7	Pilar	Mirás	2	2	Doblaje
8	Antonio	Cambados	2	2	Doblaje
9	Pedro	Fraile	3	3	Pastelería
10	Miguel	Yañez	3	3	Pastelería
11	Carlos	Sanchidrián	NULL	NULL	NULL
12	Alejandro	Rey	NULL	NULL	NULL

El siguiente diagrama de Venn ilustra el comportamiento del LEFT JOIN:

	NULL	NULL	NULL	NULL	1	Auxiliar de Enfermería
1	Pilar	Mirás	2	2	Doblaje	
2	Antonio	Cambados	2	2	Doblaje	
3	Pedro	Fraile	3	3	Pastelería	
4	Miguel	Yañez	3	3	Pastelería	

El siguiente diagrama de Venn ilustra el comportamiento del RIGHT JOIN:



CROSS JOIN

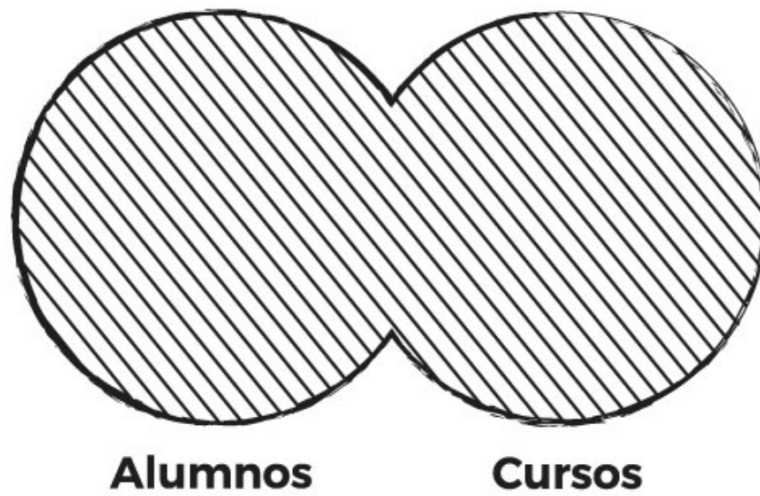
El CROSS JOIN es un tipo especial de JOIN que devuelve el producto cartesiano entre las tablas que forman parte de la unión. Se comporta igual que el INNER JOIN como si no tuviera la parte del “ON”:

```
mysql> SELECT *
      -> FROM alumnos a
```

-> CROSS JOIN cursos c;

idAlumno	nombre	apellidos	idCurso	idCurso	curso
7	Pilar	Mirás	2	1	Auxiliar de Enfermería
7	Pilar	Mirás	2	2	Doblaje
7	Pilar	Mirás	2	3	Pastelería
8	Antonio	Cambados	2	1	Auxiliar de Enfermería
8	Antonio	Cambados	2	2	Doblaje
8	Antonio	Cambados	2	3	Pastelería
9	Pedro	Fraile	3	1	Auxiliar de Enfermería
9	Pedro	Fraile	3	2	Doblaje
9	Pedro	Fraile	3	3	Pastelería
10	Miguel	Yañez	3	1	Auxiliar de Enfermería
10	Miguel	Yañez	3	2	Doblaje
10	Miguel	Yañez	3	3	Pastelería
11	Carlos	Sanchidrián	NULL	1	Auxiliar de Enfermería
11	Carlos	Sanchidrián	NULL	2	Doblaje
11	Carlos	Sanchidrián	NULL	3	Pastelería
12	Alejandro	Rey	NULL	1	Auxiliar de Enfermería
12	Alejandro	Rey	NULL	2	Doblaje
12	Alejandro	Rey	NULL	3	Pastelería

El siguiente diagrama de Venn ilustra el comportamiento del CROSS JOIN:



Si añadimos la cláusula ON, se comportará exactamente igual que el INNER JOIN:

```
mysql> SELECT *  
  
-> FROM alumnos a  
  
-> CROSS JOIN cursos c  
  
-> ON a.idCurso = c.idCurso;
```

idAlumno	nombre	apellidos	idCurso	idCurso	curso
7	Pilar	Mirás	2	2	Doblaje
8	Antonio	Cambados	2	2	Doblaje
9	Pedro	Fraile	3	3	Pastelería
10	Miguel	Yañez	3	3	Pastelería