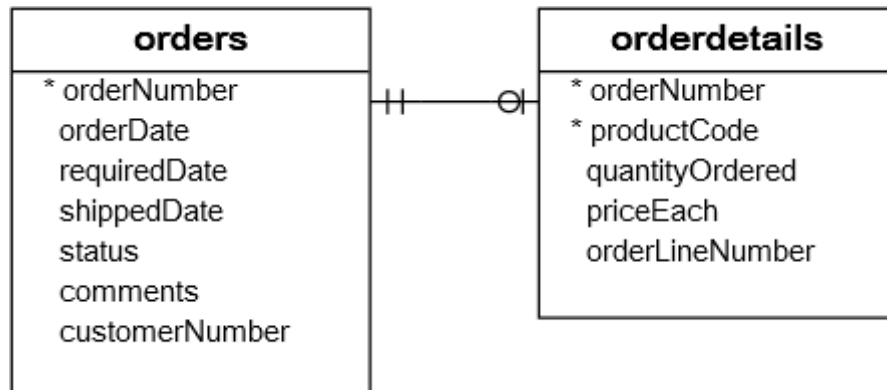


# CLÁUSULAS JOIN

## Introducción a las cláusulas JOIN de MySQL

Una base de datos relacional consta de varias tablas relacionadas que se enlazan entre sí utilizando columnas comunes que se conocen como columnas de [clave externa](#). Debido a esto, los datos de cada tabla están incompletos desde la perspectiva empresarial.

Por ejemplo, en la [base de datos de muestra](#), tenemos las tablas `orders` y `orderdetails` que están vinculadas mediante la `orderNumber` columna:



Para obtener la información completa de los pedidos, debe consultar los datos de las tablas `orders` y `orderdetails`.

Es por eso que las uniones entran en juego.

Una combinación es un método para vincular datos entre una ( [autocombinación](#) ) o más tablas en función de los valores de la columna común entre las tablas.

MySQL admite los siguientes tipos de combinaciones:

1. `INNER JOIN`
2. [LEFT JOIN](#)
3. [RIGHT JOIN](#)
4. [CROSS JOIN](#)

Para unir tablas, utilice la cláusula de combinación cruzada, combinación interna, combinación izquierda o combinación derecha para el tipo de combinación correspondiente. La cláusula de unión se utiliza en la `SELECT` declaración que aparece después de la `FROM` cláusula.

Tenga en cuenta que MySQL aún no es compatible con `FULL OUTER JOIN`.

## Configurar tablas de muestra

Primero, [cree dos tablas](#) llamadas `members` y `committees`:

```
CREATE TABLE members (
    member_id INT AUTO_INCREMENT,
    name VARCHAR(100),
    PRIMARY KEY (member_id)
);

CREATE TABLE committees (
    committee_id INT AUTO_INCREMENT,
    name VARCHAR(100),
    PRIMARY KEY (committee_id)
);
```

En segundo lugar, [inserte](#) algunas filas en las tablas `members` y `committees`:

```
INSERT INTO members(name)
VALUES('John'),('Jane'),('Mary'),('David'),('Amelia');

INSERT INTO committees(name)
VALUES('John'),('Mary'),('Amelia'),('Joe');
```

En tercer lugar, [consulte los datos](#) de las tablas `members` y `committees`:

```
SELECT * FROM members;
SELECT * FROM committees;
```

	member_id	name
▶	1	John
	2	Jane
	3	Mary
	4	David
	5	Amelia

	committee_id	name
▶	1	John
	2	Mary
	3	Amelia
	4	Joe

Algunos miembros son miembros del comité y otros no. Por otro lado, algunos miembros del comité están en la `members` mesa, otros no.

## Cláusula MySQL INNER JOIN

La cláusula de [combinación interna ó INNER JOIN](#) une dos tablas en función de una condición que se conoce como predicado de combinación.

La cláusula de combinación interna compara cada fila de la primera tabla con cada fila de la segunda tabla. Si los valores en ambas filas hacen que la condición de combinación se evalúe como verdadera, la cláusula de combinación interna 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 final. En otras palabras, la cláusula de combinación interna incluye solo filas cuyos valores coinciden.

A continuación se muestra la sintaxis básica de la cláusula de combinación interna que une dos tablas `table_1` y `table_2`:

```
SELECT column_list
FROM table_1
INNER JOIN table_2 ON join_condition;
```

Si la condición de unión usa el operador igual ( = ) y los nombres de columna en ambas tablas que se usan para la coincidencia son los mismos, puede usar la `USING` cláusula en su lugar:

```
SELECT column_list
FROM table_1
INNER JOIN table_2 USING (column_name);
```

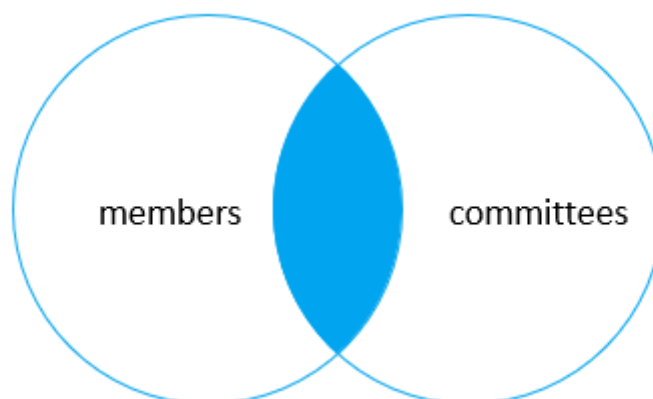
La siguiente declaración encuentra miembros que también son miembros del comité:

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
INNER JOIN committees c
    ON c.name = m.name;
```

	member_id	member	committee_id	committee
▶	1	John	1	John
	3	Mary	2	Mary
	5	Amelia	3	Amelia

En este ejemplo, la cláusula de combinación interna utilizó los valores de las `name` columnas en ambas tablas `members` y `committees` para hacer coincidir.

El siguiente diagrama de Venn ilustra la unión interna:



Debido a que las columnas de nombre son las mismas en ambas tablas, puede usar la `USING` cláusula como se muestra en la siguiente consulta:

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
INNER JOIN committees c USING(name);
```

## Cláusula MySQL LEFT JOIN

Similar a una combinación interna, una [combinación izquierda ó LEFT JOIN](#) también requiere un predicado de combinación. Al unir dos tablas usando una combinación izquierda, se introducen los conceptos de tablas izquierda y derecha.

La combinación de la izquierda selecciona datos a partir de la tabla de la izquierda. Para cada fila de la tabla de la izquierda, la combinación de la izquierda se compara con cada fila de la tabla de la derecha. Si los valores en las dos filas hacen que la condición de combinación se evalúe como verdadera, la combinación izquierda crea una nueva fila cuyas columnas contienen todas las columnas de las filas en ambas tablas e incluye esta fila en el conjunto de resultados.

Si los valores de las dos filas no coinciden, la cláusula de combinación de la izquierda aún crea una nueva fila cuyas columnas contienen columnas de la fila de la tabla de la izquierda y `NULL` columnas de la fila de la tabla de la derecha.

En otras palabras, la combinación de la izquierda selecciona todos los datos de la tabla de la izquierda, ya sea que existan filas coincidentes en la tabla de la derecha o no. En caso de que no se encuentren filas coincidentes de la tabla de la derecha, se utilizan `NULL` para las columnas de la fila de la tabla de la derecha en el conjunto de resultados final.

Aquí está la sintaxis básica de una cláusula de combinación izquierda que une dos tablas:

```
SELECT column_list
FROM table_1
LEFT JOIN table_2 ON join_condition;Lenguaje de código:  SQL (lenguaje de
consulta estructurado)  ( sql )
```

La combinación de la izquierda también admite la `USING` cláusula si la columna utilizada para la coincidencia en ambas tablas es la misma:

```
SELECT column_list
FROM table_1
LEFT JOIN table_2 USING (column_name);Lenguaje de código:  SQL (lenguaje de
consulta estructurado)  ( sql )
```

El siguiente ejemplo usa la combinación izquierda para unir el `members` con la `committees` tabla:

```

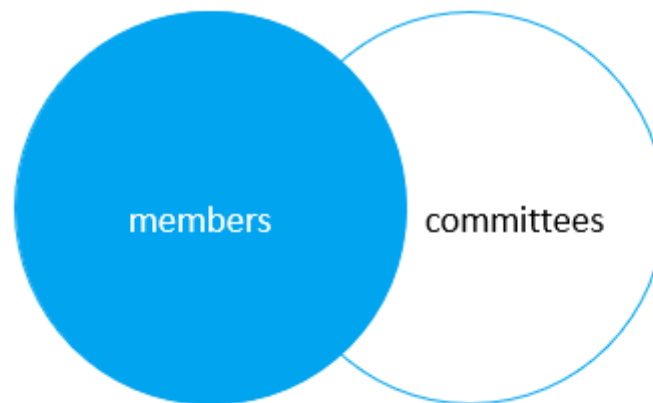
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
LEFT JOIN committees c USING(name);

```

Lenguaje de código: SQL (lenguaje de consulta estructurado) ( sql )

	member_id	member	committee_id	committee
▶	1	John	1	John
	3	Mary	2	Mary
	5	Amelia	3	Amelia
	2	Jane	NULL	NULL
	4	David	NULL	NULL

El siguiente diagrama de Venn ilustra la unión izquierda:



Esta declaración usa la cláusula de combinación izquierda con la `USING` sintaxis:

```

SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
LEFT JOIN committees c USING(name);

```

Lenguaje de código: SQL (lenguaje de consulta estructurado) ( sql )

Para encontrar miembros que no son miembros del comité, agregue una `WHERE` cláusula y un `IS NULL` operador de la siguiente manera:

```

SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
LEFT JOIN committees c USING(name)
WHERE c.committee_id IS NULL;

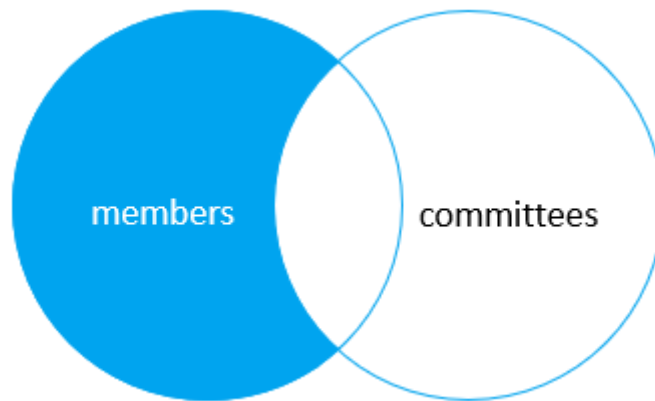
```

Lenguaje de código: SQL (lenguaje de consulta estructurado) ( sql )

	member_id	member	committee_id	committee
▶	2	Jane	NULL	NULL
	4	David	NULL	NULL

Generalmente, este patrón de consulta puede encontrar filas en la tabla de la izquierda que no tienen filas correspondientes en la tabla de la derecha.

Este diagrama de Venn ilustra cómo usar la combinación de la izquierda para seleccionar filas que solo existen en la tabla de la izquierda:



## Cláusula MySQL RIGHT JOIN

La cláusula de [combinación derecha ó RIGHT JOIN](#) es similar a la cláusula de combinación izquierda excepto que el tratamiento de las tablas se invierte. La combinación de la derecha comienza a seleccionar datos de la tabla de la derecha en lugar de la tabla de la izquierda.

La cláusula de combinación de la derecha selecciona todas las filas de la tabla de la derecha y hace coincidir las filas de la tabla de la izquierda. Si una fila de la tabla de la derecha no tiene filas coincidentes de la tabla de la izquierda, la columna de la tabla de la izquierda tendrá **NULL** en el conjunto de resultados final.

Aquí está la sintaxis de la combinación correcta:

```

SELECT column_list
FROM table_1
RIGHT JOIN table_2 ON join_condition;

```

Lenguaje de código: SQL (lenguaje de consulta estructurado) ( sql )

Similar a la cláusula de combinación izquierda, la cláusula derecha también admite la **USING** sintaxis:

```
SELECT column_list
FROM table_1
RIGHT JOIN table_2 USING (column_name);Lenguaje de código:  SQL (lenguaje de
consulta estructurado)  ( sql )
```

Para buscar filas en la tabla de la derecha que no tienen filas correspondientes en la tabla de la izquierda, también usa una `WHERE` cláusula con el `IS NULL` operador:

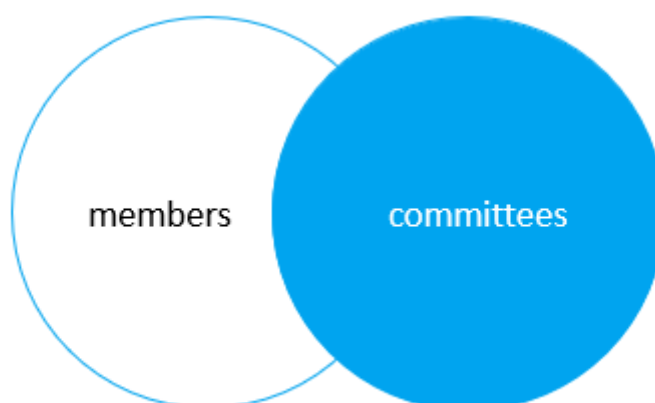
```
SELECT column_list
FROM table_1
RIGHT JOIN table_2 USING (column_name)
WHERE column_table_1 IS NULL;Lenguaje de código:  SQL (lenguaje de consulta
estructurado)  ( sql )
```

Esta declaración usa la combinación correcta para unir las tablas `members` y `committees`:

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
RIGHT JOIN committees c on c.name = m.name;Lenguaje de código:  SQL (lenguaje de
consulta estructurado)  ( sql )
```

	member_id	member	committee_id	committee
▶	1	John	1	John
	3	Mary	2	Mary
	5	Amelia	3	Amelia
	NULL	NULL	4	Joe

Este diagrama de Venn ilustra la unión derecha:



La siguiente declaración usa la cláusula de combinación correcta con la `USING` sintaxis:

```

SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
RIGHT JOIN committees c USING(name);

```

Lenguaje de código: SQL (lenguaje de consulta estructurado) ( sql )

Para encontrar los miembros del comité que no están en la `members` tabla, utilice esta consulta:

```

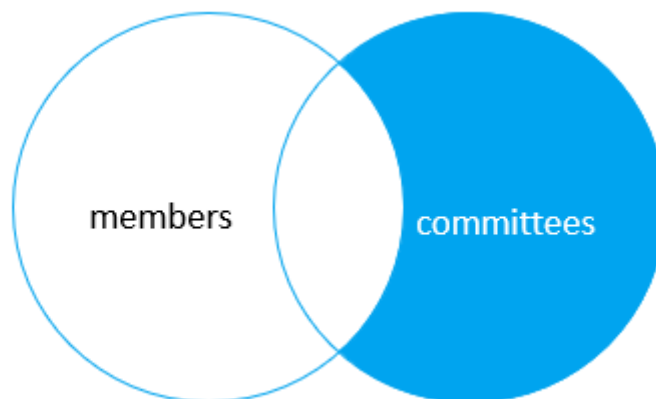
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
RIGHT JOIN committees c USING(name)
WHERE m.member_id IS NULL;

```

Lenguaje de código: SQL (lenguaje de consulta estructurado) ( sql )

	member_id	member	committee_id	committee
▶	NULL	NULL	4	Joe

Este diagrama de Venn ilustra cómo usar la combinación correcta para seleccionar datos que solo existen en la tabla de la derecha:



## Cláusula MySQL CROSS JOIN

A diferencia de la [combinación interna](#), la [combinación izquierda](#) y la [combinación derecha](#), la cláusula de [combinación cruzada](#) no tiene una condición de combinación.

La combinación cruzada crea un producto cartesiano de filas de las tablas unidas. La combinación cruzada combina cada fila de la primera tabla con cada fila de la tabla de la derecha para hacer el conjunto de resultados.

Suponga que la primera tabla tiene `n` filas y la segunda tabla tiene `m` filas. La combinación cruzada que une la primera con la segunda tabla devolverá `nxm` filas.

A continuación se muestra la sintaxis básica de la cláusula de combinación cruzada:



```
SELECT select_list
FROM table_1
CROSS JOIN table_2;Lenguaje de código:  SQL (lenguaje de consulta estructurado)
( sql )
```

Este ejemplo utiliza la cláusula de unión cruzada para unir el `members` con las `committees` tablas:

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
CROSS JOIN committees c;Lenguaje de código:  SQL (lenguaje de consulta
estructurado)  ( sql )
```

	member_id	member	committee_id	committee
▶	1	John	1	John
	1	John	2	Mary
	1	John	3	Amelia
	1	John	4	Joe
	2	Jane	1	John
	2	Jane	2	Mary
	2	Jane	3	Amelia
	2	Jane	4	Joe
	3	Mary	1	John
	3	Mary	2	Mary
	3	Mary	3	Amelia
	3	Mary	4	Joe
	4	David	1	John
	4	David	2	Mary
	4	David	3	Amelia
	4	David	4	Joe
	5	Amelia	1	John
	5	Amelia	2	Mary
	5	Amelia	3	Amelia
	5	Amelia	4	Joe

La combinación cruzada es útil para generar datos de planificación. Por ejemplo, puede realizar la planificación de ventas mediante la combinación cruzada de clientes, productos y años.

En este tutorial, ha aprendido varias declaraciones de combinación de MySQL, incluida la combinación cruzada, la combinación interna, la combinación izquierda y la combinación derecha para consultar datos de dos tablas.