

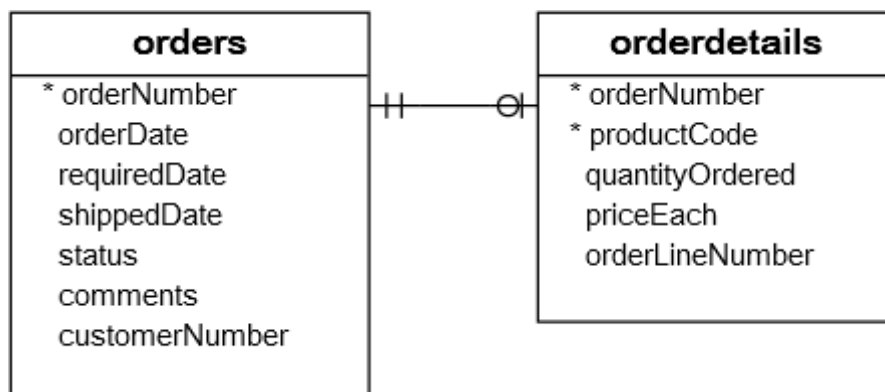
## MySQL Join

Resumen: en este tutorial, aprenderá varias cláusulas de unión de MySQL en la instrucción SELECT para consultar datos de dos tablas.

### Introducción a las cláusulas de unión de MySQL

Una base de datos relacional consta de varias tablas relacionadas que se unen mediante columnas comunes que se conocen como columnas de clave externa. Debido a esto, los datos en cada tabla están incompletos desde la perspectiva empresarial.

Por ejemplo, en la base de datos de muestra, tenemos las tablas de pedidos y detalles de pedidos que están vinculados usando la columna orderNumber:



Para obtener la información completa de los pedidos, debe consultar los datos de los pedidos y las tablas de detalles del pedido.

Es por eso que las uniones entran en juego.

Una unión es un método para vincular datos entre una (autounión) 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 unión cruzada, unión interna, unión izquierda o unión derecha para el tipo de unión correspondiente. La cláusula join se usa en la instrucción SELECT que aparece después de la cláusula FROM.

### Configurar tablas de muestra

Primero, cree dos tablas llamadas miembros y comités:

```
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 miembros y comités:

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

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

En tercer lugar, consultar datos de los miembros y comités de las tablas:

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

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

**MySQL Join - tabla de miembros**

	committee_id	name
▶	1	John
	2	Mary
	3	Amelia
	4	Joe

**MySQL Join - tabla de comités**

Algunos miembros son los miembros del comité, y otros no. Por otro lado, algunos miembros del comité están en la tabla de miembros, otros no.

### Cláusula INNER JOIN de MySQL

La cláusula de unión interna une dos tablas en función de una condición que se conoce como predicado de unión.

La cláusula de unión 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 unión se evalúe como verdadera, la cláusula de unión 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 unión interna incluye solo filas cuyos valores coinciden.

A continuación se muestra la sintaxis básica de la cláusula de unión 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 combinación utiliza el operador igual (=) y los nombres de columna en ambas tablas utilizados para la coincidencia son los mismos, puede utilizar la cláusula USING 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:

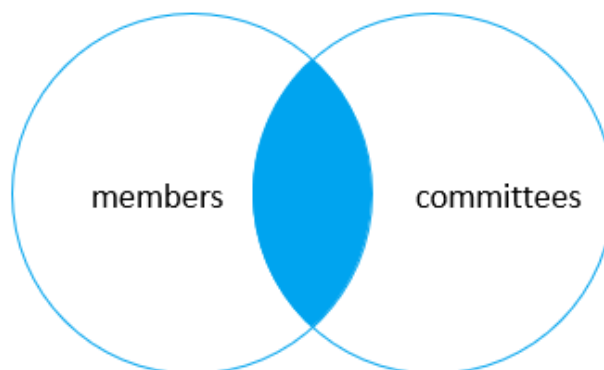
```
SELECT  
    m.member_id,  
    m.name member,  
    c.committee_id,  
    c.name committee  
FROM  
    members m  
INNER JOIN committees c  
    ON c.name = m.name;
```

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

**MySQL Join: ejemplo de INNER JOIN**

En este ejemplo, la cláusula de unión interna utiliza los valores en las columnas de nombre en los miembros de las tablas y los comités para que coincidan.

El siguiente diagrama de Venn ilustra la unión interna:



**MySql Join – INNER JOIN**

Como las columnas de nombre son las mismas en ambas tablas, puede usar la cláusula USING 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 también requiere un predicado de combinación. Al unir dos tablas con una combinación izquierda, se introducen los conceptos de tablas izquierda y derecha.

La unión izquierda selecciona datos a partir de la tabla izquierda. Para cada fila de la tabla izquierda, la unión izquierda se compara con cada fila de la tabla derecha. Si los valores en las dos filas hacen que la condición de unión se evalúe como verdadera, la unión 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 en las dos filas no coinciden, la cláusula de unión izquierda todavía crea una nueva fila cuyas columnas contienen columnas de la fila en la tabla izquierda y NULL para las columnas de la fila en la tabla derecha.

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

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

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

La combinación izquierda también admite la cláusula USING 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);
```

El siguiente ejemplo utiliza la combinación izquierda para unir a los miembros con la tabla de comités:

```
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
```

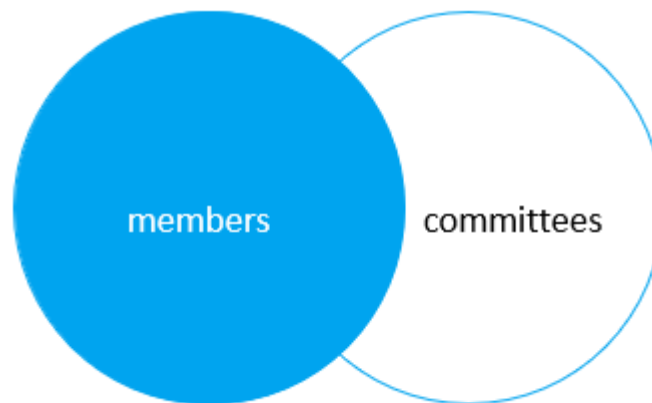
```

    c.name committee
FROM
    members m
LEFT JOIN committees c USING(name);

```

	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 unión izquierda con la sintaxis USING:

```

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

```

Para buscar miembros que no sean miembros del comité, agregue una cláusula WHERE y el operador IS NULL de la siguiente manera:

```

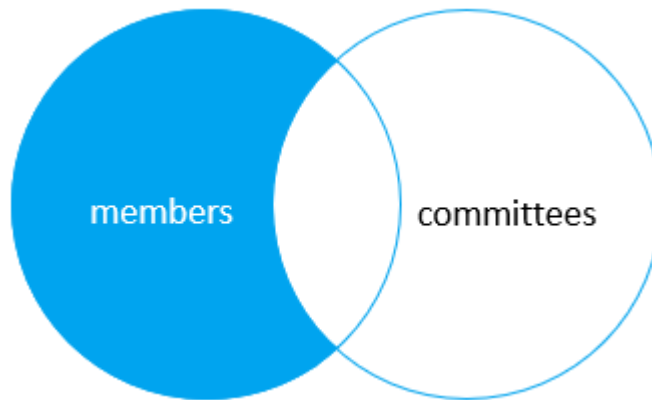
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
LEFT JOIN committees c USING(name)
WHERE c.committee_id IS NULL;

```

	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 izquierda que no tienen filas correspondientes en la tabla derecha.

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



### Cláusula MySQL RIGHT JOIN

La cláusula de unión derecha es similar a la cláusula de unión izquierda, excepto que el tratamiento de las tablas se invierte. La unión derecha comienza a seleccionar datos de la tabla derecha en lugar de la tabla izquierda.

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

Aquí está la sintaxis de la unión correcta:

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

Similar a la cláusula de unión izquierda, la cláusula derecha también admite la sintaxis de USO:

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

Para buscar filas en la tabla derecha que no tengan filas correspondientes en la tabla izquierda, también use una cláusula WHERE con el operador IS NULL:

```
SELECT column_list  
FROM table_1  
RIGHT JOIN table_2 USING (column_name)  
WHERE column_table_1 IS NULL;
```

Esta declaración utiliza la combinación correcta para unirse a las tablas de miembros y comités:

```
SELECT  
    m.member_id,
```

```

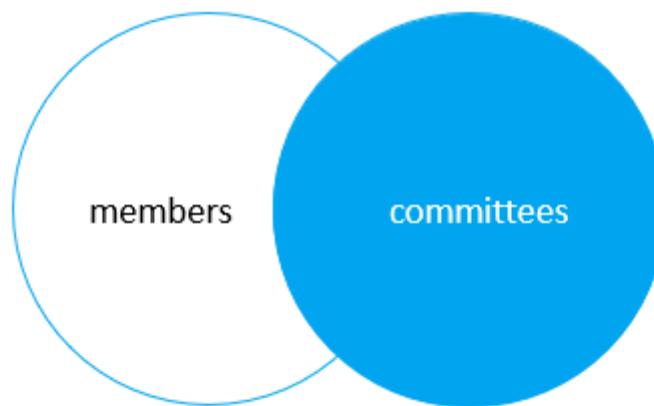
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
RIGHT JOIN committees c on c.name = m.name;

```

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

**MySQL Join: ejemplo de RIGHT JOIN**

Este diagrama de Venn ilustra la unión correcta:



**MySql Join – RIGHT JOIN**

La siguiente declaración utiliza la cláusula de unión correcta con la sintaxis USING:

```

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

```

Para encontrar los miembros del comité que no están en la tabla de miembros, use esta consulta:

```

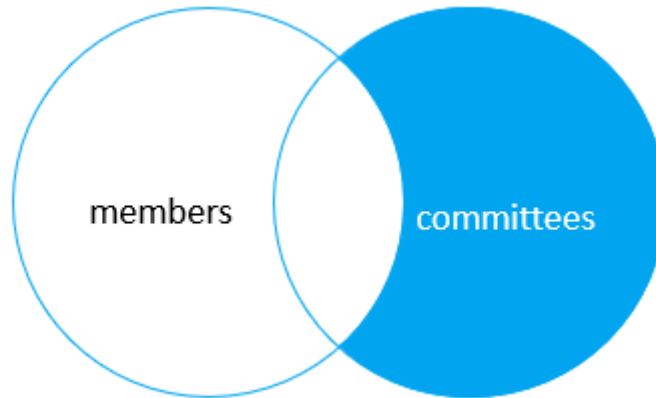
SELECT
    m.member_id,
    m.name member,
    c.committee_id,
    c.name committee
FROM
    members m
RIGHT JOIN committees c USING(name)
WHERE m.member_id IS NULL;

```

	member_id	member	committee_id	committee
▶	NULL	NULL	4	Joe

### MySQL Join – right join con la cláusula where

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



### Cláusula MySQL CROSS JOIN

A diferencia de la unión interna, la unión izquierda y la unión derecha, la cláusula de unión cruzada no tiene una condición de unión. La unión derecha crea un producto cartesiano de filas de las tablas unidas. La unión cruzada combina cada fila de la primera tabla con cada fila de la tabla derecha para hacer el conjunto de resultados.

Suponga que la primera tabla tiene n filas y la segunda tabla tiene m filas. La unión cruzada que une la primera con la segunda tabla devolverá  $n \times m$  filas.

A continuación se muestra la sintaxis básica de la cláusula cross join:

```
SELECT select_list
FROM table_1
CROSS JOIN table_2;
```

Este ejemplo utiliza la cláusula cross join para unir a los miembros con las tablas de comités:

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



	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

### MySQL Join - ejemplo de combinación cruzada

La unión cruzada es útil para generar datos de planificación. Por ejemplo, puede llevar a cabo la planificación de ventas utilizando la combinación cruzada de clientes, productos y años. En este tutorial, ha aprendido varias declaraciones de unión de MySQL, incluyendo unión cruzada, unión interna, unión izquierda y unión derecha para consultar datos de dos tablas.