

MySQL: Introducción Consultas Multitablas

Introducción consultas Multitabla

Hasta ahora, hemos realizado consultas que afectan a una sola tabla, pero es posible realizar consultas que combinen datos de múltiples tablas en MySQL. Simplemente puedes enumerar las tablas que deseas involucrar separadas por comas en la cláusula FROM de la consulta. Esto generará una combinación de todas las filas de todas las tablas mencionadas, lo que se conoce como un producto cartesiano.

Estructura Básica:

```
SELECT * FROM tabla1, tabla2;
```

Como puedes ver es muy simple combinar tablas, simplemente las separas con comas, el problema está en el resultado. En la consulta simple del ejemplo no hay cláusula WHERE, por lo tanto, no hay una condición que ordene la unión, el resultado será que cada fila de una tabla se unirá con todas las filas de la otra tabla, veámoslo con un ejemplo:

Vamos a unir la tabla empleados y departamentos:

```
SELECT * FROM empleados, departamentos;
```

id	nombre	apellido	edad	salario	departamento_id	id	nombre
1	Ana	Rodríguez	28	3707.55	1	3	Contabilidad
1	Ana	Rodríguez	28	3707.55	1	2	Recursos Humanos
1	Ana	Rodríguez	28	3707.55	1	1	Ventas

2	Carlos	López	32	3360.53	3	3	Contabilidad
2	Carlos	López	32	3360.53	3	2	Recursos Humanos
2	Carlos	López	32	3360.53	3	1	Ventas
4	Martín	González	30	3255.26	2	3	Contabilidad
4	Martín	González	30	3255.26	2	2	Recursos Humanos
4	Martín	González	30	3255.26	2	1	Ventas
5	Luis	Fernández	28	3145.80	1	3	Contabilidad
5	Luis	Fernández	28	3145.80	1	2	Recursos Humanos
5	Luis	Fernández	28	3145.80	1	1	Ventas
6	Marta	Ramírez	32	3317.00	1	3	Contabilidad
6	Marta	Ramírez	32	3317.00	1	2	Recursos Humanos
6	Marta	Ramírez	32	3317.00	1	1	Ventas
7	Lorena	Guzmán	26	2600.00	1	3	Contabilidad
7	Lorena	Guzmán	26	2600.00	1	2	Recursos Humanos
7	Lorena	Guzmán	26	2600.00	1	1	Ventas
8	Laura	Sánchez	27	3300.00	1	3	Contabilidad
8	Laura	Sánchez	27	3300.00	1	2	Recursos Humanos
8	Laura	Sánchez	27	3300.00	1	1	Ventas
9	Javier	Pérez	29	3100.00	1	3	Contabilidad
9	Javier	Pérez	29	3100.00	1	2	Recursos Humanos

9	Javier	Pérez	29	3100.00	1	1	Ventas
10	Camila	Gómez	26	3000.00	1	3	Contabilidad
10	Camila	Gómez	26	3000.00	1	2	Recursos Humanos
10	Camila	Gómez	26	3000.00	1	1	Ventas
11	Lucas	Fernández	28	3200.00	1	3	Contabilidad
11	Lucas	Fernández	28	3200.00	1	2	Recursos Humanos
11	Lucas	Fernández	28	3200.00	1	1	Ventas
12	Valentina	Rodríguez	30	3500.00	1	3	Contabilidad
12	Valentina	Rodríguez	30	3500.00	1	2	Recursos Humanos
12	Valentina	Rodríguez	30	3500.00	1	1	Ventas

📌 No incluimos las columnas de correo_electrónico y fecha_contratación por el espacio acotado

Como puedes ver son 33 filas porque por cada fila de la tabla empleados se une con las filas de la tabla departamentos, y como la tabla de empleados tiene 11 filas y la tabla de departamentos tiene 3, el resultado es $3 \times 11 = 33$.

Para evitar que suceda esto debemos tener una condición en la cláusula WHERE que permita relacionar una tabla con otra por medio de sus columnas, aquí es donde entran en juego las FOREIGN KEYS.

La tabla empleados se relaciona con la tabla departamentos a través de la columna departamento_id en la tabla empleados y la columna id de la tabla departamentos. Ahora nuestra consulta sería:

```
SELECT * FROM empleados, departamentos WHERE empleados.departamento_id = departamentos.id;
```

Cuando queremos hacer referencia a las columnas en una consulta multitabla es preferible que usemos la notación de "nombre_tabla.nombre_columna" porque es posible que haya nombres de columnas que se repitan en las tablas y

pueden producirse ambigüedades.

Puede ser muy engorroso usar los nombres completos de las tablas todo el tiempo, es por eso que podemos usar la cláusula AS para ponerles un alias y que sea más fácil armar la consulta:

```
SELECT * FROM empleados AS e, departamentos AS d WHERE e.departamento_id = d.id;  
# recuerda que el AS es opcional a la hora crear un alias:  
SELECT * FROM empleados e, departamentos d WHERE e.departamento_id = d.id;
```

id	nombre	apellido	edad	salario	departamento_id	id	nombre
1	Ana	Rodríguez	28	3707.55	1	1	Ventas
5	Luis	Fernández	28	3145.80	1	1	Ventas
6	Marta	Ramírez	32	3317.00	1	1	Ventas
7	Lorena	Guzmán	26	2600.00	1	1	Ventas
8	Laura	Sánchez	27	3300.00	1	1	Ventas
9	Javier	Pérez	29	3100.00	1	1	Ventas
10	Camila	Gómez	26	3000.00	1	1	Ventas
11	Lucas	Fernández	28	3200.00	1	1	Ventas
12	Valentina	Rodríguez	30	3500.00	1	1	Ventas
4	Martín	González	30	3255.26	2	2	Recursos Humanos
2	Carlos	López	32	3360.53	3	3	Contabilidad

Para entender lo que está sucediendo tenemos que volver a pensar que la consulta antes del WHERE devuelve el producto cartesiano de ambas tablas, por lo tanto el resultado son 33 filas, entonces las condiciones del WHERE se aplican sobre ese conjunto, lo que termina resultando en la eliminación de aquellas filas donde "e.departamento_id" y "d.id" no sean iguales.

Recuerda usar las FOREIGN KEYS para relacionar las tablas entre sí al realizar en un futuro consultas.