



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Mexicali

Ingeniería en Sistemas Computacionales

Fundamentos de base de datos

Grupo: B

Formas de hacer cross sin join en sql

Alumno: *Morales Ruiz José Luis*

Numero de control: 23490385

Mexicali, B.C., a 03 de marzo de 2025.

1. Uso de CROSS JOIN implícito con coma (,)

En SQL estándar, puedes hacer un producto cartesiano usando una coma en la cláusula FROM:

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

Esto es equivalente a:

```
SELECT *
FROM      tabla1
CROSS JOIN      tabla2;
```

2. Uso de SELECT sin JOIN

Si no quieres usar ni CROSS JOIN ni ,, puedes hacer combinaciones con subconsultas:

```
SELECT *
FROM      (SELECT * FROM tabla1) t1, (SELECT * FROM tabla2) t2;
```

Esto es útil si necesitas modificar los datos antes del producto cartesiano.

3. Uso de UNION ALL con combinaciones manuales

Si las tablas tienen un número finito de valores y quieres hacer un producto cartesiano manualmente, puedes usar UNION ALL en combinación con SELECT:

sql

```
SELECT      t1.columna1,      t2.columna2
FROM      (SELECT 'A' AS columna1 UNION ALL SELECT 'B') AS t1,
      (SELECT 1 AS columna2 UNION ALL SELECT 2) AS t2;
```

Esto generará el producto cartesiano de {A, B} con {1, 2}.

4. Uso de CROSS APPLY o OUTER APPLY (SQL Server y otros motores compatibles)

Si tu base de datos lo permite, puedes usar CROSS APPLY, que también permite hacer un producto cartesiano si no hay una condición de relación:

```
SELECT *  
FROM tabla1  
CROSS APPLY (SELECT * FROM tabla2) AS t2;
```

Uso de la Sintaxis "FROM Tabla1, Tabla2"

Una forma de unir dos tablas sin una columna en común es utilizar una sintaxis obsoleta para unir tablas. Con esta sintaxis, simplemente listamos las tablas que queremos unir en la cláusula **FROM** y luego usamos una cláusula **WHERE** para añadir condiciones de unión si es necesario.

Si lo que queremos es cada combinación de filas de dos tablas, no es necesario incluir ninguna condición de unión. Podemos utilizar una consulta como la siguiente:

```
SELECT w.name AS wine, m.name AS main_course  
FROM wine w, main_course m;
```

La consulta devolverá un **producto cartesiano** (una unión cruzada), cuyo conjunto de resultados tiene el número total de filas igual **al número de filas de la primera tabla multiplicado por el número de filas de la segunda tabla**.

wine	main_course
Merlot	Cardamom Maple Salmon
House	Cardamom Maple Salmon
Sangiovese	Cardamom Maple Salmon
Merlot	Classic Macaroni & Cheese
House	Classic Macaroni & Cheese
Sangiovese	Classic Macaroni & Cheese
Merlot	Baked Teriyaki Chicken
House	Baked Teriyaki Chicken
Sangiovese	Baked Teriyaki Chicken
Merlot	Blue Cheese Beef Tenderloin
House	Blue Cheese Beef Tenderloin
Sangiovese	Blue Cheese Beef Tenderloin

Obtenemos el resultado que queríamos. Esta es una solución viable para unir dos tablas cuando el objetivo es obtener un conjunto de resultados con todas las combinaciones posibles de las filas. Sin embargo, este no es el mejor enfoque.

Los profesionales de SQL prefieren utilizar la palabra clave explícita **CROSS JOIN** para este tipo de operaciones. Esto deja claro que estamos realizando una unión cruzada de forma intencionada y no omitiendo accidentalmente la condición de unión **WHERE**, lo que podría ocurrir al utilizar sintaxis obsoleta para unir tablas.

Lee más sobre los beneficios de la nueva sintaxis para unir tablas, en nuestro artículo [¿Cuál es la diferencia entre JOIN y múltiples tablas en FROM?](#)

Uso del operador CROSS JOIN

Como ya habrás adivinado, el segundo enfoque para obtener todas las combinaciones posibles de las filas de dos tablas es utilizar el operador **CROSS JOIN**:

```
SELECT w.name AS wine, m.name AS main_course  
FROM wine w  
CROSS JOIN main_course m;
```

Esta consulta produce exactamente el mismo conjunto de resultados que el mostrado anteriormente. Sin embargo, **es preferible esta sintaxis** ya que indica claramente la intención de realizar una **CROSS JOIN** (unión cruzada) y es más legible para otros usuarios que vayan a leer su código SQL.

Aprenda más sobre las uniones cruzadas en nuestra [guía ilustrada de SQL CROSS JOIN](#).

Fuentes de informacion: LearnSQL. (s.f.). *¿Se pueden unir dos tablas sin una columna en común?* LearnSQL. Recuperado el 3 de marzo de 2025, de <https://learnsql.es/blog/se-pueden-unir-dos-tablas-sin-una-columna-en-comun/>