

Practicar con el OUTER JOIN

Estos son unos ejemplos para que te vayas familiarizando con el OUTER JOIN.

Después de practicar con el INNER JOIN vamos a ampliar las combinaciones.

Vamos a probar estos ejemplos con la base de datos GestionA suministrada en la unidad 2.

Pero antes debes cambiar la definición de la tabla Pedidos en esa base de datos, tienes que hacer que las columnas rep, clie, fab, producto no admitan valores nulos. La tabla ya tiene datos pero como no hay ningún NULL en estas columnas, no habrá problemas.

El OUTER JOIN hay que enfocarlo en un principio como un INNER JOIN y una vez tenemos definido el ON elegimos el tipo de JOIN.

Al guión que utilizamos en ejemplos anteriores ahora tenemos que añadir lo referente al OUTER JOIN.

- ¿Dónde están los datos que necesito? La respuesta me dará la FROM.
 - Si los datos están en más de una tabla me pregunto:
 - Quiero **concatenar** filas de una tabla con filas de otra?
 - si es que **no** → necesito utilizar una de las operaciones de **conjunto**. Completa con el guión de las operaciones de conjunto.
 - Si es que **Sí** → necesito una **composición** de tablas.
 - ¿Quiero **combinar todas las filas con todas**?
 - Si es que Sí → Producto cartesiano
 - Si es que NO → INNER JOIN
 - Tengo que pensar primero en la condición de combinación → ON.
 - ¿Pueden haber filas de la 1ª tabla que no estén relacionadas con filas de la 2ª tabla?
 - Si es que Sí y nos interesa que salgan → cambiar a LEFT
 - Si es que no o es que sí pero no nos interesa que salgan → dejar INNER
 - ¿Pueden haber filas de la 2ª que no estén relaciondas con filas de la 1ª tabla?
 - Si es que Sí y nos interesa que salgan:
 - Convertir INNER → RIGHT
 - Convertir LEFT → FULL
 - Si es que no o es que sí pero no nos interesa que salgan → dejar el JOIN que había

El resto es lo mismo que vimos en los guiones anteriores.

1.- Vamos a obtener los datos de TODOS los empleados con los datos de **su** oficina. Empezamos por un INNER JOIN:

```
SELECT *
FROM empleados INNER JOIN oficinas ON empleados.oficina=oficinas.oficina;
```

En el ON el orden de los campos es indiferente se puede escribir oficinas.oficina = empleados.oficina.

Una vez montado el INNER JOIN Te preguntas:

¿Puede haber **empleados** (filas de la primera tabla) sin oficina (que no estén relacionados con filas de la segunda tabla)?

En este caso la respuesta es que Sí (todos los empleados que tienen oficina NULL).

¿Nos interesan? La respuesta es que sí (me piden TODOS los empleados), entonces cambio el INNER a LEFT (la tabla de empleados está a la izquierda).

Pero no hemos terminado, tenemos que plantear también la pregunta al revés:

¿Pueden haber oficinas (filas de la segunda tabla) que no tengan empleados (que no estén relacionadas con filas de la primera tabla)?

En este caso es que sí, ¿nos interesa que salgan? La respuesta es que no, entonces dejamos lo que había (LEFT).

```
SELECT *
FROM empleados LEFT JOIN oficinas ON empleados.oficina=oficinas.oficina;
```

Fíjate en el resultado, aparecen los empleados que tienen oficina junto con los datos de su oficina y los que no tienen oficina aparecen con valores nulos en los campos de la tabla Oficinas.

Si ponemos a la izquierda la tabla oficinas:

```
SELECT *  
FROM oficinas INNER JOIN empleados ON empleados.oficina=oficinas. oficina;
```

Las preguntas se convierten en:

¿Pueden haber oficinas (filas de la 1ª tabla) que no tengan empleados (que no estén relacionadas con filas de la 2ª tabla)?

R: Sí, ¿Nos interesa que salgan? NO (lo que me piden son empleados), luego dejamos INNER de momento.

Ahora planteamos la pregunta al revés:

¿Pueden haber **empleados** (filas de la 2ª tabla) sin oficina (que no estén relacionados con filas de la 1ª tabla)?

R: Sí, ¿Nos interesan? Sí, entonces pasamos de INNER a RIGHT:

```
SELECT *  
FROM oficinas RIGHT JOIN empleados ON empleados.oficina=oficinas. oficina;
```

Fíjate en el resultado, aparecen las oficinas que tienen empleados junto con los datos de sus empleados y las que no tienen empleados aparecen con valores nulos en los campos de la tabla Empleados.

Las filas añadidas a lo que sería el resultado de un INNER JOIN aparecen en estos ejemplos al final, es una 'casualidad' que nos viene bien para verlo mejor pero no es un comportamiento estándar, las filas añadidas pueden aparecer en medio de las filas del INNER JOIN.

2.- Obtener una lista de TODOS los clientes indicando el nombre del representante asignado al cliente.

Primero planteamos el INNER:

```
SELECT *  
FROM Clientes INNER JOIN empleados ON Clientes.repclie = empleados.numemp;
```

Luego estudiamos si necesitamos un INNER o un OUTER:

¿Pueden haber clientes (filas de la 1ª tabla) que no tengan representante asignado (que no estén relacionadas con filas de la 2ª tabla)?

R: Sí (para responder no sólo hay que fijarse en los valores que tenemos en la tabla Clientes, hay que mirar la definición de la tabla, si el campo repclie admite o no nulos. Si admite nulos, aunque en este momento en la tabla no haya ningún cliente así, puede que en algún momento sí lo haya porque la definición de la tabla lo permite. Si el campo repclie fuese no nulo entonces responderíamos No a la pregunta).

¿Nos interesan estos clientes?

R: Sí (nos piden TODOS los clientes) → pasamos el INNER a LEFT (los clientes están a la izquierda del JOIN)

Seguimos con las preguntas:

¿Pueden haber empleados (filas de la 2ª tabla) que no estén asignados a ningún cliente (que no estén relacionados con filas de la 1ª tabla)?

R: Sí, ¿Nos interesan? No, entonces dejamos el LEFT:

```
SELECT *  
FROM Clientes LEFT JOIN empleados ON Clientes.repclie = empleados.numemp;
```

El orden en que pongamos las tablas no influye siempre que pongamos el JOIN adecuado. Podíamos haber escrito primero la tabla Empleados:

```
SELECT *  
FROM Empleados INNER JOIN Clientes ON Empleados.numemp = Clientes.repclie;
```

Luego estudiamos si necesitamos un INNER o un OUTER:

¿Pueden haber empleados (filas de la 1ª tabla) que no estén asignados a ningún cliente (que no estén relacionadas con filas de la 2ª tabla)?

R: Sí.

¿Nos interesan estos empleados?

R: No (nos piden TODOS los clientes) → dejamos el INNER .

Seguimos con las preguntas:

¿Pueden haber clientes (filas de la 2ª tabla) que no tengan un representante asignado (que no estén relacionados con filas de la 1ª tabla)?

R: Sí.

¿Nos interesan?

Sí (deben aparecer todos los clientes), entonces pasamos el INNER a RIGHT (la tabla Clientes está a la derecha):

```
SELECT *  
FROM Empleados RIGHT JOIN Clientes ON Empleados.numemp = Clientes.repclie;
```

3.- Ahora nos piden un informe en el que aparezca de cada empleado los clientes a los que está asignado, deben aparecer todos los empleados y además todos los clientes.

Primero planteamos el INNER:

```
SELECT *  
FROM Empleados INNER JOIN Clientes ON Empleados.numemp = Clientes.repclie;
```

En el ON el orden de los campos es indiferente se puede escribir Clientes.repclie = Empleados.numemp

Luego estudiamos si necesitamos un INNER o un OUTER:

¿Pueden haber empleados (filas de la 1ª tabla) que no estén asignados a ningún cliente (que no estén relacionadas con filas de la 2ª tabla)?

R: Sí.

¿Nos interesan estos clientes?

R: Sí (nos piden TODOS los empleados) → pasamos el INNER a LEFT (los empleados están a la izquierda del JOIN)

Seguimos con las preguntas:

¿Pueden haber clientes (filas de la 2ª tabla) que no tengan un representante asignado (que no estén relacionados con filas de la 1ª tabla)?

R: Sí.

¿Nos interesan?

Sí (deben aparecer todos los clientes), entonces pasamos el LEFT que teníamos a FULL:

```
SELECT *  
FROM Empleados INNER JOIN Clientes ON Empleados.numemp = Clientes.repclie;
```

Si hubiésemos planteado el JOIN empezando por la tabla Clientes:

```
SELECT *  
FROM Clientes INNER JOIN empleados ON Clientes.repclie = empleados.numemp;
```

Luego estudiamos si necesitamos un INNER o un OUTER:

¿Pueden haber clientes (filas de la 1ª tabla) que no tengan representante asignado (que no estén relacionadas con filas de la 2ª tabla)?

R: Sí.

¿Nos interesan estos clientes?

R: Sí (nos piden TODOS los clientes) → pasamos el INNER a LEFT (los clientes están a la izquierda del JOIN)

Seguimos con las preguntas:

¿Pueden haber empleados (filas de la 2ª tabla) que no estén asignados a ningún cliente (que no estén relacionados con filas de la 1ª tabla)?

R: Sí.

¿Nos interesan?

Sí (deben aparecer todos los empleados), entonces pasamos el LEFT que teníamos a FULL:

```
SELECT *  
FROM Clientes FULL JOIN empleados ON Clientes.repclie = empleados.numemp;
```

4.- Obtener una lista de TODOS los pedidos indicando para cada pedido el nombre del representante que ha hecho el pedido.

Empezamos por el INNER:

```
SELECT *  
FROM Pedidos INNER JOIN Empleados ON Pedidos.rep = Empleados.numemp;
```

¿Pueden haber pedidos sin representante?

R: No (el campo rep no admite nulos, recuerda, lo has cambiado al principio del ejercicio), entonces dejamos INNER.

¿Pueden haber empleados sin pedidos?

Sí, pero no nos intersan → dejamos el INNER.

¡Ojo con la palabra TODOS, que queramos que salgan todas las filas de una tabla no implica que tengamos que poner siempre un OUTER JOIN (LEFT/RIGHT/FULL)!

5.- Obtener una lista de TODOS los clientes indicando para cada cliente sus pedidos.

Empezamos por el INNER:

```
SELECT *  
FROM Clientes INNER JOIN Pedidos ON Clientes.numclie = Pedidos.clie;
```

¿Pueden haber clientes sin pedidos?

Sí, y nos intersan → pasamos a LEFT.

¿Pueden haber pedidos sin cliente?

R: No, entonces dejamos LEFT.

```
SELECT *  
FROM Clientes LEFT JOIN Pedidos ON Clientes.numclie = Pedidos.clie;
```

6.- Cuando trabajamos con consultas reflexivas (que utilizan la misma tabla varias veces), el procedimiento es idéntico, recuerda que tienes que ver la tabla que se repite como dos tablas.

De cada empleado quiero saber el nombre de su jefe.

Empezamos por el INNER:

```
SELECT empleados.*, jefes.numemp AS Codigo_Jefe, jefes.nombre AS Nombre_Jefe  
FROM empleados INNER JOIN empleados AS jefes  
ON empleados.jefe = jefes.numemp;
```

¿Pueden haber empleados sin jefe?

Sí, y nos interesan → Paso a LEFT

¿Pueden haber jefes sin empleado asignado?

No → Dejamos LEFT

```
SELECT empleados.*, jefes.numemp AS Codigo_Jefe, jefes.nombre AS Nombre_Jefe  
FROM empleados LEFT JOIN empleados AS jefes  
ON empleados.jefe = jefes.numemp;
```

7.- De cada jefe quiero saber el nombre de los empleados a su cargo.

Empezamos por el INNER:

```
SELECT jefes.*, empleados.numemp AS Codigo_Subordinado, empleados.nombre AS  
Nombre_Subordinado, empleados.jefe AS Su_jefe  
FROM empleados INNER JOIN empleados AS jefes  
ON empleados.jefe = jefes.numemp;
```

¿Pueden haber empleados sin jefe?

Sí, pero no nos interesan → dejamos INNER

¿Pueden haber jefes sin empleado asignado? Realmente la pregunta sería pueden haber empleados que no son jefes, pero esos de todas formas no nos interesan como jefes.

No → Dejamos INNER