

# Practicar con el INNER JOIN

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

Después de practicar con el producto cartesiano, veamos ahora el INNER JOIN. Piensa que no es más que un producto cartesiano con una condición de combinación.

Puedes probarlos utilizando la base de datos GestionA suministrada en la unidad 2.

Con el JOIN (me referiré siempre con JOIN a cualquier JOIN que no sea CROSS JOIN, este último siempre será producto cartesiano) hay que fijarse muy bien en la condición de combinación, si obtenemos filas concatenadas con filas que no les corresponden, o más filas de las que se esperaba puede que sea por culpa de un error en la condición de combinación.

Al guión que utilizamos en ejemplos anteriores ahora tenemos que añadir lo referente al INNER 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 (quiero unas combinadas con su padre) → INNER JOIN

Tengo que pensar la condición de combinación.

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

1.- Vamos a obtener los datos de los empleados con los datos de **su** oficina.

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

Esta instrucción obtiene el mismo resultado que el último producto cartesiano que vimos en la anterior hoja de ejemplos, pero es mejor, se ejecuta mejor a nivel interno. Básicamente en vez de combinar todas las filas de empleados con todas las filas de oficinas, por cada fila de empleados busca directamente el registro correspondiente en la tabla de oficinas y lo combina.

Siempre que quieras combinar hijos con su padre o viceversa, utiliza un JOIN.

La condición de unión (la que pones en el ON) es fundamental si no pones la condición adecuada no se combinarán las filas correctas.

2.- Prueba la sentencia de antes pero cambiando el ON

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

Observa que el resultado es diferente. ¿Qué filas hemos obtenido? Las que cumplen que el jefe del empleado sea el director de la oficina, es decir los empleados que tienen un jefe que también es director de una oficina. Y eso no son los empleados con su oficina.

3.- Lo 'complicado' de un INNER JOIN es acertar el ON.

Al principio sobre todo lo que suele costar es encontrar la condición a poner en el ON. Para eso te recomiendo coger la hoja con las tablas y coger un ejemplo de lo que se quiere obtener. Por ejemplo Nos piden obtener los datos de los clientes junto con el nombre del representante asignado al cliente.

Pues busca en las tablas los datos de un cliente y el nombre de su representante. Por ejemplo el cliente 2101 Luis García Antón tiene asignado el representante 106 que se llama Luis Antonio. Fijate en los campos que has consultado, para averiguar que es Luis Antonio has buscado el empleado 106 en la tabla de Empleados, el valor 106 es el nexo de unión entre ese cliente y su representante, pues los campos donde está el valor son los que se igualan en el ON.

La solución es pues:

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

4.- Queremos obtener los pedidos realizados junto con los datos del cliente del pedido.

El cliente del pedido lo tenemos en el campo clie de la tabla Pedidos, para saber cómo se llama, busco el valor en la columna numclie de la tabla Clientes→ Ya tenemos el ON

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

5.- Queremos los pedidos junto con los datos del representante del pedido.

Seguimos buscando un ejemplo en las tablas. Cojemos un pedido el 1, su representante es el 110 (columna rep), ¿Cómo se llama? Busco el 110 en la tabla Empleados en la columna numemp.

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

6.- Queremos los pedidos junto con los datos del producto vendido en el pedido.

Ojo aquí recuerda que la clave de la tabla Productos está formada por fab (el código de fabricante) e idproducto (el código del producto) con lo cual para buscar el producto del pedido tienes que fijarte en fab y producto. El producto del pedido 1 es ACI – 4100z

```
SELECT *
FROM pedidos INNER JOIN productos ON producto = idproducto AND fab = idfab;
```

Prueba de quitar por ejemplo la condición fab = idfab y ordena por codigo para ver mejor lo que ocurre:

```
SELECT *
FROM pedidos INNER JOIN productos ON producto = idproducto
ORDER BY codigo;
```

¿Qué ocurre? Salen más filas que pedidos. Por ejemplo el pedido 12 sale dos veces con dos productos diferentes el art t3 y la manivela porque los dos productos tienen el mismo idproducto pero son de fabricantes diferentes.

¿Porqué? Sale cada pedido con todos los productos que tengan el mismo idproducto pero como un mismo idproducto se puede repetir en varios productos (con idfab diferente), el pedido se puede combinar con varios productos.

7.- También podemos añadir una condición de selección con una cláusula WHERE. Queremos ahora los pedidos de manivelas.

```
SELECT *
FROM pedidos INNER JOIN productos ON producto = idproducto AND fab = idfab
WHERE descripcion = 'manivela';
```

La condición la hubiéramos podido añadir al ON pero prefiero ponerla fuera para distinguir entre la condición de combinación y la de selección. Como quiero que lo sepáis distinguir quiero que lo hagáis igual. En el ON sólo deberá aparecer la condición de combinación (la que permite juntar una fila con su correspondiente en la otra tabla) y la condición de selección (la que permite seleccionar filas del origen).

8.- Quiero obtener los pedidos del cliente *Jaime Llorens*

Aunque sólo quisiéramos visualizar los datos del pedido, como el dato que nos dan es el nombre del cliente, para poder utilizar el nombre del cliente en el WHERE necesito tener en el origen la tabla de Clientes y no combinada de cualquier manera, necesito tener cada pedido con SU cliente, luego será un JOIN.

```
SELECT *
FROM Pedidos INNER JOIN Clientes ON Pedidos.clie = Clientes.numclie
WHERE nombre= 'Jaime Llorens';
```

9.- Para terminar practicaremos con el JOIN de una tabla consigo misma.

Queremos saber el nombre del jefe de cada empleado. Para ello hacemos lo de siempre para saber el ON, cojo un ejemplo, el empleado 101, ¿cómo se llama su jefe?

Para saberlo me fijo en la columna jefe, apunto el nº (104) y para saber cómo se llama el 104, busco el nº 104 en la columna numemp. Fíjate que ahora la búsqueda la he hecho en la misma tabla, luego tengo que combinar empleados con empleados:

Empleados INNER JOIN Empleados

Pero en este JOIN el primer Empleados hace el papel de tabla de Empleados (de donde he cogido el empleado 101) y el segundo Empleados hace el papel de Jefes (donde he buscado el 104 para saber su nombre, el nombre del jefe). Pues como no puedo tener en el origen dos veces la misma tabla sin definir un alias de tabla, defino un alias para la tabla de Jefes:

Empleados INNER JOIN Empleados AS Jefes

el 104 estaba en la columna jefe del empleado y luego lo he buscado en la columna numemp, entonces la condición del ON será:

Empleados.jefe = Jefes.numemp

```
SELECT *
FROM empleados INNER JOIN empleados AS jefes
ON empleados.jefe = jefes.numemp;
```

10.- Para verlo mejor saca del jefe sólo su código y nombre:

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

Cuando tenemos que combinar una tabla consigo misma, imagina que en vez de una tabla tienes dos (como hemos hecho con los empleados y jefes) y aplica el mismo razonamiento que cuando son tablas distintas.

11.- ¿Qué pasa si pones otro ON?

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

Hago que el código del empleado coincida con el código de la segunda tabla, luego estoy combinando cada empleado consigo mismo.

Y

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

He rectificado la lista de selección y añadido el ORDER BY para que se aprecie mejor el resultado.

Hago que el jefe del empleado sea el mismo jefe que el segundo. Pues voy a combinar los empleados que tienen el mismo jefe. Por ejemplo el empleado 101 lo combino con todos los empleados que tienen el jefe 104 (el jefe del 101).