

Hemos visto cómo usar registros de una tabla para encontrar registros de otra tabla, uniendo ambas tablas con "join" y enlazándolas con una condición "on" en la cual colocamos el campo en común. O sea, hacemos un "join" y asociamos registros de 2 tablas usando el "on", buscando coincidencia en los valores del campo que tienen en común ambas tablas.

Trabajamos con las tablas de una librería:

-libros: codigo (clave primaria), titulo, autor, codigoeditorial, precio, cantidad y

-editoriales: codigo (clave primaria), nombre.

Queremos saber de qué editoriales no tenemos libros.

Para averiguar qué registros de una tabla no se encuentran en otra tabla necesitamos usar un "join" diferente.

Necesitamos determinar qué registros no tienen correspondencia en otra tabla, cuáles valores de la primera tabla (de la izquierda) no están en la segunda (de la derecha).

Para obtener la lista de editoriales y sus libros, incluso de aquellas editoriales de las cuales no tenemos libros usamos:

```
select * from editoriales
left join libros
on editoriales.codigo=libros.codigoeditorial;
```

Un "left join" se usa para hacer coincidir registros en una tabla (izquierda) con otra tabla (derecha), pero, si un valor de la tabla de la izquierda no encuentra coincidencia en la tabla de la derecha, se genera una fila extra (una por cada valor no encontrado) con todos los campos seteados a "null".

Entonces, la sintaxis es la siguiente: se nombran ambas tablas, una a la izquierda del "join" y la otra a la derecha, y la condición para enlazarlas, es decir, el campo por el cual se combinarán, se establece luego de "on". Es importante la posición en que se colocan las tablas en un "left join", la tabla de la izquierda es la que se usa para localizar registros en la tabla de la derecha. Por lo tanto, estos "join" no son iguales:

```
select * from editoriales
left join libros
on editoriales.codigo=libros.codigoeditorial;
```

```
select * from libros
left join editoriales
on editoriales.codigo=libros.codigoeditorial;
```

La primera sentencia opera así: por cada valor de codigo de "editoriales" busca coincidencia en la tabla "libros", si no encuentra coincidencia para algún valor, genera una fila seteada a "null".

La segunda sentencia opera de modo inverso: por cada valor de "codigoeditorial" de "libros" busca coincidencia en la tabla "editoriales", si no encuentra coincidencia, setea la fila a "null".

Usando registros de la tabla de la izquierda se encuentran registros en la tabla de la derecha.

Luego del "on" se especifican los campos que se asociarán; no se deben colocar condiciones en la parte "on" para restringir registros que deberían estar en el resultado, para ello hay que usar la cláusula "where".

Un "left join" puede tener cláusula "where" que restrinja el resultado de la consulta considerando solamente los registros que encuentran coincidencia en la tabla de la derecha:

```
select e.nombre,l.titulo
from editoriales as e
left join libros as l
on e.codigo=l.codigoeditorial
where l.codigoeditorial is not null;  para que no salgan nulos en la segunda tabla
```

El anterior "left join" muestra los valores de la tabla "editoriales" que están presentes en la tabla de la derecha ("libros").

También podemos mostrar las editoriales que no están presentes en "libros":

```
select e.nombre,l.titulo from editoriales as e
left join libros as l
on e.codigo=l.codigoeditorial
where l.codigoeditorial is null;
```

El anterior "left join" muestra los valores de la tabla "editoriales" que no encuentran correspondencia en la tabla de la derecha, "libros".

Problema:

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1- Elimine las tablas "clientes" y "provincias", si existen:
`drop table if exists clientes, provincias;`

2- Créelas con las siguientes estructuras:

```
create table clientes (
  codigo int unsigned auto_increment,
  nombre varchar(30) not null,
  domicilio varchar(30),
  ciudad varchar(20),
  codigoProvincia tinyint unsigned,
  telefono varchar(11),
  primary key(codigo)
);

create table provincias(
  codigo tinyint unsigned auto_increment,
  nombre varchar(20),
  primary key (codigo)
);
```

3- Ingrese algunos registros para ambas tablas:
`insert into provincias (nombre) values('Cordoba');`

```

insert into provincias (nombre) values('Santa Fe');
insert into provincias (nombre) values('Corrientes');
insert into provincias (nombre) values('Misiones');
insert into provincias (nombre) values('Salta');
insert into provincias (nombre) values('Buenos Aires');
insert into provincias (nombre) values('Neuquen');

insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Lopez Marcos', 'Colon 111', 'Córdoba',1,'null');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Perez Ana', 'San Martin 222', 'Cruz del Eje',1,'4578585');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Garcia Juan', 'Rivadavia 333', 'Villa Maria',1,'4578445');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Perez Luis', 'Sarmiento 444', 'Rosario',2,null);
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Pereyra Lucas', 'San Martin 555', 'Cruz del Eje',1,'4253685');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Gomez Ines', 'San Martin 666', 'Santa Fe',2,'0345252525');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Torres Fabiola', 'Alem 777', 'Villa del Rosario',1,'4554455');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Lopez Carlos', 'Irigoyen 888', 'Cruz del Eje',1,null);
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Ramos Betina', 'San Martin 999', 'Cordoba',1,'4223366');
insert into clientes (nombre,domicilio,ciudad,codigoProvincia,telefono)
values ('Lopez Lucas', 'San Martin 1010', 'Posadas',4,'0457858745');

```

4- Queremos saber de qué provincias no tenemos clientes:

5- Queremos saber de qué provincias si tenemos clientes, sin repetir el nombre de la provincia:

6- Omita la referencia a las tablas en la condición "on" para verificar que la sentencia no se ejecuta porque el nombre del campo "codigo" es ambiguo (ambas tablas lo tienen):