

Queries de ejemplo con varias tablas

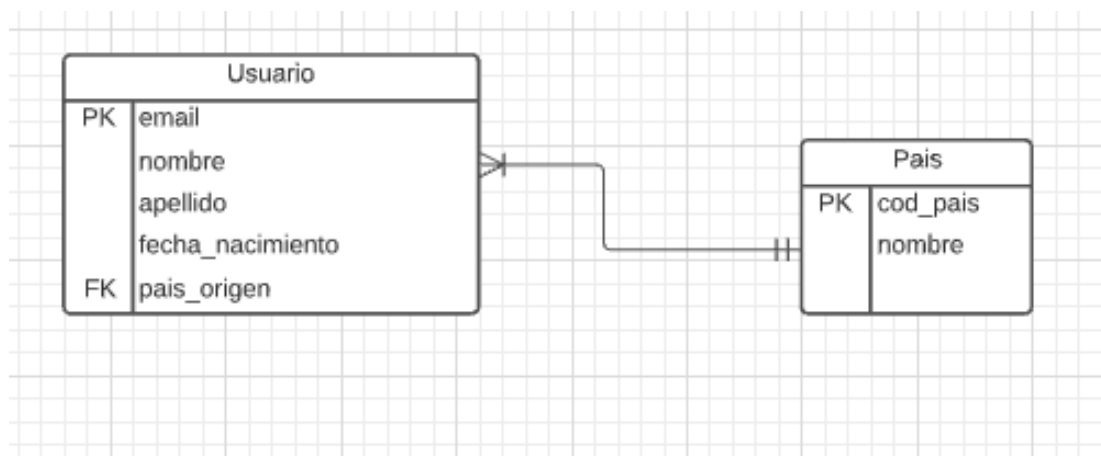
Con el fin de mejorar nuestras consultas vamos a requerir combinar varias tablas en la misma, para eso será necesario entender cómo se relacionan y qué debemos agregar en la sentencia SELECT para poder hacerlo.

Relacionando Tablas:

Antes de ver qué agregar en la sentencia debemos tener en cuenta parte de lo que vimos en el módulo referido a las claves primarias y foráneas.

Como podemos ver en el siguiente diagrama sobre las tablas Usuario y País, las mismas cuentan con una relación que en el DER lo explicamos por medio de la línea que las une y el dibujo en los extremos que nos marca la cardinalidad. Pero al llevarlo a nuestra base de datos, debemos recordar que esa relación se encuentra en la clave primaria de una tabla y la clave foránea de otra que guarda en ese atributo la clave primaria de la primera.

Si pusiéramos el ejemplo, suponiendo que el código país de Argentina es 1, sabríamos que todos los usuarios cuyo país de origen sea Argentina tendrán en el atributo referencial (clave foránea) pais_origen como valor 1. Y de esta forma podremos relacionar los registros que tenga el valor 1 con el registro en país con clave primaria 1, Argentina.



Junta natural (Inner Join):

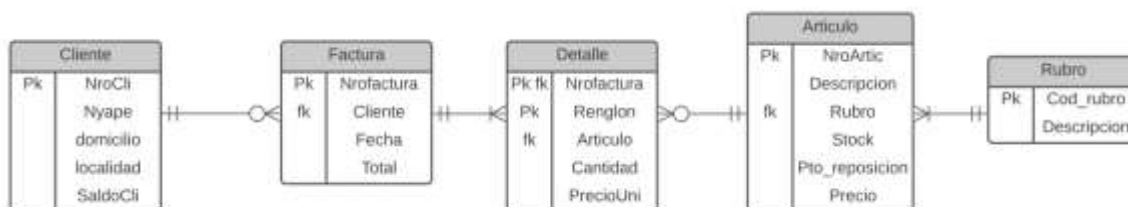
Cuando una consulta involucra varias tablas que están relacionadas mediante columnas comunes (FK y PK) se puede emplear la operación JOIN en la cláusula FROM.

Existen varias formas de juntar tablas mediante las columnas comunes, la más usada es:

INNER JOIN: devuelve solo las filas que contienen en sus columnas comunes el mismo valor.

Para entenderlo de mayor forma vamos a referirnos a un ejemplo puntual.

Supongamos que tengo una base de datos de venta de artículos representada por el siguiente diagrama:

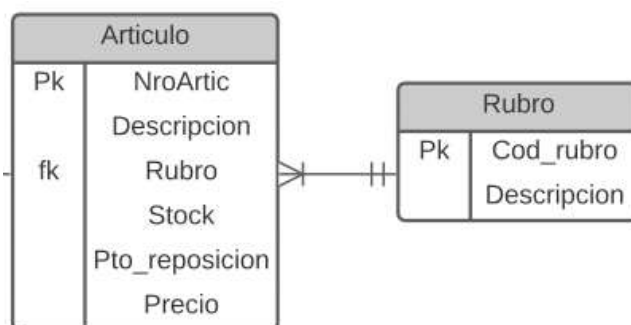


Ejemplo 1

Nos solicitan un informe que permita informar el nombre de los artículos a la venta, con el respectivo rubro al que pertenece. El listado debe tener número de artículo, descripción, precio y rubro (descripción del rubro).

Para solucionarlo debemos comprender cuál será el dominio de mi consulta, es decir, qué tablas están involucradas para poder mostrar los datos requeridos.

Como podemos ver en este caso vamos a necesitar Artículo y Rubro, dado lo que nos solicita el enunciado.



Cuando llegamos a este punto debemos saber por medio de qué atributos se relacionan las tablas, en este caso podemos ver qué artículo cuenta con un atributo referencial (clave foránea) Rubro, que contiene la clave primaria de la tabla Rubro. Ya podríamos ponerlo en nuestra consulta, conformando en pasos nuestro query.

Paso 1) Completo el FROM poniendo ambas tablas y entre medio la palabra reservada *inner join*. Es conveniente que agreguemos un alias a las tablas para evitar tener que escribirlas continuamente al referenciarlas con sus atributos. Alcanza con poner una o más letras luego de cada nombre, en nuestro caso “a” para Artículo y “r” para Rubro.

FROM Artículo a inner join Rubro r

Antes de terminar el FROM es necesario que aclaremos por medio de que atributos las tablas se referencian. Usamos la palabra reservada ON para indicar lo que es llamado “Condición de Junta”. Basta con agregar los atributo de las tablas utilizando el signo “=” entre medio (se suele poner esta parte entre paréntesis). Quedaría de esta forma en nuestro ejemplo.

FROM Artículo a inner join Rubro r ON (a.Rubro = r.Cod_rubro)

Paso 2) Verificamos si debemos agregar el WHERE para alguna condición a nivel de registros. En nuestro ejemplo no tenemos ninguna así que WHERE no aparecerá.

Paso 3) Completo el SELECT incorporando las columnas, funciones o cálculos solicitados en nuestro ejemplo son solo columnas. Y nos queda de la siguiente forma:

SELECT a.NroArtic, a.Descripcion, a.precio, r.Descripcion
FROM Artículo a inner join Rubro r ON (a.Rubro = r.Cod_rubro);

Podemos renombrar las columnas para que queden tal y como se solicitaron en el enunciado.

SELECT a.NroArtic as "Numero de Artículo",
a.Descripcion as "Descripcion Artículo",

a.precio as "Precio Articulo",

r.Descripcion as "Descripción Rubro"

FROM Articulo a inner join Rubro r ON (a.Rubro = r.Cod_rubro);

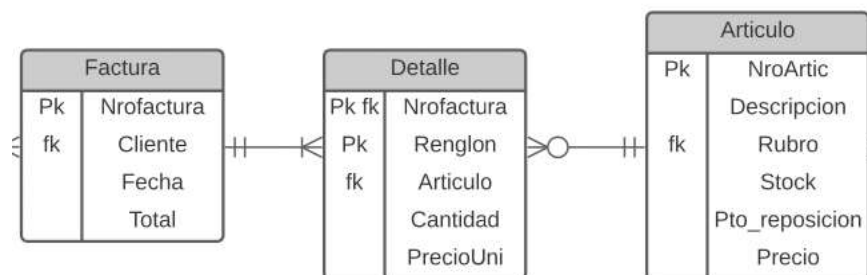
Y esto sería un ejemplo de lo que nos podría retornar nuestra consulta:

Results		Messages		
	Numero de Articulo	Descripción Articulo	Precio Articulo	Descripción Rubro
1	4000	Destornillador	300.00	Herramienta
2	4010	Pinza	700.00	Herramienta
3	4020	Tenaza	810.00	Herramienta
4	4030	Martillo	410.00	Herramienta
5	4040	Moladora	3000.00	Herramienta Eléctri...
6	4050	Clavo	8.00	Clavos
7	4060	Tuerca	7.70	Tornillos
8	4070	Tornillo	11.00	Tornillos
9	4080	Soldador	900.00	Herramienta Eléctri...
10	4090	Enchufe	40.00	Articulos de Electr...
11	4100	Cable	78.00	Articulos de Electr...
12	4110	Clavo en L	5.00	Clavos

Junta natural (Inner Join) con más de dos tablas:

Como pudimos ver, son consultas que nos van a traer mucha más información y lo único necesario es poder identificar por medio de qué atributos se relacionan las tablas. Puede ocurrir que una tabla intermedia sea quien relacione otras dos tablas. Se le suele llamar tablas “Asociativas” y su principal característica es que cuentan con las claves foráneas que apuntan a las claves primarias de las tablas que relacionan.

En el DER propuesto podemos identificar una tabla “Asociativa” entre las siguientes tablas:



La tabla **Detalle** tiene las claves foráneas Nrofactura y Articulo que guardan las claves primarias de las tablas **Factura** y **Articulo** respectivamente.

Propongamos un ejercicio utilizando estas entidades.

Ejemplo 2

Supongamos que se requiere una lista de los artículos vendidos en marzo del 2021, los datos a mostrar serían: Número de factura, Fecha de facturación, Descripción del artículo vendido, cantidad y precio unitario (en la venta).

Tal como hicimos en el ejemplo 1, comenzaremos por el FROM, pero antes explicaremos que ocurre cuando queremos unir más de una tabla en una consulta.

Inner Join de varias tablas: El *inner join* nos obliga a utilizar 2 tablas una a la izquierda y otra a la derecha con la correspondiente condición de junta, pero esto se da solo en el primer *join*. A partir del segundo *join* solo necesitamos agregar una tabla a la derecha y que la tabla a la cual hacemos referencia en la condición de junta ya haya sido enunciada previamente (en otro *join* dentro de la misma consulta).

Dicho esto, podemos sacar como conclusión que la cantidad de JOIN necesarios será igual a la cantidad de tablas a unir menos uno.

Cantidad de Join = Cantidad de Tablas - 1

Regresando el ejemplo nuestro paso 1 seria de la siguiente forma:

```
FROM Factura f INNER JOIN Detalle d ON (f.Nrofactura = d.Nrofactura)
      INNER JOIN Articulo a ON (d.Articulo = a.NroArtic)
```

Paso 2) Revisamos si tenemos condiciones a nivel de registros en el enunciado que debamos incluir en el WHERE. Nuestra clave está en esta parte del enunciado "...vendidos en enero del 2021...". Debemos poder condicionar las facturas para que sean solo las que posean fecha entre el 1 de marzo del 2021 y el 31 de marzo del 2021.

```
FROM Factura f INNER JOIN Detalle d ON (f.Nrofactura = d.Nrofactura)
      INNER JOIN Articulo a ON (d.Articulo = a.NroArtic)
WHERE f.fecha >= '2021/03/01'
AND f.fecha <= '2021/03/31'
```

Y nuestro último paso será encontrar los datos que debemos mostrar, que se encuentran al final del enunciado. Los incorporamos quedando nuestra consulta de la siguiente forma:

```
SELECT f.nrofactura as "Numero Factura", f.FECHA as "Fecha Factura",
      a.DESCRIPCION as "Descripción Articulo", d.CANTIDAD,
      d.PRECIOUNI as "Precio Unitario"
FROM Factura f INNER JOIN Detalle d ON (f.Nrofactura = d.Nrofactura)
      INNER JOIN Articulo a ON (d.Articulo = a.NroArtic)
WHERE f.fecha >= '2021/03/01'
AND f.fecha <= '2021/03/31'
```

Y como resultado obtenemos algo parecido a esto:

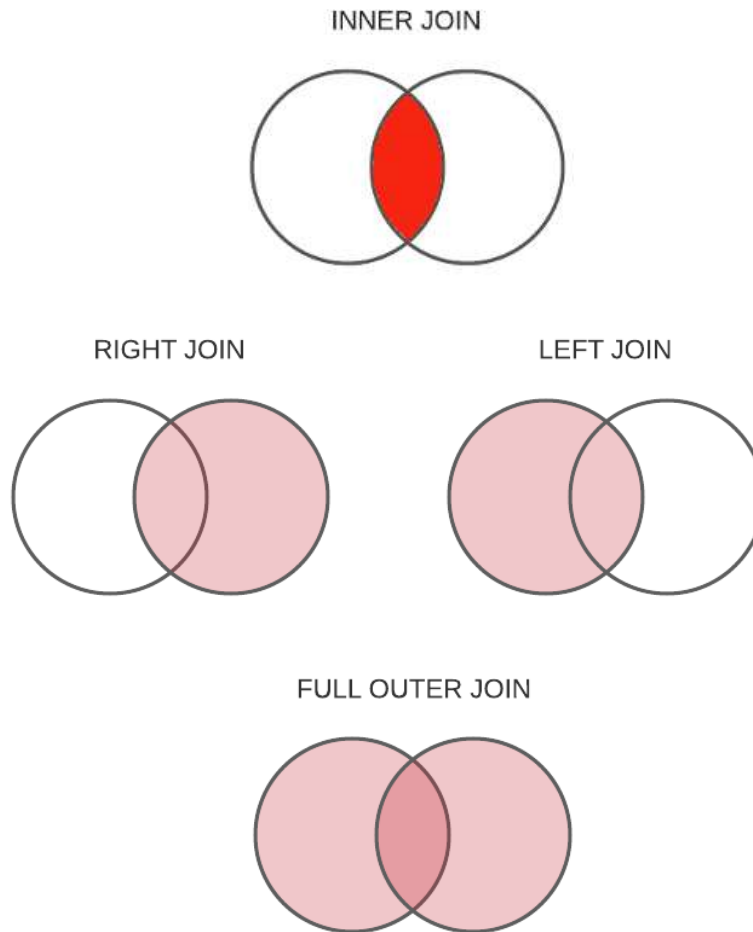
Results		Messages			
	Numero Factura	Fecha Factura	Descripción Artículo	CANTIDAD	Precio Unitario
1	1061	2021-03-12	Moladora	5	3000.00
2	1062	2021-03-12	Clavo	2	8.00
3	1062	2021-03-12	Tuerca	3	7.00
4	1063	2021-03-14	Tuerca	1	7.00
5	1064	2021-03-14	Tornillo	5	10.00
6	1064	2021-03-14	Soldador	3	900.00
7	1064	2021-03-14	Enchufe	3	40.00
8	1064	2021-03-14	Cable	3	78.00
9	1064	2021-03-14	Clavo en L	5	5.00
10	1064	2021-03-14	Destornillador	1	300.00
11	1065	2021-03-15	Soldador	2	900.00
12	1065	2021-03-15	Enchufe	6	40.00
13	1066	2021-03-15	Enchufe	3	40.00
14	1066	2021-03-15	Cable	6	78.00
15	1066	2021-03-15	Clavo en L	4	5.00
16	1067	2021-03-17	Cable	5	78.00
17	1067	2021-03-17	Clavo en L	2	5.00

Outer Join:

Al realizar un Inner Join, las filas de cualquiera de las tablas que no coinciden en la otra tabla, no cumplen con la condición de junta, no se devuelven. En un Outer Join, se pueden devolver filas no coincidentes en una o ambas tablas. Los tipos de outer join son:

- Left Join: devuelve solo filas no coincidentes de la tabla de la izquierda.
- Right Join: devuelve solo filas no coincidentes de la tabla de la derecha.
- Full Outer Join: devuelve filas no coincidentes de ambas tablas.

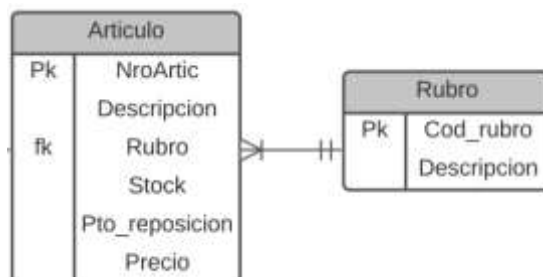
Para poder entenderlo podemos verlos con conjuntos, como si cada conjunto representará las filas de las tablas a unir:



Veamos como un ejemplo como podemos darle uso a los Outers Join.

Ejemplo 3

Dado el siguiente diagrama se requiere un listado todos los rubros, código de rubro y descripción, y sus correspondientes artículos, número de artículo y descripción, en caso que tuviera.



Arrancamos con el paso 1 conformando el FROM con las tablas que están involucradas, quedándonos de la siguiente forma y teniendo en cuenta que se requieren todos los registros que se encuentren en RUBRO aun cuando no cumplan la condición de junta.

```
FROM Rubro r RIGHT JOIN Articulo a ON (r.cod_rubro = a.Rubro)
```

En el paso 2 validamos si tenemos alguna condición que debamos agregar en el WHERE pero en este caso no tenemos ninguna así que pasamos al paso 3 donde validamos los datos que se solicitaron mostrar.

```
SELECT r.COD_RUBRO as "Código de Rubro",
       r.Descripcion as "Descripción Rubro",
       a.NroArtic as "Numero de Articulo",
       a.Descripcion as "Descripcion Articulo"
```

```
FROM Rubro r LEFT JOIN Articulo a ON (r.cod_rubro = a.Rubro)
```

Results		Messages		
	Código de Rubro	Descripción Rubro	Numero de Articulo	Descripcion Articulo
1	1	Herramienta	4000	Destornillador
2	1	Herramienta	4010	Pinza
3	1	Herramienta	4020	Tenaza
4	1	Herramienta	4030	Martillo
5	2	Tornillos	4060	Tuerca
6	2	Tornillos	4070	Tornillo
7	3	Clavos	4050	Clavo
8	3	Clavos	4110	Clavo en L
9	4	Herramienta Eléctrica	4040	Moladora
10	4	Herramienta Eléctrica	4080	Soldador
11	5	Artículos de Electricidad	4090	Enchufe
12	5	Artículos de Electricidad	4100	Cable
13	9	Electrodomesticos	NULL	NULL
14	10	Cotillon	NULL	NULL

Como se puede ver en los últimos 3 registros se completa con NULL las columnas que refieren a Artículo, y esto se debe a que los rubros “Electrodomésticos” y “Cotillón” no tienen aún ningún registro en la tabla ARTICULO que se relacione. Es decir, que no hay ningún registro con el valor 9 o 10 en el atributo referencial Rubro de la tabla Artículo.

Se podría haber aplicado el RIGHT JOIN en vez del LEFT JOIN si invertimos la posición de las tablas Rubro y Artículo.