

# Consultas de múltiples tablas y uniones

Primera parte

# Trabajar con varias tablas

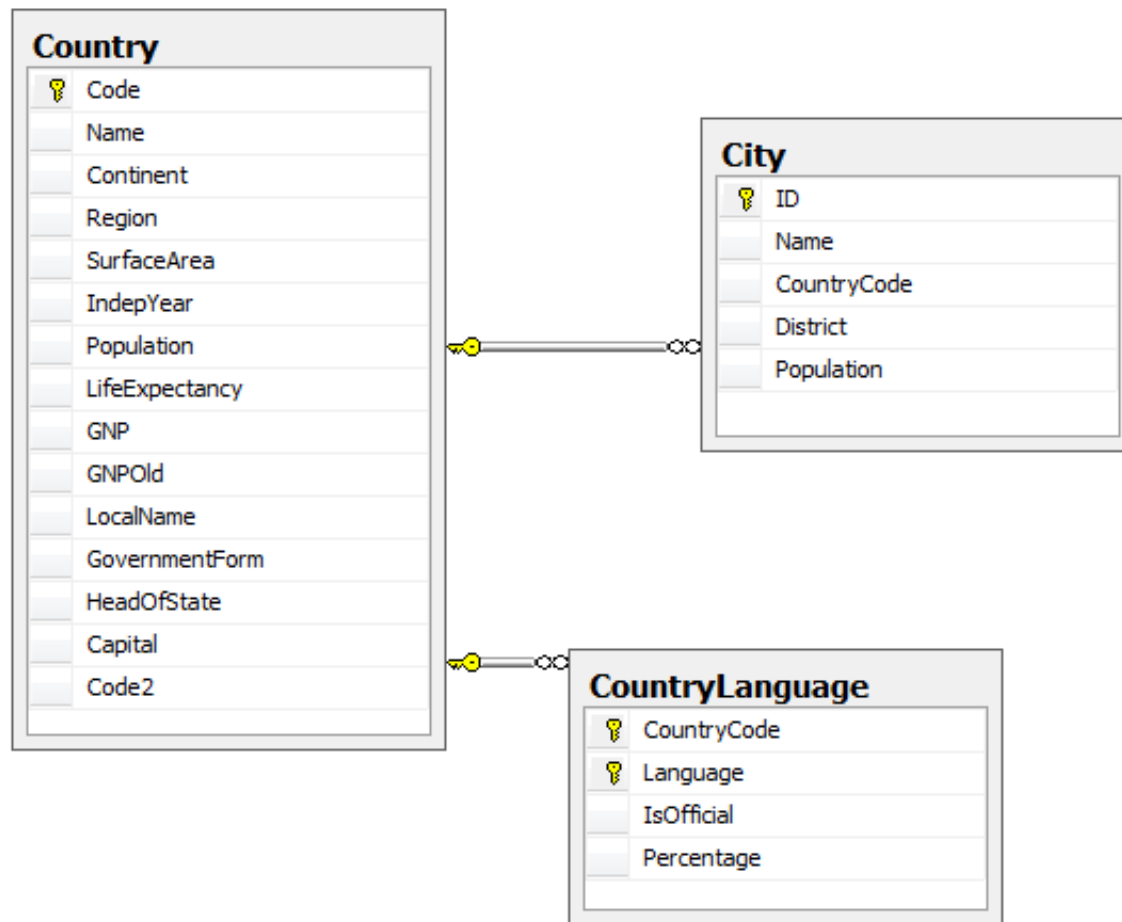
Generalmente se trabaja con varias tablas.

Se estructura la información de manera que evitemos la repetición de información almacenada y teniendo en cuenta que un campo no puede contener múltiples informaciones para un único registro.

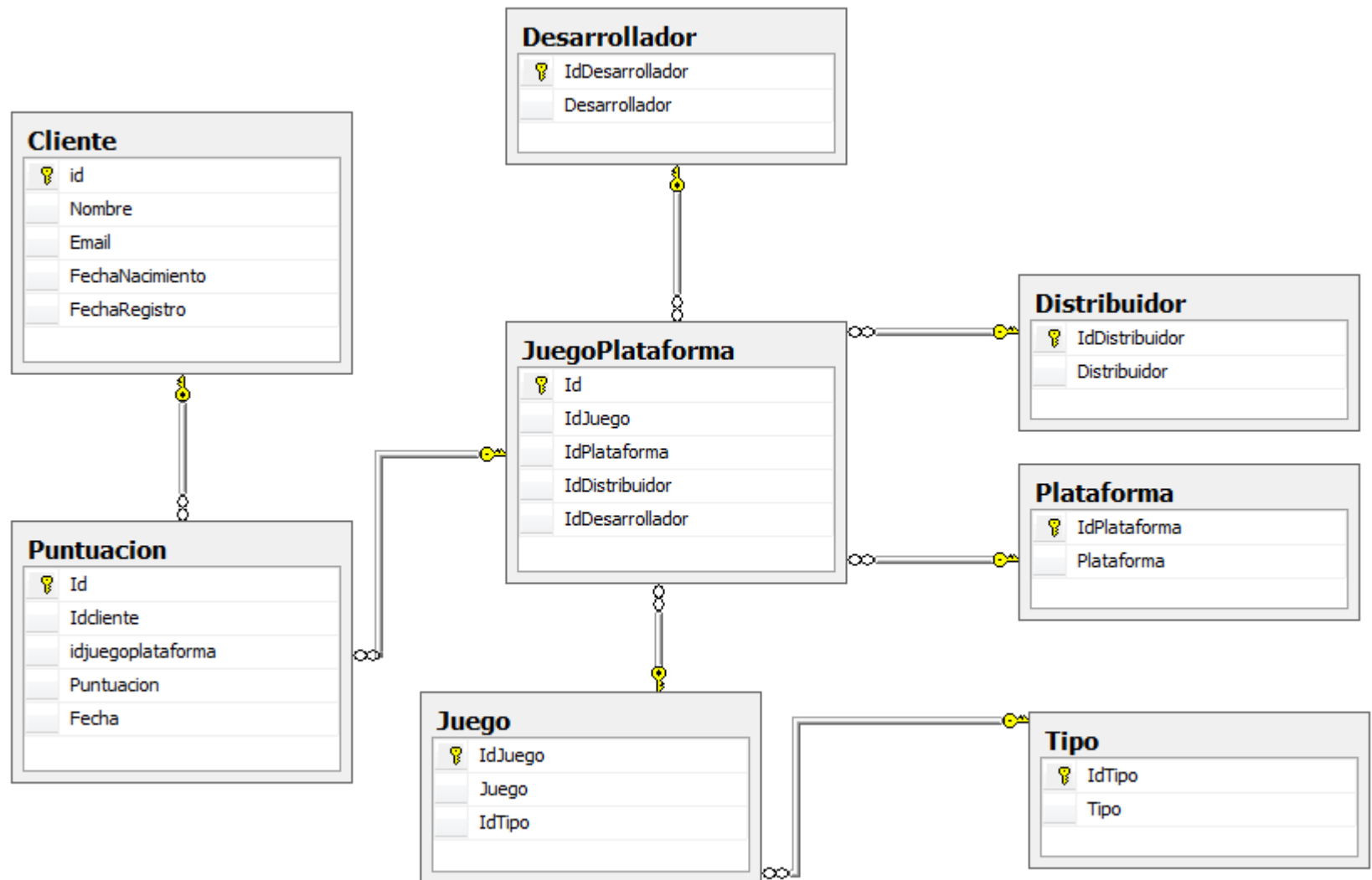
Realmente se trata de definir la estructura de la Base de Datos basada en la Normalización.

Teniendo en cuenta esto, tendremos que aprender a reconstruir la información a partir de ese conjunto de tablas.

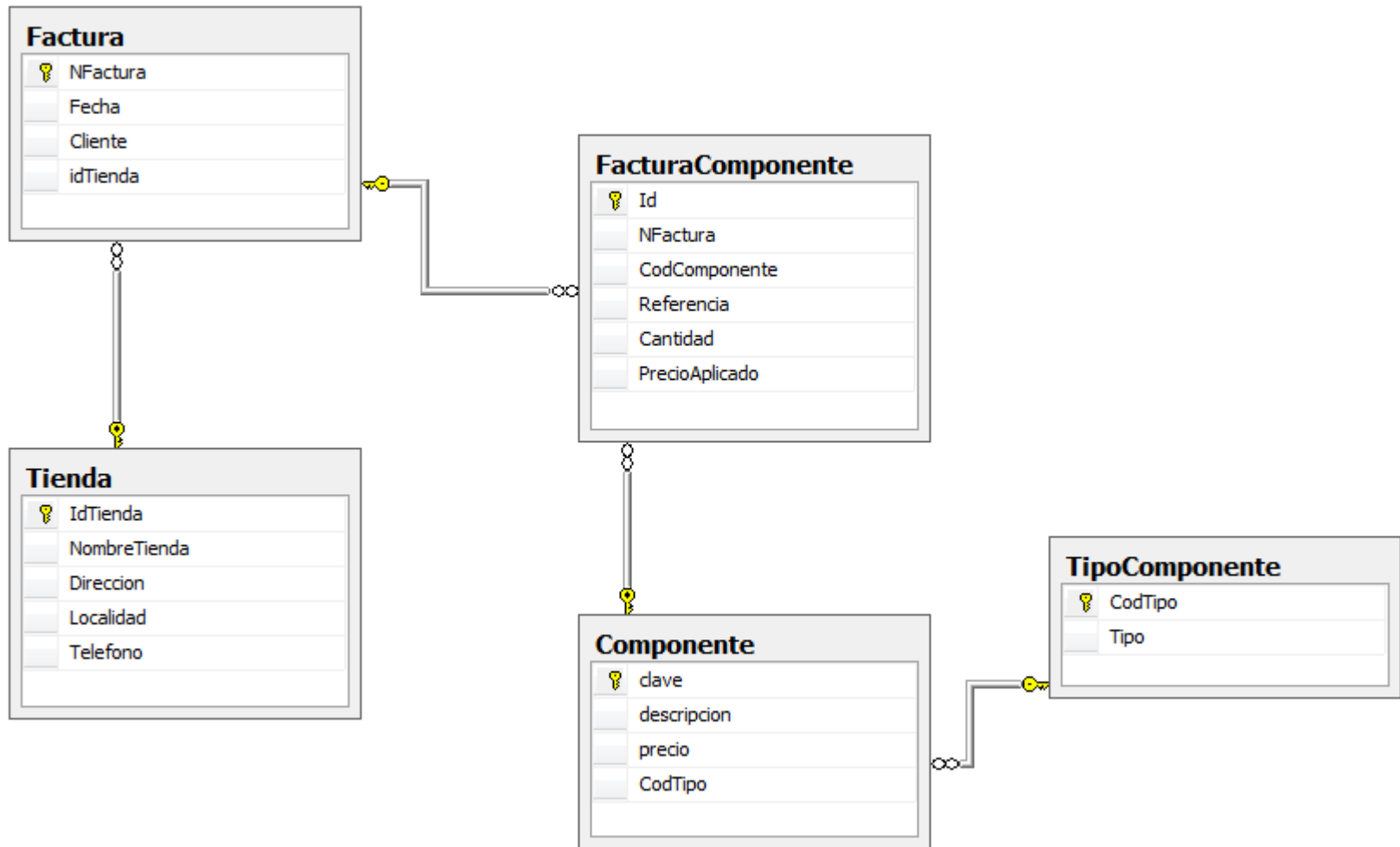
# Mundo



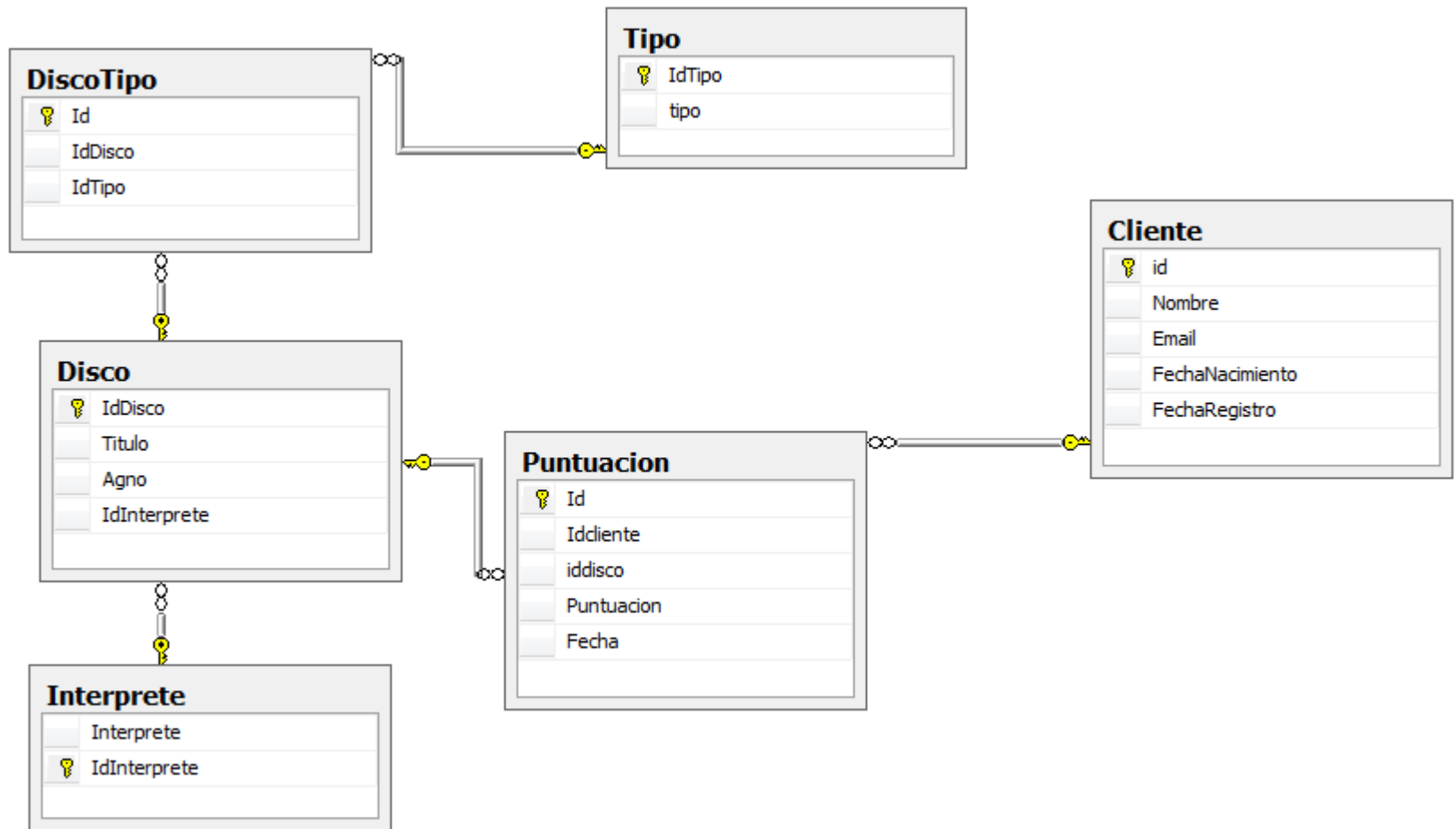
# Videojuegos



# Empresas informáticas



# Discos



Intentaremos ver las ciudades del país Spain.  
Probaremos con...

```
use mundo;
```

```
select p.Name as País, c.Name as Ciudad  
from country as p ,city as c  
where p.Name='Spain'|
```

	País	Ciudad
1	Spain	Kabul
2	Spain	Qandahar
3	Spain	Herat
4	Spain	Mazar-e-Sharif
5	Spain	Amsterdam
6	Spain	Rotterdam
7	Spain	Haag
8	Spain	Utrecht
9	Spain	Eindhoven

Mundo	00:00:01	4079 filas
-------	----------	------------

**No es lo esperado...**



Si colocamos los nombres de tablas separados con comas lo que tendremos es un producto cartesiano de las tablas.  
Es decir, por cada registro de la primera tabla tantos registros como tiene la segunda tabla.

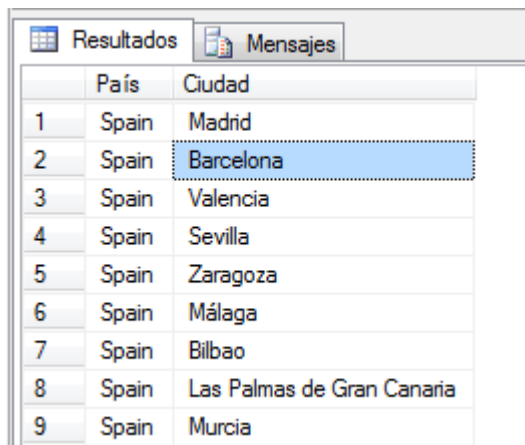
En el caso del ejemplo cada país con todas las ciudades.

La primera solución disponible es filtrar ese producto cartesiano mediante un where a través del campo que identifica el país en la tabla City.

CountryCode es el código del país en la tabla City.

```
use mundo;
```

```
select p.Name as País, c.Name as Ciudad  
from country as p ,city as c  
where p.Name='Spain'  
and (p.Code=c.CountryCode);
```



	País	Ciudad
1	Spain	Madrid
2	Spain	Barcelona
3	Spain	Valencia
4	Spain	Sevilla
5	Spain	Zaragoza
6	Spain	Málaga
7	Spain	Bilbao
8	Spain	Las Palmas de Gran Canaria
9	Spain	Murcia

0 | 59 filas

Es una solución factible  
aunque normalmente **poco  
eficiente.**

El SQL tiene cláusulas  
específicas para efectuar el  
cruce de tablas.

# INNER JOIN

combinaciones internas

```

use mundo;
go

select p.Name as País, c.Name as Ciudad
from country as p
inner join city as c
    on p.Code=c.CountryCode
where p.Name='Spain';
go

```

	País	Ciudad
1	Spain	Madrid
2	Spain	Barcelona
3	Spain	Valencia
4	Spain	Sevilla
5	Spain	Zaragoza
6	Spain	Málaga

0:00:00 | 59 filas

Muestra las ciudades de cada país, filtrados sólo para Spain

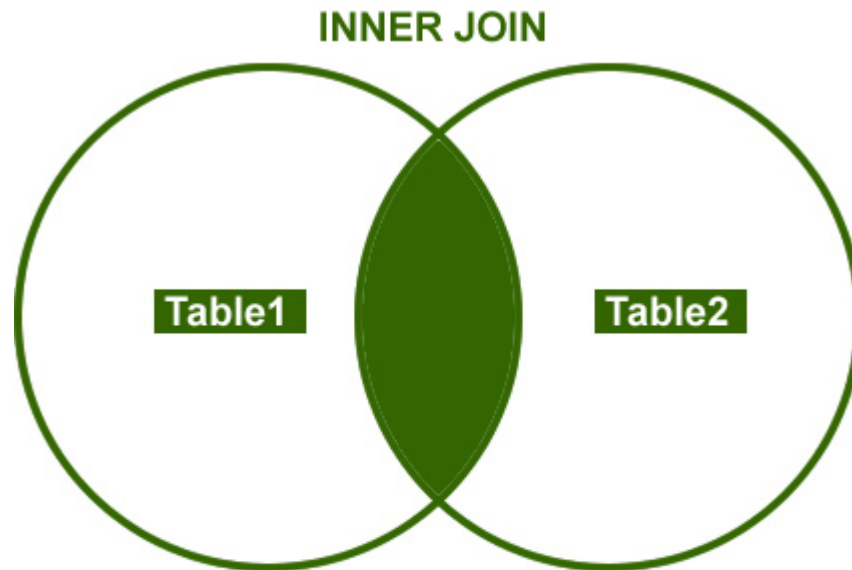
# Formato de la sentencia INNER JOIN

Especifica que se devuelvan todos los pares de filas coincidentes. Rechaza las filas no coincidentes de las dos tablas.  
Se puede poner sólo JOIN.

Dentro de la cláusula FROM

```
Tabla1  
INNER JOIN Tabla2  
on Campotabla1=Campotabla2
```

Teniendo en cuenta que los campos de enlace son los que referencian la misma información en ambas tablas.



```
SELECT *  
FROM Table1 t1  
INNER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

(C) <http://blog.SQLAuthority.com>

# Algoritmo INNER JOIN

```
Por cada tupla de la relación R :r
  Por cada tupla de la relación S :s
    Si para la tupla <r,s>
      coinciden los campos
      r.idc = s.idc
    entonces
      agregar <r,s> a la salida
```

## Tabla1

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

## Tabla2

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo

```
--inner join
select t1.id,t1.valor1,t2.id,t2.valor2
from tabla1 as t1
inner join tabla2 as t2
    on t1.id=t2.id;
go
```

	id	valor1	id	valor2
1	1	Primero	1	Primero
2	2	Segundo	2	Segundo
3	3	Tercero	3	Tercero

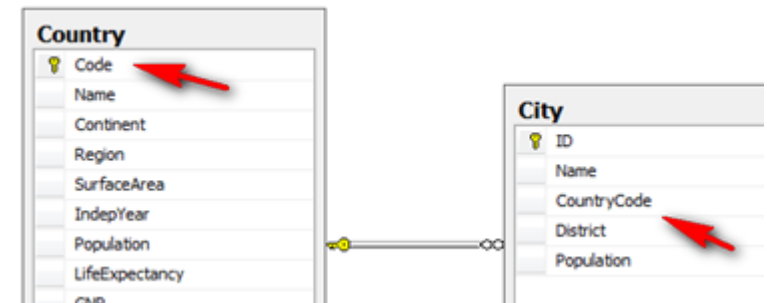


```

select p.Code,p.Name as País, c.CountryCode,c.Name as Ciudad
from country as p
inner join city as c
    on p.Code=c.CountryCode
go

```

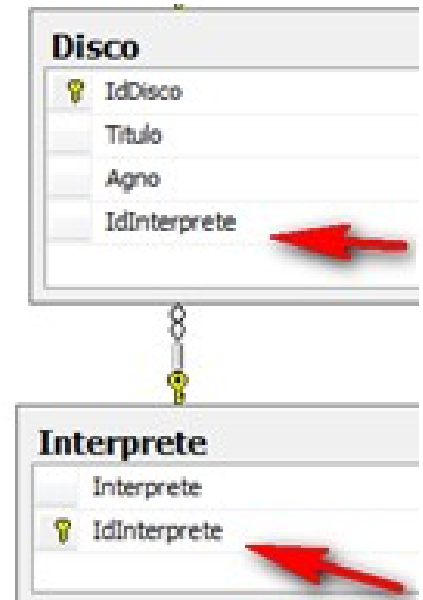
	Code	País	CountryCode	Ciudad
199	BOL	Bolivia	BOL	Potosí
200	BOL	Bolivia	BOL	Tanja
201	BIH	Bosnia an...	BIH	Sarajevo
202	BIH	Bosnia an...	BIH	Banja Luka
203	BIH	Bosnia an...	BIH	Zenica
204	BWA	Botswana	BWA	Gaborone
205	BWA	Botswana	BWA	Francistown
206	BRA	Brazil	BRA	São Paulo
207	BRA	Brazil	BRA	Rio de Janeiro
208	BRA	Brazil	BRA	Salvador
209	BRA	Brazil	BRA	Belo Horizonte
210	BRA	Brazil	BRA	Fortaleza
211	BRA	Brazil	BRA	Brasília
212	BRA	Brazil	BRA	Curitiba



## Mostrar título de disco y su intérprete

```
use discos;  
go  
  
|select titulo,interprete  
|from disco as d  
|inner join interprete as i|  
|    on d.IdInterprete=i.IdInterprete;  
go
```


Resultados			Mensajes
	titulo	interprete	
1	Dancing Queen	ABBA	
2	Ring My Bell	Anita Ward	
3	Yes Sir, I Can Boogie	Baccara	
4	Copacabana	Bary Manilow	
5	You're The First, My Last, My Everything	Bary White	
6	Love's Theme	Bary White; Love Unlimited Orchestra	
7	Saturday Night Fever	Bee Gees	
8	You Should Be Dancing	Bee Gees	
9	Stayin' Alive	Bee Gees	
10	live Talkin'	Bee Gees	



natural join

Funciona como el INNER JOIN, lo único que no se expresa la relación de campos iguales, ya que supone que son los que se llaman igual en las dos tablas.

**NO está admitido en SQL Server, pero funciona en MySql.**



ID	Valor1
1	Primero
2	Segundo
3	Tercero
4	Cuarto
5	Quinto

**tabla1**

ID	Valor2
1	Primero
2	Segundo
3	Tercero
6	Sexto
7	S?ptimo
8	Octavo

**tabla2**

```
SELECT t1.id, t1.valor, t2.id, t2.valor  
FROM tabla1 AS t1  
NATURAL JOIN tabla2 AS t2  
LIMIT 0 , 30
```

id	valor1	id	valor2
1	Primero	1	Primero
2	Segundo	2	Segundo
3	Tercero	3	Tercero

left outer join

```
select c.id,Nombre,p.id,p.idcliente,puntuacion
from cliente as c
left join puntuacion as p
      on c.id=p.idcliente
```

	id	Nombre	id	idcliente	puntuacion
1	1	Raymond Velasquez	9	1	7
2	1	Raymond Velasquez	50	1	8
3	2	Memill Leblanc	6	2	6
4	3	Dean Santos	NULL	NULL	NULL
5	4	Ryan Hooper	NULL	NULL	NULL
6	5	John Mcknight	51	5	9
7	6	John Watson	53	6	3
8	7	Lucian Pugh	48	7	6
9	7	Lucian Pugh	41	7	10

incluye registros de la primera tabla que no tienen coincidencia en la segunda, poniendo NULL en los campos de la segunda tabla

...

# Formato de la sentencia LEFT OUTER JOIN

Especifica que se devuelvan todos los pares de filas coincidentes. Incluye las filas no coincidentes de la primera tabla, rellenando a null los atributos correspondientes a la otra tabla.

Se puede poner sólo LEFT JOIN.

Dentro de la cláusula FROM

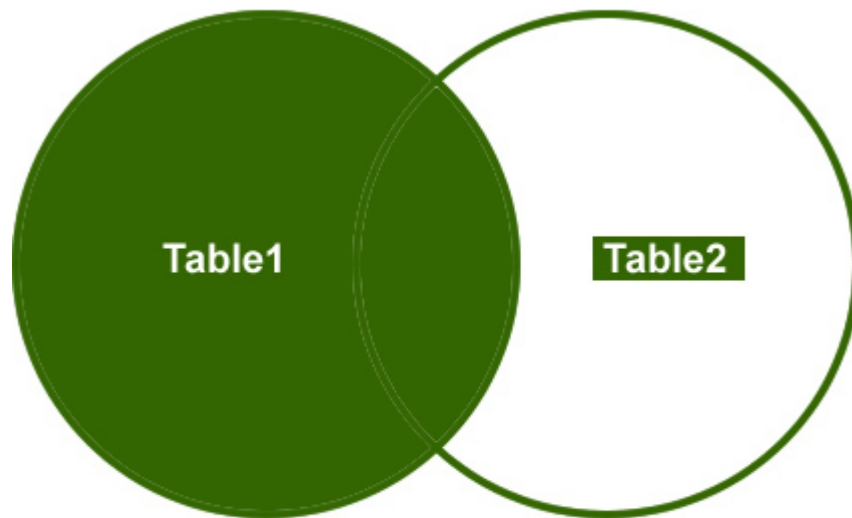
Tabla1

LEFT OUTER JOIN Tabla2

on Campotabla1=Campotabla2

Teniendo en cuenta que los campos de enlace son los que referencian la misma información en ambas tablas.

## LEFT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
LEFT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

(C) <http://blog.SQLAuthority.com>



# Algoritmo LEFT OUTER JOIN

```
Por cada tupla de la relación R :r
  Por cada tupla de la relación S :s
    Si para la tupla <r,s>
      coinciden los campos
      r.idc = s.idc
    entonces
      agregar <r,s> a la salida
  Si no hay tuplas insertadas para r
  entonces
    agregar <r, null> a la salida
```

# Tabla1

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

```
--left outer join  
select t1.id,valor1,t2.id,valor2  
from tabla1 as t1  
left outer join tabla2 as t2  
on t1.id=t2.id;  
go
```

# Tabla2

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo

	id	valor1	id	valor2
1	1	Primero	1	Primero
2	2	Segundo	2	Segundo
3	3	Tercero	3	Tercero
4	4	Cuarto	NULL	NULL
5	5	Quinto	NULL	NULL

Los  
coincidentes

Los no coincidentes de la  
tabla de la izquierda

Para las no coincidentes con null en los campos de la otra tabla

También vale con **left join**

```
select t1.id, valor1, t2.id, valor2
from tabla1 as t1
left join tabla2 as t2
      on t1.id=t2.id;
go
```

```

select descripcion,nfactura
from componente as c
left outer join facturacomponente as fc
on c.clave=fc.CodComponente

```

Resultados		Mensajes
	descripcion	nfactura
1	ACER C120 ( PROYECTOR ) P/N EY.JE001.001	28
2	ACER ASPIRE M1935 INTEL I3 2130 / 4GB DDR3 / DISC...	5
3	ACER ASPIRE M1935 INTEL I5 2330S ( 2.80 GHZ ) / 4GB ...	NULL
4	ACER X111 ( PROYECTOR ) 2700 ANSI LUMENS P/N M...	NULL
5	ACER X1240 ( PROYECTOR ) 2700 LUMENS / 3D / XGA ...	2
6	ACER VERITON X2610G INTEL I5 2320 ( 3.00 GHZ ) / 4GB ...	NULL
7	ACER VERITON X2611G INTEL I3 3220 / 4GB DDR3 / DIS...	NULL
8	ACER ASPIRE XC100 AMD E1-1200 ( DUAL CORE ) ( 1.40 ...	NULL
9	ACER ASPIRE XC600 INTEL DUAL CORE G645/ 6GB DD...	NULL
10	ACER ASPIRE XC600 INTEL DUAL CORE G645 ( 2.90 GH...	34
11	ADAPTADOR DISCO DURO 2.5 A BAHIA 3.5 P/N 10.99.00	32

Ver sólo los artículos sin ventas...

```
select descripcion,nfactura
from componente as c
left outer join facturacomponente as fc
on c.clave=fc.CodComponente
where nfactura is null|
```

	descripcion	nfactura
1	ACER ASPIRE M1935 INTEL I5 2330S ( 2.80 GHZ ) / 4GB DD...	NULL
2	ACER X111 ( PROYECTOR ) 2700 ANSI LUMENS P/N MR.J...	NULL
3	ACER VERITON X2610G INTEL I5 2320 ( 3.00 GHZ ) / 4GB D...	NULL
4	ACER VERITON X2611G INTEL I3 3220 / 4GB DDR3 / DISCO...	NULL
5	ACER ASPIRE XC100 AMD E1-1200 ( DUAL CORE ) ( 1.40 GH...	NULL
6	ACER ASPIRE XC600 INTEL DUAL CORE G645/ 6GB DDR3 ...	NULL
7	ADAPTADOR DVI/VGA	NULL
8	ALTAVOCES LOGITECH LS11 2.0 ( 980-000046 )	NULL

...

right outer join

```
--right outer join  
]select nombre,puntuacion  
from puntuacion as p  
right outer join Cliente as c  
-   on c.id=p.Idcliente
```

	nombre	puntuacion
31	Raymond Henry	6
32	Oleg Park	NULL
33	Tad Fuentes	4
34	Tad Fuentes	6
35	Tad Fuentes	2
36	Tad Fuentes	8
37	Emmanuel Cox	6
38	Bruno Baird	6
39	Bruno Baird	8
40	Nehru Atkins	NULL
41	Jordan Tanner	NULL

incluye registros de la segunda tabla que no tienen coincidencia en la primera, poniendo NULL en los campos de la primera tabla

...

# Formato de la sentencia RIGHT OUTER JOIN

Especifica que se devuelvan todos los pares de filas coincidentes. Incluye las filas no coincidentes de la segundatabla, rellenando a null los atributos correspondientes a la otra tabla.

Se puede poner sólo RIGHT JOIN.

Dentro de la cláusula FROM

Tabla1

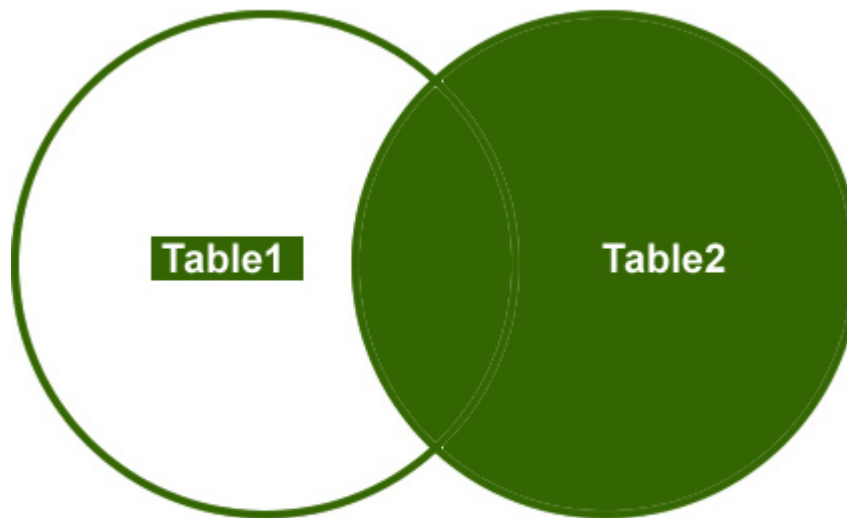
RIGHT OUTER JOIN Tabla2

on Campotabla1=Campotabla2

Teniendo en cuenta que los campos de enlace son los que referencian la misma información en ambas tablas.



## RIGHT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
RIGHT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

(C) <http://blog.SQLAuthority.com>

# Algoritmo RIGHT OUTER JOIN

```
Por cada tupla de la relación R :r
  Por cada tupla de la relación S :s
    Si para la tupla <r,s>
      coinciden los campos
      r.idc = s.idc
    entonces
      agregar <r,s> a la salida
Por cada tupla de la relación S:s
  Si no hay tuplas insertadas para s
  entonces
    agregar <null, s> a la salida
```

# Tabla1

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

```
--right outer join
select t1.id,valor1,t2.id,valor2
from tabla1 as t1
right outer join tabla2 as t2
    on t1.id=t2.id;
go
```

# Tabla2

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo

	id	valor1	id	valor2
1	1	Primero	1	Primero
2	2	Segundo	2	Segundo
3	3	Tercero	3	Tercero
4	NULL	NULL	6	Sexto
5	NULL	NULL	7	Séptimo
6	NULL	NULL	8	Octavo

Los  
coincidentes

Los no coincidentes de la  
tabla de la derecha

Para las no coincidentes con null en los campos de la otra tabla

También vale con **right join**

```
select t1.id,valor1,t2.id,valor2  
from tabla1 as t1  
right join tabla2 as t2  
    on t1.id=t2.id;  
go|
```

# Equivalencia LEFT JOIN – RIGHT JOIN

Para el caso de dos tablas podremos sustituir fácilmente el right join por el left join con sólo intercambiar el orden de las tablas en la cláusula from.

```
--right outer join
select descripcion,nfactura
from facturacomponente as fc
right outer join componente as c
on c.clave=fc.CodComponente
```

Resultados Mensajes		
	descripcion	nfactura
1	ACER C120 ( PROYECTOR ) P/N EY.JE001.001	28
2	ACER ASPIRE M1935 INTEL I3 2130 / 4GB DDR3 / DISC...	5
3	ACER ASPIRE M1935 INTEL I5 2330S ( 2.80 GHZ ) / 4GB ...	NULL
4	ACER X111 ( PROYECTOR ) 2700 ANSI LUMENS P/N M...	NULL
5	ACER X1240 ( PROYECTOR ) 2700 LUMENS / 3D / XGA ...	2
6	ACER VERITON X2610G INTEL I5 2320 ( 3.00 GHZ ) / 4GB ...	NULL
7	ACER VERITON X2611G INTEL I3 3220 / 4GB DDR3 / DIS...	NULL
8	ACER ASPIRE XC100 AMD E1-1200 ( DUAL CORE ) ( 1.40 ...	NULL
9	ACER ASPIRE XC600 INTEL DUAL CORE G645/ 6GB DD...	NULL
10	ACER ASPIRE XC600 INTEL DUAL CORE G645 ( 2.90 GH...	34
11	ADAPTADOR DISCO DURO 2.5 A BAHIA 3.5 P/N 10.99.00	32

Ver sólo los artículos sin ventas...

```
--right outer join
select descripcion,nfactura
from facturacomponente as fc
right outer join componente as c
    on c.clave=fc.CodComponente
where nfactura is null
```

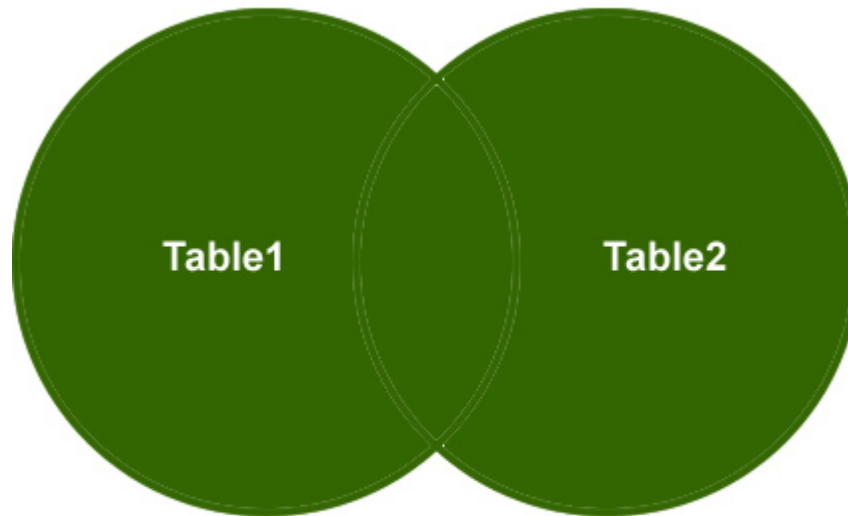
	descripcion	nfactura
1	ACER ASPIRE M1935 INTEL I5 2330S ( 2.80 GHZ ) / 4GB DD...	NULL
2	ACER X111 ( PROYECTOR ) 2700 ANSI LUMENS P/N MR.J...	NULL
3	ACER VERITON X2610G INTEL I5 2320 ( 3.00 GHZ ) / 4GB D...	NULL
4	ACER VERITON X2611G INTEL I3 3220 / 4GB DDR3 / DISCO...	NULL
5	ACER ASPIRE XC100 AMD E1-1200 ( DUAL CORE ) ( 1.40 GH...	NULL
6	ACER ASPIRE XC600 INTEL DUAL CORE G645/ 6GB DDR3 ...	NULL
7	ADAPTADOR DVI/VGA	NULL
8	ALTAVOCES LOGITECH LS11 2.0 ( 980-000046 )	NULL

...

full outer join



## FULL OUTER JOIN



```
SELECT *  
FROM Table1 t1  
FULL OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

(C) <http://blog.SQLAuthority.com>

# Tabla1

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

```
--full join
select t1.id,valor1,t2.id,valor2
from tabla1 as t1
full join tabla2 as t2
    on t1.id=t2.id;
go
```

# Tabla2

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo

	id	valor1	id	valor2
1	1	Primero	1	Primero
2	2	Segundo	2	Segundo
3	3	Tercero	3	Tercero
4	4	Cuarto	NULL	NULL
5	5	Quinto	NULL	NULL
6	NULL	NULL	6	Sexto
7	NULL	NULL	7	Séptimo
8	NULL	NULL	8	Octavo

Los  
coincidentes

Los no coincidentes de la  
tabla de la izquierda

Los no coincidentes de la  
tabla de la derecha

Para las no coincidentes con null en los campos de la otra tabla

# Algoritmo FULL JOIN

```
Por cada tupla de la relación R :r
  Por cada tupla de la relación S :s
    Si para la tupla <r,s>
      coinciden los campos
      r.idc = s.idc
    entonces
      agregar <r,s> a la salida
Por cada tupla de la relación S:s
  Si no hay tuplas insertadas para s
  entonces
    agregar <null,s> a la salida
Por cada tupla de la relación R:r
  Si no hay tuplas insertadas para r
  entonces
    agregar <r,null> a la salida
```

```
--full join
select interprete,titulo
from disco as d
full join interprete as i
on d.idinterprete=i.IdInterprete;
```

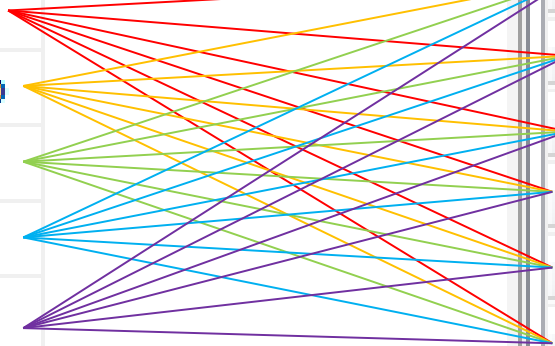
	interprete	titulo
1	NULL	Very Best Love Songs
2	NULL	Beautifull Love Songs
3	NULL	Special Love Songs
4	ABBA	Dancing Queen
5	Anita Ward	Ring My Bell
6	Baccara	Yes Sir, I Can Boogie
7	Barry Manilow	Copacabana
8	Ramy White	You're The First - My Last - My Everything

76	Wings	Silly Love Songs
77	Wings	Special Love Songs 2
78	Wings	Special Love Songs 3
79	WXX	NULL
80	Special one	NULL
81	El mejor	NULL

cross join explícito e  
implícito

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo



```
--cross join
select t1.id,valor1,t2.id,valor2
from tabla1 as t1
cross join tabla2 as t2;
go
```

	id	valor1	id	valor2
1	1	Primero	1	Primero
2	1	Primero	2	Segundo
3	1	Primero	3	Tercero
4	1	Primero	6	Sexto
5	1	Primero	7	Séptimo
6	1	Primero	8	Octavo
7	2	Segundo	1	Primero
8	2	Segundo	2	Segundo
9	2	Segundo	3	Tercero
10	2	Segundo	6	Sexto
11	2	Segundo	7	Séptimo
12	2	Segundo	8	Octavo
13	3	Tercero	1	Primero
14	3	Tercero	2	Segundo
15	3	Tercero	3	Tercero
16	3	Tercero	6	Sexto
17	3	Tercero	7	Séptimo
18	3	Tercero	8	Octavo
19	4	Cuarto	1	Primero
20	4	Cuarto	2	Segundo

...

# Algoritmo CROSS JOIN

Por cada tupla de la relación R :r  
    Por cada tupla de la relación S :s  
        Agregar <r,s> a la salida



Pares tienda/tipocomponente posibles, para componentes que comiencen por brother...

```
--cross join  
|select nombretienda,tipo  
from tienda  
cross join TipoComponente  
where tipo like 'brother%';
```

Resultados		Mensajes
	nombretienda	tipo
1	Tienda PC1	BROTHER - MULTIFUNCION LASER MONOCROMO
2	Tienda PC2	BROTHER - MULTIFUNCION LASER MONOCROMO
3	Tienda PC3	BROTHER - MULTIFUNCION LASER MONOCROMO
4	Tienda PC4	BROTHER - MULTIFUNCION LASER MONOCROMO
5	Tienda PC5	BROTHER - MULTIFUNCION LASER MONOCROMO
6	Tienda PC6	BROTHER - MULTIFUNCION LASER MONOCROMO
7	Tienda PC1	BROTHER ( MULTIFUNCIONES CHORRO DE TINTA )
8	Tienda PC2	BROTHER ( MULTIFUNCIONES CHORRO DE TINTA )
9	Tienda PC3	BROTHER ( MULTIFUNCIONES CHORRO DE TINTA )
10	Tienda PC4	BROTHER ( MULTIFUNCIONES CHORRO DE TINTA )
11	Tienda PC5	BROTHER ( MULTIFUNCIONES CHORRO DE TINTA )
12	Tienda PC6	BROTHER ( MULTIFUNCIONES CHORRO DE TINTA )

...

Igual funciona con el **CROSS JOIN Implícito**.

Sólo hace falta separar la lista de tablas con comas en la cláusula FROM

```
--cross join implícito  
-select nombretienda, tipo  
  from tienda, TipoComponente  
 where tipo like 'brother%';
```

Otros JOIN con condiciones  
diferentes a =

En los JOIN podemos cambiar la condición definida por el ON.  
Podemos enlazar con AND varios pares de campos, por ejemplo si las claves son de más de un campo.  
Pero también puede ocurrir que coloquemos una condición diferente.  
El sistema nos hará caso y evaluará para cada par de tuplas la condición.  
Colocará la tupla en el resultado si cumple la condición.

Debe usarse con precaución y sabiendo lo que queremos hacer.  
La recomendación es ceñirnos al uso visto y para otras condiciones usar el where

Cruzar los datos de las tablas disco e intérprete, mostrando los que intérprete sea mayor alfanuméricamente que titulo...

```
--otro join  
select interprete,titulo  
from disco as d  
inner join interprete as i  
on Interprete>titulo;
```

	interprete	titulo
1	ABBA	1999
2	Anita Ward	1999
3	Baccara	1999
4	Bary Manilow	1999
5	Bary White	1999
6	Bary White; Love Unlimited Orchestra	1999
7	Bee Gees	1999
8	Bee Gees	Beautifull Love Songs
9	Blondie	1999
10	Blondie	Beautifull Love Songs
11	Boney M.	1999
12	Boney M	Body Language

La Base de datos,  
obedientemente nos  
responde...  
Aunque la pregunta sea rara.

■ ■ ■

autocombinación

Es posible combinar una tabla consigo misma.

En el ejemplo del Mundo vamos a obtener pares de países posibles y sin repeticiones.

```
select p1.Name as País1, p2.Name as País2
from Country as p1
cross join Country as p2
where (p1.Code<>p2.Code) and
      (p1.Region='Southern Europe') and
      (p2.Region='Southern Europe');
```

37	Malta	Bosnia and Herzegovina
38	Portugal	Bosnia and Herzegovina
39	San Marino	Bosnia and Herzegovina
40	Slovenia	Bosnia and Herzegovina
41	Holy See (Vatican City ...	Bosnia and Herzegovina
42	Yugoslavia	Bosnia and Herzegovina
43	Albania	Spain
44	Andorra	Spain
45	Bosnia and Herzegovina	Spain
46	Gibraltar	Spain
47	Greece	Spain
48	Croatia	Spain

combinaciones con más de dos  
tablas



Podemos enlazar con join (en cualquiera de sus tipos) de más de dos tablas, para conseguir llegar a la información que necesitemos presentar o calcular. El funcionamiento es consecutivo, como con las operaciones matemáticas. Primero calcula la tabla resultante del primer join. Después hace el join entre esa tabla resultante y la siguiente. Y así sucesivamente. De izquierda a derecha y de arriba a abajo.

Podremos cambiar la secuencia en que se realiza poniendo paréntesis antes de la operación join que corresponda.

Hay que cuidar las condiciones where, que si afectan a campos que pueden ser null harán que desaparezcan esos registros salvo que lo controlemos adecuadamente.

Mostrar el título, intérprete y tipos de todos los discos...

```
select d.Titulo, i.interprete, t.Tipo
from Disco as d
inner join DiscoTipo as dt
    on d.iddisco=dt.iddisco
inner join Tipo as t
    on t.idtipo=dt.idtipo
inner join Interprete as i
    on d.idinterprete=i.IdInterprete;
go
```

Resultados		Mensajes	
	Titulo	interprete	Tipo
1	Yes Sir, I Can Boogie	Baccara	Euro disco
2	Copacabana	Barry Manilow	Disco
3	If I Can't Have You	Bee Gees	Disco
4	Jive Talkin'	Bee Gees	Disco
5	Saturday Night Fever	Bee Gees	Disco
6	You Should Be Dancing	Bee Gees	Disco
7	Daddy Cool	Boney M.	Euro disco
8	We Kill The World (Don't Kill The World)	Boney M.	Pop
9	Can't Take My Eyes off You	Boy Town Gang	Disco
10	I'm Every Woman	Chaka Khan	Disco
11	Love To Love You Baby	Donna Summer	Disco
12	No More Tears (Enough Is Enough)	Donna Summer; Barbra Streisand	Disco

Partimos de los tipos, de los que sacaremos todos, aunque no tengan discos asignados.

```
= select d.Titulo, i.interprete, t.Tipo  
from tipo as t  
left join DiscoTipo as dt  
    on t.idtipo=dt.idtipo  
left join Disco as d  
    on d.iddisco=dt.iddisco  
left join Interprete as i  
    on d.idinterprete=i.IdInterprete;  
go
```

Si pusiéramos inner join en alguno veríamos que ese enlace se hace obligatorio, por lo que se perderían en este caso los valores null.

Resultados		Mensajes	
	Titulo	interprete	Tipo
109	Rock the Boat	Hues Corporation	Oldies
110	Only The Strong Survive	Jerry Butler	Oldies
111	Soul Makossa	Manu Dibango	Oldies
112	We Kill The World (Don't Kill The World)	Boney M.	Pop
113	Meu Amigo Charlie Brown	Two Man Sound	Pop
114	Thriller	Michael Jackson	Pop
115	You're The First, My Last, My Everything	Barry White	R&B
116	Miss You	The Rolling Stones	Rock
117	Rivers of Babylon	Boney M.	Rocksteady
118	I'm So Excited	The Pointer Sisters	Soul
119	NULL	NULL	Rock espa...
120	NULL	NULL	Soul different

Mostrar todas las ciudades y los Idiomas del país al que pertenece una ciudad.

```
select c.Name as ciudad,l.Language as Idioma
from City as c
inner join Country as p
    on p.Code=c.CountryCode
inner join CountryLanguage as l
    on l.CountryCode=p.Code;
go
```

	Resultados	Mensajes
	ciudad	Idioma
19	Mazar-e-Sharif	Turkmenian
20	Mazar-e-Sharif	Uzbek
21	Amsterdam	Arabic
22	Amsterdam	Dutch
23	Amsterdam	Fries
24	Amsterdam	Turkish
25	Rotterdam	Arabic
26	Rotterdam	Dutch
27	Rotterdam	Fries
28	Rotterdam	Turkish
29	Haag	Arabic
30	Haag	Dutch

Sacamos todos los tipos e intérpretes aunque no tengan datos.

```
= select d.Titulo,i.interprete,t.Tipo
from tipo as t
left join (DiscoTipo as dt
          inner join Disco as d
            on d.iddisco=dt.iddisco)
on t.idtipo=dt.idtipo|
full join Interprete as i
on d.idinterprete=i.IdInterprete;
go
```

116	Miss You	The Rolling Stones	Rock
117	Rivers of Babylon	Boney M.	Rocksteady
118	I'm So Excited	The Pointer Sisters	Soul
119	NULL	NULL	Rock espa...
120	NULL	NULL	Soul different
121	NULL	El mejor	NULL
122	NULL	WXX	NULL
123	NULL	Special one	NULL
124	NULL	Umberto Tozzi	NULL

El paréntesis contiene una cláusula from válida. El SQL la calcula y después la enlaza como corresponda.

combinaciones, group by y  
funciones agrupadoras

El uso de múltiples tablas no interfiere con lo ya visto. Podremos filtrar datos mediante `where`. Teniendo cuidado de referenciar el campo o campos con el prefijo que indica la tabla a la que pertenece en el caso de que existan varios campos con el mismo nombre.

También podremos agrupar y filtrar la tabla resultante del `select` mediante el `group by` y el `having`.

Ordenar los datos obtenidos será tan inmediato como antes, mediante el `order by`.

Número de idiomas por cada país especificado en la BD mundo.

```
--group by
select P.Name as Pais, count(*) as NIdiomas
from Country as p
inner join Countrylanguage as l
      on p.code=l.CountryCode
group by p.Name
order by NIdiomas desc;
```

Resultados		Mensajes
	Pais	NIdiomas
1	Canada	12
2	China	12
3	India	12
4	Russian Federation	12
5	United States	12
6	South Africa	11
7	Tanzania	11
8	Uganda	10
9	Sudan	10
10	Nigeria	10
11	Philippines	10
12	Iran	10



## Sumar las puntuaciones de los juegos

```
select j.Juego, sum(p.puntuacion) as Spunt
from Juego as j
inner join JuegoPlataforma as jp
    on j.IdJuego=jp.Idjuego
inner join puntuacion as p
    on p.idjuegoplataforma=jp.id
group by j.Juego
order by Spunt desc;
```

Resultados		Mensajes
	Juego	Spunt
1	BioShock Infinite	18
2	Call of Duty: Black Ops	17
3	007 Legends	12
4	Angry Birds	10
5	BioShock	10
6	Halo 4	10
7	Star Wars Battlefront 2	10
8	World of Warcraft	10
9	Zelda II: The Adventure of Link	9
10	The Legend of Zelda	9
11	Profesor Layton y la Caja de Pandora	9

## Total de ventas en cada tienda informática

```
select t.NombreTienda, sum(fc.cantidad*fc.precioaplicado)
from Tienda as t
inner join Factura as f
    on f.idtienda=t.idtienda
inner join FacturaComponente as fc
    on fc.NFactura=f.NFactura
group by t.NombreTienda;
```

Resultados		Mensajes
	NombreTienda	(Sin nombre de columna)
1	Tienda PC1	2779
2	Tienda PC2	3931
3	Tienda PC3	887,9
4	Tienda PC4	577
5	Tienda PC5	3539,65
6	Tienda PC6	2450,5

combinaciones con update y  
delete

Formato del update:

```
UPDATE Tabla
    set campo=valor,...
FROM
    ...
WHERE
    ...
```

Pudiendo contener la cláusula from elementos  
join de las tablas.

Así podremos colocar en el where las  
descripciones y no sólo las claves.

Podemos emplear "update" o "delete" con "join" para actualizar o eliminar registros de una tabla consultando otras tablas.

Miramos los discos de un intérprete por su nombre...

```
select puntuacion,interprete,tipo
from tipo as t
inner join DiscoTipo as dt
    on t.idtipo=dt.idtipo
inner join Disco as d
    on d.iddisco=dt.iddisco
inner join Intérprete as i
    on d.idinterprete=i.IdIntérprete
inner join puntuacion as p
    on p.iddisco=d.iddisco
where Intérprete='Boney M.';
```

Resultados		Mensajes	
	puntuacion	interprete	tipo
1	4	Boney M.	Euro disco
2	3	Boney M.	Pop

Ahora vamos a modificar la puntuación del registro de tipo Pop...

```
]update puntuacion
-   set puntuacion=5
[|from tipo as t
  inner join DiscoTipo as dt
    on t.idtipo=dt.idtipo
  inner join Disco as d
    on d.iddisco=dt.iddisco
  inner join Interprete as i
    on d.idinterprete=i.IdInterprete
  inner join puntuacion as p
    on p.iddisco=d.iddisco
-where Interprete='Boney M.' and tipo='Pop';
```

Al hacer de nuevo el select  
veremos que se actualizó  
el valor.

Resultados		Mensajes	
	puntuacion	interprete	tipo
1	4	Boney M.	Euro disco
2	5	Boney M.	Pop

Igualmente sería con el delete.

```
DELETE Tabla  
FROM
```

```
    ...  
WHERE  
    ...
```

Vamos a borrar el elemento al que le cambiamos la puntuación y después veremos el resultado.

```

delete puntuacion
from tipo as t
inner join DiscoTipo as dt
    on t.idtipo=dt.idtipo
inner join Disco as d
    on d.iddisco=dt.iddisco
inner join Interprete as i
    on d.idinterprete=i.IdInterprete
inner join puntuacion as p
    on p.iddisco=d.iddisco
where Interprete='Boney M.' and tipo='Pop';
go

```

El otro registro se eliminó.

Resultados		Mensajes	
	puntuacion	interprete	tipo
1	4	Boney M.	Euro disco