

Explotación y administración de Base de datos

Juan Carlos Otaegui
jotaegui@unlam.edu.ar

SQL – Tipos de Juntas

Se usan para combinar registros de distintas tablas.

Básicamente se pueden dividir en 3 grupos:

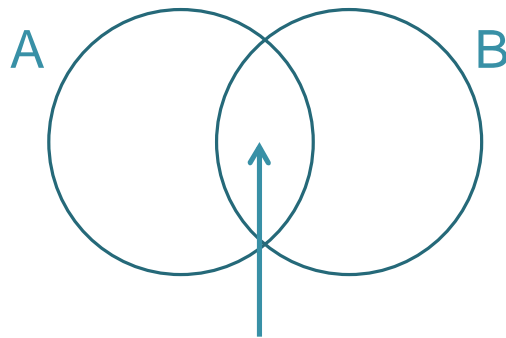
INNER

OUTER

CROSS

INNER JOIN

- Es el tipo de junta más utilizado.
- Se utilizar para traer los registros que cumplan con la condición en dos o más tablas.
- Es el tipo de junta implícito



Ejemplo

SQLQuery1.sql - SQL Server Enterprise (56)			
<pre>select ProductKey, EnglishProductName, ProductSubcategoryKey from dbo.DimProduct where ProductSubcategoryKey is not null order by 3; select ProductSubcategoryKey, EnglishProductSubcategoryName, ProductCategoryKey from dbo.DimProductSubcategory; select ProductCategoryKey, EnglishProductCategoryName from dbo.DimProductCategory;</pre>			
Results Messages			
ProductKey	EnglishProductName	ProductSubcategoryKey	
1 344	Mountain-100 Silver, 38	1	
2 345	Mountain-100 Silver, 42	1	
3 346	Mountain-100 Silver, 44	1	
4 347	Mountain-100 Silver, 48	1	
5 348	Mountain-100 Black, 38	1	
6 349	Mountain-100 Black, 42	1	
7 350	Mountain-100 Black, 44	1	
8 351	Mountain-100 Black, 48	1	
ProductSubcategoryKey	EnglishProductSubcategoryName	ProductCategoryKey	
1 1	Mountain Bikes	1	
2 2	Road Bikes	1	
3 3	Touring Bikes	1	
4 4	Handlebars	2	
5 5	Bottom Brackets	2	
6 6	Brakes	2	
7 7	Chains	2	
8 8	Cranksets	2	
ProductCategoryKey	EnglishProductCategoryName		
1 1	Bikes		
2 2	Components		

Ejemplo

SQLQuery1.sql -...istrator (56))*

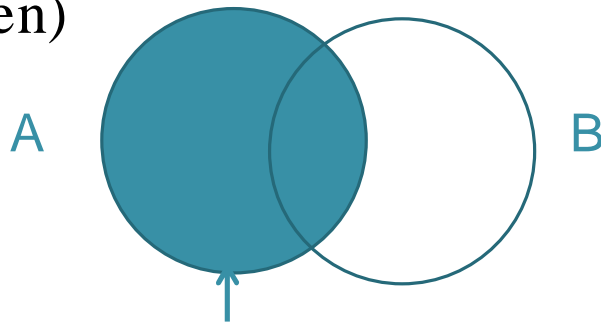
```
select ProductKey, EnglishProductName,  
PSC.ProductSubcategoryKey, PSC.EnglishProductSubcategoryName  
from dbo.DimProduct P  
inner join dbo.DimProductSubcategory PSC  
on (P.ProductSubcategoryKey = PSC.ProductSubcategoryKey)  
where P.ProductSubcategoryKey is not null order by 3;
```

Results Messages

	ProductKey	EnglishProductName	ProductSubcategoryKey	EnglishProductSubcategoryName
1	344	Mountain-100 Silver, 38	1	Mountain Bikes
2	345	Mountain-100 Silver, 42	1	Mountain Bikes
3	346	Mountain-100 Silver, 44	1	Mountain Bikes
4	347	Mountain-100 Silver, 48	1	Mountain Bikes
5	348	Mountain-100 Black, 38	1	Mountain Bikes
6	349	Mountain-100 Black, 42	1	Mountain Bikes
7	350	Mountain-100 Black, 44	1	Mountain Bikes
8	351	Mountain-100 Black, 48	1	Mountain Bikes
9	352	Mountain-200 Silver, 38	1	Mountain Bikes
10	353	Mountain-200 Silver, 38	1	Mountain Bikes
11	354	Mountain-200 Silver, 42	1	Mountain Bikes
12	355	Mountain-200 Silver, 42	1	Mountain Bikes
13	356	Mountain-200 Silver, 46	1	Mountain Bikes
14	357	Mountain-200 Silver, 46	1	Mountain Bikes
15	358	Mountain-200 Black, 38	1	Mountain Bikes
16	359	Mountain-200 Black, 38	1	Mountain Bikes
17	360	Mountain-200 Black, 42	1	Mountain Bikes
18	361	Mountain-200 Black, 42	1	Mountain Bikes
19	362	Mountain-200 Black, 46	1	Mountain Bikes
20	363	Mountain-200 Black, 46	1	Mountain Bikes

LEFT OUTER JOIN

- Se utiliza cuando queremos los registros de la tabla A y cuando sea posible juntar también los de la tabla B.
- Ejemplo: Nos interesa obtener los datos del cliente (tabla cliente) y de sus tarjetas asociadas (si existen ya que no todos la tienen)



Ejemplo

SQLQuery1.sql -...istrator (56))*

```

select ProductKey, EnglishProductName,
PSC.ProductSubcategoryKey, PSC.EnglishProductSubcategoryName
from dbo.DimProduct P
left outer join dbo.DimProductSubcategory PSC
on (P.ProductSubcategoryKey = PSC.ProductSubcategoryKey)
--where P.ProductSubcategoryKey is not null
order by 3;

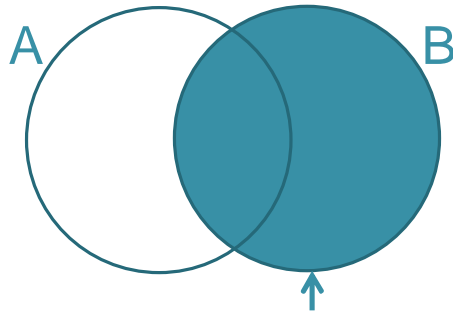
```

Results Messages

	ProductKey	EnglishProductName	ProductSubcategoryKey	EnglishProductSubcategoryName
201	201	Seat Lug	NULL	NULL
202	202	Stem	NULL	NULL
203	203	Seat Post	NULL	NULL
204	204	Steerer	NULL	NULL
205	205	Seat Stays	NULL	NULL
206	206	Seat Tube	NULL	NULL
207	207	Top Tube	NULL	NULL
208	208	Tension Pulley	NULL	NULL
209	209	Rear Derailleur Cage	NULL	NULL
210	210	HL Road Frame - Bl...	14	Road Frames
211	211	HL Road Frame - R...	14	Road Frames
212	212	Sport-100 Helmet, R...	31	Helmets
213	213	Sport-100 Helmet, R...	31	Helmets
214	214	Sport-100 Helmet, R...	31	Helmets
215	215	Sport-100 Helmet, B...	31	Helmets
216	216	Sport-100 Helmet, B...	31	Helmets
217	217	Sport-100 Helmet, B...	31	Helmets
218	218	Mountain Bike Sock...	23	Socks
219	219	Mountain Bike Sock...	23	Socks
220	220	Sport-100 Helmet, B...	31	Helmets

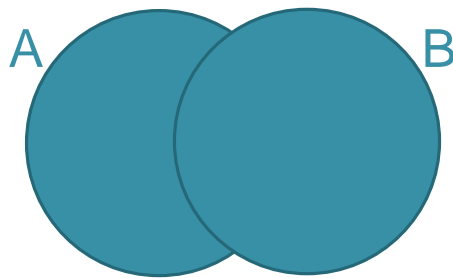
RIGHT OUTER JOIN

- Se utiliza cuando queremos los registros de la tabla B y cuando sea posible juntar también los de la tabla A.
- Idéntico al LEFT pero identifica la posición de las tablas en el clausula «from»



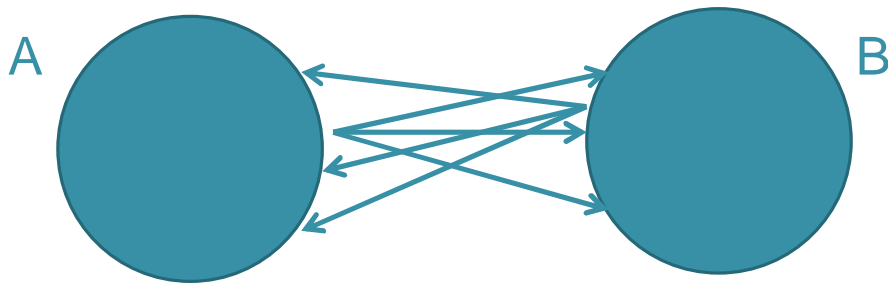
FULL OUTER JOIN

- Se utiliza cuando queremos los registros de la tabla B y de la tabla A.
- Independientemente de los registros en común.
- Ejemplo: Medios de pago



CROSS JOIN

- Se utiliza cuando no se indica condición de junta.
- Genera un producto cartesiano.



Ejemplo

SQLQuery1.sql - SQL Server Enterprise Manager (56)

```
select ProductKey, EnglishProductName,  
PSC.ProductSubcategoryKey, PSC.EnglishProductSubcategoryName  
from dbo.DimProduct P  
cross join dbo.DimProductSubcategory PSC  
--on (P.ProductSubcategoryKey = PSC.ProductSubcategoryKey)  
--where P.ProductSubcategoryKey is not null  
order by 3;
```

Results Messages

	ProductKey	EnglishProductName	ProductSubcategoryKey	EnglishProductSubcategoryName
1	1	Adjustable Race	1	Mountain Bikes
2	2	Bearing Ball	1	Mountain Bikes
3	3	BB Ball Bearing	1	Mountain Bikes
4	4	Headset Ball Bearings	1	Mountain Bikes
5	5	Blade	1	Mountain Bikes
6	6	LL Crankarm	1	Mountain Bikes
7	7	ML Crankarm	1	Mountain Bikes
8	8	HL Crankarm	1	Mountain Bikes
9	9	Chaining Bolts	1	Mountain Bikes
10	10	Chaining Nut	1	Mountain Bikes
11	11	Chaining	1	Mountain Bikes
12	12	Crown Race	1	Mountain Bikes
13	13	Chain Stays	1	Mountain Bikes
14	14	Decal 1	1	Mountain Bikes
15	15	Decal 2	1	Mountain Bikes
16	16	Down Tube	1	Mountain Bikes
17	17	Mountain End Caps	1	Mountain Bikes
18	18	Road End Caps	1	Mountain Bikes
19	19	Touring End Caps	1	Mountain Bikes
20	20	Fork End	1	Mountain Bikes

Query executed successfully. WIN-OPGKN3SKUZU (10.0 RTM) WIN-OPGKN3SKUZU\Admini... AdventureWorksDW 00:00:01 22422 rows

Clausula IN/NOT IN

Se utiliza en el «where» para indicar una restricción en cuanto a los posibles valores que puede tomar un atributo.

Ejemplo:

```
SELECT * FROM CLIENTE WHERE  
STATUS IN ('Activo','Suspendido');
```

```
SELECT * FROM CLIENTE WHERE  
STATUS NOT IN ('Baja');
```




Clausula EXISTS/NOT EXISTS

■ Similar a IN/NOT IN pero se puede indicar todo un SQL anidado para validar una regla de negocio.

■ Ejemplo:

```
SELECT *  
FROM CLIENTE C  
WHERE EXISTS ( SELECT *  
                FROM TRAMITES T  
                WHERE C.CLIENTE = T.CLIENTE  
                AND T.TIPO_TRAMITE = 'Baja '  
              )
```

Ejercicio

-  Describa (con registros ejemplo incluidos) tablas de su modelo físico descrito en la unidad de diseño lógico.
-  Luego incorpore enunciados y SQLs que den respuestas a esas consultas requeridas.
-  Las consultas deben incluir:
 1. Uso de Select con clausula INNER JOIN
 2. Uso de Select con clausula LEFT JOIN
 3. Uso de Select con clausula IN y otra con NOT IN
 4. Uso de Select con clausula EXISTS / NOT EXISTS
 5. Incluya también el dataset resultante que espera obtener