

CURSOS
INTERSEMESTRALES



PROTECO

JOINS

SQL 1

15 enero 2020

¿Qué Son?

Traducido al bruto, un join es una ligadura.
Cuando se desea realizar una consulta de datos que se encuentran en una o mas tablas es necesario relacionar, asociar o ligar (join) para extraer la información necesaria



¿Condiciones?

Para realizar la liga o asociación de tablas, siempre se realiza igualando los valores de los campos que tengan en común.

Típicamente la manera de ligar o asociar tablas es mediante el uso de las PKs y las FKs de las tablas involucradas



Clasificaciones

- Inner Joins
 - Equi Join
 - Non equi Join
 - Natural Join
- Outer Joins
 - Left Outer Join
 - Right Outer Join
- Cross Join
- Self Join



INNER JOIN

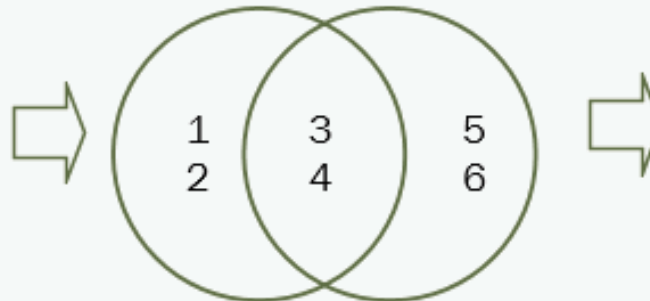
Los inner joins únicamente incluyen en la salida de la consulta los registros que tengan correspondencia con la tabla asociada.

1	3
2	4
3	5
4	6

A

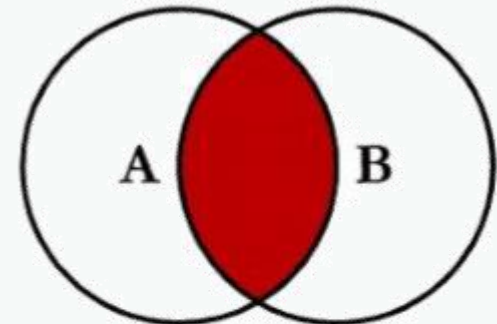
3	4
5	6

B



A INNER JOIN B

3
4



```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



PROTECO

Syntax

```
select ...  
from <table-1>  
[inner | natural | cross] join <table-2>  
[on <condition> | using <column-name>, ...], ...
```

```
SELECT A.n  
FROM A  
INNER JOIN B ON B.m = A.n;
```

```
SELECT A.n  
FROM A  
INNER JOIN B ON B.m = A.n  
INNER JOIN C ON C.o = A.n;
```



Curiosidades

El único caso donde el alias es requerido, es para los casos en donde se tengan atributos con el mismo nombre en ambas tablas.

El alias permite eliminar la ambigüedad ya que el manejador no sabría distinguir a cuál de las 2 tablas referirse. En este caso, la columna cliente_id requiere el uso de alias

Es posible emplear expresiones como c.*, t.* y *. En este último caso, la consulta mostraría todos los campos de ambas tablas, incluyendo el campo cliente_id de ambas tablas



Natural Join

Básicamente Natural Join es un Inner Join con una variante, sin embargo:

- En un Inner Join siempre se debe especificar el predicado o condición del join
- En Natural Join, el predicado se construye automáticamente por lo que puede ser omitido.
- La regla para construir el predicado de forma automática o 'natural' es considerando el nombre de las columnas.



Syntax

```
select ...  
from <table-1>  
[inner | natural | cross] join <table-2>  
[on <condition> | using <column-name>, ..., ...]
```

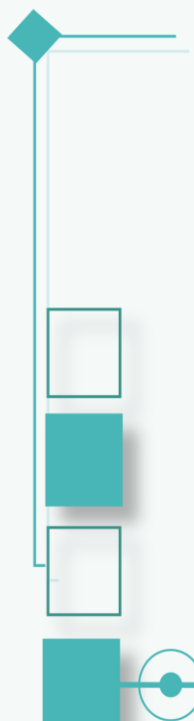
R

A	B	C	D	E
2	30	100	X	True
5	70	400	W	False
6	80	500	M	False

S

A	B	F	G	H
2	30	100	X	True
5	80	400	W	False
6	89	500	M	False





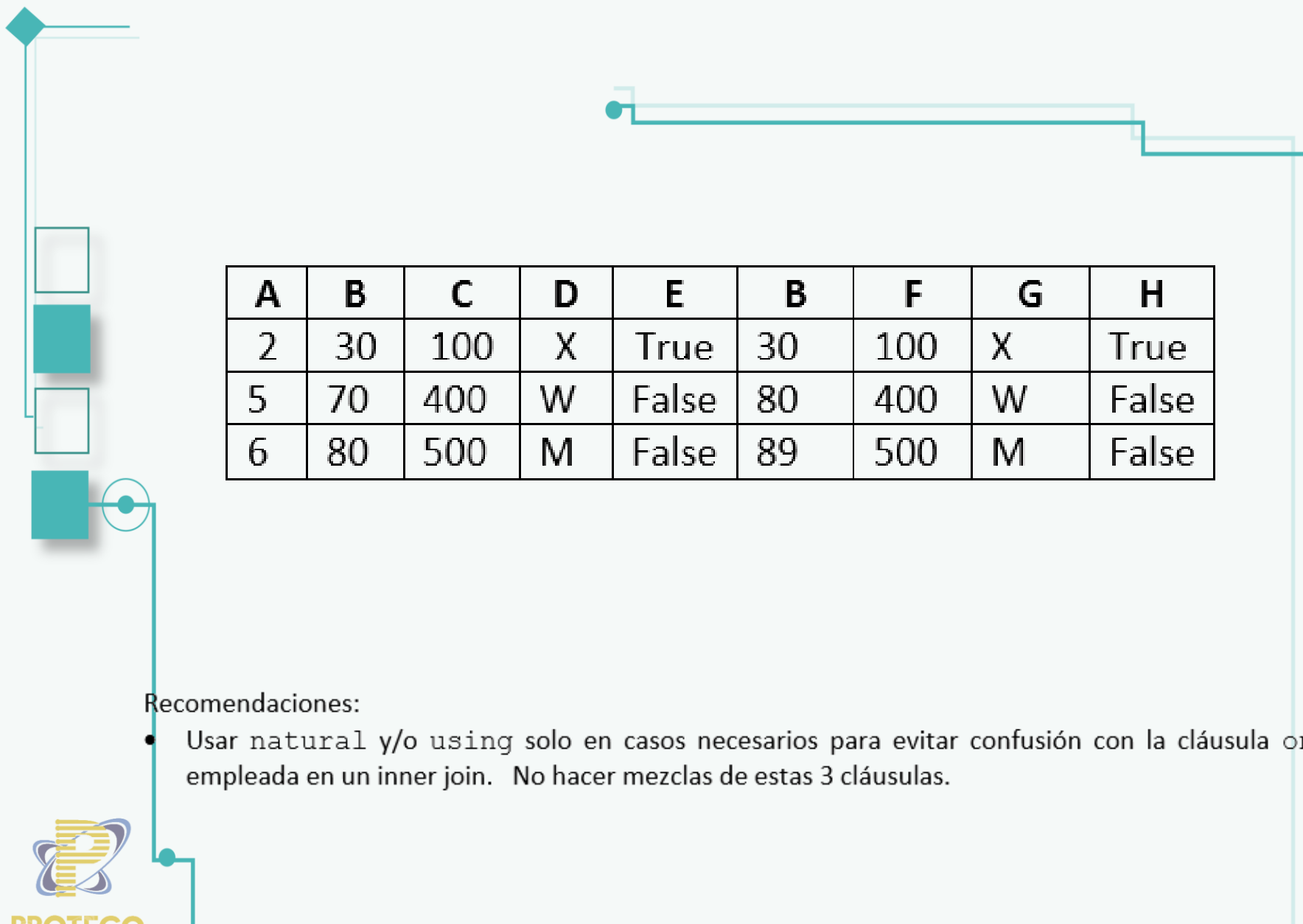
A	B	C	D	E	F	G	H
2	30	100	X	True	100	X	True



Curiosidades

- Observar que la sentencia SQL no contiene el predicado del Join. En su lugar se emplea la palabra natural.
- En este ejemplo, existen 2 columnas con el mismo nombre: A y B. Por lo tanto, el predicado que se aplica de forma automática es: $R.A = S.A$ and $R.B = S.B$
- El predicado solo se cumple para el primer registro, por lo que el resultado del natural Join solo contiene un registro.
- ¿Qué sucede si existen columnas con el mismo nombre, pero no se desea que sean tomadas en cuenta como parte del predicado del Join?
- En este caso se puede emplear una variante de Natural join a través del uso de la cláusula using.





A	B	C	D	E	B	F	G	H
2	30	100	X	True	30	100	X	True
5	70	400	W	False	80	400	W	False
6	80	500	M	False	89	500	M	False

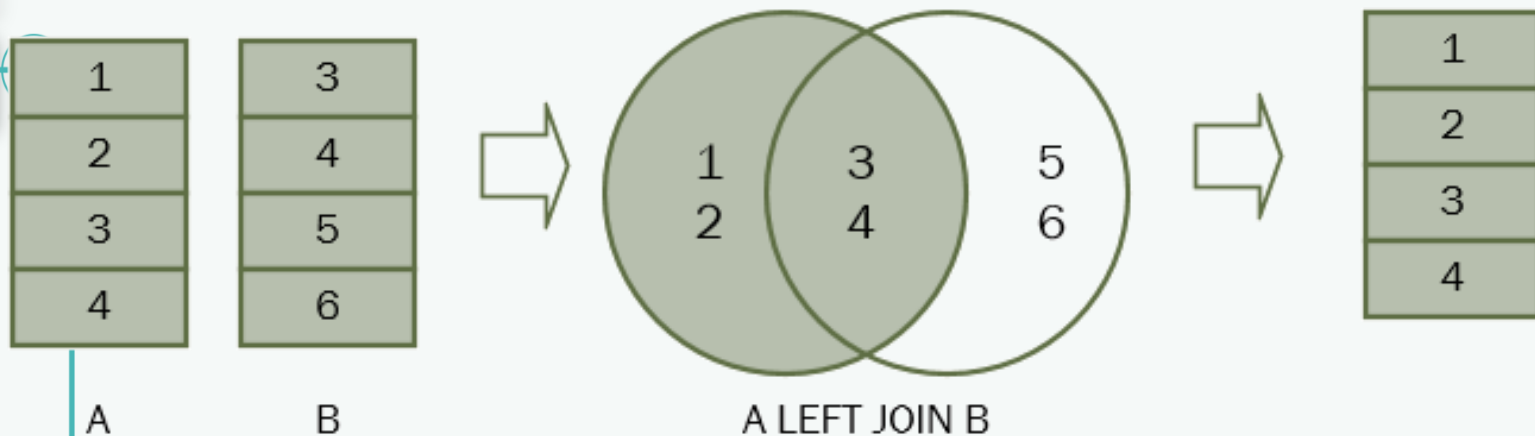
Recomendaciones:

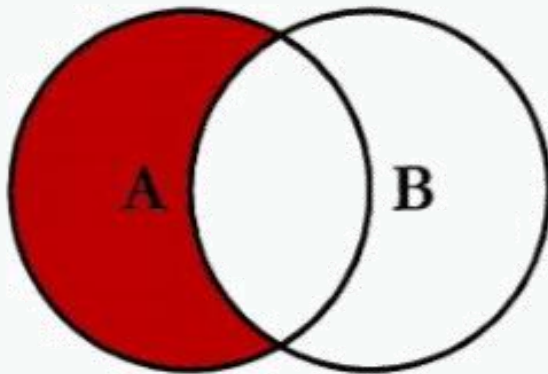
- Usar `natural` y/o `using` solo en casos necesarios para evitar confusión con la cláusula `on` empleada en un `inner join`. No hacer mezclas de estas 3 cláusulas.



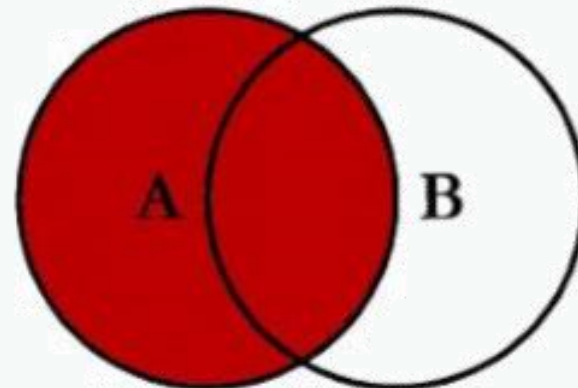
LEFT OUTER JOIN

Especifica una unión entre dos tablas con una cláusula de unión explícita, conservando las filas no coincidentes de la primera tabla





```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL.
```

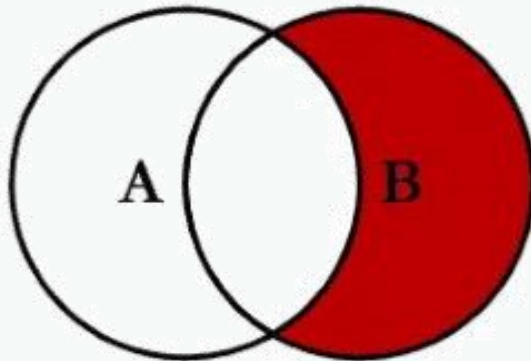


```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

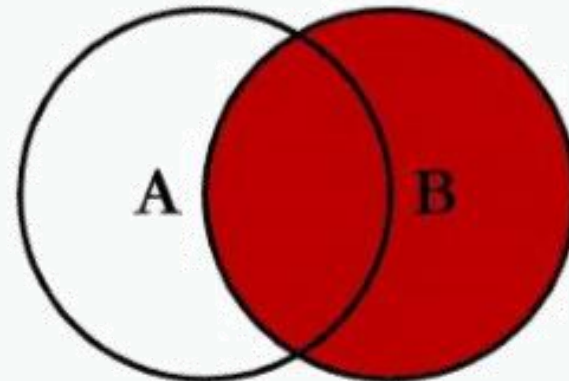


RIGHT OUTER JOIN

Especifica una unión entre dos tablas con una cláusula de unión explícita, conservando las filas no coincidentes de la segunda tabla



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



Ejemplo

Escenario a considerar:

Un estudiante puede o no tener una beca, y una beca puede o no estar asociada a un estudiante, es decir, puede haber becas vacantes. Observar los datos de estas 2 tablas.

ESTUDIANTE

ESTUDIANTE_ID	NOMBRE	AP_PATERO	AP_MATERNO	BECA_ID
2001	MARIA	JIMENEZ	SALAZAR	500
2002	HUGO	RODRIGUEZ	BENITEZ	{NULL}
2003	ARMANDO	ZARATE	MARTINEZ	501
2004	RODRIGO	MUNGUIA	SOLANO	{NULL}
2005	JULIO	LUNA	MEJIA	502

BECA

BECA_ID	TIPO	DURACION	MONTO
500	ESCOLAR	5	6000
501	ALIMENTICIA	2	3000
502	VITALICIA	1	1000
503	TEMPORAL	1	1500

Left outer join:

Mostrar todos los estudiantes y si tienen beca, mostrar los datos de la beca.

Right outer join:

Mostrar todas las becas disponibles, y en caso de que esté asociada a un estudiante, mostrar sus datos:



FULL OUTER JOIN

- En teoría, una combinación externa completa es la combinación de una combinación izquierda y una combinación derecha.
- La combinación externa completa incluye todas las filas de las tablas unidas, independientemente de que la otra tabla tenga o no la fila correspondiente.

