



## Práctica 11: Variaciones de JOIN”

### Fundamentos de Bases de Datos

#### Integrantes:

- Aceves Higareda Mayra Guadalupe  
ahigareda.mayra@ciencias.unam.mx  
310646246
- Badillo Lora Carlos  
carlosb@ciencias.unam.mx  
415083504
- Moreno Ruíz Jesús Fernando  
chuchini@ciencias.unam.mx  
414001967
- Servín Mote Edson  
edson@ciencias.unam.mx  
308292503

**Profesor:** Gerardo Avilés Rosas

**Laboratorio:** Erick Orlando Matla Cruz



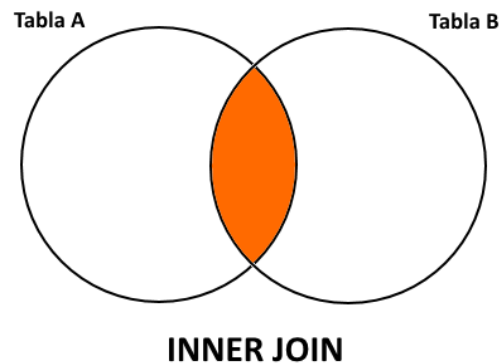
---

---

## Diferencias entre las variaciones de JOIN

### INNER JOIN

Un **inner join** o **join**, se puede representar como la intersección entre dos conjuntos, es decir, una consulta con inner join mostraría los registros de las tablas que coinciden en los campos de la unión que se ha definido en dicha consulta.



Su sintaxis es:

```
FROM TABLA1 [INNER] JOIN TABLA2 ON CONDICIONES_VINCULOS_TABLAS
```

Por ejemplo:

**R:**

A	B	C
1	2	3
6	7	8
9	7	8

**S:**

B	C	D
2	3	4
2	3	5
7	8	10

**Consulta:**

```
SELECT A, R.B AS B, S.C AS C, D  
FROM R JOIN S  
ON R.C = S.C AND R.B = S.B;
```

---

---

La consulta devuelve la siguiente tabla:

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

Cabe señalar que esto puede ocasionar la desaparición del resultado de filas de alguna de las dos tablas, por tener valores nulos, o por tener un valor que no exista en la otra tabla entre los campos/columnas que se están comparando.

### Combinaciones externas (OUTER JOIN)

Las combinaciones externas se realizan mediante la instrucción **OUTER JOIN**. Devuelven todos los valores de la tabla que hemos puesto a la derecha, los de la tabla que hemos puesto a la izquierda o los de ambas tablas según el caso, devolviendo además valores nulos en las columnas de las tablas que no tengan el valor existente en la otra tabla.

Es decir, nos permite seleccionar algunas filas de una tabla aunque éstas no tengan correspondencia con las filas de la otra tabla con la que se combina.

La sintaxis general de las combinaciones externas es:

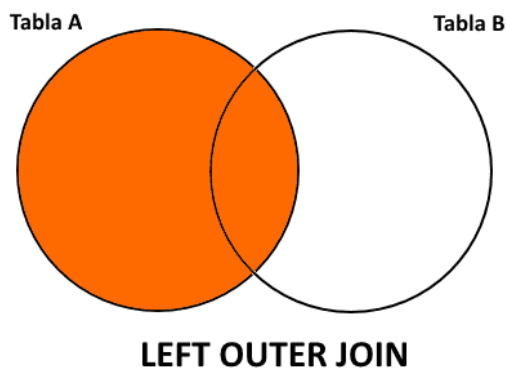
```
SELECT ATRIBUTO1, ..., ATRIBUTO N
FROM TABLA1 [LEFT/RIGHT/FULL] [OUTER] JOIN TABLA2
ON CONDICIONES_VINCULOS_TABLAS;
```

Existen tres variantes de las combinaciones externas. En todas estas combinaciones externas el uso de la palabra **OUTER** es opcional. Si utilizamos **LEFT**, **RIGHT** o **FULL** y la combinación de columnas, el sistema sobreentiende que estamos haciendo una combinación externa.

---

---

**LEFT JOIN** recoge todos los datos de la tabla que está a la izquierda de la unión en la consulta, aunque no tengan correspondencia en la tabla de la derecha.



Por ejemplo:

**R:**

A	B	C
1	2	3
6	7	18
9	7	8

**S:**

B	C	D
2	3	4
2	3	5
7	8	10
2	5	15

**Consulta:**

```
SELECT *  
FROM R LEFT OUTER JOIN S;
```

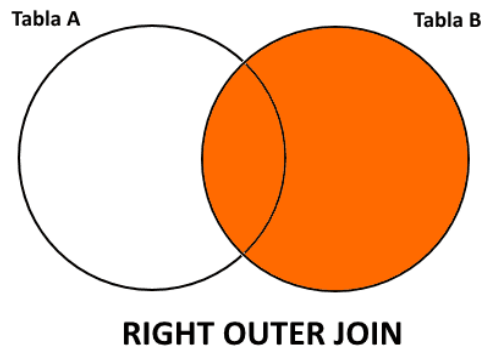
La consulta devuelve la siguiente tabla:

A	B	C	D
1	2	3	4
1	2	3	5
6	7	18	NULL
9	7	8	10

---

---

Análogamente, **RIGHT JOIN** recoge todos los datos de la tabla que está a la derecha de la unión en la consulta, aunque no tengan correspondencia en la tabla de la izquierda.



Por ejemplo:

**R:**

A	B	C
1	2	3
6	7	18
9	7	8

**S:**

B	C	D
2	3	4
2	3	5
7	8	10
2	5	15

**Consulta:**

```
SELECT *  
FROM R RIGHT OUTER JOIN S;
```

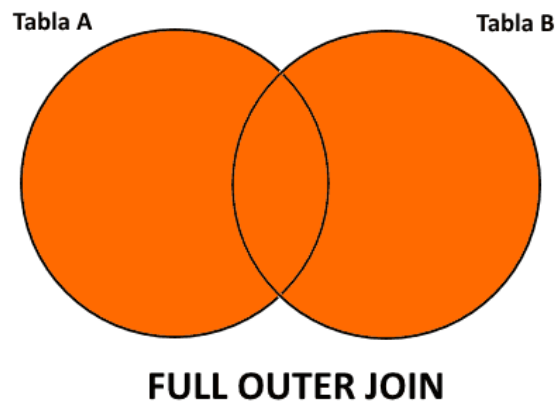
La consulta devuelve la siguiente tabla:

A	B	C	D
1	2	3	4
1	2	3	5
9	7	8	10
NULL	2	5	15

---

---

**FULL JOIN** obtiene todas las filas en ambas tablas, aunque no tengan correspondencia en la otra tabla. Es decir, todos los registros de A y de B aunque no haya correspondencia entre ellos, rellenando con nulos los campos que falten:



Es equivalente a obtener los registros comunes (con un INNER JOIN) y luego añadirle los de la tabla A que no tienen correspondencia en la tabla B, con los campos de la tabla vacíos, y los registros de la tabla B que no tienen correspondencia en la tabla A, con los campos de la tabla A vacíos.

Por ejemplo:

**R:**

A	B	C
1	2	3
6	7	18
9	7	8

**S:**

B	C	D
2	3	4
2	3	5
7	8	10
2	5	15

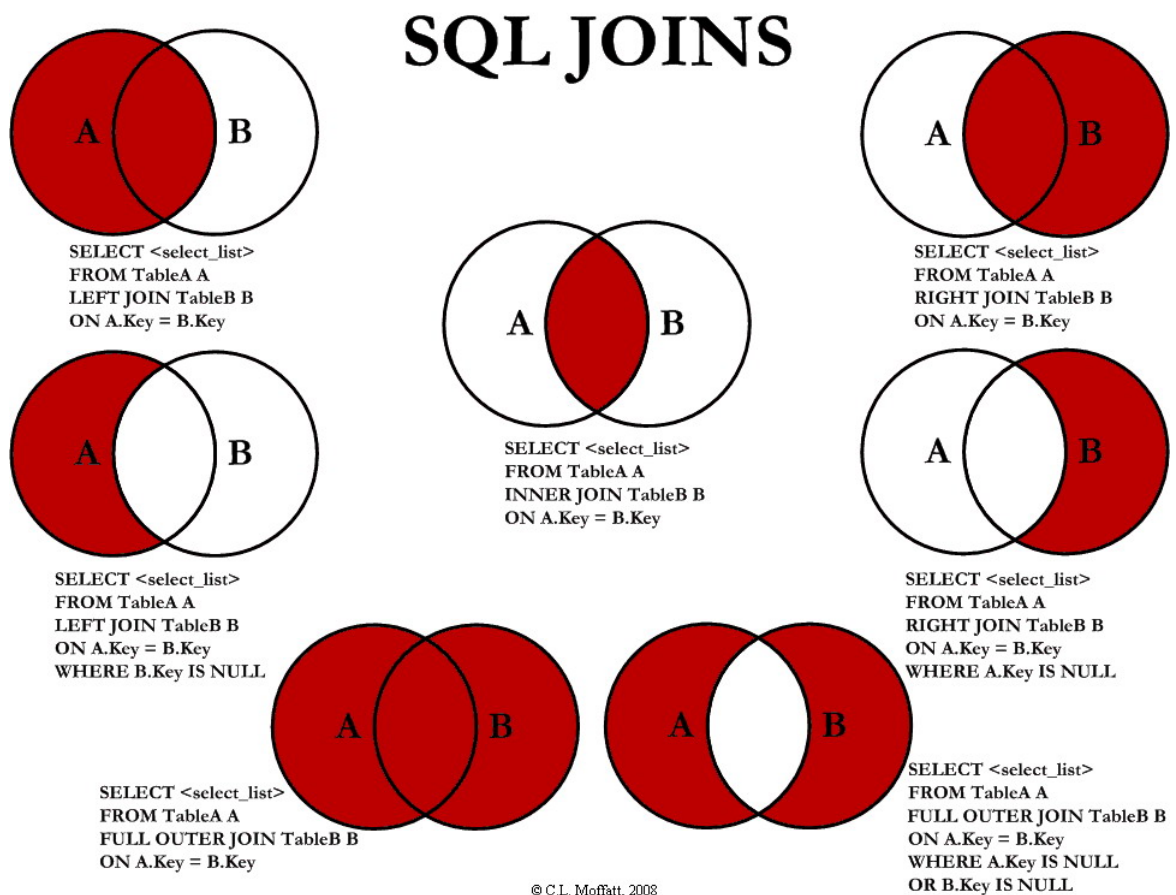
**Consulta:**

```
SELECT *  
FROM R FULL OUTER JOIN S;
```

La consulta devuelve la siguiente tabla:

A	B	C	D
1	2	3	4
1	2	3	5
6	7	18	NULL
9	7	8	10
NULL	2	5	15

La siguiente imagen ejemplifica con diagramas de Venn-Euler algunas de las operaciones que realizamos en las consultas donde se hace uso de combinaciones externas, así como del INNER JOIN:



---

---

## CROSS JOIN

Un CROSS JOIN que no incluye la cláusula **WHERE** produce el producto cartesiano de las tablas involucradas en el join. El tamaño del conjunto resultante del producto cartesiano es el número de renglones en la primera tabla multiplicado por el número de renglones en la segunda tabla.

Por ejemplo:

**R:**

A	B	C
1	2	3
6	7	8
9	7	8

**S:**

B	C	D
2	3	4
2	3	5
7	8	10

**Consulta:**

```
SELECT *  
FROM R FULL OUTER JOIN S;
```

La consulta devuelve la siguiente tabla:

R.A	R.B	R.C	S.B	S.C	S.D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	2	3	4
6	7	8	2	3	5
6	7	8	7	8	10
9	7	8	2	3	4
9	7	8	2	3	5
9	7	8	7	8	10

Sin embargo, si se añade una cláusula **WHERE**, CROSS JOIN funciona como un INNER JOIN.



---

---

## *Bibliografía*

- “Diferencia entre inner join, left join y right join. SQL”  
<https://donnierock.com/2014/03/04/diferencia-entre-inner-join-left-join-y-right-join-sql/>  
[Fecha de consulta: 28 de octubre del 2016]
- Avilés, G. (2016). Structured Query Language [diapositivas de PowerPoint]  
Recuperado de:  
<http://www.engineerhunters.com/gerardo/fbd/material/uploaden/Gerardo/Presentaciones/06SQL.pdf>
- “Fundamentos de SQL: Consultas SELECT multi-tabla - Tipos de JOIN”  
<http://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Consultas-SELECT-multi-tabla-Tipos-de-JOIN.aspx> [Fecha de consulta: 28 de octubre del 2016]