



UNIVERSIDAD TÉCNOLOGICA NACIONAL
FACULTAD REGIONAL GENERAL PACHECO

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

Base de Datos II

JOINS

ING. ARIEL HERRERA



Contenido

Join	2
Tablas de ejemplo.....	2
Combinación interna (INNER JOIN)	3
LEFT JOIN Y RIGHT JOIN	5
TODOS LOS TIPOS DE JOIN EN SQL – GUÍA DE REFERENCIA RÁPIDA	7



Join

La sentencia **JOIN** en SQL permite combinar registros de dos o más tablas en una base de datos relacional.

Tablas de ejemplo

Todas las explicaciones que están a continuación utilizan las siguientes dos tablas para ilustrar el efecto de diferentes clases de uniones **JOIN**.

Tabla Empleado	
Apellido	IDDepartamento
Andrade	31
Jordán	33
Steinberg	33
Róbinson	34
Solano	34
Gaspar	36



Tabla Departamento	
NombreDepartamento	IDDepartamento
Ventas	31
Ingeniería	33
Producción	34
Mercadeo	35

La tabla Empleado contiene a los empleados con el número del departamento al que pertenecen; mientras que la tabla Departamento, contiene el nombre de los departamentos de la empresa, se puede notar que existe un empleado que tiene asignado un número de departamento que no se encuentra en la tabla Departamento (Gaspar), igualmente, en la tabla Departamento existe un departamento al cual no pertenece empleado alguno (Mercadeo). Esto servirá para presentar algunos ejemplos más adelante.

Combinación interna (INNER JOIN)

Con esta operación se calcula el producto cruzado de todos los registros; así cada registro en la tabla A es combinado con cada registro de la tabla B; pero sólo permanecen aquellos registros en la tabla combinada que satisfacen las condiciones que se especifiquen. Este es el tipo de *JOIN* más utilizado por lo que es considerado el tipo de combinación predeterminado.

SQL:2003 especifica dos formas diferentes para expresar estas combinaciones. La primera, conocida como *explícita* usa la palabra JOIN, mientras que la segunda es *implícita* y usa ',' para separar las tablas a



combinar en la sentencia FROM de la declaración SELECT. Entonces siempre se genera el producto cruzado del cual se seleccionan las combinaciones que cumplan lo que indica la sentencia **WHERE**.

Es necesario tener especial cuidado cuando se combinan columnas con valores nulos **NULL** ya que el valor nulo no se combina con otro valor o con otro nulo, excepto cuando se le agregan predicados tales como **IS NULL** o **IS NOT NULL**.

Como ejemplo, la siguiente consulta toma todos los registros de la tabla Empleado y encuentra todas las combinaciones en la tabla Departamento. La sentencia **JOIN** compara los valores en la columna IDDepartamento en ambas tablas. Cuando no existe esta correspondencia entre algunas combinaciones, éstas no se muestran; es decir que si el número de departamento de un empleado no coincide con los números de departamento de la tabla Departamento, no se mostrará el empleado con su respectivo departamento en la tabla resultante.

Las dos consultas siguientes son similares, y se realizan de manera explícita (A) e implícita (B).

A. Ejemplo de la sentencia INNER JOIN

explícita: SELECT *

FROM empleado

INNER JOIN departamento

ON empleado.IDdepartamento = departamento.IDdepartamento

B. Ejemplo de la sentencia INNER JOIN

implícita: SELECT *

FROM empleado, departamento

WHERE empleado.IDdepartamento = departamento.IDDepartamento



Resultados:

Empleado.Apellido	Empleado.IDdepartamento	departamento.NombreDepartamento	departamento.IDDepartamento
Solano	34	Producción	34
Jordán	33	Ingeniería	33
Róbinson	34	Producción	34
Steinberg	33	Ingeniería	33
Andrade	31	Ventas	31

El empleado Gaspar y el departamento de Mercadeo no son presentados en los resultados ya que ninguno de estos tiene registros correspondientes en la otra tabla.

No existe un departamento con número 36 ni existe un empleado con número de departamento 35.

LEFT JOIN Y RIGHT JOIN

Si empleamos la cláusula INNER en la consulta se seleccionarán sólo aquellos registros de la tabla de la que hayamos escrito a la izquierda de INNER JOIN que contengan al menos un registro relacionado de la tabla que hayamos escrito a la derecha.

Para modificar esto tenemos dos cláusulas que sustituyen a la palabra clave INNER, estas cláusulas son LEFT y RIGHT. LEFT toma todos los registros de la tabla de la izquierda aunque no tengan ningún registro relacionado en la tabla de la derecha.



SELECT *

FROM empleado

LEFT [OUTER] JOIN departamento

ON empleado.IDdepartamento = departamento.IDdepartamento

RIGHT realiza la misma operación pero al contrario, toma todos los registros de la tabla de la derecha aunque no tenga ningún registro relacionado en la tabla de la izquierda.

SELECT *

FROM empleado

RIGHT [OUTER] JOIN departamento

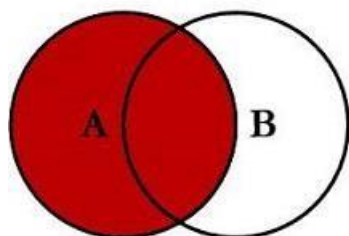
ON empleado.IDdepartamento = departamento.IDdepartamento

Aquellos registros sin correspondencia entregaran para el campo correspondiente un valor NULL, es decir si un empleado no tiene departamento, entonces su valor es null.

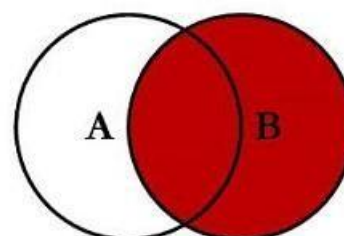


TODOS LOS TIPOS DE JOIN EN SQL – GUÍA DE REFERENCIA RÁPIDA

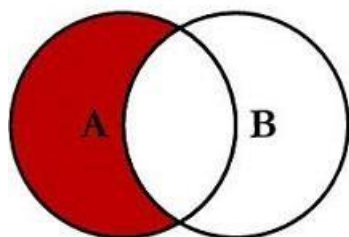
SQL JOINS



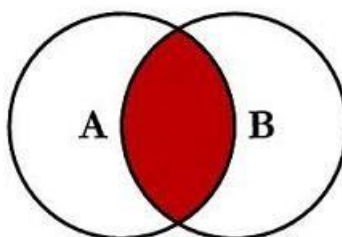
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



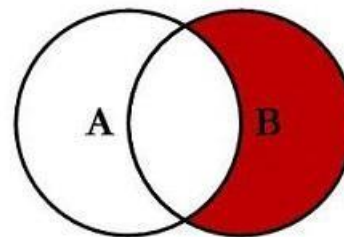
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



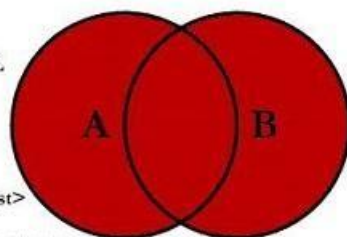
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



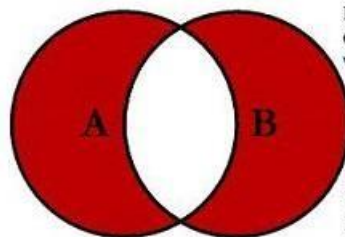
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
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  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



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

© C.L. Moffatt, 2008

<https://ingenieriadesoftware.es/tipos-sql-join-guia-referencia/>. Consultado

24/08/2020