

Introducción a Bases de Datos y SQL

Módulo 5

Otros tipos de JOIN

Cláusula JOIN

Recordemos que en un **SELECT**, cuya sintaxis básica es:

```
SELECT campos FROM tabla WHERE condición
```

podemos distinguir tres partes:

1. **Campos**: columnas a devolver.
2. **Tabla**: la o las tablas en las que se encuentran los datos.
3. **Condición**: condiciones que deben cumplirse al momento de efectuar la consulta.

Hasta ahora hemos utilizado sólo una tabla en cada consulta. A continuación, veremos cómo se hace para **combinar datos que provienen de más de una tabla**.

Cuando se quiere consultar campos que se encuentran en distintas tablas, es necesario combinar las mismas a partir de la cláusula **JOIN** mediante un campo en común.

La cantidad de **JOINS** es igual a la cantidad de tablas que participan en la consulta **-1**.

Cláusula JOIN

Sintaxis alternativas

A continuación, vamos a ver dos maneras posibles de combinar tablas mediante la cláusula **JOIN** para aquellos casos en los que el campo en común no tiene el mismo nombre. Las sintaxis son las siguientes:

```
SELECT tabla1.campos, tabla2.campos
FROM tabla1
JOIN tabla2 ON tabla1.campo1=tabla2.campo1
WHERE condición
```

```
SELECT tabla1.campos, tabla2.campos
FROM tabla1, tabla2
WHERE tabla1.campo1=tabla2.campo1
AND condición
```

- Ambas son válidas; puede utilizar el código con el cual se sienta más cómodo.
- Hay que tener en cuenta que en ambos casos, si los campos por los cuales se combinan las tablas tienen el mismo nombre, hay que escribirlos en el **SELECT** de la forma "**tabla.campo**" para especificar a qué tabla pertenece el campo.
- En todos los tipos de **JOIN**, al referirse a la tabla de la **izquierda**, estamos hablando de la tabla especificada en el **FROM**, y la tabla de la derecha es la tabla definida a continuación del **JOIN**.
- En las sintaxis **1** y **2**, la tabla de la **izquierda** es **TABLA1** y la de la derecha, **TABLA2**.

Cláusula JOIN

Otros tipos de JOIN

Los mismos pueden ser:

- **LEFT [OUTER] JOIN**
- **RIGHT [OUTER] JOIN**
- **CROSS JOIN**

A continuación, explicaremos cada **JOIN** con un ejemplo, suponiendo las siguientes tablas:

Codigo (int)	Nombre (varchar(15))
1	A
3	C
8	H

Tabla 1

Codigo (int)	Nombre2 (varchar(15))
3	Tres
5	Cinco
8	Ocho

Tabla 2

Otros tipos de JOIN

LEFT [OUTER] JOIN

Este **JOIN** devuelve todos los registros de la tabla de la izquierda y los registros que coinciden de la tabla de la derecha:

```
SELECT *  
FROM tabla1  
LEFT JOIN tabla2  
ON tabla1.codigo = tabla2.codigo;
```

obteniendo como resultado de la combinación de la **TABLA1** y la **TABLA2**, los siguientes registros:

1	A	
3	C	Tres
8	H	Ocho

Otros tipos de JOIN

LEFT [OUTER] JOIN

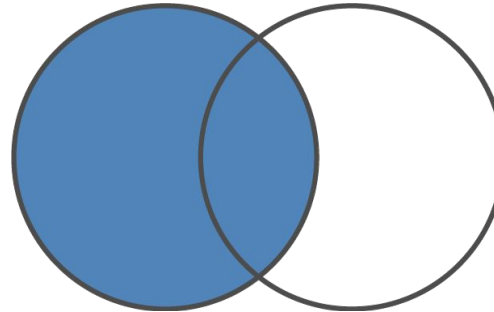
Esta consulta también puede escribirse:

```
SELECT * FROM tabla1  
LEFT OUTER JOIN tabla2  
ON tabla1.codigo = tabla2.codigo;
```

Basándonos nuevamente en los diagramas de Venn, podemos expresar el resultado del **LEFT [OUTER] JOIN** del siguiente modo:

Tabla 1

Tabla 2



Otros tipos de JOIN

RIGHT [OUTER] JOIN

Este **JOIN** devuelve todos los registros de la tabla de la derecha y los registros que coinciden de la tabla de la izquierda:

```
SELECT *  
FROM tabla1  
RIGHT JOIN tabla2  
ON tabla1.codigo = tabla2.codigo;
```

obteniendo como resultado de la combinación de la **TABLA1** y la **TABLA2**, los siguientes registros:

3	C	Tres
5		Cinco
8	H	Ocho

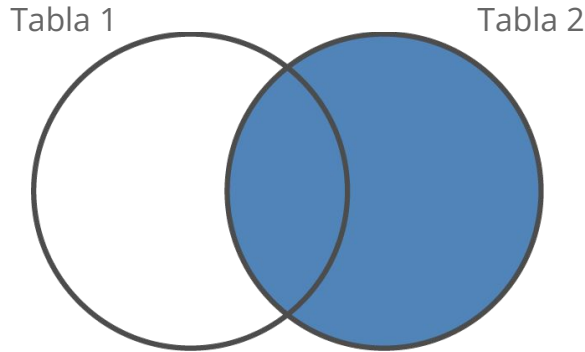
Otros tipos de JOIN

RIGHT [OUTER] JOIN

Esta consulta también puede escribirse:

```
SELECT * FROM tabla1  
RIGHT OUTER JOIN tabla2  
ON tabla1.codigo = tabla1.codigo;
```

Si seguimos con la analogía de los diagramas de Venn, podemos expresar el resultado del **RIGHT [OUTER] JOIN** del siguiente modo:



Otros tipos de JOIN

CROSS JOIN

Combina cada registro de la tabla de la izquierda con cada registro de la tabla de la derecha, sin hacer coincidir un campo en particular:

```
SELECT * FROM tabla1 CROSS JOIN tabla2;
```

obteniendo como resultado de la combinación de la **TABLA1** y la **TABLA2**, los siguientes registros:

1	A	3	Tres
1	A	5	Cinco
1	A	8	Ocho
3	C	3	Tres
3	C	5	Cinco
3	C	8	Ocho
8	H	3	Tres
8	H	5	Cinco
8	H	8	Ocho

Nota: este tipo de **JOIN** brinda la posibilidad de cruzar todos los registros con todos (producto cartesiano) y resulta imposible de dibujar con un diagrama de Venn. Se puede observar que representa un **JOIN** muy poco eficiente de ejecutar en tablas grandes.

¡Muchas gracias!

¡Sigamos trabajando!