
Unidad Didáctica 7. Consultas sobre varias tablas. Composición externa

Apuntes de BD para DAW, DAM y ASIR

José Juan Sánchez Hernández

Curso 2023/2024

Índice

1 Consultas sobre varias tablas. Composición externa	1
1.0.1 Composiciones externas	1
2 Errores comunes	7
3 Referencias	8
4 Licencia	9

Índice de figuras

Índice de cuadros

1 Consultas sobre varias tablas. Composición externa

1.0.1 Composiciones externas

- Join externa
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN (No implementada en MySQL)
 - NATURAL LEFT OUTER JOIN
 - NATURAL RIGHT OUTER JOIN

Ejemplo de LEFT OUTER JOIN:

```
1 SELECT *
2 FROM empleado LEFT JOIN departamento
3 ON empleado.codigo_departamento = departamento.codigo
```

Esta consulta devolverá todas las filas de la tabla que hemos colocado a la izquierda de la composición, en este caso la tabla `empleado`. Y relacionará las filas de la tabla de la izquierda (`empleado`) con las filas de la tabla de la derecha (`departamento`) con las que encuentre una coincidencia. Si no encuentra ninguna coincidencia, se mostrarán los valores de la fila de la tabla izquierda (`empleado`) y en los valores de la tabla derecha (`departamento`) donde no ha encontrado una coincidencia mostrará el valor `NULL`.

LEFT JOIN

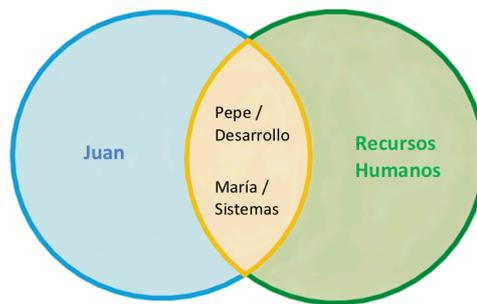
1

```
/* SQL 2 */
SELECT *
FROM empleado LEFT JOIN departamento
ON empleado.id_departamento = departamento.id
```

2

Tabla: empleado			Tabla: departamento	
id	nombre	id_departamento	id	nombre
1	Pepe	1	1	Desarrollo
2	María	2	2	Sistemas
3	Juan	NULL	3	Recursos Humanos

Estas filas quedan **fuera de la intersección**



3

El **resultado de la operación LEFT JOIN** es:

empleado. id	empleado. nombre	empleado. id_departamento	departamento. id	departamento. nombre
1	Pepe	1	1	Desarrollo
2	María	2	2	Sistemas
3	Juan	NULL	NULL	NULL



<http://josejuansanchez.org/bd>

Ejemplo de RIGHT OUTER JOIN:

```
1 SELECT *
2 FROM empleado RIGHT JOIN departamento
3 ON empleado.codigo_departamento = departamento.codigo
```

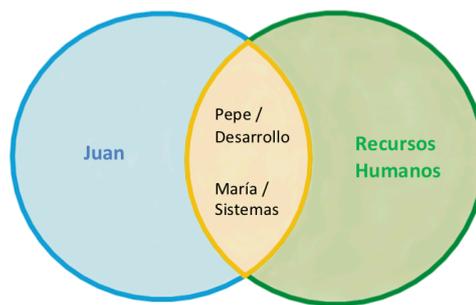
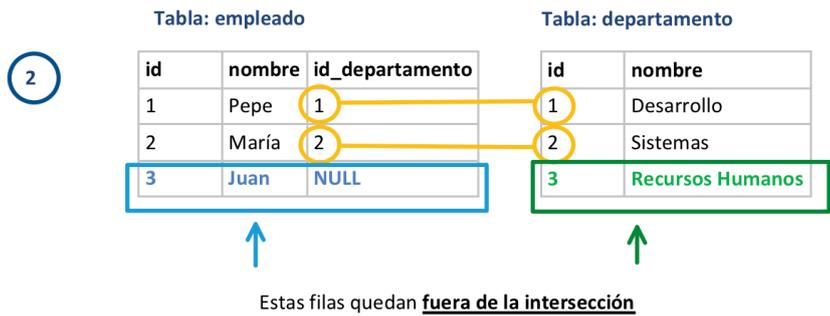
Esta consulta devolverá todas las filas de la tabla que hemos colocado a la derecha de la composición, en este caso la tabla `departamento`. Y relacionará las filas de la tabla de la derecha (`departamento`) con las filas de la tabla de la izquierda (`empleado`) con las que encuentre una coincidencia. Si no encuentra ninguna coincidencia, se mostrarán los valores de la fila de la tabla derecha (`departamento`) y en los valores de la tabla izquierda (`empleado`) donde no ha encontrado una coincidencia mostrará el valor `NULL`.

RIGHT JOIN

1

```

/* SQL 2 */
SELECT *
FROM empleado RIGHT JOIN departamento
ON empleado.id_departamento = departamento.id
    
```



3 El **resultado de la operación RIGHT JOIN** es:

empleado. id	empleado. nombre	empleado. id_departamento	departamento. id	departamento. nombre
1	Pepe	1	1	Desarrollo
2	María	2	2	Sistemas
NULL	NULL	NULL	3	Recursos Humanos



<http://josejuansanchez.org/bd>

Ejemplo de FULL OUTER JOIN:

La composición **FULL OUTER JOIN** **no está implementada en MySQL**, por lo tanto para poder simular esta operación será necesario hacer uso del operador **UNION**, que nos realiza la unión del resultado de dos consultas.

El resultado esperado de una composición de tipo **FULL OUTER JOIN** es obtener la intersección de las dos tablas, junto las filas de ambas tablas que no se puedan combinar. Dicho con otras palabras, el resultado sería el equivalente a realizar la unión de una consulta de tipo **LEFT JOIN** y una consultas de tipo **RIGHT JOIN** sobre las mismas tablas.

```
1 SELECT *
2 FROM empleado LEFT JOIN departamento
3 ON empleado.codigo_departamento = departamento.codigo
4
5 UNION
6
7 SELECT *
8 FROM empleado RIGHT JOIN departamento
9 ON empleado.codigo_departamento = departamento.codigo
```

Para poder utilizar el operador **UNION** entre dos o más consultas deberá tener en cuenta que:

- Deben tener el mismo número de columnas.
- Las columnas que se van a unir tienen que tener tipos de datos similares.

Para ordenar los resultados tras aplicar una operación de **UNION** existen dos soluciones:

- Usar la posición de la columna sobre la que queremos ordenar los resultados en el **ORDER BY**.
- Crear un alias en las columnas del primer **SELECT** sobre la que queremos ordenar los resultados y usarlo en el **ORDER BY**.

Ejemplo:

```
1 /* Solución 1 */
2 SELECT departamento.nombre, empleado.apellido1, empleado.apellido2, empleado.
   nombre
3 FROM empleado LEFT JOIN departamento
4 ON empleado.codigo_departamento=departamento.codigo
5
6 UNION
7
8 SELECT departamento.nombre, empleado.apellido1, empleado.apellido2, empleado.
   nombre
9 FROM empleado RIGHT JOIN departamento
10 ON empleado.codigo_departamento=departamento.codigo
11 ORDER BY 1, 2, 3, 4;
```

```
1 /* Solución 2 */
2 SELECT
3   departamento.nombre AS nombre_departamento,
4   empleado.apellido1, empleado.apellido2, empleado.nombre
5 FROM empleado LEFT JOIN departamento
6 ON empleado.codigo_departamento=departamento.codigo
7
8 UNION
```

```
9
10 SELECT departamento.nombre, empleado.apellido1, empleado.apellido2, empleado.
    nombre
11 FROM empleado RIGHT JOIN departamento
12 ON empleado.codigo_departamento=departamento.codigo
13 ORDER BY nombre_departamento;
```

Ejemplo de NATURAL LEFT JOIN:

```
1 SELECT *
2 FROM empleado NATURAL LEFT JOIN departamento
```

Esta consulta realiza un `LEFT JOIN` entre las dos tablas, la única diferencia es que en este caso no es necesario utilizar la cláusula `ON` para indicar sobre qué columna vamos a relacionar las dos tablas. **En este caso las tablas se van a relacionar sobre aquellas columnas que tengan el mismo nombre.** Por lo tanto, sólo deberíamos utilizar una composición de tipo `NATURAL LEFT JOIN` cuando estemos seguros de que los nombres de las columnas sobre las que quiero relacionar las dos tablas se llaman igual en las dos tablas.

2 Errores comunes

1. Cuando estamos usando `LEFT JOIN` o `RIGHT JOIN` no deberíamos tener varias condiciones en la cláusula `ON`.

Consulta incorrecta

```
1 SELECT *
2 FROM fabricante LEFT JOIN producto
3 ON fabricante.codigo = producto.codigo_fabricante AND
4    producto.codigo_fabricante IS NULL;
```

Consulta correcta.

```
1 SELECT *
2 FROM fabricante LEFT JOIN producto
3 ON fabricante.codigo = producto.codigo_fabricante
4 WHERE producto.codigo_fabricante IS NULL;
```

3 Referencias

- [Wikibook SQL Exercises](#).
- [Tutorial SQL de w3resource](#).
- [MySQL Join Types by Steve Stedman](#).
- **Bases de Datos**. 2ª Edición. Grupo editorial Garceta. Iván López Montalbán, Manuel de Castro Vázquez y John Ospino Rivas.
- **INNER JOIN**.
- **LEFT JOIN**.
- **RIGHT JOIN**.

4 Licencia

Esta página forma parte del curso Bases de Datos de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.