



Tema 4

LENGUAJE SQL

CONSULTAS MULTITABLA



Índice

1. Multiplicaciones de tablas.
2. Composiciones o Combinaciones simples (JOIN)
3. Composición o combinación natural.
4. Composiciones o Combinaciones basadas en desigualdad.
5. Composiciones o Combinaciones de una tabla consigo misma.
6. Composiciones o Combinaciones Externas (OUTER JOIN)
7. Composiciones y Subconsultas



1. Multiplicaciones de tablas

- Hasta ahora, las órdenes SQL que hemos utilizado están basadas en una única tabla, pero a menudo es necesario utilizar datos procedentes de dos o más tablas de la base de datos.
- Para poder acceder a dos o más tablas de una base de datos, SQL genera internamente una tabla en la que cada fila de una tabla se combina con todas y cada una de las filas de las demás tablas indicadas. Esta operación es el producto cartesiano de las tablas que se están accediendo y la tabla resultante contiene todas las columnas de todas las tablas que se han multiplicado.
- Pueden unirse tantas tablas como se desee (aunque la experiencia aconseja que no sean muchas por optimización).
- Comenzaremos con el formato más sencillo de multiplicación de tablas, reseñando solo las cláusulas significativas para realizar el producto.
- Pueden ser utilizadas todas las cláusulas de selección vistas anteriormente y al final indicaremos el formato completo.



1. Multiplicaciones de tablas

- El formato más sencillo es:
 - `SELECT [ALL/DISTINCT] ExpresionColumna [, ExpresionColumna]
FROM NombreTabla [AliasTabla] [, NombreTabla [AliasTabla]
.....]`
- En la SELECT pueden seleccionarse columnas de ambas tablas. Si hay columnas con el mismo nombre en las distintas tablas de la FROM, deben identificarse como `NombreTabla.NombreColumna` o `AliasTabla.NombreColumna`. Para no tener que escribir siempre el nombre de la tabla, por comodidad, se suele utilizar un alias para cada tabla eligiendo un nombre corto (1 o 2 caracteres) que identifique a cada tabla. En algún caso, que veremos más adelante, este alias será imprescindible.



1. Multiplicaciones de tablas

- **Ejemplos**

- Veremos la salida que produce la multiplicación, o producto cartesiano, de dos tablas con un ejemplo para obtener todos los empleados, indicando su número de empleado, su apellido, el nombre de su departamento y su localidad de éste.
 - `mysql> SELECT emp_no "Nº EMPLEADO", apellido "APELLIDO", dnombre "DEPARTAMENTO" , loc "LOCALIDAD"`
`FROM empleados, departamentos;`



1. Multiplicaciones de tablas

N° EMPLEADO	APELLIDO	DEPARTAMENTO	LOCALIDAD
7499	ALONSO	CONTABILIDAD	BARCELONA
7499	ALONSO	INVESTIGACION	VALENCIA
7499	ALONSO	VENTAS	MADRID
7499	ALONSO	PRODUCCION	SEVILLA
7521	LOPEZ	CONTABILIDAD	BARCELONA
7521	LOPEZ	INVESTIGACION	VALENCIA
7521	LOPEZ	VENTAS	MADRID
7521	LOPEZ	PRODUCCION	SEVILLA
7654	MARTIN	CONTABILIDAD	BARCELONA
7654	MARTIN	INVESTIGACION	VALENCIA
7654	MARTIN	VENTAS	MADRID
7654	MARTIN	PRODUCCION	SEVILLA
7698	GARRIDO	CONTABILIDAD	BARCELONA
7698	GARRIDO	INVESTIGACION	VALENCIA
7698	GARRIDO	VENTAS	MADRID
7698	GARRIDO	PRODUCCION	SEVILLA
7782	MARTINEZ	CONTABILIDAD	BARCELONA
7782	MARTINEZ	INVESTIGACION	VALENCIA
7782	MARTINEZ	VENTAS	MADRID
7782	MARTINEZ	PRODUCCION	SEVILLA
7839	REY	CONTABILIDAD	BARCELONA
7839	REY	INVESTIGACION	VALENCIA
7839	REY	VENTAS	MADRID
7839	REY	PRODUCCION	SEVILLA
7844	CALVO	CONTABILIDAD	BARCELONA
7844	CALVO	INVESTIGACION	VALENCIA
7844	CALVO	VENTAS	MADRID
7844	CALVO	PRODUCCION	SEVILLA
7876	GIL	CONTABILIDAD	BARCELONA
7876	GIL	INVESTIGACION	VALENCIA
7876	GIL	VENTAS	MADRID
7876	GIL	PRODUCCION	SEVILLA
7900	JIMENEZ	CONTABILIDAD	BARCELONA
7900	JIMENEZ	INVESTIGACION	VALENCIA
7900	JIMENEZ	VENTAS	MADRID
7900	JIMENEZ	PRODUCCION	SEVILLA



1. Multiplicaciones de tablas

- Cada empleado de la tabla de empleados aparece tantas veces como departamentos hay en la tabla de departamentos, con los correspondientes valores de cada una de las filas de la tabla departamentos.
- Se obtienen, por tanto, $9 \times 4 = 36$ filas.



2. Composiciones o Combinaciones simples (JOIN)

- Acabamos de ver, en el ejemplo anterior, que la salida producida por la multiplicación de las tablas de empleados y de departamentos no tiene mucho sentido y poca aplicación.
- La solución más adecuada del ejemplo sería enlazar la tabla empleados con la tabla departamentos de forma que para cada empleado de la tabla empleados solo tengamos de la tabla departamentos la fila correspondiente a su departamento. Es decir:
 - ```
mysql> SELECT emp_no "Nº EMPLEADO", apellido "APELLIDO", dnombre
DEPARTAMENTO, loc "LOCALIDAD"
FROM empleados, departamentos
WHERE empleados.dept_no = departamentos.dept_no;
```





## 2. Composiciones o Combinaciones simples (JOIN)

| Nº EMPLEADO | APELLIDO | DEPARTAMENTO  | LOCALIDAD |
|-------------|----------|---------------|-----------|
| 7521        | LOPEZ    | CONTABILIDAD  | BARCELONA |
| 7782        | MARTINEZ | CONTABILIDAD  | BARCELONA |
| 7839        | REY      | CONTABILIDAD  | BARCELONA |
| 7876        | GIL      | INVESTIGACION | VALENCIA  |
| 7900        | JIMENEZ  | INVESTIGACION | VALENCIA  |
| 7499        | ALONSO   | VENTAS        | MADRID    |
| 7654        | MARTIN   | VENTAS        | MADRID    |
| 7698        | GARRIDO  | VENTAS        | MADRID    |
| 7844        | CALVO    | VENTAS        | MADRID    |

- Hemos visto que la salida que produce el producto de las tablas es cada fila de una tabla combinada con todas las de otra tabla. Esta información no suele ser la deseada. Aparecen las ***composiciones o combinaciones de tablas*** que nos proporcionan la misma información pero filtrada. Una composición o combinación (join) consiste en aplicar una condición de selección a las filas obtenidas de la multiplicación de las tablas sobre las que se está realizando una consulta.



## 2. Composiciones o Combinaciones simples (JOIN)

- El formato de la combinación de tablas sería:
  - `SELECT [ALL/DISTINCT] ExpresionColumna [, ExpresionColumna .....]`
  - `FROM NombreTabla [AliasTabla] [, NombreTabla [AliasTabla]......]`
  - `[WHERE CondicionComposicion]`
- donde ***CondicionComposicion*** : es una condición que selecciona las filas de la composición de las tablas.
- Dependiendo de la condición de composición tendremos las combinaciones naturales, basadas en la igualdad, o las combinaciones basadas en la desigualdad.



## 2. Composiciones o Combinaciones simples (JOIN)

- Existen varios tipos según sea esta condición de composición
  - ***Composición natural:*** es la más sencilla. Es aquella en que la condición de selección se establece con el operador de igualdad entre las columnas que deban coincidir exactamente en tablas diferentes.
  - ***Composición basada en desigualdad:*** es menos utilizada y consiste en que la condición de selección no sea una igualdad. (no las veremos)



### 3. Composición o Combinación natural

- Como hemos dicho la composición o combinación natural es la más sencilla y natural. Es aquella en que la condición de selección se establece con el operador de igualdad entre las columnas que deben coincidir exactamente en tablas diferentes.
- Suele utilizarse para unir tablas en las que hay una relación a través de las claves ajenas, uniendo una tabla con su referenciada.
- Veamos el ejemplo anterior:
  - `mysql> SELECT emp_no "Nº EMPLEADO", apellido "APELLIDO", dnombre DEPARTAMENTO, loc "LOCALIDAD"`
    - > `FROM empleados, departamentos`
    - > `WHERE empleados.dept_no = departamentos.dept_no;`



### 3. Composición o Combinación natural

| Nº EMPLEADO | APELLIDO | DEPARTAMENTO  | LOCALIDAD |
|-------------|----------|---------------|-----------|
| 7521        | LOPEZ    | CONTABILIDAD  | BARCELONA |
| 7782        | MARTINEZ | CONTABILIDAD  | BARCELONA |
| 7839        | REY      | CONTABILIDAD  | BARCELONA |
| 7876        | GIL      | INVESTIGACION | VALENCIA  |
| 7900        | JIMENEZ  | INVESTIGACION | VALENCIA  |
| 7499        | ALONSO   | VENTAS        | MADRID    |
| 7654        | MARTIN   | VENTAS        | MADRID    |
| 7698        | GARRIDO  | VENTAS        | MADRID    |
| 7844        | CALVO    | VENTAS        | MADRID    |

- Si obtenemos el producto cartesiano de las tablas empleados por departamentos para cada empleado obtenemos la combinación con todas las filas de departamento. Pero la única que nos interesará será la del departamento al que pertenezca el empleado, es decir la que cumpla la condición: **departamento. dept\_no = empleado. dept\_no** .



### 3. Composición o Combinación natural

- SQL resuelve el anterior ejercicio con el siguiente proceso:
  - La cláusula FROM genera todas las combinaciones posibles de filas de la tabla de *empleados* (9 filas) por las de la tabla de *departamentos* (4 filas), resultando una tabla producto de  $4 \times 9 = 36$  filas.
  - La cláusula WHERE selecciona únicamente aquellas filas de la tabla producto donde coinciden los números de departamento, que necesitan el nombre de la tabla o el alias por tener el mismo nombre en ambas tablas. En total se han seleccionado 9 filas y las 27 restantes se eliminan.
  - La sentencia SELECT visualiza las columnas especificadas de las tablas producto para las filas seleccionadas.
- SQL no exige que las columnas de emparejamiento estén relacionadas como clave primaria y clave ajena, aunque suele ser lo habitual. Pueden servir cualquier par de columnas de dos tablas, siempre que tengan tipos de datos comparables.



### 3. Composición o Combinación natural

- El formato es:

**[NombreTabla1.] NombreColumna1 =**  
**[NombreTabla2.] NombreColumna2**

- Los nombres de las columnas necesitarán anteponer el nombre de la tabla o el alias si el nombre es el mismo en ambas tablas.



### 3. Composición o Combinación natural

- **Ejemplos**

- Obtener los distintos números y nombres de departamentos a los que pertenecen los empleados de la tabla *empleados*:
  - `mysql> SELECT DISTINCT d.dept_no "Nº Departamento", d.dnombre  
Departamento  
FROM empleados e, departamentos d  
WHERE e.dept_no = d.dept_no;`

| Nº Departamento | Departamento  |
|-----------------|---------------|
| 10              | CONTABILIDAD  |
| 20              | INVESTIGACION |
| 30              | VENTAS        |





### 3. Composición o Combinación natural

- **Ejemplos**

- Mostrar de los pedidos, el número de pedido, la fecha del pedido junto con la descripción del producto que contiene

```
mysql> SELECT num_pedido, fecha_pedido, descripcion
```

```
FROM pedidos pe, productos pr
```

```
WHERE pe.num_producto = pr.num_producto order by num_pedido;
```

| num_pedido | fecha_pedido | descripcion                |
|------------|--------------|----------------------------|
| 1000       | 1999-10-06   | SILLA DIRECTOR MOD. BÚFALO |
| 1001       | 1999-10-06   | ARCHIVADOR CEREZO          |
| 1002       | 1999-10-07   | MESA DESPACHO MOD. GAUIOTA |
| 1003       | 1999-10-16   | SILLA DIRECTOR MOD. BÚFALO |
| 1004       | 1999-10-20   | MESA MODELO UNIÓN          |
| 1005       | 1999-10-20   | ARMARIO NOGAL DOS PUERTAS  |
| 1006       | 1999-11-03   | DESTRUCTORA DE PAPEL A3    |
| 1007       | 1999-11-06   | ARCHIVADOR CEREZO          |
| 1008       | 1999-11-16   | MESA DESPACHO MOD. GAUIOTA |
| 1009       | 1999-11-26   | SILLA DIRECTOR MOD. BÚFALO |
| 1010       | 1999-12-08   | MESA MODELO UNIÓN          |
| 1011       | 1999-12-15   | ARMARIO NOGAL DOS PUERTAS  |
| 1012       | 1999-12-06   | MESA DESPACHO MOD. GAUIOTA |
| 1013       | 1999-12-06   | ARMARIO NOGAL DOS PUERTAS  |
| 1014       | 2000-01-07   | SILLA DIRECTOR MOD. BÚFALO |
| 1015       | 2000-01-16   | DESTRUCTORA DE PAPEL A3    |

16 rows in set (0.02 sec)



## 5. Composiciones o Combinaciones de una tabla consigo misma

- Si desde una fila de una tabla podemos acceder a otra fila de la misma tabla, duplicando la tabla, estamos realizando una composición o combinación de una tabla consigo misma.
- El formato es el mismo de la combinación o composición pero las tablas del producto son las mismas tablas.
- En este caso, como ambas tablas se llaman igual, es necesario obligatoriamente un alias para diferenciar las columnas de cada tabla



## 5. Composiciones o Combinaciones de una tabla consigo misma

- **Ejemplos**

- Obtener la lista de los empleados con los nombres de sus directores.
  - ```
mysql> SELECT e1.emp_no "Nº Empleado", e1.apellido "Nombre Empleado", e1.dir "Nº Director", e2.apellido "Nombre Director"
FROM empleados e1, empleados e2
WHERE e1.dir=e2.emp_no;
```

+-----+		+-----+		+-----+		+-----+	
N°Empleado		Nombre Empleado		N°Director		Nombre Director	
+-----+		+-----+		+-----+		+-----+	
	7499		ALONSO		7698		GARRIDO
	7521		LOPEZ		7782		MARTINEZ
	7654		MARTIN		7698		GARRIDO
	7698		GARRIDO		7839		REY
	7782		MARTINEZ		7839		REY
	7844		CALVO		7698		GARRIDO
	7876		GIL		7782		MARTINEZ
	7900		JIMENEZ		7782		MARTINEZ
+-----+		+-----+		+-----+		+-----+	



5. Composiciones o Combinaciones de una tabla consigo misma

■ Ejemplos

- Obtener los jefes de los empleados cuyo oficio sea el de VENDEDOR.

- ```
mysql> SELECT e1.emp_no "Nº Empleado", e1.apellido "Nombre Empleado", e1.dir "Nº Director", e2.apellido "Nombre Director"
FROM empleados e1, empleados e2
WHERE e1.dir=e2.emp_no AND e1.oficio = 'VENDEDOR';
```

| NºEmpleado | apellido | oficio   | NºDirector | Nombre Director |
|------------|----------|----------|------------|-----------------|
| 7499       | ALONSO   | VENDEDOR | 7698       | GARRIDO         |
| 7664       | MARTIN   | VENDEDOR | 7698       | GARRIDO         |
| 7844       | CALVO    | VENDEDOR | 7698       | GARRIDO         |



## 6. Composiciones o Combinaciones externas (OUTER JOIN)

- Cuando se realiza una composición o combinación de tablas estableciendo una determinada relación entre sus columnas, puede ocurrir que no se emparejen todas las filas que debieran por faltar correspondencia entre algunas de ellas. Esto se debe a que existen filas en alguna tabla que no tienen correspondencia en la otra tabla y al aplicar la condición de selección de la composición no se seleccionarán.
- **Por ejemplo**
  - `mysql> SELECT dnombre Departamento, loc, emp_no, apellido  
FROM empleados e, departamentos d  
WHERE e.dept_no=d.dept_no;`



## 6. Composiciones o Combinaciones externas (OUTER JOIN)

| Departamento  | localidad | emp_no | apellido |
|---------------|-----------|--------|----------|
| CONTABILIDAD  | BARCELONA | 7521   | LOPEZ    |
| CONTABILIDAD  | BARCELONA | 7782   | MARTINEZ |
| CONTABILIDAD  | BARCELONA | 7839   | REY      |
| INVESTIGACION | VALENCIA  | 7876   | GIL      |
| INVESTIGACION | VALENCIA  | 7900   | JIMENEZ  |
| VENTAS        | MADRID    | 7499   | ALONSO   |
| VENTAS        | MADRID    | 7654   | MARTIN   |
| VENTAS        | MADRID    | 7698   | GARRIDO  |
| VENTAS        | MADRID    | 7844   | CALVO    |

- Aquellos departamentos que no tengan empleados no aparecerán porque para esos departamentos no se cumplirá la igualdad `empleados.num_dep = departamentos.num_dep`.
- A veces es aconsejable que la salida, obtenida por una consulta en la que se pudiera presentar esta posibilidad, muestre todas las filas, aunque algunas con falta de información. Para conseguir este resultado se utiliza la composición o combinación externa (**OUTER JOIN**).



## 6. Composiciones o Combinaciones externas (OUTER JOIN)

- El formato es:
  - **SELECT [ALL/DISTINCT] ExpresionColumna [,ExpresionColumna .....]**  
**FROM NombreTabla [AliasTabla]**  
**[LEFT | RIGHT [OUTER] JOIN NombreTabla [AliasTabla].....]**  
**ON CondicionComposicion**
- Donde **LEFT | RIGHT [OUTER] JOIN:** indica que es un join externo y si la extensión del producto de las tablas se quiere realizar por la izquierda o por la derecha
- **CondicionComposicion:** es la misma condición de composición anterior, pero escrita aquí en lugar de en la cláusula WHERE



## 6. Composiciones o Combinaciones externas (OUTER JOIN)

- El funcionamiento de un join externo es el siguiente:
  - **LEFT JOIN:** join donde se obtienen todas las filas de la tabla de la izquierda, aunque no tenga correspondencia en la tabla de la derecha. Realiza el producto cartesiano de las tablas que se le indican, aplica la condición de composición (expresada en la cláusula ON) al resultado de este producto cartesiano y añade, por cada fila de la tabla de la izquierda que no tenga correspondencia en la tabla de la derecha, una fila con los valores de la tabla de la izquierda y en la tabla de la derecha valores NULL en todas las columnas.





## 6. Composiciones o Combinaciones externas (OUTER JOIN)

- **RIGHT JOIN:** join donde se obtienen todas las filas de la tabla de la derecha, aunque no tengan correspondencia en la tabla de la izquierda. Realiza el producto cartesiano de las tablas que se le indican, aplica la condición de composición (expresada en la cláusula ON) al resultado de este producto cartesiano y añade, por cada fila de la tabla de la derecha que no tenga correspondencia en la tabla de la izquierda, una fila con los valores de la tabla de la derecha y en la tabla de la izquierda valores NULL en todas las columnas.



- | Departamento  | localidad | Nºempleado | apellido |
|---------------|-----------|------------|----------|
| CONTABILIDAD  | BARCELONA | 7521       | LOPEZ    |
| CONTABILIDAD  | BARCELONA | 7782       | MARTINEZ |
| CONTABILIDAD  | BARCELONA | 7839       | REY      |
| INVESTIGACION | VALENCIA  | 7876       | GIL      |
| INVESTIGACION | VALENCIA  | 7900       | JIMENEZ  |
| VENTAS        | MADRID    | 7499       | ALONSO   |
| VENTAS        | MADRID    | 7654       | MARTIN   |
| VENTAS        | MADRID    | 7698       | GARRIDO  |
| VENTAS        | MADRID    | 7844       | CALVO    |
| PRODUCCION    | SEVILLA   | NULL       | NULL     |



- | Departamento  | localidad | Nºempleado | apellido |
|---------------|-----------|------------|----------|
| CONTABILIDAD  | BARCELONA | 7821       | LOPEZ    |
| CONTABILIDAD  | BARCELONA | 7782       | MARTINEZ |
| CONTABILIDAD  | BARCELONA | 7839       | REY      |
| INVESTIGACION | VALENCIA  | 7876       | GIL      |
| INVESTIGACION | VALENCIA  | 7900       | JIMENEZ  |
| VENTAS        | MADRID    | 7499       | ALONSO   |
| VENTAS        | MADRID    | 7654       | MARTIN   |
| VENTAS        | MADRID    | 7698       | GARRIDO  |
| VENTAS        | MADRID    | 7844       | CALVO    |
| PRODUCCION    | SEVILLA   | NULL       | NULL     |



## 6. Composiciones o Combinaciones externas (OUTER JOIN)

- Hay casos en los que hacer el OUTER JOIN obtendremos el mismo resultado con una combinación natural, ya que no hay filas sin correspondencia en la tabla de la correspondencia.
  - ```
mysql> SELECT emp_no "Nº empleado", apellido, dnombre  
          "Departamento", loc  
FROM departamentos d RIGHT JOIN empleados e  
ON d.dept_no=e.dept_no;
```

Nº empleado	apellido	Departamento	localidad
7499	ALONSO	VENTAS	MADRID
7521	LOPEZ	CONTABILIDAD	BARCELONA
7654	MARTIN	VENTAS	MADRID
7698	GARRIDO	VENTAS	MADRID
7782	MARTINEZ	CONTABILIDAD	BARCELONA
7839	REY	CONTABILIDAD	BARCELONA
7844	CALVO	VENTAS	MADRID
7876	GIL	INVESTIGACION	VALENCIA
7900	JIMENEZ	INVESTIGACION	VALENCIA



6. Composiciones o Combinaciones externas (OUTER JOIN)

- Obtener la lista de empleados con los nombres de sus directores, incluyendo al PRESIDENTE. (Ejemplo en autocomposiciones). (Probar a hacer JOIN solo)
 - `mysql> SELECT e1.apellido "Nombre Empleado", e2.apellido "Nombre Director"`

`FROM empleados e1 LEFT JOIN empleados e2
ON e1.dir = e2.emp_no;`

apellido	Nombre Director
ALONSO	GARRIDO
LOPEZ	MARTINEZ
MARTIN	GARRIDO
GARRIDO	REY
MARTINEZ	REY
REY	NULL
CALVO	GARRIDO
GIL	MARTINEZ
JIMENEZ	MARTINEZ