

## Unidad 6. CONSULTA SQL. Multitabla.

### Sumario

6.1. Consultas multitabla SQL-1.....	1
6.1.1. Composición cruzada (producto cartesiano).....	1
6.1.2. Composiciones internas.....	3
6.2. Consultas SQL-2.....	5
6.2.1. Composiciones cruzadas.....	5
6.2.2. Composiciones internas.....	6
6.2.3. Composiciones externas.....	6
6.3. Unir tres o más tablas./.....	8
6.4. Unir una tabla consigo misma. (self-equi-join).....	8

Las consultas multitabla nos permiten consultar información en más de una tabla. La única diferencia respecto a las consultas sencillas es que vamos a tener que especificar en la cláusula FROM cuales son las tablas que vamos a usar y cómo las vamos a relacionar entre sí.

Para realizar este tipo de consultas podemos usar dos alternativas:

- Sintaxis de SQL 1 (SQL-86), que consiste en realizar el producto cartesiano de las tablas y añadir un filtro para relacionar los datos que tienen en común.
- Sintaxis de SQL 2 (SQL-92 y SQL-2003) que incluye todas las cláusulas del tipo JOIN.

### 6.1. Consultas multitabla SQL-1.

#### 6.1.1. Composición cruzada (producto cartesiano).

El producto cartesiano de dos conjuntos, es una operación que consiste en obtener otro conjunto cuyos elementos son todas las parejas que pueden formarse entre los dos conjuntos. Por ejemplo, tendríamos que coger el primer elemento del primer conjunto y formar una pareja con cada uno de los elementos del segundo conjunto. Una vez hecho esto, repetimos el mismo proceso para cada uno de los elementos del primer conjunto.

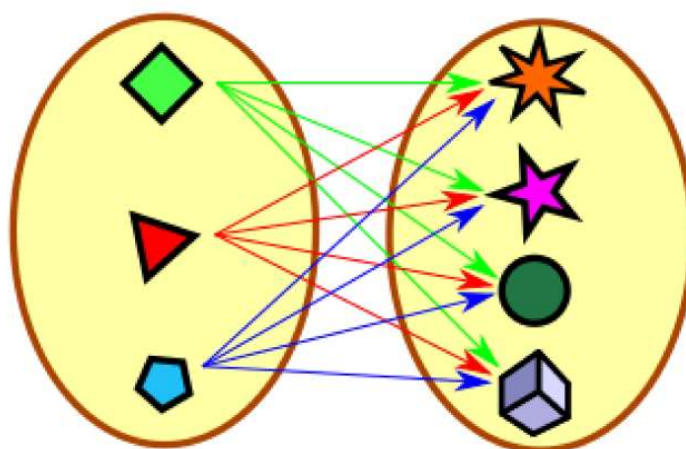


Imagen: Imagen extraída de [Wikipedia](#). Autor: [GermanX](#)

Ejemplo, supongamos las siguientes tablas: empleado y departamento

```
mysql> SELECT codigo, nif, nombre, apellido1, apellido2 FROM empleado;
```

codigo	nif	nombre	apellido1	apellido2
1	20111222x	Julio	Romero	de Torres
2	20111333y	Antonio	Reyna	Manescau
3	20111444z	Juana	de Arcos	NULL
4	20111111a	Ada	Bayron	NULL
5	20222111a	Jimena	Burgos	Madrid
6	20222555b	Remedios	Varo	Sánchez
7	20222666c	Frida	Kalo	NULL
8	20222666c	Antonio	Candela	Heredia
9	20222777r	Maria	Triana	Fuentes
10	20222888h	Fernano	Fernan	Gómez
11	33000111s	Pepe	Ruiz	Santana
12	33000222d	David	Ruiz	Santana
13	33000333e	David	Ruiz	Ruiz
14	33000444R	Pepe	Ruiz	Ruiz

14 rows in set (0.00 sec)

```
mysql> SELECT codigo, nombre, presupuesto FROM departamento;
```

codigo	nombre	presupuesto
1	Sistemas	5600
2	Desarrollo	5600
3	Recursos Humanos	2500
4	Ventas	7000
5	Contabilidad	3000
6	I+D	7800

6 rows in set (0.00 sec)

El producto cartesiano de las dos tablas tendría 84 registros resultantes de combinar cada empleado con los 6 departamentos recogidos.

Veamos como sería el resultado:

```
mysql> SELECT empleado.codigo, empleado.nif, empleado.nombre, empleado.apellido1, empleado.apellido2, departamento.codigo, departamento.nombre, departamento.presupuesto FROM empleado, departamento;
```

codigo	nif	nombre	apellido1	apellido2	codigo	nombre	presupuesto
1	20111222x	Julio	Romero	de Torres	1	Sistemas	5600
1	20111222x	Julio	Romero	de Torres	2	Desarrollo	5600
1	20111222x	Julio	Romero	de Torres	3	Recursos Humanos	2500
1	20111222x	Julio	Romero	de Torres	4	Ventas	7000
1	20111222x	Julio	Romero	de Torres	5	Contabilidad	3000
1	20111222x	Julio	Romero	de Torres	6	I+D	7800
2	20111333y	Antonio	Reyna	Manescau	1	Sistemas	5600
2	20111333y	Antonio	Reyna	Manescau	2	Desarrollo	5600
2	20111333y	Antonio	Reyna	Manescau	3	Recursos Humanos	2500
2	20111333y	Antonio	Reyna	Manescau	4	Ventas	7000
2	20111333y	Antonio	Reyna	Manescau	5	Contabilidad	3000
2	20111333y	Antonio	Reyna	Manescau	6	I+D	7800
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
13	33000333e	David	Ruiz	Ruiz	1	Sistemas	5600
13	33000333e	David	Ruiz	Ruiz	2	Desarrollo	5600
13	33000333e	David	Ruiz	Ruiz	3	Recursos Humanos	2500
13	33000333e	David	Ruiz	Ruiz	4	Ventas	7000
13	33000333e	David	Ruiz	Ruiz	5	Contabilidad	3000
13	33000333e	David	Ruiz	Ruiz	6	I+D	7800
14	33000444R	Pepe	Ruiz	Ruiz	1	Sistemas	5600
14	33000444R	Pepe	Ruiz	Ruiz	2	Desarrollo	5600
14	33000444R	Pepe	Ruiz	Ruiz	3	Recursos Humanos	2500
14	33000444R	Pepe	Ruiz	Ruiz	4	Ventas	7000
14	33000444R	Pepe	Ruiz	Ruiz	5	Contabilidad	3000
14	33000444R	Pepe	Ruiz	Ruiz	6	I+D	7800

84 rows in set (0.00 sec)

Observa que para indicar un atributo de una tabla se pone el nombre de la tabla seguido de un punto y el nombre del atributo. Esto es necesario solo si hay ambigüedad para distinguir que atributo es. En el caso del código es obligatorio hacerlo: departamento.codigo y empleado.codigo; pero con nif, apellido1, apellido2, presupuesto,.. no haría falta pues solo están esos atributos en una de las tablas.

### 6.1.2. Composiciones internas.

La composición interna se obtiene al aplicar una intersección de las tablas que intervienen.

La intersección de dos conjuntos es una operación que da como resultado otro conjunto que contiene sólo los elementos comunes que existen en los conjunto a los que se les ha aplicado la intersección.

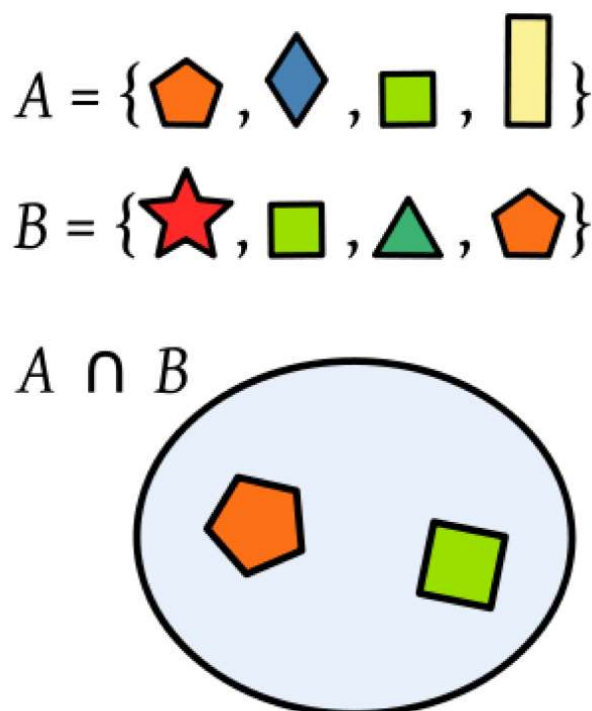


Imagen: Imagen extraída de [Wikipedia](#). Autor: Kismalac.

Para poder realizar una operación de intersección entre dos tablas debemos utilizar la cláusula WHERE para indicar la columna con la que queremos relacionarlas.

Ejemplo: obtener un listado de los empleados y el departamento donde trabaja cada uno de ellos podemos realizar la siguiente consulta.

```
mysql> SELECT empleado.nif, empleado.nombre, empleado.apellido1, empleado.apellido2, departamento.nombre
FROM empleado, departamento WHERE empleado.codigo_departamento=departamento.codigo;
```

nif	nombre	apellido1	apellido2	nombre
20111222x	Julio	Romero	de Torres	Sistemas
20111444z	Juana	de Arcos	NULL	Sistemas
33000333e	David	Ruiz	Ruiz	Sistemas
33000444R	Pepe	Ruiz	Ruiz	Sistemas
20221111a	Jimena	Burgos	Madrid	Desarrollo
20222555b	Remedios	Varo	Sánchez	Desarrollo
33000111s	Pepe	Ruiz	Santana	Desarrollo
20222666c	Frida	Kalo	NULL	Recursos Humanos
20222666c	Antonio	Candela	Heredia	Recursos Humanos
33000222d	David	Ruiz	Santana	Recursos Humanos
20111333y	Antonio	Reyna	Manescau	I+D
20111111a	Ada	Bayron	NULL	I+D

12 rows in set (0.00 sec)

En la operación de intersección sólo obtendremos los elementos que existan en ambos conjuntos. Por lo tanto, puede ocurrir que haya filas en la tabla de empleado que no aparezcan en el resultado y filas de la tabla de departamento que tampoco estén. Serán empleados que no estén asignados a ningún departamento y departamentos que no tengan asignado a ningún empleado.

Comparando la solución SQL-1 con SQL-2 tendríamos:

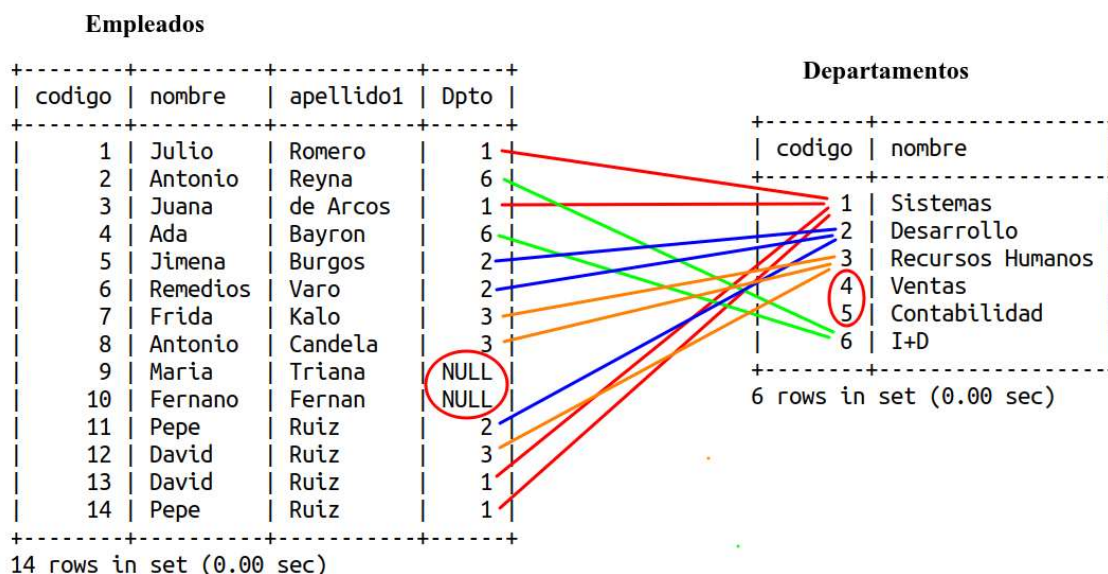
SQL-1:

```
mysql>SELECT empleado.nombre, empleado.apellido1, departamento.nombre
FROM empleado, departamento
WHERE empleado.codigo_departamento=departamento.codigo;
```

SQL-2:

```
mysql>SELECT empleado.nombre, empleado.apellido1, departamento.nombre
FROM empleado INNER JOIN departamento
ON empleado.codigo_departamento=departamento.codigo;
```

En ambas soluciones se mostrarán solo los registros relacionados y no aparecerán los empleados que no pertenezcan a ningún departamento, ni los departamentos sin empleados:



No aparecerán en la intersección:

- Las filas de empleado:
  - Maria Triana
  - Fernando Fernan
- Las filas de departamento:
  - Ventas
  - Contabilidad

Los resultados obtenidos son independientes del orden en el que se pongan las tablas:

```
mysql>SELECT empleado.nombre, empleado.apellido1, departamento.nombre
      FROM empleado INNER JOIN departamento
      ON empleado.codigo_departamento=departamento.codigo;
```

es equivalente a:

```
mysql>SELECT empleado.nombre, empleado.apellido1, departamento.nombre
      FROM departamento INNER JOIN empleado
      ON empleado.codigo_departamento=departamento.codigo;
```

codigo	nombre	apellido1	Dpto	codigo	nombre
1	Julio	Romero	1	1	Sistemas
3	Juana	de Arcos	1	1	Sistemas
13	David	Ruiz	1	1	Sistemas
14	Pepe	Ruiz	1	1	Sistemas
5	Jimena	Burgos	2	2	Desarrollo
6	Remedios	Varo	2	2	Desarrollo
11	Pepe	Ruiz	2	2	Desarrollo
7	Frida	Kalo	3	3	Recursos Humanos
8	Antonio	Candela	3	3	Recursos Humanos
12	David	Ruiz	3	3	Recursos Humanos
2	Antonio	Reyna	6	6	I+D
4	Ada	Bayron	6	6	I+D

12 rows in set (0.00 sec)

## 6.2. Consultas SQL-2.

### 6.2.1. Composiciones cruzadas.

La composición cruzada (producto cartesiano) se hace con la cláusula: [CROSS] JOIN. La palabra CROSS es opcional.



```
mysql> SELECT empleado.codigo, empleado.nif, empleado.nombre, empleado.apellido1, empleado.apellido2,
departamento.codigo, departamento.nombre, departamento.presupuesto FROM empleado CROSS JOIN departamen
to;
```

codigo	nif	nombre	apellido1	apellido2	codigo	nombre	presupuesto
1	20111222x	Julio	Romero	de Torres	1	Sistemas	5600
1	20111222x	Julio	Romero	de Torres	2	Desarrollo	5600
1	20111222x	Julio	Romero	de Torres	3	Recursos Humanos	2500
1	20111222x	Julio	Romero	de Torres	4	Ventas	7000
1	20111222x	Julio	Romero	de Torres	5	Contabilidad	3000
1	20111222x	Julio	Romero	de Torres	6	I+D	7800
2	20111333y	Antonio	Reyna	Manescau	1	Sistemas	5600
2	20111333y	Antonio	Reyna	Manescau	2	Desarrollo	5600
2	20111333y	Antonio	Reyna	Manescau	3	Recursos Humanos	2500
2	20111333y	Antonio	Reyna	Manescau	4	Ventas	7000
2	20111333y	Antonio	Reyna	Manescau	5	Contabilidad	3000
2	20111333y	Antonio	Reyna	Manescau	6	I+D	7800
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
13	33000333e	David	Ruiz	Ruiz	1	Sistemas	5600
13	33000333e	David	Ruiz	Ruiz	2	Desarrollo	5600
13	33000333e	David	Ruiz	Ruiz	3	Recursos Humanos	2500
13	33000333e	David	Ruiz	Ruiz	4	Ventas	7000
13	33000333e	David	Ruiz	Ruiz	5	Contabilidad	3000
13	33000333e	David	Ruiz	Ruiz	6	I+D	7800
14	33000444R	Pepe	Ruiz	Ruiz	1	Sistemas	5600
14	33000444R	Pepe	Ruiz	Ruiz	2	Desarrollo	5600
14	33000444R	Pepe	Ruiz	Ruiz	3	Recursos Humanos	2500
14	33000444R	Pepe	Ruiz	Ruiz	4	Ventas	7000
14	33000444R	Pepe	Ruiz	Ruiz	5	Contabilidad	3000
14	33000444R	Pepe	Ruiz	Ruiz	6	I+D	7800

84 rows in set (0.00 sec)

Se obtiene el mismo resultado que con la opción SQL-1

## 6.2.2. Composiciones internas.

Para realizar una composición interna o intersección se disponen de dos opciones:

- [INNER] JOIN tabla ON condición. La palabra INNER es opcional. La cláusula ON nos permite especificar la intersección, si no se pone estaríamos ante una composición cruzada (producto cartesiano).
- NATURAL JOIN tabla. En este caso se devuelve la intersección de las dos tablas, pero utilizando para ello las columnas cuyos nombres coincidan, es por ello que no necesita la cláusula ON. De nuestro ejemplo, código y nombre son atributos que aparecen en las dos tablas, mostraría los registros cuyo código de empleado coincida con el código de departamento y que el nombre de empleado coincida con el nombre de departamento, es decir, ninguno.

Para usar la cláusula NATURAL, se ha de estar seguro de que las columnas por las que vamos a realizar la intersección se llaman igual en las dos tablas. Lo normal es que esto no ocurra, por lo que lo más utilizado es INNER JOIN o JOIN.

## 6.2.3. Composiciones externas.

En las composiciones externas se incluyen todos los registros de la tabla que aparece a la derecha o a la izquierda, según la opción, de la cláusula JOIN, estén o no relacionados con algún registro de la otra tabla.

Las distintas composiciones externas son:

- LEFT [OUTER] JOIN.
- RIGHT [OUTER] JOIN.
- FULL [OUTER] JOIN (no implementado en MySQL).
- NATURAL LEFT [OUTER] JOIN.
- NATURAL RIGHT [OUTER] JOIN.

Al usar “empleado LEFT JOIN departamento” mostrará la intersección y todos los registros que aparecen en la tabla a la izquierda de JOIN: empleado. Se incluye: María Triana y Fernando Fernan, sin embargo no aparecen los registros de departamentos que no tienen relación: “ni Ventas ni Contabilidad”.

```
mysql> SELECT empleado.codigo, empleado.nif, empleado.nombre, empleado.apellido1, empleado.apellido2, departamento.codigo, departamento.nombre, departamento.presupuesto FROM empleado LEFT JOIN departamento ON empleado.codigo_departamento=departamento.codigo;
```

codigo	nif	nombre	apellido1	apellido2	codigo	nombre	presupuesto
1	20111222x	Julio	Romero	de Torres	1	Sistemas	5600
2	20111333y	Antonio	Reyna	Manescau	6	I+D	7800
3	20111444z	Juana	de Arcos	NULL	1	Sistemas	5600
4	20111111a	Ada	Bayron	NULL	6	I+D	7800
5	20222111a	Jimena	Burgos	Madrid	2	Desarrollo	5600
6	20222555b	Remedios	Varo	Sánchez	2	Desarrollo	5600
7	20222666c	Frida	Kalo	NULL	3	Recursos Humanos	2500
8	20222666c	Antonio	Candela	Heredia	3	Recursos Humanos	2500
9	20222777r	Maria	Triana	Fuentes	NULL	NULL	NULL
10	20222888h	Fernando	Fernan	Gómez	NULL	NULL	NULL
11	33000111s	Pepe	Ruiz	Santana	2	Desarrollo	5600
12	33000222d	David	Ruiz	Santana	3	Recursos Humanos	2500
13	33000333e	David	Ruiz	Ruiz	1	Sistemas	5600
14	33000444R	Pepe	Ruiz	Ruiz	1	Sistemas	5600

14 rows in set (0.00 sec)

En el caso de “empleado RIGHT JOIN departamento”, además de los registros de la intersección se incluyen todos los que son de la tabla a de derecha del JOIN. Ahora, aparecen: “Ventas y Contabilidad”, pero no aparecen “ni María Triana ni Fernando Fernan”.

```
mysql> mysql> SELECT empleado.codigo, empleado.nif, empleado.nombre, empleado.apellido1, empleado.apellido2, departamento.codigo, departamento.nombre, departamento.presupuesto FROM empleado RIGHT JOIN departamento ON empleado.codigo_departamento=departamento.codigo;
```

codigo	nif	nombre	apellido1	apellido2	codigo	nombre	presupuesto
1	20111222x	Julio	Romero	de Torres	1	Sistemas	5600
3	20111444z	Juana	de Arcos	NULL	1	Sistemas	5600
13	33000333e	David	Ruiz	Ruiz	1	Sistemas	5600
14	33000444R	Pepe	Ruiz	Ruiz	1	Sistemas	5600
5	20222111a	Jimena	Burgos	Madrid	2	Desarrollo	5600
6	20222555b	Remedios	Varo	Sánchez	2	Desarrollo	5600
11	33000111s	Pepe	Ruiz	Santana	2	Desarrollo	5600
7	20222666c	Frida	Kalo	NULL	3	Recursos Humanos	2500
8	20222666c	Antonio	Candela	Heredia	3	Recursos Humanos	2500
12	33000222d	David	Ruiz	Santana	3	Recursos Humanos	2500
NULL	NULL	NULL	NULL	NULL	4	Ventas	7000
NULL	NULL	NULL	NULL	NULL	5	Contabilidad	3000
2	20111333y	Antonio	Reyna	Manescau	6	I+D	7800
4	20111111a	Ada	Bayron	NULL	6	I+D	7800

14 rows in set (0.00 sec)

La consulta FULL OUTER JOIN no está implementada en MySQL, pero se podría simular con el operador UNION, que realizaría la unión de dos consultas que se hagan. Se aplicará a una consulta LEFT JOIN y otra RIGHT JOIN.

El resultado de FULL JOIN es obtener la intersección de dos tablas y todos los elementos de cada tabla que no estén en la intersección. En nuestro caso de tabla empleados y departamentos deben aparecer:

- Registros de la intersección. Es decir, empleados que están asignados a un departamento. Los datos del empleado y del departamento, que se especifiquen.
- Registros de empleados que no están asignados a ningún departamento. En este caso aparecerán en NULL los campos del departamento.
- Registros de departamentos que no tienen empleados. Igualmente aparecerán en NULL los atributos de empleado.

```
mysql> SELECT empleado.codigo, nif, empleado.nombre, apellido1, apellido2, departamento.codigo, departamento.nombre, presupuesto FROM empleado LEFT JOIN departamento ON empleado.codigo_departamento=departamento.codigo UNION SELECT empleado.codigo, nif, empleado.nombre, apellido1, apellido2, departamento.codigo, departamento.nombre, departamento.presupuesto FROM empleado RIGHT JOIN departamento ON empleado.codigo_departamento=departamento.codigo;
```

codigo	nif	nombre	apellido1	apellido2	codigo	nombre	presupuesto
1	20111222x	Julio	Romero	de Torres	1	Sistemas	5600
2	20111333y	Antonio	Reyna	Manescau	6	I+D	7800
3	20111444z	Juana	de Arcos	NULL	1	Sistemas	5600
4	20111111a	Ada	Bayron	NULL	6	I+D	7800
5	20222111a	Jimena	Burgos	Madrid	2	Desarrollo	5600
6	20222555b	Remedios	Varo	Sánchez	2	Desarrollo	5600
7	20222666c	Frida	Kalo	NULL	3	Recursos Humanos	2500
8	20222666c	Antonio	Candela	Heredia	3	Recursos Humanos	2500
9	20222777r	Maria	Triana	Fuentes	NULL	NULL	NULL
10	20222888h	Fernano	Fernan	Gómez	NULL	NULL	NULL
11	33000111s	Pepe	Ruiz	Santana	2	Desarrollo	5600
12	33000222d	David	Ruiz	Santana	3	Recursos Humanos	2500
13	33000333e	David	Ruiz	Ruiz	1	Sistemas	5600
14	33000444r	Pepe	Ruiz	Ruiz	1	Sistemas	5600
NULL	NULL	NULL	NULL	NULL	4	Ventas	7000
NULL	NULL	NULL	NULL	NULL	5	Contabilidad	3000

16 rows in set (0.00 sec)

Respecto a las composiciones con NATURAL LEFT JOIN y NATURAL RIGHT JOIN, Hay que tener en cuenta que la composición se hará sobre columnas que se llamen igual, luego no hace falta la cláusula ON, igualmente se incluirán en la consulta los registros que no coincidan en la intersección de la tabla a la izquierda o de la tabla a la derecha, respectivamente.

## 6.3. Unir tres o más tablas./

La unión de más de dos tablas se realiza añadiendo cláusulas JOIN.

Ejemplo:

```
mysql> SELECT * FROM cliente
      INNER JOIN empleado ON cliente.codigo_empleado_venta=empleado.codigo
      INNER JOIN pago ON cliente.codigo=pago.codigo_cliente;
```

## 6.4. Unir una tabla consigo misma. (self-equi-join)

Para poder hacer una intersección de una tabla consigo misma, es necesario crear un alias. El alias se crea con la cláusula AS.

Añadamos una columna a la tabla de empleados indicando el jefe de cada departamento:



```
mysql> SELECT codigo, nif, nombre, apellido1, apellido2, codigo_departamento AS Dpto, codigo_jefe AS jefe from empleado ORDER BY codigo_departamento, apellido1;
```

codigo	nif	nombre	apellido1	apellido2	Dpto	jefe
10	20222888h	Fernano	Fernan	Gómez	NULL	NULL
9	20222777r	Maria	Triana	Fuentes	NULL	NULL
3	20111444z	Juana	de Arcos	NULL	1	NULL
1	20111222x	Julio	Romero	de Torres	1	3
13	33000333e	David	Ruiz	Ruiz	1	3
14	33000444R	Pepe	Ruiz	Ruiz	1	3
5	20222111a	Jimena	Burgos	Madrid	2	NULL
11	33000111s	Pepe	Ruiz	Santana	2	5
6	20222555b	Remedios	Varo	Sánchez	2	5
8	20222666c	Antonio	Candela	Heredia	3	NULL
7	20222666c	Frida	Kalo	NULL	3	8
12	33000222d	David	Ruiz	Santana	3	8
4	20111111a	Ada	Bayron	NULL	6	NULL
2	20111333y	Antonio	Reyna	Manescau	6	4

14 rows in set (0.00 sec)

Ejemplo, mostrar un listado donde aparezca de cada empleado y quien es su jefe:

```
mysql> SELECT empleado.nombre, empleado.apellido1, empleado.apellido2, jefe.nombre, jefe.apellido1, jefe.apellido2
FROM empleado INNER JOIN empleado AS jefe
ON empleado.codigo_jefe=jefe.codigo;
```

```
mysql> SELECT empleado.codigo AS Cod, empleado.nombre AS Nombre, empleado.apellido1 AS Apellido1, empleado.apellido2 AS Apellido2, jefe.nombre, jefe.apellido1, jefe.apellido2 FROM empleado JOIN empleado AS jefe ON empleado.codigo_jefe=jefe.codigo;
```

Cod	Nombre	Apellido1	Apellido2	nombre	apellido1	apellido2
1	Julio	Romero	de Torres	Juana	de Arcos	NULL
2	Antonio	Reyna	Manescau	Ada	Bayron	NULL
6	Remedios	Varo	Sánchez	Jimena	Burgos	Madrid
7	Frida	Kalo	NULL	Antonio	Candela	Heredia
11	Pepe	Ruiz	Santana	Jimena	Burgos	Madrid
12	David	Ruiz	Santana	Antonio	Candela	Heredia
13	David	Ruiz	Ruiz	Juana	de Arcos	NULL
14	Pepe	Ruiz	Ruiz	Juana	de Arcos	NULL

8 rows in set (0.00 sec)