

A4. Consultas entre varias tablas.

Índice.

1. Introducción a las consultas entre varias tablas.....	2
2. Consultas multitable SQL 1.....	3
2.1. Composiciones cruzadas (producto cartesiano).....	3
2.2. Composiciones internas (intersección).....	3
3. Consultas multitable SQL 2.....	4
3.1. Composiciones cruzadas.....	4
3.2. Composiciones internas.....	4
3.3. Composiciones externas.....	5

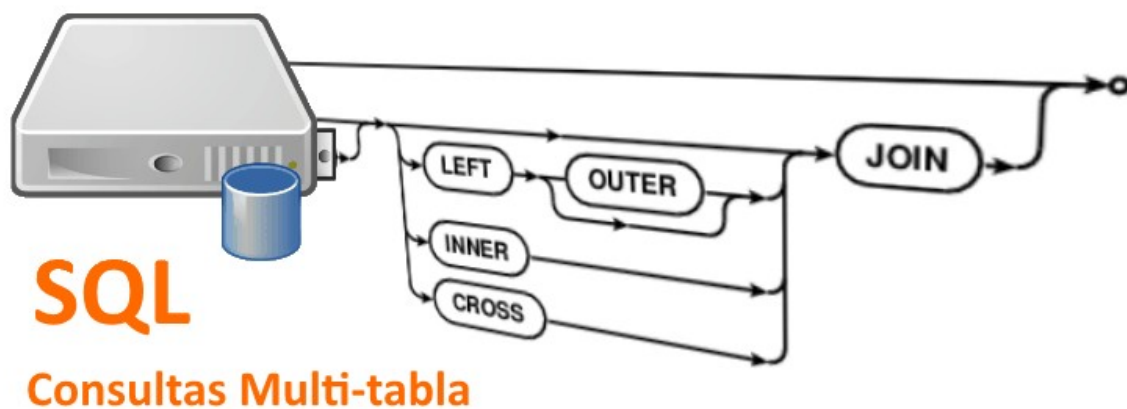
1. Introducción a las consultas entre varias tablas.

Una consulta entre varias tablas (o multitabla) es la que se produce al consultar información contenida en más de una tabla, aprovechándonos de los campos relacionados que haya entre las tablas para unirlos (join).

La única diferencia con las consultas sencillas se halla en la cláusula FROM, en la que se especifica las tablas que se van a usar y cómo se van a relacionar.

Este tipo de consultas se puede realizar a través de dos alternativas:

- Sintaxis de SQL 1(SQL-86) → se realiza el producto cartesiano de las tablas y se añade un filtro para los datos que hay en común.
- Sintaxis de SQL 2(SQL-92 y SQL-2003) → incluye todas las cláusulas de tipo JOIN.



2. Consultas multitabla SQL 1.

Supongamos que tenemos las siguientes dos tablas Empleado y Departamento con los siguientes datos:

Empleado							Departamento				
	codEmpleado	NIF	nombre	apellido1	apellido2	codigo		codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	▶	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2		2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2		3	Recursos Humanos	280000	25000

2.1. Composiciones cruzadas (producto cartesiano).

El **producto cartesiano** de dos conjuntos (o tablas) es una operación que consiste en obtener otro conjunto cuyos elementos son todas las parejas que se pueden formar entre los elementos de los dos conjuntos.

```
SELECT * FROM Empleado, Departamento;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	1	Desarrollo	120000	6000
	3	55756324Y	Rosa María	Granero	Reyes	2	1	Desarrollo	120000	6000
	1	35323825B	Sancho	García	Rodríguez	1	2	Sistemas	150000	21000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000
	1	35323825B	Sancho	García	Rodríguez	1	3	Recursos Humanos	280000	25000
	2	32481595F	Federico	Rivero	Gómez	2	3	Recursos Humanos	280000	25000
	3	55756324Y	Rosa María	Granero	Reyes	2	3	Recursos Humanos	280000	25000

Esta operación genera un conjunto de resultados con todas las combinaciones posibles entre las filas de las dos tablas y con todas las columnas.

2.2. Composiciones internas (intersección).

La **intersección** de dos conjuntos es una operación que resulta en otro conjunto y que contiene sólo los elementos comunes que existen en ambos conjuntos.

```
SELECT * FROM Empleado, Departamento
WHERE Empleado.codigo = Departamento.codigo;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000

Esta operación genera un conjunto de resultados con todas las combinaciones posibles entre las filas de las dos tablas y con todas las columnas, al que se aplica un filtro a través de la condición WHERE.

3. Consultas multitabla SQL 2.

Seguimos con las dos tablas anteriores.

Empleado						Departamento					
	codEmpleado	NIF	nombre	apellido1	apellido2	codigo		codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	▶	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2		2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2		3	Recursos Humanos	280000	25000

3.1. Composiciones cruzadas: CROSS JOIN.

La **composición cruzada** devuelve el producto cartesiano de dos tablas.

```
SELECT * FROM Empleado CROSS JOIN Departamento;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	1	Desarrollo	120000	6000
	3	55756324Y	Rosa María	Granero	Reyes	2	1	Desarrollo	120000	6000
	1	35323825B	Sancho	García	Rodríguez	1	2	Sistemas	150000	21000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000
	1	35323825B	Sancho	García	Rodríguez	1	3	Recursos Humanos	280000	25000
	2	32481595F	Federico	Rivero	Gómez	2	3	Recursos Humanos	280000	25000
	3	55756324Y	Rosa María	Granero	Reyes	2	3	Recursos Humanos	280000	25000

3.2. Composiciones internas.

Las composiciones internas pueden ser de dos tipos:

- **[INNER] JOIN**

Esta composición devuelve la intersección entre las dos tablas.

```
SELECT * FROM Empleado INNER JOIN Departamento
ON Empleado.codigo = Departamento.codigo;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000

- **NATURAL JOIN**

Esta composición devuelve la intersección de las dos tablas, pero utiliza las columnas con el mismo nombre para relacionarlas. Sólo se deberá utilizar si estamos seguros que los nombres de las columnas que se desea relacionar se llaman igual en ambas tablas.

```
SELECT * FROM Empleado NATURAL JOIN Departamento;
```

3.3. Composiciones externas.

Las composiciones externas pueden ser de los siguientes tipos:

- **LEFT OUTER JOIN**

Esta consulta devuelve las filas de la tabla que se han ubicado a la izquierda de la composición (Empleado), relacionándolas con las filas de la tabla de la derecha (Departamento) con las que encuentre coincidencia.

Si no hay ninguna coincidencia, se mostrará los de las filas de la tabla de la izquierda (Empleado) y en los valores de la tabla de la derecha, aparecerá el valor NULL.

```
SELECT * FROM Empleado LEFT OUTER JOIN Departamento
ON Empleado.codigo = Departamento.codigo;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000

- **RIGHT OUTER JOIN**

Esta consulta devuelve las filas de la tabla que se han ubicado a la derecha de la composición (Empleado), relacionándolas con las filas de la tabla de la izquierda (Departamento) con las que encuentra coincidencia.

Si no hay ninguna coincidencia, se mostrará los de las filas de la tabla de la derecha (Departamento) y en los valores de la tabla de la izquierda, aparecerá el valor NULL.

```
SELECT * FROM Empleado RIGHT OUTER JOIN Departamento
ON Empleado.codigo = Departamento.codigo;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000
	NULL	NULL	NULL	NULL	NULL	3	3	Recursos Humanos	280000	25000

- **FULL OUTER JOIN (no implementada en MySQL)**

Esta consulta devuelve la intersección de dos tablas junto con las filas que no se pueden combinar → unión entre una consulta de tipo LEFT JOIN y una consulta RIGHT JOIN sobre las mismas tablas.

```
SELECT * FROM Empleado FULL JOIN Departamento;
```

Esta consulta no está implementada en MySQL, pero se puede construir utilizando el operador UNION.

Para poder utilizar el operador UNION entre varias consultas hay que tener en cuenta los siguientes aspectos:

- Las tablas deben tener el mismo número de columnas.
- Las columnas que se van a unir han de tener tipos de datos similares.

```
SELECT * FROM Empleado LEFT JOIN Departamento ON Empleado.codigo = Departamento.codigo
UNION
SELECT * FROM Empleado RIGHT JOIN Departamento ON Empleado.codigo = Departamento.codigo;
```

	codEmpleado	NIF	nombre	apellido1	apellido2	codigo	codigo	nombre	presupuesto	gastos
▶	1	35323825B	Sancho	García	Rodríguez	1	1	Desarrollo	120000	6000
	2	32481595F	Federico	Rivero	Gómez	2	2	Sistemas	150000	21000
	3	55756324Y	Rosa María	Granero	Reyes	2	2	Sistemas	150000	21000
	NULL	NULL	NULL	NULL	NULL	3	3	Recursos Humanos	280000	25000

UD 5. REALIZACIÓN DE CONSULTAS

- NATURAL LEFT OUTER JOIN**

En esta consulta, se compara todas las columnas que tengan el mismo nombre en ambas tablas y se devuelve sólo una, en este caso la de la tabla de la izquierda (Empleado). Los demás datos de la tabla de la derecha (Departamento) se omiten con NULL.

```
SELECT * FROM Empleado NATURAL LEFT JOIN Departamento;
```

	nombre	codigo	codEmpleado	NIF	apellido1	apellido2	presupuesto	gastos
▶	Sancho	1	1	35323825B	García	Rodríguez	NULL	NULL
	Federico	2	2	32481595F	Rivero	Gómez	NULL	NULL
	Rosa María	2	3	55756324Y	Granero	Reyes	NULL	NULL

- NATURAL RIGHT OUTER JOIN**

En esta consulta, se compara todas las columnas que tengan el mismo nombre en ambas tablas y se devuelve sólo una, en este caso la de la tabla de la derecha (Departamento). Los demás datos de la tabla de la izquierda (Empleado) se omiten con NULL.

```
SELECT * FROM Empleado NATURAL RIGHT JOIN Departamento;
```

	codigo	nombre	presupuesto	gastos	codEmpleado	NIF	apellido1	apellido2
▶	1	Desarrollo	120000	6000	NULL	NULL	NULL	NULL
	2	Sistemas	150000	21000	NULL	NULL	NULL	NULL
	3	Recursos Humanos	280000	25000	NULL	NULL	NULL	NULL