

TEMA 07 – CONSULTAS SOBRE VARIAS TABLAS. COMPOSICIÓN INTERNA Y CRUZADA

Consultas multitabla

- Inner join

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

- Inner es opcional
- Si omitimos el “on”, nos devuelve el producto cartesiano
- Cuando se llaman igual la PK y la FK

```
1 SELECT *
2 FROM empleado INNER JOIN departamento
3 USING (codigo_departamento)
```

- Natural join
 - Cuando se llaman igual la PK y la FK se puede automatizar todavía más:

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

- Composiciones externas
 - “LEFT OUTER JOIN”
 - Devolverá todas las filas de la tabla que hemos colocado a la izquierda de la composición
 - Relacionará las filas de la tabla izquierda con las filas de la tabla de la derecha con las que encuentre una coincidencia
 - Si no encuentra ninguna coincidencia, se mostrarán los valores de la fila de la tabla izquierda y “null” en la derecha

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

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



El resultado de la operación LEFT JOIN es:

3

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

- "RIGHT OUTER JOIN"
 - Devolverá todas las filas de la tabla que hemos colocado a la derecha de la composición
 - Relacionará las filas de la tabla derecha con las filas de la tabla de la izquierda con las que encuentre una coincidencia

- Si no encuentra ninguna coincidencia, se mostrarán los valores de la fila de la tabla derecha y “null” en la derecha

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

1

/* SQL 2 */

```
SELECT *
FROM empleado RIGHT 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**



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

3

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

- “FULL OUTER JOIN”
 - No implementada en MySQL
 - Para poder simularla, será necesario el uso del operador “UNION”
 - El resultado de una composición de tipo “FULL OUTER JOIN” es obtener la intersección de las dos tablas, junto a las filas de ambas tablas que no se puedan combinar

```
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
```

- “NATURAL LEFT JOIN”
 - Realiza un “LEFT JOIN” entre las dos tablas
 - No es necesario utilizar la cláusula “ON”
 - Cuando los nombres de las columnas sobre las que queremos relacionar las 2 tablas se llamen igual en las 2 tablas
 - Vamos, igual que el “NATURAL JOIN”

Unir 3 o más tablas

```
1 SELECT *
2 FROM cliente INNER JOIN empleado
3 ON cliente.codigo_empleado_rep_ventas = empleado.codigo_empleado
4 INNER JOIN pago
5 ON cliente.codigo_cliente = pago.codigo_cliente;
```

Unir una tabla consigo misma (self-join)

```
1 SELECT empleado.nombre, empleado.apellido1, empleado.apellido2, jefe.nombre,
   jefe.apellido1, jefe.apellido2
2 FROM empleado INNER JOIN empleado AS jefe
3 ON empleado.codigo_jefe = jefe.codigo_empleado
```

```
1 SELECT empleado.nombre, empleado.apellido1, empleado.apellido2, jefe.nombre,
   jefe.apellido1, jefe.apellido2
2 FROM empleado INNER JOIN empleado jefe
3 ON empleado.codigo_jefe = jefe.codigo_empleado
```

Ejemplo: Cómo unir una tabla consigo misma varias veces.

```
1 SELECT empleado.*, jefe.*, super_jefe.*
2 FROM empleado INNER JOIN empleado AS jefe
3 ON empleado.codigo_jefe = jefe.codigo_empleado
4 INNER JOIN empleado AS super_jefe
5 ON jefe.codigo_jefe = super_jefe.codigo_empleado
```

```
1 SELECT empleado.*, jefe.*, super_jefe.*
2 FROM empleado INNER JOIN empleado jefe
3 ON empleado.codigo_jefe = jefe.codigo_empleado
4 INNER JOIN empleado super_jefe
5 ON jefe.codigo_jefe = super_jefe.codigo_empleado
```

Unir tablas de diferentes bases de datos

```
1 SELECT *
2 FROM db1.tabla1 AS t1 INNER JOIN db2.tabla2 AS t2
3 ON t1.codigo = t2.codigo
```

Uniones equivalentes (equi-joins) y no equivalentes (non-equi-joins)