

Módulo de adaptación

Master en Business Intelligence y Big Data

PROFESOR/A
Antonio Sarasa Cabezuelo

Consultas sobre más de una tabla

Consulta de más de una tabla.

- En la cláusula FROM es posible especificar más de una tabla cuándo se quieren consultar columnas de tablas diferentes.
- Existen varios casos:
 - **Combinación.** Se crea una sola tabla a partir de las tablas especificadas, haciendo coincidir los valores de las columnas relacionadas de las tablas.

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla1 JOIN tabla2  
{ON condiciones|USING (columna [, columna...])}  
[WHERE condiciones];
```

Consulta de más de una tabla.

- La opción ON permite expresar condiciones con cualquiera de los operadores de comparación sobre las columnas especificadas.
- Es posible utilizar una misma tabla dos veces usando alias diferentes para diferenciarlas.
- Puede ocurrir que las tablas consideradas tengan columnas con los mismos nombres. En este caso es obligatorio diferenciarlas especificando en cada columna a que tabla pertenecen.

Consulta de más de una tabla.

- Por ejemplo se quiere obtener el nif del cliente y el precio de los proyectos desarrollados para el cliente con código 30.

```
SELECT p.precio, c.nif FROM clientes c JOIN proyectos p  
ON c.codigo_cli = p.codigo_cliente WHERE c.codigo_cli = 30;
```

- Alternativamente se podría obtener con la siguiente consulta:

```
SELECT p.precio, c.nif FROM clientes c, proyectos p  
WHERE c.codigo_cli = p.codigo_cliente AND c.codigo_cli = 20;
```

Consulta de más de una tabla.

- Por ejemplo si se quieren los códigos de los proyectos que son más caros que el proyecto con código 30

```
SELECT p1.codigo_proyec  
FROM proyectos p1 JOIN proyectos p2 ON p1.precio > p2.precio  
WHERE p2.codigo_proyec = 30;
```

Consulta de más de una tabla.

- **Combinación natural.** Consiste en una combinación en la que se eliminan las columnas repetidas.

```
SELECT nombre_columnas_a_seleccionar  
FROM tabla1 NATURAL JOIN tabla2 [WHERE condiciones];
```

- Por ejemplo si se quiere obtener los empleados cuyo departamento se encuentra situado en Madrid:

```
SELECT * FROM empleados NATURAL JOIN departamentos  
WHERE ciudad = 'Madrid';
```

- De forma equivalente se podría consultar:

```
SELECT * FROM empleados JOIN departamentos  
USING (nombre_dep, ciudad_dep) WHERE ciudad = 'Madrid';
```

Consulta de más de una tabla.

- **Combinación interna vs externa.**
 - Interna(inner join). Sólo se consideran las filas que tienen valores idénticos en las columnas de las tablas que compara.

```
SELECT nombre_columnas_a_seleccionar  
FROM t1 [NATURAL] [INNER] JOIN t2  
  {ON condiciones | | USING (columna [,columna...])}  
  [WHERE condiciones];
```

- Externa(outer join). Se consideran los valores de la tabla derecha, de la izquierda o de ambas tablas.

```
SELECT nombre_columnas_a_seleccionar  
FROM t1 [NATURAL] [LEFT | RIGHT | FULL] [OUTER] JOIN t2  
  {ON condiciones | [USING (columna [,columna...])}  
  [WHERE condiciones];
```


Consulta de más de una tabla.

- **Combinación de más de 2 tablas.** Para combinar más de 2 tablas basta añadirlas en el FROM de la consulta y establecer las relaciones necesarias en el WHERE, o bien combinar las tablas por pares de manera que la resultante es el primer componente del siguiente par.
- Por ejemplo si se quieren combinar las tablas empleados, proyectos y clientes:

```
SELECT * FROM empleados, proyectos, clientes  
WHERE num_proyec = codigo_proyec AND codigo_cliente =  
codigo_cli;
```

O bien:

```
SELECT *  
FROM (empleados JOIN proyectos ON num_proyec =  
codigo_proyec) JOIN clientes ON codigo_cliente = codigo_cli;
```

Operaciones entre consultas

- Entre 2 tablas se pueden definir las siguientes operaciones:
 - **Unión.** Permite unir los resultados de 2 o más consultas.

```
SELECT columnas FROM tabla [WHERE condiciones]  
UNION [ALL]  
SELECT columnas FROM tabla[WHERE condiciones];
```

Observar que la cláusula ALL indica si se quieren obtener todas las filas de la unión(incluidas las repetidas)

Operaciones entre consultas

- Por ejemplo si se quiere obtener todas las ciudades que aparecen en las tablas de la base de datos:

```
SELECT ciudad FROM clientes  
UNION  
SELECT ciudad_dep FROM departamentos;
```

Operaciones entre consultas

- **Intersección.** Permite hacer la intersección entre los resultados de 2 o más consultas.

```
SELECT columnas FROM tabla [WHERE condiciones]  
INTERSECT [ALL]  
SELECT columnas FROM tabla [WHERE condiciones];
```

Observar:

- La cláusula ALL indica si se quieren obtener todas las filas de la intersección(incluidas las repetidas)

Operaciones entre consultas

- Se puede simular usando:

- **IN**

```
SELECT columnas FROM tabla WHERE columna  
IN (SELECT columna FROM tabla [WHERE condiciones]);
```

- **EXISTS**

```
SELECT columnas FROM tabla WHERE EXISTS (SELECT *  
FROM tabla WHERE condiciones);
```

Operaciones entre consultas

- Por ejemplo si se quiere saber las ciudades de los clientes en las que hay departamentos:
 - Usando la intersección.

```
SELECT ciudad FROM clientes INTERSECT  
SELECT ciudad_dep FROM departamentos;
```

- Usando IN

```
SELECT c.ciudad FROM clientes c WHERE c.ciudad IN  
(SELECT d.ciudad_dep FROM departamentos d);
```

- Usando EXISTS

```
SELECT c.ciudad FROM clientes c WHERE EXISTS (SELECT *  
FROM departamentos d WHERE c.ciudad = d.ciudad_dep;
```

Operaciones entre consultas

- **Diferencia.** Permite hacer la diferencia entre los resultados de 2 o más consultas.

```
SELECT columnas FROM tabla [WHERE condiciones]  
EXCEPT [ALL]  
SELECT columnas FROM tabla [WHERE condiciones];
```

Observar:

- La cláusula ALL indica si se quieren obtener todas las filas de la intersección(incluidas las repetidas)

Operaciones entre consultas

- Se puede simular usando:

- **NOT IN**

```
SELECT columnas FROM tabla WHERE columna NOT IN  
(SELECT columna FROM tabla [WHERE condiciones]);
```

- **NOT EXISTS**

```
SELECT columnas FROM tabla WHERE NOT EXISTS  
(SELECT *FROM tabla WHERE condiciones);
```


Operaciones entre consultas

- Por ejemplo si se quiere saber las ciudades de los clientes en las que no hay departamentos:
 - Usando la intersección.

```
SELECT ciudad FROM clientes EXCEPT  
SELECT ciudad_dep FROM departamentos;
```

- Usando IN

```
SELECT c.ciudad FROM clientes c WHERE c.ciudad NOT IN  
(SELECT d.ciudad_dep FROM departamentos d);
```

- Usando EXISTS

```
SELECT c.ciudad FROM clientes c WHERE NOT EXISTS  
(SELECT * FROM departamentos d WHERE c.ciudad =  
d.ciudad_dep;
```