

Más ejemplos en SQL

Francisco Moreno

CP

S

<u>sn</u>	<u>snombre</u>	<u>situacion</u>	<u>ciudad</u>
S1	Salazar	20	Londres
S2	Jaramillo	10	París
S3	Bernal	30	París
S4	Caicedo	20	Londres
S5	Aldana	30	Atenas

CP

P

<u>pn</u>	<u>pnombre</u>	<u>color</u>	<u>peso</u>	<u>ciudad</u>
P1	Tuerca	Rojo	12	Londres
P2	Perno	Verde	17	París
P3	Tornillo	Azul	17	Roma
P4	Tornillo	Rojo	14	Londres
P5	Leva	Azul	12	París
P6	Rueda	Rojo	19	Londres

CF
hacia S

CF
hacia P

SP

<u>sn</u>	<u>pn</u>	<u>cantidad</u>
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P3	200
S4	P2	200
S4	P4	300
S4	P5	400

CP

- Nótese que en estos ejemplos, se cambió el nombre del atributo **s#** por **sn** y del atributo **p#** por **pn**, ya que el carácter **#** está reservado en SQL (aunque en algunos SGBD funciona).
- Obtener todos los datos de los proveedores de París que tengan una situación mayor a 10

```
SELECT *
```

```
FROM s
```

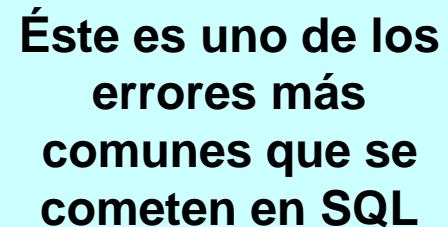
```
WHERE ciudad = 'París' AND situacion > 10;
```

- Obtener solo los códigos de los proveedores de Londres

```
SELECT sn  
FROM s  
WHERE ciudad = 'Londres' ;
```

- Imprimir todos los datos de los productos que no tienen color asignado (color es nulo)

```
SELECT *  
FROM p  
WHERE color IS NULL;
```



Éste es uno de los errores más comunes que se cometen en SQL

Nota: La comparación `color = NULL`, no devuelve verdadero aunque color esté en NULL

También, existe el operador `IS NOT NULL`.

- Imprimir todos los datos de los proveedores que suministran el producto P2. Veamos tres soluciones:

a) `SELECT s.*`
 `FROM sp, s`
 `WHERE sp.sn = s.sn AND pn = 'P2' ;`

b) `SELECT *`
 `FROM s` 
 `WHERE EXISTS (SELECT *`
 `FROM sp`
 `WHERE sp.sn = s.sn AND`
 `sp.pn = 'P2') ;`

Correlación



c) SELECT *

 FROM s

 WHERE sn IN (SELECT sn

 FROM sp

 WHERE pn = ' P2') ;

- Imprimir todos los datos de los proveedores que **NO** suministran el producto P2

```
a) SELECT *
FROM s
WHERE NOT EXISTS (SELECT * FROM SP
                  WHERE sp.sn = s.sn AND
                        sp.pn = 'P2' );
```

```
b) SELECT *
FROM s
WHERE sn NOT IN (SELECT sn FROM SP
                  WHERE pn = 'P2');
```


b) Una solución alternativa usando agregados:

```
SELECT *  
FROM s  
WHERE (SELECT COUNT (*)  
        FROM sp  
        WHERE s.sn = sp.sn) = (SELECT COUNT (*)  
                                FROM p) ;
```

Es decir, imprimir los proveedores si su conteo de productos en la tabla sp es igual al conteo de productos que hay en la tabla p.

- Obtener los nombres de los proveedores que suministran por lo menos un producto rojo

a) Mediante subconsultas:

```
SELECT *  
  FROM s  
 WHERE sn IN (SELECT sn  
              FROM sp  
              WHERE pn IN (SELECT pn  
                          FROM p  
                          WHERE color = 'Rojo'  
                          )  
              )  
);
```

b) Mediante reuniones (*joins*):

```
SELECT DISTINCT s.*  
FROM s, sp, p  
WHERE s.sn = sp.sn AND  
      p.pn = sp.pn AND  
      color = 'Rojo' ;
```

Nótese la cláusula `DISTINCT` para evitar tuplas duplicadas en la salida

- Imprimir los códigos de las parejas de proveedores que están situados en la misma ciudad

```
SELECT s1.sn, s2.sn  
FROM s AS s1, s AS s2  
WHERE s1.ciudad = s2.ciudad AND  
       s1.sn < s2.sn;
```

- Nótese la condición `s1.sn < s2.sn` para evitar que salga cada proveedor consigo mismo y evitar además las parejas “contrarias”.
- Recordar: en el SQL estándar para renombrar se requiere AS, en algunos SGBD no es correcto.

- Obtener el código y el número de productos que suministra cada proveedor

```
SELECT sn, COUNT(*)  
FROM sp  
GROUP BY sn;
```

- La misma consulta anterior pero solo para aquellos proveedores con más de dos productos. Imprimir en orden por el conteo.

```
SELECT sn, COUNT(*) AS conteo  
FROM sp  
GROUP BY sn  
HAVING COUNT(*) > 2  
ORDER BY 2;
```

Significa que ordene por la segunda columna del **SELECT**, aquí, conteo.

Nótese el renombrado para la función **COUNT**.

- Obtener los códigos de los proveedores que están ubicados en Londres o que suministran el producto P2

```
SELECT sn
FROM s
WHERE ciudad = 'Londres'
UNION
SELECT sn
FROM sp
WHERE pn = 'P2' ;
```

Nota: El operador **UNION** **elimina tuplas duplicadas**; si se desean duplicados usar el operador **UNION ALL**.

- Obtener los códigos de los proveedores que son de Londres pero que NO suministran el producto P2

SELECT sn

FROM s

WHERE ciudad ='Londres'

EXCEPT

SELECT sn

FROM sp

WHERE pn = 'P2' ;



En algunos SGBD se usa la palabra **MINUS** en vez de **EXCEPT**

- Obtener los códigos de los proveedores que son de Londres y que suministran también el producto P2

```
SELECT sn
```

```
FROM s
```

```
WHERE ciudad = 'Londres'
```

```
INTERSECT
```

```
SELECT sn
```

```
FROM sp
```

```
WHERE pn = 'P2' ;
```

Manejo de *joins*:

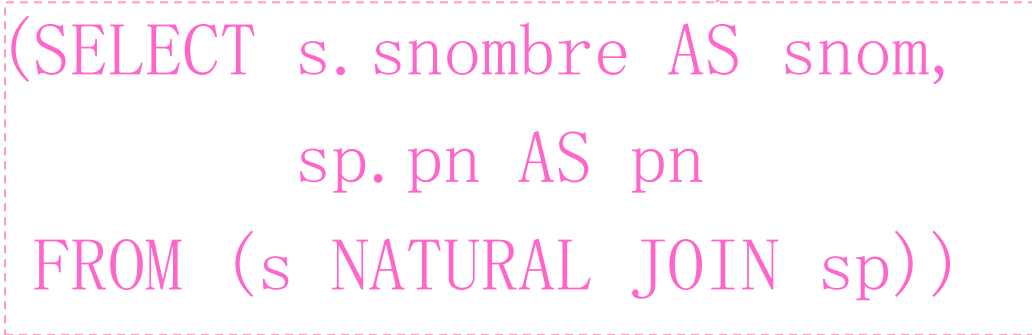
- Imprimir el nombre de cada proveedor y el nombre de cada producto que suministra

a) `SELECT s.snOMBRE, p.pnOMBRE
FROM s, sp, p
WHERE s.sn = sp.sn AND p.pn = sp.pn;`

b) SELECT s. snombre, p. pnombre
FROM (s INNER JOIN sp ON (s. sn = sp. sn))
INNER JOIN p ON (p. pn = sp. pn);

c) SELECT snom, pnombre
FROM (SELECT s. snombre AS snom,
sp. pn AS pn
FROM (s NATURAL JOIN sp))
NATURAL JOIN p;

Inline view



- Nótese que en la última consulta se evita el atributo ciudad de las tablas s y p, para hacer el NATURAL JOIN **solo** por el atributo pn con la relación P.
- Usar *inline views* solo cuando sea necesario:

```
SELECT sn FROM (SELECT * FROM s);
```

- Recuerde, aunque funciona ¡esto es simplemente!:

```
SELECT sn FROM s;
```