

Algunas cuestiones sobre paréntesis, códigos y DISTINCT

1. ¡Cuidado con el uso de paréntesis!

Por ejemplo, ¿hay algún error en la siguiente consulta?

Encontrar todas las parejas de ciudades tales que la primera sea de un proveedor y la segunda sea la de un proyecto entre los cuales haya algún suministro.

$$\pi_{proveedor.ciudad, proyecto.ciudad}(\sigma_{ventas.codpj=proyecto.codpj}(VENTAS \bowtie PROVEEDOR) \times PROYECTO)$$

2. ¡No es lo mismo el código que el nombre!

Por ejemplo, ¿qué problema puede darnos la siguiente consulta?

Pieza con más peso entre las que pesan menos de 100:

$$A = \sigma_{peso < 100}(PIEZA)$$

$$B = A$$

$$\pi_{nompie}(A) - \pi_{B.nompie}(\sigma_{B.peso < A.peso}(B \times A))$$

¿Y esta de SQL?

Encontrar parejas de proveedores que no viven en la misma ciudad:

```
SELECT a.nompro, b.nompro
FROM proveedor a, proveedor b
WHERE a.ciudad < b.ciudad;
```

3. ¡Ojo con el uso del DISTINCT!

¿Qué diferencia hay entre estas dos consultas? ¿Cómo habría que modificar la segunda para que el resultado sea equivalente?

Encuentra los proyectos que están en una ciudad donde se fabrica alguna pieza:

```
SELECT codpj
FROM proyecto
WHERE ciudad IN (SELECT DISTINCT ciudad FROM pieza);

SELECT codpj
FROM proyecto, pieza
WHERE proyecto.ciudad = pieza.ciudad;
```

¿Se usa bien DISTINCT en esta consulta?

Encuentra los códigos de aquellos proyectos que no utilizan ninguna pieza roja que esté suministrada por un proveedor de Londres.

```
SELECT DISTINCT codpj
FROM proyecto
WHERE codpj NOT IN (
    SELECT codpj
    FROM ventas, pieza, proveedor
    WHERE ventas.codpie=pieza.codpie
    AND ventas.codpro=proveedor.codpro
    AND color='Rojo' AND proveedor.ciudad='Londres');
```

¿Se usa DISTINCT de manera coherente aquí?

Encontrar aquellos proveedores que envían piezas procedentes de todas las ciudades donde hay un proyecto.

```
SELECT codpro FROM proveedor WHERE NOT EXISTS (
    SELECT ciudad FROM proyecto
    MINUS
    SELECT DISTINCT ciudad
    FROM ventas NATURAL JOIN pieza
    WHERE ventas.codpro=proveedor.codpro
);
```

¿Qué diferencia hay entre las siguientes consultas?

```
select count(*) from ventas;
select count(codpro) from ventas;
select count(distinct codpro) from ventas;
```

1. La selección no aplica sobre lo que queremos.

$$\pi_{proveedor.ciudad, proyecto.ciudad}(\sigma_{ventas.codpj=proyecto.codpj}((VENTAS \bowtie PROVEEDOR) \times PROYECTO))$$

2. Si hay dos piezas con el mismo nombre, no funcionaría.

$$A = \sigma_{peso < 100}(PIEZA)$$

$$B = A$$

$$\pi_{codpie}(A) - \pi_{B.codpie}(\sigma_{B.peso < A.peso}(B \times A))$$

En la de SQL, si hubiera dos proveedores con el mismo nombre, no los distinguiríamos en el resultado.

```
SELECT a.codpro, a.nompro, b.codpro, b.nompro
FROM proveedor a, proveedor b
WHERE a.ciudad < b.ciudad;
```

3. En la segunda consulta hace falta poner DISTINCT porque los proyectos en este caso pueden salir duplicados.

```
SELECT DISTINCT codpj
FROM proyecto, pieza
WHERE proyecto.ciudad = pieza.ciudad;
```

En la consulta sobre piezas rojas y proveedores de Londres, el DISTINCT no tiene sentido (los códigos en la tabla proyecto no están duplicados). Si se pone en la subconsulta, donde sí pueden aparecer duplicados, tratamos de ayudar al sistema. En ninguno de los dos casos, afecta al resultado.

```
SELECT codpj
FROM proyecto
WHERE codpj NOT IN (
    SELECT DISTINCT codpj
    FROM ventas, pieza, proveedor
    WHERE ventas.codpie=pieza.codpie
    AND ventas.codpro=proveedor.codpro
    AND color='Rojo' AND proveedor.ciudad='Londres');
```

En la consulta sobre proveedores que envían piezas procedentes de todas las ciudades donde hay un proyecto, si se pone DISTINCT en el sustraendo, ¿por qué no se pone en el minuendo? En los dos casos pueden aparecer duplicados. En cualquier caso, no afecta al resultado.

```
SELECT codpro FROM proveedor WHERE NOT EXISTS (
    SELECT DISTINCT ciudad FROM proyecto
    MINUS
    SELECT DISTINCT ciudad
    FROM ventas NATURAL JOIN pieza
    WHERE ventas.codpro=proveedor.codpro
);
```

Finalmente, con respecto al uso del count:

La primera cuenta tuplas. La segunda cuenta tuplas con codpro distinto de null (lo cual no tiene sentido en una tabla como la de ventas). La tercera cuenta valores distintos de codpro.