DAW/DAM. UD 6. MODELO FÍSICO DQL. ACTIVIDADES NO EVALUABLES. BOLETÍN D. EXTRA

DAW/DAM. Bases de datos (BD)

UD 6. MODELO FÍSICO DQL

Extra. Boletín D. Prácticas no evaluables (con soluciones)

Abelardo Martínez y **Pau Miñana**

Curso 2023-2024

Aspectos a tener en cuenta

Importante

Estas actividades son opcionales y no evaluables pero es recomendable hacerlas para un mejor aprendizaje de la asignatura.

> **Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo**. Ten en cuenta que **ChatGPT no es infalible ni todopoderoso**.
>
> Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje. Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el “Aula Virtual”.

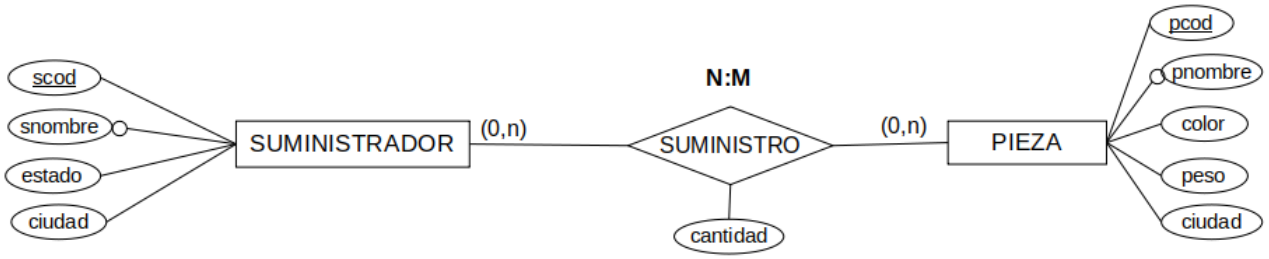
Recomendaciones

Importante

- **No uses NUNCA tildes, ni eñes, ni espacios, ni caracteres no alfanuméricos** (salvo el guión bajo) **en los metadatos** (nombres de elementos de una base de datos).
- Sé coherente con el uso de mayúsculas/minúsculas.

1. BD Suministros. Esquema

Disponemos del siguiente diagrama ER de la base de datos:



2. Creación de la BD

El siguiente paso será crear la base de datos junto con sus tablas correspondientes.

Actividad no evaluable

El SGBD a utilizar es MySQL. El archivo que contiene el *script* con la base de datos lo puedes descargar del Aula Virtual. Lo ejecutaremos y comprobaremos que se ha creado la base de datos y las tablas correctamente.

Base de datos:

```
mysql> CREATE DATABASE IF NOT EXISTS DBSuministros CHARACTER SET utf8 COLLATE
Query OK, 1 row affected, 2 warnings (0,01 sec)

mysql>
mysql> USE DBSuministros;
Database changed
```

Tablas:

```
mysql> show tables;
+-------------------------+
| Tables_in_DBSuministros |
+-------------------------+
| pieza                   |
| suministrador           |
| suministropieza         |
+-------------------------+
3 rows in set (0,00 sec)
```

3. Consultas. Nivel básico

Actividad no evaluable

Realiza las siguientes consultas en **MySQL**.

3.1. Ejercicio

a) Obtener el número total de suministradores.

b) Obtener el número total de suministradores que proveen alguna pieza.

Solución

a) COUNT(*) cuenta el total de registros de la tabla.

```
mysql> SELECT COUNT(*) AS Total_suministradores
    -> FROM suministrador;
+----------------------+
| Total_suministradores |
+----------------------+
|                    5 |
+----------------------+
1 row in set (0,00 sec)
```

b) La cláusula DISTINCT evita los registros duplicados. Si no se pone, se contarían los proveedores por cada pieza que suministran (cada proveedor puede suministrar muchas piezas), pero solo nos interesa contarlo una vez.

```
mysql> SELECT COUNT(DISTINCT scod) AS Suministrador_provee_piezas
    -> FROM suministropieza;
+---------------------------+
| Suministrador_provee_piezas |
+---------------------------+
|                         4 |
```

```
+----------------------------+
1 row in set (0,00 sec)
```

3.2. Ejercicio

Obtener la cantidad total suministrada de la pieza 'p2'.

Solución

```
mysql> SELECT SUM(cantidad) AS 'Total suministro p2'
    -> FROM suministropieza
    -> WHERE pcod = 'p2';
+---------------------+
| Total suministro p2 |
+---------------------+
|                 800 |
+---------------------+
1 row in set (0,00 sec)
```

3.3. Ejercicio

Obtener el código de aquellos suministradores de París con un estado superior a 20.

Solución

```
mysql> SELECT scod AS 'Paris estado > 20'
    -> FROM suministrador
    -> WHERE ciudad = 'París'
    ->     AND estado > 20;
+-------------------+
| Paris estado > 20 |
+-------------------+
| s3                |
+-------------------+
1 row in set (0,01 sec)
```

3.4. Ejercicio

Obtener los pares de código de suministrador y código de pieza tales que tengan la misma ciudad, ordenados por código de suministrador y código de pieza.

Solución

a) Forma implícita:

```
mysql> SELECT S.scod AS Suministrador, P.pcod AS Pieza
    -> FROM suministrador S, pieza P
    -> WHERE S.ciudad = P.ciudad
    -> ORDER BY S.scod, P.pcod;
+---------------+-------+
| Suministrador | Pieza |
+---------------+-------+
| s1            | p1    |
| s1            | p4    |
| s1            | p6    |
| s2            | p2    |
| s2            | p5    |
| s3            | p2    |
| s3            | p5    |
| s4            | p1    |
| s4            | p4    |
| s4            | p6    |
+---------------+-------+
10 rows in set (0,00 sec)
```

b) Forma explícita:

```sql
SELECT S.scod AS Suministrador, P.pcod AS Pieza
FROM suministrador S
INNER JOIN pieza P
ON S.ciudad = P.ciudad
ORDER BY S.scod, P.pcod;
```

3.5. Ejercicio

Obtener los códigos de las piezas que pesan entre 16 y 19 kg.

Solución

```
mysql> SELECT pcod AS 'Peso entre 16 y 19'
    -> FROM pieza
    -> WHERE peso BETWEEN 16 AND 19;
+--------------------+
| Peso entre 16 y 19 |
+--------------------+
| p2                 |
| p3                 |
| p6                 |
+--------------------+
3 rows in set (0,00 sec)
```

3.6. Ejercicio

Obtener el nombre de los suministradores que proveen entre 100 y 200 unidades de alguna pieza, ordenado alfabéticamente por nombre.

Solución

a) Forma implícita:

```
mysql> SELECT DISTINCT(S.snombre) AS Suministrador
    -> FROM suministrador S, suministropieza SP
    -> WHERE S.scod = SP.scod
    ->     AND SP.cantidad BETWEEN 100 AND 200
    -> ORDER BY S.snombre;
+---------------+
| Suministrador |
+---------------+
| Blake         |
| Clark         |
| Smith         |
+---------------+
3 rows in set (0,00 sec)
```

Si no usamos DISTINCT, saldrían duplicados y no sería correcto:

```
mysql> SELECT S.snombre AS Suministrador
    -> FROM suministrador S, suministropieza SP
    -> WHERE S.scod = SP.scod
    ->     AND SP.cantidad BETWEEN 100 AND 200
```

```
    -> ORDER BY S.snombre;
+---------------+
| Suministrador |
+---------------+
| Blake         |
| Clark         |
| Smith         |
| Smith         |
| Smith         |
| Smith         |
+---------------+
6 rows in set (0,00 sec)
```

b) Forma explícita:

```sql
SELECT DISTINCT(S.snombre) AS Suministrador
FROM suministrador S
INNER JOIN suministropieza SP
ON S.scod = SP.scod
WHERE SP.cantidad BETWEEN 100 AND 200
ORDER BY S.snombre;
```

3.7. Ejercicio

Obtener el nombre y código de piezas cuyo nombre empieza por 'T', ordenado por nombre descendente.

Solución

```
mysql> SELECT pnombre AS 'Nombre pieza', pcod AS 'Código'
    -> FROM pieza
    -> WHERE pnombre LIKE 'T%'
    -> ORDER BY pnombre DESC;
+--------------+---------+
| Nombre pieza | Código  |
+--------------+---------+
| Tuerca       | p1      |
| Tornillo     | p3      |
| Tornillo     | p4      |
+--------------+---------+
3 rows in set (0,00 sec)
```

3.8. Ejercicio

Obtener el nombre de piezas que contengan algún guión bajo (_), ordenado por nombre.

Solución

Se utiliza el símbolo de barra invertida (\) como carácter de escape para poder poner el guión bajo.

```
mysql> SELECT pnombre AS 'Nombre pieza'
    -> FROM pieza
    -> WHERE pnombre LIKE '%\_%'
    -> ORDER BY pnombre;
Empty set (0,00 sec)
```

3.9. Ejercicio

Obtener el código y nombre de los suministradores que no tienen estado.

Solución

```
mysql> SELECT scod AS 'Código', snombre AS 'Nombre sumin.'
    -> FROM suministrador
    -> WHERE estado IS NULL;
Empty set (0,00 sec)
```

Como podemos apreciar, la consulta no devuelve ningún resultado (0 registros); sin embargo, eso no significa que esté mal. En este caso, no hay ningún dato que cumpla las condiciones de la consulta; es decir, todos los suministradores tienen un estado.

3.10. Ejercicio

Obtener el nombre de las piezas cuyo peso sea 12, 16 ó 18 kg, ordenado por nombre.

Solución

```
mysql> SELECT pnombre AS 'Nombre pieza'
    -> FROM pieza
    -> WHERE peso IN (12, 16, 18)
    -> ORDER BY pnombre;
+--------------+
| Nombre pieza |
+--------------+
| Leva         |
| Tuerca       |
+--------------+
2 rows in set (0,00 sec)
```

También podemos utilizar una expresión equivalente para la condición de los pesos:

WHERE peso = 12 OR peso = 16 OR peso = 18

3.11. Ejercicio

Obtener el nombre de suministradores que proveen la pieza 'p2', ordenado por nombre.

Solución

a) Forma implícita:

```
mysql> SELECT DISTINCT(S.snombre) AS 'Sumin. de pieza p2'
    -> FROM suministrador S, suministropieza SP
    -> WHERE S.scod = SP.scod
    ->     AND SP.pcod = 'p2'
    -> ORDER BY S.snombre;
+--------------------+
| Sumin. de pieza p2 |
+--------------------+
| Blake              |
| Jones              |
| Smith              |
+--------------------+
3 rows in set (0,00 sec)
```

b) Forma explícita:

```sql
SELECT DISTINCT(S.snombre) AS 'Sumin. de pieza p2'
FROM suministrador S
INNER JOIN suministropieza SP
ON S.scod = SP.scod
```

```
WHERE SP.pcod = 'p2'
ORDER BY S.snombre;
```

3.12. Ejercicio

Obtener el nombre de suministradores que proveen alguna pieza roja, ordenado por nombre.

Solución

a) Forma implícita:

```
mysql> SELECT DISTINCT(S.snombre) AS 'Sumin. de piezas rojas'
    -> FROM suministrador S, suministropieza SP, pieza P
    -> WHERE S.scod = SP.scod
    ->     AND P.pcod = SP.pcod
    ->     AND P.color = 'Rojo'
    -> ORDER BY S.snombre;
+------------------------+
| Sumin. de piezas rojas |
+------------------------+
| Clark                  |
| Jones                  |
| Smith                  |
+------------------------+
3 rows in set (0,00 sec)
```

Si no usamos DISTINCT, los suministradores saldrían duplicados y no sería correcto:

```
mysql> SELECT S.snombre AS 'Sumin. de piezas rojas'
    -> FROM suministrador S, suministropieza SP, pieza P
    -> WHERE S.scod = SP.scod
```

```
    ->     AND P.pcod = SP.pcod
    ->     AND P.color = 'Rojo'
    -> ORDER BY S.snombre;
+------------------------+
| Sumin. de piezas rojas |
+------------------------+
| Clark                  |
| Clark                  |
| Jones                  |
| Smith                  |
| Smith                  |
| Smith                  |
+------------------------+
6 rows in set (0,00 sec)
```

b) Forma explícita:

```sql
SELECT DISTINCT(S.snombre) AS 'Sumin. de piezas rojas'
FROM suministrador S
INNER JOIN suministropieza SP
ON S.scod = SP.scod
INNER JOIN pieza P
ON P.pcod = SP.pcod
WHERE P.color = 'Rojo'
ORDER BY S.snombre;
```

3.13. Ejercicio

Obtener los pares de ciudades tales que un suministrador de la primera suministre una pieza de la segunda, ordenados por ciudad del suministrador y por ciudad de la pieza.

Solución

a) Forma implícita:

```
mysql> SELECT DISTINCT S.ciudad AS 'Sumin. ciudad', P.ciudad AS 'Pieza ciudad'
    -> FROM suministrador S, pieza P, suministropieza SP
    -> WHERE S.scod = SP.scod
    ->     AND P.pcod = SP.pcod
    -> ORDER BY S.ciudad, P.ciudad;
+---------------+--------------+
| Sumin. ciudad | Pieza ciudad |
+---------------+--------------+
| Londres       | Londres      |
| Londres       | París        |
| Londres       | Roma         |
| París         | Londres      |
| París         | París        |
+---------------+--------------+
5 rows in set (0,00 sec)
```

b) Forma explícita:

```
SELECT DISTINCT S.ciudad AS 'Sumin. ciudad', P.ciudad AS 'Pieza ciudad'
FROM suministrador S
```

```
INNER JOIN suministropieza SP
ON S.scod = SP.scod
INNER JOIN pieza P
ON P.pcod = SP.pcod
ORDER BY S.ciudad, P.ciudad;
```

3.14. Ejercicio

Obtener los códigos de suministradores tales que ambos tienen la misma ciudad.

Solución

a) Forma implícita:

```
mysql> SELECT S1.scod AS 'Sumin 1', S2.scod AS 'Sumin 2'
    -> FROM suministrador S1, suministrador S2
    -> WHERE S1.ciudad = S2.ciudad
    ->     AND S1.scod > S2.scod;
+---------+---------+
| Sumin 1 | Sumin 2 |
+---------+---------+
| s3      | s2      |
| s4      | s1      |
+---------+---------+
2 rows in set (0,00 sec)
```

Si se pone <> (en lugar de >) saldrían los simétricos (S1,S2) y (S2, S1).

b) Forma explícita:

```sql
SELECT S1.scod AS 'Sumin 1', S2.scod AS 'Sumin 2'
FROM suministrador S1
INNER JOIN suministrador S2
ON S1.ciudad = S2.ciudad
WHERE S1.scod > S2.scod; -- si se pone <> saldrían los simétricos (S1,S2) y (S
```

3.15. Ejercicio

Obtener el código de los suministradores que proporcionan al menos una pieza que suministra 's2', ordenado por código.

Solución

a) Forma implícita:

```
mysql> SELECT DISTINCT SP1.scod AS Suministrador
    -> FROM suministropieza SP1, suministropieza SP2
    -> WHERE SP1.pcod = SP2.pcod
    ->     AND SP1.scod <> 's2'
    ->     AND SP2.scod = 's2'
    -> ORDER BY SP1.scod;
+---------------+
| Suministrador |
+---------------+
| s1            |
| s3            |
| s4            |
+---------------+
3 rows in set (0,00 sec)
```

Si no ponemos la condición ***SP1.scod <> 's2'*** saldría también el propio suministrador 's2':

```
mysql> SELECT DISTINCT SP1.scod AS Suministrador
    -> FROM suministropieza SP1, suministropieza SP2
    -> WHERE SP1.pcod = SP2.pcod
```

```
    ->     AND SP2.scod = 's2'
    -> ORDER BY SP1.scod;
+---------------+
| Suministrador |
+---------------+
| s1            |
| s2            |
| s3            |
| s4            |
+---------------+
4 rows in set (0,00 sec)
```

b) Forma explícita:

```sql
SELECT DISTINCT SP1.scod AS Suministrador
FROM suministropieza SP1
INNER JOIN suministropieza SP2
ON SP1.pcod = SP2.pcod
WHERE SP1.scod <> 's2'
    AND SP2.scod = 's2'
ORDER BY SP1.scod;
```

4. Consultas. Nivel medio

Actividad no evaluable

Realiza las siguientes consultas en **MySQL**.

4.1. Ejercicio

Para cada pieza suministrada, obtener código y cantidad máxima y mínima suministrada de esa pieza, excluyendo los suministros de 's1', ordenado por código de pieza.

Solución

```
mysql> SELECT pcod AS Pieza, MAX(cantidad), MIN(cantidad)
    -> FROM suministropieza
    -> WHERE scod <> 's1'
    -> GROUP BY pcod
    -> ORDER BY pcod;
+-------+---------------+---------------+
| Pieza | MAX(cantidad) | MIN(cantidad) |
+-------+---------------+---------------+
| p1    |           300 |           200 |
| p2    |           400 |           200 |
| p4    |           300 |           300 |
| p5    |           400 |           400 |
+-------+---------------+---------------+
4 rows in set (0,01 sec)
```

4.2. Ejercicio

Obtener el código de las piezas suministradas por más de un proveedor, ordenado por código de pieza.

Solución

```
mysql> SELECT pcod AS 'Código pieza'
    -> FROM suministropieza
    -> GROUP BY pcod
    -> HAVING COUNT(*) > 1
    -> ORDER BY pcod;
+---------------+
| Código pieza  |
+---------------+
| p1            |
| p2            |
| p4            |
| p5            |
+---------------+
4 rows in set (0,00 sec)
```

4.3. Ejercicio

Para las piezas azules y rojas que se suministren en cantidad total superior a 350 (sin tener en cuenta suministros inferiores a 200), obtener el código, máxima cantidad suministrada, color y peso en gramos. Ordenar por cantidad (descendente) y código de pieza.

Solución

a) Forma implícita:

```
mysql> SELECT P.pcod AS 'Código pieza', MAX(cantidad) AS 'Máxima cantidad', P.
    -> FROM pieza P, suministropieza SP
    -> WHERE P.pcod = SP.pcod
    ->     AND SP.cantidad >= 200
    ->     AND (P.color = 'Rojo' OR P.color = 'Azul')
    -> GROUP BY P.pcod, P.peso, P.color
    -> HAVING SUM(cantidad) > 350
    -> ORDER BY 2 DESC, 1;
+---------------+------------------+----------------+-------+
| Código pieza  | Máxima cantidad  | Peso en gramos | Color |
+---------------+------------------+----------------+-------+
| p3            |              400 |          17000 | Azul  |
| p5            |              400 |          12000 | Azul  |
| p1            |              300 |          12000 | Rojo  |
| p4            |              300 |          14000 | Rojo  |
+---------------+------------------+----------------+-------+
4 rows in set (0,00 sec)
```

Consideraciones:

- Es equivalente poner SELECT SP.pcod (en lugar de P.pcod).
- Al agrupar por el código de pieza (clave primaria), solo hay un peso y un color (el de la pieza).

b) Forma explícita:

```
SELECT P.pcod AS 'Código pieza', MAX(cantidad) AS 'Máxima cantidad', P.peso*10
FROM pieza P
INNER JOIN suministropieza SP
ON P.pcod = SP.pcod
WHERE SP.cantidad >= 200
    AND (P.color = 'Rojo' OR P.color = 'Azul')
GROUP BY P.pcod, P.peso, P.color
HAVING SUM(cantidad) > 350
ORDER BY 2 DESC, 1;
```

5. Consultas. Nivel avanzado

Actividad no evaluable

Realiza las siguientes consultas en **MySQL**.

5.1. Ejercicio

Obtener el código de aquellos suministradores que tengan un estado menor que el máximo estado de un suministrador.

Solución

```
mysql> SELECT S1.scod AS 'Suministrador'
    -> FROM suministrador S1
    -> WHERE S1.estado < (SELECT MAX(S2.estado)
    ->                    FROM suministrador S2);
+---------------+
| Suministrador |
+---------------+
| s1            |
| s2            |
| s4            |
+---------------+
3 rows in set (0,01 sec)
```

5.2. Ejercicio

Obtener el código de aquellos suministradores que vivan en la misma ciudad que el suministrador 's1'.

Solución

```
mysql> SELECT S1.scod AS 'Suministrador'
    -> FROM suministrador S1
    -> WHERE S1.scod <> 's1'
    ->     AND S1.ciudad = (SELECT S2.ciudad
    ->                      FROM suministrador S2
    ->                      WHERE S2.scod = 's1');
+---------------+
| Suministrador |
+---------------+
| s4            |
+---------------+
1 row in set (0,00 sec)
```

Si no ponemos la condición ***S1.scod <> 's1'*** , saldría también el propio suministrador 's1':

```
mysql> SELECT S1.scod AS 'Suministrador'
    -> FROM suministrador S1
    -> WHERE S1.ciudad = (SELECT S2.ciudad
    ->                      FROM suministrador S2
    ->                      WHERE S2.scod = 's1');
+---------------+
```

```
| Suministrador |
+---------------+
| s1            |
| s4            |
+---------------+
2 rows in set (0,00 sec)
```

5.3. Ejercicio

Obtener el nombre de suministradores que proveen la pieza 'p2', ordenado por nombre.

Solución

a) OPCIÓN 1

Usando el código de suministrador.

```
mysql> SELECT S.snombre AS 'Sumin. de pieza p2'
    -> FROM suministrador S
    -> WHERE S.scod IN
    ->     (SELECT SP.scod
    ->     FROM suministropieza SP
    ->     WHERE SP.pcod = 'p2')
    -> ORDER BY S.snombre;
+--------------------+
| Sumin. de pieza p2 |
+--------------------+
| Blake              |
| Jones              |
| Smith              |
+--------------------+
3 rows in set (0,00 sec)
```

b) OPCIÓN 2

Usando el código de pieza.

```sql
SELECT S.snombre AS 'Sumin. de pieza p2'
FROM suministrador S
WHERE 'p2' IN
    (SELECT SP.pcod
    FROM suministropieza SP
    WHERE SP.scod = S.scod)
ORDER BY S.snombre;
```

5.4. Ejercicio

Obtener el nombre de suministradores que proveen alguna pieza roja, ordenado por nombre.

Solución

a) OPCIÓN 1

Subconsulta con predicado IN.

```
mysql> SELECT S.snombre AS 'Sumin. de piezas rojas'
    -> FROM suministrador S
    -> WHERE S.scod IN
    ->     (SELECT SP.scod
    ->     FROM suministropieza SP
    ->     WHERE SP.pcod IN
    ->         (SELECT P.pcod
    ->             FROM pieza P
    ->             WHERE P.color = 'Rojo'))
    -> ORDER BY S.snombre;
+------------------------+
| Sumin. de piezas rojas |
+------------------------+
| Clark                  |
| Jones                  |
| Smith                  |
+------------------------+
3 rows in set (0,00 sec)
```

b) OPCIÓN 2

Con **GROUP BY**.

```
SELECT S.snombre AS 'Sumin. de piezas rojas'
FROM suministrador S, suministropieza SP, pieza P
WHERE S.scod = SP.scod
    AND SP.pcod = P.pcod
    AND P.color = 'Rojo'
GROUP BY S.scod, S.snombre
ORDER BY S.snombre;
```

c) OPCIÓN 3

Con **DISTINCT**.

```
SELECT DISTINCT S.snombre AS 'Sumin. de piezas rojas'
FROM suministrador S, suministropieza SP, pieza P
WHERE S.scod = SP.scod
    AND SP.pcod = P.pcod
    AND P.color = 'Rojo'
ORDER BY S.snombre;
```

5.5. Ejercicio

Obtener el código de los suministradores que proporcionan al menos una pieza que suministra 's2', ordenado por código.

Solución

a) OPCIÓN 1

Subconsulta con **IN**.

```
mysql> SELECT DISTINCT SP1.scod AS Suministrador
    -> FROM suministropieza SP1
    -> WHERE SP1.scod <> 's2'
    ->     AND SP1.pcod IN
    ->         (SELECT SP2.pcod
    ->             FROM suministropieza SP2
    ->             WHERE SP2.scod = 's2')
    -> ORDER BY SP1.scod;
+---------------+
| Suministrador |
+---------------+
| s1            |
| s3            |
| s4            |
+---------------+
3 rows in set (0,00 sec)
```

Si no ponemos la condición ***SP1.scod <> 's2'*** saldría también el propio suministrador 's2'.

b) OPCIÓN 2

Subconsulta con **EXISTS**.

```
SELECT DISTINCT SP1.scod AS Suministrador
FROM suministropieza SP1
WHERE SP1.scod <> 's2'
    AND EXISTS
        (SELECT SP2.pcod
            FROM suministropieza SP2
            WHERE SP2.scod = 's2')
ORDER BY SP1.scod;
```

5.6. Ejercicio

Obtener el nombre de los suministradores que proveen al menos una pieza.

Solución

a) OPCIÓN 1

Con predicado **EXISTS**.

```
mysql> SELECT S.snombre AS 'Sumin. de piezas'
    -> FROM suministrador S
    -> WHERE EXISTS (SELECT *
    ->                 FROM suministropieza SP
    ->                 WHERE SP.scod = S.scod)
    -> ORDER BY S.snombre;
+------------------+
| Sumin. de piezas |
+------------------+
| Blake            |
| Clark            |
| Jones            |
| Smith            |
+------------------+
4 rows in set (0,00 sec)
```

b) OPCIÓN 2

Con predicado **IN**.

```
SELECT S.snombre AS 'Sumin. de piezas'
FROM suministrador S
WHERE S.scod IN (SELECT SP.scod
                 FROM suministropieza SP)
ORDER BY S.snombre;
```

c) OPCIÓN 3

Con predicado **ANY**.

```
SELECT S.snombre AS 'Sumin. de piezas'
FROM suministrador S
WHERE S.scod = ANY (SELECT SP.scod
                    FROM suministropieza SP)
ORDER BY S.snombre;
```

5.7. Ejercicio

Obtener el código de los suministradores que suministran alguna pieza en cantidad mayor que todos los suministros de 's1', ordenado por código.

Solución

a) OPCIÓN 1

Con predicado **EXISTS**.

```
mysql> SELECT DISTINCT SP1.scod AS 'Sumin. cant > s1'
    -> FROM suministropieza SP1
    -> WHERE NOT EXISTS (SELECT *
    ->                     FROM suministropieza SP2
    ->                     WHERE SP2.scod = 's1'
    ->                         AND SP1.cantidad <= SP2.cantidad)
    -> ORDER BY SP1.scod;
Empty set (0,00 sec)
```

b) OPCIÓN 2

Con predicado **ALL**.

```
SELECT DISTINCT SP1.scod AS 'Sumin. cant > s1'
FROM suministropieza SP1
WHERE SP1.cantidad > ALL (SELECT SP2.cantidad
                          FROM suministropieza SP2
                          WHERE SP2.scod = 's1')
ORDER BY SP1.scod;
```

5.8. Ejercicio

Obtener el código de suministradores que suministran más piezas que el suministrador de código 's1', ordenado por código.

Solución

```
mysql> SELECT DISTINCT SP1.scod AS 'Sumin. piezas > s1'
    -> FROM suministropieza SP1
    -> GROUP BY SP1.scod
    -> HAVING COUNT(*) > (SELECT COUNT(*)
    ->                    FROM suministropieza SP2
    ->                    WHERE SP2.scod = 's1')
    -> ORDER BY SP1.scod;
Empty set (0,00 sec)
```

6. Bibliografía

- MySQL 8.0 Reference Manual. https://dev.mysql.com/doc/refman/8.0/en/
- Oracle Database Documentation. https://docs.oracle.com/en/database/oracle/oracle-database/index.html
- MySQL Tutorial. https://www.w3schools.com/mysql/
- GURU99. Tutorial de MySQL para principiantes Aprende en 7 días. https://guru99.es/sql/
- SQL Tutorial - Learn SQL. https://www.sqltutorial.net/