UNIT 5
PHYSICAL MODELING
DATA QUERY LANGUAGE (DQL)

BASES DE DATOS 2022/2023
CFGS DAW

FULL DQL WORKSHOP (SELECTS)
SOLUTIONS WORKSHOP A (BASIC LEVEL)

Reviewed by:
Sergio Badal
Author:
Paco Aldarias
Date: 03/07/23

SOLUTION QUERY 11

Mostrar las ciudades en las que la empresa de jardinería tiene clientes. Fíjate en el resultado obtenido y si detectas errores de algún tipo, corrígelos.

```
SELECT DISTINCT ciudad FROM clientes;
```

```
mysql> use jardineria;
Database changed
mysql> SELECT DISTINCT ciudad FROM clientes;
+--------------------------+
| ciudad                   |
+--------------------------+
| San Francisco            |
| Miami                    |
| New York                 |
| Fuenlabrada              |
| Madrid                   |
| San Lorenzo del Escorial |
| Montornes del valles     |
| Santa cruz de Tenerife   |
| Barcelona                |
| Canarias                 |
|  Barcelona               |
| Sotogrande               |
| Humanes                  |
| Getafe                   |
| Fenlabrada               |
| Paris                    |
| Sydney                   |
| London                   |
+--------------------------+
18 rows in set (0.00 sec)
```

Look at the result, as you can see, despite having used the distinct clause, two Barcelona appear, but if you look closely, you will see that the second one is a little separated from the column. This is a common error, when you have typed the data you have included a blank space in front of the name, so the database is taking it as two different cities. Normally these things should be controlled by the programme that allows the data to be entered, but obviously these are not perfect and sometimes errors like this occur. If anomalies of this type are observed, they must be corrected. To change this record and place it properly we have several options, I am going to do one of them, but I hope you can think of at least a couple of others.

```
mysql> UPDATE clientes SET ciudad = 'Barcelona' WHERE ciudad =  ' Barcelona';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Notice that in the filter, after the first quotation mark, I have left a blank space to reproduce the erroneous data in the field. If you now run the previous query again, you will see that only one city, Barcelona, appears.

THIS MUST BE REPEATED FOR EVERY MISSPELLING (Fuenlabrada instead of ~~Fenlabrada~~, París instead of ~~Paris~~...)

SOLUTION QUERY 12

Mostrar cuántos clientes hay en una columna denominada Número de clientes.

```
SELECT COUNT(*) AS 'Número de clientes'
FROM clientes;
```

```
mysql> SELECT COUNT(*) AS 'Número de clientes'
    -> FROM clientes;
+--------------------+
| Número de clientes |
+--------------------+
|                 36 |
+--------------------+
1 row in set (0.00 sec)
```

SOLUTION QUERY 13

Mostrar el nombre, la cantidad en almacén y el precio de compra (proveedor) de los productos de la gama Herramientas ordenado alfabéticamente por el nombre del producto. Las columnas aparecerán como Prod, Stock y PVP.

```
SELECT nombre AS Prod, cantidadenstock AS Stock, precioproveedor AS PVP
FROM productos
WHERE gama='Herramientas'
ORDER BY nombre; -- BEST PRAXIS
-- ORDER BY 1; -- BEST PRAXIS
-- ORDER BY Prod; -- BAD PRAXIS
```

```
mysql> SELECT nombre AS Prod, cantidadenstock AS Stock, precioproveedor AS PVP
    -> FROM productos
    -> WHERE gama='Herramientas'
    -> ORDER BY nombre; -- BEST PRAXIS
+----------------------+-------+-------+
| Prod                 | Stock | PVP   |
+----------------------+-------+-------+
| Azadón               |    15 | 11.00 |
| Pala                 |    15 | 13.00 |
| Rastrillo de Jardín  |    15 | 11.00 |
| Sierra de Poda 400MM |    15 | 11.00 |
+----------------------+-------+-------+
4 rows in set (0.00 sec)
```

SOLUTION QUERY 14

Mostrar la valoración del almacén de cada producto de la gama Herramientas junto con el nombre del producto, ordenando la salida por valoración y nombre del producto. La valoración del almacén multiplica los productos en stock por el precio de compra y la salida deseada debe tener como encabezados "Valor" y "Producto". Cuando no indican asc/desc, se sobreentiende que es alfabético para texto y de menor a mayor para números.

```
SELECT (cantidadenstock*precioproveedor) AS Valor, nombre AS Producto
FROM productos
WHERE gama='Herramientas'
ORDER BY 1,2; -- BEST PRAXIS
-- BEST PRAXIS: ORDER BY (cantidadenstock*precioproveedor), nombre;
-- BAD PRAXIS: ORDER BY 1, nombre;
-- BAD PRAXIS: ORDER BY 1, Producto;
-- BAD PRAXIS: ORDER BY (cantidadenstock*precioproveedor), Producto;
-- BAD PRAXIS:  ORDER BY Valor, Producto;
```

```
mysql> SELECT (cantidadenstock*precioproveedor) AS Valor, nombre AS Producto
    -> FROM productos
    -> WHERE gama='Herramientas'
    -> ORDER BY 1,2; -- BEST PRAXIS
+--------+----------------------+
| Valor  | Producto             |
+--------+----------------------+
| 165.00 | Azadón               |
| 165.00 | Rastrillo de Jardín  |
| 165.00 | Sierra de Poda 400MM |
| 195.00 | Pala                 |
+--------+----------------------+
4 rows in set (0.00 sec)
```

SOLUTION QUERY 15

Beneficio en la venta de cada producto de Herramientas (en columna Beneficio). Primero los productos más rentables. (beneficio = precio venta - precio proveedor)

```
SELECT nombre AS Producto, (precioventa-precioproveedor) as Beneficio
FROM productos
WHERE gama='Herramientas'
ORDER BY 2 DESC, 1; -- BEST PRAXIS
```

```
mysql> SELECT nombre AS Producto, (precioventa-precioproveedor) as Beneficio
    -> FROM productos
    -> WHERE gama='Herramientas'
    -> ORDER BY 2 DESC, 1; -- BEST PRAXIS
+----------------------+-----------+
| Producto             | Beneficio |
+----------------------+-----------+
| Sierra de Poda 400MM |      3.00 |
| Azadón               |      1.00 |
| Pala                 |      1.00 |
| Rastrillo de Jardín  |      1.00 |
+----------------------+-----------+
4 rows in set (0.00 sec)
```

SOLUTION QUERY 16

Mostrar el beneficio máximo de los productos en stock (si no tiene stock no cuenta). La columna debe llamarse "Beneficio máximo". Se considera beneficio a precioventa – precioproveedor.

```
SELECT max(precioventa - precioproveedor) as 'Beneficio máximo'
FROM productos
WHERE cantidadenstock>0;
```

```
mysql> SELECT max(precioventa - precioproveedor) as 'Beneficio máximo'
    -> FROM productos
    -> WHERE cantidadenstock>0;
+------------------+
| Beneficio máximo |
+------------------+
|            93.00 |
+------------------+
1 row in set (0.00 sec)
```

SOLUCIÓN CONSULTA 17

Mostrar el Código del pedido (idPedido), la fecha del pedido (Pedido), el código del cliente (idCliente), la fecha de entrega (Entrega) y la fecha esperada (Estimación) del pedido para todos aquellos con fecha de entrega posterior a la fecha esperada. Indica también los días de retraso (restando las fechas) y ordena por días de retraso y fecha de pedido, visualizando primero los pedidos con mayor retraso. (pista: son 32 filas)

```
SELECT codigopedido AS idPedido, fechapedido AS Pedido, codigocliente AS idCliente, fechaentrega AS Entrega, fechaesperada AS Estimación, fechaentrega-fechaesperada AS Retraso
FROM pedidos
WHERE fechaesperada<fechaentrega
ORDER BY 6 DESC, 2;
```

```
mysql> SELECT codigopedido AS idPedido, fechapedido AS Pedido, codigocliente AS idCliente,
fechaentrega AS Entrega, fechaesperada AS Estimación, fechaentrega-fechaesperada AS Retraso

    -> FROM pedidos
    -> WHERE fechaesperada<fechaentrega
    -> ORDER BY 6 DESC, 2;
+----------+------------+-----------+------------+------------+---------+
| idPedido | Pedido     | idCliente | Entrega    | Estimación | Retraso |
+----------+------------+-----------+------------+------------+---------+
|      113 | 2008-10-28 |        36 | 2009-01-09 | 2008-11-09 |    9000 |
|      115 | 2008-11-29 |        36 | 2009-02-27 | 2009-01-26 |     101 |
|      112 | 2009-03-05 |        36 | 2009-05-07 | 2009-04-06 |     101 |
|       31 | 2008-09-04 |        13 | 2008-10-04 | 2008-09-30 |      74 |
|       92 | 2009-04-19 |        27 | 2009-05-03 | 2009-04-30 |      73 |
|      128 | 2008-11-10 |        38 | 2008-12-29 | 2008-12-10 |      19 |
|       32 | 2007-01-07 |         4 | 2007-01-27 | 2007-01-19 |       8 |
|       16 | 2009-01-06 |         7 | 2009-01-15 | 2009-01-07 |       8 |
|      106 | 2009-05-13 |        30 | 2009-05-20 | 2009-05-15 |       5 |
|      126 | 2009-05-13 |        30 | 2009-05-20 | 2009-05-15 |       5 |
|      103 | 2009-01-15 |        30 | 2009-01-24 | 2009-01-20 |       4 |
|      123 | 2009-01-15 |        30 | 2009-01-24 | 2009-01-20 |       4 |
|       42 | 2009-03-22 |        19 | 2009-03-27 | 2009-03-23 |       4 |
|       28 | 2009-02-10 |         3 | 2009-02-20 | 2009-02-17 |       3 |
|       68 | 2009-02-10 |         3 | 2009-02-20 | 2009-02-17 |       3 |
|       40 | 2009-03-09 |        19 | 2009-03-13 | 2009-03-10 |       3 |
|       44 | 2009-03-26 |        23 | 2009-03-30 | 2009-03-27 |       3 |
|       45 | 2009-04-01 |        23 | 2009-03-07 | 2009-03-04 |       3 |
|       17 | 2009-01-08 |         7 | 2009-01-11 | 2009-01-09 |       2 |
|       22 | 2009-01-11 |         9 | 2009-01-13 | 2009-01-11 |       2 |
|      114 | 2009-01-15 |        36 | 2009-01-31 | 2009-01-29 |       2 |
|       39 | 2009-03-06 |        19 | 2009-03-09 | 2009-03-07 |       2 |
|       43 | 2009-03-25 |        23 | 2009-03-28 | 2009-03-26 |       2 |
|       96 | 2008-03-20 |        35 | 2008-04-13 | 2008-04-12 |       1 |
|       49 | 2008-07-12 |        26 | 2008-07-23 | 2008-07-22 |       1 |
|       55 | 2008-12-10 |        14 | 2009-01-11 | 2009-01-10 |       1 |
|        9 | 2008-12-22 |         1 | 2008-12-28 | 2008-12-27 |       1 |
|       60 | 2008-12-22 |         1 | 2008-12-28 | 2008-12-27 |       1 |
|       18 | 2009-01-05 |         9 | 2009-01-07 | 2009-01-06 |       1 |
|       13 | 2009-01-12 |         7 | 2009-01-15 | 2009-01-14 |       1 |
|       38 | 2009-03-05 |        19 | 2009-03-07 | 2009-03-06 |       1 |
|       46 | 2009-04-03 |        23 | 2009-03-05 | 2009-03-04 |       1 |
+----------+------------+-----------+------------+------------+---------+
32 rows in set (0.00 sec)
```

SOLUTION QUERY 18

Obtener cuántos pedidos nos han realizado todos los clientes con códigos entre 30 y 40, excluyendo el cliente con código 30.

```
SELECT count(*) AS 'Número de pedidos'
FROM pedidos
WHERE codigocliente BETWEEN 31 AND 40;
-- WHERE codigocliente BETWEEN 30 AND 40 AND codigocliente<>30;
-- WHERE codigocliente > 30 AND 40 codigocliente<=40;
```

```
mysql> SELECT count(*) AS 'Número de pedidos'
    -> FROM pedidos
    -> WHERE codigocliente BETWEEN 31 AND 40;
+--------------------+
| Número de pedidos |
+--------------------+
|                 15 |
+--------------------+
1 row in set (0.00 sec)
```

SOLUTION QUERY 19.1

Se ha detectado que hay errores en nuestros datos. Se han encontrado pedidos con fecha de entrega nula y estado Entregado. Se desea encontrar esas inconsistencias y mostrarlas ordenadas por la fecha de pedido. También nos piden incluir los pedidos con código 3, 5 o 10.

```
SELECT codigocliente, fechapedido, estado
FROM pedidos
WHERE (fechaentrega IS NULL AND Estado = 'Entregado')
OR codigocliente IN (3, 5, 10)
-- OR codigocliente=3 OR codigocliente=5 OR codigocliente=10
ORDER BY fechapedido;
```

```
mysql> SELECT codigocliente, fechapedido, estado
    -> FROM pedidos
    -> WHERE (fechaentrega IS NULL AND Estado = 'Entregado')
    -> OR codigocliente IN (3, 5, 10)
    -> -- OR codigocliente=3 OR codigocliente=5 OR codigocliente=10
    -> ORDER BY fechapedido;
+---------------+-------------+-----------+
| codigocliente | fechapedido | estado    |
+---------------+-------------+-----------+
|             5 | 2006-01-17  | Entregado |
|             5 | 2007-10-23  | Entregado |
|             5 | 2008-06-20  | Rechazado |
|             5 | 2008-12-30  | Rechazado |
|             3 | 2009-01-15  | Pendiente |
|             3 | 2009-01-15  | Pendiente |
|             5 | 2009-01-20  | Pendiente |
|             3 | 2009-01-24  | Entregado |
|            28 | 2009-01-24  | Entregado |
|             3 | 2009-02-06  | Rechazado |
|             3 | 2009-02-06  | Rechazado |
|             3 | 2009-02-07  | Entregado |
|             3 | 2009-02-07  | Entregado |
|             3 | 2009-02-10  | Entregado |
|             3 | 2009-02-10  | Entregado |
+---------------+-------------+-----------+
15 rows in set (0.00 sec)
```

SOLUTION QUERY 19.2

El cliente tiene un problema con ADUANAS. Nos pide, de manera urgente, un listado de las lineas de pedido que contienen productos con códigos que NO acaben en 0, 2, 4 o 6 que vienen de Francia (el código de producto empieza por FR) y cuyo de precio de venta al público es superior a 70 euros. Todo debe ir ordenado por pedido, linea, y producto.

Nos indican una captura para que sepamos exactamente lo que quieren y los nombres de cada columna:

```
SELECT codigopedido AS Pedido, numerolinea AS Línea, codigoproducto AS Producto, preciounidad AS PVP
FROM detallepedidos
WHERE preciounidad>70 AND codigoproducto LIKE 'FR%'
    AND codigoproducto NOT LIKE '%2'
    AND codigoproducto NOT LIKE '%4'
    AND codigoproducto NOT LIKE '%6'
    AND codigoproducto NOT LIKE '%0'
ORDER BY 1, 2, 3;
```

```
mysql> SELECT codigopedido AS Pedido, numerolinea AS Línea, codigoproducto AS Producto,
 preciounidad AS PVP
    -> FROM detallepedidos
    -> WHERE preciounidad>70 AND codigoproducto LIKE 'FR%'
    -> AND codigoproducto NOT LIKE '%2'
    -> AND codigoproducto NOT LIKE '%4'
    -> AND codigoproducto NOT LIKE '%6'
    -> AND codigoproducto NOT LIKE '%0'
    -> ORDER BY 1, 2, 3;
+--------+-------+----------+--------+
| Pedido | Línea | Producto | PVP    |
+--------+-------+----------+--------+
|      8 |     3 | FR-11    | 100.00 |
|      9 |     2 | FR-69    |  91.00 |
|     10 |     1 | FR-91    |  75.00 |
|     13 |     3 | FR-11    | 100.00 |
|     28 |     3 | FR-11    |  99.00 |
|     57 |     4 | FR-69    |  91.00 |
|     80 |     3 | FR-11    | 100.00 |
|     92 |     2 | FR-11    | 100.00 |
|     94 |     3 | FR-11    | 100.00 |
|    107 |     1 | FR-11    | 100.00 |
+--------+-------+----------+--------+
10 rows in set (0.00 sec)
```