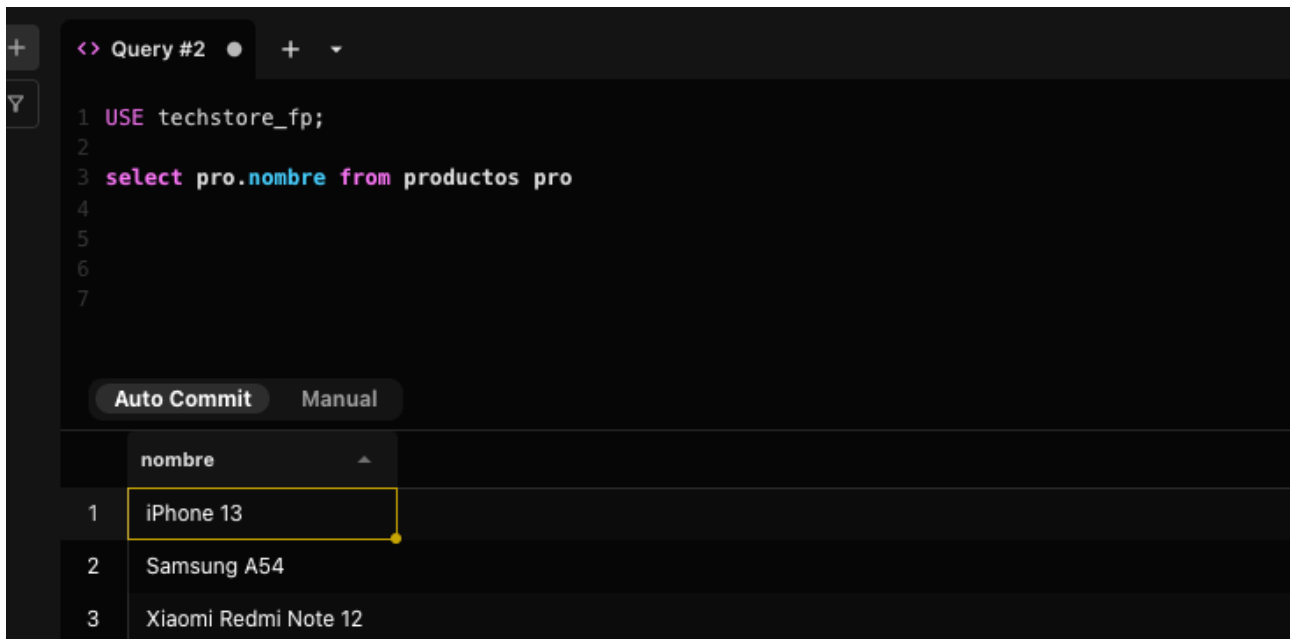


EJERCICIOS DE CONSULTAS SOBRE UNA TABLA 02

Realice las siguientes consultas teniendo en cuenta la base de datos tienda.

1) Lista el nombre de todos los productos que hay en la tabla producto.

`select nombre from producto;`



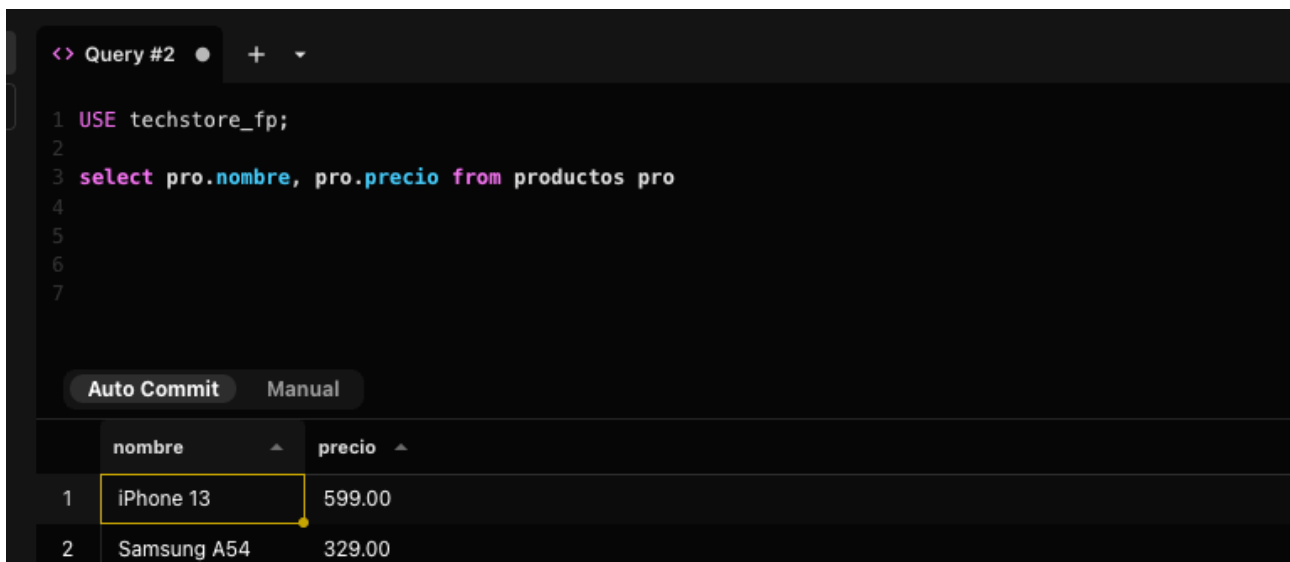
```
<> Query #2 ● + -
1 USE techstore_fp;
2
3 select pro.nombre from productos pro
4
5
6
7
```

Auto Commit Manual

	nombre
1	iPhone 13
2	Samsung A54
3	Xiaomi Redmi Note 12

2) Lista los nombres y los precios de todos los productos de la tabla producto.

`select nombre, precio from producto;`



```
<> Query #2 ● + -
1 USE techstore_fp;
2
3 select pro.nombre, pro.precio from productos pro
4
5
6
7
```

Auto Commit Manual

	nombre	precio
1	iPhone 13	599.00
2	Samsung A54	329.00

3) Lista todas las columnas de la tabla producto.

```
select * from producto;
```

Xampp MacOS - Beekeeper Studio - Free Version

```
Query #2
1 USE techstore_fp;
2
3 select * from productos pro
4
5
6
7
```

Auto Commit Manual

	id_producto	nombre	precio	stock	id_categoria
1	1	iPhone 13	599.00	10	1
2	2	Samsung A54	329.00	15	1

4) Lista el nombre de los productos, el precio en euros y el precio en dólares estadounidenses (USD).

```
select nombre, concat(precio, ' ', '€'), concat(truncate(precio*1.11 ,2), ' ', '$') from
producto;
```

techstore_fp *localhost Script-3 X

```
USE techstore_fp;
SELECT p.nombre,
CONCAT(p.precio, '€') AS Precio_Euros,
CONCAT((p.precio * 1.11), '$') AS Precio_Dolar
FROM productos p
```

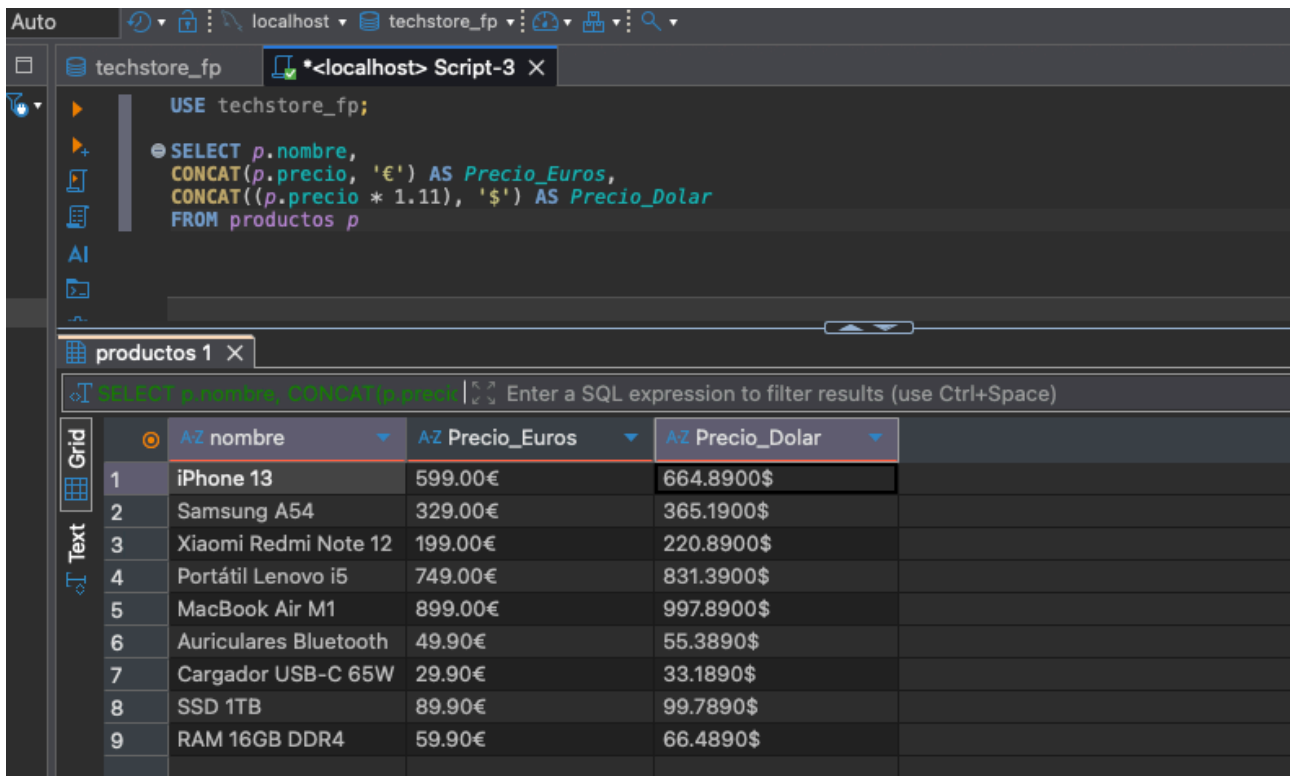
productos 1 X

SELECT p.nombre, CONCAT(p.precio, '€') AS Precio_Euros, CONCAT((p.precio * 1.11), '\$') AS Precio_Dolar

Enter a SQL expression to filter results (use Ctrl+Space)

	Az nombre	Az Precio_Euros	Az Precio_Dolar
1	iPhone 13	599.00€	664.8900\$
2	Samsung A54	329.00€	365.1900\$
3	Xiaomi Redmi Note 12	199.00€	220.8900\$
4	Portátil Lenovo i5	749.00€	831.3900\$
5	MacBook Air M1	899.00€	997.8900\$
6	Auriculares Bluetooth	49.90€	55.3890\$
7	Cargador USB-C 65W	29.90€	33.1890\$
8	SSD 1TB	89.90€	99.7890\$
9	RAM 16GB DDR4	59.90€	66.4890\$

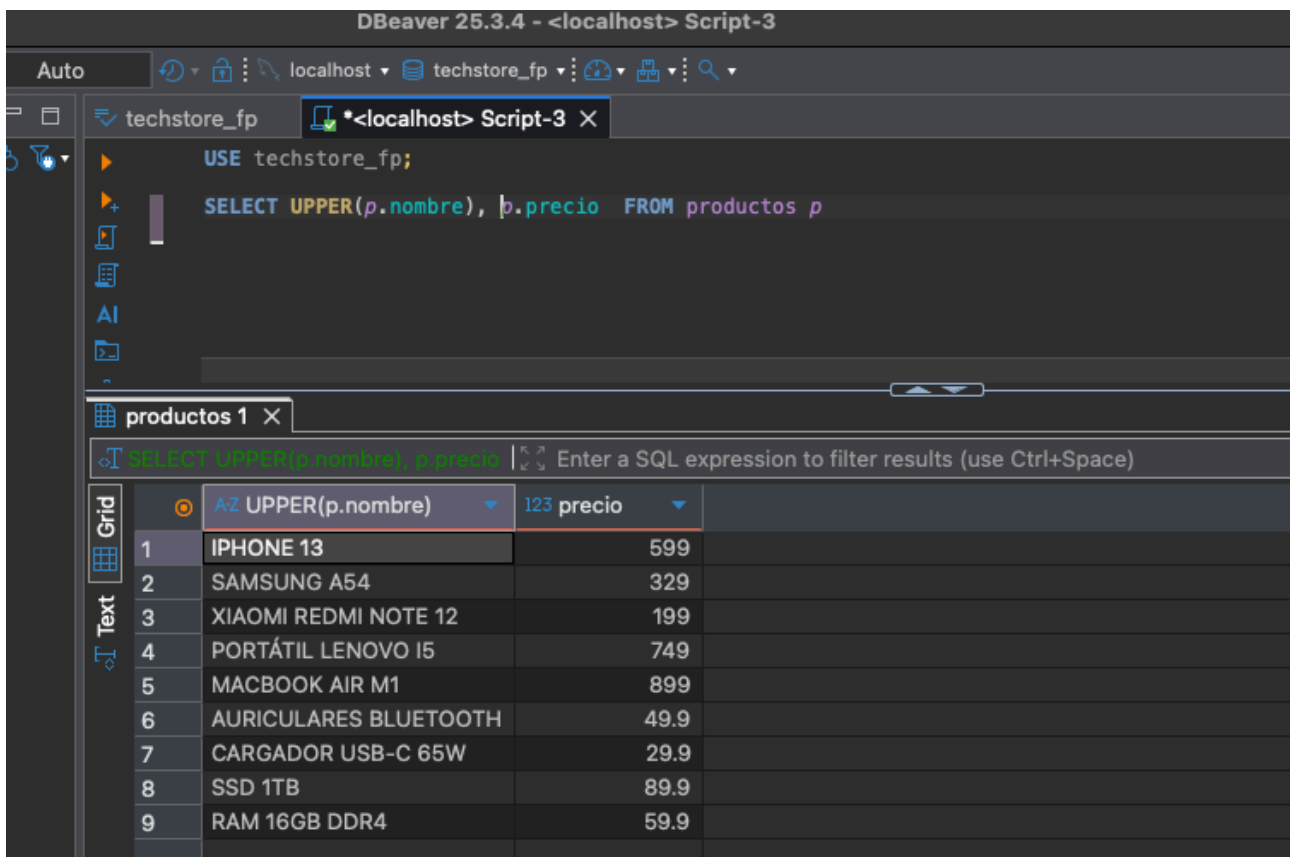
5) Lista el nombre de los productos, el precio en euros y el precio en dólares estadounidenses (USD). Utiliza los siguientes alias para las columnas: nombre de producto, euros, dólares.
`select nombre, concat(precio, ' ', '€') as euros, concat(truncate(precio*1.11,2), ' ', '$') as dolares from producto;`



The screenshot shows the DBeaver interface with a SQL script named 'Script-3' executed on the 'techstore_fp' database. The query selects product names, price in Euros, and price in US Dollars. The results are displayed in a grid view with 9 rows of data.

	AZ nombre	AZ Precio_Euros	AZ Precio_Dolar
1	iPhone 13	599.00€	664.8900\$
2	Samsung A54	329.00€	365.1900\$
3	Xiaomi Redmi Note 12	199.00€	220.8900\$
4	Portátil Lenovo i5	749.00€	831.3900\$
5	MacBook Air M1	899.00€	997.8900\$
6	Auriculares Bluetooth	49.90€	55.3890\$
7	Cargador USB-C 65W	29.90€	33.1890\$
8	SSD 1TB	89.90€	99.7890\$
9	RAM 16GB DDR4	59.90€	66.4890\$

6) Lista los nombres y los precios de todos los productos de la tabla producto, convirtiendo los nombres a mayúscula.
`select upper(nombre),precio from producto;`

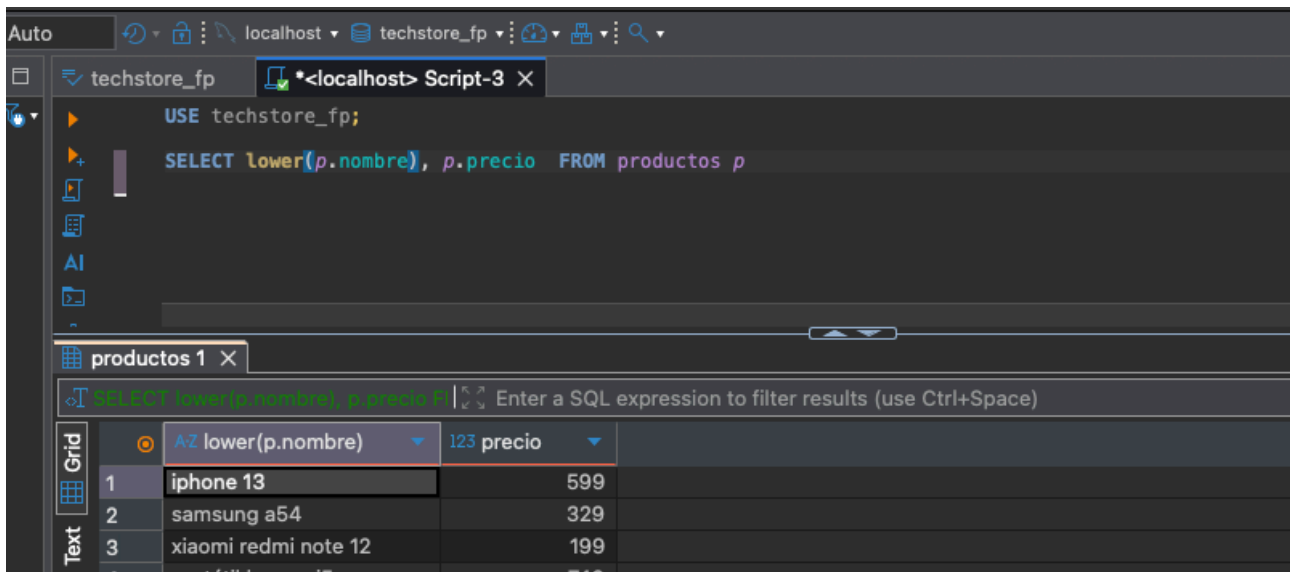


The screenshot shows the DBeaver interface with a SQL script named 'Script-3' executed on the 'techstore_fp' database. The query selects product names converted to uppercase and their prices. The results are displayed in a grid view with 9 rows of data.

	AZ UPPER(p.nombre)	123 precio
1	IPHONE 13	599
2	SAMSUNG A54	329
3	XIAOMI REDMI NOTE 12	199
4	PORTÁTIL LENOVO I5	749
5	MACBOOK AIR M1	899
6	AURICULARES BLUETOOTH	49.9
7	CARGADOR USB-C 65W	29.9
8	SSD 1TB	89.9
9	RAM 16GB DDR4	59.9

7) Lista los nombres y los precios de todos los productos de la tabla producto, convirtiendo los nombres a minúscula.

```
select lower(nombre),precio from producto;
```



The screenshot shows a SQL IDE interface. The top pane contains the following SQL query:

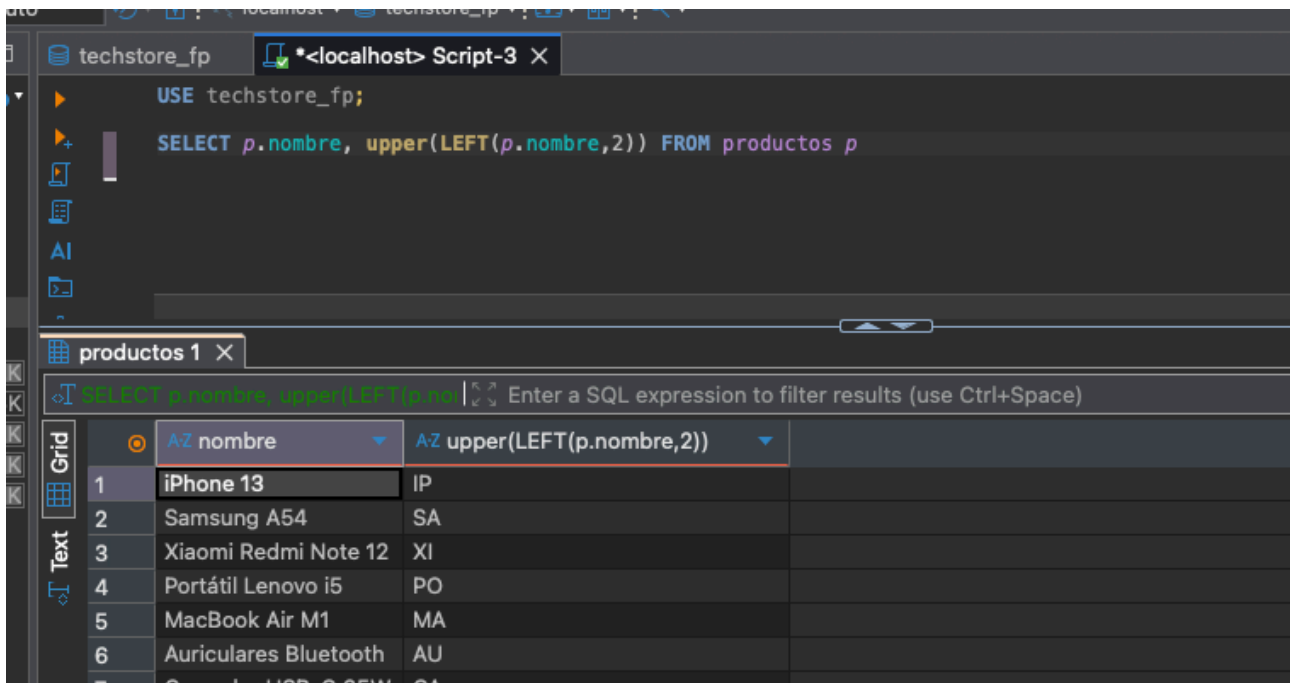
```
USE techstore_fp;  
  
SELECT lower(p.nombre), p.precio FROM productos p
```

The bottom pane displays the results of the query in a table named 'productos 1'. The table has two columns: 'lower(p.nombre)' and 'precio'. The data is as follows:

	lower(p.nombre)	precio
1	iphone 13	599
2	samsung a54	329
3	xiaomi redmi note 12	199
4	portátil lenovo i5	719

8) Lista el nombre de todos los fabricantes en una columna, y en otra columna obtenga en mayúsculas los dos primeros caracteres del nombre del fabricante.

```
select nombre, upper( left(nombre,2)) from fabricante;
```



The screenshot shows a SQL IDE interface. The top pane contains the following SQL query:

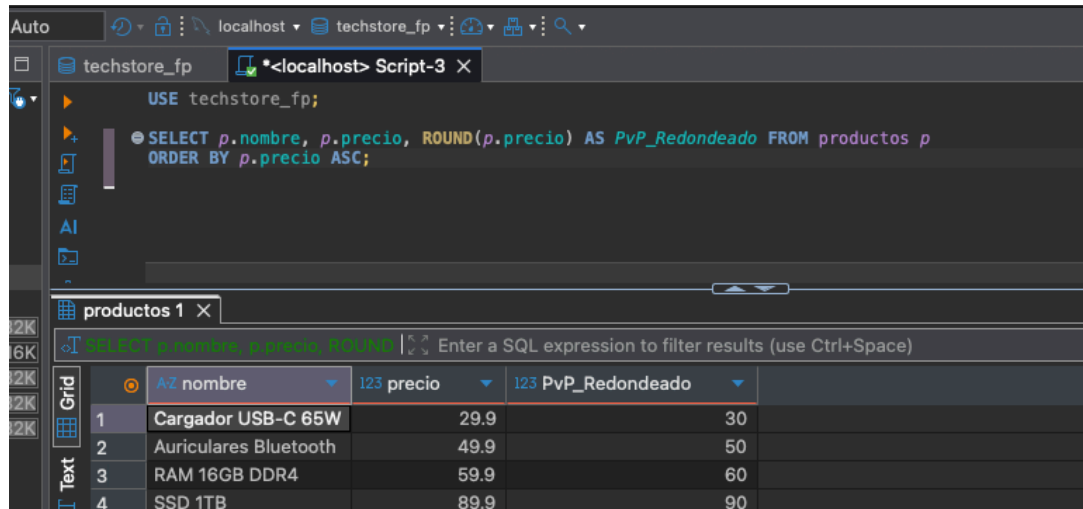
```
USE techstore_fp;  
  
SELECT p.nombre, upper(LEFT(p.nombre,2)) FROM productos p
```

The bottom pane displays the results of the query in a table named 'productos 1'. The table has two columns: 'nombre' and 'upper(LEFT(p.nombre,2))'. The data is as follows:

	nombre	upper(LEFT(p.nombre,2))
1	iPhone 13	IP
2	Samsung A54	SA
3	Xiaomi Redmi Note 12	XI
4	Portátil Lenovo i5	PO
5	MacBook Air M1	MA
6	Auriculares Bluetooth	AU
7	Cargador USB-C 65W	CA

9) Lista los nombres y los precios de todos los productos de la tabla producto, redondeando el valor del precio.

`select nombre, round(precio) from producto;`



The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL code:

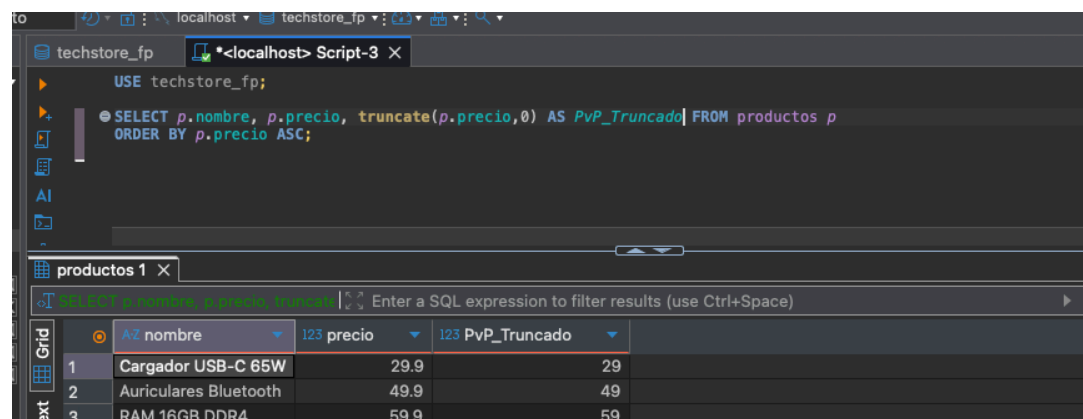
```
USE techstore_fp;  
  
SELECT p.nombre, p.precio, ROUND(p.precio) AS PvP_Redondeado FROM productos p  
ORDER BY p.precio ASC;
```

The results pane shows a table with the following data:

	AZ nombre	123 precio	123 PvP_Redondeado
1	Cargador USB-C 65W	29.9	30
2	Auriculares Bluetooth	49.9	50
3	RAM 16GB DDR4	59.9	60
4	SSD 1TB	89.9	90

10) Lista los nombres y los precios de todos los productos de la tabla producto, truncando el valor del precio para mostrarlo sin ninguna cifra decimal.

`select nombre, truncate(precio,0) from producto;`



The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL code:

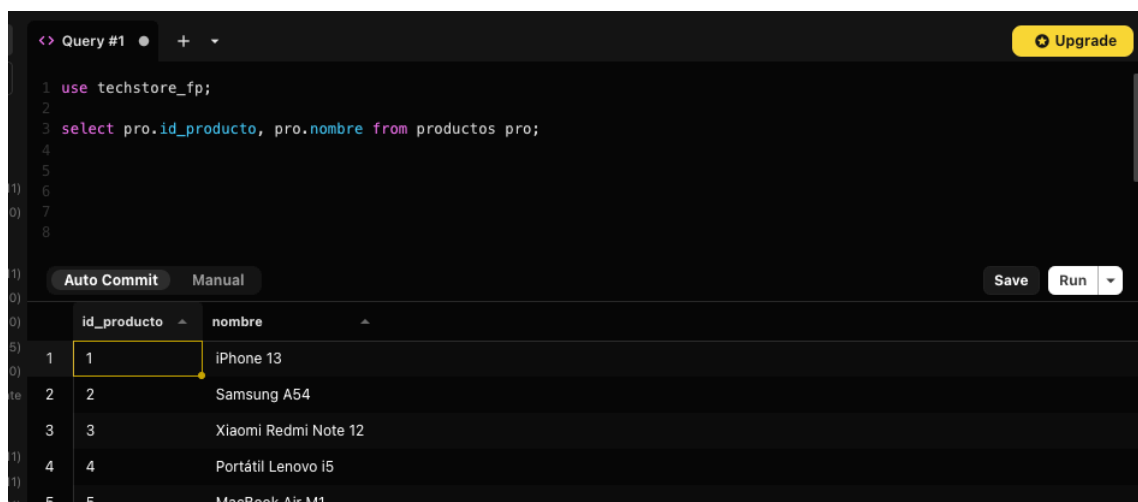
```
USE techstore_fp;  
  
SELECT p.nombre, p.precio, truncate(p.precio,0) AS PvP_Truncado FROM productos p  
ORDER BY p.precio ASC;
```

The results pane shows a table with the following data:

	AZ nombre	123 precio	123 PvP_Truncado
1	Cargador USB-C 65W	29.9	29
2	Auriculares Bluetooth	49.9	49
3	RAM 16GB DDR4	59.9	59

11) Lista el código de los fabricantes que tienen productos en la tabla producto.

`select codigo_fabricante from producto;`



The screenshot shows a SQL query editor with the following SQL code:

```
use techstore_fp;  
  
select pro.id_producto, pro.nombre from productos pro;
```

The results pane shows a table with the following data:

	id_producto	nombre
1	1	iPhone 13
2	2	Samsung A54
3	3	Xiaomi Redmi Note 12
4	4	Portátil Lenovo i5
5	5	MacBook Air M1

12) Lista el código de los fabricantes que tienen productos en la tabla producto, eliminando los códigos que aparecen repetidos.

```
select distinct(codigo_fabricante) from producto;
```

```
1 use techstore_fp;
2
3 select distinct(pro.id_producto), pro.nombre from productos pro;
```

Auto Commit Manual Save Run

	id_producto	nombre
1	1	iPhone 13
2	2	Samsung A54
3	3	Xiaomi Redmi Note 12
4	4	Portátil Lenovo i5
5	5	MacBook Air M1

13) Lista los nombres de los fabricantes ordenados de forma ascendente.

```
select nombre from fabricante order by nombre;
```

```
1 use techstore_fp;
2
3 select pro.nombre from productos pro
4 order by pro.nombre
```

Auto Commit Manual Save Run Upgrade

	nombre
1	Auriculares Bluetooth
2	Cargador USB-C 65W
3	iPhone 13
4	MacBook Air M1

14)Lista los nombres de los fabricantes ordenados de forma descendente.

`select nombre from fabricante order by nombre desc;`

The screenshot shows a SQL IDE interface. At the top, there's a tab labeled 'Query #1' and a yellow 'Upgrade' button. The SQL editor contains the following code:

```
1 use techstore_fp;
2
3 select pro.nombre from productos pro
4   order by pro.nombre desc;
5
6
7
8
```

Below the editor, there are buttons for 'Auto Commit', 'Manual', 'Save', and 'Run'. The 'Run' button is highlighted. Below the buttons, a table of results is displayed with the column header 'nombre'.

	nombre
1	Xiaomi Redmi Note 12
2	SSD 1TB
3	Samsung A54
4	RAM 16GB DDR4

15)Lista los nombres de los productos ordenados en primer lugar por el nombre de forma ascendente y en segundo lugar por el precio de forma descendente.

`select nombre from producto order by nombre;`

`select nombre from producto order by nombre desc;`

The screenshot shows a SQL IDE interface. The SQL editor contains the following code:

```
1 use techstore_fp;
2
3 select pro.nombre from productos pro
4   order by pro.nombre desc, pro.precio;
5
6
7
8
```

Below the editor, there are buttons for 'Auto Commit', 'Manual', 'Save', and 'Run'. The 'Run' button is highlighted. Below the buttons, a table of results is displayed with the column header 'nombre'.

	nombre
1	Xiaomi Redmi Note 12
2	SSD 1TB
3	Samsung A54
4	RAM 16GB DDR4

16)Devuelve una lista con las 5 primeras filas de la tabla fabricante.

`select * from fabricante limit 5;`

The screenshot shows a SQL IDE interface. The SQL editor contains the following code:

```
1 use techstore_fp;
2
3 select pro.nombre from productos pro
4   limit 5;
5
6
7
8
```

Below the editor, there are buttons for 'Auto Commit', 'Manual', and 'Save'. The 'Auto Commit' button is highlighted. Below the buttons, a table of results is displayed with the column header 'nombre'.

	nombre
1	iPhone 13
2	Samsung A54
3	Xiaomi Redmi Note 12
4	Portátil Lenovo i5
5	MacBook Air M1

17) Devuelve una lista con 2 filas a partir de la cuarta fila de la tabla fabricante. La cuarta fila también se debe incluir en la respuesta.

```
select * from fabricante limit 5 offset 3;
```

```
1 use techstore_fp;
2
3 select pro.nombre from productos pro
4   limit 2 offset 3;
```

Auto Commit Manual

	nombre
1	Portátil Lenovo i5
2	MacBook Air M1

18) Lista el nombre y el precio del producto más barato. (Utilice solamente las cláusulas ORDER BY y LIMIT)

```
select * from producto order by precio limit 1;
```

```
1 use techstore_fp;
2
3 select pro.nombre, pro.precio from productos pro
4   order by pro.precio asc
5   limit 1;
```

Auto Commit Manual Save

	nombre	precio
1	Cargador USB-...	29.90

19) Lista el nombre y el precio del producto más caro. (Utilice solamente las cláusulas ORDER BY y LIMIT)

```
select * from producto order by precio desc limit 1;
```

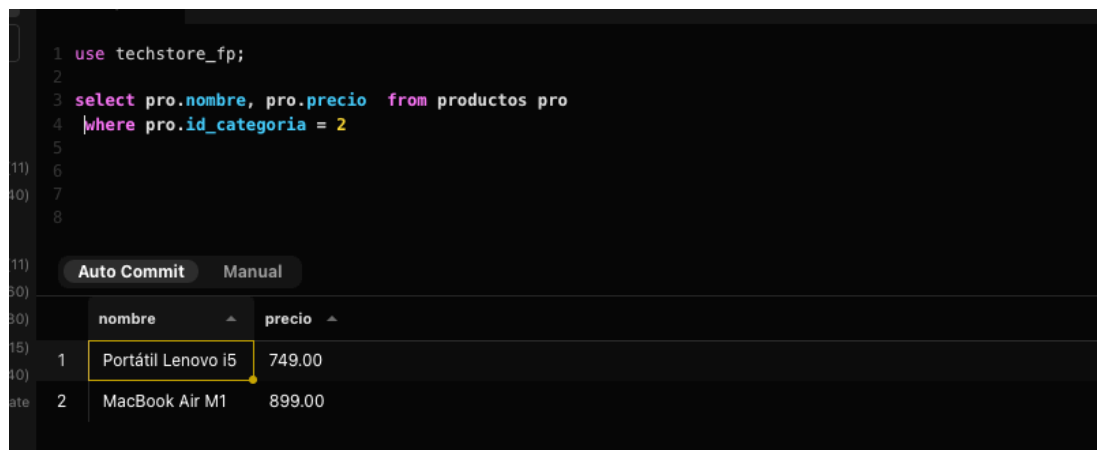
```
<> Query #1 ● + -
1 use techstore_fp;
2
3 select pro.nombre, pro.precio from productos pro
4   order by pro.precio desc
5   limit 1;
```

Auto Commit Manual Save

	nombre	precio
1	MacBook Air M1	899.00

20)Lista el nombre de todos los productos del fabricante cuyo código de fabricante es igual a 2.

`select nombre from producto where codigo_fabricante = 2;`

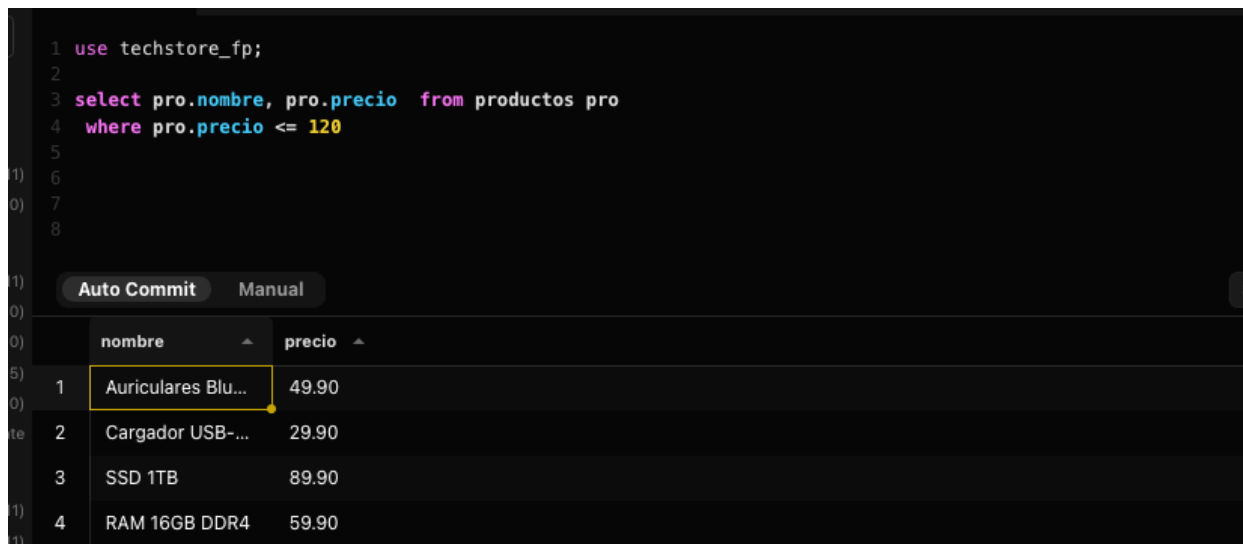


The screenshot shows a SQL IDE with a query editor and a results pane. The query is: `use techstore_fp; select pro.nombre, pro.precio from productos pro where pro.id_categoria = 2`. The results pane shows two rows of data.

	nombre	precio
1	Portátil Lenovo i5	749.00
2	MacBook Air M1	899.00

21)Lista el nombre de los productos que tienen un precio menor o igual a 120€.

`select nombre from producto where precio <= 120;`

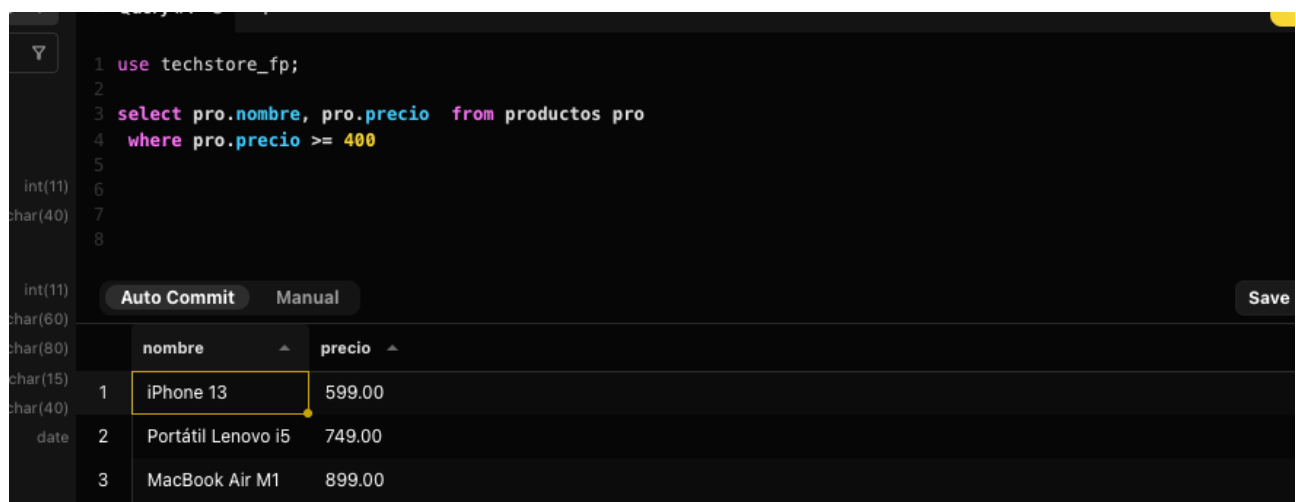


The screenshot shows a SQL IDE with a query editor and a results pane. The query is: `use techstore_fp; select pro.nombre, pro.precio from productos pro where pro.precio <= 120`. The results pane shows four rows of data.

	nombre	precio
1	Auriculares Blu...	49.90
2	Cargador USB-...	29.90
3	SSD 1TB	89.90
4	RAM 16GB DDR4	59.90

22)Lista el nombre de los productos que tienen un precio mayor o igual a 400€.

`select nombre from producto where precio >= 400;`



The screenshot shows a SQL IDE with a query editor and a results pane. The query is: `use techstore_fp; select pro.nombre, pro.precio from productos pro where pro.precio >= 400`. The results pane shows three rows of data.

	nombre	precio
1	iPhone 13	599.00
2	Portátil Lenovo i5	749.00
3	MacBook Air M1	899.00

23)Lista el nombre de los productos que no tienen un precio mayor o igual a 400€.

`select nombre from producto where precio <= 400;`

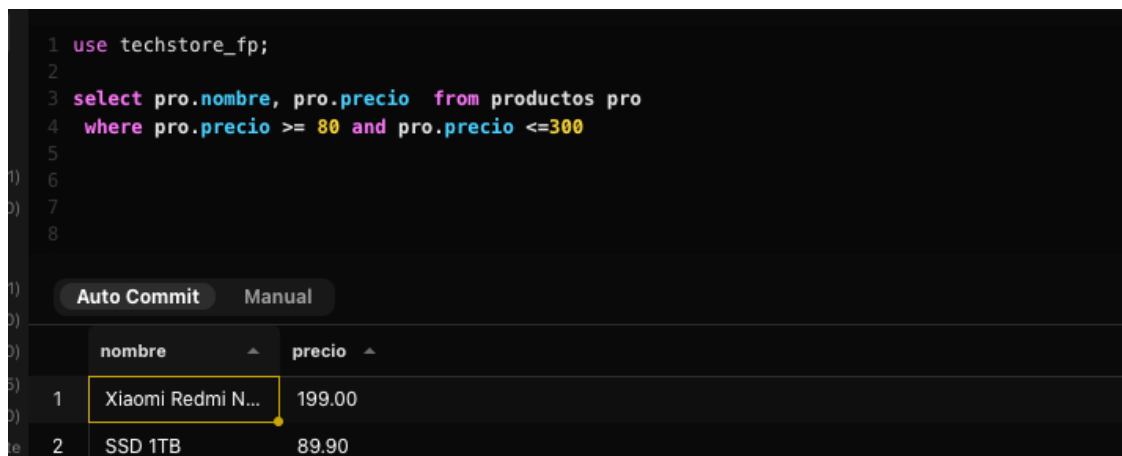


The screenshot shows a SQL IDE with a query editor and a results table. The query is: `use techstore_fp; select pro.nombre, pro.precio from productos pro where pro.precio <= 400`. The results table has two columns: 'nombre' and 'precio'. It contains six rows of data.

	nombre	precio
1	Samsung A54	329.00
2	Xiaomi Redmi N...	199.00
3	Auriculares Blu...	49.90
4	Cargador USB-...	29.90
5	SSD 1TB	89.90
6	RAM 16GB DDR4	59.90

24)Lista todos los productos que tengan un precio entre 80€ y 300€. Sin utilizar el operador BETWEEN.

`select nombre from producto where precio >= 80 and precio <= 300;`

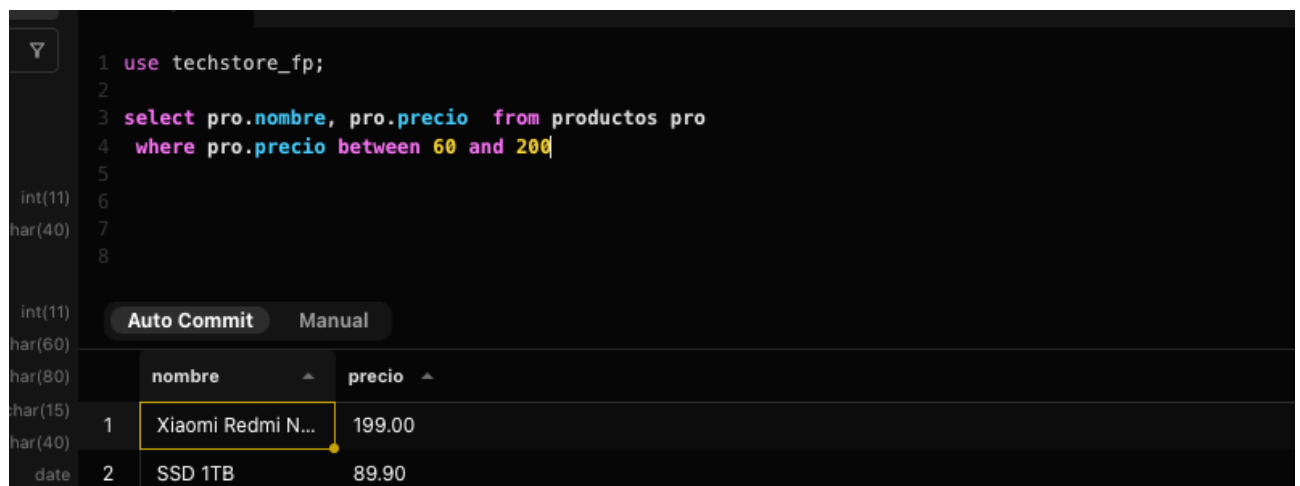


The screenshot shows a SQL IDE with a query editor and a results table. The query is: `use techstore_fp; select pro.nombre, pro.precio from productos pro where pro.precio >= 80 and pro.precio <= 300`. The results table has two columns: 'nombre' and 'precio'. It contains two rows of data.

	nombre	precio
1	Xiaomi Redmi N...	199.00
2	SSD 1TB	89.90

25)Lista todos los productos que tengan un precio entre 60€ y 200€. Utilizando el operador BETWEEN.

`select nombre from producto where precio between 60 and 200;`

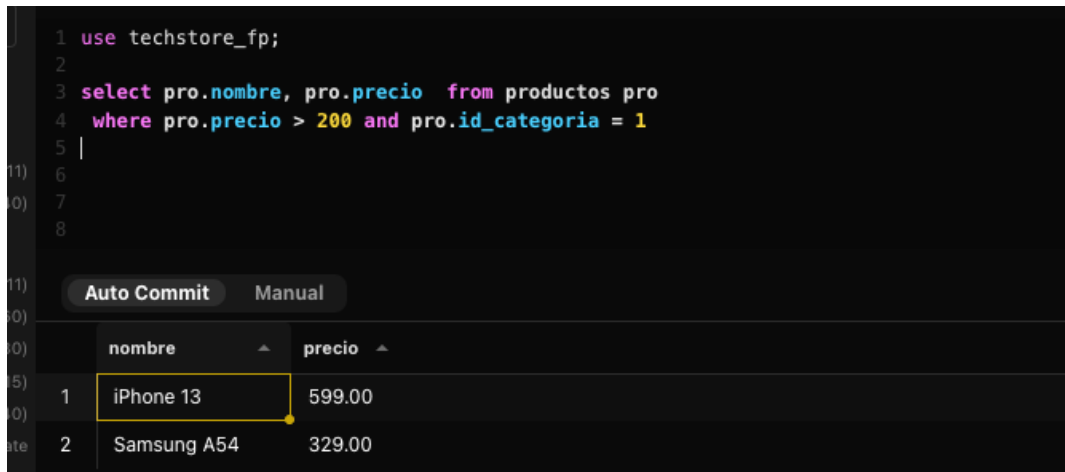


The screenshot shows a SQL IDE with a query editor and a results table. The query is: `use techstore_fp; select pro.nombre, pro.precio from productos pro where pro.precio between 60 and 200`. The results table has two columns: 'nombre' and 'precio'. It contains two rows of data.

	nombre	precio
1	Xiaomi Redmi N...	199.00
2	SSD 1TB	89.90

26)Lista todos los productos que tengan un precio mayor que 200€ y que el código de fabricante sea igual a 6.

```
select nombre from producto where precio > 200 and codigo_fabricante = 6;
```



The screenshot shows a SQL IDE with a dark theme. The query editor contains the following SQL code:

```
1 use techstore_fp;
2
3 select pro.nombre, pro.precio from productos pro
4 where pro.precio > 200 and pro.id_categoria = 1
5 |
6
7
8
```

Below the query editor, there are two buttons: "Auto Commit" and "Manual". Below the buttons, a table of results is displayed with two columns: "nombre" and "precio".

	nombre	precio
1	iPhone 13	599.00
2	Samsung A54	329.00

27)Lista todos los productos donde el código de fabricante sea 1, 3 o 5. Sin utilizar el operador IN.

```
select nombre from producto where codigo_fabricante = 1 or codigo_fabricante = 3 or
codigo_fabricante = 5;
```

28)Lista todos los productos donde el código de fabricante sea 1, 3 o 5. Utilizando el operador IN.

```
select nombre from producto where codigo_fabricante in (1,3,5);
```

29)Lista el nombre y el precio de los productos en céntimos (Habría que multiplicar por 100 el valor del precio). Cree un alias para la columna que contiene el precio que se llame céntimos.

```
select nombre,precio*100 as centimos from producto;
```

30)Lista los nombres de los fabricantes cuyo nombre empiece por la letra S.

```
select nombre from fabricante where nombre like 'S%';
```

31)Lista los nombres de los fabricantes cuyo nombre termine por la vocal e.

```
select nombre from fabricante where nombre like '%e';
```

32)Lista los nombres de los fabricantes cuyo nombre contenga el carácter w.

```
select nombre from fabricante where nombre like '%w%';
```

33)Lista los nombres de los fabricantes cuyo nombre sea de 4 caracteres.

```
select nombre from fabricante where nombre like '____';
```

34)Devuelve una lista con el nombre de todos los productos que contienen la cadena Portátil en el nombre.

```
select nombre from producto where nombre like '%Portatil%';
```

35)Devuelve una lista con el nombre de todos los productos que contienen la cadena Monitor en el nombre y tienen un precio inferior a 215 €.

```
select nombre from producto where nombre like '%Monitor%' and precio >215;
```

36)Lista el nombre y el precio de todos los productos que tengan un precio mayor o igual a 180€. Ordene el resultado en primer lugar por el precio (en orden descendente) y en segundo lugar por el nombre (en orden ascendente).

```
select nombre, precio from producto where precio >= 180 order by precio desc;select nombre,
precio from producto where precio >= 180 order by nombre;
```