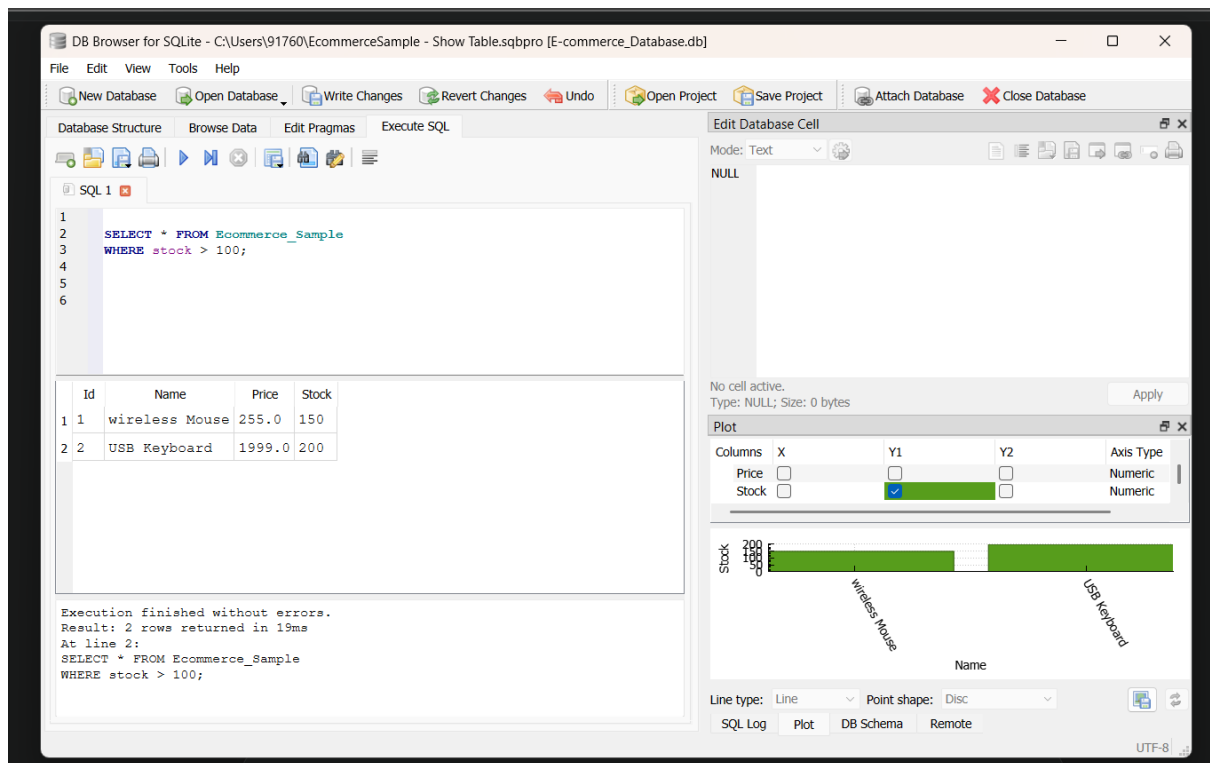
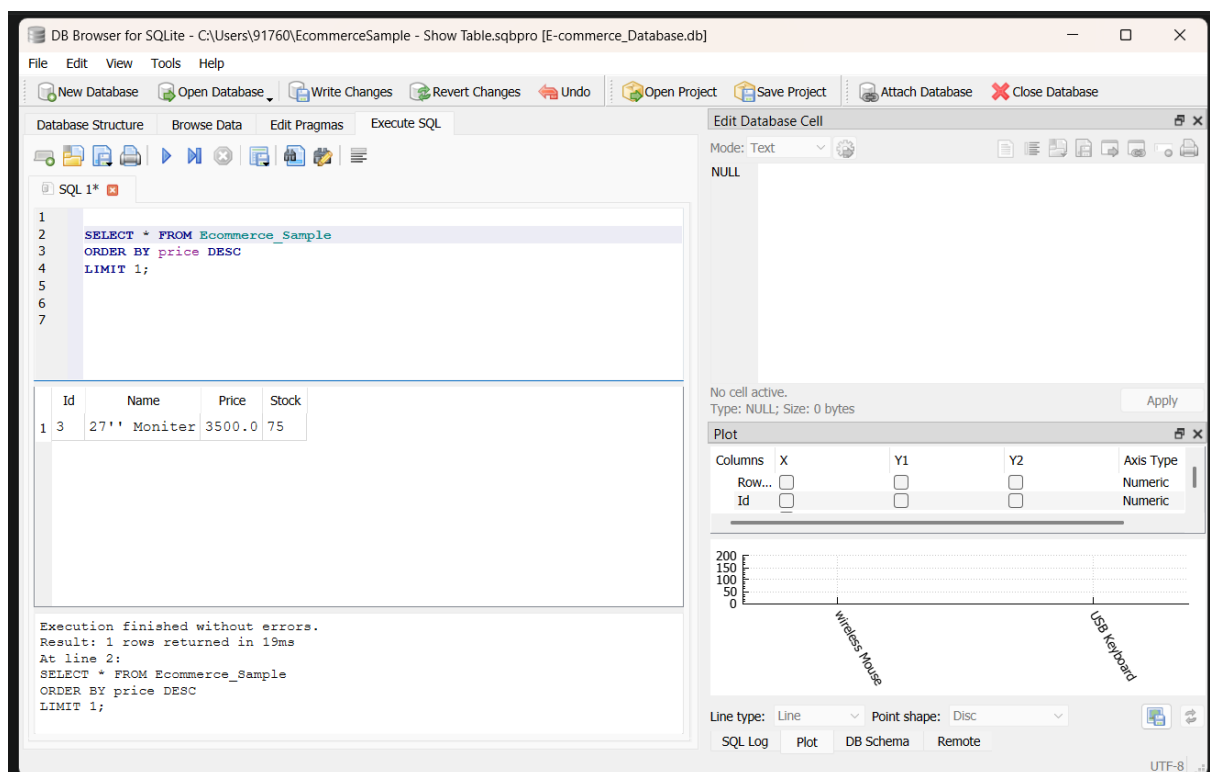


Queries I Practiced

1. Show only products with stock greater than 100

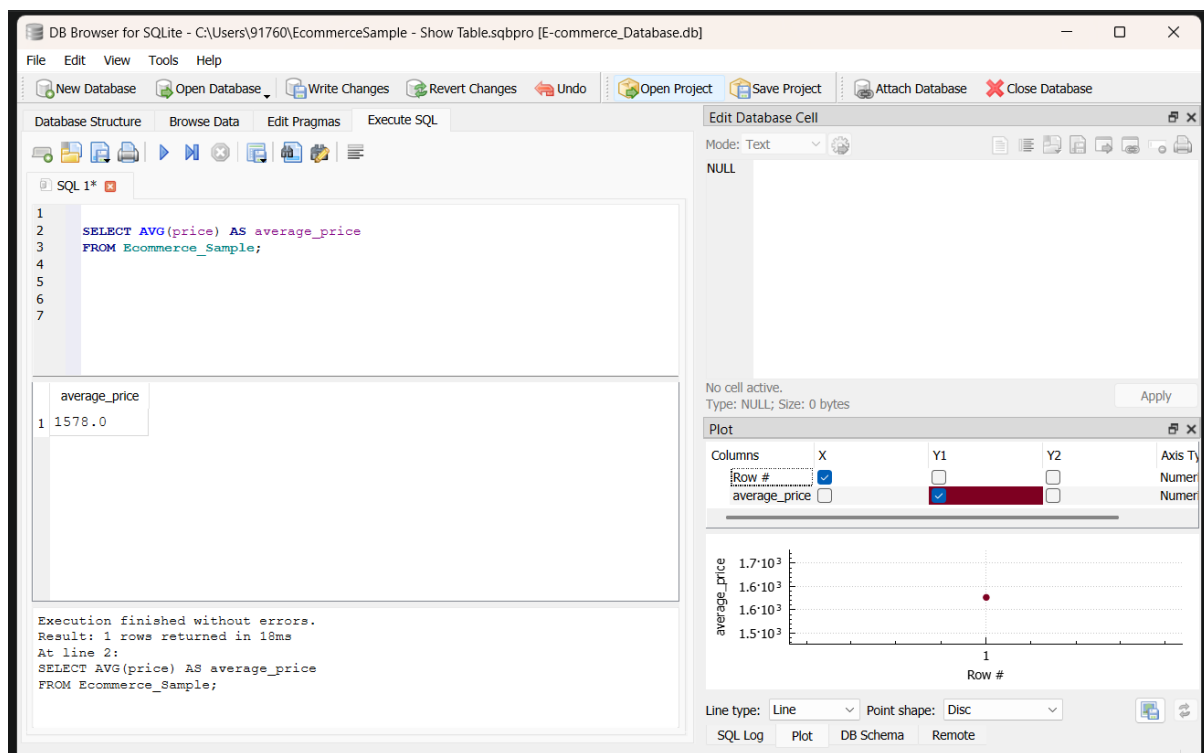


2. Show the most expensive product

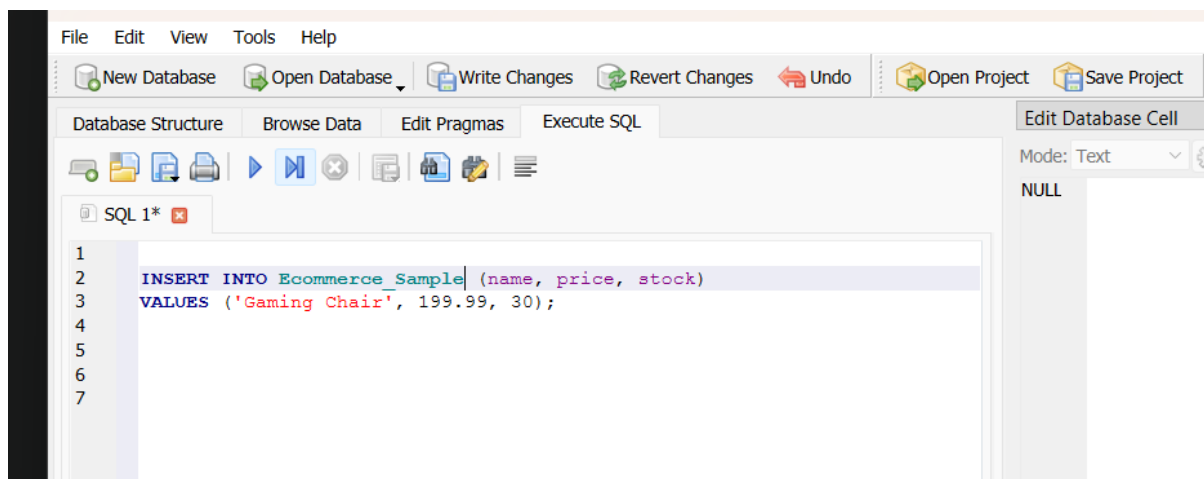


Queries I Practiced

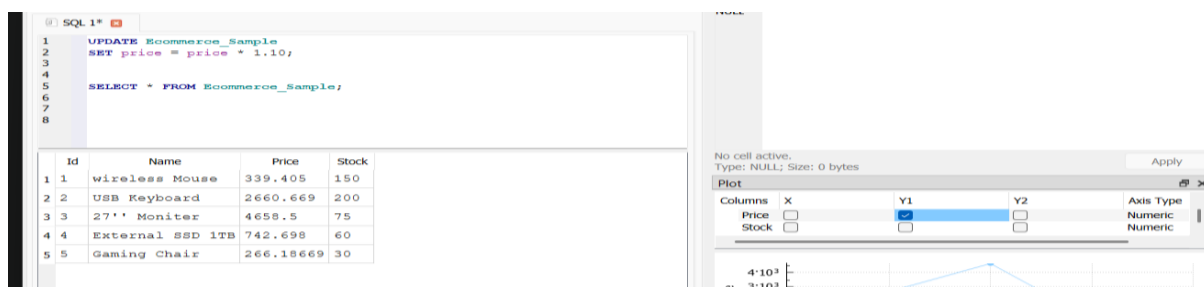
3. Calculate the average price of products



4. Add a new product



5. Increase all prices by 10%



Queries I Practiced

6. Search for products containing a specific word

The screenshot shows a SQL IDE interface with tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a query in a text editor. The query is:

```
1  
2 SELECT * FROM Ecommerce_Sample  
3 WHERE name LIKE '%Mouse%';  
4  
5  
6
```

Below the editor, the results are displayed in a table:

	Id	Name	Price	Stock
1	1	wireless Mouse	339.405	150

At the bottom, the execution status is shown:

Execution finished without errors.
Result: 1 rows returned in 17ms
At line 2:
SELECT * FROM Ecommerce_Sample
WHERE name LIKE '%Mouse%';

7. Count total number of products

The screenshot shows a SQL IDE interface with a query in the text editor:

```
1  
2 SELECT COUNT(*) AS total_products  
3 FROM Ecommerce_Sample;  
4  
5  
6
```

Below the editor, the results are displayed in a table:

	total_products
1	5

On the right side of the IDE, there are labels for 'No' and 'Type', and a 'Plc' button at the bottom.

Queries I Practiced

8. Show products that cost more than 100 and have stock less than 100

The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database is 'E-commerce_Database.db'. The 'Execute SQL' tab is active, displaying the following query:

```
1  
2 SELECT * FROM Ecommerce_Sample  
3 WHERE price > 100 AND stock < 100;  
4  
5  
6  
7
```

Below the query editor, the results are displayed in a table with 4 columns: Id, Name, Price, and Stock. The results show 3 rows of data:

	Id	Name	Price	Stock
1	3	27'' Moniter	4658.5	75
2	4	External SSD 1TB	742.698	60
3	5	Gaming Chair	266.18669	30

At the bottom, a status message reads: 'Execution finished without errors. Result: 3 rows returned in 19ms. At line 2: SELECT * FROM Ecommerce_Sample WHERE price > 100 AND stock < 100;'. On the right side, the 'Edit Data' panel shows 'Mode: Text' and 'NULL'.

9. Show products sorted by stock from highest to lowest

The screenshot shows the DB Browser for SQLite interface. The 'Execute SQL' tab is active, displaying the following query:

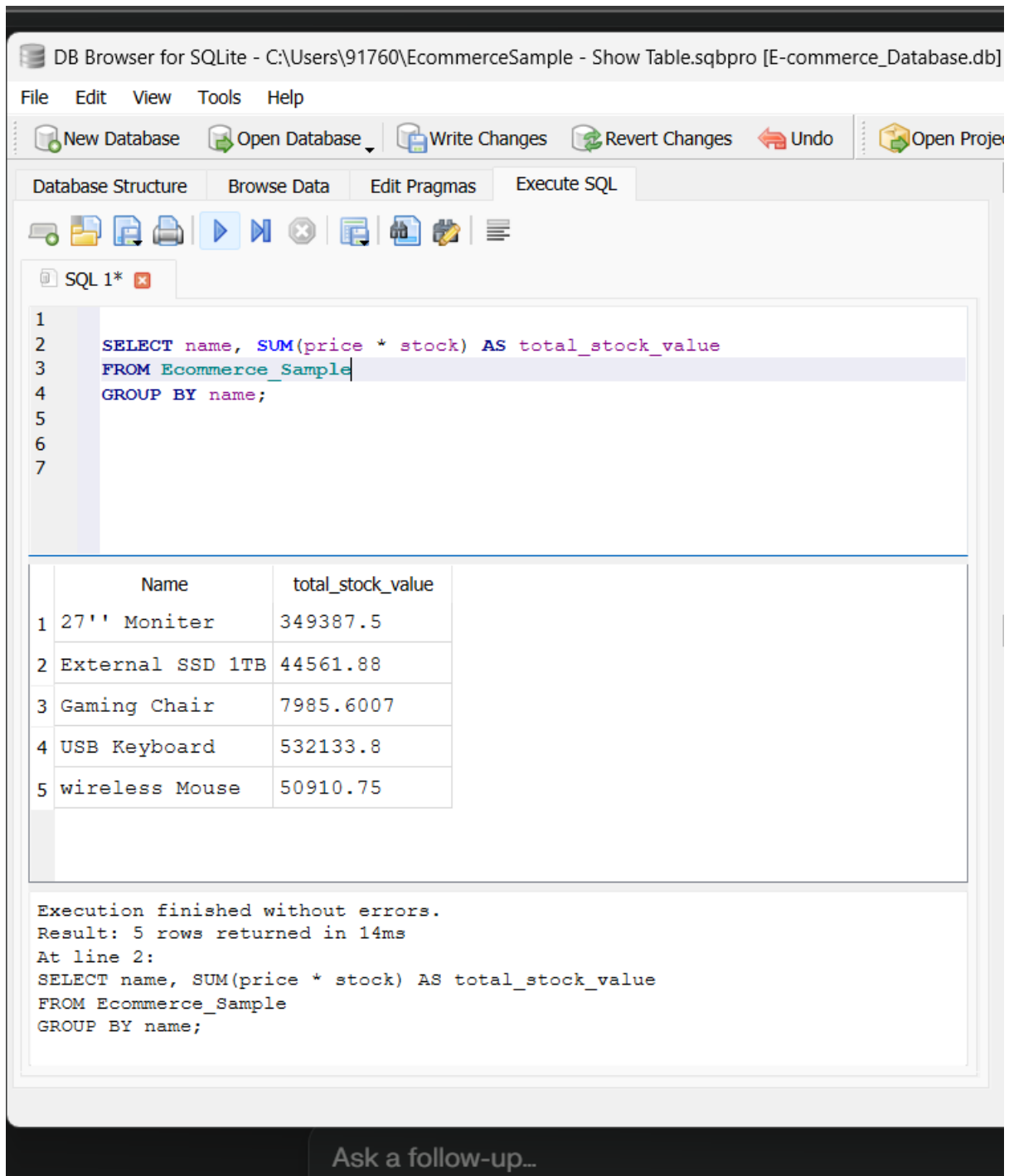
```
2 SELECT * FROM Ecommerce_Sample  
3 ORDER BY stock DESC;  
4  
5  
6
```

Below the query editor, the results are displayed in a table with 4 columns: Id, Name, Price, and Stock. The results show 5 rows of data, sorted by stock in descending order:

	Id	Name	Price	Stock
1	2	USB Keyboard	2660.669	200
2	1	wireless Mouse	339.405	150
3	3	27'' Moniter	4658.5	75
4	4	External SSD 1TB	742.698	60
5	5	Gaming Chair	266.18669	30

Queries I Practiced

10. Total stock value for each type of product.



The screenshot shows the DB Browser for SQLite application. The title bar indicates the database is 'C:\Users\91760\EcommerceSample - Show Table.sqbpro [E-commerce_Database.db]'. The 'Execute SQL' tab is active, displaying the following SQL query:

```
1  
2 SELECT name, SUM(price * stock) AS total_stock_value  
3 FROM Ecommerce_Sample  
4 GROUP BY name;  
5  
6  
7
```

Below the query editor, the results are displayed in a table with 5 rows:

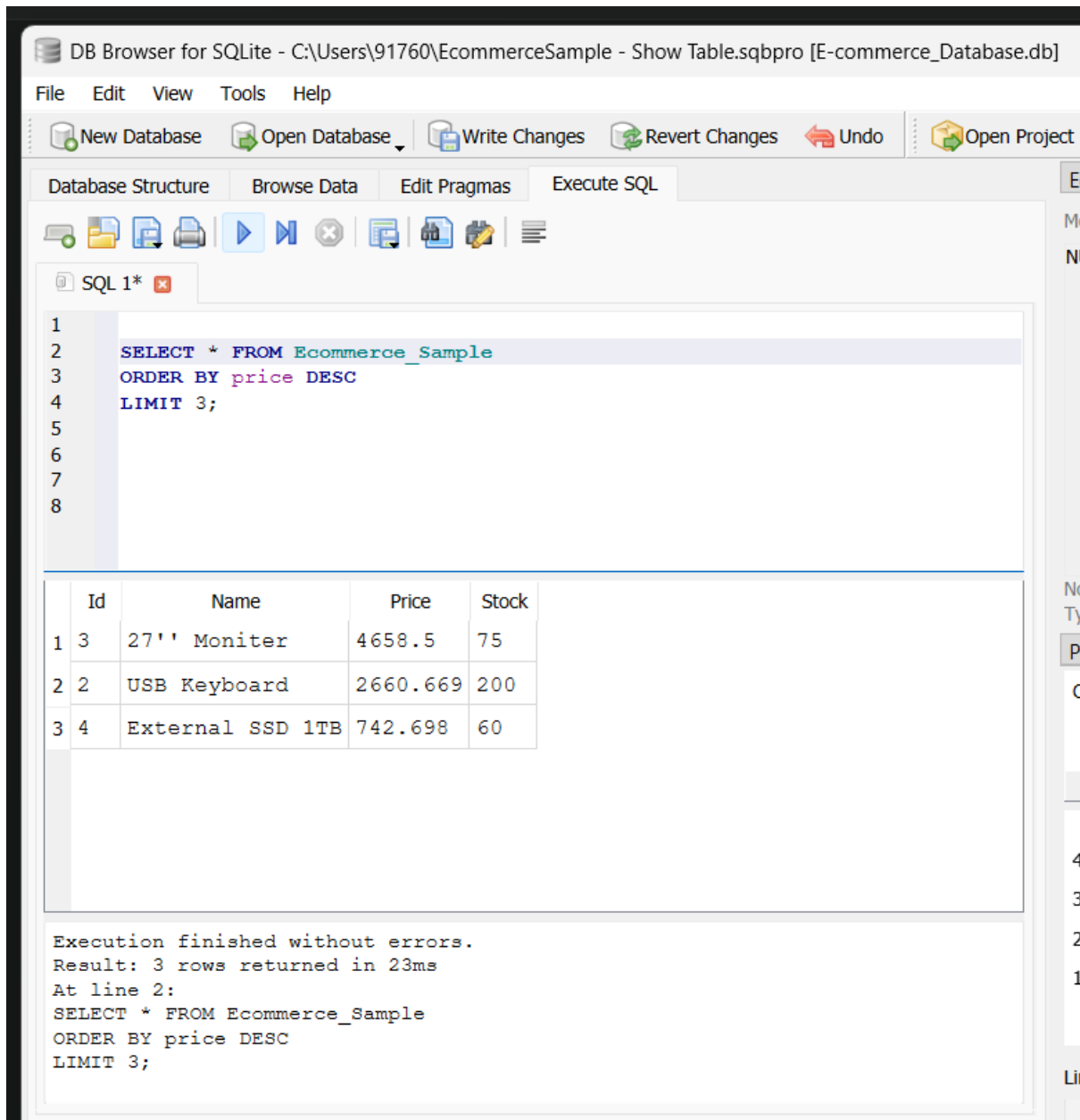
	Name	total_stock_value
1	27'' Moniter	349387.5
2	External SSD 1TB	44561.88
3	Gaming Chair	7985.6007
4	USB Keyboard	532133.8
5	wireless Mouse	50910.75

At the bottom, a status message reads: 'Execution finished without errors. Result: 5 rows returned in 14ms'. Below this, the executed SQL query is repeated.

Ask a follow-up...

Queries I Practiced

11. Top 3 most expensive products.



The screenshot shows the DB Browser for SQLite application. The title bar indicates the database is 'C:\Users\91760\EcommerceSample - Show Table.sqbpro [E-commerce_Database.db]'. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains icons for New Database, Open Database, Write Changes, Revert Changes, Undo, and Open Project. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The SQL editor shows a query to select the top 3 most expensive products from the 'Ecommerce_Sample' table, ordered by price in descending order.

```
1  
2 SELECT * FROM Ecommerce_Sample  
3 ORDER BY price DESC  
4 LIMIT 3;  
5  
6  
7  
8
```

	Id	Name	Price	Stock
1	3	27'' Moniter	4658.5	75
2	2	USB Keyboard	2660.669	200
3	4	External SSD 1TB	742.698	60

Execution finished without errors.
Result: 3 rows returned in 23ms
At line 2:
SELECT * FROM Ecommerce_Sample
ORDER BY price DESC
LIMIT 3;

Queries I Practiced

12. Delete a Product (use with caution!)

The screenshot shows the DB Browser for SQLite application. The title bar indicates the file path: C:\Users\91760\EcommerceSample - Show Table.sqbpro [E-commerce_Database.db]. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains icons for New Database, Open Database, Write Changes, Revert Changes, Undo, and Open Project. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a query editor with the following SQL code:

```
1  
2 DELETE FROM Ecommerce_Sample  
3 WHERE name = 'USB Keyboard';  
4  
5  
6
```

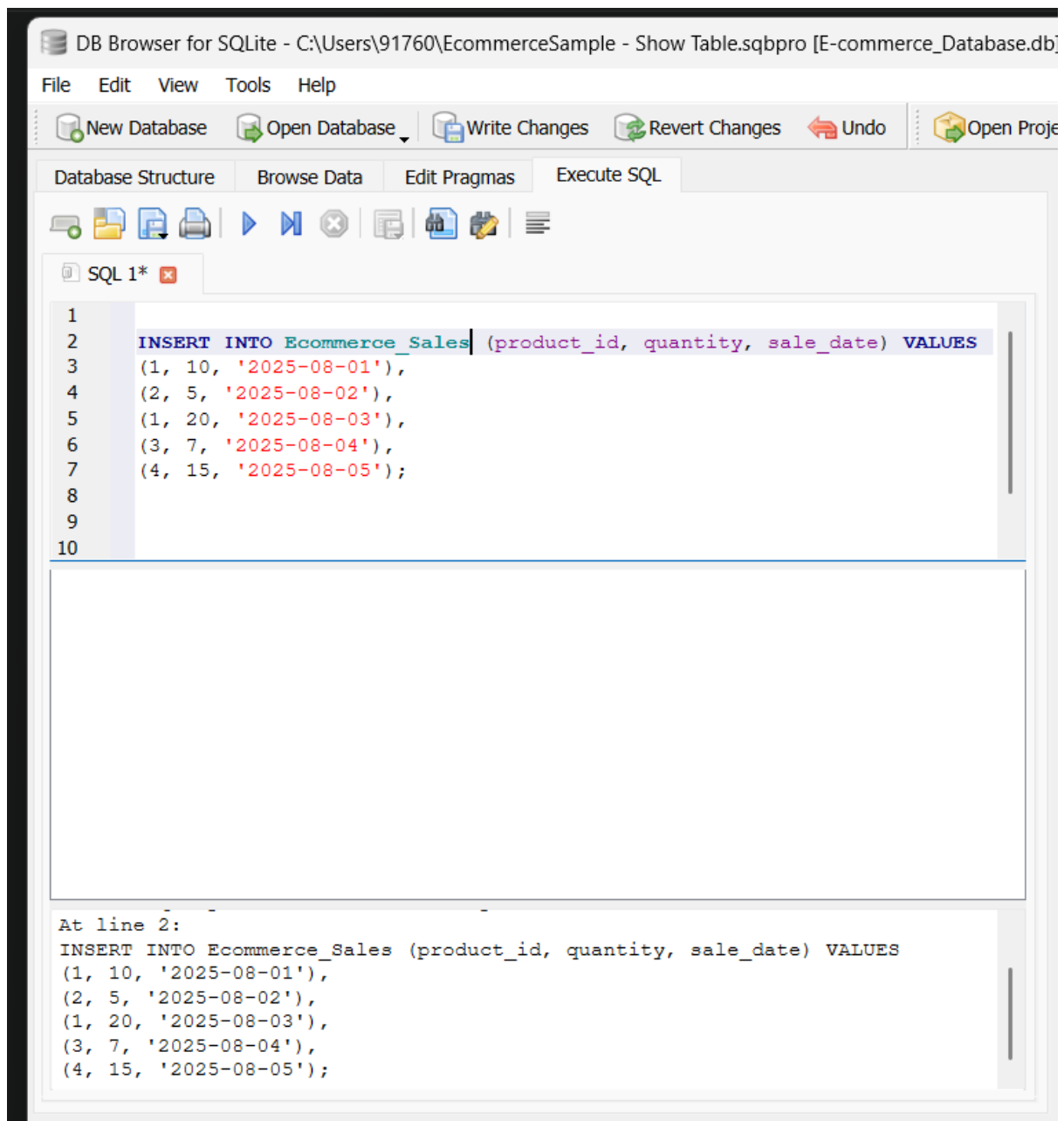
Below the query editor, the execution results are displayed:

```
Execution finished without errors.  
Result: query executed successfully. Took 1ms, 1 rows affected  
At line 2:  
DELETE FROM Ecommerce_Sample  
WHERE name = 'USB Keyboard';
```

On the right side of the interface, there are several panels. The 'Mode' panel shows 'NULL'. The 'No cell Type:' panel shows 'Plot'. The 'Colour' panel shows a color bar. The 'Line ty' panel shows 'SQL'.

Queries I Practiced

13. Insert Sample Data into sales



Queries I Practiced

14. Join sales and products to Analyze Sales Data

DB Browser for SQLite - C:\Users\91760\EcommerceSample - Show Table.sqbpro [E-commerce_Database.db]

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Project

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```
1
2 SELECT p.name, s.quantity, s.sale_date
3 FROM Ecommerce_Sales s
4 JOIN Ecommerce_Sample p ON s.product_id = p.id;
5
6
```

	Name	Quantity	Sale_date
1	wireless Mouse	10	2025-08-01
2	wireless Mouse	20	2025-08-03
3	27'' Monitor	7	2025-08-04
4	External SSD 1TB	15	2025-08-05

Execution finished without errors.
Result: 4 rows returned in 21ms
At line 2:
SELECT p.name, s.quantity, s.sale_date
FROM Ecommerce_Sales s
JOIN Ecommerce_Sample p ON s.product_id = p.id;

Queries I Practiced

15. See total quantity sold for each product

DB Browser for SQLite - C:\Users\91760\EcommerceSample - Show Table.sqbpro [E-commerce_Database.dl

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Undo Open Pro

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

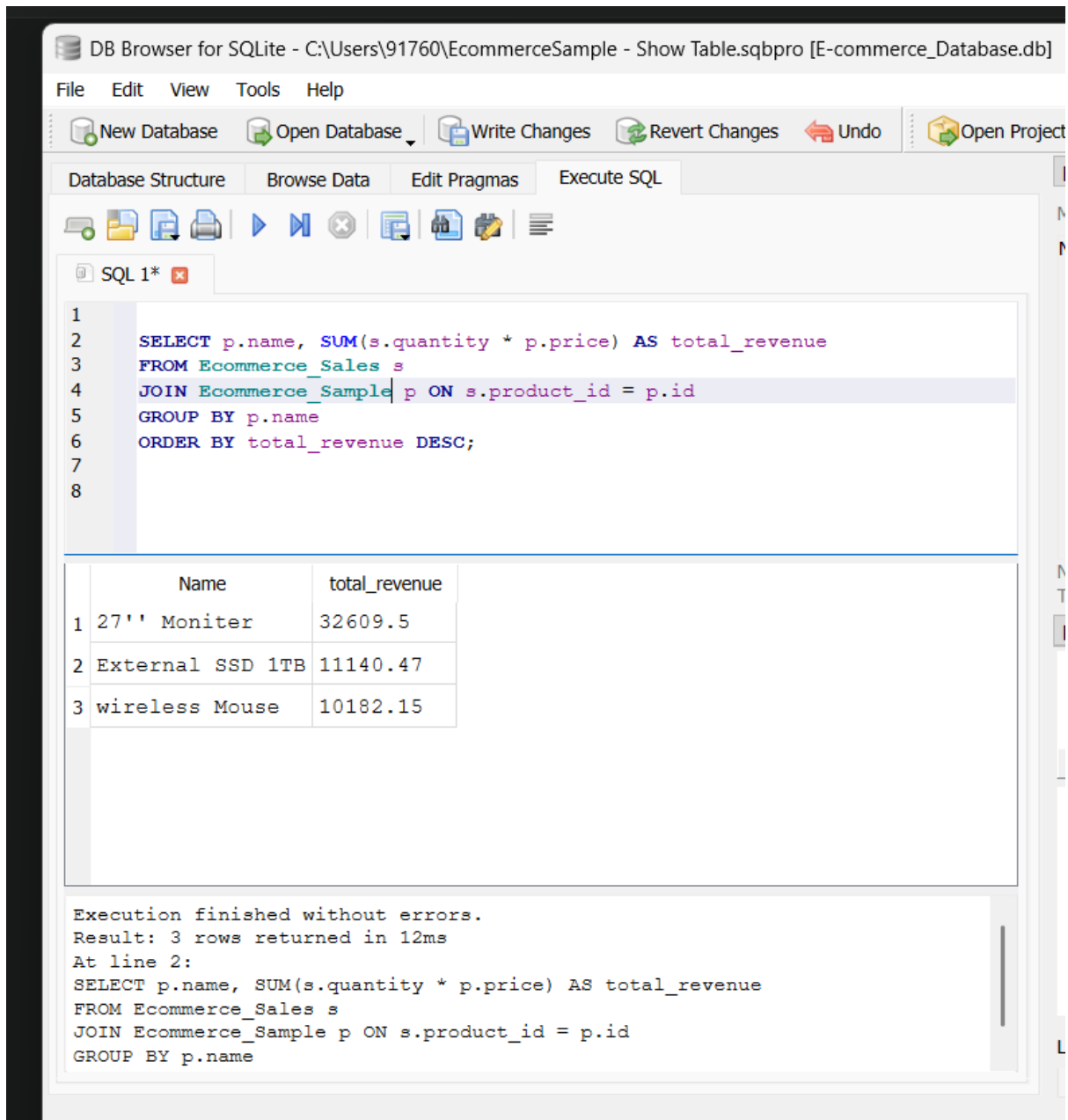
```
1
2 SELECT p.name, SUM(s.quantity) AS total_quantity_sold
3 FROM Ecommerce_Sales s
4 JOIN Ecommerce_Sample p ON s.product_id = p.id
5 GROUP BY p.name;
6
7
```

	Name	total_quantity_sold
1	27'' Moniter	7
2	External SSD 1TB	15
3	wireless Mouse	30

Execution finished without errors.
Result: 3 rows returned in 14ms
At line 2:
SELECT p.name, SUM(s.quantity) AS total_quantity_sold
FROM Ecommerce_Sales s
JOIN Ecommerce_Sample p ON s.product_id = p.id
GROUP BY p.name;

Queries I Practiced

16. Calculate total revenue for each product



The screenshot shows the DB Browser for SQLite interface. The title bar indicates the database is 'E-commerce_Database.db'. The 'Execute SQL' tab is active, displaying the following SQL query:

```
1  
2 SELECT p.name, SUM(s.quantity * p.price) AS total_revenue  
3 FROM Ecommerce_Sales s  
4 JOIN Ecommerce_Sample p ON s.product_id = p.id  
5 GROUP BY p.name  
6 ORDER BY total_revenue DESC;  
7  
8
```

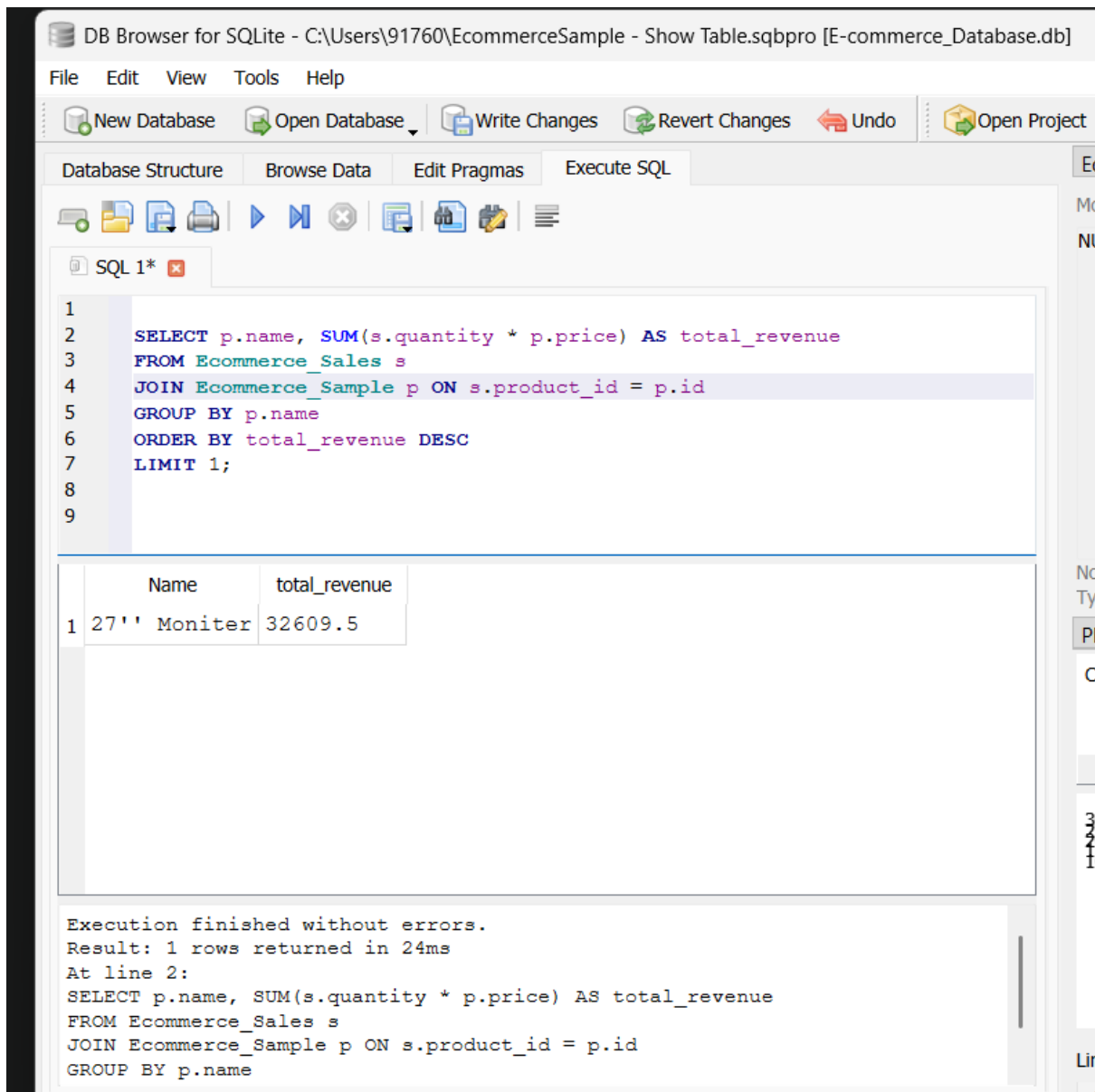
Below the query editor, the results are displayed in a table with 3 rows:

	Name	total_revenue
1	27'' Moniter	32609.5
2	External SSD 1TB	11140.47
3	wireless Mouse	10182.15

At the bottom, the execution status is shown: 'Execution finished without errors. Result: 3 rows returned in 12ms'. The SQL query is repeated in the execution log.

Queries I Practiced

17. Find the best-selling product (by total revenue)



The screenshot shows the DB Browser for SQLite application. The title bar indicates the database is 'C:\Users\91760\EcommerceSample - Show Table.sqbpro [E-commerce_Database.db]'. The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains icons for New Database, Open Database, Write Changes, Revert Changes, Undo, and Open Project. The main window has tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. The Execute SQL tab is active, showing a query in the SQL editor. Below the editor, the results are displayed in a table with two columns: Name and total_revenue. The table contains one row: '27'' Monitor' with a total_revenue of 32609.5. At the bottom, a status bar indicates 'Execution finished without errors. Result: 1 rows returned in 24ms' and shows the SQL query that was executed.

```
1
2 SELECT p.name, SUM(s.quantity * p.price) AS total_revenue
3 FROM Ecommerce_Sales s
4 JOIN Ecommerce_Sample p ON s.product_id = p.id
5 GROUP BY p.name
6 ORDER BY total_revenue DESC
7 LIMIT 1;
8
9
```

	Name	total_revenue
1	27'' Monitor	32609.5

Execution finished without errors.
Result: 1 rows returned in 24ms
At line 2:
SELECT p.name, SUM(s.quantity * p.price) AS total_revenue
FROM Ecommerce_Sales s
JOIN Ecommerce_Sample p ON s.product_id = p.id
GROUP BY p.name