Task 2 SQL

Database creation

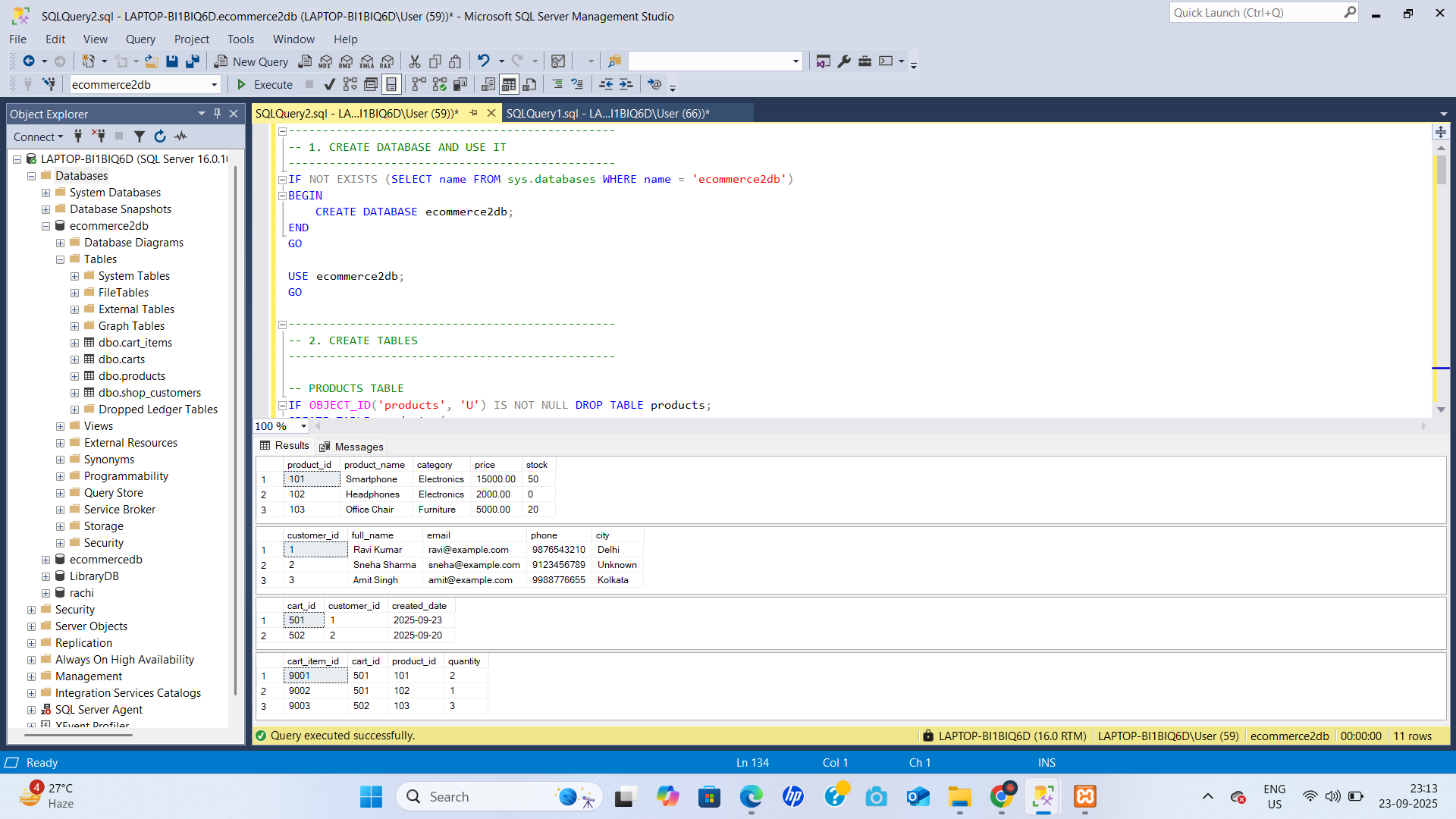
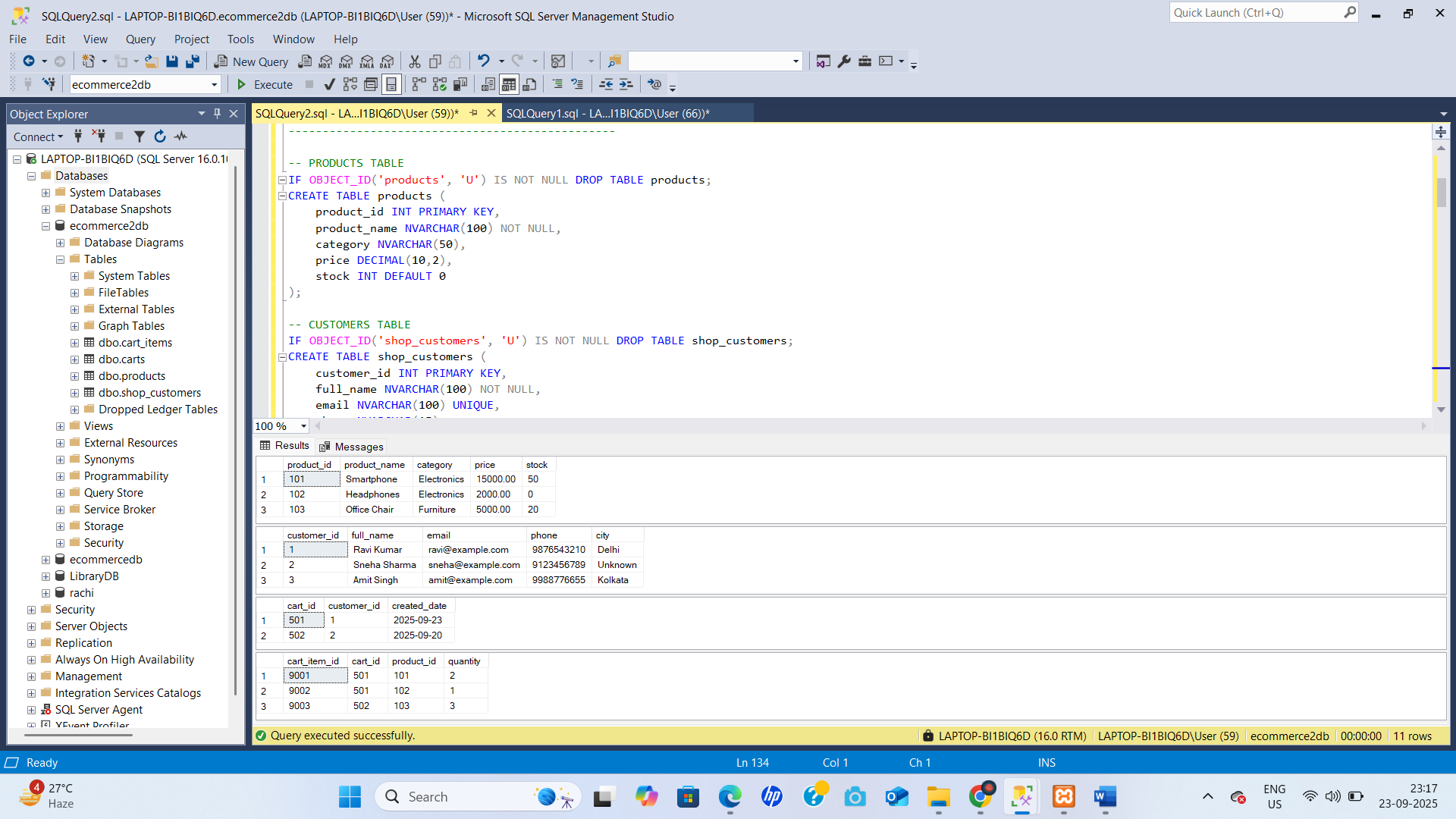
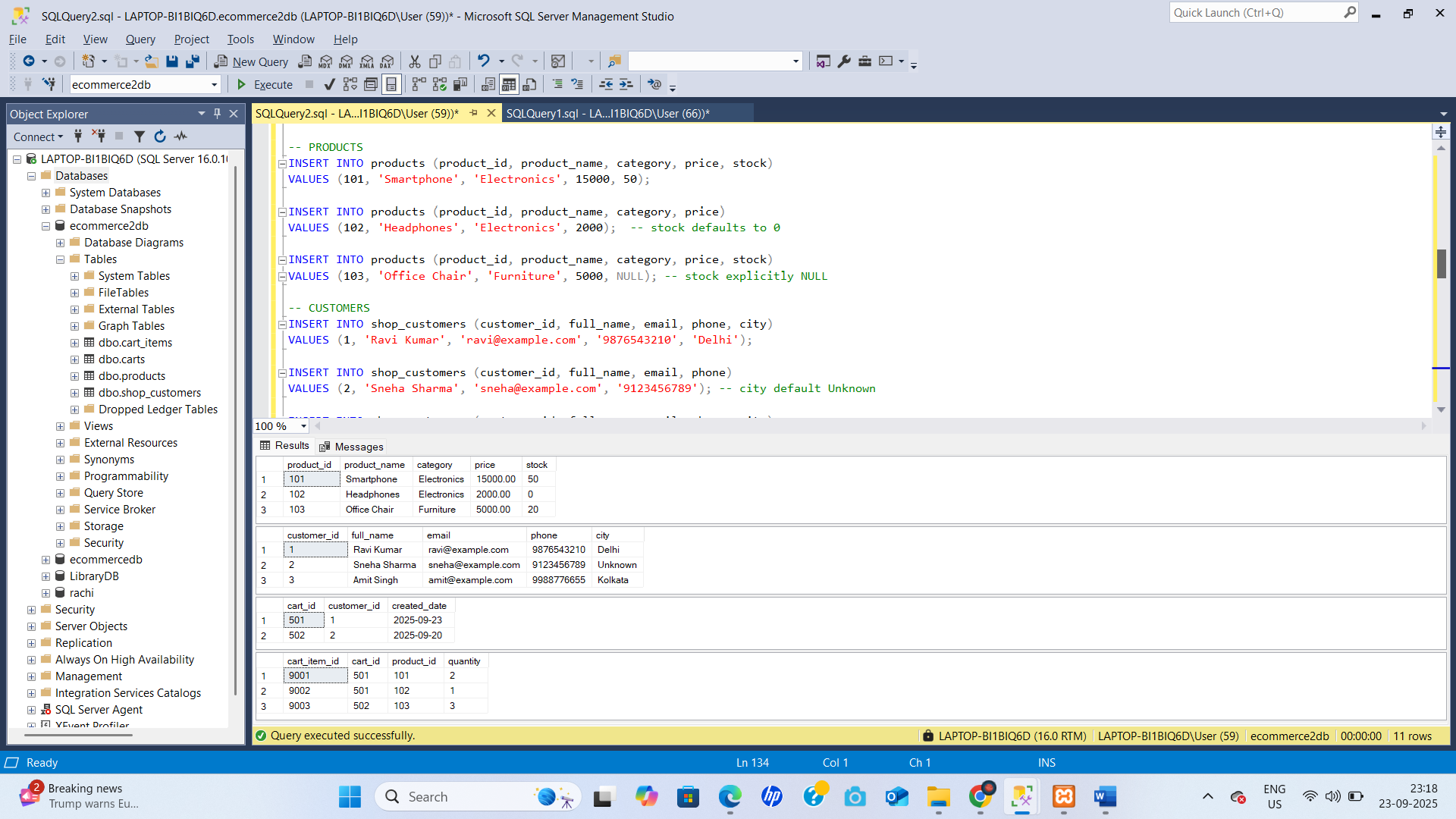


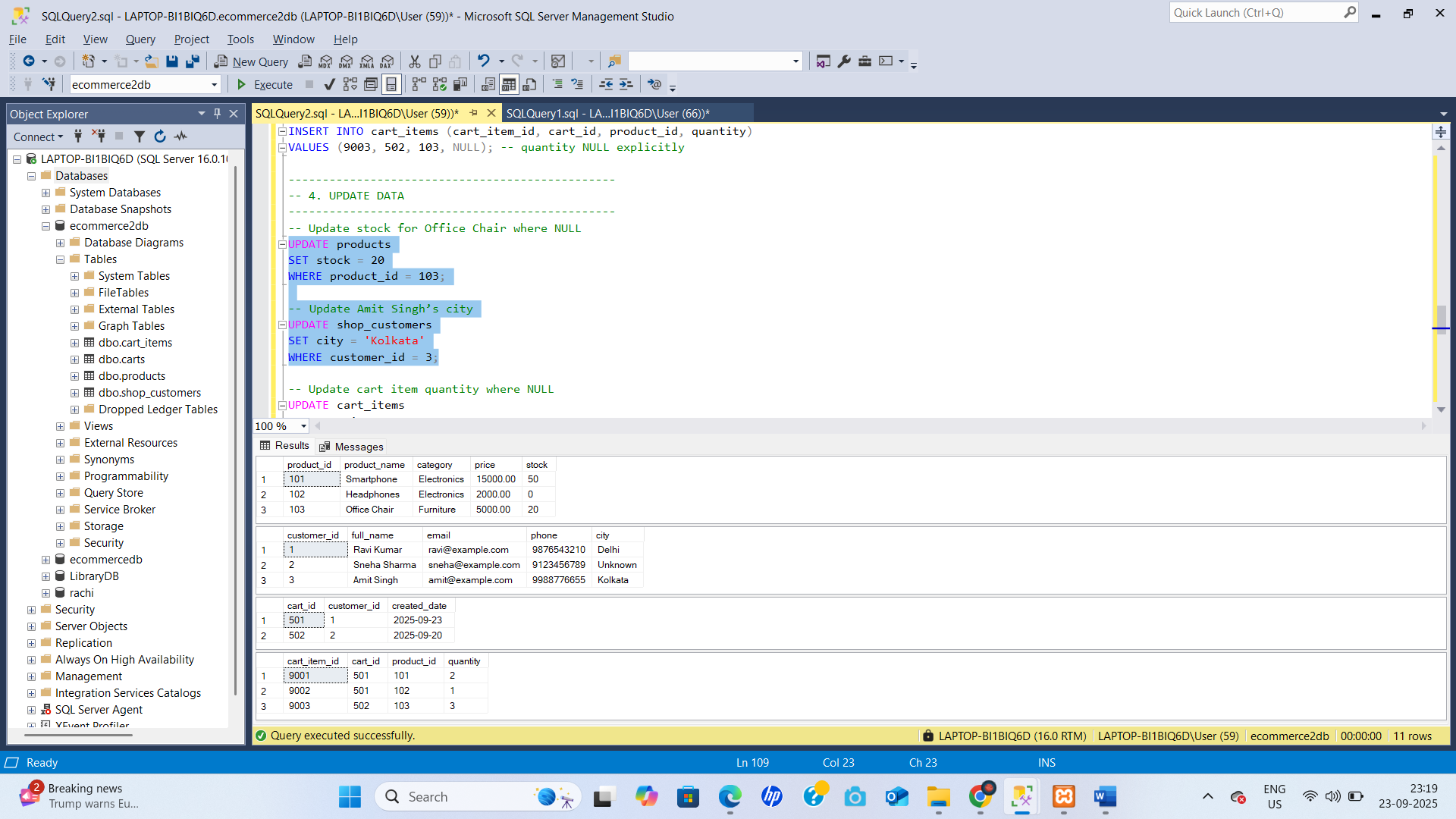
Table Creation

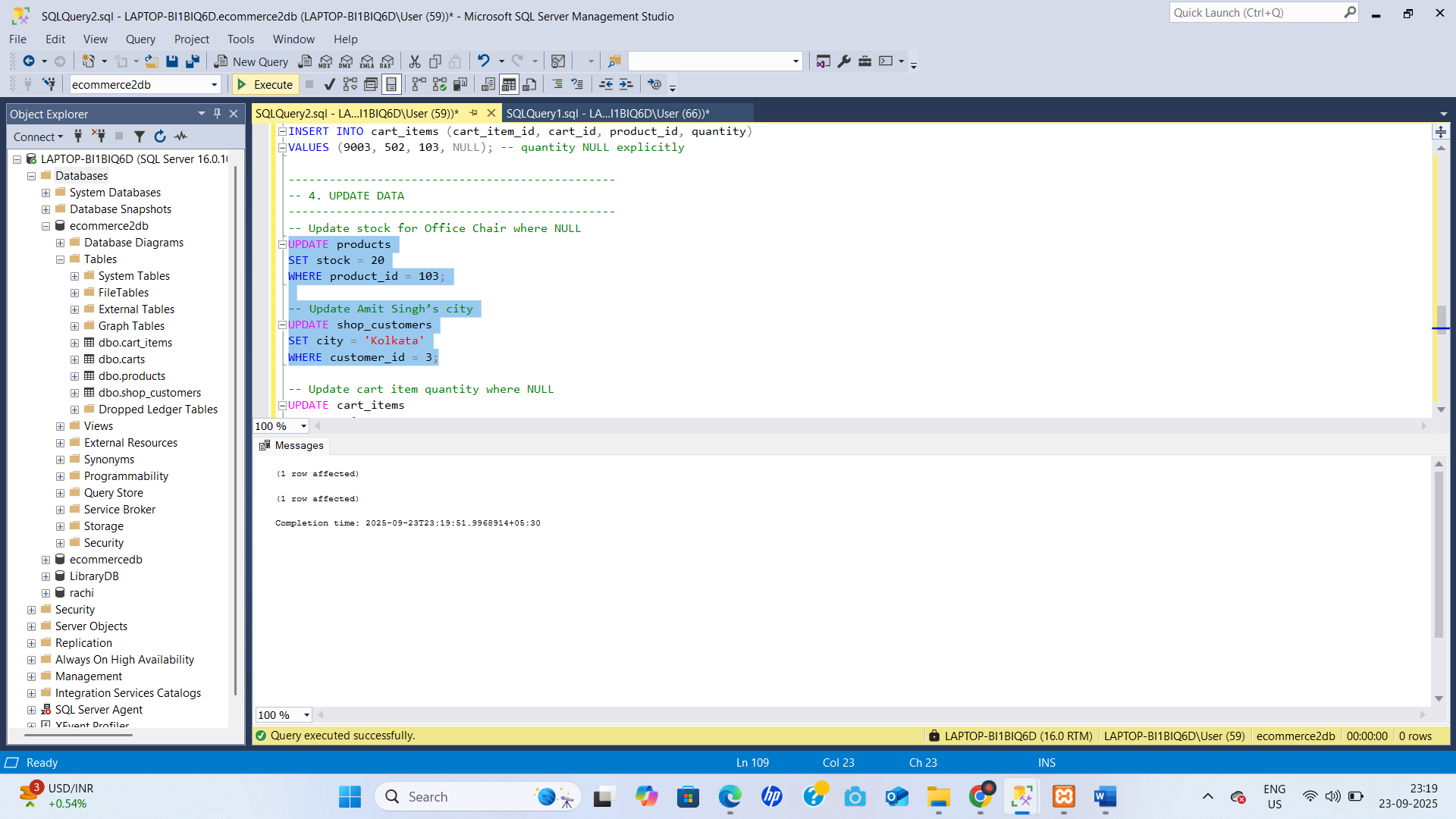


Data insert

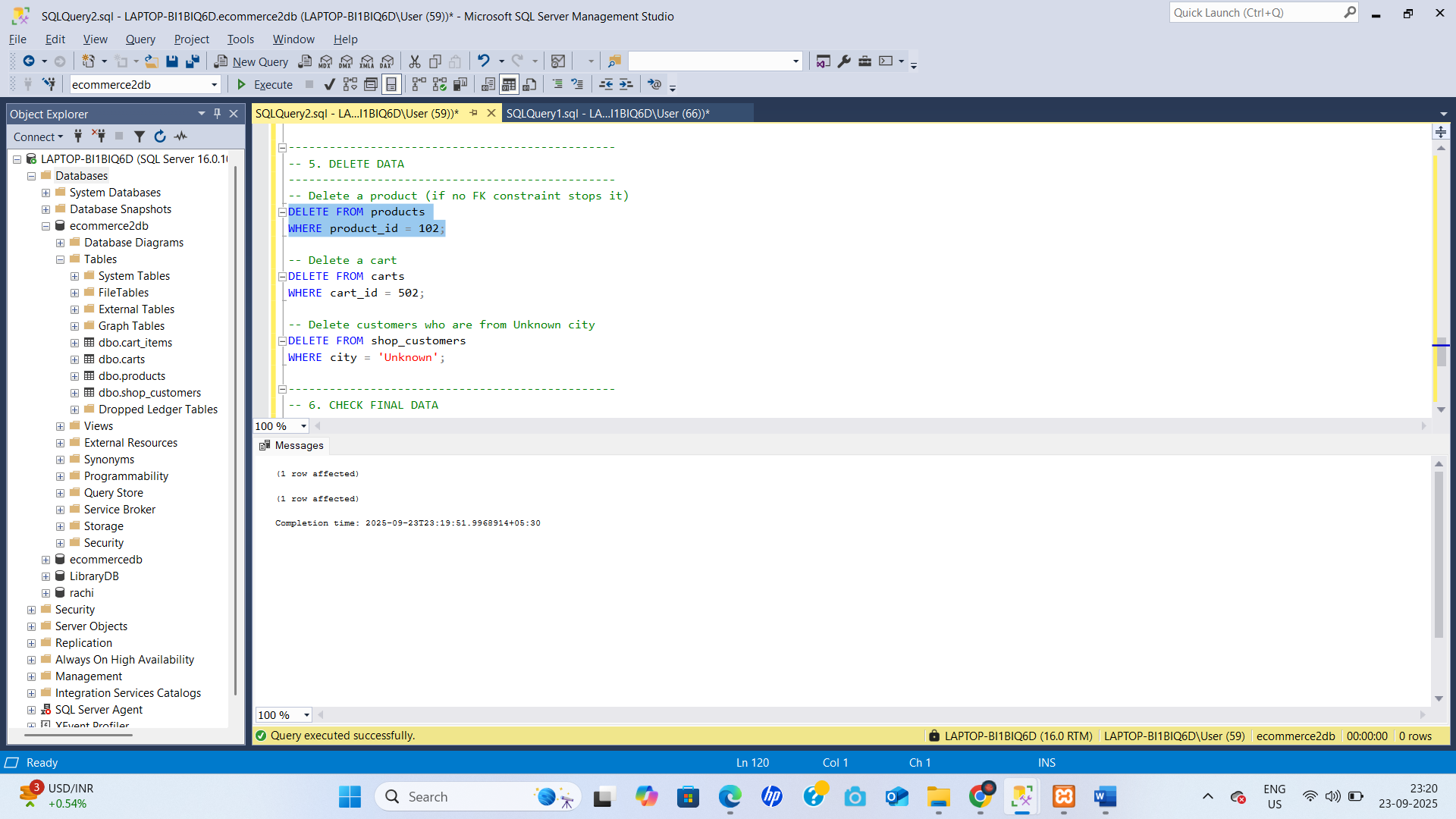


Update Data

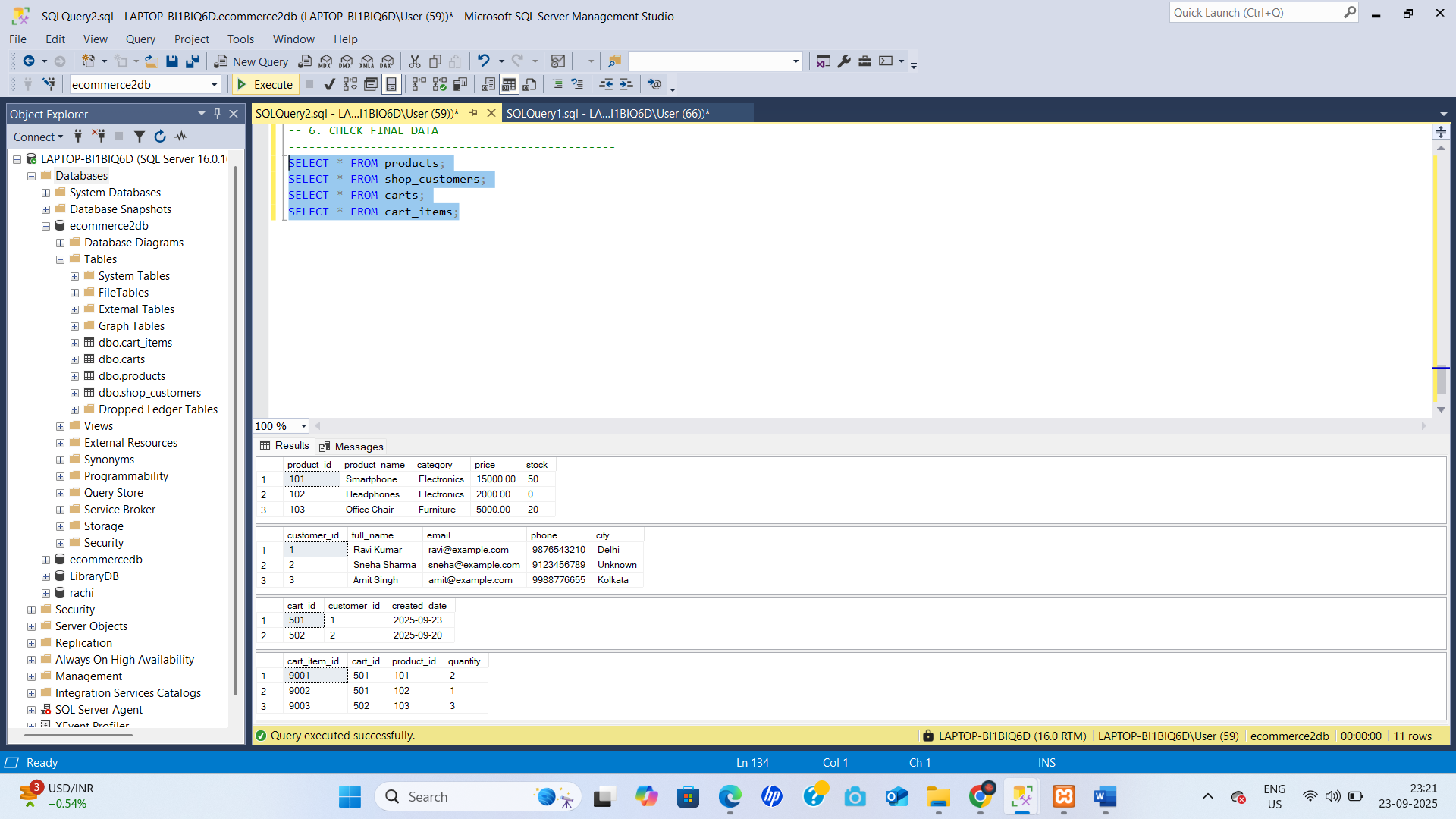




Delete product



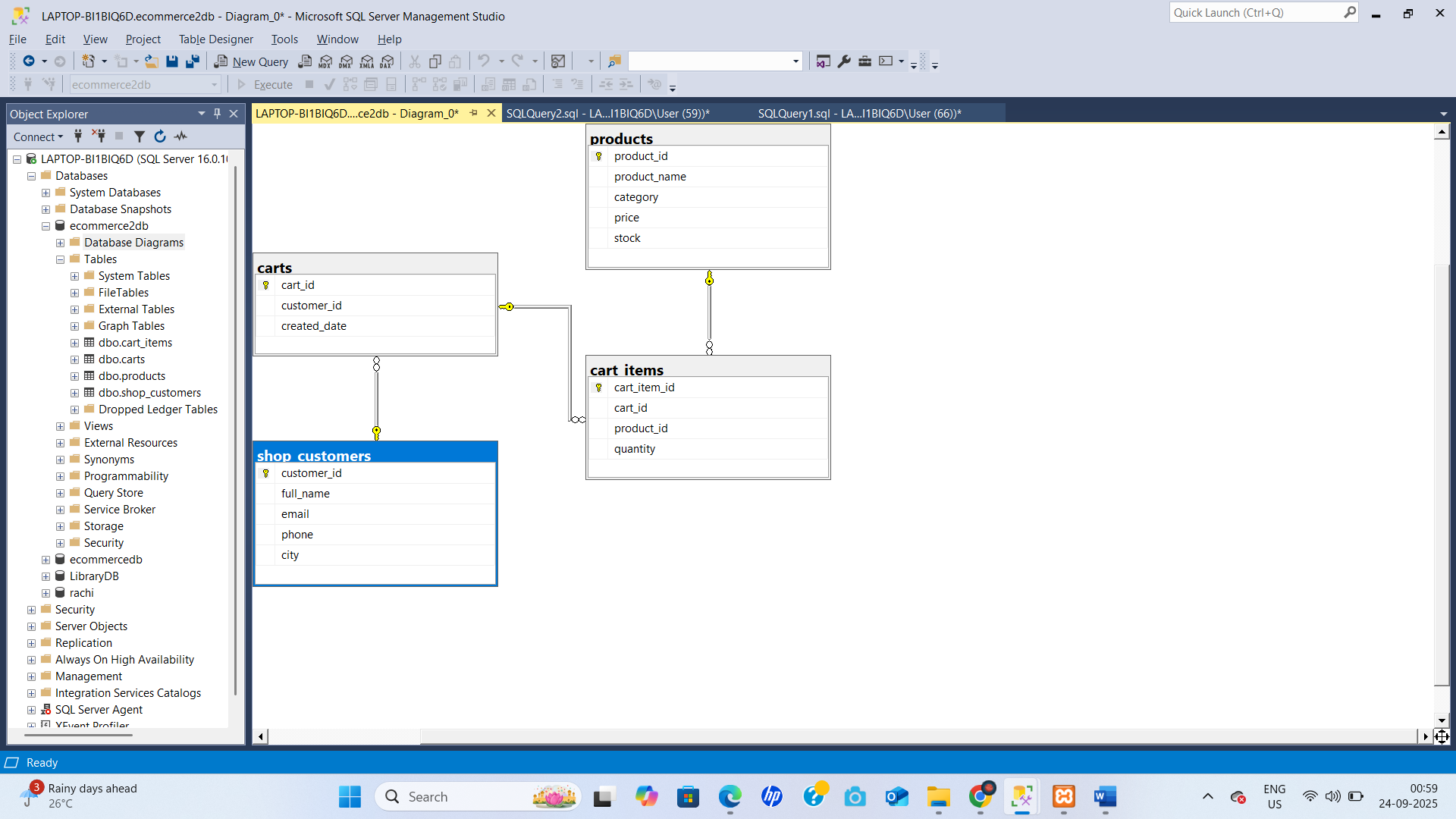
Select Data



 **DML (Data Manipulation Language):** INSERT, UPDATE, DELETE.

 **Null Handling:** Use NULL, DEFAULT, IS NULL, and ON DELETE CASCADE carefully.

 **Transactions:** Use BEGIN TRANSACTION and ROLLBACK to undo changes.



**1️⃣ Difference between NULL and 0**

| **Feature** | **NULL** | **0** |
| --- | --- | --- |
| Meaning | Unknown / missing / no value | Actual numeric value 0 |
| Storage | No actual data stored | Numeric zero stored |
| Comparison | NULL <> 0 (not equal) | Can compare normally |
| Example | Age column left blank → NULL | Age column entered as 0 |

**Key point:** NULL ≠ 0.

**2️⃣ What is a Default Constraint?**

A **Default Constraint** automatically inserts a predefined value into a column when **no value is provided** during an INSERT.

Example:

CREATE TABLE payments (

payment\_id INT PRIMARY KEY,

method NVARCHAR(20) DEFAULT 'UPI'

);

INSERT INTO payments (payment\_id) VALUES (1);

-- method becomes 'UPI'

**3️⃣ How does IS NULL work?**

IS NULL is used in SQL to **check for missing/NULL values**.  
Example:

SELECT \* FROM customers WHERE city IS NULL;

This returns rows where city has no value.

**4️⃣ How do you update multiple rows?**

Use UPDATE with a WHERE condition that matches multiple rows.  
Example:

UPDATE orders

SET status = 'Shipped'

WHERE order\_date < '2025-09-01';

All rows matching the condition are updated at once.

**5️⃣ Can we insert partial values?**

Yes — if you **specify the column list** in the INSERT statement and the remaining columns have DEFAULT or allow NULL.

Example:

INSERT INTO customers (customer\_id, name)

VALUES (5, 'New Customer');

Columns not listed get NULL or default values automatically.

**6️⃣ What happens if a NOT NULL field is left empty?**

* SQL Server **throws an error** and the insert/update fails.
* You **must provide a value** for all NOT NULL columns or use a default.

Example:

CREATE TABLE test (

id INT NOT NULL

);

INSERT INTO test DEFAULT VALUES; -- ❌ Error

**7️⃣ How do you rollback a deletion?**

* If inside a **transaction**, you can use ROLLBACK TRANSACTION to undo the delete.
* If you’ve already committed, you can’t rollback — you must restore from a backup.

Example:

BEGIN TRANSACTION;

DELETE FROM customers WHERE customer\_id = 5;

ROLLBACK TRANSACTION; -- Undo the delete

**8️⃣ Can we insert values into specific columns only?**

Yes — specify the column names in the INSERT statement.

Example:

INSERT INTO orders (order\_id, customer\_id)

VALUES (201, 1);

Other columns get defaults or NULL.

**9️⃣ How to insert values using SELECT?**

Use INSERT … SELECT to insert rows from one table into another.

Example:

INSERT INTO orders\_archive (order\_id, customer\_id, order\_date, status)

SELECT order\_id, customer\_id, order\_date, status

FROM orders

WHERE status='Shipped';

**🔟 What is ON DELETE CASCADE?**

It’s a **referential action** on a foreign key.  
When a **parent row is deleted**, all **child rows** in the referencing table are automatically deleted.

Example:

CREATE TABLE orders (

order\_id INT PRIMARY KEY

);

CREATE TABLE orderitems (

item\_id INT PRIMARY KEY,

order\_id INT,

FOREIGN KEY (order\_id) REFERENCES orders(order\_id) ON DELETE CASCADE

);

If you delete a row from orders, all related orderitems rows will be deleted automatically.

**🔹 Key Concepts Recap**

* **DML (Data Manipulation Language):** INSERT, UPDATE, DELETE.
* **Null Handling:** Use NULL, DEFAULT, IS NULL, and ON DELETE CASCADE carefully.
* **Transactions:** Use BEGIN TRANSACTION and ROLLBACK to undo changes.

**1️⃣ Difference between NULL and 0**

| **NULL** | **0** |
| --- | --- |
| Represents **no value / unknown / missing** | Actual numeric zero |
| Takes no storage for the value itself | Stores a number 0 |
| Cannot be compared with = directly (need IS NULL) | Can be compared with normal operators |

Example:

SELECT \* FROM Orders WHERE Amount IS NULL; -- checks missing values

SELECT \* FROM Orders WHERE Amount = 0; -- checks zero values

**2️⃣ What is a Default Constraint?**

A **default constraint** automatically inserts a value when none is provided.

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY,

Method NVARCHAR(20) DEFAULT 'UPI'

);

INSERT INTO Payments (PaymentID) VALUES (1);

-- Method automatically becomes 'UPI'

**3️⃣ How does IS NULL work?**

Used to test for NULL values.

SELECT \* FROM Customers WHERE City IS NULL;

Without IS NULL, = or <> won’t detect NULL.

**4️⃣ How do you update multiple rows?**

Use one UPDATE with a WHERE clause matching several rows.

UPDATE Orders

SET Status='Shipped'

WHERE OrderDate<'2025-09-01';

All matching rows get updated.

**5️⃣ Can we insert partial values?**

Yes, if you specify columns. Remaining columns must allow NULL or have defaults.

INSERT INTO Customers (CustomerID, Name)

VALUES (10, 'Amit'); -- other columns become NULL/default

**6️⃣ What happens if a NOT NULL field is left empty?**

SQL Server returns an **error** — you must supply a value or default.

CREATE TABLE Test (ID INT NOT NULL);

INSERT INTO Test DEFAULT VALUES; -- ❌ Error (no value)

**7️⃣ How do you rollback a deletion?**

If the delete is inside a transaction:

BEGIN TRANSACTION;

DELETE FROM Customers WHERE CustomerID=5;

ROLLBACK TRANSACTION; -- undo delete

If you’ve already committed, you need a backup to restore.

**8️⃣ Can we insert values into specific columns only?**

Yes, list the columns explicitly:

INSERT INTO Orders (OrderID, CustomerID)

VALUES (201, 1);

Other columns get defaults or NULL.

**9️⃣ How to insert values using SELECT?**

Use INSERT … SELECT to copy from another table:

INSERT INTO Orders\_Archive (OrderID, CustomerID, Status)

SELECT OrderID, CustomerID, Status

FROM Orders

WHERE Status='Shipped';

**🔟 What is ON DELETE CASCADE?**

It’s a foreign key option. If you delete a row in the **parent** table, related **child** rows are automatically deleted.

CREATE TABLE Orders (

OrderID INT PRIMARY KEY

);

CREATE TABLE OrderItems (

ItemID INT PRIMARY KEY,

OrderID INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE

);

Deleting from Orders also deletes matching OrderItems.