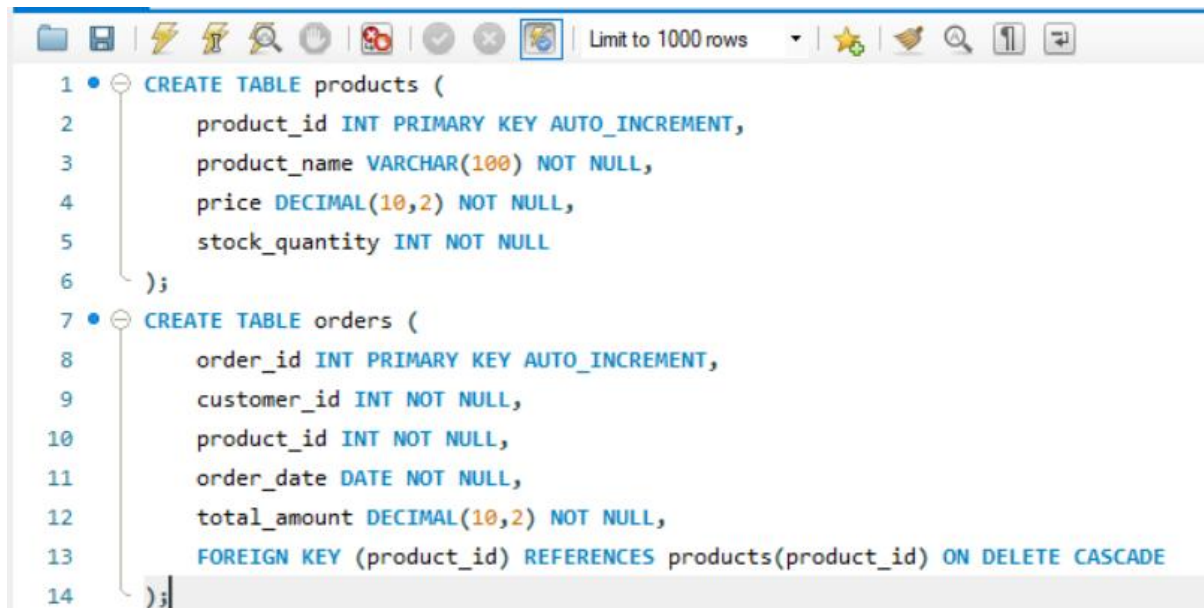


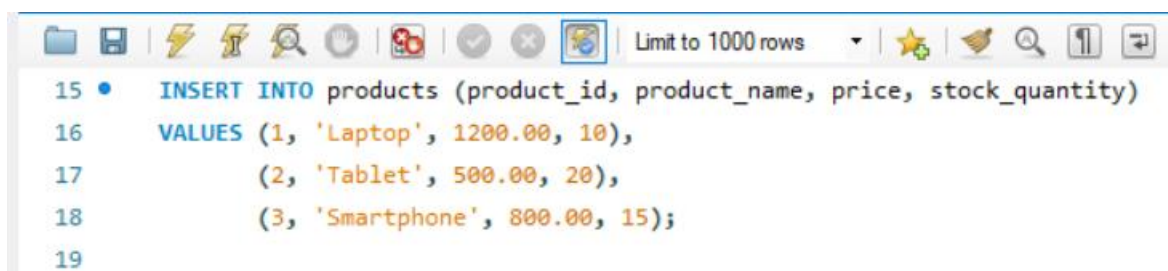
**Assignment 5:** Begin a transaction, perform a series of INSERTs into 'orders', setting a SAVEPOINT after each, rollback to the second SAVEPOINT, and COMMIT the overall transaction.

**Step 1 :** Create the products and orders tables.




```
1 • CREATE TABLE products (  
2     product_id INT PRIMARY KEY AUTO_INCREMENT,  
3     product_name VARCHAR(100) NOT NULL,  
4     price DECIMAL(10,2) NOT NULL,  
5     stock_quantity INT NOT NULL  
6 );  
7 • CREATE TABLE orders (  
8     order_id INT PRIMARY KEY AUTO_INCREMENT,  
9     customer_id INT NOT NULL,  
10    product_id INT NOT NULL,  
11    order_date DATE NOT NULL,  
12    total_amount DECIMAL(10,2) NOT NULL,  
13    FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE CASCADE  
14 );
```

**Step 2 :** Insert data to products table.



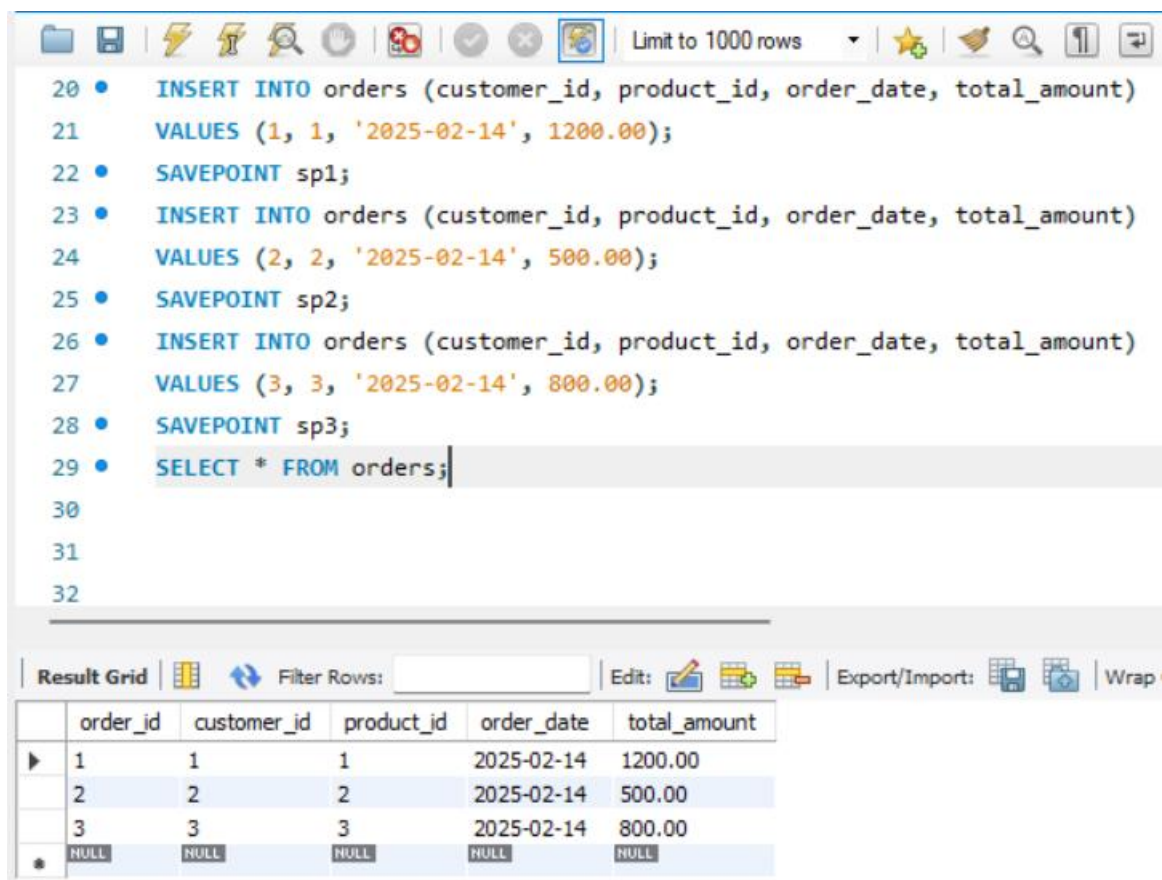
```
15 • INSERT INTO products (product_id, product_name, price, stock_quantity)  
16 VALUES (1, 'Laptop', 1200.00, 10),  
17          (2, 'Tablet', 500.00, 20),  
18          (3, 'Smartphone', 800.00, 15);  
19
```

**Step 3 :** Start a Transaction.



```
19 • START TRANSACTION;
```

#### Step 4 : Insert Orders with SAVEPOINTS & Verify insertion.



```
20 • INSERT INTO orders (customer_id, product_id, order_date, total_amount)
21   VALUES (1, 1, '2025-02-14', 1200.00);
22 • SAVEPOINT sp1;
23 • INSERT INTO orders (customer_id, product_id, order_date, total_amount)
24   VALUES (2, 2, '2025-02-14', 500.00);
25 • SAVEPOINT sp2;
26 • INSERT INTO orders (customer_id, product_id, order_date, total_amount)
27   VALUES (3, 3, '2025-02-14', 800.00);
28 • SAVEPOINT sp3;
29 • SELECT * FROM orders;
```

Result Grid

	order_id	customer_id	product_id	order_date	total_amount
▶	1	1	1	2025-02-14	1200.00
	2	2	2	2025-02-14	500.00
	3	3	3	2025-02-14	800.00
*	NULL	NULL	NULL	NULL	NULL

#### Step 5 : Rollback to the Second SAVEPOINT & Check Orders After ROLLBACK.

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query window contains the following SQL code:

```
30 • ROLLBACK TO SAVEPOINT sp2;  
31 • SELECT * FROM orders;  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42
```

Below the query window is the 'Result Grid' section. It includes a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Wrap Cell Content'. The result grid displays the following data:

	order_id	customer_id	product_id	order_date	total_amount
▶	1	1	1	2025-02-14	1200.00
	2	2	2	2025-02-14	500.00
*	NULL	NULL	NULL	NULL	NULL

**Step 6 :** Commit the Transaction.

The screenshot shows a SQL editor with the following query:

```

32 • COMMIT;
33 • SELECT * FROM orders;
34
35
36
37
38
39
40
41
42
43
44

```

The results are displayed in a table below the editor:

	order_id	customer_id	product_id	order_date	total_amount
▶	1	1	1	2025-02-14	1200.00
	2	2	2	2025-02-14	500.00
*	NULL	NULL	NULL	NULL	NULL