Assignment 4: Compose SQL statements to BEGIN a transaction, INSERT a new record into the 'orders' table, COMMIT the transaction, then UPDATE the 'products' table, and ROLLBACK the transaction.

SOLUTION:

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(10, 2)
);

INSERT INTO orders (order_id, customer_id, order_date, total_amount) VALUES
(101, 1, '2024-05-01', 100.00),
(102, 3, '2024-05-03', 150.00),
(103, 2, '2024-05-05', 200.00),
(104, 2, '2024-05-10', 180.00),
(105, 4, '2024-05-12', 120.00);

CREATE TABLE products (
```

```
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    stock_quantity INT
);

INSERT INTO products (product_id, product_name,
stock_quantity) VALUES
(123, 'Product A', 50),
(124, 'Product B', 100),
(125, 'Product C', 75);

START TRANSACTION;

INSERT INTO orders (order_id, customer_id, order_date,
total_amount)
VALUES (106, 1, '2024-05-15', 180.00);

COMMIT;

UPDATE products
```

```
SET stock_quantity = stock_quantity - 1

WHERE product_id = 123;


ROLLBACK;
```

OUTPUT:

```
1 • SELECT * FROM assgn11.orders;
```

Result Grid | Filter Rows: | Edit:

order_id	customer_id	order_date	total_amount
101	1	2024-05-01	100.00
102	3	2024-05-03	150.00
103	2	2024-05-05	200.00
104	2	2024-05-10	180.00
105	4	2024-05-12	120.00
106	1	2024-05-15	180.00
NULL	NULL	NULL	NULL

```
1 • SELECT * FROM assgn11.products;
```

Result Grid | Filter Rows: | Edit:

product_id	product_name	stock_quantity
123	Product A	49
124	Product B	100
125	Product C	75
NULL	NULL	NULL