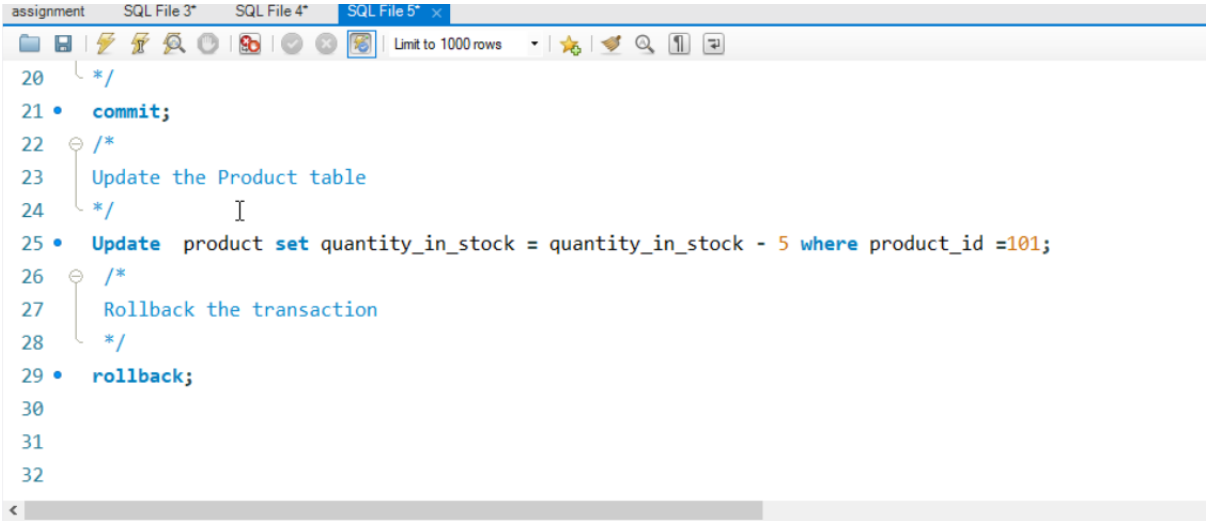
Assignment 4:

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.

```
/*Begin the transaction*/
BEGIN TRANSACTION;

/* Insert a new record into the 'orders' table*/
INSERT INTO orders (order_id, product_id, quantity, order_date)
VALUES (1, 101, 5, '2024-05-10');

/* Commit the transaction */
COMMIT;

/* Update the 'products' table */
UPDATE products
SET quantity_in_stock = quantity_in_stock - 5
WHERE product_id = 101;

/*  Rollback the transaction (if needed) */
ROLLBACK;
```

```sql
set autocommit=0;
select * from orders;
/*
Insert new record in order table
*/
insert into orders(order_id, product_id, quantity, order_date)
values(1, 101, 5, '2024-5-10');
/*
Commit the transactions
*/
commit;
/*
```



In this example:

- We begin the transaction using `BEGIN TRANSACTION;`.
- We insert a new record into the 'orders' table with an order ID of 1, product ID of 101, quantity of 5, and the current date as the order date.
- We commit the transaction using `COMMIT;`, which saves the changes made in the transaction to the database.
- We then update the 'products' table to deduct 5 units from the quantity in stock for the product with ID 101.
- Finally, we use `ROLLBACK;` to undo the changes made in the transaction if needed. In this example, it's placed after the commit and won't have any effect since the transaction has already been committed. However, if an error occurred before the commit, the rollback would undo the changes made by the transaction.