**Assignment 7 : 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.**

The SQL statements demonstrating a transaction with both successful commit and rollback scenarios:

**Scenario 1: Successful Transaction (Commit)**

SQL

BEGIN TRANSACTION;

INSERT INTO orders (customer\_id, order\_date, total\_amount)

VALUES (123, CURDATE(), 100.00);

COMMIT TRANSACTION;

**Explanation:**

1. BEGIN TRANSACTION: This marks the beginning of a transaction. All subsequent SQL statements within this block will be treated as part of the same transaction.
2. INSERT INTO orders ...: This statement inserts a new record into the orders table with specified values for customer\_id, order\_date (current date using CURDATE()), and total\_amount.
3. COMMIT TRANSACTION: This statement finalizes the transaction and makes the changes to the database permanent.

**Scenario 2: Transaction Rollback**

SQL

BEGIN TRANSACTION;

INSERT INTO orders (customer\_id, order\_date, total\_amount)

VALUES (123, CURDATE(), 100.00);

UPDATE products SET stock\_level = stock\_level - 5 -- Assuming product ID is known

WHERE product\_id = YOUR\_PRODUCT\_ID; -- Replace with actual product ID

ROLLBACK TRANSACTION;

**Explanation:**

1. The initial statements are the same as the successful transaction.
2. UPDATE products ...: This statement attempts to update the stock level of a product (assuming the product ID is known) by subtracting 5 from the existing stock.
3. ROLLBACK TRANSACTION: Since this statement comes after the UPDATE, it undoes all changes made within the transaction. This means the inserted order record is removed, and the product stock level remains unchanged.

**Important Notes:**

* Replace YOUR\_PRODUCT\_ID with the actual product ID you want to update.
* This example demonstrates a simple update. Depending on your database system, additional checks or error handling might be necessary before updating related tables.
* Always ensure proper database connection handling and error handling in your application code.