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.

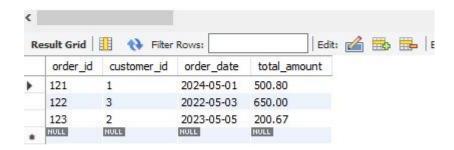
```
SOLUTION:
CREATE TABLE orders(
order id INT PRIMARY KEY,
customer id INT,
order_date DATE,
total_amount DECIMAL(10,2)
);
INSERT INTO
orders(order_id,customer_id,order_date,total_amount)
VALUES
(121,1, '2024-05-01', 500.80),
(122, 3, '2022-05-03', 650.00),
(123, 2, '2023-05-05', 200.67);
```

START TRANSACTION;

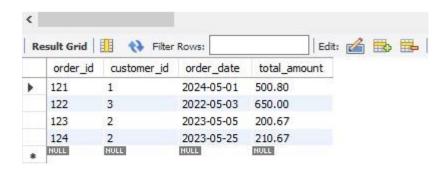
```
INSERT INTO
orders(order_id,customer_id,order_date,total_amount)
VALUES
(124, 2, '2023-05-25', 210.67);
SAVEPOINT first savepoint;
INSERT INTO
orders(order_id,customer_id,order_date,total_amount)
VALUES
(125, 5, '2024-05-22', 810.67);
SAVEPOINT second_savepoint;
INSERT INTO
orders(order_id,customer_id,order_date,total_amount)
VALUES
(126, 5, '2024-05-21', 610.67);
ROLLBACK to second savepoint;
COMMIT;
```

OUTPUT:

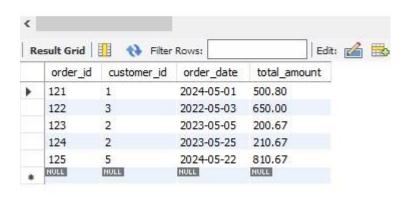
1 • SELECT * FROM assgn12.orders;

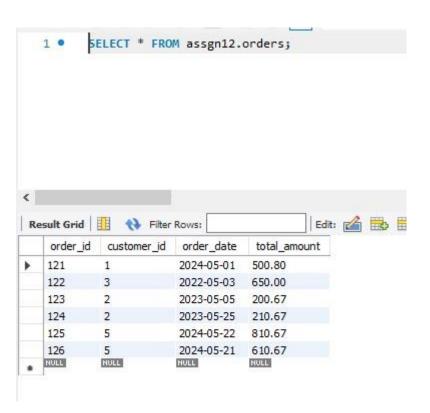


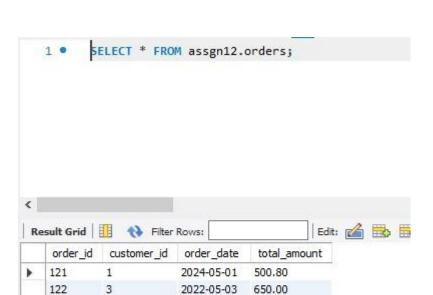
1 • SELECT * FROM assgn12.orders;



1 • SELECT * FROM assgn12.orders;







2023-05-05

2023-05-25

2024-05-22

200.67

210.67

810.67

NULL

123

124

125

NULL

2

2

5