**Level-B Assignment**

**Test cases:**

**Order details Table:**

**A table with numbers and text

AI-generated content may be incorrect.**

**Products Table**

****

**After execution:**

**A table with numbers and text

AI-generated content may be incorrect.**



UnitInStock value decrease from Product Table.



****



**UpdateOrderdetails Procedure:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**GetOrderDetails Procedure:**

**Query:**

EXEC GetOrderDetails 10621;

A table with numbers and letters

AI-generated content may be incorrect.

EXEC GetOrderDetails 99999;

A close-up of a number

AI-generated content may be incorrect.

**DeleteOrderDetails Procedure:**

BEGIN TRANSACTION.

EXEC DeleteOrderDetails 10621, 19

ROLLBACK.

A white background with black text

AI-generated content may be incorrect.

EXEC DeleteOrderDetails 10621, 999;

A black text on a white background

AI-generated content may be incorrect.

**Trigger:**

**Q) If someone cancels an order in Northwind database**, then you want to delete that order from the Orders table. But you will not be able to delete that Order before deleting the records from Order Details table for that order due to referential integrity constraints. **Create an INSTEAD OF DELETE trigger on Orders table** so that if someone tries to delete an Order, that trigger gets fired and that trigger should first delete everything in Order Details table and then delete that order from the Orders table.

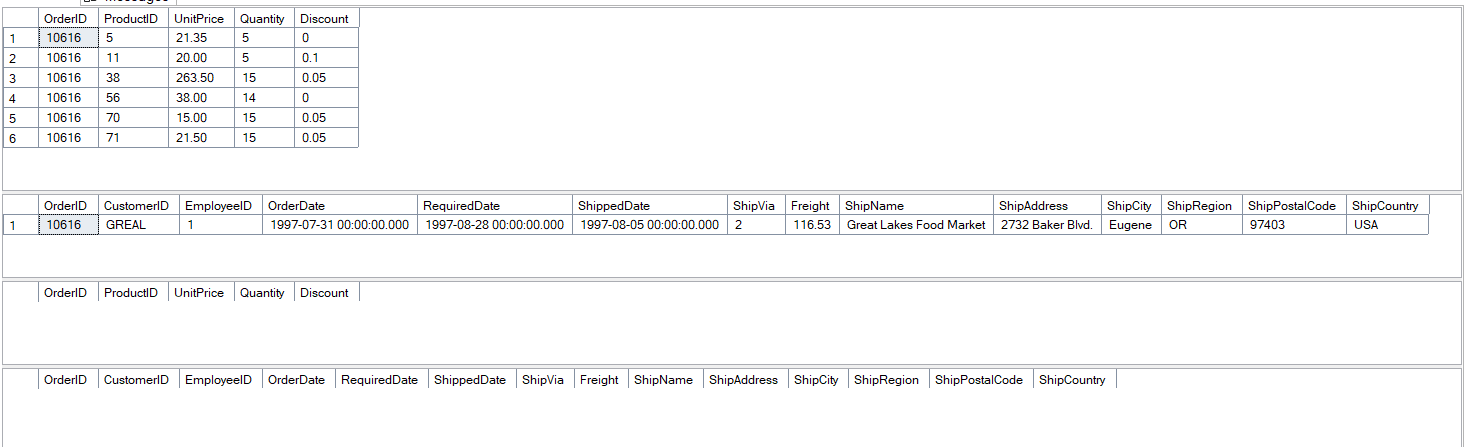
SELECT \* FROM [Order Details] WHERE OrderID = 10616;

SELECT \* FROM Orders WHERE OrderID = 10616

DELETE FROM Orders WHERE OrderID = 10616;

SELECT \* FROM [Order Details] WHERE OrderID = 10616;

SELECT \* FROM Orders WHERE OrderID = 10616;



As you can notice, the instances from both table is deleted for OrderID=10616 as we have performed a trigger which will first delete an everything from Order Details and then delete from Orders Table.

**Q) When an order is placed for X units of product Y**, we must first check the Products table to ensure that there is sufficient stock to fill the order. This trigger will operate on the Order Details table. If sufficient stock exists, then fill the order and decrement X units from the UnitsInStock column in Products. If insufficient stock exists, then refuse the order (*i.e.*, do not insert it) and notify the user that the order could not be filled because of insufficient stock.