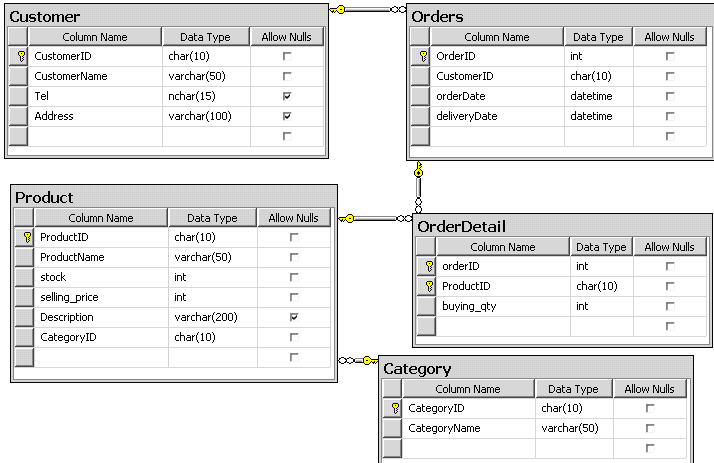
**Triggers**

Consider the SELL\_MANAGE database which has the following relationship:



**Write the SQL statements to:**

1. Create a trigger **Product\_trigger\_update** that update price of item in Product table.

The trigger should check for the value of selling price being updated. If selling price

is updated less than current price, the update operation is not allowed to succeed.

1. Create a trigger named **OrderDetail\_trigger** that will activate whenever a new row

is being inserted to the Order\_Detail. The trigger should check for the value of quantity. If buying quantity is greater than stock quantity, the insert operation will terminate. Otherwise, it will insert a new record into Order\_Detail table and decrease stock column of Product table with corresponding quantity.

1. Create a trigger that will activate whenever a new row is being inserted to the Order table. The maximum limit between OrderDate and deliveryDate that the customer has ordered is 7 days.
2. Implement a cascade delete by using a trigger named **casc\_delete** such that if an item is deleted from the Product table, rows in the corresponding table are also deleted.
3. Display the list of triggers created in the SELL\_MANAGE database using (sys.triggers view).
4. View the definition of trigger with named **casc\_delete**.
5. Create a DDL trigger named **ProhibitDelete** that will activate whenever a user is

trying to delete a table from Test\_Item database. The trigger must not allow a user

to perform deletes and must display a message ***‘You are not allowed to delete tables***

***in this database’***.