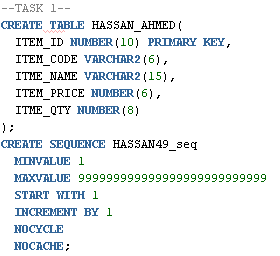
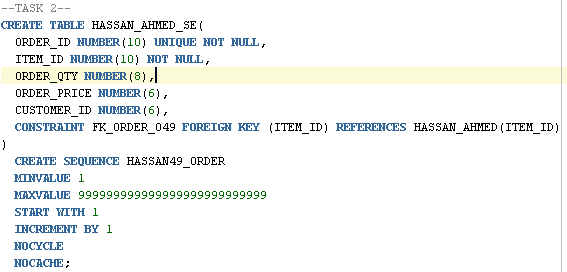
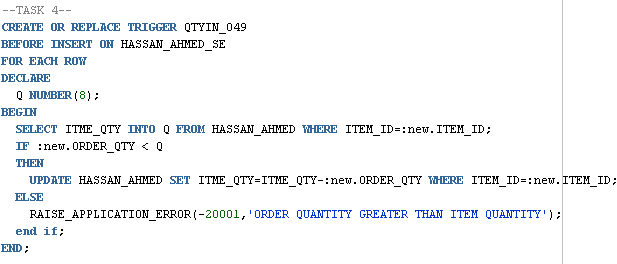
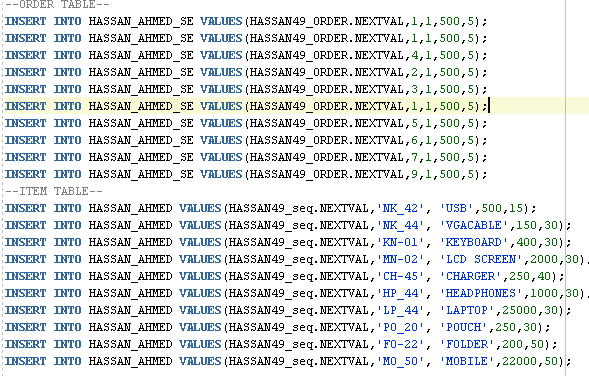
**OPEN ENDED LAB:**

**Scenario:**

****

****

****

**SCENERIO 2:**

**Tables:**

**Highschooler(ID int, name text, grade int);**

**Friend(ID1 int, ID2 int);**

**Likes(ID1 int, ID2 int);**

1 - Write one or more triggers to maintain symmetry in friend relationships. Specifically, if (A,B) is deleted from Friend, then (B,A) should be deleted too. If (A,B) is inserted into Friend then (B,A) should be inserted too. Don't worry about updates to the Friend table

**QUERY:**

**CREATE TRIGGER FRIEND\_INSERT**

**AFTER INSERT ON friend**

**FOR EACH ROW**

**BEGIN**

**INSERT INTO friend (ID1,ID2) VALUES (:new.id2, :new.id1);**

**END;**

**CREATE TRIGGER FRIEND\_DELETE**

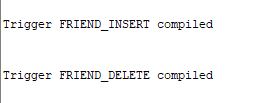
**AFTER DELETE ON friend**

**FOR EACH ROW**

**BEGIN**

**DELETE FROM friend WHERE id1=:old.id2 AND id2=:old.id1;**

**END;**



2 - Write a trigger that automatically deletes students when they graduate, i.e., when their grade is updated to exceed 12. In addition, write a trigger so when a student is moved ahead one grade, then so are all of his or her friends.

**QUERY:**

**CREATE TRIGGER graduate\_trigger**

**AFTER UPDATE on highschooler**

**FOR EACH ROW**

**when (new.grade > 12)**

**BEGIN**

**DELETE FROM highschooler WHERE id=:new.id;**

**END ;**

**CREATE TRIGGER moving\_ahead**

**AFTER UPDATE on highschooler**

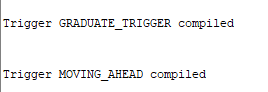
**for EACH ROW**

**WHEN (new.grade = old.grade + 1)**

**BEGIN**

**UPDATE highschooler SET grade = grade+1 WHERE id in (SELECT id2 FROM friend WHERE id1=:new.id);**

**END;**



3 - Write a trigger to enforce the following behavior: If A liked B but is updated to A liking C instead, and B and C were friends, make B and C no longer friends. Don't forget to delete the friendship in both directions, and make sure the trigger only runs when the "liked" (ID2) person is changed but the "liking" (ID1) person is not changed.

CREATE TRIGGER like\_Trigger

AFTER UPDATE ON likes

WHEN (SELECT \* FROM friend WHERE id1=new.id2 AND id2=old.id2

AND new.id1 = old.id1 AND NOT new.id2 = old.id2)

BEGIN

DELETE FROM friend where id1=:new.id2 AND :id2=old.id2);

DELETE FROM friend WHERE id1=:old.id2 AND :id2=new.id2);

END;