

EXERCISE 7

19.10.2020

1. Product (BarCode, PName, Price, QuantityInStock)

Sale (SaleID, DeliveryAddress, CreditCard)

SaleItem (SaleID, BarCode, Quantity)

Create a trigger called updateAvailableQuantity that updates the quantity in stock in the Product table, for every product sold. The trigger should be executed after each insert operation on the SaleItem table: for the product with the given barcode (the one inserted into SaleItem), update the available quantity in Product table to be the old quantity minus the sold quantity.

2. Activities (ActivityID, ActivityName)

Performers (PerformerID, PerformerName, Street, City, State, Zip, ActivityID)

Arenas (ArenaID, ArenaName, City, ArenaCapacity)

Concerts (PerformerID, ArenaID, ConcertDate, TicketPrice)

Create a trigger called deletePerformer that prevents the last performer for a particular activity to be deleted from the database. The trigger should be associated with a delete operation on the Performers table: if the performer to be deleted is the last one for his/her particular ActivityID, then the performer cannot be deleted. Otherwise, the performer should be deleted.

3. Employee(Emp_no, Dept_no, Job,

Salary) Dept(Dept_no, Dept_name)

1. Write a TRIGGER to ensure that DEPT TABLE does not contain duplicate of null values in DEPTNO column.
2. Write a Trigger to carry out the following action: on deleting a deptno from dept table , all the records with that deptno has to be deleted from the emp table
3. Write a trigger that raises an User defined error message and does not allow updating and insertion.