CSE 4508 – RDBMS Programming Lab

Lab 7

A. An ecommerce business has two tables inside their database called CustomerData and PurchaseData. Description of those two tables are,

CustomerData(userId int, name varchar, status varchar) [underscore means primary key]

PurchaseData(purchaseId int, purchaseAmount int, userId int [foreignKey referencing CustormerData's userId])

So the status column of CustomerData can have only three values: 'normal', 'member' and 'regular' (use check constraint to make sure we can insert only these three values). Next make sure that no one can alter the value of status to any other value than 'normal', 'member' and 'regular'.

Write a PL/SQL block to update the status of a customer like,

1. Customer is of 'normal' status but his/her total purchaseAmount is between 4000 to 5999 (4000<=purchaseAmount<6000) then he/she will be upgraded to 'member'.
2. Customer is of 'normal' status but his/her total purchaseAmount is greater than or equal to 6000 then he/she will be upgraded to 'regular'.
3. Customer is of 'member' status but his/her total purchaseAmount is greater than or equal to 6000 then he/she will be upgraded to 'regular'.
4. No need to degrade the customer's status. So once a regular is always a regular.

After updating, use an implicit cursor to display the number of customers whose status has been changed. **(Don't use explicit cursor, you can only use implicit cursor)**

Helper file with input values (Your code should work with the given input values)

PTO

B. A game where player needs to terminate different demons to win has below two tables,

PlayerPoint(playerId int, demonTerminatingPoint int, gamePlayDate date)

Ranking(rankingName varchar, requiredLowpoint int, requiredHighPoint int)

Write a PL/SQL function called printRanking which will take a date as input and print the rankName of a player till input date. **(Must use explicit cursor)**

Explanation :

Ranking table is identical to the below table

rankName	requiredLowpoin	requiredHighPoint
newbie	0	499
pupil	500	999
expert	1000	1e9

Now let's say our PlayerPoint table is,

playerId	demonTerminatingPoint	gamePlayDate
1	300	2019-10-25
1	300	2020-12-01

So if we do printRanking('2019-12-31') then output will be : "Player 1 was a newbie till 31-DEC-19" (since 0 <= 300 <= 499). Again if we do printRanking('2023-12-31') then output will be : "Player 1 was a pupil till 31-DEC-23" (since 500 <= 600 <= 999).

Helper file with tables and input values (Your code should work with the given input values)