Politecnico di Torino
Database Management Systems

February 6^{th} 2012

Trigger - Draft solution

1. (8 Points) The following relations are given (primary keys are underlined, optional attributes are denoted with *):

   PRODUCT (ProductCode, PName, Price, ProductPoints)
   LOYALTY_CARD (CardCode, CustomerName, TotalPoints)
   PURCHASE (PurchaseCode, ProductCode, Date, CardCode*, NumberOfItems)
   PRIZE (PrizeCode, PrizeDescription, NeededPoints)
   NOTIFICATION_REQUEST (NCode, CardCode, PrizeCode, PrizeDescription)

   A supermarket wants to manage some activities relating to loyalty cards. For each loyalty card, the LOYALTY_CARD table contains the total points (attribute TotalPoints) acquired by a customer. The PRIZE table describes the available prizes. For each prize, the NeededPoints attribute defines the value of the prize in points. Write the triggers managing the following activities.

   (1) *Assignment of points for a purchase and possible selection of the prize.* Write the trigger to update the state of the loyalty card of the customer who made the purchase. When a new purchase is made (insert in the PURCHASE table), the total points achieved by the customer must be updated. For each product, the ProductPoints attribute contains the points related to the purchase of a single item. To calculate the total points achieved in the purchase, you should consider the total number of purchased items (attribute NumberOfItems). When the product is not worth the acquisition of any points, the attribute ProductPoints is equal to zero. Note that purchases are not necessarily associated with a loyalty card. When the purchase is not associated with a loyalty card, the value of the CardCode attribute is NULL and the LOYALTY_CARD table should not be updated.

   Next, you must check if the total points accumulated by the customer allow her to obtain a prize, i.e., if there is at least one prize having value (attribute NeededPoints) less than or equal to the total points accumulated by the customer. If so, among the prizes that can be received with the accumulated points, the prize with the maximum value must be chosen. Assume that there is at most one prize satisfying this condition. Finally, a request to notify the possibility of receiving the selected prize for the cardholder must be inserted in the NOTIFICATION_REQUEST table. The NCode primary key is a counter, which should be incremented each time a new notification request is inserted (note that fully processed notification requests could be removed from database).

```
create trigger PointsUpdate
after insert on purchase
for each row
when (NEW.CardCode IS NOT NULL)
declare
  myPrizeDescription varchar(10);
  N number;
  P number;
  Z number;
  PCode number;

begin

--read points for the product
  select ProductPoints into P
  from PRODUCT
  where ProductCode = :NEW.ProductCode;

  if (P <> 0) then
     --update loyalty card
     update LOYALTY_CARD
     set TotalPoints = TotalPoints + P * :NEW.NumberOfItems
     where CardCode = :NEW.CardCode;

     --read maximum needed points for a prize
     select max(NeededPoints) INTO Z
     from PRIZE
     where NeededPoints <= (select TotalPoints
                            from LOYALTY_CARD
                            where CardCode = :NEW.CardCode);

     if (Z IS NOT NULL) then

        -- select the prize
        select PrizeCode, PrizeDescription into PCode, myPrizeDescription
        from PRIZE
        where NeededPoints = Z;

        --read maximum NCode
        select max(NCode) into
        from NOTIFICATION_REQUEST;

        if (N IS NULL) then
           N := 0;
        end if;
```

```
        --insert notification
        insert into NOTIFICATION_REQUEST (NCode, CardCode, PrizeCode, PrizeDescription)
        values (N+1, :NEW.CardCode, PCode, myPrizeDescription);
     end if;
  end if;

end;
```

(2) *Integrity constraint on the maximum value in points for a product.* For each product, the value in points (attribute `ProductPoints` in the `PRODUCT` table) can not exceed 1/10 of the sale price (attribute `Price`). Write the trigger managing the integrity constraint, by assigning the maximum value, equal to 1/10 of the sale price, when the maximum allowed value is exceeded.

```
EVENT: insert on PRODUCT
       update of Price, ProductPoints on PRODUCT
EXECUTION GRANULARITY: row level (one tuple at a time is checked)
EXECUTION MODE: before (to correct constraint violation)
CONDITION: ProductPoints > Price/10
ACTION: assign ProductPoints := Price/10

create or replace trigger checkProductPoints
before insert or update of ProductPoints, Price on PRODUCT
for each row
when (NEW.ProductPoints > NEW.Price/10)
begin
     :NEW.ProductPoints := :NEW.Price/10;
end;
```