

**SGBD- TD3.1 PLSQL****Exercice 1**

Soit le schéma relationnel de la base de données GESCOM suivant :

**Articles** (Refart, Libart, Coultart, Pvert, Qstart, Paart)

**Clients** (Codeclt, Nomclt, Prenonclt, Cateclt, Adreclt, Cpelt, Villeclt)

**Commandes** (Numcom, Datecom, #Codeclt)

**LigCom** (#Numcom, #Refart, Qtecom)

- Créer les blocs PL/SQL qui répondent aux besoins suivants :
  - Cherche et affiche le prix de vente le plus élevé de la table Articles.
  - Détermine les articles qui ont le prix de vente le plus élevé
  - Réduit de 2% le prix de vente de tous les articles de couleur rouge. Afficher le nombre d'article mis à jour.
  - Retourne la liste des numéros des commandes dans lesquelles la quantité commandée de l'article 'A200' est supérieure à 4.
- Transformer le Bloc PL/SQL de la question 1.4 en procédure.
- Créer un bloc PL/SQL qui donne la liste des articles les moins chers à l'achat (référence, libellé, prix d'achat). Le bloc PL/SQL doit faire appel à une fonction qui retourne le prix d'achat le moins cher. Il ne faut pas oublier de traiter les exceptions.
- Ecrire une procédure en PL/SQL procnbrCat qui cherche pour toutes les catégories de client le nombre total de clients de cette catégorie et retourne celles qui sont supérieures à 50 clients. Il ne faut pas oublier de traiter les exceptions.
- Ecrire une procédure procMajpv (valdif Articles.pvert%TYPE, paugt NUMBER (2)) qui cherche les articles dont la différence entre le prix de vente et le prix d'achat est inférieur ou égale à la variable valdif. Pour les articles sélectionnés, le prix de vente doit augmenter de (paugt/100). Il ne faut pas oublier de traiter les exceptions.

**Exercice 2**

Nous considérons le schéma relationnel suivant modélisant les activités d'une agence de location de voitures.

Clients (idclt, nomclt, adresse, ville, numpermis)

Voitures (nummat, modele, nb\_places, #idmarque)

Marques (idmarque, nommarque, Fabricant)

Locations (idloc, datedeb\_loc, datefin\_loc, #idclt, #nummat)

1. Ecrire une procédure **Proc\_MarquesAddict**(Mq Marque.nommarque%type,anneeloc number) qui affiche pour une marque donnée tous les modèles acquis par l'agence de voyage de cette marque et affiche pour chaque modèle les clients (nomclt, numpermis) qui ont loué des voitures de ce modèle à partir de l'année 2016.

Pour ce faire, utiliser :

- Une procédure Proc\_info\_client (numclt Clients.idclt%type) qui affiche les informations d'un client compte tenu de son identifiant.
- Un curseur curModele affichant les modèles acquis par l'agence de voyage pour une marque donnée
- Un curseur curClientModele qui affiche les identifiants des clients qui ont loué un modèle voiture à partir de l'année 2019

2. Définir une procédure **Proc\_derniereloc** qui affiche pour un client donné, son nom, la date (date fin), le numéro et la marque de la dernière voiture louée.

3. Compléter la procédure proc\_location qui va permettre d'insérer une ligne dans la table location.

Avant d'insérer une ligne, la procédure vérifie que :

- La voiture à louer existe dans la table voitures.
- La période de location est correcte.
- La voiture n'est pas déjà louée.

```
CREATE proc_location (dd DATE, df DATE, c client.idclt%TYPE, numv voiture.nummat%TYPE) IS
    nbr NUMBER;
    nbrv NUMBER ;
```

```
vloc BOOLEAN;
voitlouee EXCEPTION ;
probdte EXCEPTION ;
probvoiture EXCEPTION;
problt EXCEPTION;
```

BEGIN

```
SELECT COUNT (*) INTO nbrv FROM voitures WHERE nummat=numv ;
IF (nbrv>0) THEN
    IF (df>dd) THEN
        vloc:= voiturelouee(dd,df,numv)
        IF vloc = FALSE THEN
            INSERT INTO locations VALUES (seqloc.NEXTVAL, dd,df,c,numv)
        ELSE
            RAISE voitlouee ;
        END IF ;
    ELSE
        RAISE probdte ;
    END IF ;
ELSE
    RAISE probvoiture ;
END IF ;
```

EXCEPTION

**END procloc1 ;**

3. Définir la fonction **voiturelouee** qui vérifie si la voiture est déjà en location durant la période choisie par le client.

4. Compléter le programme PLSQL suivant en se basant sur , entre autres, sur les données saisies par l'utilisateur selon les instructions suivantes:

**SET SERVEROUTPUT ON**

**ACCEPT c1 PROMPT 'Entrer le numéro du permis du client:'**

**ACCEPT d1 PROMPT 'Entrer la date du début de la location:'**

**ACCEPT d2 PROMPT 'Entrer la date de la fin de la location:'**

**SET SERVEROUTPUT ON**

DECLARE

vclient Clients.idclt%TYPE;

Vmat Clients.matricule%type := '210 TUN 3050'

BEGIN

--vérifier si le client de permis saisi existe ou non

--Afficher la date de la dernière location

proc\_derniereloc (vclient);

--Ajouter la nouvelle location à la table location