

Cartouche du document

Année : ING 2

Activité : Travail dirigé

Objectifs

L'objectif de cette série d'exercices est :

- de se familiariser avec la gestion d'erreur dans un bloc PL/SQL
- de maîtriser la conception de fonctions et procédures stockées
- d'apprendre à utiliser des curseurs dans un bloc PL/SQL

Sommaire des exercices

- 1 - Gestion de commandes
- 2 - Gestion de classes

Corps des exercices

1 - Gestion de commandes

Énoncé :

Soit le MLD suivant :

```
produit: idproduit, designationprod, prixprod, stockprod  
commande : idcommande, datecom, montanttotalcom, #idclient  
lignecommande : idligcom, quantiteprod, #idproduit, #idcommande  
ligneerreur : idcommande, idligcom, idproduit, quantiteprod, stockprod
```

On désire, à chaque insertion d'une ligne de commande mettre à jour automatiquement le montant total de la commande correspondante ainsi que le stock du produit concerné par la ligne de commande.

Question 1)

Énoncé de la question

Ecrire le code correspondant en gérant le fait qu'une ligne de commande peut ne pas être satisfaite (stock inférieur à la quantité commandée).

Une ligne de commande non satisfaite devra faire l'objet d'une insertion dans la table LIGNEERREUR par l'intermédiaire d'une gestion d'exception.

1) Pour exécuter dans une fonction une transaction indépendante de la transaction courante, on utilise la clause PRAGMA AUTONOMOUS_TRANSACTION;. Cette clause apparaît dans la zone de déclaration des variables du bloc PL/SQL

2) Pour provoquer un rollback de la transaction courante en raison d'une erreur, on utilise la fonction raise_application_error(noErreur,msgErreur) où noErreur doit être un nombre

négatif strictement inférieur à -20000 (les autres codes d'erreur sont réservés par Oracle).

Ci-dessous, vous trouverez le script de création des tables en vue tester le trigger.

```
drop table produit cascade constraints;
drop table commande cascade constraints;
drop table lignecommande cascade constraints;
drop table ligneerreur cascade constraints;
create table produit
(idproduit varchar2(5) primary key,
desprod varchar(50),
prixprod number(8,2),
stockprod number(4)
);
create table commande
(idcommande varchar2(5) primary key,
datecde date,
montanttotalcom number(10,2),
idclient varchar2(5)
);
create table lignecommande
(idlig varchar2(5),
idcommande varchar2(5) references commande(idcommande),
quantiteprod number(3),
idproduit varchar2(5) references produit(idproduit),
primary key (idlig,idcommande)
);
create table ligneerreur (
idcommande varchar2(5),
idlig varchar2(5),
idproduit varchar2(5),
quantiteprod number(3),
stockprod number(4)
);
insert into produit values ('P1',' ',50,50);
insert into produit values ('P2',' ',500,0);
insert into commande values ('cde1',' ',0,'c1');
```

Question 2)

Énoncé de la question

On désire maintenant afficher la facture correspondant à une commande. Sur cette facture doit apparaître : le client, le numéro de la commande et la date de la commande, le montant total ainsi que pour chaque produit commandé, la quantité, le prix et le montant à payer correspondant.

Ecrire la procédure qui acceptera en paramètre le numéro de la commande et qui permettra d'afficher la facture.

2 - Gestion de classes

Énoncé :

On gère des classes dans un collège. Une classe est composée de collégiens. Un collégien est décrit par la relation Collégien : idEtudiant,Nom,Prenom.

Question 1)

Énoncé de la question

Ecrire une procédure qui reçoit un triplet idEtudiant,Nom,Prenom et qui

- insère un nouveau collégien si la clé idEtudiant n'existe pas dans la table ;
- met à jour le nom et le prénom d'un nouveau collégien si la clé idEtudiant existe dans la table.