

Cartouche du document

Année : ING 2

Activité : Travail dirigé

Objectifs

Approfondir le langage PL/SQL : package fonction et procédure.

Sommaire des exercices

- 1 - création de package
- 2 - Encore des triggers ...

Corps des exercices

1 - création de package

Enoncé :

Soit le script de table suivant :

```
drop table commande cascade constraints;
```

```
create table commande
(idcommande varchar2(5) primary key,
datecde date,
montanttotalcom number(10,2),
idclient varchar2(5),
idvendeur varchar2(4)
);
```

```
insert into commande values('cd1',' ',100,'cl1','v1');
insert into commande values ('cd2',' ',200,'cl1','v1');
insert into commande values('cd3',' ',300,'cl2','v1');
insert into commande values('cd4',' ',300,'cl2','v2');
insert into commande values('cd5',' ',200,'cl2','v3');
insert into commande values('cd6',' ',150,'cl3','v3');
```

Le but de cet exercice est de créer un package qui regroupera les objets (fonctions et procédures) permettant de créer un rapport d'activité détaillant pour chaque vendeur son ca global et le ca des clients.

Un exemple de rapport est donné ci-après :

Rapport d activité du jj mm aa

Nom vendeur1, CA gloabl

idclient1, ca client1 pour le vendeur1

idclient2,ca client2 pour le vendeur1

nom vendeur2, ca global

idclient1, ca client1 pour le vendeur2

idclient3, ca client3 pour le vendeur2

idclient4,ca client4 pour le vendeur2

Question 1)

Enoncé de la question

Créer une fonction qui permettra de calculer le ca global d'un vendeur

Solution de la question

Question 2)

Enoncé de la question

Créer une fonction qui permettra de calculer le ca d'un client pour un vendeur donné

Solution de la question

Question 3)

Enoncé de la question

Créer la procédure qui permettra d'obtenir le rapport décrit ci-dessus.

Solution de la question

Question 4)

Enoncé de la question

Créer un package qui contiendra les trois objets définis ci-dessus.

Solution de la question

create or replace package ca_pk

```

is
function ca_client (Pidclient commande.idclient%type,pidvendeur commande.idvendeur%type) return number ;
function ca_vendeur (Pidvendeur commande.idvendeur%type) return number ;
procedure aff_ca;
end ca_pk ;
/

create or replace package body ca_pk
is
function ca_client(pidclient commande.idclient%type,pidvendeur commande.idvendeur%type)
return number
is
Vca_client commande.montanttotalcom%type ;
begin
select sum(montanttotalcom) into Vca_client
from commande
where idclient = pidclient
and idvendeur=pidvendeur
group by pidvendeur,pidclient;
return vca_client ;
exception
when no_data_found then return -1 ;
end ca_client ;
-- fonction calcul ca pour le vendeur
function ca_vendeur (Pidvendeur commande.idvendeur%type)
return number
is
Vca_vendeur commande.montanttotalcom%type ;
begin
select sum(montanttotalcom) into vca_vendeur from commande where idvendeur=Pidvendeur
group by idvendeur ;
return vca_vendeur ;
exception
when no_data_found then return -1 ;
end ca_vendeur ;
-- procédure aff_ca
procedure aff_ca
is
vidvendeur commande.idvendeur%type ;
vidclient commande.idclient%type ;
vca_client commande.montanttotalcom%type ;
vca_vendeur commande.montanttotalcom%type ;
cursor cur_vendeur is select distinct idvendeur from commande ;
cursor cur_client(vidvendeur commande.idvendeur%type) is select distinct idclient from commande where
idvendeur=vidvendeur;
begin
dbms_output.put_line('compte rendu d activité') ;
dbms_output.put_line('-----');
for vcur_vendeur in cur_vendeur loop
vca_vendeur := ca_vendeur(vcur_vendeur.idvendeur);
dbms_output.put_line('vendeur'||' '||vcur_vendeur.idvendeur||' ' || vca_vendeur) ;
for vcur_client in cur_client(vcur_vendeur.idvendeur) loop
vca_client := ca_client(vcur_client.idclient,vcur_vendeur.idvendeur) ;
dbms_output.put_line('client'||' '||vcur_client.idclient|| ' ' || vca_client) ;
end loop ;
dbms_output.put_line('-----');
end loop ;
end aff_ca ;
end ca_pk;
/

pour excuter la procédure :
execute ca_pk.aff_ca

```

2 - Encore des triggers ...

Enoncé :

Soit la table définie comme suit :

```
drop table table_log ;
create table table_log
(
  utilisateur varchar2(50) not null,
  action varchar2(30) not null ,
  date_action date not null
) ;
```

La colonne utilisateur correspond au username de l'utilisateur.

La colonne action permet de savoir ce qui a déclenché le trigger (insert, update ou delete). La colonne date_action correspond à la date à laquelle a été effectué l'action.

Question 1)

Enoncé de la question

Créer un trigger qui à chaque action effectuée sur la table commande ira mettre à jour la table table_log

Solution de la question

```
create or replace trigger maj_log
after insert or update or delete
on commande
declare
  vutilisateur table_log.utilisateur%type;
  vaction table_log.action%type;
  vdate_action table_log.date_action%type;
begin
  select user,sysdate into vutilisateur,vdate_action from dual ;
  if inserting then vaction := 'insert' ;
  elsif deleting then vaction := 'delete' ;
  elsif updating then vaction := 'update';
end if;
insert into table_log values (vutilisateur,vaction,vdate_action) ;
end ;
```