DENIZOT Hugo

F2

COMPTE RENDU TP BDD

Voici le compte rendu du TP de BDD. Pour les tests je n'ai pas réussi à me reconnecter à Guacamole durant les vacances j'ai donc pris des captures d'écrans des résultats que j'avais copié lors de l'exécution de certaines questions. Pour le code j'ai fait une capture d'écran de mon code afin que cela soit plus lisible.

Package :

Voici le code du package avec ses procédures :

```
CREATE OR REPLACE Package hudenizot_pack
AS
    TYPE tuple_employe IS RECORD(numero emp.empno%type, nom emp.ename%type);
    CURSOR emp_par_dep_hudenizot(deptno number) RETURN tuple_employe;
    function salok_hudenizot (jobs  SalIntervalle_hudenizot.job%type, salaire SalIntervalle_hudenizot.lsal%type) return number;
    procedure raisesalary_hudenizot(emp_id IN emp.empno%type, amount in emp.sal%type);
    procedure afficher_emp_hudenizot(deptno number);
end hudenizot_pack;
/

CREATE OR REPLACE Package hudenizot_pack
AS
    CURSOR emp_par_dep_hudenizot(deptno number) RETURN tuple_employe IS
        SELECT empno,ename w

    function salok_hudenizot(jobs IN SalIntervalle_hudenizot.job%type, salaire in SalIntervalle_hudenizot.lsal%type)
    RETURN number
    IS
        retour number;
        salairemin SalIntervalle_hudenizot.lsal%type;
        salairemax SalIntervalle_hudenizot.hsal%type;

    BEGIN
        select lsal,hsal into salairemin,salairemax FROM SalIntervalle_hudenizot Where job=jobs;
        if salaire>salairemin and salaire<salairemax then
            retour:=1;
        else
            retour:=0;
        end if;
        return(retour);
    END salok_hudenizot;
    /

    procedure raisesalary_hudenizot(emp_id IN emp.empno%type, amount in emp.sal%type)
    IS
        retour number;
        nom_job emp.job%type;
        sal_actuelle emp.sal%type;
        futur_sal emp.sal%type;

    BEGIN
        select job,sal into nom_job,sal_actuelle from emp where empno=emp_id;
        if nom_job is not null then
            futur_sal:=sal_actuelle+amount;
            select salok_hudenizot(nom_job,futur_sal) into retour from dual;
            if retour=1 then
                UPDATE emp SET sal=sal_actuelle+amount Where empno=emp_id ;
            else
                dbms_output.put_line('Le salaire ne sera plus dans l''intervale');
            end if;
        end if;
    END raisesalary_hudenizot;
    procedure afficher_emp_hudenizot(nodept  emp.deptno%type)
    IS
    Begin
        for rec in emp_par_dep_hudenizot(nodept)
        loop
            dbms_output.put_line(rec.numero);
            dbms_output.put_line(rec.nom);
        end loop;


    end afficher_emp_hudenizot;
end hudenizot_pack;
/
```

Je n'ai pas pu faire de test sur cette partie je ne peux donc pas savoir si elles sont fonctionnelles mais c'est comme ça que je les aurais codé et il me semble que cela se créait.

Trigger :

Q1)

Voici le code du trigger de la question 1 :

```
/*TRIGGER Q1*/
CREATE OR REPLACE trigger hudenizot.raise_hudenizot
BEFORE UPDATE of sal
ON hudenizot.emp
For each row
BEGIN
    if :old.sal>:new.sal then
        Raise_application_error(-20003,'Pas le droit d''un salaire moins bon');
    end if;
END;
/
```

J'ai oublié de garder le résultat des tests mais ceux-ci marchaient, on ne peut pas diminuer le salaire avec un update, cependant par la suite cela sera possible lors du changement de job si le salaire est trop haut (voir question 8).

Q2)

Voici le code du trigger de la question 2 :

```
/*TRIGGER Q2*/
CREATE OR REPLACE trigger hudenizot.numdept_hudenizot
BEFORE UPDATE of deptno or INSERT
ON hudenizot.dept
for each row
BEGIN
    if :new.deptno<61 or :new.deptno>69 then
        Raise_application_error(-20003,'Le departement doit etre compris entre 61 et 69' );
    end if;
end;
/
```

Idem pour cette question je n'ai pas pris le temps de recopier le résultat des tests mais le trigger se créait et fonctionnait, les seules départements possibles sont ceux compris entre 61 et 69 inclus.

Q3)

Voici le code du trigger de la question 3 :

```
/*TRIGGER Q3*/
CREATE OR REPLACE trigger hudenizot.dept_hudenizot
BEFORE UPDATE of deptno or INSERT
ON hudenizot.emp
for each row
DECLARE
nombredept number;
BEGIN
    select count(*) into nombredept from dept where deptno=:new.deptno;
    if nombredept=0 then
        INSERT INTO DEPT VALUES(:new.deptno,'A SAISIR','A SAISIR');
    end if;
end;
/
```

Et voici ce qui se passe lorsqu'on le test :

```
AVANT:
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        12 bdd            Clermont

SQL> INSERT INTO emp VALUES(9999,'blabla','test',7000,null,2500,null,62);

1 ligne creee.

APRES:
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        12 bdd            Clermont
        62 A SAISIR       A SAISIR

APRES:

SQL> UPDATE emp set deptno=63 WHERE empno=9999;

1 ligne mise a jour.

SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        12 bdd            Clermont
        63 A SAISIR       A SAISIR
        62 A SAISIR       A SAISIR
```

On peut donc voir que lorsque l'on rajoute ou modifie un employé avec un département inconnu cela le créé.

Q4)

Voici le code du trigger de la question 4 :

```
set serveroutput on
CREATE OR REPLACE trigger hudenizot.noweek_hudenizot
BEFORE UPDATE of mgr
ON hudenizot.emp
for each row
DECLARE
jour  VARCHAR(80);
BEGIN
    select TO_CHAR(SYSDATE, 'D') into jour from dual;
    if jour=1 or jour=7 then
        Raise_application_error(-20003,'Pas le droit de changer la relation un week-end');
    else
        dbms_output.put_line(jour);
    end if;
end;
/
```

Et voici ce qui se passe quand on le test :

```
SQL> update emp set mgr=7934 where empno=9999;
update emp set mgr=7934 where empno=9999
       *
ERREUR a la ligne 1 :
ORA-20003: Pas le droit de changer la relation un week-end
ORA-06512: a "HUDENIZOT.NOWEEK_HUDENIZOT", ligne 6
ORA-04088: erreur lors d'execution du declencheur 'HUDENIZOT.NOWEEK_HUDENIZOT'
```

Q5)

Voici la commande pour désactiver un trigger :

```
ALTER TRIGGER hudenizot.noweek_hudenizot DISABLE;
```

Et voici ce que ça donne quand on reteste une fois modifié :

```
SQL> ALTER TRIGGER hudenizot.noweek_hudenizot DISABLE
  2  ;

Declencheur modifie.

SQL> update emp set mgr=7934 where empno=9999;

1 ligne mise a jour.
```

Q6)

Voici la commande pour réactiver un trigger :

```
ALTER TRIGGER hudenizot.noweek_hudenizot ENABLE;
```

Et voici ce que ça donne quand on reteste une fois modifié :

```
SQL> ALTER TRIGGER hudenizot.noweek_hudenizot ENABLE
  2  ;

Declencheur modifie.

SQL> update emp set mgr=7934 where empno=9999;
update emp set mgr=7934 where empno=9999
       *
ERREUR a la ligne 1 :
ORA-20003: Pas le droit de changer la relation un week-end
ORA-06512: a "HUDENIZOT.NOWEEK_HUDENIZOT", ligne 6
ORA-04088: erreur lors d'execution du declencheur 'HUDENIZOT.NOWEEK_HUDENIZOT'
```

Q7)

Voici le code du trigger de la question 7 et la création de la table associée :

```
CREATE TABLE STATS_hudenizot(TypeMaj VARCHAR(80) PRIMARY KEY,NbMaj number,Date_derniere_maj DATE);

INSERT INTO STATS_hudenizot VALUES('INSERT',0,null);
INSERT INTO STATS_hudenizot VALUES('UPDATE',0,null);
INSERT INTO STATS_hudenizot VALUES('DELETE',0,null);

CREATE OR REPLACE trigger hudenizot.modif_stats_hudenizot
AFTER UPDATE or INSERT or DELETE
ON hudenizot.emp
for each row
BEGIN
    IF INSERTING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=SYSDATE WHERE TypeMaj='INSERT';
    END if;
    IF UPDATING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=SYSDATE WHERE TypeMaj='UPDATE';
    END if;
    IF DELETING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=SYSDATE WHERE TypeMaj='DELETE';
    END if;
end;
/
```

Voici ce qui se passe quand on test le trigger :

```
CREATE OR REPLACE trigger hudenizot.modif_stats_hudenizot
BEFORE UPDATE or INSERT or DELETE
ON hudenizot.emp
for each row
DECLARE
BEGIN
    IF INSERTING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=TO_CHAR(SYSDATE,'dd-mm-yyyy') WHERE TypeMaj='INSERT';
    END if;
    IF UPDATING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=TO_CHAR(SYSDATE,'dd-mm-yyyy') WHERE TypeMaj='UPDATE';
    END if;
    IF DELETING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=TO_CHAR(SYSDATE,'dd-mm-yyyy') WHERE TypeMaj='DELETE';
    END if;
end;
 17  /

Declencheur cree.

SQL> update emp set sal=sal+1 where empno=9999;

1 ligne mise a jour.

SQL> select * from stats_hudenizot;

TYPEMAJ
--------------------------------------------------------------------------------
     NBMAJ DATE_DER
---------- --------
INSERT
         0

UPDATE
         1 06/02/23

DELETE
         0


SQL> update emp set sal=sal+1;

16 lignes mises a jour.

SQL> select * from stats_hudenizot;

TYPEMAJ
--------------------------------------------------------------------------------
     NBMAJ DATE_DER
---------- --------
INSERT
         0

UPDATE
        17 06/02/23

DELETE
         0
```

Et maintenant voici ce qui se passe si on retire la ligne « for each row »

```
SQL> select * from stats_hudenizot;

TYPEMAJ
--------------------------------------------------------------------------------
     NBMAJ DATE_DER
---------- --------
INSERT
         0

UPDATE
        33 06/02/23

DELETE
         0


CREATE OR REPLACE trigger hudenizot.modif_stats_hudenizot
BEFORE UPDATE or INSERT or DELETE
ON hudenizot.emp
DECLARE
BEGIN
    IF INSERTING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=TO_CHAR(SYSDATE,'dd-mm-yyyy') WHERE TypeMaj='INSERT';
    END if;
    IF UPDATING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=TO_CHAR(SYSDATE,'dd-mm-yyyy') WHERE TypeMaj='UPDATE';
    END if;
    IF DELETING then
        UPDATE STATS_hudenizot set NbMaj=NbMaj+1, Date_derniere_maj=TO_CHAR(SYSDATE,'dd-mm-yyyy') WHERE TypeMaj='DELETE';
    END if;
end;
 16  /

Declencheur cree.

SQL> update emp set sal=sal+1;

16 lignes mises a jour.

SQL> select * from stats_hudenizot;

TYPEMAJ
--------------------------------------------------------------------------------
     NBMAJ DATE_DER
---------- --------
INSERT
         0

UPDATE
        34 06/02/23

DELETE
         0
```

On peut donc voir que sans le for each row on aura pas exactement le nombre de ligne modifié à chaque insert update ou delete mais le nombre d’insert,update ou delete réalisé.

```
SQL> DELETE From emp where empno=9999;

1 ligne supprimee.

SQL> select * from stats_hudenizot;

TYPEMAJ
--------------------------------------------------------------------------------
    NBMAJ DATE_DER
---------- --------
INSERT
         0

UPDATE
        34 06/02/23

DELETE
         1 06/02/23

SQL> INSERT INTO EMP VALUES(9999,'blabla','blabla',7000,null,2100,null,63);

1 ligne creee.

SQL> select * from stats_hudenizot;

TYPEMAJ
--------------------------------------------------------------------------------
    NBMAJ DATE_DER
---------- --------
INSERT
         1 06/02/23

UPDATE
        34 06/02/23

DELETE
         1 06/02/23
```

Voici le résultat du trigger lors d'un insert et d'un delete.

Q8)

Voici le code du trigger de la question 8 :

```
CREATE OR REPLACE trigger hudenizot.checksal_hudenizot
BEFORE UPDATE of job
ON hudenizot.emp
for each row WHEN (old.job!='PRESIDENT')
DECLARE
lowsal number;
highsal number;
retourJob number;
BEGIN
    select count(*) into retourJob FROM salintervalle_hudenizot where job=:new.job;
    if retourJob=0 then
        Raise_application_error(-20003,'Ceci n''est pas un metier connu');
    else
        select lsal,hsal into lowsal,highsal from salintervalle_hudenizot where job=:new.job;
        if lowsal> (:old.sal+100) then
            :new.sal:=lowsal;
        else if highsal< (:old.sal+100) then
            :new.sal:=highsal;
        else
            :new.sal:=:old.sal+100;
        end if;
        end if; /*Bien fermé les else if*/
    end if;

end;
/
```

J'ai oublié de garder le résultat des tests du trigger mais ceux-ci fonctionnaient bien, lorsque l'on change de métier si le salaire maximum du nouveau métier est inférieur au salaire actuel alors il aura le salaire maximum du nouveau métier et si le salaire minimum du nouveau métier est supérieur au salaire actuel alors il aura le salaire minimum du nouveau métier. Si un président change de métier il ne sera pas affecté et gardera son salaire actuel, cependant si cette personne rechange de métier alors le trigger se déclenchera.