Collège de Rosemont

Formation Continue

Groupe : 0283

**EVALUATION 1**

420-A34-RO Approfondissement des bases de données

Professeur : Florent Adima

29/05/2017

Nom : **Marchand Magali**

Numéro étudiant : 1699241

**PARTIE 1 : Fonctions – Procédures – Triggers**

**Énoncé 1 - 1**

CREATE OR REPLACE FUNCTION trouverEmploye (f\_no\_Emp IN NUMBER)

RETURN NUMBER IS

v\_no\_Emp NUMBER;

CURSOR c1 IS SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID = f\_no\_Emp;

BEGIN

OPEN c1;

FETCH c1 INTO v\_no\_Emp;

IF c1%NOTFOUND THEN

v\_no\_Emp := 0;

DBMS\_OUTPUT.PUT\_LINE('Il n y a pas d employe a ce nom');

END IF;

CLOSE c1;

RETURN v\_no\_Emp;

END;

**Énoncé 1 - 2**

create or replace procedure supprimerEmploye (f\_no\_Emp IN NUMBER) is

begin

DELETE FROM EMPLOYEES

WHERE EMPLOYEE\_ID = f\_no\_Emp;

end;

**Énoncé 1 - 3**

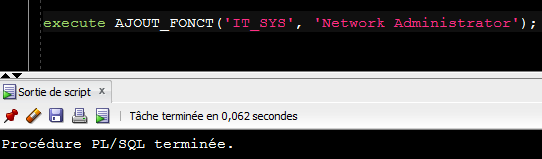
create or replace procedure AJOUT\_FONCT (p\_id\_job in VARCHAR2, p\_title\_job in VARCHAR2) is

begin

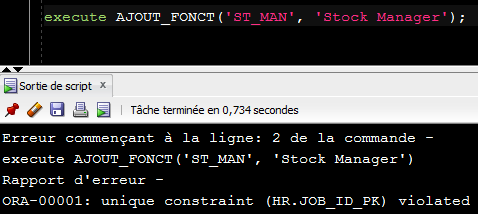
INSERT INTO JOBS(JOB\_ID,JOB\_TITLE)

values(p\_id\_job,p\_title\_job);

end;







Erreur : On ne peut pas insérer ce poste car il existe déja et possède une contrainte de non duplication sur la colonne JOB\_ID qui est la clé primaire de la table JOBS.

**Énoncé 1 - 4**

CREATE OR REPLACE TRIGGER unique\_president

BEFORE INSERT ON EMPLOYEES

FOR EACH ROW

DECLARE

nb\_pres NUMBER;

BEGIN

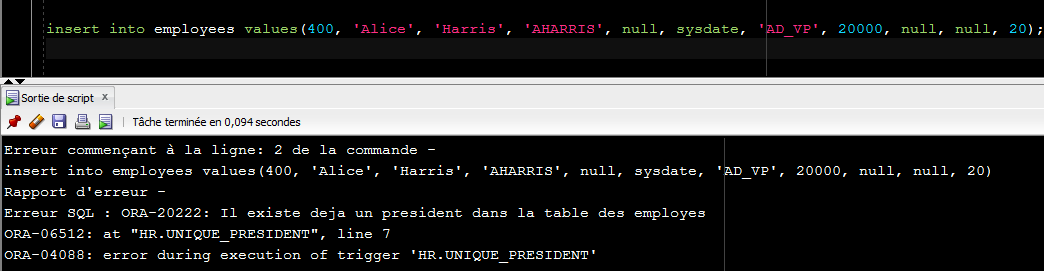
select count(JOB\_ID) into nb\_pres from EMPLOYEES where JOB\_ID = 'AD\_PRES';

if nb\_pres = 1 then

Raise\_application\_error (-20222, 'Il existe deja un president dans la table des employes');

end if;

END;



**Énoncé 2 - a**

CREATE OR REPLACE FUNCTION GET\_JOB\_COUNT (p\_id\_emp IN NUMBER)

RETURN NUMBER IS

v\_nb1 NUMBER;

v\_nb2 NUMBER;

v\_nb3 NUMBER;

notfound\_id EXCEPTION;

CURSOR c1 IS SELECT distinct JOB\_ID FROM JOB\_HISTORY WHERE EMPLOYEE\_ID = p\_id\_emp;

CURSOR c2 IS SELECT JOB\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID = p\_id\_emp;

CURSOR c3 IS SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE EMPLOYEE\_ID = p\_id\_emp;

BEGIN

OPEN c1;

OPEN c2;

OPEN c3;

v\_nb1 := 0;

v\_nb2 := 0;

v\_nb3 := 0;

WHILE c1%FOUND LOOP

v\_nb1 := v\_nb1 + 1;

END LOOP;

IF c1%NOTFOUND THEN

v\_nb1 := 0;

END IF;

IF c2%FOUND THEN

v\_nb2 := 1;

else IF c2%NOTFOUND THEN

v\_nb2 := 0;

END IF;

IF c3%NOTFOUND THEN

RAISE notfound\_id;

END IF;

v\_nb3 := v\_nb1 + v\_nb2;

EXCEPTION

WHEN notfound\_id THEN

raise\_application\_error (-20001,'Cet id employe ne se trouve pas dans la base de donnees.');

CLOSE c1;

CLOSE c2;

CLOSE c3;

RETURN v\_nb3;

END;

**PARTIE 2 : Affichage d’informations et curseurs**

**Énoncé 1**

CREATE OR REPLACE PROCEDURE listeEmployes

IS

v\_prenom VARCHAR2(20 BYTE);

v\_nom VARCHAR2(25 BYTE);

v\_salaire NUMBER(8,2);

CURSOR c1 IS SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM EMPLOYEES

WHERE JOB\_ID = 'IT\_PROG';

BEGIN

OPEN c1;

WHILE c1%FOUND LOOP

FETCH c1 INTO v\_prenom, v\_nom, v\_salaire;

DBMS\_OUTPUT.PU\_LINE('Employe: ' || v\_prenom || 'Nom: ' || v\_nom || 'Salaire: ' || v\_salaire);

END LOOP;

CLOSE c1;

END;