

SGBD TP6 :

PL/SQL : Fonction, procédure, Exception (implicite et explicite)

- Durée 1h30.
- Pour ce TP vous avez besoin de la BD Gest_Appart.
- Le compte rendu de ce TP doit être sous la forme suivante : n°question, bloc PL/SQL, capture écran du résultat.

Le schéma relationnel de la BD Gest_Appart :

Immeuble (idimm, adresse, codep, nbrEtages, dateConstr)

Personne (idpers, nom, prenom, adrOfficielle, codep, dateN, profession, tel, Persmorale)

Appartement (numapp, #idimm, descript, nbrpiece, superficie, etage, occupe)

ContratAchat(numcontA, #(numapp, idimm), dateAchat, PrixAchat)

PropAppart (#numcontA, #idpers)

ContratLoc (numcontLoc, #(numapp, idimm), dateLoc, #idpers, datedepart, datedepR, loyer,Caution)

Travail à faire :

1. Soit le bloc PL/SQL ci dessous qui n'est pas complet :

```

set serveroutput on;
ACCEPT dated
PROMPT 'Donner la date du début de la période : '
ACCEPT dateF
PROMPT 'Donner la date de la fin de la période : '
DECLARE
Vmax contratloc.loyer%TYPE ;
vdated DATE :=&dated ;
vdateF DATE :=&datef ;

tabvid EXCEPTION;

BEGIN
Vmax := f_maxLoyer(vdated,vdateF) ;
IF Vmax=0 THEN
    RAISE tabvid;
ELSE
    DBMS_OUTPUT.PUT_LINE('Le loyer le plus élevé entre le '||vdated||' et le '||vdatef||' est : '||vmax);
END IF ;
EXCEPTION
WHEN tabvid THEN DBMS_OUTPUT.PUT_LINE('Pas d'appartements louées entre le '||vdated||' et le '||vdatef);
WHEN ErDate THEN DBMS_OUTPUT.PUT_LINE ('Erreur dans le choix de la période : la date début doit être inférieur à la date fin et à la date d''aujourd''hui!');
END ;
/

```

- 1.1. Ecrire une fonction f_maxLoyer (dated,datef) qui cherche le loyer maximum pour une période donnée.
- 1.2. Compléter le traitement de l'exception **ErDate** et exécuter le bloc PLSQL.

2. On veut chercher le nombre et la liste des appartements non occupés depuis une période dans l'immeuble où habite une personne (nom et prenom) donner. Voir bloc PL/SQL ci-dessous :

```
DECLARE
Vadr immeuble.adresse%TYPE ;
Vidim Immeuble.idimm%TYPE;
Vnomp personne.nom%TYPE := 'Kaabi';
Vprenomp personne.prenom%TYPE := 'mohamed' ;
Errempty EXCEPTION ;
Errmany EXCEPTION ;
BEGIN
  Vidim :=f_chercherIm(vnomp,vprenomp) ;
  IF vidim = 0 THEN RAISE ERRempty ;
  ELSE IF vidim = -1 THEN RAISE ERRmany ;
    ELSE
      Vadr := f_chercheradrIm(vidim) ;
      p_appartLibre(vidim) ;
    END IF ;
  END IF ;
EXCEPTION
WHEN errempty THEN DBMS_OUTPUT.PUT_LINE('Cette personne n'est pas locataire !');
WHEN errmany THEN DBMS_OUTPUT.PUT_LINE('Plusieurs personnes avec ce nom et ce prénom !');
END ;
```

2.1. Ecrire la fonction **f_chercherIm(vnomp,vprenomp)** qui retourne l'identifiant de l'Immeuble où habite une personne (recherche par le nom et le prénom).

2.2. Traiter les exceptions dans la procédure **p_appartLibre** ci-dessous :

```
CREATE OR REPLACE PROCEDURE p_appartLibre(idim Immeuble.idimm%TYPE) IS

CURSOR curAppLibre IS
SELECT numapp, descript,nbrpiece,superficie,etage
FROM Appartement
WHERE UPPER(occupe) LIKE ('non') AND idimm=idim;

Vcurapp curAppLibre%ROWTYPE ;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Les appartements libres de l'immeuble : '|| idim) ;
  OPEN curAppLibre ;

  LOOP
    FETCH curAppLibre INTO vcurapp ;
    EXIT WHEN curAppLibre%NOTFOUND OR curAppLibre%NOTFOUND IS NULL ;
    DBMS_OUTPUT.PUT_LINE('- '|| vcurapp.numapp||', '|| vcurapp.descript|| '- '||
vcurapp.nbrpiece||'
    pieces '|| vcurapp.superficie|| ' m2 '|| vcurapp.etage ||' etage') ;
  END LOOP ;
  CLOSE curAppLibre ;

END ;
```

2.3. Traiter les exceptions dans la fonction **f_chercherIm** ci-dessous :

```
CREATE OR REPLACE FUNCTION f_chercheradrIm (idim Immeuble.idimm%TYPE) RETURN Immeuble.
Adresse%TYPE IS
result Immeuble. Adresse%TYPE ;
BEGIN
  SELECT adresse INTO result FROM Immeuble WHERE idimm=idim ;
  RETURN result ;
END f_chercheradrIm;
```