TP1 : PL/SQL (séance sécurité des schémas)

1. Schéma de base de données

Nous allons travailler sur le schéma relationnel Employé (schéma de test classique pour Oracle). Le script de création des tables vous est donné (ScriptCreation.sql).

— Dept(**n_dept**, nom, lieu)
— Emp(nom, **num**, fonction, n_sup, embauche, salaire, com, n_dept)
avec Emp(n_dept)⊆Dept(n_dept)
et Emp(n_sup)⊆Emp(num)

2. Contraintes clés primaires et étrangères

Vous exécuterez le script de création une fois connecté à Oracle (BD master).

```
Exemple pour le compte exxxxxx et le mdp yyyy :
rlwrap sqlplus exxxxxx/yyyy@oracle.etu.umontpellier.fr:1523/pmaster

Si pas de compte ou pb :
aller sur
https://sapiens.umontpellier.fr/
```

Listing 1 – connexion

Vous définirez les contraintes de clés comme indiquées dans le schéma relationnel, au moyen d'un ordre ALTER TABLE. Pensez à exploiter le "on delete cascade" pour les contraintes de clé étrangère, qui permettra par exemple de supprimer les tuples des employés qui travaillent dans un département qui n'existe plus.

```
alter table emp add constraint emp_pk primary key (num);
alter table dept add constraint dept_pk primary key (n_dept);
alter table emp add constraint emp_fk1 foreign key (n_sup) references emp(num) on delete
    cascade;
alter table emp add constraint emp_fk2 foreign key (n_dept) references dept(n_dept) on
    delete cascade;
```

Listing 2 – ajout de contraintes

3. Triggers LMD

Une partie des questions est à traiter en non présentiel.

1. Construisez un trigger nommé *rennesMille* qui vérifie que le salaire des employés est toujours supérieur à 1000 euros, à l'insertion comme à la mise à jour, mais uniquement pour des employés qui travaillent dans le département localisé à Rennes.
2. Les triggers ont souvent une écriture très simple (règle ECA) et s'appuient sur des procédures qui en masquent la complexité. Vous reprendrez l'écriture du trigger ouvrable (donné dans le cours). Vous définirez une procédure *JoursEtHeuresOuvrables* sans argument qui vérifie que la date du jour n'est pas un samedi ni un dimanche et qui renvoie un message d'erreur autrement. Vous redéfinirez le trigger *ouvrable* qui fera appel à cette procédure, dans le contexte de la table Emp. Vous en testerez les effets.
3. Les triggers ont souvent un rôle de monitoring auprès des administrateurs de bases de données (indicateurs de la bonne santé des bases de données dont ils ont la charge). Vous créerez une table historique (dateOperation, nomUsager, typeOperation) qui va permettre de conserver la trace de toutes les opérations réalisées sur la table *Dept*. Vous créerez le trigger nommé *monitor* qui va permettre d'alimenter cette table.
   Le nom de l'usager et la date système sont connus au travers des descripteurs USER et SYSDATE (cf. select user, sysdate from dual ;)
4. Lors de la définition de contraintes de clés étrangères, Oracle autorise certaines fonctionnalités telle que la suppression en cascade des tuples dépendants au travers de la syntaxe réservée *on delete cascade*. A la différence de PostgreSQL, Oracle ne permet pas l'exploitation de la syntaxe *on update cascade* qui permettrait de modifier les tuples dépendants en conséquence.
   Vous construirez un trigger nommé *cascade* qui porte sur la table Dept et qui se charge à chaque évènement de suppression ou de modification d'un département (n_dept) dans *Dept* de supprimer ou de modifier dans la table *Emp*, les tuples d'employés dépendants de ce département. **Pensez à supprimer au préalable la contrainte de clé étrangère sur n_dept de la table (pose des problèmes de conflit sinon). Pensez ensuite à annuler les effets des suppressions ou modifications par rollback sur la transaction.**

4. Triggers LDD

Vous construirez un trigger qui se déclenche au niveau de votre schéma utilisateur et qui affiche un message indiquant un changement du modèle à chaque ordre de création d'objet (table ou autre objet du schéma).

5. Triggers d'instance

Vous construirez un trigger qui se déclenche au niveau de chaque connexion utilisateur (évènement qui se produit au niveau de la base de données). L'action découlant de cet évènement devra être une insertion dans une table nommée QuiSeConnecte que vous aurez préalablement définie. Prenez garde de définir des tailles de chaîne variable suffisamment élevées pour la variable OS_USER (au moins 60 caractères). La définition de cette catégorie de trigger nécessite le privilège : "administer database trigger", et donc l'octroi de ce privilège par le DBA, de la manière suivante :

```
grant administer database trigger to public;
```

Vous utiliserez la fonction sys_context qui donne des informations sur l'environnement de l'utilisateur et qui peut être exploitée à partir de la table à tout faire dual.

```
select sys_context('USERENV','IP_ADDRESS') from dual ;
select sys_context('USERENV','SESSION_USER') from dual ;
```