TP Appropriation de différents mécanismes de manipulation

1. Objectifs

Les objectifs du TP sont de différentes natures. Il s'agit cependant d'explorer différents mécanismes qui vont faciliter la manipulation d'une base de données. Les points abordés sont listés ici

- suppression d'objets d'un schéma de base de données (procédures PL/SQL)
- chargement de tables à partir de fichiers de données tabulés (format csv pour comma separated value) (utilitaire SQL Loader)
- restitution du contenu d'une table (utilisation adaptée de SQL en tant que DML)
- restitution du schéma d'une base de données en vue de migration de la base de données (paquetage PL/SQL DBMS_Metadata)
- facilités pour calculer le temps de réponse à une requête (variables d'environnement de l'interpréteur sqlplus, paquetage PL/SQL DBMS_Utility et classes JDBC dédiées)

2. Suppression du contenu d'un schéma de base de données

L'idée est de faire place nette au sein de votre schéma utilisateur. Vous exploiterez à cet effet la procédure PL/SQL de suppression de tables construite dans les précédents TP. Procédez par ordre, suivez la démarche énumérée ci-dessous :

1. affichez les objets de votre schéma et gardez une trace du résultat avec le mécanisme de "spool" qui vous permet de conserver au sein d'un fichier (portant une extension .lst) le contenu de la mémoire tampon (buffer).

   ```
   col object_name for a30
   col object_type for a30
   spool on
   spool fichier_objet
   select object_name, object_type from user_objects;
   spool off
   ```

2. analyser les résultats obtenus, quels sont les types d'objets qui peuplent votre schéma ? Accédez vous à tous les objets du schéma ?
3. exécuter la procédure de suppression des tables
4. afficher à nouveau les objets de votre schéma, quels sont les objets qui restent, quels sont ceux qui ont été supprimés ? Le résultat vous semble t'il cohérent par rapport à ce que vous en attendiez ? Quelle est la vue du méta-schéma à consulter pour vérifier que les contraintes portant sur les tabes ont également été supprimées ? Conservez le code de certaines de vos procédures (vue user_source) puis supprimer ces derniers éléments du schéma.

3. Construction d'un nouveau schéma de base de données

3.1 Première manipulation

Vous allez maintenant travailler sur un nouveau schéma de bases de données. Le schéma de la table commune vous est donné, vous créerez cette table. L'alimentation se fera au travers de l'utilitaire de chargement Oracle nommé SQL Loader. Vous disposez, pour ce faire, d'un fichier au format tabulé nommé Commune_Tuples.csv. Vous spécifierez un fichier de contrôle en vous aidant des exemples du Tp1 pour charger ces données. Quels sont les problèmes rencontrés ? Un fichier de contrôle réglant l'ensemble de ces problèmes vous sera proposé en cours de Tp.

3.2 Construction de l'ensemble des tables et chargement

Un fichier archive vous est fourni avec un script SQL précisant l'ordre et les ordres SQL nécessaires à la définition et au chargement de l'ensemble du schéma. Les fichiers de données et les fichiers de contrôle de chargement sont également présents dans l'archive. Un modèle conceptuel vous est donné qui retranscrit les entités considérées dans le schéma.

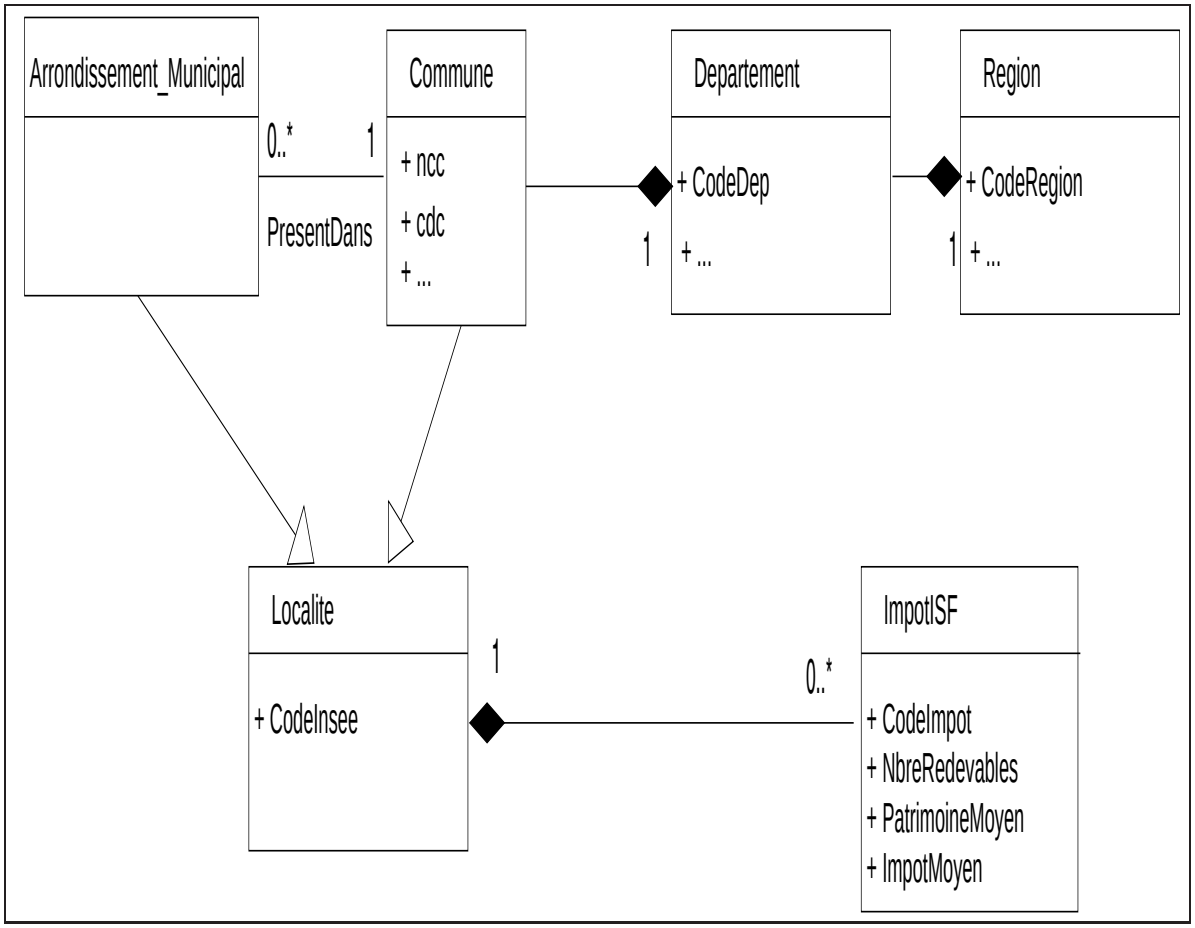

Figure 1 – Diagramme de classes UML

- proposez quelques requêtes qui vous semblent pertinentes. Par exemple, les informations concernant l'imposition sur l'ISF pour les plus grandes villes : Marseille, Paris et Lyon ne sont connues qu'au travers des arrondissements municipaux. Ecrivez les ordres SQL qui vont permettre de calculer et de rajouter cette information.
- exploiter différents mécanismes qui permettent d'avoir une première idée du temps de réponse pour une requête donnée
  - variable d'environnement de sqlplus : timing (set timing on). L'ensemble des variables d'environnement disponibles au sein du module sqlplus est listé au travers de la commande show all
  - exloitation du paquetage dbms_utility et de la fonction get_time qui retourne les centièmes de seconde écoulés depuis un instant t arbitraire. Il est possible d'obtenir une description du contenu du paquetage dbms_utility avec la commande desc dbms_utility
    Des exemples d'utilisation de get_time sont donnés ci-dessous. Vous les adapterez à d'autres requêtes.

```
DECLARE
-- temps en centiemes de sec.
   l_start NUMBER DEFAULT dbms_utility.get_time;
BEGIN
   FOR i IN
        (SELECT ncc
         FROM   cog)
 LOOP
        dbms_output.put_line ('name    : '|| i.ncc);
   END LOOP;
   dbms_output.put_line(round( (dbms_utility.get_time-l_start)/100, 2 ) || ' seconds...');
END;
/

SELECT hsecs, dbms_utility.get_time
FROM   v$timer;
```

- Exploiter la vue statique dba_objects pour vous informer sur les objets de type package dont le nom débute par "DBMS".

4. Exploiter différents paquetages prédéfinis de PL/SQL

4.1 Migration de schémas et le paquetage DBMS_METADATA

La librairie DBMS_METADATA permet, entre autres, d'obtenir le schéma d'une table au travers de son ordre de creation en SQL ou de sa définition en XML.

```
set long 4000000

-- ordre de creation de l'usager user1
select dbms_metadata.get_ddl( 'USER', 'USER1' ) from dual;

-- ordre de creation d'une table nommee cog dans le schema utilisateur
```

```
select dbms_metadata.get_ddl('TABLE','COG','USER1')
from dual;

select dbms_metadata.get_xml('TABLE','COG','USER1')
from dual;
```

Il est également possible d'en préciser le mode d'affichage au travers d'une autre procédure du paquetage DBMS_METADATA nommée SET_TRANSFORM_PARAM et dont la signature est la suivante SET_TRANSFORM_PARAM(transform_handle, name, value). D'autres paramétrages vont sont donnés dans les scripts exemples.

```
set long 4000000

execute DBMS_METADATA.SET_TRANSFORM_PARAM(DBMS_METADATA.SESSION_TRANSFORM,'PRETTY',true);
execute dbms_metadata.set_transform_param(dbms_metadata.session_transform,'SQLTERMINATOR',true);
execute dbms_metadata.set_transform_param(dbms_metadata.session_transform,'STORAGE',false);

-- ordre de creation d'une table nommee cog dans le schema utilisateur
select dbms_metadata.get_ddl('TABLE','COG','USER1')
from dual;
```

Vous manipulerez ces différentes commandes sur les différentes tables de votre schéma. Réfléchissez à comment exploiter le fonction dbms_metadata.get_ddl et la vue du méta-schéma user_tables pour restituer tout le schéma utilisateur.

5. Procéder à un export des données

Des modules d'import/export sont en général rendus disponibles au sein des SGBDs. Pour Oracle, il s'agit des modules exp et imp que nous n'utiliserons pas ici. L'idée, lorsque l'on procède à de la migration de bases de données, est de pouvoir gérer ce qui est désigné par dump (chargement) des tables. Nous allons simplement aborder l'export des données au format tabulaire avec une commande select qui retourne toutes les données d'une table et une utilisation appropriée des variables d'édition de sqlplus. Le paquetage utl_file donne des fonctionnalités pour la création, l'ouverture, l'écriture et la lecture dans des fichiers externes à la base de données. Nous ne l'utiliserons pas ici.

```
set pages 0
set feedback off
set heading off
set trimspool off
set termout off
set verify off
set colsep ""
set tab off
set linesize 100

SPOOL ON
SPOOL file_out
select *  from region ;
SPOOL OFF
```

Vous réfléchirez à partir de l'exemple suivant à construire une procédure qui permet de prendre en charge un caractère de séparation en bonne et due forme pour le fichier résultat (tabulation, ;, ...).

```
set serveroutput on

create or replace procedure factory_region is
cursor reg is select * from region;
begin
for reg_t in reg
loop
dbms_output.put_line(reg_t.region||chr(9)||reg_t.ncc||chr(9)||reg_t.cheflieu) ;
end loop ;
exception
when others then dbms_output.put_line('Pb sur l''affichage ') ;
end ;
/

spool on
spool file_reg
execute factory_region;
spool off
```

Vérifier la validité de votre fichier de sortie en chargeant son contenu tabulé dans une table créée pour l'occasion.