

TP2 : PL/SQL (séance 2)

1. Schéma de base de données

Nous allons continuer à travailler sur la base de données `Communes Françaises`

2. Constructions de fonctions

Vous définirez une fonction de calcul de distance kilométrique. Vous gérerez pour cette fonction différentes exceptions de manière à prévenir une grande majorité des erreurs pouvant impacter le schéma de la table commune. Le calcul de la distance en kilomètres entre deux points a et b ayant respectivement une latitude et une longitude notées `lat_a`, `long_a`, et `lat_b`, `long_b` (référentiel WGS84) se fait de la manière suivante :

```
6366*acos(cos(radians(lat_a))*cos(radians(lat_b))*cos(radians(long_b)-radians(long_a))
+sin(radians(lat_a))*sin(radians(lat_b)));
```

Listing 1 – distance Km

Les valeurs des latitudes et longitudes de la table `Commune` sont en degrés. Il vous faudra convertir les degrés en radians en définissant la fonction `radians` qui effectue le calcul :

```
valeurEnDegre/57.295779513082;
```

Listing 2 – degre vers radian

Vous pouvez construire différentes variantes de la fonction, avec une fonction qui prend pour arguments d'entrée les deux noms de communes considérées.

3. Constructions de paquetages

Vous construirez deux paquetages :

1. un paquetage nommé `UrbanUnits` qui visera à proposer des fonctions et procédures pour mieux caractériser les communes, les départements et les régions. La fonction de calcul kilométrique viendra se placer dans ce paquetage. Vous pourrez aussi y définir différentes fonctions de comptage des populations. Vous proposerez des exemples d'utilisation du paquetage ainsi construit.
2. un paquetage nommé `Supervision` qui visera à proposer des fonctions et procédures pour "monitorer" la base de données. Notamment vous exploiterez les vues du méta-schéma pour :
 - (a) manipuler les vues structurelles `user_constraints`, `user_tab_columns` et `user_cons_columns` pour définir une procédure qui renvoie pour une table donnée, toutes les informations sur ses attributs, le type de ces attributs, les contraintes qui s'appliquent éventuellement sur

ces attributs et le type de ces contraintes. Un exemple de ce qui est attendu est donné ci-dessous :

```
COLUMN_NAME      DATA_TYPE      CONSTRAINT_NAME  C
-----
NUM               NUMBER         EMP_PK          P
N_SUP            NUMBER         EMP_FK1         R
EMBAUCHE         DATE
COMM             NUMBER
NOM              VARCHAR2
SALAIRE          NUMBER
N_DEPT           NUMBER
FONCTION         VARCHAR2

8 rows selected.
```

FIGURE 1 – Exploiter la fonction

- (b) Vous écrirez une fonction qui renvoie les utilisateurs connectés (ainsi que le nom du terminal sur lequel ils sont connectés et le type de client qu'ils utilisent)
- (c) L'idée est, ici, d'exploiter la notion de SQL dynamique au travers de la commande `execute immediate`. Il s'agit de permettre l'exécution de requêtes faisant le lien avec des objets qui ne sont pas encore connus au moment de la compilation et qui ne le seront qu'en tout dernier lieu à l'exécution. Ainsi il sera rendu possible d'interroger les vues du méta-schéma et de supprimer, par exemple, à la volée toutes les tables, vues ou autres objets du méta-schéma. Vous construirez deux procédures : une procédure qui supprimera une table dont le nom sera passé en paramètre, une procédure qui supprimera toutes les tables du schéma. Vous pouvez choisir de travailler sur un autre objet de la base (vue, trigger, ...) si vous préférez conserver vos tables.