

# INSIA – SIGL 3

## Bases de données

### PL-SQL – ORACLE

#### TP 1

Bertrand LIAUDET

#### Exercice 0 : installation de l'environnement

- *Ouvrez une calculette SQL\*PLUS en tant que SYSDBA*

```
sqlplus / as sysdba
```

- *Affichez le nom de l'utilisateur*

```
show user
```

- *Affichez vos droits*

```
select * from user_role_privs;
```

- *Créez un utilisateur TPPLSQL qui pourra faire du PLSQL :*

```
CREATE USER tpplsql  
IDENTIFIED BY "tpplsql";  
GRANT CONNECT, RESOURCE TO tpplsql;
```

CONNECT est un rôle prédéfini qui permet la création de tables et donc leur modification, suppression, consultation.

RESOURCE est un rôle prédéfini qui permet de faire du PL-SQL : création de triggers et de procédures stockées.

- *Déconnectez l'utilisateur courant et connectez vous en tant que tpplsql*

```
connect tpplsql/tpplsql
```

- *Affichez le nom de l'utilisateur*

- *Affichez vos droits*

- *Affichez le répertoire courant pour la calculette.*

```
host cd
```

- *Fermez la calculette puis rouvrez la à partir du répertoire dans lequel vous mettrez vos scripts.*

Créez ce répertoire.

Créez un fichier .bat contenant la commande de lancement d'une calculette :

```
mode con cols=120  
mode con lines=500
```

```
set NLS_LANG=FRENCH_FRANCE.WE8PC850 // pour PC
sqlplus tpplsql / tpplsql
```

Créez un fichier login.sql dans lequel vous mettrez des valeurs pertinentes pour les variables pagesize et linesize :

```
set linesize 120
set pagesize 500
```

## Exercice 1 : prise en main du PL-SQL

### 01-bonjour.sql - Affichage

1. Exécuter le script 01-bonjour.sql dans la calculatrice sql\_plus.
2. Modifier le script pour que l’affichage se fasse.
3. Modifier le script pour que le temps d’exécution s’affiche.
4. Mettez à jour le fichier login.sql pour que l’affichage se fasse à chaque fois que vous vous connecterez.

### 02-variable.sql – empdept.sql - &

1. Exécutez le script empdept.sql dans la calculatrice sql\_plus.
2. Listez les tables de votre catalogue.
3. Exécutez le script 02-variable.sql
4. Exécutez une deuxième fois le script 02-variable.sql. Que constatez-vous ?
5. Regardez le code et explicitez toute la syntaxe.

### 03- AffEmpDept.sql (empdept) - Procédure

1. Exécutez le script 03-AffEmpDept.sql dans la calculatrice sql\_plus.
2. Affichez les erreurs repérées par le système  

```
Show errors;
```
3. affichez le code pris en compte.  

```
Col text format A80
Select line, text
From user_source
where name = 'AFFEMPDEPT';
```
4. Corrigez la ou les erreurs. Explicitez toute la syntaxe.
5. Appelez la procédures pour toutes les valeurs de département.
6. Mettez à jour la procédure pour qu’elle prenne correctement en compte le cas où le numéro de département n’existe pas.

### 04- moySalDept.sql (empdept) - Fonction

1. Exécutez le script 04-moySalDept.sql dans la calculatrice sql\_plus.
2. affichez le code pris en compte.  

```
Col text format A80
Select line, text
```

```
From user_source  
where name = 'MOYSALDEPT';
```

3. Explicitez toute la syntaxe.
4. Testez la fonction en utilisant la pseudo-table DUAL
5. Ecrire une requête avec une variable de substitution qui permette d'afficher, pour un département dont on saisira la valeur à l'exécution de la requête, tous les employés qui gagnent plus que la moyenne des salaires du département.

## 05- Curseurs

1. Exécutez les scripts 05-lesEmployes\_01.sql et 05-lesEmployes\_02.sql dans la calculatrice sql\_plus.
2. Comparez les résultats.
3. Comparez les codes des deux scripts.
4. Transformer les deux scripts en procédure, la procédure prenant le numéro du département en paramètre et l'affichage se limitant aux employés du département traité. Affichez un message spécial si le département est vide.

## 06- SQL dynamique - exercice

1. Ecrire une procédure (ou un bloc) qui permet d'afficher, pour chaque table de la BD, le nom de la table, le nombre de tuples, et les 5 premiers tuples de chaque table.

```
Select * from cat;  
Select * from emp where rownum <6;
```