Université de Provence
Licence d'Informatique
Troisième année

Bases de données
TP No 1
SQL_PLUS d'Oracle

1) Structured Query Language (SQL)

SQL est un :

- Langage de Définition de Données (LDD) : création et modification de la structure des Bases de Données
- Langage de Manipulation de Données (LMD) : insertion et modification des données des Bases de Données
- Langage de Contrôle des Données (LCD) : gestion de la sécurité, confidentialité et Contraintes d'Intégrité

Petit lexique entre le modèle relationnel et SQL :

Modèle relation	SQL
relation	table
attribut	colonne
tuple	ligne

2) SQL_PLUS

a) Configuration requise

Sous capucine, modifier votre fichier de configuration d'environnement (fichier .cshrc ou fichier .bash_profile) de la façon suivante :

- Dans le fichier .bash_profile que vous éditez avec emacs par exemple :
  - Ajouter le chemin '`/usr/local/oracle/bin`' à la variable d'environnement PATH :
    - `PATH=$PATH:/usr/local/oracle/bin`
  - Ajoutez les instructions suivantes qui permettent la définition de variable d'environnement nécessaires à la bonne exécution d'Oracle :
    - `ORACLE_HOME=/usr/local/oracle`
    - `LD_LIBRARY_PATH=$ORACLE_HOME/lib`
    - `ORACLE_DOC=$ORACLE_HOME/doc`
    - `ORACLE_SID=ens`
    - `export ORACLE_HOME LD_LIBRARY_PATH ORACLE_DOC ORACLE_SID`
- Dans le fichier .cshrc que vous éditez, ajouter les instructions suivantes :
  - `setenv ORACLE_HOME /usr/local/oracle`
  - `setenv LD_LIBRARY_PATH $ORACLE_HOME/lib`
  - `setenv ORACLE_DOC $ORACLE_HOME/doc`
  - `setenv ORACLE_SID ens`

b) Les trois commandes les plus utilisées

Lancement de SQL_PLUS	Quitter SQL_PLUS	Aide de SQL_PLUS

`sqlplus`
`Enter user-name:`
`Enter password:`
	`exit`	`Help <commande SQL>`

c) Editeur d'Oracle

Les instructions SQL sont mémorisées dans un buffer de travail (buffer SQL) que nous pouvons manipuler à l'aide des commandes suivantes :

- `l` Liste des lignes de la commande
- `l n` Affichage de la ligne `n`
- `l n m` Affichage des lignes `n` à `m`
- `l *` Affichage de la ligne courante

- `n` La ligne `n` devient la ligne courante
- `c/ch1/ch2` Changement de la chaîne `ch1` par la chaîne `ch2` dans la ligne courante
- `a ch` Ajout d'une chaîne `ch` à la fin de la ligne courante
- `i` Insertion après la ligne courante
- `del` Suppression de la ligne courante

d) Sauvegarde

- **`Save`** `<nom de fichier>` **`append`** Ajoute le contenu du buffer de travail à la fin du fichier
- **`Save`** `<nom de fichier>` **`[create]`** Sauvegarde le contenu du buffer de travail dans le fichier si ce fichier n'existe pas
- **`Save`** `<nom de fichier>` **`rep[lace]`** Sauvegarde le contenu du buffer de travail dans un fichier existant

e) Exécution des instructions SQL

Un commande SQL_PLUS est exécutée par :
- '`;`' en fin de commande
- '`/`' seul sur la ligne

Sinon, par défaut, l'éditeur vous propose de continuer la saisie de la commande.

Pour ré-exécuter les instructions du buffer de travail, vous pouvez utiliser :
- '`/`' pour l'exécution sans affichage des commandes
- '**`R[un]`**' pour l'exécution avec affichage des commandes

Pour charger dans le buffer de travail les instructions contenues dans un fichier :
- '**`get`** `<nom de fichier>`' pour charger le contenu du fichier dans le buffer de travail

Pour charger dans le buffer de travail et exécuter les instructions contenues dans ce fichier :
- '**`R[un]`** `<nom de fichier>`' pour charger et exécuter avec affichage des instructions
- '**`START`** `<nom de fichier>`' pour charger et exécuter sans affichage des instructions
- '`@ <nom de fichier>`' pour charger et exécuter sans affichage des instructions

Remarques : les commandes '`<host>`' et '`!`' permettent d'exécuter toutes les commandes du système d'exploitation (shell d'UNIX). Cela permet donc de lancer un éditeur de texte quelconque depuis le buffer SQL.

3) SQL LDD

a) Types syntaxiques (presque les domaines)

La notion de domaine n'est pas prise en compte dans SQL_PLUS. Il nous faut donc nous limites à la définition des types syntaxiques suivants :
- **`VARCHAR2`**`(n)` Chaîne de caractères de longueur variable (maximum `n`)
- **`CHAR`**`(n)` Chaîne de caractères de longueur fixe (`n` caractères)
- **`NUMBER`** Nombre entier (40 chiffres maximum)
- **`NUMBER`**`(n,m)` Nombre de longueur totale `n` avec `m` décimales
- **`DATE`** Date (DD-MON_YY est le format par défaut)
- **`LONG`** Flot de caractères

b) Création de table

- **`CREATE TABLE`** `<nom de la table> (<nom de colonne> <type> [`**`NOT NULL`**`] [, <nom de colonne> <type>]…, [<contraintes>]…);`

Où `<contraintes>` représente la liste des contraintes d'intégrité structurelles concernant les colonnes de la table crée. Elle s'exprime sous la forme suivante :
- **`CONSTRAINT`** `<nom de contrainte> <sorte de contrainte>`

Où <sorte de contrainte> est :

- **PRIMARY KEY** (attribut1, [attribut2…])
- **FOREIGN KEY** ((attribut1, [attribut2…]) **REFERENCES** <nom de table associée> (attribut1, [attribut2…])
- **CHECK** (attribut <condition>) avec <condition> qui peut être une expression booléenne « simple » ou de la forme **IN** (liste de valeurs) ou **BETWEEN** <borne inférieure> **AND** <borne supérieure>

c) Modification de la structure d'une table

Ajout de colonne :

- **ALTER TABLE** <nom de la table> **ADD** ([<nom de colonne> <type>][,<contrainte>][,<nom de colonne> <type>][,<contrainte>]…);

Modification de colonne :

- **ALTER TABLE** <nom de la table> **MODIFY** ([<nom de colonne> <nouveau type>][,<nom de colonne> <nouveau type>]…);

d) Destruction de table

- **DROP TABLE** <nom de la table>;

e) Consultation de la structure d'une base

- **DESCRIBE** <nom de la table>;

4) SQL LMD

a) Interrogation

- **SELECT [DISTINCT]** <nom de colonne>[,<nom de colonne>]…
- **FROM** <nom de la table>[,<nom de la table>]…
- **WHERE** <condition>
- **GROUP BY** <nom de colonne>[<nom de colonne>]…
- **HAVING** <condition avec calcul verticaux>
- **ORDER BY** <nom de colonne>[<nom de colonne>]…

b) Insertion de données

- **INSERT INTO** <nom de la table> [(colonne,…)] **VALUES** (valeurs,…)
- **INSERT INTO** <nom de la table> [(colonne,…)] **SELECT**_

c) Modification de données

- **UPDATE** <nom de la table> **SET** colonne=valeur,… **WHERE** <condition>
- **UPDATE** <nom de la table> **SET** colonne=**SELECT**_

d) Suppression de données

- **DELETE FROM** <nom de la table> **WHERE** <condition>

5) Prise en main

a. Lancer l'interprète SQL. Modifier le mot de passe (à noter !) en utilisant la commande `password`.

b. Créer, sous l'éditeur SQL_PLUS, la table correspondant à la relation :

`Pays(numPays,nom,nbHabitants,superficie)`

Pour cela, il faut utiliser la commande **`CREATE TABLE`**. Ne pas oublier de déclarer `numPays` comme clé primaire (**`CONSTRAINT PRIMARY KEY`**).

c. Créer, toujours sous l'éditeur SQL_PLUS, des données (des tuples). Pour cela, il faut utiliser la commande **`INSERT INTO`**.

d. Effacer cette table et ses données (commande **`DROP TABLE`** `<table>` **`CASCADE CONSTRAINTS`**).

e. Editer un fichier d'extension `.sql` contenant les commandes de créations effectuées dans les questions 2 et 3. Charger ce fichier sous SQL_PLUS en utilisant la commande **`START`** ou **`@`**.

f. Editer un autre fichier d'extension `.sql` qui contiendra les commandes de création de la table correspondant à la relation `Ville`, ainsi que la création de quelques tuples de cette table :

`Ville(numVille,nom)`

g. Charger ce dernier fichier sous SQL_PLUS. Modifier, sous l'éditeur SQL_PLUS et non pas dans le fichier, la table `Ville` en lui rajoutant une colonne correspondant au nombre d'habitants de cette ville (commande **`ALTER TABLE`**). Modifier les données en conséquence (commande **`UPDATE`**).

h. Quelle commande utiliser pour obtenir la description de la table Pays ?

i. Quelle commande utiliser pour obtenir la liste des tuples de la table Pays ?

j. Modifier, sous l'éditeur SQL_PLUS, la table Ville en lui rajoutant une colonne `refPays` dont les valeurs seront des clés vers la table `Pays`. Pour cela, il faut ajouter aussi une contrainte **`CONSTRAINT FOREIGN KEY`**.

k. Modifier les données de la classe `Ville` en conséquence.