

## SQLPLUS

SQL Plus, est une interface qui permet d'interagir en direct avec des bases de données Oracle, c'est à dire manipuler et de soumettre des requêtes SQL en direct. On peut par exemple :

- *Editer* (saisir, modifier, enregistrer, retrouver) et *exécuter* des commandes SQL.
- *Consulter les informations* sur les tables (savoir quels sont les schémas de relation)
- *Copier des données* entre des bases de données SQL
- *Manipuler le résultat* d'une requête, le sauvegarder ou l'imprimer sous forme de rapport

### 1. Connexion

---

Pour travailler avec SQLPlus en mode commande (fenêtre DOS) :

1. Commencez par ouvrir l'invite de commande. Sur Windows, tapez « cmd » dans le menu Démarrer ou sur l'écran d'accueil et choisissez « Invite de commande ».
2. La commande de connexion à SQLPlus est: sqlplus, suivie du nom d'utilisateur, du mot de passe. On peut résumer la commande ainsi :

```
SQL> sqlplus nom_utilisateur/mot_de_passe
```

### 2. Gestion des utilisateurs

---

Pour connecter à une BD Oracle, il faut se connecter en administrateur et créer un compte utilisateur (nom et mot de passe) avec les lignes:

```
SQL> connect / as sysdba
```

1. La commande suivante permet de créer un utilisateur :

```
SQL> create user utilisateur_x identified by mot_de_passe;
```

2. Pour changer le mot de passe d'un utilisateur voici la commande que vous devrez utiliser :

```
SQL> ALTER USER utilisateur_x IDENTIFIED BY nouveau_mot_de_passe ;
```

3. Pour affecter des droits à un utilisateur :

```
SQL> GRANT privilege ON objet TO utilisateur_x ; /*donne un privilège à un utilisateur_x sur l'objet mentionné parmi les privilèges : insert, select, update,delete, alter */
```

```
SQL> GRANT ALL ON objet TO utilisateur_x ; /*donne tous les privilèges à utilisateur_x */
```

```
SQL> GRANT privilege ON objet TO PUBLIC /*donne droit à tout le monde*/
```

```
SQL> REVOKE privilege ON objet FROM utilisateur_x ; /*révocation des droits de manière identique au GRANT
```

4. Pour supprimer un utilisateur :

```
SQL> drop user utilisateur_x;
```

## Exemple :

---

Exécutez les commandes suivantes pour créer un utilisateur Ali et lui affecter quelques droits :

```
SQL> create user Ali identified by passali;      /* user 'Ali' de mot de passe  
'passali'  
SQL> grant connect, resource, create view to Ali;      /* affectation de  
droits */
```

Vous pouvez vérifier alors que vous pouvez créer une table (en mode admin), par exemple :

```
SQL> create table t (a integer);
```

NB : Ne pas oublier de faire "drop table t ;

Plus tard, quand vous (re)lancez SQLPlus, tapez la commande suivante pour se connecter à Oracle en tant qu'utilisateur "Ali" de password "passali" :

```
SQL> connect Ali/passali      /* sans ; en fin de ligne*/
```

Exécutez la commande suivante pour supprimer l'utilisateur « Ali » :

```
SQL> drop user Ali;
```

## 3. Les commandes de type DDL et DML

---

On peut utiliser SQLPLUS pour exécuter des commandes SQL de type DDL (création de tables) et DML (recherche, mises-à-jour, ...).

On peut entrer directement les commandes en les tapant sous SQLPLUS, mais en cas de faute de frappe ou d'erreur, il est difficile de corriger le texte. Il est donc fortement recommandé de procéder de la manière suivante:

1. Avec *bloc-notes* ou *notepad++*, tapez votre(vos) commande(s), et sauvegardez-la(les) dans un fichier (par exemple, entrez `create table t (a integer);` dans *bloc-notes* ou *notepad++* et enregistrez ce texte dans le fichier `TP.sql`).
2. Sous **SQLPLUS**: demandez l'exécution du fichier `TP.sql` comme suit:

```
SQL> @EmplacementDuFichier/TP
```

3. Si quelque chose cloche, corrigez avec *bloc-notes* ou *notepad++*, enregistrez le fichier, et ré-exécutez-le.

Créez ainsi un fichier pour vos commandes: cela vous permettra de ne rien perdre et d'avoir un minimum de frappe clavier.

Sous **SQLPLUS**, vous avez quelques commandes utiles dont voici une brève liste.

```
SQL> desc tab      // Donne le schema de la table 'tab'.  
SQL> select table_name  
      from user_tables;      // Liste des tables que vous avez creees.  
SQL> list      // Affiche le dernier ordre execute  
SQL> spool fic      // Copie l'affichage a l'ecran dans 'fic.lst'  
SQL> spool off      // Stoppe la copie dans 'fic.lst'  
SQL> exit;      // Sortir de SQLPLUS
```

## Exemple :

---

Les commandes suivante permettent la création de la base de données "Officiel des spectacles".

1. Avec *bloc-notes* ou *notepad++*, tapez et sauvegardez ces commandes dans un fichier TP.sql.
2. Sous **SQLPLUS**: exécutez le fichier *TP.sql*.

```
CREATE TABLE Artiste (Nom VARCHAR2 (20) NOT NULL,
                      Prenom VARCHAR2 (15),
                      Annee_naissance NUMBER(4) ,
                      PRIMARY KEY (Nom));

CREATE TABLE film (ID_film          NUMBER(10) NOT NULL,
                   Titre             VARCHAR2(30),
                   Annee              NUMBER(4),
                   Nom_Realisateur VARCHAR2(20),
                   PRIMARY KEY (ID_film),
                   FOREIGN KEY (Nom_realisateur) REFERENCES Artiste);

CREATE TABLE Role (Nom_role        VARCHAR2(20) NOT NULL,
                   ID_film          NUMBER (10) NOT NULL,
                   Nom_acteur       VARCHAR2 (20) NOT NULL,
                   PRIMARY KEY (ID_film, nom_acteur),
                   FOREIGN KEY (ID_film) REFERENCES Film
                   ON DELETE CASCADE,
                   FOREIGN KEY (Nom_acteur) REFERENCES Artiste
                   ON DELETE CASCADE);

CREATE TABLE cinema (Nom_cinema    VARCHAR2 (10) NOT NULL,
                      Arrondissement NUMBER (2),
                      Adresse        VARCHAR2 (30),
                      PRIMARY KEY (Nom_cinema));

CREATE TABLE salle (Nom_cinema    VARCHAR2(10) NOT NULL,
                     No_salle      NUMBER(2) NOT NULL,
                     Climatise     CHAR(1),
                     Capacite       NUMBER(4),
                     PRIMARY KEY (Nom_cinema, No_salle),
                     FOREIGN KEY (Nom_cinema) REFERENCES cinema
                     ON DELETE CASCADE);

CREATE TABLE seance (Nom_cinema    VARCHAR2(10) NOT NULL,
                     No_salle       NUMBER(2) NOT NULL,
                     No_seance      NUMBER(2) NOT NULL,
                     Heure_debut    NUMBER (4,2),
                     Heure_fin      NUMBER (4,2),
                     ID_film        NUMBER(10) NOT NULL,
                     PRIMARY KEY (Nom_cinema, No_salle, No_seance),
                     FOREIGN KEY (Nom_cinema, No_salle) REFERENCES salle
                     ON DELETE CASCADE);
```

Les commandes DROP TABLE permettent d'exécuter le fichier de création plusieurs fois de suite, en détruisant d'abord les tables éventuellement créés lors des exécutions précédentes.

Attention : Il faut détruire les tables dans l'ordre inverse de création, afin de ne pas violer les contraintes de FOREIGN KEY.

```
DROP TABLE seance;  
DROP TABLE salle;  
DROP TABLE cinema;  
DROP TABLE role;  
DROP TABLE film;  
DROP TABLE artiste;
```

### Remarques importantes :

- TOUTES LES COMMANDES SQL DOIVENT SE TERMINER PAR UN ';'. Si vous oubliez le ';', une ligne '2' vous est proposée. Dans ce cas tapez un ';' pour finir la commande.
- Les chaînes de caractères s'écrivent avec une simple quote: 'texte' et pas ``texte''.
- Les majuscules et les minuscules sont interprétées différemment. Par exemple 'texte' est considéré comme différent de 'Texte' ou 'TEXTE'. Pensez-y en faisant des sélections! Un moyen d'éviter les problèmes est d'utiliser la fonction UPPER qui met tout en majuscule. Par exemple:  

```
select * from film where UPPER(titre) = 'VERTIGO';
```
- On ne peut pas utiliser de sous-requêtes dans une clause CHECK.