

TP1 Memento pour bien démarrer avec Oracle (HAI725I)

1. Documentation SQL et environnement SQLPLUS d'Oracle

1.1 Résumé des commandes utiles pour la série de TP

Dans la suite du support, les paramètres sont en minuscules et les mots clés en majuscules, mais il s'agit d'une simple convention d'écriture, `sqlplus` acceptant aussi les mots clés en minuscules. Le client `sqlplus` est à fois un interpréteur de commandes SQL et un éditeur de commandes. à ce titre, il possède également ses propres commandes comme les commandes suivantes (plus de détail plus bas) :

- `desc nomTable`
- `set linesize 160`
- `show all`
- `show user`
- `column nomColonne format a20`
- ...

Il est à noter que les commandes de l'éditeur n'ont pas besoin de se terminer par ";" Parmi les commandes SQL les plus utiles :

- `SELECT * FROM CAT;` – pour voir les objets de la base, CAT pour CATALOGUE

1.2 SQL en tant que LDD (Langage de Définition des Données)

1.2.1 Création de table

```
CREATE TABLE nomTable (col1 type1 [contrainte1], col2 type2 [contrainte2],  
[contrainteTable] ...);
```

- Le type de données pouvant être `number(n)`, `char(n)` ou mieux `varchar(n)`, `date`, `long`, `integer`, `float`, ...
- Une contrainte de colonne pouvant être prise parmi : `UNIQUE`, `PRIMARY KEY`, `NOT NULL`, `REFERENCES nomtable2(colsource) ON DELETE CASCADE`, `DEFAULT valeurpardéfaut`, `CHECK(condition)`
- une contrainte de table pouvant être `PRIMARY KEY(col1, col2)`, `CHECK()`, ...
- une contrainte peut être nommée en la préfixant de `CONSTRAINT nomContrainte`

1.2.2 Création de table à partir d'une requête

```
CREATE TABLE nomtable AS SELECT ...;
```

1.2.3 Evolution du schéma d'une table

`ALTER TABLE` `nomtable` `ADD` `nomcol1` `type1` ;

`ALTER TABLE` `nomtable` `ADD` `contraintel1` `expression_contrainte` ;

`ALTER TABLE` `nomtable` `MODIFY` `nomcol2` `type2` ;

`ALTER TABLE` `nomtable` `DROP` `nomcontrainte` ;

`ALTER TABLE` `nomtable` `ENABLE` `nomcontrainte` ;

`ALTER TABLE` `nomtable` `DISABLE` `nomcontrainte` ;

1.2.4 Suppression d'une table

`DROP TABLE` `nomtable`;

1.2.5 Renommage d'une table

`RENAME` `ancienobjet` `TO` `nouvelobjet`;

1.2.6 Création d'un index

`CREATE INDEX` `nomindex` `ON` `nomtable`(`nomcol1` `ASC`, `nomcol2` `DESC`);

1.2.7 Suppression d'un index

`DROP INDEX` `nomindex`;

1.2.8 Création d'une vue

`CREATE VIEW` `nomvue` `AS SELECT` ...;

1.2.9 Suppression d'une vue

`DROP VIEW` `nomvue`;

1.3 SQL en tant que LMD (Langage de Manipulation des Données)

1.3.1 Consultation

`SELECT` [`ALL`|`DISTINCT`] `nomcol1`, `nomcol2`, ... `FROM` `table1`, ... `WHERE` `condition1`
`GROUP BY` `nomcol3`, `nomcol4` `HAVING` `condition2`;

La condition du `WHERE` peut être une requête imbriquée

1.3.2 Insertion

`INSERT INTO` `nomtable` `VALUES` (`valeur1`, `valeur2`, ...);

1.3.3 Insertion avec requête associée

`INSERT INTO` `nomtable` `SELECT` ...;

1.3.4 Mise à jour

`UPDATE` `nomtable` `SET` `nomcol` = `expression` `WHERE` `condition`;

1.3.5 Suppression de tuples

`DELETE FROM` nomtable `WHERE` condition ;

1.3.6 Dictionnaire des Données

ORACLE, tout comme l'ensemble des SGBD commerciaux les plus connus, maintient une metabase qui est consultable au travers de vues diverses notamment :

Les vues des objets de l'utilisateur avec pour préfixe `USER_`
 Les vues des objets de tous les utilisateurs avec pour préfixe `ALL_`
 Les vues des objets de toute la base avec pour préfixe `DBA_`

1.4 L'interface sqlplus

Nous listons ci-dessous certaines commandes spécifiques de `sqlplus`. Elles ne sont pas à confondre avec des instructions du standard SQL et ne sont d'ailleurs pas à ponctuer par un `;` (à la différence des instructions SQL). Elles possèdent par ailleurs des abréviations (indiquées ci-dessous entre parenthèses) et permettent surtout de mieux exploiter l'environnement d'édition de `sqlplus`.

- `DESCRIBE` (`DESC`) nomtable : décrit la structure de la table
- `SHOW` : visualise la valeur d'une variable d'environnement de l'éditeur. `SHOW ALL` permet de lister l'ensemble des variables disponibles ainsi que leurs valeurs par défaut. `SHOW linesize` donne la longueur de la ligne (par défaut 80 colonnes).
- `SET` : permet de changer la valeur d'une variable. Par exemple `SET linesize 120`, étendra la taille de la ligne.
- `DEFINE` (`DEF`) nomvar = valeur
permet notamment de faire `DEF _EDITOR = gedit`
- `EDIT` (`ED`) pour éditer le buffer de commandes avec l'éditeur retenu (ici `gedit`) puis sauver et quitter. Il faut avoir fait le `DEFINE` précédent !
- `RUN` (`R`) pour exécuter le buffer éventuellement modifiée par `EDIT`.
- `/` pour exécuter le buffer sans afficher son contenu.
- `HOST` (!) pour sortir temporairement de `sqlplus` et pouvoir exécuter des commandes Unix ; on retourne à `sqlplus` par `exit`.
- `SPOOL` fichier ...`SPOOL OFF`
pour enregistrer la trace d'une session dans fichier.lst
- `SAVE` requete1
pour enregistrer la dernière requête (buffer) dans un fichier requete1.sql.
- `START` requete1 (@)
pour charger dans le buffer puis exécuter requete1.sql (`GET` puis `RUN`)
- `INPUT` ... (ligne vide)
pour ajouter une ou plusieurs lignes de clauses au buffer courant. On peut utiliser `SAVE` à la fin pour sauver.
- `DELETE` (`DEL`) pour effacer la ligne courante (dernière) du buffer
- `LIST` [n] pour lister les(a) lignes(n) du buffer
- `HELP` sujet pour obtenir l'aide.
- `QUIT`|`EXIT` pour quitter `sqlplus`.

Exemple de spool :

```
spool on
select nomc from client where solde<0;
spool off
```

permet de sauver dans le fichier cn.lst tout ce qui s'est passé entre "spool on et spool off" (requêtes et résultats).