

# Chapitre 6 : Initiation Oracle © INF3080 BASES DE DONNÉES (SGBD)

Guy Francoeur

Aucune reproduction sans autorisation

3 septembre 2019

**UQÀM** | **Département d'informatique**

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL

- ▶ Les droits de lecture sont accordés aux étudiants inscrits au cours INF3080-030 A2019 uniquement;
- ▶ Aucun droit pédagogique ou reproduction n'est accordé sans autorisation;

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL

- ▶ Besoin de 4 personnes pour partager les solutions aux questions de l'atelier sur l'algèbre relationnelle;
- ▶ Questions !

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL

# SQL\*Plus - Origine et définition

- ▶ System R : Relationnal Approach to Database Management
  - ▶ M. M. Astrahan et al., 1976
  - ▶ IBM Research Laboratory (San Jose, CA)
- ▶ Specifying queries as relational expressions: the SQUARE data sublanguage
  - ▶ R. F. Boyce et al., 1975
  - ▶ IBM Research Laboratory (San Jose, CA) + 1 MIT
- ▶ SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control
  - ▶ D. D. Chamberlin, M. M. Astrahan, K. P. Eswaran, P. P. Griffiths, R. A. Lorie, J. W. Mehl, P. Reisner, B. W. Wade,

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL



- ▶ SQL\*plus est une application qui permet de connecter le SGBDR Oracle;
- ▶ Il est un terminal similaire a ssh ou putty;
- ▶ Une fois connecter vous pouvez écrire vos des requêtes SQL;
- ▶ SQL\*plus est unique à Oracle;

- ▶ Le SGBDR est un service, un processus, qui s'exécute en mémoire dans un ordinateur/serveur;
- ▶ Le processus SGBDR écoute sur un port précis;
- ▶ En plus du SGBDR il y a un autre service qui s'appelle le LISTENER;
- ▶ Le rôle du LISTENER est d'accueillir les connexions des applications;
- ▶ C'est une architecture modulaire;
- ▶ *AppJava* → *Listener* → *SGBDR* → *donnes*
- ▶ *SQL \* plus* → *Listener* → *SGBDR* → *donnes*

# Configuration

Un aperçu du tnsnames.ora :

```
1 # tnsnames.ora Network Configuration File:
2 # /u01/oracle/product/11.2/db_1/network/admin/tnsnames.ora
3 # Generated by Oracle configuration tools.
4
5 BACLAB =
6   (DESCRIPTION =
7     (ADDRESS = (PROTOCOL = TCP)(HOST = zeta2.ens.uqam.ca)(PORT =
8       1521))
9     (CONNECT_DATA =
10      (SERVER = SHARED)
11      (SERVICE_NAME = baclab)
12    )
13  )
```

```
$ sqlplus --help
```

```
1 {<username>[/<password>][@<connect_identifier>] | / }  
2 [AS {SYSDBA | SYSOPER | SYSASM}] [EDITION=value]
```

```
$ sqlplus codems/password@BACLAB
```

- ▶ BACLAB est nommé serviceName ou *TNSnames-alias*
- ▶ Connectons-nous !

# SQL\*plus - environnement

```
--pour changer l'éditeur par default (facultatif)
DEFINE _EDITOR = nano

--Éditer la dernière commande (Connaissance Vi exigé):
edit

--Executer la dernière commande:
run

--Lister la dernière commande:
list

--Effacer l'écran:
clear screen

--Executer un script
@/home/folder/script.sql
```

# SQL\*plus - environnement

```
--Exécuter des commandes système (linux):  
host ls -lah  
  
--Démarrer le spool (capture) dans un fichier:  
spool file.txt  
  
--Fermer le fichier (sauvegarde):  
spool off  
  
--pour permettre une sortie (output) sur une meme ligne:  
set linesize 32767  
  
--supression des entetes a chaque page size  
-- (entete sera affichee une seul fois)  
set pagesize 10000  
  
--cas particulier de formatage de colonne de type LONG:  
set long 4000  
set longchunksize 4000  
  
--Pour éditer vos fichiers .sql .txt (si nano est disponible):  
SQL> ho nano exemple.sql
```

Un peu de magie. Il est possible de préparer votre environnement, le mettre à votre goût :

- ▶ SQL\*plus exécute automatiquement **login.sql** au lancement;
- ▶ login.sql doit être présent (droits) dans répertoire de lancement;

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL



# Le premier objet

- ▶ DUAL est un objet utile afin de respecter la norme ISO:SQL86+;
- ▶ Il existe un formalisme d'une requête SQL DML;
- ▶ `SELECT a/v/f FROM t/v/s WHERE c GROUP BY a HAVING c ORDER BY n/a;`
- ▶ deux instructions sont obligatoires, les autres sont facultatives;

```
SQL> SELECT * FROM DUAL;
```

# Les objets Oracle

```
SQL> SELECT OBJECT_TYPE FROM ALL_OBJECTS;
```

- Tous les objets qui composent, définissent, configurent le SGBDR sont inclus dans le SGBDR;

```
SQL> SELECT COUNT(*) FROM ALL_OBJECTS;
```

```
SQL> SELECT OBJECT_NAME  
FROM ALL_OBJECTS  
WHERE OBJECT_NAME LIKE 'USER_%';
```

# Les objets Oracle

Les objets les plus utiles, surtout dans le cadre de ce cours, sont surement ceux qui listent vos propres objets :

- ▶ USER\_TABLES, USER\_VIEWS
- ▶ USER\_INDEXES, USER\_SEQUENCES, USER\_PROCEDURES, ...

```
--Pour lister les tables (ou vues) contenues dans mon compte
Oracle :
SQL> SELECT TABLE_NAME FROM user_tables;
SQL> SELECT VIEW_NAME FROM user_views;
```

```
1 #Pour obtenir la liste des attributs d'une table ou vue:
2 SQL> DESC <nom entite>
```

```
--Exemples
SQL> DESCRIBE ALL_OBJECTS
SQL> DESC user_tables
```

Nous parlons de table de vue, ouille, mais c'est quoi tout ça ?

- ▶ Une table est un objet dans lequel nous pouvons insérer des tuples;
- ▶ *tuples*  $\equiv$  *rows*  $\equiv$  *lignes*;
- ▶ Une vue est une **projection (AR)** d'une vue ou table;
- ▶ La vue est construite à l'aide d'une requête SQL;
- ▶ Il est possible d'appliquer une **sélection (AR)** sur une vue construite;

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL

# DDL - Data Definition Language

**acronyme** LDD - Langage de définition de données;

**définition** Il est utile pour créer, supprimer ou modifier des objets dans le dictionnaire;

► CREATE

► DROP

► ALTER

► RENAME

```
SQL> CREATE VIEW mes_objets as  
SELECT table_name  
FROM user_tables;
```

```
SQL> SELECT * FROM mes_objets;
```

```
SQL> SELECT * FROM mes_objets WHERE like 'C%';
```

# DDL - Data Definition Language

```
SQL> DROP TABLE Connexion;  
SQL> CREATE TABLE Connexion (  
pConnexion NUMERIC NOT NULL,  
cCourriel VARCHAR(100) NULL,  
cPassword VARCHAR(50) NULL  
);
```

```
SQL> ALTER TABLE Connexion ADD cCommentaire VARCHAR(100) NULL;
```

```
SQL> ALTER TABLE Connexion MODIFY cCourriel VARCHAR(200);
```

```
SQL> RENAME Connexion TO Login;
```

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
- 6. DML**
7. TCL



# DML - Data Manipulation Language

acronyme LMD - Langage de manipulation de données;

définition Il est utile pour insérer, supprimer ou modifier les données dans les tables de la base de données;

- ▶ SELECT
- ▶ INSERT
- ▶ DELETE
- ▶ UPDATE
- ▶ MERGE (UPSERT)

```
SQL> INSERT INTO Connexion  
(pConnexion, cCourriel)  
VALUES (1, 'bidon');
```

```
SQL> UPDATE Connexion SET cCourriel='lamiss@uqam.ca';
```

# DML - Data Manipulation Language

```
SQL> UPDATE Connexion SET cCourriel='lamiss@uqam.ca'  
WHERE pConnexion=1;
```

```
SQL> DELETE FROM Connexion  
WHERE pConnexion=2;
```

```
SQL> INSERT INTO Connexion  
(pConnexion, cCourriel)  
VALUES (1, 'un autre');
```

# Table des matières

1. Au dernier cours
2. Origine
3. SQL\*plus
4. Les objets Oracle
5. DDL
6. DML
7. TCL

# TCL - Transaction Control Language

**acronyme** LCT - Langage de contrôle de transaction;

**définition** Il est utile pour créer, supprimer ou modifier des objets dans le dictionnaire;

► COMMIT

► ROLLBACK

Le TCL *Transaction Control Language* est assez important en oracle. Une transaction débute à chaque fois qu'une écriture est effectuée.

# conclusion - résumé

