



## B.D. avancées (Oracle)

Pr. EL BELRHITI EL ALAOUI Abdelbaki  
a.elbelrhiti@fs-umi.ac.ma

Pr. EL BELRHITI EL ALAOUI

1

## Avant - propos

- Ce cours est suite logique du module BD du semestre 5 de la filière SMI.
- Avant d'aller plus loin dans Oracle (PL-SQL, clients/serveur, gestion des utilisateurs,...), le cours abordera les notions du langage SQL non traitées en S5 (séquences, agrégation, jointures, vues,... ).
- Les exemples de ce cours seront testés dans les BD créées en semestre 5.

2

## Notations

- { ... } : une liste d'éléments
  - | : un choix.
  - [ ... ] : un caractère optionnel d'une directive au sein d'une commande. Exemple :
- CREATE TABLE Avion (...) [ORGANISATION INDEX];

Pr. EL BELRHITI EL ALAOUI

3

## Plan

- I. Architecture d'oracle
- II. Prise en main de SQL plus
- III. SQL avancé
- IV. PL/SQL
- V. Contrôle de données

Pr. EL BELRHITI EL ALAOUI

4

## I. Architecture d'oracle

### I.A Généralités

### I.B Architecture

Pr. EL BELRHITI EL ALAOUI

5

## I.A Généralités

- Un SGBD Objet-relationnel multiplateforme (Linux, Windows, UNIX, Solaris (SUN), Aix (IBM), etc.)
- Les versions d'Oracle :
  - ✓ Oracle Server Enterprise Edition : toutes les fonctionnalités du logiciel.
  - ✓ Oracle Server Standard : limitée en nombre de processeurs.
  - ✓ Oracle database Express Edition : version bridée
    - Une seule BD avec une taille limitée.
    - Un seul processeur utilisé.
    - Plusieurs outils non disponibles.

Pr. EL BELRHITI EL ALAOUI

6

## Historique / Société

- 1977 : Software Development Laboratories (SDL)
- 1979 : Relational Software Inc. (RSI)
- 1983 : Oracle Corp.
- 2009 : Oracle achète Sun (Java) qui avait acheté MySQL en 2008.

Pr. EL BELRHITI EL ALAOUI

7

## Historique / Logiciel

- 1979 : Oracle 2, 1<sup>ère</sup> version du SGBD, écrit en assembleur
- 1983 : Oracle 3 (verrous,...) codée dans le langage C.
- 1984 : Oracle 4 (transactions...).
- 1986 : Oracle 5 (client-serveur,...).
- 1988 : Oracle 6 (sauvegarde/restauration, PL/SQL,...)
- 1992 : Oracle 7 (Contraintes référentielles, trigger, procédures stockées,...).
- 1997 : Oracle 8 (Objet-relationnel, Java,...).

Pr. EL BELRHITI EL ALAOUI

8

## Historique / Logiciel

- 2001 : Oracle9i (i comme Internet, Services Web, XML,...)
- 2004 : Oracle10g (g comme Grid computing).
- 2007 : Oracle11g (Amélioration des performances et de l'administration)
- 2009 : Oracle 11g Release 2
- 2013 : Oracle 12c (c pour Cloud computing).
- Depuis 2018, Oracle publie chaque année une nouvelle version de son SGBD, disponible sur sa plateforme Live SQL

Pr. EL BELRHITI EL ALAOUI

9

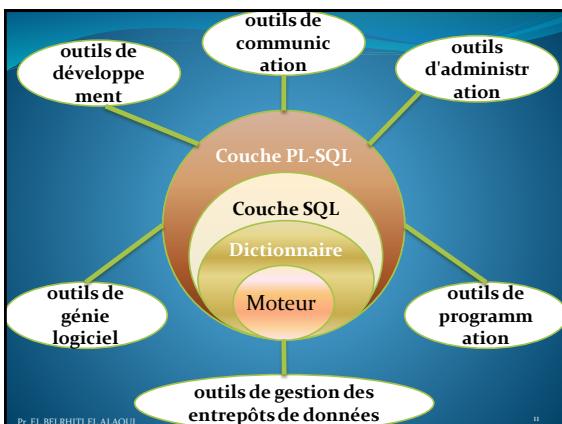
## I.B Architecture

- Autour du serveur de données, Oracle offre une multitude d'outils.

Evoluent en permanence → Site d'Oracle (Jungle)

Pr. EL BELRHITI EL ALAOUI

10



### I.B.1 Le serveur

#### Le moteur

- Le cœur du système.
- Assure, entre autres, les fonctions suivantes :
  - ✓ Stockage physique des données.
  - ✓ La cohérence des données.
  - ✓ L'intégrité des données.
  - ✓ La gestion des accès concurrents.

Pr. EL BELRHITI EL ALAOUI

12

## I.B.1 Le serveur

### *Le dictionnaire de données*

- Un ensemble de tables systèmes (une métabase) contenant les informations relatives à la structure de la BD.
- Créer au moment de la création de la BD.

Pr. EL BELRHITI EL ALAOUI

13

## I.B.1 Le serveur

### *Le dictionnaire de données*

- Décrit d'une façon dynamique les caractéristiques des objets de la BD :
- ✓ Tables, vues, index, clusters, triggers, packages,....
- ✓ Utilisateurs de la base (privileges, rôle,...).
- ✓ Contraintes d'intégrité (verreux,...).
- ✓ Ressources physiques allouées à la base.

Pr. EL BELRHITI EL ALAOUI

14

## I.B.1 Le serveur

### *La couche SQL*

- Interprète les commandes SQL.
- Optimisation de l'exécution des requêtes.

Pr. EL BELRHITI EL ALAOUI

15

## I.B.1 Le serveur

### *La couche PL-SQL*

- Une extension de la couche SQL.
- Le langage procédural d'Oracle qui permet d'incorporer nativement tout ordre SQL dans un programme.
- Un moteur qui peut exécuter des Blocs de commandes PL/SQL.

Pr. EL BELRHITI EL ALAOUI

17

## I.B.2 Les outils d'administration

### **Enterprise Manager (Database Control) :**

- Application web pour administrer graphiquement (sauvegardes, supervision, etc.) des BD Oracle locales ou distantes.

### **Locale Builder**

- Service de régionalisation (jeux de caractères, sens d'écriture, système monétaire,...)

Pr. EL BELRHITI EL ALAOUI

18

## I.B.2 Les outils d'administration

### **Assistant de Mise à niveau (DBUA, Database Upgrade Assistant)**

- Faire migrer une BD oracle vers une version supérieur du SGBD.

### **Assistant configuration de base de données**

- Créer, configurer et supprimer une BD Oracle.
- Gérer les modèles de BD.

Pr. EL BELRHITI EL ALAOUI

19

## I.B.2 Les outils d'administration

### Universal Installer

- Installer et Configurer les produits Oracle.

### SQL \*Loader

- Alimenter une BD à partir de fichiers externes.

Pr. EL BELRHITI EL ALAOUI

20

## I.B.3 Les outils de communication

### Oracle Net Manager

- Créer et configurer les processus d'écoute (Listener).
- Définir des alias aux descripteurs des connexions
- Résolution des noms.

### Wallet Manager

- Gérer les connexions sécurisées.

Pr. EL BELRHITI EL ALAOUI

21

## I.B.4 Les outils de développement

### SQL plus

- Console client historique.
- Disponible pour toutes les versions.
- Interface d'accès interactive en mode ligne de commandes
  - ✓ Exécution de commandes SQL et des scripts PL-SQL.
  - ✓ Administration en ligne.
  - ✓ Paramétriser l'environnement de travail (longueur d'une ligne, etc.).
  - ✓ Exécuter des commandes OS.

Pr. EL BELRHITI EL ALAOUI

23

## I.B.4 Les outils de développement

### SQL Developer

- Interface graphique développée en Java.
- Plusieurs extensions (Oracle ou sociétés tierces) peuvent y être intégrées.
- Permet, entre autres, de :
  - ✓ Naviguer dans les objets de la base.
  - ✓ Éditer, exécuter et déboguer les codes SQL et PL- SQL.
  - ✓ Administrer la BD.
  - ✓ Faire migrer les BD d'éditeurs tiers vers Oracle.

Pr. EL BELRHITI EL ALAOUI

24

## I.B.4 Les outils de développement

### ApEx (Oracle Application Express)

- Environnement de développement intégré à partir d'un navigateur web
- Permet de développer rapidement des applications exploitant des BD oracle.

### Oracle Forms

- Développer des applications graphiques (fenêtres, formulaires, ...) exploitant les données d'une BD Oracle sans écrire des requêtes SQL.

Pr. EL BELRHITI EL ALAOUI

26

## I.B.5 Les outils de programmation

### Les pré-compilateurs

- Des interfaces (Pro\*C, Pro\*Fortran, ...) permettant aux programmes de s'interfacer avec les BD Oracle.

## I.B.6 Les outils de génie logiciel

### Oracle Designer

- AGL (Atelier de Génie Logiciel, CASE : Computer Aided Systems Engineering).
- Réalisation des systèmes d'informations de la modélisation à la génération du code.

Pr. EL BELRHITI EL ALAOUI

28

## I.B.7 Les outils de gestion des entrepôts de données

### Warehouse Builder

- Outil décisionnel pour concevoir, implémenter, planifier des traitements d'ETL (Extraction-Transfer-Loading) d'un DataWarehouse.

Pr. EL BELRHITI EL ALAOUI

29

## I. Architecture d'oracle

### I.A Généralités

### I.B Architecture

Pr. EL BELRHITI EL ALAOUI

30

## II. Prise en main de SQL plus

### II.A Oracle Database Express Edition

### II.B Prise en main de SQL plus

Pr. EL BELRHITI EL ALAOUI

31

## II.A Oracle Database Express Edition

### Présentation

- Version bridée :
- ✓ Une seule instance de BD avec une taille limitée.
- ✓ Un seul processeur utilisé, même si on dispose de plusieurs.
- ✓ Plusieurs outils non disponibles.
- Une seule installation par machine.
- Peut cohabiter avec d'autres versions d'Oracle, Standard et Entreprise Editions.
- Idéal pour un premier contact :
- ✓ Rien (ou presque) à paramétrier.
- ✓ BD prête à l'emploi.

Pr. EL BELRHITI EL ALAOUI

32

## II.B Oracle Database Express Edition

Compte rendu de l'installation sous Windows

- Le serveur : oracle.exe
- Listener : TNSLSNR.exe
- Les composants installés :
  - ✓ Start/Stop Database.
  - ✓ Backup /Restore Database.
  - ✓ Get Started.
  - ✓ Run SQL Command line .

Pr. EL BELRHITI EL ALAOUI

34

## II.A Oracle Database Express Edition

Compte rendu de l'installation sous Windows

- Répertoires :
- ✓ Données (.DBF)  
C:\app\<username>\product\21c\oradata\XE
- ✓ Réseau:  
C:\app\<username>\product\21c\homes\OraDB21Home1\network
- Services :
- ✓ OracleServiceXE.
- ✓ OracleXETNSListener .
- ✓ OracleXEClrAgent.
- ✓ OracleMTSRecoveryService.

Pr. EL BELRHITI EL ALAOUI

35

## II.A Oracle Database Express Edition

### Désinstallation

1. Arrêter tous les services Oracle.
2. Désinstaller le produit à partir du panneau de configuration.
3. Supprimer les répertoires d'installation (C:\app)
4. Nettoyer les registres :
  - a. à l'aide d'un logiciel de nettoyage des registres.
  - b. Puis, supprimer toutes les entrées dans la base des registres faisant référence à oracle xe (regedit / F3).
5. Redémarrer

Pr. EL BELRHITI EL ALAOUI

36

## II-B. Prise en main de SQL plus

- Interface client en ligne de commandes permettant :
  - ✓ d'accéder au serveur → utilisateur valide.
  - ✓ de faire du SQL et du PL/SQL.
- Exemples de commandes SQLplus :
  - ✓ help index | nomCommande
  - ✓ host, spool, start,...
- Utilisateurs par défaut : SYSTEM et SYS
- connect & disconnect
- Requête test : select username from all\_users ;

Pr. EL BELRHITI EL ALAOUI

49

- Remarque SQLplus :
- ❖ Nécessite : BD et listener démarrés, connexions (SYSTEM) valide
  - ❖ SQLplus n'est pas sensible à la casse.
  - ❖ Commandes SQLplus :
    - ✓ describe NomTable; // visualiser une table
    - ✓ start, @ : pour exécuter un fichier.
    - ✓ show, set : pour voir et affecter des variables d'environnement
    - ✓ host : exécuter une commande SE
    - ✓ Save : enregistrer le buffer dans un fichier.
    - ✓ Get : pour lire un fichier dans le buffer
    - ✓ Exécuter un script :
      - @nomFichier : exécute le fichier .sql par défaut
      - PAUSE : pour arrêter l'exécution d'un script
    - ❖ spool :
      - Envoyer les résultats des requêtes vers un fichier ou l'imprimante.
      - lancer la sauvegarde : SPOOL nom\_fichier.txt
      - Arrêter la sauvegarde : SPOOL OFF

Pr. EL BELRHITI EL ALAOUI

50

➤ Remarque accès:

- L'utilisateur nouvellement créé n'a aucun droit : pas même celui de se connecter. Toutefois, il apparaît dans la liste des utilisateurs.
- CONNECT, rôle prédéfini, permet de se connecter au serveur puis d'accéder à des objets (tables, etc.) appartenant à d'autres utilisateurs à condition que les droits nécessaires aient été accordés.
- RESOURCE, rôle prédéfini, permet de créer et utiliser presque tous les objets ORACLE : tables, séquence, index, procédures, triggers, clusters.
- Tests :
 

```
select * from all_users;
select table_name from user_tables;
CREATE USER alaoui IDENTIFIED BY "alaoui";
GRANT CONNECT, RESOURCE TO alaoui;
connect alaoui/alaoui;
select table_name from user_tables;
```

Pr. EL BELRHITI EL ALAOUI

51

## III. Plus loin dans SQL

- ### III.A Séquences
- ### III.B Rupture du lien Père/fils
- ### III.C Vues
- ### III.D Agrégation
- ### III.E Jointure externe
- ### III.F Les opérateurs ensemblistes
- ### III.G Fonctions prédéfinies

Pr. EL BELRHITI EL ALAOUI

53

## III. Plus loin dans SQL

### III-A. Les séquences

- Une séquence:
  - ✓ un objet virtuel qui ne correspond à aucune information,
  - ✓ gérée indépendamment des tables,
  - ✓ peut être partagée par plusieurs utilisateurs,
  - ✓ utilisée pour générer automatiquement des entiers.

Pr. EL BELRHITI EL ALAOUI

54

## III-A. Les séquences

### III-A-1. Création d'une séquence

- Privilèges requis :
- ✓ Dans votre schéma : CREATE SEQUENCE.
- ✓ Autre schéma : CREATE ANY SEQUENCE.
- ✓ Remarque : Un schéma peut avoir plusieurs séquences..

Pr. EL BELRHTI EL ALAOUI

55

#### ➤ Syntaxe :

```
CREATE SEQUENCE [schéma.]nom_séquence
```

-- Par défaut, schéma courant

#### ✓ Options les plus utilisées :

[INCREMENT BY entier1 ]	-- Par défaut, 1
-------------------------	------------------

[START WITH entier2 ]	-- Par défaut, 1
-----------------------	------------------

#### ✓ Exemple :

```
CREATE SEQUENCE SeqTest
```

```
INCREMENT BY 2
```

```
START WITH 1
```

✓ Autres options existent : fixer une limite (supérieure ou inférieure), recycler des valeurs, etc.

Pr. EL BELRHTI EL ALAOUI

56

### III-A-2. Utilisation d'une séquence

- Uniquement deux fonctions (pseudo-colonnes ou directives) peuvent être appliquées à une séquence :
- ✓ CURRVAL : retourne la valeur courante de la séquence (lecture seule),
- ✓ NEXTVAL : incrémente la séquence et retourne la nouvelle valeur (écriture et lecture).
- Syntaxe : NomSéquence.NomFonction
- Exemples :
- ✓ INSERT INTO commandes(numcmd, datecmd, Bonus)  
VALUES (**SeqTest.NEXTVAL**, '15-02-2015' , 300);
- ✓ SELECT **SeqTest.curval** FROM dual -- La pseudo-table DUAL permet d'évaluer une expression et de l'afficher.

Pr. EL BELRHTI EL ALAOUI

57

### III-A-2. Utilisation d'une séquence

#### ➤ Les principales restrictions d'utilisation :

- ✓ dans un SELECT d'une vue
- ✓ sous-interrogation dans une instruction DELETE, SELECT, ou UPDATE
- ✓ dans un SELECT utilisant les opérateurs ensemblistes (GROUP BY, etc.)
- ✓ en tant que valeur par défaut (DEFAULT) d'une colonne d'un CREATE|ALTER TABLE ;
- ✓ dans la condition d'une contrainte CHECK d'un CREATE | ALTER TABLE

Pr. EL BELRHTI EL ALAOUI

58

### III-A-3. Modification & Suppression d'une séquence

- Modification :
- ✓ ALTER SEQUENCE [schéma.]nom\_séquence  
INCREMENT BY entier
- ✓ On ne peut pas modifier le numéro de démarrage de la séquence.
- ✓ Exemple : ALTER SEQUENCE SeqTest INCREMENT BY 1
- Suppression :
- DROP SEQUENCE [schéma.]nom\_séquence

Pr. EL BELRHTI EL ALAOUI

59

## III-B. Intégrité référentielle : Rupture du lien Père/fils

- Pour assurer l'intégrité référentielle, Oracle possède trois alternatives pour résoudre le problème de la rupture du lien Père/fils :
- ✓ Le blocage de toute suppression met en jeu ce lien.
- ✓ L'option **ON DELETE** dans la définition des clés étrangères.
- ✓ L'option **CASCADE CONSTRAINTS** au moment de la suppression d'une table.

Pr. EL BELRHTI EL ALAOUI

60

## III-B. Intégrité référentielle : ...

### III-B-1. ON DELETE

FOREIGN KEY (colonne1 [,colonne2]...) REFERENCES [schéma.]TablePere (colonne1 [,colonne2]...)

[ON DELETE { CASCADE | SET NULL }]

➤ La directive ON DELETE dispose de deux options :

- ✓ CASCADE : effectue des suppressions en cascade dans les tables en relation,
- ✓ SET NULL : positionne les clés étrangères de tous les enregistrements fils associés à l'enregistrement père supprimé à NULL.

Pr. EL BELRHTI EL ALAOUI

61

### III-B-1. ON DELETE

➤ Tester des suppressions cassant le lien père-fils dans les exemples suivants :

- ✓ Exemple 1 : Clients(NumClt, NomClt) Commandes0(NumCmd, DateCmd, #NumClt)
- ✓ Exemple 2 : Dans le même schéma, créer la table Commandes1(NumCmd, DateCmd, #NumClt) en tenant compte du fait qu'une suppression d'un client entraînera la suppression de ses commandes.
- ✓ Exemple 3 : Dans le même schéma, créer la table Commandes2 en permettant la suppression des clients tout en gardant les informations concernant leurs commandes.

62

### III-B-2. CASCADE CONSTRAINTS

➤ L'option CASCADE CONSTRAINTS de la requête DROP TABLE permet de :

- ✓ supprimer une table même si elle est référencée par une contrainte référentielle.
- ✓ de détruire les contraintes référentielles des tables 'fils' sans toucher à ses données.

➤ Commenter les requêtes suivantes :

DROP TABLE Commandes2 CASCADE CONSTRAINTS;

DROP TABLE clients CASCADE CONSTRAINTS;

Pr. EL BELRHTI EL ALAOUI

63

### III-B-3. Remarque

➤ La norme SQL2 offre autres options n'ont implémentées dans Oracle :

- ✓ ON DELETE SET DEFAULT : propage une valeur par défaut.
- ✓ ON UPDATE CASCADE | SET NULL | SET DEFAULT.

64

## III-C. Les Vues

- Vue : table virtuelle dérivée d'une ou plusieurs tables.
- Intérêts :
  - ✓ Elle permet à chaque utilisateur d'avoir sa propre vision des données (indépendance logique).
  - ✓ Consultation des données simplifiée : noms d'attributs traduits et explicites, requêtes complexes pré définis dans la vue...
  - ✓ Sécurité : accès aux vues et non aux tables
  - ✓ Modifier les tables sans changer les programmes

Pr. EL BELRHTI EL ALAOUI

65

### ➤ Syntaxes

#### ❖ Création:

- ✓ Privilèges requis :
  - Dans votre schéma : CREATE VIEW
  - Autre schéma : CREATE ANY VIEW

#### ✓ Syntaxe simplifié :

```
CREATE VIEW [schéma.]Nom_Vue
AS Sélection
[WITH READ ONLY];
      ▪ WITH READ ONLY : INSERT, UPDATE et DELETE non autorisés.
```

#### ❖ Effacement d'une vue : DROP VIEW Nom\_Vue

Pr. EL BELRHTI EL ALAOUI

66

### III-C. Les Vues

- Exercice :
- Créer puis tester la vue 'vue\_cde\_N1' qui regroupe les informations des articles de la commande numéro 1.
- Modification des données :
  - ❖ Problème :
    - ✓ Vue multi tables
    - ✓ Colonne NOT NULL non visible dans la vue
  - ❖ Solution : Déclencheur INSTEAD OF

Pr. EL BELRHTI EL ALAOUI

67

### III-D. Agrégation

```
SELECT remise , SUM(prixuv) "Somme des prix "
FROM articles
GROUP BY remise
HAVING SUM(prixuv) > 60 ;
```

Pr. EL BELRHTI EL ALAOUI

68

### III-E. Jointure externe

- La jointure est une opération permettant de combiner des informations venant de plusieurs tables.
- Différents types de jointures.
  - ✓ Jointure interne.
  - ✓ Jointure externe.

Pr. EL BELRHTI EL ALAOUI

69

#### a. Jointure Interne

- Seules les lignes ayant correspondance dans les deux tables jointes sont affichées.
- Exemple :
 

Sélectionner les articles commandés.

```
SELECT a.nomart, a.numart , b.numart
FROM articles a JOIN lignes_commande b
ON a.numart = b.numart;    -- Condition de jointure
```

Pr. EL BELRHTI EL ALAOUI

70

#### b. Jointure externe

- Correspondance non obligatoire.
- Il existe trois jointures externes (OUTER JOIN) :
  - ✓ Gauche (LEFT)
  - ✓ Droite (RIGHT)
  - ✓ Entière (FULL)

Pr. EL BELRHTI EL ALAOUI

71

#### b. Jointure externe

- Jointure externe gauche (LEFT OUTER JOIN). Exemple :
  - ✓ Sélectionner les articles même ceux qui n'ont pas été commandés.
  - ✓ Correspondance non obligatoire sur la table lignes\_commande.
- Jointure externe gauche : RIGHT OUTER JOIN
  - ✓ Correspondance non obligatoire sur la table articles.
- Jointure externe entière : FULL OUTER JOIN
  - ✓ Correspondance non obligatoire sur les deux tables.
- Remarques
  - ✓ Auto-jointure : jointure sur la même table.
  - ✓ Jointure interne = Jointure naturelle.

72

## III-F. Les opérateurs ensemblistes

- Union (UNION, UNION ALL) :  
retourne toutes les lignes des deux tables.
- Intersection (INTERSECT) :  
extrait uniquement les lignes présentes dans les deux tables.
- Différence (MINUS , non commutatif) :  
retourne les lignes présentes uniquement dans la première table.

Pr. EL BELRHITI EL ALAOUI

73

- Dans une requête utilisant des opérateurs ensemblistes :
  - ✓ Tous les SELECT doivent avoir le même nombre de colonnes sélectionnées, et leurs types doivent être un à un identiques.
  - ✓ Les doubles sont supprimés automatiquement (DISTINCT implicite) à moins d'utiliser UNION ALL.
  - ✓ Les noms de colonnes sont ceux du premier SELECT.

Pr. EL BELRHITI EL ALAOUI

74

### Exemple 1 :

Quels sont les articles et les fournisseurs(numfour, nomfour, remise) dont la remise est supérieure à 2.

- ```
(SELECT nomart as "Articles et Fournisseurs", remise
FROM articles WHERE remise >2 ) UNION
(SELECT nomfour, remise FROM fournisseurs WHERE remise > 2)
```
- Exemple 2 : Quelles sont les remises communes entre articles et fournisseurs ?
  - Exemple 3 : Quels sont les couples (numfour, remise) de la table 'fournisseurs' qu'on n'a pas dans la table 'articles' (numart, remise) ?

Pr. EL BELRHITI EL ALAOUI

75

## III-G-2. Fonctions numériques

- ABS(n), POWER(n, e), ROUND(n[, p]), SQRT(n), etc.
- ✓ ROUND : 0 par défaut
- Exemple :  

```
SELECT POWER(10, 2) FROM dual ;
```

Pr. EL BELRHITI EL ALAOUI

77

## III-G. Fonctions prédéfinies

### III-G-1. Fonctions d'agrégation

- AVG([DISTINCT | ALL] expression) : Moyenne.
- COUNT([DISTINCT|ALL]expression|\*) : Nombre de lignes  
MAX, MIN, SUM, ...
- Les fonctions d'agrégat ne tiennent pas compte des valeurs NULL à l'exception de count(\*) .
- DISTINCT : la fonction ne prend en compte que des valeurs distinctes (ALL est par défaut)
- Exemple : 

```
SELECT COUNT(*) , AVG(prix)
FROM articles WHERE remise > 10000
```

Pr. EL BELRHITI EL ALAOUI

76

### III-G-3. Les fonctions de chaînes de caractères

- CONCAT(c1,c2) : équivalent à l'opérateur || .
- LENGTH(chaine) : longueur d'une chaîne. Exemple :  

```
SELECT nomfour, LENGTH(nomfour) AS "Longueur nomfour"
FROM fournisseurs ORDER BY "Longueur nomfour"
```
- SUBSTR(chaine, Début [,longueur ]) :
  - ✓ Extraire une chaîne de caractère
  - ✓ Si la longueur n'est pas précisée, on prend tous les caractères jusqu'à la fin de la chaîne
  - ✓ Exemple :  

```
SELECT SUBSTR(prenom_four, 1, 1) || ' ' || nomfour
FROM fournisseurs
```

Pr. EL BELRHITI EL ALAOUI

78

- TO\_DATE(chaine[, format]) : convertir chaîne en date.

- ✓ Exemples de formats d'affichage :

DDD numéro du jour dans l'année  
 DD numéro du jour dans le mois  
 D numéro du jour dans la semaine  
 MM numéro du mois  
 YYYY année  
 Etc.

- ✓ Exemple :

```
UPDATE commandes
SET datecmd = TO_DATE('10-Février-1980', 'DD/MM/YYYY')
WHERE numcmd = 1
```

Pr. EL BELRHITI EL ALAOUI

79

### III-G-3. Les fonctions de chaînes de caractères

- TO\_CHAR (date[, format]) TO\_DATE

- ✓ Même format que TO\_DATE.

- ✓ Exemple :

- Insérer deux commandes datées respectivement 10/12/1915 et 23/12/2015 (J/M/A).
- Sélectionner les commandes datées comme \*\*/\*2/\*\*15.

- INSTR, REPLACE, TRANSLATE, etc.

Pr. EL BELRHITI EL ALAOUI

80

## III-G. Fonctions prédéfinies

### III-G-4. Fonctions manipulant les dates

- TRUNC | ROUND ( Date,{' YY | MM |DD | etc. '})
- ✓ Arrondir le mois de la date de votre système.
- EXTRACT( {YEAR | MONTH | DAY | etc.} FROM Expression)
- ✓ Extraire le numéro de mois de la date de votre système.
- ADD\_MONTHS (Date, Nombre)
- ✓ Sélectionner la date qui sera dans trois mois d'aujourd'hui.
- SYSDATE, NEXT\_DAY, MONTHS\_BETWEEN, etc.

Pr. EL BELRHITI EL ALAOUI

81

## IV. PL/SQL (Procedural Langage/SQL)

### IV-A. Les premiers pas

### IV-B. Branchements, boucles, tableaux et structures

### IV-C. Transactions

### IV-D. Sous-programmes stockés

### IV-E. Triggers

### IV-F. Exceptions

### IV-G. Curseurs

Pr. EL BELRHITI EL ALAOUI

82

## IV. PL/SQL

### IV-A. Les premiers pas

#### IV-A-1. Généralités

- Objectif : Effectuer des traitements complexes sur les BD.
- Hérite d'Ada.
- Avantages :
  - ✓ Portable : un programme PL/SQL ne dépend pas du SE hébergeant le serveur Oracle.
  - ✓ Modulaire : les sous-programmes et les packages.

Pr. EL BELRHITI EL ALAOUI

83

#### IV-A-1. Généralités

- Avantages (suite) :

- ✓ Réduction du trafic du réseau : les sous-programmes stockés.
- ✓ Traiter les erreurs : les exceptions.
- ✓ Parcourir les résultats de requêtes : les curseurs.
- ✓ Etendre la logique des contraintes : les triggers.
- ✓ Simplifier l'implémentation des règles d'intégrités de la BD : les mécanismes pour programmer les transactions.

Pr. EL BELRHITI EL ALAOUI

84

## IV-A-1. Généralités

- Bloc PL/SQL, syntaxe :

```
DECLARE
    Déclaration des variables
BEGIN
    Instructions
END;
```

- Les sous-blocs d'un bloc qui déclare un objet (variable, type, curseur, exception, etc.) peuvent accéder à l'objet en question, ce dernier n'est pas visible des blocs supérieurs.

Pr. EL BELRHTI EL ALAOUI

85

## IV-A. Les premiers pas

### IV-A-2. Types PL/SQL & Variables

- Types PL/SQL :

- ✓ Les types SQL.
- ✓ Types personnalisés
- ✓ INTEGER : NUMBER(38,0) Entiers sans décimales.
- ✓ SIGNTYPE : {-1, 0, 1}.
- ✓ ...

- Déclaration d'une variable|constante. Syntaxe :

Identificateur [CONSTANT] type [NOT NULL] [:= initialisation];

Pr. EL BELRHTI EL ALAOUI

86

## IV-A-2. Types PL/SQL & Variables

- La directive INTO :

- ✓ Affecter à une variable le résultat d'une requête SELECT

- ✓ Syntaxe :

```
SELECT liste INTO { variable [,variable]... } FROM...;
```

- ✓ Une requête SELECT ... INTO doit retourner un et un seul enregistrement, si :

- ❑ aucun enregistrement : exception NO\_DATA\_FOUND.

- ❑ plusieurs enregistrements :

- ❖ exception TOO\_MANY\_ROWS,

- ❖ curseurs.

Pr. EL BELRHTI EL ALAOUI

87

## IV-A-2. Types PL/SQL & Variables

- Assigner à une variable le type :

- ✓ d'un champ d'une table (vue) ou d'une autre variable : %TYPE

NomVariable [Table|Vue].Colonne | identificateur %TYPE ;

- ✓ d'un enregistrement : %ROWTYPE.

NomVariable Table|Vue % ROWTYPE;

- Variables de substitution :

- ✓ Variables d'entrée d'un bloc PL/SQL anonyme.

- ❑ Lire la variable au clavier : la directive ACCEPT

- ❑ Accéder au contenu de la variable : &

Pr. EL BELRHTI EL ALAOUI

88

## IV-A-2. Types PL/SQL & Variables

- Variables de session :

- ✓ Déclaration : directive VARIABLE

- ✓ Utilisation dans un bloc PL/SQL :

- : Nom\_variable

- ✓ Affichage sous sqlplus : directive PRINT.

Pr. EL BELRHTI EL ALAOUI

89

## IV-A-3. Les packages

- Ensemble d'objets (variables, procédures, exceptions, curseurs,..) groupés pour offrir un ensemble de services homogènes.

- Avantages :

- modularité & encapsulation (certains traitements sont masqués)

- Facilite la maintenance de l'application (modularité, extensibilité, réutilisabilité).

Pr. EL BELRHTI EL ALAOUI

90

### IV-A-3. Les packages

- Packages prédéfinis :
- ✓ DBMS\_OUTPUT : gestion des entrées/sorties sous PL/SQL,
- ✓ DBMS\_LOCK : gérer les verrous,
- ✓ DBMS\_RANDOM : générer des nombres aléatoires,
- ✓ etc.
- Créer son package : CREATE PACKAGE ...
- Appel : NomPackage.NomService
- ✓ NomService : procédure, fonction...du package en question

Pr. EL BELRHTI EL ALAOUI

91

### IV-A-3. Les packages

#### DBMS\_OUTPUT

- Service : écrire et lire des lignes dans un buffer depuis un bloc PL/SQL.
- Activation du (session) : SET SERVEROUTPUT ON
- Exemples de procédures :
  - ✓ PUT :
    - met dans le buffer le contenu d'une variable de type VARCHAR.
    - le contenu du buffer est affiché à l'écran à la fin du bloc.
    - syntaxe :
  - DBMS\_OUTPUT.PUT(ligne\_buffer IN VARCHAR|DATE| NUMBER);
- ✓ PUT\_LINE : PUT + NEW\_LINE

Pr. EL BELRHTI EL ALAOUI

92

## II. PL/SQL

### IV-B. Branchements, boucles, tableaux et structures

#### IV-B-1. Branchements et boucles

- IF :
 

```
IF (expression-booléenne) THEN instruction(s);
[ ELSE instruction(s) ];
END IF;
```
- :
 

```
NULL AND FALSE → FALSE
NULL OR FALSE → NULL
NULL OR TRUE → TRUE
```

Pr. EL BELRHTI EL ALAOUI

98

### IV-B-1. Branchements et boucles

#### CASE :

##### CASE

```
WHEN expression-booléenne 1 THEN instructions1;
WHEN expression-booléenne 2 THEN instructions2;
...
WHEN expression-booléenne N THEN instructionsN;
[ELSE instructionsN+1;]
END CASE;
```

#### La boucle Tant que :

```
WHILE condition LOOP
  instructions;
END LOOP;
```

Pr. EL BELRHTI EL ALAOUI

99

#### IV-B-1. Branchements et boucles

##### La boucle Répéter

```
LOOP
  instructions;
EXIT WHEN condition;
END LOOP;
```

##### La boucle Pour :

```
FOR compteur IN [REVERSE] Borne_Inf .. Borne_Sup LOOP
  instructions;
END LOOP;
```

Pr. EL BELRHTI EL ALAOUI

100

#### IV-B-2. Tableaux : TABLE

- Tableaux dynamiques accessibles uniquement sous PL/SQL.
- Un tableau à deux dimensions :
  - ✓ colonne clé primaire (de type BINARY\_INTEGER)
  - ✓ colonne pour stocker des éléments de même type (NUMBER, CURSEUR, TYPE, ROWTYPE, RECORD,...).
- Syntaxes :
  - ✓ Définition du type tableau :
 

```
TYPE NomTypeTab IS TABLE OF
Type_éléments [NOT NULL] INDEX BY BINARY_INTEGER;
```
  - ✓ Déclaration d'une variable de type tableau :
 

```
NomTableau NomTypeTab;
```

Pr. EL BELRHTI EL ALAOUI

101

## IV-B-2. Structures

- Syntaxe :

```
TYPE nomRecord IS RECORD
(
    NomChamp1 type [NOT NULL] [:= expression]
    [, NomChamp2 type... ] ...
);
```

Pr. EL BELRHTI EL ALAOUI

102

## IV. PL/SQL

### IV-C. Les transactions

- Suite d'opérations faisant passer la BD d'un état cohérent à un autre état cohérent.
- Caractéristiques (ACID) :
  - ✓ Atomicité : tout ou rien;
  - ✓ Cohérence : préservation de la cohérence de la BD;
  - ✓ Isolation : 'mise en série' des transactions;
  - ✓ Durabilité : une fois la transaction est terminée, les M.AJ. sont définitives.

Pr. EL BELRHTI EL ALAOUI

106

## IV-C. Les transactions

- Une transaction débute :

- ✓ Implicitement, soit :
  - à la connexion,
  - à la première requête SQL,
  - à la fin de la transaction précédente.
- ✓ Explicitement : n'existe pas sous oracle.
- Une transaction prend fin :
- ✓ Implicitement :
  - à la première requête LDD ou LCD rencontrée,
  - à la fin de la session (avec ou sans déconnexion)
- ✓ Explicitement : instructions COMMIT ou ROLLBACK

Pr. EL BELRHTI EL ALAOUI

107

## IV-C. Les transactions

- Une transaction est validée :

- ✓ Explicitement : COMMIT
- ✓ Implicitement :
  - Requêtes : CREATE, DROP, ALTER, GRANT ...
  - Ordres : EXIT, CONNECT, DISCONNECT,...

- Une transaction est annulée :

- ✓ Explicitement : ROLLBACK
- ✓ Implicitement : interruption normale d'une requête, panne du client ou du serveur, RESET,...

Pr. EL BELRHTI EL ALAOUI

108

## IV-C. Les transactions

- Autocommit

- ✓ active | désactive la gestion automatique des transactions.
- ✓ variable d'environnement :
  - SQL\*Plus : SET AUTOCOMMIT **ON** | **OFF**
  - SQL Developper : Outils → Préférences → BD → Paramètres de feuille de calcul → Validation automatique
- ✓ ON :
  - Une instruction = une transaction
  - COMMIT implicite après chaque instruction.

Pr. EL BELRHTI EL ALAOUI

109

## IV-C. Les transactions

- Autocommit (suite)

- ✓ OFF :
  - COMMIT pour valider les MAJ.
  - Utiliser pour programmer des transactions portant sur plusieurs instructions.
- SAVEPOINT:
  - ✓ Instruction PL/SQL
  - ✓ Associe à un point de validation (étiquette) → permet d'annuler les MAJ effectuées depuis l'étiquette.

Pr. EL BELRHTI EL ALAOUI

110

## IV. PL/SQL

### IV-D. Sous-programmes stockés

- Procédures et fonctions compilées et stockées dans la BD.
- Contrairement aux procédures, les fonctions ne peuvent retourner qu'un seul résultat
- Intérêt :
  - ✓ Programmation modulaire :
    - Réutilisé et partagé les modules.
    - Réduire les efforts de la programmation client et de la maintenance.

Pr. EL BELRHITI EL ALAOUI

113

### IV-D. Sous-programmes stockés

- Intérêt (suite) :
  - ✓ Rapidité : optimisées à la compilation.
  - ✓ Réduction du trafic du réseau : une seule instruction est envoyée sur le réseau.
  - ✓ Sécurité : exécuter une fonction ou une procédure sans avoir le droit
    - d'exécuter les instructions qui la constituent.
    - d'accéder aux objets (table, vue, variable...) qu'elle manipule.

Pr. EL BELRHITI EL ALAOUI

114

### IV-D-1. Syntaxes

- Création :
  - ✓ Procédure
- ```
CREATE PROCEDURE [schéma.]Nom_Procédure
[(paramètre1 [ IN | OUT | IN OUT ] type
[,paramètre2 [ IN | OUT | IN OUT ] type]... )]
IS BEGIN
    Instructions
END ;
```

Pr. EL BELRHITI EL ALAOUI

115

### IV-D-1. Syntaxes

- Création (suite) :
    - ✓ Fonction
- ```
CREATE FUNCTION [schéma.]Nom_Fonction
[(paramètre1 type [,paramètre2 type]... )]
RETURN type
IS BEGIN
    Instructions -- RETURN est obligatoire
    END;
```
- Suppression :
- ```
DROP PROCEDURE | FUNCTION
[schéma.] Nom_Procédure|Fonction;
```

Pr. EL BELRHITI EL ALAOUI

116

## IV. PL/SQL

### IV-E. Triggers

- Intérêt :
  - ✓ étendre la logique des contraintes : héritage, contrôle de saisie...
  - ✓ Réduire les efforts de la programmation client et de la maintenance.
  - ✓ Renforcé la sécurité du système.
  - ✓ La réPLICATION des données dans les BD distribuées.

Pr. EL BELRHITI EL ALAOUI

112

### IV-E. Triggers

- Procédure stockée attachée à un événement :
- ✓ instruction LMD : INSERT, UPDATE ou DELETE;
  - Le trigger est toujours lié à une table ou une vue.
- ✓ instruction LDD : CREATE, ALTER, ou DROP d'un objet (table, séquence, etc.);
- ✓ démarrage | arrêt de la base,
- ✓ connexion | déconnexion d'un utilisateur,
- ✓ erreur définie (NO\_DATA\_FOUND, etc.)
- Déclenchés avant (BEFORE), après (AFTER) ou à la place (INSTEAD OF) de la requête événement.

Pr. EL BELRHITI EL ALAOUI

113

## IV-E. Triggers

- Tester un trigger → Déclencher l'événement correspondant.
  - Syntaxe réduite :
- ```
CREATE TRIGGER NomDéclencheur
{ BEFORE | AFTER | INSTEAD OF }
{ DELETE | INSERT | UPDATE[OF col1 [,col2]...] }
[OR { DELETE | INSERT | UPDATE[OF col1 [,col2]...] }]...
ON { NomTable | NomVue }
[FOR EACH ROW]
[WHEN ( condition ) ]
BEGIN instructions END ;
```

Pr. EL BELRHITI EL ALAOUI

124

## IV-E. Triggers

- Les options :
- ✓ FOR EACH ROW
  - Si elle est mentionnée, le traitement dans le trigger doit être exécuté pour chaque ligne concernée par l'événement.
  - Sinon, le traitement n'est exécuté qu'une seule fois par événement.
- ✓ WHEN : conditionne l'exécution du déclencheur.
- Restrictions:
  - ✓ Un déclencheur ne peut pas valider ou annuler une transaction :
    - Pas de COMMIT, ROLLBACK et SAVEPOINT dans le corps du trigger.
  - ✓ INSTEAD OF ne peut porter que sur une vue.

Pr. EL BELRHITI EL ALAOUI

125

## IV-E. Triggers

- Outils :
- ✓ INSERTING, UPDATING et DELETING : variables booléennes permettant d'identifier l'événement qui a déclenché le trigger.
- ✓ Les variables :NEW et :OLD
  - Se référer aux anciennes et nouvelles valeurs des lignes.
  - Utilisable uniquement avec l'option FOR EACH ROW.
  - Uniquement après UPDATE que les deux variables sont affectées :
    - DELETE, uniquement :OLD.
    - INSERT, uniquement :NEW.

Pr. EL BELRHITI EL ALAOUI

126

## IV-E. Triggers

- Gestion des déclencheurs :
- ✓ Recompilation d'un déclencheur :
`ALTER TRIGGER nomDéclencheur COMPILE;`
- ✓ Désactivation | Réactivation :
  - d'un déclencheur :
`ALTER TRIGGER nomDéclencheur DISABLE | ENABLE;`
  - de tous les déclencheur d'une table :
`ALTER TABLE nomTable DISABLE | ENABLE ALL TRIGGERS;`
- ✓ Suppression :
`DROP TRIGGER nomDéclencheur;`

Pr. EL BELRHITI EL ALAOUI

127

## IV-E-2. Exercice

1. Ajouter l'attribut Qt\_en\_stock à la table Articles avec une valeur par défaut nulle (0).
  2. MAJ cette colonne suite aux
    - a. insertions de nouvelles commandes.
    - b. aux annulations des commandes.
    - c. mises à jour des commandes.
- Tester les déclencheurs créés.

Pr. EL BELRHITI EL ALAOUI

131

## IV. PL/SQL

### IV-F. Les Exceptions

#### IV-F-1. Généralités

- Permet de :
- ✓ Rattraper les erreurs en cours d'exécution lors de l'exécution d'un programme PL/SQL ou d'une requête SQL.
- ✓ Eviter qu'un programme de s'arrêter à la première erreur rencontrée.
- Une exception est détectée (levée) au cours d'exécution d'un programme (entre un BEGIN et un END).
- Deux types d'exceptions :
  - ✓ Exceptions prédéfinies,
  - ✓ Exceptions utilisateurs.

Pr. EL BELRHITI EL ALAOUI

132

**Syntaxe :**

```

DECLARE
    // Déclarations
BEGIN
    // Instructions
EXCEPTION
    WHEN Exception1 THEN Instructions;
    [WHEN Exception2 THEN Instructions;]
    ...
    [WHEN OTHERS THEN instructions;]
END;

```

- ✓ Si aucune exception n'est levée, le bloc EXCEPTION est ignoré.
- ✓ Si une exception est levée,
  - l'exécution du bloc PL/SQL courant est abandonnée.
  - le traitement de l'exception est recherché dans le bloc EXCEPTION.

Pr. EL BELRHTI EL ALAOUI

133

**IV. PL/SQL****IV-F. Les Exceptions****IV-F-2. Exceptions prédefinies**

- Les erreurs Oracle sont identifiées par des codes qui ont la forme ORA-XXXX.
- Certains de ces codes ont un alias, mais chacune de ces erreurs correspond à une exception (les exceptions prédefinies).

Pr. EL BELRHTI EL ALAOUI

134

**> Exemples d'exceptions prédefinies :**

| Code d'erreur | Alias de l'erreur | Commentaire                                                    |
|---------------|-------------------|----------------------------------------------------------------|
| ORA-01476     | ZERO_DIVIDE       | Division par zéro.                                             |
| ORA-01403     | NO_DATA_FOUND     | Requête ne retournant aucun résultat.                          |
| ORA-01422     | TOO_MANY_ROWS     | Requête retournant plusieurs lignes.                           |
| ORA-06592     | CASE_NOT_FOUND    | Aucun des choix de la structure CASE sans ELSE n'est effectué. |

- Remarque : Contrairement à une requête SELECT, l'exception NO\_DATA\_FOUND ne se lève pas avec les requêtes UPDATE et DELETE.

Pr. EL BELRHTI EL ALAOUI

135

**> Syntaxe (suite) :**

```

DECLARE
    ...
    exception1 EXCEPTION;
    ...
BEGIN
    ...
    RAISE exception1;
    ...
EXCEPTION
    ...
    WHEN exception1 THEN instructions
    ...
    WHEN OTHERS THEN instructions
END;

```

Pr. EL BELRHTI EL ALAOUI

137

**IV. PL/SQL****IV-F. Les Exceptions****IV-F-3. Exceptions utilisateurs****IV. PL/SQL****IV-F. Les Exceptions****IV-F-3. Exceptions utilisateurs**

- Exceptions définies par les utilisateurs.
- Syntaxe :
  - ✓ Déclaration se fait dans le bloc DECLARE par :
 

```
nomException EXCEPTION;
```
  - ✓ Déclenchement est programmé dans le corps du programme (entre BEGIN et bloc EXCEPTION) par :
 

```
RAISE nomException;
```

Pr. EL BELRHTI EL ALAOUI

136

**IV. PL/SQL****IV-G. Curseurs****IV-G-1. Généralités**

- Un curseur est une variable dynamique qui permet de parcourir les lignes renvoyées par une requête SQL.
- De son ouverture à sa fermeture, le curseur pointe toujours sur la ligne courante.
- C'est une solution, entre autres, au problème des requêtes type 'SELECT ... INTO ...' retournant plusieurs lignes.
- Parcourir le curseur en récupérant les lignes une par une dans une variable locale.
- Deux types de curseurs : prédefinis et d'utilisateurs.

Pr. EL BELRHTI EL ALAOUI

140

## IV. PL/SQL

### IV-G. Curseurs

#### IV-G-2. Curseur prédefini : SQL

- Crée automatiquement par Oracle sur les requêtes LMD :
- INSERT, UPDATE et DELETE.
- Porte le nom SQL.
- Exemples de ses attributs :
  - ✓ SQL%ROWCOUNT : Nombre de lignes affectées par l'instruction LMD précédente.
  - ✓ SQL%FOUND : Booléen valant TRUE si l'instruction précédente affecte au moins une ligne.

Pr. EL BELRHITI EL ALAOUI

141

## IV. PL/SQL

### IV-G. Curseurs

#### IV-G-3. Curseurs utilisateurs

- Crées par l'utilisateur pour traiter le résultat de requêtes retournant plusieurs lignes.
- Plusieurs curseurs peuvent être définis dans un bloc ou un programme PL/SQL.

Pr. EL BELRHITI EL ALAOUI

143

#### ➤ Exemple 13 :

```

DECLARE
    NbreFournSup NUMBER;
BEGIN
    DELETE FROM fournisseurs where num_fourn = 4;
    NbreFournSup := SQL%ROWCOUNT ;
    DBMS_OUTPUT.PUT_LINE(' Le nombre de
                           fournisseurs supprimer par la
                           requête précédente est ' || NbreFournSup);
END;
  
```

Pr. EL BELRHITI EL ALAOUI

142

#### ➤ Syntaxe :

- ✓ Déclaration d'un curseur :
  - Dans la zone déclarative du programme,
  - **CURSOR** Nom\_curseur **IS** (clause SELECT);
- ✓ Ouverture d'un curseur :
 

```
OPEN nom_curseur ;
```
- ✓ Récupération des données de la ligne courante du curseur :
 

```
FETCH nom_curseur INTO Var_1, Var_2, ...;
```

  - Les Var\_i correspondent aux colonnes de la clause SELECT.
  - Suite à cette instruction, le curseur se positionne sur la ligne suivante.
- ✓ Fermeture d'un curseur :
 

```
CLOSE nom_curseur ;
```

Pr. EL BELRHITI EL ALAOUI

144

## V. Contrôle de données

- Toute manipulation dans une BD nécessite d'avoir le(s) privilège(s) adéquat(s).
- Le contrôle de données a comme objectifs la gestion des utilisateurs, des privilèges et des rôles.

### V-A. Les utilisateurs

### V-B. Les priviléges

### V-C. Les rôles

Pr. EL BELRHITI EL ALAOUI

148

## V. Contrôle de données

### V-1. Les utilisateurs

- Un utilisateur (user),
- ✓ Est identifié par son nom (et son mot de passe).
- ✓ Dispose d'un espace de stockage pour ses objets.
- ✓ Ses objets sont stockés dans son schéma qui porte son nom.
- ✓ Peut être assimilé à son schéma.
- ✓ A tous les droits sur les objets de son schéma.
- ✓ Selon ses priviléges, peut accéder aux objets de schémas tiers.
- ✓ Par défaut, nouvellement créé, il n'a aucun droit, même pas celui de se connecter. Toutefois, il apparaît dans la liste des utilisateurs.

Pr. EL BELRHITI EL ALAOUI

149

- Quelques utilisateurs créés automatiquement lors de la création de la BD :
  - ✓ SYS :
    - le plus puissant.
    - le propriétaire des tables du dictionnaire de données.
  - ✓ SYSTEM :
    - le DBA créé automatiquement par Oracle.
    - permet d'effectuer les tâches d'administration
  - ✓ PUBLIC : tous les utilisateurs.

Pr. EL BELRHITI EL ALAOUI

150

- Création d'un utilisateur, syntaxe avec les options les plus utilisées (**suite**) :
- Espace disque (tablespaces) :
  - DEFAULT TABLESPACE : associe un espace de travail à l'utilisateur.
  - Si l'option tablespaces n'est pas spécifiée, tablespace SYSTEM est associé à l'utilisateur en tant qu'espace de travail.
  - USERS est la tablespace recommandée comme espaces de travail.
- PASSWORD EXPIRE : oblige l'utilisateur à changer son MP à la première connexion.

Pr. EL BELRHITI EL ALAOUI

152

- Création d'un utilisateur, syntaxe avec les options les plus utilisées :
 

```
CREATE USER nom_user
  IDENTIFIED BY { mot_passe | EXTERNALLY | GLOBALLY AS 'nom_externe' }
  [ DEFAULT TABLESPACE nom_tablespace_permanent
    [QUOTA { taille | K | M } | UNLIMITED } ON nom_tablespace ]
  [PASSWORD EXPIRE ];
```
- IDENTIFIED BY :
  - mot\_passe : authentification par mot de passe (utilisateur local).
  - EXTERNALLY : utilisation de l'authentification du SE.
  - GLOBALLY : authentification à l'aide d'un système d'annuaire.

Pr. EL BELRHITI EL ALAOUI

151

- Exemples :
- ✓ Utilisateur user1 avec les paramètres par défaut :
 

```
CREATE USER user1 IDENTIFIED BY mpuser1 ;
```
- ✓ Utilisateur user2 avec les paramètres
  - L'espace réservé pour ses objets est 15M,
  - Ses objets seront stockés dans USERS,
  - Il doit modifier son mot de passe à la première connexion.

```
CREATE USER user2 IDENTIFIED BY mpuser2
  DEFAULT TABLESPACE USERS QUOTA 15M ON USERS
  PASSWORD EXPIRE;
```
- Pour visualiser les propriétés des utilisateurs, la vue DBA\_USERS du dictionnaire des données peuvent être utilisée.

Pr. EL BELRHITI EL ALAOUI

153

- Suppression d'un utilisateur :
 

```
DROP USER Nom_utilisateur ;
```
- ✓ Conditions :
  - l'utilisateur à supprimer est déconnecté.
  - l'utilisateur ne possède pas d'objets dans son schéma.

L'option CASCADE permet de supprimer un utilisateur et ses objets.
- ✓ Exemple :
 

```
DROP USER user1 CASCADE ;
```
- Modification d'un utilisateur :
 

```
ALTER USER Nom_utilisateur ;
```

Pr. EL BELRHITI EL ALAOUI

154

- ## V. Contrôle de données
- ### V-2. Les priviléges
- Les priviléges permettent de donner des droits sur :
    - ✓ la base de données : priviléges système,
    - ✓ les données de la base : priviléges objets.
  - Les priviléges système diffèrent sensiblement d'un SGBD à un autre.
  - Quelques exemples de priviléges :
    - ✓ le droit de se connecter à une base de données (ouvrir une session),
    - ✓ le droit de créer une table,
    - ✓ le droit de sélectionner des lignes dans une table.

Pr. EL BELRHITI EL ALAOUI

155

- Les droits limitent l'accès des utilisateurs à certaines informations de la base et leur interdisent certaines actions.
- Vues utilisées pour visualiser les informations sur les priviléges octroyés :
  - ✓ Systèmes : DBA\_SYS\_PRIVS
  - ✓ Objets : DBA\_TAB\_PRIVS
  - ✓ à l'utilisateur courant : SESSION\_PRIVS
- Gestion de priviléges :
  - ✓ GRANT permet d'attribuer un privilège système ou objet à un utilisateur ou à un groupe d'utilisateurs.
  - ✓ REVOKE permet d'annuler un privilège système ou objet d'un utilisateur ou d'un groupe d'utilisateurs.

Pr. EL BELRHITI EL ALAOUI

156

## V. Contrôle de données

### V-2. Les priviléges

#### V-2-1. Privilèges système

- Les priviléges système permettent aux utilisateurs qui en disposent d'administrer une base de données Oracle.
- ✓ Exemples de priviléges système :
  - accès à la BD,
  - création ou suppression des utilisateurs,
  - création ou suppression des tables.
- ✓ Réservés aux DBA.

Pr. EL BELRHITI EL ALAOUI

157

- Attribution de priviléges système :
- ✓ Syntaxe :
 

```
GRANT { privilège_système | nom_ rôle | ALL PRIVILEGES }
      [, { privilège_système ...}]
      TO { utilisateur | PUBLIC }[,{ utilisateur ...}]
      [ WITH ADMIN OPTION ];
      • WITH ADMIN OPTION : accorde le droit à l'utilisateur
        de transmettre ces priviléges.
```
- ✓ Exemple 1:
 

```
GRANT CONNECT TO Alaoui;
      • Avec le compte Alaoui, essayez de créer une table.
```

Pr. EL BELRHITI EL ALAOUI

158

#### Attribution de priviléges système (Suite):

##### Exemple 2:

```
GRANT CREATE ANY TABLE TO Alaoui;
```

- La mention ANY donne la possibilité au bénéficiaire de s'en servir dans tout schéma (par défaut, n'inclue pas celui de l'utilisateur SYS).
- Essayez de créer une table avec la connexion Alaoui.

Pr. EL BELRHITI EL ALAOUI

159

- Révocation de priviléges système
- ✓ Pour pouvoir révoquer un privilège ou un rôle, vous devez détenir au préalable ce privilège avec l'option WITH ADMIN OPTION.
- ✓ Syntaxe :
 

```
REVOKE { privilège_Système | nom_Rôle | ALL PRIVILEGES }
      [, { privilège_Système ...}]
      FROM { utilisateur | PUBLIC } [, { utilisateur ... }];
      • PUBLIC pour annuler le(s) privilège(s) à chaque utilisateur
        ayant reçu ce(s) privilège(s) par l'option PUBLIC.
```

Pr. EL BELRHITI EL ALAOUI

160

## V. Contrôle de données

### V-2. Les priviléges

#### V-2-2. Privilèges objets

- Relatifs aux données de la base et aux actions sur les objets.
- Chaque type d'objet a différents priviléges associés.
- Le créateur d'un objet
  - ✓ possède tous les droits sur ledit objet : lecture, modification, suppression, etc.
  - ✓ peut accorder des droits sur cet objet à d'autres utilisateurs
  - ✓ les autres utilisateurs, hormis le DBA, n'ont aucun droit.

Pr. EL BELRHITI EL ALAOUI

161

➤ Exemples de privilèges objets :

| Privilèges     | Objets               |
|----------------|----------------------|
| <b>SELECT</b>  | Table, Vue, Séquence |
| <b>DELETE</b>  | Table, Vue           |
| <b>UPDATE</b>  | Table, Vue, Colonne  |
| <b>ALTER</b>   | Table, Vue           |
| <b>EXECUTE</b> | Procédure, Fonction  |

Pr. EL BELRHTI EL ALAOUI

162

➤ Attribution de privilèges objets (GRANT)

✓ L'utilisateur qui exécute cette commande doit avoir reçu le droit de transmettre ces privilèges.

✓ Syntaxe :

```
GRANT { privilège_objet | nom_rôle | ALL PRIVILEGES }
```

```
[ (colonne_1 [,colonne_2]...) ] [, { privilège_objet ... }]
```

```
ON [schéma.]nom_objet
```

```
TO {utilisateur | PUBLIC} [, { utilisateur ... }]
```

```
[WITH GRANT OPTION];
```

- colonne\_i : la ou les colonnes sur lesquelles porte le privilège.

✓ Exemple :

Affectation des privilèges de lecture de la table 'clients' et de modification des colonnes 'nom client' et 'prénom client' à l'utilisateur Alaoui.

```
GRANT SELECT , UPDATE(nom_clt,prenom_clt)
```

```
ON clients TO user3;
```

163

➤ Révocation de privilèges objets

✓ Conditions :

- avoir reçu le privilège système ANY OBJECT PRIVILEGE
- Il n'est pas possible d'annuler un privilège objet qui a été accordé avec l'option WITH GRANT OPTION

✓ Syntaxe :

```
REVOKE { privilège_objet | nom_rôle | ALL PRIVILEGES }
```

```
[ (colonne1 [,colonne2]...) ] [, { privilège_objet ... }]
```

```
ON [schéma.]nom_Objet
```

```
FROM { utilisateur | PUBLIC } [, { utilisateur ... }];
```

✓ Exemple :

Révoquez les privilèges de modification et de lecture de la table 'clients' à l'utilisateur 'Alaoui'.

```
REVOKE UPDATE , SELECT
```

```
ON clients FROM Alaoui;
```

164

## V. Contrôle de données

### V-2. Les privilèges

#### V-2-3. Privilèges prédéfinis

➤ Facilitent la gestion des droits.

Pr. EL BELRHTI EL ALAOUI

165

➤ Exemples :

✓ SYSDBA : Offre tous les privilèges pour administrer la base de données.

✓ SYSOPER : Hérite de tous les privilèges de SYSDBA sauf :

- CREATE DATABASE
- ALTER DATABASE BEGIN/END BACKUP
- RECOVER DATABASE
- RESTRICTED SESSION.

✓ GRANT ANY PRIVILEGE : Autorisation de transmettre tout privilège système.

➤ En utilisant une connexion avec le privilège

✓ SYSDBA, vous êtes dans le schéma de SYS.

✓ SYSOPER, vous êtes dans le schéma PUBLIC.

Pr. EL BELRHTI EL ALAOUI

166

## V. Contrôle de données

### V-3. Les rôles

➤ Un rôle est un ensemble nommé de privilèges système et/ou objets regroupés pour simplifier la gestion des privilèges.

➤ Un rôle peut être assigné aux utilisateurs ou attribué à un autre rôle.

Pr. EL BELRHTI EL ALAOUI

167

➤ Exemples de rôles prédéfinis :

- ✓ CONNECT. Ses privilèges :
  - Autorise la connexion à une base Oracle.
  - Accéder à certains objets (tables, etc.)
- ✓ RESOURCE. Ses privilèges :
  - créer et utiliser presque tous les objets : tables, séquence, index, procédures, triggers, clusters.
  - N'est pas autorisé à créer des vues.
- ✓ DBA ( DataBase Administrator )
  - Regroupe tous les privilèges systèmes avec WITH ADMIN OPTION.
  - Les utilisateurs SYS et SYSTEM ont par défaut le rôle DBA.

Pr. EL BELRHITI EL ALAOUI

168

➤ Création d'un rôle (CREATE ROLE)

- ✓ Pour pouvoir créer un rôle, vous devez posséder le privilège CREATE ROLE
- ✓ Au départ, seul le DBA dispose de ce droit.
- ✓ Syntaxe :

```
CREATE ROLE nom_rôle
[ NOT IDENTIFIED | IDENTIFIED BY { mot_passe |
    EXTERNALLY | GLOBALLY AS 'nom_externe' } ] ;
```

- NOT IDENTIFIED : l'utilisation de ce rôle est autorisée sans MP.
- IDENTIFIED : l'utilisation de ce rôle est autorisée par une authentification locale (mot\_passe), externe à Oracle (EXTERNALLY, SE) ou globale (GLOBALLY, annuaire).
- Un rôle doit être activé s'il est créé avec l'option IDENTIFIED :

```
SET ROLE nom_rôle ;
```

Pr. EL BELRHITI EL ALAOUI

169

➤ Création d'un rôle (Suite)

- ✓ Exemple :

```
CREATE ROLE test_role;          -- Crédit du rôle test_role.
GRANT CREATE TABLE, CREATE VIEW
    TO test_role;           -- Affecter des privilèges au rôle créé.
GRANT test_role TO user3;      -- Attribution du rôle.
```

➤ Révocation d'un rôle : REVOKE

- ✓ Pour pouvoir annuler un rôle, vous devez détenir au préalable ce rôle avec l'option ADMIN OPTION ou avoir reçu le privilège système GRANT ANY ROLE.
- ✓ Syntaxe :

```
REVOKE nom_rôle [, nom_rôle ...]
FROM {utilisateur | nom_rôle | PUBLIC} [, {utilisateur | nom_rôle | PUBLIC}]...;
```

- ✓ Exemple : REVOKE test\_role FROM user3;

Pr. EL BELRHITI EL ALAOUI

170

➤ Suppression d'un rôle :

- ✓ Syntaxe : DROP ROLE nom\_rôle ;
- ✓ Désaffecte aussi le rôle aux bénéficiaires.
- ✓ Les utilisateurs des sessions ouvertes ne seront affectés par cette suppression qu'après déconnexion.

➤ Informations sur les rôles

- ✓ DBA\_ROLES : liste les rôles de la base.
- ✓ DBA\_ROLE\_PRIVS : liste les rôles et les privilèges octroyés à tous les utilisateurs de la base.
- ✓ SESSION\_ROLES : liste les rôles activés pour la session courante.

Pr. EL BELRHITI EL ALAOUI

171

## VI. Bibliographie et Webographie

- <http://www.oracle.com/>
- SQL pour Oracle, Christian Soutou, 2008 EYROLLES.
- Cours "BASES DE DONNÉES – ORACLE" - Bertrand LIAUDET, INSIA
- <http://oracle.developpez.com/>
- Introduction au PL/SQL Oracle, Alexandre Mesié, 2011

Pr. EL BELRHITI EL ALAOUI

172