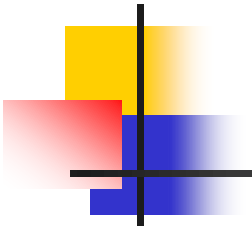
UNIVERSITÉ PRIVÉE DE MARRAKECH

UNIVERSITÉ RECONNUE PAR L'ÉTAT

جامعة معترف بها من طرف الدولة

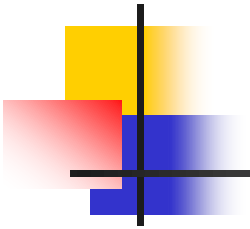
DBA: Gestion des utilisateurs

Module: Administration des bases de données Année universitaire: 2022/2023



Plan

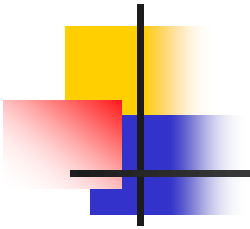
- **Géstion d'ORACLE**
  - Profils utilisateur
  - Roles et Privilèges
  - Utilisateur



Qu’est ce qu’un DBA?

- D.B.A = Data Base Administrator = Administrateur de bases de données.

- Il est responsable du bon fonctionnement de toutes les bases de données de l‘entreprise.



Qu'est ce qu'un DBA?

Les interlocuteurs du DBA :

- les développeurs
- les administrateurs système (unix/linux, windows, etc.)
- le support technique ORACLE
- le support technique interne
- l'administrateur sécurité (s'il existe)
- les utilisateurs

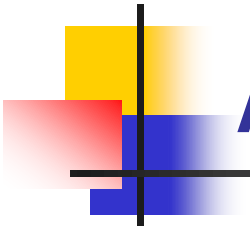
Les tâches de l'administrateur de la Base de données

Typologies des utilisateurs

- **Les différentes catégories d'utilisateurs :**

En pratique il existe 5 catégories d'utilisateurs, qui utilisent différents outils en fonction de leur profil et du SGBD ciblé. Ceci est résumé dans le tableau suivant :

Type d'utilisateur	Profil	Outils	Ex d'outils
Utilisateur final	non informaticien	application client serveur ou web, ou progiciel, ou appli bureautique, outil d'aide à la décision	Excel, appli maison, Oarcle application, B.O
utilisateur final évolué	non infrmaticien éclairé	client QBE (Query by Example)	MS QUery
Développeur	Technicien ou ingénieur	Bloc note, environnement de developpement, Atelier de développement	TOAD, designer, webdb
Administrateur application	Utilisateur final hiérarchiquement privilégié	l'application elle même, via un accès privilégié	NA
DBA	Technicien ou ingénieur	langage SQL, console d'admin texte, console graphique, console web	sqlplus, OEM CS ou web, TOAD,



Authentification

Authentification locale au niveau du système d'exploitation :

- C'est une forme d'authentification externe, en ce sens que ce n'est pas ORACLE qui contrôle la connexion grâce à son référentiel interne. On se connecte directement (via telnet ou ssh par exemple) au système qui héberge le serveur de données, puis à la base locale.

- Ce type de connexion ne nécessite pas d'identifiant ni de mot de passe ORACLE, mais d'être un utilisateur privilégié au niveau O.S.

Authentification

Authentification via fichier de mots de passe :

- Dans ce cas de figure, les privilèges seront contrôlés à partir d'un fichier de mot de passe cryptés local.
- exemple de création du fichier :

$ORAPWD FILE=monfic PASSWORD=monpasse ENTRIES=100

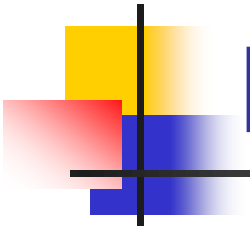
  - ORAPWD : utilitaire pour créé Le password file.
  - PASSWORD : le mot de passe de l'utilisateur par default du système.
  - ENTRIES : le nb max d'utilisateurs référençable dans le fichier.

DBAs et privilèges ORACLE

- Un utilisateur ORACLE (déclaré au sein de la base, à distinguer de l'utilisateur au niveau OS).
- Le DBA a tous les '**privilèges système**' au sein de la base.

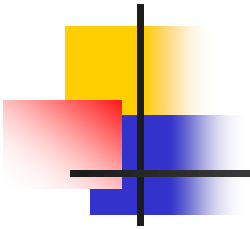
- Grace à quoi, il peut essentiellement :
  - consulter et mettre à jour (SELECT, UPDATE, INSERT, DELETE) toutes les données utilisateur de la base.
  - créer, modifier des structures de données utilisateur (CREATE, ALTER, DROP) n'importe ou (ANY TABLESPACE).
  - gérer des utilisateurs et des droits (CREATE/DROP USER, GRANT, REVOKE).
  - consulter la totalité du dictionnaire.

DBAs et privilèges ORACLE



- exécuter des ordres d'administration (CREATE DATABASE, DATAFILE, TABLESPACE)
- Il y a 2 utilisateurs privilégiés prédéfinis, **SYS** et **SYSTEM** (dont les mots de passe sont définis à la création de la base ou par 'ORAPWD')
- Ils sont tous les 2 DBA, mais SYS est plus privilégié en ce sens qu'il est propriétaire des tables et des vues du dictionnaire.
- Il existe un ensemble de privilèges (ROLE) prédéfinis nommés 'DBA' qui donne les privilèges nécessaires à un DBA.
- Pour donner les privilèges d'un DBA a un utilisateur :

  GRANT DBA TO ali;



Gestion

- Profils utilisateurs
- Privilége et rôles
- Utilisateurs ORACLE

Profils utilisateurs

- Nombre maximal de tentatives de connexion à la base,
- le temps de vérouillage d'une compte
- limiter les ressources système allouées à un utilisateur afin d'éviter une surcharge inutile du serveur.

Comment ?

ORACLE nous propose une solution efficace et pratique pour mettre en place ce type d'action : les PROFILS.

Profils utilisateurs

- Un PROFIL est un ensemble de **limitations** système.
- Une fois qu'un PROFIL a été assigné à un utilisateur celui-ci ne pourra plus dépasser les limitations imposées.

Deux types de limitations:

- Les limitations du mots de passe
- Les limitations des ressources système

Les limitations du mot de passe

- Quelque options très intéressantes permettant d'augmenter la sécurité de vos mots de passe.

OPTION	DESCRIPTION
FAILED_LOGIN_ATTEMPTS	le nombre maximal de tentatives de connexion.
PASSWORD_LOCK_TIME	la durée de vérouillage du compte utilisateur après avoir bloqué le compte avec le paramètre FAILED_LOGIN_ATTEMPTS.
PASSWORD_LIFE_TIME	la durée d'utilisation du même mot de passe.
PASSWORD_GRACE_TIME	Définit en jours le temps de grace qui vous sera alloué pour changer votre mot de passe.

Les limitations des ressources

- Voici la liste de quelques limitations que vous pourrez mettre en place.

OPTION	DESCRIPTION
SESSIONS_PER_USER	le nombre de session maximum qu'un utilisateur pourra ouvrir.
IDLE_TIME	le temps en minutes pour la durée d'inactivité maximale d'une session.
CPU_PER_SESSION	le temps de processeur maximum en centièmes de secondes qu'une session pourra utiliser.
CPU_PER_CALL	le temps de processeur maximum en centièmes de secondes qu'un "appel serveur" pourra utiliser.
LOGICAL_READS_PER_SESSION	le nombre maximal de bloc lus durant une session. On parlera ici des blocs lus sur le disque et dans la mémoire.
LOGICAL_READS_PER_CALL	le nombre maximal de bloc lus durant un "appel serveur".
CONNECT_TIME	le temps en minutes pour la durée de connexion maximale d'une session
PRIVATE_SGA	la taille en Kbytes ou MBytes que pourra utiliser une session.
COMPOSITE LIMIT	le coût total des limitations autorisée pour une session.

Mise en place d'un profil

- La méthode pour mettre en place est très simple :

1. Etablir les limitations de mot de passe et les limitations système.
2. Créer le profil
3. Attribuer le profil aux utilisateurs qui devront être limités

Création d'un profil

- La syntaxe de création d'un profil est :

CREAT PROFIL profil1
LIMIT
--ressource_parameters
<param> <valeur>
--password_parameters
<param> <valeur>

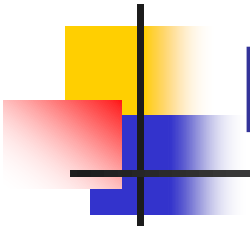
Les valeurs peuvent etre **UNLIMITED, DEFAULT** ou integer.

Assigner un profil a un utilisateur

- Par défaut un utilisateur se voit assigner le profil **DEFAULT** lors de sa création.

- Si vous souhaitez lui assigner un nouveau profil cela sera possible soit lors de la création soit avec la commande

ALTER USER <nom> PROFIL <profil1>



Modification d'un profil

- Avant de pouvoir modifier des limitations de ressources système, vous devez disposer du privilège système **ALTER** profil et vous devrez disposer des privilèges **ALTER** profil et **ALTER USER** pour modifier des limitations de mot de passe.

Modification d'un profil

- La syntaxe en elle même est quasiment identique à la syntaxe de **CREAT PROFIL**.

ALTER profil profil
LIMIT
--ressource_parameters
<param> <valeur>
--password_parameters
<param> <valeur>

Suppression d’un profil

- Il existe 2 cas de figure possibles.

1. Le premier cas le plus simple consiste à supprimer un profil qui n'a été assigné à personne. Vous pourrez donc le supprimer sans action supplémentaire.

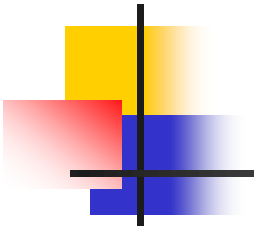
2. Si ce profil a été assigné à un utilisateur vous devrez alors utiliser l'option CASCADE qui demandera à ORACLE de supprimer le profil et d'assigner le profil DEFAULT à tout les utilisateurs qui possédaient le profil qui vient d'être supprimé.

Suppression d'un profil

- Exemple

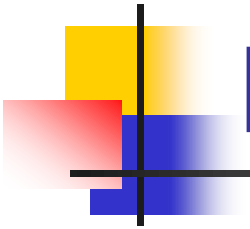
DROP PROFIL <profil> CASCADE

- Ici tous les utilisateurs qui disposaient du profil **<profil>** se verront automatiquement assigner le profil **DEFAULT**.



Roles et Privilèges

- Ces concepts sont mis en œuvre pour protéger les données en accordant (ou retirant) des privilèges a un utilisateur ou un groupe d'utilisateurs .
- Il n'existe pas de notion de groupe d'utilisateur sous ORACLE, mais la notion de rôle, qui permet de nommer un groupe de privilèges.
- Un rôle est un regroupement de privilèges. Une fois créé il peut être assigné à un utilisateur ou à un autre rôle



Privilèges

- Les privilèges sont de deux types :

  - Les privilèges de niveau objet
  - Les privilèges de niveau système

Privilèges objets

- Implicitement le créateur d'un objet (TABLE, VUE , INDEX, etc…) est son propriétaire, et possède tous les privilèges et droits sur le contenu et le contenant :
  - consultation, mises à jour, insertion , suppression des lignes

  et aussi modification et suppression de la structure.

- les privilèges SELECT, INSERT, UPDATE, DELETE sur une table **ali**.x par exemple permettent à l'utilisateur qui les a reçu de sélectionner, ajouter, modifier et supprimer des lignes dans la table x appartenant à l'utilisateur **ali**.

Privilèges systèmes

- Les privilèges systèmes sont des privilèges qui à la différence des privilèges d'accès aux objets s'intéressent plutôt au contenant qu'au contenu.
- Ils concernent principalement des ordres de création, de modification de structures et de suppression d'objets.

Par exemple CREAT TABLE, CREAT VIEW.

Assigner des privilèges système à un utilisateur

- Lorsqu'un utilisateur est créé avec l'instruction CREATE USER, il ne dispose encore d'aucun droit car aucun privilège ne lui a encore été assigné.
- Il ne peut même pas se connecter à la base !

Assigner des privilèges système à un utilisateur

- Il faut donc lui assigner les privilèges nécessaires

  Il doit pouvoir se connecter, créer des tables, des vues, des séquences.
  Pour lui assigner ces privilèges de niveau système il faut utiliser l'instruction **GRANT** .

Assigner des privilèges système à un utilisateur

Exemple

- Pour que l'utilisateur puisse simplement se connecter à la base, il doit bénéficier du privilège système **CREATE SESSION**

GRANT CREATE SESSION TO <user> ;

- Ensuite il faut lui assigner des droits de création de table

GRANT CREATE TABLE TO <user> ;

L'ensemble de ces privilèges peuvent être assignés au sein d'une même commande.

Assigner des privilèges objet à un utilisateur

- Pour assigner à l'utilisateur le droit de sélectionner, insérer, modifier et supprimer des lignes dans la table **x** de l'utilisateur **ali**.

```
GRANT SELECT ,INSERT ,UPDATE
,DELETE
ON <ali.x>
TO <user>;
```

Assigner des privilèges objet à un utilisateur

- Une liste de colonnes peut être indiquée dans l'instruction afin de restreindre davantage les droits sur une table :

GRANT
UPDATE (<col1>,< col2>)
ON <ali.x>
TO <user> ;

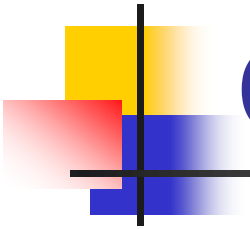
Créer des rôles

- un **rôle** et se créé avec l'instruction **CREATE ROLE**

- Lorsque le rôle est créé, il ne contient rien et il faut l'alimenter à l'aide d'instructions **GRANT**

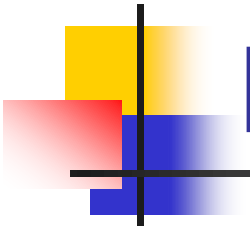
```
CREATE ROLE comptabilite ;
GRANT SELECT, INSERT, UPDATE, DELETE
ON comptable.facture
TO comptabilite ;
GRANT SELECT, INSERT, UPDATE, DELETE
ON comptable.tax
TO comptabilite ;

GRANT SELECT, INSERT, UPDATE, DELETE
ON comptable.journal
TO comptabilite ;
```

Créer des rôles



- Une fois le rôle créé, il peut être assigné à un utilisateur ou à un autre rôle

GRANT comptabilite TO <user> ;



Rôles déjà existants

- Trois rôles existent en standard

  - **CONNECT**
  - **RESOURCE**
  - **DBA**

Suppression d'un rôle

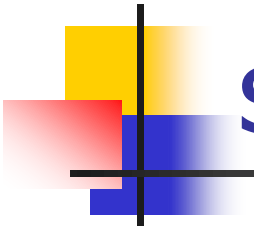
- Un rôle peut être supprimé en utilisant l'instruction

  DROP ROLE

DROP ROLE <nom_rôle>;

- Le rôle spécifié ainsi que tous les privilèges qui lui sont associés sont supprimés de la base et également retiré à tous les utilisateurs qui en bénéficiaient.

Supprimer des Privilèges system



- Les privilèges système qui ont été assignés à des utilisateurs ou à des rôles peuvent être retirés avec l'instruction **REVOKE**

- Les arguments sont identiques à ceux décrits pour l'instruction **GRANT**

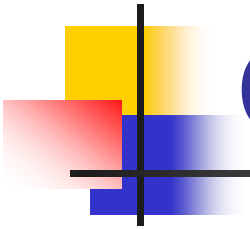
REVOKE <system_Privilège>
FROM <user|rôle>

Supprimer des Privilèges objet

- Les privilèges objet qui ont été assignés à des utilisateurs ou à des rôles peuvent être retirés avec l'instruction **REVOKE**

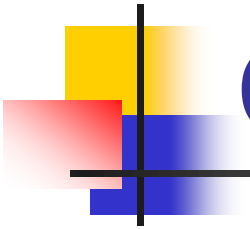
Exemple

- **REVOKE <object_Privilège>**
- **FROM <user|rôle>**



Gestion des utilisateurs

- **Création des utilisateurs**
- **Modifications d'un utilisateur**
- **Suppressions d'un utilisateur**
- **Informations sur les utilisateurs**



Création des utilisateurs

- Voici les différentes étapes qui seront nécessaire à la création d'un utilisateur ORACLE :

1. Choisir un nom d'utilisateur
2. Choisir une méthode d'authentification
3. Choisir les TABLESPACEs que l'utilisateur pourra utiliser
4. Créer l'utilisateur
5. Assigner les rôles et privilèges à l'utilisateur

Schema et utilisateur

- Un schéma est un ensemble nommé d'objets tels que des tables, vues, procédure et packages associés à un utilisateur précis.

- Un utilisateur ne pourra alors être associé qu'à un seul schéma et réciproquement.

- Un utilisateur de base de données va correspondre à un login qui aura reçu certains privilèges. Cet utilisateur sera stocké dans le dictionnaire de données et disposera d'un espace de stockage pour ses objets qui seront alors stockés dans son schéma.

  En ORACLE on pourra assimiler un utilisateur avec son schéma.

Choix du nom de l'utilisateur

- il est fortement recommandé de metttre une stratégie de nommage en place.
- **Par exemple** :
  Ali ARABI donnera comme login arabi_a.

- Il convient ensuite de connaitre les limitations et règles de nommage à respecter:
  - Taille maximale 30 caractères.
  - Ne devra contenir que des lettres de [a-z] et des chiffres [0-9]. Tout les caractères accentués ou autres sont à éviter. Vous pourrez également utiliser les symboles #, $, _.

Choisir la méthode d'authentification

- Il existe différents type d'authentification :

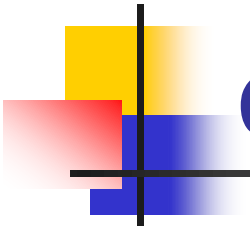
  - Authentification par la base de données.
  - Authentification par le système d'exploitation.
  - Authentification par le réseau.

Choisir les TABLESPACEs que l'utilisateur pourra utiliser

- Il va maintenant falloir choisir le domaine d'action du nouvel utilisateur.
- En effet, pour des raisons de sécurités évidentes, nous allons restreindre le champ d'action de l'utilisateur en choisissant les tablespaces que celui-ci sera en mesure d'utiliser.

- Il va donc falloir identifier tous les tablespaces nécessaire à l'utilisateur, que ce soit des tablespaces de données, d'index, ou temporaire.

Choisir les TABLESPACEs par défaut de l'utilisateur

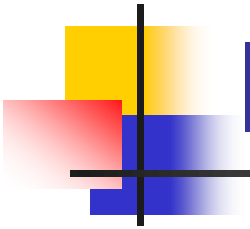


- Cette étape est absolument indispensable, elle va permettre de définir le tablespace de données et le tablespace temporaire de l'utilisateur.

- Cette étape est indispensable pour éviter toute écriture dans le tablespace SYSTEM (qui est assigné si aucun tablespace par défaut n'est défini).

- Vous devrez définir ces deux tablespaces avec les options **DEFAULT TABLESPACE** pour le tablespace de données et **TEMPORARY TABLESPACE** pour le tablespace temporaire.

Définir les QUOTAs de l'utilisateur



- Une fois les tablespaces identifiés, l'étape suivante va consister à définir l'espace alloué à l'utilisateur sur chacun des tablespaces.

Voici les différentes options disponibles pour les quotas :

- Une taille en K (KiloBytes) ou en M (MegaBytes)
- **Unlimited**

Creation de l'utilisateur

```
CREATE USER ali
IDENTIFIED BY password
DEFAULT TABLESPACE tbs_users
QUOTA 10M ON tbs_users
TEMPORARY TABLESPACE tmp_users
QUOTA 5M ON tmp_users
PASSWORD EXPIRE
ACCOUNT LOCK
```

Modification d’un utilisateur

- Si vous souhaitez changer le mot de passe d'un utilisateur voici la commande que vous devrez utiliser:

ALTER USER < login de l'utilisateur >
IDENTIFIED BY < nouveau mot de passe >

Suppression d’un utilisateur

- Il est important de noter qu'un utilisateur actuellement connecté à la base ne pourra pas être supprimé.

Suppression d'un utilisateur

- On peut soit supprimer un utilisateur seul :

DROP USER utilisateur

- Soit supprimer l'utilisateur et son schéma :

DROP USER utilisateur CASCADE

Informations sur l'utilisateur

- Quelques vues utile pour obtenir des informations sur les utilisateurs :

- **DBA_USERS** qui contiendra les informations sur les DBA
- **USER_USERS** qui contiendra les informations de l'utilisateur courant
- **USER_TS_QUOTAS** qui contiendra les informations sur les quotas de l'utilisateur courant