TP N°1 : Administration d'Oracle

Document à rendre :
un rapport de TP qui contient
les commandes utilisées,
avec description et références à l'énoncé,
***ainsi que les résultats* et commentaires éventuels**

1) Lancer **start Database** puis **sqlplus**

Il existe 2 utilisateurs par défaut :

- SYS : propriétaire des tables et des vues du dictionnaire
- SYSTEM : peut que consulter les tables et vues du dictionnaire

Ces 2 utilisateurs ont par défaut le **rôle DBA** :

- Accès à tous les objets de tous les autres utilisateurs de la base
- Ont le droit d'exécuter certaines commandes d'exploitation et d'administration.

SQLPLUS est une interface en mode commande et console. Il permet :

- Exécution des commandes SQL.
- Exécution de code PL/SQL
- Exécution des commandes **SQLPLUS**.
- Exécution de fichiers scripts.
- La création de rapports.
- L'administration de la base de données.

2) Connectez-vous en tant que system

User : system
Mot de passe : celui que vous avez saisi lors de l'installation d'Oracle

Si la connexion réussit, vous obtiendrez le prompt :

SQL >

Pour paramétrer de façon correcte la fenêtre d'affichage, après chaque connexion tapez :

SQL> **set pause on** (Pressez Entrée pour aller à la page suivante)
SQL> **set pagesize 40** (40 lignes par page)
SQL> **set linesize 100** (100 colonnes par pages)

3) **Le dictionnaire de données (DD)**

Le dictionnaire de données (dictionary ou le synonyme dict) est un ensemble de tables et de vues qui contient toutes les informations concernant la structure de stockage et tous les objets de la base de données Oracle. Le propriétaire du dictionnaire de données est lôutilisateur «SYS» **.**

Dict est une vue : côest la méta-base.

Le dictionnaire de données stocke les informations sur :

– La structure logique de la base de données.
– La structure physique de la base de données.
– Les noms et les définitions des objets.
– Les contraintes dôintégrité définies pour les objets dôune base de données.
– Les noms des utilisateurs valides de la base de données et les privilèges attribuées à chaque utilisateur de la base de données.
– Lôaudit sur une base de données.

Ce dictionnaire est généralement exploité par lôadministrateur de base de données, mais côest aussi une source dôinformation utile pour les développeurs et les utilisateurs.

Pour connaître lôutilisateur qui sôest connecté : **show user**

a) Quels sont les attributs de la table dict ? Quels sont les types de données de chaque attribut ainsi que leurs tailles ?

b) Afficher les tuples de la table **dict**

c) Combien y a-t-il de tables répertoriées dans la table dict ?

Les noms des objets dans le dictionnaire de données Oracle débutent par lôun des trois préfixes suivants :

– Les vues **DBA** contiennent des informations sur les objets de tous les schémas.

– Les vues **ALL** incluent les enregistrements des vues USER et des informations sur les objets pour lesquels des privilèges ont été octroyés au groupe PUBLIC ou à lôutilisateur courant.

– Les vues **USER** contiennent des informations sur les objets appartenant au compte qui exécute la requête.

Les vues USER, ALL et DBA sont disponibles pour quasiment tous les objets de base de données.

Il existe aussi les vues commençant par le préfixe **V**$ (accessibles uniquement par lôutilisateur **SYS**) utilisées pour fournir des données relatives aux performances telles que des informations sur les fichiers de données et les structures de la mémoire.

. d) combien y a-t-il de vues commençant par

- **DBA ?**
- **ALL ?**
- **USER** ?

4) Ecrire un script nommé **listingTables.sql** permettant d'afficher les noms des tables (ou vues) de préfixe saisi par l'utilisateur. La liste des tables étant envoyée dans un fichier intitulé liste_des_tables.txt.

Exemple d'exécution :

SQL> Entrer le nom de la vue : dba
SQL> pas de table

SQL> Entrer le nom de la vue : DBA
SQL>

```
1 Vue : DBA_2PC_NEIGHBORS
2 Vue : DBA_2PC_PENDING
3 Vue : DBA_ADDM_FDG_BREAKDOWN
4 Vue : DBA_ADDM_FINDINGS
5 Vue : DBA_ADDM_INSTANCES
```

Aide :

La commande **spool** permet d'envoyer les résultats de l'écran dans un fichier texte qu'on pourra ensuite consulter. L'envoi des résultats se fait lorsqu'on arrête l'envoi par **off**

Exemple : spool c:\requetes.txt

La requête select * from dba_tables va envoyer les résultats de la requête dans le fichier requêtes dès qu'on tapera **spool off**

Pour inviter l'utilisateur à entrer une valeur qui sera saisie dans une variable avec un message :

accept <nom_variable> **prompt** "<message>"

Exemple : SQL> accept Nom prompt "Entrer votre nom : "

➔ Entrer votre nom : Martin

La variable Nom contiendra la chaîne de caractères Martin

prompt Bonjour M. &Nom → Bonjour M. Martin

bloc pl/sql anonyme (langage basé sur la langage ADA)

```
declare
  é é .
begin
  é é
end ;
```

Dans la section **declare**, on définit les variables

Nom varchar2(10) ;
N number(1) ;

Dans la section **begin é ..end**, on écrit les instructions du programme

La structure si : **if é . Then .... endif** ou **if é then é ..else é . endif**

Les structures de boucle :

for i in é ..loop	**loop**	**while é .. loop**
end loop	**é é**	**é é é é é ..**
	exit when é ..	**end loop**
	en loop	

Lôaffectation dôune variable se fait avec les **:=**

N := 10 ;
Nom :=ôMartinô;

Pour afficher un message à lôécran dans un bloc pl/sql, on utilise un paquetage **dbms_output** permet de faire des sorties écran.

Un paquetage est un fichier qui contient un ensemble de fonctions ou de procédures déjà écrire que le programmeur pourra utiliser.

Pour afficher un message à lôécran :dbms_output.**put_line**(ôJe suis en coursô) ;

Mais sqlplus ne gère pas, par défaut, les sorties pl/sql. Il faut lancer la commande sqlplus : **set serveroutput on**

Voici un petit programme pl/sql anonyme à taper dans un fichier premier_programme.sql (fichier script) que vous lancez avec la commande start

```
set serveroutput on
accept n prompt "Entrer un entier :"
spool c:\tp\multiplication.txt
declare
   x number(1) :=6 ;
begin
  for i in 1..&n loop
    dbms_output.put_line(i || Nfj Nj x || Nej Nj i*x) ;
  end loop ;
end ;
/
spool off
```

Si on nôa pas lancé la commande **set serveroutput on**, on nôaura pas dôaffichage.

5) La vue **dict** contient toutes les vues du dictionnaire de données
La vue **dict_columns** contient toutes les colonnes (attributs) de toutes les vues du dictionnaire de données.

a) Afficher, dans un fichier script nommé liste_vues.sql , toutes les vues (leurs noms) dont le nom contient ôVIEWSô Combien y a-t-il de vues ?

b) Afficher, dans un fichier script nommé liste_colonnes, toutes les vues (leurs noms) qui contiennent la colonne (ou attribut) ôTABLE_NAMEô Combien y-a-t-il de vues ?

6) **Les objets utilisateur**

Lôensemble des objets appartenant à un utilisateur est désigné par le terme **catalogue** ; il en existe un seul par utilisateur. Un catalogue affiche tous les objets dont lôutilisateur peut sélectionner les enregistrements.

Le nom de la vue qui désigne le catalogue est : **dba_catalog.** Cette vue nôest pas accessible à un utilisateur non system.

Le nom de la vue qui désigne le *catalogue utilisateur* est : **cat**

a) Quels sont les attributs de la vue dba_catalog ?

b) Combien y-a-t-il de vues ?

La vue **dba_objects** contient tous les types objets : les vues, les fonctions, les procédures, les index, les paquetages, les triggers, les séquences etc é ..
Cette vue nôest pas accessible pour un utilisateur non system.

La vue **user_objects** contient tous les types objets dôun utilisateur.

a) Quels sont les attributs de la vue **dba_objects** ?

7) **Les tables**

Toutes les vues dbaé . Ne sont pas accessible aux utilisateurs non system

La vue **dba_tables** contient toutes les tables de la base avec des informations plus détaillées comme le nom de lôespace de stockage, les statistiques et autres.

a) Combien dôattributs contient la vue **dba_tables** ?

La vue **dba_tab_columns** contient les noms des attributs, leurs types, leurs tailles et la vue ou table dans laquelle se trouve lôattribut.

a) Afficher les noms des attributs, leurs types et leurs tailles de la table dba_tables sans utiliser **desc**. Combien y a-t-il dôattributs ?

b) Quel est le type et la taille de lôattribut num_rows ?

La vue **dba_constraints** contient les noms des contraintes, le type et la condition dans le cas dôune contrainte check.

La vue **dba_cons_columns** permet de connaître les attributs avec leurs contraintes

Les mêmes vues avec le préfixe **user** pourront être consultées par les utilisateurs non system : **user_tables, user_constraints, user_tab_columns**

8) **Les tablespaces**

Un tablespace est un espace logique qui contient les objets stockés dans la base de données comme les tables ou les indexes.

Un tablespace est composé d'au moins dôun datafile, c'est à dire un fichier de données qui est physiquement présent sur le serveur à l'endroit stipulé lors de sa création.

Par défaut, il existe un tablespace intitulé SYSTEM qui contient le dictionnaire de données.

SURTOUT, il ne faut pas quûun utilisateur utilise le tablespace SYSTEM

Il faut créer un tablespace pour un ou plusieurs utilisateurs.

Syntaxe simplifiée :

create tablespace <Nom> **datafile** <chemin\nom_fichier> **size** <taille> **autoextend**<on/off> **next** <taille> **maxsize** <taille ou unlimited>;

La taille doit être exprimée en K ou M

a) Créer le tablespace **tbs_toto** destiné à recevoir les données de lûutilisateur toto avec les paramètres suivants :

nom : tbs_toto
fichier : c:\TP\tp1.dat
taille : 10M
autoexend : on
next : 10M
maxsize : 100M

Pour effacer un tablespace : **drop tablespace** <non_tablespace> ; attention, la suppression dûun tablespace ne supprime pas le datafile, il faut le faire "manuellement"

Pour modifier un tablespace : **alter tablespace** é é é .. ;

b) Vérifier la création du tablespace **tbs_....** ainsi que le fichier de données**,** la taille du fichier de données ainsi que la taille maxi du tablespace.

On utilisera, pour cela, les 2 vues **dba_tablespaces** et **dba_data_files**

9) **Les utilisateurs**

Chaque utilisateur créé doit être associé à un tablespace pour y créer ses objets (tables, vues, index, etc é) avec un quota sur le tablespace

Syntaxe pour créer un utilisateur :

create user <nom_utilisateur**> identified by** <mot _de_passe> **default tablespace** <nom_tablespace> **quota** <taille ou unlimited> on <nom_tablespace>

a) Créer un utilisateur avec les paramètres suivant :

nom : toto
mot de passe : toto
tablespace attribué : tbs_toto
quota 5M

b) Vérifier la création de lûutilisateur toto avec ses paramètres. On utilisera la vue **dba_users** et **dba_ts_quotas**

c) On peut modifier ou supprimer un utilisateur.

- Supprimer un utilisateur [avec tous ses objets] : **drop user** <user> **[cascade] ;**
- Modifier un mot de passe : **alter user** <user> **identified by** <mot_de_passe> ;
- Modifier un quota : **alter user** <user> **quota** <nouvelle_taille> **on** <nom_tablespace> ;

d) Connectez-vous avec l'utilisateur toto. Que se passe-t-il ?

e) Quand on crée un utilisateur, celui-ci ne dispose d'aucun privilège. Le DBA doit on attribuer à cet utilisateur des privilèges (ou des droits dans l'environnement linux).

10) **Les privilèges**

Il existe des **privilèges de niveau système** et des **privilèges de niveau objet**

- **Les privilèges de niveau système**

Privilège système	Operations autorisées
CREATE SESSION	se connecter à la base de données
CREATE TABLE	créer des tables dans le schéma de l'utilisateur
CREATE SEQUENCE	créer des séquences dans le schéma de l'utilisateur
CREATE VIEW	créer des vues dans le schéma de l'utilisateur
CREATE PROCEDURE	créer des procédures stockées, des fonctions ou des packages dans le schéma de l'utilisateur

Pour donner des privilèges de niveau système :

grant <privilege> [, <privilege>,é] **to** <user> [, <user> ou <role> ou **PUBLIC**] **;**

Exemple : L'utilisateur Scott a reçu les privilèges permettant d'ouvrir des sessions, de créer des tables, des séquences et des vues

grant create session, create table, create sequence, create view to Scott ;

f) Donner à l'user toto le droit de se connecter et d'ouvrir une session

g) Vérifier que ce privilège a bien été donné. Pour cela, on pourra consulter la vue **dba_sys_privs** ou la vue **session_privs**

h) Combien de privilèges de niveau système à l'utilisateur SYSTEM ?

i) L'utilisateur peut connaître ces privilèges par la vue **user_sys_privs** ou par la vue **session_privs.** Vérifier le privilège

- **Les privilèges de niveau objet**

Privilège objet	Table	Vue	Séquence	Procédure
ALTER	✓		✓	
DELETE	✓	✓		
EXECUTE				✓
INDEX	✓			
INSERT	✓	✓		
REFERENCES	✓			
SELECT	✓	✓	✓	
UPDATE	✓	✓		

Pour donner des privilèges objet :

GRANT object_priv [**(columns)**]**|ALL ON** <object>
TO {<user>| ou <role> ou PUBLIC}
[**WITH GRANT OPTION**];

Exemples :

- donner aux utilisateurs Sue et Rich le privilège permettant d'interroger votre table **Employees**

  grant select on Employees to Sue, Rich ;

- donner à l'utilisateur Scott le privilège permettant de mettre à jour les attributs NomDepart et Location de la table Departement

  grant update (NomDepart, Location) on Departement to Scott ;

- donner à l'utilisateur Scott le privilège **SELECT** et **INSERT** sur votre table **Departement.** Scott pourra accorder ces privilèges aux autres.

  grant select, insert on Departement to Scott with grant option ;

- autoriser tous les utilisateurs du système à interroger les données de la table **Departement** d'Alice.

  grant select on alice.Departement to PUBLIC ;

La vue **dba_tab_privs** (user_tab_privs) permet de connaître les privilèges de niveau objet accordés à tous les utilisateurs (à l'utilisateur connecté)

La vue **dba_col_privs** (user_col_privs) permet de connaître tous les privilèges sur les colonnes donnés aux utilisateur (à l'utilisateur connecté).

j) Donner le privilège à toto pour créer des tables.

k) Vérifier le privilège en étant SYSTEM puis en étant l'user toto

Pour supprimer des privilèges à des utilisateurs :

revoke {**privilege [, privilege...]** ou **ALL**} **on** <objet> **from** <user> ou <role> ou **PUBLIC ;**

Exemple : L'utilisateur Alice peut retirer les privilèges **SELECT** et **INSERT** donnés à l'utilisateur Scott sur la table **Departement**

revoke SELECT, INSERT on Departement from Scott ;

Autres commandes donnant des privilèges :

grant connect to <user> : donne le droit à l'user de se connecter et d'ouvrir une session

grant all privileges to <user> : donne à l'user tous les privilèges

grant resource to <user> : donne à l'user le droit de créer des objets

connect et ressource sont des **rôles prédéfinis** (voir plus loin)

11) **Les rôles**

Le rôle permet de nommer un groupe de privilèges pour l'affecter à un ou plusieurs utilisateurs.

Il existe un certain nombre de rôles prédéfinis :

Rôle	Privileges associés à ce rôle
CONNECT	ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE VIEW
RESOURCE	CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER
DBA	All system privileges WITH ADMIN OPTION
EXP_FULL_DATABASE	SELECT ANY TABLE, BACKUP ANY TABLE, INSERT, DELETE, AND UPDATE ON THE TABLES SYS.INCVID, SYS.INCFIL, AND SYS.INCEXP
IMP_FULL_DATABASE	BECOME USER

Syntaxe de création d'un rôle :

create role <nom_role> ;

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

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

grant <nom_role> **to** <user> ;

La suppression d'un rôle : **drop role** <nom_role> ;

La vue **dba_roles** permet d'afficher les rôles créés.

La vue **dba_role_privs** permet de connaître les rôles assignés aux utilisateurs

La vue **user_role_privs** permet de connaître les rôles assignés à l'utilisateur connecté

a) Afficher la liste des rôles créés

b) Quelles sont les rôles assignés à l'utilisateur SYSTEM ?

c) Créer le rôle nommé role_cnam qui permet de créer des procédures, des triggers et des sequences (create procedure, é)

d) Afficher la liste des privilèges du role role_cnam

e) Assigner le rôle role_cnam à l'utilisateur toto

f) Afficher la liste des rôles pour l'utilisateur toto.

12) **Les profils**

Un profil permet de définir un certain nombre de **limitations**. Une fois établit, ce profil peut être assigné à un utilisateur.

Un profil par défaut est assigné à tout utilisateur créé, toutes les valeurs par défaut sont unlimited.

Il existe 2 types de limitations :

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

Quelques limitations du mot de passe

Paramètre	signification
password_life_time	Durée maximale d'un mot de passe
failed_login_attempts	Nombre d'essais de connexion infructueux avant le blocage du compte
password_lock_time	Nombre de jours e blocage du compte après un fail_login_attemps

Quelques limitations des ressources du système

Paramètre	signification
sessions_per_user	Limite le nombre de sessions actives simultanément pour un utilisateur
connect_time	Indique la durée maximale de connexion autorisée pour la session
cpu_per_session	Indique la durée CPU maximale autorisée pour la session active, exprimée en centièmes de seconde
Idel_time	Indique la durée maximale avant fermeture automatique de la session, exprimée en minutes

Syntaxe pour créer un profil :

create profile <nom_profil>**limit** [paramètre {entier ou unlimited ou default} paramètre {entier ou unlimited ou default} é] ;

Pour supprimer un profil : **drop profile** <nom_profil> [**cascade**]

↳ Permet de remplacer par le profil DEFAULT

La vue **dba_profiles** permet de consulter la liste des profils avec les paramètres

L'assignation d'un profil se fait à la création d'un utilisateur :

create user é é é é é é .. **profile** <nom_profil>

Si l'utilisateur a été créé, on peut modifier le profil par défaut assigné à la création :

alter user <user> **profile** <nom_profil>

a) Quel est le nombre de sessions autorisés du profil par défaut ?
b) Combien d'essais de connexion sont-ils autorisés du profil par défaut ?
c) Créer un profil nommé profil_cnam avec les paramètres suivants :
   - Nombre de sessions actives simultanément 3
   - Temps de connexion 10 minutes
   - Durée d'inactivité avant fermeture de session 5 minutes
   - Nombre d'essais de connexion 2
d) Assigner le profil profil_cnam à l'utilisateur toto