

## TP1 ORACLE

Question 1 :

La requête permettant d'afficher les attributs d'une table est :

"DESCRIBE NOM\_TABLE"

Question 2 :

La requête permettant d'afficher les tuples d'une table est :

"SELECT \* FROM NOM\_TABLE"

Question 3 :

La requête permettant d'afficher le nombre de lignes de la table correspondante est :

"SELECT count(\*) FROM NOM\_TABLE"

```
SQL> SELECT COUNT(*) FROM DICT;  
  
COUNT(*)  
-----  
4669
```

Question 4 :

La requête permettant d'afficher le nombre d'entrées commençant par DBA :

SELECT COUNT(\*) FROM DICT WHERE TABLE\_NAME LIKE 'DBA%';

Pour ALL :

SELECT COUNT(\*) FROM DICT WHERE TABLE\_NAME LIKE 'ALL%';

Pour USER :

SELECT COUNT(\*) FROM DICT WHERE TABLE\_NAME LIKE 'USER%';

```
SQL> SELECT COUNT(*) FROM DICT WHERE TABLE_NAME LIKE 'USER%';  
  
COUNT(*)  
-----  
481
```

Question 5 :

Pour cette question, nous allons décrire l'utilisation de spool :

On crée d'abord le fichier où l'on va insérer le contenu de notre SELECT comme ceci :

```
spool C:\Users\Theo\Desktop\Cours\Oracle\TP1\liste_tables_dba.txt
```


On va ensuite exécuter la requête SQL SELECT voulue comme ci-dessous :

```
SELECT TABLE_NAME FROM DICT WHERE TABLE_NAME LIKE  
          'DBA%' ;
```

Et enfin on met fin à la commande spool afin de refermer le fichier et de le sauvegarder :

```
spool off
```

On peut alors exécuter ce cheminement pour chaque SELECT voulu, en n'oubliant pas de créer avec spool des fichiers textes différents pour chaque requête.



- liste\_tables\_all.txt
- liste\_tables\_dba.txt
- liste\_tables\_user.txt

Question 6 :

On nous demande à nouveau d'utiliser spool afin cette fois-ci de stocker une requête SQL comme celle-ci :

```
SELECT TABLE_NAME FROM DICT WHERE TABLE_NAME LIKE  
          '%VIEWS%' ;
```

Cette requête SQL va permettre, grâce aux %, de récupérer toutes les entrées ayant VIEWS dans leur TABLE\_NAME.

Ainsi, après avoir exécuté la fin de spool avec SPOOL OFF, les résultats de cette requête est stocké dans le fichier texte préalablement créé, que l'on aura ici nommé : liste\_views.txt.

## LES OBJETS UTILISATEURS :

### Question 7 :

Pour obtenir les attributs de la table dba\_catalog on utilise a nouveau la commande :

```
describe dba_catalog
```

### Question 8 :

Le nombre de vues s'obtient grâce à cette commande :

```
SELECT COUNT(*) FROM dba_catalog.
```

### Question 9 :

Ici bas les attributs de la table user\_objects.

```
OBJECT_NAME          VARCHAR2(128)
SUBOBJECT_NAME        VARCHAR2(128)
OBJECT_ID             NUMBER
DATA_OBJECT_ID        NUMBER
OBJECT_TYPE           VARCHAR2(23)
CREATED               DATE
LAST_DDL_TIME         DATE
TIMESTAMP             VARCHAR2(19)
STATUS               VARCHAR2(7)
TEMPORARY             VARCHAR2(1)
GENERATED             VARCHAR2(1)
SECONDARY            VARCHAR2(1)
NAMESPACE            VARCHAR2(1)
EDITION_NAME         VARCHAR2(128)
SHARING              VARCHAR2(18)
EDITIONABLE          VARCHAR2(1)
ORACLE_MAINTAINED    VARCHAR2(1)
APPLICATION          VARCHAR2(1)
DEFAULT_COLLATION    VARCHAR2(100)
DUPLICATED           VARCHAR2(1)
SHARDED              VARCHAR2(1)
CREATED_APPID        NUMBER
CREATED_VSNID        NUMBER
MODIFIED_APPID       NUMBER
MODIFIED_VSNID       NUMBER
```

## LES TABLES :

Question 10 :

On dénombre 26 attributs dans la table dba\_objects.

```
SQL> SELECT COUNT(COLUMN_NAME) FROM dba_tab_columns WHERE TABLE_NAME LIKE 'DBA_OBJECTS';  
COUNT(COLUMN_NAME)  
-----  
26
```

Question 11 :

Pour en afficher le nom, type et taille des différents attributs de la table dba on fait :

```
SELECT COLUMN_NAME, DATA_TYPE, DATA_LENGTH FROM  
DBA_TAB_COLUMNS WHERE TABLE_NAME LIKE 'DBA_TABLES'
```

COLUMN\_NAME : Nom

DATA\_TYPE : Type

DATA\_LENGTH : Taille

Et pour en savoir le nombre on utilise COUNT sur COLUMN\_NAME.

```
SQL> SELECT COUNT(COLUMN_NAME) FROM dba_tab_columns WHERE TABLE_NAME LIKE 'DBA_TABLES';  
COUNT(COLUMN_NAME)  
-----  
84
```

Question 12 :

Comme montré ci dessous sur la capture, l'attribut NUM\_ROWS est de type NUMBER et possède une longueur de 22.

```
SQL> SELECT DATA_TYPE, DATA_LENGTH FROM dba_tab_columns WHERE TABLE_NAME LIKE 'DBA_TABLES' AND COLUMN_NAME LIKE 'NUM_ROWS';  
DATA_TYPE  
-----  
DATA_LENGTH  
-----  
NUMBER  
22
```

Question 13 :

On utilise la commande :

```
"CREATE TABLESPACE tbs_toto DATAFILE  
'C:\Users\Theo\Desktop\Cours\Oracle\TP1\TP1.dat"
```

On peut également vérifier que cette tablespace a bien été créé avec une commande SELECT.

```
SQL> SELECT TABLESPACE_NAME, MAX_SIZE FROM DBA_TABLESPACES WHERE TABLESPACE_NAME LIKE 'TBS_TOTO';  
TABLESPACE_NAME      MAX_SIZE  
-----  
TBS_TOTO              2147483645
```

LES UTILISATEURS :

Question 14 :

On va maintenant créer l'utilisateur TOTO en utilisant la précédente tablespace créée.

```
"CREATE USER toto IDENTIFIED BY toto DEFAULT TABLESPACE  
tbs_toto QUOTA 5M ON tbs_toto"
```

On peut vérifier la présence de l'utilisateur avec une commande SELECT comme ci dessous.

```
SQL> SELECT USERNAME,PASSWORD FROM DBA_USERS WHERE USERNAME LIKE 'TOTO';  
USERNAME  
-----  
PASSWORD  
-----  
TOTO
```

Question 15 :

Nous ne pouvons pas avoir accès au compte toto car l'utilisateur toto créé ne dispose pas des bons privilèges

```
Entrez le nom utilisateur : toto
Entrez le mot de passe :
ERROR:
ORA-01045: l'utilisateur TOTO n'a pas le privilège CREATE SESSION ; connexion
refusée
```

LES PRIVILEGES :

Question 16 :

La commande ci dessous permet d'ajouter le droit de créer des sessions a TOTO :

“GRANT CREATE SESSION TO TOTO”

On vérifie la présence des privilèges avec un SELECT.

```
SQL> SELECT GRANTEE,PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE LIKE 'TOTO';

GRANTEE
-----
PRIVILEGE
-----
TOTO
CREATE SESSION
```

Question 17 :

On va maintenant sélectionner et afficher les privilèges de l'utilisateur principal, dit SYSTEM :

```
SQL> SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE LIKE 'SYSTEM';  
  
PRIVILEGE  
-----  
CREATE TABLE  
SELECT ANY TABLE  
DEQUEUE ANY QUEUE  
GLOBAL QUERY REWRITE  
ENQUEUE ANY QUEUE  
CREATE MATERIALIZED VIEW  
UNLIMITED TABLESPACE  
MANAGE ANY QUEUE  
  
8 lignes sélectionnées.
```

Comme la capture d'écran le montre il en dispose de 8.

Question 18 :

Accordons le droit à TOTO de créer des tables avec :

“GRANT CREATE TABLE TO TOTO”

Vérifions à nouveau que TOTO dispose de deux droits, et plus qu'un seul.

```
SQL> SELECT PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE LIKE 'TOTO';  
  
PRIVILEGE  
-----  
CREATE SESSION  
CREATE TABLE
```

## LES ROLES :

### Question 19 :

Voici une liste des rôles déjà présents dans Oracle, récupérés via une requête SQL :

```
SQL> SELECT ROLE FROM DBA_ROLES;

ROLE
-----
CONNECT
RESOURCE
DBA
PDB_DBA
AUDIT_ADMIN
AUDIT_VIEWER
SELECT_CATALOG_ROLE
EXECUTE_CATALOG_ROLE
CAPTURE_ADMIN
EXP_FULL_DATABASE
IMP_FULL_DATABASE
ROLE
-----
CDB_DBA
APPLICATION_TRACE_VIEWER
LOGSTDBY_ADMINISTRATOR
DBFS_ROLE
GSMUSER_ROLE
GSMROOTUSER_ROLE
AQ_ADMINISTRATOR_ROLE
AQ_USER_ROLE
DATAPUMP_EXP_FULL_DATABASE
DATAPUMP_IMP_FULL_DATABASE
ADM_PARALLEL_EXECUTE_TASK
```

Ici les rôles attribués à l'utilisateur SYSTEM :

```
SQL> SELECT GRANTED_ROLE FROM USER_ROLE_PRIVS;

GRANTED_ROLE
-----
AQ_ADMINISTRATOR_ROLE
DBA
```

On va donc créer le rôle ROLE\_CNAM auquel on attribue par la suite les différents privilèges :

```
SQL> alter session set "_ORACLE_SCRIPT"=true;

Session modifiée.

SQL> CREATE ROLE ROLE_CNAM;

Rôle créé.
```

Attribution des privilèges :

```
SQL> GRANT CREATE PROCEDURE, CREATE TRIGGER, CREATE SEQUENCE TO ROLE_CNAM;

Autorisation de privilèges (GRANT) acceptée.
```



Et on attribue enfin le rôle ROLE\_CNAM à l'utilisateur créé avant, TOTO :

```
SQL> GRANT ROLE_CNAM TO TOTO;

Autorisation de privilèges (GRANT) acceptée.
```

On peut vérifier via un SELECT la liste des rôles attribués à l'utilisateur TOTO :

```
SQL> SELECT GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTEE='TOTO';

GRANTED_ROLE
-----
ROLE_CNAM
```

LES PROFILS :

Affichage du nombre de sessions autorisés pour le profil par défaut :

```
SQL> SELECT LIMIT FROM DBA_PROFILES WHERE RESOURCE_NAME='SESSIONS_PER_USER' AND PROFILE='DEFAULT';

LIMIT
-----
UNLIMITED
```

Nombres de connexions autorisés :

```
SQL> SELECT LIMIT FROM DBA_PROFILES WHERE RESOURCE_NAME='FAILED_LOGIN_ATTEMPTS' AND PROFILE='DEFAULT';

LIMIT
-----
10
```

Création du profil\_CNAM avec paramètres demandés :

```
SQL> CREATE PROFILE PROFIL_CNAM LIMIT SESSIONS_PER_USER 3 CONNECT_TIME 10 IDLE_TIME 5 FAILED_LOGIN_ATTEMPTS 2;

Profil créé.
```

Il nous suffit maintenant d'attribuer le profil à l'utilisateur TOTO via la commande ALTER USER TOTO PROFILE PROFIL\_CNAM :

```
SQL> ALTER USER TOTO PROFILE PROFIL_CNAM;

Utilisateur modifié.
```