

PROJET ADMINISTRATION DES BASES DE DONNEES ORACLE 10G

TP N°1

I-/

- 1. CREATE USER Gest_Vols IDENTIFIED BY password;
GRANT DBA TO Gest_Vols;
- 2. CONNECT Gest_Vols/password;

```
CREATE TABLE Pilote (  
    NP NUMBER PRIMARY KEY,  
    Nom VARCHAR2(25),  
    Dnaiss DATE,  
    Adr VARCHAR2(40),  
    Tel NUMBER(10),  
    Salaire NUMBER(7),  
    Age NUMBER(2)  
);
```

```
CREATE TABLE Avion (  
    NA NUMBER PRIMARY KEY,  
    Nom VARCHAR2(12),  
    Capacite NUMBER(6),  
    Localite VARCHAR2(10),  
    Remarq VARCHAR2(100)  
);
```

```
CREATE TABLE Vol(  
    NV NUMBER(6) PRIMARY KEY,  
    NP NUMBER,  
    NA NUMBER,  
    VD VARCHAR2(10),  
    VA VARCHAR(10),  
    HD NUMBER,  
    HA NUMBER,  
    DATE_VOL DATE  
    CONSTRAINT Fk_Pilote FOREIGN KEY(NP) REFERENCES Pilote(NP),  
    CONSTRAINT Fk_Avion FOREIGN KEY(NA) REFERENCES Avion(NA)  
);
```

3. INSERT INTO Pilote VALUES(1, 'Miranda', '18-08-52','Sophia-Antipolis', '93548254', 18009.0,50);
INSERT INTO Pilote VALUES(2, 'St-exupery', '16-10-32', 'Lyon', '91548254', 12300.0,50);
INSERT INTO Avion VALUES(1, 'A300', 300, 'Nice', 'En service');
INSERT INTO Avion VALUES(2, 'A300', 300, 'Nice', 'En service');
INSERT INTO Vol VALUES(100, 1,1,'Nice', 'Paris', '1345', '1500','3-03-89');
INSERT INTO Vol VALUES(110, 3,6,'Nice', 'Toulouse', '1230', '1325','6-03-89');

II-/

1. INSERT INTO Avion VALUES(100, 'AIRBUS', 300, 'LOME', 'En service');
INSERT INTO Avion VALUES(101, 'B737', 250, 'CASA', 'En service');
INSERT INTO Avion VALUES(102, 'B737', 220, 'DAKAR', 'En service');

2. DELETE FROM Pilote WHERE NP=5;
DELETE FROM Pilote WHERE NP=10;

3. UPDATE Pilote SET Salaire=Salaire+Salaire*0.25 WHERE NP=20;

4. UPDATE Pilote SET Salaire=Salaire-Salaire*0.11 WHERE NP=15;

III-/ Requêtes

1. SELECT * FROM Avion;

2. SELECT * FROM Avion ORDER BY Nom ASC;

3. SELECT Nom, Capacite FROM Avion;

4. SELECT DISTINCT(Localite) FROM Avion;

5. SELECT Vol.NV NumVol, Avion.NA NumAvions, Pilote.Nom NomPilotes FROM Vol v, Avion a, Pilote p WHERE v.NV = p.NP;

6. SELECT * FROM Avion WHERE Localite='LOME' OR Localite='CASA';

7. UPDATE Avion SET Capacite=220 WHERE NA=101;

8. DELETE FROM Avion WHERE Capacite<200;

9. SELECT MIN(Capacite) MINIMAL, MAX(Capacite) MAXIMAL, AVG(Capacite) MOYENNE FROM Avion;
10. SELECT * FROM Avion WHERE Capacite=MIN(Capacite);
11. SELECT * FROM Avion WHERE Capacite>AVG(Capacite);
12. SELECT Nom, Adr FROM Pilote, Vol WHERE Pilote.NP=Vol.NP AND NV='IT100' AND NV='IT104';
13. SELECT NP FROM Vol;
14. SELECT NP FROM Pilote WHERE NP NOT IN (SELECT NP FROM Vol);
15. SELECT Pilote.Nom FROM Pilote, Vol, Avion WHERE Pilote.NP=Vol.NP AND Avion.NA=Vol.NA AND Avion.Nom='AIRBUS';

Remarques : Nous avons relevé des erreurs dans l'exercice.

- La clé primaire de la table Vol est de type texte alors que dans les insertions, les clés primaires sont de type de NUMBER

- Le nombre de colonne de la table Vol est 7 mais dans les insertions, le nombre de colonne est 8. La dernière colonne dans les insertions qui contient les dates ne se trouve pas dans la structure de la table. Nous avons ajouté une colonne DATE à la structure de la table afin de pouvoir faire les insertions.

TP N°2

1. Schéma de relation correspondant à DICT :

DESCRIBE DICT;

2. Visualisation du contenu complet de DICT à l'aide de la commande SELECT :

SELECT * FROM DICT;

Résultats : 1870 lignes sélectionnées.

3. Rôles et structures des tables et vues ALL_CATALOG, ALL_USERS, ALL_COL_COMMENTS, ALL_CONSTRAINTS et ALL_TAB_PRIVS :

* ALL_CATALOG liste tous les index, les tables, les vues, les clusters, les synonymes et les séquences accessibles à l'utilisateur courant.

Visualiser la structure de la table ALL_CATALOG : DESCRIBE ALL_CATALOG;

* ALL_USERS répertorie tous les utilisateurs de la base de données visibles par l'utilisateur courant.

Visualiser la structure de la table ALL_USERS : DESCRIBE ALL_USERS;

* ALL_COL_COMMENTS affiche tous les commentaires sur les colonnes des tables et vues accessibles à l'utilisateur courant.

Visualiser la structure de la table ALL_COMMENTS : DESCRIBE ALL_COL_COMMENTS;

* ALL_CONSTRAINTS décrit les définitions de contraintes sur les tables accessibles à l'utilisateur courant.

Visualiser la structure de la table ALL_CONSTRAINTS : DESCRIBE ALL_CONSTRAINTS;

* ALL_TAB_PRIVS décrit les types de subventions suivantes :

- subventions d'objets pour lesquels l'utilisateur actuel est le propriétaire, l'octroi ou le bénéficiaire de l'objet.
- subventions d'objets pour lesquels un rôle activé ou PUBLIC est le bénéficiaire.

Visualiser la structure de la table DESCRIBE ALL_TAB_PRIVS : DESCRIBE ALL_TAB_PRIVS;

4. La requête SQL pour voir les types d'objets d'une base de données est simple car cette information est stockée dans la colonne OBJECT_TYPE de la vue DBA_OBJECTS.

Requête: SELECT OBJECT_TYPE, COUNT(*) FROM DBA_OBJECTS GROUP BY
OBJECT_TYPE ORDER BY COUNT(*) DESC;

5. Nombre d'objets référencés dans les tables ALL_CATALOG, ALL_TABLES, ALL_VIEWS, ALL_SYNONYMS et ALL_SEQUENCES.

SELECT COUNT(*) FROM ALL_CATALOG;

Résultat : 25439 éléments

SELECT COUNT(*) FROM ALL_VIEWS;

Résultat : 3670 éléments

SELECT COUNT(*) FROM ALL_TABLES;

Résultat : 1600 éléments

SELECT COUNT(*) FROM ALL_SYNONYMS;

Résultat : 20026 éléments

SELECT COUNT(*) FROM ALL_SEQUENCES;

Résultat : 141

6. Schéma de relation correspondant à la table USER_USERS : DESCRIBE USER_USERS;

Pour avoir la liste de tous les utilisateurs :

SELECT username, password FROM DBA_USERS;

Pour voir le nom de l'utilisateur courant : SELECT USER FROM DUAL;

7. Comparaison des contenus des tables ALL_CATALOG et USER_CATALOG

```
SELECT COUNT(*) FROM ALL_CATALOG;
```

Résultat : 25439 éléments

```
SELECT COUNT(*) FROM USER_CATALOG;
```

Résultat : 3701 éléments

La table ALL_CATALOG a 25439 enregistrements tandis que la table USER_CATALOG a 3701 enregistrements.

```
DESCRIBE ALL_CATALOG;
```

```
DESCRIBE USER_CATALOG;
```

Dans la table ALL_CATALOG, nous avons USER_CATALOG en tant que Vue (VIEW) et USER_CATALOG en tant que SYNONYM.

Par contre dans la table USER_CATALOG, nous avons ALL_CATALOG en tant que VIEW seulement.

8. Ensemble des tables ayant un même propriétaire.

```
SELECT owner, table_name FROM ALL_TABLES;
```

Cette requête affiche uniquement les tables qui sont accessibles par l'utilisateur courant selon qu'il soit administrateur ou simple utilisateur.

TP N°3

1. CONNECT sys/toor as sysdba;
2. CREATE USER User_Oracle10G IDENTIFIED BY Oracle10G;
3. GRANT DBA TO User_Oracle10G;
4. ALTER USER User_Oracle10G IDENTIFIED BY SYSOracle10G;
5. ALTER USER User_Oracle10G ACCOUNT LOCK;
6. ALTER USER User_Oracle10G ACCOUNT UNLOCK;

TP N°4

Rôle des composants de l'architecture Oracle :

Serveur Oracle = instance Oracle + base de données Oracle

Instance Oracle : c'est un moyen pour accéder à une base de données Oracle. Elle est composée de la mémoire SGA, cette dernière renferme d'autres composants.

La SGA (System Global Area) représente la zone mémoire déterminante d'une instance, tant par sa taille que par son rôle. Elle est composée de :

Shared pool : permet de stocker les informations comme les opérations SQL récentes et les données du dictionnaire de données récemment utilisées.

Database Buffer cache : est utilisé pour stocker des blocs de données en mémoire afin d'accélérer l'interrogation et /ou la modification

Buffer Redo Log : c'est un buffer circulaire qui stocke les modifications réalisées sur la base de données avec les opérations INSERT, DELETE, UPDATE, CREATE, ALTER et DROP. Il permet à Oracle de reconstruire les modifications des données en cas de panne. L'information Redo reste dans le buffer Redo log jusqu'à ce qu'Oracle la stocke sur le disque. Sa taille est définie par LOG_BUFFER.

Zone de mémoire LARGE POOL : zone facultative qui fournit des espaces importants d'allocation de mémoire à des processus volumineux, tels que les opérations de sauvegarde et de récupération Oracle et les processus serveur d'E/S.

Zone de mémoire Java : Zone de mémoire nécessaire pour la machine virtuelle Java intégrée dans Oracle. Cette zone permet d'exécuter le code Java stocké dans le noyau Oracle.