

**Filière :** Ingénierie de données (DI1-S2) **Module :** Administration et optimisation de données avancées

## Série n° 2

**Partie 1 :** le but général de ce TP est :

- La création des utilisateurs, l'attribution des privilèges, des rôles à des utilisateurs,....
- L'interrogation d'une base de données relationnelle (jointures, fonctions SQL...).

Pour cela, on considère la base de données suivante fournie par Oracle, toutes les tables sont à recopier dans votre compte à partir du schéma *Scott* par : *CREATE TABLE emp as SELECT \* FROM scott.emp;*

**EMP** de clé primaire *EmpNo* et de clé étrangère *DeptNo*

| EMPNO | ENAME  | JOB      | MGR  | HIREDATE | SAL  | COMM | DEPTNO |
|-------|--------|----------|------|----------|------|------|--------|
| 7369  | SMITH  | CLERK    | 7902 | 17/12/80 | 800  |      | 20     |
| 7499  | ALLEN  | SALESMAN | 7698 | 20/02/81 | 1600 | 300  | 30     |
| 7521  | WARD   | SALESMAN | 7698 | 22/02/81 | 1250 | 500  | 30     |
| 7566  | JONES  | MANAGER  | 7839 | 02/04/81 | 2975 |      | 20     |
| 7654  | MARTIN | SALESMAN | 7698 | 28/09/81 | 1250 | 1400 | 30     |

### DEPT

| DEPTNO | DNAME      | LOC      |
|--------|------------|----------|
| 10     | ACCOUNTING | NEW YORK |
| 20     | RESEARCH   | DALLAS   |
| 30     | SALES      | CHICAGO  |
| 40     | OPERATIONS | BOSTON   |

### SalGrade

| GRADE | LOSAL | HISAL |
|-------|-------|-------|
| 1     | 700   | 1200  |
| 2     | 1201  | 1400  |
| 3     | 1401  | 2000  |
| 4     | 2001  | 3000  |
| 5     | 3001  | 9999  |

### Données pour les différentes tables :

**Employé(EMP) EMPNO :** numéro, **ENAME :** nom, **JOB :** profession, **MGR :** chef, **HIREDATE :** date d'embauche, **SAL :** salaire, **COMM :** commission, **DEPTNO :** numéro du département où l'employé est affecté

**Département (DEPT) :** **DEPTNO :** numéro, **DNAME :** nom, **LOC :** adresse

**Catégorie Salaire (SALGRADE) :** **GRADE :** catégorie, **LOSAL :** salaire minimum, **HISAL :** salaire maximum

1. Connectez-vous au compte exemple *SCOTT* et affichez les tables : *[SELECT TABLE\_NAME FROM USER\_TABLES;]*

- Créez un utilisateur *USER1* auquel vous donnerez le droit *SELECT* sur la table *emp*; à partir du compte *SCOTT* [*Grant select on emp to user1*, puis copiez la table *emp* dans le compte *USER1*
  2. Donner la liste des employés ayant des salaires supérieurs à 1000
  3. Donner la liste des employés travaillant dans le département *SALES*
  4. Donner le nom de chaque employé ainsi que son salaire total (incluant sa commission)
  5. Afficher les noms des employés et leurs commissions. Si un employé n'a pas de commission, afficher 'Sans commission'. Etiqueter cette colonne *Commission* (alias).
  6. Afficher les noms des employés, leurs dates d'embauche et la date de la première négociation de leurs salaires, qui est le premier Lundi après 6 mois de travail, nommer cette colonne « Date Négociation ».
  7. Afficher pour chaque employé le nom, la date d'embauche et le jour de la semaine auquel il a commencé à travailler. Ordonner le résultat suivant le jour.
  8. Un fichier *fichier1.sql* que vous créerez, sera appelé pour exécution par '@fichier1' avec son chemin d'accès complet.
  9. Ecrire la requête qui affiche le nom de chaque employé et le nom et le code de son chef. Appeler les colonnes *Employé*, *NoEmp*, *Chef*, *NoChef*. Insérer cette requête dans *fichier1.sql* à exécuter.
  10. Afficher tous les employés, y compris King (qui n'a pas de chef).
  11. Afficher le nom, le métier, le nom du département, le salaire et la catégorie du salaire de chaque employé. (On utilisera aussi la table *SALGRADE*).
  12. **Afficher par métier** la somme des salaires pour chaque département, ainsi que le total des salaires de tous les départements, en utilisant 2 méthodes : *GROUP BY*, *DECODE*. On ne considèrera que les départements qui contiennent des employés des départements : 10, 20 et 30.
- | Job     | Dep10 | Dep20 | Dep30 | Total |
|---------|-------|-------|-------|-------|
| Analyst |       |       |       |       |
| Manager |       |       |       |       |
13. Obtenir les numéros des départements où il n'y a pas d'employés, en utilisant la clause *MINUS*.
  14. Donner le nom du plus ancien employé.
  15. Afficher par ordre croissant les noms des 3 employés qui ont les salaires les plus élevés.
  16. Afficher des informations complètes concernant les départements où il n'y a aucun vendeur (salesman)
  17. Afficher le numéro et le nom des employés qui gagnent plus que le salaire moyen et qui travaillent dans le même département qu'au moins un employé dont le nom contient la lettre 'T'.