

## TP4 – Base de Données

Rendre le TP avant minuit en le chargeant sur *Tomuss*. Le rapport doit contenir vos requêtes SQL et/ou vos réponses aux questions, avec vos commentaires/explications et captures d'écran.

### Exercice 1

On considère la base de données de gestion des salariés suivante :

**EMP** (**EmpNo**, **EName**, **Job**, **#MGR**, **HireDate**, **Sal**, **Comm**, **#DeptNo**)  
**DEPT** (**DeptNo**, **DName**, **Loc**, **NbEmp**)  
**SALGRADE** (**Grade**, **LoSal**, **HiSal**)

Le contenu des différentes tables :

**EMP** (**EmpNo**, **EName**, **Job**, **#MGR**, **HireDate**, **Sal**, **Comm**, **#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     |
| 7698  | BLAKE  | MANAGER   | 7839 | 01/05/81 | 2850 | -    | 30     |
| 7782  | CLARK  | MANAGER   | 7839 | 09/06/81 | 2450 | -    | 10     |
| 7788  | SCOTT  | ANALYST   | 7566 | 19/04/87 | 3000 | -    | 20     |
| 7839  | KING   | PRESIDENT | -    | 17/11/81 | 5000 | -    | 10     |
| 7844  | TURNER | SALESMAN  | 7698 | 08/09/81 | 1500 | 0    | 30     |
| 7876  | ADAMS  | CLERK     | 7788 | 23/05/87 | 1100 | -    | 20     |
| 7900  | JAMES  | CLERK     | 7698 | 03/12/81 | 950  | -    | 30     |
| 7902  | FORD   | ANALYST   | 7566 | 03/12/81 | 3000 | -    | 20     |
| 7934  | MILLER | CLERK     | 7782 | 23/01/82 | 1300 | -    | 10     |

**DEPT** (**DeptNo**, **DName**, **Loc**)

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

**SALGRADE (Grade, LoSal, HiSal)**

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

**N.B. :** un employé est identifié par son numéro. Un employé peut avoir un supérieur direct qui est aussi un employé. Chaque employé est associé à un département. Un département est identifié par son numéro. Un grade est déterminé par son numéro.

Exprimez les requêtes suivantes en langage SQL.

1. Créez les structures des tables et récupérer leurs contenus en exécutant le script TP4.sql. Ajouter les clés primaires et les clés étrangères en utilisant la syntaxe `ALTER TABLE ... ADD`.
2. Listez tous les employés.
3. Donner la liste des employés dont le salaire est inférieur à 1000.
4. Donnez la liste des employés qui touchent un salaire supérieur à celui de tous les employés du département 20.
5. Quels sont les employés qui ont le même job et le même salaire que l'employé FORD ?
6. Listez les employés dont la commission est inférieure à 25% du salaire.
7. Quel est le total des salaires des employés du département 30 ?
8. Donnez le nom et le job de l'employé ayant le salaire le plus élevé.
9. Quel est le nombre d'employés du département 10.
10. Donnez, pour chaque département, le salaire le plus élevé.
11. Donnez les salaires moyens par département (et par job).
12. Quel est le nombre d'employés du département ayant le plus d'employés ?
13. Quels sont le numéro et le nom de l'employé qui dirige le plus grand nombre d'employés ?
14. Quel est le département ayant le plus d'employés ?
15. Quels sont les départements qui n'ont pas d'employés ?
16. Donnez les noms des employés qui ont un salaire supérieur à la moyenne des salaires de leurs départements.
17. Donnez le nom du plus ancien employé de chaque département.

18. Affichez par ordre croissant (salaire) les noms des 3 employés qui ont les plus hauts salaires.
19. Affichez les noms des départements qui ont plus de 3 employés.
20. Affichez les noms de départements et les catégories de salaire qui s'y trouvent.
21. Donnez le nom du département et le plus petit salaire du département ayant le plus grand salaire moyen.
22. Affichez des informations complètes concernant les départements où il n'y a aucun vendeur (salesman).
23. Affichez le code et le nom des employés qui travaillent dans les départements qui ont moins de 4 employés.
24. Créez une vue contenant les employés du département 30. Affichez son contenu.

## Exercice 2

Le but de cet exercice est de mettre en place les tables de la base de données simplifiée suivante du système de gestion des notes des étudiants :

```

Etudiant(numE, nom, prenom, dateNais, email)
Cours(numC, formation, nom, nbcredit)
Notes(#numE, #numC, notefinal)
Statut(#numE, annee, formation, statut)

```

La définition des différents champs de la base :

| Nom       | Signification                                | Type        |
|-----------|----------------------------------------------|-------------|
| NUME      | Numéro de l'étudiant                         | Number(8)   |
| NOM       | Nom de l'étudiant                            | Varchar(20) |
| PRENOM    | Prénom de l'étudiant                         | Varchar(20) |
| DATENAIS  | Date de naissance de l'étudiant              | Date        |
| EMAIL     | L'email de l'étudiant                        | Varchar(30) |
| ANNEE     | Début de l'année scolaire                    | Number(4)   |
| FORMATION | La formation que l'étudiant a suivie         | Varchar(20) |
| STATUT    | Statut de la formation suivie par l'étudiant | Varchar(10) |
| NUMC      | Numéro du cours                              | Number(4)   |
| NOM       | Nom du cours                                 | Varchar(20) |
| NBCREDIT  | Nombre de crédits                            | Number(2)   |
| NOTEFINAL | Note finale du cours                         | Number(2)   |

Le contenu des différentes tables :

```

Etudiant(numE, nom, prenom, dateNais, email)

```

| NumE    | Nom     | Prenom   | DateNais   | EMail                  |
|---------|---------|----------|------------|------------------------|
| 3999133 | Martin  | Amandine | 04-10-1990 | martin.a@hotmail.fr    |
| 3891012 | Denis   | Nicolas  | 21-03-1991 | nicolasd@gmail.com     |
| 3996423 | Clément | Jacques  | 01-12-1993 | jacquesc@yahoo.com     |
| 3984125 | Benoit  | Amélie   | 09-05-1992 | -                      |
| 4000014 | Hubert  | Emmanuel | 18-08-1993 | mhubert@hotmail.com    |
| 4000021 | Collet  | Sarah    | 30-09-1994 | sarah.collet@gmail.com |
| 3997862 | Vincent | Georges  | 15-04-1993 | gvincent@hotmail.com   |

**Cours(numC, formation, nom, nbcredit)**

| NumC | Formation       | Nom                       | NbCredit |
|------|-----------------|---------------------------|----------|
| 1    | L1-Informatique | Base de données           | 6        |
| 2    | L1-Informatique | Programmation             | 6        |
| 3    | L1-Informatique | Algèbre                   | 3        |
| 4    | L1-Informatique | Systèmes d'exploitation   | 3        |
| 5    | L2-Informatique | Génie logiciel            | 3        |
| 6    | L2-Informatique | Analyse de données        | 6        |
| 7    | L2-Informatique | Web                       | 6        |
| 8    | L3-Informatique | Intelligence artificielle | 3        |
| 9    | L3-Informatique | Projet                    | 12       |

**Notes(#numE, #numC, notefinal)**

| NumE    | NumC | NoteFinal |
|---------|------|-----------|
| 4000014 | 5    | 15        |
| 4000014 | 7    | 12        |
| 3999133 | 8    | 11        |
| 3999133 | 9    | 7         |
| 3891012 | 8    | 12        |
| 3891012 | 9    | 14        |
| 3996423 | 1    | 10        |
| 3996423 | 2    | 15        |
| 3996423 | 3    | 17        |
| 3996423 | 4    | 9         |
| 3984125 | 5    | 12        |
| 3984125 | 7    | 5         |
| 4000021 | 1    | 15        |
| 4000021 | 2    | 10        |
| 4000021 | 3    | 6         |
| 4000021 | 4    | 11        |
| 3997862 | 2    | 16        |
| 3997862 | 3    | 11        |

**Statut(#numE, annee, formation, statut)**

| NumE    | Annee | Formation       | Statut   |
|---------|-------|-----------------|----------|
| 4000014 | 2011  | L1-Informatique | Validé   |
| 4000014 | 2012  | L2-Informatique | En_cours |
| 3999133 | 2012  | L3-Informatique | En_cours |
| 3891012 | 2012  | L3-Informatique | En_cours |
| 3996423 | 2011  | L1-Informatique | Nvalidé  |
| 3984125 | 2012  | L2-Informatique | En_cours |
| 4000021 | 2012  | L1-Informatique | En_cours |
| 3997862 | 2012  | L1-Informatique | En_cours |

Exprimez les requêtes suivantes en langage SQL.

1. Définissez l'ordre chronologique de la création des tables.
2. Créez toutes les tables avec les contraintes d'intégrités.
3. Saisissez, table par table, le contenu de toutes les tables.
4. Donnez la note 15 à l'étudiant « Georges » pour le cours « Base de données »
5. Donnez le nom et la note du meilleur étudiant pour chaque cours.
6. Supprimez le cours « Web » et toutes les notes de ce cours.
7. Modifiez le statut de l'étudiant de l'année actuelle en « Validé » si la moyenne des notes de la formation (que l'étudiant suit) est supérieur à 10 et aucune note est inférieur à 8.  
**Attention !** Il faut vérifier que toutes les notes correspondantes au cours de la formation suivie sont disponibles.