

Rapport TP 1 – Base de données avancées

Nom : Lyes Ahfir
Date : 30 avril 2025

Exercice 1 :

1. Définition de la table **section**

Pour modéliser les sections de cours, on crée la table `section` avec le schéma suivant :

```
create table section(  
    course_id    varchar(8),  
    sec_id       varchar(8),  
    semester     varchar(6),  
    check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),  
    year         numeric(4,0),  
    building     varchar(15),  
    room_number  varchar(7),  
    time_slot_id varchar(4),  
    primary key (course_id, sec_id, semester, year),  
    foreign key (course_id) references course,  
    foreign key (building, room_number) references classroom  
);
```


Exercice 2 :

Cet exercice consiste à interroger la base de données relationnelle à l'aide de requêtes SQL afin d'extraire des informations sur les cours, les enseignants, les étudiants et les départements.

1. Afficher la structure et le contenu de la relation `section` :

```
DESC section;
SELECT * FROM section;
```

2. Afficher les renseignements sur les cours (relation `course`) :

```
SELECT * FROM course;
```

3. Afficher les titres des cours et les départements qui les proposent :

```
SELECT title, dept_name FROM course;
```

4. Afficher les noms des départements et leur budget :

```
SELECT dept_name, budget FROM department;
```

5. Afficher les noms des enseignants et leur département :

```
SELECT name, dept_name FROM teacher;
```

6. Enseignants avec un salaire strictement supérieur à 65 000 \$:

```
SELECT name FROM teacher WHERE salary > 65000;
```

7. Enseignants avec un salaire entre 55 000 \$ et 85 000 \$:

```
SELECT name FROM teacher
WHERE salary BETWEEN 55000 AND 85000;
```

8. Noms des départements distincts depuis la table `teacher` :

```
SELECT DISTINCT dept_name FROM teacher;
```

9. Enseignants du département Informatique avec salaire $> 65\,000$ \$:

```
SELECT name FROM teacher
WHERE salary > 65000 AND dept_name = 'Comp. Sci.';
```

10. Informations sur les cours proposés au printemps 2010 :

```
SELECT * FROM section
WHERE semester = 'Spring' AND year = 2010;
```

11. Titres des cours du département Informatique avec plus de 3 crédits :

```
SELECT title FROM course
WHERE dept_name = 'Comp. Sci.' AND credit > 3;
```

12. Noms des enseignants, leur département et le bâtiment associé :

```
SELECT teacher.name, teacher.dept_name, department.building
FROM teacher, department
WHERE teacher.dept_name = department.dept_name;
```

13. Étudiants ayant suivi au moins un cours en informatique :

```
SELECT DISTINCT student.name
FROM student, takes, course
WHERE student.id = takes.id
      AND takes.course_id = course.course_id
      AND course.dept_name = 'Comp. Sci.';
```

14. Étudiants ayant suivi un cours assuré par Einstein :

```
SELECT DISTINCT student.name
FROM student, teacher, takes, teaches
WHERE student.id = takes.id
      AND takes.course_id = teaches.course_id
      AND takes.sec_id = teaches.sec_id
      AND takes.semester = teaches.semester
      AND takes.year = teaches.year
      AND teaches.id = teacher.id
      AND teacher.name = 'Einstein';
```

15. Identifiants des cours et enseignants correspondants :

```
SELECT name, course_id
FROM teacher, teaches
WHERE teacher.id = teaches.id;
```

16. Nombre d'inscrits par enseignement au printemps 2010 :

```
SELECT takes.course_id, takes.sec_id, takes.semester, takes.year, COUNT(*)
FROM takes
WHERE takes.semester = 'Spring' AND takes.year = 2010
GROUP BY takes.course_id, takes.sec_id, takes.semester, takes.year;
```

17. Noms des départements et salaires maximum :

```
SELECT dept_name, MAX(salary) FROM teacher
GROUP BY dept_name;
```

18. Nombre d'inscrits pour chaque enseignement proposé :

```
SELECT takes.course_id, takes.sec_id, takes.semester, takes.year, COUNT(*)
FROM takes
GROUP BY takes.course_id, takes.sec_id, takes.semester, takes.year;
```

19. Nombre total de cours par bâtiment pour automne 2009 et printemps 2010 :

```
SELECT building, COUNT(*)
FROM section
WHERE (semester, year) IN (('Fall', 2009), ('Spring', 2010))
GROUP BY building;
```

20. Nombre de cours dispensés par chaque département dans leur bâtiment :

```
SELECT department.dept_name, COUNT(*)
FROM section, department, teacher, teaches
WHERE (section.course_id = teaches.course_id
      AND section.sec_id = teaches.sec_id
      AND section.semester = teaches.semester
      AND section.year = teaches.year
      AND teaches.id = teacher.id
      AND teacher.dept_name = department.dept_name
      AND department.building = section.building)
GROUP BY department.dept_name;
```

21. Titres des cours proposés et les enseignants correspondants :

```
SELECT course.title, teacher.name
FROM section, teacher, teaches, course
WHERE (section.course_id = teaches.course_id
      AND section.sec_id = teaches.sec_id
      AND section.semester = teaches.semester
      AND section.year = teaches.year
      AND teaches.id = teacher.id
      AND section.course_id = course.course_id)
ORDER BY course.title;
```

22. Nombre total de cours par semestre :

```
SELECT section.semester, COUNT(*) FROM section
GROUP BY section.semester;
```

23. Crédits obtenus par chaque étudiant sur des cours hors de son département :

```
SELECT student.name, SUM(credits)
FROM student, course, takes
WHERE student.id = takes.id
      AND takes.course_id = course.course_id
      AND student.dept_name != course.dept_name
GROUP BY student.name;
```

24. Crédits totaux par bâtiment pour chaque département :

```
SELECT section.building, SUM(course.credits)
FROM section, course
WHERE section.course_id = course.course_id
GROUP BY section.building;
```