

Exercice N°:4

- 1) Planification des cours : les horaires prévus et les enseignements programmés pour chaque jour de la semaine peuvent être utilisés pour planifier les cours à venir. Les enseignants peuvent visualiser leur emploi du temps et préparer leurs cours en conséquence.

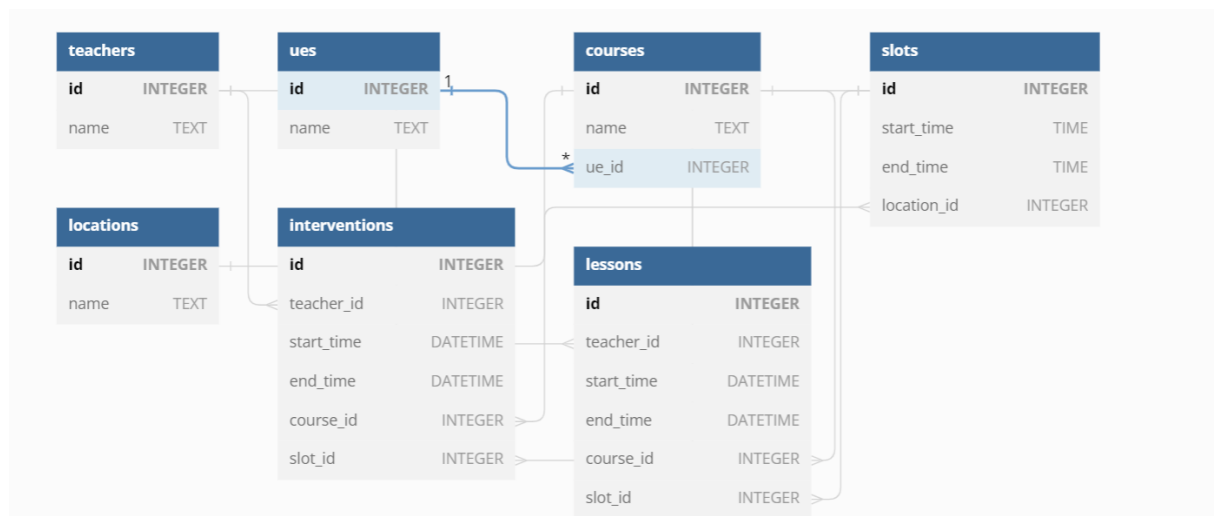
Suivi des heures de cours : les heures réalisées permettent de suivre le nombre d'heures de cours effectuées pour chaque enseignement. Cela peut être utile pour calculer les salaires des enseignants et évaluer la charge de travail de chaque enseignant.

Gestion des salles de classe : la base de données contient également des informations sur les salles de classe où se déroulent les cours. Cette information peut être utilisée pour planifier l'utilisation des salles de classe et éviter les conflits de réservation.

Suivi des enseignants : les coordonnées des enseignants sont incluses dans la base de données, ce qui peut être utile pour contacter les enseignants en cas de besoin. Cette information peut également être utilisée pour évaluer la charge de travail de chaque enseignant et assurer une répartition équitable des enseignements.

Analyse des performances des étudiants : en utilisant les données sur les enseignements et les heures réalisées, il est possible d'analyser les performances des étudiants en fonction du nombre d'heures de cours suivies pour chaque enseignement. Cette analyse peut aider à identifier les enseignements qui nécessitent une attention particulière et les étudiants qui ont besoin d'un soutien supplémentaire.

2) Schéma de la BDD



3) Ajouter des entrées dans la base de données de la semaine 2 de votre calendrier

```
INSERT INTO slots (date, time_start, time_end, duration, lesson_id,
teacher_id, location_id)
VALUES
('2023-01-09', '09:00:00', '12:30:00', '03:30:00', 7, 4, 3),
('2023-01-09', '13:30:00', '17:00:00', '03:30:00', 8, 4, 3),
('2023-01-10', '09:00:00', '12:30:00', '03:30:00', 7, 4, 3),
('2023-01-10', '13:30:00', '17:00:00', '03:30:00', 8, 4, 3),
('2023-01-11', '09:00:00', '12:30:00', '03:30:00', 7, 4, 3),
('2023-01-11', '13:30:00', '17:00:00', '03:30:00', 8, 4, 3),
('2023-01-12', '09:00:00', '12:30:00', '03:30:00', 7, 4, 3),
('2023-01-12', '13:30:00', '17:00:00', '03:30:00', 8, 4, 3),
('2023-01-13', '09:00:00', '12:30:00', '03:30:00', 9, 5, 1),
('2023-01-13', '13:30:00', '17:00:00', '03:30:00', 10, 5, 1);
```

4) Ecrivez les requêtes SQL permettant de répondre aux questions suivantes

4.1) Écrivez une requête pour récupérer les noms de tous les enseignants et les noms des cours qu'ils ont enseignés dans la table lessons, triés par ordre alphabétique du nom de l'enseignant.

```
SELECT teachers.name AS enseignant, lessons.name AS cours
FROM lessons
JOIN teachers ON lessons.teacher_id = teachers.id
ORDER BY enseignant ASC;
```

4.2) Écrivez une requête pour récupérer les noms de tous les cours, les noms des UE auxquels ils appartiennent et le nombre d'évaluations de chaque UE dans les tables courses et ues.

```
SELECT courses.name AS cours, ues.name AS ue, COUNT(DISTINCT
evaluations.id) AS nb_evaluations
FROM courses
JOIN ues ON courses.ue_id = ues.id
LEFT JOIN evaluations ON evaluations.course_id = courses.id
GROUP BY courses.id, ues.id
ORDER BY ue ASC, cours ASC;
```

4.3) Écrivez une requête pour récupérer les noms de tous les créneaux et les noms des emplacements où des cours sont planifiés dans les tables slots et locations.

```
SELECT slots.name AS creneau, locations.name AS emplacement
FROM slots
JOIN locations ON slots.location_id = locations.id;
```

4.4) Écrivez une requête pour compter le nombre de cours enseignés dans chaque emplacement, trié par le nombre de cours par ordre décroissant.

```
SELECT locations.name AS emplacement, COUNT(*) AS nb_cours
FROM slots
JOIN locations ON slots.location_id = locations.id
GROUP BY locations.id
ORDER BY nb_cours DESC;
```

4.5) Écrivez une requête pour récupérer les noms de tous les enseignants et le nombre total d'heures qu'ils ont enseignées dans des interventions et des cours dans les tables teachers, interventions et lessons, trié par le nombre total d'heures par ordre décroissant.

```
SELECT teachers.name AS enseignant,
       SUM(TIME_TO_SEC(TIMEDIFF(interventions.end_time,
                                interventions.start_time)) + TIME_TO_SEC(TIMEDIFF(lessons.end_time,
                                lessons.start_time))) / 3600 AS nb_heures_enseignees
FROM teachers
LEFT JOIN interventions ON interventions.teacher_id = teachers.id
LEFT JOIN lessons ON lessons.teacher_id = teachers.id
GROUP BY teachers.id
ORDER BY nb_heures_enseignees DESC;
```