

# tp\_gestion\_cours

khadija bouargalne

January 2025

## 1 Introduction

Ce rapport technique décrit la création et la gestion d'une base de données nommée `gestion_cours`. Cette base de données permet de gérer les informations relatives aux étudiants, professeurs, cours et inscriptions. Le projet inclut la conception des tables, l'insertion des données, et l'exécution de requêtes SQL spécifiques.

## 2 Création de la Base de Données et des Tables

### 2.1 Création de la Base de Données

```
CREATE DATABASE gestion_cours;  
USE gestion_cours;
```

### 2.2 Création des Tables

```
CREATE TABLE etudiants (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nom VARCHAR(20),  
    prenom VARCHAR(20),  
    email VARCHAR(40)  
);
```

```
CREATE TABLE professeurs (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nom VARCHAR(20),  
    prenom VARCHAR(20),  
    email VARCHAR(40)  
);
```

```
CREATE TABLE cours (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    titre VARCHAR(20),
```

```

        date_debut DATE,
        date_fin DATE,
        id_professeur INT,
        FOREIGN KEY (id_professeur) REFERENCES professeurs(id)
    );

CREATE TABLE inscriptions (
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_etudiant INT,
    id_cours INT,
    FOREIGN KEY (id_etudiant) REFERENCES etudiants(id),
    FOREIGN KEY (id_cours) REFERENCES cours(id)
);

```

### 3 Insertion des Données

```

INSERT INTO etudiants (nom, prenom, email) VALUES
    ('khadija', 'bouargalne', 'khadija123@gmail.com'),
    ('fatima', 'hasnaoui', 'fati123@gmail.com'),
    ('hiba', 'sahal', 'hiba2006@gmail.com');

INSERT INTO professeurs (nom, prenom, email) VALUES
    ('mohamed', 'hano', 'mohamed123@gmail.com'),
    ('fatima', 'halimi', 'fatiha123@gmail.com'),
    ('asmae', 'dawi', 'asma2006@gmail.com');

INSERT INTO cours (titre, date_debut, date_fin, id_professeur) VALUES
    ('math', '2023-02-01', '2023-04-05', 1),
    ('physique', '2024-09-01', '2024-11-05', 2),
    ('info', '2025-02-02', '2025-04-04', 3);

INSERT INTO inscriptions (id_etudiant, id_cours) VALUES
    (1, 1),
    (2, 2),
    (3, 3);

```

## 4 Requêtes SQL

### 4.1 Requête 1

Afficher tous les cours dont la date de fin est après le 1er janvier 2023 et qui ont un titre contenant le mot "Math".

```
SELECT titre FROM cours WHERE date_fin > '2023-01-01' AND titre LIKE '%math%';
```

## 4.2 Requête 2

Afficher les étudiants qui ont un email se terminant par "@example.com" et qui ne sont pas inscrits à un cours.

```
SELECT etudiants.nom, etudiants.prenom
FROM etudiants
LEFT JOIN inscriptions ON etudiants.id = inscriptions.id_etudiant
WHERE etudiants.email LIKE '%@example.com' AND inscriptions.id_etudiant IS NULL;
```

## 4.3 Requête 3

Afficher les cours qui commencent après le 1er janvier 2023 et qui ont une date de fin avant le 31 décembre 2023.

```
SELECT titre FROM cours WHERE date_debut > '2023-01-01' AND date_fin < '2023-12-31';
```

## 4.4 Requête 4

Afficher les étudiants dont le prénom commence par "A" et qui ont un email se terminant par "@gmail.com".

```
SELECT * FROM etudiants WHERE prenom LIKE 'A%' AND email LIKE '%@gmail.com';
```

## 4.5 Requête 5

Afficher le nombre total de cours qui sont enseignés par un professeur dont le prénom est "Ahmed" et qui commencent après le 1er janvier 2024.

```
SELECT COUNT(*)
FROM professeurs
JOIN cours ON professeurs.id = cours.id_professeur
WHERE professeurs.prenom = 'Ahmed' AND cours.date_debut > '2024-01-01';
```

## 5 Conclusion

Ce rapport présente la conception, l'insertion des données et l'exécution des requêtes pour la base de données **gestion\_cours**. Le système implémente une gestion efficace des étudiants, professeurs, cours, et inscriptions, et répond aux besoins définis par les requêtes SQL.