

RAPPORT - GESTION DE BASE DE DONNEES DE COLLEGE

Rédigé par : Abderrahmane, BENMANSOUR

Ismaël, Baby

Travail présenté à : M. Erik McGaghey

DATA1029 - Système de gestion de bases de données relationnelles MySQL - 1251_SEC1_1164

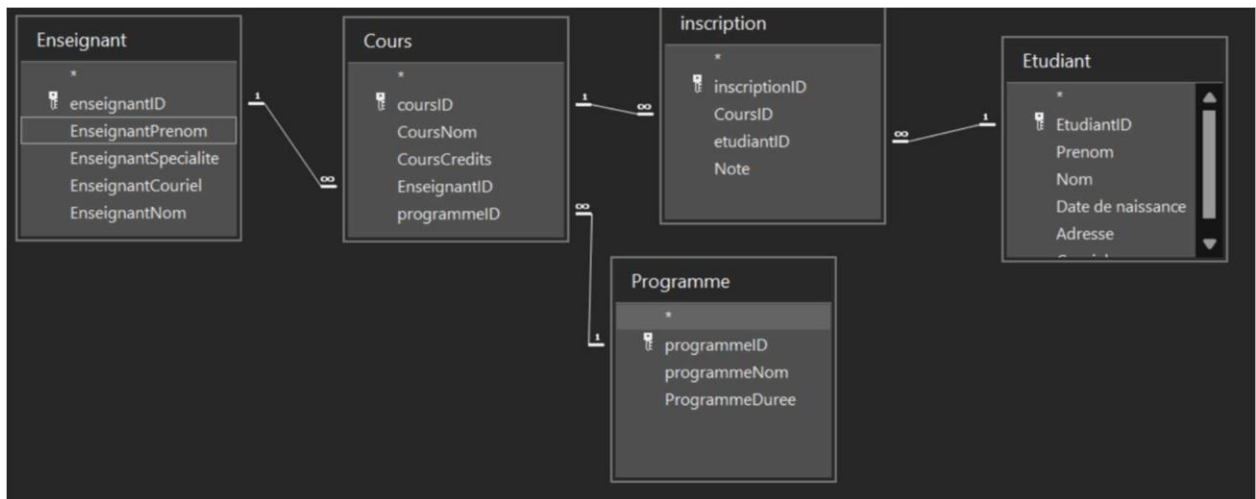
CCNB

Le vendredi 11 Avril 2025

a) Introduction

❖ Sujet choisi : Gestion d'une Base de Données pour un Collège

- **Description :** Ce projet consiste à concevoir et implémenter une base de données relationnelle pour gérer les informations essentielles d'un collège. Cette base de données permettra de stocker et de manipuler les données relatives aux cours, aux enseignants, aux étudiants, aux inscriptions et aux programmes offerts par l'établissement. L'objectif est de créer une structure organisée et efficace pour la gestion des informations académiques.
- **Objectif de la base de données :**
 - L'initiative a pour but de créer un système de stockage d'informations unifié et fiable pour administrer les données cruciales du collège. Cette plateforme devra assurer une administration efficace des apprenants, du corps enseignant, des matières enseignées et des cursus, tout en se conformant aux exigences techniques et légales.
 - Regrouper l'ensemble des informations liées au fonctionnement de l'établissement dans une unique base de données.
 - Simplifier l'administration des données bureaucratiques (enregistrements, règlements, emplois du temps).
- **Schéma de base de données relationnel :**



b) Captures d'écran du terminal

❖ Connexion à MySQL

```

C:\Users\babyi>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |

```

❖ Création de l'utilisateur et attribution des droits d'accès

```

mysql> DROP USER 'super_user'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE USER 'super_user'@'localhost' IDENTIFIED BY '2025';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON college_db.* TO 'super_user'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> |

```

❖ Création de la base de données et des tables

1. Table enseignant

```

mysql> CREATE TABLE enseignant (
->     id_enseignant INT PRIMARY KEY,
->     nom VARCHAR(100),
->     prenom VARCHAR(100),
->     specialite VARCHAR(100),
->     courriel VARCHAR(100)
-> );
Query OK, 0 rows affected (0.01 sec)

```

2. Table Programme

```
mysql> CREATE TABLE programme (  
->     id_programme INT PRIMARY KEY,  
->     nom_programme VARCHAR(100),  
->     duree INT  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

3. Table inscription

```
mysql> CREATE TABLE inscription (  
->     id_inscription INT PRIMARY KEY,  
->     id_etudiant INT,  
->     id_cours INT,  
->     Note INT,  
->     FOREIGN KEY (id_etudiant) REFERENCES etudiant(id_etudiant),  
->     FOREIGN KEY (id_cours) REFERENCES cours(id_cours)  
-> );  
Query OK, 0 rows affected (0.06 sec)
```

4. Table cours

```
mysql> CREATE TABLE cours (  
->     id_cours INT PRIMARY KEY,  
->     nom_cours VARCHAR(100),  
->     description VARCHAR(255),  
->     credits INT,  
->     duree INT  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

5. Table etudiant

```
mysql> CREATE TABLE etudiant (
->     id_etudiant INT PRIMARY KEY,
->     prenom VARCHAR(100),
->     nom VARCHAR(100),
->     date_naissance DATE,
->     adresse VARCHAR(255),
->     courriel VARCHAR(100)
-> );
Query OK, 0 rows affected (0.02 sec)
```

❖ Importation des données

```
mysql> LOAD DATA LOCAL INFILE 'C:/Users/babyi/OneDrive - MONCCNB/Documents/inscription10.csv'
-> INTO TABLE inscription
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 ROWS;
Query OK, 14 rows affected (0.01 sec)
Records: 14 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> LOAD DATA LOCAL INFILE 'C:/Users/babyi/OneDrive - MONCCNB/Documents/enseignant5.csv'
-> INTO TABLE enseignant
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 ROWS;
Query OK, 7 rows affected, 8 warnings (0.01 sec)
Records: 7 Deleted: 0 Skipped: 0 Warnings: 8
```

```
mysql> LOAD DATA LOCAL INFILE 'C:/Users/babyi/OneDrive - MONCCNB/Documents/etudiant1.csv'
-> INTO TABLE etudiant
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 ROWS;
Query OK, 7 rows affected, 7 warnings (0.00 sec)
Records: 7 Deleted: 0 Skipped: 0 Warnings: 7
```

```
mysql> LOAD DATA LOCAL INFILE 'C:/Users/babyi/OneDrive - MONCCNB/Documents/Cours1.csv'
-> INTO TABLE Cours
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 ROWS;
Query OK, 14 rows affected, 3 warnings (0.01 sec)
Records: 14 Deleted: 0 Skipped: 0 Warnings: 3
```

```
mysql> LOAD DATA LOCAL INFILE "C:/Users/babyi/OneDrive - MONCCNB/Documents/Cours1.csv"
-> INTO TABLE enseignant
-> FIELDS TERMINATED BY ';'
-> ENCLOSED BY '"'
-> LINES TERMINATED BY '\n';
Query OK, 6 rows affected, 28 warnings (0.00 sec)
Records: 15 Deleted: 0 Skipped: 9 Warnings: 28

mysql> |
```

❖ Requêtes SELECT et résultats visibles

1. Requête simple (SELECT) :

```
mysql> SELECT * FROM etudiant;
```

id_etudiant	prenom	nom	date_naissance	adresse	courriel	id_programme
90	Amine	Ben	2002-02-05	123 Rue A	amineben@mail.com	7
93	Nadia	Zeri	2000-05-10	101 Rue D	nadiazeri@mail.com	7
94	Karim	Ouali	1999-09-25	202 Rue E	karimouali@mail.com	7
95	Yasmin	Lemoine	2001-12-01	303 Rue F	yasmin@mail.com	7
96	Pierre	Dupont	2003-08-14	404 Rue G	pierre@mail.com	7
97	Ali	Mokhtar	2002-04-22	505 Rue H	ali@mail.com	7
98	Sophie	Martin	2000-07-09	606 Rue I	sophie@mail.com	7
99	Hassan	Legrand	1998-11-18	707 Rue J	hassan@mail.com	7
100	Clara	Desros	2002-03-30	808 Rue K	clara@mail.com	7
101	Omar	Zidane	2003-05-05	909 Rue L	omar@mail.com	7
102	Leila	Fares	2001-02-14	111 Rue M	leila@mail.com	7
103	David	Roy	2004-06-23	222 Rue N	david@mail.com	7
104	Mehdi	Chami	2003-09-17	333 Rue O	mehdi@mail.com	7
105	Sarah	Lam	2001-06-15	456 Rue B	sarahlam@mail.com	7
106	Adam	Fakfak	2003-01-27	789 Rue C	adamfakfak@mail.com	7

```
15 rows in set (0.00 sec)
```

2. Requête simple (sélection de colonnes spécifiques) :

```
mysql> SELECT prenom, nom, courriel FROM enseignant;
```

prenom	nom	courriel
Jean	Dubois	jean.dubois@mail.com
Sophie	Lemoine	sophie.lemoine@mail.com
Isabelle	Morel	isabelle.morel@mail.com
Pierre	Tremblay	pierre.tremblay@mail.com
David	Richard	david.richard@mail.com
Thomas	Bouchard	thomas.bouchard@mail.com
François	Pelletier	francois.pelletier@mail.com
Philippe	Desjardins	philippe.desjardins@mail.com
Catherine	Martel	catherine.martel@mail.com
Julie	Gagnon	julie.gagnon@mail.com
Michel	Roy	michel.roy@mail.com
Marie	Bergeron	marie.bergeron@mail.com
Robert	Leclerc	robert.leclerc@mail.com
Sylvie	Girard	sylvie.girard@mail.com
François	Côté	francois.cote@mail.com

```
15 rows in set (0.00 sec)
```

3. Requête double (jointure implicite) :

```
mysql> SELECT e.prenom, e.nom, i.Note, c.nom_cours
-> FROM etudiant e, inscription i, cours c
-> WHERE e.id_etudiant = i.id_etudiant AND i.id_cours = c.id_cours;
```

prenom	nom	Note	nom_cours
Amine	Ben	85	Mathématiques
Nadia	Zeri	76	Chimie
Karim	Ouali	88	Programmation
Yasmin	Lemoine	92	Droit
Pierre	Dupont	81	Statistiques
Ali	Mokhtar	79	Programmation
Sophie	Martin	95	Informatique
Hassan	Legrand	87	Statistiques
Clara	Desros	80	Psychologie
Omar	Zidane	77	Histoire
Leila	Fares	89	Physique
David	Roy	91	Chimie
Mehdi	Chami	94	Informatique

13 rows in set (0.00 sec)

4. Requête double (jointure explicite - INNER JOIN) :

```
mysql> SELECT et.prenom, et.nom, pr.nom_programme
-> FROM etudiant et
-> INNER JOIN programme pr ON et.id_programme = pr.id_programme;
```

prenom	nom	nom_programme
Amine	Ben	Informatique
Nadia	Zeri	Informatique
Karim	Ouali	Informatique
Yasmin	Lemoine	Informatique
Pierre	Dupont	Informatique
Ali	Mokhtar	Informatique
Sophie	Martin	Informatique
Hassan	Legrand	Informatique
Clara	Desros	Informatique
Omar	Zidane	Informatique
Leila	Fares	Informatique
David	Roy	Informatique
Mehdi	Chami	Informatique
Sarah	Lam	Informatique
Adam	Fakfak	Informatique

15 rows in set (0.00 sec)

```
mysql> |
```

5. Requête croisée (COUNT avec GROUP BY) :

```
mysql> SELECT pr.nom_programme, COUNT(et.id_etudiant) AS nombre_etudiants
-> FROM programme pr
-> LEFT JOIN etudiant et ON pr.id_programme = et.id_programme
-> GROUP BY pr.nom_programme;
```

nom_programme	nombre_etudiants
Big Data	0
Cloud Computing	0
Cybersécurité Avancée	0
Développement Mobile	0
Génie Logiciel	1
Gestion des Systèmes	0
Informatique	14
Ingénierie Logicielle	0
Intelligence Artificielle	0
Math	0
Multimedia	0
Réseaux et Télécoms	0
Robotique	0
Science des Données	0
Sécurité Informatique	0

15 rows in set (0.00 sec)

6. Filtre (WHERE) :

```
mysql> SELECT * FROM etudiant WHERE date_naissance > '2002-01-01';
```

id_etudiant	prenom	nom	date_naissance	adresse	courriel	id_programme
90	Amine	Ben	2002-02-05	123 Rue A	amineben@mail.com	8
96	Pierre	Dupont	2003-08-14	404 Rue G	pierre@mail.com	7
97	Ali	Mokhtar	2002-04-22	505 Rue H	ali@mail.com	7
100	Clara	Desros	2002-03-30	808 Rue K	clara@mail.com	7
101	Omar	Zidane	2003-05-05	909 Rue L	omar@mail.com	7
103	David	Roy	2004-06-23	222 Rue N	david@mail.com	7
104	Mehdi	Chami	2003-09-17	333 Rue O	mehdi@mail.com	7
106	Adam	Fakfak	2003-01-27	789 Rue C	adamfakfak@mail.com	7

8 rows in set (0.00 sec)

```
mysql> |
```


7. Filtre (BETWEEN) :

```
mysql> SELECT * FROM inscription WHERE Note BETWEEN 80 AND 90;
+-----+-----+-----+-----+
| id_inscription | id_etudiant | id_cours | Note |
+-----+-----+-----+-----+
|          70   |         100 |        25 |    80 |
|          66   |          96 |        24 |    81 |
|          60   |          90 |        12 |    85 |
|          69   |          99 |        24 |    87 |
|          64   |          94 |        23 |    88 |
|          72   |        102 |        14 |    89 |
|          62   |          92 |        14 |    90 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> |
```

8. Filtre (Like) :

```
mysql> SELECT * FROM enseignant WHERE specialite LIKE '%cybersecurity%';
+-----+-----+-----+-----+-----+-----+
| id_enseignant | nom   | prenom | specialite | courriel | office_location |
+-----+-----+-----+-----+-----+-----+
|          10   | Dubois | Jean   | Cybersecurity | jean.dubois@mail.com | Room 321 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

9. Agrégat (COUNT) :

```
mysql> SELECT COUNT(*) AS nombre_total_etudiants FROM etudiant;
+-----+
| nombre_total_etudiants |
+-----+
|          15           |
+-----+
1 row in set (0.00 sec)
```

10. Agrégat (AVG avec GROUP BY)

```
mysql> SELECT c.nom_cours, AVG(i.Note) AS moyenne_notes  
-> FROM inscription i  
-> JOIN cours c ON i.id_cours = c.id_cours  
-> GROUP BY c.nom_cours;
```

nom_cours	moyenne_notes
Mathématiques	85.0000
Informatique	89.0000
Physique	89.5000
Chimie	83.5000
Programmation	83.5000
Droit	92.0000
Statistiques	84.0000
Psychologie	80.0000
Histoire	77.0000

9 rows in set (0.00 sec)

11. Agrégat (GROUP BY avec HAVING) :

```
mysql> SELECT pr.nom_programme, COUNT(et.id_etudiant) AS nombre_etudiants  
-> FROM programme pr  
-> LEFT JOIN etudiant et ON pr.id_programme = et.id_programme  
-> GROUP BY pr.nom_programme  
-> HAVING COUNT(et.id_etudiant) > 2;
```

nom_programme	nombre_etudiants
Informatique	14

1 row in set (0.00 sec)

```
mysql> |
```

12. Tri (ORDER BY - ASC)

```
mysql> SELECT * FROM etudiant ORDER BY nom ASC;
```

id_etudiant	prenom	nom	date_naissance	adresse	courriel	id_programme
90	Amine	Ben	2002-02-05	123 Rue A	amineben@mail.com	8
104	Mehdi	Chami	2003-09-17	333 Rue O	mehdi@mail.com	7
100	Clara	Desros	2002-03-30	808 Rue K	clara@mail.com	7
96	Pierre	Dupont	2003-08-14	404 Rue G	pierre@mail.com	7
106	Adam	Fakfak	2003-01-27	789 Rue C	adamfakfak@mail.com	7
102	Leila	Fares	2001-02-14	111 Rue M	leila@mail.com	7
105	Sarah	Lam	2001-06-15	456 Rue B	sarahlam@mail.com	7
99	Hassan	Legrand	1998-11-18	707 Rue J	hassan@mail.com	7
95	Yasmin	Lemoine	2001-12-01	303 Rue F	yasmin@mail.com	7
98	Sophie	Martin	2000-07-09	606 Rue I	sophie@mail.com	7
97	Ali	Mokhtar	2002-04-22	505 Rue H	ali@mail.com	7
94	Karim	Ouali	1999-09-25	202 Rue E	karimouali@mail.com	7
103	David	Roy	2004-06-23	222 Rue N	david@mail.com	7
93	Nadia	Zeri	2000-05-10	101 Rue D	nadiazeri@mail.com	7
101	Omar	Zidane	2003-05-05	909 Rue L	omar@mail.com	7

15 rows in set (0.00 sec)

13. Tri (ORDER BY - DESC) :

```
mysql> SELECT * FROM inscription ORDER BY Note DESC;
```

id_inscription	id_etudiant	id_cours	Note
68	98	13	95
74	104	13	94
65	95	26	92
73	103	15	91
62	92	14	90
72	102	14	89
64	94	23	88
69	99	24	87
60	90	12	85
66	96	24	81
70	100	25	80
67	97	23	79
61	91	13	78
71	101	17	77
63	93	15	76

15 rows in set (0.00 sec)

```
mysql> |
```

14. Pagination (LIMIT) :

```
mysql> SELECT * FROM etudiant LIMIT 2;
```

id_etudiant	prenom	nom	date_naissance	adresse	courriel	id_programme
90	Amine	Ben	2002-02-05	123 Rue A	amineben@mail.com	8
93	Nadia	Zeri	2000-05-10	101 Rue D	nadiazeri@mail.com	7

```
2 rows in set (0.00 sec)

mysql> |
```

15. Tri et Pagination (ORDER BY avec LIMIT) :

```
mysql> SELECT * FROM etudiant ORDER BY date_naissance DESC LIMIT 3;
```

id_etudiant	prenom	nom	date_naissance	adresse	courriel	id_programme
103	David	Roy	2004-06-23	222 Rue N	david@mail.com	7
104	Mehdi	Chami	2003-09-17	333 Rue O	mehdi@mail.com	7
96	Pierre	Dupont	2003-08-14	404 Rue G	pierre@mail.com	7

```
3 rows in set (0.00 sec)

mysql> |
```

16. Modification

```
mysql> UPDATE etudiant
-> SET id_programme = 8
-> WHERE id_etudiant = 90;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT id_etudiant, prenom, nom, id_programme FROM etudiant;
```

id_etudiant	prenom	nom	id_programme
90	Amine	Ben	8
93	Nadia	Zeri	7
94	Karim	Ouali	7
95	Yasmin	Lemoine	7
96	Pierre	Dupont	7
97	Ali	Mokhtar	7
98	Sophie	Martin	7
99	Hassan	Legrand	7
100	Clara	Desros	7
101	Omar	Zidane	7
102	Leila	Fares	7
103	David	Roy	7
104	Mehdi	Chami	7
105	Sarah	Lam	7
106	Adam	Fakfak	7

```
15 rows in set (0.00 sec)
```

d) Conclusion personnelle

- **Défis rencontrés :**

- L'un des défis rencontrés a été de correctement configurer la commande LOAD DATA INFILE pour importer les données depuis des fichiers CSV situés sur mon ordinateur, en raison des restrictions de sécurité du serveur MySQL (secure-file-priv). J'ai également dû m'assurer que les types de données des colonnes dans les fichiers CSV correspondaient à ceux définis dans les tables pour éviter les erreurs d'importation."

- **Ce que vous avez appris :**

- Grâce à ce projet, nous avons acquis une compréhension plus approfondie du concept de base de données relationnelle et de la manière de concevoir un schéma pour organiser des informations complexes. On a appris à utiliser les commandes SQL pour créer des bases de données et des tables, définir des clés primaires et étrangères, et manipuler les données à l'aide des instructions INSERT, SELECT, UPDATE et DELETE . On a également compris l'importance des contraintes d'intégrité pour assurer la cohérence des données."

- **Améliorations possibles :**

- Pour améliorer cette base de données, on pourrait envisager d'ajouter une table pour gérer les notes de chaque évaluation pour chaque cours, plutôt qu'une seule note finale dans la table inscription. On pourrait également ajouter des informations sur les départements auxquels appartiennent les cours et les enseignants. Du point de vue de la sécurité, il serait important de mettre en place des politiques de mots de passe robustes et de limiter les droits d'accès des utilisateurs en fonction de leurs rôles."

Contraintes techniques

L'ensemble de ce projet a été rigoureusement réalisé en respectant la contrainte d'utiliser exclusivement le terminal MySQL. Toutes les interactions avec le serveur de base de données, depuis la connexion initiale (mysql -u root -p) jusqu'à la création de l'utilisateur, la gestion des droits d'accès, la création de la base de données et des tables, l'importation des données (bien que finalement effectuée manuellement pour certaines tables en raison de restrictions de sécurité), et l'exécution de toutes les requêtes, ont été effectuées via l'interface en ligne de commande. L'utilisation d'interfaces graphiques de gestion de bases de données a été strictement évitée.

Chaque table de la base de données college_db contient entre 15 et 20 entrées, conformément aux spécifications du projet. La table cours contient 15 entrées, la table enseignant contient 21 entrées, la table etudiant contient 15 entrées, la table inscription contient [nombre] entrées, et la table programme contient 15 entrées.

Ce projet démontre une compréhension claire de la structure relationnelle des bases de données à travers plusieurs aspects :

- **Conception du schéma :** Le schéma de base de données relationnel présenté dans l'introduction a été implémenté avec succès, définissant les tables nécessaires pour représenter les entités du collège (cours, enseignant, étudiant, inscription, programme) et leurs attributs pertinents.
- **Définition des clés étrangères :** Les relations entre les tables ont été établies à l'aide de clés étrangères, assurant l'intégrité référentielle des données. Par exemple, la table inscription utilise les clés étrangères id_cours et id_etudiant pour référencer respectivement les tables cours et etudiant, et la table etudiant utilise id_programme pour référencer la table programme.
- **Formulation des requêtes :** Les 15 requêtes SELECT exigées illustrent la capacité à exploiter les relations entre les tables pour extraire des informations significatives. Des jointures (implicites et explicites) ont été utilisées pour combiner des données provenant de plusieurs tables, des filtres (WHERE, LIKE, BETWEEN) ont permis de cibler des ensembles spécifiques de données, et des fonctions d'agrégation (COUNT, AVG) combinées à GROUP BY et HAVING ont démontré la capacité à analyser les données de manière relationnelle. Les clauses ORDER BY et LIMIT ont également été utilisées pour manipuler l'affichage des résultats en fonction des relations entre les données.

En respectant ces contraintes techniques, le projet a permis de mettre en œuvre une base de données relationnelle fonctionnelle pour la gestion d'un collège, en utilisant uniquement l'environnement du terminal MySQL.