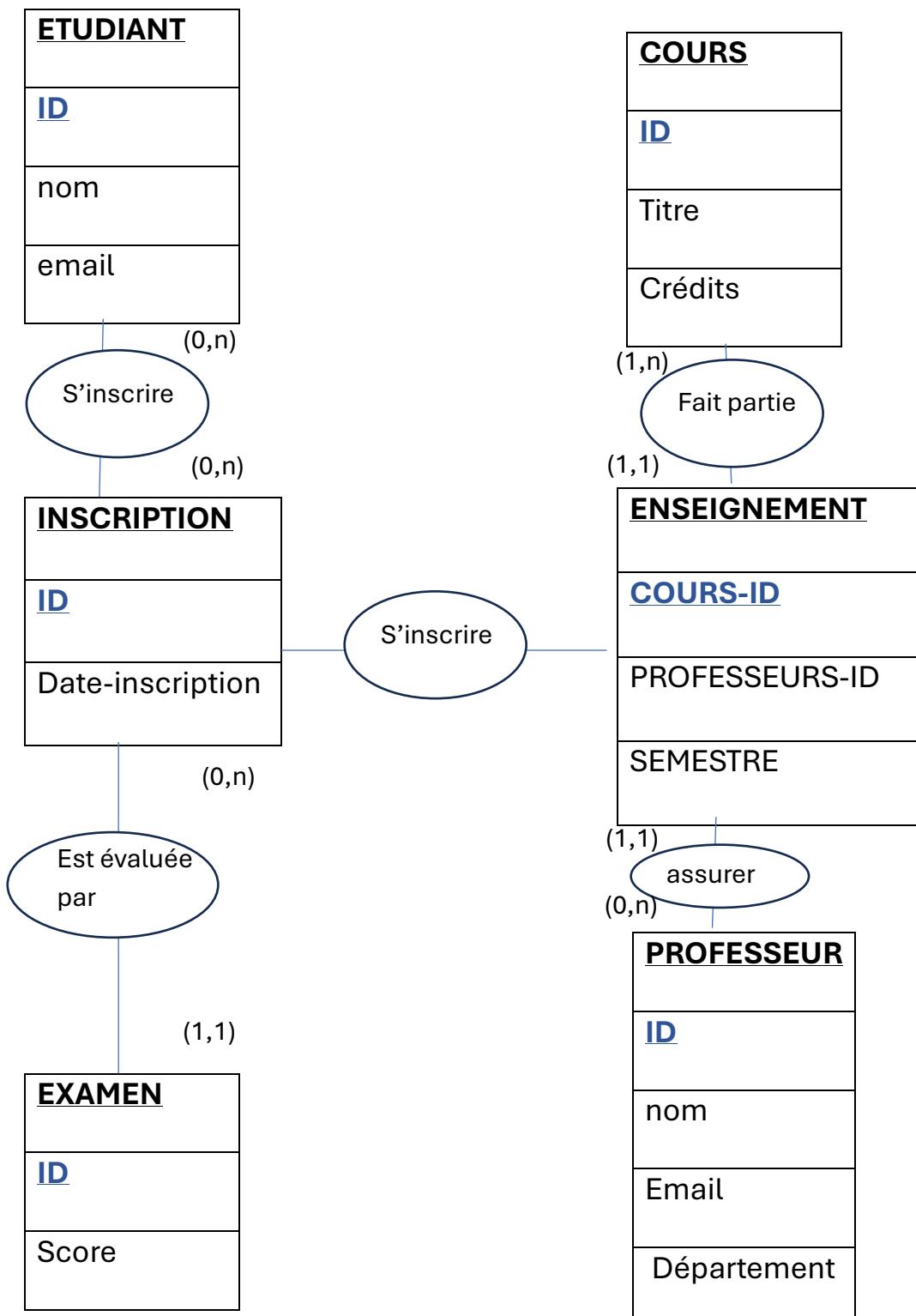


Exercice 1: University Database – Explantations

MCD



MLD TABLES:

- 1. ETUDIANT**(#id, nom, email UNIQUE)
- 2. PROFESSEUR**(#id, nom, email UNIQUE, departement)
- 3. COURS**(#id, titre, code UNIQUE, credits)
- 4. ENSEIGNEMENT**(#id, semestre, annee, #cours_id, #professeur_id)
- 5. INSCRIPTION**(#id, date_inscription, #etudiant_id, #enseignement_id)
- 6. EXAMEN**(#id, score, date_examen, #inscription_id)

CLÉS ÉTRANGÈRES:

- ENSEIGNEMENT.cours_id → COURS.id
- ENSEIGNEMENT.professeur_id → PROFESSEUR.id
- INSCRIPTION.etudiant_id → ETUDIANT.id
- INSCRIPTION.enseignement_id → ENSEIGNEMENT.id
- EXAMEN.inscription_id → INSCRIPTION.id

1. DICTIONNAIRE DE DONNÉES

Table: ETUDIANT

<u>Colonne</u>	<u>Type</u>	<u>Description</u>	<u>Contrainte</u>
<u>id</u>	INT	Identifiant unique étudiant	PRIMARY KEY, AUTO_INCREMENT
<u>nom</u>	VARCHAR(100)	Nom complet étudiant	NOT NULL
<u>email</u>	VARCHAR(100)	Email étudiant	UNIQUE, NOT NULL

Table: PROFESSEUR

<u>Colonne</u>	<u>Type</u>	<u>Description</u>	<u>Contrainte</u>
<u>id</u>	INT	Identifiant unique professeur	PRIMARY KEY, AUTO_INCREMENT
<u>nom</u>	VARCHAR(100)	Nom professeur	NOT NULL
<u>email</u>	VARCHAR(100)	Email professeur	UNIQUE, NOT NULL
<u>département</u>	VARCHAR(50)	Département d'affiliation	

Table: COURS

<u>Colonne</u>	<u>Type</u>	<u>Description</u>	<u>Contrainte</u>
<u>id</u>	INT	Identifiant unique cours	PRIMARY KEY, AUTO_INCREMENT
<u>titre</u>	VARCHAR(200)	Intitulé du cours	NOT NULL
<u>code</u>	VARCHAR(20)	Code du cours	UNIQUE, NOT NULL
<u>credits</u>	INT	Nombre de crédits ECTS	NOT NULL, CHECK (credits > 0)

Table: ENSEIGNEMENT

Colonne	Type	Description	Contrainte
<u>id</u>	INT	Identifiant unique enseignement	PRIMARY KEY, AUTO_INCREMENT
<u>semestre</u>	VARCHAR(20)	Semestre académique	NOT NULL
<u>annee</u>	INT	Année académique	NOT NULL
<u>cours_id</u>	INT	Référence au cours	FOREIGN KEY, NOT NULL
<u>professeur_id</u>	INT	Référence au professeur	FOREIGN KEY

Table: INSCRIPTION

Colonne	Type	Description	Contrainte
<u>id</u>	INT	Identifiant unique inscription	PRIMARY KEY, AUTO_INCREMENT
<u>date_inscription</u>	DATE	Date d'inscription	DEFAULT (CURRENT_DATE)
<u>etudiant_id</u>	INT	Référence à l'étudiant	FOREIGN KEY, NOT NULL
<u>enseignement_id</u>	INT	Référence à l'enseignement	FOREIGN KEY, NOT NULL

Table: EXAMEN

<u>Colonne</u>	<u>Type</u>	<u>Description</u>	<u>Contrainte</u>
<u>id</u>	INT	Identifiant unique examen	PRIMARY KEY, AUTO_INCREMENT
<u>score</u>	DECIMAL(4,2)	Note sur 20	CHECK (score BETWEEN 0 AND 20)
<u>date_examen</u>	DATE	Date de l'examen	DEFAULT (CURRENT_DATE)
<u>inscription_id</u>	INT	Référence à l'inscription	FOREIGN KEY, NOT NULL

LES FORMES NORMALES

ETUDIANT: RESPECTE 1NF, 2NF, 3NF

PROFESSEUR: RESPECTE 1NF, 2NF, 3NF

COURS: RESPECTE 1NF, 2NF, 3NF

ENSEIGNEMENT: RESPECTE 1NF, 2NF, 3NF

INSCRIPTION: RESPECTE 1NF, 2NF, 3NF

EXAMEN: RESPECTE 1NF, 2NF, 3NF

A. Création du schéma

- Base de données créée avec encodage UTF8MB4 pour supporter tous les caractères
- 6 tables créées avec ENGINE=InnoDB pour les transactions et clés étrangères
- CHECK(score BETWEEN 0 AND 20) implémenté dans la table EXAMEN

B. Contraintes d'intégrité

4. ON DELETE SET NULL dans ENSEIGNEMENT(professeur_id):

Effet: Lorsqu'un professeur est supprimé, son ID dans les enseignements devient NULL.

Avantage: Les enseignements ne sont pas supprimés, seulement le lien avec le professeur.

Risque: Peut laisser des enseignements sans professeur assigné.

5. Contrainte anti-doublon dans INSCRIPTION:

Implémentée via PRIMARY KEY(etudiant_id, enseignement_id)

Cela empêche un étudiant de s'inscrire deux fois au même enseignement.

6. Utilisation de UNIQUE():

- UNIQUE(email) dans ETUDIANT et PROFESSEUR:

Garantit l'unicité des emails (pas deux comptes avec le même email)

- UNIQUE(code) dans COURS:

Garantit que chaque cours a un code unique (ex: "CS101" ne peut exister qu'une fois)

Importance: Évite les doublons, assure l'intégrité des données, facilite les recherches.

C. Insertion et tests

2 professeurs, 3 cours, 2 étudiants, 2 enseignements, 4 inscriptions insérés

Test CHECK contrainte: Insertion avec score=25 échoue comme attendu

4 examens valides insérés avec scores entre 0 et 20

G. Maintenance du schéma

19. ALTER TABLE EXAMEN ADD COLUMN commentaire TEXT AFTER score;

- Ajoute une colonne pour les commentaires sur les examens

- Positionnée après score pour une organisation logique

20. Déploiement en production avec outils de migration:

Étape 1: Créer un fichier de migration (ex: V2__add_comment_to_examen.sql)

Étape 2: Versionner dans Git avec les autres migrations

Étape 3: Exécuter via l'outil (Flyway/Liquibase) qui:

- a. Vérifie la version actuelle
- b. Applique la migration
- c. Met à jour le registre des migrations
- d. Rollback possible en cas d'erreur

Avantages des outils de migration:

- Historique des changements
- Déploiement cohérent entre environnements
- Rollback automatisé
- Collaboration facilitée