

Université Évry Paris-Saclay

Projet : Analyse Entité-Relation pour une Plateforme Universitaire

Rapport rédigé par :

Ait Braham Rayane
Benrabah Salah

Année universitaire : 2025/2026



Table des matières

1	Introduction	2
2	Schéma relationnel	3
3	Associations du modèle	4
3.1	Association Enseigner	4
3.2	Association Programmer	4
3.3	Association Encadre	4
3.4	Association Avoir	4
3.5	Association Recevoir	4
3.6	Association Corrige	4
4	Cardinalités et justification des choix de modélisation	5
4.1	Étudiant – Inscription – Cours	5
4.2	Enseignant – Enseigner – Cours	5
4.3	Cours – Programmer – Examen	5
4.4	Enseignant – Encadre – Examen	5
4.5	Examen – Avoir – Note	5
4.6	Note – Recevoir – Étudiant	5
4.7	Enseignant – Corrige – Examen	5
5	Choix des clés primaires et étrangères	6
5.1	Clés primaires	6
5.2	Clés étrangères dans les associations	6
5.2.1	Inscription	6
5.2.2	Enseigner	6
5.2.3	Corrige	6
5.2.4	Recevoir	6
6	Explication des décisions de modélisation	7
6.1	Choix des relations et des cardinalités	7
6.2	Choix des attributs	7
6.3	Clés primaires et étrangères	7
6.4	Justification globale	7
7	Diagramme Entité–Relation	8
8	Conclusion	9

1 Introduction

Ce document présente l'analyse Entité-Relation (ER) réalisée pour la gestion d'une plateforme universitaire. Le but est de proposer un modèle conceptuel et relationnel cohérent qui couvre les besoins suivants : gestion des étudiants, des enseignants, des cours, des examens, des inscriptions et des notes.

Les entités principales retenues sont : **Étudiant**, **Cours**, **Enseignant**, **Examen**, **Note**. Les relations de type association sont : **Inscription**, **Enseigner**, **Corrige**, **Recevoir**.

2 Schéma relationnel

Entités principales :

- **Etudiant**(id_etudiant, nom, prenom, email, date_naissance)
- **Cours**(id_cours, intitule, semestre, coefficient)
- **Enseignant**(id_enseignant, nom, prenom, email, grade, specialite)
- **Examen**(id_examen, date_examen, #id_enseignant, #id_cours)
- **Note**(id_note, valeur, #id_examen)

Associations :

- **Inscription**(id_cours, id_etudiant, date_inscription, statut)
- **Enseigner**(id_enseignant, #id_cours)
- **Corrige**(id_examen, #id_enseignant)
- **Recevoir**(id_note, #id_etudiant)

3 Associations du modèle

d.

3.1 Association Enseigner

Relie un **Enseignant** et un **Cours**. Un enseignant peut enseigner plusieurs cours, un cours peut être enseigné par plusieurs enseignants.

3.2 Association Programmer

Relie un **Cours** et un **Examen**. Chaque examen est programmé pour un cours spécifique.

3.3 Association Encadre

Relie un **Enseignant** et un **Examen**. Chaque examen est encadré par un enseignant unique.

3.4 Association Avoir

Relie un **Examen** et une **Note**. Un examen génère plusieurs notes.

3.5 Association Recevoir

Relie une **Note** et un **Étudiant**. Chaque note appartient à un étudiant.

3.6 Association Corrige

Relie un **Enseignant** et un **Examen**. Chaque examen peut être corrigé par plusieurs enseignants, et un enseignant peut corriger zéro ou plusieurs examens. Cardinalités : 0,N – 1,N

4 Cardinalités et justification des choix de modélisation

4.1 Étudiant – Inscription – Cours

M :N Un étudiant peut s'inscrire à plusieurs cours, et un cours peut accueillir plusieurs étudiants.

4.2 Enseignant – Enseigner – Cours

M :N Plusieurs enseignants peuvent enseigner plusieurs cours. Un cours peut être enseigné par plusieurs enseignants.

4.3 Cours – Programmer – Examen

0,N – 1,1 Un cours peut avoir zéro ou plusieurs examens. Chaque examen appartient à un seul cours.

4.4 Enseignant – Encadre – Examen

1,N – 1,1 Un enseignant peut encadrer plusieurs examens, mais un examen possède toujours un seul enseignant responsable.

4.5 Examen – Avoir – Note

1,N – 1,1 Un examen génère plusieurs notes. Chaque note provient d'un examen unique.

4.6 Note – Recevoir – Étudiant

1,N – 1,N Un étudiant peut recevoir plusieurs notes. Une note peut être reçue par plusieurs étudiants.

4.7 Enseignant – Corrige – Examen

0,N – 1,N Un enseignant peut corriger zéro ou plusieurs examens. Un examen peut être corrigé par un ou plusieurs enseignants.

5 Choix des clés primaires et étrangères

5.1 Clés primaires

- **Etudiant** : id_etudiant
- **Cours** : id_cours
- **Enseignant** : id_enseignant
- **Examen** : id_examen
- **Note** : id_note
- **Tables d'association** : la clé primaire est composée de l'ensemble des clés étrangères

5.2 Clés étrangères dans les associations

5.2.1 Inscription

- id_etudiant (FK) → Étudiant
- id_cours (FK) → Cours

5.2.2 Enseigner

- id_enseignant (FK) → Enseignant
- id_cours (FK) → Cours

5.2.3 Corrige

- id_enseignant (FK) → Enseignant
- id_examen (FK) → Examen

5.2.4 Recevoir

- id_note (FK) → Note
- id_etudiant (FK) → Étudiant

6 Explication des décisions de modélisation

6.1 Choix des relations et des cardinalités

- Étudiant – Inscription – Cours (M :N) : un étudiant peut s'inscrire a plusieurs cours, un cours peut etre inscrist par plusieurs étudiants.
- Enseignant – Enseigner – Cours (M :N) : plusieurs enseignants peuvent dispenser le même cours.
- Cours – Programmer – Examen (0,N – 1,1) : un cours peut avoir zéro ou plusieurs examens.
- Enseignant – Encadre – Cours (0,N – 1,1) : un enseignant peut encadrer zéro ou plusieurs cours, un cours est encadré par un seul enseignant.
- Examen – Avoir – Note (1,N – 1,1) : un examen génère plusieurs notes.
- Note – Recevoir – Étudiant (1,N – 1,N) : un étudiant peut recevoir plusieurs notes, une note peut être reçue par plusieurs étudiants.
- Enseignant – Corrige – Examen (0,N – 1,N) : un enseignant peut corriger zéro ou plusieurs examens, un examen peut être corrigé par un ou plusieurs enseignants.

6.2 Choix des attributs

- Seuls les attributs essentiels sont retenus pour les entités.
- Les associations contiennent des attributs supplémentaires si nécessaire (statut, date_inscription, coefficient).

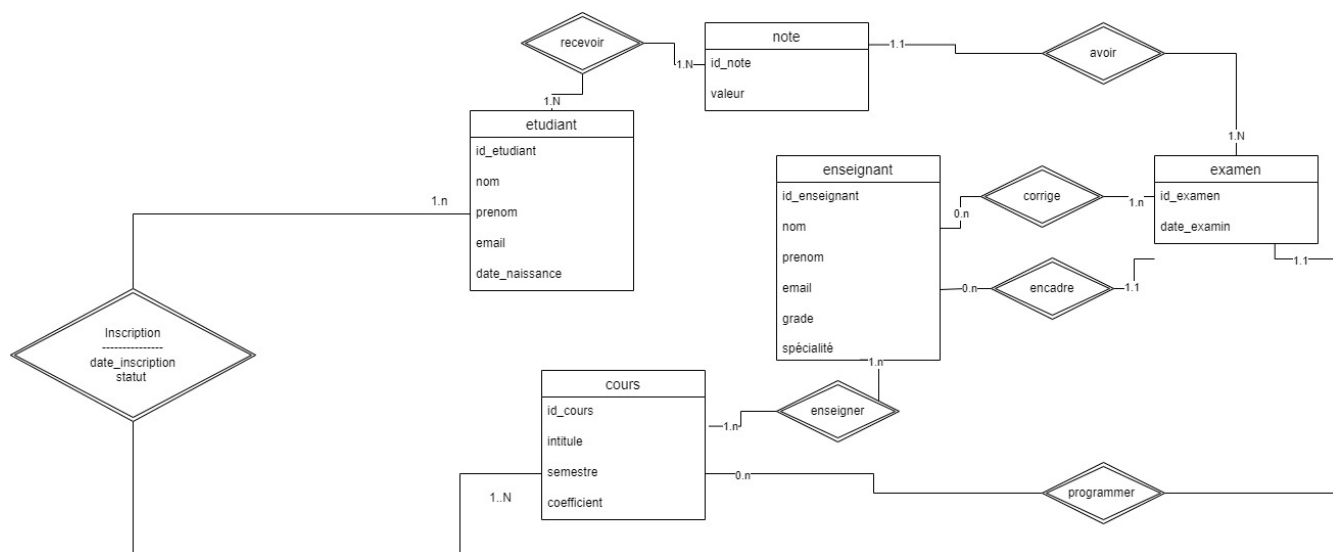
6.3 Clés primaires et étrangères

- Chaque entité a une clé primaire unique.
- Les tables d'association utilisent des clés étrangères pour garantir l'intégrité référentielle.

6.4 Justification globale

- M :N via tables d'association pour gérer les attributs supplémentaires et la normalisation.
- 1 :N et 0,N-1,1 explicites pour refléter la réalité du système.
- Séparation entités/associations pour flexibilité et clarté.

7 Diagramme Entité-Relation



8 Conclusion

Le schéma proposé couvre les besoins fonctionnels principaux d'une plateforme universitaire : gestion des étudiants, des enseignants, des cours, des examens et du suivi des notes. La séparation entités/associations permet une grande flexibilité .