

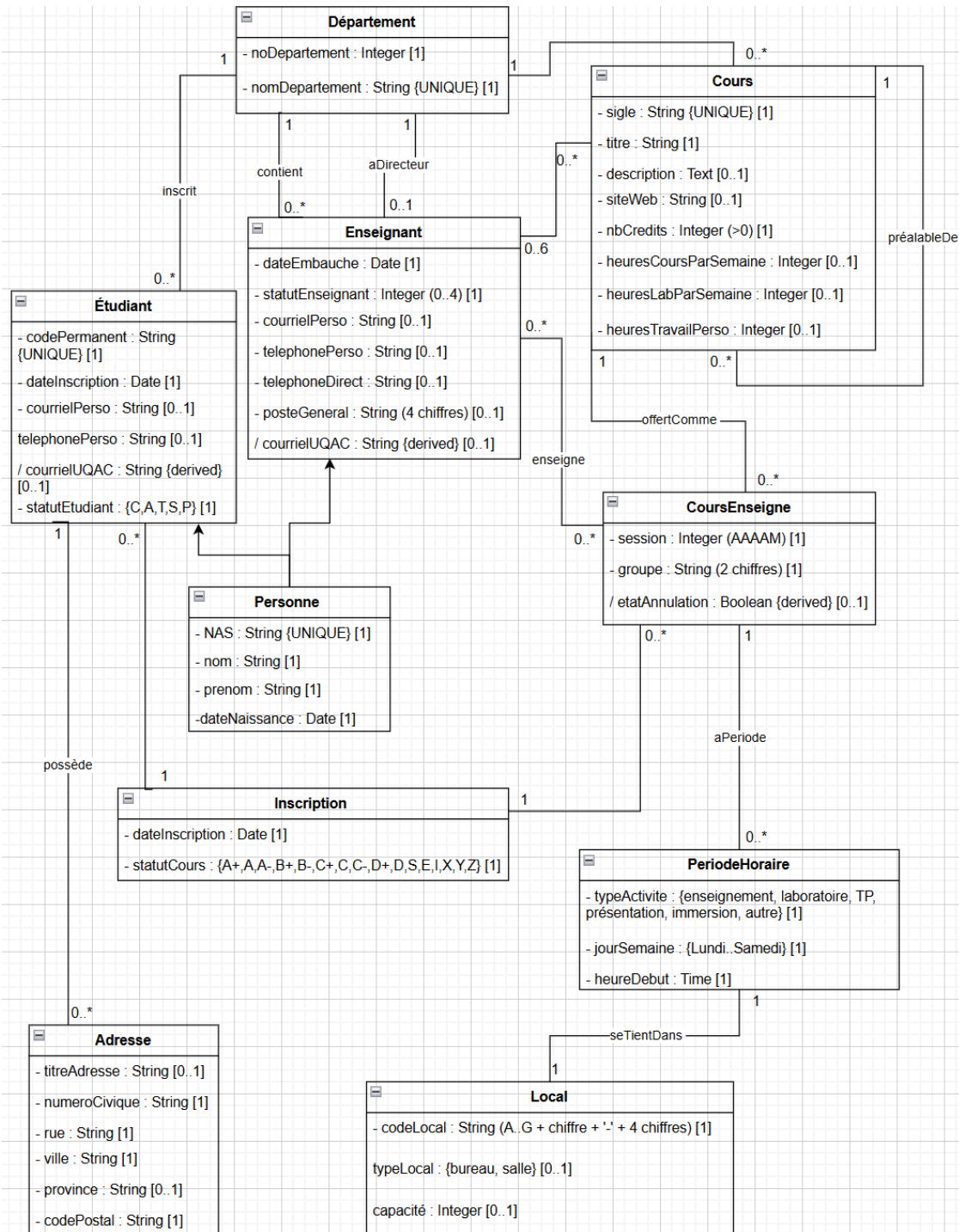
Travail pratique #1 : Modèles de conception

Étudiant 1	Hajar CHAKIR
Étudiant 2	
Date	

Rapport	
Schéma conceptuel	/ 50
Schéma relationnel	/ 30
Justifications	/ 10
Question théorique	/ 10
Qualité du français	
Total	/ 100

1 Schéma conceptuel

[Insérez le diagramme de classe du système correspondant à la tâche 1 de l'énoncé du travail. Faites attention à ce que le diagramme soit clair et qu'il y ait moins possible de recouvrements entre les lignes.]



Note 1 : Le directeur doit être un enseignant du même département.

Note 2 : Un enseignant peut être responsable de maximum 6 cours.

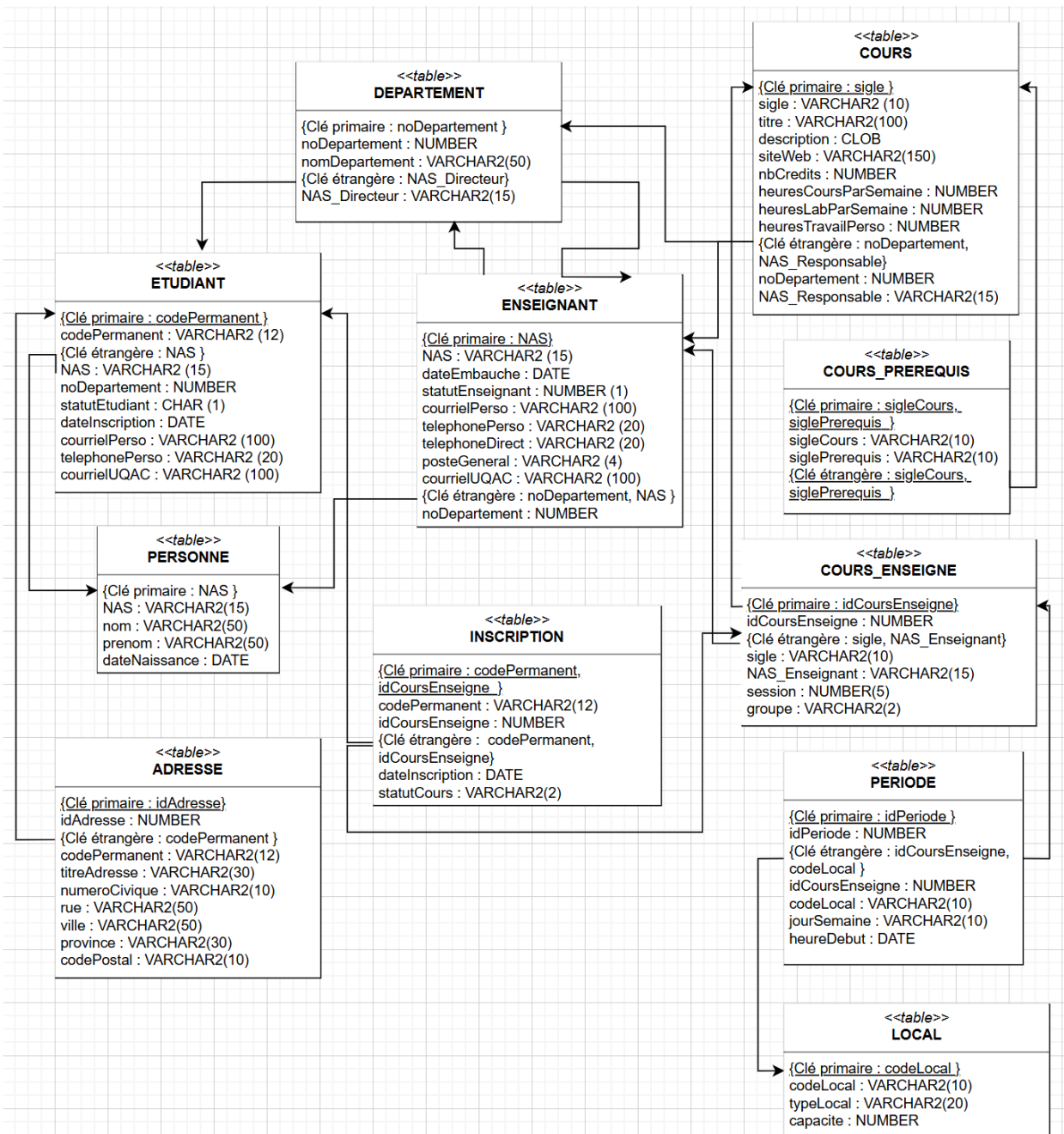
Note 3 : Un coursEnseigne est annulé si < 8 inscriptions.

Note 4 : courrielUQAC est dérivé automatiquement (selon les règles de l'UQAC).

Note 5 : Unité : NAS (Personne), codePermanent (Étudiant), sigle (Cours), nomDépartement (Département).

2 Schéma relationnel

[Insérez le diagramme UML du schéma relationnel tel que décrit à la tâche 2 de l'énoncé du travail. Faites attention à ce que le diagramme soit clair et qu'il y ait le moins possible de recouvrements entre les lignes.]



3 Justification des choix de conception

[Décrivez brièvement les stratégies employées dans la conception de votre schéma, en regard des critères de robustesse aux changements, de facilité de maintenance, de modularité et sécurité, d'efficacité, etc.]

Dans la conception de mon schéma conceptuel et relationnel, j'ai appliqué plusieurs stratégies afin d'assurer que la base de données soit robuste, maintenable, modulaire, sécurisée et efficace.

Robustesse aux changements

J'ai choisi de séparer clairement les entités principales (étudiants, enseignants, départements, cours) et de créer des tables associatives pour représenter les relations plusieurs-à-plusieurs (par exemple, l'inscription d'un étudiant à un cours enseigné, ou les cours prérequis). Cette approche me permet d'intégrer facilement de nouvelles règles de gestion sans avoir à modifier la structure complète de la base.

Facilité de maintenance

J'ai veillé à ce que chaque table ait une clé primaire bien définie (code permanent pour un étudiant, NAS pour une personne ou un enseignant, sigle pour un cours, etc.). De plus, les clés étrangères permettent de garder la cohérence entre les tables. Grâce à cette normalisation, il est plus facile de corriger ou de mettre à jour des informations dans une seule table sans introduire d'erreurs ailleurs.

Modularité et sécurité

J'ai regroupé les données par thèmes (par exemple, les adresses des étudiants dans une table séparée ADRESSE, les périodes d'un cours enseigné dans PERIODE). Cette modularité permet de limiter les accès : une application qui gère seulement les horaires pourra interroger la table PERIODE sans avoir besoin d'accéder aux informations personnelles des étudiants. Cela contribue à renforcer la sécurité et la confidentialité des données.

Efficacité

J'ai choisi des types de données adaptés (par exemple, VARCHAR2 pour les textes de taille variable, NUMBER pour les entiers, DATE pour les dates). J'ai aussi prévu des clés étrangères pour faciliter les jointures entre tables, ce qui permet des requêtes plus rapides et optimisées.

4 Conventions de nommage

[Décrivez brièvement les conventions utilisées pour nommer vos tables, colonnes, clés primaires/étrangères, procédures, etc.]

Afin de garder une cohérence et de faciliter la compréhension de mon schéma relationnel, j'ai appliqué les conventions suivantes :

- **Tables** : j'ai utilisé des noms en majuscules, au singulier, reflétant l'entité représentée.
 - Exemple : ETUDIANT, ENSEIGNANT, DEPARTEMENT, COURS.
- **Colonnes** : j'ai utilisé le camel case simplifié, en mélangeant lisibilité et précision.
 - Exemple : codePermanent, noDepartement, dateNaissance.
- **Clés primaires (PK)** : j'ai choisi des attributs naturellement uniques quand c'était possible (ex. codePermanent pour un étudiant, sigle pour un cours). Quand un attribut naturel ne suffisait pas, j'ai introduit une clé technique (ex. idCoursEnseigne, idPeriode).
- **Clés étrangères (FK)** : pour plus de clarté, j'ai repris le nom de la clé primaire référencée en ajoutant le préfixe de la table d'origine.
 - Exemple : noDepartement dans ETUDIANT fait référence à DEPARTEMENT.noDepartement.
- **Procédures/fonctions** (si elles sont ajoutées plus tard dans les TP suivants) : j'adopterai un nommage en majuscules, avec un verbe décrivant l'action.
 - Exemple : AJOUTER_ETUDIANT, MODIFIER_COURS.

Ces conventions permettent de garder une base de données homogène, lisible et facile à maintenir, tout en respectant les bonnes pratiques en entreprise.

5 Planification des tâches

[Décrivez brièvement comment le travail a été divisé dans votre équipe. Estimez, pour chaque tâche de l'énoncé, le pourcentage du travail effectué par chacun des membres de votre équipe.]

Je suis seule pour ce devoir.

6 Question théorique

[Donnez deux avantages de la normalisation des tables et dites si les tables de votre schéma sont en 3FN. Nommez les tables qui ne sont pas en 3FN.]

La normalisation des tables en base de données présente plusieurs avantages.

1. **Élimination des redondances :**

La normalisation évite de stocker plusieurs fois la même information. Par exemple, le nom d'un département n'est présent qu'une seule fois dans la table **DEPARTEMENT**, et les étudiants ou enseignants y font référence par une clé étrangère (noDepartement). Cela permet de réduire l'espace mémoire et de simplifier les mises à jour (un seul changement suffit).

2. **Amélioration de la cohérence et de l'intégrité des données :**

Grâce aux relations entre les tables (clés primaires et étrangères), on s'assure que les données restent cohérentes. Par exemple, un étudiant ne peut être lié qu'à un département existant, et une inscription à un cours doit pointer vers un cours réellement offert. Cela diminue fortement le risque d'erreurs et d'incohérences.

Vérification de la 3FN (Troisième forme normale) :

La 3FN impose trois règles principales :

1. Chaque attribut doit dépendre entièrement de la clé primaire.
2. Aucun attribut ne doit dépendre d'une partie de la clé (dans le cas de clés composées).
3. Aucun attribut ne doit dépendre transitivement d'un autre attribut non clé.

En analysant mon schéma relationnel :

- Les tables **ETUDIANT**, **ENSEIGNANT**, **DEPARTEMENT**, **COURS**, **COURS_ENSEIGNE**, **PERIODE**, **INSCRIPTION** et **ADRESSE** respectent bien la 3FN. Chaque attribut dépend uniquement de la clé primaire de sa table.
- Les tables associatives comme **COURS_PREREQUIS** sont également en 3FN, car elles ne contiennent que des clés étrangères formant leur clé primaire.

Exceptions :

Certaines informations pourraient être discutables :

- Dans **ENSEIGNANT** et **ETUDIANT**, le courriel institutionnel peut être considéré comme dérivé (il dépend du prénom, du nom et d'un compte). Si on l'inclut dans la table telle quelle, on pourrait dire qu'il introduit une dépendance fonctionnelle non nécessaire.
- Toutefois, pour des raisons pratiques et de performance, j'ai choisi de le conserver dans les tables afin d'éviter de le recalculer dynamiquement.

Pour conclure, je considère donc que mon schéma est globalement en **3FN**, à l'exception des colonnes calculées comme le **courriel institutionnel**. Cette décision a été faite volontairement afin d'allier respect des principes de normalisation et efficacité dans l'utilisation réelle de la base de données.