



Université de Nouakchott Al Aasriya
Faculté des Sciences et Techniques
Département Mathématiques et Informatiques



Mémoire de Projet BDA



Normalisation d'une Base de Données d'Inscriptions des Étudiants

Réalisé par :

Mohamed Lemine Abdallahi Tah

C12896

Encadré par :

Dr Cheikhne Mohamed Mahmoud

Année universitaire : 2024 – 2025

Projet réalisé dans le cadre du Master SSD – Statistiques et Sciences des Données

Table des matières

Introduction générale à la normalisation	2
1 Première Forme Normale (1NF)	2
1.1 Partie théorique	2
1.2 Partie pratique	2
2 Deuxième Forme Normale (2NF)	3
2.1 Partie théorique	3
2.2 Partie pratique	3
3 Troisième Forme Normale (3NF)	4
3.1 Partie théorique	4
3.2 Partie pratique	4
4 Exemples d'enregistrements dans la base de données	6
Conclusion	7

Note*

Ce projet porte sur le thème : « *Normalisation d'une base de données relationnelle : application des formes normales 1NF, 2NF et 3NF* ».

Les objectifs sont les suivants :

- Identifier les anomalies dans une base non normalisée.
- Appliquer les règles des trois premières formes normales.
- Structurer une base cohérente en 3NF avec intégrité référentielle.

Note*

Une copie du thème et des objectifs du projet est disponible dans le fichier `rapport.pdf`, situé dans le dépôt GitHub suivant :

<https://github.com/MoLemineAb/normalisation>

Le script SQL se trouve dans le fichier `normalisation_bdd.sql`

Introduction générale à la normalisation

La normalisation est un processus fondamental dans la conception des bases de données relationnelles. Elle vise à :

- Réduire la redondance des données,
- Éviter les anomalies d'insertion, de mise à jour ou de suppression,
- Garantir la cohérence et l'intégrité des informations stockées.

Elle se fait selon plusieurs formes dites "normales", désignées par 1NF, 2NF, 3NF, etc. Dans ce projet, nous nous concentrons sur les trois premières formes normales.

Le script SQL complet est disponible sur le dépôt GitHub suivant :

https://github.com/MoLemineAb/normalisation/blob/main/normalisation_bdd.sql

1 Première Forme Normale (1NF)

1.1 Partie théorique

Une table est en 1NF si :

- Toutes les colonnes contiennent des valeurs atomiques (une seule par cellule),
- Il n'y a pas de colonnes avec des listes ou groupes répétitifs,
- Une clé primaire est définie pour identifier de façon unique chaque ligne.

1.2 Partie pratique

Dans la base initiale, certains champs violaient la 1NF :

- `matière` : CalculDiff., Géométrie différentielle, Analyse
- `note_tp` : 0, 0, 0
- `note_cf` : 13,12,14

Problème : Ces champs contiennent plusieurs valeurs pour un même enregistrement.

Extrait de la base initiale :

id_etudiant	matière	note_tp	note_cc	note_cf
C00001	CalculDiff, Géométrie différentielle, Analyse	0,0,0	11,15,16	13,10,14

TABLE 1 – Extrait de la base non normalisée (violant la 1NF)

Transformation appliquée : chaque ligne est éclatée de façon à ne contenir qu'une seule matière par étudiant :

id_etudiant	matière	note_tp	note_cf
C00001	CalculDiff.	0	13
C00001	Géométrie différentielle	0	10
C00001	Analyse	0	14

TABLE 2 – Table normalisée en 1NF

La table respecte maintenant la 1NF.

2 Deuxième Forme Normale (2NF)

2.1 Partie théorique

Une table est en 2NF si :

- Elle est déjà en 1NF,
- Tous les attributs non-clés dépendent de la totalité de la clé primaire.

Cette étape élimine les dépendances partielles : un attribut ne doit pas dépendre d'une partie seulement d'une clé composée.

2.2 Partie pratique

Dans la base initiale, certaines informations concernant l'étudiant (`nom`, `prénom`, `date_naissance`, etc.) se répètent à chaque matière suivie. Cela indique une dépendance partielle de ces attributs à `id_etudiant` uniquement, et non à la clé composée (`id_etudiant`, `matière`).

Exemple de redondance :

id_etudiant	nom	matière	note_tp	note_cf
C00001	Mohamed Ahmed	Analyse	0	14
C00001	Mohamed Ahmed	Géométrie différentielle	0	10

TABLE 3 – Violation de la 2NF : redondance des informations personnelles

Correction appliquée :

décomposition en plusieurs tables :

- **Etudiants** : `id_etudiant`, `nom`, `prénom`, `date_naissance`, etc.
- **Notes** : `id_etudiant`, `matière`, `note_tp`, `note_cf`, etc.

Chaque attribut dépend maintenant uniquement de la clé de sa propre table.

Remarque sur les filières :

Initialement, chaque étudiant possédait un champ `code_filiere`. Cependant, un étudiant peut suivre des matières provenant de plusieurs niveaux (ex. valider une matière de L2 tout en étant en L3).

Solution correcte : créer une table d'association **Inscriptions** permettant de relier chaque étudiant à une ou plusieurs filières :

```
CREATE TABLE Inscriptions (
    id_etudiant VARCHAR(10),
    code_filiere VARCHAR(20),
    PRIMARY KEY (id_etudiant, code_filiere),
    FOREIGN KEY (id_etudiant) REFERENCES Etudiants(id_etudiant),
    FOREIGN KEY (code_filiere) REFERENCES Filières(code_filiere)
);
```

3 Troisième Forme Normale (3NF)

3.1 Partie théorique

Une table est en 3NF si :

- Elle est déjà en 2NF,
- Aucun attribut non-clé ne dépend d'un autre attribut non-clé.

Cette forme élimine les dépendances transitives.

3.2 Partie pratique

Problèmes identifiés :

- `code_filiere` → `filiere`, `niveau`
- `matière` → `semestre`, `crédit`

Extrait :

<u>code_filiere</u>	filière	niveau	matière	semestre	crédit
C16L3	MA	L3	CalculDiff	S5	6

TABLE 4 – Dépendances transitives entre champs non-clés

Solution :

- Créer une table **Filières** avec `code`, `nom` et `niveau`,
- Créer une table **Matières** avec `nom`, `semestre`, `crédit`,
- Supprimer ces champs des autres tables et utiliser des clés étrangères.

Les dépendances transitives ont été supprimées, assurant la conformité avec la 3NF.

Remarque sur les champs dérivés : mfe et decision

Dans la base initiale, deux champs posaient problème du point de vue de la 3NF :

- **mfe** : moyenne finale, souvent calculée à partir des notes,
- **decision** : validation ou non selon la moyenne.

Règle métier observée :

La moyenne est calculée selon la formule pondérée suivante :

$$\text{mfe} = \frac{(\text{note_tp} \times 1) + (\text{note_cc} \times 2) + (\text{note_cf} \times 3)}{6}$$

La **decision** dépend de cette moyenne :

- Si $\text{mfe} \geq 10$ alors "V" (Validé),
- Sinon, on essaie avec la meilleure note entre **note_cf** et **note_sc**.

Exemple de calcul SQL (non stocké, mais utilisable dans une vue ou requête) :

```
SELECT id_etudiant, nom_matiere,
ROUND(
CASE
WHEN ((note_tp*1 + note_cc*2 + note_cf*3)/6) >= 10
THEN ((note_tp*1 + note_cc*2 + note_cf*3)/6)
ELSE ((note_tp*1 + note_cc*2 + GREATEST(note_cf, note_sc)*3)/6)
END, 2
) AS mfe,
CASE
WHEN ((note_tp*1 + note_cc*2 + note_cf*3)/6) >= 10
THEN 'V'
WHEN ((note_tp*1 + note_cc*2 + GREATEST(note_cf, note_sc)*3)/6) >= 10
THEN 'V'
ELSE 'NV'
END AS decision
FROM Notes;
```

Note* : Pour respecter strictement la 3NF, **mfe** et **decision** ne sont pas stockés dans la base. Ils sont calculés dynamiquement à partir des autres champs (atomiques).

4 Exemples d'enregistrements dans la base de données

Voici un aperçu de deux lignes d'exemples pour chaque table, permettant de mieux visualiser le contenu de la base après normalisation.

1. Table Etudiants

id	Nom	Prénom	Date	Lieu	Adresse	Téléphone	Genre
C00001	Mohamed	Ahmed	2000-12-31	Néma	NKTT	12121212	M
C00002	Abou	Mohamed	2000-12-31	Aioun	NKTT	12121213	M

TABLE 5 – Extrait de la table Etudiants

2. Table Filières

Code Filière	Filière	Niveau
C13L2	MAI	L2
C16L3	MA	L3

TABLE 6 – Extrait de la table Filières

3. Table Inscriptions

id_etudiant	code_filiere
C00001	C16L3
C00001	C13L2

TABLE 7 – Extrait de la table Inscriptions

4. Table Matieres

Matière	Semestre	Crédit
CalculDiff	5	6
Géometrie différentielle	6	3

TABLE 8 – Extrait de la table Matieres

5. Table Notes

id_etudiant	Matière	TP	CC	CF	SC	Année
C00001	CalculDiff	0.0	11.0	13.0	0.0	2024-2025
C00001	Géometrie différentielle	0.0	15.0	10.0	0.0	2024-2025

TABLE 9 – Extrait de la table Notes

Conclusion

Ce travail a permis de transformer une base brute non normalisée en un modèle relationnel structuré et cohérent. Chaque étape de la normalisation (1NF, 2NF, 3NF) a été appliquée méthodiquement pour garantir l'intégrité, la non-redondance et la maintenabilité des données.