

# TP — Normalisation SQL & Modélisation MongoDB

Atelier pratique : Du fichier Excel au schéma de base de données

**Joseph AZAR** — IUT Nord Franche-Comté — [joseph.azar@univ-fcomte.fr](mailto:joseph.azar@univ-fcomte.fr)

# Contexte et Mise en Situation

---

## La société LuminaTech

**LuminaTech** est une agence web de 12 employés basée à Besançon. Elle réalise des projets numériques (sites web, applications mobiles, identité graphique) pour des clients variés : boulangeries, cabinets juridiques, hôtels, fromageries...

Jusqu'à présent, le directeur gère tout dans un **fichier Excel unique** avec 6 onglets. Mais avec la croissance de l'entreprise, il vous demande de créer une **vraie base de données**.

**Fichier fourni :** `TP_LuminaTech_Données.xlsx` — 6 feuilles contenant les données réelles de l'entreprise :

- **Employés** (12 lignes) — identité, poste, département, salaire, responsable hiérarchique
- **Clients** (8 lignes) — coordonnées, secteur d'activité
- **Projets** (7 lignes) — description, client, chef de projet, budget, statut
- **Affectations** (26 lignes) — quel employé travaille sur quel projet, heures, taux
- **Factures** (12 lignes) — facturation par projet, TVA, paiements
- **Technologies** (25 lignes) — stack technique utilisée par projet

**Votre mission :** Analyser ce fichier Excel, identifier les problèmes, puis concevoir deux bases de données : une en **SQL (3FN)** et une en **MongoDB (NoSQL)**. Vous devrez justifier chaque choix.

# 1 Analyse du fichier Excel

## Exercice 1.1 — Identifier les redondances

1. Ouvrez le fichier `TP_LuminaTech_Données.xlsx`
2. Pour chaque feuille, identifiez les **données redondantes** (informations répétées inutilement).
3. Listez au moins **5 exemples concrets** de redondance avec le nom de la feuille, les colonnes concernées, et le nombre de répétitions.

**Indice :** Regardez attentivement la feuille « Affectations » — des noms et prénoms sont-ils répétés ? Et dans « Projets » — le nom du client apparaît combien de fois ?

## Exercice 1.2 — Identifier les anomalies

Pour chaque scénario, expliquez le problème qui se poserait avec le fichier Excel actuel :

1. **Modification :** L'employée Marie Dupont (EMP001) change d'email. Combien de cellules faut-il modifier ? Dans quelles feuilles ?
2. **Insertion :** Un nouveau client « Librairie du Centre » se présente, mais n'a pas encore de projet. Peut-on le stocker quelque part sans problème ?
3. **Suppression :** Le projet PRJ003 (Refonte Logo EcoVert) est archivé et supprimé de la feuille « Projets ». Quelles informations perd-on ?
4. **Incohérence :** Que se passe-t-il si quelqu'un modifie le nom du projet dans la feuille « Factures » mais pas dans « Affectations » ?

## Exercice 1.3 — Identifier les dépendances fonctionnelles

Pour chaque feuille, listez toutes les dépendances fonctionnelles. Utilisez la notation  $A \rightarrow B$ .

**Exemple :** Dans la feuille Employés : `ID_Employé`  $\rightarrow$  `Nom`, `Prénom`, `Email`, `Téléphone`, `Poste`, `Département`, `Date_Embauche`, `Salaire`, `Responsable`

1. Listez les DF de la feuille **Clients**
2. Listez les DF de la feuille **Projets** — Attention, certaines colonnes créent des dépendances partielles ou transitives !

3. Listez les DF de la feuille **Affectations** — Quelle est la clé primaire ?
4. Listez les DF de la feuille **Factures**
5. Listez les DF de la feuille **Technologies**

### Exercice 2.1 — Normalisation pas à pas

En partant des feuilles Excel, normalisez les données en appliquant les étapes suivantes :

1. **1FN** : Les données sont-elles déjà atomiques ? Y a-t-il des champs à éclater ?
2. **2FN** : Identifiez les dépendances partielles dans les feuilles ayant une clé composite (Affectations, Technologies). Proposez une décomposition.
3. **3FN** : Identifiez les dépendances transitives. Par exemple :
  - Dans « Projets » : `ID_Projet → ID_Client → Raison_Sociale`
  - Dans « Factures » : `N°_Facture → ID_Projet → Nom_Projet` et `N°_Facture → ID_Client → Raison_Sociale`
  - Dans « Affectations » : `(ID_Projet, ID_Employé) → ID_Projet → Nom_Projet`

Proposez la décomposition pour chaque cas.

### Exercice 2.2 — Schéma final en 3FN

1. Dessinez le **schéma relationnel** final (tables, colonnes, clés primaires, clés étrangères).
2. Pour chaque table, précisez :
  - Le nom de la table
  - La liste des colonnes avec leurs types de données
  - La clé primaire (soulignée)
  - Les clés étrangères (avec la table référencée)
3. Combien de tables avez-vous au final ? Justifiez chaque table.

### Exercice 2.3 — Script SQL de création

Écrivez le script SQL `CREATE TABLE` complet pour votre schéma. Incluez :

- Les `PRIMARY KEY`
- Les `FOREIGN KEY ... REFERENCES ...`
- Les contraintes `NOT NULL` où nécessaire

- Les types de données appropriés ( `VARCHAR` , `INT` , `DECIMAL` , `DATE` , `ENUM` ...)

### Exercice 2.4 — Vérification de la 3FN

Pour **chaque table** de votre schéma, démontrez qu'elle respecte la 3FN en répondant :

1. **1FN** : Toutes les valeurs sont-elles atomiques ?
2. **2FN** : Y a-t-il des dépendances partielles ? (Si clé simple → automatiquement en 2FN)
3. **3FN** : Y a-t-il des dépendances transitives entre attributs non-clé ?

**Rappel :** En MongoDB, on modélise en fonction des **requêtes fréquentes**, pas en fonction de la structure des données. La question n'est pas « quelles sont mes entités ? » mais « quelles questions je pose à ma base ? »

### Exercice 3.1 — Analyse des patterns d'accès

Voici les requêtes les plus fréquentes de LuminaTech. Pour chacune, indiquez si elle favorise l'**embarquement** (embed) ou le **référencement** (normalize) :

1. "Afficher un projet avec tous ses membres, heures et technologies"
2. "Lister tous les projets d'un client donné avec le montant total facturé"
3. "Mettre à jour le téléphone d'un employé"
4. "Afficher la fiche complète d'un employé avec tous ses projets en cours"
5. "Générer la facture d'un projet avec les détails client"
6. "Voir quels employés connaissent une technologie donnée"

### Exercice 3.2 — Trois modélisations MongoDB

Proposez **trois modélisations différentes** en écrivant un document JSON d'exemple pour chaque approche :

#### Approche A — Normalisée (une collection par feuille Excel)

Créez une collection par entité avec des références `_id`.

- Combien de collections ?
- Montrez un document exemple de la collection `projets`
- Comment répondriez-vous à la requête Q1 ? (Combien de requêtes nécessaires ?)

#### Approche B — Tout embarqué

Créez une seule collection `projets` où TOUT est embarqué (client, employés, factures, technologies).

- Montrez un document complet pour le projet PRJ001
- Quelle est la taille estimée de ce document ?

- Identifiez au moins 3 problèmes concrets de cette approche

## Approche C — Hybride (recommandée)

Concevez un modèle qui embarque intelligemment et référence quand nécessaire :

- Quelles données embarquer dans les projets ?
- Quelles entités garder en collections séparées ?
- Y a-t-il des données à dupliquer volontairement (dénormalisation) ?
- Montrez un document exemple pour chaque collection

### Exercice 3.3 — Comparaison et justification

Remplissez le tableau comparatif suivant :

Critère	Approche A (Norm.)	Approche B (Embed)	Approche C (Hybride)
Nombre de collections			
Redondance de données			
Requêtes pour Q1			
Facilité de mise à jour			
Risque d'incohérence			
Taille max document			

Ensuite, rédigez un paragraphe justifiant **quelle approche vous choisiriez** pour LuminaTech et pourquoi.

### Exercice 3.4 — Requêtes MongoDB

En utilisant votre modèle hybride (Approche C), écrivez les requêtes MongoDB pour :

1. Trouver tous les projets « En cours »
2. Afficher le projet PRJ002 avec tous ses détails
3. Lister tous les projets où travaille l'employé EMP001
4. Calculer le total des heures réalisées par employé sur tous les projets



5. Trouver tous les projets utilisant la technologie « React »
6. Mettre à jour le téléphone de l'employé EMP004

## 4 Synthèse SQL vs MongoDB – Analyse critique

### Exercice 4.1 — Comparaison des approches

Rédigez une analyse comparative (15-20 lignes minimum) répondant aux questions suivantes :

1. Quels sont les avantages concrets du schéma SQL normalisé pour LuminaTech ?
2. Dans quels scénarios le modèle MongoDB serait-il préférable ?
3. Si LuminaTech passe de 12 à 200 employés et gère 500 projets simultanés, quel impact sur chaque modèle ?
4. Si l'entreprise développe un tableau de bord temps réel pour suivre les heures, quel modèle est le plus adapté ?

### Exercice 4.2 — Recommandation finale

Le directeur de LuminaTech vous demande : « *On prend SQL ou MongoDB ?* »

Rédigez une recommandation argumentée (10-15 lignes) en prenant en compte :

- La nature des données (structurée ? hiérarchique ?)
- Les besoins de cohérence (factures, salaires → critique !)
- La fréquence de lecture vs écriture
- L'évolution future de l'entreprise
- Les compétences de l'équipe technique

**Attention :** Il n'y a pas de « bonne réponse ». Les deux choix sont défendables ! Ce qui compte, c'est la **qualité de votre argumentation**.

# Livrables attendus

---

## À rendre en fin de séance :

1. **Rapport d'analyse** (exercices 1.1 à 1.3) — anomalies, redondances, DF
2. **Schéma relationnel SQL en 3FN** + script `CREATE TABLE` (exercices 2.1 à 2.4)
3. **3 modèles MongoDB** (documents JSON) + tableau comparatif (exercices 3.1 à 3.4)
4. **Synthèse comparative** + recommandation (exercices 4.1 et 4.2)

## Conseils :

- Commencez par lire tout le fichier Excel attentivement
- Dessinez les schémas sur papier avant de coder
- N'oubliez pas les clés étrangères !
- Justifiez TOUS vos choix