

Dépendances Fonctionnelles & Normalisation

De l'Excel au Schéma Relationnel Propre

Joseph AZAR

IUT Nord Franche-Comté
joseph.azar@univ-fcomte.fr

Appuyez sur **0** ou **Échap** pour la vue d'ensemble

Plan du cours

Partie 1 — Le Problème

1. Le scénario : un client avec son fichier Excel
2. Les anomalies (insertion, modification, suppression)

Partie 2 — L'outil

3. Les dépendances fonctionnelles
4. Clés candidates et clés primaires

Partie 3 — La Solution

5. 1^{re} Forme Normale (1FN)
6. 2^e Forme Normale (2FN)
7. 3^e Forme Normale (3FN)
8. Forme Normale de Boyce-Codd (BCNF)

Partie 4 — Pratique 9. Exercices guidés

Partie 1

Le Problème

Quand Excel ne suffit plus...

Le scénario : Mme Dupont et sa boutique

La cliente

Mme Dupont gère une boutique en ligne. Elle stocke TOUT dans un seul fichier Excel : clients, commandes, produits, fournisseurs...

Le scénario : Mme Dupont et sa boutique

La cliente

Mme Dupont gère une boutique en ligne. Elle stocke TOUT dans un seul fichier Excel : clients, commandes, produits, fournisseurs...

Voici son fichier :

NºCmd	DateCmd	Client	Ville	Produit	PrixUnit	Qté	Fournisseur	TelFourn
C001	15/01	Ali	Paris	Clavier	25€	2	TechPlus	01.23.45
C001	15/01	Ali	Paris	Souris	15€	1	TechPlus	01.23.45
C002	16/01	Sara	Lyon	Écran	200€	1	ScreenCo	04.56.78
C003	17/01	Ali	Paris	Clavier	25€	3	TechPlus	01.23.45

Le scénario : Mme Dupont et sa boutique

La cliente

Mme Dupont gère une boutique en ligne. Elle stocke TOUT dans un seul fichier Excel : clients, commandes, produits, fournisseurs...

Voici son fichier :

NºCmd	DateCmd	Client	Ville	Produit	PrixUnit	Qté	Fournisseur	TelFourn
C001	15/01	Ali	Paris	Clavier	25€	2	TechPlus	01.23.45
C001	15/01	Ali	Paris	Souris	15€	1	TechPlus	01.23.45
C002	16/01	Sara	Lyon	Écran	200€	1	ScreenCo	04.56.78
C003	17/01	Ali	Paris	Clavier	25€	3	TechPlus	01.23.45

Voyez-vous les problèmes ? Regardons de plus près...

Problème n°1 : La Redondance

NºCmd	DateCmd	Client	Ville	Produit	PrixUnit	Qté	Fournisseur	TelFourn
C001	15/01	Ali	Paris	Clavier	25€	2	TechPlus	01.23.45
C001	15/01	Ali	Paris	Souris	15€	1	TechPlus	01.23.45
C002	16/01	Sara	Lyon	Écran	200€	1	ScreenCo	04.56.78
C003	17/01	Ali	Paris	Clavier	25€	3	TechPlus	01.23.45

Problème n°1 : La Redondance

NºCmd	DateCmd	Client	Ville	Produit	PrixUnit	Qté	Fournisseur	TelFourn
C001	15/01	Ali	Paris	Clavier	25€	2	TechPlus	01.23.45
C001	15/01	Ali	Paris	Souris	15€	1	TechPlus	01.23.45
C002	16/01	Sara	Lyon	Écran	200€	1	ScreenCo	04.56.78
C003	17/01	Ali	Paris	Clavier	25€	3	TechPlus	01.23.45

Les données répétées :

- Ali / Paris apparaît 3 fois
- Prix du Clavier (25€) apparaît 2 fois
- TechPlus / 01.23.45 apparaît 3 fois

- Gaspillage d'espace disque
- Risque d'incohérence

Conséquences :

- Maintenance difficile

Problème n°2 : Anomalie de Modification



Scénario : TechPlus change de numéro de téléphone

Nouveau numéro : **09.87.65**

Problème n°2 : Anomalie de Modification



Scénario : TechPlus change de numéro de téléphone

Nouveau numéro : **09.87.65**

Il faut modifier TOUTES les lignes où TechPlus apparaît !

N°Cmd	Produit	Fournisseur	TelFourn
C001	Clavier	TechPlus	01.23.45 → 09.87.65
C001	Souris	TechPlus	01.23.45 → 09.87.65
C003	Clavier	TechPlus	01.23.45 → 09.87.65

Problème n°2 : Anomalie de Modification



Scénario : TechPlus change de numéro de téléphone

Nouveau numéro : **09.87.65**

Il faut modifier TOUTES les lignes où TechPlus apparaît !

N°Cmd	Produit	Fournisseur	TelFourn
C001	Clavier	TechPlus	01.23.45 → 09.87.65
C001	Souris	TechPlus	01.23.45 → 09.87.65
C003	Clavier	TechPlus	01.23.45 → 09.87.65

Et si on oublie UNE ligne ? On a deux numéros différents pour TechPlus = données incohérentes !

Problème n°3 : Anomalie d'Insertion



Scénario : Un nouveau fournisseur "AudioMax" se présente

On veut stocker son nom et son téléphone, mais il n'a pas encore de commande...

Problème n°3 : Anomalie d'Insertion



Scénario : Un nouveau fournisseur "AudioMax" se présente

On veut stocker son nom et son téléphone, mais il n'a pas encore de commande...

NºCmd	DateCmd	Client	Produit	Fournisseur	TelFourn
???	???	???	???	AudioMax	06.11.22

Problème n°3 : Anomalie d'Insertion



Scénario : Un nouveau fournisseur "AudioMax" se présente

On veut stocker son nom et son téléphone, mais il n'a pas encore de commande...

N°Cmd	DateCmd	Client	Produit	Fournisseur	TelFourn
???	???	???	???	AudioMax	06.11.22

Impossible ! On ne peut pas ajouter un fournisseur sans créer une fausse commande. Les colonnes obligatoires (N°Cmd, Client...) seraient vides = violation de clé primaire !

Problème n°4 : Anomalie de Suppression



Scénario : Sara annule sa commande C002

On supprime la ligne de la commande C002...

Problème n°4 : Anomalie de Suppression



Scénario : Sara annule sa commande C002

On supprime la ligne de la commande C002...

NºCmd	Client	Produit	Fournisseur	TelFourn
C002	Sara	Écran	ScreenCo	04.56.78

Problème n°4 : Anomalie de Suppression



Scénario : Sara annule sa commande C002

On supprime la ligne de la commande C002...

NºCmd	Client	Produit	Fournisseur	TelFourn
C002	Sara	Écran	ScreenCo	04.56.78

Catastrophe ! En supprimant la commande de Sara, on perd AUSSI :

- L'existence de la cliente Sara (et qu'elle habite Lyon)
- Le prix de l'Écran (200€)
- L'existence du fournisseur ScreenCo et son téléphone

Résumé : Les 3 Anomalies



Modification

Changer une info oblige à modifier **plusieurs lignes**

Risque d'incohérence



Insertion

Impossible d'ajouter une info
sans données fictives

Données manquantes



Suppression

Supprimer une ligne fait **perdre des infos utiles**

Perte de données

Résumé : Les 3 Anomalies



Modification

Changer une info oblige à modifier **plusieurs lignes**

Risque d'incohérence



Insertion

Impossible d'ajouter une info
sans données fictives

Données manquantes



Suppression

Supprimer une ligne fait **perdre des infos utiles**

Perte de données

La cause de TOUS ces problèmes ? On a mélangé des informations qui n'ont rien à voir dans une seule table ! La solution : **la normalisation.**

Partie 2

Les Dépendances Fonctionnelles

L'outil pour comprendre et corriger

C'est quoi une Dépendance Fonctionnelle ?



Pensez à votre vie quotidienne...

C'est quoi une Dépendance Fonctionnelle ?



Pensez à votre vie quotidienne...

Si je connais votre numéro étudiant, je peux retrouver votre nom.

NumÉtudiant → NomÉtudiant

C'est quoi une Dépendance Fonctionnelle ?



Pensez à votre vie quotidienne...

Si je connais votre numéro étudiant, je peux retrouver votre nom.

NumÉtudiant → NomÉtudiant

Autres exemples :

CodePostal → Ville

Si je connais le code postal, je connais la ville

N°SécuSociale → Nom, Prénom, DateNaissance

Le N° sécu détermine l'identité

C'est quoi une Dépendance Fonctionnelle ?



Pensez à votre vie quotidienne...

Si je connais votre numéro étudiant, je peux retrouver votre nom.

NumÉtudiant → NomÉtudiant

Autres exemples :

CodePostal → Ville

Si je connais le code postal, je connais la ville

N°SécuSociale → Nom, Prénom, DateNaissance

Le N° sécu détermine l'identité

Définition : On dit que **A détermine B** (noté $A \rightarrow B$) si, à chaque valeur de A, correspond **une et une seule** valeur de B.

Attention : DF ≠ coïncidence !

C'est une DF si...

- C'est **toujours vrai**, pas juste dans les données actuelles
- C'est une **règle métier** (du monde réel)
- NumÉtudiant → Nom (un étudiant a toujours un seul nom)

Ce n'est PAS une DF si...

- C'est vrai **par hasard** dans quelques lignes
- Nom → Ville (deux personnes s'appellent « Ali » mais habitent dans des villes différentes !)

Attention : DF ≠ coïncidence !

✓ C'est une DF si...

- C'est **toujours vrai**, pas juste dans les données actuelles
- C'est une **règle métier** (du monde réel)
- NumÉtudiant → Nom (un étudiant a toujours un seul nom)

✗ Ce n'est PAS une DF si...

- C'est vrai **par hasard** dans quelques lignes
- Nom → Ville (deux personnes s'appellent « Ali » mais habitent dans des villes différentes !)

Règle d'or : Une DF vient du **sens des données** (la sémantique), pas d'un simple coup d'œil sur le tableau. C'est une contrainte qui doit être vraie pour **toutes les données possibles**, pas juste celles qu'on voit.

Revenons à la boutique de Mme Dupont

Identifions les dépendances fonctionnelles dans son fichier :

N°Cmd	DateCmd	Client	Ville	Produit	PrixUnit	Qté	Fournisseur	TelFourn
C001	15/01	Ali	Paris	Clavier	25€	2	TechPlus	01.23.45
C001	15/01	Ali	Paris	Souris	15€	1	TechPlus	01.23.45
C002	16/01	Sara	Lyon	Écran	200€	1	ScreenCo	04.56.78
C003	17/01	Ali	Paris	Clavier	25€	3	TechPlus	01.23.45

Revenons à la boutique de Mme Dupont

Identifions les dépendances fonctionnelles dans son fichier :

N°Cmd	DateCmd	Client	Ville	Produit	PrixUnit	Qté	Fournisseur	TelFourn
C001	15/01	Ali	Paris	Clavier	25€	2	TechPlus	01.23.45
C001	15/01	Ali	Paris	Souris	15€	1	TechPlus	01.23.45
C002	16/01	Sara	Lyon	Écran	200€	1	ScreenCo	04.56.78
C003	17/01	Ali	Paris	Clavier	25€	3	TechPlus	01.23.45

Les DF trouvées :

N°Cmd → DateCmd, Client

Client → Ville

Produit → PrixUnit, Fournisseur

Fournisseur → TelFourn

(N°Cmd, Produit) → Qté

Les 3 types de Dépendances Fonctionnelles

Les 3 types de Dépendances Fonctionnelles

1. DF Totale (complète)

L'attribut dépend de **toute** la clé, pas d'une partie.

(N°Cmd, Produit) → Qté

La quantité dépend de la commande ET du produit ensemble. Ni l'un ni l'autre seul ne suffit.

Les 3 types de Dépendances Fonctionnelles

1. DF Totale (complète)

L'attribut dépend de **toute** la clé, pas d'une partie.

(N°Cmd, Produit) → Qté

La quantité dépend de la commande ET du produit ensemble. Ni l'un ni l'autre seul ne suffit.

2. DF Partielle

L'attribut ne dépend que d'une **partie** de la clé composite.

(N°Cmd, Produit) → DateCmd ⚡ car N°Cmd → DateCmd suffit !

La date dépend seulement du N°Cmd, pas besoin du Produit.

Les 3 types de Dépendances Fonctionnelles

1. DF Totale (complète)

L'attribut dépend de **toute** la clé, pas d'une partie.

(N°Cmd, Produit) → Qté

La quantité dépend de la commande ET du produit ensemble. Ni l'un ni l'autre seul ne suffit.

2. DF Partielle

L'attribut ne dépend que d'une **partie** de la clé composite.

(N°Cmd, Produit) → DateCmd ⚡ car N°Cmd → DateCmd suffit !

La date dépend seulement du N°Cmd, pas besoin du Produit.

3. DF Transitive

A → B → C, donc A détermine C **indirectement**.

N°Cmd → Client → Ville

La ville dépend du client, qui lui-même dépend du N°Cmd. C'est indirect !

Clés : L'identifiant unique

Clé candidate : Le plus petit ensemble d'attributs qui détermine **tous** les autres attributs de la table. Si on enlève un attribut, ça ne marche plus.

Clés : L'identifiant unique

Clé candidate : Le plus petit ensemble d'attributs qui détermine **tous** les autres attributs de la table. Si on enlève un attribut, ça ne marche plus.

Clé Primaire

La clé candidate choisie pour identifier chaque ligne. Une seule par table.

Exemple : **N°Cmd** dans la table Commandes

Super-clé

Tout ensemble contenant une clé candidate. Détermine tout, mais n'est pas forcément minimal.

Exemple : **(N°Cmd, DateCmd)** — fonctionne mais DateCmd est inutile

Clés : L'identifiant unique

Clé candidate : Le plus petit ensemble d'attributs qui détermine **tous** les autres attributs de la table. Si on enlève un attribut, ça ne marche plus.

Clé Primaire

La clé candidate choisie pour identifier chaque ligne. Une seule par table.

Exemple : **N°Cmd** dans la table Commandes

Super-clé

Tout ensemble contenant une clé candidate. Détermine tout, mais n'est pas forcément minimal.

Exemple : **(N°Cmd, DateCmd)** — fonctionne mais DateCmd est inutile

Astuce : Dans le fichier de Mme Dupont, quelle est la clé ?

(N°Cmd, Produit) — car c'est le couple minimal qui détermine toutes les autres colonnes.

Partie 3

La Normalisation

Découper pour mieux organiser

C'est quoi, la normalisation ?



Analogie : Ranger un appartement

Tout mettre dans un seul tiroir = le bazar. Ranger par catégorie = efficace !

C'est quoi, la normalisation ?



Analogie : Ranger un appartement

Tout mettre dans un seul tiroir = le bazar. Ranger par catégorie = efficace !

Normalisation = Décomposer une grosse table en plusieurs petites tables bien organisées, en **éliminant les redondances** et les anomalies.

C'est quoi, la normalisation ?



Analogie : Ranger un appartement

Tout mettre dans un seul tiroir = le bazar. Ranger par catégorie = efficace !

Normalisation = Décomposer une grosse table en plusieurs petites tables bien organisées, en **éliminant les redondances** et les anomalies.

La feuille de route :



Chaque étape corrige un type de problème. On avance pas à pas.

Première Forme Normale

Valeurs atomiques uniquement

1FN – La règle : tout doit être atomique

Règle : Chaque cellule contient **une seule valeur**, simple et indivisible (= atomique). Pas de listes, pas de groupes répétitifs.

1FN – La règle : tout doit être atomique

Règle : Chaque cellule contient **une seule valeur**, simple et indivisible (= atomique). Pas de listes, pas de groupes répétitifs.

Scénario : Le fichier de Mme Dupont (version pire)

Mme Dupont a aussi un onglet "Clients" avec les téléphones...

1FN – La règle : tout doit être atomique

Règle : Chaque cellule contient **une seule valeur**, simple et indivisible (= atomique). Pas de listes, pas de groupes répétitifs.

Scénario : Le fichier de Mme Dupont (version pire)

Mme Dupont a aussi un onglet "Clients" avec les téléphones...

✗ PAS en 1FN

Client	Téléphones
Ali	06.11, 07.22, 01.33
Sara	06.44, 07.55

Plusieurs valeurs dans une cellule !

✓ En 1FN

Client	Téléphone
Ali	06.11
Ali	07.22
Ali	01.33
Sara	06.44
Sara	07.55

Une valeur par cellule !

1FN – Autre exemple : adresse structurée

✗ PAS en 1FN

Client	Adresse
Ali	5, Rue Victor Hugo, 75001, Paris
Sara	12, Av. Pasteur, 69002, Lyon

L'adresse contient 4 informations en une !

✓ En 1FN

Client	N°	Rue	CP	Ville
Ali	5	V. Hugo	75001	Paris
Sara	12	Pasteur	69002	Lyon

Chaque info a sa propre colonne !

1FN – Autre exemple : adresse structurée

✗ PAS en 1FN

Client	Adresse
Ali	5, Rue Victor Hugo, 75001, Paris
Sara	12, Av. Pasteur, 69002, Lyon

L'adresse contient 4 informations en une !

✓ En 1FN

Client	N°	Rue	CP	Ville
Ali	5	V. Hugo	75001	Paris
Sara	12	Pasteur	69002	Lyon

Chaque info a sa propre colonne !

Résumé 1FN : Éliminer les attributs multivalués et les attributs composés. Chaque cellule = UNE valeur atomique.

Deuxième Forme Normale

Éliminer les dépendances partielles

2FN – La règle

Règle : La table est en 1FN, ET chaque attribut non-clé dépend de **TOUTE** la clé primaire, pas seulement d'une partie.

2FN – La règle

Règle : La table est en 1FN, ET chaque attribut non-clé dépend de **TOUTE** la clé primaire, pas seulement d'une partie.

Note importante : Si la clé primaire est un **seul attribut** (pas composite), la table est automatiquement en 2FN ! Ce problème ne concerne que les **clés composites**.

2FN – La règle

Règle : La table est en 1FN, ET chaque attribut non-clé dépend de **TOUTE** la clé primaire, pas seulement d'une partie.

Note importante : Si la clé primaire est un **seul attribut** (pas composite), la table est automatiquement en 2FN ! Ce problème ne concerne que les **clés composites**.

Retour chez Mme Dupont

Sa table a la clé composite (**N°Cmd, Produit**). Vérifions si tous les attributs dépendent bien de la clé *entièr*e...

2FN – Identifier les dépendances partielles

NºCmd	Produit	DateCmd	Client	Ville	PrixUnit	Qté	Fournisseur	TelFourn
C001	Clavier	15/01	Ali	Paris	25€	2	TechPlus	01.23.45

2FN – Identifier les dépendances partielles

NºCmd	Produit	DateCmd	Client	Ville	PrixUnit	Qté	Fournisseur	TelFourn
C001	Clavier	15/01	Ali	Paris	25€	2	TechPlus	01.23.45

Analyse attribut par attribut :

Attribut	Dépend de...	Type	Problème ?
Qté	(NºCmd, Produit)	Totale	<input checked="" type="checkbox"/> OK

2FN – Identifier les dépendances partielles

NºCmd	Produit	DateCmd	Client	Ville	PrixUnit	Qté	Fournisseur	TelFourn
C001	Clavier	15/01	Ali	Paris	25€	2	TechPlus	01.23.45

Analyse attribut par attribut :

Attribut	Dépend de...	Type	Problème ?
Qté	(NºCmd, Produit)	Totale	✓ OK
DateCmd	NºCmd seul	Partielle	⚠ Problème !

2FN – Identifier les dépendances partielles

NºCmd	Produit	DateCmd	Client	Ville	PrixUnit	Qté	Fournisseur	TelFourn
C001	Clavier	15/01	Ali	Paris	25€	2	TechPlus	01.23.45

Analyse attribut par attribut :

Attribut	Dépend de...	Type	Problème ?
Qté	(NºCmd, Produit)	Totale	✓ OK
DateCmd	NºCmd seul	Partielle	⚠ Problème !
Client	NºCmd seul	Partielle	⚠ Problème !

2FN – Identifier les dépendances partielles

NºCmd	Produit	DateCmd	Client	Ville	PrixUnit	Qté	Fournisseur	TelFourn
C001	Clavier	15/01	Ali	Paris	25€	2	TechPlus	01.23.45

Analyse attribut par attribut :

Attribut	Dépend de...	Type	Problème ?
Qté	(NºCmd, Produit)	Totale	✓ OK
DateCmd	NºCmd seul	Partielle	⚠ Problème !
Client	NºCmd seul	Partielle	⚠ Problème !
PrixUnit	Produit seul	Partielle	⚠ Problème !

2FN – Identifier les dépendances partielles

N°Cmd	Produit	DateCmd	Client	Ville	PrixUnit	Qté	Fournisseur	TelFourn
C001	Clavier	15/01	Ali	Paris	25€	2	TechPlus	01.23.45

Analyse attribut par attribut :

Attribut	Dépend de...	Type	Problème ?
Qté	(N°Cmd, Produit)	Totale	✓ OK
DateCmd	N°Cmd seul	Partielle	⚠ Problème !
Client	N°Cmd seul	Partielle	⚠ Problème !
PrixUnit	Produit seul	Partielle	⚠ Problème !
Fournisseur	Produit seul	Partielle	⚠ Problème !

2FN – La décomposition

Recette : On *sort* chaque dépendance partielle dans sa propre table.

2FN – La décomposition

Recette : On *sort* chaque dépendance partielle dans sa propre table.

Table Commandes

N°Cmd	DateCmd	Client
C001	15/01	Ali
C002	16/01	Sara
C003	17/01	Ali

N°Cmd → DateCmd, Client

Table Produits

Produit	PrixUnit	Fournisseur
Clavier	25€	TechPlus
Souris	15€	TechPlus
Écran	200€	ScreenCo

Produit → PrixUnit, Fourn.

Table LignesCommande

N°Cmd	Produit	Qté
C001	Clavier	2
C001	Souris	1
C002	Écran	1
C003	Clavier	3

(N°Cmd, Prod.) → Qté

2FN – La décomposition

Recette : On *sort* chaque dépendance partielle dans sa propre table.

Table Commandes

N°Cmd	DateCmd	Client
C001	15/01	Ali
C002	16/01	Sara
C003	17/01	Ali

N°Cmd → DateCmd, Client

Table Produits

Produit	PrixUnit	Fournisseur
Clavier	25€	TechPlus
Souris	15€	TechPlus
Écran	200€	ScreenCo

Produit → PrixUnit, Fourn.

Table LignesCommande

N°Cmd	Produit	Qté
C001	Clavier	2
C001	Souris	1
C002	Écran	1
C003	Clavier	3

(N°Cmd, Prod.) → Qté

Résultat : Plus de dépendances partielles ! Chaque info liée au N°Cmd est dans Commandes, chaque info liée au Produit est dans Produits, et seule la Qté (qui dépend des deux) reste dans LignesCommande.

2FN – Est-ce qu'on a tout résolu ?

Regardons la table **Commandes** :

NºCmd	DateCmd	Client	Ville
C001	15/01	Ali	Paris
C002	16/01	Sara	Lyon
C003	17/01	Ali	Paris

2FN – Est-ce qu'on a tout résolu ?

Regardons la table **Commandes** :

NºCmd	DateCmd	Client	Ville
C001	15/01	Ali	Paris
C002	16/01	Sara	Lyon
C003	17/01	Ali	Paris

Encore des redondances ! « Ali / Paris » est répété. Et dans Produits, « TechPlus » est répété.
Pourquoi ? Il reste des *dépendances transitives* :

NºCmd → Client → Ville

La Ville ne dépend pas directement de NºCmd, mais du Client !

2FN – Est-ce qu'on a tout résolu ?

Regardons la table **Commandes** :

NºCmd	DateCmd	Client	Ville
C001	15/01	Ali	Paris
C002	16/01	Sara	Lyon
C003	17/01	Ali	Paris

Encore des redondances ! « Ali / Paris » est répété. Et dans Produits, « TechPlus » est répété.
Pourquoi ? Il reste des *dépendances transitives* :

NºCmd → Client → Ville

La Ville ne dépend pas directement de NºCmd, mais du Client !

Il faut passer en **3^e Forme Normale** pour corriger cela.

Troisième Forme Normale

Éliminer les dépendances transitives

3FN – La règle

Règle : La table est en 2FN, ET aucun attribut non-clé ne dépend d'un **autre attribut non-clé**. Tout doit dépendre directement de la clé, et de rien d'autre.

3FN – La règle

Règle : La table est en 2FN, ET aucun attribut non-clé ne dépend d'un **autre attribut non-clé**. Tout doit dépendre directement de la clé, et de rien d'autre.

La phrase magique pour retenir :

*"Chaque attribut non-clé dépend de la clé, **de toute la clé**, et de **rien d'autre que la clé**."*

— Inspiré du serment judiciaire

3FN – La règle

Règle : La table est en 2FN, ET aucun attribut non-clé ne dépend d'un **autre attribut non-clé**. Tout doit dépendre directement de la clé, et de rien d'autre.

La phrase magique pour retenir :

*"Chaque attribut non-clé dépend de la clé, **de toute la clé**, et de **rien d'autre que la clé**."*

— Inspiré du serment judiciaire

Dépendance transitive = Quand $A \rightarrow B \rightarrow C$ (A détermine B , B détermine C).

C ne dépend pas *directement* de A , mais *à travers* B . Il faut séparer B et C dans leur propre table.

3FN – Identifier les dépendances transitives

Table Commandes (en 2FN) :

N°Cmd	DateCmd	Client	Ville
C001	15/01	Ali	Paris
C002	16/01	Sara	Lyon
C003	17/01	Ali	Paris

Transitivité détectée !

N°Cmd → Client → Ville

La Ville dépend du Client, pas du N°Cmd directement.

Table Produits (en 2FN) :

Produit	PrixUnit	Fournisseur	TelFourn
Clavier	25€	TechPlus	01.23.45
Souris	15€	TechPlus	01.23.45
Écran	200€	ScreenCo	04.56.78

Transitivité détectée !

Produit → Fournisseur → TelFourn

Le téléphone dépend du Fournisseur, pas du Produit.

3FN – La décomposition finale

Recette : On sort chaque dépendance transitive dans sa propre table.

3FN – La décomposition finale

Recette : On sort chaque dépendance transitive dans sa propre table.

Clients

Client	Ville
Ali	Paris
Sara	Lyon

Commandes

N°Cmd	DateCmd	Client (FK)
c001	15/01	Ali
c002	16/01	Sara
c003	17/01	Ali

Fournisseurs

Fournisseur	TelFourn
TechPlus	01.23.45
ScreenCo	04.56.78

Produits

Produit	PrixUnit	Fourn. (FK)
Clavier	25€	TechPlus
Souris	15€	TechPlus
Écran	200€	ScreenCo

3FN – La décomposition finale

Recette : On sort chaque dépendance transitive dans sa propre table.

Clients

Client	Ville
Ali	Paris
Sara	Lyon

Commandes

N°Cmd	DateCmd	Client (FK)
c001	15/01	Ali
c002	16/01	Sara
c003	17/01	Ali

Fournisseurs

Fournisseur	TelFourn
TechPlus	01.23.45
ScreenCo	04.56.78

Produits

Produit	PrixUnit	Fourn. (FK)
Clavier	25€	TechPlus
Souris	15€	TechPlus
Écran	200€	ScreenCo

+ LignesCommande (inchangée)

N°Cmd	Produit	Oté
C001	Clavier	2
C001	Souris	1
C002	Écran	1
C003	Clavier	3

3FN – Vérifions : les anomalies sont-elles résolues ?

Modification

TechPlus change de tel ? On modifie **UNE seule ligne** dans la table Fournisseurs.

Insertion

Nouveau fournisseur « AudioMax » ? On l'ajoute dans la table Fournisseurs, **sans fausse commande**.

Suppression

On supprime la commande C002 ? Sara reste dans la table Clients. ScreenCo reste dans Fournisseurs.

3FN – Vérifions : les anomalies sont-elles résolues ?

Modification

TechPlus change de tel ? On modifie **UNE seule ligne** dans la table Fournisseurs.

Insertion

Nouveau fournisseur « AudioMax » ? On l'ajoute dans la table Fournisseurs, **sans fausse commande.**

Suppression

On supprime la commande C002 ? Sara reste dans la table Clients. ScreenCo reste dans Fournisseurs.

Avant vs Après

Avant (1 table, 9 colonnes) :

Redondances, anomalies, cauchemar de maintenance

Après (5 tables propres) :

Chaque info stockée une seule fois, zéro anomalie

Forme Normale de Boyce-Codd

La 3FN « renforcée » (3.5 FN)

BCNF – La règle

Règle : Pour **toute** dépendance fonctionnelle $X \rightarrow Y$ dans la table, X doit être une **super-clé** (c'est-à-dire contenir une clé candidate).

BCNF – La règle

Règle : Pour **toute** dépendance fonctionnelle $X \rightarrow Y$ dans la table, X doit être une **super-clé** (c'est-à-dire contenir une clé candidate).

3FN dit :

Les attributs non-clé ne dépendent que de la clé.
(mais tolère certains cas avec des clés candidates multiples)

BCNF dit :

TOUT déterminant (le côté gauche d'une DF) doit être une super-clé.
Aucune exception.

BCNF – La règle

Règle : Pour **toute** dépendance fonctionnelle $X \rightarrow Y$ dans la table, X doit être une **super-clé** (c'est-à-dire contenir une clé candidate).

3FN dit :

Les attributs non-clé ne dépendent que de la clé.
(mais tolère certains cas avec des clés candidates multiples)

BCNF dit :

TOUT déterminant (le côté gauche d'une DF) doit être une super-clé.
Aucune exception.

En pratique : La plupart des tables en 3FN sont déjà en BCNF. La différence n'apparaît que dans des cas particuliers avec **plusieurs clés candidates qui se chevauchent**.

BCNF – Un exemple concret

Scénario : Cours à l'université

Un étudiant suit des matières, chaque matière est enseignée par un seul prof, et chaque prof n'enseigne qu'une seule matière.

BCNF – Un exemple concret

Scénario : Cours à l'université

Un étudiant suit des matières, chaque matière est enseignée par un seul prof, et chaque prof n'enseigne qu'une seule matière.

Étudiant	Matière	Professeur
Ali	BDD	M. Martin
Ali	Java	Mme Leroy
Sara	BDD	M. Martin
Sara	Web	M. Petit

BCNF – Un exemple concret

Scénario : Cours à l'université

Un étudiant suit des matières, chaque matière est enseignée par un seul prof, et chaque prof n'enseigne qu'une seule matière.

Étudiant	Matière	Professeur
Ali	BDD	M. Martin
Ali	Java	Mme Leroy
Sara	BDD	M. Martin
Sara	Web	M. Petit

Les DF :

(Étudiant, Matière) → Professeur

Professeur → Matière

Problème BCNF : La DF « Professeur → Matière » a pour déterminant **Professeur**, qui n'est **pas une super-clé** ! (La clé est (Étudiant, Matière))

BCNF – La décomposition

Avant (pas en BCNF) :

Étudiant	Matière	Professeur
Ali	BDD	M. Martin
Ali	Java	Mme Leroy
Sara	BDD	M. Martin
Sara	Web	M. Petit

Anomalie : Si M. Martin change de matière, il faut modifier TOUTES les lignes.

Après (en BCNF) :

Professeur	Matière
M. Martin	BDD
Mme Leroy	Java
M. Petit	Web

Étudiant	Professeur
Ali	M. Martin
Ali	Mme Leroy
Sara	M. Martin
Sara	M. Petit

BCNF – La décomposition

Avant (pas en BCNF) :

Étudiant	Matière	Professeur
Ali	BDD	M. Martin
Ali	Java	Mme Leroy
Sara	BDD	M. Martin
Sara	Web	M. Petit

Anomalie : Si M. Martin change de matière, il faut modifier TOUTES les lignes.

Après (en BCNF) :

Professeur	Matière
M. Martin	BDD
Mme Leroy	Java
M. Petit	Web

Étudiant	Professeur
Ali	M. Martin
Ali	Mme Leroy
Sara	M. Martin
Sara	M. Petit

Maintenant : Chaque déterminant (Professeur, et (Étudiant, Professeur)) est une super-clé dans sa table respective. C'est en BCNF !

Résumé : Les Formes Normales en un coup d'œil

Forme	Exigence	Problème corrigé	Comment faire
1FN	Valeurs atomiques	Listes et groupes répétitifs	Éclater les cellules multi-valeurs
2FN	1FN + pas de DF partielles	Attribut dépend d'une partie de la clé	Séparer dans une nouvelle table
3FN	2FN + pas de DF transitives	Attribut dépend d'un non-clé	Séparer dans une nouvelle table
BCNF	Tout déterminant = super-clé	Déterminant non-clé	Séparer dans une nouvelle table

Résumé : Les Formes Normales en un coup d'œil

Forme	Exigence	Problème corrigé	Comment faire
1FN	Valeurs atomiques	Listes et groupes répétitifs	Éclater les cellules multi-valeurs
2FN	1FN + pas de DF partielles	Attribut dépend d'une partie de la clé	Séparer dans une nouvelle table
3FN	2FN + pas de DF transitives	Attribut dépend d'un non-clé	Séparer dans une nouvelle table
BCNF	Tout déterminant = super-clé	Déterminant non-clé	Séparer dans une nouvelle table

La phrase à retenir :

*"Chaque attribut non-clé dépend de **la clé** (1FN), de **toute la clé** (2FN), et de **rien d'autre que la clé** (3FN)."*

La méthode pas à pas

La méthode pas à pas

1 Lister les dépendances fonctionnelles

À partir des règles métier, pas des données !

La méthode pas à pas

1 Lister les dépendances fonctionnelles

À partir des règles métier, pas des données !

2 Identifier la clé primaire

Le plus petit ensemble qui détermine tout.

La méthode pas à pas

1 Lister les dépendances fonctionnelles

À partir des règles métier, pas des données !

2 Identifier la clé primaire

Le plus petit ensemble qui détermine tout.

3 Vérifier la 1FN

Toutes les valeurs sont atomiques ?

La méthode pas à pas

1 Lister les dépendances fonctionnelles

À partir des règles métier, pas des données !

2 Identifier la clé primaire

Le plus petit ensemble qui détermine tout.

3 Vérifier la 1FN

Toutes les valeurs sont atomiques ?

4 Vérifier la 2FN — Chercher les DF partielles

Un attribut dépend d'une partie de la clé ? → Nouvelle table !

La méthode pas à pas

1 Lister les dépendances fonctionnelles

À partir des règles métier, pas des données !

2 Identifier la clé primaire

Le plus petit ensemble qui détermine tout.

3 Vérifier la 1FN

Toutes les valeurs sont atomiques ?

4 Vérifier la 2FN — Chercher les DF partielles

Un attribut dépend d'une partie de la clé ? → Nouvelle table !

5 Vérifier la 3FN — Chercher les DF transitives

Un non-clé dépend d'un autre non-clé ? → Nouvelle table !

La méthode pas à pas

1 Lister les dépendances fonctionnelles

À partir des règles métier, pas des données !

2 Identifier la clé primaire

Le plus petit ensemble qui détermine tout.

3 Vérifier la 1FN

Toutes les valeurs sont atomiques ?

4 Vérifier la 2FN — Chercher les DF partielles

Un attribut dépend d'une partie de la clé ? → Nouvelle table !

5 Vérifier la 3FN — Chercher les DF transitives

Un non-clé dépend d'un autre non-clé ? → Nouvelle table !

6 Vérifier la BCNF — Tous les déterminants sont des super-clés ?

Sinon → Décomposer !

Partie 4

Exercices guidés

À vous de jouer !

Exercice 1 : La bibliothèque municipale

Scénario

La bibliothécaire Mme Lambert gère les emprunts dans un fichier Excel :

NºEmprunt	DateEmprunt	Adhérent	TelAdh	ISBN	TitreLivre	Auteur	Rayon
E001	10/01	Luc	06.11	978-1	SQL pour Nuls	Dupont	Info
E001	10/01	Luc	06.11	978-2	Java Facile	Martin	Info
E002	12/01	Marie	07.22	978-1	SQL pour Nuls	Dupont	Info
E003	15/01	Luc	06.11	978-3	Le Petit Prince	Saint-Ex	Littérature

Exercice 1 : La bibliothèque municipale

Scénario

La bibliothécaire Mme Lambert gère les emprunts dans un fichier Excel :

NºEmprunt	DateEmprunt	Adhérent	TelAdh	ISBN	TitreLivre	Auteur	Rayon
E001	10/01	Luc	06.11	978-1	SQL pour Nuls	Dupont	Info
E001	10/01	Luc	06.11	978-2	Java Facile	Martin	Info
E002	12/01	Marie	07.22	978-1	SQL pour Nuls	Dupont	Info
E003	15/01	Luc	06.11	978-3	Le Petit Prince	Saint-Ex	Littérature

Questions :

Exercice 1 : La bibliothèque municipale

Scénario

La bibliothécaire Mme Lambert gère les emprunts dans un fichier Excel :

NºEmprunt	DateEmprunt	Adhérent	TelAdh	ISBN	TitreLivre	Auteur	Rayon
E001	10/01	Luc	06.11	978-1	SQL pour Nuls	Dupont	Info
E001	10/01	Luc	06.11	978-2	Java Facile	Martin	Info
E002	12/01	Marie	07.22	978-1	SQL pour Nuls	Dupont	Info
E003	15/01	Luc	06.11	978-3	Le Petit Prince	Saint-Ex	Littérature

Questions :

1. Identifiez les **dépendances fonctionnelles**

Exercice 1 : La bibliothèque municipale

Scénario

La bibliothécaire Mme Lambert gère les emprunts dans un fichier Excel :

NºEmprunt	DateEmprunt	Adhérent	TelAdh	ISBN	TitreLivre	Auteur	Rayon
E001	10/01	Luc	06.11	978-1	SQL pour Nuls	Dupont	Info
E001	10/01	Luc	06.11	978-2	Java Facile	Martin	Info
E002	12/01	Marie	07.22	978-1	SQL pour Nuls	Dupont	Info
E003	15/01	Luc	06.11	978-3	Le Petit Prince	Saint-Ex	Littérature

Questions :

1. Identifiez les **dépendances fonctionnelles**
2. Quelle est la **clé primaire** ?

Exercice 1 : La bibliothèque municipale

Scénario

La bibliothécaire Mme Lambert gère les emprunts dans un fichier Excel :

NºEmprunt	DateEmprunt	Adhérent	TelAdh	ISBN	TitreLivre	Auteur	Rayon
E001	10/01	Luc	06.11	978-1	SQL pour Nuls	Dupont	Info
E001	10/01	Luc	06.11	978-2	Java Facile	Martin	Info
E002	12/01	Marie	07.22	978-1	SQL pour Nuls	Dupont	Info
E003	15/01	Luc	06.11	978-3	Le Petit Prince	Saint-Ex	Littérature

Questions :

1. Identifiez les **dépendances fonctionnelles**
2. Quelle est la **clé primaire** ?
3. Normalisez jusqu'en **3FN**

Exercice 1 : La bibliothèque municipale

Scénario

La bibliothécaire Mme Lambert gère les emprunts dans un fichier Excel :

NºEmprunt	DateEmprunt	Adhérent	TelAdh	ISBN	TitreLivre	Auteur	Rayon
E001	10/01	Luc	06.11	978-1	SQL pour Nuls	Dupont	Info
E001	10/01	Luc	06.11	978-2	Java Facile	Martin	Info
E002	12/01	Marie	07.22	978-1	SQL pour Nuls	Dupont	Info
E003	15/01	Luc	06.11	978-3	Le Petit Prince	Saint-Ex	Littérature

Questions :

1. Identifiez les **dépendances fonctionnelles**
2. Quelle est la **clé primaire** ?
3. Normalisez jusqu'en **3FN**
4. Donnez un exemple d'**anomalie** dans la table originale

Exercice 1 – Correction

1. Dépendances fonctionnelles :

N°Emprunt → DateEmprunt, Adhérent

Adhérent → TelAdh

ISBN → TitreLivre, Auteur, Rayon

(N°Emprunt, ISBN) → tout

Exercice 1 – Correction

1. Dépendances fonctionnelles :

N°Emprunt → DateEmprunt, Adhérent

Adhérent → TelAdh

ISBN → TitreLivre, Auteur, Rayon

(N°Emprunt, ISBN) → tout

2. Clé primaire : (N°Emprunt, ISBN)

Exercice 1 – Correction

1. Dépendances fonctionnelles :

N°Emprunt → DateEmprunt, Adhérent

Adhérent → TelAdh

ISBN → TitreLivre, Auteur, Rayon

(N°Emprunt, ISBN) → tout

2. Clé primaire : (N°Emprunt, ISBN)

3. Décomposition en 3FN :

Adhérents

Adhérent	TelAdh
Luc	06.11
Marie	07.22

Livres

ISBN	Titre	Auteur	Rayon
978-1	SQL...	Dupont	Info
978-2	Java...	Martin	Info
978-3	Petit P.	Saint-Ex	Litt.

Emprunts

N°Emp	Date	Adh.(FK)
E001	10/01	Luc
E002	12/01	Marie
E003	15/01	Luc

LignesEmprunt

N°Emp	ISBN(FK)
E001	978-1
E001	978-2
E002	978-1
E003	978-3

Exercice 2 : L'agence immobilière

Scénario

M. Renaud gère ses locations dans une table unique :

N°Contrat	Locataire	TelLoc	Appart	Adresse	Loyer	Proprio	TelProprio	DateDébut
L001	Paul	06.33	A12	5 Rue Paix	800€	M. Blanc	01.44	01/03
L002	Julie	07.55	A15	10 Av. Gare	650€	M. Blanc	01.44	01/06
L003	Paul	06.33	A20	3 Rue Hugo	900€	Mme Noir	02.66	15/09

Exercice 2 : L'agence immobilière

Scénario

M. Renaud gère ses locations dans une table unique :

N°Contrat	Locataire	TelLoc	Appart	Adresse	Loyer	Proprio	TelProprio	DateDébut
L001	Paul	06.33	A12	5 Rue Paix	800€	M. Blanc	01.44	01/03
L002	Julie	07.55	A15	10 Av. Gare	650€	M. Blanc	01.44	01/06
L003	Paul	06.33	A20	3 Rue Hugo	900€	Mme Noir	02.66	15/09

À vous !

1. Listez les DF
2. Identifiez les anomalies possibles
3. Normalisez en 3FN
4. Vérifiez si c'est en BCNF

Exercice 2 – Correction

DF identifiées :

N°Contrat → Locataire, Appart, DateDébut

Locataire → TelLoc

Appart → Adresse, Loyer, Proprio

Proprio → TelProprio

Exercice 2 – Correction

DF identifiées :

N°Contrat → Locataire, Appart, DateDébut

Locataire → TelLoc

Appart → Adresse, Loyer, Proprio

Proprio → TelProprio

Décomposition en 3FN (4 tables) :

Locataires

Locataire	TelLoc
Paul	06.33
Julie	07.55

Propriétaires

Proprio	TelProprio
M. Blanc	01.44
Mme Noir	02.66

Appartements

Appart	Adresse	Loyer	Proprio(FK)
A12	5 Rue Paix	800€	M. Blanc
A15	10 Av. Gare	650€	M. Blanc
A20	3 Rue Hugo	900€	Mme Noir

Contrats

N°Contrat	Locataire(FK)	Appart(FK)	DateDébut
L001	Paul	A12	01/03
L002	Julie	A15	01/06
L003	Paul	A20	15/09

Exercice 2 – Correction

DF identifiées :

N°Contrat → Locataire, Appart, DateDébut

Locataire → TelLoc

Appart → Adresse, Loyer, Proprio

Proprio → TelProprio

Décomposition en 3FN (4 tables) :

Locataires

Locataire	TelLoc
Paul	06.33
Julie	07.55

Propriétaires

Proprio	TelProprio
M. Blanc	01.44
Mme Noir	02.66

Appartements

Appart	Adresse	Loyer	Proprio(FK)
A12	5 Rue Paix	800€	M. Blanc
A15	10 Av. Gare	650€	M. Blanc
A20	3 Rue Hugo	900€	Mme Noir

Contrats

N°Contrat	Locataire(FK)	Appart(FK)	DateDébut
L001	Paul	A12	01/03
L002	Julie	A15	01/06
L003	Paul	A20	15/09

BCNF ? Oui ! Dans chaque table, tout déterminant est une super-clé. Déjà en BCNF.

Partie 5

Et en NoSQL / MongoDB ?

Même données, autre philosophie

SQL vs NoSQL : deux philosophies

SQL (Relationnel)

- Tables avec lignes et colonnes
- Schéma **fixe** et strict
- Relations via clés étrangères + JOIN
- Normalisation = la règle
- Ex : MySQL, PostgreSQL, Oracle

NoSQL (MongoDB)

- **Documents** JSON (pas de tables !)
- Schéma **flexible**
- Données imbriquées (embedded)
- Pas de JOIN natif
- Ex : MongoDB, CouchDB, Firebase

SQL vs NoSQL : deux philosophies

SQL (Relationnel)

- Tables avec lignes et colonnes
- Schéma **fixe** et strict
- Relations via clés étrangères + JOIN
- Normalisation = la règle
- Ex : MySQL, PostgreSQL, Oracle

NoSQL (MongoDB)

- **Documents** JSON (pas de tables !)
- Schéma **flexible**
- Données imbriquées (embedded)
- Pas de JOIN natif
- Ex : MongoDB, CouchDB, Firebase

Question clé : Comment modéliser le fichier Excel de Mme Dupont en MongoDB ? Il existe **3 approches** possibles.

Rappel : Les données de Mme Dupont

Son fichier Excel contient ces feuilles (après normalisation SQL en 3FN) :

Clients	
Ali	Paris
Sara	Lyon

Fournisseurs	
TechPlus	01.23.45
ScreenCo	04.56.78

Produits		
Clavier	25€	TechPlus
Souris	15€	TechPlus
Écran	200€	ScreenCo

Commandes		
C001	15/01	Ali
C002	16/01	Sara
C003	17/01	Ali

LignesCommande		
C001	Clavier	2
C001	Souris	1
C002	Écran	1
C003	Clavier	3

Rappel : Les données de Mme Dupont

Son fichier Excel contient ces feuilles (après normalisation SQL en 3FN) :

Clients	
Ali	Paris
Sara	Lyon

Fournisseurs	
TechPlus	01.23.45
ScreenCo	04.56.78

Produits		
Clavier	25€	TechPlus
Souris	15€	TechPlus
Écran	200€	ScreenCo

Commandes		
C001	15/01	Ali
C002	16/01	Sara
C003	17/01	Ali

LignesCommande		
C001	Clavier	2
C001	Souris	1
C002	Écran	1
C003	Clavier	3

Voyons les 3 façons de mettre ces données dans MongoDB...

Approche 1 : Normalisée (une collection par table)

Référencée

```
// Collection "clients"
{ "_id": "Ali", "ville": "Paris" }
{ "_id": "Sara", "ville": "Lyon" }

// Collection "fournisseurs"
{ "_id": "TechPlus", "tel": "01.23.45" }
{ "_id": "ScreenCo", "tel": "04.56.78" }

// Collection "produits"
{
  "_id": "Clavier",
  "prix": 25,
  "fournisseur_id": "TechPlus"
}
```

```
// Collection "commandes"
{
  "_id": "C001",
  "date": "2025-01-15",
  "client_id": "Ali",
  "lignes": [
    { "produit_id": "Clavier", "qte": 2 },
    { "produit_id": "Souris", "qte": 1 }
  ]
}
```

Principe : Comme en SQL, chaque « table » = une collection.
On utilise des `_id` pour référencer les autres collections.

Approche 1 : Normalisée (une collection par table)

Référencée

```
// Collection "clients"
{ "_id": "Ali", "ville": "Paris" }
{ "_id": "Sara", "ville": "Lyon" }

// Collection "fournisseurs"
{ "_id": "TechPlus", "tel": "01.23.45" }
{ "_id": "ScreenCo", "tel": "04.56.78" }

// Collection "produits"
{
  "_id": "Clavier",
  "prix": 25,
  "fournisseur_id": "TechPlus"
}
```

```
// Collection "commandes"
{
  "_id": "C001",
  "date": "2025-01-15",
  "client_id": "Ali",
  "lignes": [
    { "produit_id": "Clavier", "qte": 2 },
    { "produit_id": "Souris", "qte": 1 }
  ]
}
```

Principe : Comme en SQL, chaque « table » = une collection.
On utilise des `_id` pour référencer les autres collections.

- Pas de redondance (comme en SQL)
- Modification facile (un seul endroit)

 **Avantages** • Familiar si on vient du monde SQL

✗ Inconvénients

- Pas de `JOIN` natif → requêtes multiples
- Performances médiocres en lecture
- **Anti-pattern MongoDB !** On perd l'intérêt du NoSQL

Approche 2 : Tout embarqué (un seul document)

Embedded

```
// Collection "commandes" - TOUT est embarqué dans le document
{
  "_id": "C001",
  "date": "2025-01-15",
  "client": {
    "nom": "Ali",
    "ville": "Paris"
  },
  "lignes": [
    {
      "produit": "Clavier",
      "prix_unitaire": 25,
      "quantite": 2,
      "fournisseur": { "nom": "TechPlus", "tel": "01.23.45" }
    },
    {
      "produit": "Souris",
      "prix_unitaire": 15,
      "quantite": 1,
      "fournisseur": { "nom": "TechPlus", "tel": "01.23.45" }
    }
  ]
}
```

Approche 2 : Tout embarqué (un seul document)

Embedded

```
// Collection "commandes" - TOUT est embarqué dans le document
{
  "_id": "C001",
  "date": "2025-01-15",
  "client": {
    "nom": "Ali",
    "ville": "Paris"
  },
  "lignes": [
    {
      "produit": "Clavier",
      "prix_unitaire": 25,
      "quantite": 2,
      "fournisseur": { "nom": "TechPlus", "tel": "01.23.45" }
    },
    {
      "produit": "Souris",
      "prix_unitaire": 15,
      "quantite": 1,
      "fournisseur": { "nom": "TechPlus", "tel": "01.23.45" }
    }
  ]
}
```

- **UNE seule requête** pour tout récupérer
- Lecture ultra-rapide



Avantages

- Pas besoin de JOIN

✗ Inconvénients

- **Redondance massive !** (TechPlus/01.23.45 répété partout)
- Anomalies de modification (comme le fichier Excel initial !)
- Limite 16 Mo par document

Approche 3 : Hybride (le meilleur des deux mondes)

Recommandée

```
// Collection "commandes"
// On EMBARQUE ce qui est lu ensemble
// On RÉFÉRENCE ce qui change souvent
{
  "_id": "C001",
  "date": "2025-01-15",
  "client": {
    "nom": "Ali",
    "ville": "Paris"
  },
  "lignes": [
    {
      "produit": "Clavier",
      "prix_unitaire": 25,
      "quantite": 2,
      "fournisseur_id": "TechPlus"
    },
    {
      "produit": "Souris",
      "prix_unitaire": 15,
      "quantite": 1,
      "fournisseur_id": "TechPlus"
    }
  ]
}
```

```
// Collection "fournisseurs"
// Séparée car partagée entre
// beaucoup de produits et
// peut changer (tel, adresse)
{
  "_id": "TechPlus",
  "tel": "01.23.45",
  "adresse": "12 Rue Tech, Paris"
}
{
  "_id": "ScreenCo",
  "tel": "04.56.78",
  "adresse": "5 Av. Écran, Lyon"
}
```

Principe :

Embarquer ce qu'on lit toujours ensemble (client + lignes dans la commande).

Référencer ce qui change souvent ou est partagé (fournisseurs).

Comment choisir ? Les règles d'or

Comment choisir ? Les règles d'or

EMBARQUER (embed) quand :

- Les données sont **toujours lues ensemble** (commande + ses lignes)
- Relation **1-àquelques** (un client a 2-3 adresses)
- Les sous-données **n'existent pas seules** (une ligne de commande sans sa commande ?)
- Les données embarquées **changent rarement**

Comment choisir ? Les règles d'or

EMBARQUER (embed) quand :

- Les données sont **toujours lues ensemble** (commande + ses lignes)
- Relation **1-àquelques** (un client a 2-3 adresses)
- Les sous-données **n'existent pas seules** (une ligne de commande sans sa commande ?)
- Les données embarquées **changent rarement**

RÉFÉRENCER (normalize) quand :

- Les données sont **partagées** entre plusieurs documents (un fournisseur fournit 100 produits)
- Relation **1-à-beaucoup** ou **N-à-N**
- Les données **changent souvent** (prix, téléphone, stock...)
- Le document risque de dépasser **16 Mo**

Comment choisir ? Les règles d'or

EMBARQUER (embed) quand :

- Les données sont **toujours lues ensemble** (commande + ses lignes)
- Relation **1-àquelques** (un client a 2-3 adresses)
- Les sous-données **n'existent pas seules** (une ligne de commande sans sa commande ?)
- Les données embarquées **changent rarement**

RÉFÉRENCER (normalize) quand :

- Les données sont **partagées** entre plusieurs documents (un fournisseur fournit 100 produits)
- Relation **1-à-beaucoup** ou **N-à-N**
- Les données **changent souvent** (prix, téléphone, stock...)
- Le document risque de dépasser **16 Mo**

Lecture fréquente ensemble ? → Embarquer | Écriture/modification fréquente ? → Référencer

Comparaison des 3 approches MongoDB

Critère	Normalisée (réf.)	Tout embarqué	Hybride
Redondance	Aucune	Beaucoup	Minimale
Vitesse lecture	Lente (N requêtes)	Ultra rapide	Rapide
Vitesse écriture	Rapide	Lente (N mises à jour)	Correcte
Cohérence	Facile	Difficile	Gérable
Complexité requêtes	\$lookup partout	Simple find()	find() + rare \$lookup
Taille documents	Petits	Gros (risque 16 Mo)	Raisonnables

Comparaison des 3 approches MongoDB

Critère	Normalisée (réf.)	Tout embarqué	Hybride
Redondance	Aucune	Beaucoup	Minimale
Vitesse lecture	Lente (N requêtes)	Ultra rapide	Rapide
Vitesse écriture	Rapide	Lente (N mises à jour)	Correcte
Cohérence	Facile	Difficile	Gérable
Complexité requêtes	\$lookup partout	Simple find()	find() + rare \$lookup
Taille documents	Petits	Gros (risque 16 Mo)	Raisonnables

En pratique : l'approche hybride est quasi toujours la meilleure !

On embarque les données « proches » et on référence les données « partagées ».

Exemple pas à pas : Excel → MongoDB

Scénario : M. Karim a un fichier Excel « Gestion des projets »

3 feuilles : Employés, Projets, Affectations

Exemple pas à pas : Excel → MongoDB

Scénario : M. Karim a un fichier Excel « Gestion des projets »

3 feuilles : Employés, Projets, Affectations

Feuille « Employés »

ID	Nom	Poste	Départ.
E01	Fatima	Dev	IT
E02	Omar	Designer	Créa
E03	Léa	Chef projet	IT

Feuille « Projets »

ID	NomProjet	Client	Budget
P01	Site Web	ABC Corp	15 000€
P02	App Mobile	XYZ SA	30 000€

Feuille « Affectations »

Employé	Projet	Heures	Rôle
E01	P01	120	Backend
E02	P01	80	UI/UX
E01	P02	200	Fullstack
E03	P02	50	Gestion

Exemple pas à pas : Excel → MongoDB

Scénario : M. Karim a un fichier Excel « Gestion des projets »

3 feuilles : Employés, Projets, Affectations

Feuille « Employés »

ID	Nom	Poste	Départ.
E01	Fatima	Dev	IT
E02	Omar	Designer	Créa
E03	Léa	Chef projet	IT

Feuille « Projets »

ID	NomProjet	Client	Budget
P01	Site Web	ABC Corp	15 000€
P02	App Mobile	XYZ SA	30 000€

Feuille « Affectations »

Employé	Projet	Heures	Rôle
E01	P01	120	Backend
E02	P01	80	UI/UX
E01	P02	200	Fullstack
E03	P02	50	Gestion

Question : Comment modéliser cela en MongoDB ? Analysons les 3 approches...

Étape 1 : Analyser les patterns d'accès

En MongoDB, on ne part pas des données mais des QUESTIONS qu'on pose !

Étape 1 : Analyser les patterns d'accès

En MongoDB, on ne part pas des données mais des QUESTIONS qu'on pose !

Questions typiques de M. Karim :

Q1 : "Affiche-moi le projet P01 avec tous ses membres et leurs heures"

Q2 : "Liste tous les projets de Fatima (E01)"

Q3 : "Mets à jour le téléphone de l'employé E02"

Étape 1 : Analyser les patterns d'accès

En MongoDB, on ne part pas des données mais des QUESTIONS qu'on pose !

Questions typiques de M. Karim :

Q1 : "Affiche-moi le projet P01 avec tous ses membres et leurs heures"

Q2 : "Liste tous les projets de Fatima (E01)"

Q3 : "Mets à jour le téléphone de l'employé E02"

Q1 → Embarquer les affectations dans les projets
(lues ensemble)

Q3 → Référencer les employés (modifiés
indépendamment)

Étape 2 : Le modèle hybride recommandé

```
// Collection "employes"
// Séparée : données de référence
// qui changent (tel, poste)
{
  "_id": "E01",
  "nom": "Fatima",
  "poste": "Développeuse",
  "departement": "IT",
  "email": "fatima@entreprise.fr"
}
```

```
// On DUPLIQUE le nom (rarement
// modifié) pour éviter un $lookup
// à chaque lecture de projet
```

```
// Collection "projets"
// Affectations EMBARQUÉES
{
  "_id": "P01",
  "nom": "Site Web",
  "client": "ABC Corp",
  "budget": 15000,
  "membres": [
    {
      "employe_id": "E01",
      "nom": "Fatima",
      "heures": 120,
      "role": "Backend"
    },
    {
      "employe_id": "E02",
      "nom": "Omar",
      "heures": 80,
      "role": "UI/UX"
    }
  ]
}
```

Étape 2 : Le modèle hybride recommandé

```
// Collection "employes"
// Séparée : données de référence
// qui changent (tel, poste)
{
  "_id": "E01",
  "nom": "Fatima",
  "poste": "Développeuse",
  "departement": "IT",
  "email": "fatima@entreprise.fr"
}
```

```
// On DUPLIQUE le nom (rarement
// modifié) pour éviter un $lookup
// à chaque lecture de projet
```

```
// Collection "projets"
// Affectations EMBARQUÉES
{
  "_id": "P01",
  "nom": "Site Web",
  "client": "ABC Corp",
  "budget": 15000,
  "membres": [
    {
      "employe_id": "E01",
      "nom": "Fatima",
      "heures": 120,
      "role": "Backend"
    },
    {
      "employe_id": "E02",
      "nom": "Omar",
      "heures": 80,
      "role": "UI/UX"
    }
  ]
}
```

Astuce : On duplique le `nom` de l'employé dans les membres (dénormalisation contrôlée). Si le nom change, il faudra mettre à jour les projets aussi — mais c'est rare et ça évite des \$lookup coûteux en lecture.

SQL vs MongoDB : Résumé final

Aspect	SQL (3FN)	MongoDB (Hybride)
Philosophie	Pas de redondance, normaliser au maximum	Optimiser pour les requêtes de lecture
Modélisation	D'abord les données, puis les requêtes	D'abord les requêtes, puis les données
Relations	Clés étrangères + JOIN	Embedded + références (\$lookup)
Redondance	Zéro (si bien normalisé)	Contrôlée (dénormalisation volontaire)
Lecture complexe	JOIN multiples	Un seul find()
Cohérence	Transactions ACID	Cohérence éventuelle
Idéal pour	Banque, comptabilité, ERP	Réseaux sociaux, IoT, e-commerce

SQL vs MongoDB : Résumé final

Aspect	SQL (3FN)	MongoDB (Hybride)
Philosophie	Pas de redondance, normaliser au maximum	Optimiser pour les requêtes de lecture
Modélisation	D'abord les données, puis les requêtes	D'abord les requêtes, puis les données
Relations	Clés étrangères + JOIN	Embedded + références (\$lookup)
Redondance	Zéro (si bien normalisé)	Contrôlée (dénormalisation volontaire)
Lecture complexe	JOIN multiples	Un seul find()
Cohérence	Transactions ACID	Cohérence éventuelle
Idéal pour	Banque, comptabilité, ERP	Réseaux sociaux, IoT, e-commerce

Ni l'un ni l'autre n'est « meilleur » – ils répondent à des besoins différents !

Les points clés à retenir

Les points clés à retenir

1. Une **dépendance fonctionnelle** $A \rightarrow B$ signifie que si on connaît A, on connaît B (et c'est toujours vrai).

Les points clés à retenir

1. Une **dépendance fonctionnelle** $A \rightarrow B$ signifie que si on connaît A, on connaît B (et c'est toujours vrai).
2. Les **anomalies** (modification, insertion, suppression) viennent des **redondances** causées par un mauvais schéma.

Les points clés à retenir

1. Une **dépendance fonctionnelle** $A \rightarrow B$ signifie que si on connaît A, on connaît B (et c'est toujours vrai).
2. Les **anomalies** (modification, insertion, suppression) viennent des **redondances** causées par un mauvais schéma.
3. La normalisation **élimine ces redondances** en décomposant les tables, étape par étape : 1FN \rightarrow 2FN \rightarrow 3FN \rightarrow BCNF.

Les points clés à retenir

1. Une **dépendance fonctionnelle** $A \rightarrow B$ signifie que si on connaît A, on connaît B (et c'est toujours vrai).
2. Les **anomalies** (modification, insertion, suppression) viennent des **redondances** causées par un mauvais schéma.
3. La normalisation **élimine ces redondances** en décomposant les tables, étape par étape : 1FN \rightarrow 2FN \rightarrow 3FN \rightarrow BCNF.
4. La 2FN élimine les **DF partielles**, la 3FN élimine les **DF transitives**, la BCNF exige que tout déterminant soit une **super-clé**.

Les points clés à retenir

1. Une **dépendance fonctionnelle** $A \rightarrow B$ signifie que si on connaît A, on connaît B (et c'est toujours vrai).
2. Les **anomalies** (modification, insertion, suppression) viennent des **redondances** causées par un mauvais schéma.
3. La normalisation **élimine ces redondances** en décomposant les tables, étape par étape : 1FN \rightarrow 2FN \rightarrow 3FN \rightarrow BCNF.
4. La 2FN élimine les **DF partielles**, la 3FN élimine les **DF transitives**, la BCNF exige que tout déterminant soit une **super-clé**.
5. Toujours commencer par **identifier les DF** à partir des règles métier avant de normaliser !

Fin du Cours

Questions ?

Joseph AZAR

joseph.azar@univ-fcomte.fr
IUT Nord Franche-Comté

Ressources :

[Redgate — Functional Dependencies](#)
[Redgate — Update Anomalies](#)

