



كلية العلوم
والتقنيات - مراكش
FACULTE DES SCIENCES
ET TECHNIQUES - MARRAKECH

Faculté des Sciences et Techniques
Marrakech



Module Bases de Données *La Normalisation*

Prof. Sara Qassimi

Introduction à la Normalisation

Définition

- La normalisation est un processus utilisé en **modélisation de bases de données relationnelles** pour structurer les informations de manière optimale.
- Son objectif principal est de **minimiser la redondance des données** et d'**éviter les anomalies** (d'insertion, de mise à jour et de suppression).
- Elle repose sur l'application de **formes normales**, chacune imposant un niveau supplémentaire de structuration.

Pourquoi normaliser une base de données ?

1. **Éliminer les anomalies** : Une base de données non normalisée peut poser des problèmes d'insertion, de mise à jour et de suppression des données.
2. **Améliorer la cohérence** : En normalisant, on assure que chaque donnée est stockée à un seul endroit, ce qui évite les incohérences.
3. **Faciliter les requêtes et la maintenance** : Une base bien organisée est plus facile à interroger et à maintenir.
4. **Optimiser l'espace de stockage** : La réduction de la redondance diminue la taille globale de la base.

Introduction à la Normalisation - Comprendre les anomalies

Donnée :

| Num_Étudiant | Nom | Cours 1 | Prof 1 | Cours 2 | Prof 2 |
|--------------|-----|---------|-----------|----------|------------|
| 123 | Ali | Math | M. Dupont | Physique | Mme Martin |
| 456 | Sam | Math | M. Dupont | Chimie | M. Leroy |

Questions :

1. Identifier les anomalies d'insertion, de suppression et de mise à jour possibles.
2. Proposer une structure normalisée pour résoudre ces problèmes.

Introduction à la Normalisation - Comprendre les anomalies

Donnée :

| Num_Étudiant | Nom | Cours 1 | Prof 1 | Cours 2 | Prof 2 |
|--------------|-----|---------|-----------|----------|------------|
| 123 | Ali | Math | M. Dupont | Physique | Mme Martin |
| 456 | Sam | Math | M. Dupont | Chimie | M. Leroy |

Questions :

1. Identifier les anomalies d'insertion, de suppression et de mise à jour possibles.
2. Proposer une structure normalisée pour résoudre ces problèmes.

Correction :

- **Anomalie d'insertion** : Si un nouvel étudiant s'inscrit mais n'a pas encore de cours, on ne peut pas l'ajouter sans valeurs pour **Cours 1** et **Prof 1**.
- **Anomalie de mise à jour** : Si **M. Dupont** change de nom, il faut mettre à jour **toutes les lignes** où il apparaît.
- **Anomalie de suppression** : Si l'on supprime l'étudiant **123**, on perd l'information qu'un **professeur enseigne une matière**.

Solution : Séparer en plusieurs tables :

1. **Étudiants** (Num_Étudiant, Nom)
2. **Cours** (Code_Cours, Nom_Cours)
3. **Professeurs** (ID_Prof, Nom_Prof)
4. **Inscription** (Num_Étudiant, Code_Cours, ID_Prof)

Dépendances Fonctionnelles

Définition

Une **dépendance fonctionnelle** est une contrainte logique entre les attributs d'une relation qui exprime une **relation de détermination** :

$X \rightarrow Y$ signifie que **pour chaque valeur de X, il existe une valeur unique de Y**.

Autrement dit, si deux lignes d'une table ont **la même valeur pour X**, elles doivent aussi avoir la **même valeur pour Y**.

Exemple :

- $\text{NuméroSS} \rightarrow \text{Nom}, \text{Prénom}$ (Un numéro de sécurité sociale détermine un nom unique).

Exercice : Identifier les dépendances fonctionnelles

Donnée :

| ID_Employé | Nom | Département | Numéro_Bureau |
|------------|-------|-------------|---------------|
| E1 | Jean | IT | B101 |
| E2 | Marie | Finance | B201 |
| E3 | Paul | IT | B101 |

Correction :

- $\text{ID_Employé} \rightarrow \text{Nom}$ ✓ (Chaque employé a un identifiant unique).
- $\text{Numéro_Bureau} \rightarrow \text{Département}$ ✗ (Un bureau peut être partagé entre plusieurs départements, donc ce n'est pas toujours vrai).

Dépendances Fonctionnelles

1. Dépendance Fonctionnelle Triviale

- Une dépendance $X \rightarrow Y$ est **triviale** si **Y est un sous-ensemble de X**.
- Cela signifie que **les attributs de Y sont déjà inclus dans X**, donc cette dépendance ne fournit pas de nouvelle information.

Exemple :

- $\{\text{Nom, Prénom}\} \rightarrow \text{Nom}$ ✓
- $\{\text{Numéro étudiant, Nom}\} \rightarrow \text{Numéro étudiant}$ ✓
- Ces dépendances sont triviales car **l'attribut dépendant fait déjà partie de l'ensemble déterminant**.

Exercice 1 : Identifier les dépendances triviales

| A | B | C |
|----|----|----|
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a2 | b2 | c3 |
| a2 | b2 | c4 |

1. Trouver les dépendances triviales.
2. Justifier pourquoi elles sont triviales.

Correction :

- **Dépendances triviales identifiées :**
 - $\{A, B\} \rightarrow A$
 - $\{A, B\} \rightarrow B$
 - $\{A, B, C\} \rightarrow C$
- **Justification :** Ces dépendances sont triviales car **l'attribut dépendant est déjà inclus dans l'ensemble déterminant**.

Exercice 1 : Identifier les dépendances triviales

| A | B | C |
|----|----|----|
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a2 | b2 | c3 |
| a2 | b2 | c4 |

1. Trouver les dépendances triviales.
2. Justifier pourquoi elles sont triviales.

Dépendances Fonctionnelles

1. Dépendance Fonctionnelle Non Triviale

- Une dépendance $X \rightarrow Y$ est **non triviale** si **Y n'est pas un sous-ensemble de X**.

Exemple :

- $\{\text{Numéro étudiant}\} \rightarrow \{\text{Nom}\}$ ✓
- $\{\text{Code postal}\} \rightarrow \{\text{Ville}\}$ ✓
- Une dépendance non triviale est **utile** car elle montre une **relation réelle entre des attributs distincts**, ce qui permet **d'éliminer les redondances** et de mieux structurer les données.

Identifier les Dépendances Non triviales:

Correction :

- $\{\text{Référence}\} \rightarrow \{\text{Produit}, \text{Fournisseur}\}$

L'attribut **Produit** et **Fournisseur** dépendent uniquement de **Référence**, qui est unique pour chaque produit.

| Produit | Référence | Fournisseur |
|------------|-----------|-------------|
| Ordinateur | P123 | Dell |
| Souris | P456 | Logitech |
| Clavier | P789 | HP |

Dépendances Fonctionnelles

1. Dépendance Fonctionnelle Élémentaire

- Une dépendance fonctionnelle $X \rightarrow Y$ est dite **élémentaire** si elle est **non triviale** et **aucun attribut de X ne peut être retiré sans rendre la dépendance invalide**.

Exemple :

- $\{\text{Numéro étudiant}\} \rightarrow \text{Nom}$ ✓ (Dépendance élémentaire car Numéro étudiant est nécessaire pour déterminer Nom).
- $\{\text{Matricule}\} \rightarrow \text{Salaire}$ ✓ (Le matricule est le seul attribut nécessaire pour déterminer le salaire).

Identifier une dépendance élémentaire

| ID Employé | Nom | Département |
|------------|-------|-------------|
| E1 | Jean | IT |
| E2 | Marie | Finance |
| E3 | Paul | IT |

Correction : $\{\text{ID Employé}\} \rightarrow \{\text{Nom}\}$ ✓

ID Employé est **suffisant et nécessaire** pour déterminer Nom, donc cette dépendance est élémentaire.

Les Règles d'Armstrong

Les **règles d'Armstrong** sont un ensemble d'axiomes qui permettent de **déduire toutes les dépendances fonctionnelles valides** dans une base de données.

♦ **1. Réflexivité** : Si $Y \subseteq X$, alors $X \rightarrow Y$.

- **Exemple** : $\{\text{Nom}, \text{Prénom}\} \rightarrow \text{Nom}$ ✓ (Nom est inclus dans {Nom, Prénom}).

♦ **2. Augmentation** : Si $X \rightarrow Y$, alors $XZ \rightarrow YZ$.

- **Exemple** : $\{\text{ID Étudiant}\} \rightarrow \{\text{Nom}\}$ implique $\{\text{ID Étudiant}, \text{Classe}\} \rightarrow \{\text{Nom}, \text{Classe}\}$ ✓.

♦ **3. Transitivité** : Si $X \rightarrow Y$ et $Y \rightarrow Z$, alors $X \rightarrow Z$.

- **Exemple** : $\{\text{Numéro étudiant}\} \rightarrow \{\text{Nom}\}$ et $\{\text{Nom}\} \rightarrow \{\text{Adresse}\}$ alors $\{\text{Numéro étudiant}\} \rightarrow \{\text{Adresse}\}$ ✓.

♦ **4. Union** : Si $X \rightarrow Y$ et $X \rightarrow Z$, alors $X \rightarrow YZ$.

- **Exemple** : $\{\text{Numéro étudiant}\} \rightarrow \{\text{Nom}\}$ et $\{\text{Numéro étudiant}\} \rightarrow \{\text{Classe}\}$ implique $\{\text{Numéro étudiant}\} \rightarrow \{\text{Nom, Classe}\}$ ✓.

♦ **5. Décomposition** : Si $X \rightarrow YZ$, alors $X \rightarrow Y$ et $X \rightarrow Z$.

- **Exemple** : $\{\text{Matricule}\} \rightarrow \{\text{Salaire, Fonction}\}$ implique $\{\text{Matricule}\} \rightarrow \{\text{Salaire}\}$ et $\{\text{Matricule}\} \rightarrow \{\text{Fonction}\}$ ✓.

♦ **6. Pseudo-transitivité** : Si $X \rightarrow Y$ et $WY \rightarrow Z$, alors $WX \rightarrow Z$.

- **Exemple** : $\{\text{ID Produit}\} \rightarrow \{\text{Catégorie}\}$ et $\{\text{Catégorie, Stock}\} \rightarrow \{\text{Fournisseur}\}$ implique $\{\text{ID Produit, Stock}\} \rightarrow \{\text{Fournisseur}\}$ ✓

Les Règles d'Armstrong (à revoir)

Appliquer les règles d'Armstrong donnée: $B \rightarrow C$

| A | B | C |
|----|----|----|
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a2 | b2 | c3 |
| a2 | b2 | c4 |

✓ Correction :

- **Transitivité** : $A \rightarrow B$, $B \rightarrow C \Rightarrow A \rightarrow C$ ✓
- **Décomposition** : $A \rightarrow BC$ implique $A \rightarrow B$ et $A \rightarrow C$ ✓

Exercice

Une entreprise de vente en ligne souhaite structurer sa base de données en respectant les principes de normalisation. L'extrait suivant représente une partie de la base :

| Numéro Commande | ID Client | Nom Client | Adresse | ID Produit | Nom Produit | Quantité | Prix Unitaire |
|--------------------|--------------|---------------|---------|---------------|----------------|----------|------------------|
| CMD001 | C001 | Alice | Paris | P1001 | Ordinateur | 1 | 1200 € |
| CMD002 | C002 | Bob | Lyon | P2002 | Souris | 2 | 25 € |
| CMD003 | C001 | Alice | Paris | P3003 | Clavier | 1 | 45 € |
| CMD004 | C003 | Charlie | Lille | P1001 | Ordinateur | 1 | 1200 € |

Questions :

1. **Identifier les dépendances fonctionnelles triviales** et expliquer pourquoi elles sont triviales.
2. **Identifier les dépendances fonctionnelles non triviales** et expliquer leur impact sur la redondance des données.
3. **Identifier une dépendance fonctionnelle élémentaire** et justifier pourquoi elle est minimale.
4. **Appliquer les règles d'Armstrong** pour déduire une nouvelle dépendance fonctionnelle.
5. **Proposer une décomposition** en relations normalisées en évitant les anomalies.

Correction

1. Identification des dépendances fonctionnelles triviales :

Une dépendance est triviale si l'attribut dépendant fait déjà partie de l'ensemble déterminant.

Dépendances triviales identifiées :

- $\{\text{Numéro Commande, ID Client}\} \rightarrow \text{Numéro Commande}$
- $\{\text{ID Produit, Nom Produit}\} \rightarrow \text{Nom Produit}$
- $\{\text{Numéro Commande, Quantité}\} \rightarrow \text{Quantité}$

Explication :

Ces dépendances ne fournissent **aucune nouvelle information** car elles répètent une relation déjà évidente dans la structure des données.

Correction

2. Identification des dépendances fonctionnelles non triviales :

Une dépendance est non triviale lorsque l'attribut dépendant n'est pas déjà un sous-ensemble de l'ensemble déterminant.

Dépendances non triviales identifiées :

- $\{\text{Numéro Commande}\} \rightarrow \{\text{ID Client, Nom Client, Adresse}\}$
- $\{\text{ID Produit}\} \rightarrow \{\text{Nom Produit, Prix Unitaire}\}$

Explication :

- La **commande** (Numéro Commande) détermine l'**identité du client** (ID Client, Nom, Adresse), ce qui montre une **dépendance logique importante**.
- L'**ID Produit** détermine le **Nom du Produit et son Prix**, ce qui signifie que les prix unitaires des produits ne changent pas selon les commandes.

➡ **Problème de redondance :**

Si **Nom Client** et **Adresse** sont stockés avec chaque commande, ils seront **dupliqués** à chaque nouvelle commande du même client, ce qui pose un risque d'anomalie de mise à jour.

Correction

3. Identification d'une dépendance fonctionnelle élémentaire :

Une dépendance est élémentaire si elle est non triviale et que **tous les attributs du côté gauche sont nécessaires**.

 **Dépendance élémentaire identifiée :**

- $\{\text{ID Client}\} \rightarrow \{\text{Nom Client}, \text{Adresse}\}$

 **Explication :**

Si nous connaissons **ID Client**, nous pouvons **retrouver** **Nom Client** et **Adresse**, et nous ne pouvons pas simplifier **{ID Client}** davantage. C'est donc une dépendance élémentaire.

Correction

4. Application des règles d'Armstrong :

Nous allons utiliser les règles pour déduire une nouvelle dépendance.

Règles utilisées :

- **Transitivité** : Si $\text{Numéro Commande} \rightarrow \text{ID Client}$ et $\text{ID Client} \rightarrow \text{Nom Client, Adresse}$, alors $\text{Numéro Commande} \rightarrow \text{Nom Client, Adresse}$ ✓
- **Décomposition** : $\text{ID Produit} \rightarrow \text{Nom Produit, Prix Unitaire}$ implique $\text{ID Produit} \rightarrow \text{Nom Produit}$ et $\text{ID Produit} \rightarrow \text{Prix Unitaire}$ ✓



Explication :

- Grâce à la **transitivité**, nous avons prouvé que Numéro Commande permet de retrouver **toutes les informations clients**, évitant ainsi les répétitions inutiles.
- Avec la **décomposition**, nous savons que Nom Produit et Prix Unitaire peuvent être extraits séparément.

Correction

5. Proposition d'une décomposition normalisée :

Pour éviter les anomalies et assurer une structure efficace, nous proposons trois relations :

🚩 Table Clients :

| ID Client | Nom Client | Adresse |
|-----------|------------|---------|
| C001 | Alice | Paris |
| C002 | Bob | Lyon |
| C003 | Charlie | Lille |

🚩 Table Produits :

| ID Produit | Nom Produit | Prix Unitaire |
|------------|-------------|---------------|
| P1001 | Ordinateur | 1200 € |
| P2002 | Souris | 25 € |
| P3003 | Clavier | 45 € |

🚩 Table Commandes :

| Numéro Commande | ID Client | ID Produit | Quantité |
|-----------------|-----------|------------|----------|
| CMD001 | C001 | P1001 | 1 |
| CMD002 | C002 | P2002 | 2 |
| CMD003 | C001 | P3003 | 1 |
| CMD004 | C003 | P1001 | 1 |



Explication :

❶ Clients : Séparés des commandes pour éviter de répéter **Nom Client** et **Adresse**.

❷ Produits : Séparés pour éviter la redondance de **Nom Produit** et **Prix Unitaire**.

❸ Commandes : Contient uniquement les clés de **Clients** et **Produits**, ce qui réduit la redondance et améliore l'intégrité des données.

Première forme normale (1FN)

- Une relation est en première forme normale si, et seulement si, tout attribut contient une valeur atomique (non multiples, non composées).
- Par exemple, le pseudo schéma de relation **Personne** n'est pas en première forme normale.

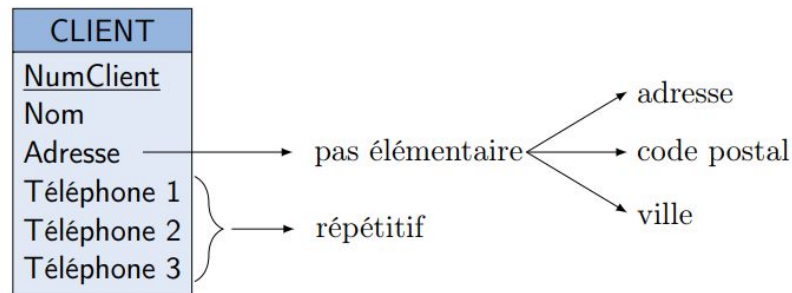
Personne(num-personne, nom, prénom, rue, ville, prénoms-enfants)

Il faut le décomposer en :

Personne(num-personne, nom, prénom, rue, ville)

Prénoms-enfants(#num-personne, # num-prénom)

Prénoms(num-prénom, prénom)



Deuxième forme normale (2FN)

- Une relation est en deuxième forme normale si, et seulement si, elle est en première forme normale et si toutes les dépendances fonctionnelles entre la clé et les autres attributs sont élémentaires.
- Autrement dit, une relation est en deuxième forme normale si, et seulement si, elle est en première forme normale et si tout attribut n'appartenant pas à la clé ne dépend pas que d'une partie de la clé.
- Une relation peut être en deuxième forme normale par rapport à une de ses clés candidates et ne pas l'être par rapport à une autre. Une relation avec une clé primaire réduite à un seul attribut est, par définition, forcément en deuxième forme normale.
- Par exemple une relation avec comme identifiant A et B, si $B \rightarrow C$ alors la relation n'est pas en 2FN

Deuxième forme normale (2FN)

- Soit, par exemple, le schéma de relation suivant :

Commande (**Num-Commande**, **Num-Client**, **Quantité**, **Nom_Client**, **Date**)

- Cette relation indique qu'un client (identifié par Num-Client) a passé une commande (identifiée par Num-Commande). Elle est bien en première forme normale.
- Par contre, les attributs **Quantité**, **Date** et **Nom_Client** ne dépendent pas de
- **Num-Commande**, et de **Num-Client**.

Num-Client, Num-Commande $\dashv\rightarrow$ **Quantité, Date, Nom_Client**

- Cette relation n'est donc pas en deuxième forme normale. Une solution simple pour la normaliser est de la remplacer par :

Client (**Num-Client**, **Nom_Client**)

Commande (**Num-Commande**, **Quantité**, **Date**, **#Num-Client**)

Troisième forme normale (3FN)

- Une relation est en troisième forme normale si, et seulement si, elle est en deuxième forme normale et si toutes les dépendances fonctionnelles entre la clé et les autres attributs sont élémentaires et directes.
- Autrement dit, une relation est en troisième forme normale si, et seulement si, elle est en deuxième forme normale et si tout attribut n'appartenant pas à la clé ne dépend pas d'un attribut non-clé.
 - Exemple d'une relation avec comme identifiant A et B, si $C \rightarrow D$ alors la relation n'est pas en 3FN.
 - Exemple d'une relation avec comme identifiant A et B, si $C \rightarrow B$ alors la relation n'est pas en 3FN.

Troisième forme normale (3FN)

- Soit, par exemple, le schéma de relation suivant :

CommandeLivre(Num-Commande, #Num-Client, *Titre -Livre*, Auteur, Quantité, Prix)

- Cette relation est bien en deuxième forme normale. Par contre, les attributs **Auteur** et **Prix** dépendent de l'attribut **Titre -Livre**.

***Titre -Livre* > Auteur, Prix**

- La relation n'est donc pas en troisième forme normale. Pour la normaliser, il faut la décomposer de la manière suivante :

CommandeLivre(Num-Commande, #Num-Client, #*Titre -Livre*, Quantité)

Livre(*Titre -Livre*, Auteur, Prix)

Forme normale de BOYCE-CODD

- Une relation est en forme normale de BOYCE-CODD (BCNF ou 4FN) si, et seulement si, elle est en troisième forme normale et si les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut.
- Cette forme normale permet de renforcer certaines lacunes de la troisième forme normale. Par exemple une relation avec comme identifiant A et B, si $A \rightarrow B$ alors la relation n'est pas en 4FN.
- Soit, par exemple, le schéma relationnel décrivant l'enseignement d'une matière donnée à une classe par un enseignant :
 - Matière(nom-matière)
 - Classe(num-classe)
 - Enseignant(nom-enseignant)
 - Enseignement(nom-enseignant, num-classe, nom-matière)

Forme normale de BOYCE-CODD

- Supposons, de plus, qu'une matière n'est enseignée qu'une seule fois dans une classe et que par un seul enseignant, et qu'un enseignant n'enseigne qu'une seule matière. Chacune des relations respecte bien la troisième forme normale. Cependant, dans la relation Enseignement, nous avons les dépendances fonctionnelles élémentaires suivantes :

1. nom-matière, num-classe -> nom-enseignant

2. nom-enseignant -> nom-matière

Il existe donc des dépendances fonctionnelles élémentaires dont la source n'est pas la clé de la relation.

Forme normale de BOYCE-CODD

| nom-enseignant | num-classe | nom-matière |
|-----------------------|-------------------|--------------------|
| George | 5 | Physique |
| George | 6 | Physique |
| George | 7 | Physique |
| George | 8 | Physique |
| Michael | 5 | Mathématiques |
| Michael | 6 | Mathématiques |
| Michael | 7 | Mathématiques |
| Michael | 8 | Mathématiques |

Exemple de relation présentant une redondance due au non respect de la forme normale de BOYCE-CODD.

Forme normale de BOYCE-CODD

- Le non respect de la forme normale de BOYCE-CODD entraîne une redondance illustrée par la table pour chaque nom-enseignant identifiant un enseignant, il faut répéter le nom-matière identifiant la matière qu'il enseigne.
- Pour normaliser la relation Enseignement, il faut la décomposer pour aboutir au schéma relationnel suivant :
 - Matière(nom-matière)
 - Classe(num-classe)
 - Enseignant(nom-enseignant, #nom-matière)
 - Enseigner(#num-classe, #nom-matière, #nom-enseignant)

Exercice

Pourquoi la relation **COMMANDE V4** n'est pas en 4FN ? Proposer une solution pour que la relation soit en 4FN.

COMMANDE V4 :

| # Nom four | <u>N° com</u> | <u>N° client</u> | Date com |
|------------|---------------|------------------|------------|
| DURAND | C54 | 15 | 12/01/2016 |
| DUPONT | C55 | 15 | 10/05/2016 |
| BRUNEAU | C17 | 18 | 01/08/2016 |
| DURAND | C574 | 85 | 30/12/2015 |

Exercices:

- Soit le schéma de BD Relationnel suivant :

COMMANDE(numCom, datCom, numCli, nomCli, adrCli, RédCom)

ARTICLES (numCom, numArt, nomArt, desArt, qtéCom, PUArt, PT, RédAcc)

- RG1 : Une commande est constituée de plusieurs articles.
- RG2 : Un article commandé l'est pour une certaine quantité et pour un prix total (PT) donné.

Ce schéma est-il en forme normale ? Pourquoi ?

Appliquer les 4 règles de normalisation sur le modèle pour vérification