

CHAPITRE 2.2



LA NORMALISATION DU MODÈLE RELATIONNEL

plan



- 1) Introduction à la normalisation des relations
 - 1.1 Qu'est ce que la normalisation
 - 1.2 Objectifs
- 2) Les dépendances fonctionnelles
 - 1.1 Définitions
 - 1.2 Représentation des DFs sous forme de graphes
- 3) Normalisation du modèle relationnel
 - 1.1 Première forme normale
 - 1.2 Deuxième forme normale
 - 1.3 Troisième forme normale
- 4) exemples

Notion de normalisation



Etymologie : du latin *norma*, règle.

Plusieurs sens (dans plusieurs domaines)

- **Sens 1**

Une **norme** est une **règle**, une loi auxquelles **on doit se conformer**.

- **Sens 2**

Une **norme** est un **ensemble de caractéristiques décrivant et régissant** un domaine particulier, **un objet**, un produit, un être.

- **Sens 3...**

Objectifs de la normalisation (1)



- L'objectif étant de construire un schéma de base de données ***cohérent***.
- Le but essentiel de la normalisation est d'éviter les anomalies transactionnelles pouvant découler d'une mauvaise modélisation des données et ainsi éviter un certain nombre de problèmes potentiels tels que les anomalies de lecture, les anomalies d'écriture, la redondance des données et la contre-performance.

Objectifs de la normalisation (2)



La normalisation des modèles de données permet de vérifier la robustesse de leur conception pour **améliorer la modélisation** (et donc obtenir une meilleure représentation) et faciliter la mémorisation des données **en évitant la redondance et les problèmes sous-jacents de mise à jour ou de cohérence.**

Conséquence:

L'élimination des redondances doit aboutir à une minimisation de l'espace de stockage.

Mauvaise modélisation: conséquences



Un mauvais schéma logique peut conduire à un certain nombre d'anomalies pendant la phase d'exploitation de la base de donnée.

Problème de la redondance



Exemple: Soit la relation **COMMANDE_PRODUIT**.

<i>Num Prod</i>	<i>Qté</i>	<i>Num Four</i>	<i>Adr</i>
101	300	901	Pêcherie, Alger
104	1000	902	Pêcherie, Bouharoun
112	78	904	Pêcherie, Bousmail
103	250	901	Pêcherie, Alger
104	3000	901	Pêcherie, Alger
104	500	903	Pêcherie, Douaouda
112	150	901	Pêcherie, Alger

Anomalies de modification

Mise à jour de l'adresse d'un fournisseur: il faut le faire pour tous les tuples concernés.

Anomalies d'insertion :

Ajout d'un nouveau fournisseur: il faut obligatoirement fournir des valeurs pour *NumProd* et *Qté*.

Anomalies de suppression :

Suppression du produit *104*: perte de toutes les données sur le fournisseur *902*.

Modèle normalisé



- un modèle relationnel est dit **normalisé**, s'il respecte certaines contraintes (règles ou normes) appelées **les formes normales**.
- Les formes normales s'appuient sur les dépendances fonctionnelles entre attributs.

DÉPENDANCE FONCTIONNELLE (DF)



DÉFINITION:

Deux rubriques R_1 et R_2 sont dites **en Dépendance Fonctionnelle** si le fait de connaître la valeur de R_1 permet de connaître une valeur et une seule de R_2 .

On écrit: $R_1 \rightarrow R_2$.

R_1 est la source de la DF et R_2 le but.

Exemples:

numéro client ----- \rightarrow nom client

matricule étudiant ---- \rightarrow âge, adresse

La question fondamentale qui se pose : « *connaissant une valeur de la source, peut-on connaître une valeur unique du but?* ».

Dépendance fonctionnelle à partie gauche composée



Il peut arriver que ce soit la combinaison de plusieurs attributs (en source) qui permettent de connaître une valeur unique du but.

Exemple : un numéro de facture + un code produit, nous donne la quantité facturée.

numéro de facture, code produit \longrightarrow quantité facturée

On parlera de dépendance fonctionnelle à partie gauche composée (DFPGC).

Dépendance fonctionnelle élémentaire

Une dépendance fonctionnelle $R_1 \xrightarrow{\quad} R_2$ est élémentaire s'il n'existe pas une R_3 , sous-ensemble de R_1 , qui assure elle-même une dépendance fonctionnelle $R_3 \longrightarrow R_2$.

Référence article	\longrightarrow	nom article
(Numéro facture, référence article)	\longrightarrow	quantité facturée
(Numéro facture, référence article)	\longrightarrow	nom article

- La première est élémentaire car la référence article permet directement de connaître son nom.
- La deuxième est élémentaire car le numéro de facture seul ou la référence article seule ne permet pas de connaître la quantité facturée.
- La troisième n'est pas élémentaire car on peut se passer du numéro de facture pour trouver le nom de l'article.

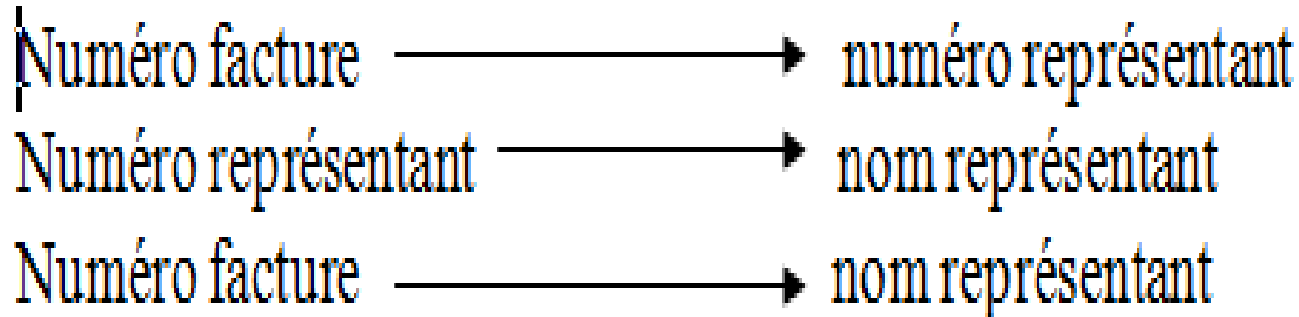
Dépendance fonctionnelle directe



Une dépendance fonctionnelle $R1 \longrightarrow R2$ est directe s'il n'existe pas une $R3$ (ou une collection de rubriques) qui engendrerait une dépendance fonctionnelle transitive de telle sorte que l'on pourrait écrire :



Exemple

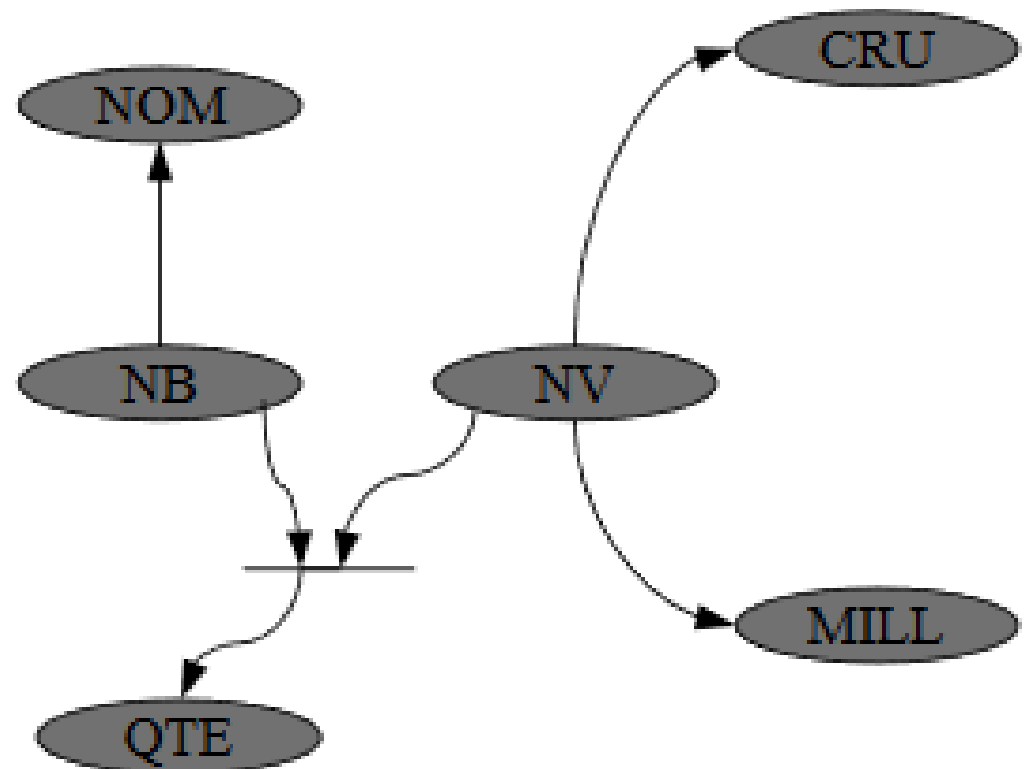


Graphe de DF

Graphe de dépendances fonctionnelles élémentaires

– Exemple

- $NV \rightarrow CRU$
- $NV \rightarrow MILL$
- $NB \rightarrow NOM$
- $(NV, NB) \rightarrow QTE$



La 1^{ère} forme normale(1FN)



Définition :

Est en 1FN, une relation (ayant par définition une clé) dont les attributs possèdent tous une valeur **sémantiquement atomique**.

Atomicité : un attribut est dit « atomique » si aucune subdivision de l'information initiale n'apporte une information supplémentaire ou complémentaire.

Normalisation en 1F:

Un attribut non atomique doit être décomposé en ses différents attributs atomique.

La 1^{ère} forme normale(1FN)



Exemple:

ÉTUDIANT (Mat, Nom, Prénoms, DateNaiss)

Normaliser en 1FN.

La 1^{ère} forme normale(1FN)



ÉTUDIANT (Mat, Nom, Prénoms, DateNaiss)

Analyse:

La relation ÉTUDIANT possède une clé primaire.
Mais l'attribut Prénoms n'est pas élémentaire.

Qu'en est-il de l'attribut DateNaiss?

Relativité de la notion d'atomicité

L'atomicité d'un attribut est souvent relative : on peut décider qu'un attribut contenant une date n'est pas atomique (et que le jour, le mois et l'année constituent chacun une valeur), ou bien que l'attribut est de domaine date et donc qu'il est atomique.

Conclusion:

La relation ÉTUDIANT n'est pas en 1FN.

La 1^{ère} forme normale(1FN)



Normalisation:

Un **étudiant** donnée peut avoir un ou plusieurs **prénoms**.

Il faut ajouter une nouvelle relation PRÉNOM.

Cette relation contient les différents prénoms des étudiants.

Cette relation est à lier à la relation ÉTUDIANT par un lien Père (ÉTUDIANT)-Fils (PRÉNOM).

La 1^{ère} forme normale(1FN)



Schéma en intension:

ÉTUDIANT (Mat, Nom, DateNaiss)

PRÉNOM (#Mat, prénom)

Schéma en extension:

ÉTUDIANT		
Mat	Nom	DateNaiss
<u>69</u>	Halimi	14-05-1980
<u>14</u>	Biskri	09-06-1982
<u>101</u>	Freudi	04-11-1962

PRÉNOM	
<u>#Mat</u>	<u>prénoms</u>
<u>14</u>	<u>Omar</u>
<u>69</u>	<u>Mahmoud</u>
<u>69</u>	<u>Fawzi</u>
<u>101</u>	<u>Ahmed</u>
<u>101</u>	<u>Amine</u>

La 1^{ère} forme normale(1FN)

Exemple 1 : La relation LIVRE (N° Livre, Titre, Auteurs) telle que

- *N° Livre -> Titre*
- *N° Livre -> Auteurs*

est-elle en Première Forme Normale ?

<i>LIVRE/</i>	<i>N° Livre</i>	<i>Titre</i>	<i>Auteurs</i>
	1	<i>Astérix le Gaulois</i>	<i>Goscinny, Uderzo</i>
	2	<i>Les diaboliques</i>	<i>Boileau-Narcejac</i>
	3	<i>Tintin en Amérique</i>	<i>Hergé</i>
	4	<i>Paris brûle-t-il ?</i>	<i>Lapierre, Collins</i>

NON, l'attribut Auteurs peut avoir deux valeurs (il pourrait y en avoir plus) ...

*Comment peut-on transformer la relation **LIVRE** pour obtenir une solution en 1^{ère} forme normale ?*

La 1^{ère} forme normale(1FN)

LIVRE (n° Livre, titre)

AUTEUR (n°auteur, nom auteur)

ECRIRE (n° livre#, n°auteur#)

LIVRE	N° Livre	Titre
	1	Astérix le Gaulois
	2	Les diaboliques
	3	Tintin en Amérique
	4	Paris brûle-t-il ?
AUTEUR	N°auteur	Nom auteur
	1	Goscinnny
	2	Uderzo
	3	Boileau-Narcejac
	4	Hergé
	5	Lapierre
	6	Collins
ECRIRE	N° Livre	N°auteur
	1	1
	1	2
	2	3
	3	4
	4	5
	4	6

La 2^{ème} forme normale(2FN)



- **Définition :**

Une relation est en 2FN ssi :

- ✦ Elle est en 1FN ;
- ✦ Chacun des attributs ne faisant pas partie de la clé primaire est en Dépendance Fonctionnelle élémentaire avec la clé primaire (toute entière et pas une partie).

Conséquence :

Une relation en 1FN avec un seul attribut comme clé primaire est *automatiquement* en 2FN.

Le problème se pose uniquement pour les tables qui ont une clé primaire composée.

Rappel :

Un attribut est en dépendance fonctionnelle élémentaire avec la clé primaire lorsqu'il n'est en dépendance fonctionnelle avec aucune partie de la clé.

La 2^{ème} forme normale(2FN)



CLIENT (NumCli, Nom, DateNaiss)

Analyse:

Elle est en 1FN.

Sa clé est simple.

Conclusion:

La relation est en 2FN.

La 2^{ème} forme normale(2FN)

COMMANDE_PRODUIT (NumProd, NumFour, Quantité, VilleFour)

Analyse:

Elle est en 1FN (existence d'une clé, tous les attributs sont atomiques).

$\text{NumProd, NumFour} \rightarrow \text{Quantité}$

La Quantité ne peut être connue que si on connaît le NumProd et le NumFour, donc toute la clé primaire.

mais

$\text{NumFour} \rightarrow \text{VilleFour}$

Donc, le NumFour, qui est une partie de la clé primaire, est suffisant pour connaître la ville du fournisseur: il joue le rôle de la clé.

Donc: la DF

$\text{NumProd, NumFour} \rightarrow \text{VilleFour}$
n'est pas élémentaire.

Conclusion:

La relation n'est pas en 2FN.

La 2^{ème} forme normale(2FN)



Normalisation:

Méthode: Décomposer la relation initiale en plusieurs nouvelles relations (leur nombre est lié au nombre des DFs non élémentaires).

Astuce: Pour éviter toute confusion, renommer la relation initiale X.

COMMANDE_PRODUIIT (NumProd, NumFour, Quantité, VilleFour)

Écriture utilisant les DFs:

COMMANDE_PRODUIIT (NumProd, NumFour → Quantité; NumProd, NumFour → VilleFour; NumFour → VilleFour)

Renomons cette relation X comme suit:

X (NumProd, NumFour → Quantité; NumProd, NumFour → VilleFour; NumFour → VilleFour)

Il faut éliminer la DF non élémentaire.

Nous obtenons les deux nouvelles relations suivantes:

X1 (NumProd, NumFour → Quantité)

X2 (NumFour → VilleFour)

La 2^{ème} forme normale(2FN)



Soit :

X1 (**NumProd**, **NumFour**, Quantité)

X2 (**NumFour**, VilleFour)

Maintenant, nous pouvons renommer les deux relations en fonction de la nature de la base de données: la première relation concerne les données sur les commandes (les fournisseurs qui ont émis ces commandes et les quantités commandées) et la deuxième les données sur les fournisseurs.

COMMANDE (**NumProd**, **NumFour**, Quantité)

FOURNISSEUR (**NumFour**, VilleFour)

Les deux relations COMMANDE et FOURNISSEUR sont en 2FN.

La 2^{ème} forme normale(2FN)

*Exemple 2 : La relation **ELEVE** (N° Elève, N° Matière, N° Classe, Moyenne) est-elle en deuxième forme normale ?*

Tout d'abord est-elle en 1FN ? OUI

En 2FN ?

NON car les DF sont les suivantes :

N° Elève, N° Matière -> moyenne : DF élémentaire.

N° Elève -> N° classe : donc N°Elève, n° matière → n° classe est une DF non élémentaire.

Il y a un attribut en Dépendance fonctionnelle avec une partie de la clé primaire.

Pour le rendre en 2FN on crée une relation supplémentaire telle que

***ELEVE** (N° Elève, N° Classe)*

***RESULTAT** (N° Elève, N° Matière, moyenne)*

La 3^{ème} forme normale(3FN)



- **Définition :**

Une relation est en 3FN ssi :

- ✦ Elle est en 2FN ;
 - ✦ Chacun des attributs ne faisant pas partie de la clé primaire est en Dépendance Fonctionnelle élémentaire et directe avec la clé primaire et uniquement avec elle (autrement dit: (i) il n'existe aucune DF entre deux attributs non clé primaire et (ii) chaque attribut non clé doit obligatoirement dépendre fonctionnellement de la clé primaire).
- Rappel : un attribut est en dépendance fonctionnelle directe avec la clé primaire lorsque cette dépendance fonctionnelle n'est pas obtenue par transitivité.

La 3^{ème} forme normale(3FN)



La relation :

COMPAGNIE (Vol, Avion, Pilote)

avec les DF :

Vol \rightarrow Avion

Avion \rightarrow Pilote

Vol \rightarrow Pilote

est en 2FN mais pas en 3FN.

La 3^{ème} forme normale(3FN)



Anomalies de mise à jour sur la relation COMPAGNIE : il n'est pas possible d'introduire un nouvel avion sur un nouveau vol sans préciser le pilote correspondant.

La décomposition suivante donne deux relations en 3FN qui permettent de retrouver (par transitivité) toutes les DF :

R1 (**Vol**, #Avion) ;

R2 (**Avion**, Pilote).

La 3^{ème} forme normale(3FN)



*Exemple 3 : La relation **ELEVE** (N° Elève, nom élève, N° Classe, nom classe) est-elle en deuxième forme normale ?*

Tout d'abord est-elle en 1FN ? OUI

En 2FN ?

OUI car la clé primaire est une propriété simple.

En 3FN ?

NON, car N° élève \rightarrow nom classe n'est pas une dépendance fonctionnelle directe.

En effet : N° élève \rightarrow N° classe et N° classe \rightarrow nom classe

|

Pour la rendre en 3FN on crée une relation supplémentaire telle que

***ELEVE** (N° Elève, nom élève, N° Classe#)*

***CLASSE** (N° Classe, nom classe)*

Résumé



- **Modèle relationnel normalisé = relations avec**
 - une clé, qui permet de distinguer chaque occurrence
 - des attributs élémentaires (1FN)
 - en dépendance de TOUTE la clé (2FN),
 - et RIEN QUE de la clé (3FN)

Illustration : maximiser l'espace de stockage (1)

Prenons l'exemple 2: *ELEVE* (*N° Elève*, *N° Matière*, *N° Classe*, *moyenne*)

Prenons les hypothèses suivantes :

Il existe 2000 élèves au sein d'un lycée. Chaque élève travaille au plus sur 10 matières. Nous sommes d'accord que le nombre de tuples de la table élève est de 20 000 au maximum.

Posons d'autres hypothèses sur la taille de chaque attribut :

N° élève : 5 caractères, N° matière : 2 car, N° classe : 3car, Moyenne : 4 dont 2 décimales.

- La taille d'un tuple est de 14 caractères.
- On peut déduire que le volume total de la table **ELEVE** est de $20\ 000 * 14 = 280\ 000$ caractères.

Illustration : maximiser l'espace de stockage (2)

Prenons à nouveau l'exemple 2 avec la solution normalisée :

ELEVE (N° Elève, N° Classe)

RESULTAT (N° Elève, N° Matière, moyenne)

La table **ELEVE** possède 2000 tuples. La taille de chacun est égale à $5+3=8$ caractères.

Le volume de la table **ELEVE** est égal à $2000 * 8 = 16\ 000$ caractères.

La table **RESULTAT** possède 20 000 tuples. La taille de chacun est égale à $5+2+4=11$ caractères.

Le volume de la table **RESULTAT** est égal à $20\ 000 * 11 = 220\ 000$ caractères.

Le volume total est égal à $220\ 000 + 16\ 000 = 236\ 000$ caractères $< 280\ 000$ caractères.

On se rend compte que la solution normalisée utilise moins d'espace disque que la solution non normalisée.

En fait dans cette solution, le n° classe était redondant : répété pour chaque matière, donc il y avait perte de place.