



Université Abdelhamid Mehri, Constantine 2

Faculté NTIC

Département MI

2^{ème} Année MI

Année Universitaire 2016-2017

Introduction aux Bases de Données

F. Magra-Benchikha



Chapitre 4

**Conception de bases de données relationnelles :
La normalisation**

1. Problématique
2. Les dépendances fonctionnelles (DF)
 - 2.1. Définition
 - 2.2. Types de dépendance fonctionnelles
 - DF élémentaire
 - DF directe
 - 2.3. Propriétés des dépendances fonctionnelles
 - 2.4. Fermeture transitive et couverture minimale
 - 2.5. Graphe de dépendances fonctionnelles
 - 2.6. Clé minimale de relation
 - 2.7. Application
3. La normalisation
 - 3.1. Degrés de normalisation
 - 3.2. Les formes normales (1FN, 2FN, 3FN)
 - 3.3. Schéma relationnel normalisé
 - 3.4. Décomposition de relation (Algorithme de synthèse)
 - 3.5. Application1
 - 3.6. Application2
4. Application de synthèse

Problématique

On désire concevoir une base de données relationnelle gérant le stock des produits dans des dépôts.

- On a le dictionnaire de données et les règles de gestion suivantes :

Information	Codification
Code du produit	code_prod
Code du dépôt	code_dep
Prix du produit	prix
Nom produit	nom_prod
Adresse du dépôt	adr_dep
Quantité en stock du produit	qté

Règles de gestion :

- Un produit peut être stocké dans plusieurs dépôts
- Dans un dépôt sont stockés plusieurs produits
- Un produit est stocké dans un dépôt avec une certaine quantité

On considère les deux propositions de conception suivantes :

Proposition 1 : La base de données est représentée par une seule table (relation) contenant tous les attributs. On appelle cette table la **relation universelle** (RU).

RU (code_prod, nom_prod, prix, code_dep, adr_dep, qté)

Soit l'extension suivante de RU :

code_prod	nom_prod	prix	code_dep	adr_dep	qté
100	Crayon	20	D1	Adr1	12
100	Crayon	20	D2	Adr2	54
200	Stylo	30	D1	Adr1	76
100	Crayon	20	D3	Adr3	78
200	Slylo	30	D3	Adr3	45

Problèmes : il y'a de la redondance d'information qui engendre les anomalies suivantes :

1. **Anomalie de mise à jour :** si on change le prix d'un produit, on doit le changer dans toutes les lignes de son stockage dans les différents dépôts,
2. **Anomalie d'insertion :** on ne peut pas ajouter un nouveau dépôt (code et adr) vide (sans produits stockés),
3. **Anomalie de suppression :** si on supprime le dernier produit d'un dépôt D1, on perd toute information (code et adr) concernant ce dépôt.

Proposition 2 : La base de données est représentée par les trois tables suivantes :

Produit (code_prod, nom_prod, prix)

Dépôt (code_dep, adr_dep)

Stock (code_prod, code_dep, qté)

Soit l'extension suivante de la BD :

code_prod	nom_prod	Prix
100	Crayon	20
200	Slylo	30

code_dep	adr_dep
D1	Adr1
D2	Adr2
D3	Adr3

code_prod	code_dep	qté
100	D1	12
100	D2	54
200	D1	76
100	D3	78
200	D3	45

Cette base de données ne présente aucune anomalie ➡ elle est **normalisée**

La normalisation

- La normalisation est le processus qui permet de construire des relations complètes et non redondantes.
- Cette théorie est basée sur les "dépendances fonctionnelles" (DF) qui **traduisent des contraintes sur les données**.
- Les dépendances fonctionnelles et des propriétés particulières, sont à la base d'une suite de formes normales (FN).
- Une relation est dite normalisée si, au moins, elle vérifié la troisième forme normale (3FN).
- La troisième forme normale est obtenue au moyen d'algorithmes de décomposition. Le point de départ de ces algorithmes est la relation universelle, **c'est-à-dire la relation qui regroupe toutes les informations à stocker (dans notre exemple, le schéma de la première proposition représente cette relation universelle) .**
- le but est d'obtenir, en sortie, une représentation des données (3^{ème} forme normale) présentant un minimum de redondance à l'intérieur de chaque relation et un maximum d'indépendance entre les différentes relations .

Notion de dépendance fonctionnelle

Définition

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation, X et Y des sous-ensembles de (A_1, A_2, \dots, A_n) . Il existe une **dépendance fonctionnelle (DF) de X vers Y** , notée $X \rightarrow Y$, si et seulement si : étant donné une valeur de X , il lui correspond une et une seule valeur de Y . On dit que X détermine Y ou Y dépend fonctionnellement de X .

Exemple :

code_prod \rightarrow prix,
code_prod \rightarrow nom-prod
code_dep \rightarrow adr_dep

Remarque : la **partie gauche** d'une dépendance fonctionnelle peut être formée de la concaténation de plusieurs attributs. Cette concaténation est indissociable.

Exemple : code_prod, code_dep \rightarrow qté (code_prod ~~\rightarrow~~ qté **et** code_dep ~~\rightarrow~~ qté)

Types de dépendances fonctionnelles

1. Dépendance fonctionnelle élémentaire

Une dépendance fonctionnelle $X \rightarrow Y$ est dite **élémentaire** si :

- Y est un attribut unique n'appartenant pas à X,
- il n'existe pas W inclus dans X tel que $W \rightarrow Y$ (en d'autres termes, Y ne dépend pas fonctionnellement d'une partie de X)

Exemple :

code_prod, nom_prod \rightarrow prix (DF non élémentaire car on a: code_prod \rightarrow prix)

code_prod, code_dep \rightarrow qté (DF élémentaire)

code_dep \rightarrow adr_dep (DF élémentaire)

2. Dépendance fonctionnelle directe

Une dépendance fonctionnelle $X \rightarrow Y$ est dite **directe** si :

- il n'existe aucun attribut Z tel que l'on puisse avoir $X \rightarrow Z$ et $Z \rightarrow Y$ (En d'autres termes, X ne détermine pas Y de façon transitive)

Exemple :

Code_etudinat \rightarrow id_salle **n'est directe car on a :**

Code_etudiant \rightarrow code_enseignant

Code_enseignant \rightarrow id_salle

Propriétés des dépendances fonctionnelles (Axiomes d'Armstrong)

Les dépendances fonctionnelles obéissent à certaines propriétés connues sous le nom d'axiomes d'Armstrong.

- **Réflexivité** :

$$Y \subseteq X$$

ALORS $X \rightarrow Y$

- Augmentation :

$$X \rightarrow Y$$

ALORS $X, Z \rightarrow Y, Z$

- Transitivité :

$$X \rightarrow Y \text{ et } Y \rightarrow Z$$

ALORS $X \rightarrow Z$

- D'autres propriétés se déduisent de ces axiomes :

- Union :

$$X \rightarrow Y \text{ et } X \rightarrow Z$$

ALORS $X \rightarrow Y, Z$

- Pseudo-transitivité :

$$X \rightarrow Y \text{ et } Y, W \rightarrow Z$$

ALORS $X, W \rightarrow Z$

- Décomposition :

$$X \rightarrow Y \text{ et } Z \subset Y$$

ALORS $X \rightarrow Z$

Fermeture transitive et couverture minimale d'un ensemble de DFs

1. Fermeture transitive

- La **fermeture transitive** d'un ensemble de dépendances fonctionnelles est ce même ensemble enrichi de toutes les dépendances fonctionnelles déduites par transitivité.

2. Couverture minimale

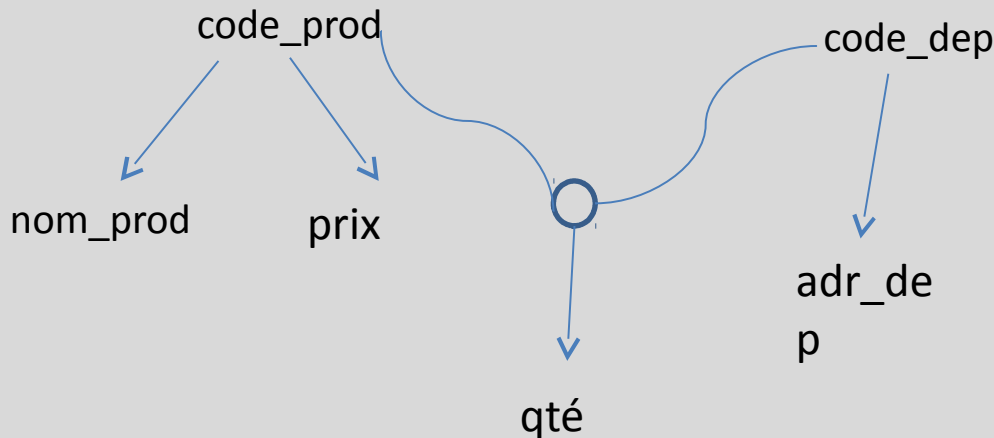
- On appelle la **couverture minimale** d'un ensemble de dépendances fonctionnelles F , un sous ensemble des dépendances fonctionnelles de F qui sont élémentaires et non déduites. La couverture minimale F' est équivalente à F en ce sens que toute dépendance fonctionnelle de F peut être déduite de F' .

Grphe de dépendances fonctionnelles (GDF)

Le **graphe de dépendances fonctionnelles (GDF)** est la représentation graphique d'un ensemble de DFs.

Grphe minimal des dépendances fonctionnelles

Le GDF est dit **minimal** s'il représente une couverture minimale d'un ensemble de dépendances fonctionnelles.



Exemple de GDF

Clé minimale de relation

Une **clé minimale X d'une relation R** est un attribut (ou groupe d'attributs) de R dont :

- X est le plus petit ensemble d'attributs qui identifie R,
- X détermine tous les autres attributs de R.

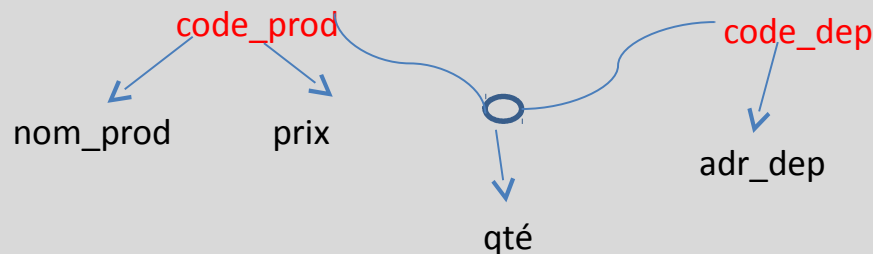
Remarque

Tout graphe (minimum ou pas) des DFs peut être employé pour la recherche des identifiants (clés) des relations de la façon suivante:

Chercher, sur le graphe des DFs de la relation, tout ensemble minimum d'attributs, X, tel que tous les chemins partant de X et suivant les DF atteignent tous les autres attributs du graphe. Alors X est un identifiant de la relation.

Exemple :

La clé primaire de la relation Stock est : (code-prod + code-dep)



Application 1

Soit la relation R (A , B , C , D , E , F , G , H) avec la suite de dépendances fonctionnelles suivantes :

DF = { $A \rightarrow B$,
 $A \rightarrow C$,
 $D \rightarrow E$,
 $F \rightarrow A$,
 $F \rightarrow C$,
 $F \rightarrow G$,
 $AD \rightarrow C$,
 $AD \rightarrow H$ }

1. Démontrer en utilisant les axiomes d'Amstrong et les règles d'inférence la validité de la dépendance fonctionnelle suivante : $FD \rightarrow CE$
2. Proposer une couverture minimale pour DF.
3. Trouver une clé minimale de R en considérant la couverture minimale de DF.

Application 1 (Corrigé)

1. Démontrer en utilisant les axiomes d'Amstrong et les règles d'inférence la dépendance fonctionnelle suivante : $FD \rightarrow CE$
 - a. $F \rightarrow C$ (Donnée)
 - b. $FD \rightarrow CD$ (Augmentation de (a) par D)
 - c. $D \rightarrow E$ (Donnée)
 - d. $CD \rightarrow CE$ (Augmentation de (c) par C)
 - e. $FD \rightarrow CE$ (Transitivité de (b) et (d))

2. Proposer une couverture minimale pour DF.

DF = { $A \rightarrow B$,
 $A \rightarrow C$,
 $D \rightarrow E$,
 $F \rightarrow A$,
 ~~$F \rightarrow C$~~ , (non directe ou transitive)
 $F \rightarrow G$,
 ~~$AD \rightarrow C$~~ , (non élémentaire)
 $AD \rightarrow H$ }

Couverture minimale est donc :

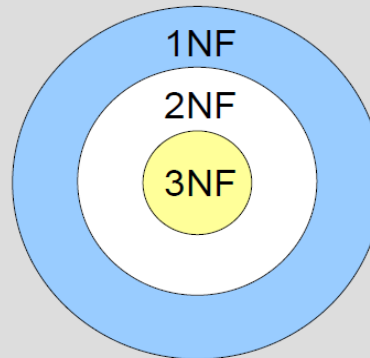
DF' = { $A \rightarrow B$,
 $A \rightarrow C$,
 $D \rightarrow E$,
 $F \rightarrow A$,
 $F \rightarrow G$,
 $AD \rightarrow H$ }

3. Trouver une clé minimale de R

La clé minimale de R est : **(F + D)**

Degré de normalisation

- Il existe plusieurs degrés de normalisation, de la 1^{ère} à la 5^{ème} forme normale:
 - 1NF - première forme normale
 - 2NF – deuxième forme normale
 - **3NF – troisième forme normale (« minimum »)**
 - BCNF – forme normale de Boyce-Codd
 - 4NF – quatrième forme normale
 - 5NF – cinquième forme normale



1^{ère} Forme Normale (1FN)

Une relation R est dite en 1FN si et seulement si :

- Tous les attributs de R sont **atomiques** et **monovalués**.

Exemple1:

Dans la relation suivante, tout attribut est atomique et monovalué.

Produit (codeProd, nomProd, prix, couleur)

Produit est en 1FN

Exemple 2

Pour l'exemple précédent, si on suppose qu'un produit peut être de plusieurs couleurs. La relation Produit n'est plus en 1FN car l'attribut "couleur" est dans ce cas multivalué.

Solution : Décomposition de la relation Produit en :

Produit (code_prod, nom_prod, prix)

Couleur-Produit (code_prod, couleur)

2^{ème} Forme Normale (2FN)

Définition 1

Une relation R est en 2FN si et seulement si :

- R est en 1FN;
- tout attribut n'appartenant pas à une clé de R (attribut non clé) **ne dépend pas** d'une **partie** de cette clé.

Définition 2

On dit qu'une relation R est en 2FN si et seulement si :

- R est en 1FN,
- toutes les DFs entre la clé et les autres attributs sont **élémentaires**.

Exemple

La relation : **RU** (code_prod, nom_prod, prix, code_dep, adr_dep, qté)

N'est pas en 2FN car on a : **code_prod** → **nom_prod**

3^{ème} Forme Normale (3FN)

Définition 1 :

Une relation R est en 3FN ssi :

- R est en 2FN;
- tout attribut **non clé ne dépend pas** d'un attribut **non clé**.

Définition 2 :

On dit qu'une relation est en 3FN si toutes les DFs entre la clé et les autres attributs sont **élémentaires** et **directes**.

Exemple

Etudiant (Num-E, NomE, Dept, NomDept)

DF = {Matr \rightarrow NomE, Dept
Dept \rightarrow NomD }

La relation Etudiant n'est pas en 3FN car on : Dept \rightarrow NomD

Schéma relationnel normalisé

Définition

Un schéma relationnel est dit **normalisé** si toutes les relations qui le composent sont, au moins, dans la troisième forme normale (3FN).

Normalisation

Normaliser un schéma relationnel c'est le transformer en un schéma relationnel normalisé équivalent, c'est-à-dire, contenant les mêmes informations que le schéma de départ et ne présentant pas de redondance. La normalisation se base principalement sur la **décomposition** des relations non normalisées en plusieurs relations normalisées.

Décomposition

On dit que la relation R est **décomposable** en relations R1, R2, ..., Rn **sans perte d'informations et sans perte de DFs** si :

$$R = R1 \bowtie R2 \bowtie \dots \bowtie Rn$$

Important : Pour toute relation non normalisée, il existe au moins une décomposition en 3FN sans perte d'information et sans perte de DFs. Une telle décomposition est générée par un algorithme dit **algorithme de synthèse**.

Algorithme de synthèse

Entrée : La relation R non normalisée et une couverture minimale de l'ensemble des DFs.

Sortie : Des relations en 3FN.

A partir du GDF minimal (ou de la couverture minimale) :

1. Éditer l'ensemble des attributs isolés dans une même relation,
2. Rechercher le plus grand ensemble X d'attributs qui déterminent d'autres attributs, A_1, A_2, \dots, A_n ,
3. Éditer la relation R (X, A_1, A_2, \dots, A_n) avec X sa clé primaire,
4. Supprimer les DFs ($X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$) du GDF minimal (ou de C),
5. Supprimer les attributs isolés du GDF,
6. Répéter l'opération de réduction à partir de l'étape 2, jusqu'à ce que le GDF (ou C) soit vide.

Fin du Chapitre 4

Application 2

Soit $R(A, B, C, D, E, F)$ une relation avec l'ensemble de dépendances fonctionnelles suivant:
 $DF = \{ AB \rightarrow D, B \rightarrow C, D \rightarrow E, D \rightarrow F \}$

1. Quelle est la clé minimale de R ?
2. Quelle est la forme normale de R ?
3. On décompose la relation R en R_1 et R_2 : $R_1(A, B, D, E, F)$ et $R_2(B, C)$. Quelles sont les formes normales des relations R_1 et R_2 ?
4. Normaliser R_1 et R_2 si elles ne le sont pas déjà.
5. Démontrer qu'il n'y a pas de perte de données dans le schéma relationnel normalisé.

Application 2 (Corrigé)

Soit $R(A, B, C, D, E, F)$ une relation avec l'ensemble de dépendances suivant:
 $DF = \{ AB \rightarrow D, B \rightarrow C, D \rightarrow E, D \rightarrow F \}$

1. La clé minimale de R est $(A + B)$.
2. R est en 1FN ,
 R n'est pas en 2FN ($B \rightarrow C$). **Donc la forme normale de R est 1FN.**
3. - $R1(\underline{A}, \underline{B}, D, E, F)$, $DF1 = \{ AB \rightarrow D, D \rightarrow E, D \rightarrow F \}$.
 $R1$ est en 2FN
- $R2(B, C)$, $DF2 = \{ B \rightarrow C \}$.
 $R2$ est en 3FN (**$R2$ est normalisée**)
4. On normalise $R1$ en la décomposant en deux relations :
 $R11(\underline{A}, \underline{B}, D)$ $DF11 = \{ AB \rightarrow D \}$.
 $R12(\underline{D}, E, F)$ $DF12 = \{ D \rightarrow E, D \rightarrow F \}$.
5. Il n'y a pas de perte de données dans le schéma relationnel normalisé car on :
 $R = R12 [R12.D = R11.D] R11 [R11.B = R2.B] R2$
(R est retrouvée par la jointure des trois relations $R11$, $R12$ et $R2$)

$R(\underline{A}, \underline{B}, C, D, E, F)$
est normalisée par
décomposition en :
 $R11(\underline{A}, \underline{B}, D)$
 $R12(\underline{D}, E, F)$
 $R2(\underline{B}, C)$

Application 3

Soit la relation : Réparation (n-client , panne , n-réparateur)

on suppose que :

a/ chaque réparateur est spécialiste dans une seule panne

b/ pour chaque panne, chaque client se rend chez un seul réparateur.

1. Traduire les règles ci-dessus en termes de dépendances fonctionnelles.
2. Quelles est la clé minimale de la relation Réparation ?
3. Vérifier sur la relation les formes normales : 1FN, 2FN , 3FN.
4. On suppose maintenant qu'un réparateur peut réparer plusieurs pannes :
 - Vérifier les différentes formes normales,
 - Normaliser la relation.

Application de synthèse

Pour réaliser un livre d'exercices sur les bases de données, un groupe d'auteurs a décidé de classer les exercices proposés en plusieurs types et que chaque exercice a un niveau de difficulté qui permet de donner une estimation sur la durée de résolution suivant le type de l'exercice. L'élaboration du modèle conceptuel du livre a permis d'avoir les attributs suivants:

- numéro-exercice
- type-exercice
- libellé-type-exercice
- niveau-difficulté
- libellé-niveau-difficulté
- nom-auteur (identifiant de l'auteur)
- carrière-auteur
- durée-résolution-estimée : l'estimation est faite par type d'exercice et par niveau de difficulté.
- énoncé-exercice
- part-auteur : un exercice peut être réalisé par plusieurs auteurs et on enregistre ainsi la part de participation de chaque auteur dans un exercice donnée.

Questions

- Trouver la base de données relationnelle normalisée qui gère cette bibliothèque.
- Écrire en SQL les requêtes suivantes :

1. Quels sont les noms des auteurs n'ayant pas participé aux exercices d'application (type-exercice= 'Application'). La liste doit être triée par ordre alphabétique.
2. Quels sont les noms des auteurs dont le taux de participation dans chaque exercice dépasse 25%.
3. Quels sont les exercices dont les auteurs inclus au moins Gardarin et Boudjlida.
4. Quels sont les exercices (numéro-exercice) les plus difficiles et avec une durée de résolution maximale.
5. Quels sont les auteurs qui ont participé dans plus d'exercices d'application que Gardarin.
6. Quels sont les exercices faits par un seul auteur.