Institut Supérieur d'Informatique et de Gestion de Kairouan

Bases de Données Relationnelles

Cours 4 : Normalisation d'une base de données relationnelle

Riadh ZAAFRANI

2ème année Licence Fondamentale : Sciences de l'informatique

Octobre 2009

Qu'est-ce qu'une BD relationnelle 'correcte' ?

Un ensemble de relations tel que:

- chaque relation décrit un fait élémentaire avec les seuls attributs qui lui sont directement liés
- il n'y a pas de redondance d'information, génératrices de problèmes lors des MAJ
- il n'y a pas de perte d 'information
  - Personne (nom, prénom), Adresse (no, rue, ville) Qui habite où ? Impossible de répondre !

Qu'est-ce qu'une BD relationnelle 'incorrecte' ?

Une relation n'est pas correcte si:

- elle implique des répétitions au niveau de sa population
- elle pose des problèmes lors des MAJ insertions / modifications / suppressions
- Les conditions pour qu'une relation soit correcte peuvent être définies formellement:

=> règles de normalisation

Normalisation

- Processus de transformation d'une relation posant des problèmes lors des MAJ en relations ne posant pas de problèmes
- On mesure la qualité d'une relation par son degré de normalisation
- 1NF, 2NF, 3NF, BCNF, etc.

Formes normales: 1FN

- Une relation est en **1FN** si chaque valeur de chaque attribut de chaque tuple est une valeur simple (tous les attributs sont simples et monovalués).
- Exemple:

  Livraison (N°fourn, adr.f, N°prod, prix.p, qté)

Exemple de mauvaise relation en 1FN

Livraison (N°fourn, adr.f, N°prod, prix-p, qté)

3	Kairouan	52	65	10
22	Sousse	10	15	5
22	Sousse	25	10	12
3	Kairouan	25	10	5
3	Kairouan	10	15	20

- L'adresse du fournisseur ne dépend pas du produit.
- Le prix du produit ne dépend pas du fournisseur

  ⇒ **REDONDANCES**
- Anomalies de mise à jour

Exemple de mauvaise relation en 1FN

Livraison (N°fourn, adr.f, N°prod, prix-p, qté)

3	Kairouan	52	65	10
22	Sousse	10	15	5
22	Sousse	25	10	12
3	Kairouan	25	10	5
3	Kairouan	10	15	20

- Si un fournisseur change d'adresse et qu'un seul tuple (une seule ligne) est mis à jour:
  ⇒ **incohérence**
- Si une nouvelle ligne est insérée pour un fournisseur connu, mais avec une adresse différente:
  ⇒ **incohérence**

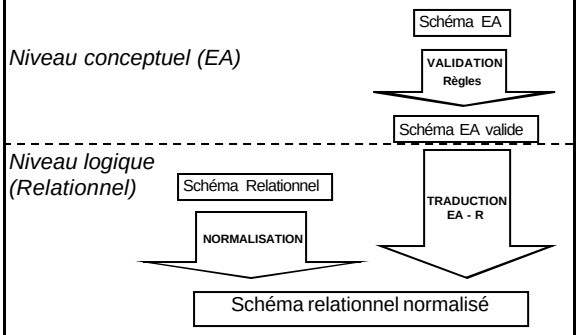
Exemple de mauvaise relation en 1FN

Livraison (N°fourn, adr.f, N°prod, prix-p, qté)

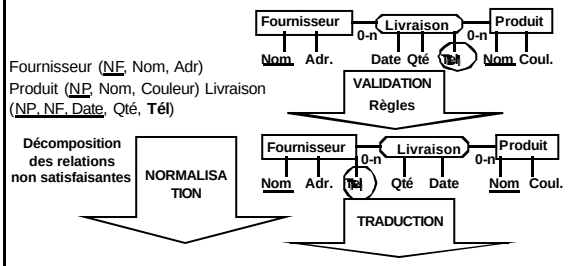
3	Tunis	52	65	10
~~22~~	~~Sousse~~	~~10~~	~~15~~	~~5~~
~~22~~	~~Sousse~~	~~25~~	~~10~~	~~12~~
3	Kairouan	25	10	5
3	Kairouan	10	15	20

- Si un fournisseur change d'adresse et qu'un seul tuple (une seule ligne) est mis à jour:
  ⇒ **incohérence**
- Si une nouvelle ligne est insérée pour un fournisseur connu, mais avec une adresse différente:
  ⇒ **incohérence**

Normalisation ou traduction EA



Exemple



Produit (NP, Nom, Couleur)
Fournisseur (NF, Nom, Adr, **Tél**)
Livraison (NP, NF, Date, Qté)

Formalisation du problème

- L'adresse d'un fournisseur ne dépend que du fournisseur, …..
  ⇒ DÉPENDANCE FONCTIONNELLE (DF)
- soit une table T (x, y, z)
- il existe une DF: x → y si et seulement si dans T à une même valeur de x correspond toujours une même valeur de y

Formalisation

- T :

X	Y	Z
x1	y1	z1
.....		
x1	y1	z2
.....		

- X → Y : X détermine Y Y dépend de X
- X: source de la DF, Y: cible de la DF
- la source peut être un ensemble d'attributs:
  (nom, prénom) ⟶ adresse

Propriétés des DF (Armstrong's Axioms)

- Transitivité:
  si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$
  - (DF déduite)
- Augmentation:
  si $X \rightarrow Y$ alors $(A, X) \rightarrow Y$ quelque soit A
  - (DF non élémentaire)
- Reflexivité:
  si $Y \subseteq X$ alors $X \rightarrow Y$
  - (DF triviale)
- On ne s'intéresse qu'aux DF élémentaires non déduites et non triviale.

Graphe des DF

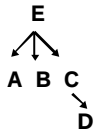
- Pour chaque table il faut connaître les DF intéressantes. Il est facile alors de les représenter sous forme de graphe:

  graphe minimum des DF (orienté)

- Exemple: T (A, B, C, D, E)

  $E \rightarrow A$ $E \rightarrow B$ $E \rightarrow C$ *(° $E \rightarrow A, B, C$)*

  $C \rightarrow D$

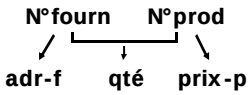


Exemple de graphe des DF

Livraison (N°fourn, adr-f, N°prod, prix-p, qté)

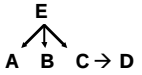
- N°fourn → adr-f
  - l'adresse d'un fournisseur ne dépend que du fournisseur
- N°prod → prix-p
  - le prix d'un produit ne dépend que du produit
- (N°fourn, N°prod) → qté
  - la quantité livrée dépend du produit **et** du fournisseur

[faux: *N°fourn → qté , N°prod → qté*]



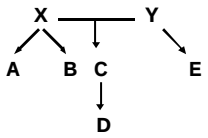
DFs et identifiants

- Le graphe minimum des DF permet de trouver les identifiants de la table
- L'identifiant d'une table est l'ensemble (minimal) des nœuds du graphe minimum à partir desquels on peut atteindre tous les autres nœuds (via les DF)
- Exemple: T1 (A, B, C, D, E)



DFs et identifiants

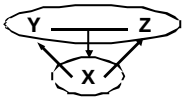
- Autre exemple: T2 (A, B, C, D, E, X, Y)



T2 (A, B, C, D, E, <u>X, Y</u>)

DFs et identifiants

- Autre exemple: T3 (X, Y, Z)
- DF: $X \rightarrow YZ$, $YZ \rightarrow X$



Normalisation

- Que faire si une table n'est pas « normalisée » ?

  ⇒ DECOMPOSITION

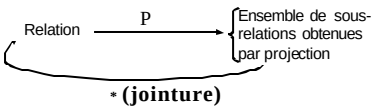
- La table doit être remplacée par un ensemble de tables (plus petites: moins d'attributs)

Décomposition d'une relation

- Soit une relation non satisfaisante, trouver un ensemble de relations satisfaisantes qui décrive les mêmes informations
- La méthode consiste à décomposer la relation en deux (ou plusieurs) relations.
- Pour cela, il est nécessaire de disposer de deux opérations qui permettent, l'une, de découper une relation en sous-relations **(la projection)**, et l'autre, de recomposer la relation à partir de ses sous-relations **(la jointure)**.

Décomposition d'une relation

- Pour vérifier qu'une relation est décomposable sans perte d'information:



- Nombreux algorithmes de décomposition

  Mais attention: les relations sont normalisées, mais peuvent être sémantiquement non significatives.

Projection

- Opération unaire
- la syntaxe : $\pi_{A1,A2,...An}(R)$ (avec *A1, ... An* des attributs de R)
- la sémantique : tous les n-uplets de R, mais avec seulement les attributs *A1, ... An*
- le schéma : schéma($\pi_{A1,A2,...An}(R)$) = A1, ... An
- d'éventuelles remarques : $|\pi_{A1,A2,...An}(R)| <= |R|$
- un exemple :

S

B	C	D
b	c	d
a	a	b
a	a	c

$\pi_{B,D}(S)$

B	D
b	d
a	b
a	c

Jointure

- Opération binaire
- la syntaxe : R ⋈ S
- la sémantique :
  Si schéma(R) ∩ schéma(S) ≠ ∅ = A1, ..., An,
  Si t1∈R et t2∈S sont égaux sur A1, .. An, on les « recolle » (avec t1 et t2 des n-uplets)
  Si schéma(R) ∩ schéma(S) = ∅, c'est le produit cartésien
- le schéma : schéma(R ⋈ S) = schéma(R) ∪ schéma(S)

Exemple

R

A	B
a	b
a	c
b	b

S

B	C	D
b	c	d
a	a	b
a	a	c

R ⋈ S

A	B	C	D
a	b	c	d
b	b	c	d

Méthode formelle de décomposition

- Théorème de Heath

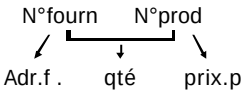
  T (x, y, z) est décomposable sans perte d'information en

  T1 (x, y) et T2 (x, z)

  si x → y

Application de Heath

Livraison (N°fourn, adr.f, N°prod, prix.p, qté)



- N°fourn → adr.f. ➔ (N°fourn, adr.f.) **ok**
  (N°fourn, N°prod, prix.p., qté)
- N°prod → prix.p ➔ (N°prod, prix.p) **ok**
  (N°prod, N°fourn, qté)
- N°prod, N° fourn → qté **ok**

Qualité d'une décomposition

- Une « bonne » décomposition est une décomposition
  1) sans perte d'information
  2) sans perte de DF
- Sans perte de DF:
  toute DF doit être dans l'une des tables obtenues par décomposition

Sans perte d'information

- La « recomposition » de la table à partir des « morceaux » doit redonner la table initiale
- soit T (x, y, z) décomposée en T1 (x, y) T2 (x, z)
  tel que: T1 = PROJECTION [x, y] T
  T2 = PROJECTION [x, z] T
- la décomposition est sans perte d'informations
  si et seulement si : T = JOINTURE (T1, T2)

Exemple: bonne décomposition

- T (NomEmp, adresse, poste, age)

Karim	Kairouan	secrétaire	27
Foued	Sousse	secrétaire	32
Yasmine	Tunis	directeur	38

T1 (NomEmp, adresse, poste) T2 (NomEmp, age

Karim	Kairouan	secrétaire
Foued	Sousse	secrétaire
Yasmine	Tunis	directeur

Karim	27
Foued	32
Yasmine	38

- **T = JOINTURE (T1, T2)**

Exemple: mauvaise décomposition

T12 (NomEmp, adresse, poste) T22 (poste, age)

Karim	Kairouan	secrétaire
Foued	Sousse	secrétaire
Yasmine	Tunis	directeur

secrétaire	27
secrétaire	32
directeur	38

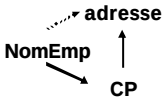
JOINTURE (T12, T22) =

Karim	Kairouan	secrétaire	27
Karim	Kairouan	secrétaire	32
Foued	Sousse	secrétaire	27
Foued	Sousse	secrétaire	32
Yasmine	Tunis	directeur	38



Exemple: perte de DF

- T (NomEmp, adresse, CP)



Karim	Kairouan	3000
Foued	Sousse	2000
Yasmine	Tunis	1000

T1 (NomEmp, CP) T2 (NomEmp, adresse)

Karim	3000
Foued	2000
Yasmine	1000

Karim	Kairouan
Foued	Sousse
Yasmine	Tunis

Exemple: perte de DF

- **JOINTURE (T1, T2) =T**
  - ◆sans perte d 'information
- MAJ de adresse en T2
  - ◆DF CP → adresse
  - ◆⇒ **incohérence entre T1 et T2 si pas de contrôle par application**

Formes normales: 1FN

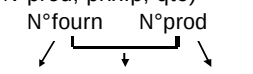
- Une relation est en **1FN** si chaque valeur de chaque attribut de chaque tuple est une valeur simple (tous les attributs sont simples et monovalués).
- Exemple:

  Livraison (N°fourn, adr.f, N°prod, prix.p, qté)

2ème forme normale: 2FN

- Permet d 'éliminer les attributs qui ne décrivent pas l '« objet » traduit par la relation

  Livraison (N°fourn, adr.f, N°prod, prix.p, qté)



- mélange la description:
  - de la livraison (?fourn., ?produit, quantité-livrée)
  - du fournisseur (?fourn., adr.-fourn)
  - du produit (?produit, prix)

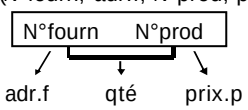
Exemple de mauvaise relation en 1FN

Livraison (N°fourn, adr.f, N°prod, prix-p, qté)

3	Kairouan	52	65	10
22	Sousse	10	15	5
22	Sousse	25	10	12
3	Kairouan	25	10	5
3	Kairouan	10	15	20

2ème forme normale: définition

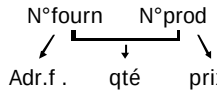
Livraison (N°fourn, adr.f, N°prod, prix.p, qté)



- des DF partent de composants de l 'identifiant: Livraison **n'est pas** en 2FN
- une table est en 2FN si
  - elle est en 1FN, et
  - chaque attribut qui ne fait pas partie de l'identifiant dépend d'un identifiant **entier**

Application de Heath

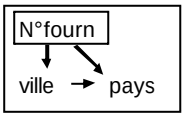
Livraison (N°fourn, adr.f, N°prod, prix.p, qté)



- N°fourn → adr.f. ➔ (N°fourn, adr.f.) **ok**
  (N°fourn, N°prod, prix.p., qté)
- N°prod → prix.p ➔ (N°prod, prix.p) **ok**
  (N°prod, N°fourn, qté)
- N°prod, N° fourn → qté **ok**

3FN: 3ème forme normale

- Permet d'éliminer des sous-relations incluses dans une relation
- Exemple: Fournisseur (N°fourn, ville, pays)



- doit être décomposée en

F (N° fourn, ville)
G (ville, pays)

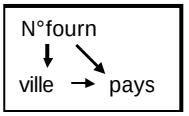
Exemple

- Fournisseur (N°fourn., ville, pays)

3	Kairouan	Tunisie
21	Paris	France
22	Paris	France
4	Sousse	Tunisie
5	Sousse	Tunisie

3ème forme normale: définition

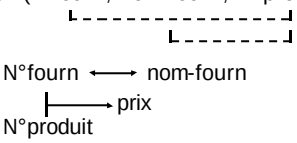
- Fournisseur (N°fourn., ville, pays)



- Profondeur de l 'arbre des DF > 1 :
  pas en 3FN
- 3FN: si T est en 2FN et chaque attribut qui ne fait pas partie de l'identifiant dépend **directement** d'un identifiant entier

3ème forme normale: définition

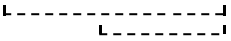
- Plusieurs identifiants:
- 3FN: chaque attribut qui ne fait pas partie de l'un des identifiants dépend **directement** de l'identifiant entier
- Fournisseur (N°fourn, nom-fourn, N°produit, prix)



FNBC Forme normale de Boyce-Codd (BCNF)

- Généralise la 3FN aux relations avec plusieurs identifiants

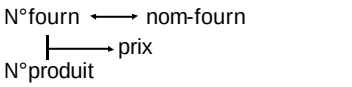
Fournisseur (N°fourn, nom-fourn, N°produit, prix)



- Une table est en FNBC ~~si elle est en 3FN et~~ si toute source complète de DF (élémentaire) est un identifiant entier

FNBC: example Fournisseur

- Fournisseur (N°fourn, nom-fourn, N°produit, prix)

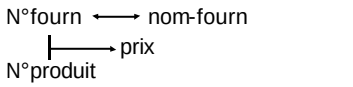


est en 3FN mais pas en FNBC

- le passage en FNBC n'est pas toujours possible sans perte de dépendances

Décomposition Fournisseur

- Fournisseur (N°fourn, nom-fourn, N°produit, prix)

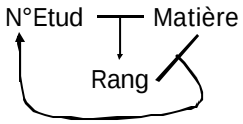


- F1 (N°fourn, nom-fourn)
- F2 (N°fourn, N°produit, prix) ou

  F2' (nom-fourn, N°produit, prix)

FNBC: example Place

- Place (N°Etud, Matière, Rang)

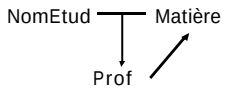


- Identifiants : 1) N°Etud + Matière

  2) Rang + Matière
- Place est en 3FN et est en FNBC

FNBC contre-example

- Enseignement (NomEtud, Matière, Prof)



- Identifiants : 1) NomEtud + Matière

  2) NomEtud + Prof
- Enseignement est en 3FN mais n'est pas en FNBC

Décomposition Enseignement

- Enseignement (NomEtud, Matière, Prof)
- Soit la décomposition:

  T1 (Prof, Matière) T2 (NomEtud, Prof)
- T1 + T2 : sans perte d 'information

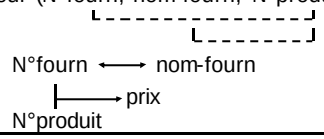
  mais avec perte de la DF:

  (NomEtud, Matière) →Prof
- il faut ajouter la C.I.:

  un étudiant suit une matière donnée avec un seul professeur

3FN: définition plus générale

- Plusieurs identifiants et DF élémentaire A->B
  - Cond1: A est un identifiant
  - Cond2: B fait part d'un identifiant
- FNBC si Cond1
- 3NF (plusiers identifiants) si Cond1 OR Cond2
- Fournisseur (N°fourn, nom-fourn, N°produit, prix)



Conception d'une BD relationnelle

(niveau conceptuel)

- Méthode "classique"
- Objectif: arriver à un ensemble de relations tel que

  - chaque relation décrit un type d' « objet » avec les seuls attributs qui lui sont directement liés

  - les redondances d'information (génératrices de problèmes lors des mises à jour) sont éliminées

Conception d'une BD relationnelle

- Méthode: Décomposition par projections

  sans perte d'information et sans perte de dépendances
- Outils:
  - dépendances fonctionnelles
  - règles de normalisation
- Limites:
  - ne garantit pas que le schéma soit celui qu'il faut
  - ne garantit pas que le schéma soit efficace
- (« overnormalization »)

Conception d'une BD relationnelle

- Méthode "moderne"
- Conception d'un schéma entité-association, puis traduction de celui-ci en schéma relationnel