Dépendances fonctionnelles et normalisation

L2A

Semestre 4

Mehdi Benzine

Introduction

Il est possible de modéliser le réel lié à une problématique métier sous la forme d'une relation universelle.

Une relation universelle est une relation incluant l'ensemble des attributs du domaine étudié.

Pour le domaine "Gestion des affectations des employés".

R(Num_Employé, Num_Projet, Début_Affect, Fin_Affect, Nom, Prénom, Date_Naissance, Fonction, Supérieur, Est_Cadre, Salaire, Description, Date_Début, Date_Fin, Budget)

Schéma mal conçu (redondance des données)

Num_Employé	Num_Projet	Début_Affect	Nom	Fonction	Date_Début
1009	122	07/03/2011	Touati	Concepteur	NULL
1001	122	08/03/2011	Belaid	Chef de projet	NULL
1023	122	15/06/2011	Djabi	Développeur	NULL
1009	103	12/09/2010	Touati	Concepteur	NULL
1001	208	15/06/2011	Belaid	Analyste	NULL
1018	NULL	NULL	Malek	Analyste	NULL
1015	NULL	NULL	Grir	Analyste	NULL
NULL	420	NULL	NULL	NULL	01/05/2012
NULL	430	NULL	NULL	NULL	19/05/2012
NULL	431	NULL	NULL	NULL	15/05/2012

A chaque nouvelle affectation, toutes les informations sur l'employé et le projet concernés sont répétées!!!!

Schéma mal conçu (problème de mise à jour)

Mise à jour, posant problème, de la fonction de l'employé 1009.
UPDATE Affectation_2 SET Fonction = 'Chef de projet' WHERE Num_Employé = 1009 AND Num_Projet = 103;

Num_Employé	Num_Projet	Début_Affect	Nom	Fonction	Date_Début
1009	122	07/03/2011	Touati	Concepteur	NULL
1001	122	08/03/2011	Belaid	Chef de projet	NULL
1023	122	15/06/2011	Djabi	Développeur	NULL
1009	103	12/09/2010	Touati	Chef de projet	NULL
1001	208	15/06/2011	Salem	Analyste	NULL
1018	NULL	NULL	Malek	Analyste	NULL
1015	NULL	NULL	Grir	Analyste	NULL
NULL	420	NULL	NULL	NULL	01/05/2012
NULL	430	NULL	NULL	NULL	19/05/2012
NULL	431	NULL	NULL	NULL	15/05/2012

Quelle est la fonction de l'employé 1009 ???

Schéma mal conçu (contraintes d'intégrité difficiles à définir)

Num_Employé	Num_Projet	Début_Affect	Nom	Fonction	Date_Début
1009	122	07/03/2011	Touati	Concepteur	NULL
1001	122	08/03/2011	Belaid	Chef de projet	NULL
1023	122	15/06/2011	Djabi	Développeur	NULL
1009	103	12/09/2010	Touati	Chef de projet	NULL
1001	208	15/06/2011	Belaid	Analyste	NULL
1018	NULL	NULL	Malek	Analyste	NULL
1015	NULL	NULL	Grir	Analyste	NULL
NULL	420	NULL	NULL	NULL	01/05/2012
NULL	430	NULL	NULL	NULL	19/05/2012
NULL	431	NULL	NULL	NULL	15/05/2012

Comment imposer que Num_Employé et Nom ne soient pas nuls ???

…

Dépendances fonctionnelles (DF)

Dépendance fonctionnelle sur R (A1, A2, …, An, B)

A1, A2, …, An → B

"Si deux tuples de R ont les mêmes valeurs pour les attributs de A1, A2, …, An alors ils ont même valeur pour les attributs de B. »

Exemple:

Num_Employé → Nom, Prénom

Num_Employé, Num_Projet → Début_Affect

Code_Postal → Ville

De façon plus générale, soient A1, A2, …,An, B1,B2,…,Bp des attributs

A1, A2, …, An → B1, B2, …, Bp si la donnée d´une valeur pour chacun des attributs A1, A2, …, An détermine au plus une valeur pour chacun des attributs B1, B2, …, Bp.

Exemple:

Num_Employé, Num_Projet → Début_Affect, Fin_Affect, Supérieur

Axiomes d'Armstrong

Soient W, X, Y et Z des ensembles d'attributs.

- **Axiomes d'Armstrong :**
  - **Réflexivité** : si $Y \subseteq X \Rightarrow X \rightarrow Y$
  - **Augmentation** : Si $X \rightarrow Y \Rightarrow \forall Z \;\; XZ \rightarrow YZ$
  - **Transitivité** : Si $X \rightarrow Y$ et $Y \rightarrow Z \Rightarrow X \rightarrow Z$

- **On déduit :**
  - **Union :** $\{X \rightarrow Y, X \rightarrow Z\} \models \{X \rightarrow YZ\}$
  - **Pseudo-transitivité** : $\{X \rightarrow Y, WY \rightarrow Z\} \models \{XW \rightarrow Z\}$
  - **Décomposition :** Si $X \rightarrow Y$ et $Z \subseteq Y$ alors $X \rightarrow Z$ ou
    Si $X \rightarrow ZY$ alors $X \rightarrow Z$ et $X \rightarrow Y$
  - **Composition :** Si $X \rightarrow Y$ et $Z \rightarrow W$ alors $XZ \rightarrow YW$
  - **Auto-détermination :** $X \rightarrow X$

A |= B: Inférence. Déduction logique de B à partir de A

Définitions

Une dépendance $A_1, A_2, \ldots, A_n \rightarrow B_1, B_2, \ldots, B_m$ est :

- **triviale** : si l'ensemble $\{B_1, B_2, \ldots, B_m\}$ est un sous-ensemble de $\{A_1, A_2, \ldots, A_n\}$
Num_Employé, Num_Projet → Num_Projet

- **non triviale** : si au moins un B_i n'appartient pas à $\{A_1, A_2, \ldots, A_n\}$
Num_Employé, Num_Projet → Num_Employé, Début_Affect

- **complètement non triviale** : si aucun des B_i n'appartient à $\{A_1, A_2, \ldots, A_n\}$
Num_Employé, Num_Projet →Debut_Affect, Fin_Affect, Supérieur

- **élémentaire** : si elle ne contient qu'un seul B_i. Que ce B_i n'appartient pas à l'ensemble $\{A_1, A_2, \ldots, A_n\}$ et qu'il n'existe pas de sous-ensemble $X \subset \{A_1, A_2 \ldots A_n\}$ tel que $X \rightarrow B_i$
Num_Employé, Num_Projet →Debut_Affect

- **directe** : une dépendance fonctionnelle est directe si elle est élémentaire et qu'elle ne peut pas être déduite par transitivité d'autres dépendances fonctionnelles.

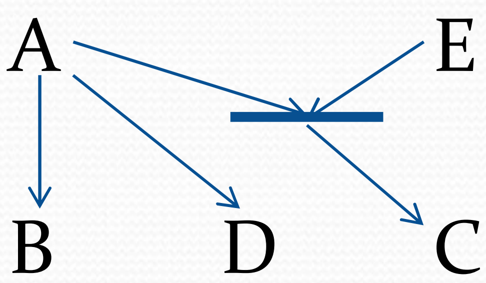
Graphe de dépendances

Graphe orienté représentant des dépendances fonctionnelles. Les sommets du graphe représentent les attributs et les arrêtes les liens de dépendances fonctionnelles.

Exemple:

$\{A \rightarrow B, A \rightarrow D, AE \rightarrow C\}$

Graphe de DF:



Fermeture transitive

Fermeture transitive d'une famille de DF

Définition 1:

On appelle fermeture transitive d´une famille de dépendances fonctionnelles (noté F^+) l'ensemble des DF impliquées par F.

$$\mathbf{F}^+ = \{X \rightarrow Y \setminus F \mid= X \rightarrow Y\}$$

Définition 2:

On appelle fermeture transitive d'une famille de dépendances fonctionnelles (notée F^+) l'ensemble des DF pouvant être déduite de F par transitivité.

Fermeture transitive d'un ensemble d'attributs X par rapport à une famille de dépendances fonctionnelles F (noté $[X]^+$)

$[X]^+$: ensemble des attributs A pour lesquels la DF $X \rightarrow A$ est dans la fermeture transitive de F.

$$[\mathbf{X}]^+ = \{A \setminus F \mid= X \rightarrow A\}$$

Fermeture transitive

$F = \{A \to B, B \to C\}$

$F^+ = \{A \to A, A \to B, B \to C, A \to C, AB \to A, AB \to B, AC \to A, AC \to B, AC \to C, ABC \to A, ABC \to B, ABC \to C\}$

On ne conserve, dans la fermeture transitive, que les dépendances fonctionnelles élémentaires.

$F^+ = \{A \to B, B \to C, A \to C\}$

Fermeture transitive d'un ensemble d'attributs

Algorithme

Soit X l'ensemble des attributs qui vont éventuellement appartenir à la fermeture de {A1, A2, …, An}

$X^+ := X$

Répéter

$AUX := X^+$

pour chaque DF $Y \rightarrow Z$ de F faire :

Si $Y \subseteq X^+$ alors $X^+ := X^+ \cup Z$

Jusqu'à $AUX = X^+$ ou $X^+ = R$

Exemple

Considérons la relation R avec pour attributs A, B, C, D, E et G et l´ensemble F = {AB → C , BC → AD , D → E , CG → B }. Calculons la fermeture $[AB]^+$:

$X^+ = \{A, B\}$

AUX = {A, B}

On a AB → C d´où $X^+ = \{A, B, C\}$

On a BC → AD, soit par décomposition BC → A et BC → D, d´où $X^+ = \{A, B, C, D\}$

On a D → E d´où $X^+ = \{A, B, C, D, E\}$

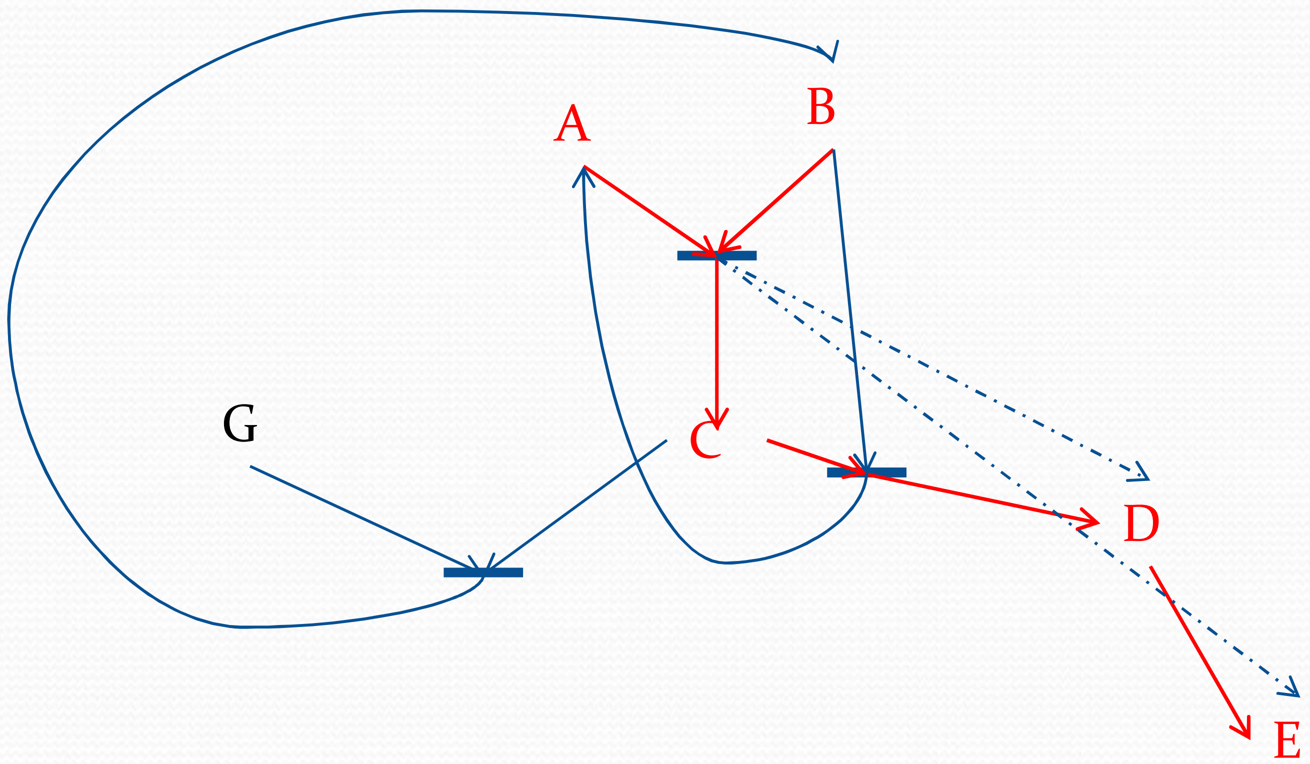
La dépendance fonctionnelle CG → B ne peut être utilisée car G n´appartient pas a X^+

Par conséquent $[AB]^+ = \{A, B, C, D, E\}$

Fermeture transitive $[AB]^+$

$F = \{AB \rightarrow C\ ,\ BC \rightarrow AD\ ,\ D \rightarrow E\ ,\ CG \rightarrow B\ \}$

$[AB]^+ = \{A, B, C, D, E\}$



Equivalence et couverture

Deux familles de dépendances fonctionnelles F et G sont **équivalentes** si $\mathbf{F^+ = G^+}$

Si $F^+ \subset G^+$ alors G est **une couverture** de F

Exemple :

Montrons que $F = \{A \rightarrow BC , B \rightarrow C \}$ et $G = \{A \rightarrow B , B \rightarrow C \}$ sont équivalentes :

Par décomposition de $A \rightarrow BC$ on a que $F = \{A \rightarrow B, A \rightarrow C, B \rightarrow C\}$

Par la suite, on montre que $F^+ = G^+$

Couverture minimale

Une famille de dépendances fonctionnelles F est minimale si :

- En partie droite de toute dépendance de F, il n'y a qu'un seul attribut.

- Il n'y a pas de dépendance fonctionnelle $X \rightarrow A$ dans F telle que
  $(F - \{X \rightarrow A\})$ soit équivalente à F (pas de DF pouvant être obtenues par transitivité)
  Ville $\rightarrow$ Pays
  Pays $\rightarrow$ Monnaie
  Ville $\rightarrow$ Monnaie

- Il n'y a pas de dépendance fonctionnelle $X \rightarrow A$ et $Z \subset X$ tels que
  $(F - \{X \rightarrow A\}) \cup \{Z \rightarrow A\}$ soit équivalente à F (pas de DF non élémentaires)
  Num_Employé, Num_Projet $\rightarrow$ Nom
  Num_Employé $\rightarrow$ Nom

Attention: Pas d'unicité des couvertures minimales

Algorithme couverture minimale

1. On réécrit les dépendances fonctionnelles de F de telle sorte que chaque dépendance doit avoir un seul attribut à droite (décomposition)
2. On détermine une suite $F_0, F_1, \ldots, F_p$ telle que :
   a) $F_0 = F$
   b) $F_{i+1} = F_i - \{X_i \rightarrow A\}$ avec $X_i \rightarrow A$ une dépendance fonctionnelle de F_i et F_{i+1} équivalente à F_i .

- Tant que on peut enlever des dépendances. Le résultat F' est équivalent à F

1. On détermine une suite $F'_0, F'_1, \ldots, F'_p$ telle que :
   a) $F'_0 = F'$
   b) $F'_{i+1} = F'_i - \{X_j \rightarrow A\} \cup \{Y_j \rightarrow A\}$ où $Y_j \subset X_j$ et F'_{i+1} équivalente à F'_i

- Tant que on peut enlever des attributs à gauche des dépendances. Le résultat est F'' équivalent à F.
- F'' est une couverture minimale de F

Exemple 1

- Soient la relation R(A, B, C) et F = {$A \rightarrow B$, $A \rightarrow C$, $B \rightarrow A$, $B \rightarrow C$, $C \rightarrow A$, $C \rightarrow B$}

1. Toutes les dépendances fonctionnelles de F ont un seul attribut à droit.
2. On pose $F_0 = F$

- $F_1 = F_0 - \{A \rightarrow C\}$ car $A \rightarrow C$ peut être déduite par transitivité de $A \rightarrow B$ et $B \rightarrow C$.
- $F_2 = F_1 - \{C \rightarrow B\}$ car $C \rightarrow B$ peut être déduite par transitivité de $C \rightarrow A$ et $A \rightarrow B$.
- $F_3 = F_2 - \{B \rightarrow A\}$ car $B \rightarrow A$ peut être déduite par transitivité de $B \rightarrow C$, $C \rightarrow A\}$

F_3 est une couverture minimal de F.
$F_3 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

Exemple 2

- Soient la relation R(A, B, C) et F = {$A \rightarrow BC$, $B \rightarrow C$, $A \rightarrow B$, $AB \rightarrow C$}

1. On réécrit F comme F = {$A \rightarrow B$, $A \rightarrow C$, $B \rightarrow C$, $AB \rightarrow C$}
2. On pose $F_0 = F$

- $F_1 = F_0 -$ {$A \rightarrow C$} car $A \rightarrow C$ peut être déduite par transitivité de $A \rightarrow B$ et $B \rightarrow C$.
- F_1 est équivalente à F

3. On pose $F'_0 = F_1$

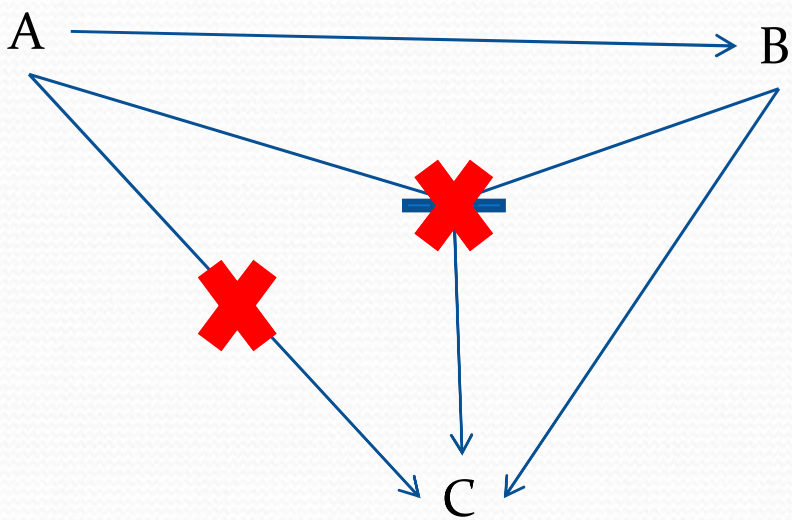
- $F'_1 = F'_0 -$ {$AB \rightarrow C$} car $B \subset AB$ et que $F'_0 -$ {$AB \rightarrow C$} $\cup$ {$B \rightarrow C$} est équivalente à F'_0

Une couverture minimal de F est {$A \rightarrow B$, $B \rightarrow C$}

Exemple 2

R(A, B, C) et F = {$A \rightarrow BC$, $B \rightarrow C$, $A \rightarrow B$, $AB \rightarrow C$}

Une couverture minimal de F est {$A \rightarrow B$, $B \rightarrow C$}



Clé primaire

- Soit $R(A_1, A_2, \ldots, A_n)$ et F une famille de DF sur R
- Un sous-ensemble X de $\{A_1, A_2, \ldots, A_n\}$ est une **clé primaire** de R si
    1. La dépendance fonctionnelle $X \rightarrow A_1, A_2, \ldots, A_n \in F^+$
    2. $\forall\, Y \subset X$, on a pas $Y \rightarrow A_1, A_2, \ldots, A_n$
- Si X n'est pas un ensemble minimal alors X est une **surclé**
- Les dépendances fonctionnelles permettent de déduire les clés des relations

Propriétés de la clé primaire

- Tout attribut qui ne figure pas dans le membre droit d´une DF non triviale de F doit appartenir à toute clé de R.
- Si l'ensemble des attributs de R qui ne figure pas en membre droit d'une DF non triviale de F est une clé, alors F possède une clé minimale unique formée de l'ensemble de ces attributs.
- Un schéma de relation muni d'une seule DF possède une clé minimale unique.
- Si une relation possède plusieurs clés elles sont appelées clés candidates. Parmi ces clés candidates, une seule doit être choisie pour être la clé primaire de la relation.

Décomposition de schéma

La décomposition de schéma doit garantir:

1. La préservation de l'information

2. La préservation des dépendances fonctionnelles

Décomposition de schéma

On appelle *décomposition d'un schéma de relation* R = (A_1, A_2, …, A_n) où A_1, A_2, …, A_n sont des attributs, le remplacement de R par un ensemble de schémas de relation R_1, R_2, …, R_p ($p \geq 1$), obtenus à partir de R par projection et de telle sorte que la réunion des attributs de R_i ($1 \leq i \leq p$) par jointure soit égale à R.

Exemple : Soient R = (COURS, ETUDIANT, NOTE, PROF)
R1 = (COURS, ETUDIANT, NOTE)
R2 = (COURS, PROF)
R1 et R2 forment une décomposition de R.

Décomposition sans perte d'information (SPI)

- On dit qu'une *décomposition* de R en R1, R2, … Rp, est *sans perte d'information*, ce que l'on note SPI, si tous les tuples r sur R considérés sont égaux à la jointure des tuples ri ($1 \leq i \leq p$) obtenus par projection de r sur les schémas Ri.

$R1 = \pi_{(att1,\ att2...)}\ R$

$R2 = \pi_{(att'1,\ att'2...)}\ R$

.

$Rp = \pi_{(att''1,\ att''2...)}\ R$

$R1 \bowtie R2 \ldots \bowtie Rp = R$

$\{att1, att2...\} \cap \{att'1, att'2...\} \cap \ldots \cap \{att''1, att''2...\} \neq \emptyset$

Décomposition sans perte d'information (SPI)

Condition nécessaire et suffisante :

Théorème de Heath:

Soit R=(X, Y, Z) où X, Y et Z sont des ensembles d'attributs. R est munie d'une famille de DF F. Si on suppose que $X \rightarrow Y$ appartient à F+, alors la décomposition de R en S = (X, Y) et T = (X, Z) est SPI. Réciproquement, si la décomposition de R en S et T est SPI, alors $X \rightarrow Y$ ou $X \rightarrow Z$ appartient à F+.

Exemple

R = (COURS, ETUDIANT, NOTE, PROF)

F = {COURS $\rightarrow$ PROF, (COURS, ETUDIANT) $\rightarrow$ NOTE}

Selon le théorème de Heath: R1 et R2 sont une décomposition SPI de R munie de F.

R1 = (COURS, ETUDIANT, NOTE)

R2 = (COURS, PROF)

$R1 = \pi_{(COURS,\ ETUDIANT,\ NOTE)}\ R$

$R2 = \pi_{(COURS,\ PROF)}\ R$

$R1 \bowtie R2 = R$

Décomposition sans perte de dépendances (SPD)

- Soit R décomposée en $R_1, R_2, .., R_p$. On appelle *restriction* de F à Ri, notée Fi, l'ensemble des DF de F+ formées d'attributs de Ri uniquement.

Exemple:

Restriction de F à R1 : F1 = {(COURS, ETUDIANT) → NOTE}

Restriction de F à R2 : F2 = {COURS → PROF}

- On dit que la décomposition de R en $R_1, R_2, .., R_p$ préserve les DF, ou encore est *sans perte de dépendances*, ce que l'on note SPD, si l'union des fermetures transitives des Fi ($1 \leq i \leq p$) est égale à F+.

Les formes normales

Les 3 premières formes normales ont pour objectif de permettre la décomposition de relations sans perdre d'informations à partir de la notion de dépendance fonctionnelle. L'objectif de cette décomposition est d'obtenir un schéma conceptuel représentant les entités et les associations du monde réel.

Reconstituer les entités et les associations à partir des liens (DF) entre les attributs.

Première forme normale (1FN)

Un schéma de relation est dit en première forme normale (1FN) si tous les attributs qui le composent sont atomiques et indivisibles.

Une relation en 1FN ne contient pas d'attributs composées ni d'attributs multi-valués.

Relation Employé en 1FN

Num_Employé	Nom	Prénom	Date_Naissance	Fonction	Est_Cadre
1001	Belaid	Toufik	12/05/1965	Concepteur	true
1009	Touati	Rachid	13/09/1941	Chef de projet	true
1023	Kadri	Amine	23/11/1970	Développeur	true
1053	Djabi	Fatiha	04/06/1980	Analyste	false
1026	Bouras	Kamel	19/04/1968	Administrateur	true
1005	Djabi	Fatiha	22/08/1976	Développeur	false

Relation Employé en NF2 (pas en 1FN)

Num_Employé	Nom	Prénom	Date_Naissance	Fonction	Est_Cadre
1001	Belaid	Toufik	12/05/1965	Concepteur Développeur	true
1009	Touati	Rachid	13/09/1941	Chef de projet	true
1023	Kadri	Amine	23/11/1970	Développeur	true
1053	Djabi	Fatiha	04/06/1980	Analyste Développeur	false
1026	Bouras	Kamel	19/04/1968	Administrateur Concepteur	true
1005	Djabi	Fatiha	22/08/1976	Développeur	false

Non First Normal Form (NF2):
Pas en 1ère forme normale

Deuxième forme normale (2FN)

Un schéma de relation R est **2FN** si et seulement si :

- le schéma est en 1FN
- $\forall$ A attribut $\not\in$ une clé, A ne dépend pas d’une partie de la clé. C'est à dire Ė(A **une dépendance fonctionnelle partielle**)

Relation Affectation en 2FN

Num_Employé	Num_Projet	Début_Affect	Fin_Affect	Supérieur
1009	122	07/03/2011	13/11/2011	NULL
1001	122	08/03/2011	28/06/2011	1009
1023	122	15/06/2011	04/10/2011	1009
1009	103	12/09/2010	01/11/2010	NULL
1001	208	15/06/2011	12/10/2011	1009
1009	208	15/06/2011	06/03/2012	NULL
1023	208	01/09/2011	17/12/2011	1009
1009	133	06/11/2011	19/02/2012	NULL
1053	208	01/09/2011	06/03/2012	1026
1026	208	19/08/2011	06/03/2012	1009

Num_Employé, Num_Projet ì Début_Affect, Fin_Affect, Supérieur

Relation Affectation (pas en 2FN)

Num_Employé	Num_Projet	Début_Affect	Fin_Affect	Supérieur	Fonction
1009	122	07/03/2011	13/11/2011	NULL	Concepteur
1001	122	08/03/2011	28/06/2011	1009	Chef de projet
1023	122	15/06/2011	04/10/2011	1009	développeur
1009	103	12/09/2010	01/11/2010	NULL	Concepteur
1001	208	15/06/2011	12/10/2011	1009	Chef de projet
1009	208	15/06/2011	06/03/2012	NULL	Concepteur
1023	208	01/09/2011	17/12/2011	1009	Développeur
1009	133	06/11/2011	19/02/2012	NULL	Concepteur
1053	208	01/09/2011	06/03/2012	1026	Administrateur
1026	208	19/08/2011	06/03/2012	1009	Analyste

Num_Employé, Num_Projet ì Début_Affect, Fin_Affect, Supérieur

Num_Employé ì Fonction

Troisième forme normale (3FN)

Un schéma de relation R est en 3FN ssi :

- le schéma est en 2FN
- $\neg$($\exists$ une dépendance fonctionnelle transitive) c'est à dire une dépendance A $\rightarrow$ B telle que A et B soit des attributs n'appartenant pas à la clé.

Relation Affectation en 3FN

Num_Employé	Num_Projet	Début_Affect	Fin_Affect	Supérieur
1009	122	07/03/2011	13/11/2011	NULL
1001	122	08/03/2011	28/06/2011	1009
1023	122	15/06/2011	04/10/2011	1009
1009	103	12/09/2010	01/11/2010	NULL
1001	208	15/06/2011	12/10/2011	1009
1009	208	15/06/2011	06/03/2012	NULL
1023	208	01/09/2011	17/12/2011	1009
1009	133	06/11/2011	19/02/2012	NULL
1053	208	01/09/2011	06/03/2012	1026
1026	208	19/08/2011	06/03/2012	1009

Num_Employé, Num_Projet → Début_Affect, Fin_Affect, Supérieur

Relation Affectation (pas en 3FN)

Num_Employé	Num_Projet	Début_Affect	Fin_Affect	Supérieur	Nom_Supérieur
1009	122	07/03/2011	13/11/2011	NULL	NULL
1001	122	08/03/2011	28/06/2011	1009	Touati
1023	122	15/06/2011	04/10/2011	1009	Touati
1009	103	12/09/2010	01/11/2010	NULL	NULL
1001	208	15/06/2011	12/10/2011	1009	Touati
1009	208	15/06/2011	06/03/2012	NULL	NULL
1023	208	01/09/2011	17/12/2011	1009	Touati
1009	133	06/11/2011	19/02/2012	NULL	NULL
1053	208	01/09/2011	06/03/2012	1026	Bouras
1026	208	19/08/2011	06/03/2012	1009	Touati

Num_Employé, Num_Projet ì Début_Affect, Fin_Affect, Supérieur, Nom_Supérieur
Supérieur ì Nom_supérieur

Décomposition en 3FN

L'algorithme de Bernstein permet de décomposer un schéma de relation en 3ème forme normale sans perte d'information ni de dépendance fonctionnelle.

Données :

- R est un schéma de relation, R (A1, A2, …,An)
- F est un ensemble de dépendances fonctionnelles définies sur R.

Résultat :

- *Une décomposition de R* muni de F en schémas de relation en 3FN *sans perte d'information ni de dépendance fonctionnelle*.

Algorithme de Bernstein

Étape 1 :

- On remplace F par une couverture minimale de F (cf. algorithme précédent). On cherche les clés minimales de R et on teste si R est en 3FN. Si oui, on s'arrête.

Étape 2 :

- On regroupe les DF $X \rightarrow Ai$ (i entre 1 et p), ayant même membre gauche X. Pour chaque membre gauche X, on définit un schéma de relation contenant tous les attributs intervenant dans ces DF, soit Rx (X, A1, A2, …, Ap). Le schéma Rx est muni de l'ensemble des DF $X \rightarrow Ai$ (i entre 1 et p).

Étape 3 :

- Si aucun des schémas Rx définis à l'étape 2 ne contient de clé de R, on ajoute un schéma Rk = (K), où K est une clé minimale de R, muni d'aucune DF.

Exemple 1

Soient $R(A, B, C, D)$ et $F = \{A \rightarrow BC, A \rightarrow D, C \rightarrow D\}$

Etape 1: Une couverture minimale de $F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D\}$
Une clé de cette relation est A car $A \rightarrow ABCD$.
R n'est pas en 3FN car $C \rightarrow D$.

Etape 2:
$A \rightarrow B$ et $A \rightarrow C \Rightarrow R_A(\underline{A}, B, \mathit{C})$ C est une clé étrangère
$C \rightarrow D \Rightarrow R_C(\underline{C}, D)$

Etape3: La clé de R se trouve dans R_A. Fin d'algorithme.
$R_A(\underline{A}, B, \mathit{C})$
et
$R_C(\underline{C}, D)$
sont une décomposition en 3FN SPI et SPD de R.

Exemple 2

Soient R(A, B, C, D, E, F, G) et F = {AB→C, AB→D, B→D, E→FG}

Etape 1: Une couverture minimale de F = {AB→C, B→D, E→F, E→G}
Une clé de cette relation est ABE car AB→CD et E→FG donc ABE→ABCDEFG.
R n'est pas en 2FN (ni en 3FN)car B→D.

Etape 2:
AB→C ⇒ R_{AB}(<u>A, *B*</u>, C) B est une clé étrangère
B→D ⇒R_B(<u>B</u>, D)
E→F et E→G ⇒R_E(<u>E</u>, F, G)

Etape3: La clé de R ne se trouve dans aucune des relations obtenues. Il faut donc ajouter R_{ABE}(<u>A, B, E</u>).
R_{AB}(<u>A, *B*</u>, C)
R_B(<u>B</u>, D)
R_E(<u>E</u>, F, G)
R_{ABE}(<u>*A, B, E*</u>) A, B, et E sont des clés étrangères
sont une décomposition en 3FN SPI et SPD de R.

Conclusion

La décomposition de schéma relationnel offre des avantages et des inconvénients:

Avantages:

- Réduit le redondance de données
- Elimine les anomalies de mise à jour
- Optimise les requêtes de mise à jour

Inconvénients:

- Augmente la complexité des requêtes (nombre de jointure)
- Augmente le coût d'exécution des requêtes de sélection