Chapitre II : MODELE RELATIONNEL-NORMALISATION DE RELATION

Plan du Chapitre

Mme Z.TAHAKOURT

1- Introduction

Le modèle relationnel a été inventé par Mr CODD en 1970. Il est basé sur des concepts très simples. C'est le modèle de données le plus utilisé par les SGBDs actuels. C'est un modèle de données plus simple que celui de l'E/A.

2- Définitions

Les objets et associations du monde réel sont représentés par un concept unique qui est la « relation ». les relations sont des tableaux à deux dimensions appelés « tables ».

Une relation est définie par :

1. Son nom,
2. liste des couples (attribut, domaine),
3. Son (ses) identifiant(s).

Ces trois informations constituent le schéma ou « intention » de la relation.

La population ou « extension » d'une relation est constituée de l'ensemble des tuples de la relation.

On appelle schéma d'une BDR (BD relationnelle) l'ensemble des schémas de ses relations.

On appelle BDR l'extension de toutes ses relations.

On appelle cardinalité (arité) d'une relation le nombre de tuples que son extension contient.

On appelle degré d'une relation le nombre d'attributs que son intension contient.

3- Concept de normalisation

Afin de dégager quelques propriétés d'un schéma relationnel mal conçu, considérons le schéma de l'exemple suivant sur la livraison des fournitures. Dans cet exemple on voudrait garder les informations concernant les produits et leurs fournisseurs :

FOURNITURE(<u>NF, NP</u>, NomF, AdrF,designationP , QTE)

dont l'extension est la suivante :

<u>NF</u>	<u>NP</u>	NomF	Date	AdrF	designationP	QTE
01	A	Ali	01/05/1985	Béjaia	Pain	205
…						

Cette exemple soulève plusieurs types de problèmes :

1. <u>Redondance</u> : l'adresse et le nom d'un fournisseur est répétée pour chaque pièce qu'il fournit

2. <u>Anomalie de mise à jour</u> : lors d'une mise à jour, on risque de modifier l'adresse d'un fournisseur dans un tuple mais pas dans les autres. Un même fournisseur possèdera plusieurs adresses !

3. <u>Anomalies d'insertion</u> : on n'enregistre pas l'adresse d'un fournisseur, si on n'achète pas au moins une pièce à ce fournisseur.

4. Anomalies de suppression : si on supprime toutes les pièces proposées par un fournisseur ce dernier n'apparaît plus.

Le schéma précédent peut se décomposer selon deux schémas :

Fournisseur (NF,NomF, AdrF) , Produit (NP, DesignationP) et Livraison (NF,NP,date, Qte)

Le processus de transformation d'une relation posant des problèmes lors des m-à-j en relations n'ayant pas ces problèmes, est appelé processus de normalisation ou décomposition.

Cette décomposition élimine les problèmes précédents. En revanche, pour trouver l'adresse d'un fournisseur qui fournit une pièce donnée, il faut recourir à une jointure entre Fournisseur, et Livraison. L'opération peut se révéler coûteuse en temps, mais si on doit choisir entre la cohérence de la base de données et le gain de temps c'est la cohérence qui prime.

On mesure la qualité d'une relation (ou sa capacité à représenter le monde réel) par son degré de normalisation. Une relation peut être de la moins bonne à la meilleure : 1FN, 2FN, 3FN, BCNF, 4FN, 5FN,…).

Les procédures qui permettent de décomposer une relation en " bonnes relations", reposent sur le concept de *dépendance fonctionnelle*(DF).

4- Concept de dépendance fonctionnelle(DF)

Une *Dépendance fonctionnelle* (DF) sur un schéma relationnel R, est une proposition de la forme Si , pour tous tuples t1 et t2 de R qui ont la même valeur pour l'attribut A, on a t1 et t2 qui ont les même valeurs pour l'attribut B, alors on dit que A détermine fonctionnellement B.. On note : A → B. A est appelé : *Prémisse*, et B : *conclusion*.

Remarque

Si l'attribut A détermine plusieurs attributs B ,C et D…
On écrit : A → BCD

Exemple :R(A,B,C)
On a :
A → B ; A → C ; c'est tout. Toutes les autres DF ne sont pas satisfaites.

A	B	C
1	5	4
2	5	6
3	6	6

4-1 Dépendance fonctionnelle élémentaire (L-Réduite: Réduite à gauche)

Une dépendance fonctionnelle X→A est dite élémentaire ou réduite à gauche, si :

1. A est un attribut n'appartenant pas à X ;
2. $\nexists X' \in X /$ tel que X' → A,

en d'autres mots:

1. A est un attribut n'appartenant pas à X ;
2. $\forall X' \in X$: $A \notin X'^{+}$.

Quand la DF X→A n'est pas élémentaire, (X-X') sont appelés attributs *accessoires*.

Exemple: Soit une relation R(A,B,C) et F un ensemble de dépendances fonctionnelles.
F ={ A→B, A→BC,AB→C }

1. $A \rightarrow B$: DF élémentaire.
2. $A \rightarrow BC$: DF élémentaire
3. $AB \rightarrow C$:

on vérifie $A^+=\{A, B, C\}$ ici on a $C \in A^+$ donc la DF $AB \rightarrow C$ est non élémentaire, et B est un attribut accessoire.

4-2 Dépendance fonctionnelle triviale

Une dépendance fonctionnelle $X \rightarrow A$ est dite triviale si et Seulement si :

1. $A \in X$

Exemple : Soit une relation R(A,B,C). $AB \rightarrow B$: DF triviale. $A \rightarrow BC$: DF non triviale

4-3 Dépendance fonctionnelle déduite ou redondante

Une dépendance fonctionnelle $X \rightarrow A$ est dite déduite si partant de X on peut arriver à A sans utiliser la DF $X \rightarrow A$. en d'autres mots, $X \rightarrow A$ est redondante dans F si elle est conséquence de $F - \{X \rightarrow A\}$.

Exemple
R(A,B,C,D) ;
$F=\{A \rightarrow B,$ non déduite
$B \rightarrow C,$ non déduite
$A \rightarrow C\}$ déduite

5- Propriété des dépendances fonctionnelles : Axiomes d'Armstrong

Soit A, B, C, D des sous-ensembles quelconques d'attributs d'une relation donnée R. Notons AB l'union de A et de B. Les règles permettant de produire de nouvelles DF à partir d'un ensemble donné de DF sont les suivantes, et sont connues sous le nom d'axiomes d'Armstrong.

1. **Réflexivité** si $B \subseteq A$, alors $A \rightarrow B$.

   Cette règle stipule que tout ensemble d'attribut détermine lui-même ou une partie de lui même.
2. **Augmentation** : si $A \rightarrow B$, alors $AC \rightarrow BC$
3. **Transitivité** : si $A \rightarrow B$ et $B \rightarrow C$ alors $A \rightarrow C$

D' autres règles peuvent se déduire de ces axiomes:

1. **Décomposition** : si $A \rightarrow BC$, alors $A \rightarrow B$ et $A \rightarrow C$
2. **Union** : si $A \rightarrow B$ et $A \rightarrow C$, alors $A \rightarrow BC$
3. **Pseudo-transitivité** : si $A \rightarrow B$ et $BC \rightarrow D$ alors $AC \rightarrow D$

6- La fermeture d'un ensemble d'attributs

Soit X un ensemble d'attribut et F un ensemble de dépendances fonctionnelles. On appelle fermeture de X sous F, notée $\mathbf{X^+}$, l'ensemble des attributs de R qui peuvent être déduits de X à partir de F en appliquant les axiomes d'Armstrong. Ainsi, Y sera inclus dans X^+ ssi $X \rightarrow Y$.

Algorithme de Calcul de la fermeture d'un ensemble d'attributs :

1. initialiser $(X)^+$ à X,
2. trouver une dépendance fonctionnelle f de F possédant en partie gauche des attributs inclus dans $(X)^+$,
3. ajouter dans $(X)^+$ les attributs placés en partie droite de la DF f,
4. répéter les étapes 2) et 3) jusqu'à ce que X^+ n'évolue plus.

Exemple

Soit F = { A → D ; AB → E ; BI → E ; CD → I ; E → C }.
Question : calculer la fermeture, sous F, de AE.
Solution : au départ, $(AE)^+$ = {A,E},
A → D permet d'ajouter D : $(AE)^+$ = {A,E,D}
E → C permet d'ajouter C : $(AE)^+$ = AEDC,
CD → I permet d'ajouter I : $(AE)^+$ = AEDCI.
Question : calculer la fermeture, sous F, de BE.
Solution : au départ, $(BE)^+$ = BE,
E → C permet d'ajouter C : $(BE)^+$ = BEC.

Exercice :

Soit F = { AB → C ; B → D ; CD → E ; CE → GH ; G →A }.
1. Montrer en utilisant les axiomes d'Armstrong que AB → E.
B → D => AB → D par augmentation puis décomposition,
AB → C et AB → D => AB → CD par union,
AB → CD et CD → E => AB → E par transitivité.

1. Montrer en utilisant la notion de fermeture d'attributs que AB → E.
$(AB)^+$={A, B, C, D, E, G, H} comme E ∈ $(AB)^+$ donc AB → E

7- La couverture minimale d'un ensemble de DFs

Un ensemble de DF F est dit couverture minimale, si aucune DF f de F ne peut être déduite à partir de F en appliquant les propriétés des DFs.

Algorithme de calcul de couverture minimale

1. Mettre F sous forme canonique: décomposer chaque DF de façon à avoir un seul attribut à droite. (Décomposition)
2. Réduire à gauche les DF non élémentaires: Pour toute DF X → Y, s'il existe un Z ∈ X tel que Y ∈ Z^+ alors: remplacer la DF X → Y par la DF Z→Y
3. Supprimer les DF redondantes (qu'on peut obtenir par calcul à base des axiomes d'Armstrong à partir des autres DF)

Remarque: Pour chaque ensemble de DF, correspond au moins une couverture minimale (il peu y en avoir plusieurs, cela dépendra de l'ordre des réductions que l'on effectuera.

Exemple1:

Soit la relation R(A, B, C, D, E) avec les DF F ={A→B, AB→C, B→DE, A→D } . Donner une couverture minimale de F.

on applique l'algorithme:

Etape1: Mettre F sous forme canonique:

F 1={A→B, AB→C, B→D, B→E, A→D }

Etape2: Réduction à gauche des DF de F:

Pour les DF A→B, B→D, B→E, A→D } elles sont toutes réduite à gauche
On va tester si AB→C est réduite à gauche ou pas c-à-d s'il existe un Z ∈ {AB} tel que C ∈ Z^+ alors: A^+={A, B, C, D, E}; on voit que C ∈ A^+ donc B est un attribut accessoire on peut donc le supprimer, et remplacer la DF AB→C par A→C
F 2= {A→B, A→C, B→D, B→E, A→D }

Etape3: Supprimer les DF redondante de F:
Les DF: A→B, A→C, B→D, B→E ne sont pas redondantes par contre A→D l'est:
A→B et B→D par transitivité on a A→D.
La couverture minimale F ' = {A→B, A→C, B→D, B→E }

Exemple2 :Soit R(A,B,C,D,E). F ={A →B, BC →D, AC →BDE, D →E }

Etape1: Mettre sous forme canonique F:
A →B, BC →D, AC →B, AC →D, AC →E, D →E

Etape2: Réduction à gauche des DF de F:
Les DF A →B et D →E sont réduite à gauche, vérifions pour les DF BC →D, AC →B, AC →D, AC →E

1. BC →D: B^+={B}, C^+={C} , D n'appartient ni à B^+ ni à C^+ donc la DF BC →D est réduite à gauche.
2. AC →B: A^+={A, B}, B∈ A^+ l'attribut C est accessoire, la DF AC →B est remplacée par A→B
3. AC →D: A^+={A, B}, C^+={C}, D n'appartient ni à A^+ ni à C^+ donc la DF AC →D est réduite à gauche.
4. AC →E: A^+={A, B}, C^+={C}, E n'appartient ni à A^+ ni à C^+ donc la DF AC →E est réduite à gauche.

F ={ A →B, BC →D, A →B, AC →D, AC →E, D →E}

Etape3: Supprimer les DF redondante de F:

1. La DF (A →B) est redondante.
2. La DF (AC →D) est déduite (redondante) car:
   A →B =>(Augmentation) AC →BC et on a BC →D => (Transitivité) AC →D
3. De même (AC →E) est déduite car:
   A→B =>(augmentation) AC → BC, BC→D =>(Transitivité) AC → D ,D →E => (Transitivité) AC→E .

Donc la couverture minimale est: F '={A →B, BC →D, D →E}

Exemple3: Soit R(A,B,C,D,E). F ={AB→C ,B→E, C→ED, E→D, BC→D }. Montrer que F n'est pas une couverture minimale. Trouver sa couverture minimale

F '={AB→C ,B→E, E→D, C→E}.

Exercice:

Soit la relation R(A, B, C, D, E) avec les Dfs F ={A→CD, C→BDE, D→CE} . Donner deux couvertures minimales F.

Réponse: F1 ={A→C, C→B, C →D, D→C,D→E}   F2 ={ A→D, C→B,C →D, D→C,C→E}

8- Graphe de DFs

Un Graphe des dépendances est un ensemble de nœuds et d'arcs tels que :

1. Chaque nœud du graphe est un attribut atomique, ou non.
2. Chaque arc du graphe est une dépendance fonctionnelle.
3. Les arcs sont orientés.

Dans le cas où toutes les DFs sont non déduite, on obtient un *graphe minimum de DFs.*

Exemple : Soit R(A,B,C,D,E) une relation. F ={AB→C, B→E, C→B, C→E, D→C}

1. Donner le graphe de DF de F.
2. Donner le graphe minimum de DF de F.

9- Clé minimale, clé candidate, clé primaire, clé alternative et super clé

Notion de clé minimale

Soit R (A1, ..., An) une relation. X un sous ensemble de {A1, ..., An}. On dit que X est une « clé minimale » de la relation R, si X respecte les deux contraintes suivantes :

1. Unicité. Deux n-uplets distincts de R ne peuvent avoir même valeur de X.
2. Irréductibilité (ou minimalité). Il n'existe pas de sous-ensemble strict de X garantissant la règle d'unicité.

En d'autres mots :

1. X → A1 ... An (X^+=Attr(R)), et
2. $\nexists X' \in X /$ tel que X' → A1 ... An (X'^+=Attr(R)).

Si la relation comporte plusieurs clés minimales, celles-ci sont appelées clés *candidates*, le modèle relationnel exige que seul une de ces clés candidates soit choisie comme clé *primaire* pour cette relation; les autres clés, s'il y en a, sont appelées clés *alternatives*.

Soit K une clé minimale d'une relation. On appelle « super clé» d'une relation, un sous-ensemble d'attributs K' de la relation, telle que : K∈K'. C-à-d que K' est une clé ne garantissant pas la contrainte de minimalité.

Exemple

On considère la relation Fournisseur(Nom, Article, Prix, Adresse)
F={Nom,Article → Prix ; Nom → Adresse}
Montrer que (nom,article) est une clé minimale de fournisseur.

Solution

1- $(nom,article)^+$={nom, article, prix, adresse}=Attr(Fournisseur)
2- Nom^+={nom, adresse}≠ Attr(Fournisseur)
3- $Article^+$={article}}≠ Attr(Fournisseur)

Donc, (nom,article) est une clé minimale de fournisseur.

10- Propriété des clés minimales

Soit R une relation et F un ensemble de DF. Alors :

1. Propriét1 : tout attribut de R qui ne figure pas dans les conclusions des DFs doit appartenir à toutes les clés minimales.
2. Propriét2 : si l'ensemble des attributs de R qui ne figurent pas dans les conclusions des DFs forment une clé minimale, alors celle-ci est unique.

Exemple

On considère la relation R(A,B,C,D) ; F={A→BC ; C→B}
Trouver toutes les clés minimales de R.

Solution

Les attributs qui ne figure pas dans les conclusions des DFs, sont : AD, on va calculer $(AD)^+$.
$(AD)^+$ = {A, D, B, C} =Attr(R). D'après la Propriét1 {AD} appartiendrait à toutes les clés min de R
A^+= {A, B, C} ≠ Attr(R). D^+={D}≠ Attr(R). D'après la Propriét2 AD est l'unique clé minimale de R.

11- Décomposition des relations

Revenons à notre objectif initial, qui est la décomposition d'une relation afin d'éviter les redondances et les anomalies qui en découlent. On peut décomposer R en R1,...,Rn en s'assurant de ne pas perdre de l'information ni de DF. On définit pour cela les deux propriétés suivantes :

12-1 Décomposition sans perte d'information

Une décomposition est dite sans perte d'information si la jointure naturelle des sous-relations calcule exactement tous les tuples de la relation initiale. Autrement dit, si on a : R = R1 ⋈R2 ⋈... ⋈Rn

Considérons la relation R suivante :

R	A	B	C	D
	a	5	x	2
	b	5	y	1
	c	5	x	2

R1	B	C	D
	5	x	2
	5	y	1

R2	A	B
	a	5
	b	5
	c	5

Cette relation a pour schéma R(A,B,C,D) que l'on décompose en R1(A,B) et R2(B,C,D). Cette décomposition n'est pas sans perte d'information puisque R ne vaut pas R1 ⋈R2.
Par contre, si l'on décompose R en R1(A,B,C) et R2(A,D), on obtient une décomposition sans perte d'information car R1 et R2 vallent : et dans ce cas, R = R1 ⋈R2.

R1	A	B	C
	a	5	x
	b	5	y
	c	5	x

R2	A	D
	A	2
	b	1
	c	2

12-2 Décomposition sans perte de dépendances fonctionnelles

On aimerait aussi que la décomposition préserve les dépendances fonctionnelles. Si R est une relation qui satisfait un ensemble de dépendance F, et si R se décompose en R1,...,Rn, on note F_i l'ensemble des DF de R qui portent sur Attr(R_i). La décomposition préserve les dépendances fonctionnelles si ($F_1 \cup ... \cup F_n$) est équivalent à F.

Reprenons l'exemple. La relation R(A,B,C,D) avec l'ensemble de DF F={A → BC,C → D,D → B} La décomposition de R en R1(A,B,C) et R2(A,D) ne préserve pas les DF. En effet F_1, ensemble des DF qui portent sur les attributs de R1 est {A → BC} et l'ensemble F_2 des DF qui portent sur les attributs de R2 est vide.

13. Les formes normales

13.1 Les type d'attributs

1. Attribut *simple* = non divisible (ex. âge).
2. Attribut *composé* = subdivisé en attributs simples sous forme d'une hiérarchie (exemple : adresse postale = rue + code postal + ville + pays).
3. Attribut *monovalué* = qui a une seule valeur par tuple.
4. Attribut *multivalué* = qui a plusieurs valeurs par tuple (Possibilité d'imbriquer composition et multivaluation ⇒ objets complexes).
5. Attribut *dérivé* = dont la valeur est calculée (ex. prix T.T.C. à partir du prix H.T.).

Exemple

Soit la relation Etudiant suivante

Etudiant (matricule, nom, prenom, adresse, n°rue, nomrue, datenaissance, journais, moisnais, anneenais, {telephones})

13.2 La première forme normale (1FN)

Une relation est dite en première forme normale (1FN) , si tous ses attributs sont simples et mono-valués.

Remarque : si rien n'est précisé sur les attributs d'une relation, alors ces derniers sont considérés comme simples et monovalués.

Exemple

On considère une entreprise possédant des départements éparpiés dans plusieurs régions. On note par *Departement* larelation suivante : Departement (N°Dpt, NomDpt, {LieuDpt})

N°Dpt	NomDpt	LieuDpt
1	Recherche	{Alger, T.Ouzou}
2	Administration	Oran

3	Commerce	{Alger, Annaba}

Cette relation *Departement* n'est pas en 1FN car l'attribut LieuDpt est multivalué. Cependant il est possible de décomposer la relation *Departement* en 2 relations de 1FN :

N°Dpt	NomDpt
1	Recherche
2	Administration
3	Commerce

N°Dpt	LieuDpt
1	Alger
1	T.Ouzou
2	Oran
3	Alger
3	Annaba

Departement (N°<u>Dpt</u>, NomDpt) et Localisation (<u>N°Dpt, LieuDpt</u>)

13.3 La deuxième forme normale (2FN)

Soit R une relation. On dit que R est en 2FN, s :

1. R est en 1FN ;
2. Les attributs non clé (n'appartiennent à aucune clé) de R, ne dépendent pas d'une partie de la clé.

<u>Exemple</u>

Soit la relation Livraison(<u>NF, NP</u>, Prix, AdrF, TelF)

F={NP,NF → Prix ; NF→ AdrF ; NF→ TelF}. Livraison est-elle en 2FN ?

$(NP,NF)^+$=Attr(Livraison) ; NP^+≠Attr(Livraison) ; NF^+≠Attr(Livraison) donc (NP,NF) est l'unique clé minimale de Livraison.

Livraison n'est pas en 2FN car il existe un attribut non clé(AdrF) qui dépend d'unepartie de la clé(NF).

La relation Livraison peut être décomposé en 2 relation de 2FN :

Produit(<u>NP,NF</u>,Prix) et Fournisseur(<u>NF</u>,AdrF,TelF).

13.4 La troisième forme normale (3FN)

Soit R une relation. On dit que R est en 2FN, s :

1. R est en 2FN ;
2. Les attributs non clé (n'appartiennent à aucune clé) de R, ne dépendent pas d'un autre attribut non clé.

<u>Exemple1</u>

Soit la relation R(A,B,C) avec F={A→B, B→C}. R est-elle en 3FN ?
On va vérifier si R est en 2FN : la clé minimale de R est A. Oui R est en 2FN.
Non R n'est pas en 3FN, car l'attribut non clé C dépend d'un autre attribut non clé (B).

<u>Exemple2</u>

Soit la relation R(A,B,C) avec F={A→B, BC→A}. R est-elle en 3FN ?
On va vérifier si R est en 2FN :
R comporte deux clés minimales : BC et AC.
Si on choisit AC comme Clé primaire, R serait en 1FN mais pas en 2FN.
Si on choisit BC comme Clé primaire, R serait en 1FN, 2FN et en 3FN.

Algorithme de décomposition d'une relation non 3FN en plusieurs relations 3FN

Entrée :R : schéma de relation et F : ensemble de DFs

Sortie : R_i [Attr(R_i) ; F_i] relations de 3FN.

Début

1. Calculer la couverture minimale F' de F.
2. Calculer les clés minimales de R
3. Regrouper les DFs de F'ayant même prémisse, {Gi→A1 ; Gi→A2 ; … ; Gi→An}
4. Former des relations R_i [($\underline{G_i}$,A_1,A_2,…A_n) ; {Gi→A_1, Gi→A_1 , …, Gi→A_n}].
5. Si la clé K de R ne figure dans aucune des relations R_i construite en 4, alors ajouter une relation R_Z[(<u>K</u>), {K→K}].

Fin

<u>Exemple</u>

Soit la relation R(A,B,C,D,E) avec F={A→B, A→D, BC→D, D→E, A→E}. R est-elle en 3FN ? sinon proposer une décomposition en plusieurs relations de3FN

Calculons la couverture minimale de R : F'= {A→B, A→D, BC→D, D→E}
Trouvons la ou les clés minimales de R :
AC ne figure pas sur les prémisses des DF donc AC appartient à toutes les clés min.
$(AC)^+$=Attr(R) ; $A^+ \neq$attr(R) ; $C^+ \neq$attr(R), donc AC est l'unique clé minimale de R.
R n'est pas en 2FN car il existe un attribut non clé qui dépend d'un autre attribut non clés (D→E).
Donc R n'est pas en 3FN également.
On va décomposer R en suivant l'algorithme de décomposition, et on obtient :
R1[(<u>A</u>,B,D) ;{A→B, A→D}] ; R2[(<u>B,C</u>,D) ;{ BC→D}]; R3[(<u>D</u>, E) ;{D→E}]
Comme la clé AC ne figure dans aucune des 3 relations alors on ajoute la relation RZ[(<u>A,C</u>) ; {AC→AC}]

<u>Remarque très importante:</u> l'algorithme de décomposition en relations de 3FN garantit une décomposition sans perte d'information et sans perte de DFs.

13.5 La forme normale de BOYCE-CODD(BCNF)

Une relation est dite en forme normale de BOYCE-CODD(BCNF), si

1. Elle est en 1FN ;
2. Toutes les DFs sont de la forme Clé minimale→ attribut.

Algorithme de décomposition d'une relation non BCNF en plusieurs relations BCNF

Entrée :R : schéma de relation et F : ensemble de DFs

Sortie : D= R_i [Attr(R_i) ; F_i] relations de BCNF

Début

1. Au départ, D vaut {R}
2. Tant qu'il existe un schéma R_i dans D qui n'est pas BCNF, alors
   - Chercher une dépendance non triviale X → Y dans F^+, et {X, Y} inclus dans R_i, telle que X ne soit pas une clef de R_i.
   - Décomposer Ri selon R_{i1}[(X,Y) ; {X→Y}] et R_{i2}[(Attr(Ri)-Y) ; F_i-{X→Y}]
3. S'il existe R_i et R_j dans D avec Attr(R_i) sous-ensemble de Attr(R_j) alors supprimer R_i de D.

Fin

Remarque très importante: l'algorithme de décomposition en relations de BCNF permet de trouver une décomposition sans perte d'information. Il existe en général plusieurs décompositions BCNF d'une relation, et la décomposition résultat de l'algorithme précédent dépend de l'ordre dans lequel on considère les DFs. La décomposition calculée par l'algorithme ne préserve pas nécessairement les DFs.

Exemple1

Soit le schéma relationnel R (A, B, C, D, E) satisfaisant les DFs F= {A B→ C, AB → D, C → A }
Quelle est la FN de R ? proposer une décomposition en BCNF.

Il y a deux clé candidates : AB et CB.
Sion choisit BC comme clé primaire : R non 2FN
Sion choisit AB comme clé primaire : R en 3FN mais pas en BCNF.
Decomposition R1[(A,B,C,D);{ A B→ C, AB → D }] et R2 [(C,A);{ C → A }]

Exemple2

Soit le schéma relationnel R (A, B, C, D, E) satisfaisant les DFs F= {A → B, A → C, CD → E, B → D}

1. Voici une première application de l'algorithme de décomposition :

étape 1 : R(A,B,C,D,E) n'est pas BCNF. Sa seule clef est A. donc R pas BCNF à cause de B → D et CD → E. On considère cette fois la dépendance CD → E et on décompose R en R1(C,D,E) et R2(A,B,C,D) .
étape 2 : R2 n'est pas BCNF à cause de B → D et A clef. On décompose donc R2 en R21(B,D) et R22(A,B,C).
R1(C,D,E) est BCNF (vérifie CD → E), R21(B,D) est BCNF (vérifie B → D), R22(A,B,C) est BCNF (vérifie A → B et A → C)
Cette décomposition est sans perte et préserve les DF. En fait, c'est la décomposition que l'on obtient avec l'algorithme de normalisation en 3NF.

2. Voici une deuxième application de l'algorithme de décomposition :

étape 1 : R(A,B,C,D,E) n'est pas BCNF. Sa seule clef est A. donc R pas BCNF à cause de B → D et CD→E.
On considère la dépendance B → D et on décompose R en R1 (B, D) et R2 (A, B, C, E) .
étape 2 : La clef de R2 est A. R2 (A, B, C, E) n'est pas BCNF. En effet BC → E qui est conséquence de l'ensemble des DF (∈ F^+) n'admet pas une clef en partie gauche. On décompose R2 en R21(B,C,E) et R22(A,B,C).
R1(B,D) est BCNF (vérifie B → D), R21(B,C,E) est BCNF (vérifie BC → E), R22(A,B,C) est BCNF (vérifie A → B et A → C)

La décomposition de R en R1(B,D), R21(B,C,E), R22(A,B,C) est sans perte mais elle ne conserve pas les DF. A partir de {B → D, BC → E, A → B, A → C}, on ne peut pas déduire la dépendance CD → E. Donc la décomposition ne vérifie pas les même DF que R.