

Bases de données avancées

Normalisation

Wiesław Zielonka

Abstract

Ces notes ne sont pas corrigées, mais peut-être vous les trouverez quand même utiles pour préparer l'examen ou projet. Ne pas distribuer.

1 Dépendances fonctionnelles

Un *schéma relationnel* $R(A_1, \dots, A_n)$ est composé d'un nom de la relation R et d'une suite (A_1, \dots, A_n) d'attributs.

Un tuple de type R est une fonction de l'ensemble d'attributs $\{A_1, \dots, A_n\}$ vers les valeurs, nous notons par $t(A_i)$ la valeur de l'attribut A_i du tuple t .

Pour simplifier nous allons ignorer le fait qu'en SQL la valeur $t(A_i)$ d'un attribut A_i peut être null. Une relation r de type R (ou une relation conforme au schéma R) est un ensemble de tuples de type R .

Un schéma d'une base de donnée est un ensemble de schémas relationnels.

D'habitude notre base de données doit satisfaire un certain nombre de contraintes et on considère que les relations sont valides si elles satisfont toutes les contraintes.

Nous allons étudier d'abord des contraintes données par des dépendances fonctionnelles.

1.1 Dépendances fonctionnelles

Pour un tuple t de type $R(A_1, \dots, A_n)$ et un ensemble non vide $X \subset \{A_1, \dots, A_n\}$ d'attributs on note $\pi_X(t)$ la projection de t sur X . Donc $t' = \pi_X(t)$ est un tuple avec les attributs X tel que, pour chaque $A_i \in X$, $t'(A_i) = t(A_i)$.

Soit X, Y deux ensembles non vides d'attributs, $X, Y \subseteq \{A_1, \dots, A_n\}$ de $R(A_1, \dots, A_n)$. Une relation r de type R satisfait une dépendance fonctionnelle $X \rightarrow Y$ si, pour chaque couple de tuples t_1, t_2 de r , si $\pi_X(t_1) = \pi_X(t_2)$ alors $\pi_Y(t_1) = \pi_Y(t_2)$.

Pour alléger la notation nous utiliserons souvent deux conventions:

Si $\{A_{i_1}, \dots, A_{i_k}\}$ et $\{B_{j_1}, \dots, B_{j_m}\}$ deux ensembles d'attributs alors nous écrivons

$$A_{i_1} \dots A_{i_k} \longrightarrow B_{j_1} \dots B_{j_m}$$

pour noter la dépendance $\{A_{i_1}, \dots, A_{i_k}\} \longrightarrow \{B_{j_1}, \dots, B_{j_m}\}$.

De plus si X et Y sont deux ensembles d'attributs alors nous écrirons souvent XY pour désigner l'union $X \cup Y$ de X et Y .

2 Anomalies de mise à jour - décomposition de tables

Supposons que nous avons une relation (ou plutôt un schéma relationnelle)

Emp_Dept(nom_employe, SSN, date_naissance, adresse, num_dept, nom_dept, chef_dept_SSN)

qui regroupe des informations concernant les employés et les départements.
Les numéro de la sécu SSN détermine l'employé ce qui nous donne la dépendance

$SSN \longrightarrow nom_employe, SSN, date_naissance, adresse$

Le numéro de département identifie le département ce qui implique la dépendance

$num_dept \longrightarrow nom_dept, chef_dept_SSN$

Anomalie de suppression. Si on supprime le dernier employé d'un département alors toute information concernant le département disparaît aussi.

Anomalie de modification. Si on change le nom du département 5, par exemple, il faudra le faire de façon uniforme pour tous les employés de ce département.

Anomalie d'insertion Impossible d'insérer un département sans employés, sauf si on décide de mettre NULL pour les données concernant l'employé. Mais dans ce cas si on met le premier employé dans le département l'employé fictif désigné par NULL devrait disparaître.

Les problèmes viennent de la redondance de l'information, l'information concernant le département est inutilement dupliquée pour chaque employé. (Cela fait augmenter aussi la taille de la mémoire, mais c'est moins important.)

Notre problème: **comment décomposer la table pour éviter les problèmes.**

Mais d'abord qu'est-ce que c'est une décomposition d'une relation r ?

2.1 Décompositions de taes

le scé(que) dans rection

Bien sur la clôture F^+ de F contient toutes les dépendances de F . Elle contient aussi toutes les dépendances triviales qui sont satisfaites par chaque relation:

Une dépendance $X \longrightarrow Y$ est **triviale** si $Y \subseteq X$. Évidemment dépendances triviales sont satisfaites par toute relation.

Mais la clôture peut contenir aussi d'autres dépendances plus difficiles à trouver. Par exemple si F contient des dépendances $X \longrightarrow Y$ et $Y \longrightarrow Z$, alors la clôture contient $X \longrightarrow Z$ (si les valeurs de X déterminent de façon unique les valeurs de Y et les valeurs de Y déterminent les valeurs de Z alors les valeurs de X déterminent les valeurs de Z).

Armstrong a proposé trois règles de déductions (**axiomes d'Armstrong**) :

- (1) **la réflexivité** Si $X \supseteq Y$ alors $X \longrightarrow Y$.
- (2) **l'augmentation** Si $X \longrightarrow Y$ alors $XZ \longrightarrow YZ$.
- (3) **la transitivité** Si $X \longrightarrow Y$ et $Y \longrightarrow Z$ alors $X \longrightarrow Z$.

La réflexivité nous donne toutes les dépendances triviales. Il est facile de voir que les trois règles d'Armstrong sont saines, c'est-à-dire si une relation r satisfait un ensemble F de dépendances fonctionnelles alors elle satisfait aussi toutes les dépendances que nous pouvons obtenir à partir de F en appliquant les axiomes d'Armstrong (c'est-à-dire chaque dépendance dérivée à partir de l'ensemble F de dépendances fonctionnelles par les règles d'Armstrong est dans la clôture F^+).

Armstrong a démontré que l'ensemble de trois règles est aussi complet, c'est-à-dire si une dépendance appartient à F^+ alors elle peut être déduite de F en appliquant les règles d'Armstrong.

Cela nous donne un algorithme (pas très pratique) de calcul de F^+ : appliquer les règles d'Armstrong tant que cela produit des nouvelles dépendances.

Nous donnons ci-dessous un autre algorithme de calcul de F^+ .

3.1 Calcul de la clôture.

Soit X un ensemble d'attributs et S un ensemble de DF.

Par définition la **clôture** X^+ de X par rapport à F est un ensemble d'attributs tels que A appartient à X^+ si et seulement si la dépendance $X \longrightarrow A$ est dans la clôture F^+ de F . Autrement, X^+ est composé de tous les attributs dont les valeurs sont déterminées par les valeurs de X si la relation r satisfait les dépendances F .

L'algorithme suivant permet de calculer X^+ :

```
entree : un ensemble  $X$  d'attributs
 $X^+ := X$  //initialisation
repete
     $U := X^+$ 
    pour chaque DF  $Y \longrightarrow Z$  dans  $F$  faire
        si  $X^+ \supseteq Y$  alors  $X^+ := X^+ \cup Z$ 
tant que  $U \neq X^+$ 
```

Définition 2. Deux ensembles de DF sont **équivalents** lorsqu'ils ont la même clôture.

Définition 3. Soit $X \subseteq \{A_1, \dots, A_n\}$ un ensemble d'attributs d'un schéma $R(A_1, \dots, A_n)$. X est une **clé candidate** si

1. $X \longrightarrow A_1 \dots A_n$ et
2. pour chaque $Y \subseteq X$, si $Y \longrightarrow A_1 \dots A_n$ alors $X = Y$.

Donc X est une clé candidat si c'est le plus petit ensemble d'attributs qui implique tous les attributs de R .

Parmi les clés candidates on choisit une seule clé *primaire* qui nous sert de clé pour la table qui implémente le schéma relationnel en SQL, mais cela n'a pas d'incidence sur notre algorithme de normalisation.

Exercice 1. Soit $ABCDEF$ un ensemble d'attributs avec les dépendances:

$$AB \rightarrow C, D \rightarrow C, D \rightarrow E, CE \rightarrow F, E \rightarrow A$$

Calculer D^+ , $(AB)^+$, $(CE)^+$.

Exercice 2. Soit $R(A, B, C, D)$ et F est composé de $\{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$. Trouver les clés candidates. (Réponse CB et AB).

3.2 Dépendance fonctionnelle irréductibles

Définition 4. $X \rightarrow A$ est une dépendance **irréductible** (dans F^+) si A est un attribut n'appartenant pas à X ($X \rightarrow A$ est non triviale) et pour tout l'ensemble Y d'attributs, si $Y \subset X, Y \neq X$ alors $Y \rightarrow A$ n'est pas dans F^+ .

Donc, intuitivement $X \rightarrow A$ est irréductible si l'attribut A n'est pas dans X et il est impossible de supprimer aucun attribut de X .

Les dépendances irréductibles sont faciles à trouver si on a déjà la clôture F^+ .

3.3 Dépendances redondantes et couverture irredondante (minimale)

Soit F un ensemble de dépendances irréductibles.

Une dépendance $X \rightarrow A$ de F est **redondante** si $X \rightarrow A$ appartient à la clôture

$$(F - \{X \rightarrow A\})^+.$$

Autrement, si $X \rightarrow A$ est une dépendance dans F telle que F et $F \setminus \{X \rightarrow A\}$ sont équivalents alors cette dépendance est redondante et elle peut être supprimée de F .

On appelle **couverture irredondante** (ou couverture minimale) d'un ensemble G de dépendances un ensemble F de dépendances tel que

- $G^+ = F^+$ (F et G sont équivalents),
- toutes les dépendances de F sont irréductibles,
- F ne contient pas de dépendances redondantes.

Exemple 1. Soit G :

$$A \rightarrow B, BC \rightarrow D, D \rightarrow E, AC \rightarrow D, AC \rightarrow E$$

Les trois premières dépendances forment une couverture irredondante de G . En effet, si $F = \{A \rightarrow B, BC \rightarrow D, D \rightarrow E\}$ alors F^+ contient aussi les dépendances $AC \rightarrow D$ et $AC \rightarrow E$ (vérifiez!) et si on supprime une dépendance de F alors on la clôture change.

Donc F est un plus petit sous-ensemble de G^+ tel que $F^+ = G^+$.

4 Décompositions sans perte d'information

Définition 5. Soit r une relation de type $R = (A_1, \dots, A_n)$. Soit X, Y deux ensembles non vides d'attributs tels que $X \cup Y = \{A_1, \dots, A_n\}$. On prenant les projections $r_1 = \pi_X(r)$ et $r_2 = \pi_Y(r)$ de r nous obtenons une *décomposition* de r en deux relations r_1 et r_2 . Cette décomposition est **sans perte**¹ si $r_1 \bowtie r_2 = r$, où \bowtie est la jointure naturelle de r_1 et r_2 .

Si la décomposition est sans perte alors nous pouvons oublier la relation r et garder uniquement $\pi_X(r)$ et $\pi_Y(r)$ parce que r est restructurable (par la jointure naturelle) à partir de $\pi_X(r)$ et $\pi_Y(r)$.

Notons que toujours $\pi_X(r) \bowtie \pi_Y(r)$ contient tous les tuples de r , c'est-à-dire toujours r est inclus dans $\pi_X(r) \bowtie \pi_Y(r)$. Mais nous avons un problème si $\pi_X(r) \bowtie \pi_Y(r)$ contient des nouveaux tuples qui n'existaient pas dans la relation r . Dans ce cas la décomposition de r introduira de tuples parasites.

Le théorème de Heath indique quand la décomposition sans perte est possible:

Theorem 6 (Heath). *Soit r une relation de type $R(X)$ une relation qui satisfait toutes les dépendances de F .*

Soit $X = X_1 \cup X_2$. Alors la décomposition $\pi_{X_1}(r), \pi_{X_2}(r)$ est sans perte si

- *soit $X_1 \cap X_2 \rightarrow X_1$,*
- *soit $X_1 \cap X_2 \rightarrow X_2$.*

En pratique on utilise le résultat suivant qui découle immédiatement du théorème de Heath :

Theorem 7. *Théorème de décomposition Soit $R(Z)$ un schéma relationnel avec F comme l'ensemble de DF.*

Soit $X \rightarrow Y$ une DF de F .

Alors la décomposition $R_1(XY) = \pi_{X \cup Y}(R)$, $R_2(Z \setminus Y) = \pi_{Z \setminus Y}(R)$ est sans perte.

Les applications successives de décompositions sans perte donne une décompositions sans perte.

Il ne faut jamais faire de décomposition si elle peut engendrer une perte d'information.

5 Décomposition avec préservation des dépendances fonctionnelles

Intuitivement, F_i ce sont les dépendances valables dans $\pi_{X_i}(R)$ et la condition (1) signifie que avec F_i on est capable de reconstituer F^+ .

Exemple 2. Examinons

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

avec les dépendances $\{A \rightarrow BC, C \rightarrow D, D \rightarrow B\}$. La décomposition $R_1(A, B, C)$, $R_2(A, D)$ est sans perte d'information (pour le voir appliquer le théorème 7) mais si on prend les dépendances valables dans $R_1(A, B, C)$ et dans $R_2(A, D)$ on ne retrouve plus la dépendance $D \rightarrow B$, alors il n'y a pas de préservation de dépendances fonctionnelles.

On préfère les décompositions qui préservent les dépendances fonctionnelles mais parfois on est prêt à sacrifier cette propriété pour pouvoir décomposer une relation. Par contre, répétons encore une fois, on n'admet jamais de décomposition avec perte d'information.

Noter aussi que les deux conditions définie dans ce chapitre et dans le chapitre précédent sont indépendantes, nous pouvons avoir une décomposition sans perte d'information mais qui ne préservent pas de DF (comme dans l'exemple précédent) et aussi nous pouvons avoir des décompositions qui préservent les DF mais qui engendrent une perte d'information.

En particulier notons que la décomposition du théorème 7 garantie que le résultat soit sans perte d'information mais elle ne garantie rien en ce qui concerne la préservation d'indépendances fonctionnelles.

6 Formes normales

6.1 Première forme normale

La relation est en première forme normale si les valeurs d'attributs ne sont pas décomposable.

6.2 Deuxième forme normale 2NF

Un ensemble de dépendances F est clos si F est égal à sa clôture, $F^+ = F$.

Soit $R(A_1, \dots, A_n)$ un schéma relationnel avec un ensemble clos F de DF.

Définition 9. Un attribut A est **primaire** s'il appartient à une clé candidate. Un attribut est non primaire s'il n'est pas primaire.

Le schéma R est en *seconde forme normale* (2NF) si il est 1NF et pour chaque attribut non primaire A et chaque clé candidate X , la dépendance $X \rightarrow A$ est irréductible

Voir la définition 4 pour la notion de dépendances irréductibles..

Notons que si X est une clé candidate alors $X \rightarrow A$ pour chaque attribut A (par la définition de la clé candidate). Si de plus A est non primaire alors A n'appartient pas à X (parce que A n'appartient à aucune clé candidate). Donc la DF $X \rightarrow A$ est non triviale. Donc intuitivement la condition qui définit la seconde forme normale indique que pour déterminer la valeur d'un attribut non primaire A il faut connaître les valeurs de *tous* les attributs d'une clé candidate X .

Exemple 3. Soit

A	B	C
$a1$	$b1$	$c1$
$a2$	$b1$	$c1$

et $AB \rightarrow C, B \rightarrow C$. Cette relation n'est pas 2NF, la seule clé candidate est AB mais l'attribut non primaire C dépend d'une partie de la clé AB , puisque nous avons déjà $B \rightarrow C$.

Nous n'allons pas utiliser 2NF ni maintenant ni à l'examen.

Par contre deux formes normales qui suivent, 3NF et BCNF, sont très importantes.

6.3 Troisième forme normale – 3NF

Soit $R(A_1, \dots, A_n)$ un schéma relationnel avec un ensemble clos F de DF.

Le schéma R est en troisième forme normale (3NF) si pour chaque dépendance irréductible $X \rightarrow A$ où A n'est pas primaire, X est une clé candidate.

Autrement, les attributs non primaires dépendent uniquement de clés candidates.

Notez que si R est 3NF alors R est 2NF.

Exemple 4. $R(A, B, C)$ avec $A \rightarrow BC, B \rightarrow C$. La seule clé candidate c'est $\{A\}$. Comme la clé est composée d'un seul attribut cette relation est en 2NF.

Mais l'attribut non primaire C dépend de B par $B \rightarrow C$, donc ce n'est pas une relation 3NF.

Le théorème suivant indique pourquoi nous nous sommes intéressés à la troisième forme normale :

Theorem 10. *Toute relation R admet une décomposition **sans perte d'information** et **avec la préservation des dépendances fonctionnelles** en relations R_1, \dots, R_k qui sont toutes en troisième forme normale.*

6.3.1 Algorithme de normalisation en 3NF

Nous donnons ici un algorithme qui décompose une relation en relations 3NF. Cette décomposition est sans perte d'information et elle préserve les dépendances fonctionnelles.

Entrée: $R(X)$ et un ensemble F de DF.

Sortie: Un ensemble R_1, \dots, R_n des relations en 3NF.

- (1) rechercher une couverture irrédundante G de F ,
- (2) partitionner G en G_1, \dots, G_k tels que les DF du même groupe aient la même partie gauche,
- (3) construire les projections $R_i = \pi_{X_i}(R)$ où X_i les attributs qui apparaissent dans G_i , $i = 1, \dots, k$,
- (4) si aucune de clés candidates ne figure pas dans une des relations R_i alors il est nécessaire de rajouter une relation dont les attributs constituent une clé candidate.

Exemple 5. $R(A, B, C, D, E)$ et les dépendances

$$A \rightarrow B, \quad A \rightarrow C, \quad CD \rightarrow E, \quad B \rightarrow D.$$

Ces DF forment déjà une couverture irrédundante, il est impossible d'enlever une de ces dépendances. Il y a trois groupes de dépendances avec la même partie gauche :

$$\begin{aligned} &\{A \rightarrow B, \quad A \rightarrow C\} \\ &\{CD \rightarrow E\} \\ &\{B \rightarrow D\} \end{aligned}$$

Donc l'algorithme donne la décomposition en trois relations $R_1(A, B, C), R_2(C, D, E), R_3(B, D)$. Noter que A est une clé candidate de R et elle est dans R_1 donc le dernier pas (4) de l'algorithme ne s'applique pas.

6.4 Forme normale Boyce-Codd (BCNF)

Un schéma relationnel $R(A_1, \dots, A_n)$ avec un ensemble F de DF est en forme normale de Boyce-Codd (BCNF) si pour chaque DF irréductible $X \longrightarrow A$ dans F^+ , la partie gauche X est une clé candidate.

La condition qui définit BCNF est une simplification de 3NF. BCNF est plus stricte que 3NF, si une relation est BCNF alors elle 3NF.

Exemple 6. $R(A, B, C)$ avec $F = \{AB \longrightarrow C, C \longrightarrow B\}$ est 3NF mais pas BCNF, dans $C \longrightarrow B$ la partie gauche n'est pas une clé candidate.

Theorem 11. *Toute relation admet une décomposition en BCNF sans perte d'information mais parfois au prix de perte de dépendances fonctionnelles.*

6.5 Algorithme de décomposition en BCNF.

Tant qu'il existe une relation $R(Z)$ qui n'est pas en BCNF

- chercher une dépendance non triviale $X \longrightarrow Y$ dans R telle que X ne soit pas une clé candidate et
- décomposer R comme indiqué dans le théorème de décomposition, c'est-à-dire en $R_1(XY)$ et $R_2(Z \setminus Y)$.

On répète cette procédure tant qu'il existent des relations qui ne sont pas BCNF.

A la fin, s'il existe des relations $R_i(X_i)$ et $R_j(X_j)$ dans la décomposition telle que $X_i \subset X_j$ alors on supprime R_i .

7 Exemples

Exemple 7. $R(A, B, C, D, E)$ avec le DF

$$F = \{A \longrightarrow B, \quad A \longrightarrow C, \quad CD \longrightarrow E, \quad B \longrightarrow D\}.$$

Décomposez R en relations BCNF.

Toutes ces dépendances sont non triviales, seule clé candidate A . Calculons les clôtures qui permettent d'obtenir les dépendances non irréductibles :

$$\begin{aligned} A^+ &= ABCDE \\ B^+ &= BD \\ (CD)^+ &= CDE \\ (BC)^+ &= BCDE \end{aligned} \tag{2}$$

Donc A est la seule clé candidate.

Les dépendances fonctionnelles $B \longrightarrow D$ et $CD \longrightarrow E$ violent les conditions BCNF.

Première décomposition. Pour décomposer en BCNF nous pouvons prendre la dépendance $B \longrightarrow D$. L'algorithme de décomposition donne deux relations :

$$R_1(B, D) \quad \text{et} \quad R_2(A, B, C, E).$$

R_1 est déjà BCNF avec une seule DF non triviale $B \longrightarrow D$.

Par contre R_2 n'est pas BCNF, en regardant (2) nous trouvons que les dépendances

$$A \longrightarrow B, \quad A \longrightarrow C, \quad BC \longrightarrow E \tag{3}$$

dont les parties gauche et droite sont dans R_2 sont valides dans R_2 et que toutes les autres dépendances valides dans R_2 sont dans la clôture de (3). Encore une fois A est la seule clé dans R_2 et la dépendance $BC \longrightarrow E$ viole la condition BCNF.

Donc l'algorithme de décomposition nous demande de décomposer R_2 en deux relations $R_{21}(B, C, E)$ et $R_{22}(A, B, C)$. Dans $R_{21}(B, C, E)$ nous avons une dépendance non triviale : $BC \longrightarrow E$. Dans $R_{22}(A, B, C)$ nous avons deux dépendances non triviales : $A \longrightarrow B$ et $A \longrightarrow C$.

Avec ces dépendances les deux relations $R_{21}(B, C, E)$ et $R_{22}(A, B, C)$ sont BCNF.

Donc finalement nous avons décomposé R en trois relation :

$$R_1(B, D) \quad \text{avec} \quad B \longrightarrow D,$$

$$R_{21}(B, C, E) \quad \text{avec} \quad BC \longrightarrow E$$

et

$$R_{22}(A, B, C) \quad \text{avec} \quad A \longrightarrow B \quad \text{et} \quad A \longrightarrow C.$$

Notons que cette décomposition préserve les DF.

Deuxième décomposition On peut aussi commencer avec la DF

$$CD \longrightarrow E \tag{4}$$

qui viole la condition BCNF.

En utilisant l'algorithme de décomposition nous obtenons deux relations

$$R_1(C, D, E) \quad \text{avec la DF} \quad CD \longrightarrow E$$

et

$$R_2(A, B, C, D) \quad \text{avec trois DF :} \quad A \longrightarrow B, A \longrightarrow C, B \longrightarrow D.$$

R_1 est déjà BCNF.

Dans R_2 , A est la seule clé candidate mais $B \longrightarrow D$ viole BCNF. On décompose,

$$R_{21}(B, D) \quad \text{avec} \quad B \longrightarrow D$$

et

$$R_{22}(A, B, C) \quad \text{avec} \quad A \longrightarrow B \quad \text{et} \quad A \longrightarrow C.$$

Donc à la fin on obtient la même décomposition que précédemment.

Exemple 8. Soit les DF $A \longrightarrow BC, B \longrightarrow C, A \longrightarrow B, AB \longrightarrow C, AC \longrightarrow D$. Trouver couverture minimale (irredondante).

On calcule les clôtures :

$$\begin{aligned} A^+ &= ABCD \\ B^+ &= BC \end{aligned} \tag{5}$$

On voit que toutes les autres dépendances peuvent être obtenues à partir de (5).

Cela donne 4 dépendances non triviales $A \longrightarrow C, A \longrightarrow D, A \longrightarrow B, B \longrightarrow C$ mais la première est dans la clôture de trois dernières donc finalement la couverture irredondante est composée de trois dépendances:

$$A \longrightarrow D, \quad A \longrightarrow B, \quad B \longrightarrow C$$