

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

HMIN112M : Théorie de la normalisation

I.Mougenot

UM Faculté des Sciences Département Informatique

2020

Introduction

Conception des bases de données relationnelles

- disposer de schémas dits standardisés
- garantir efficacité et robustesse des schémas
- une sorte de "certification" sur la manière de définir un schéma

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Problèmes posés par un schéma mal construit

Exemple d'une relation unique (dite universelle) nommée ici RU

Monde de la banque

RU(nomAgence, numPrêt, client, montant, chiffreAffaires)

Sémantique associée : Une agence de nom **nomAgence** possède un **chiffre d'affaires** et consent des prêts de **numéros précisés** à des **clients** pour un **montant** déterminé

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Exemples de problèmes

pouvant apparaître sur un schéma mal construit

- répétition de données
- perte d'informations
- et par suite inaptitude du schéma face aux besoins

Intuitivement à partir de l'exemple donné :
recopie du chiffre d'affaires de l'agence à chaque nouveau prêt consenti par l'agence : **répétition**
si l'agence ne consent aucun prêt, aucune donnée concernant l'agence n'est disponible : **perte d'information**

Dépendance fonctionnelle (DF)

Notion fondamentale dans la théorie de la normalisation

- constitue un type de contrainte d'intégrité essentiel
- sous-tend la contrainte de clé primaire
- exprime une dépendance forte entre attributs

Dans l'exemple, et intuitivement, le nom de l'agence détermine son chiffre d'affaires, et que le couple (client, numéro de prêt) détermine le montant du prêt.

nomAgence → chiffreAffaires

client numPrêt → montant

Dépendance fonctionnelle : définition

Soit $\mathcal{R}(A_1, A_2, \dots, A_n)$, et G et $D \subset (A_1, A_2, \dots, A_n)$. On dit que G identifie D , ou que D dépend fonctionnellement de G , et on note $G \rightarrow D$, si pour toute relation R de \mathcal{R} , tous les n -uplets u, v de R ayant même valeur dans G , ont même valeur dans D .

Définition

ssi $(\forall R \text{ instance de } \mathcal{R}) (\forall u, v \in R \text{ tel que } u[G] = v[G] \Rightarrow u[D] = v[D])$

Deux n -uplets de R qui coïncident sur G , coïncident aussi sur D .
 $u[G]$ note la projection de u sur G .

Illustration : une relation en général

G : partie gauche de la DF, D : partie droite de la DF (G et D constitués d'un ou de plusieurs attributs)

...	G	...	D	...
...	XXXXXX	...	YYYYYY	...
...	ZZZZZZ	...	WWWWWWW	...
...	XXXXXX	...	YYYYYY	...
...	ZZZZZZ	...	WWWWWWW	...
...

$G \rightarrow D$, si deux n-uplets partagent la même valeur à gauche, alors ils doivent partager la même valeur pour D (ici $D \rightarrow G$ tient aussi)

Généralités
 Dépendance Fonctionnelle
 Clé primaire
 Illustration
 Ensemble de DFs
 Fermeture transitive
 Fermeture transitive d'un attribut
 Ensembles équivalents
 CIM
 Décomposition
 Formes normales

Exemple Banque ; Une instance possible

NOMAGENCE	NUMPRÊT	CLIENT	MONTANT	CHIFFREAFFAIRES
Montpellier	17	Dupont	1000	90M
Montpellier	23	Durand	2000	90M
Béziers	15	Nestor	1500	85M
Nîmes	20	Adam	500	85M
Béziers	16	Achille	2000	85M
Nîmes	18	Désiré	1500	85M
...

nomAgence → chiffreAffaires et client numPrêt → montant

Remarque : si DG constitue l'ensemble des attributs de \mathcal{R} , si deux n-uplets $u, v \in R$ instance de \mathcal{R} coïncident sur G, alors ils sont égaux. Or les relations sont des ensembles, et les doublons sont interdits

Définition de clé primaire à partir des DFs

Définition 1 : surclé K d'un schéma relationnel

K est surclé de \mathcal{R} sur U, ssi $K \rightarrow U$ est une DF sur \mathcal{R} .

Deux n-uplets distincts ne peuvent avoir la même surclé (sinon la réalisation R de \mathcal{R} contient 2 tuples égaux). U est toujours une surclé (triviale).

Définition 2 : clé K d'un schéma relationnel

K est une clé ssi K est une surclé de \mathcal{R} , et K est minimale

En d'autres termes K est clé ssi $\neg \exists A \in K$ tel que $K - \{A\} \rightarrow U$

Si plusieurs clés candidates : en choisir une et une seule

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Banque : trouver toutes les DFs et définir une clé

NOMAGENCE	NUMPRÊT	CLIENT	MONTANT	CHIFFREAFFAIRES
Montpellier	17	Dupont	1000	90M
Montpellier	23	Durand	2000	90M
Béziers	15	Nestor	1500	85M
Nîmes	20	Adam	500	85M
Béziers	16	Achille	2000	85M
Nîmes	18	Désiré	1500	85M
...

nomAgence → chiffreAffaires

client numPrêt → montant nomAgence

client numPrêt : clé

Pbs posés : une agence se crée et l'information n'est pas consignée.

Une agence change de CA, il faut passer par de la recopie

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Trouver une bonne décomposition

Dans l'exemple, chiffresAffaires dépend indirectement de la clé, ce qui est la source des répétitions

Utiliser les DFs pour décomposer (les garder toutes)

Exemple de décomposition

Agence(nomAgence, chiffreAffaires) avec $F_{\text{Agence}} = \{\text{nomAgence} \rightarrow \text{chiffreAffaires}\}$

Prêt(numPrêt, client, montant, nomAgence) avec $F_{\text{Pret}} = \{\text{numPrêt client} \rightarrow \text{montant nomAgence}\}$

Généralités
 Dépendance Fonctionnelle
 Clé primaire
 Illustration
 Ensemble de DFs
 Fermeture transitive
 Fermeture transitive d'un attribut
 Ensembles équivalents
 CIM
 Décomposition
 Formes normales

En extension

Agence en extension

NOMAGENCE	CHIFFREAFFAIRES
Montpellier	90M
Nîmes	85M
Béziers	85M

Prêt en extension

NUMPRÊT	CLIENT	MONTANT	NOMAGENCE
17	Dupont	1000	Montpellier
23	Durand	2000	Montpellier
15	Nestor	1500	Béziers
20	Adam	500	Nîmes
16	Achille	2000	Béziers
18	Désiré	1500	Nîmes

Reconstruire RU : Agence \bowtie Prêt

Autre décomposition

"Mauvaise" décomposition

Certaines des DFs sont perdues ($\text{numPrêt client} \rightarrow \text{montant et numPrêt client} \rightarrow \text{nomAgence}$)

Décomposition avec perte d'information

$\mathcal{R}_1(\text{nomAgence, numPrêt, montant})$ $\mathcal{R}_2(\text{client, montant})$
 $\mathcal{R}_3(\text{nomAgence, chiffreAffaires})$

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Exemple illustré de la perte d'information

Une requête parmi d'autres

Décomposition avec perte d'information

$\Pi_{nomAgence}(\sigma_{client='Achille'}(R2) \bowtie R1)$

La jointure se fait sur montant et comme Durand a aussi un prêt de 2000 euros mais dans une autre agence. Le résultat donnera Montpellier et Béziers, alors que la réponse exacte avec RU aurait été Béziers.

Equivalence entre ensembles de DFs

Trouver l'ensemble de DFs qui va assurer la meilleure décomposition

- trouver les DFs est la seule étape subjective de la démarche
- selon le modélisateur, l'ensemble de dépendances fonctionnelles peut être différent
- Le problème sera de savoir si les ensembles de DF aboutissent au même schéma normalisé

Conséquence logique et fermeture transitive d'un ensemble de DFs

Etant donné un schéma relationnel $\mathcal{R}(U, F)$, une DF f sera une conséquence logique de F , si f est vérifiée pour toute réalisation R de \mathcal{R} .

Etant donné un ensemble F de DFs, d'autres DFs peuvent en être la conséquence logique. Comme les DFs sont en nombre fini (si U est fini), l'ensemble des conséquences existe et s'appelle la **fermeture transitive** de F , notée F^+

Système de dérivation de DFs

règles de dérivation appelées *axiomes d'ARMSTRONG*

Règles principales (soit X, Y et Z des attributs ou ensembles d'attributs)

- R1 **réflexivité** si $Y \subseteq X \subseteq U$ alors $X \rightarrow Y$
- R2 **augmentation** si $X \rightarrow Y$, et Z est un ensemble d'attributs tel que $Z \subseteq U$, alors $XZ \rightarrow YZ$
- R3 **transitivité** si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

L'ensemble de ces règles de dérivation constitue un ensemble de DFs valide (sain et consistant) et complet

Ensemble de DFs valide et complet

- Un système de règles de dérivation est valide (sain ou consistant) si lorsque f est dérivée de F à l'aide du système de règles, f est une conséquence logique de F (le système de règles ne permet donc d'engendrer que des DF valables).
- Un système de règles de dérivation est complet si lorsque f est une conséquence logique de F alors f est dérivable de F en utilisant ce système de règles

On utilise aussi des conséquences de ces règles qui sont utiles en pratique lors des manipulations de DFs

Autres règles pouvant être déduites

Axiomes d'Armstrong additionnels (soit X, Y, W et Z des attributs ou ensembles d'attributs)

- R4 **pseudo-transitivité** si $X \rightarrow Y$ et $WY \rightarrow Z$, alors $WX \rightarrow Z$
- R5 **union** si $X \rightarrow Y$ et $X \rightarrow Z$, alors $X \rightarrow YZ$
- R6 **décomposition** si $X \rightarrow YZ$, alors $X \rightarrow Y$ et $X \rightarrow Z$

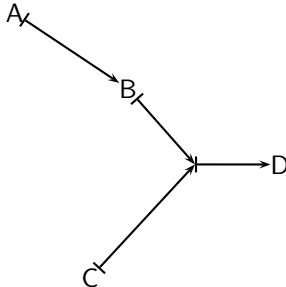
Fermeture transitive d'un attribut (ou ensemble d'attributs), noté X

L'ensemble des attributs directement atteignables à partir de X par rapport à F est appelé fermeture transitive des attributs de X par rapport à F et notée X^+ .

Exemple

Soit $R(ABCD)$ soumis à $F = \{A \rightarrow B, BC \rightarrow D\}$

Fermeture transitive et construction d'un hypergraphe



Exemple : calculs à partir de l'hypergraphe

L'hypergraphe permet de naviguer d'un attribut à l'autre et de calculer les fermetures transitives d'attributs ou d'ensembles d'attributs

Exemples

$$A^+/F = AB$$

$$B^+/F = B$$

$$C^+/F = C$$

$$D^+/F = D$$

$$BC^+/F = BCD$$

$$AC^+/F = ABCD$$

Autre façon de faire

Algorithme simple de calcul de fermeture transitive

Algorithme Fermeture_Trans(X, F, X^+)

{ calcule dans X^+ la fermeture de X par rapport à F ,
ensemble de DF }

$X^+ := X$; $ENU := F$; {ENU ensemble des DF non utilisées}

tant que $\langle \exists \text{ une DF } G \rightarrow D \text{ dans ENU tel que } G \in X^+ \rangle$

faire choisir (et enlever) un arc $G \rightarrow D$ de ENU tel que $G \in X^+$

$X^+ := X^+ \cup \{D\}$;

Exemple : calcul de AC^+/F avec l'algorithme

En mode pas à pas

$AC^+/F = AC$ et $ENU = \{A \rightarrow B, BC \rightarrow D\}$

$AC^+/F = ACB$ et $ENU = \{BC \rightarrow D\}$

$AC^+/F = ACBD$ et $ENU = \{ \}$

Clé

À noter que AC détermine tous les autres attributs et est donc clé, et la seule clé de la relation

Equivalence entre ensembles de DFs

Deux ensembles de dépendances fonctionnelles F et F' sont équivalents ssi

- $F^+ = F'^+$
- $F^+ \in F'^+$ et $F'^+ \in F^+$
- $F \in F'^+$ et $F' \in F^+$
- toute DF de $F \in F'^+$ et toute DF de $F' \in F^+$
- pour toute DF de F , $G \rightarrow D$ alors G^+ par rapport à F' contient D et
- pour toute DF de F' , $G \rightarrow D$ alors G^+ par rapport à F contient D

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Chasser la redondance

Equivalence entre ensembles de DFs

Théorème

Soit F un ensemble de DFs, la dépendance $G \rightarrow D$ est dérivée de F ($G \rightarrow D \in F^+$) ssi $D \in G^+/F$

Simplification

On ramène le calcul de la fermeture transitive de l'ensemble des dépendances fonctionnelles F à celui des fermetures des parties gauches des dépendances fonctionnelles de F .

Ensemble de DFs utiles : calcul d'une CIM

Éliminer la redondance : arriver à une Couverture Irredondante Minimale en 3 étapes

a. Décomposition des DFs non élémentaires

Décomposer $G \rightarrow D_1 D_2 D_3 \dots D_n$ en n DFs **élémentaires**
(un seul attribut à droite) : $G \rightarrow D_1, G \rightarrow D_2, G \rightarrow D_3 \dots$

b. redondance d'une DF ($G \rightarrow D$) par rapport à F

Définition : $G \rightarrow D$ est redondant par rapport à F

ssi $D \in G^+$ par rapport à $(F - \{G \rightarrow D\})$

ssi $(F - \{G \rightarrow D\})^+ = F^+$

Ensemble de DFs utiles : calcul d'une CIM suite

Eliminer la redondance : Dernière étape

c. minimisation des parties gauches de DF

Définition : $A \in G$ est redondant dans $G \rightarrow D$ par rapport à F ssi $D \in \{G - \{A\}\}^+$ par rapport à F

Une DF est dite **minimale** si elle ne contient aucun attribut redondant en partie gauche

Ensemble de DFs utiles : calcul d'une CIM fin

couverture irredondante minimale (CIM) d'un ensemble F de DFs

Il s'agit donc d'un ensemble F' de DFs équivalent à F , qui ne contient aucune DF redondante et dont les parties gauches sont minimales. Pour en obtenir une (il n'y a pas forcément unicité), il faut appliquer les tests a et b dans l'ordre donné

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Sans perte d'information
Sans perte de dépendance

Comment décomposer les schémas

Propriétés d'une décomposition

La normalisation permet de décomposer un schéma relationnel $\mathcal{R}(U, F)$ en sous-schémas relationnels, incorporant l'ensemble des DFs F , réduisant la redondance des données et par suite s'affranchissant des difficultés de mise à jour.

On appelle décomposition d'un schéma \mathcal{R} défini sur U en sous-schémas \mathcal{R}_i définis sur U_i sous ensembles de U ssi $\mathbf{U} = \bigcup \text{des } \mathbf{U}_i$ (recouvrement de l'ensemble des attributs)

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Sans perte d'information
Sans perte de dépendance

Définitions :

Propriétés d'une décomposition

On appelle **décomposition sans perte** d'un schéma \mathcal{R} soumis à un ensemble de dépendances fonctionnelles F en sous-schémas \mathcal{R}_i définis sur $\{U_i\}$ ssi pour toutes les instances R de \mathcal{R} vérifiant F , on a $R = \Pi_{U_1}(R) \bowtie \Pi_{U_2}(R) \bowtie \dots \bowtie \Pi_{U_n}(R)$

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Sans perte d'information
Sans perte de dépendance

Test d'une décomposition :

Algorithme Test_Sans_Perte($\mathcal{R}, \{U_i\}, F$) : booléen

{retourne vrai si $\{U_i\}$ est une décomposition sans perte}

construire un tableau T de n colonnes (n : arité de U) et de k lignes (k : nombre de sous-schémas) : la ligne i correspond au schéma \mathcal{R}_i et la colonne j correspond à l'attribut j de U.

Pour tous les attributs A_j si $A_j \in \mathcal{R}_i$ mettre $T(i,j)=a_j$ sinon $T(i,j)=b_{ij}$

Pour toute DF G D de F **faire**

Pour toutes les lignes qui ont même partie gauche (sur tous les attributs A_j de G) **faire** coïncider les attributs de D de la manière suivante :

- si l'un d'entre eux est de la forme a_j , mettre les autres à a_j
- sinon les mettre égaux à l'un d'entre eux (b_{ij} par exemple)

Résultat := la table T possède au moins une ligne remplie de a_i .

2 théorèmes pour une décomposition sans perte

Théorème 1 Si \mathcal{R} , soumis à F , se décompose en $\{\mathcal{R}_1, \mathcal{R}_2\}$, alors cette décomposition est sans perte par rapport à F ssi $(U_1 \cap U_2) \rightarrow (U_1 - U_2)$ ou $(U_1 \cap U_2) \rightarrow (U_2 - U_1)$
ssi les attributs communs sont clés de l'un ou de l'autre des schémas décomposés

Théorème 2

Si \mathcal{R} soumis à F , se décompose en $\{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_n\}$ sans perte, et si \mathcal{R}_1 soumis à $\Pi_{U_1}(F)$ se décompose en $\{V_1, V_2\}$ sans perte,
 alors $\{V_1, V_2, \mathcal{R}_2, \dots, \mathcal{R}_n\}$ est une décomposition sans perte de \mathcal{R} soumis à F .

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Sans perte d'information
Sans perte de dépendance

Décomposition avec préservation des DF

Définition :

La projection d'un ensemble de DF sur un ensemble d'attributs Z , est l'ensemble des DF de F^+ qui ont tous leurs attributs dans Z .

$$\Pi_Z(F) = \{G \rightarrow D \mid G \rightarrow D \in F^+ \text{ et } GD \in Z\}$$

Une décomposition préserve un ensemble F de DF ssi $\{\bigcup \Pi_{\mathcal{R}_i}(F)\}^+ = F^+$ ssi "l'union des projections est équivalent à l'ensemble initial"

Généralités
Dépendance Fonctionnelle
Clé primaire
Illustration
Ensemble de DFs
Fermeture transitive
Fermeture transitive d'un attribut
Ensembles équivalents
CIM
Décomposition
Formes normales

Sans perte d'information
Sans perte de dépendance

Décomposition avec préservation des DF

Exemples :

Soit \mathcal{R} défini sur ABC, et soumis à $F = \{A \rightarrow B, B \rightarrow C\}$

- a) la décomposition sur AC, BC préserve-t'elle les DF ?

F^+ sans les dépendances triviales = $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

$\Pi_{AC}(F) = \{A \rightarrow C\}, \Pi_{BC}(F) = \{B \rightarrow C\}$

$\Pi_{AC}(F) \cup \Pi_{BC}(F) = \{A \rightarrow C, B \rightarrow C\}$

et $A \rightarrow B$ n'est pas retrouvée donc la décomposition ne préserve pas les DF

- b) au contraire la décomposition AB, BC les préserve.

Formes normales

Exemples :

La théorie de la normalisation s'accompagne d'un ensemble de formes dites normales (FN), à même de garantir la qualité des schémas relationnels

1^{ière} FN

Un schéma relationnel \mathcal{R} défini sur l'ensemble d'attributs U , est en première forme normale si tous les attributs de U sont atomiques (ou monovalués).

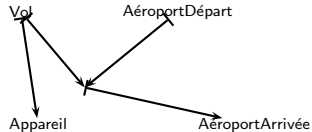
2^{ème} FN

Deuxième forme normale

Un schéma relationnel \mathcal{R} défini sur U , et soumis à F est en 2^{ème} FN, si il est en 1^{ère} FN, et ssi tout attribut n'appartenant pas à une clé dépend **directement**, et **totale**ment d'une clé

Exemple : Soit $\mathcal{R}(\text{Appareil}, \text{Vol}, \text{AéroportDépart}, \text{AéroportArrivée})$ avec $F = \{\text{Vol} \rightarrow \text{Appareil}, \text{Vol AéroportDépart} \rightarrow \text{AéroportArrivée}\}$.

2^{ème} FN



La clé de \mathcal{R} est Vol AéroportDépart.

Appareil dépend d'une partie de la clé, donc le schéma n'est pas en 2^{ème} FN.

3^{ème} FN

Troisième forme normale

Un schéma relationnel \mathcal{R} défini sur U et soumis à F , est en 3^{ème} FN, si il est en 2^{ème} FN et,

- ssi les parties gauches des DFs sont clés
- ou ssi pour les DFs dont la partie gauche n'est pas clé, la partie droite appartient à une clé

En d'autres termes, ssi pour toute DF $G \rightarrow D$ de F ,

- **soit G est une clé**
- **soit D appartient à une clé**

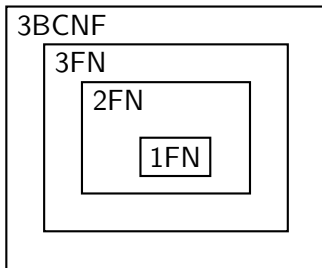
BCNF ou forme normale de Boyce-Codd

BCNF

un schéma de relation \mathcal{R} soumis à F (CIM) est en BCNF, si il est en 3^{ième} FN, et ssi pour toute DF $G \rightarrow D$ de F , **G est une clé**.

Niveau de normalisation attendu pour les schémas relationnels

Imbrication des diverses formes normales



Théorème de décomposition en 3^{ème} FN

Algorithme de Bernstein : conduit à une décomposition en 3BCNF dans le meilleur des cas

- En entrée, un schéma \mathcal{R} soumis à F , et en sortie, $\{\mathcal{R}_i(U_i, F_i)\}$ tel que tout \mathcal{R}_i est en 3FN.
- calculer une CIM (F1)
- trouver les clés de F1
- regrouper les DF de F1 ayant même partie gauche. Fabriquer les schémas de relation maximaux par l'inclusion ensembliste
 $\mathcal{R}_i = \{U_i = \text{Max}(G \cup D); \{G_j \rightarrow D_j ; G_j \cup D_j \in U_i\}\}$
- si aucune clé ne figure dans une des relations produites à l'étape précédente, ajouter un schéma de relation
 $\mathcal{R}_i = \{K ; K \rightarrow K\}$