Transparents de Jeffrey Ullman (Stanford University)
Traduits/adaptés par C. Retoré (U. Bordeaux 1)

Formes Normales et Normalisation

Anomalies
Boyce-Codd BCNF
3e Forme Normale 3NF

Anomalies et Redondance

Types d'anomalies
Exemple d'erreurs de conception

Anomalies

- Bon schéma relationnel:
  - pas de redondance
    - (c.-à-d. plusieurs fois la même information)
  - pas d'anomalies.
    - *Anomalie de mise à jour* : une occurrence d'une information est modifiée et pas les autres
    - *Anomalie de suppression* : une information pertinente est perdue en détruisant un n-uplet.

Exemple de schéma mal conçu

Drinkers(name, addr, beersLiked, manf, favBeer)

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	???	WickedAle	Pete's	???
Spock	Enterprise	Bud	???	Bud

Les données sont redontantes:
les Dépendances Fonctionnelles
name -> addr favBeer
beersLiked -> manf.
déterminent les ???

Erreurs de conception et Anomalies

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	Voyager	WickedAle	Pete's	WickedAle
Spock	Enterprise	Bud	A.B.	Bud

- Anomalie de Mise à jour: si Janeway part pour l'*Intrepid*, pensera-t-on à changer tous les n-uplets?
- Anomalie de Descruction: si personne n'aime Bud, on perd le faut que son fabricant Anheuser-Busch.

Forme Normale de Boyce Codd BCNF

Exemple
Mise en forme normale

Forme Normale de Boyce-Codd BCNF

- Une relation R est dite en *BCNF* ssi pour toute Dépendance Fonctionnelle non triviale (c.-à-d. X ne contient pas A) *X ->A* sur les attributs de R
  X est une clé (pas forcément minimale)

Exemple

Drinkers(name, addr, beersLiked, manf, favBeer)
FD's: name->addr favBeer, beersLiked->manf

- Une seule clé minimale {name, beersLiked}.
- Pour chacune DF:
  la partie droite n'est pas une clé
- *Drinkers* n'est pas en BCNF

(prendre l'une des deux DF Au choix)

Autre Exemple

Beers(name, manf, manfAddr)
FD's: name->manf, manf->manfAddr

- Une seule clé minimale {name} .
- name->manf ne contredit pas BCNF, mais par contre la relation n'est pas BCNF à cause de: manf->manfAddr .

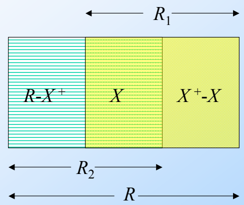
Décomposition en BCNF

- Donnée: une relation R avec des DF F.
- Chercher les DF $X \rightarrow B$ telles que X ne soit pas une clé
  - Si R pas BCNF, il y en a au moins une.
- Calculer X^+.
  - Qui ne contient pas tous les attributs, sinon X serait une clé.

Décomposer R en suivant $X \rightarrow B$

- Remplacer R par deux relations dont les attributs sont:
  1. $R_1 = X^+$.
  2. $R_2 = R - (X^+ - X)$.
- *Projeter* les DF sur ces schémas (attention! c'est la clôture de F, Cl(F), qu'on projette sur R_1 et R_2

Schéma de la Décomposition



Exemple

Drinkers(<u>name</u>, addr, <u>beersLiked</u>, manf, favBeer)
F = name->addr,
name -> favBeer,
beersLiked->manf

- Trouver une DF qui fait que Drinkers n'est pas en BCNF: name->addr.
- Clôture des attributs de gauche: $\{name\}^+ = \{name, addr, favBeer\}$.
- On obtient deux relations:
  1. Drinkers1(<u>name</u>, addr, favBeer)
  2. Drinkers2(<u>name</u>, <u>beersLiked</u>, manf)

Exemple, suite

- Il faut maintenant voir si Drinkers1 et Drinkers2 sont en BCNF
- Projection des DF assez facile pour ce cas particulier -- calcul de la clôture Cl(F)
- Pour Drinkers1(<u>name</u>, addr, favBeer), les DF pertinentes sont name->addr et name->favBeer.
  - {name} est la seule clé et Drinkers1 est en BCNF.

Exemple, suite

- Pour Drinkers2(<u>name</u>, <u>beersLiked</u>, manf), la seule DF est beersLiked->manf, et la seule clé est {name, beersLiked}.
  - Pas BCNF, on recommence.
- $beersLiked^+ = \{beersLiked, manf\}$, on décompose *Drinkers2* en:
  1. Drinkers3(<u>beersLiked</u>, manf)
  2. Drinkers4(<u>name</u>, <u>beersLiked</u>)

Exemple, fin

- Décomposition de *Drinkers* :
  1. Drinkers1(<u>name</u>, addr, favBeer)
  2. Drinkers3(<u>beersLiked</u>, manf)
  3. Drinkers4(<u>name</u>, <u>beersLiked</u>)
- Remarque:
  - *Drinkers1* parle des personnes,
  - *Drinkers3* parle des bières,
  - *Drinkers4* relation entre personnes et bières

3e Forme Normale 3NF

Motivation
Décomposition

3e Forme Normale - Motivation

- Il y a des DF qui posent problème quand on décompose
  $AB \rightarrow C$
  $C \rightarrow B$.
  - Exemple (Paris, Marseille, Lyon, Rennes, …)
    A = adresse,
    B = ville,
    C = code postal.
- IL y a deux clés, $\{A,B\}$ et $\{A,C\}$.
- $C \rightarrow B$ contredit BCNF, il faudrait décomposer en AC, BC.

On ne peut pas garder les DF

- Si décomposition *AC* et *BC* on ne peut pas retrouver *AB ->C* à partir des DF projetées.
- Exemple avec
  A = adresse,
  B = ville,
  C = code postal
  sur le transparent suivant.

Une DF non préservée

Ville	Code postal
Paris	75005
Paris	75006

Adresse	Code postal
80, Bd St-Germain	75005
80, Bd St-Germain	75006

Adresse	Ville	Code postal
80, Bd St-Germain	Paris	75005
80, Bd St-Germain	Paris	75006

Bien que toutes les DF soient vérifiées
dans chacune des tables,
la DF adresse ville -> code_postal n'est pas respectée.

3NF: évite ce problème

- 3e Forme Normale (3NF) assouplit la condition de BCNF pour ne pas décomposer dans ce cas.
- Un attribut est dit *premier* s'il fait partie d'une clé minimale.
- Une relation n'est pas en 3NF ssi on peut trouver une DF *X ->A* telle que *X* n'est pas une clé et A ne fait pas partie d'une clé minimale.

Exemple

- Dans notre exemple avec les DF *AB ->C* et *C ->B*, les clés minimales sont *AB* et *AC*.
- Chaque attribut *A*, *B*, ou *C* est premier.
- Bien que *C ->B* contredise BCNF, le schéma A, B,C est en 3NF.

Décomposition SPI

Test pour SPI
Théorème de Heath

Décomposition SPI Sans Perte d'Information

$R(A_1,\ldots\ldots,A_n)$
Décomposition de R suivant
$X_1,\ldots,X_k$ ensembles d'attributs $\subset \{A_1,\ldots\ldots,A_n\}$
$R_i = \pi_{Xi}(R)$ pour $1 \leq i \leq k$
$\mu(R)=R_1 \ |X|\ldots\ldots|X|\ R_k$
Jointure notée |X|,

a-t-on: $R=\mu(R)$?

Décomposition SPI

Quelques propriétés:

1. $\mu(R)=R_1 \ |X|\ldots\ldots|X|\ R_k \supset R$
2. $\pi_{Xi}(\mu(R))=R_i$
3. $\mu(\mu(R))=\mu(R)$

Démonstration au tableau de 1.

Décomposition SPI

- R(A,B,C,D,E)
- Dépendances fonctionnelles:
  - A -> C
  - B->C
  - C->D
  - DE->C
  - CE->A
- Décomposition:
  - X1=AD
  - X2=AB
  - X3=BE
  - X4=CDE
  - X5=AE

Décomposition SPI

	A	B	C	D	E
X1	A_1	B_{12}	B_{13}	A_4	B_{15}
X2	A_1	A_2	B_{23}	B_{24}	B_{25}
X3	B_{31}	A_2	B_{23}	B_{24}	B_{25}
X4	B_{41}	A_2	B_{33}	B_{34}	A_5
X5	A_1	B_{52}	B_{53}	B_{54}	A_5

Décomposition SPI

- Remplir la table
- Forcer les égalités jusqu'à ce que les DF soient vraies
- Si une ligne est $A_1 \ldots\ldots A_n$ décomposition SPI
- Sinon, il y a perte d'information, c.-à-d. $\mu(R) \not\subset R$

Décomposition SPI

- Si la ligne $A_1 \ldots\ldots A_n$ n'apparaît pas, on a une relation qui satisfait les DFs et $A_1 \ldots\ldots A_n$ est dans la jointure des projections mais pas dans R
- Sinon, il n'y a pas de perte d'information, c.-à-d. $\mu(R) = R$ (explication au tableau)

Décomposition SPI

- Théorème de Heath: une décomposition en X_1 et X_2 est SPI ssi
  - $(X_1 \cap X_2) \to X_1 - X_2$

    ou
  - $(X_1 \cap X_2) \to X_2 - X_1$
- Application du test précédent

Décomposition 3NF SPI et sans perte de DF

Algo de base

Raffinement

Décomposition 3NF

- Calculer une couverture minimale
- Créer une table par DF + une par attribut isolé
  - Le résultat est 3NF
- Si on veut une décomposition SPI ajouter il faut une table avec les attributs d'une clé minimale
  - Découle du test SPI

Décomposition 3NF

- Correction de la décomposition:
- DF préservées, OK
- Si pb avec X->A dans YB (de Y->B)
  - A=B
    X pas clé donc X strictement inclus Y
    X->A peut remplacer Y->B
    couverture pas minimale
  - A≠B
    soit Z clé minimale de YB incluse dans Y
    A pas dans Z car A premier et Z clé de YB
    Z->B et Z strictement inclus dans Y
    couverture pas minimale

Décomposition 3NF

- C->P
- HP->S
- HE->S
- HS->C
- CE->N
- HC->P

Décomposition efficace en BCNF SPI

Le premier algo donné
impose de calculer les projections des DFs,
et donc leur clôture ce qui est coûteux.

On peut faire mieux…

Quelques propriétés

Prop. 1

Si on décompose R en $R_1R_2 \ldots R_k$ SPI
et si on décompose R_1 en S_1 et S_2 SPI
alors la décomposition de R en
$S_1S_2\, R_2 \ldots R_k$ est SPI

Raison: la jointure est associative.

Quelques propriétés

Prop. 2

Toute schéma à deux attributs est BCNF
Si R n'est pas BCNF alors
  il existe des attributs A et B tels que
  (R-AB) -> A

Réciproque fausse:
R(A,B,C)
C->A
C->B
ABC-AB->A et pourtant R est BCNF

Quelques propriétés

Explication Prop. 2

pas BCNF à cause de X-> A

Si pas d'autres attributs que X et A, X clé.

Donc il existe B tq
  R-AB->X (réflexivité)
  X->A
  et donc (R-AB)->X

Quelques propriétés

Prop. 3

SI
- on projette un ensemble de DF F (sa clôture) sur $R_1 \subset R$ et qu'on obtient G
- et si on projette G (sa clôture) sur $R_2 \subset R_1$

ALORS
On obtient la même chose que si on projette F (sa clôture) sur R_2

Algo décomposition BCNF SPI

Z:le schéma à décomposer

Init: Y:=Z
Pour chaque paire d'attributs A B
  si Y-AB->A
  faire Y:=Y-B et C:=A
Si aucune paire ne marche ou si |Y|=2
Y est BCNF
Décomposer Z en Y et Z-C.

Recommencer avec Z-C.

Algo décomposition BCNF SPI

- Cours
- Heure
- Salle
- Prof
- Etudiant
- Note

- C->P
- HP->S
- HE->S
- HS->C
- CE->N

Attention les DFs ne sont pas conservées

Algo décomposition BCNF SPI

Correction de l'algo:

Le schéma extrait est BCNF (Prop. 2)

On décompose Z en XA et Z-A avec X->A
(pas de perte par jointure, th; de Heath)

La combinaison des décomposition SPI est SPI (Prop. 1)

Les X^+ sont calculés avec F (Prop. 3)

Algo décomposition BCNF SPI

Peut décomposer un schéma déjà en BCNF

Identifiant
Nom
Départment
Salaire

I->N
I->D
I->S

Décomposé en IN, ID, IS mais déjà BCNF!

Comparaison 3NF BCNF

Décompositions
- sans perte d'information
- avec préservation des DF

Propriétés de 3NF et de BCNF

- Il y a deux propriétés importantes des décompositions:
  1. *Décomposition SPI: Sans perte d'information* : on peut projeter la relation de départ sur chacine des composantes et reconstruire la relation de départ.
  2. *Préservation des dépendances fonctionnelles* : on peut vérifier dans les relations projetées que les dépendances originales sont préservées.

3NF et BCNF, fin

- On peut toujours mettre en BCNF sans perte d'information.
- On peut toujours mettre en 3NF
  - Sans perte d'information
  - En préservant les Dépendances Fonctionnelles
- Par contre il n'y a pas toujours de forme BCNF sans perte d'information et sans perte de dépendance fonctionnelle
  - Adresse rue ville est un exemple.