

Bases de données avancées

Cours 3 : dépendances fonctionnelles (à partir
d'un cours de Ullman, Stanford Univ.)

Conception de bases de données

- Un bon schéma relationnel doit éviter les redondances d'information :
 - base inutilement grosse
 - anomalie de mise à jour : une occurrence d'information est modifiée et pas les autres
 - anomalie de suppression : une information pertinente est perdue en effaçant un n-uplet

Les dépendances fonctionnelles

- Si R est une relation de schéma $\{A_1:d_1, \dots, A_n:d_n\}$, on dit que R satisfait la dépendance fonctionnelle $B_1 \dots B_m \rightarrow C_1 \dots C_p$ si:
 - pour tout v et w de R prenant les mêmes valeurs pour les attributs $B_1 \dots B_m$, v et w prennent les mêmes valeurs pour les attributs $C_1 \dots C_p$
 - Rappel : $B_1 \dots B_m$ est une super-clé de R si R satisfait $B_1 \dots B_m \rightarrow A_1 \dots A_m$, c'est une clé si $B_1 \dots B_m$ est minimale.

Inférer les dépendances fonctionnelles

Axiomes d'Armstrong

Reflexivité

$$\frac{\{C_1, \dots, C_p\} \subseteq \{B_1, \dots, B_m\}}{B_1 \dots B_m \rightarrow C_1 \dots C_p}$$

Augmentation

$$\frac{B_1 \dots B_m \rightarrow C_1 \dots C_p}{B_1 \dots B_m D_1 \dots D_q \rightarrow C_1 \dots C_p D_1 \dots D_q}$$

Transitivité

$$\frac{B_1 \dots B_m \rightarrow C_1 \dots C_p \quad C_1 \dots C_p \rightarrow D_1 \dots D_q}{B_1 \dots B_m \rightarrow D_1 \dots D_q}$$

Exemple de schéma mal conçu

- Buveur: {Nom, Bar, Potable, Prod, Pref}

Nom	Bar	Potable	Prod	Pref
Jean	F&R	Stella	InBev	Aflieghem
Jean	???	Aflieghem	Heineken int.	???
Paul	Ox. A	Kilkenny	St Fr. Ab	Kilkenny

- Les données sont redondantes, les dépendances fonctionnelles : Nom → Bar, Pref et Potable → Prod déterminent les ???

Erreur de conception et anomalies

Nom	Bar	Potable	Prod	Pref
Jean	F&R	Stella	InBev	Aflieghem
Jean	F&R	Aflieghem	Heineken int.	Aflieghem
Paul	Ox. A	Kilkenny	Diageo	Kilkenny

- Anomalies de mise à jour : si Jean va au Connemara, il faut changer tous les n-uplets le concernant
- Anomalies d'effacement : si personne n'aime la Kilkenny on perd l'info qu'elle est fabriquée par Diageo

Gérer les anomalies

- Pour les anomalies de mises à jour, on peut utiliser le système de contraintes du SGBD en codant les dépendances fonctionnelles, si il le permet (coût supplémentaire à chaque mise à jour)
- Pour les anomalies d'effacement, il n'y a pas de solution en conservant la relation telle qu'elle.

Formes normales

- Les relations sous forme normale permettent d'éviter les anomalies :
 - les mises à jour ne doivent plus que vérifier des contraintes simples (mais la lecture de l'information peut être moins efficace)
 - les effacements ne font perdre que l'information pertinente

Décomposition de relation

- La mise sous forme normale d'une relation R vérifiant un ensemble de dépendances fonctionnelles S revient à décomposer R en un ensemble de relation $R_1, \dots, R_n$. On cherche à avoir les propriétés suivantes :
 - On évite les anomalies
 - $R = R_1 \text{ join } \dots \text{ join } R_n$
 - On peut retrouver les DF, de S , à partir de celles de R_i .
- Remarque on ne peut en général pas satisfaire toutes ces contraintes.

Vérifier la conservation de l'information

- Soit $R : \{A_1, \dots, A_n\}$ vérifiant les dépendances fonctionnelles, S , et vérifions si :
 - $R = \pi_{L_1}(R) \text{ join } \dots \text{ join } \pi_{L_m}(R)$
 - il faut tout d'abord que $L_1 \cup \dots \cup L_m = \{A_1, \dots, A_n\}$

Exemple

- $R : \{A, B, C, D\}$, $L1 = \{A, D\}$, $L2 = \{A, C\}$ et $L3 = \{B, C, D\}$
- DF : $A \rightarrow B$, $B \rightarrow C$, $CD \rightarrow A$

	A	B	C	D
L1	a	b1	c1	d
L2	a	b2	c	d2
L3	a3	b	c	d

Exemple

- $R : \{A, B, C, D\}$, $L1 = \{A, D\}$, $L2 = \{A, C\}$ et $L3 = \{B, C, D\}$
- DF : $A \rightarrow B$, $B \rightarrow C$, $CD \rightarrow A$

	A	B	C	D
L1	a	b1	c1	d
L2	a	b1	c	d2
L3	a3	b	c	d

Exemple

- $R : \{A, B, C, D\}$, $L1 = \{A, D\}$, $L2 = \{A, C\}$ et $L3 = \{B, C, D\}$
- DF : $A \rightarrow B$, $B \rightarrow C$, $CD \rightarrow A$

	A	B	C	D
L1	a	b1	c	d
L2	a	b1	c	d2
L3	a3	b	c	d

Exemple

- $R : \{A, B, C, D\}$, $L1 = \{A, D\}$, $L2 = \{A, C\}$ et $L3 = \{B, C, D\}$
- DF : $A \rightarrow B$, $B \rightarrow C$, $CD \rightarrow A$

	A	B	C	D
L1	a	b1	c	d
L2	a	b1	c	d2
L3	a	b	c	d

Exemple

- $R : \{A, B, C, D\}$, $L1 = \{A, D\}$, $L2 = \{A, C\}$ et $L3 = \{B, C, D\}$
- DF : $A \rightarrow B$, $B \rightarrow C$, $CD \rightarrow A$

	A	B	C	D
L1	a	b1	c	d
L2	a	b1	c	d2
L3	a	b	c	d

Théorème de Heath

- Une décomposition des attributs X d'une relation R en X_1 et X_2 est sans perte d'information ssi
 - ou bien $X_1 \cap X_2 \rightarrow X_1 - X_2$
 - ou bien $X_1 \cap X_2 \rightarrow X_2 - X_1$

Forme normale Boyce Codd (BCNF)

- Une relation R satisfaisant un ensemble de dépendances fonctionnelles S est en BCNF ssi :
 - toute dépendance fonctionnelle non-triviale $X \rightarrow A$ (A n'est pas dans X) que R vérifie est telle que X est une super-clé.

Exemple

- Buveur: {Nom, Bar, Potable, Prod, Pref}
- DF : Nom $\rightarrow$ Bar Pref et Potable $\rightarrow$ Prod
- Clé minimale (ici unique) : {Nom, Potable}
- Pour chaque DF : la partie gauche n'est pas une super-clé
- Buveur n'est pas en BCNF (prendre l'une des deux DF au choix)

Autre exemple

- Biere : {Nom, Prod, Adresse}
- DF : Nom $\rightarrow$ Prod, Prod $\rightarrow$ Adresse
- Une seule clé minimale {Nom}
- Nom $\rightarrow$ Prod ne contredit pas BCNF, mais la relation n'est pas en BCNF à cause de la DF :
Prod $\rightarrow$ Adresse

Décomposition BCNF

- Donnée : une relation R avec des DF S .
- Chercher les DF $X \rightarrow B$ telles que B ne soit pas dans X et X ne soit pas une super-clé (si R n'est pas en BCNF, il y en a au moins une)
- Calculer X_+ : i.e. l'ensemble des attributs qui dépendent fonctionnellement de X . X_+ ne contient pas tous les attributs sinon X serait une super-clé.

Décomposer R suivant

$$X \rightarrow B$$

- Remplacer R d'attributs Q par deux relations
 - $R_1 = \pi_L(R)$ avec $L = X_+$
 - $R_2 = \pi_M(R)$ avec $M = Q - (X_+ - X)$
- Projeter les DF sur ces relations
(Attention à ne pas en oublier)

$$R1 \text{ join } R2 = R$$

- $R1 = \pi_L(R)$ avec $L = X_+$
- $R2 = \pi_M(R)$ avec $M = Q - (X_+ - X)$
- Les attributs communs de $R1$ et $R2$ sont X , de plus tous les attributs de L sont déterminés par X ce qui implique que $R1 \text{ join } R2 = R$

Exemple

- Buveur: {Nom, Bar, Potable, Prod, Pref}
- DF : Nom $\rightarrow$ Bar, Nom $\rightarrow$ Pref, Potable $\rightarrow$ Prod
- La DF Nom $\rightarrow$ Bar implique que Buveur n'est pas en BCNF
- Calcul de $\{\text{Nom}\}_+ = \{\text{Nom}, \text{Bar}, \text{Pref}\}$
- On obtient deux nouvelles relations :
 - Buveur1 : {Nom, Bar, Pref}
 - Buveur2 : {Nom, Potable, Prod} = {Nom, Bar, Potable, Prod, Pref} - {Bar, Pref}

Exemple (suite)

- Pour Buveur1 : {Nom, Bar, Pref}, la projection des DF donne : $\text{Nom} \rightarrow \text{Bar}$, $\text{Nom} \rightarrow \text{Pref}$
- donc {Nom} est la seule clé de Buveur1, donc Buveur1 est en BCNF.

Exemple (suite)

- Buveur2 : {Nom, Potable, Prod} la seule DF est Potable $\rightarrow$ Prod et la seule clé est {Nom, Potable}, donc Buveur2 n'est pas en BCNF
- $\{\text{Potable}\}^+ = \{\text{Potable}, \text{Prod}\}$ on décompose Buveur2 en :
 - Buveur3 : $\{\text{Potable}, \text{Prod}\} = \{\text{Potable}\}^+$
 - Buveur4 : $\{\text{Nom}, \text{Potable}\} = \{\text{Nom}, \text{Potable}, \text{Prod}\} - \{\text{Prod}\}$

Exemple fin

- Décomposition de Buveur :

- Buveur1(Nom, Bar, Pref)

- Buveur3(Pref, Prod)

- Buveur4(Nom, Pref)

- Remarque :

- Buveur1 parle des personnes

- Buveur3 parle des bières

- Buveur4 relation entre personnes et bières

3eme forme normale – motivation

- Il y a des DF qui posent problème quand on les décompose : $AB \rightarrow C$ et $C \rightarrow B$
- On peut prendre A =adresse, B =ville et C =code postal
- Il y a deux clés $\{A, B\}$ et $\{A, C\}$
- $C \rightarrow B$ contredit BNCF, il faudrait décomposer en AC et BC

Les DF ne sont plus représentées

- Si on décompose la relation en AC et BC on ne peut retrouver la DF $AB \rightarrow C$ à partir des DF projetées

Ville	Code
Paris	75005
Paris	75006

Adresse	Code
80 bd St Germain	75005
80 bd St Germain	75006

Ville	Adresse	Code
Paris	80 bd St Germain	75005
Paris	80 bd St Germain	75006

Imposer la cohérence

- Pour retrouver la cohérence, il faut reconstruire une contrainte sur la jointure des relations, ce qui peut rendre la mise à jour coûteuse... et qui contredit un peu les motivations initiales...

Vers un compromis

- La 3ème forme normale (3NF) assouplit la condition de BCNF pour ne pas décomposer dans ce cas.
- Un attribut est dit premier si il fait partie d'une clé (minimale)
- Une relation n'est pas en 3NF si elle vérifie une DF $X \rightarrow A$ tq X n'est pas une super-clé et A n'est pas premier
- Une relation est 3NF ssi pour toute DF $X \rightarrow A$ qu'elle satisfait, X est une super-clé ou A est premier.

Exemple

- Dans notre exemple avec les DF $AB \rightarrow C$ et $C \rightarrow B$ les clés sont $\{A,B\}$ et $\{A,C\}$
- Chaque attribut A, B ou C est premier.
- Bien que $C \rightarrow B$ contredise BCNF la relation $R:\{A,B,C\}$ est en 3NF, car B est un élément d'une clé.

Remarque sur 3NF

- Dans le cas de 3NF, on n'élimine pas les anomalies et il faut donc vérifier plusieurs dépendances fonctionnelles lors d'une mise à jour ; de plus on peut éliminer des informations lors d'effacement.

Mise en 3NF

- Calculer une base minimale G pour les DF de R
- pour chaque $X \rightarrow A$ de G ajouter la relation $\pi_L(R)$ à la décomposition (avec $L = X \cup \{A\}$)
- si aucun schéma de l'une des relations ajouter n'est une clé pour R , ajouter une relation dont le schéma est une clé pour R