

Bases de Données Relationnelles

TD 4 : Dépendances, Formes Normales

2ème année

November 28, 2005

1 Vérification de dépendances

Soit r l'instance de la relation suivante:

r	A	B	C	D	E
	a_1	b_1	c_1	d_1	e_1
	a_1	b_2	c_2	d_2	e_1
	a_2	b_1	c_3	d_3	e_1
	a_2	b_1	c_4	d_3	e_1
	a_3	b_2	c_5	d_1	e_1

Quelles sont les dépendances vérifiées par r :

1. $A \rightarrow D$?
 2. $AB \rightarrow D$?
 3. $C \rightarrow BDE$?
 4. $E \rightarrow A$?
 5. $A \rightarrow E$?
1. non, les deux premiers tuples ont la même valeur a_1 en colonne A mais des valeurs différentes en colonne D
 2. oui, les deux seuls tuples (le troisième et le quatrième) ayant la même paire de valeurs en colonne A, D ont la même valeur sur la colonne D .
 3. oui, car il n'y a pas deux tuples différents ayant même valeur en C
Il suffit de remarquer que toutes les valeurs de C sont différentes
 4. non à cause par exemple du premier et du troisième tuple.
 5. oui trivialement, une seule valeur étant présente en colonne E .

2 Relation clé/dépendance fonctionnelle

Montrer qu'une instance r satisfait la dépendance $X \rightarrow Y$ si et seulement si X est une superclé de la relation $\Pi_{XY}(r)$.

X est une super clé de la relation $\Pi_{XY}(r)$ si et seulement si X détermine fonctionnellement tous les autres attributs de la relation $\Pi_{XY}(r)$, c'est à dire X détermine fonctionnellement Y

Je leur demande en général de démontrer les deux implications

3 Algorithme de vérification d'une dépendance

Donner un algorithme pour vérifier qu'une instance r satisfait une dépendance fonctionnelle

En fait je leur demande de proposer un algorithme, non une requête SQL mais on peut demander les deux !

Voici deux idées d'algorithmes "standards" :

- En $O(n \log n)$ avec éventuellement une copie de la table (ou du moins des colonnes qui correspondent à X et Y)
 1. Trier la table (ou la copie) suivant X ;
 2. Pour $i = 1$ à n vérifier que si $X_i = X_{i+1}$ alors $Y_i = Y_{i+1}$
- En $O(n)$ mais avec un coût mémoire un peu plus important (20-30%). L'idée consiste à utiliser un algorithme de "Hash-coding" :
 - Stocker toutes les paires (X, Y) en utilisant X pour calculer l'adresse;
 - En cas de collision vérifier que les Y sont identiques (et peut être aussi que les X sont identiques..)

Soit $X \rightarrow Y$, la dépendance fonctionnelle que l'on veut vérifier :

select * from r , r as copie_r where r.X=copie_r.X and r.Y<> copie_r.Y

doit rendre un ensemble vide de tuple.

4 Axiomes

Montrer que les axiomes 3,4 et 5 se déduisent tous des axiomes 1,2 et 6 et de la propriété d'idempotence $XX \equiv X$.

1. Reflexivité : $X \rightarrow X$
2. Augmentation: $X \rightarrow Y \vdash XZ \rightarrow Y$
3. Additivité : $X \rightarrow Y, X \rightarrow Z \vdash X \rightarrow YZ$
4. Projectivité : $X \rightarrow YZ \vdash X \rightarrow Z$
5. Transitivité : $X \rightarrow Y, Y \rightarrow Z \vdash X \rightarrow Z$
6. Pseudo- transitivité : $X \rightarrow Y, YZ \rightarrow W \vdash XZ \rightarrow W$

Pour la correction au tableau j'utilise la notation : $\frac{X \rightarrow Y}{XZ \rightarrow Y}$ qui permet de raisonner dans les "deux sens"

- Additivité : $X \rightarrow Y, X \rightarrow Z \vdash$ (réflexivité) $X \rightarrow Y, X \rightarrow Z, YZ \rightarrow YZ \vdash$ (pseudo transitivité) $X \rightarrow Y, XY \rightarrow YZ \vdash$ (pseudotransitivité) $XX \rightarrow YZ \vdash$ (idempotence) $X \rightarrow YZ$
- Projectivité : $X \rightarrow YZ \vdash$ (réflexivité) $X \rightarrow YZ, Z \rightarrow Z \vdash$ (augmentation) $X \rightarrow YZ, XZ \rightarrow Z \vdash$ (augmentation) $X \rightarrow YZ, XYZ \rightarrow Z \vdash$ (pseudotransitivité) $XX \rightarrow Z \vdash$ (idempotence) $X \rightarrow Z$
- Transitivité : $X \rightarrow Y, Y \rightarrow Z \vdash$ (augmentation) $X \rightarrow Y, XY \rightarrow Z \vdash$ (pseudotransitivité) $XX \rightarrow Z \vdash$ (idempotence) $X \rightarrow Z$

5 Recherche de clés

Considérons la relation $R(A,B,C,D)$ avec les dépendances fonctionnelles : $AB \rightarrow C, C \rightarrow D, D \rightarrow A$,

1. Quelles sont les dépendances non triviales qu'on peut déduire de ces 3 dépendances ?
2. Quelles sont les clés de cette relation?
3. Quelles sont les super clés qui ne sont pas des clés ?
 1. $\{A, B\}^+ = \{C, B\}^+ = \{D, B\}^+ = \{A, B, C, D\}$; fermeture $\{C\} = \{A, C, D\}$;
 2. clés: $\{A, B\}$, $\{D, B\}$ et $\{C, B\}$; on peut noter qu'il suffit d'ajouter à la fermeture de $\{C\}$ l'élément manquant ce qui est l'intuition de l'algorithme de saturation.
 3. tous les sur-ensembles de $\{A, B\}$, $\{D, B\}$ et $\{C, B\}$ construits avec $\{A, B, C, D\}$

6 Formes normales

Considérons la relation $R(A,B,C,D)$ avec les dépendances fonctionnelles : $B \rightarrow C, C \rightarrow D$,

1. Indiquez quelles sont les violations de la 3NF
 2. Donnez une décomposition en 3NF de cette relation
1. fermeture $\{A, B\} = \{A, B, C, D\}$ et $\{A, B\}$ est donc une clé.
D'où: $B \rightarrow C$ viole la 2NF et $C \rightarrow D$ viole la 3NF
 2. décomposition : $\{\underline{A, B}\} \{\underline{B}, C\} \{\underline{C}, D\}$