Bases de Données Relationnelles

TD 3 : Dépendances, Formes Normales

SI3, SI4, MAM4

November 28, 2016

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. $A \rightarrow D$?

---

non, les deux premiers tuples ont la même valeur a_1 en colonne A mais des valeurs différentes en colonne D

---

2. $AB \rightarrow D$?

---

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. $C \rightarrow BDE$?

---

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. $E \rightarrow A$?

---

non à cause par exemple du premier et du troisième tuple.

---

5. $A \rightarrow E$?

---

oui trivialement, une seule valeur étant présente en colonne E.

---

2 Relation clé/dépendance fonctionnelle

Montrer qu'une instance r satisfait la dépendance $X \to Y$ 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

---

3 Algorithme de vérification d'une dépendance

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

---

Voici deux idées d'algorithmes "standards" :

- En $O(nlogn)$ 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 verifier que les Y sont identiques (et peut etre aussi que les X sont identiques..)

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

Règle 3:

$$\dfrac{\dfrac{X \to Z(h_2) \qquad \dfrac{X \to Y(h_1) \qquad \dfrac{}{YZ \to YZ}\,(1)}{XZ \to YZ}\,(6)}{XX \to YZ}\,(6)}{X \to YZ}\,(\textsc{idempotence})$$

Règle 4:

$$\dfrac{\dfrac{X \to YZ(h) \qquad \dfrac{\dfrac{\dfrac{}{Z \to Z}\,(1)}{XZ \to Z}\,(2)}{YXZ \to Z}\,(2)}{XX \to Z}\,(6)}{X \to Z}\,(\textsc{idempotence})$$

Règle 5:

$$\dfrac{\dfrac{X \to Y(h_1) \qquad \dfrac{Y \to Z(h_2)}{XY \to Z}\,(2)}{XX \to Z}\,(6)}{X \to Z}\,(\textsc{idempotence})$$

---

5 Dependances Fonctionnelles

Soit un schéma de Bases de données, $DB = \{R_1, R_2, R_3\}$, où

- R_1 a comme attributs ABC
- R_2 a comme attributs ADE
- R_3 a comme attributs CE

et où l'ensemble des dépendances fonctionnelles est :
$DF = \{AB \to C, C \to E, E \to C, C \to D, AB \to E\}$

Trouver un ensemble DF' équivalent à DF, où chacune des dépendances de DF' a tous ses attributs dans une des relations R_i.

$DF' = \{AB \to C, C \to E, E \to C, E \to D\}$
En effet, $DF \Rightarrow DF'$ car $E \to C, C \to D \vdash E \to D$
D'autre part $DF' \Rightarrow DF$ car $C \to E, E \to D \vdash C \to D$ et $AB \to C, C \to E \vdash AB \to E$

---

6 Décomposition 3NF

On considère une relation R avec

- Attributs ={Vol, Ville_Départ, Ville_Arrivée, Heure_Départ, Heure_Arrivée, Durée, Type_Avion, Capacité_Première_Classe, Capacité_Tourisme Capacité_Totale, Repas}
- Clés :
  - Vol
  - Ville_Départ, Ville_Arrivée, Heure_Départ
  - Ville_Départ, Ville_Arrivée, Heure_Arrivée
- Dépendances :
  - Type_Avion $\to$ Capacité_Première_Classe, Capacité_Tourisme, Capacité_Totale
  - Heure_Départ, Durée $\to$ Repas –
  - Heure_Arrivée, Durée $\to$ Repas
  - Capacité_Première_Classe, Capacité_Tourisme $\to$ Capacité_Totale
  - Capacité_Première_Classe, Capacité_Totale $\to$ Capacité_Tourisme

- Capacité_Tourisme, Capacité_Totale → Capacité_Première_Classe

Trouver un décomposition en forme 3NF de cette relation .

- R_1={Vol, Ville_Départ, Ville_Arrivée, Heure_Départ, Heure_Arrivée, Durée, Type_Avion}
- R_2={Heure_Départ, Durée, Repas}
- R_3={Type_Avion, Capacité_Première_Classe, Capacité_Tourisme }
- R_4={Capacité_Première_Classe, Capacité_Tourisme , Capacité_Totale }

- R_1 a pour clés les trois clés de la relation initiale et il n'y a aucune autre dépendance fonctionnelle
- R_2 a pour clé (H_Départ, Durée) et il n'y a aucune autre dépendance fonctionnelle
- R_3 a pour clé Type_Avion et il n'y a aucune autre dépendance fonctionnelle
- R_4 a pour clés (Capacité_Première_Classe, Capacité_Tourisme), (Capacité_Première_Classe, Capacité_Totale), (Capacité_Tourisme, Capacité_Totale) et il n'y a pas d'autres dépendance fonctionnelle.
