Contraintes d'intégrité et dépendances

Serge Abiteboul

INRIA & ENS Cachan

2014

Organisation

Introduction et décomposition

Les contraintes d'intégrité et les dépendances

Ce sont des connaissances supplémentaires sur le monde réel.

Pourquoi les utiliser ?

1. protection des données
2. performance (indexation des tables)
3. **conception de schéma**

Les dépendances. C'est quoi :

- Une syntaxe pour des formules logiques (p. ex. calcul)
- Une sémantique : états de la bd $\rightarrow$ {Vrai, Faux}

$$\forall x, y, z\, [ami(x, y) \wedge ami(y, z) \Rightarrow ami(x, z)]$$
$$\forall x, y, z\, [personne(x, y, z) \Rightarrow integer(z) \wedge z \geq 0 \wedge z \leq 125]$$

Conception de schéma

- Typiquement de nombreuses façons de choisir un schéma
- Comment choisir ?
- Outils de conception spécialisés
- S'appuient sur les dépendances

Redondances et anomalies

Exemple : personne, enfant, voiture

toto	*lulu*	*bmw*
toto	*lulu*	*2cv*
toto	*zaza*	*bmw*
toto	*zaza*	*2cv*

Redondances et anomalies d'insertion et de suppression

Décomposition - solution miracle

Nom	*Enfant*	*Adresse*
toto	*titi*	*Paris*
toto	*lulu*	*Paris*
toto	*titi*	*Romorantin*
toto	*lulu*	*Romorantin*

Le concepteur repère les anomalies

Il décompose la relation entre

- Nom Enfant
- Nom Adresse

Les problèmes disparaissent

1. plus de redondance
2. plus d'anomalies d'insertion, de suppression, de modification

Et on voudrait que ce soit

- Sans perte d'information
- Sans perte de sémantique/contraintes - on verra plus loin

Conception de schéma : Autre exemple

- Apply(SSN, Nom, Université-demandée, Lycée, Ville-Lycée, Hobby)
- Anomalies : redondance, suppression, insertion
- SSN détermine Nom et Lycée
- Lycée détermine Ville-lycéee...

- Student(SSN, Nom)
- Apply(SSN, Université-demandée)
- HighSchool(SSN, Lycée)
- Located(Lycée, Ville-Lycée)
- Hobbies(SSN, Hobby)

On fait quoi :

- On part d'une relation avec tous les attributs/concepts
- On décompose en plus petites automatiquement
- Pour cela on se base sur ce que l'on sait des propriétés des données : les dépendances
- On continue jusqu'à ce que les relations aient les bonnes propriétés ; qu'elles soient sous formes normales

Décomposition - 2 dépendances utilisées en pratique

1. Dépendance fonctionnelle et Forme Normale Boyce-Codd
2. Dépendance multivaluée et 4ème Forme normale

Dépendances fonctionnelles

Dépendances fonctionnelles

Sur une seule table $R(A_1, \ldots, A_n)$
Dépendance fonctionnelle sur R :

- $X \to Y$ (X détermine Y)
- $X, Y \subseteq \{A_1, \ldots, A_n\}$

Exemple $R = \{Nom, Adresse, SS\#, Enfants\}$

$$\{Nom\} \to \{Adresse,\ SS\#\} \ \ (Nom \to Adresse\ SS\#)$$

R **satisfait** $X \to Y$, $(R \models X \to Y)$ si

$$\forall u, v \in R \ \ (\pi_X(u) = \pi_X(v) \Rightarrow \pi_Y(u) = \pi_Y(v))$$

Exemple

R	A	B	C	D	E	F
	0	1	2	3	5	1
	1	2	0	4	5	2
	0	1	2	3	5	1
	2	0	0	4	6	2
	2	0	0	4	7	2
	1	4	2	4	5	3

$R \not\models A \rightarrow B$

$R \not\models A \rightarrow E$

$R \models AB \rightarrow CD$

$R \not\models AB \rightarrow DE$

$R \models ABE \rightarrow CD$

Clés

Cas particulier des dépendances fonctionnelles
R(ABC) et A est une clé : $A \to ABC$

Note : AB est aussi une clé. On parle parfois de superclé. Alors que A est une clé minimale.

Les clés représentent des réalités du monde que l'on cherche à décrire :

- Deux personnes n'ont pas le même numéro de SS
- Une personne ne peut pas être dans deux endroits à la fois
- Un vin ne peut pas être à la fois rouge et blanc

Observations

- Trivialité : $ABC \to A$ (trivial) ; $AB \to CD$ (non trivial) ; $AB \to AC$ (en partie trivial)
- Couper à droite : Si $AB \to CD$ alors $AB \to C$ et $AB \to D$
- Couper à gauche : Si $AB \to C$ alors $A \to C$?
- Grouper à droite : Si $A \to B$ et $A \to C$ alors $A \to BC$
- Transitivité : : Si $A \to B$ et $B \to C$ alors $A \to C$

Formaliser l'implication

Exemple
Si $R \models AB \rightarrow CD$ alors $R \models ABE \rightarrow CD$
Si $R \models AB \rightarrow CD$, $R \models CD \rightarrow F$ alors $R \models AB \rightarrow F$

Définition : $F \Rightarrow f$ (F *implique* f) si toute relation R, si $R \models F$ alors $R \models f$

Axiomatiser l'implication : Axiomes d'Armstrong

Soit U un ensemble d'attributs

$$
\begin{array}{lll}
\text{Réflexivité,} & Y \subseteq X \subseteq U & \vdash \quad X \rightarrow Y \\
\text{Augmentation,} & X \rightarrow Y, Z \subseteq U & \vdash \quad XZ \rightarrow YZ \\
\text{Transitivité,} & X \rightarrow Y, Y \rightarrow Z & \vdash \quad X \rightarrow Z
\end{array}
$$

Notation $\vdash$: **Symbole d'implication syntaxique**
L'ensembles des DFs que l'on obtient en utilisant ces règles

Théorème $F \vdash f \Leftrightarrow F \Rightarrow f$

Les axiomes – suite

Utiliser les axiomes d'Armstrong

$$F = \{A \to B, B \to C, AC \to E, EB \to F\}$$

Exercice : Trouver des f tels que $F \vdash f$

Faciliter l'emploi des axiomes :

Axiomes complémentaires

$$\begin{array}{c} X \to Y \\ X \to Z \end{array} \quad \vdash \quad X \to YZ$$

$$\begin{array}{c} X \to Y \\ WY \to Z \end{array} \quad \vdash \quad XW \to Z$$

$$\begin{array}{c} X \to Y \\ Z \subseteq Y \end{array} \quad \vdash \quad X \to Z$$

Exercice : prouver les axiomes complémentaires

Quelques remarques simples

$X \to A_1 \dots A_n$ ssi $X \to A_i$ pour tout i
donc il suffit de consider $X \to A$ pour un seul attribut A

Si on veut déterminer que F implique $X \to A$

1. on démarre de $X^+ := X$
2. Pour chaque $Y \to B$ dans F, si $Y \subseteq X^+$ on rajoute B à X^+
3. Quand on ne peut plus appliquer (2), $X \to A$ si $A \in X^+$

Dépendances fonctionnelles - les clés du miracle

Les dépendances fonctionnelles

$R(XYZ)$ est *décomposable sans perte d'information* sur XY, XZ si

$$R = \pi_{XY}(R) \bowtie \pi_{XZ}(R)$$

Décomposition – suite

Théorème[1] Soit $R(XYZ)$ une relation. Si R vérifie $X \rightarrow Y$ alors la décomposition sur (XY, XZ) est s.p.i.

Théorème Soit $R(XYZ)$ une relation satisfaisant un ensemble F de DF. La décomposition sur (XY, XZ) est s.p.i. ssi

$$F \Rightarrow X \rightarrow Y \quad OU \quad F \Rightarrow X \rightarrow Z$$

1. X,Y,Z sont disjoints

Forme normale de Boyce-Codd

But : décomposer pour éviter des anomalies
Définition : Un schéma $(R(U), F)$ est sous-FNBC

$$si \quad \begin{array}{c} F \Rightarrow X \rightarrow A \\ A \notin X \end{array} \quad \text{alors } X \text{ est une clé}$$

Décomposition sous FNBC

- Choisir une DF $X \rightarrow Y$ telle que

$$F \Rightarrow X \rightarrow Y, \quad U = XYZ, \quad X, Y, Z \neq \emptyset, \quad Y \cap Z = \emptyset$$

- Décomposer suivant XY et XZ
- Continuer à décomposer les blocs résultats

Rechercher aussi : pas de perte de dépendances

Problème avec FNBC

$AB \rightarrow C$ et $C \rightarrow B$
Exemple : Adresse, Ville, CodePostal

$C \rightarrow B$ est une violation de BCNF (C n'est pas une clé)

Décomposition sous CB et CA

Oups : on ne peut plus imposer $AB \rightarrow C$
(On ne peut plus la vérifier sur les projections sur CB et CA)

On a inventé la 3NF qui est moins contraignante que BCNF

BCNF

Résultat

- Toujours possible de passer sous FNBC
- Pas toujours sans perte des DF

Conception de schéma

On part d'un ensemble d'attributs

On spécifie les dépendances fonctionnelles

On décompose la relation jusqu'à satisfaire BCNF (ou 3NF)

Dépendances multivaluées

Décomposition

Nom	*Enfant*	*Adresse*
toto	*titi*	*Paris*
toto	*lulu*	*Paris*
toto	*titi*	*Romorantin*
toto	*lulu*	*Romorantin*

Problèmes

1. redondance
2. anomalies d'insertion, de suppression, de modification
3. Les dépendances fonctionnelles ne suffisent pas

Décomposition, 4FN et Dépendances Multivaluées

Exemple : personne, enfant, voiture

toto	*lulu*	*bmw*
toto	*lulu*	*2cv*
toto	*zaza*	*bmw*
toto	*zaza*	*2cv*

Redondances et anomalies : Mais aucune dépendance fonctionnelle

$R(X, Y, Z)$ satisfait la dépendance multivaluée $X \twoheadrightarrow Y$ si pour tout u, v dans R avec $\pi_X(u) = \pi_X(v)$, il existe w tel que

- $\pi_{XY}(w) = \pi_{XY}(u)$
- $\pi_{XZ}(w) = \pi_{XZ}(v)$

Décomposition souhaitable en XY, XZ

Conduit à 4ème Forme Normale

Une relation est en 4FN si pour chaque MVD nontriviale $A \twoheadrightarrow B$, A est une clé.

Observations pour les Dépendances multivaluées

- Si $X \rightarrow Y$ alors $X \twoheadrightarrow Y$

  Si $A \rightarrow B$ alors $A \twoheadrightarrow B$ alors A est une clé
  Donc 4FN $\rightarrow$ FNBC

- Intersection : Si $A \twoheadrightarrow BC$ et $A \twoheadrightarrow CD$ alors $A \twoheadrightarrow C$
- Transitivité : Si $A \twoheadrightarrow B$ et $B \twoheadrightarrow C$ alors $A \twoheadrightarrow C$

Algorithme de décomposition

- Repeat until toutes les relations sont en 4FN :
- Choisir R' avec U $\twoheadrightarrow$ V qui viole 4FN
- Décompose R' en R1(U, V) and R2(U, reste)
- Calculer FDs and MVDs pour R1 et R2

Dépendances plus générales

Dépendances plus générales

Dépendances de plus en plus complexes

1. Equality generating dependencies (EGDs) :

$$\forall x_1, \ldots, x_n (R(u_1) \wedge \ldots \wedge R(u_m) \rightarrow x_i = x_j)$$

   généralisation des dépendances fonctionnelles

2. Tuple generating dependencies (TGDs) :

$$\forall x_1, \ldots, x_n (R(u_1) \wedge \ldots \wedge R(u_m) \rightarrow R(v))$$

   généralisation des dépendances multivaluées

3. Embedded TGDs

$$\forall x_1, \ldots, x_n (R(u_1) \wedge \ldots \wedge R(u_m) \rightarrow \exists y_1, \ldots, y_k (R(v)))$$

   Exemple : si un auteur écrit un livre selon Auteur, ce livre est enregistré dans la relation Livre

$$\forall x_1, x_2 (Auteur(x_1, x_2) \rightarrow \exists y_1, y_2 (Livre(x_2, y_1, y_2)))$$

Une vision algébrique des dépendances

En général : si $\varphi \subseteq \psi$

Exemples

$R(AB) \models A \rightarrow B$ si $\sigma_{B \neq B'}(R \bowtie \rho_{B:B'}(R)) \subseteq \emptyset$

$R(ABC) \models A \twoheadrightarrow B$ si $\pi_{AB}(R) \bowtie \pi_{AC}(R) \subseteq R$

Inclusion dependencies

Mettent en jeu deux relations

$R(U) \subseteq S(V)$

$I \models R(U) \subseteq S(V)$ if $\forall t \in I(R), \exists t' \in I(S), \; t(U) = t'(V)$.

embedded TGD : exemple

- $R(AB), S(CD), R(A) \subseteq S(D)$
- $\forall x, y (R(x, y) \Rightarrow \exists z S(z, x))$

Implication

Théorème L'implication est décidable pour les EGDs et (total) TGDs

- Technique : le *chase*

Théorème L'implication est indécidable pour les EGDs et embedded TGDs

- Déjà indécidable avec dépendances fonctionnelles et d'inclusion seules

Contraintes utilisées en pratique

Contraintes utilisées en pratique

- Valeur non nulle : le SSN ne peut pas être une valeur nulle
- Dépendances domaines : l'attribut taille est un réel
- Dépendances fonctionnelles et clés
- Dépendances multivaluées
- Dépendances d'inclusion
- *Clés étrangères*
- *Triggers*

Clé étrangère

Clé

- utilisée pour identifier uniquement un nuplet
- ne peut pas avoir de valeur nulle
- E.g., SS#

R a une *clé étrangère* qui référence des nuplets de S en utilisant la clé de S.

Exemple :

- Livre(Titre,Année,Editeur) a Titre comme clé
- Auteur(Nom,Titre) a Titre comme clé étrangère
  - Attention : ce n'est pas une clé d'Auteur
  - $\forall x_1, x_2 (Auteur(x_1, x_2) \rightarrow \exists y_1, y_2 (Livre(x_2, y_1, y_2)))$

Clé étrangères (referential integrity)

- problème d'intégrité : que se passe-t-il si je détruis le tuple correspondant à un titre utilisé dans la relation Livre – error
- si je le mets à jour - cela met à jour automatiquement la relation Auteur

Triggers

Une contrainte est une propriété des données

Un trigger est un programme à exécuter quand une condition est satisfaite

- condition : n'importe quelle expression Booléenne (p. ex. en SQL)

Exemple

Pour imposer $A \rightarrow B$, si on insère (a, b') et qu'il existe déjà (a, b), commencer par supprimer (a, b)

Event-Condition-Action

Quand un évènement arrive,
si une condition est vérifié
alors réaliser un programme (par exemple, une mise-à-jour)

- Auteur, Livre et la clé externe
- ```
  CREATE TRIGGER LivreBidon AFTER INSERT ON Auteur
    FOR EACH ROW
    WHEN (NEW.Titre NOT IN (SELECT Titre FROM Livre))
      INSERT INTO Livre(Titre) VALUES(Nouveau.Titre);
  ```

`FOR EACH ROW` : précise de les traiter un après l'autre ; il est aussi possible de les traiter tous ensemble
  ```

Contraintes Plus complexes

Contraintes dynamiques : dépendent de l'évolution

Exemple 1 : Le salaire d'un employé ne peut pas décroitre

Exemple 2 : On ne peut annuler sa commande que si on a pris l'assurance annulation

Exemple 3 : Workflow d'un site de commerce électronique :

- Dans l'étape 2, on ne peut plus rien ajouter à son panier
- Dans l'étape 3, on ne peut pas non plus enlever

Contraintes d'intégrités avec MySQL

- types de données et autres attributs (p. ex. `UNSIGNED`)
- clefs, clefs uniques : dépendances fonctionnelles dont la partie droite est toute la table (force à mettre en BCNF !)
- clefs étrangères

```
ALTER TABLE Annuaire ADD FOREIGN KEY (ville)
  REFERENCES Villes (id)
```

- pas de contraintes plus complexes (`CHECK` dans d'autres SGBD)
- des triggers, mais :
  - droits de superutilisateur nécessaires
  - pas de `WHEN`

Il est toujours possible de vérifier les contraintes d'intégrité « à la main », avec des procédures stockées appelées régulièrement.

Merci