Bases de données avancées

Introduction à la normalisation

Équipe pédagogique BD

https://perso.liris.cnrs.fr/marc.plantevit/doku/doku.php?id=lifbdw2_2018a

Version du 7 octobre 2018

Problème et motivation

Anomalies de mise-à-jour et redondance

Décomposition et pertes d'information

Objectifs

Modéliser

Modéliser consiste à définir un monde abstrait qui coïncide avec les manifestations apparentes du monde réel.

- il s'agit donc de déterminer l'ensemble des attributs, des relations et des contraintes qui constitueront le modèle.

Nous allons voir dans la suite :

- Quelles sont les propriété attendues d'une bonne modélisation ;
- Comment les obtenir.

Contexte

Modélisation

- en intelligence artificielle : représentation des connaissances,
- en bases de données : organisation des données (en relations),
- en génie logiciel : organisation des programmes,
- en mathématiques : formalisation du réel.

Pourquoi tant d'effort ?

- Intérêt pratique, bagage de base d'un informaticien : on pourra s'affranchir de normaliser que quand on *sait* normaliser.
- Importance des données vis-à-vis du code : en général, les données sont plus stables dans le temps que les codes qui les accèdent, il y a donc intérêt à leur porter toute notre attention sur la qualité de leur organisation.

Exemple

Soit $\mathcal{U} = \{id, nom, adresse, cnum, desc, note\}$ un univers décrivant des étudiants et des cours. Soient les deux schémas de BD suivants :

- $R1 = \{Donnees\}$ avec $schema(Donnees) = \mathcal{U}$ [1].
- $R2 = \{Etudiant, Cours, Affectation\}$ avec
  - $schema(Etudiant) = \{id, nom, adresse\}$
  - $schema(Cours) = \{cnum, desc\}$
  - $schema(Affectation) = \{id, cnum, note\}$

Comment évaluer ces deux schémas ?

- Lequel est meilleur ?
- Pourquoi ?
- Selon quels critères ?

1. C'est la façon dont on gèrerait les données dans un tableur

Exemple

Donnees	*id*	*nom*	*adresse*	*cnum*	*desc*	*note*
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Quels sont les problèmes de cette modélisation ?

Exemple

Donnees	*id*	*nom*	*adresse*	*cnum*	*desc*	*note*
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Quels sont les problèmes de cette modélisation ?

L'information est *redondante*.

Anomalie de *modification*

Données	id	nom	adresse	cnum	desc	note
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Anomalie de *modification*

- Une modification sur *une* ligne peut nécessiter des modifications sur *d'autres* lignes.
- Exemple : on souhaite modififer l'adresse de Paul : deux lignes sont impactées.

Anomalie de *suppression*

Données	id	nom	adresse	cnum	desc	note
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B

Anomalie de *suppression*

- Certaines informations dépendent de l'existence *d'autres informations*.
- Exemple : le cours 'CS24' dépend de l'inscription de Paul.

Anomalie d'*insertion*

Données	id	nom	adresse	cnum	desc	note
	124	Jean	Paris	F234	Philo I	A
	456	Emma	Lyon	F234	Philo I	B
	789	Paul	Marseille	M321	Analyse I	C
	124	Jean	Paris	M321	Analyse I	A
	789	Paul	Marseille	CS24	BD I	B
	145	Evariste	Aubenas	???	???	???

Anomalie d'*insertion*

- La possibilité d'enregistrer un tuple implique la connaissance de toutes les informations qui lui sont liées : problème de valeurs manquantes.
- Exemple : soit '145, Evariste, Aubenas' un nouvel étudiant. On ne peut l'insérer que si l'on connait un de ses cours et sa note dans ce cours, à moins de permettre les valeurs nulles.

Comment formaliser tout ça ?

Le moyen qui permet d'éviter ces problèmes est l'étude des dépendances (fonctionnelles, multi-valuées, de jointure...)

Quelques définitions

`http://en.wikipedia.org/wiki/Database_normalization`

Élémentaire (ou *minimale*) une DF $X \to Y$ est *élémentaire* ssi $\forall . X' \subsetneq X \Rightarrow X' \not\to Y$

Directe une DF $X \to Y$ est *directe* ssi $\nexists Z . X \to Z \wedge Z \not\to X \wedge Z \to Y$.

Clé un ensemble d'attributs X est *clé* ssi $\forall A \in R . X \to A$. On dit aussi que la *dépendance* $X \to R$ est *clé*.

Super clé un ensemble d'attributs X est *super clé* ssi $\exists K . K$ est clé et $K \subseteq X$.

Clé candidate (ou *minimale*) un ensemble d'attributs X est *clé candidate* ssi la DF associée $X \to R$ est *élémentaire*.

Clé primaire c'est le *choix d'une clé* parmi les candidates.

Problème et motivation

Anomalies de mise-à-jour et redondance

Décomposition et pertes d'information

- Une *anomalie de mise à jour* à lieu lorsqu'*à la suite d'une modification de la base, des contraintes sémantiques valides se trouvent violées*.
- Des mécanismes de contrôle sont intégrés aux SGBDR pour éviter ce genre de problèmes mais ils supposent :
  - une perte de temps dans la gestion de la base, certains contrôles pouvant être assez lourds ;
  - une implémentation rigoureuse de toutes les contraintes par le concepteur de la base. Sous Oracle, celà passe bien souvent par la mise en place de déclencheurs en PL/SQL.

Compromis

- On fait l'hypothèse suivante le concepteur n'implémente que les clés et les clés étrangères.
- Le contrôle automatique de ces contraintes est peu coûteux par le SGBD, et leur implémentation est toujours intégrée.
- Ainsi, on considère que toute mise à jour *respecte les clés*.

Anomalie de m-à-j

Définition

Une relation r a *une anomalie de mise-à-jour par rapport à* F si $r \models F$ et qu'il est possible d'insérer un tuple t tel que :

- $r \cup \{t\} \models CLE(F)$, où $CLE(F)$ est l'ensemble des clés induites par F.
- $r \cup \{t\} \not\models F$.

Soit le schéma $ETUDIANT(id, nom, ville, CP, dpt.)$ muni de l'ensemble de DF $\{id \rightarrow \{nom, ville, CP\}, \{ville, CP\} \rightarrow dpt.\}$

- La seule clé minimale de la relation est id (Toutes les autres clés sont des sur-ensembles de id).
- Supposons qu'on insère un nouvel étudiant, avec une ville et un CP déjà présent mais un autre département.
- La clé ne sera pas violée (pas de doublon sur id) mais la DF $\{ville, CP\} \rightarrow dpt.$ ne sera plus satisfaite.
- La relation $ETUDIANT$ possède une anomalie de mise à jour.

Redondances

- La notion de *redondance* est une autre façon de considérer les problèmes de mises à jour.
- Elle se définit sur les relations, alors les problèmes de mise à jour portent sur des schémas.

Definition

Une relation r sur R est *redondante* par rapport à un ensemble F de DF sur R ssi :

- $r \models F$ et
- il existe $X \rightarrow A \in F$ et $t_1 \neq t_2 \in r$ tels que $t_1[XA] = t_2[XA]$.

Sur le schéma $ETUDIANT(id, nom, ville, CP, dpt.)$ muni de l'ensemble de DF $\{id \rightarrow \{nom, ville, CP\}, \{ville, CP\} \rightarrow dpt.\}$

ETUDIANT	*id*	*nom*	*ville*	*CP*	*dpt.*
	1	Fagin	Lyon	69003	Rhône
	2	Armstrong	Lyon	69001	Rhône
	3	Bunneman	Clermont	63000	Puy-de-Dôme
	4	Codd	Lyon	69001	Rhône

Cette relation est bien correcte car elle respecte les DF. Néanmoins, elle est *redondante* car il existe un doublon sur $(ville, CP)$: l'information du département de Lyon 1er apparaît deux fois.

Liens entre anomalies de m-à-j et redondances

- On voit que les notions d'anomalie de mise à jour et de redondance sont très liées.
- Elles sont en fait équivalentes, selon le résultat suivant.

Théorème : il y a équivalence entre

- R a une *anomalie de mise à jour* par rapport à F,
- Il existe une relation r sur R *redondante* par rapport à F.

Pour éviter les anomalies

- Le principe est de *décomposer* les relations de telle sorte d'éviter les anomalies ;
- c'est-à-dire de transformer une relation en plusieurs relations

Difficulté

Le risque en décomposant est de perdre de l'information :

- on doit pouvoir retrouver toutes les informations initiales,
- et avoir les même dépendances satisfaites.

Perte d'information

Principe

Il faut que toutes les informations de la base de donnée initiale puissent être retrouvées en effectuant des *jointures* sur les relations issues de la décomposition.

Perte de jointures

- Soit R un *schéma de relation* (c'est à dire un ensemble d'attributs), que l'on décompose en un *schéma de base de données* (un ensemble de relations) $\mathbf{R} = \{R_1, \ldots, R_n\}$.
- $\mathbf{R}$ est sans perte de jointures par rapport à un ensemble F de DF ssi pour toute relation r sur R telle que $r \models F$ on a :

$$r = \pi_{R_1}(r) \bowtie \ldots \bowtie \pi_{R_n}(r)$$

Sur le schéma $ETUDIANT(id, nom, ville, CP, dpt.)$ muni de l'ensemble de DF $\{id \rightarrow \{nom, ville, CP\}, \{ville, CP\} \rightarrow dpt.\}$

Supposons que pour régler le problème de redondance on découpe le schéma *ETUDIANT* en deux relations R_1 et R_2 de façon à obtenir les relations suivantes :

R_1	*id*	*nom*
	1	Fagin
	2	Armstrong
	3	Bunneman
	4	Codd

R_2	*ville*	*CP*	*dpt.*
	Lyon	69003	Rhône
	Lyon	69001	Rhône
	Clermont	63000	Puy-de-Dôme

Peut-on reconstruire *r* (avec une jointure) ?

Perte de dépendances fonctionnelles

- Il ne faut pas que la décomposition « coupe » des DFs,
- ceci conduirait à une *perte sémantique*.
- On va caractériser cette notion de perte avec la notion de projection d'un ensemble de dépendances fonctionnelles.

Projections d'un ensemble de DF

Soit F un ensemble de DF sur R, et S un schéma de relation tel que $S \subseteq R$. *La projection de l'ensemble F sur S* est définie par

$$F[S] = \{X \to Y \mid X \to Y \in F^+ \wedge XY \subseteq S\}$$

La projection sur un *schéma de bases de données* est l'union des projections sur chaque relation du schéma

$$F[\mathbf{R}] = \bigcup \{F[R] \mid R \in \mathbf{R}\}$$

Décomposition qui préserve les dépendances

Soit R un schéma de relation et F un ensemble de DF sur R. Un schéma de relation $\mathbf{R}$ est une *décomposition qui préserve les dépendances* de R par rapport à F ssi :

$$F[\mathbf{R}]^+ = F^+$$

Informellement

Une projection est sans perte de dépendances ssi les DFs que l'on avait avant la décomposition peuvent toutes être retrouvées à partir des DFs encore vérifiées sur les relations décomposées.

Exemple

Soit la relation *Edition* définie sur le schéma $R = \{isbn, titre, editeur, pays\}$ qui décrit des livres et leurs éditeurs et $F = \{isbn \rightarrow \{titre, editeur, pays\};\ editeur \rightarrow pays\}$ l'ensemble des dépendances vérifiées. Soit r l'instance donnée :

isbn	*titre*	*editeur*	*pays*
2-212-09283-0	Bases de données	Eyrolles	France
2-7117-8645-5	Fondements des BD	Vuibert	USA
0-201-70872-8	Databases	Addison Wesley	USA
2-212-09069-2	Internet/Intranet etBD	Eyrolles	France

- Exhibez des redondances et des exemples d'anomalies d'*insertion*, de *m-à-j*, de *suppression* ?
- Est-ce que la décomposition *Livre*(*isbn*, *titre*, *editeur*) et *Edite*(*editeur*, *pays*) préserve l'information *et* les DFs ?

Solution aux anomalies

La solution à ces problèmes consiste à normaliser la relation en cause en la décomposant en plusieurs relations.

- Cette décomposition s'appuie sur les dépendances qui existent entre les attributs de la relation initiale :
  - dépendances fonctionnelles,
  - dépendances multivaluées (généralisent les DFs, voir la 4FN).

Les formes normales permettent de spécifier formellement la notion *intuitive* de bon schéma

- Pour les DFs, plusieurs Formes Normales (FN) de plus en plus restrictives :
  - 1FN, 2FN, 3FN, FN de Boyce-Codd.
- Pour les DMVs, on a la 4FN.
- D'autres encore ont été étudiées au delà.

Quand ne pas normaliser ?

La normalisation n'est pas une obligation on peut vouloir s'en passer :

- Pour retrouver « toutes » les données (originales), il faut calculer des jointures, qui peuvent être coûteuses :
  - elle sont généralement nombreuses car la décomposition est maximale,
  - leur calcul n'est pas toujours performant, en particulier si les index ne sont pas adaptés.
- La normalisation peut être difficile, et donc coûteuse en travail humain surtout pour obtenir des formes normales élevées
- On en a pas nécessairement besoin quand la base n'a pas une très grande durée de vie.

Fin.