

Systèmes de Gestion de Bases de Données

L3 Informatique : parcours ASR, Informatique, MIAGE initial et apprentissage

S. Cerrito

premier semestre 2011-2012

Plan du cours

1. Introduction et notions de base des BD relationnelles;
2. Fondements des langages de requête : Algèbre Relationnelle et Calcul Relationnel à variables n -uplets;
3. SQL
4. Conception de schéma :
 - (a) Contraintes d'intégrité : les dépendances fonctionnelles;
 - (b) La méthode Entité-Association (EA, ou ER);
 - (c) Raffinements du schéma :
 - Anomalies;
 - Décompositions SPI et SPD;
 - Formes Normales
5. Si temps : stockage physique des données (indexes etc.)

Supports de cours en ligne sur :

<http://www.ibisc.univ-evry.fr/~serena/>

Mon adresse e-mail :

Serena.Cerrito@ibisc.univ-evry.fr

1 Introduction

SGBD= **S**ystème de **G**estion d'une **B**ase de **D**onnées

Quelles sont les spécificités d'un SGBD ?

- Très grande quantité de données à gérer, qui doivent être stockées dans plusieurs fichiers, voir plusieurs sites.
- Besoin d'interroger et/ou mettre à jour souvent, rapidement et facilement ces données.
- Besoin d'accès concurrents.
- Besoin de sécurité.
- Besoin important de gérer des pannes éventuelles.

INTRODUCTION, suite

Important : indépendance du niveau “logique” (vision “conceptuelle” des données) par rapport au niveau physique (implémentation), car :

1. Utilisateur d’une BD (base de données) : pas forcément un pro de l’implémentation. Il doit juste comprendre comment les données sont “logiquement” organisées.
2. L’implémentation peut changer, sans que le “schéma” (la “forme conceptuelle”) de la BD change.
3. Modèle logique clair $\Rightarrow$
 - (a) possibilité d’un *langage de requêtes* facile pour l’utilisateur
 - (b) si l’implémentation change, pas besoin d’écrire un nouveau programme pour poser la même question à la base !
4. Idem pour le *langage de mise à jour*.

INTRODUCTION, suite

Historique

- Avant 1970 : BD=fichiers d'enregistrements, “modèles” *réseaux* et *hiérarchique*; pas de vraie indépendance logique/physique.
- En 1970 : modèle *relationnel* (Codd) : vraie indépendance logique/physique.
- Années 80 et 90 : nouveaux modèles :
 - modèle à objets
 - modèle à base de règles (Datalog)
- Fin années 90 : données dites *semi-structurées* (XML).

Ce cours : modèle relationnel, le plus utilisé dans la pratique (même si XML...
Mais voir le cours de BDA du M1 !).

2 Notions essentielles des BD relationnelles

Mots clés :

- Univers U , Attributs $A_1, \dots, A_n$
- Domaine $Dom(A)$ d'un attribut A
- Schéma d'une relation dont le nom est R .
- n -uplet sur un ensemble E d'attributs
- Relation (ou “table”) sur un schéma de relation
- Schéma d'une BD
- Base de données B sur un schéma de base

Un *univers* U est un ensemble fini et non-vidé de noms, dits *attributs*.

Le *domaine* d'un attribut A ($Dom(A)$) est l'ensemble des valeurs possibles associé à A .

Exemple :

$U = \{NomFilm, Realisateur, Acteur, Producteur, NomCinema, Horaire\}$

$Dom(NomFilm) = Dom(Realisateur) = Dom(Acteur) = Dom(Producteur) = Dom(NomCinema) =$ chaînes de caractères.

$Dom(Horaire) = \{h.m \mid h \in [0, \dots, 23], m \in [0, \dots, 59]\}$

Un *schéma* d'une relation dont le nom est R est un sous-ensemble non-vide de l'univers U .

Suite de l'exemple :

- Schéma de la relation $Film = \{NomFilm, Realisateur, Acteur, Producteur\}$
- Schéma de la relation $Projection = \{NomFilm, NomCinema, Horaire\}$

Intuition : Format de deux tables.

Film :

NomFilm	Realisateur	Acteur	Producteur
⋮	⋮	⋮	⋮

Projection :

NomFilm	NomCinema	Horaire
⋮	⋮	⋮

Soit $E = \{A_1, \dots, A_n\}$ le schéma d'une relation. Un n -uplet n sur E est une suite de n éléments de la forme : $v_i : A_i$ où $1 \leq i \leq n$ et $v_i \in Dom(A_i)$.

Si $E' \subset E$, la restriction de n à E se note $n(E')$.

Exemple.

Un n -uplet possible sur le schéma de *Projection* :

$\langle "Bird" : NomFilm, "Gaumont Alesia" : NomCinema, 13.35 : Horaire \rangle$

Sa restriction à $\{NomCinema, NomFilm\}$:

$\langle "Bird" : NomFilm, "Gaumont Alesia" : NomCinema \rangle$.

Si pas de confusion possible, on notera plus simplement :

$\langle "Bird", "Gaumont Alesia", 13.35 \rangle$

$\langle "Bird", "Gaumont Alesia" \rangle$.

Pourquoi, alors, on mentionne les attributs, dans la définition formelle de n -uplet ?

Une *relation* (table) r sur un schéma de relation S est un ensemble fini de n -uplets sur S . On dit aussi : S est le schéma de r .

Exemple.

Film :

NomFilm	Réalisateur	Acteur	Producteur
nf1	r1	a1	p1
nf1	r1	a2	p1
nf2	r2	a1	p2
nf3	r2	a1	p2

Projection :

NomFilm	NomCinema	Horaire
nf1	nc1	h1
nf1	nc2	h2
nf2	nc1	h3
nf3	nc2	h1

Un schéma $\mathcal{S}$ d'une base sur un univers U est un ensemble non-vidé d'expressions de la forme $N(S)$ où S est un schéma de relation et N un nom de relation.

Exemple (on omet les $\{\}$ dans les schémas des relations).

$U =$
 $\{NomFilm, Realisateur, Acteur, Producteur, NomCinema, Horaire, Spectateur\}$

$\mathcal{S} =$
 $\{$
 $Film(NomFilm, Realisateur, Acteur, Producteur),$
 $Projection(NomFilm, NomCinema, Horaire), Aime(Spectateur, NomFilm)$
 $\}$

Schéma de la base = Format des données de la base.

Quel est le format de la base de l'exemple ?

- Une *base de données* B sur un schéma de base $\mathcal{S}$ (avec univers U) est un ensemble de relations (finies) $r_1, \dots, r_n$ où chaque r_i est associée à un nom de relation N_i et est telle que si $N_i(S) \in \mathcal{S}$, alors r_i a S comme schéma. (Si le schéma $\mathcal{S}$ de la base dit que toute relation nommée N_i doit avoir l'ensemble d'attribut S , alors la relation r_i “obeit”...)
- On peut aussi imposer des *contraintes* sur les données. Par exemple : les *dépendances fonctionnelles* (DF, à voir), qui fixent, entre autres, les *clés* des relations (à voir).
- Ces contraintes, dites d'*intégrité*, font aussi partie de la spécification du format des données de la base.

Exemple d'une base.

<i>Film</i>			
NomFilm	Réalisateur	Acteur	Producteur
nf1	r1	a1	p1
nf1	r1	a2	p1
nf2	r2	a1	p2
nf3	r2	a1	p2
nf4	r1	a1	p1

<i>Projection</i>		
NomFilm	NomCinema	Horaire
nf1	nc1	h1
nf1	nc2	h2
nf2	nc1	h3
nf3	nc2	h1

<i>Aime</i>	
NomFilm	Spectateur
nf1	s1
nf1	s2
nf2	s1
nf3	s3

Un ex. de contrainte (qui n'est pas une DF) : *Toute valeur de la colonne NomFilm de Projection doit apparaître aussi dans la colonne NomFilm de Film.*

3 Fondements des Langages de Requête

- Informellement : *Requête sur une base* = question que l'on pose à la base.
- *Langage de requête* = langage permettant d'écrire des requêtes
- Importance d'un langage de requête formel et rigoureux :
 1. Conception de langages commerciaux (SQL etc.)
 2. Evaluation de la puissance d'expression de chaque langage commercial
 3. Possibilité de déterminer ce qu'un langage commercial ne pourra pas exprimer
 4. Notion d'équivalence entre deux expressions de requête $\Rightarrow$ Optimisation "logique" de l'évaluation d'une requête

Langage formel à voir dans ce cours : *algèbre relationnelle*

Mais d'autres formalismes, fondés sur la logique du premier ordre, existent.

Langage commercial SQL : des notions viennent de l'algèbre, d'autres d'un formalisme logique (dit “calcul relationnel”).

3.1 Les opérateurs de l'algèbre relationnelle

- Opérateurs ensemblistes : union ($\cup$), intersection ($\cap$), différence ($\setminus$), produit cartésien ($\times$)
- projection sur un ensemble d'attributs E (π_E), sélection d'un ensemble de n -uplets selon une condition C (σ_C), jointure “naturelle” ($\bowtie$), division ($\div$), renommage (ρ).

Union, intersection, différence. Arguments : 2 relations r et r' de même schéma S . Résultat : une nouvelle relation, encore sur S .

Notation : ici et après, n indique un n -uplet.

$$r \cup r' = \{n \mid n \in r \text{ ou } n \in r'\}$$

$$r \cap r' = \{n \mid n \in r \text{ et } n \in r'\}$$

$$r \setminus r' = \{n \mid n \in r \text{ et } n \notin r'\}$$

Projection π . Arguments : 1 relation r . Résultat : une nouvelle relation dont le schéma est inclus dans celui de r .

S = schéma de r , $E \subseteq S$.

$$\pi_E(r) = \{n(E) \mid n \in r\}$$

Ecriture équivalente :

$$\pi_E(r) = \{m \mid \exists n (n \in r \text{ et } m = n(E))\}$$

Sélection.

- **Condition de Sélection C .**

Atomes : $A_i \text{ op } A_j$ ou $A_i \text{ op } v$ où :

A_i et A_j sont des attributs, $v \in \text{Dom}(A_i)$, $\text{op} \in \{=, \neq, >, <, \geq, \leq\}$.

C est une formule booléenne construite à partir des atomes.

- **Opérateur de Sélection σ_C .** Arguments : 1 relation r . Résultat : une nouvelle relation sur le même schéma que r .

$$\sigma_C(r) = \{n \mid n \in r \text{ et } n \text{ satisfait } C\}$$

- **Produit Cartésien** $\times$. Arguments : 2 relations r et r' , de schémas S et S' , telles que S et S' sont disjoints. Résultat : une nouvelle relation dont le schéma est $S \cup S'$.

$$r \times r' = \{n \text{ sur } S \cup S' \mid n(S) \in r \text{ et } n(S') \in r'\}$$

- **Jointure “naturelle”** $\bowtie$. Arguments : 2 relations r et r' , de schémas S et S' . Résultat : une nouvelle relation dont le schéma est $S \cup S'$.

$$r \bowtie r' = \{n \text{ sur } S \cup S' \mid n(S) \in r \text{ et } n(S') \in r'\}$$

N.B. : Si S et S' sont disjoints, le résultat de $r \bowtie s$ est le même que celui de $r \times s$. Donc : $\times =$ cas particulier de $\bowtie$.

Division. Arguments : 2 relations r et r' , de schémas S et S' tels que $S' \subset S$.

Résultat : une nouvelle relation dont le schéma est $S \setminus S'$.

Notation : si n et x sont 2 n -uplets, notons $n \bowtie x$ l'unique élément de la relation $\{n\} \bowtie \{x\}$.

$$r \div r' = \{n \text{ sur } S \setminus S' \mid n \in \pi_{S \setminus S'}(r) \text{ et } \forall x \in r', n \bowtie x \in r\}$$

Exemple.

AimeLivre

Personne	NomLivre
p1	l1
p2	l2
p1	l2

Livre

NomLivre
l1
l2

“Qui aime tous les livres ?” : $AimeLivre \div Livre$

Renommage. Arguments : une relations r , de schéma S , un attribut $A \in S$ et un nouveau attribut $A' \notin S$. Résultat : une copie la relation r où l'attribut A est renommé en A' .

Schéma de la copie : $(S \setminus \{A\}) \cup \{A'\}$.

Ecriture : $\rho_{A \rightsquigarrow A'}(r)$

Exemple : $\rho_{NomLivre \rightsquigarrow NomLivre2}(AimeLivre) =$

Personne	NomLivre2
p1	l1
p2	l2
p1	l2

“Qui aime au moins deux livres ?” :

$\pi_{Personne}(\sigma_{NomLivre \neq NomLivre2}(AimeLivre \bowtie \rho_{NomLivre \rightsquigarrow NomLivre2}(AimeLivre)))$

Est-il possible de s'en passer de ρ ??

Requêtes

Requête : expression d'un langage L qui, évaluée sur une BD calcule une relation.

Par ex. pour une BD sur le cinéma (schéma déjà vu) :

- requête R1, L =français :

Quels spectateurs aiment au moins deux films différents réalisés par Ken Loach ?

- requête R1, L = langage de l'algèbre relationnelle

$$\pi_{Spe}[\sigma_{NomFi \neq NomFi2}(Aime \bowtie \rho_{NomFi \rightsquigarrow NomFi2}(Aime)) \bowtie \sigma_{Real=K.L.}(Film)]$$

Requêtes et expressions algébriques

U : univers, $\mathcal{S}$: schéma de base sur U

- Expression E de l'algèbre relationnelle = mot construit “proprement” en utilisant les opérateurs de l'algèbre.
- Les expressions de l'algèbre calculent des requêtes : la réponse à la requête E_1 pour une BD est la relation résultat de l'évaluation de l'expression algébrique E_1 sur BD.

Equivalence entre expressions algébriques

- E est équivalent à E' ($E \equiv E'$) ssi E et E' , évaluées sur la même base, calculent toujours la même requête.
- Pour *démontrer* que $E_1 \equiv E_2$ on montre que : qqes soit le n -uplet n , $n \in \text{réponse à } E_1$ ssi $n \in \text{réponse à } E_2$.

Utilisation des définitions des opérateurs de l'algèbre.

Exemple : Si tout attribut de $C \in \text{schéma de } r$, alors

$$\sigma_C(r \bowtie s) \equiv (\sigma_C(r)) \bowtie s.$$

- Pour montrer q'une $\equiv$ est fausse un contre-exemple suffit.

Exemple. Soit S le schéma de r_1 et r_2 , soit $A \in S$. C'est faux que

$$\pi_A(r_1 \cap r_2) \equiv (\pi_A(r_1)) \cap (\pi_A(r_2)).$$

Prendre $S = \{personne, livre\}$, $A = personne$, r_1 et r_2 non-vides mais disjointes.

- Utilité des $\equiv$: *optimisation algébrique*. Par ex., quelle est l'expression la plus coûteuse parmi $\sigma_C(R \bowtie S)$ et $(\sigma_C(R)) \bowtie S$. ?