

Bases de données - Introduction

Serge Abiteboul

INRIA

April 3, 2009

Contact

Serge Abiteboul
I.N.R.I.A., Saclay, Ile de France
Serge.Abiteboul@inria.fr
Google: Serge Abiteboul

Plan du cours

- 1 Introduction
- 2 Le Modèle Relationnel
- 3 La Gestion de Fichiers
- 4 Optimisation de Requêtes
- 5 Transactions et Concurrency
- 6 Pannes, BD Distribuées
- 7 (Les Contraintes d'intégrité)
- 8 (SGBD Déductifs)

Et un peu de Web si on a le temps

Bibliographie

Les bases

Korth H.F., Silberschatz A., Database System Concepts

Plus complet

Ullman J.D., *Principles of Database and Knowledge-Base Systems*, 2 volumes, Computer Science Press

La théorie sous-jacente

Abiteboul S, R. Hull et V. Vianu, *Foundations of Databases*, Addison-Wesley

Les bases en français

Delobel A. et M. Adiba, *Bases de Données et Systèmes Relationnels*, Dunod Informatique

Introduction

Début de l'informatique :
systèmes pour calculer (e.g. équations différentielles)

Tendance actuelle :
gestion de grosses quantités d'information (e.g. banque,
bibliothèque, minitel)

Problème : comment stocker et manipuler ces données ?

DONNES → *BASE DE DONNES*
LOGICIEL → *SGBD*

Intermédiaire entre un être humain et les données

Introduction

Caractéristiques :

- persistance des données (années)
- taille de ces données (Giga octets)
- accès diversifiés (partage, utilisateurs, programmes)
- protection des données

Exemple d'applications :

- Gestion (paye, stock,...)
- Transactionnel (banque, centrale d'achat...)
- Documentation (bibliothèque,...)
- Génie logiciel
- Engineering
- IA

Diversité → complexité

- 1 Utilisations:
 - 1 Utilisateurs: administrateurs, programmeurs, secrétaires...
 - 2 interfaces: langages BD, menus, saisies, rapports...
- 2 Modèles de données: réseau, hiérarchique, relationnel, entité-relation, orienté-objet, modèles sémantiques...
- 3 Supports physiques:
 - 1 Nature: disque (optique), bandes, réseaux...
 - 2 Organisation: index, arbre-B, hachage...
 - 3 Géographie: BD distribuées

Fonctionnalités et architecture

Niveau physique

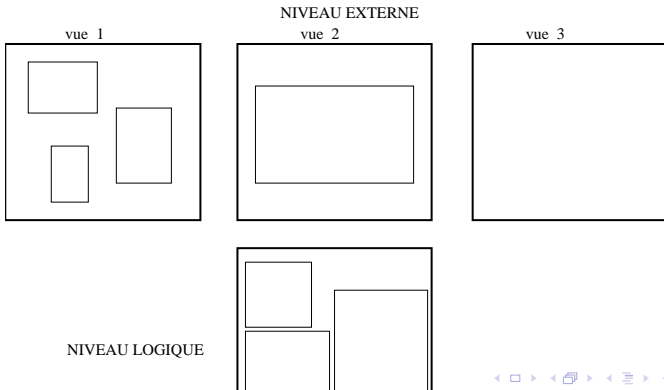
- Gestion de mémoire secondaire données (fichiers), schéma, index
- Gestion concurrence et pannes

Niveau conceptuel

- Compilation (langage de description de données (LDD), langage de manipulation (LMD), etc.)
- Optimisation des requêtes
- Gestion de la confidentialité et intégrité

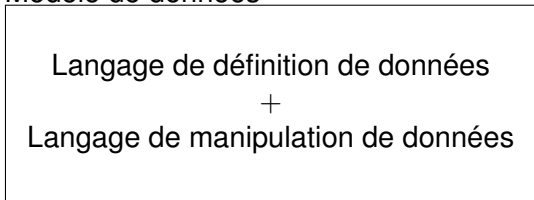
Niveau externe

- environnement de programmation
- interfaces graphiques
- outils d'aides (e.g. conception de schémas)
- débogueurs



Historique

Modèle de données =



60	Cobol	
mi-60	Hiérarchique IMS/IBM	navigationnel
	Réseau	navigationnel
73	Relationnel (Codd)	assertionnel
76	Entité-relation	?
77	Fonctionnel	fonctionnel
Fin 80	Orienté-objets	navig + assert
	Relationnel étendu	assert + navig

Grandes étapes

60-70	Hiérarchique et réseau
80	Succès du relationnel Part croissante de marche Explosion sur micro
90	Nouvelles applis - sgbd-oo Relationnels étendus
00	Web et PHP/mySQL

Idée principale

Séparer la définition logique des données de leur organisation réelle sur support physique.

Architecture ansi-sparc (75) : 3 niveaux

Organisation physique peut être très différente ou pas:
peu de différences dans modèle réseau, beaucoup
modèle relationnel.

Niveau physique et niveau logique

Exemple : un seul fichier étudiant

(nom, prénom, # ss, date naissance, année)

Je veux faire deux types d'accès:

- 1 direct par nom
- 2 séquentiel par # ss

Approche classique

je choisis une implantation :

- 1 un H-code sur les noms
- 2 l'ensemble des enregistrements chaînés par #ss

J'écris mes programmes d'application :

Un beau jour, je modifie l'implantation physique

- H-code sur #ss
- index sur les noms

⇒ Il faut ré-écrire les applications !!!

Physique et logique – suite

Approche BD :

- 1 Je décide de mes structures physique/logique
- 2 J'écris les programmes d'application en utilisant la structure logique

Il faut les **compiler** en programmes exécutables

Les performances dépendront de cette compilation.

C'est l'indépendance des données :

Indépendance physique: On peut modifier l'implantation physique sans modifier le niveau logique.

Indépendance logique: On peut modifier le niveau logique sans toucher à l'implantation.

Question : Pour quel utilisateur?

Niveau logique et niveau externe

Vue du Prof : des cours + pour chaque cours :

- Horaires et salles
- Liste d'étudiants (nom, prénom, note)

Vue de la Paye : un ensemble de profs

(nom, prénom, adresse, indice, nombre d'heures...)

Vue des administrateurs d'emploi du temps : ...

Vue de d'administration (suivi des élèves) : ...

Vues de la base de données

- 1 Aucun ne voit tout
- 2 Chacun utilise sa propre organisation/terminologie

Intégration de ces vues

- 1 On laisse chaque usager avec sa vision du monde
- 2 Intégration dans un **schéma conceptuel**
- 3 Pour chacune de ces vues on définit un mapping
schéma externe → schéma conceptuel

3 aspects importants

- 1 le traitement d'une requête
de sa conception et son exécution
- 2 les connaissances
contraintes d'intégrité et déduction
- 3 le facteur humain
les personnes impliquées

Le traitement d'une requête

- analyse syntaxique
- modification de la requête
basée sur les vues et les contraintes d'intégrité
- optimisation
génération d'un programme optimise a partir de connaissances
(e.g., statistiques) et des structures physiques
- exécution pour obtenir le résultat

Plus de sémantique

Contraintes d'intégrité

- schéma = structure + contraintes
- e.g., nom caractère 15, not null;
age integer between 0 and 120;
débit \leq crédit
- de manière générale, formule logique
- but principal: protéger les données

Autorisation

- grant select on employee to jean
- revoke alter on employee to pierre

Déduction

- Certaines données ne sont pas stockées mais dérivées
- Des règles permettent de les obtenir

Le facteur humain

- Le constructeur du SGBD
- L'administrateur de la base
 - ▶ discute avec les différents utilisateurs
 - ▶ conception des différentes vues
 - ▶ intégration de ces vues en un schéma conceptuel
 - ▶ conception du schéma physique
 - ▶ gère l'évolution de la base (nouveaux besoins, utilisateurs)
 - ▶ Outils à sa disposition: langages de définition de schémas (physique, logique, externe)
- Concepteur et programmeur d'application
à partir : besoins des différents utilisateurs
écrit l'application pour des utilisateurs "naïfs"

Le facteur humain (2)

- Utilisateur sophistiqué: informaticien connaissant langages programmation et langages BD (e.g., SQL)
- Utilisateur naïf: saisie, secrétaire, cadre, non spécialiste

Impact du Web

Protocole d'échange standard

Facilité pour exporter des données

Distribution des données

HTML: manque de structure + mélange données et traitement

XML: plus de structure + feuilles de présentation

- mieux adapté pour les BD

Architectures à la mode

Entrepôts de données

Médiateurs

Portails (d'entreprise)

Modèle hiérarchique

IMS-IBM (64) encore utilisé

Bonnes performances

Ne pas ré-écrire les applications

```
tree realestate
  record regions root
    1 rname char(20)
  record offices parent=regions
    1 city char(20)
    1 oaddr char(30)
  record agents parent=offices
    1 aname char(20)
    1 sales integer
  record emps parent=offices
    1 emp_no integer
    1 ename char(20)
    1 eaddr char(30)
    1 salary integer
  record listings parent=offices
    1 addr char(30)
    1 price integer
  record clients parents=agents
```


Modèle hiérarchique

```
get leftmost agents
  where rname = "Sud" and city = "Aix" and sales > 10000

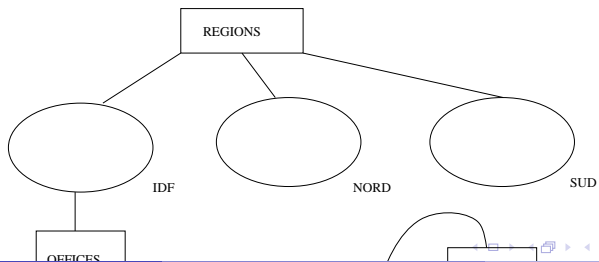
/* les agents avec un chiffre d'affaire plus grand que 10000 */

get leftmost agents
  where sales > 10000
while not fail do
  print agents.aname
  get next agents
    where sales > 10000
endwhile

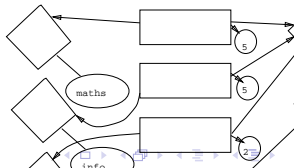
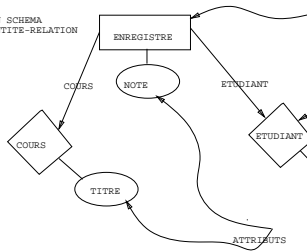
/* les clients du premier Dupond */

get leftmost agents
  where ename = "Dupond"
get next within parent clients
while not fail do
```

Modèle réseau – même esprit



UN SCHEMA
ENTITE-RELATION



Merci