Bases de données avancées
ENSEIRB

Akka Zemmari

LaBRI - Université de Bordeaux

2009-2010

Plan du cours

- Rappels : algébre relationnelle, stockage, etc
- Optimisation
- Transactions

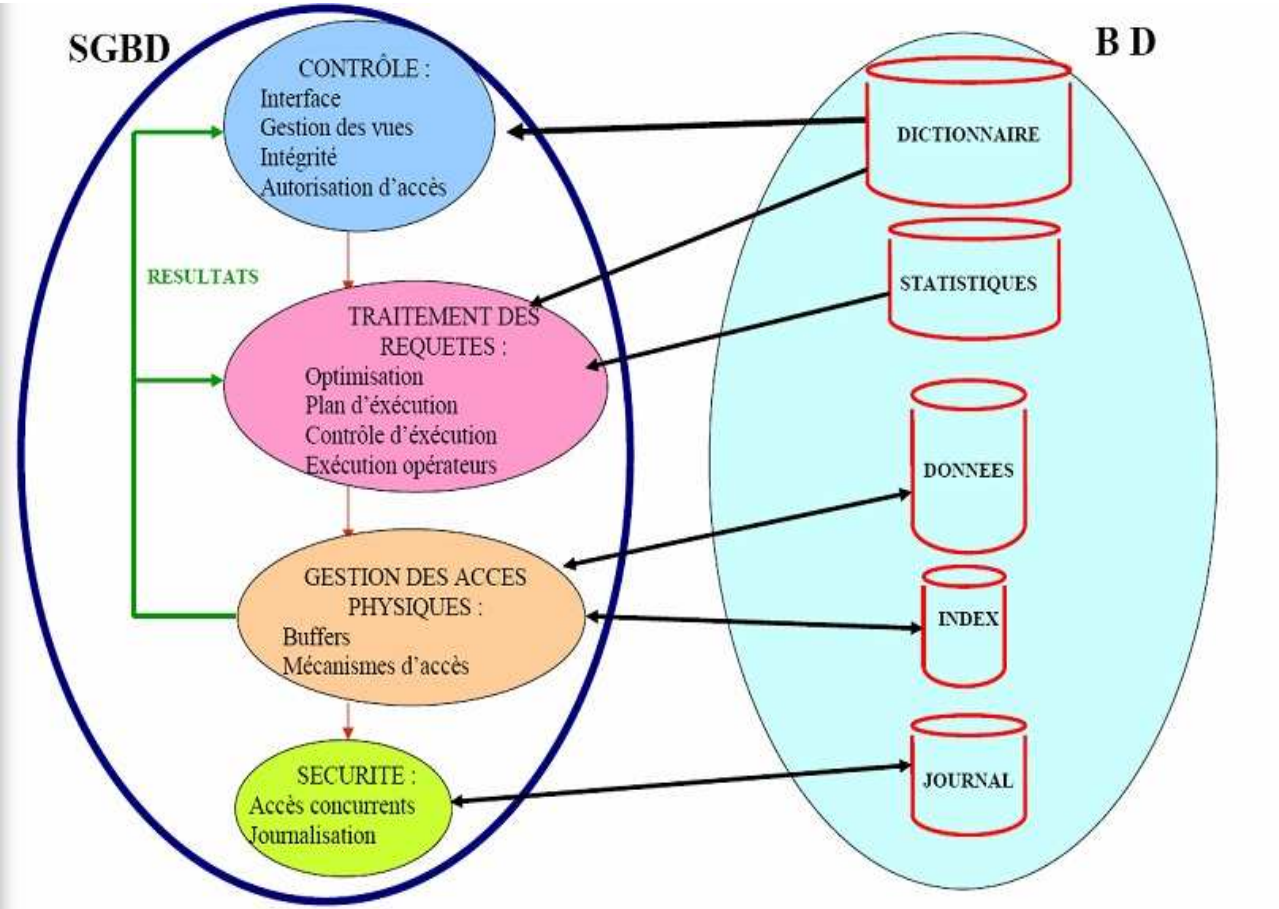
Fonctionnalités d'un SGBD

- Objectifs :
  - Stocker et centraliser des données (BD) et les mettre à disposition des utilisateurs.
  - Manipuler (de manière transparente pour l'utilisateur) des données (SGBD).
- Gestion du stockage : Tailles énormes de données, éviter (dans la limite du possible) les redondances.
- Persistance : Les données survivent aux programmes qui les créent
- Fiabilité : Mécanismes de reprise sur pannes (logiciel ou matériel)
- Sécurité - Confidentialité : Contrôle des utilisateurs et des droits d'accès aux données

Fonctionnalités d'un SGBD (suite)

- Cohérence : Contraintes d'intégrité
- Contrôle de concurrence : Conflits d'accès (notion de transaction)
- Répercussions sur la cohérence
- Interfaces homme-machine : Convivialité + différents types d'utilisateurs
- Distribution : Données stockées sur différents sites
- Optimisation : Transferts MC-MS

Architecture fonctionnelle d'un SGBD



Modèle relationnel

Plan

- Structure d'une BD relationnelle
- Algèbre relationnelle
- Calcul relationnel

Structure d'une BD relationnelle

- Les données sont structurées en **tables** (**relations**)
- Etant donnés les ensembles $A_1, \ldots A_n$, une `relation` r est un sous ensemble de $A_1 \times A_2 \times \cdots \times A_n$.
- Une relation est un `ensemble` de `n-uplets` (ou tuples) de la forme $\langle a_1, \ldots a_n \rangle$ avec $a_i \in A_i$.

Structure d'une BD relationnelle

Exemple : On a trois ensembles : Nom, Num_Cte et Rue avec
Nom = {Durand, Dupont, Dupond}
Num_Cte = {123, 124, 235 , 226}
Rue = {Neuve, vieille, Courte }
Alors

$$\{ \quad \langle \text{Dupont, 123, Neuve} \rangle, \langle \text{Dupont, 124, Neuve} \rangle, \langle \text{Dupond, 235, Neuve} \rangle, \langle \text{Durand, 123, Vieille} \rangle \quad \}$$

est une relation sur Nom $\times$ Num_Cte $\times$ Rue

Schéma de relation

- Une table est une relation (au sens mathématique) qui a un nom
- $A_1, \ldots, A_n$ sont des `attributs`
- $R(A_1, \ldots, A_n)$ est un schéma de relation.
- R est le nom du schéma de la relation.
- On note $Att(R)$ pour désigner l'ensemble des attributs de R
- L'`arité` de R est la cardinalité de $Att(R)$
- Le domaine de A_i (noté $dom(A_i)$) est l'ensemble des valeurs associées à A_i. Cet ensemble peut être fini ou non

Instance de relation

Emp	Nom	Num_Cte	Rue
	Dupont	124	Neuve
	Dupond	235	Neuve
	Durand	123	Vieille

$Att(Emp) = \{Nom, Num\_Cte, Rue\}$
Arité(Emp) $= 3$
$Dom(Num\_Cte) =$ les entiers naturels (infini)
$Dom(Nom) =$ chaînes de moins de 20 caractères (fini)

Langages de requête

Langages qui permettent d'interroger la BD

(i) Langages relationnels "purs"

- Algèbre relationnelle
- Calcul relationnel par n-uplet
- Calcul relationnel par domaine

(ii) Langages pratiques

- SQL (Structured Query Language)
- QUEL (Query Language)
- SEQUEL (Structured English as a Query Language)
- QBE (Query By Example)

Algèbre relationnelle

Six opérations de base

1. Projection
2. Sélection
3. Union
4. Différence
5. Produit cartésien
6. Renommage

Certaines sont unaires d'autres sont binaires

Projection

Notation : $\pi_{A_1,\ldots,A_k}(r)$ où :

- r : nom de relation et
- $\forall 1 \leq i \leq k \ A_i \in Att(r)$.

Le résultat de cette opération est une relation avec k colonnes

Projection (Exemple)

On veut extraire les noms des employés de la relation <<Emp>> ci-dessous

Emp	Nom	Num_Cte	Rue
	Dupont	124	Neuve
	Dupont	235	Neuve
	Durand	123	Vieille

$\pi_{Nom}(Emp) =$

Nom
Dupont
Durand

Sélection

Notation : $\sigma_{Cond}(r)$ où

- r est le nom d'une relation
- *Cond* est une condition de la forme
  1. $Att_i\ \theta\ Att_j$ ou
     $Att_i\ \theta$ constante avec
     $\theta \in \{<, \leq, =, \geq, >, \neq\}$, ou bien
  2. une `conjonction` ($\wedge$) ou une `disjonction` ($\vee$) de conditions

Le résultat est une relation qui contient tous les n-uplets de r qui satisfont la condition *Cond*

Sélection (Exemple)

On veut avoir les informations concernant les employés dont le nom est Dupont

Emp	Nom	Num_Cte	Rue
	Dupont	124	Neuve
	Dupont	235	Neuve
	Durand	123	Vieille

$\sigma_{(Nom=Dupont)} =$

Nom	Num_Cte	Rue
Dupont	124	Neuve
Dupont	235	Neuve

Union, Différence et Intersection

- Opérations ensemblistes classiques
- Notation : $r \cup s$; $r - s$; $r \cap s$
- $r \cup s = \{t \mid t \in r \textbf{ ou } t \in s\}$
- $r - s = \{t \mid t \in r \textbf{ et } t \notin s\}$
- $r \cap s = \{t \mid t \in r \textbf{ et } t \in s\}$
- Opérations binaires
- Il faut que $Att(r) = Att(s)$

Union, Différence et Intersection

r	A	B
	α	1
	α	2
	β	1

s	A	B
	α	2
	β	3

$r - s =$

A	B
α	1
β	1

$r \cup s =$

A	B
α	1
α	2
β	1
β	3

$r \cap s =$

A	B
α	2

Produit cartésien

Notation : $r \times s$

- $r \times s = \{tv \mid t \in r \ \textbf{et} \quad v \in s\}$
- tv est la concaténation des tuples t et v

Cette opération n'est pas définie si $Att(r) \cap Att(s) \neq \emptyset$
$Att(r \times s) = Att(r) \cup Att(s)$

Produit cartésien (Exemple)

r	A	B
	α	1
	β	2

s	C	D	E
	α	10	$+$
	β	10	$-$

$r \times s =$

A	B	C	D	E
α	1	α	10	$+$
α	1	β	10	$-$
β	2	α	10	$+$
β	2	β	10	$-$

Renommage

Notation : $\rho_{Att_i \rightarrow Att'_i}(r)$
Permet de renommer l'attribut Att_i par Att'_i
Le résultat est la relation r avec un nouveau schéma

Renommage (Exemple)

r	A
	10
	20

$$\rho_{A \to B}(r) =$$

r	B
	10
	20

Composition des opérateurs

On peut appliquer un opérateur de l'algèbre au résultat d'une autre opération

Exemple : $\pi_A\Big(\sigma_{B=20}(r)\Big)$

On dit que l'algèbre relationnelle est un langage `fermé` car chaque opération prend une ou deux relations et retourne une relation.

Soient les schémas de relation $Tit(Id, Nom, Adresse)$ et $Cte(Num, Solde, Id\_Tit)$.

Le compte de numéro *Num* appartient au client identifié par *Id_Tit*.

On veut avoir (i) le numéro, (ii) le solde et (iii) le nom du titulaire de chaque compte débiteur.

Id	Nom	Adresse
A25	Dupond	rue neuve
B212	Durand	rue vieille

Tit

Num	Solde	Id_Tit
120	25234.24	A25
135	-100	A25
275	230	B212

Cte

Composition des opérateurs (Suite)

1. $Cte \times Tit$ retourne une relation qui associe à chaque tuple de Cte, tous les tuples de Tit
2. $\sigma_{Id=Id\_Tit}(Cte \times Tit)$ élimine les tuples où le compte n'est pas associé au bon titulaire
3. $\sigma_{Solde<0}\left(\sigma_{Id=Id\_Tit}(Cte \times Tit)\right)$ retient les comptes débiteurs
4. $\pi_{Nom,Num,Solde}\left(\sigma_{Solde<0}\left(\sigma_{Id=Id\_Tit}(Cte \times Tit)\right)\right)$ élimine les attributs non demandés

Comment aurait-on pu faire si dans Cte on avait Id au lieu de Id_Tit comme nom d'attribut ?

Jointure

Notation : $r \bowtie s$

- $Att(r \bowtie s) = Att(r) \cup Att(s)$
- Résultat : Soient $t_r \in r$ et $t_s \in s$. $t_r t_s \in r \bowtie s$ ssi $\forall A \in Att(r) \cap Att(s) t_r.A = t_s.A$

Jointure (Exemple)

r	A	B
	α	10
	α	15
	β	1

s	B	C
	10	$+$
	1	$-$

$r \bowtie s =$

A	B	C
α	10	$+$
β	1	$-$

Jointure (Exemple)

- Noter que le même résultat peut être obtenu comme suit :
  1. $temp_1 := \rho_{B \to B_1}(s)$
  2. $temp_2 := r \times temp_1$
  3. $temp_3 := \sigma_{B=B_1}(temp_2)$
  4. $res := \pi_{A,B,C}(temp_3)$
- La jointure n'est pas une opération de base de l'algèbre relationnelle

Calcul relationnel par n-uplet

- Les requêtes sont de la forme $\{t \mid P(t)\}$
- C'est l'ensemble des n-uplets tels que le `prédicat` $P(t)$ est vrai pour t
- t est une variable n-uplet et $t[A]$ désigne la valeur de l'attribut A dans t
- $t \in r$ signifie que t est un n-uplet de r
- P est une formule de la logique de premier ordre

Rappel sur le calcul des prédicats

- Des ensembles d'attributs, de constantes, de comparateurs $\{<, \dots\}$
- Les connecteurs logiques 'et' $\wedge$, 'ou' $\vee$ et la négation $\neg$
- Les quantificateurs $\exists$ et $\forall$:
  - $\exists t \in r\Big(Q(t)\Big)$: Il existe un tuple t de r tel que Q est vrai
  - $\forall t \in r\Big(Q(t)\Big)$: Q est vrai pour tout t de r

Exemples de requêtes

Considérons les schémas de relations suivants :
Film(Titre, Réalisateur, Acteur) instance f
Programme(NomCiné, Titre, Horaire) instance p
f contient des infos sur tous les films et p concerne le programme à Bordeaux

- Les films réalisés par Bergman

$$\{t \mid t \in f \wedge t[\text{Réalisateur}] = \text{"Bergman"}\}$$

- Les films où Jugnot et Lhermite jouent ensembles

$$\{t \mid t \in f \wedge \exists s \in f \Big(t[\text{Titre}] = s[\text{Titre}] \wedge t[\text{Acteur}] = \text{"Jugnot"} \wedge s[\text{Acteur}] = \text{"Lhermite"} \Big)\}$$

Exemples de requêtes(suite)

Les titres des films programmés à Bordeaux :

$$\{t \mid \exists s \in p\Big(t[\text{Titre}] = s[\text{Titre}]\Big)\}$$

Les films programmés à l'UGC mais pas au Trianon :

$$\{t \mid \exists s \in p\Big(s[\text{Titre}] = t[\text{Titre}] \wedge s[\text{NomCiné}] = \text{“UGC”} \wedge$$
$$\neg\exists u \in p\Big(u[\text{NomCiné}] = \text{“Trianon”} \wedge u[\text{Titre}] = t[\text{Titre}]\Big)\Big)\}$$

Les titres de films qui passent à l'UGC ainsi que leurs réalisateurs :

$$\{t \mid \exists s \in p\Big(\exists u \in f\Big(s[\text{NomCiné}] = \text{“UGC”} \wedge$$
$$s[\text{Titre}] = u[\text{Titre}] = t[\text{Titre}] \wedge t[\text{Réal}] = u[\text{Réal}]\Big)\Big)\}$$

Expressions “non saines”

Il est possible d'écrire des requêtes en calcul qui retournent une relation infinie.
Exemple : Soit NumCte(Num) avec l'instance n et la requête $\{t \mid \neg t \in n\}$ i.e les numéros de compte non recensés. Si on considère que le $Dom(Num) = N$, alors la réponse à cette requête est infinie.
Une requête est *saine si quelle que soit l'instance de la base dans laquelle on l'évalue, elle retourne une réponse finie.*
Dépendance du domaine.

Calcul relationnel par domaine

Les requêtes sont de la forme :

$$\{\langle x_1, \ldots, x_n \rangle \mid P(x_1, \ldots, x_n)\}$$

Les x_i représentent des variables de domaine.
$P(x_1, \ldots, x_n)$ est une formule similaire à celles qu'on trouve dans la logique des prédicats.
Exemple : Les titres de films programmés à l'UGC de Bordeaux

$$\{\langle t \rangle \mid \exists \langle nc, t, h \rangle \in p(nc = \text{“}UGC\text{”})\}$$

Relation entre les 3 langages

- Toute requête exprimée en algèbre peut être exprimée par le calcul.
- Toute requête “saine” du calcul peut être exprimée par une requête de l’algèbre.
- Les 3 langages sont donc équivalents d’un point de vue puissance d’expression.
- L’algèbre est un langage *procédurale* (quoi et comment) alors que le calcul ne l’est pas (seulement quoi)