

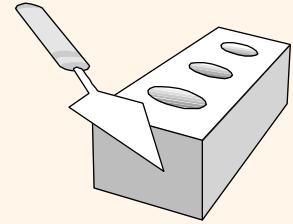
Bases de Données Avancées



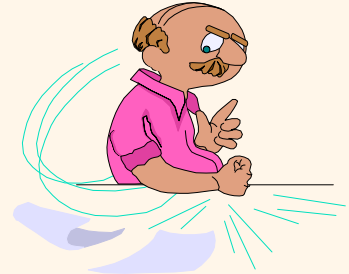
Ioana Ileana
Université Paris Descartes

Cours basé sur le livre (et les diapositives de):
Database Management Systems 3ed, R. Ramakrishnan et J. Gehrke

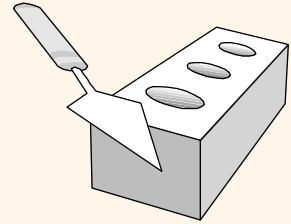




Cours 7a: rappels algèbre relationnelle



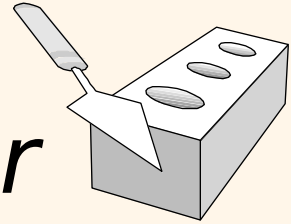
- ❖ Diapos traduites et adaptées du matériel fourni en complément du livre Database Management Systems 3ed, par Ramakrishnan et Gehrke ; un grand merci aux auteurs pour la réalisation et la disponibilité de ce matériel !
- ❖ Les diapos originales (en anglais) sont disponibles ici :
<http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/slides/slides3ed.html>
- ❖ Plus particulièrement, ce cours touche aux éléments dans les Chapitre 4A du livre ci-dessus; lecture conseillée! ;)



Langages de requête

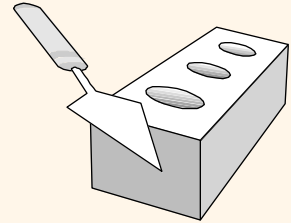
- ❖ Langages de requête (query languages = QL): permettent la consultation et la manipulation des données
- ❖ Le modèle relationnel supporte des langages de requête simples ET puissants:
 - De très solides fondations formelles basées sur la logique.
 - Permettant une vaste gamme d'optimisations!
- ❖ Langages de requête **!=** langages de programmation !
 - Pas censés être utilisés pour des calculs complexes.
 - Permettent en revanche un accès facile et efficace à des jeux de données de taille importante.

Langages formels de requête sur le modèle relationnel



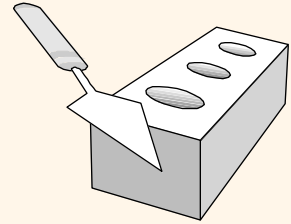
- ❖ Deux langages de requête formels (“mathématiques”) se posent à la base des langages de requête “réels” (SQL :)) sur le modèle relationnel – et à la base des implémentations dans les SGBD!
 - Algèbre relationnelle: Plus **opérationnel**, ce langage est très utile pour représenter les *plans d'exécution* des requêtes!
 - Calcul relationnel: Langage déclaratif (**non-opérationnel**), permet de décrire ce que l'utilisateur “veut” et non pas comment obtenir ce résultat!

Requêtes (queries) et schémas relationnels



- ❖ Une requête est appliquée à *un ensemble d'instances de relations*, et le résultat d'une requête est lui-même une instance de relation!
 - *Les schémas des relations "d'entrée" (input)* pour une requête sont *fixés* (mais la requête s'exécutera sur n'importe quel ensemble d'instances correspondant à ces schémas!)
 - Le *schéma du résultat* de la requête est aussi *fixé*! Il est en effet déterminé par la définition des éléments du langage de requêtes!

Exemple



- ❖ Relations: “Marins” et “Reservations”
- ❖ Instances: M1 et M2 de “Marins”, et R1 de “Reservations”
- ❖ Dans ce qui suit, on utilisera les noms des colonnes, mais aussi des fois leurs indices (“notation positionnelle”)

R1

<u>mid</u>	<u>bid</u>	<u>jour</u>
22	101	10/10/96
58	103	11/12/96

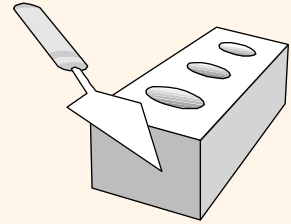
M1

<u>mid</u>	nom	note	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

M2

<u>mid</u>	nom	note	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

Algèbre relationnelle



❖ Opérations de base:

- Sélection (restriction) (σ) sélectionne un sous-ensemble de lignes (tuples) d'une relation.
- Projection (π) retient les colonnes (champs, attributs) spécifiés (enlève les autres);
- Produit cartésien (\times) permet de combiner deux relations.
- Différence ($-$) Tuples qui sont dans R1 mais pas dans R2
- Union (\cup) Tuples qui sont en R1 ou en R2.

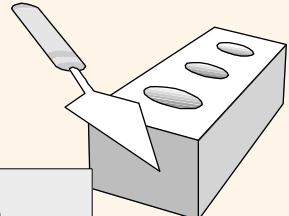
❖ D'autres opérations:

- Intersection, jointure, division, renommage: ne sont pas essentielles, mais (très!) utiles.

❖ Chaque opération retourne une relation, elles peuvent donc être composées! (algèbre "fermée").

Projection

- ❖ Efface les attributs qui ne sont pas dans la *liste de projection*.
- ❖ *Le schéma* du résultat contient exactement les champs de la liste de projection, avec les mêmes noms que dans la relation input.
- ❖ L'opérateur de projection doit éliminer les *doublons*!

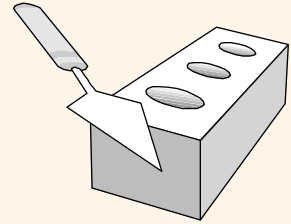


nom	note
yuppy	9
lubber	8
guppy	5
rusty	10

$\pi_{\text{nom, note}}(M2)$

age
35.0
55.5

$\pi_{\text{age}}(M2)$



Sélection

- ❖ Sélectionne des lignes qui correspondent à une *condition de sélection*.
- ❖ Pas de doublons dans le résultat!
- ❖ *Le schéma* du résultat est identique au schéma de la relation entrée (input).
- ❖ *La relation résultat* peut être l'entrée d'une autre opération d'algèbre relationnelle (*composition des opérateurs*).

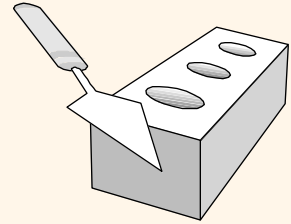
mid	nom	note	age
28	yuppy	9	35.0
58	rusty	10	35.0

$\sigma_{\text{note} > 8} (M2)$

nom	note
yuppy	9
rusty	10

$\Pi_{\text{nom, note}} (\sigma_{\text{note} > 8} (M2))$

Union, Intersection, Différence



❖ Prennent en entrée deux relations compatibles:

- Même nombre de champs
- Les champs “correspondants” ont le même type

mid	nom	note	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$M1 \cup M2$

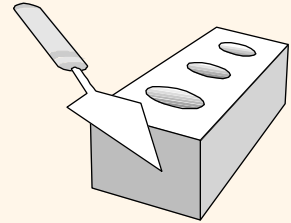
❖ Quel schéma pour le résultat?

mid	nom	note	age
22	dustin	7	45.0

$M1 - M2$

mid	nom	note	age
31	lubber	8	55.5
58	rusty	10	35.0

$M1 \cap M2$



Produit cartésien

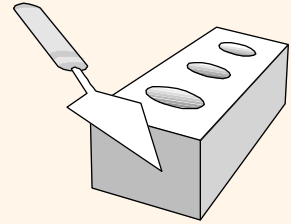
- ❖ Chaque ligne de la première relation est combinée avec chaque ligne de la deuxième
- ❖ *Le schéma du résultat* a un champ pour chaque champ de chacune des relations (avec des noms “hérités” si possible) .

M1xR1 :

(mid)	nom	note	age	(mid)	bid	jour
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Conflit: M1 et R1 ont chacune un champ *mid*; pour les distinguer, on doit s’y référer par la position, mais on peut aussi faire un renommage (renaming)

$$\rho(\text{Result}(1 \rightarrow \text{mid1}, 5 \rightarrow \text{mid2}), M1 \times R1)$$



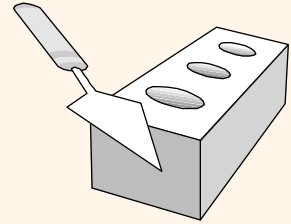
Jointures

❖ Jointure conditionnelle: $R \bowtie_c S = \sigma_c(R \times S)$

(mid)	nom	note	age	(mid)	bid	jour
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	58	103	11/12/96

$$M1 \bowtie_{M1.mid < R1.mid} R1$$

- ❖ *Le schéma du résultat* : comme pour le produit cartésien.
- ❖ Moins de tuples que le produit cartésien, potentiellement à calculer de manière plus efficace
- ❖ Appelée aussi *thêta-jointure*.



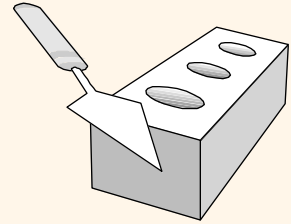
Jointures

- ❖ Equi-Jointure: Cas spécial de jointure conditionnelle où la condition c contient uniquement des **égalités**

mid	nom	note	age	bid	jour
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

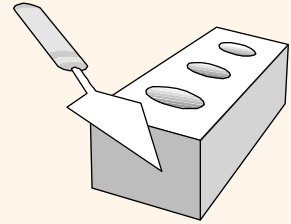
$$M1 ><_{mid} R1$$

- ❖ Le schéma résultat: similaire au produit cartésien, mais avec un seul exemplaire / paire de champs qui participent à une égalité .
- ❖ Jointure Naturelle: Equi-jointure sur *tous* les champs communs.



Division

- ❖ N'est pas un opérateur “de base”, mais très utile pour exprimer des requêtes comme:
*Trouver les marins qui ont réservé **tous** les bateaux*
- ❖ Soit les rel. A avec champs x et y et B avec champ y :
 - $A/B = \{ \langle x \rangle \mid \exists \langle x, y \rangle \in A \forall \langle y \rangle \in B \}$
 - **A/B contient toutes les valeurs de x t.q. pour chaque valeur y dans B il existe un tuple (x,y) dans A**
 - *Ou*: une valeur de x appartient à la division ssi dans les tuples de A elle figure associée à toutes les valeurs de y dans B !
- ❖ En général, x et y sont des listes de champs; y est la liste des champs de B , et $x \cup y$ est la liste des champs de A .



Exemples de division A/B

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

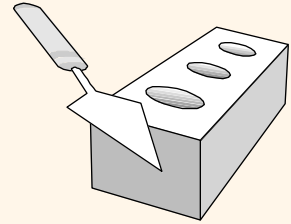
A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3



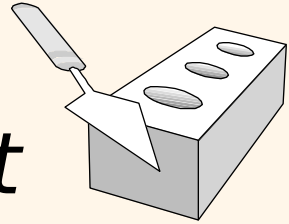
La division A/B avec des opérateurs de base

- ❖ La division est “un raccourci utile”, mais elle n’est pas essentielle.
 - (Pareil pour les jointures, mais les jointures sont tellement répandues que les systèmes en font des implémentations spécifiques...)
- ❖ **Idée:** Pour A/B , trouver les valeurs de x qui ne sont pas “disqualifiées” par une valeur y dans B .
 - Une valeur de x est *disqualifiée* si en lui rajoutant y nous obtenons un tuple xy qui n’est pas dans A .

Valuers x disqualifiés: $\Pi_x((\Pi_x(A) \times B) - A)$

A/B : $\Pi_x(A) - \text{disqualifiés}$

Trouver les noms des marins qui ont réservé le bateau #103

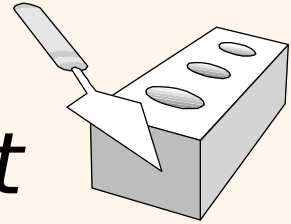


❖ **Solution 1:** $\Pi_{nom}((\sigma_{bid=103} Reservations) ><_{mid} Marins)$

❖ **Solution 2:** $\rho(Temp1, \sigma_{bid=103} Reservations)$
 $\rho(Temp2, Temp1 ><_{mid} Marins)$
 $\Pi_{nom}(Temp2)$

❖ **Solution 3:** $\Pi_{nom}(\sigma_{bid=103}(Reservations ><_{mid} Marins))$

Trouver les noms des marins qui ont réservé un bateau rouge



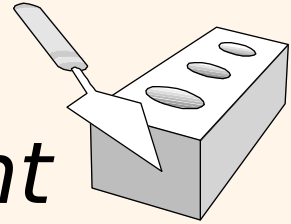
- ❖ Infos sur la couleur dans la table Bateaux; besoin donc d'une jointure supplémentaire:

$$\Pi_{nom} \left(\left(\sigma_{couleur='rouge'} Bateaux \right) ><_{bid} Reservations ><_{mid} Marins \right)$$

- ❖ Une solution plus efficace:

$$\Pi_{nom} \left(\Pi_{mid} \left(\left(\Pi_{bid} \sigma_{couleur='rouge'} Bateaux \right) ><_{bid} Reservations \right) ><_{mid} Marins \right)$$

Un optimiseur de requêtes peut trouver la deuxième formule, à partir de la première!



Trouver les noms des marins qui ont réservé un bateau rouge ou vert

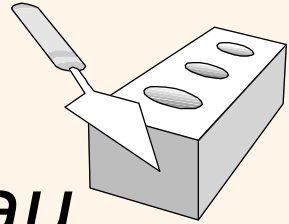
- ❖ D'abord identifier les bateaux rouges ou verts, ensuite trouver les marins qui ont réservé un de ces bateaux

$$\rho\left(\text{Tempbateaux}, \left(\sigma_{\text{couleur} = \text{'red'} \vee \text{couleur} = \text{'vert'}} \text{Bateaux}\right)\right)$$

$$\Pi_{\text{nom}}\left(\text{Tempbateaux} \bowtie \text{Reservations} \bowtie \text{Marins}\right)$$

- ❖ On peut aussi définir Tempbateaux avec de l'union! (Comment?)
- ❖ Que se passe-t-il si on remplace le « ou » par un « et » logique?

Trouver les noms des marins qui ont réservé un bateau rouge ET un bateau vert



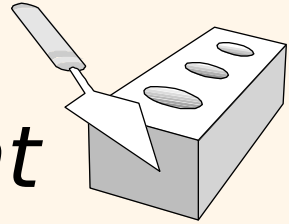
- ❖ L'approche précédente ne marche pas! Nous devons trouver les marins qui ont réservé des bateaux rouges, ceux qui ont réservé des bateaux verts, et faire l'intersection (rappel: *mid* est une clé pour Marins):

$$\rho(\text{Temprouge}, \Pi_{\text{mid}}((\sigma_{\text{couleur}='rouge'} \text{Bateaux}) \bowtie \text{Reservations}))$$

$$\rho(\text{Tempvert}, \Pi_{\text{mid}}((\sigma_{\text{couleur}='vert'} \text{Bateaux}) \bowtie \text{Reservations}))$$

$$\Pi_{\text{nom}}((\text{Tempvert} \cap \text{Temprouge}) \bowtie \text{Marins})$$

Trouver les noms des marins qui ont réservé tous les bateaux



- ❖ Avec division; attention au choix des schémas des relations qu'on donne en entrée à la division:

$$\rho(\text{Tempmids}, (\Pi_{\text{mid}, \text{bid}} \text{Reservations}) / (\Pi_{\text{bid}} \text{Bateaux}))$$

$$\Pi_{\text{nom}}(\text{Tempmids} \succ \text{Marins})$$