UNIVERSITE BADJI MOKHTAR-ANNABA
FACULTÉ DE TECHNOLOGIE
DÉPARTEMENT D'INFORMATIQUE

Algèbre Relationnelle

BDD 2LMD

Présenté par : Pr BELLEILI Habiba

Algèbre relationnelle
Définition

ÅUne algèbre est un ensemble **d'opérateurs de base**, formellement définis, qui peuvent être combinés pour construire des **expressions algébriques**

ÅUne algèbre est dite fermée si le résultat de tout opérateur est du même type que les opérandes (ce qui est indispensable pour construire des expressions)

ÅComplétude: toute manipulation pouvant être souhaitée par les utilisateurs devrait pouvoir être exprimable par une expression algébrique

Algèbre relationnelle

- Lô***algèbre relationnelle*** est une collection dôopérations formelles qui agissent sur des relations et produisent les relations en résultats
- **Côest la force** du modèle relationnel
- Il y a deux types dôopérations de base :
  - Les opérations ensemblistes traditionnelles (une relation doit être traité comme un ensemble). Ces opérations sont **des opérations binaires** (à partir de deux relations elles en construisent une troisième) (lôunion, l'intersection, la différence et le produit cartésien)
  - Les opérations spécifiques sont les **opérations unaires** de projection et restriction qui, à partir dôune relation , en construisent une autre, et **lôopération binaire** de jointure

Les opérations ensemblistes

1. Union : opération portant sur deux relations ***<u>de même schéma R et S</u>*** consistant à construire une relation de même schéma T ayant pour tuples ceux appartenant à R ou S ou aux deux relations.

Notations :

ï $R \cup S$

ï UNION (R, S)

Précondition:

Schéma(R) = Schéma(S) = Schéma (T)

A	B
α	1
α	2
β	1

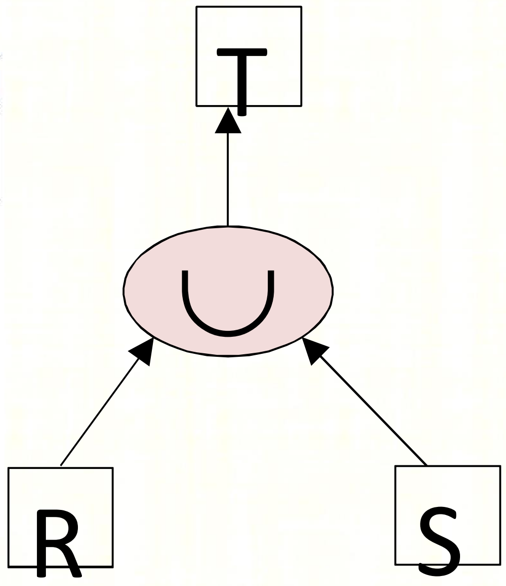
r

A	B
α	2
β	3

s

A	B
α	1
α	2
β	1
β	3

$T = R \cup S$



Différence

2. Différence : opération portant sur deux relations de **même schéma R et S** consistant à construire une relation de <u>même schéma T</u> ayant pour tuples ceux appartenant à R et nôappartenant pas à S.

Notations :

ï R - S

ï DIFFERENCE (R, S)

ï MINUS (R, S)

Précondition:

Schéma (R)=Schéma(S)= Schéma(T)

A	B
α	1
α	2
β	1

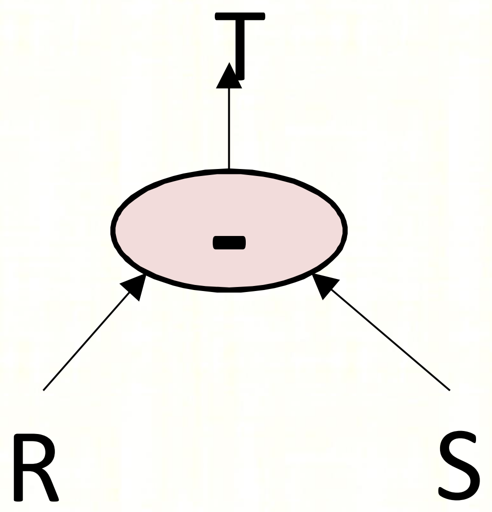
r

A	B
α	2
β	3

s

A	B
α	1
β	1

T = R - S



Produit cartésien

3. Produit cartésien : opération portant sur **<u>deux relations R et S</u>**, consistant à construire une relation T ayant pour <u>schéma la concaténation des schémas des relations opérandes (R et S)</u> et pour <u>tuples les combinaisons des tuples des relations opérandes</u>.

Notations :

- $R \times S$
- PRODUCT (R, S)

Précondition: pas de précondition

Schéma(T)=Schéma(R) union Schéma(S)

A	B
α	1
β	2

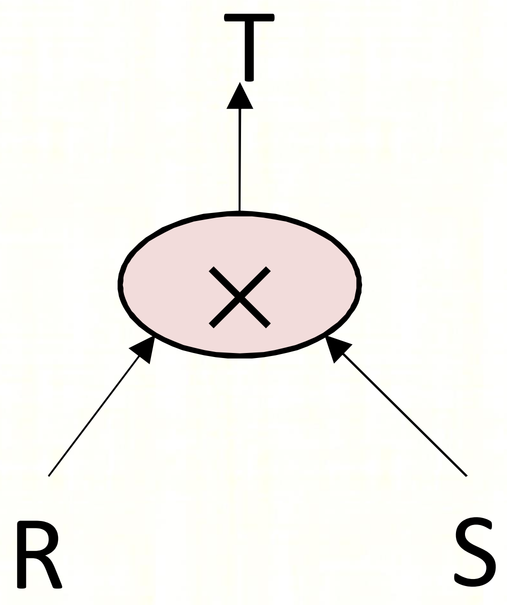
r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

$T = R \times S$



Intersection

4. Intersection : opération portant sur deux relations de même schéma R et S consistant à construire une relation de même schéma T ayant pour tuples ceux appartenant à la fois à R et S

Notations :

ï R ∩ S

ï INTERSECT(R, S)

ï AND (R, S)

Å PRECONDITION:

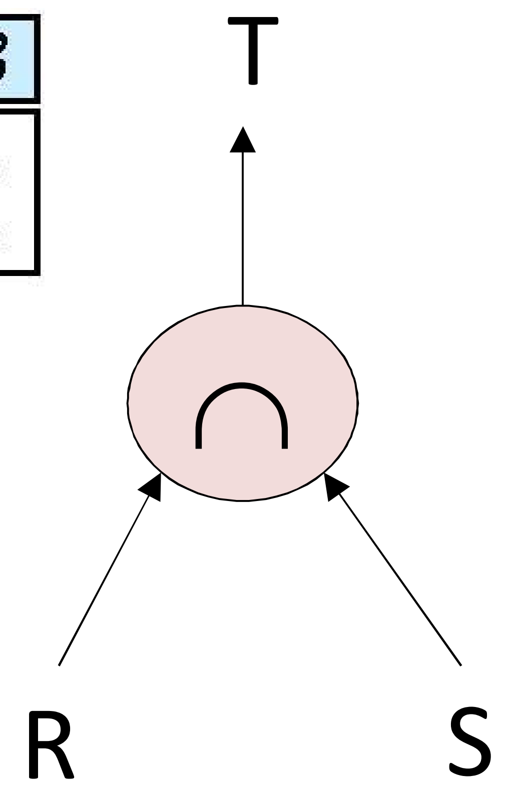
ï R et S doivent être de même sch

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s



A	B
α	2

$T = R \cap S$

Intersection (remarque)

Å Lôopération dôintersection sôobtient
à partir de lôopération de différence :

ï $R \cap S = R$ ï $(R-S)$

ï $R \cap S=S-(S-R)$

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

A	B
α	1
β	1

r - S

r- (r – S)

A	B
α	2

Les opérations spécifiques

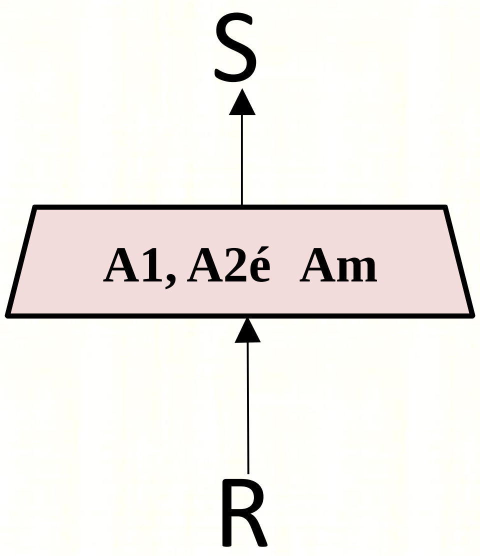
1. Projection: opération sur une relation R (unaire), consistant à composer une relation S en enlevant à la relation initiale tous les attributs non mentionnés en opérandes (aussi bien au niveau du schéma que des tuples) et <u>en éliminant les tuples en double</u> qui sont conservés une seule fois.

Notations : $Attribut_i, Attribut_j$***é*** $Attribut_m$ **:** sont les attributs de projection

ï $\prod_{Attribut_i, Attribut_j \ldots Attribut_m} (R)$

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

R



A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

$S = \prod_{A,C}(R)$

Réstriction ou sélection

2.2. Restriction ou sélection : opération sur une relation R (unaires) produisant une relation S de même schéma, mais comportant les seuls tuples qui vérifient la condition précisée en argument.

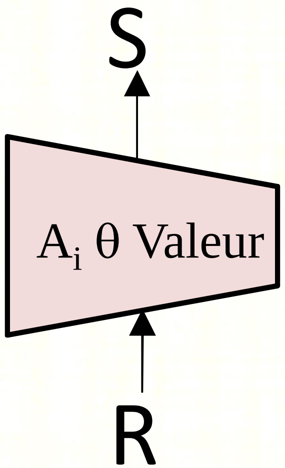
- Les conditions possibles sont des formules dans le calcul propositionnel constituées de termes connectés par : $\wedge$ (**et**), $\vee$ (**ou**), $\neg$ (**non**). Chaque terme est du type :
  - <Attribut> <Opérateur> <Attribut> ou <Constante>, Tel que <Opérateur> $\in \{=, >, \geq, <, \leq, \neq\}$,

Notations :

- $\sigma_{\text{condition}}(R)$

R

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10



$\theta \in \{=, >, \geq, <, \leq, \neq\}$

A	B	C	D
α	α	1	7
β	β	23	10

$S = \sigma_{A=B \text{ and } D > 5}(R)$

Jointure

3. Jointure : Opération consistant <u>à rapprocher selon une condition</u> les tuples de deux relations R et S afin de former une troisième relation T qui contient tous les tuples obtenus <u>en concaténant un tuple de R et un tuple de S vérifiant la condition de rapprochement</u>

Å Syntaxe : R JOIN S ou R*S ou R $\bowtie$ S

Å La condition de jointure est du type :

- ï ***<Attribut1> <opérateur Comparaison> <Attribut2>***,
- ï tel que Attribut1 $\in$ R et Attribut2 $\in$ S

Å Si **<Opérateur> est ó=ó a**lors on parle de lõ***équi-jointure*** qui est une véritable composition de relation au sens mathématique

Å **Si <Operateur>** $\in$**{>,** $\geq$**, <,** $\leq$**,** $\neq$**}**alors on parle de lõ***inéqui-jointure***

Jointure Naturelle

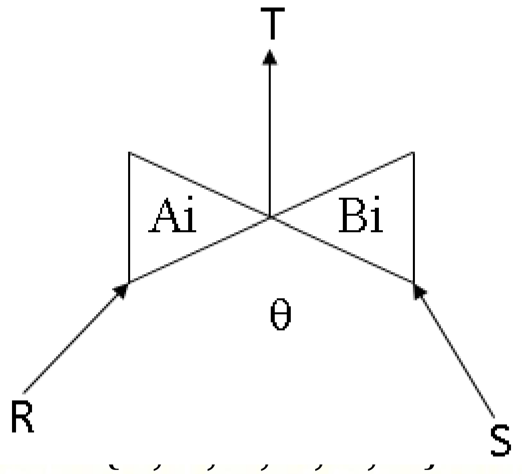
Jointure naturelle : Opération consistant à <u>rapprocher selon une condition </u>les tuples de deux relations R et S afin de former une troisième relation T dont les attributs sont lôunion des attributs de R et S, et dont les tuples sont obtenus en composant un tuple de R et un tuple de S **<u>ayant mêmes valeurs pour les attributs de même nom</u>**.

Notations :

- $R \bowtie_{Condition} S$
- JOIN(R, S, Condition)

- **<u>PRECONDITION: Au moins un attributs de même nom en commun entre les deux relations</u>**
- Schéma (R JOIN S)=(Schéma(R) $\cup$ Schéma(S)) ï (Schéma(R)$\cap$schéma(S))

Jointure Naturelle exemple



A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

$$T = R \bowtie_{B, C} S$$

Théta Jointure

Theta-jointure $\bowtie [p]$

- but: créer toutes les combinaisons significatives entre tuples de deux relations
  - significatif = critère de combinaison explicitement défini en paramètre de l'opération
- précondition: les deux relations n'ont pas d'attribut de même nom
- exemple : R $\bowtie$ [B ≠ C] S

R

A	B
a	b
b	c
c	b

S

C	D	E
b	c	d
b	a	b
c	a	c

R$\bowtie$ [B ≠ C] S

A	B	C	D	E
a	b	c	a	c
b	c	b	c	d
b	c	b	a	b
c	b	c	a	c

Division

Division : Opération consistant à construire le quotient de la relation

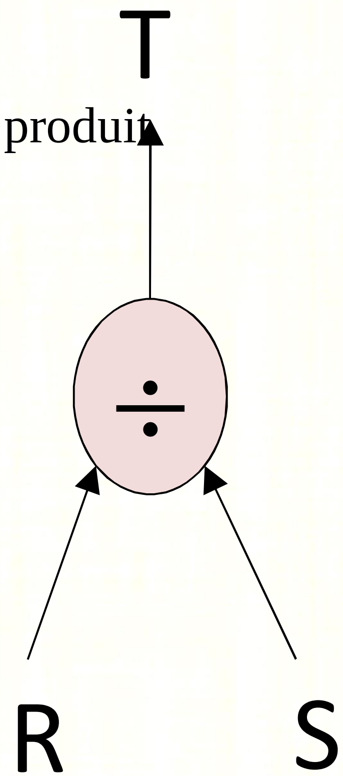
R (A_1, A_2é A_p, A_{p+1}é A_n) par la relation S (A_{p+1}é A_n) comme la relation Q (A_1, A_2é A_p) dont les tuples sont ceux concaténés à tout tuple de S donnent un tuple de R

Notations :

- ï R ÷ S
- ï DIVISION (R, S)

Å Lôopération de division sôobtient à partir de lôopération de différence, du produit cartésien et de la projection:

- ï $R \div S = R1 - R2$
- ï $R1 = \Pi_{A1,\ A2é\ \ Ap}(R)$
- ï $R2 = \Pi_{A1,\ A2é\ \ Ap}((R1 \times S) - R)$



Algèbre relationnelle

Lôopération de division sôobtient à partir de lôopération de différence, du produit cartésien et de la projection:

$R1=\Pi_{A1,\, A2é\;\; Ap}(R)$

$R2= \Pi_{A1,\, A2é\;\; Ap}((R1\times S)-R)$

$R \div S = R1-R2$

A	B	C
a1	b1	c1
a1	b2	c2
a2	b1	c1
a2	b2	c1
a3	b3	c1

R

B	C
b1	c1
b2	c1

S

A
a2

Q=R÷S

R1= Π_A(R)

A
a1
a2
a3

X

S

B	C
b1	c1
b2	c1

=

R1 X S

A	B	C
a1	b1	c1
a1	b2	c1
a2	b1	c1
a2	b2	c1
a3	b1	C1
a3	b2	c1

(R1 X S) - R

A	B	C
a1	b2	c1
a3	b1	c1
a3	b2	c1

R2= Π_A(R1 X S) - R

A
a1
a3

A
a2

R ÷S =R1- R2

Complément

Complément : Ensemble des tuples du produit cartésien des domaines des attributs dõune relation R nõappartenant pas à cette relation. Le complément suppose a priori que les domaines sont finis (sinon on obtient des relations infinies).

Notations :

ï $\neg$***R***

ï -R

Å Opération peu utilisée, car elle permet de générer des tuples qui ne sont pas dans la base, en général très nombreux.

Å Si lõon note par $\times D_i$ le produit cartésien des domaines, le complément dune relation R est obtenu à partir de la différence comme suit :

✖ $\neg \boldsymbol{R} = \times D_i - \boldsymbol{R}$

Algèbre relationnelle

3.3. Complément

Exemple :

Domaines:

Å A={a1, a2, a3}

Å B= {b1, b2, b3}

R	A	B
	a1	b1
	a1	b2
	a2	b1
	a3	b2
	a3	b3

¬R	A	B
	a1	b3
	a2	b2
	a2	b3
	a3	b1

lôopérateur de renommage

opération permettant de renommer, les attributs et les noms des relations

- On renomme les attributs dôune relation

Exemple:

  - Soit R (A1,A2,A3)
  - R1=α (A1:B1, A2,:B2,A3:B3) R : crée une nouvelle relation R1 en renommant tous les attributs de R

  - On peut renommer quôun sous ensemble des attributs dôune relation:
  - Exemple:
    - R2 = α (A1:B1,A3:C1)R

Jointure externe

Jointure externe (External Join) : opération générant une relation T à partir de deux relations R et S, par jointure de ces deux relations et ajout de tuples de relations R et S ne participant pas à la jointure, avec des valeurs nulles pour les attributs de lõautre relation

Notations :

ï T=R $\underset{\bullet}{\bowtie}$ S, ou T=R $\overline{\underline{\;}}\!\bowtie\!\overline{\underline{\;}}$ S

ï T=EXT-JOIN(R, S)

Å **La jointure externe permet de conserver les tuples « sans correspondant « avec des valeurs nulles associées quand nécessaire.**

Å Opération très utile en pratique, en particulier pour composer des vues sans perte dõinformations (ex., on peut joindre des tables CLIENTS et COMMANDES sur un numéro de client commun, en gardant les clients sans commande et les commandes sans client associé

Å On en distingue deux types :

ï La ***jointure externe droite*** qui garde seulement les tuples sans correspondant de la relation droite. Celle-ci est notée $\underset{\quad\bullet}{\bowtie}$, $\bowtie\!\overline{\underline{\;}}$, ou REXT-JOIN

ï ***La jointure externe gauche*** qui garde seulement les tuples sans correspondant de la relation gauche. Celle-ci est notée $\underset{\bullet\quad}{\bowtie}$, $\overline{\underline{\;}}\!\bowtie$, ou LEXT-JOIN

NumClient	NomClient
123	xxxx
226	yyy
336	aaa
233	vvv

NumCom	NumClient	NumPro	Qte
789	123	111	20
790	226	222	13
791	123	333	10
755	-	77	12

NumClient	Nomclient	NumCom	NumPro	Qté
123	xxx	789	111	20
123	xxx	791	333	10
226	yyy	790	222	13

Jointure naturelle

NumClient	Nomclient	NumCom	NumPro	Qté
123	xxx	789	111	20
123	xxx	791	333	10
226	yyy	790	222	13
336	aaa	-	-	-
233	vvv	-	-	-
-	-	755	77	12

Jointure externe

Jointure externe

R	A	B
	a1	b1
	a1	b2
	a2	b1
	a1	b4

S	B	C
	b1	c1
	b3	c1
	b5	c2

R ⟗ S

T	A	B	C
	a1	b1	c1
	a1	b2	-
	a2	b1	c1
	a1	b4	-
	-	b3	c1
	-	b5	c2

R ⟕ S

T	A	B	C
	a1	b1	c1
	a1	b2	-
	a2	b1	c1
	a1	b4	-

R ⟖ S

T	A	B	C
	a1	b1	c1
	a2	b1	c1
	-	b3	c1
	-	b5	c2

Algèbre relationnelle

4. Les expressions de lôalgèbre relationnelle

- A partir des opérations de lôalgèbre relationnelle, il est possible de composer un langage dôinterrogation de bases de données.
- Chaque requête (question) peut être représentée par arbre dôopérateurs relationnels
- Le paraphrasage en anglais de telles expressions est à la base du langage SQL.

4.1. Langage algébrique

- La réponse à la plupart des questions que lôon peut poser à une base de données relationnelle peut sôélaborer en utilisant des expressions dôopérations de lôalgèbre relationnelle.

Algèbre relationnelle

4.1. Langage algébrique

Å Soit le schéma relationnel dôune base de données touristiques (les clés sont soulignées)

```
STATION (NUMSTA, NOMSTAT, ALTITUDE, REGION)
HÔTEL (NUMHOT, NOMHOT, CATEGORIE, NUMSTA)
CLIENT (NUMCLI, NOMCLI, ADRCLI, TELCLI)
CHAMBRE (NUMCH, NUMHOT, NBLITS)
RESERVER (NUMCLI, NUMCH, NUMHOT, DATEDEB, DATEFIN,
NBPERS)
```

Exemple de requêtes

Å Ces requêtes peuvent être exprimées comme des expressions dôopérations, ou comme des opérations successive appliquées sur des relations intermédiaires ou de base, générant des relations intermédiaires.

Algèbre relationnelle

Soit le schéma relationnel d'une base de données touristiques (les clés sont soulignées)

STATION (<u>NUMSTA</u>, NOMSTAT, ALTITUDE, REGION)
HÔTEL (<u>NUMHOT</u>, NOMHOT, CATEGORIE, NUMSTA)
CLIENT (<u>NUMCLI</u>, NOMCLI, ADRCLI, TELCLI)
CHAMBRE (<u>NUMCH, NUMHOT</u>, NBLITS)
RESERVER (<u>NUMCLI, NUMCH, NUMHOT, DATEDEB</u>, DATEFIN, NBPERS)

- ***(Q1) Donner les noms de station de la région Haute-Normandie et d'altitude > 1000 M :***
  - $R1 = \sigma_{REGION=\text{" Haute-Normandie"}}(STATION)$
  - $R2 = \sigma_{ALTITUDE>1000}(STATION)$
  - $R3 = R1 \cap R2$
  - $RESULTAT = \Pi_{NOMSTA}(R3)$
- La requête (Q1) peut simplement s'exprimer comme suit

$$\prod_{NOMSTA}\left(\sigma_{\substack{REGION=\text{"HAUTE-NORMANDIE"}\\ AND\ ALTITUDE>1000}}(STATION)\right)$$

Algèbre relationnelle

Soit le schéma relationnel d'une base de données touristiques (les clés sont soulignées)

```
STATION (NUMSTA, NOMSTAT, ALTITUDE, REGION)
HÔTEL (NUMHOT, NOMHOT, CATEGORIE, NUMSTA)
CLIENT (NUMCLI, NOMCLI, ADRCLI, TELCLI)
CHAMBRE (NUMCH, NUMHOT, NBLITS)
RESERVER (NUMCLI, NUMCH, NUMHOT, DATEDEB, DATEFIN,
NBPERS)
```

- ***(Q2) Donner les numéros et noms des hôtels de la station de Paris OU d'Alsace :***
  - $R1 = \sigma_{REGION="Paris"}(STATION)$
  - $R2 = \sigma_{REGION="Alsace"}(STATION)$
  - $R3 = R1 \cup R2$ *schéma (R3) =schéma(STATION)*
  - $R4 = R3 \underset{NUMSTA}{\bowtie} HÔTEL$ *schéma(R4)= Schéma(R5) union Schéma(Hotel)*
  - $RESULTAT = \Pi_{NUMHOT,NOMHOT}(R4)$

Ou tout simplement

$$\Pi_{NUMHOT,NOMHOT}\left(\sigma_{\substack{REGION="PARIS" \\ OR \\ REGION="ALSACE"}}(STATION) \bowtie HÔTEL\right)$$

Algèbre relationnelle

- ***(Q3) Donner les noms et adresses des clients ayant réservé plus de 5 places dans un hôtel de 3 étoiles avec le nom de cet hôtel :***

- $R1 = \sigma_{NBPERS>5}(RESERVER)$
- $R2 = \sigma_{CATEGORIE\ =''***''}(HÔTEL)$
- $R3 = \Pi_{NUMCLI,NUMHOT}(R1)$ *// schéma R3 (NUMCLI,NUMHOT)*
- $R4 = CLIENT \bowtie R3$ *// pour chercher le nom du clients schéma(CLIENT) union*
- *schéma (R3)*
- $R5 = R2 \bowtie R3$ *// pour chercher le nom de lάhotel schéma de R2 union schéma(R3)*
- $R6 = R4 \bowtie R5$
- RESULTAT = $\Pi_{NOMCLI,ADRCLI,NOMHOT}(R6)$

ou tout simplement

$$\prod_{NOMCLI,\ ADRCLI,\ NOMHOT}(((\prod_{NUMCLI,\ NUMHOT}(\sigma_{NBPERS>5}(RESERVER)))\bowtie(\sigma_{CATEGORIE\ =''***''}(HÔTEL)))\bowtie CLIENT\ )$$

```
STATION (NUMSTA, NOMSTAT, ALTITUDE, REGION)
HÔTEL (NUMHOT, NOMHOT, CATEGORIE, NUMSTA)
CLIENT (NUMCLI, NOMCLI, ADRCLI, TELCLI)
CHAMBRE (NUMCH, NUMHOT, NBLITS)
RESERVER (NUMCLI, NUMCH, NUMHOT, DATEDEB, DATEFIN,
NBPERS)
```

STATION (<u>NUMSTA</u>, NOMSTAT, ALTITUDE, REGION)
HÔTEL (<u>NUMHOT</u>, NOMHOT, CATEGORIE, NUMSTA)
CLIENT (<u>NUMCLI</u>, NOMCLI, ADRCLI, TELCLI)
CHAMBRE (<u>NUMCH, NUMHOT</u>, NBLITS)
RESERVER (<u>NUMCLI, NUMCH, NUMHOT, DATEDEB</u>, DATEFIN, NBPERS)

- ***Q4) Donner le nom des clients nâayant réservé que dans des hôtels 4 étoiles :***

  - $R1 = CLIENT \bowtie RESERVER \bowtie HÔTEL$
  - $R2 = \sigma_{CATEGORIE = "****"}(R1)$
  - $R3 = \Pi_{NOMCLI}(R2)$
  - $R4 = \sigma_{CATEGORIE \neq "****"}(R1)$
  - $R5 = \Pi_{NOMCLI}(R4)$,
  - $RESULTAT = R3 - R5$ $schéma(R3) = Schéma(R5)$

$$\Pi_{NOMCLI}(\sigma_{CATEGORIE = "****"}(HÔTEL) \bowtie RSERVER \bowtie CLIENT) -$$
$$\Pi_{NOMCLI}(\sigma_{CATEGORIE \neq "****"}(HÔTEL) \bowtie RSERVER \bowtie CLIENT)$$

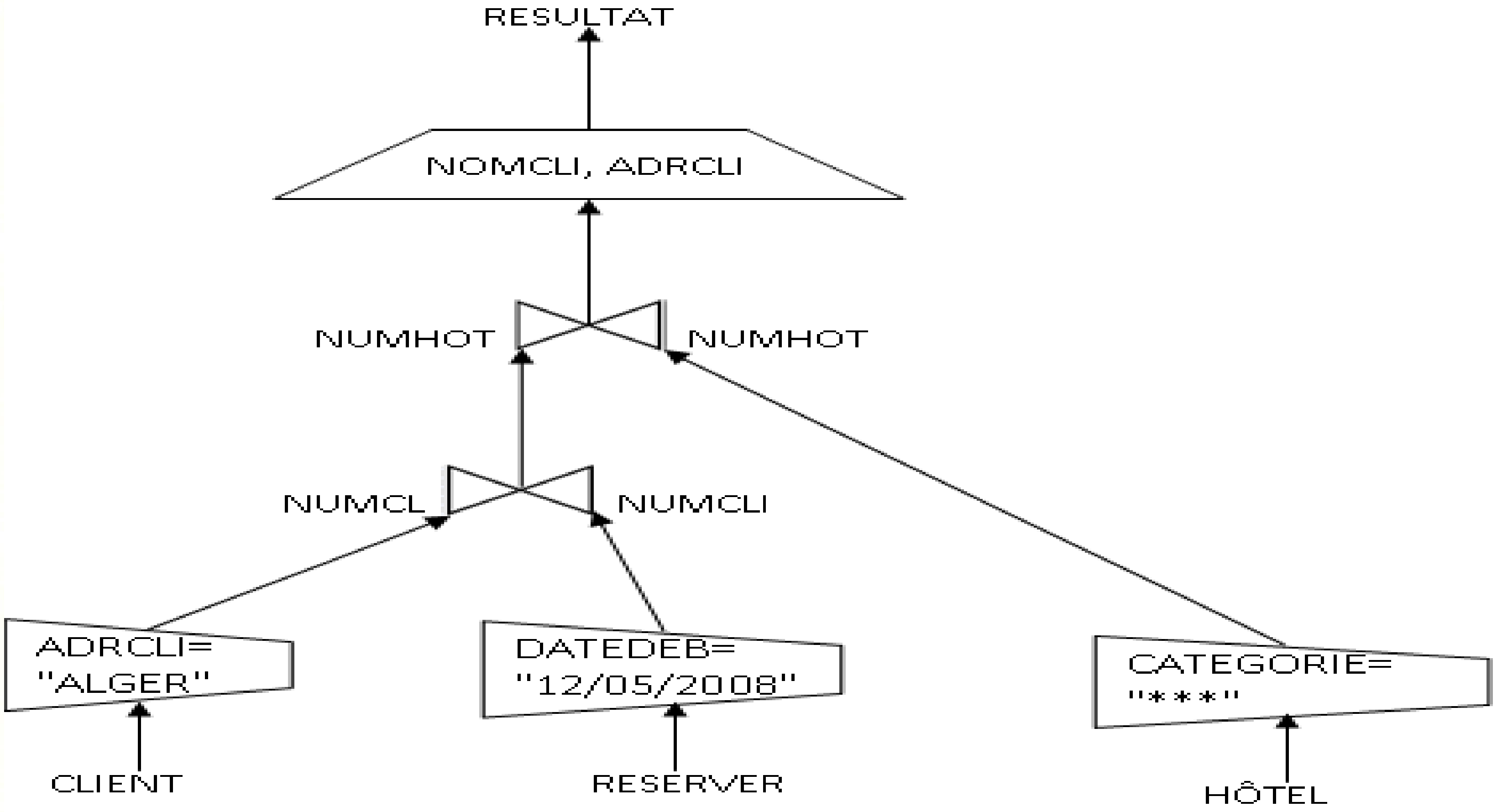
Arbre algébrique

- Une question exprimée sous forme d'un programme d'opérations de l'algèbre relationnelle peut être représentée par un ***arbre relationnel***.
- L'arbre relationnel est un arbre dont les nî uds correspondent aux représentations graphiques des opérations de l'algèbre relationnelle et les arcs à des relations temporaires ou de base représentant des flots de données entre opérations.
- Plusieurs arbre équivalents peuvent être déduits d'un arbre donné à l'aide de règles de transformation simples, telles que la permutation des jointures et restrictions, la permutation des projections et des jointures, le regroupement des intersections sur une même relation, etc.
- Ces transformations sont à la base des techniques d'optimisation de question
- La composition d'opérations de l'algèbre relationnelle ne nécessite pas toujours d'attendre le résultat de l'opération précédente pour exécuter l'opération suivante.
- Les opérations de restrictions, jointures et projections peuvent ainsi être exécutées par des algorithmes à flots de données.
- Les opérateurs qui se succèdent dans un arbre peuvent être exécutés en parallèle par des algorithmes "pipe-line"
- Les opérateurs figurants sur des branches distinctes d'un arbre peuvent aussi être exécutés en parallèle de manière indépendante.
- La représentation par arbre algébrique met ainsi en évidence les possibilités de parallélisme et les enchaînements nécessaires. Ces propriétés sont importantes pour optimiser les questions dans des contextes parallèles.

Algèbre relationnelle

Exemple :

La question "Noms et Adresses des clients habitant Alger ayant effectué une réservation le 12/05/2008 dans un hôtel de trois étoiles" peut être exprimée à l'aide de l'arbre suivant :



Fin Algèbre relationnelle