François Jacquenet
Professeur d'Informatique
Faculté des Sciences
Laboratoire Hubert Curien – UMR CNRS 5516
18 rue Benoit Lauras
42000 Saint-Etienne
Tél : 04 77 91 58 07
e-mail : Francois.Jacquenet@univ-st-etienne.fr
Web : http://labh-curien.univ-st-etienne.fr/~fj/bd

Licence de Sciences et Techniques

Unité d'enseignement **BASES DE DONNEES**

Le modèle relationnel
L'algèbre relationnelle

Que verrons nous dans ce cours ?

- Le modèle relationnel
- L’algébre relationnelle

Partie I
Le modèle relationnel

Le modèle relationnel

- Modèle de niveau logique, très simple

- Défini par en ; prix Turing en 1986. Développé au centre de recherche d'IBM (Almaden, Californie)

- Aujourd'hui utilisé par beaucoup de SGBD commerciaux (Oracle, Informix, DB2, Ingres, Sybase, dBase, Access …) et SIG

- Modèle à deux concepts:

Concepts de base

Etudiant (N° Etud, nom, prénom, age)

Etudiant

N° Etud	nom	prénom	age
136	Dupont	Jean	19
253	Aubry	Annie	20
101	Duval	André	21
147	Dupont	Marc	21

Schéma relationnel

- une BD =

- schéma d'une BD relationnelle =

- schéma d'une relation =

  Ri = (A1/d1, A2/d2, …, Ay/dy)

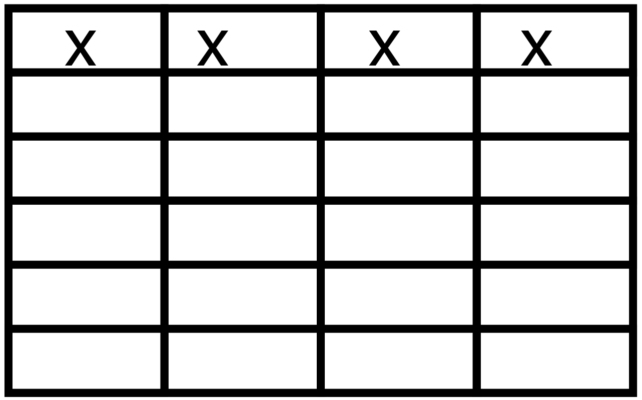
  ou, plus simplement,

  Ri = (A1, A2, …, Ay)

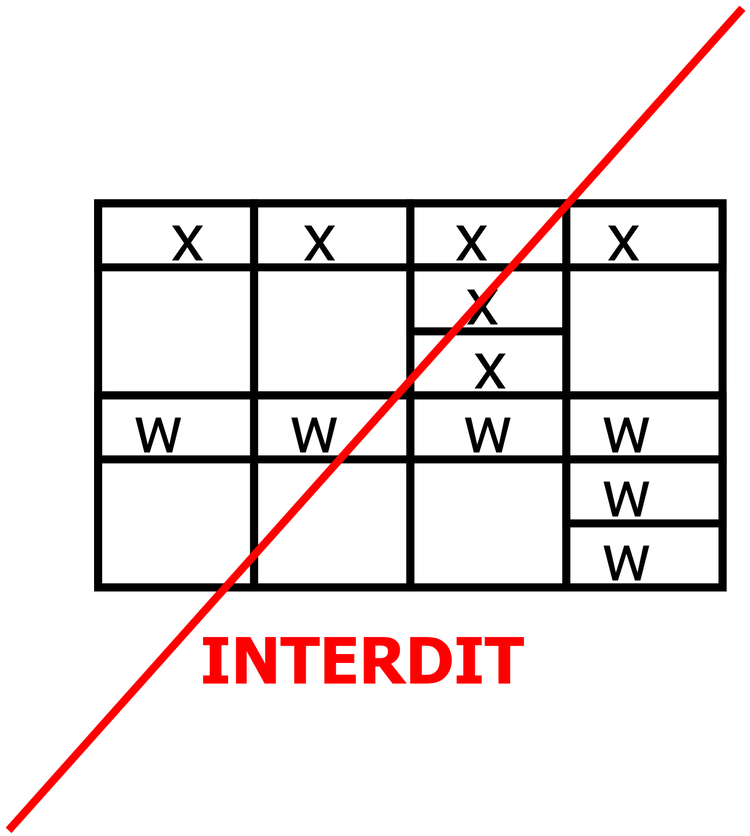
Règles de structuration

- attributs: et (domaine de valeurs atomiques)
- structure plate régulière

tuple



x: une et une seule valeur atomique par attribut



Valeurs nulles

- Un attribut peut ne pas être valué pour un tuple: on dit alors qu'il a une
  - exemple : on ne connaît ni l'age d'Annie ni le prénom de Duval

136	Dupont	Jean	19
253	Aubry	Annie	
101	Duval	21	
147	Dupont	Marc	21

Les identifiants

- Toute relation possède un
  - il ne peut y avoir deux tuples identiques dans la même relation
- ATTENTION :

Etudiant	<u>N° Etud</u>	nom	prénom	age
	136	Dupont	Jean	19
	253	Aubry	Annie	20
	101	Duval	André	21
	147	Dupont	Marc	21

Identifiant externe

- Cours(<u>NomC</u>,horaire,prof)

BD	Mercredi 15-17	Duval
SE	Mardi 16-19	Malin

- Suit(N° Etud,NomC)

253	SE
136	BD
253	BD
101	SE

Suit traduit un TA entre Etudiant et Cours.
Elle comporte les identifiants de Etudiant et de Cours.
Suit.NomC est un identifiant externe sur Cours.

Domaine de valeurs

- Domaine = Ensemble de valeurs atomiques que peut prendre un attribut

- Exemples de domaines:
  - Dnom : chaînes de caractères de longueur maximale 30
  - Dnum : entiers compris entre 0 et 99999
  - Dcouleur : {"bleu", "vert", "jaune"}
  - Dâge : entiers compris entre 16 et 65

Définition d'une relation

- Une relation est définie par :
  - son
  - sa <nom d'attribut : domaine>
  - son (ses) (s)
  - la (une phrase)

- Exemple :
  - **Etudiant** (**N° Etud : Dnum, Nom : Dnom, Prénom : Dnom, Age : Dâge**)

    Identifiant : N° Etud

    Définition : tout étudiant actuellement immatriculé à l'UJM

Contraintes de modélisation

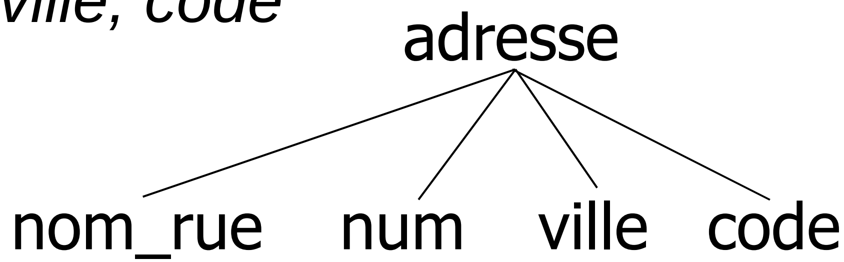
- Les notions d'attribut multivalué ou complexe n'existent pas dans le modèle relationnel.
  Il faut donc les modéliser autrement.

- Pour un attribut complexe, il faut choisir entre le composé ou les composants

- Pour un attribut multivalué, il faut créer une autre relation (ceci pour chaque attribut multivalué)

Représentation d'attribut complexe

- Soit *adresse : nom_rue , num , ville, code*



- Solution 1
  - un attribut par composant:

    nom_rue , num , ville, NPA

    "Rue de la presse", "5", "Saint-Etienne", "42000"
  - il est éventuellement possible de définir par ailleurs une vue restituant la notion globale d'adresse

- Solution 2
  - un attribut adresse dont le domaine est une chaine de caractères

    "Rue de la presse 5 Saint-Etienne 42000"

Représentation d'attribut multivalué

Exemple: mémoriser les différents prénoms des étudiants

- Solution incorrecte

  Plusieurs attributs : Prénom1, Prénom2,…
- Solution correcte : créer une relation supplémentaire:

PrénomsEtudiants(Num_Etud , Prénom)

136	Jean
136	Marie
101	André
253	Annie
253	Claudine

Ou liste ordonnée:

EtudPrénoms2 (N° Etud, N° Prénom, Prénom)

Identifiant d'une relation

- Une relation peut avoir plusieurs identifiants

PrénomsEtudiants2 (N° Etud, N° Prénom, Prénom)

  - N° Etud + N° Prenom
  - N° Etud + Prenom

- Identifiant =

- →

Identifiants externes

- Décrivent
- Suit (N° Etud : Dnum, NomC : Dnom)

  N° Etud **référence un** Etudiant

  NomC **référence un** Cours
- Si la relation référencée possède plusieurs identifiants, il faut préciser:

  N° Etud **référence un** Etudiant.N° Etud

- La vérification de est assurée par le SGBD: les identifiants externes doivent nécessairement désigner des tuples existants.

Récapitulatif

Un schéma relationnel se compose:

- pour chaque relation de:
  - nom de la relation
  - définition
  - attributs + domaines
  - identifiant(s)
  - éventuellement identifiant(s) externe(s)
  - contraintes d'intégrité associées
- et des autres contraintes d'intégrité qui portent sur plusieurs relations.

Exemple de schéma relationnel

Domaines :

- Dnom : chaînes de caractères de longueur inférieure à 30
- Dch100 : chaînes de caractères de longueur inférieure à 100
- Dannée : [1970 , 1990]
- Dnote : [0.0 , 20.0]
- Ddate : [1 , 31] / [1 , 12] / [1920 , 1990]

Exemple de schéma relationnel

- **Relation** : *Personne*

  Attributs : *n° P* : entier sans nul
  nom : *Dnom* sans nul
  adr : *Dch100* sans nul

  Identifiant: : *n° P*

  Définition: *tout étudiant et tout enseignant de l'école (état actuel).*

- **Relation** : *PersonnePrénoms*

  (permet d'avoir une liste de prénoms pour une personne donnée)

  Attributs : *n° P* : entier sans nul
  prénom : *Dnom* sans nul

  Identifiant: : *n° P + prénom*

  Identifiant externe : *n° P* référence une *Personne*

  Définition : *prénoms des personnes*

Exemple de schéma relationnel

- **Relation** : *Etudiant*

  Attributs : *n° P* : entier sans nul
  n° E : entier sans nul
  dateN : *Ddate* sans nul

  Identifiants : (*n° E*)
  (*n° P*)

  Identifiant externe : *n° P* référence une *Personne*

  Définition : *tout individu qui est actuellement inscrit à l'école, ou qui a déjà passé avec succès un des cours de l'école*

- **Relation** : *EtudiantEtudes*

  Attributs : *n° E* : entier sans nul
  année : *Dannée* sans nul
  diplôme : *Dnom* sans nul

  Identifiant: (*n° E* + *diplôme*)

  Identifiant externe : *n° E* référence un *Etudiant.n° E*
  Définition : *études antérieures des étudiants*

Exemple de schéma relationnel

- **Relation** : *Enseignant*

  Attributs: *n° P* : entier sans nul
  tel: : entier sans nul
  statut : *Dnom* sans nul
  banque : *Dnom* sans nul
  agence : *Dnom* sans nul
  compte : entier sans nul

  Identifiant : (*n° P*)
  Identifiant externe : *n° P* référence une *Personne*
  Définition : *tout individu assurant actuellement un ou plusieurs cours à l'école*

- **Relation** : *Cours*

  Attributs : *nomC* : *Dnom* sans nul
  cycle : entier sans nul
  n° Ens : entier sans nul

  Identifiant : (*nomC*)
  Identifiant externe : *n° Ens* référence un *Enseignant*
  Définition : *tout cours actuellement offert par l'école*

Exemple de schéma relationnel

- **Relation** : *Obtenu*

  Attributs: *n° E* : entier sans nul
  nomC : *Dnom* sans nul
  note : *Dnote* sans nul
  année : *Dannée* sans nul

  Identifiant : (*n° E* + *nomC*)

  Identifiants externes : *n° E* référence un *Etudiant.n° E*
  nomC référence un *Cours*

  Définition : *l'étudiant n° E a réussi le cours nomC telle année et a obtenu telle note*

- **Relation** : *Inscrit*

  Attributs: *n° E* : entier sans nul
  nomC : *Dnom* sans nul
  Identifiant : (*n° E* + *nomC*)
  Identifiants externes : *n° E* référence un *Etudiant.n° E*
  nomC référence un *Cours*
  Définition : *actuellement, l'étudiant n° E est inscrit au cours nomC*

Exemple de schéma relationnel

- **Relation** : *Prérequis*

  Attributs: *nomC* : *Dnom* sans nul

  nomCprérequis : *Dnom* sans nul

  Identifiant : (*nomC* + *nomCprérequis*)

  Identifiants externes : *nomC* référence un *Cours*

  nomCprérequis référence un *Cours*

  Définition: *le cours nomCprérequis est un prérequis pour le cours nomC*

  Contrainte d'intégrité : *dans tout tuple, nomCprérequis doit être différent de nomC*

Exemple de schéma relationnel

Contraintes d'intégrité :

- Pour tout tuple de Prérequis <nomC, nomCprérequis>, le cycle de nomCprérequis dans Cours doit être inférieur ou égal à celui de nomC:
  - <x,y> in Prérequis => x.cycle > y.cycle
- Pour tout tuple de EtudiantEtudes <n° E,année,diplôme>, soit dateN la date de naissance des étudiants dans la relation Etudiant, alors:
  dateN < année:
  - <x,y,z> in EtudiantEtudes, <a,x,d> in Etudiant
    => d < y

Exemple de schéma relationnel

Contraintes d'intégrité (suite):

- Pour tout tuple de Etudiant : cette_année - dateN = 18, où cette_année est une variable du SGBD
- Pour tout tuple de Obtenu <n° E, nomC, note, année>, soit dateN la date de naissance de l'étudiant dans la relation Etudiant, alors : dateN < année
- Pour tout tuple de Inscrit <n° E, nomC>, le n° E doit exister dans Obtenu associé à tous les cours existant dans Prérequis associés à nomC.

Partie II
L’Algèbre relationnelle

Les opérateurs

- Les opérateurs de l'algèbre relationnelle
  - Sélection
  - Projection
  - Renommage
  - Produit cartésien
  - Jointure
  - Union
  - Intersection
  - Différence
  - Division

Langages de manipulation

- Langages : base théorique solide
- Langages : version plus ergonomique

- Langages : définissent comment deriver le résultat souhaité
- Langages ou : définissent le résultat souhaité

LMD classiques

- Langages
  - langages : définissent un ensemble d'opérateurs de manipulation
  - langages : définissent le résultat souhaité en utilisant des expressions de logique
- Langages
  - Langages : SQL
  - Langage : QBE, QUEL

L’approche algèbrique

- Une algèbre est un de base, formellement définis, qui peuvent être combinés à souhait pour construire des expressions algèbriques
- Une algèbre est dite si le résultat de tout opérateur est du même type que les opérandes (ce qui est indispensable pour construire des expressions)
- : toute manipulation pouvant être souhaitée par les utilisateurs devrait pouvoir être exprimable par une expression algèbrique

L'algèbre relationnelle

- Opérandes:
- Fermeture:

- Complétude: permet (à peu prêt) toute opération
- Opérations unaires (une seule opérande):

  (noté σ), (π), (α)

- Opérations binaires:

  ($\times$), ($\bowtie$), ($\cup$),

  ($\cap$), ($-$), ($/$)

Préambule

- Pour chacune de ces 9 opérations, on va donner :
  - l 'opération
  - la syntaxe (notation)
  - la sémantique (résultat attendu)
  - le schéma
  - d'éventuelles remarques
  - un exemple

Sélection σ

-

- Syntaxe:

p: prédicat de sélection (condition de sélection)

< prédicat-élémentaire opérateur-logique prédicat-élémentaire >

opérateur-logique $\in$ { et, ou }

prédicat-élémentaire :

[non] attribut opérateur-de-comparaison constante-ou-attribut

attribut est un attribut de la relation R

opérateur-de-comparaison $\in \{=, \leq, <, >, \geq, \neq\}$

Sélection σ

- sémantique : de population l'ensembles des tuples de R qui satisfont le prédicat p

- schéma (résultat) schéma (opérande)

- population (résultat) population (opérande)

- exemple : **Petit-pays** **=** **σ** **[surface < 100]** **Pays**

Sélection σ

Exemple : soit la relation Pays

Pays	nom	capitale	population	surface
	Autriche	Vienne	8	83
	UK	Londres	56	244
	Suisse	Berne	7	41

On ne veut que les pays dont la valeur de surface est inférieure à 100 :

Petit-pays = σ [surface < 100] Pays

Petit-pays	nom	capitale	population	surface
	Autriche	Vienne	8	83
	UK	Londres	56	244
	Suisse	Berne	7	41

Projection π

-

- syntaxe:
  - attributs: liste l'ensemble d'attributs de R à conserver dans le résultat

- sémantique : crée une nouvelle relation de population l'ensembles des tuples de R réduits aux seuls attributs de la liste spécifiée

- schéma (résultat) schéma (opérande)

- nb tuples (résultat) nb tuples (opérande)

Projection π

Exemple : soit la relation Pays

Pays	nom	capitale	population	surface
	Autriche	Vienne	8	83
	UK	Londres	56	244
	Suisse	Berne	7	41

On ne veut que les attributs nom et capitale:

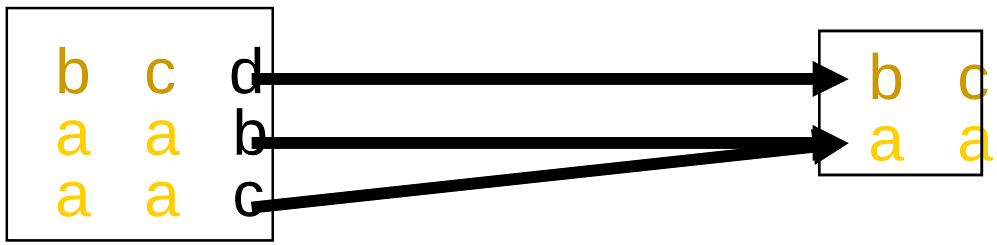
Capitales = π [nom, capitale] Pays

Capitales	nom	capitale	population	surface
	Autriche	Vienne	8	83
	UK	Londres	56	244
	Suisse	Berne	7	41

Effet de bord de la projection

- Elimination des doublons
  - une projection qui ne conserve pas la clé de la relation peut générer dans le résultat deux tuples identiques (à partir de deux tuples différents de l'opérande)
  - le résultat ne gardera que des tuples différents (fermeture)

$R\,(X, Y, Z)$ $\qquad\qquad$ $\pi\,(X, Y)\ R$



trois tuples $\qquad\qquad$ deux tuples

Sélection-projection

- On veut les capitales des petits pays :
  - **Petit-pays = σ [surface < 100] Pays**
  - **Capitales = π [nom , capitale] Petit-pays**

Capitale-petit-pays = π [nom, capitale] σ [surface < 100] Pays

Pays

nom	capitale	population	surface
Irlande	Dublin	3	70
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Renommage α

- but: résoudre des problèmes de compatibilité entre noms d'attributs de deux relations opérandes d'une opération binaire
-
- syntaxe :
- sémantique : les tuples de R avec un nouveau nom de l'attribut
- schéma : schéma (α [n:m] R) le même que schéma (R) avec n renommé en m
- précondition : le nouveau nom n'existe pas déjà dans R
- exemple : **R1**

A	B
a	b
y	z
b	b

R2 = α [B: C] R1

R2

A	C
a	b
y	z
b	b

Produit cartésien $\times$

-

- but : construire toutes les combinaisons de tuples de deux relations (en général, en vue d'une sélection)
- syntaxe :
- sémantique : chaque tuple de R est combiné avec chaque tuple de S

- schéma (R $\times$ S) **=** schéma(R) schéma(S)
- précondition: R et S n'ont pas d'attributs de même nom (sinon, renommage des attributs avant de faire le produit)

Produit cartésien $\times$

Exemple

R

A	B
a	b
b	c
c	b

n tuples

S

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

m tuples

R $\times$ **S**

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

n x m tuples

Jointure naturelle $\bowtie$

-

- syntaxe : $\bowtie$

- sémantique : combine certains tuples

- schéma (R $\bowtie$ S) = schéma (R) schéma (S)
  - les attributs de même nom n'apparaissent qu'une seule fois

- la combinaison exige l'égalité des valeurs de tous les attributs de même nom de R et de S
  - si R et S n'ont pas d'attributs de même nom la jointure peut être dynamiquement remplacée par un produit cartesien

Jointure naturelle $\bowtie$

- but : créer toutes les combinaisons significatives entre tuples de deux relations
  - significatif = portent la même valeur pour les attributs de même nom
- précondition : les deux relations ont au moins un attribut de même nom
- exemple :

R

A	B
a	b
b	c
c	b

S

B	C	D
b	c	d
a	a	b
d	a	c

R $\bowtie$ S

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

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 $\neq$ 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 $\neq$ 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

Theta-jointure $\bowtie[p]$

- opération binaire

- syntaxe : R $\bowtie$ [p] S
  - p: prédicat/condition de jointure
  - < prédicat-élémentaire et/ou prédicat-élémentaire >

- sémantique : combine les tuples qui satisfont le prédicat

- schéma (R $\bowtie$ [p] S) = schéma (R) $\cup$ schéma (S)

Union ∪

-
- syntaxe :
- sémantique : réunit dans une même relation les tuples de R et ceux de S
- schéma(R ∪ S) schéma(R) schéma(S)
- précondition : schéma(R) = schéma(S)
- exemple :

R1

A	B
a	b
b	b
y	z

R2

A	B
u	v
y	z

R1 ∪ R2

A	B
a	b
b	b
y	z
u	v

Intersection ∩

-
- syntaxe :
- sémantique : sélectionne les tuples qui sont à la fois dans R et S
- schéma (R ∩ S)   schéma (R)   schéma (S)
- précondition : schéma (R) = schéma (S)
- exemple :

R1

A	B
a	b
y	z
b	b

R2

A	B
u	v
y	z

R1 ∩ R2

A	B
y	z

Différence -

-
- syntaxe :
- sémantique : sélectionne les tuples de R qui ne sont pas dans S
- schéma (R - S) schéma (R) schéma (S)
- précondition : schéma (R) = schéma (S)
- exemple :

R1

A	B
a	b
y	z
b	b

R2

A	B
u	v
y	z

R1 - R2

A	B
a	b
b	b

La division /

But : traiter les requêtes du style «les … tels que TOUS les …»

soient R(a1, …, an) et V(a1, …, am)

avec n>m et A1, …, Am des attributs de même nom dans R et V

$R/V = \{ \langle am+1, am+2, \ldots, an \rangle \;/\; \forall \langle a1, a2, \ldots, am \rangle \in V,$
$\exists \langle a1, a2, \ldots, am, am+1, am+2, \ldots, an \rangle \in R \}$

- exemples :

R

A	B	C
1	1	1
1	2	0
1	2	1
1	3	0
2	1	1
2	3	3
3	1	1
3	2	0
3	2	1

V

B	C
1	1
2	0

R/V

A
1
3

V’

B	C
1	1

R/V’

A
1
2
3

V’’

B	C
3	5

R/V’’

A

La division /

Exemple

R

Etudiant	Cours	Obtenu
François	RDB	yes
François	Maths	yes
François	Prog	yes
Jacques	RDB	yes
Pierre	Prog	yes
Pierre	RDB	no

V

Cours	Obtenu
Prog	yes
RDB	yes

R/V

Etudiant
François

Equivalences

- $R \cap S = S \cap R$, $R \bowtie S = S \bowtie R$, etc.
- $\sigma\ [p1]\ (\sigma\ [p2]\ R) = \sigma\ [p2]\ (\sigma\ [p1]\ R) = \sigma\ [p2 \text{ et } p1]\ R$
- $\sigma\ [p]\ (\pi\ [a]\ R) = \pi\ [a]\ (\sigma\ [p]\ R)$ si $\text{attributs}(p) \subseteq a$
- etc.

Exemples de requêtes algébriques

- Soient les relations suivantes :

Journal (code_j, titre, prix, type, périodicité)

Dépôt (no_depot, nom_depot, adresse)

Livraison (no_depot, code_j, date_liv, quantite_livree)

Exemples de requêtes algébriques

- Quel sont les prix de tous les journaux ?
  -
- Donnez tous les renseignements connus sur les journaux hebdomadaires.
  -
- Donnez les codes des journaux livrés à Lyon.
  -

Exemples de requêtes algébriques

- Donnez les numéros des dépôts qui reçoivent plusieurs journaux différents.
  -