SHAKE THE FUTURE

CENTRALE NANTES

Bases de Données

Algèbre Relationnelle

JY Martin

Plan

Les relations, comment obtenir des informations ?

On dispose d’un schéma contenant des relations

Qu’est-il possible de faire avec ces relations pour obtenir des informations à partir des données de la population des relations ?

Est-il possible de formaliser ces opérations pour faire en sorte que les opérations soient mathématiquement prouvable ?

Peut-on établir des propriétés sur les opérations possibles en vue d’optimiser le comportement des outils qui effectueront les actions en questions ?

Le principe de l'Algèbre Relationnelle

Contexte

Opérandes : les relations
Opérateurs : des opérations sur les opérandes

Principe

Sélectionner les informations désirées, les exprimer sous forme d'une relation obtenue par applications successives d'opérateurs unaires ou binaires sur les opérandes.

L'Algèbre Relationnelle : les opérandes

Les opérandes de l'algèbre relationnelles sont les relations.
Relation(Attribut_1, Attribut_2, ... Attribut_n)

Les relations possèdent un identifiant.

L'Algèbre Relationnelle : les opérateurs

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

- Propriété des algèbres : fermeture
  Le résultat de tout opérateur est du même type que les opérandes
- Propriété souhaitée :
  Toute manipulation pouvant être souhaitée par les utilisateurs devrait pouvoir être exprimable par une expression algébrique

L'Algèbre Relationnelle : les opérateurs

Les opérateurs de l'algèbre relationnelles sont appliqués sur des relations et doivent fournir une relation comme résultat.

- Unaires
  - Sélection
  - Projection
  - Renommage
  - Négation
- Binaires
  - Produit cartésien, Jointure et opérations dérivées
  - Opérations ensemblistes (Union, Intersection, Différence)
  - Division

Plan

Les opérateurs unaires

- Ils s'appliquent sur une seule relation
- Ils permettent :
  - De sélectionner des tuples dans la population de la relation
  - De ne sélectionner que certains attributs
  - De renommer les attributs
  - prendre le complément de la relation

La sélection : σ

But : ne retenir que certains tuples dans une relation

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

Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Petit-Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
Suisse	Berne	7	41

La sélection : σ

Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Petit-Pays = σ [Surface <100] Pays

Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

La sélection : σ

- Opération Unaire
- Syntaxe σ [condition] R
  - R : relation
  - condition est une condition composée d'attributs de la relation, de valeurs fixes, d'opérateurs de comparaison

La sélection : σ

- σ [nom = capitale] Pays

Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41
Luxembourg	Luxembourg	0,92	0,002

- σ [((surface >100) AND (surface <500)) OR (Population >30) AND (Population <300))] Pays

La sélection : σ

- Opération Unaire
- Syntaxe σ [condition] R
    - R : relation
    - condition est une condition composée d'attributs de la relation, de valeurs fixes, d'opérateurs de comparaison
- Sémantique : crée une nouvelle relation avec la population correspondant à l'ensemble des tuples qui satisfont la condition
- Schéma(résultat) = schéma(R)
- population(résultat) <= population(opérande)

La projection : π

But : ne retenir que certains attributs dans une relation

Exemple : On ne veut que le nom et la capitale

Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Capitales	
Nom	Capitale
Autriche	Vienne
UK	Londres
Suisse	Berne

La projection : π

Pays			
Nom	Capitale	Population	Surface
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Capitales = π [Nom, Capitale] Pays

Capitales	
Nom	Capitale
Autriche	Vienne
UK	Londres
Suisse	Berne

La projection : π

- Opération Unaire
- Syntaxe π [attributs] R
  - R : relation
  - attributs est une liste d'attributs appartenant au schéma de R
- Sémantique : crée une nouvelle relation avec une population égale à l'ensemble des tuples de la relation d'origine réduits aux seuls attributs de la liste spécifiée
- Schéma(résultat) = π [attributs] schéma(R)
- population(résultat) <= population(opérande)

La projection : π

population(résultat) <= population(opérande) ?
Ne devrait-elle pas être égale ?

Livres			
Numéro	Genre	Auteur	Titre
JRR01	Fantasy	JRR Tolkien	Le Seigneur des Anneaux
JRR02	Fantasy	JRR Tolkien	Le Silmarillon
HM001	Manga	Hiro Mashima	Fairy Tail
KK01	Manga	Kumo Kagyu	Goblin Slayer
DE001	Fantasy	David Eddings	La Belgariade

TypesAuteurs = π [Genre, Auteur] Livres ?

La projection : π

Livres			
Numéro	Genre	Autheur	Titre
JRR01	Fantasy	JRR Tolkien	Le Seigneur des Anneaux
JRR02	Fantasy	JRR Tolkien	Le Silmarillon
HM001	Manga	Hiro Mashima	Fairy Tail
KK01	Manga	Kumo Kagyu	Goblin Slayer
DE001	Fantasy	David Eddings	La Belgariade

TypesAuteurs = π [Genre, Auteur] Livres

TypesAuteurs	
Genre	Autheur
Fantasy	JRR Tolkien
Manga	Hiro Mashima
Manga	Kumo Kagyu
Fantasy	David Eddings

On ne conserve pas les n-uplets identiques

La projection : π

Remarque

Toute projection qui ne conserve pas l'identifiant de la relation d'origine peut avoir une population inférieur à celle d'origine.

Remarques

Seules les projections qui conservent l'identifiant de la relation d'origine assurent que la population de la projection est égale à la population d'origine

Renommage : α

But : L'opérateur de renommage a été introduit pour résoudre des problèmes de compatibilités dans certains opérateurs binaires. Il consiste à renommer certains attributs de la relation d'origine.

Livres			
Numéro	Genre	Autheur	Titre
JRR01	Fantasy	JRR Tolkien	Le Seigneur des Anneaux
JRR02	Fantasy	JRR Tolkien	Le Silmarillon
HM001	Manga	Hiro Mashima	Fairy Tail
KK01	Manga	Kumo Kagyu	Goblin Slayer

LivresBis = α [Genre - >Type] Livres

LivresBis			
Numéro	Type	Autheur	Titre
JRR01	Fantasy	JRR Tolkien	Le Seigneur des Anneaux
JRR02	Fantasy	JRR Tolkien	Le Silmarillon
HM001	Manga	Hiro Mashima	Fairy Tail
KK01	Manga	Kumo Kagyu	Goblin Slayer

Renommage : α

- Opération Unaire
- Syntaxe α [attribut - >nouvelattribut, ...] R
  - R : relation
  - entre crochets figurent les renommage des attributs
    Les nouveaux noms d'attributs ne doivent pas déjà exister dans le schéma de la relation
- Sémantique : crée une nouvelle relation en renommant les attributs de la relation
- Schéma(résultat) = α [attribut ->nouvelattribut, ...] schéma(R)
- population(résultat) = population(opérande)

Complément : $\neg$

But : Construire une nouvelle relation avec le même schéma mais dont la population est le complément de la relation d'origine

Exemple :

- R(A, B)
- D(A) = {a,b}
- D(B) = {c,d}

R	
A	B
a	c
a	d
b	c

NotR	
A	B
b	d

Complément : $\neg$

- Opération Unaire
- Syntaxe $\neg$ R
  - R : relation
  - Les attributs de R doivent avoir un domaine de valeur fini
- Sémantique : crée une nouvelle relation dont la population est le complément de lma relation d'origine
- Schéma(résultat) = schéma(R)
- population(résultat) = produit des cardinalités des domaines - population(opérande)

Plan

Les opérateurs binaires

- Les opérateurs binaires s'appliquent sur 2 relations
- ils comportent
  - des opérations de jointures internes
    produit cartésien, jointure naturelle, théta-jointure, équi-jointure, semi-jointure, anti-jinture
  - des opérations de jointure externe
    jointure externe entière, gauche et droite
  - des opérations ensemblistes
    union, intersection, différence, division

Le produit cartésien ou jointure : x

But : construire toutes les combinaisons possibles de tuples de deux relations

R1	
A	B
a	b
x	y

R2	
C	D
c	d
z	t

R1 x R2			
A	B	C	D
a	b	c	d
a	b	z	t
x	y	c	d
x	y	z	t

Le produit cartésien ou jointure : x

- Opération Binaire
- Syntaxe R1 x R2
  - R1, R2 : relations
  - R1 et R2 n'ont pas d'attribut commun
- Sémantique : crée une nouvelle relation en combinant tous les tuples de R1 avec tous les tuples de R2
- Schéma(résultat) = schéma(R1) x schema (R2)
- population(résultat) = population(R1) x population(R2)

La jointure naturelle : *

But : créer toutes les combinaisons significatives entre tuples de deux relations
Significatif = ont la même valeur pour tous les attributs **de même nom**

R1 * R2

R1	
A	B
a	b
x	b

R2	
B	C
b	c
z	t

R1 * R2		
A	B	C
a	b	c
x	b	c

La jointure naturelle : *

Exemple

Pays		
Nom	Population	Surface
Autriche	8	83
UK	56	244
Suisse	7	41

Capitale	
Nom	Ville
Autriche	Vienne
Suisse	Berne
France	Paris

PaysEtCapitale = Pays * Capitale

PaysEtCapitale			
Nom	Population	Surface	Ville
Autriche	8	83	Vienne
Suisse	7	41	Berne

La jointure naturelle : *

- Opération Binaire
- Syntaxe R1 * R2
  - R1, R2 : relations
  - R1 et R2 ont au moins un attribut commun
- Sémantique : crée une nouvelle relation en combinant tous les tuples de R1 avec tous les tuples de R2, et en vérifiant si les colonnes de même nom ont la même valeur
- Schéma(résultat) = schéma(R1) * schema (R2)
- population(résultat) <= population(R1) x population(R2)

NB : si les relations n'ont pas d'attributs communs, il s'agit d'un produit cartésien

La theta-jointure : *[c]

But : créer toutes les combinaisons significatives entre tuples de deux relations
Significatif = la condition explicitée est vraie

R = R1 *[B=C] R2

R1	
<u>A</u>	B
a	b
x	b

R2	
<u>C</u>	D
b	c
z	t

R			
A	B	C	D
a	b	b	c
x	b	b	c

La theta-jointure : *[c]

- Opération Binaire
- Syntaxe R1 *[c] R2
  - R1, R2 : relations
  - c est une condition
- Sémantique : crée une nouvelle relation en combinant tous les tuples de R1 avec tous les tuples de R2, et en vérifiant si la condition est respectée pour chacune des lignes
- Schéma(résultat) = schéma(R1) x schema (R2)
- population(résultat) <= population(R1) x population(R2)

NB : mathématiquement, R1 *[c] R2 <= > σ [c] (R1 x R2)

La semi-jointure : $\ltimes$

But : jointure naturelle entre deux relations, en ne gardant que les attributs d'une relation.

On distingue la semi-jointure à droite et la semi-jointure à gauche

La semi-jointure : $\ltimes$

- Opération Binaire
- Syntaxe R1 $\ltimes$ R2
  - R1, R2 : relations
- Sémantique : crée une nouvelle relation en combinant tous les tuples de R1 avec tous les tuples de R2, en vérifiant si les colonnes de même nom ont la même valeur, et en ne gardant que les colonnes d'une des relations
  - la semi-jointure à gauche garde les attributs de R1
  - la semi-jointure à droite garde les attributs de R2
- Schéma(résultat) = schéma(R1) ou schéma (R2) suivant le type de semi-jointure
- population(résultat) <= population(R1) ou population (R2)

Autres types de jointures

- Anti-jointure = semi-jointure retournant le complément des tuples sélectionnés
- Jointures externes = même principe qu'une jointure naturelle, mais si la condition de jointure n'est pas respectée, on garde tout de même le tuple en mettant NULL pour les attributs manquants

L'Union : $\cup$

But : Fusionner les populations de deux relations de schéma identiques

R = R1 $\cup$ R2

R1	
A	B
a	b
x	y

R2	
A	B
c	d
z	t

R1 $\cup$ R2	
A	B
a	b
c	d
x	y
z	t

L'Union : ∪

- Opération Binaire
- Syntaxe R1 ∪ R2
  - R1, R2 : relations
  - R1 et R2 doivent avoir le même schéma
- Sémantique : crée une nouvelle relation avec tous les tuples de R1 et tous les tuples de R2
- Schéma(résultat) = schéma(R1) = schema (R2)
- population(résultat) <= population(R1) + population(R2)

L'Intersection : $\cap$

But : Prendre les tuples communs de deux relations de schéma identiques

$R = R1 \cap R2$

R1	
<u>A</u>	B
a	b
x	y

R2	
<u>A</u>	B
c	d
a	b

$R1 \cap R2$	
<u>A</u>	B
a	b

L'Intersection : $\cap$

- Opération Binaire
- Syntaxe R1 $\cap$ R2
  - R1, R2 : relations
  - R1 et R2 doivent avoir le même schéma
- Sémantique : crée une nouvelle relation avec tous les tuples communs de R1 et R2
- Schéma(résultat) = schéma(R1) = schema (R2)
- pop(résultat) <= min (pop(R1) , pop(R2))

La différence : -

But : Prendre les tuples de la première relation qui ne sont pas dans la deuxième relation

R = R1 - R2

R1	
A	B
a	b
c	d

R2	
A	B
c	d
z	t

R1 - R2	
A	B
a	b

La différence : -

- Opération Binaire
- Syntaxe R1 - R2
  - R1, R2 : relations
  - R1 et R2 doivent avoir le même schéma
- Sémantique : crée une nouvelle relation avec tous les tuples de R1 qui ne sont pas R2
- Schéma(résultat) = schéma(R1) = schema (R2)
- population(résultat) <= population(R1)

La Division : /

But : Permettre de résoudre les requêtes particulières “tous les tuples qui ont un lien avec tous les tuples d'une autre relation”

Etudiants		
Nom	Prénom	Matière
STIRLING	Lindsey	Musique
STIRLING	Lindsey	Cinéma
DEL REY	Lana	Musique
TARANTINO	Quentin	Cinéma

Cours
Matière
Musique
Cinéma

Quels sont les étudiants inscrits aux cours de Musique ET Cinéma ?

La Division : /

Soient les relations

- R = (A1, A2, ... An)
- V = (B1, B2, ... Bm)

tels que tous les Bi sont dans R (il existe un j tel que Aj=Bi)

On désigne par R / V la relation dont le schéma est celui de R privé des attributs de V et dont la population est telle que tout tuple de R/V combinés à n'importe quel tuple de V appartient à la jointure naturelle de R et V.

La Division : /

Etudiants		
Nom	Prénom	Matière
STIRLING	Lindsey	Musique
STIRLING	Lindsey	Cinéma
DEL REY	Lana	Musique
TARANTINO	Quentin	Cinéma

Cours
Matière
Musique
Cinéma

Etudiants / Cours	
Nom	Prénom
STIRLING	Lindsey

La division : /

- Opération Binaire
- Syntaxe R1 / R2
  - R1, R2 : relations
  - Les attributs de R2 sont également dans R1
- Sémantique : crée une nouvelle relation avec tous les tuples de R1 pour lesquels les attributs de communs à R2 sont tous dans R2
- Schéma(résultat) = schéma(R1) - schema (R2)
- population(résultat) <= population(R1)

Plan

Combinaison des opérateurs

Comme nous travaillons sur une algèbre, nous pouvons combiner les opérateurs, les appliquer successivement pour obtenir des résultats plus complexes.

Question :

Peut-on combiner les opérations, les effectuer successivement ?

Question :

Quelles sont les propriétés sur les combinaisons d'opérations (Commutativité, Réflexivité, Transitivité, ...) ?

Combiner les opérateurs

On désire la surface des petits pays

Pays			
Nom	Capitale	Population	Surface
Irlande	Dublin	30	70
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Surface-petit-Pays	
Nom	Surface
Irlande	70
Autriche	83
Suisse	42

CENTRALE NANTES

Combiner les opérateurs

Petit-Pays = σ [Surface <100] Pays
Surface-Pays = π [Nom, Surface] Pays

Surface-Petit-Pays = π [Nom, Surface] Petit-Pays
= π [Nom, Surface] σ [Surface <100] Pays

Pays			
Nom	Capitale	Population	Surface
Irlande	Dublin	30	70
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Combiner les opérateurs

Mais aussi

Surface-Petit-Pays = σ [Surface <100] Surface-Pays
= σ [Surface <100] π [Nom, Surface] Pays

Pays			
Nom	Capitale	Population	Surface
Irlande	Dublin	30	70
Autriche	Vienne	8	83
UK	Londres	56	244
Suisse	Berne	7	41

Combiner les opérateurs : commutativité ?

Petit-Pays = σ [Surface <100] Pays
Surface-Pays = π [Nom, Surface] Pays
Capitale-pays = π [Nom, Capitale] Pays

Capitale des petits pays ?

Capitale-Petit-Pays = π [Nom, Capitale] Petit-Pays
= π [Nom, Capitale] σ [Surface <100] Pays

Capitale-Petit-Pays = σ [Surface <100] Capitale-Pays
= σ [Surface <100] π [Nom, Capitale] Pays
=>ERREUR !!!!

Combiner les opérateurs : commutativité ?

Conclusion

La combinaison des opérateurs n'est pas commutative

Cas particulier

Les opérations de Jointure (produit cartésien), Jointure Naturelle, Théta Jointure, Union, Intersection sont commutatives.
R1 x R2 = R2 x R1

Propriétés

Associativité

Les opérations de Jointure (produit cartésien), Jointure Naturelle, Théta Jointure, Union, Intersection sont associatives.

(R1 x R2) x R3 = R1 x (R2 x R3)

Autres propriétés

Regroupement des sélections :

σ [A2=b] σ [A1=a] R1 = σ [((A2=b) et (A1=a))] R1
(σ [A1=a] R1) U (σ [A2=b] R1) = σ [(A1=a) ou (A2=b)] R1

Combinaison des sélections et des unions :

σ [A1=a] (R1 U R2) = (σ [A1=a] R1) U (σ [A1=a] R2)

Autres propriétés

Combinaison des sélections et des projections

π [A1 ... Ap] σ [Aj=a] R1
si Aj $\in$ A1...Ap
= σ [Aj=a] π [A1 ... Ap] R1
sinon
= π [A1...Ap] σ [Aj=a] π[A1...Ap, Aj] R1

Combinaison des sélections et des jointures

R1 = (A1 ... An)
R2 = (B1 ... Bm)
σ [Ai=a et Bj=b] R1 x R2 = (σ [Ai=a] R1) x (σ [Bj=b] R2)

Plan

Utilisation de l'Algèbre Relationnelle

- Vérification des requêtes
- Mise en œuvre des requêtes
- Optimisation des requêtes

Optimisation des requêtes

L'optimiseur applique des modifications pour minimiser l'espace mémoire utilisé, minimiser le nombre de sonnées sur lequel il travaille et accélérer le temps de calcul.

En général, il applique la stratégie suivante :

- effectuer en premier les sélections et projections
- faire les produits et jointures avec le minimum de données
- regrouper les opérations successives de sélection et projection
- regrouper les suites de produits ou jointures et de projections

Exemple de mise en œuvre de requête

Supposons les relations suivantes et leur population

Pays		
Nom	Population	Surface
Autriche	8	83
UK	56	244
Suisse	7	41

Capitale		
Nom	Capitale	Habitants
Autriche	Vienne	1 730 000
UK	Londres	8 308 000
Suisse	Berne	124 000

On veut les capitales des petits pays.

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

Exemple de mise en œuvre de requête

Pays		
Nom	Population	Surface
Autriche	8	83
UK	56	244
Suisse	7	41

Capitale		
Nom	Capitale	Habitants
Autriche	Vienne	1 730 000
UK	Londres	8 308 000
Suisse	Berne	124 000

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

Nom	Population	Surface	Capitale	Habitants
Autriche	8	83	Vienne	1 730 000
UK	56	244	Londres	8 308 000
Suisse	7	41	Berne	124 000

Exemple de mise en œuvre de requête

Nom	Population	Surface	Capitale	Habitants
Autriche	8	83	Vienne	1 730 000
UK	56	244	Londres	8 308 000
Suisse	7	41	Berne	124 000

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

Nom	Population	Surface	Capitale	Habitants
Autriche	8	83	Vienne	1 730 000
UK	56	244	Londres	8 308 000
Suisse	7	41	Berne	124 000

Exemple de mise en œuvre de requête

Nom	Population	Surface	Capitale	Habitants
Autriche	8	83	Vienne	1 730 000
Suisse	7	41	Berne	124 000

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

Nom	Population	Surface	Capitale	Habitants
Autriche	8	83	Vienne	1 730 000
Suisse	7	41	Berne	124 000

Exemple de mise en œuvre de requête

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

Nom	Capitale
Autriche	Vienne
Suisse	Berne

Exemple de mise en œuvre de requête

Constats :

- Lors de la jointure, on a mis des colonnes qui ne nous servent à rien (population, habitants)
- Si on avait retiré les petits pays, on n'aurait pas eu à traiter les tuples concernés dans la jointure.

Exemple de mise en œuvre de requête

Essayons d'optimiser...
On commence par réécrire un peu la requête

$$
\begin{aligned}
\text{Capitale-petit-pays} &= \pi\ [\ \text{nom, capitale}\ ]\ \sigma\ [\text{surface} < 100]\ \text{Pays} * \text{Capitale} \\
&= \pi\ [\ \text{nom, capitale}\ ]\ (\ \sigma\ [\text{surface} < 100]\ \text{Pays}\ ) * \text{Capitale} \\
&= \pi\ [\ \text{nom, capitale}\ ]\ (\ \sigma\ [\text{surface} < 100]\ \text{Pays}\ ) \\
&\qquad\qquad * (\ \pi\ [\ \text{nom, capitale}\ ]\ \text{Capitale}\ ) \\
&= \pi\ [\ \text{nom, capitale}\ ] \\
&\qquad\qquad (\ \sigma\ [\text{surface} < 100]\ \pi\ [\ \text{nom, surface}\ ]\ \text{Pays}\ ) \\
&\qquad\qquad * (\ \pi\ [\ \text{nom, capitale}\ ]\ \text{Capitale}\ )
\end{aligned}
$$

Exemple de mise en œuvre de requête

Pays		
<u>Nom</u>	Population	Surface
Autriche	8	83
UK	56	244
Suisse	7	41

Capitale		
<u>Nom</u>	Capitale	Habitants
Autriche	Vienne	1 730 000
UK	Londres	8 308 000
Suisse	Berne	124 000

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

Pays		
<u>Nom</u>	Population	Surface
Autriche	8	83
UK	56	244
Suisse	7	41

Capitale		
<u>Nom</u>	Capitale	Habitants
Autriche	Vienne	1 730 000
UK	Londres	8 308 000
Suisse	Berne	124 000

Exemple de mise en œuvre de requête

Pays	
Nom	Surface
Autriche	83
UK	244
Suisse	41

Capitale	
Nom	Capitale
Autriche	Vienne
UK	Londres
Suisse	Berne

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

Pays	
Nom	Surface
Autriche	83
UK	244
Suisse	41

Capitale	
Nom	Capitale
Autriche	Vienne
UK	Londres
Suisse	Berne

Exemple de mise en œuvre de requête

Pays	
Nom	Surface
Autriche	83
Suisse	41

Capitale	
Nom	Capitale
Autriche	Vienne
UK	Londres
Suisse	Berne

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

Nom	Surface	Capitale
Autriche	83	Vienne
Suisse	41	Berne

Exemple de mise en œuvre de requête

Nom	Surface	Capitale
Autriche	83	Vienne
Suisse	41	Berne

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

Nom	Surface	Capitale
Autriche	83	Vienne
Suisse	41	Berne

Exemple de mise en œuvre de requête

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

Nom	Capitale
Autriche	Vienne
Suisse	Berne

Exemple de mise en œuvre de requête

Bilan de la modification

- Moins d'attributs manipulés (on a éliminé ceux qui n'étaient pas utilisé)
- Moins de tuples manipulés (on a éliminés les tuples avant la jointure)
- Plus d'opérations, ... mais sur moins de données

En clair :

- On a occupé moins d'espace mémoire
- Le temps de traitement est plus rapide
- Certaines opérations peuvent se faire en parallèle

Plan

Conclusion

Obtention d'une information
->traduction en algèbre relationnelle

Optimisation des instructions
Utilisation des propriétés de l'algèbre relationnelle

Recherche sur l'algèbre relationnelle
->requêtes de plus en plus optimisées


Success
NEXT EXIT