

Évaluation des requêtes dans le relationnel



Bernard ESPINASSE
Professeur à Aix-Marseille Université (AMU)
Ecole Polytechnique Universitaire de Marseille



Mars 2013

Analyse des requêtes

- Analyse syntaxique/sémantique
- Usage des multi-graphes de connexion
- Requêtes équivalentes par transitivité

Ordonnement des requêtes par restructuration

- Arbre algébrique
- Règles et heuristiques
- Estimation des volumes d'information manipulées

Évaluation des Requêtes dans le relationnel

3 Étapes :

1. Analyse

- Syntaxique
- Sémantique
 - > Vérifier correction
 - > 1° simplification du critère de recherche

2. Ordonnement

- > Décomposer la requête en séquences d'opérations élémentaires (Algèbre relationnelle).
- > Déterminer un ordre +- optimal.
Génération d'un plan d'exécution

3. Exécution

- > Exécution en // ou en séquences des opérations élémentaires selon le plan d'exécution défini en 2

Rôle des SGBD Relationnels

Analyse des Requêtes

1. Syntaxique :

- **Vérification :**
 - attributs cités appartiennent au schéma
 - qualification correcte (types)
- **Mise en forme :**
 - Normale Conjonctive (ET de OU) (si opérateurs élémentaires supportent le OU)
 - ou
 - Normale Disjonctive (OU de ET) (=> plusieurs requêtes)

2. Sémantique :

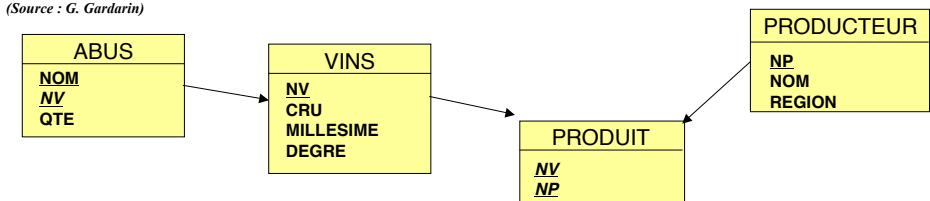
- **Vérification requête bien formulée :** Certaines parties apparaissent inutiles --> jointure oubliée ? ...
- **Requête non contradictoire :** (> 14 et < 13, ...)
- **Recherche de requêtes équivalentes par :** Manipulation de la qualification (transitivité), de contraintes d'intégrités (+rare), ...

Outils Disponibles : Multi-Graphes de connexion :

- des relations
- des attributs

Graphes de Connexion

(Source : G. Gardarin)



ABUS (NOM, NV*, QTE) (*NOM du buveur, NV*= clé étrangère*)
PRODUCTEUR (NP, NOM, REGION)
VINS (NV, CRU, MILLESIME, DEGRE)
PRODUIT (NV*, NP*)

Soit la requête (SQL) :

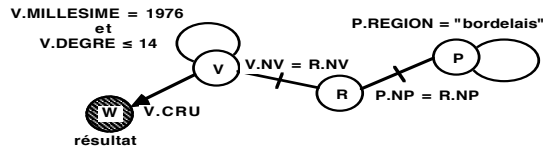
```
Select V.CRU
From PRODUCTEUR P, VINS V, PRODUIT R
Where V.MILLESIME = "1976"
and V.DEGRE ≤ 14
and P.REGION = "bordelais"
and P.NP = R.NP
and R.NV = V.NV;
```

Graphe de Connexion des Relations [ULLMAN 80]

Soit la requête (SQL) : Q1

```

Select V.CRU
From PRODUCTEUR P, VINS V, PRODUIT R
Where V.MILLESIME = "1976"
and V.DEGRE ≤ 14
and P.REGION = "bordeaux"
and P.NP = R.NP
and R.NV = V.NV;
    
```



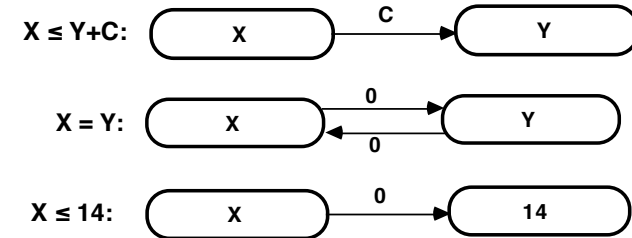
Conventions :

- **sommet**: relation
- **arc**:
 - jointure entre 2 relations
 - restriction sur une relation (boucle)

Graphe de Connexion des Attributs [ROSENKRANTZ 80]

Conventions :

- **sommet**: attribut ou constante
- **arc**:
 - jointure (entre 2 attributs)
 - restriction (attribut - constante)
- **valuation de l'arc**:



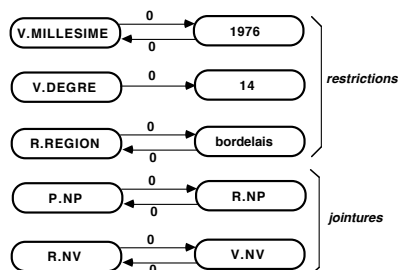
Graphe de Connexion des Attributs [ROSENKRANTZ 80]

Soit la requête (SQL) : Q1

```

Select V.CRU
From PRODUCTEUR P, VINS V, PRODUIT R
Where V.MILLESIME="1976"
and V.DEGRE≤14
and P.REGION="bordeaux"
and P.NP=R.NP
and R.NV=V.NV;
    
```

Graphe de connexion des attributs:



Graphes de connexion : correction

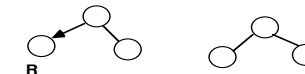
• **Correction de la Requête**

1. **Requête mal formulée** : Certaines parties semblent inutiles -> *Oublie de jointure par l'utilisateur*
2. Requête présente **qualification contradictoire** : ex: >14 et <12

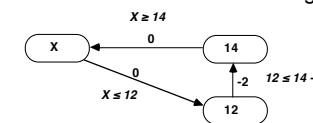
• **Utilisation des Graphes de Connexion**

Correction des requêtes conjonctives (sans OU) avec opérateurs de comparaison =, <, >, ≥, ≤, (pas de ≠) :

- Une requête est **mal formulée** si son **graphe de connexion des relations** n'est pas connexe [WONG 76]:

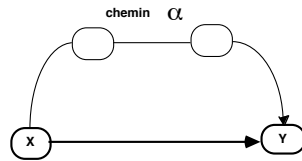


- Une requête est **contradictoire** si son **graphe de connexion des attributs** normalisé présente un cycle dont la somme des valuations est négative [ROSENKRANTZ 80]:



Graphes de connexion : requêtes équivalentes par transitivité

Utilisation des Graphes de Connexion : Transitivité d'un graphe de connexion des attributs [GOMEZ 78, BERNSTEIN 79]:



- Si:
- $X \leq Y + C$
 - il existe un chemin α joignant X à Y tel que $\sum \alpha$ valuations = C
- alors il y a fermeture transitive X -> Y

Soit G1 et G2, **sous-graphes de connexion d'attributs** associés à 2 requêtes Q1 et Q2 :

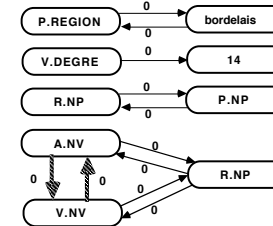
Si fermeture transitive (G1) = fermeture transitive (G2) alors Q1 \Leftrightarrow Q2

Graphes de connexion : requêtes équivalentes par transitivité

Soit la requête (SQL) : Q2

```
Select A.NOM
From BUVEURS A, PRODUCTEUR P, VINS V, PRODUIT R
Where
and A.NV=R.NV;
and R.NV=V.NV
and R.NP=P.NP
and P.REGION="bordelais"
and V.DEGRE<=14;
```

Graphe de connexion des relations:

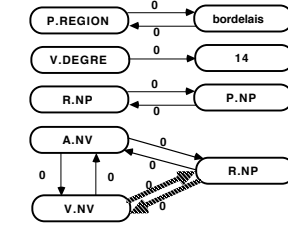


**même fermeture transitive, d'où: Q2 et Q2' équivalentes :
=> Choisir la jointure la plus facile à évaluer.**

Soit la requête (SQL) : Q2'

```
Select A.NOM
From BUVEURS A, PRODUCTEUR P, VINS V, PRODUIT R
Where
and A.NV=R.NV;
and A.NV=V.NV
and R.NP=P.NP
and P.REGION="bordelais"
and V.DEGRE<=14;
```

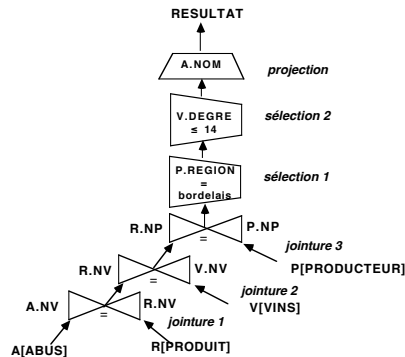
Graphe de connexion des relations:



Ordonnement par restructurations algébriques

• **Arbre algébrique :**

Retrieve A.NOM Where(A.NV=R.NV) \wedge (R.NV=V.NV) \wedge (R.NP=P.NP) \wedge
(P.REGION="bordelais") \wedge (V.DEGRE 14)



- A une question correspond plusieurs arbres (ordre \neq des opérations algébriques).
- Méthode simple : {J} + {R} + proj. Finale

Ordonnement par restructurations algébriques

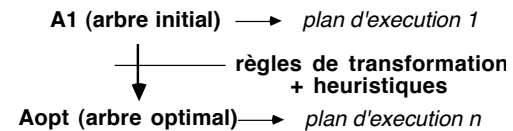
• **Arbre algébrique --> Plan d'exécution (feuille -> racine)**

• **Exécution en //**

• **Optimisation des temps de réponse :**

- Optimiser le nombre de E/S
- Le parallélisme entre E/S
- La taille des tampons nécessaires à l'exécution
- Le temps de CPU nécessaire.

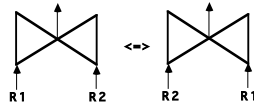
• **Optimisation par ordonnancement optimal des opérations**



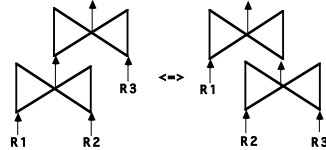
A1 \Leftrightarrow Aopt

Règles de Transformation des Arbres Algébriques [SMITH75;ULLMAN80]

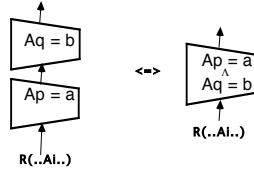
Règle 1 : Commutativité des Jointures



Règle 2 : Associativité des Jointures

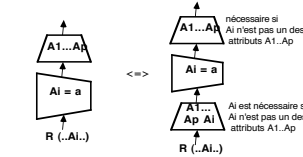


Règle 3 : Regroupement des Restrictions

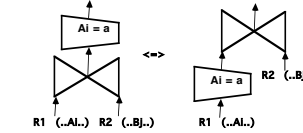


Règles de Transformation des Arbres Algébriques [SMITH75;ULLMAN80]

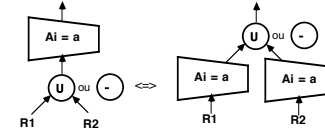
Règle 4 : Commutation des Restrictions et Projections



Règle 5 : Commutation des Restrictions et Jointures

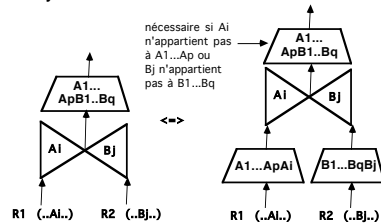


Règle 6 : Commutation des Restrictions et Unions ou des Restrictions et Différences

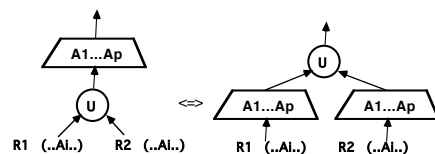


Règles de Transformation des Arbres Algébriques [SMITH75;ULLMAN80]

Règle 7 : Commutation des Projections et Jointures



Règle 8 : Commutation des Projections et Unions



Heuristiques d'Optimisation [ULLMAN 80]

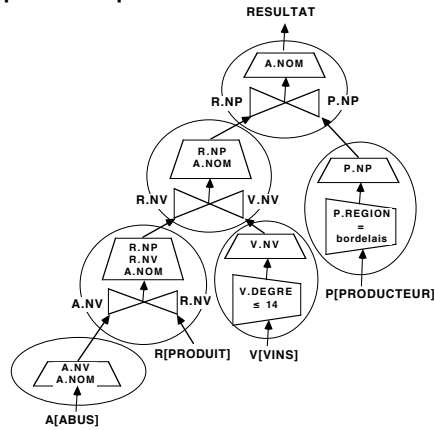
1. **Séparer les restrictions comportant plusieurs prédicats** à l'aide de la règle 3
2. **Descendre les restrictions aussi bas que possible** à l'aide des règles 4, 5 et 6. (Exécution d'abord dans opérations unaires)
3. **Regrouper les restrictions successives portant sur une même relation**
4. **Descendre les projections aussi bas que possible** à l'aide des règles 7 et 8
5. **Regrouper les projections successives** en conservant les attributs restants et éliminer d'éventuelles projections inutiles qui auraient pu apparaître.

Afin d'éviter plusieurs passes sur une même relation, **sont exécutées simultanément:**

- une restriction suivie par une projection
- une jointure suivie par une projection.

Heuristiques d'Optimisation [ULLMAN 80]

- Cet heuristique appliqué au cas précédent donne :



- Arbre pas forcément optimal : la descente des restrictions et projections n'est qu'une heuristique - l'ordre des opérations binaires n'a pas été optimisé.

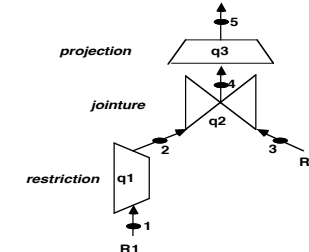
Estimation des volumes d'informations manipulées par une requête

Soit une requête Q, $V_{inf\ i}$ = volume d'information transitant sur l'arc i de l'arbre algébrique associé à Q, on a :

$$V_{inf\ Q} = \sum_i V_{inf\ i} = \sum \text{volumes relations primaires (From)} + \sum \text{volumes relations temporaires}$$

on note : IRI = volume de la relation R = (nb tuples R X nb car/tuple R)

Par exemple :

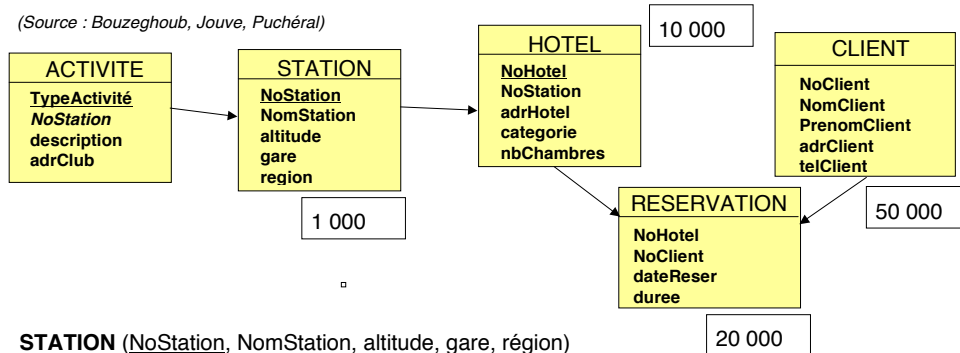


$$V_{inf\ Q} = |R1| + |R1|_{[q1]} + |R2| + |R1|_{[q1]} \bowtie |R2| + |\Pi_{q3}(R_{[q1]} \bowtie R2)|$$

1 2 3 4 5

Cas Station : la base

(Source : Bouzeghoub, Jouve, Puchéral)

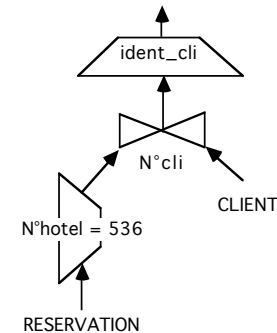


- STATION** (NoStation, NomStation, altitude, gare, région)
ACTIVITE (typeActivité, NoStation*, description, adrClub)
HOTEL (NoHôtel, NoStation*, adrHotel, catégorie, nbChambres)
RESERVATION (NoHôtel*, NoClient*, dateReser, durée)
CLIENT (NoClient, nomClient, prenomClient, adrClient, telClient)
 (attribut = clé primaire ; attribut* = clé étrangère)

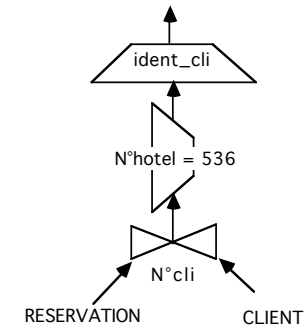
Cas Station : optimisation

Requête : Identité des clients ayant passé réservation à l'hôtel N° 536?

Select NomClient From client, reservation
 Where Nohôtel = "536" And
 reservation.NoClient = client.NoClient



Select NomClient From client, reservation
 Where reservation.NoClient = client.NoClient
 And NoHôtel="536"



Cas station : estimation des volumes

Identité des clients ayant passé réservation à l'hôtel N° 536 ?

Requête 1 :

$$V_{inf} = /reservation/ + /reservation[NoHôtel=536]/ + /client/$$

$$+ /(reservation[NoHôtel=536]Xclient/ \ /J/$$

$$+ /Π ident_cli (reservation[NoHôtel=536]Xclient)/ \ /Π/$$

soit :

$/reservation/ = 20\ 000 \times 17\ c = 340\ 000 = 3,4\ 10^5\ c$

$/reservation[NoHôtel=536]/ = (20\ 000/10\ 000) \times 17\ c = 34\ c$

(hypothèse : répartition uniforme des réservations entre les hôtels)

$/client/ = 25\ 000 \times 133 = 3\ 325\ 000 = 3,325\ 10^6\ c$

calcul de $/J/$:

reservation > client soit : $/J/ = 2(17+133\ c) = 300\ c$

(un seul tuple de reservation avec un seul tuple de client)

calcul de $/Π/$:

conserve que l(attribut NomClient (40 c) soit : $/Π/ = 2 \times 40 = 80\ c$

d'où : $V_{inf} = 3,4\ 10^5 + 34 + 3,325\ 10^6 + 300 + 80 = 3,665414\ 10^6\ \text{caractères}$

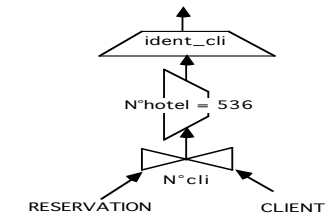
Requête 2 :

$$V_{inf} = /reservation/ + /client/ + /reservation[N°hôtel=536]/ + /Π ident_cli (reservationXclient)/$$

$$V_{inf} = 3,4\ 10^5 + 3,325\ 10^6 + (20\ 000(17+133)) + 300 + 80 = 3,9650380\ 10^6\ \text{caractères}$$

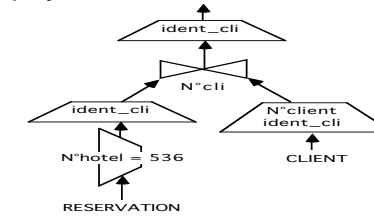
Cas station : restructuration algébrique (1)

Requête 2 :



- commutativité des restrictions/jointures
- commutativité des projections/jointures (remontée des jointures)

=> redescente des projections :



Cas station : restructuration algébrique (2)

- projection rajoutée sur CLIENT pas bénéfique car si elle réduit le volume d'information :
 - elle induit un traitement supplémentaire sur CLIENT
 - si un chemin d'accès existe sur CLIENT (index), il est perdu pour la jointure...

d'où restructuration retenue :

