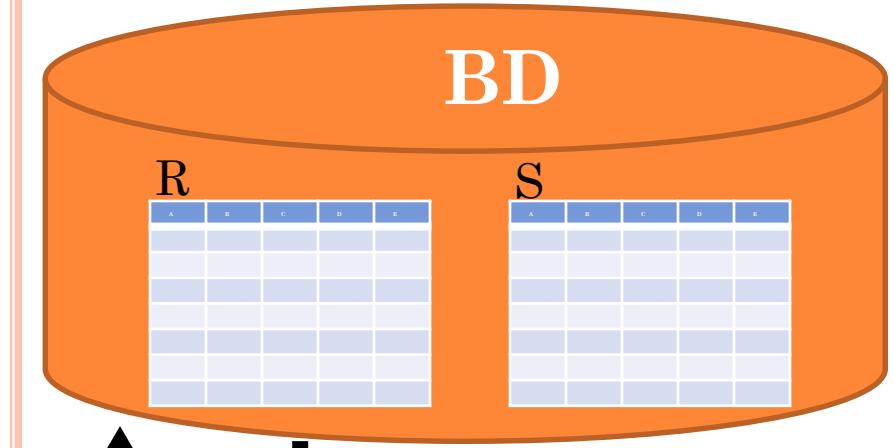
BDW1 : Bases de données et Programmation Web

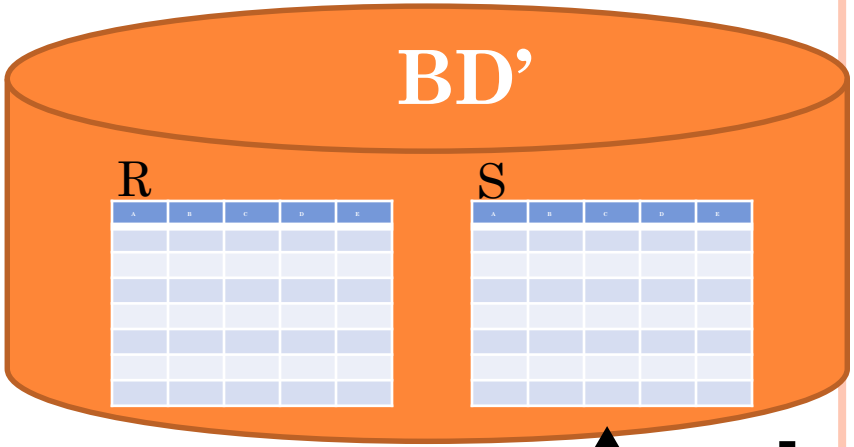
Algébre Relationnelle & Optimisation

Nicolas Lumineau

nicolas.lumineau@univ-lyon1.fr

Traitement d'une requête





Q

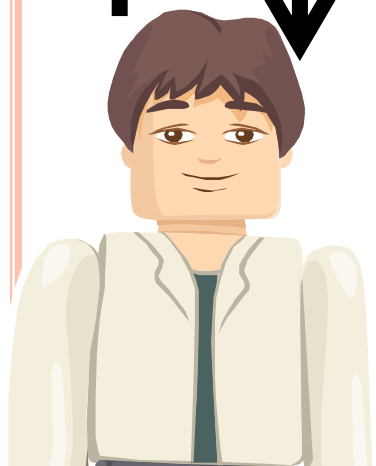
tuples

Requête Q :

```
SELECT R.A, S.C
FROM R,S
WHERE R.B=S.B;
```

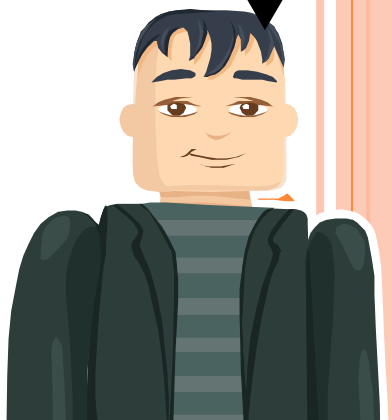
Q

tuples



Exécutée
en 5 ms

Exécutée
en 105 ms

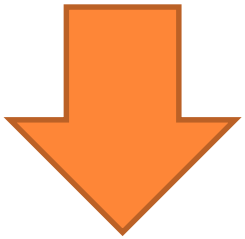


Une même requête peut s'exécuter de différentes manières

Questions

- Comment passe-t-on d'une requête SQL (définie dans un langage déclaratif) à un programme (impératif) manipulant les données ?

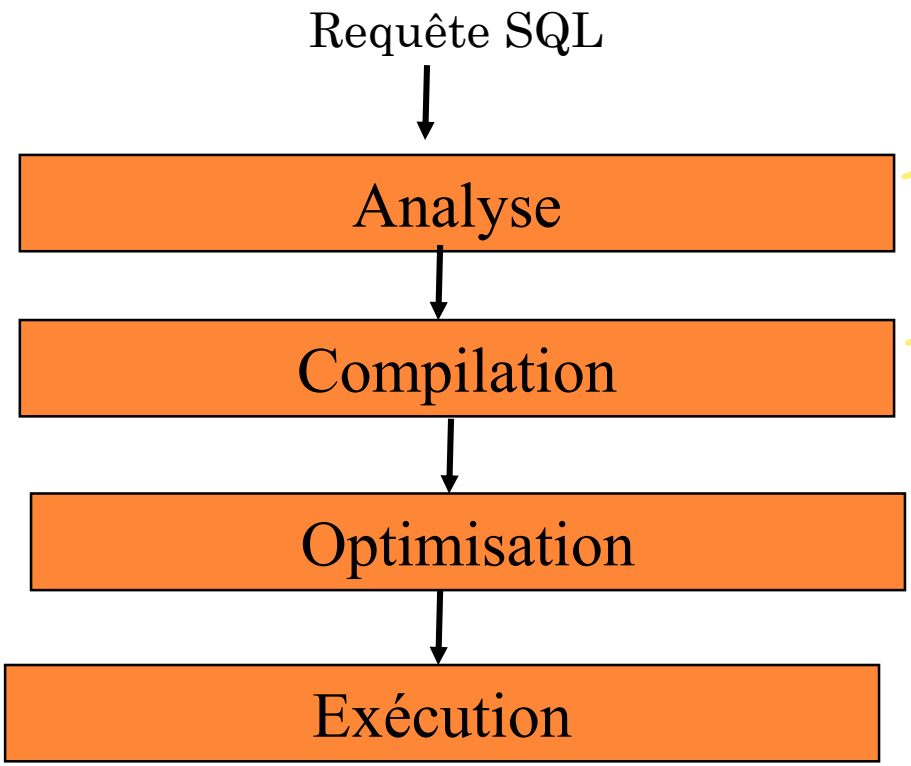
- Comment optimise-t-on une requête afin de l' exécuter efficacement pour trouver un résultat correct?



Analyse des différentes étapes du traitement d'une requête

BDW1

Traitement de requêtes



Validation de la syntaxe de la requête

Traduction de la requête en **arbre algébrique**

Recherche de **l'arbre algébrique optimal**

Exécution du programme correspondant à l'arbre algébrique optimal

Arbre algébrique ≈ Plan d'exécution

Langages de requêtes relationnels

- Langages « théoriques »
  - Objectif : étudier le modèle relationnel, prouver des propriétés et obtenir une mise en œuvre efficace
    - Langage procédural (approche algébrique)
      - **Algèbre Relationnelle**
    - Langage déclaratif (approche par prédicats)
      - **Calcul Relationnel**
        - à variable tuple (CRVT)
        - à variable domaine (CRVD)
- Langages « commerciaux » (approche plus conviviale pour l'utilisateur)
  - **SQL** (basé sur le CRVT)
  - QBE (basé sur le CRVD)

ALGEBRE RELATIONNELLE

BDW1

Algèbre relationnelle

- C'est un langage procédural proposé par E. Codd, 1969
- Définition
  - Une algèbre relationnelle est un ensemble d'opérations agissant sur des relations et produisant des relations
- Principe :
  - combiner des relations à travers des opérateurs pour formuler une requête
- **Langage fermé** : les opérandes et les résultats sont *toujours* des relations

BDW1

Les opérateurs

- Opérateurs de base :
  - **Cinq opérateurs de base** : sélection, projection, union, différence et produit.
  - **Un opérateur syntaxique**: le renommage, qui ne fait que modifier le schéma et pas les n-uplets.
- Opérateurs déduits :
  - Raccourcis syntaxiques pratiques:
    - intersection, Θ-jointure, semi-jointure et division.
- On peut les regrouper en deux catégories :
  - **opérateurs ensemblistes** : union, intersection, différence, produit cartésien
  - **opérateurs spécifiques BD** : sélection, projection, jointures, division, renommage

BDW1

Union

$R \cup S$

Union *ensembliste* entre deux tables :

$$R \cup S = \{t \mid t \in R \text{ ou } t \in S\}$$

où R et S sont des relations, t est une variable n-uplet

- Le résultat contient les n-uplets qui sont dans R ou dans S (élimination des doublons)
- *Contrainte : Les schémas de R et S* doivent être compatibles pour l'union (même arité et domaines d'attributs)

BDW1

Exemple d'Union

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Réalisateur

Nom	Prénom	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

Artiste = Acteur ∪ Réalisateur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

Intersection $R \cap S$

Intersection de deux tables:

$$R \cap S = \{t \mid t \in R \text{ et } t \in S\}$$

où

R et S sont deux relations compatibles pour l'union, , t est une variable n-uplet .

- Le résultat contient les n-uplets qui sont à la fois dans R et dans S

BDW1

Exemple d'Intersection

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Réalisateur

Nom	Prénom	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

ActeurRéalisateur = Acteur ∩ Réalisateur

Nom	Prénom	Nationalité
Allen	Woody	USA

BDW1

Différence

$R - S$ ou $R \setminus S$

- Différence *ensembliste* entre deux tables :

$$R - S = \{t \mid t \in R \text{ et } t \notin S\}$$

où

  - R et S sont des relations, t est une variable n-uplet

- Le résultat contient tous les n-uplets qui sont dans R, mais pas dans S.

- *Attention :* $R - S \neq S - R$

- *Question : Que représente* $R - (R - S)$ *?*

BDW1

Exemple de Différence

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Réalisateur

Nom	Prénom	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

ActeurNonRéal = Acteur - Réalisateur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK

BDW1

Produit cartésien $R \times S$

- Produit cartésien entre deux tables :

$$R \times S = \{t.A_1 \ldots A_{k_1} A_{k_1+1} \ldots A_{k_1+k_2} \mid t.A_1 \ldots A_{k_1} \in R \text{ et } t.A_{k_1+1} \ldots A_{k_1+k_2} \in S\}$$

où

- R est une table de degré k_1, cardinalité n_1
- S est une table de degré k_2, cardinalité n_2

- Le résultat est une relation de *degré* $(k_1 + k_2)$ et contient $(n_1 * n_2)$ n-uplets, où chaque n-uplet est la concaténation d'un nuplet de R avec un n-uplet de S.

BDW1

Exemple de Produit Cartésien

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Pays

Code	NomPays	Nat.
ESP	Espagne	Espagnol
FR	France	Français
USA	Etats-Unis	Américain
UK	Royaume-Uni	Anglais

Acteur × Pays

Nom	Prénom	Nationalité	Code	NomPays	Nat.
Cruz	Penelope	ESP	ESP	Espagne	Espagnol
Cruz	Penelope	ESP	FR	France	Français
Cruz	Penelope	ESP	USA	Etats-Unis	Américain
Cruz	Penelope	ESP	UK	Royaume-Uni	Anglais
Maura	Carmen	ESP	ESP	Espagne	Espagnol
….	…	…	…	…	….
Downey	Robert	UK	ESP	Espagne	Espagnol
Downey	Robert	UK	FR	France	Français
Downey	Robert	UK	USA	Etats-Unis	Américain
Downey	Robert	UK	UK	Royaume-Uni	Anglais

Renommage (Rho)

Renommage d'un sous-ensemble de la relation :

$$\rho_{A1/B1,\ldots An/Bn}(R)=\{t \mid t \in R \text{ et } A1 \text{ renommé en } B1, \ldots An \text{ renommé en } Bn\}$$

où :

- $R(A1, \ldots An)$ est une relation,
- Utile en cas de problème d'homonymie ou avant certaines opérations ensemblistes.

Exemple de Renommage

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Réalisateur

Nom	Prénom	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

ActeurBis =

$\rho_{Nom/NomA\ ,\ Prénom/PrénomA,\ Nationalité/Pays}$ (Acteur)

Nom A	PrénomA	Pays
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

RealBis=

$\rho_{Nom/NomR\ ,\ Prénom/PrénomR,}$ (Réalisateur)

NomR	PrénomR	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

Sélection (Sigma)

- Sélection d'un sous-ensemble de la relation opérande :

$$\sigma_F(R)=\{t \mid t \in R \text{ et } F(t) \text{ est vrai}\}$$

où

- R est une relation, t est un n-uplet variable
- F est une *formule logique sans quantificateur* composée de :
  - opérandes: constantes et attributs
  - opérateurs de comparaison :<, >, =, ≠, ≤, ≥
  - opérateurs logiques : ∧, ∨, ¬

BDW1

Exemple de Sélection

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Réalisateur

Nom	Prénom	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

$ActeurFrancais = \sigma_{Nationalité='FR'}(Acteur)$

Nom	Prénom	Nationalité
Villeret	Jacques	FR
Lhermite	Thierry	FR

$RealEspagnol = \sigma_{Pays='ESP'}(Réalisateur)$

Nom	Prénom	Pays
Almodovar	Pedro	ESP

BDW1

Projection (Pi)

- Projection sur un ensemble d'attributs d'une relation

$$\Pi_{A_1,\ldots,A_n}(R)=\{t.A_1 \ldots A_n \mid t \in R\}$$

où

- R est une relation, t est une variable n-uplet
- $\{A_1,\ldots, A_n\}$ est un sous-ensemble des attributs de R

- Remarque :
  - la projection *élimine les n-uplets doublons*.
  - Les SGBD relationnels (et SQL) permettent
    - projection (pure)
    - projection sans élimination des doublons

BDW1

Exemple de Projection

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Réalisateur

Nom	Prénom	Pays
Almodovar	Pedro	ESP
Allen	Woody	USA
Veber	Francis	FR
Favreau	John	USA

NomActeur = $\pi_{\text{Nom , Prenom}}$ (Acteur)

Nom	Prénom
Cruz	Penelope
Maura	Carmen
Johansson	Scarlett
Villeret	Jacques
Lhermite	Thierry
Downey	Robert
Allen	Woody

PaysReal= π_{Pays} (Réalisateur)

Pays
ESP
USA
FR

BDW1

JOINTURE (θ-JOINTURE)

- θ-Jointure entre les attributs de deux relations

$$R \bowtie_F S = \{t.A_1 \ldots A_n B_1 \ldots B_m | \; t.A_1 \ldots A_n \in R \text{ et } t.B_1 \ldots B_m \in S \text{ et } F \text{ est vrai}\}$$

où

- *R et S* sont des relations
- *t* est une variable n-uplet
- *F* est une formule logique composée d'atomes de la forme $A_i \; \theta \; B_j$ où $\theta \in \{<, >, =, \neq, \leq, \geq\}$.

- La jointure s'exprime aussi par un produit cartésien suivi d'une sélection: $R \bowtie_F S = \sigma_F(R \times S)$

BDW1

Exemple de Jointure

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Pays

Code	NomPays	Nat.
ESP	Espagne	Espagnol
FR	France	Français
USA	Etats-Unis	Américain
UK	Royaume-Uni	Anglais

$$\text{Acteur} \bowtie_{\text{Nationalité=Code}} \text{Pays}$$

Nom	Prénom	Nationalité	NomPays	Nat.
Cruz	Penelope	ESP	Espagne	Espagnol
Maura	Carmen	ESP	Espagne	Espagnol
Johansson	Scarlett	USA	Etats-Unis	Américain
Villeret	Jacques	FR	France	Français
Lhermite	Thierry	FR	France	Français
Downey	Robert	UK	Royaume-Uni	Anglais
Allen	Woody	USA	Etats-Unis	Américain

BDW1

Types de jointure

- Equi-jointure
  - la formule F n'utilise que l'opérateur d'égalité (=)
  - $R \bowtie_{R.A=S.B} S$
- Jointure naturelle : R(X,Y), S(X, Y')
  - Equi-jointure où on élimine les attributs en commun
  - $R \bowtie S = \Pi_{R.X,\ R.Y,\ S.Y'}(\sigma_F(R \times S)) = \Pi_{S.X,\ R.Y,\ S.Y'}(\sigma_F(R \times S))$
  - la condition de jointure F est R.X = S.X (X représente tous les attributs en commun entre R et S)

BDW1

SEMI-JOINTURE

- Semi-Jointure entre les attributs de deux relations

$$R \ltimes_F S = \{t.A_1 \ldots A_n \mid t.A_1 \ldots A_n \in R \text{ et } F \text{ est vrai pour } B_1, \ldots B_n \in S\}$$

où

- R *et* S sont des relations
- t est une variable n-uplet
- F est une formule logique composée d'atomes de la forme $A_i \ \theta \ B_j$ où $\theta \in \{<, >, =, \neq, \leq, \geq\}$.

- La semi-jointure s'exprime aussi par une jointure suivit d'une projection sur les attributs de R:

$$R \ltimes_F S = \pi_R (R \bowtie_F S)$$

BDW1

Exemple de Semi-Jointure

Acteur

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Downey	Robert	UK
Allen	Woody	USA

Pays

Code	NomPays	Nat.
ESP	Espagne	Espagnol
FR	France	Français
USA	Etats-Unis	Américain

$$\text{Acteur} \ltimes_{\text{Nationalité=Code}} \text{Pays}$$

Nom	Prénom	Nationalité
Cruz	Penelope	ESP
Maura	Carmen	ESP
Johansson	Scarlett	USA
Villeret	Jacques	FR
Lhermite	Thierry	FR
Allen	Woody	USA

Division

- Soient les relations R et S tels que *le schéma de S contient tous les attributs de R*, alors on peut réordonner les attributs de R tel que :
  - $R(A_1,\ldots,A_k,A_{k+1},\ldots,A_{k+n})$ est de degré k+n et
  - $S(A_1,\ldots,A_k)$ de degré *k*.

- La division de R par S :

$$T(A_{k+1},\ldots,A_{k+n}) = R \div S$$

est une relation de degré n qui contient tous les n-uplets t tels que $\{t\} \times S \subseteq R$.

BDW1

Exemple de Division

Acteur

Nom	Prénom	Titre
Cruz	Penelope	Vicky Cristina Barcelona,
Cruz	Penelope	Fanfan la Tulipe
Johansson	Scarlett	Vicky Cristina Barcelona
Cruz	Penelope	Volver
Hayek	Salma	Bandidas
Cruz	Penelope	Bandidas

Film

Titre
Vicky Cristina Barcelona
Volver
Bandidas
Fanfan la Tulipe

Acteur ÷ Film

Nom	Prénom
Cruz	Penelope

BDW1

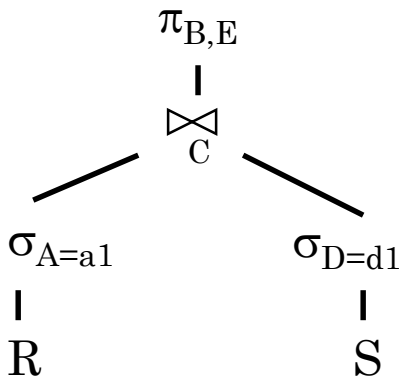
Représentation algébrique

Soit R(A, B ,C) et S(C, D, E)

- Différents modes de représentation de requêtes algébriques :
  - Expression algébrique

$$\pi_{B,E}\,((\,\sigma_{A=a1}\ (R))\bowtie_{C}(\ \sigma_{D=d1}\,(S)\,))$$

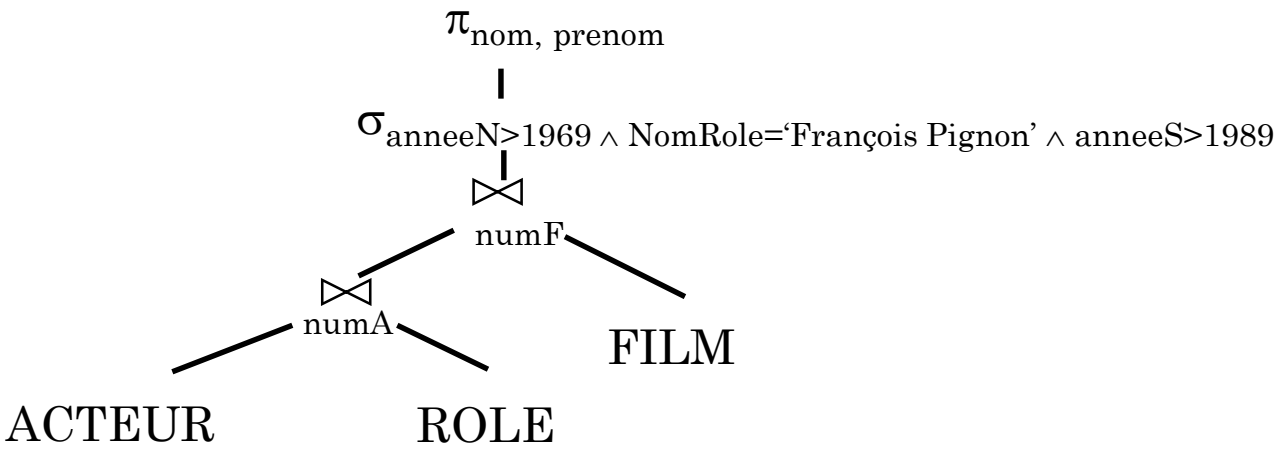
  - Arbre algébrique



EXERCICE

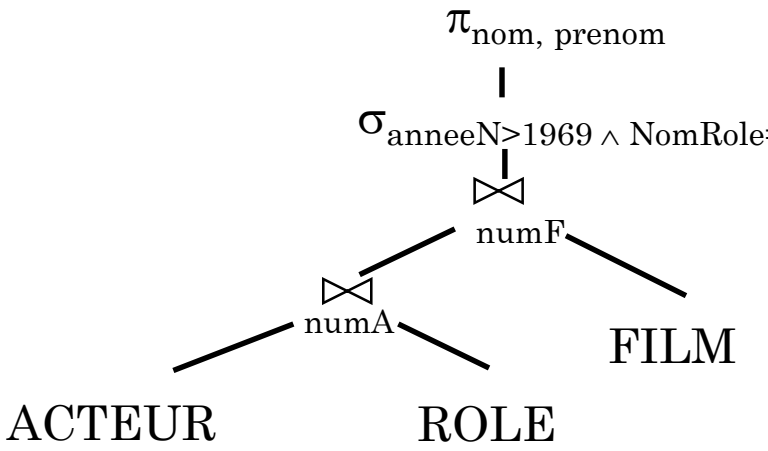
- Soit les relations
  - ACTEUR(numA, nom, prenom, anneeN)
  - FILM(numF, titre, duree, anneeS)
  - ROLE (numF, numA, nomRole)

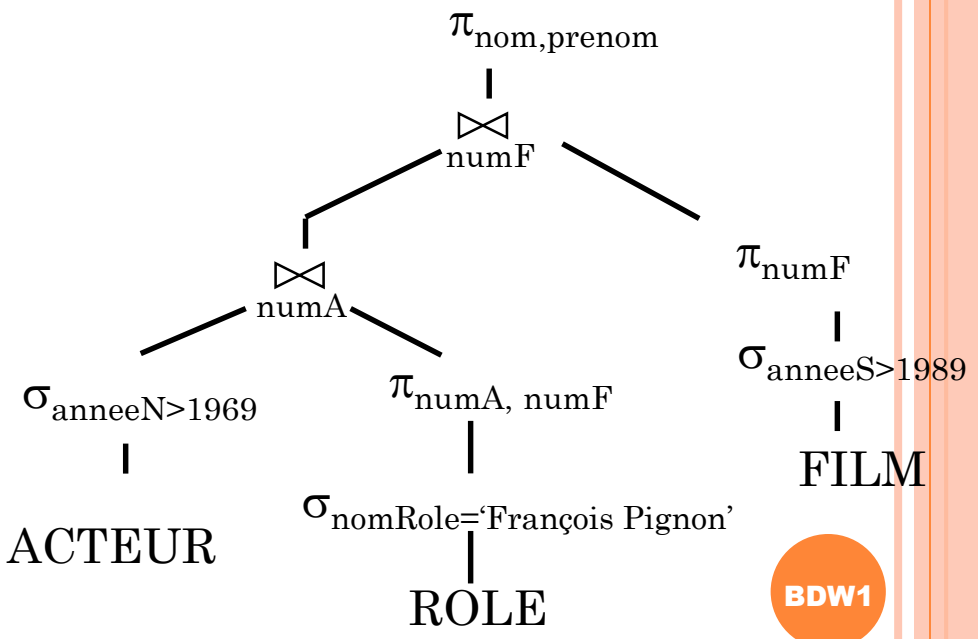
- Donner le nom et le prénom des acteurs nés après 1970 (inclus) qui ont joué le rôle de François Pignon dans des films sortis après 1990.



Remarque

- Question :
  - Quelle différence entre les deux requêtes suivantes ?

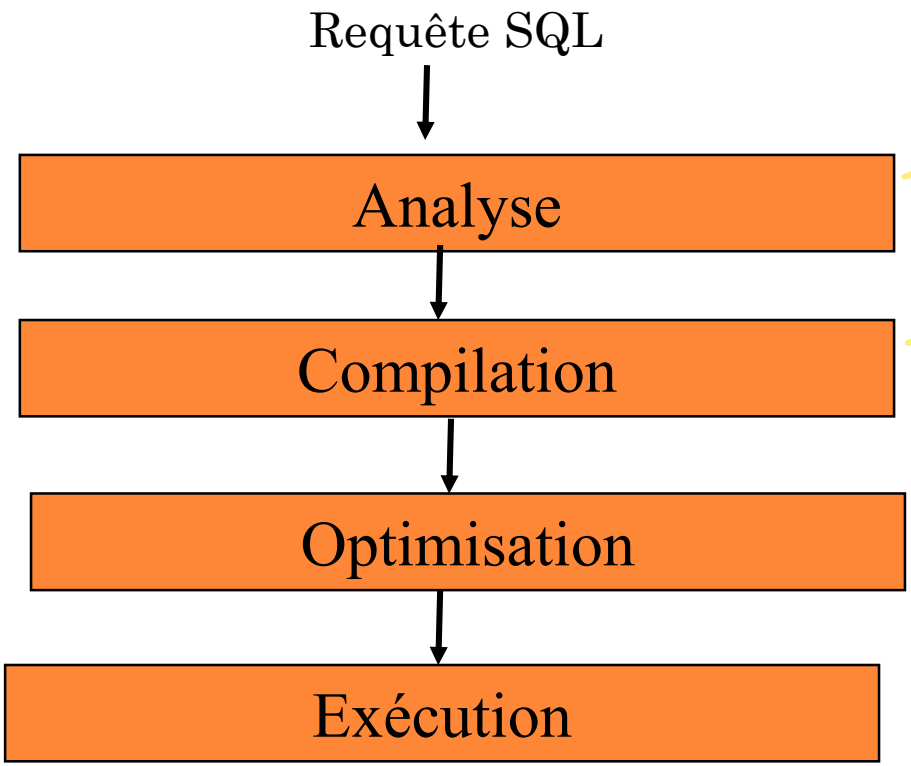




AUCUNE en terme de résultat retourné, mais les temps de traitements seront différents

Retour à la notion d'optimisation

Traitement de requêtes



Validation de la syntaxe de la requête

Traduction de la requête en **arbre algébrique**

Recherche de **l'arbre algébrique optimal**

Exécution du programme correspondant à l'arbre algébrique optimal

Arbre algébrique ≈ Plan d'exécution

TRAITEMENT D'UNE REQUÊTE

R(A,B,C) et S(B,C,D, E)

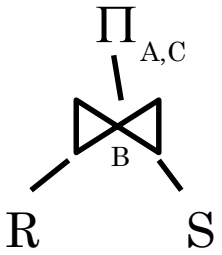
SELECT R.A, S.C
FROM R, S
WHERE R.B=S.B;

Validation OK

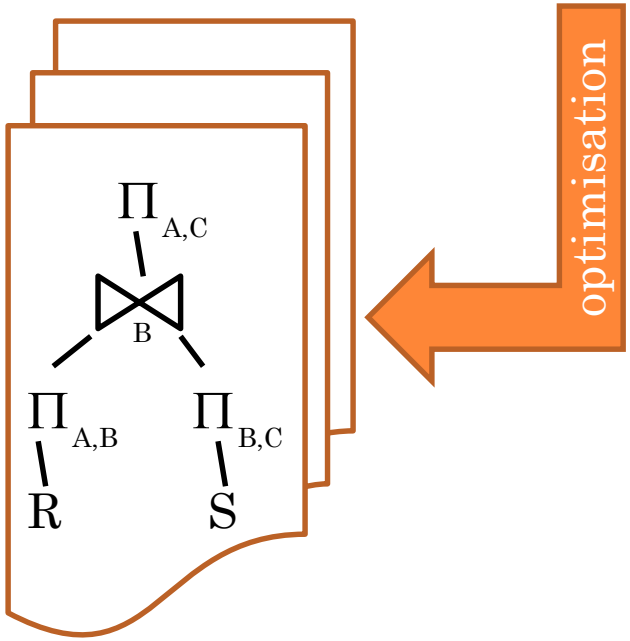
Analyse

SELECT R.A, S.C
FROM R, S
WHERE R.B=S.B;

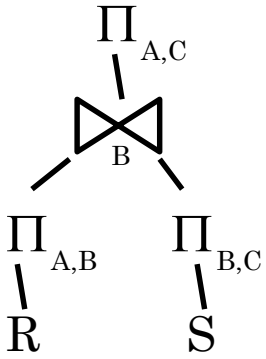
compilation



optimisation



Election du plan optimal



Choix des algorithmes

Prog.

exécution

Résultats

Analyse

- **Analyse lexicale et syntaxique** :
  - Validation par rapport à la syntaxe SQL
  - Vérification des types :
    - présence des attributs et relations dans le schéma
    - compatibilité des types dans les prédicats
  - Décomposition en requêtes/sous-requêtes

Compilation

- Règle de passage d'une requête SQL en AR
  - SELECT …. : permet de définir une projection
  - FROM … : permet de définir les feuilles de l'arbre
  - WHERE ….: permet de définir
    - Pour les comparaison attribut/valeur : des sélections
    - Pour les comparaison attribut/attribut : des jointures

Optimisation

- Optimiser c'est trouver le plan d'exécution d'une requête dont le coût soit minimal (i.e. le temps de réponse à la requête soit le plus rapide possible)

- Qu'est-ce qui peut être optimiser dynamiquement?
  - Les accès disques

Il est indispensable que le temps lié à l'optimisation soit négligeable par rapport au temps imparti à l'exécution

Espace de recherche

- Caractérisé par les plans “équivalents” pour une même requête
  - ceux qui donnent le même résultat
  - générés en appliquant des règles de transformation
- Le coût de chaque plan est en général différent
- L'ordre des jointures est important

Optimisation algébrique

Choix algorithmiques

- Chaque opérateur peut être implémenté de différentes façons.
  - La complexité de l'algorithme a un impact sur le temps de traitement de la requêtes
- La façon dont les jointures sont implémentées est important

Optimisation physique

BDW1

Exécution

- Application pour chaque opération du plan d ’exécution des programmes associés
  - MySQL, Postgres : C, C++
  - HSQL : JAVA
  - …

Illustration des plans d'exécution

Employe (Eid, Enom, Titre, Ville)
Projet(Pid, Pnom, Budget, Ville)
Travaux(Eid, Pid, Respid, Durée)

Requête :
« le nom et le titre des employés qui travaillent dans des projets avec un budget > 250 »

```
SELECT  DISTINCT Enom, Titre
FROM    Employe E, Projet P, Travaux T
WHERE   P.Budget > 250
AND     E.Eid=T.Eid
AND     P.Pid=T.Pid ;
```

BDW1

Un plan d'exécution possible

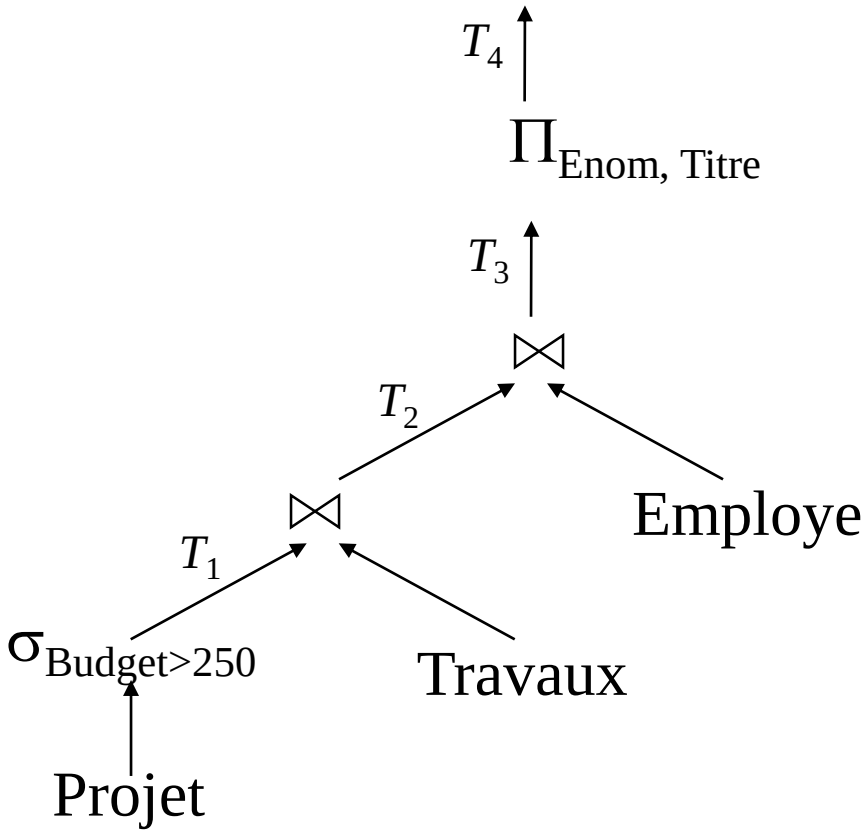
$T_1 \leftarrow$ *Lire* la table `Projet` et *sélectionner* les tuples de `Budget` > 250

$T_2 \leftarrow$ *Joindre* T_1 avec la relation `Travaux`

$T_3 \leftarrow$ *Joindre* T_2 avec la relation `Employe`

$T_4 \leftarrow$ *Projeter* T_3 sur `Enom`, `Titre`

Représentation graphique



Un autre plan d'exécution possible

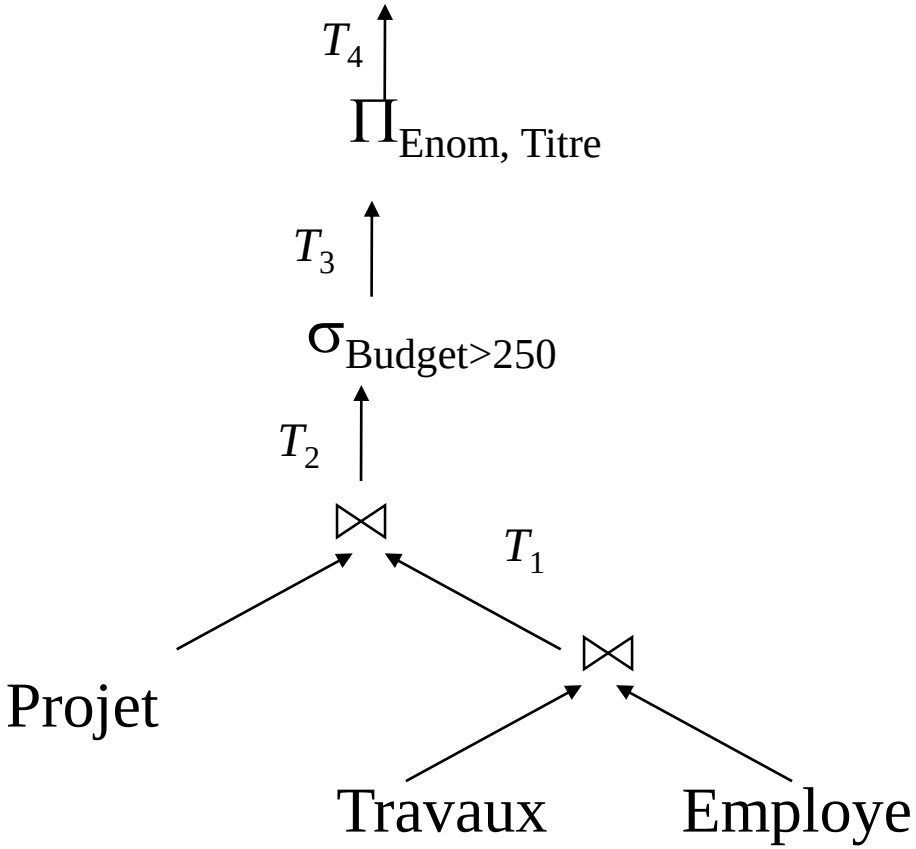
$T_1 \leftarrow$ Joindre la relation `Travaux` et la relation `Employe`

$T_2 \leftarrow$ *Joindre* T_1 avec la relation `Projet`

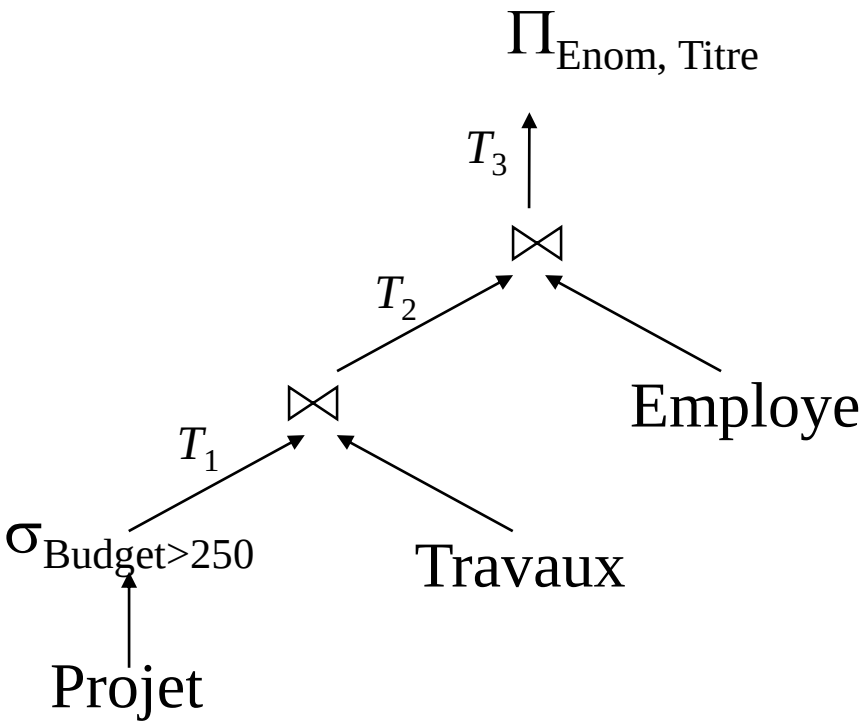
$T_3 \leftarrow$ *Lire* T_2 et *sélectionner* les tuples de `Budget` > 250

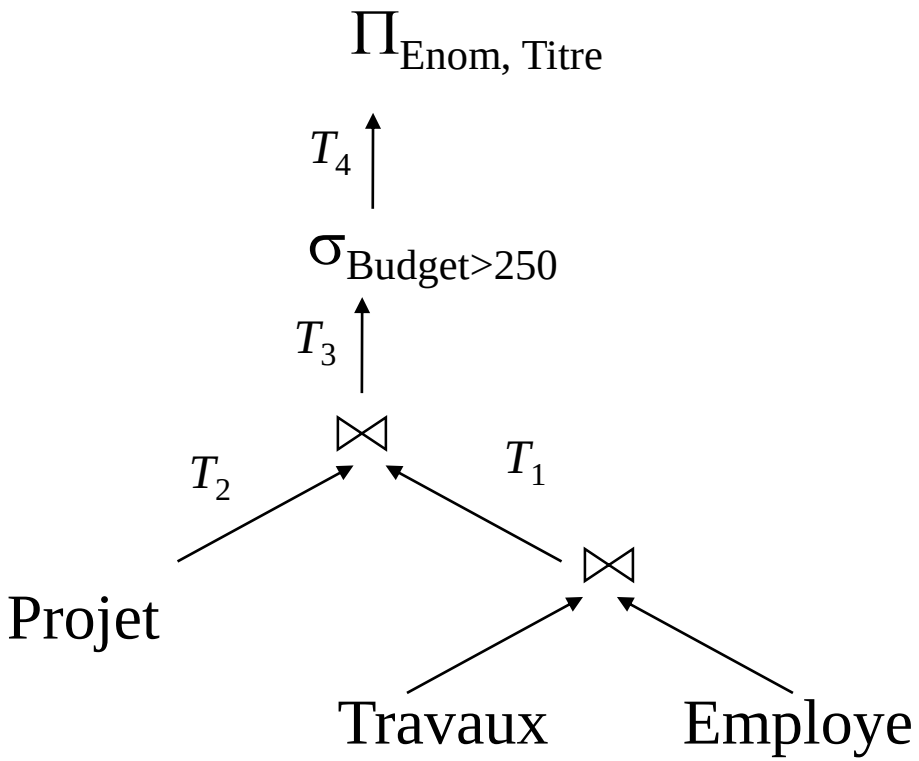
$T_4 \leftarrow$ *Projeter* T_3 sur `Enom`, `Titre`

Représentation graphique



Quelle différence entre les deux plans d'exécution ?





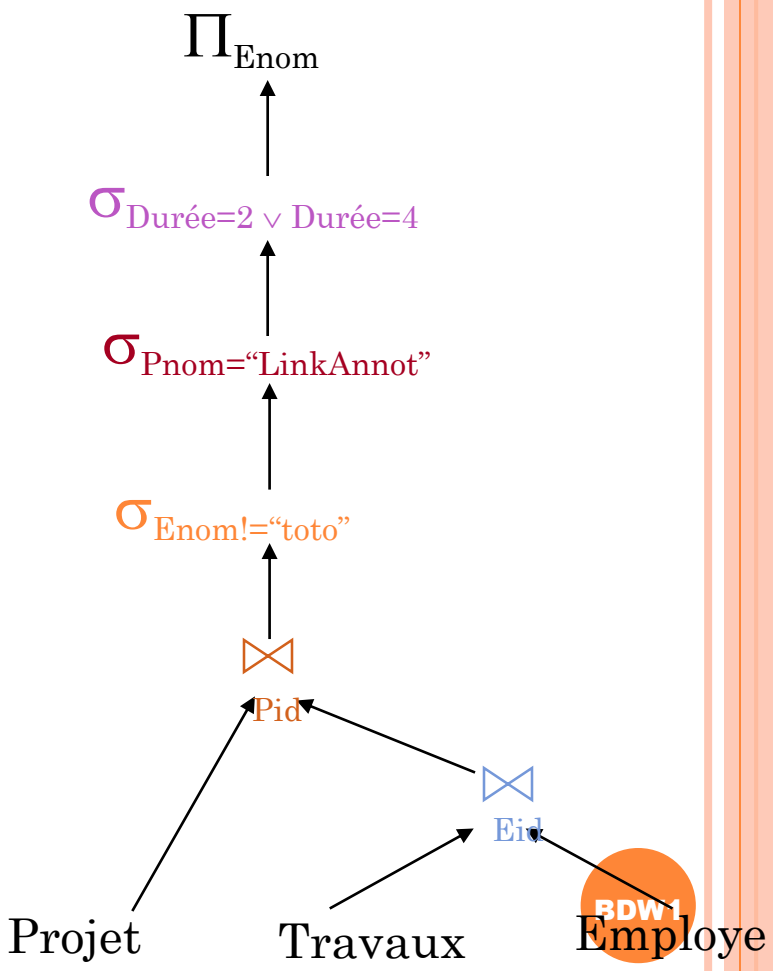
Comment optimiser?

- Réécriture des compositions de projections et/ou sélections
- Réduire au plus tôt la taille et le nombre de tuples manipulés
  - Tuples moins nombeux : grâce à la sélection
  - Tuples plus petit : grâce à la projection
- Réordonner les jointures, en fonction de la taille des relations manipulées
- Choisir des algorithmes de jointures efficaces selon les relations manipulées

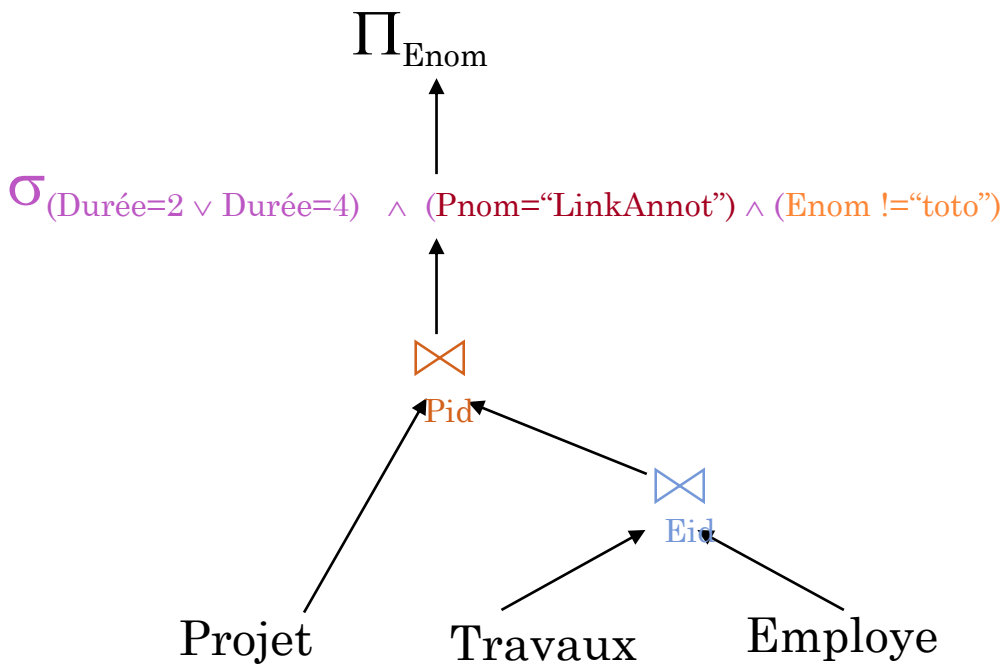
Réécriture des compositions de projections et/ou de sélections

- Exemple :

```
SELECT  Enom
FROM    Employe E, Travaux T,
Projet P
WHERE   E.Eid = T.Eid
AND     T.Pid = P.Pid
AND     Enom !='Toto'
AND     Pnom = 'Linkannot'
AND     (Durée=2 OR Durée=4)
```



Réécriture des compositions de projections et/ou de sélections



Optimisation algébrique

- Principe:

Réduction au plus tôt du nombre et de la taille des tuples manipulés

- Manipulation algébrique
  - Exploitation des règles de transformation de certains opérateurs
    - commutativité
    - associativité

Règles de transformation

- Commutativité des opérations binaires
  - $R \times S \equiv S \times R$
  - $R \bowtie S \equiv S \bowtie R$
  - $R \cup S \equiv S \cup R$
- Associativité des opérations binaires
  - $(R \times S) \times T \equiv R \times (S \times T)$
  - $(R \bowtie S) \bowtie T \equiv R \bowtie (S \bowtie T)$
- Idempotence des opérations unaires
  - $\Pi_{A1\ldots An}(\Pi_{B1\cdots Bm}(R)) \equiv \Pi_{A1\ldots An}(R)$, Bi inclus dans Ai
  - $\sigma_{F1}(\sigma_{F2}(R)) \equiv \sigma_{F1 \wedge F2}(R)$

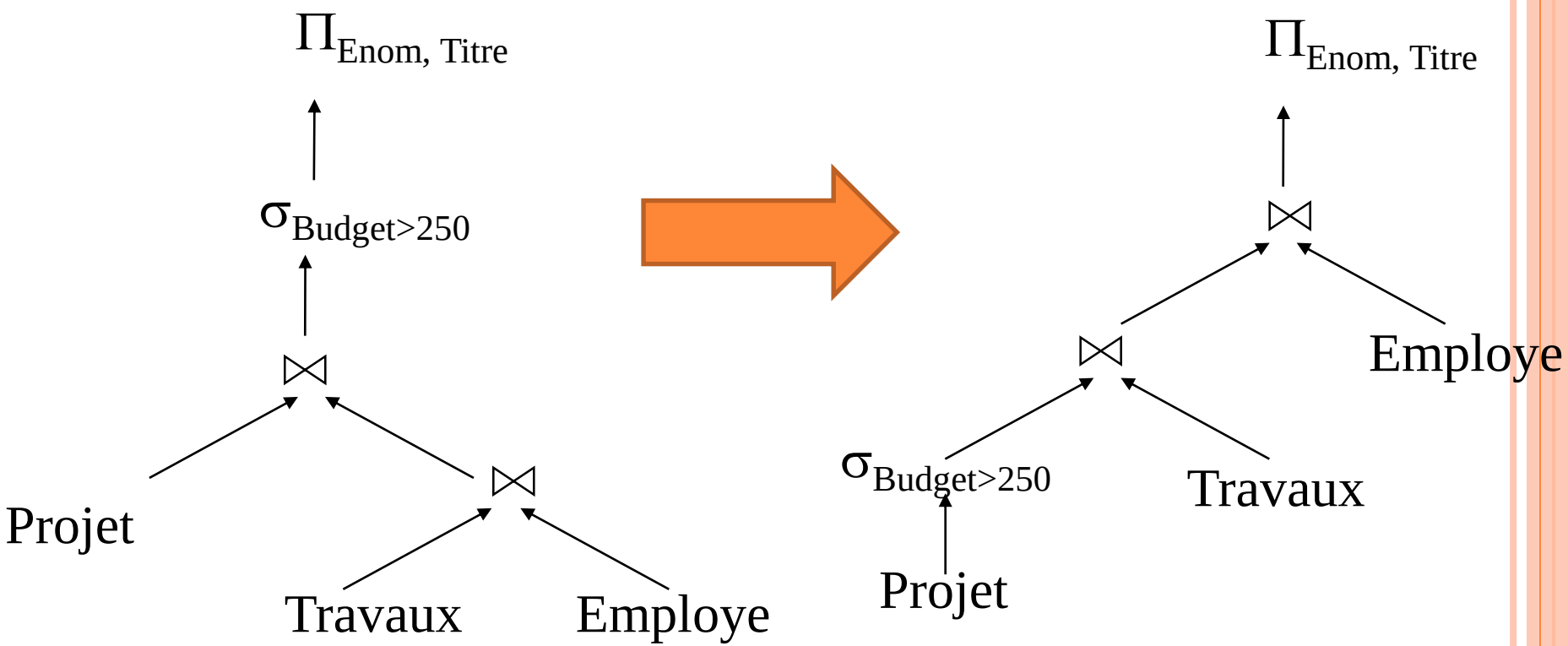
Règles de transformation

- Commutation sélection et projection
  - Si *F ne porte que sur A1,…, An,*
    - $\Pi_{A1,\ldots,An}(\sigma_F(R)) = \sigma_F(\Pi_{A1,\ldots,An}(R))$
  - Si *F porte aussi sur B1,…,Bm,*
    - $\Pi_{A1,\ldots,An}(\sigma_F(R)) = \Pi_{A1,\ldots,An}(\sigma_F(\Pi_{A1,\ldots,An,B1,\ldots,Bm}(R)))$

Règles de transformation

- Commutativité de la sélection avec les opérations binaires
  - $\sigma_F(R \bowtie S) = \sigma_F(R) \bowtie \sigma_F(S)$
  - $\sigma_F(R \times S) = \sigma_F(R) \times \sigma_F(S)$
  - $\sigma_F(R \cup S) = \sigma_F(R) \cup \sigma_F(S)$
  - $\sigma_F(R - S) = \sigma_F(R) - \sigma_F(S)$
- Commutativité de la projection avec les opérations binaires
  - $\Pi_A(R \times S) \equiv \Pi_A(R) \times \Pi_A(S)$
  - $\Pi_A(R \cup S) \equiv \Pi_A(R) \cup \Pi_A(S)$

COMMENT PASSER DE L'UN À L'AUTRE?

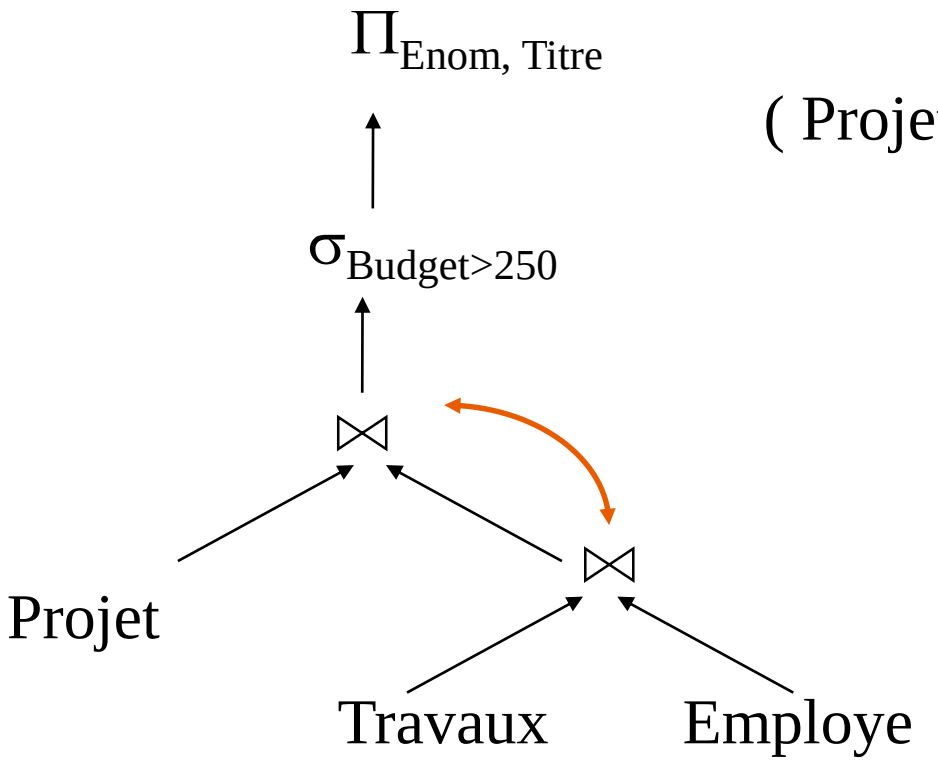


Comment passer de l'un à l'autre?

Associativité de la jointure

Projet ⋈ (Travaux ⋈ Employe)

(Projet ⋈ Travaux) ⋈ Employe

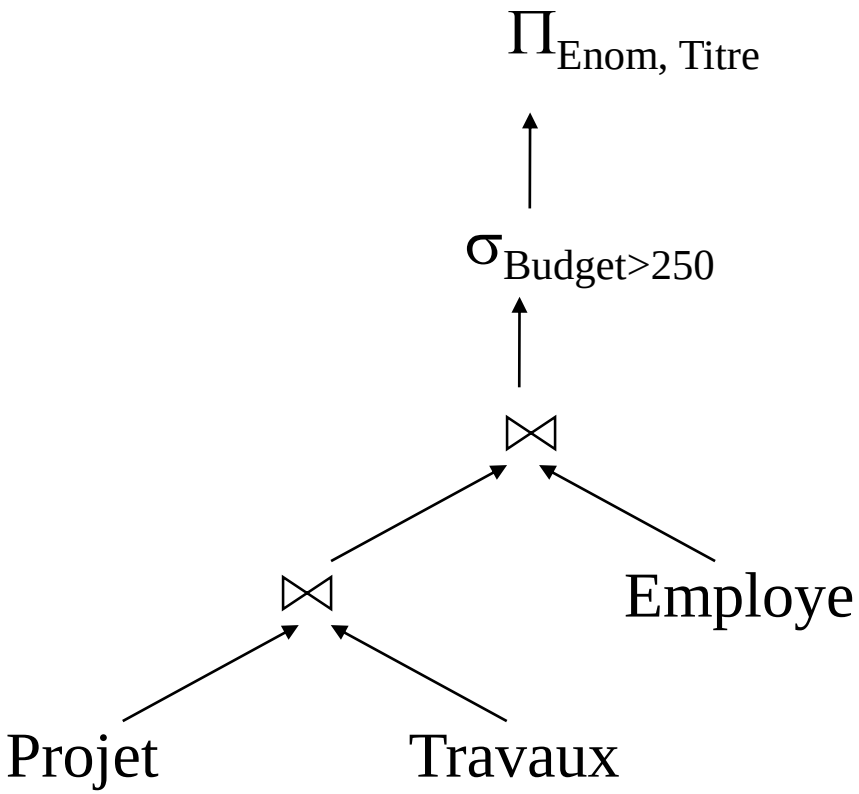


Comment passer de l'un à l'autre?

Associativité de la jointure

Projet ⋈ (Travaux ⋈ Employe)

(Projet ⋈ Travaux) ⋈ Employe

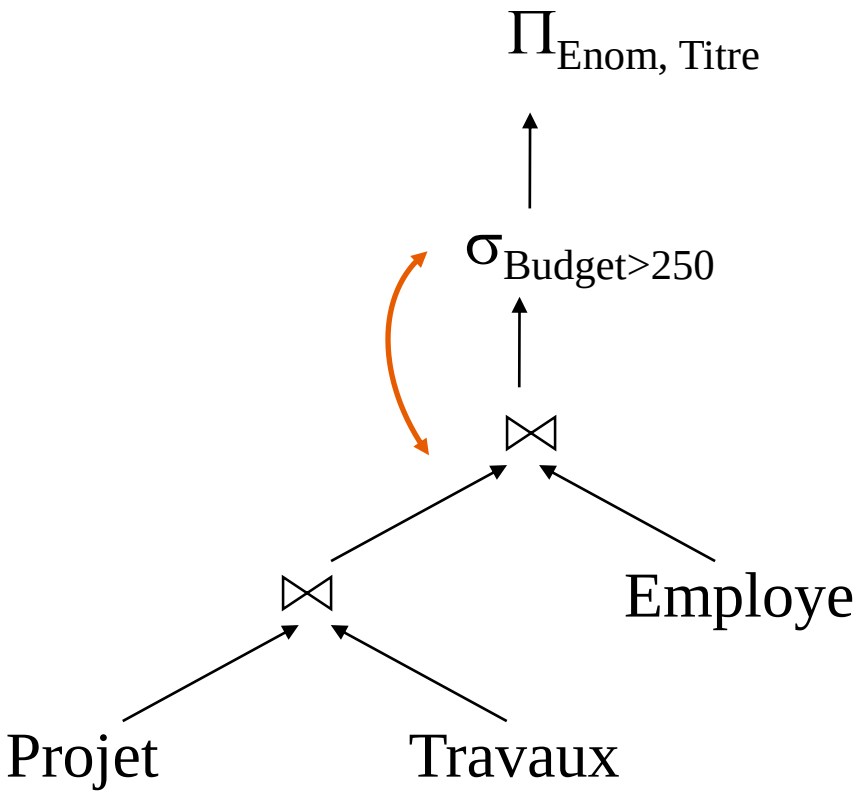


Comment passer de l'un à l'autre?

Commutativité de la sélection et de la jointure

$$\sigma_{Budget>250}\ (\ E \bowtie (\ P \bowtie T)\ )$$

$$(\ E \bowtie \sigma_{Budget>250}(\ P \bowtie T)\ )$$

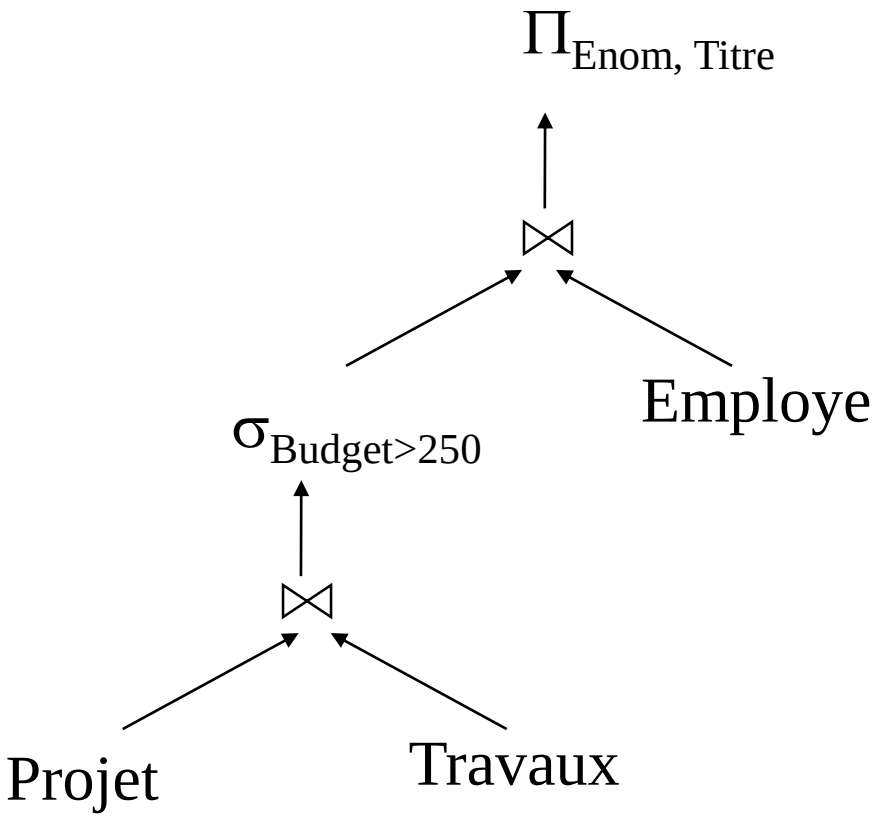


Comment passer de l'un à l'autre?

Commutativité de la sélection et de la jointure

$$\sigma_{Budget>250}\ (\ E \bowtie (\ P \bowtie T)\ )$$

$$(\ E \bowtie \sigma_{Budget>250}(\ P \bowtie T)\ )$$

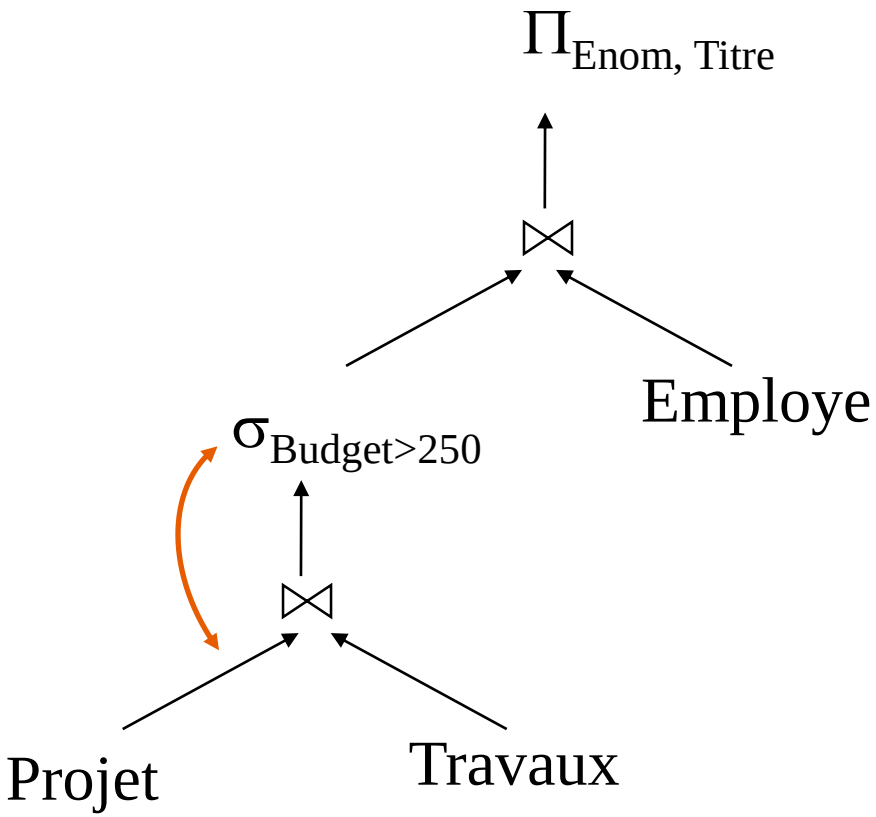


Comment passer de l'un à l'autre?

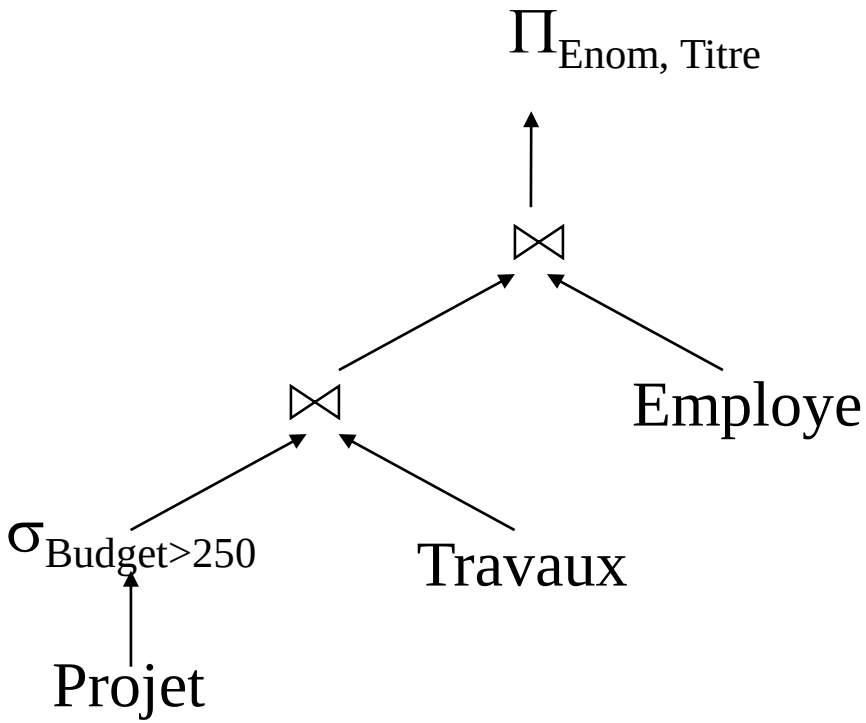
Commutativité de la sélection et de la jointure

$$(E \bowtie \sigma_{Budget>250} (P \bowtie T))$$

$$(E \bowtie (\sigma_{Budget>250} (P) \bowtie T))$$



Comment passer de l'un à l'autre?



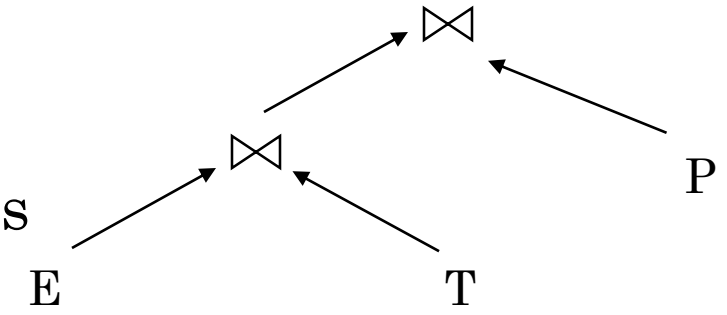
Commutativité de la sélection et de la jointure

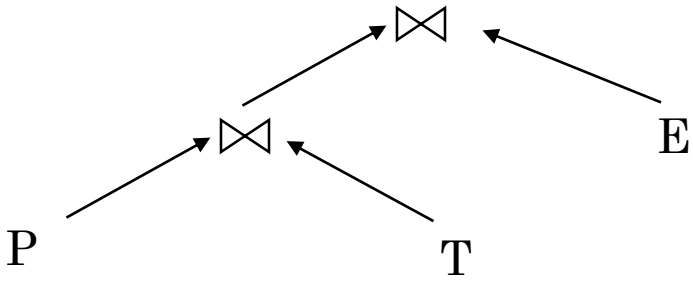
$$(E \bowtie \sigma_{Budget>250} (P \bowtie T))$$

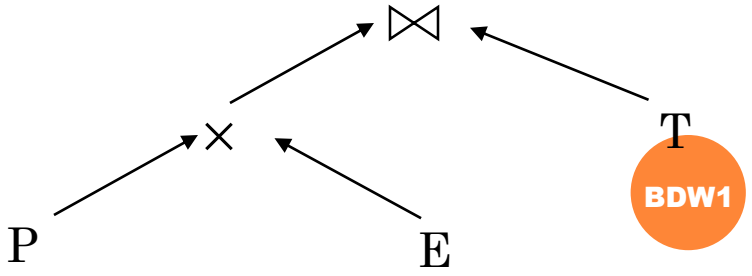
$$(E \bowtie (\sigma_{Budget>250} (P) \bowtie T))$$

Arbres de jointures

- Des arbres de jointures équivalents peuvent être obtenus en appliquant les règles de commutativité et d'associativité





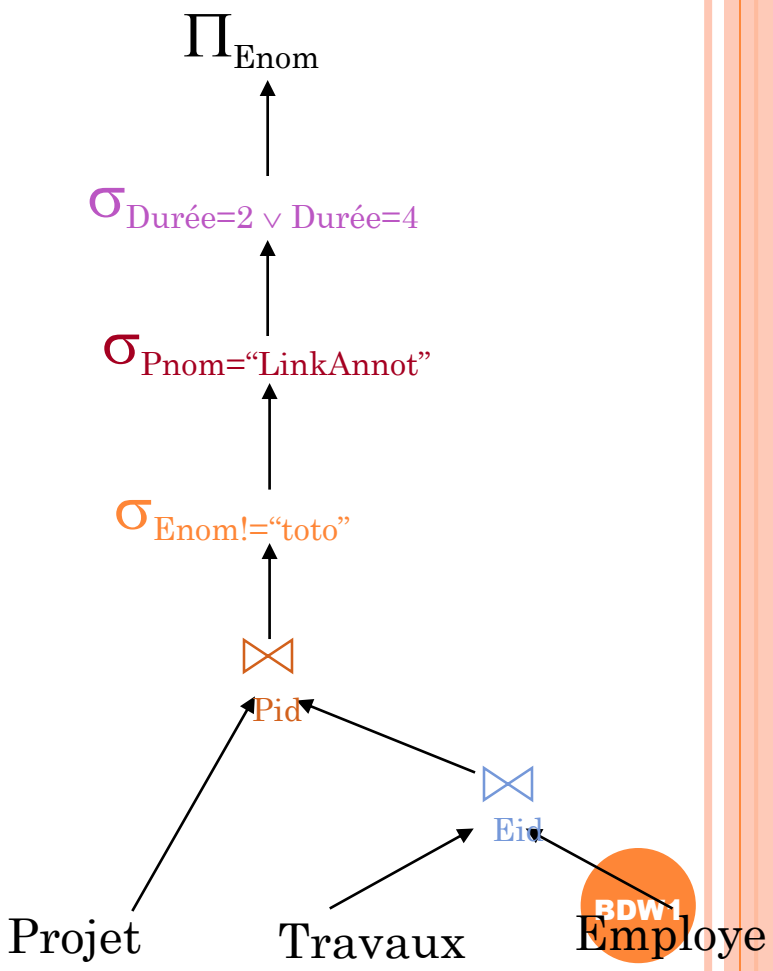



```
SELECT    *
FROM      E, T, P
WHERE     E.Eid=T.Eid
AND       T.Pid=P.Pid
```

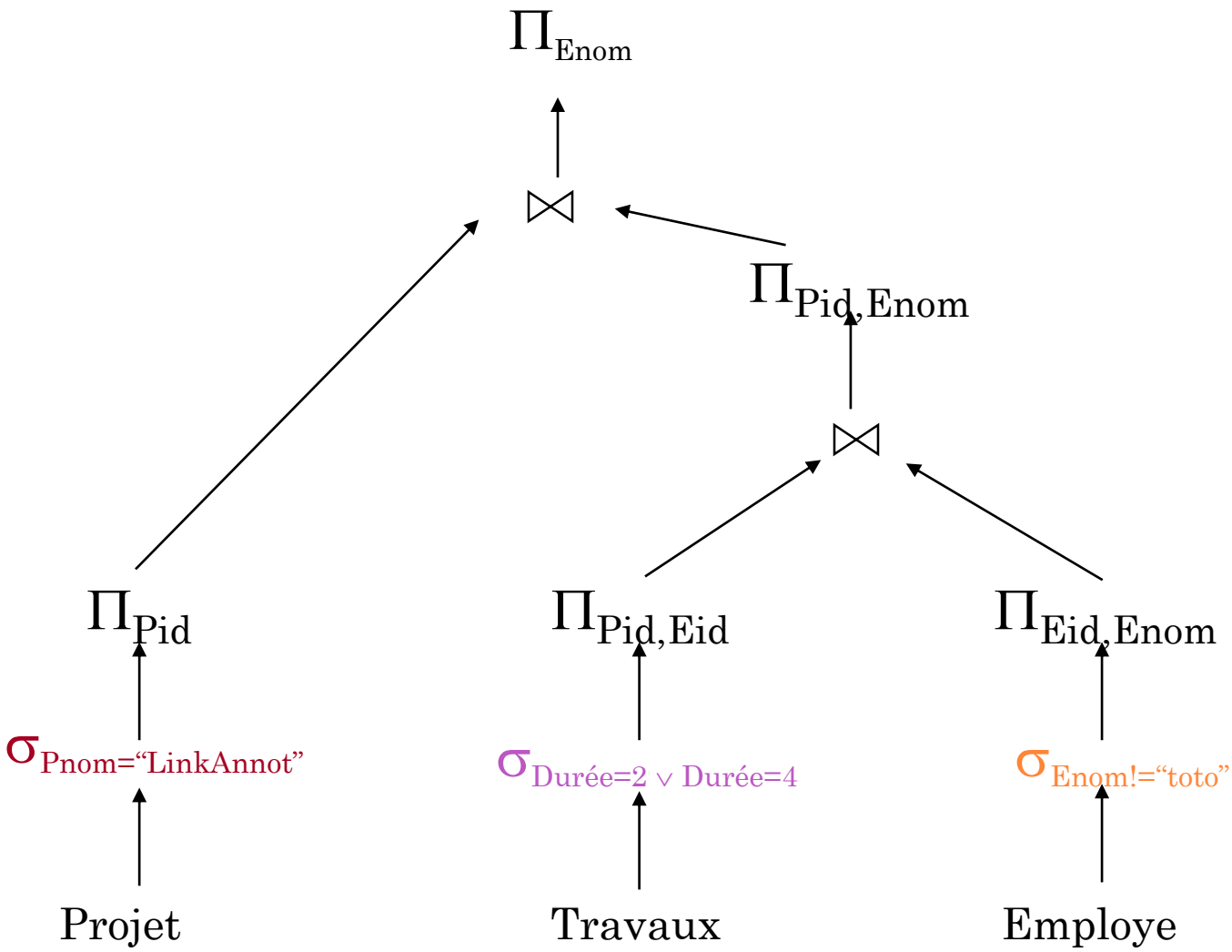
Rappel

- Exemple :

```
SELECT  Enom
FROM    Employe E, Travaux T,
Projet P
WHERE   E.Eid = T.Eid
AND     T.Pid = P.Pid
AND     Enom !='Toto'
AND     Pnom = 'Linkannot'
AND     (Durée=2 OR Durée=4)
```



Autre plan équivalente

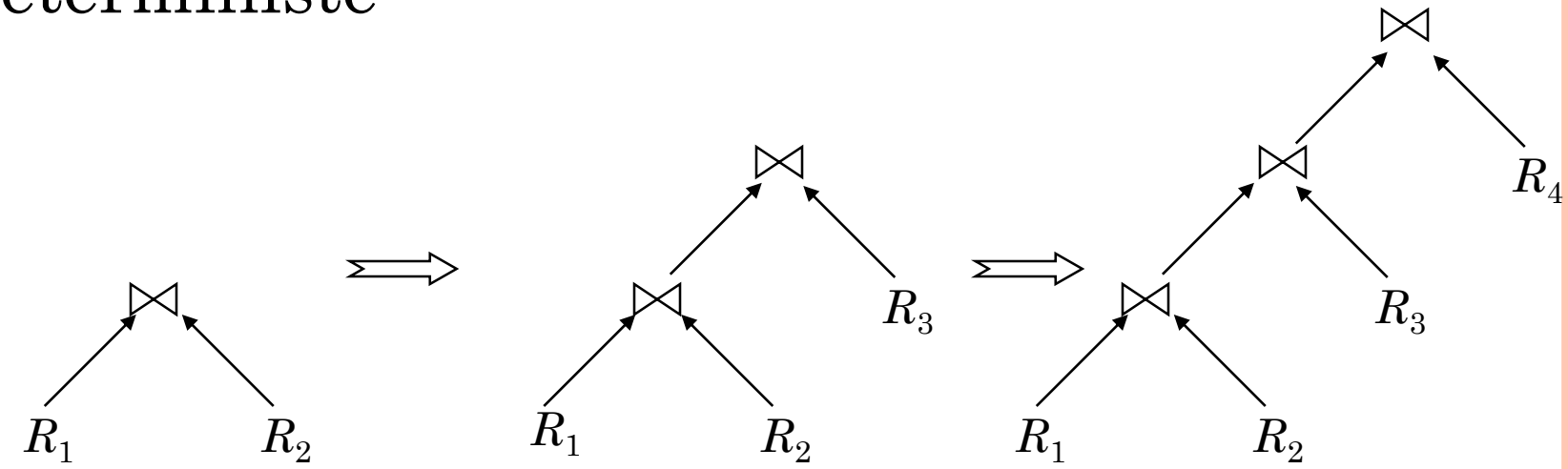


Stratégie de recherche

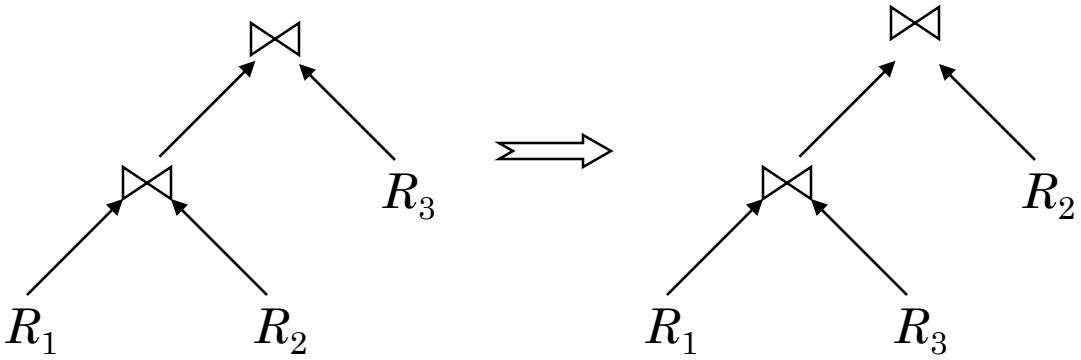
- Déterministe :
  - part des relations de base et construit les plans en ajoutant une relation à chaque étape
  - programmation dynamique: largeur-d'abord
  - excellent jusqu'à 5-6 relations
- Aléatoire :
  - recherche de l'optimalité autour d'un point de départ aléatoire
  - réduit le temps d'optimisation (au profit du temps d'exécution)
  - meilleur avec > 5-6 relations
  - recuit simulé
  - amélioration itérative

Stratégies de recherche

- Déterministe



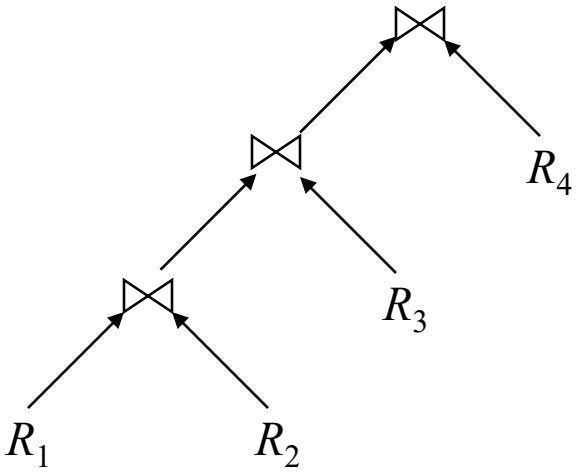
- Aléatoire



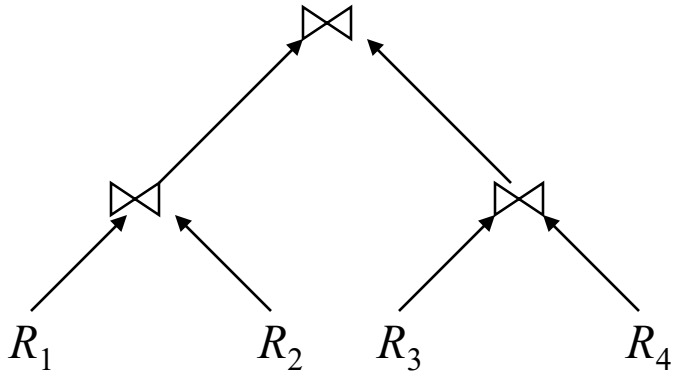
Algorithmes de recherche

- Limiter l'espace de recherche
  - heuristiques
    - par ex. appliquer les opérations unaires avant les autres
  - limiter la forme des arbres

Arbre linéaire



Arbre touffu



Les algorithmes de jointures

- Les algorithmes de jointures sont couteux
- Il existe différents algorithmes d'implantation des jointures
  - Jointure par boucles imbriquées
  - Jointure par tri fusion
  - Jointure par hachage
- Exemple :
  - On considère la jointure $R \bowtie_A S$ *avec* $R(A,B)$ *et* $S(A,C)$

Jointure par boucle imbriquée

- Algorithme le plus simple pour évaluer la jointure
- Principe:
  - compare chaque nuplet de la relation **R** à *tous* les nuplets de la relation **S**,
  - concaténer les nuplets pour lesquels la condition de jointure est vérifiée
  - ajouter le nouveau tuple au résultat.

```
T = {};
Pour chaque tuple r de R faire
    Pour chaque tuple s de S faire
      Si r.A==s.A Alors T = T ∪ (r.A, r.B, s.C)
    FinPour
FinPour
```

- Principe: Amélioration possible par la création d'un index sur A

JOINTURE PAR TRI-FUSION

- Algorithme utile quand il n'y a pas d'index sur des tables de grandes tailles
- Principe:
  - Trier R et S par rapport à l'attribut de jointure A
  - Fusionner R et S :
    - Positionner un pointeur courant pt_R et pt_S respectivement sur le premier nuplet de R et S
    - Comparer les attributs de jointure de ces deux nuplets.
      - Si l'attribut A du nuplet pointé par pt_R est égal à l'attribut **A** du nuplet pointé par pt_S,

        Alors on génère un nuplet résultat et on incrémente pt_R
      - Sinon, si l'attribut A du nuplet pointé par pt_R est supérieur à l'attribut **A** du nuplet pointé par pt_S,

        Alors on incrémente pt_S
      - Sinon, si l'attribut **A** du nuplet pointé par pt_R est inférieur à l'attribut **A** du nuplet pointé par pt_S

        Alors on incrémente pt_R afin qu'il pointe sur le nuplet suivant de **R**.

JOINTURE PAR HACHAGE

- Algorithme utile quand il y a un index sur l'attribut de jointure et qu'une des deux relations peut être mis en mémoire
- Principe:

  On suppose que S est une relation de taille plus petite que R
  - Hacher *S* avec une fonction de hachage H (clé=A; valeur=*tuple de S*)
  - Pour chaque tuple r de *R, rechercher dans la structure de hachage la clé r.A*

Comment optimiser ?

- Le processus d'optimisation s'appuie sur :
  1. Le schéma logique de la base, description des tables
  2. Le schéma physique de la base, indexes et chemins d'accès, tailles des blocs
  3. Des statistiques : taille des tables, des index, distribution des valeurs
  4. Des statistiques : taux de mises-à-jour
  5. Des algorithmes : il peuvent différer selon les systèmes

Estimer le coût d'un plan

- La fonction de coût donne les temps I/O et CPU
  - nombre d'instructions et d'accès disques (écriture/lecture)
- Estimation du *nombre d'accès disque pendant l'évaluation de chaque nœud* de l'arbre algébrique
  - utilisation de pipelines (en mémoire) ou de relations temporaires (écrits sur disque)
- Estimation de la *taille du résultat* de chaque nœud par rapport à ses entrées :
  - sélectivité des opérations

Estimation de la taille des résultats intermédiaires : Statistiques

- Relation R:
  - cardinalité : card(R)
  - taille d'un tuple : largeur de R
  - fraction de tuples participant une jointure
  - …
- Attribut A:
  - domaine : max(A), min(A), …
  - nombre de valeurs distinctes
  - distribution des valeurs (histogrammes)

- *Hypothèses :*
  - indépendance entre différentes valeurs d'attributs
  - distribution uniforme des valeurs d'attribut dans leur domaine

Tailles des relations intermédiaires

- Sélection :

$$taille(R) = card(R) * largeur(R)$$
$$card(\sigma_F(R)) = SF_\sigma(F) * card(R)$$

- où SF_σ est une *estimation* de la sélectivité du prédicat :

$$SF_\sigma(A = valeur) = \frac{1}{card(\prod_A(R))}$$

$$SF_\sigma(A > valeur) = \frac{max(A) - valeur}{max(A) - min(A)}$$

$$SF_\sigma(A < valeur) = \frac{valeur - min(A)}{max(A) - min(A)}$$

$$SF_\sigma(p(A_i) \wedge p(A_j)) = SF_\sigma(p(A_i)) * SF_\sigma(p(A_j))$$

$$SF_\sigma(p(A_i) \vee p(A_j)) = SF_\sigma(p(A_i)) + SF_\sigma(p(A_j)) - (SF_\sigma(p(A_i)) * SF_\sigma(p(A_j)))$$

$$SF_\sigma(A \in valeur) = SF_\sigma(A = valeur) * card(\{valeurs\})$$

Tailles des relations intermédiaires

- Projection

  $card(\Pi_A(R)) <= card(R)$ (égalité si A est unique)

- Produit cartésien

  $card(R \times S) = card(R) * card(S)$

- Union

  borne sup. : $card(R \cup S) = card(R) + card(S)$

  borne inf. : $card(R \cup S) = max\{card(R), card(S)\}$

- Différence

  borne sup. : $card(R–S) = card(R)$

  borne inf. : 0

Tailles des relations intermédiaires

- Jointure :
  - cas particulier: A est clé de R et B est clé étrangère de S :

$$card(R \bowtie_{A=B} S) = card(S)$$

  - plus généralement

$$card(R \bowtie S) = SF_{A=B} * card(R) * card(S)$$

Conclusion

Maintenant vous savez ce qui se passe après un clic sur le bouton ‘Execute’

BDW1