

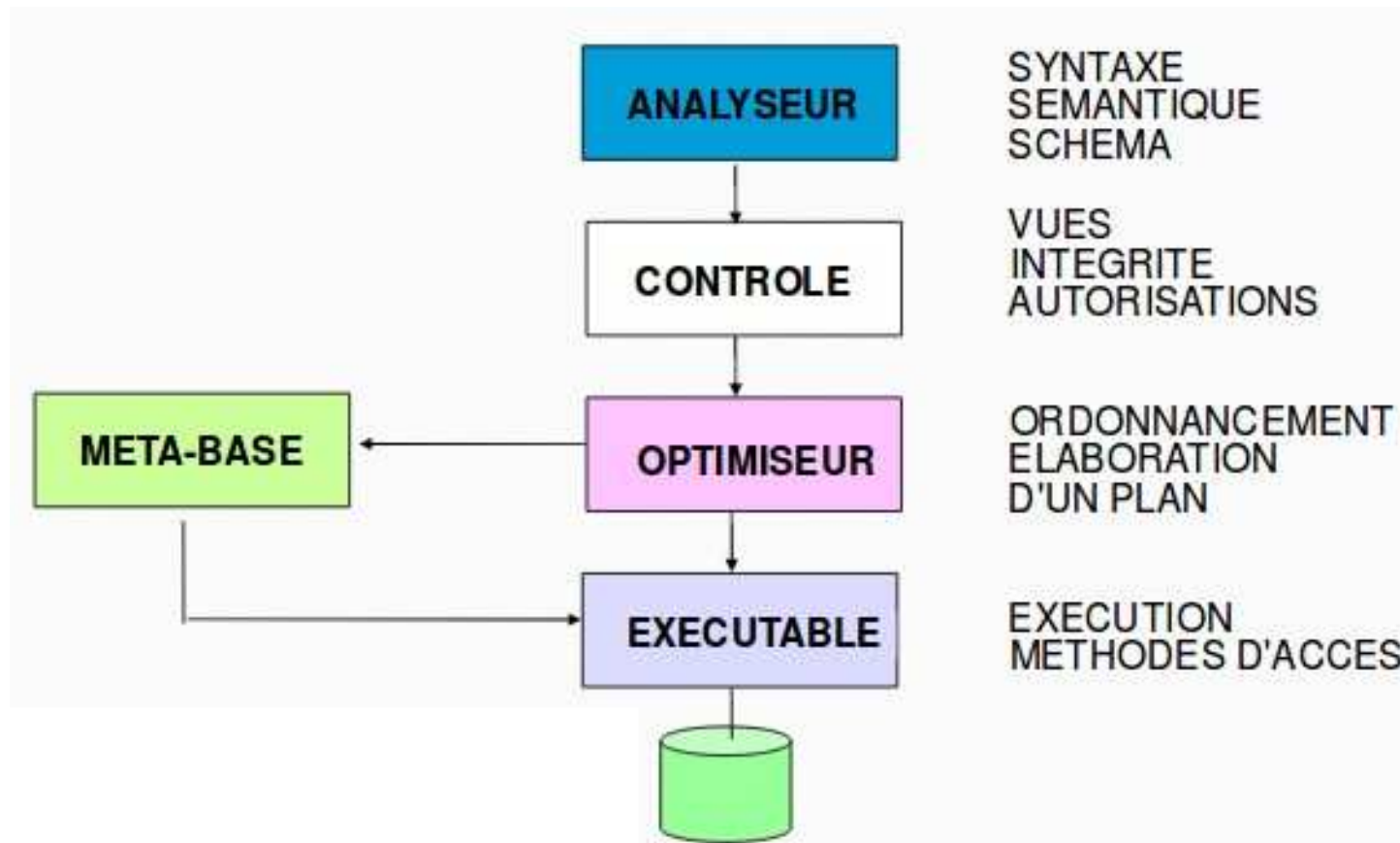
Entrepôts de Données et Big Data - HAI708I

Optimisation de Requête - Plan d'exécution logique

(Référence : Serge Abiteboul)



Architecture type d'un SGBD



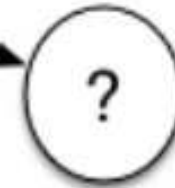
Pourquoi on s'intéresse à l'optimisation ?

- **Car cela a un impact sur le mouvement des blocs de données en mémoire**
 - Volume de données (>~1000 lignes)
 - Requêtes consommatrices de temps et de ressources
 - **Optimisation = tâche du SGBD**
 - **Mais l'utilisateur a un rôle à jouer :**
 - requête mal écrite
 - ➡ mauvaise optimisation
 - possibilité d'influer sur l'optimisation faite par le SGBD
 - ➡ meilleure optimisation
- ➡ **Comprendre les mécanismes de l'optimisation de requêtes afin d'écrire des requêtes bien construites**

Qu'est-ce que « optimiser » ?

```
select a1, a2, ...  
from T1, T2, ...  
where ...
```

Forme
déclarative



Forme
opérateur



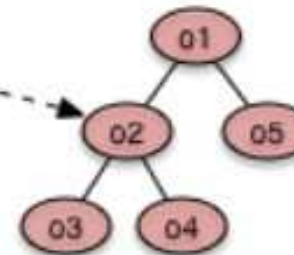
Résultat

- Requête SQL déclarative : elle ne dit pas comment calculer le résultat.
- Besoin d'un programme : le plan d'exécution = arbre constitué d'opérateurs

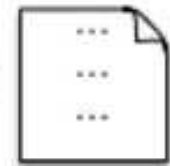
```
select a1, a2, ...  
from T1, T2, ...  
where ...
```

Forme
déclarative

?



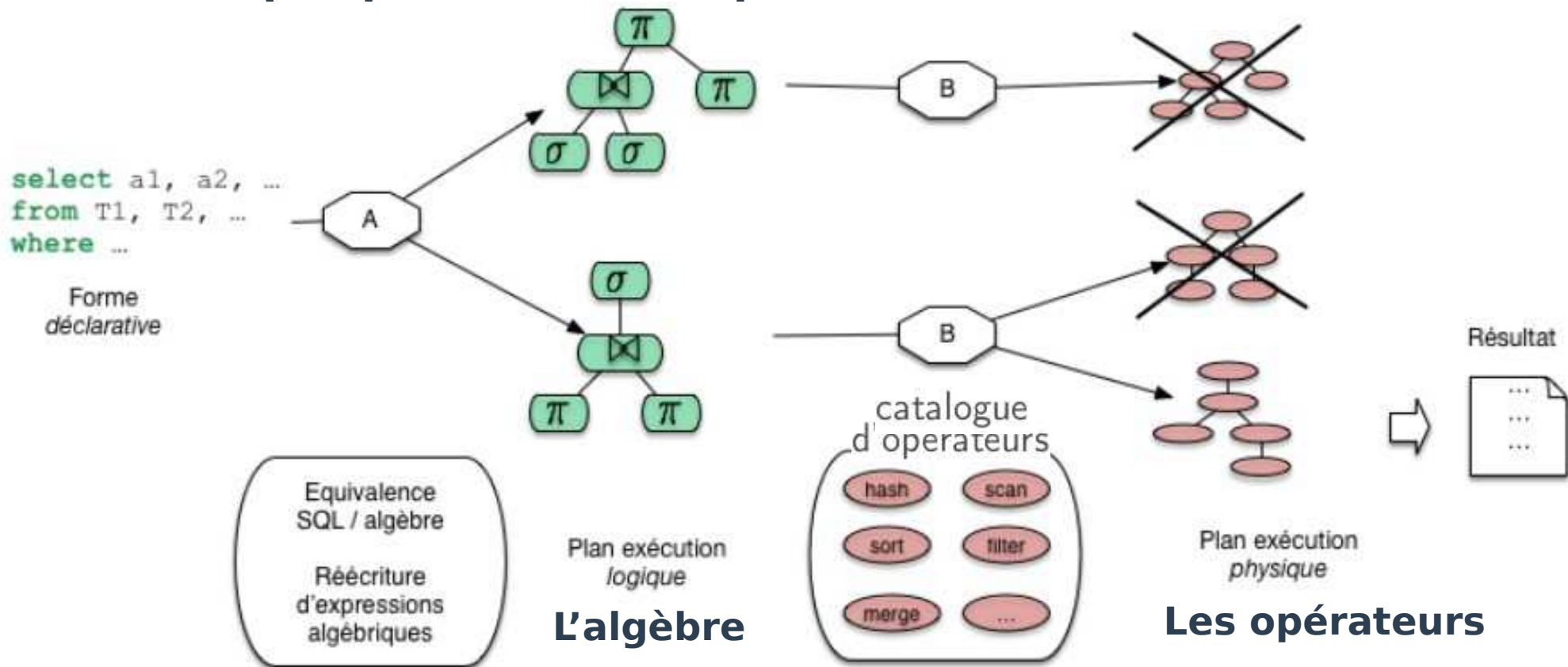
Forme
opérateur



Résultat

Qu'est-ce que « optimiser » ?

- Deux étapes pour obtenir le plan d'exécution



- À chaque étape, plusieurs choix : le SGBD les évalue et choisit le « meilleur »

Un exemple

- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL

```
select titre
from   Film f, Role r
where  nom_role = 'Ferguson'
and    f.id = r.id_ilm
and    f.annee = 1958
```

2 sélections
1 jointure
1 projection

Un exemple

- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL

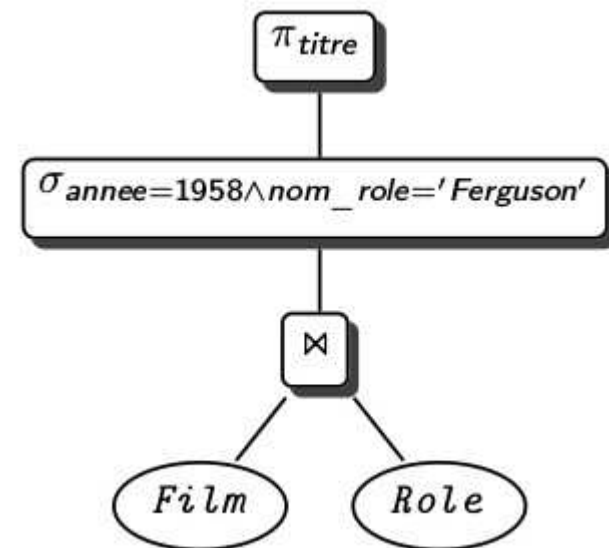


Plan d'exécution logique (l'algèbre)

```
select titre
from   Film f, Role r
where  nom_role = 'Ferguson'
and    f.id = r.id_ilm
and    f.annee = 1958
```

2 sélections
1 jointure
1 projection

$\pi_{titre}(\sigma_{annee=1958 \wedge nom_role='Ferguson'}(Film \bowtie_{id=id_film} Role))$



Un exemple

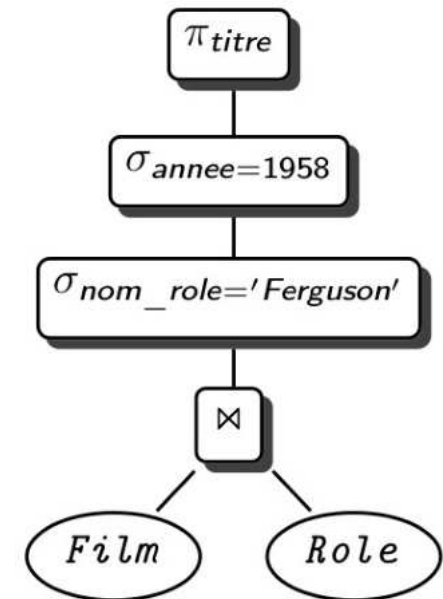
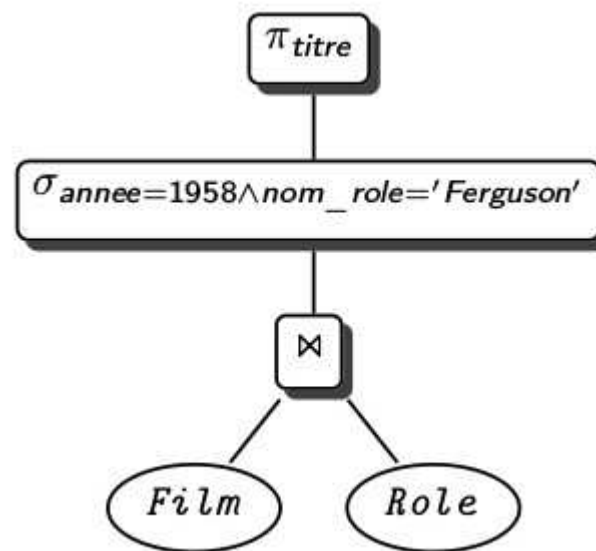
- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL



Plan d'exécution logique (l'algèbre)

```
select titre
from   Film f, Role r
where  nom_role = 'Ferguson'
and    f.id = r.id_ilm
and    f.annee = 1958
```



Un exemple

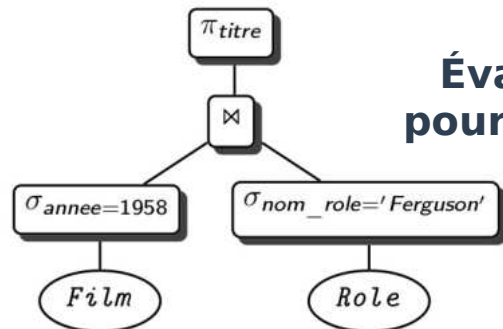
- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL



Plan d'exécution logique (l'algèbre)

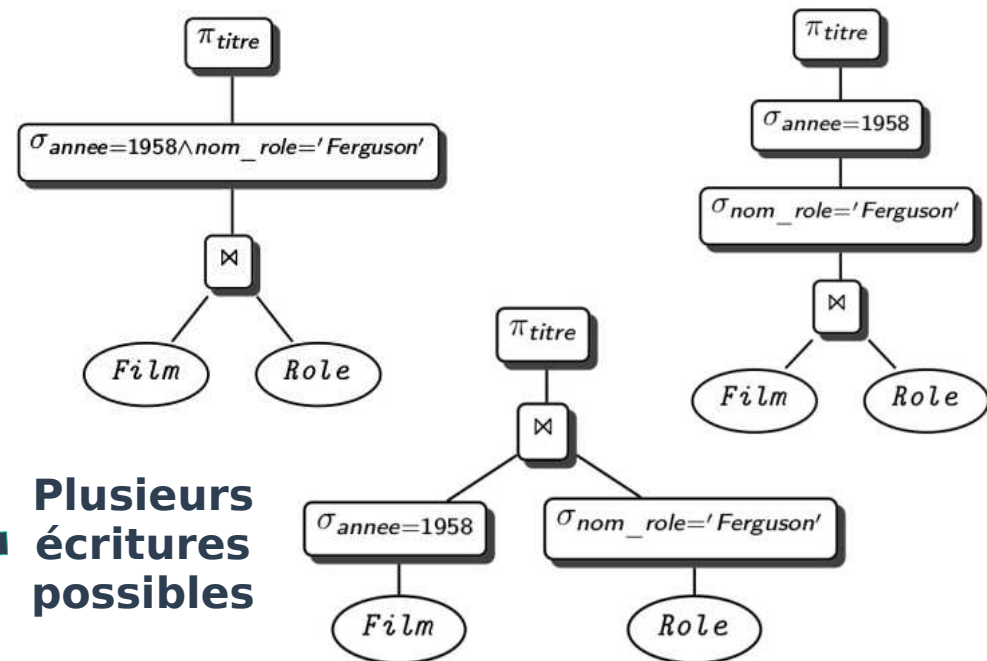
```
select titre
from Film f, Role r
where nom_role = 'Ferguson'
and f.id = r.id_ilm
and f.annee = 1958
```



Évaluation des coûts
pour trouver le meilleur
plan logique



Plusieurs
écritures
possibles



Un exemple

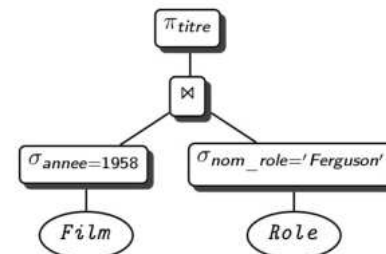
- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL



Plan d'exécution logique (l'algèbre)

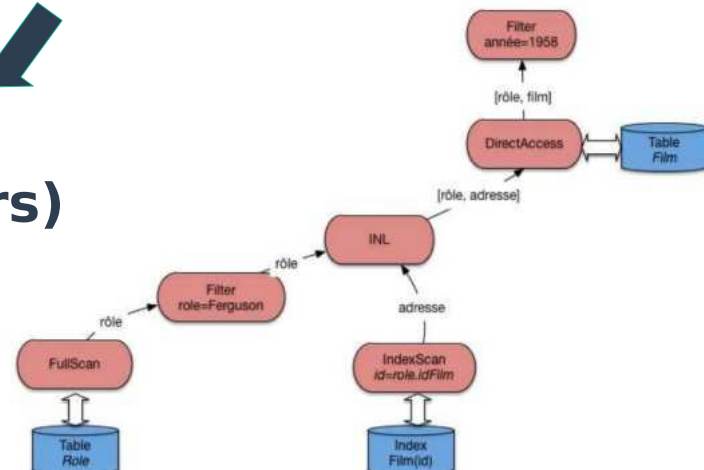
```
select titre
from   Film f, Role r
where  nom_role = 'Ferguson'
and    f.id = r.id_ilm
and    f.annee = 1958
```



Plan d'exécution physique (opérateurs)

Un opérateur = une opération

Plusieurs algorithmes par opération



Un exemple

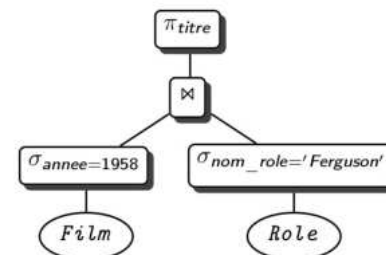
- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL



Plan d'exécution logique (l'algèbre)

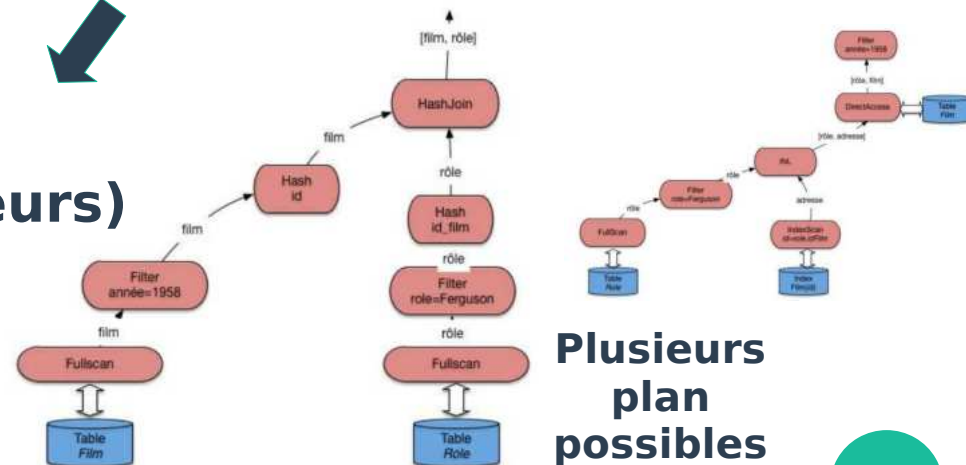
```
select titre
from Film f, Role r
where nom_role = 'Ferguson'
and f.id = r.id_film
and f.annee = 1958
```



Plan d'exécution physique (opérateurs)

Un opérateur = une opération

Plusieurs algorithmes par opération



Plusieurs
plan
possibles

Un exemple

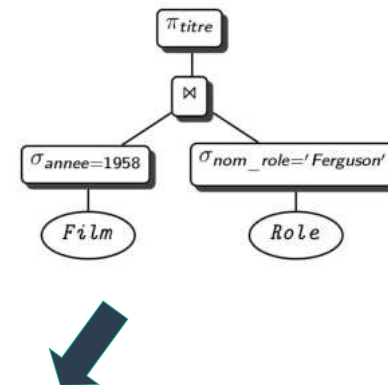
- Titre des films parus en 1958, où l'un des acteurs joue le rôle de John Ferguson.

Requête SQL



Plan d'exécution logique (l'algèbre)

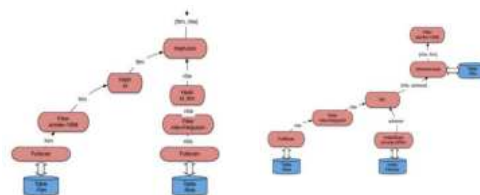
```
select titre
from Film f, Role r
where nom_role = 'Ferguson'
and f.id = r.id_ilm
and f.annee = 1958
```



Plan d'exécution physique (opérateurs)

Un opérateur = une opération

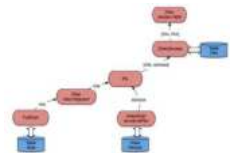
Plusieurs algorithmes par opération



Plusieurs plans possibles



Évaluation des coûts pour trouver le meilleur plan physique



Le rôle de l'optimiseur

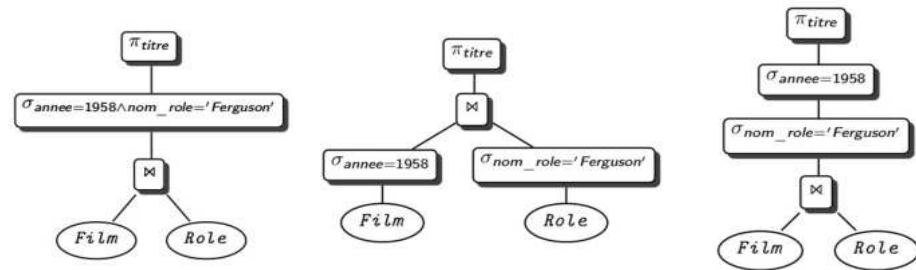
Trouver les expressions
équivalentes

Requête SQL



Plan d'exécution logique - PEL (l'algèbre)

```
select titre
from Film f, Role r
where nom_role = 'Ferguson'
and f.id = r.id_ilm
and f.annee = 1958
```

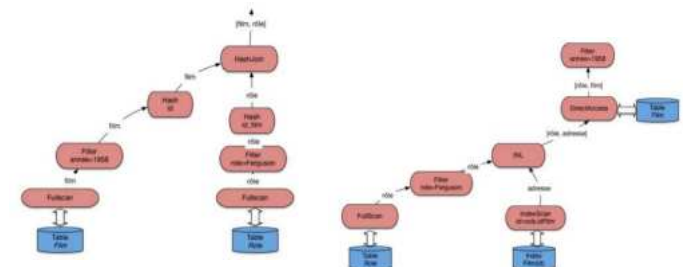


Choisir le bon algorithme
pour chaque opération

Plan d'exécution physique - PEP (opérateurs)

Un opérateur = une opération

Plusieurs algorithmes par opération



Le rôle de l'optimiseur

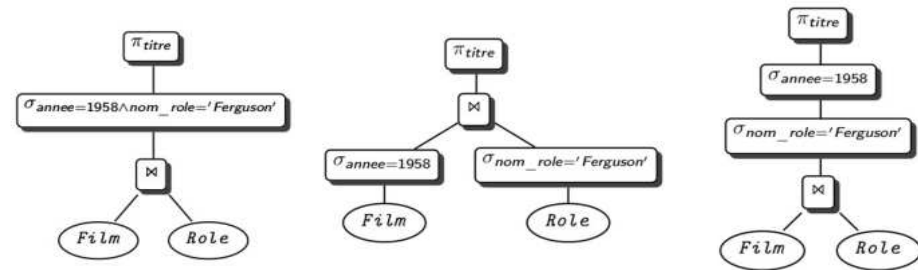
Trouver les expressions équivalentes

Requête SQL



Plan d'exécution logique - PEL (l'algèbre)

```
select titre
from   Film f, Role r
where  nom_role = 'Ferguson'
and    f.id = r.id_ilm
and    f.annee = 1958
```

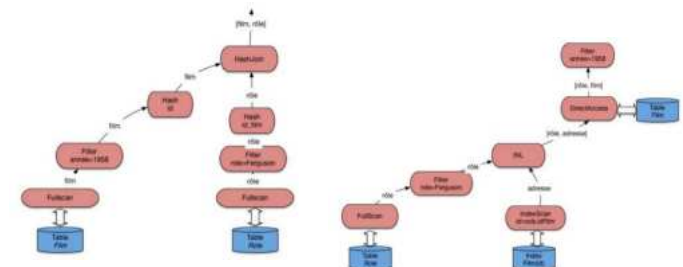


Choisir le bon algorithme pour chaque opération

Plan d'exécution physique - PEP (opérateurs)

Un opérateur = une opération

Plusieurs algorithmes par opération



Le rôle de l'optimiseur : la réécriture algébrique (PEL)

- **Problème :**

- suivant l'ordre des opérateurs algébriques dans un arbre, le coût d'exécution est différent

- **Pourquoi?**

- le coût des opérateurs varient en fonction du volume des données traitées : plus le nombre de n-uplets des relations traitées est petit, plus les coûts cpu et d'E/S sont minimisés
- certains opérateurs diminuent le volume des données (restriction, projection, ...)

➡ **Restructuration algébrique**

Le rôle de l'optimiseur : la réécriture algébrique (PEL)

- **Trouver les expressions équivalentes**

- l'algèbre permet d'obtenir une version opératoire de la requête
- les équivalences algébriques permettent d'explorer un ensemble de plans
- l'optimiseur évalue le coût (entrée / sortie) de chaque plan : différentes fonctions / modèles de coût existantes

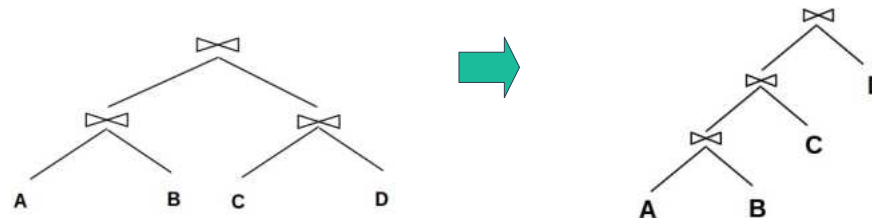
Exemples de règles de réécriture

1. Commutativité des jointures : $R \bowtie S \equiv S \bowtie R$
2. Associativité des jointures : $(R \bowtie S) \bowtie T \equiv R \bowtie (S \bowtie T)$
3. Regroupement des sélections : $\sigma_{A='a' \wedge B='b'}(R) \equiv \sigma_{A='a'}(\sigma_{B='b'}(R))$
4. Commutativité des sélections et projections : $\pi_{A_1, A_2, \dots, A_p}(\sigma_{A_i='a'}(R)) \equiv \sigma_{A_i='a'}(\pi_{A_1, A_2, \dots, A_p}(R))$
5. Commutativité des sélections et jointures : $\sigma_{A='a'}(R[\dots A \dots] \bowtie S) \equiv \sigma_{A='a'}(R) \bowtie S$
6. Commutativité des projections et jointures :

$$\pi_{A_1 \dots A_p B_1 \dots B_q}(R \bowtie_{A_i=B_j} S) \equiv \pi_{A_1 \dots A_p}(R) \bowtie_{A_i=B_j} \pi_{B_1 \dots B_q}(S) \quad i \in \{1, \dots, p\}, j \in \{1, \dots, q\}$$

Le rôle de l'optimiseur : la réécriture algébrique (PEL)

- **Trouver les expressions équivalentes**
 - MAIS impossible d'énumérer tous les plans possibles
 - ➡ Utilisations d'heuristiques
 - Heuristique classique = réduction de la taille des données
 - ➡ Opérations réductrices (sélections et projections) groupées sur chaque relation le plus tôt possible, et jointures regroupées
 - grouper les sélections aux feuilles (dégrouper puis descendre)
 - descendre les projections
 - regrouper les jointures du même côté de l'arbre (deep-left plan) ou réordonner les jointures (choisir en 1^{er} les relations de taille minimum)



Le rôle de l'optimiseur : la réécriture algébrique - fonction de coût (PEL)

Fonction / Modèle de coût : un exemple

- **Facteur de sélectivité s** : utile pour estimer le coût si on ne connaît pas la répartition exacte des données
 - Proportion de n -uplets d'une relation qui satisfont une condition

- **Exemple :**

SELECT * FROM R1, R2

➡ $s = 1$

SELECT * FROM R1 WHERE A = valeur

➡ $s = 1 / \text{CARD}(A)$ avec un modèle uniforme

Le rôle de l'optimiseur : la réécriture algébrique - fonction de coût (PEL)

Sélectivité des restrictions / sélections

$TAILLE(\sigma(R)) = s * TAILLE(R)$ avec :

- $\square s(A = \text{valeur}) = 1 / \text{CARD}(A)$
- $\square s(A > \text{valeur}) = (\max(A) - \text{valeur}) / (\max(A) - \min(A))$
- $\square s(A < \text{valeur}) = (\text{valeur} - \min(A)) / (\max(A) - \min(A))$
- $\square s(A \text{ IN liste valeurs}) =$
 $(1/\text{CARD}(A)) * \text{CARD}(\text{liste valeurs})$
- $\square s(P \text{ et } Q) = s(P) * s(Q)$
- $\square s(P \text{ ou } Q) = s(P) + s(Q) - s(P) * s(Q)$
- $\square s(\text{not } P) = 1 - s(P)$

Le rôle de l'optimiseur : la réécriture algébrique - fonction de coût (PEL)

Sélectivité des Jointures

- **$TAILLE(R1 \bowtie_{R1.A=R2.B} R2) = s * TAILLE(R1) * TAILLE(R2)$**
 - s dépend du type de jointure et de la corrélation des colonnes :
 - $s = 0$ si aucun n-uplet n'est joint
 - $s = 1 / \text{MIN}(\text{CARD}(A), \text{CARD}(B))$ si distribution uniforme équiprobable des attributs A et B sur un même domaine
 - $s = 1$ si produit cartésien
- **Cas particulier :**
 - Si A est clé de R1 et B est clé étrangère dans R2 référençant A alors
 $TAILLE(R1 \bowtie_{R1.A=R2.B} R2) = TAILLE(R2)$
 - $s = 1 / \text{MIN}(\text{CARD}(A), \text{CARD}(B)) = 1 / \text{CARD}(A)$
 - $s \sim 1 / TAILLE(R1)$

Un exemple de réécriture algébrique (PEL)

- **Soit le schéma relationnel (notation simplifiée) :**

Cinéma (ID-cinéma, nom, adresse)

Salle (ID-salle, ID-cinéma, capacité)

Séance (ID-salle, heure-début, film)

- **Requête: quels films commencent au Multiplex à 20 heures?**

```
SELECT Séance.film
```

```
FROM Cinéma, Salle, Séance
```

```
WHERE Cinéma.nom = 'Multiplex' AND
```

```
    Séance.heure-début = 20 AND
```

```
    Cinéma.ID-cinéma = Salle.ID-cinéma AND
```

```
    Salle.ID-salle = Séance.ID-salle ;
```

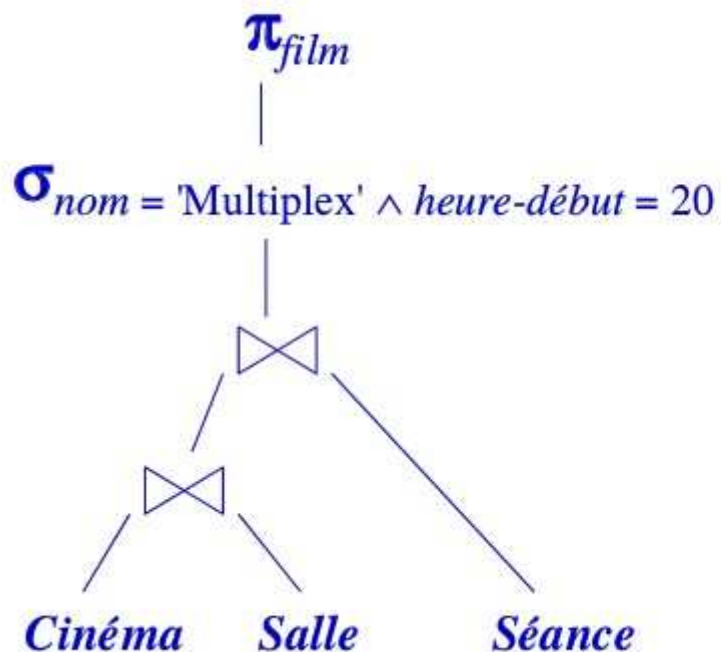
- **Expression algébrique**

$$\pi_{\text{film}} (\sigma_{\text{nom} = \text{'Multiplex'} \wedge \text{heure-début}=20} ((\text{Cinéma} \bowtie \text{Salle}) \bowtie \text{Séance}))$$

Un exemple de réécriture algébrique (PEL)

- Arbre algébrique de requête

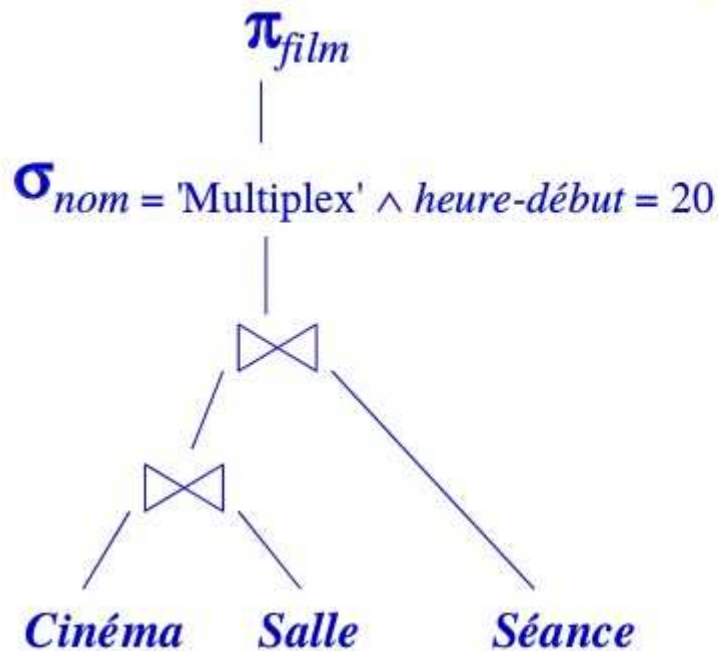
$\pi_{film} (\sigma_{nom = 'Multiplex' \wedge heure-début=20} ((Cinéma \bowtie Salle) \bowtie Séance))$



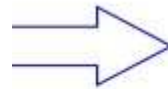
Un exemple de réécriture algébrique (PEL)

- Arbre algébrique de requête

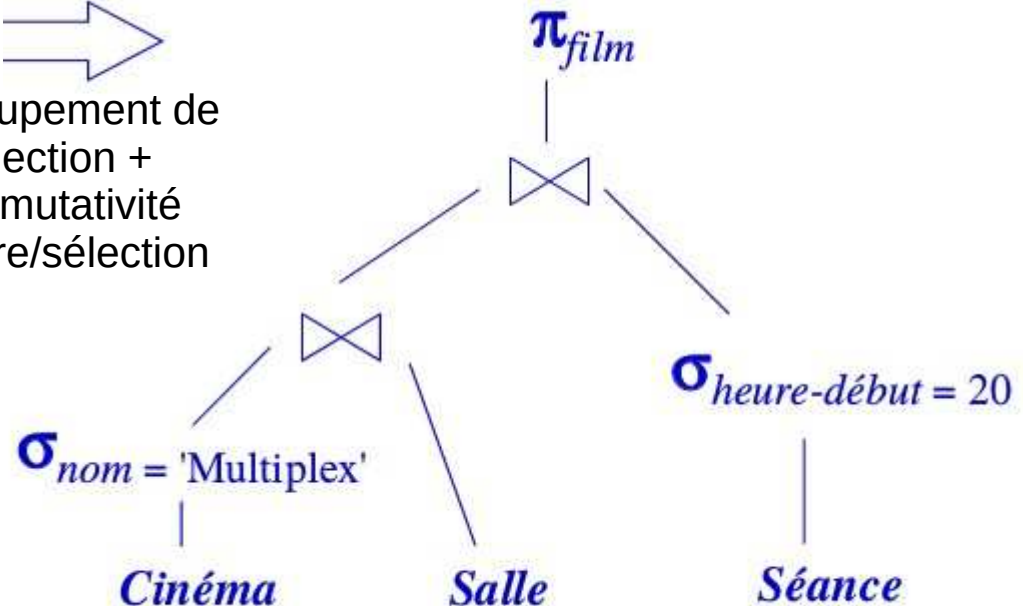
$\pi_{film}(\sigma_{nom = 'Multiplex' \wedge heure-début=20}((Cinéma \bowtie Salle) \bowtie Séance))$



Règles de réécriture utilisées :



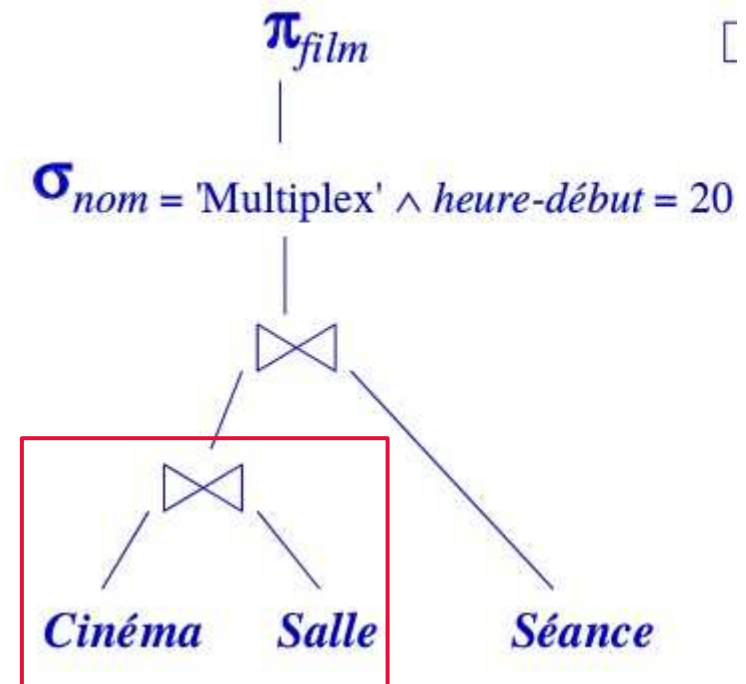
Regroupement de sélection + commutativité jointure/sélection



$\pi_{film}(\sigma_{heure-début=20}Séance) \bowtie ((\sigma_{nom = 'Multiplex'}Cinéma) \bowtie Salle))$

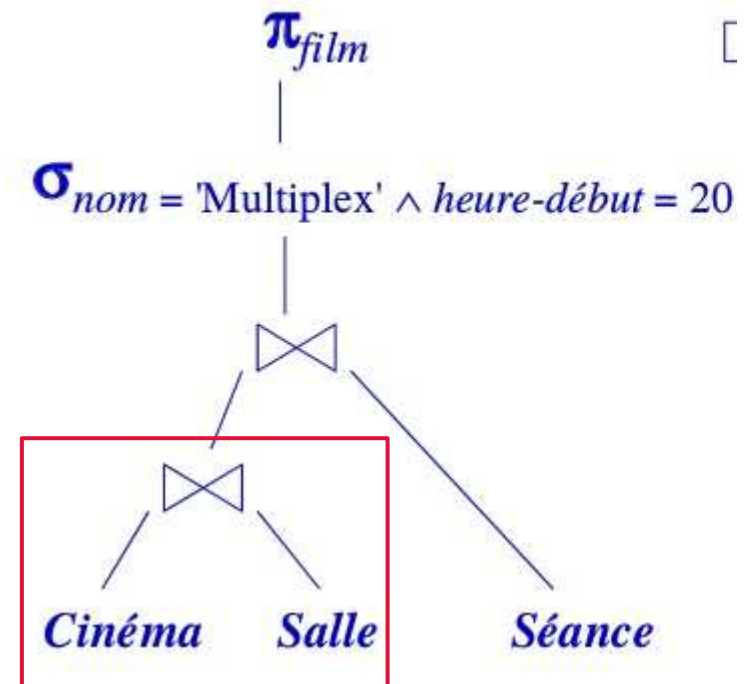
Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèses (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h
- **Plan 1 :**
 - Jointure : on lit $4 * 6 = 24$ lignes
et on produit $50 \% * 6 = 3$ lignes



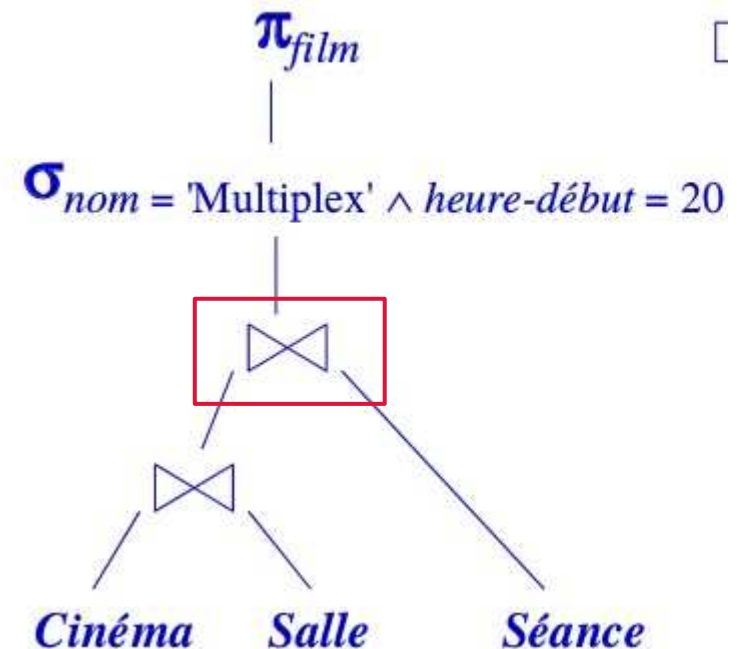
Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèses (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h
- **Plan 1 :**
 - Jointure : on lit $4 * 6 = 24$ lignes
et on produit $50 \% * 6 = 3$ lignes
=> Sélectivité de la jointure = 0,5
(la moitié des salles sont des salles de Cinéma)



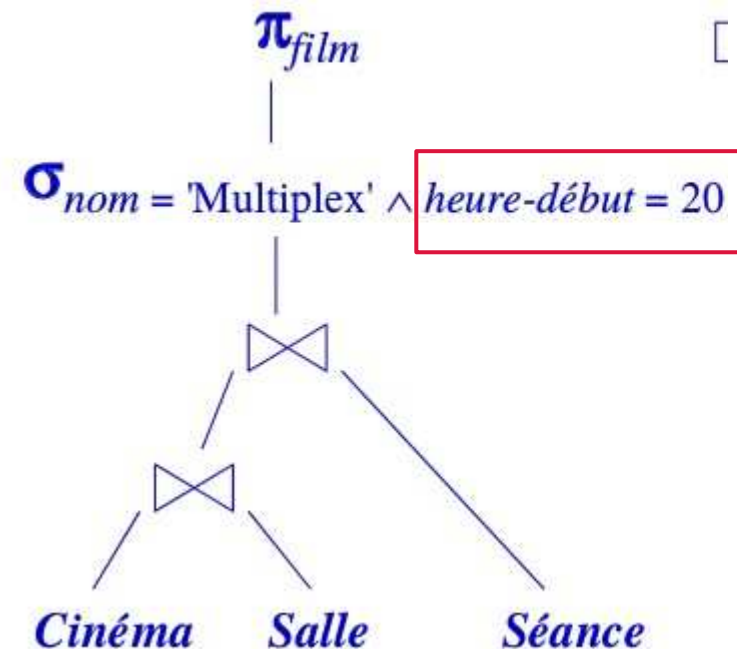
Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèses (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h
- **Plan 1 :**
 - Jointure : on lit $4 * 6 = 24$ lignes
et on produit $50 \% * 6 = 3$ lignes
 - Jointure : on lit $3 * 50 = 150$ lignes
et on produit 50 lignes
=> Sélectivité de la jointure = 1
(toutes les séances sont des séances de salles de cinéma)



Un exemple de réécriture algébrique : calcul de coût (PEL)

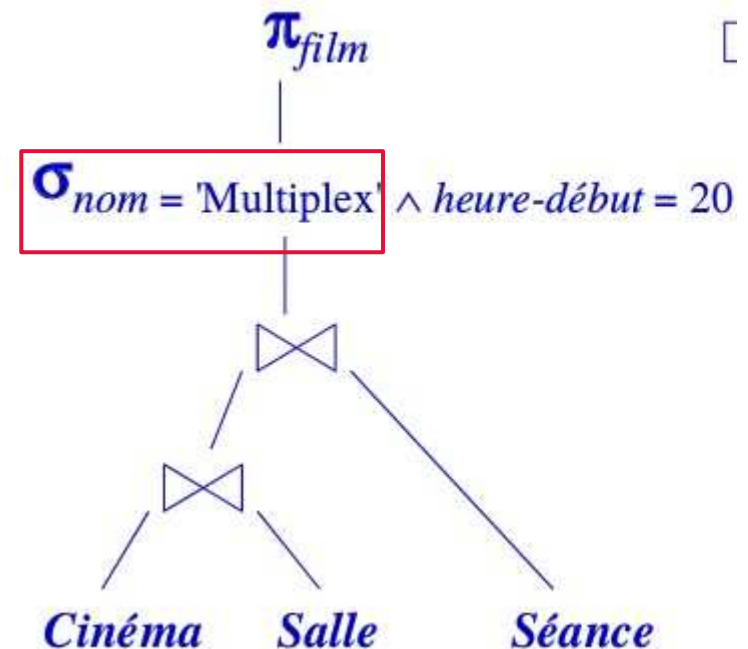
- **Hypothèses (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h
- **Plan 1 :**
 - Jointure : on lit $4 * 6 = 24$ lignes
et on produit $50 \% * 6 = 3$ lignes
 - Jointure : on lit $3 * 50 = 150$ lignes
et on produit 50 lignes
 - Sélection : on lit 50 lignes
et on produit $50 \% * 50 = 25$ lignes
=> Sélectivité de la restriction = 0,5
(la moitié des séances sont après 20h)



Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèses (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h

- **Plan 1 :**
 - Jointure : on lit $4 * 6 = 24$ lignes
et on produit $50 \% * 6 = 3$ lignes
 - Jointure : on lit $3 * 50 = 150$ lignes
et on produit 50 lignes
 - Sélection : on lit 50 lignes
et on produit $50 \% * 50 = 25$ lignes
 - Sélection : on lit 25 lignes
et on produit $20 \% * 25 = 5$ lignes
=> Sélectivité de la restriction = 0,2
(20 % des cinémas sont des Multiplex)



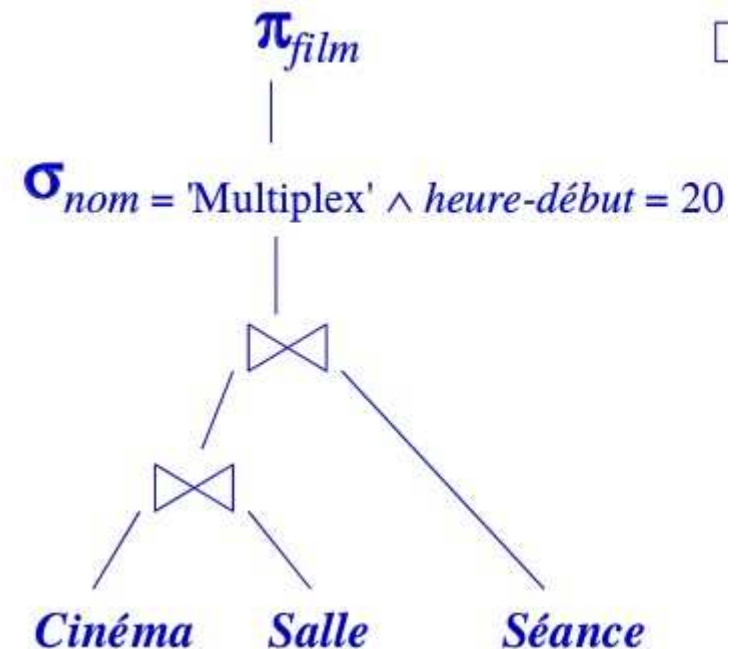
Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèses (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h

- **Plan 1 :**

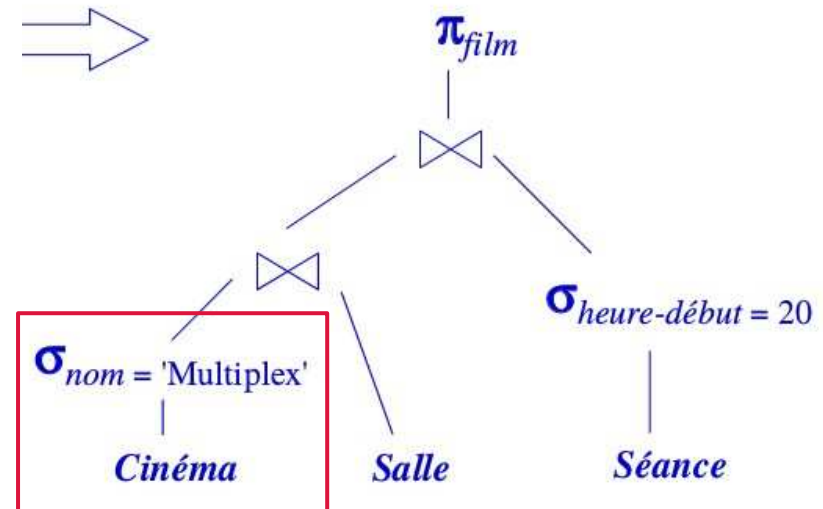
- Jointure : on lit $4 * 6 = 24$ lignes
et on produit $50 \% * 6 = 3$ lignes
- Jointure : on lit $3 * 50 = 150$ lignes
et on produit 50 lignes
- Sélection : on lit 50 lignes
et on produit $50 \% * 50 = 25$ lignes
- Sélection : on lit 25 lignes
et on produit $20 \% * 25 = 5$ lignes
- On laisse de côté la projection (même coût dans les deux cas et même nombre de lignes)

➡ Coût (E/S) : $24E + 3S + 150E + 50S + 50E + 25S + 25E + 5S = 332$ lignes E/S



Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèse (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h
- **Plan 1 :** coût (E/S) = 332 lignes E/S
- **Plan 2 :**
 - Sélection : on lit 4 lignes
et on produit $20\% * 4 = 1$ lignes
=> Sélectivité de la restriction = 0,2
(20 % des cinémas sont des Multiplex)



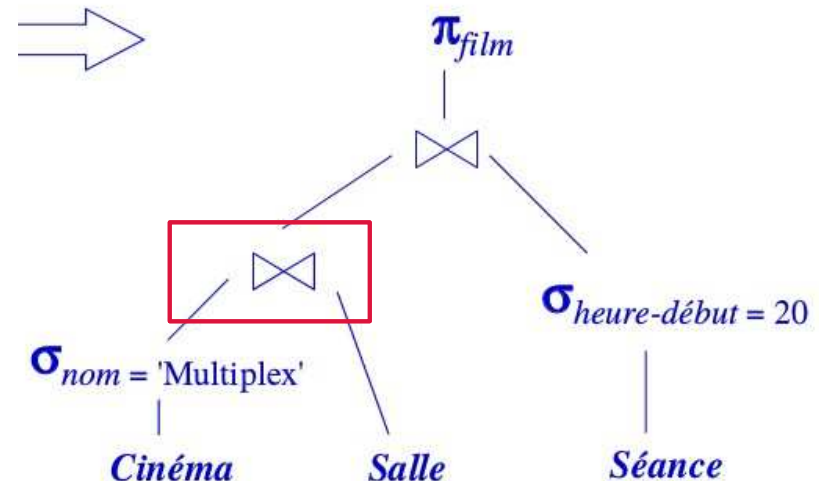
➡ Plan 2 optimal

Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèse (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h

- **Plan 1 :** coût (E/S) = 332 lignes E/S

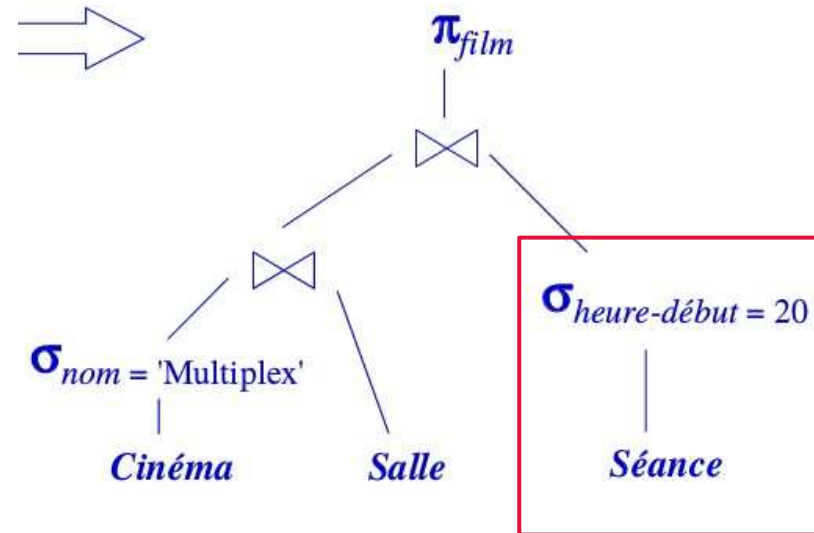
- **Plan 2 :**
 - Sélection : on lit 4 lignes
et on produit $20\% * 4 = 1$ lignes
 - Jointure : on lit $1 * 6 = 6$ lignes
et on produit $50\% * 6 = 3$ lignes
=> Sélectivité de la jointure = 0,5
=> Nombre de lignes MAX : dans le pire des cas, toutes les salles sont des salles du cinéma 'Mutiplex'



➡ Plan 2 optimal

Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèse (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h
- **Plan 1 :** coût (E/S) = 332 lignes E/S
- **Plan 2 :**
 - Sélection : on lit 4 lignes
et on produit $20\% * 4 = 1$ lignes
 - Jointure : on lit $1 * 6 = 6$ lignes
et on produit $50\% * 6 = 3$ lignes
 - Sélection : on lit 50 lignes
et on produit $50\% * 50 = 25$ lignes
=> Sélectivité de la restriction = 0,5
(la moitié des séances sont après 20h)



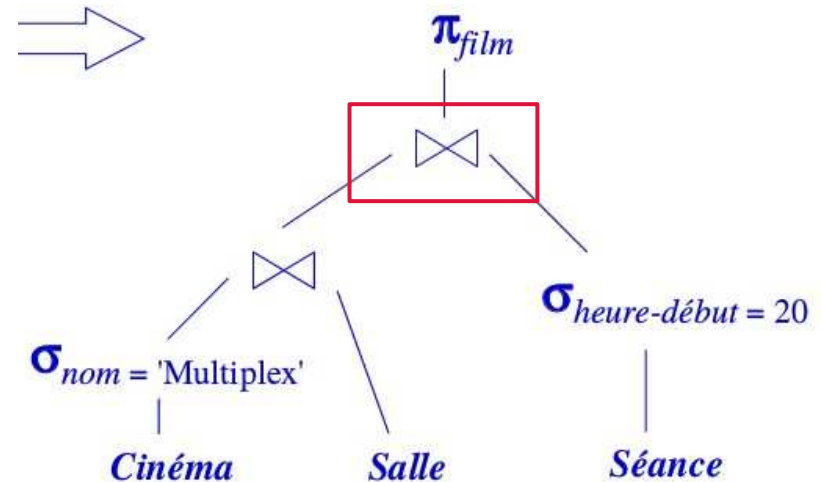
➡ Plan 2 optimal

Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèse (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h

- **Plan 1 :** coût (E/S) = 332 lignes E/S

- **Plan 2 :**
 - Sélection : on lit 4 lignes
et on produit $20\% * 4 = 1$ lignes
 - Jointure : on lit $1 * 6 = 6$ lignes
et on produit $50\% * 6 = 3$ lignes
 - Sélection : on lit 50 lignes
et on produit $50\% * 50 = 25$ lignes
 - Jointure : on lit $25 * 3 = 75$ lignes
et on produit 25 lignes
=> Sélectivité de la jointure = 1
=> Nombre de lignes MAX



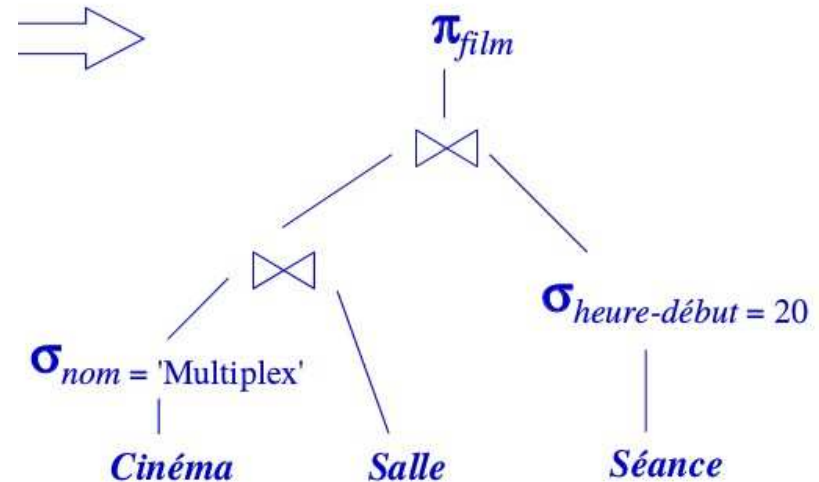
➡ Plan 2 optimal

Un exemple de réécriture algébrique : calcul de coût (PEL)

- **Hypothèse (en nombre de lignes) :**
 - Cinéma : 4 lignes dont 20 % de Multiplex
 - Salle : 6 lignes dont 50 % des salles de Cinéma
 - Séance : 50 lignes et 50 % des séances après 20h

- **Plan 1 :** coût (E/S) = 332 lignes E/S

- **Plan 2 :**
 - Sélection : on lit 4 lignes
et on produit 20 % * 4 = 1 lignes
 - Jointure : on lit 1 * 6 = 6 lignes
et on produit 50 % * 6 = 3 lignes
 - Sélection : on lit 50 lignes
et on produit 50 % * 50 = 25 lignes
 - Jointure : on lit 25 * 3 = 75 lignes
et on produit 25 lignes
 - On laisse de côté la projection (même coût dans les deux cas et même nombre de lignes)



➡ Coût (E/S) : 4E + 1S + 6E + 3S + 50E + 25S + 75E + 25S = 189 lignes E/S

➡ Plan 2 optimal

Un exemple de réécriture algébrique qui échoue (PEL)

- **Question: le plan ainsi obtenu est-il toujours optimal?**

- Réponse: NON, d'autres facteurs peuvent intervenir

- **On rajoute une table Film, en plus de Cinéma, Salle, Séance**

Film (film, réalisateur, année)

- **Requête: les réalisateurs des films qu'on peut voir après 14h**

```
SELECT Film.réalisateur
FROM Film, Séance
WHERE Séance.heure-début > 14 AND Film.film = Séance.film
```

- **Expressions algébrique**

- Initiale: $\pi_{\text{réalisateur}} (\sigma_{\text{heure-début} > 14} (\text{Film} \bowtie \text{Séance}))$
 - Optimisée: $\pi_{\text{réalisateur}} (\text{Film} \bowtie \sigma_{\text{heure-début} > 14} (\text{Séance}))$

- **Hypothèses**

- Film occupe 8 lignes
 - Séance occupe 50 lignes, 90% des séances sont après 14h et 20 % des séances concernent des films

Un exemple de réécriture algébrique qui échoue (PEL)

- **Plan initial:** $\pi_{\text{réalisateur}} (\sigma_{\text{heure-début} > 14} (\text{Film} \bowtie \text{Séance}))$

- Jointure: on lit $8 * 50 = 400$ lignes et on produit $20\% * 50 = 10$ lignes
- Sélection: on produit $90\% * 10 = 9$ lignes de séances après 14h
- On laisse de côté la projection (même coût dans les deux cas)

➡ **Coût (E/S): $400E + 10S + 10E + 9S = 429$ lignes E/S**

- **Plan optimisé:** $\pi_{\text{réalisateur}} (\text{Film} \bowtie \sigma_{\text{heure-début} > 14} (\text{Séance}))$

- Sélection: on lit 50 lignes et on produit $90\% * 50 = 45$ lignes de séances
- Jointure: on lit $8 * 45 = 360$ lignes et on produit $20\% * 45 = 9$ lignes

➡ **Coût (E/S): $50E + 45S + 360E + 9S = 464$ lignes E/S**

➡ **D'après la fonction de coût utilisée, le meilleur plan est le plan initial !**
Cas rare: ici la jointure est plus sélective que la sélection

Conclusion sur la réécriture algébrique (PEL)

- **La réécriture algébrique est nécessaire, mais pas suffisante**
- **Il faut tenir compte d'autres critères:**
 - **Les chemins d'accès aux données (selon l'organisation physique)**
 - On peut accéder aux données d'une table par accès séquentiel, par index, par hachage, etc.
 - **Les différents algorithmes possibles pour réaliser un opérateur**
 - Il existe par exemple plusieurs algorithmes pour la jointure
 - Souvent ces algorithmes dépendent des chemins d'accès disponibles
 - **Les propriétés statistiques de la base de données**
 - Taille des tables
 - Sélectivité des attributs
 - etc.

Le rôle de l'optimiseur

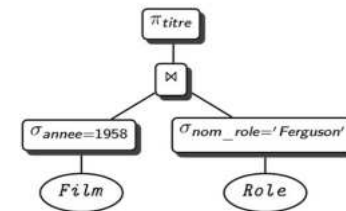
Trouver les expressions
équivalentes

Requête SQL



Plan d'exécution logique - PEL (l'algèbre)

```
select titre
from Film f, Role r
where nom_role = 'Ferguson'
and f.id = r.id_ilm
and f.annee = 1958
```



Choisir le bon algorithme
pour chaque opération

Plan d'exécution physique - PEP (opérateurs)

Un opérateur = une opération

Plusieurs algorithmes par opération

