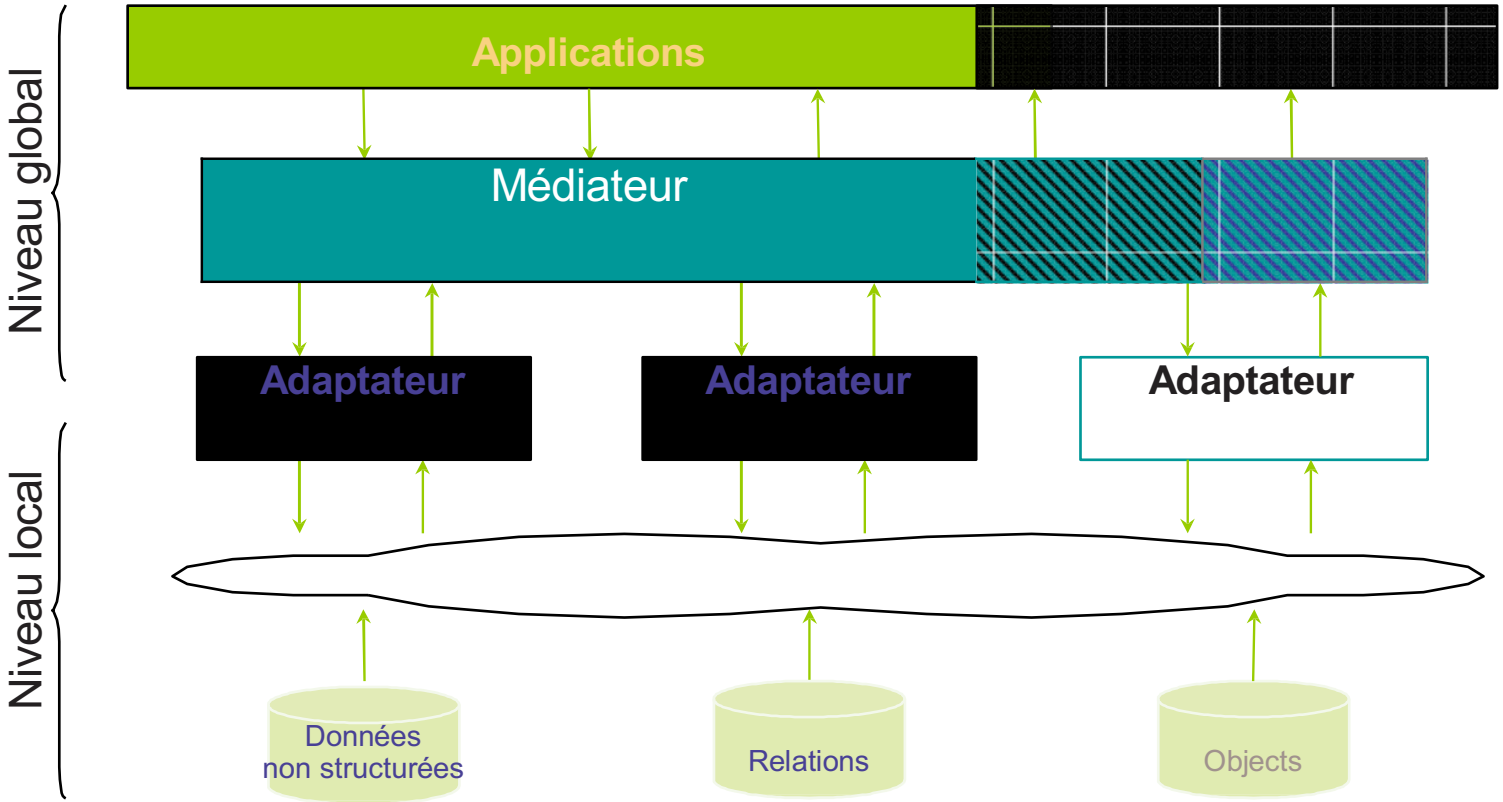
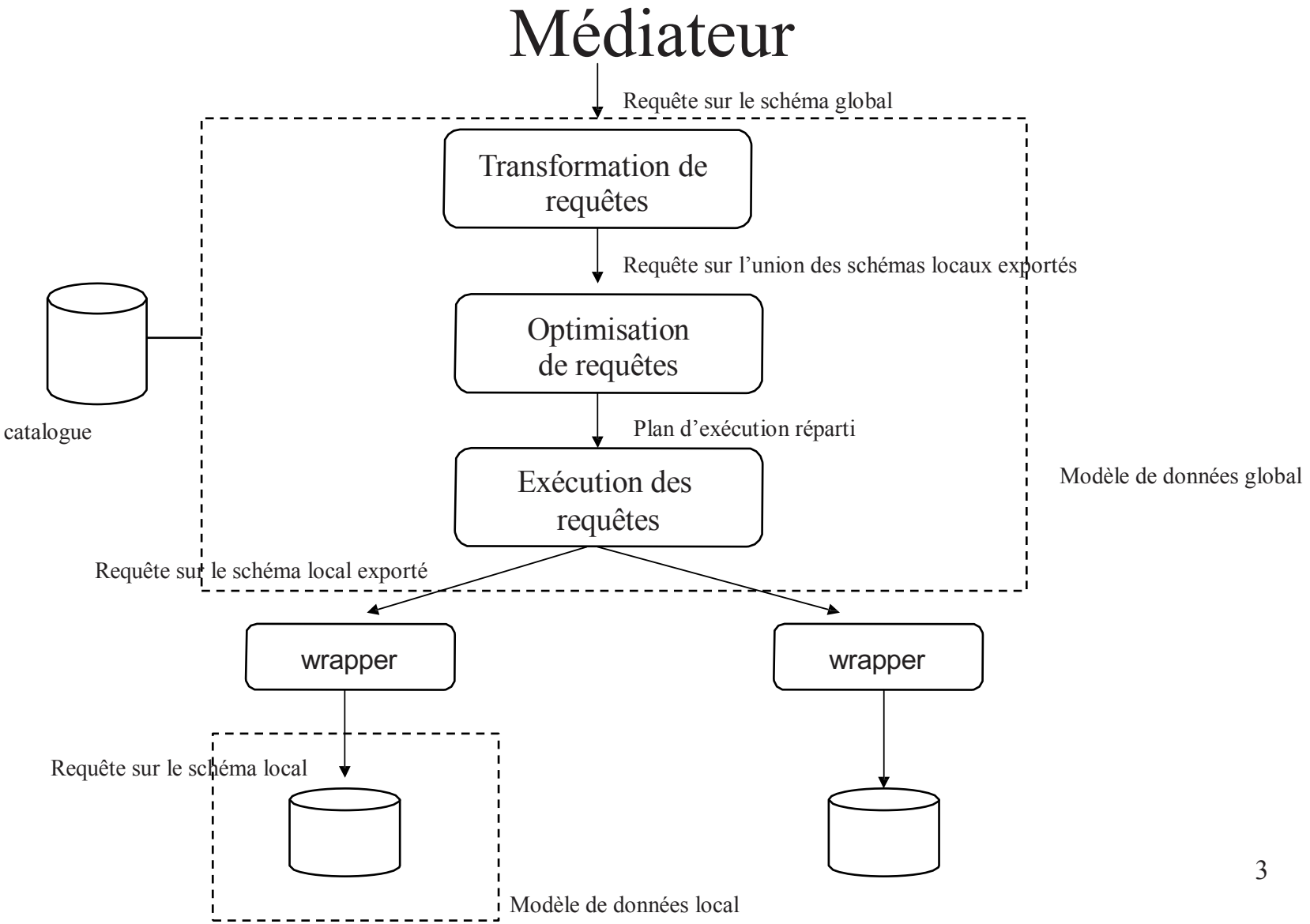
Médiateurs

Médiateurs


Niveau global
Niveau local
Applications
Médiateur
Adaptateur
Adaptateur
Adaptateur
Données
non structurées
Relations
Objects


Médiateur
Requête sur le schéma global
Transformation de requêtes
Requête sur l'union des schémas locaux exportés
Optimisation de requêtes
catalogue
Plan d'exécution réparti
Exécution des requêtes
Modèle de données global
Requête sur le schéma local exporté
wrapper
wrapper
Requête sur le schéma local
Modèle de données local

Médiateur

Le médiateur s'occupe de la répartition des sources :

- Localisation des sources
- Accepte les requêtes des clients
- Réécrit (décompose) et optimise les requêtes (optimisation répartie)
- Envoie les plans d'exécution à faire exécuter par les wrappers des différentes sources
- Combine (recompose) les résultats des wrappers et effectue éventuellement quelques opérations supplémentaires

Attention : le médiateur ne comprend pas de code spécifique aux sources!

Catalogue

Le catalogue du médiateur comprend toutes les méta-informations :

- le schéma global,
- les schémas externes des sources tels qu'ils sont exportés,
- les propriétés physiques des sources et du réseau,
- des statistiques sur les données,
- la fiabilité des sources,
- des éléments de description des sources : contenu, contraintes, complétude (des informations), fréquence des mises à jour, etc., qui permettent d'aider à la reformulation des requêtes (garantie du contenu, mieux cibler la source de données).

Sources de données

- Une source de données peut être décrite par
  - Localisation : référence, protocole de communication, moyen d'accès (JDBC, ODBC, API), support (SGBD, pages Web)
  - Type de données qu'elle gère : structuré (relationnel, objet), semi-structuré (XML, OEM), non-structuré (image, multimédia)
  - Possibilité d'interrogation : SQL, OQL, moteur de recherche
  - Format des résultats : XML, HTML, relations, textes

Description des sources

- Pouvoir d'expression : pouvoir faire la distinction entre les sources ayant des données semblables, de façon à éviter d'accéder à des sources non pertinentes.
- Faciliter l'ajout de nouvelles informations sur les sources.
- Etre capable de retrouver facilement les sources pertinentes pour une requête donnée : reformuler la requête de façon à obtenir des garanties sur les sources auxquelles on accède : réponse efficace et correcte.

Réécriture des requêtes

Etant donné

- une requête R sur le schéma global de médiation
- la description des données sources

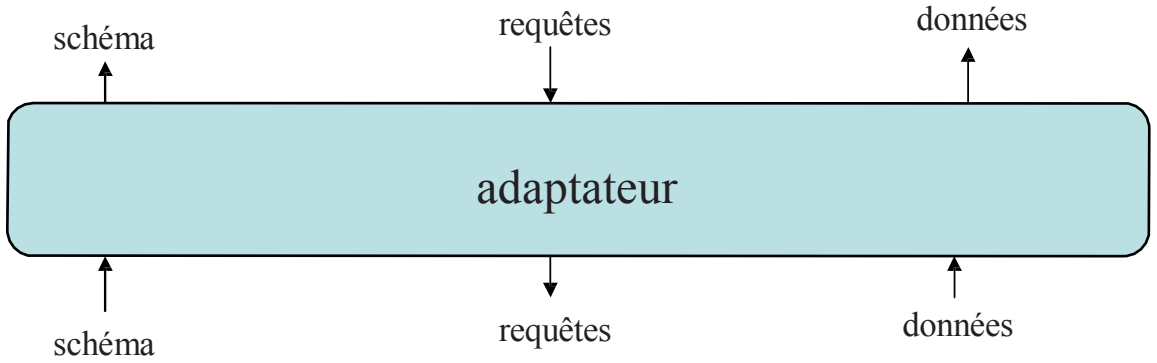
Trouver une requête R' utilisant seulement les données des relations sources, telle que

- la réponse est correcte et
- R' fournit toutes les réponses possibles à R

Assurer la correction et l'équivalence entre la requête et sa reformulation

Adaptateur (Wrapper)

- Cache l'hétérogénéité au médiateur



- Traduit le schéma des sources en termes du schéma global
- Traduit les requêtes du médiateur en termes compréhensibles par les sources
- Traduit les résultats renvoyés par la source en termes du schéma global
- Un adaptateur par source (peut constituer un obstacle à l'intégration d'un nombre important de sources)
- Assez difficiles à écrire
- Peuvent être « intelligents » : effectuer des optimisations spécifiques aux sources
- Sont généralement associés aux sources, mais peuvent aussi se trouver dans le médiateur

Exemple

Transformer :

```
<b>Data on the Web</b>
<i>Abiteboul S.</i>
<i>Buneman P</i>
<i>Suciu D.</i>
Morgan Kaufman, 1999
```

En :

```
<livre>
<titre>Data on the Web</titre>
<auteur>Abiteboul S.</auteur>
<auteur>Buneman P</auteur>
<auteur>Suciu D.</auteur>
<editeur>Morgan Kaufman </editeur>
<annee> 1999</annee>
</livre>
```

Communication médiateur/adaptateur

- Pour faciliter le travail d'intégration, on définit
  - Un langage commun dans lequel le médiateur interrogera les adaptateurs
  - Un format de résultat commun dans lequel les adaptateurs répondront au médiateur
- Le langage et le format du résultat peuvent être standardisés ou propriétaires

Schéma global

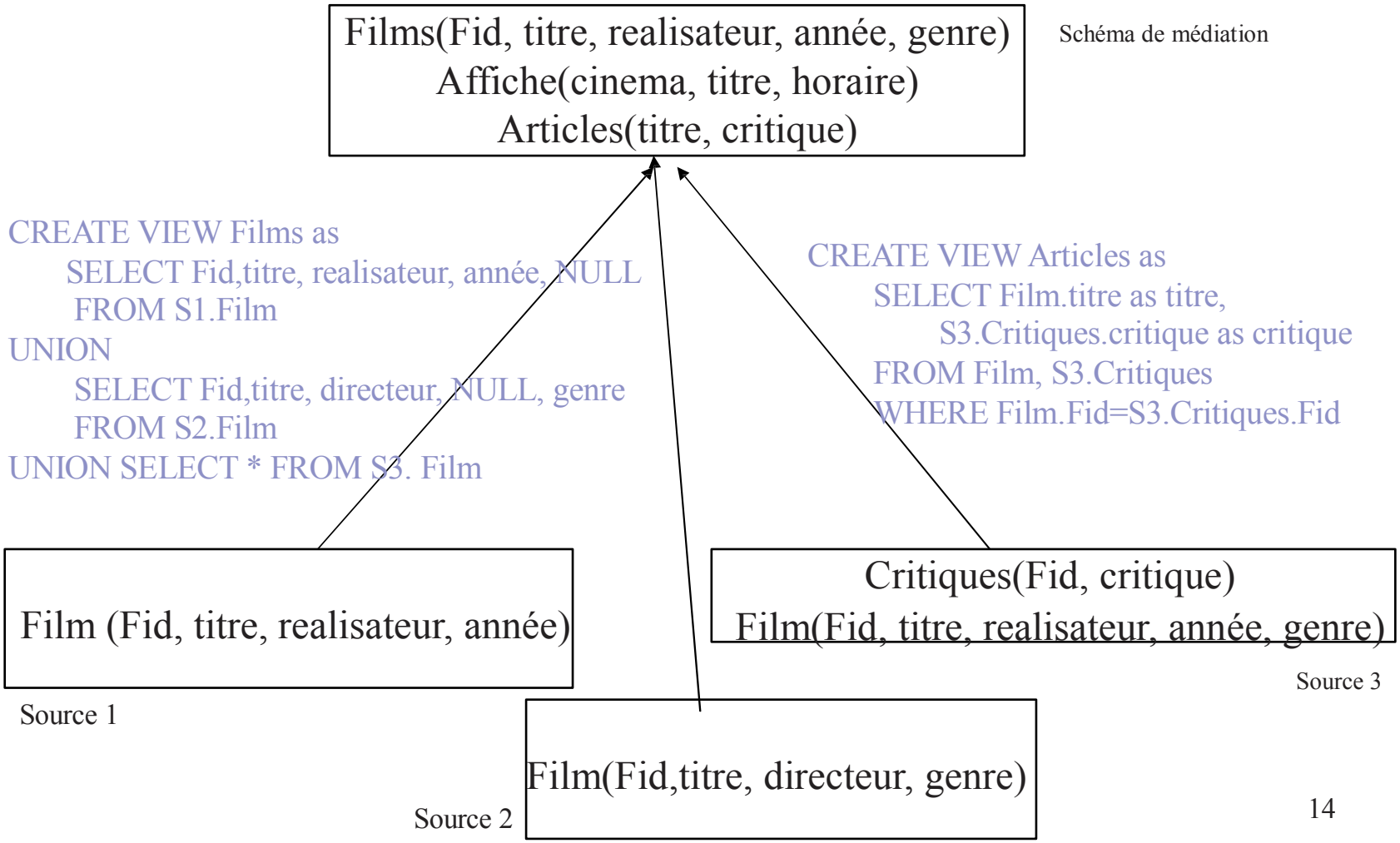
- Choix d'un modèle commun (et du langage de requêtes)
  - Relationnel (Information Manifold, Le Select,XPeranto),
  - orienté-objet (Garlic, Disco),
  - semi-structuré (Tsimmis, Yat, Nimble, Xylème),
  - Logiques de description (SIMS, Observer)
- Plusieurs approches pour définir le schéma global :
  - Global as View (Tsimmis, Disco, Yat, Garlic, XPeranto)
  - Local as View (AGORA, Information Manifold)
  - Combinaison des deux ?
- Ces approches sont déterminantes pour la réécriture des requêtes, et pour l'évolution du système d'intégration (ajout de sources)

Global as view

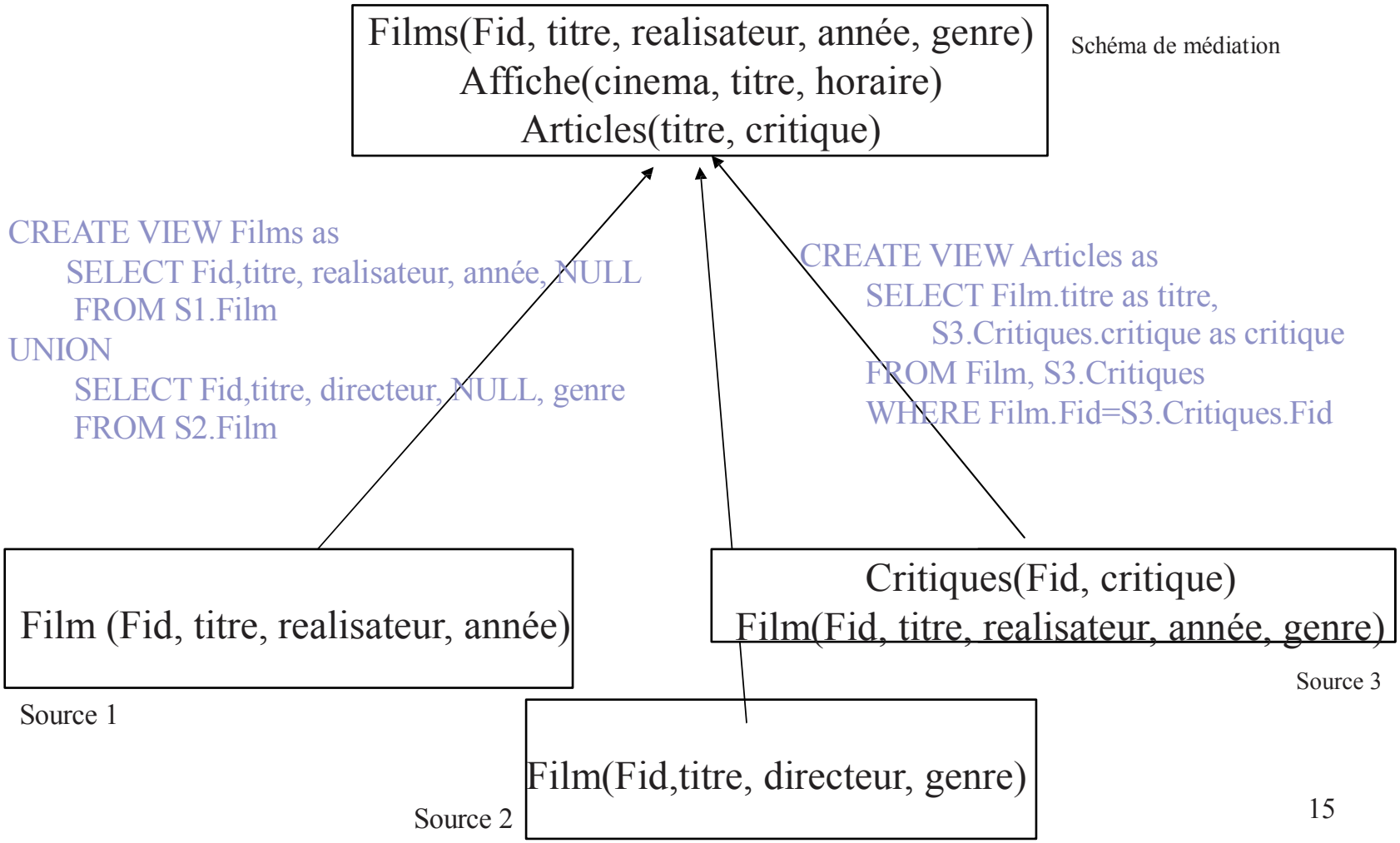
- Définir le schéma global en fonction du schéma des sources à intégrer : les relations du schéma global sont définies comme des vues sur les relations sources.
- Approche ascendante depuis les sources vers le médiateur
- Les données restent dans les sources.
- Une requête sur le schéma global se traduit en termes de schémas de sources en remplaçant les vues par leurs définitions (dépliement des requêtes). La requête dépliée est évaluée sur les sources. Il peut y avoir des redondances.

Hypothèse : les sources sont connues à l'avance

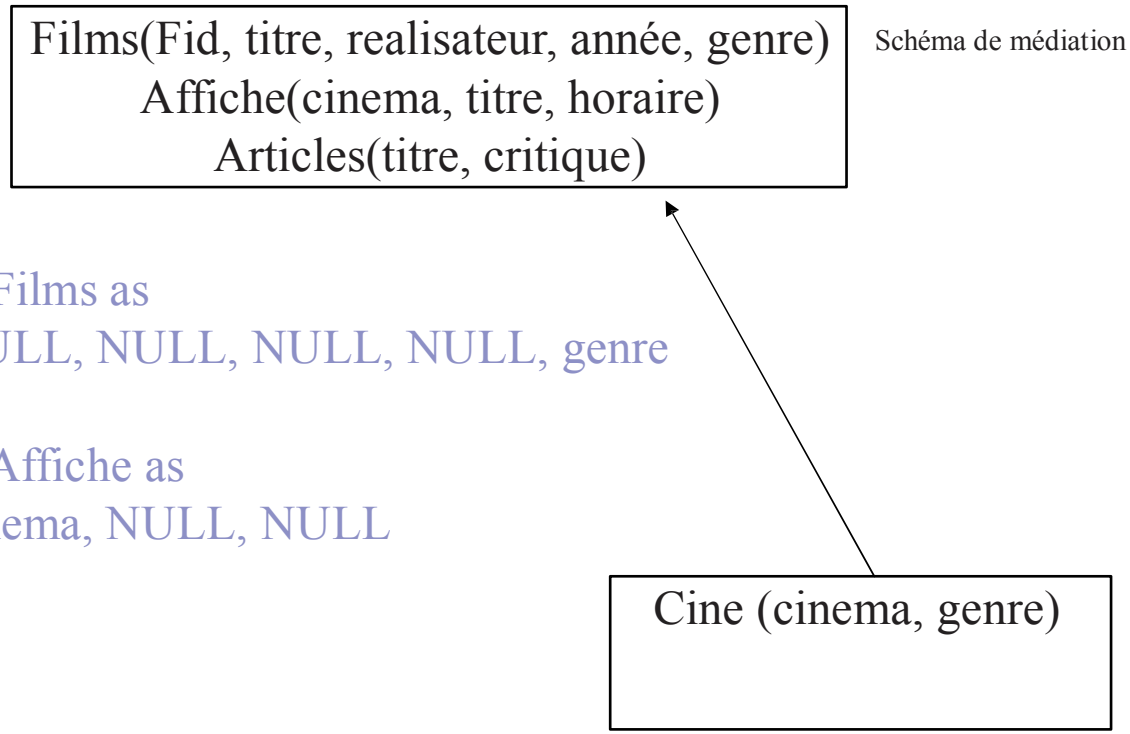
Global as view



Global as view (2ème exemple)



Global as view (3ème exemple)




```
CREATE VIEW Films as
      SELECT NULL, NULL, NULL, NULL, genre
      FROM S4
CREATE VIEW Affiche as
      SELECT cinema, NULL, NULL
      FROM S4
```

Quels sont les cinémas qui jouent des comédies ?

Global as View

- Conception facile
- La réécriture des requêtes est simple (dépliement des vues)
- Possibilité de construire des hiérarchies de schémas de médiation
- Risque de perdre de l'information
- L'ajout de sources est difficile, et demande de prendre en compte l'ensemble des sources disponibles. Passe mal à l'échelle.

Exemple de requête en GAV

Quelles sont les critiques des films réalisés par W. Allen ?

SELECT Films.titre, Critiques.critique

FROM Films, Articles

WHERE Films.Realisateur = 'W. Allen' AND Films.titre=Articles.titre

Traduction de la requête :

SELECT S1.Film.titre, S2.Film.titre, S3.Critiques.critique

FROM S1.Film, S2.Film, S3.Critiques

WHERE (S1.Realisateur='W.Allen' AND S1.Film.Fid=S3.Critiques.Fid)

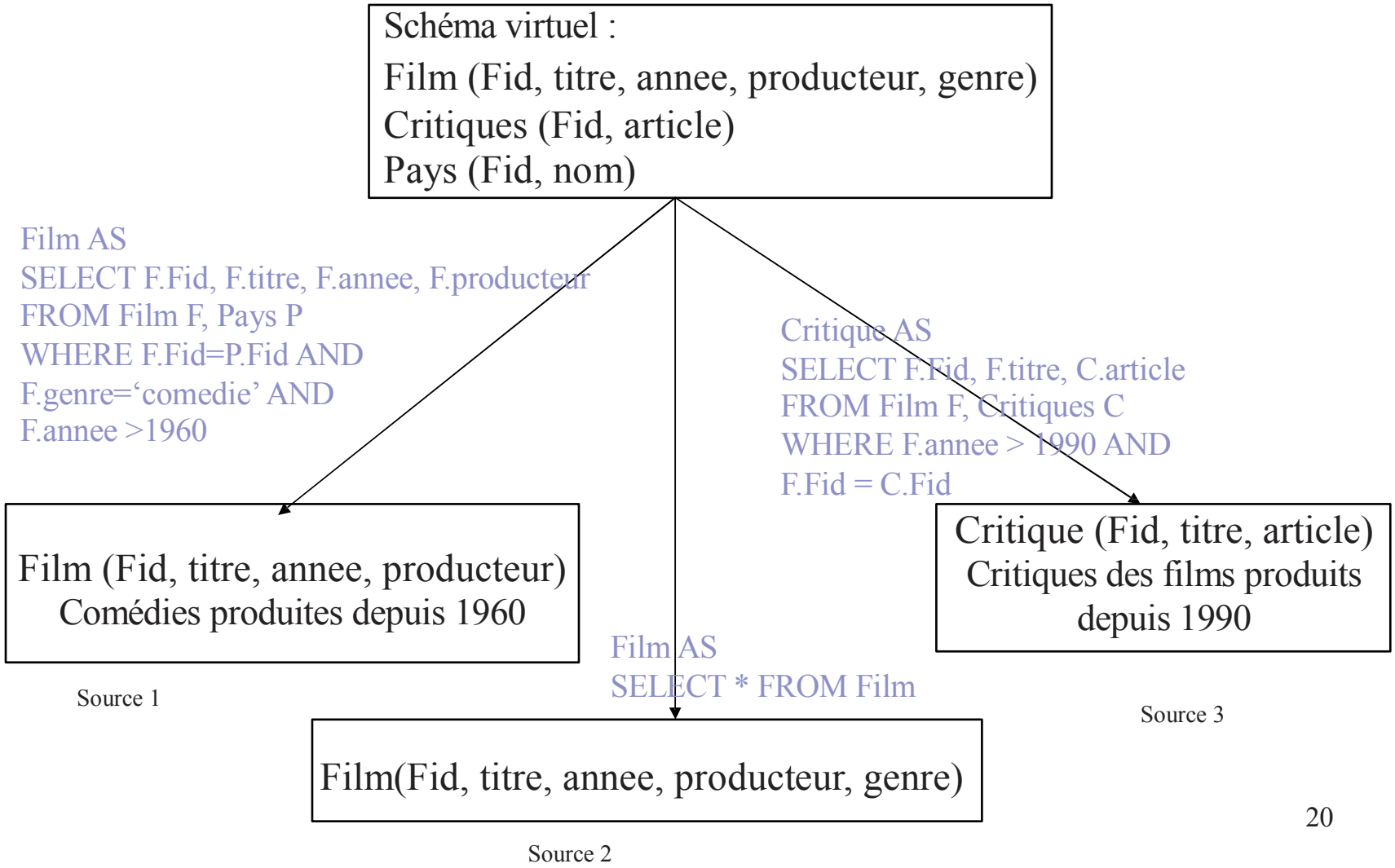
OR

(S2.Directeur='W.Allen' AND S2.Film.Fid=S3.Critiques.Fid)

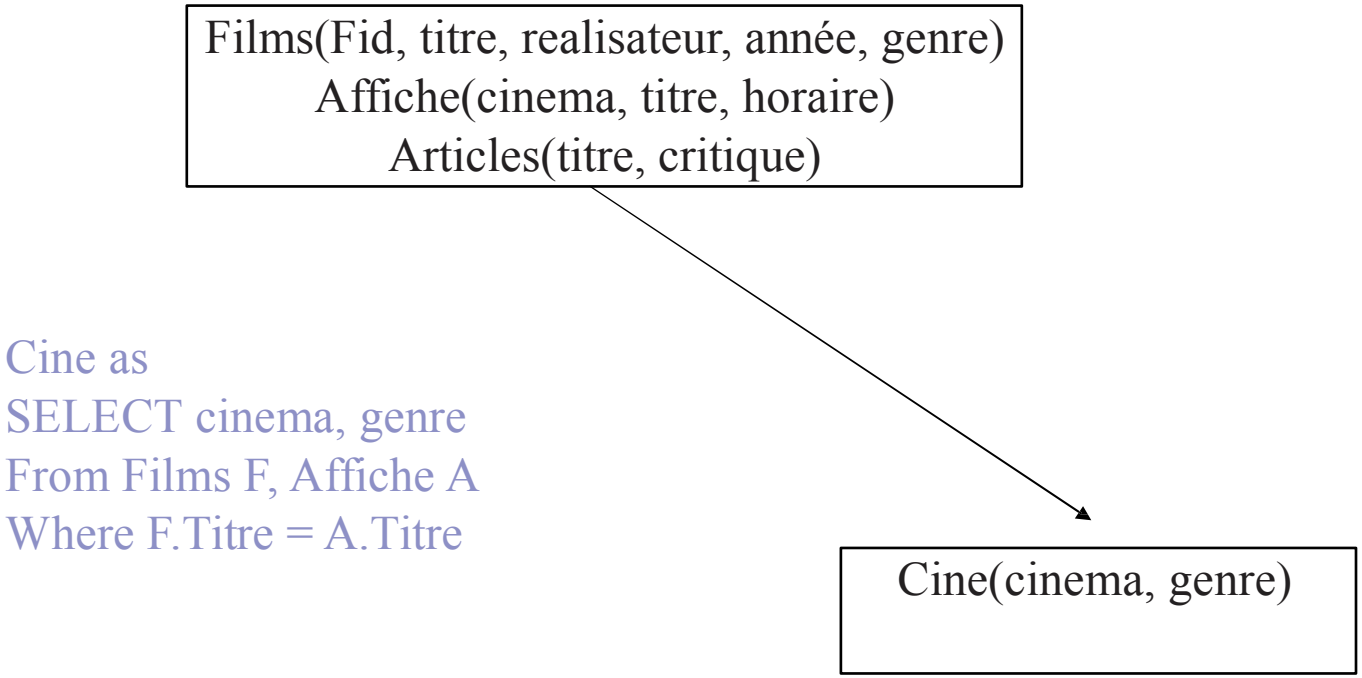
Local as View (LAV)

- Définir les schémas locaux en fonction du schéma global : les relations des schémas locaux (sources) sont définies comme des vues (requêtes) sur le schéma global.
- Les données restent dans les sources
- Une requête sur le schéma global doit être traduite en termes des schémas locaux (réécriture des requêtes)

Local as view



Local as View (Exemple 2)



Quels sont les cinémas qui jouent des comédies ?

Exemple de requête en LAV

Quelles sont les critiques des comédies depuis 1950 ?

SELECT Films.titre, Critiques.article

FROM Film, Critiques

WHERE Film.Fid=Critiques.Fid AND Film.genre='comedie' AND Film.annee >=1950

Traduction de la requête :

SELECT S1.Film.titre, S3.Critiques.article

FROM S1.Film, S3.Critiques

WHERE S1.Film.Fid=S3.Critiques.Fid

Pb: la requête reformulée n'est pas équivalente à la requête originale, car elle renvoie seulement les critiques des films depuis 1990.

GaV - LaV

Global as View :

+ la réécriture d'une requête est simple : dépliage

+ Conception naturelle et intuitive

-Evolution difficile (l'ajout d'une source est compliqué et nécessite de reconstruire le schéma global)

- nombre de sources limité (passage à l'échelle difficile)

Local as View :

+ très souple. On dispose de toute la puissance du langage de requêtes pour définir le contenu des sources.

+ chaque source est décrite de façon isolée

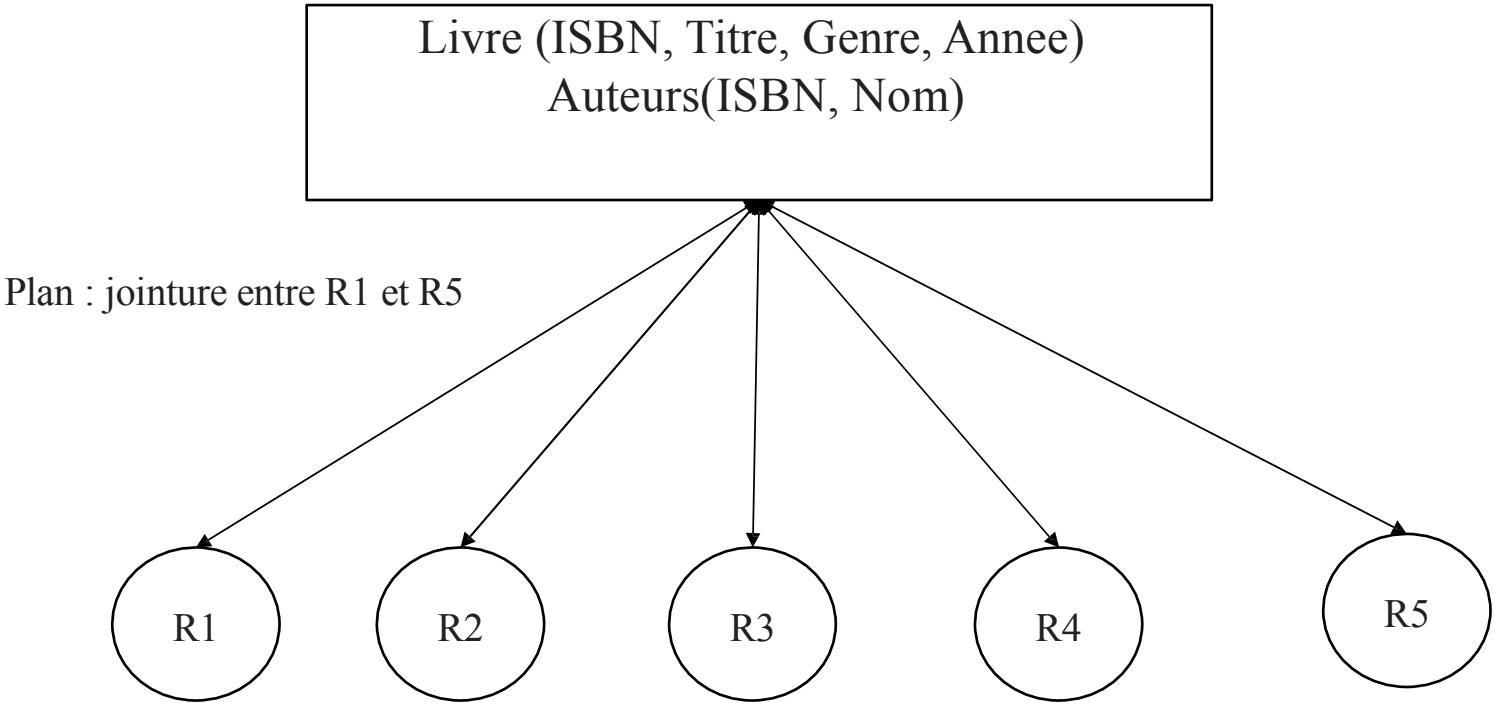
+ evolution facile : ajouter une source = écriture d'une requête, pas d'effet sur le schéma global

+ le contenu des sources est bien spécifié (spécifier des conditions dans la requête)

- la reformulation des requêtes est complexe (réécriture en termes de vues)

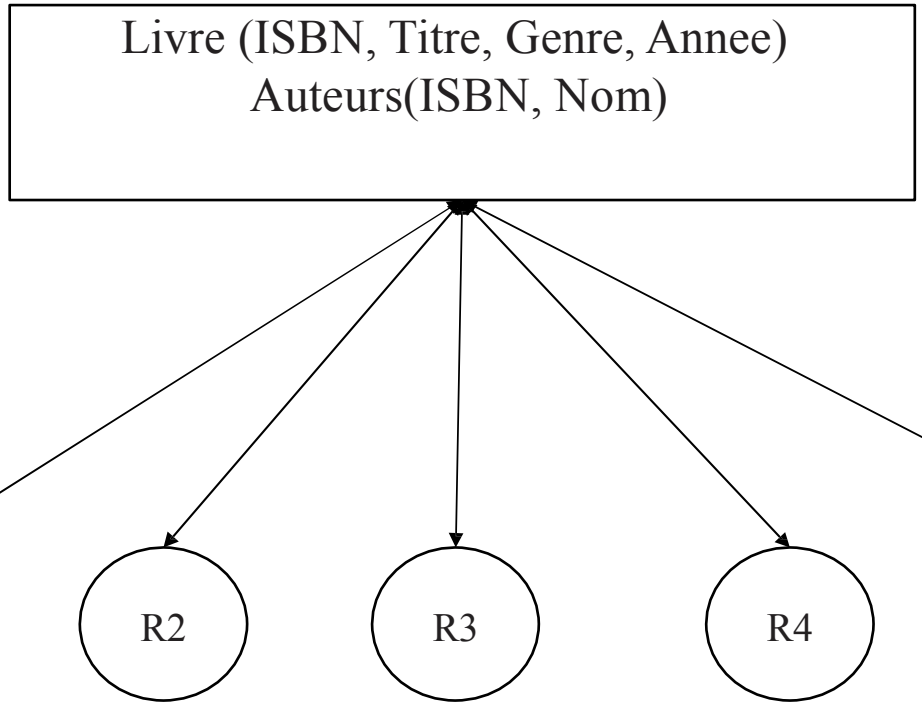
Reformulation des requêtes

Quels sont les auteurs de livres d'humour ?



Reformulation des requêtes

Quels sont les auteurs de livres d'humour publiés avant 1960 ?



Plan : ?

R1

R2

R3

R4

R5

R1(ISBN, Titre, nom)

Livres publiés avant 1970

R5(ISBN, titre)

Livres d'humour

Reformulation des requêtes

- Etant donné un ensemble de vues et une requête R, peut-on répondre à R en utilisant uniquement les réponses des vues ?
- Algorithmes de réécriture
- Complétude des sources
- Optimisation de requêtes

Traduction et réécriture de requêtes

- Pb : le langage de requête du schéma global n'est pas toujours le même que celui des cibles. Les requêtes globales doivent être traduites dans le langage des sources
  - (ex : Xquery -> SQL)
- Dépend de la transformation du modèle sous-jacent
- Doit prendre en compte la puissance d'expression du langage cible (peut nécessiter des extensions)
  - Ex : comment représenter en SQL les expressions de chemins avec récursion des langages de requêtes pour XML ?
- La réécriture des requêtes en termes des schémas locaux doit prendre en compte l'hétérogénéité structurelle ET sémantique des schémas.

Algorithme de réécriture

Les requêtes et les sources sont écrites sous forme de requêtes conjonctives. A priori, le nombre de réécritures possibles de la requête en fonction des vues est exponentiel par rapport à la taille de la requête.

Principe des algorithmes : réduire le nombre de possibilités en considérant les sous-requêtes (buts partiels) et en déterminant quelles vues sont pertinentes pour répondre à ces sous-requêtes.

Conceptuellement : on dispose d'un ensemble de requêtes précalculées (les sources) et on souhaite les utiliser pour répondre aux requêtes.

Ex. d'algorithmes : Bucket, inverse-rules, MiniCon

Description des sources

- Eléments de description :
  - Contenu : une source contient des films, leurs réalisateurs, etc.
  - Contraintes : les films répertoriés sur cette source ont été produits après 1965
  - Complétude : la source contient tous les films français
  - Capacités : les requêtes sur cette source doivent avoir une forme particulière, contenir au moins l'attribut titre, possibilité de faire des sélections, impossibilité de faire un parcours séquentiel complet, etc.

Complétude locale

En principe, il est nécessaire d'interroger toutes les sources pertinentes. En exploitant la complétude locale, on peut éviter des accès inutiles aux sources.

Pb : les sources ne sont pas toutes complètes, ou au contraire, elles ont des données qui se recouvrent.

Film (Fid, titre, réalisateur, année) sur la source S1 est complète pour année > 1960

Film(Fid, titre, annee, producteur, genre) est complète.

Peut-on garantir qu'une réponse est complète étant donnée la complétude locale des sources ?

Exemple1: Réponse incomplète

Film(titre, réalisateur, année) (incomplète avant 1960)

Salle(titre, cinéma, ville, heure)

R1 : Quels sont les films (et leurs réalisateurs) qu'on peut voir à Paris ?

SELECT F.titre, F. realisateur

FROM FILM F, SALLE S

WHERE F.titre=S.Titre AND ville='Paris'

Des modifications dans la relation Film peuvent modifier la réponse.

Exemple 2: réponse complète

Film(titre, réalisateur, année)

Oscar(titre, année)

R2: *Quels sont les réalisateurs qui ont eu des oscars depuis 1965 ?*

SELECT F. realisateur FROM Film F, oscar O

WHERE F. titre = O. titre AND

F. annee = O.annee AND

O. Annee >= 1965

Ce n'est pas parce qu'une source est incomplète que les réponses sont forcément incomplètes

Optimisation de requêtes

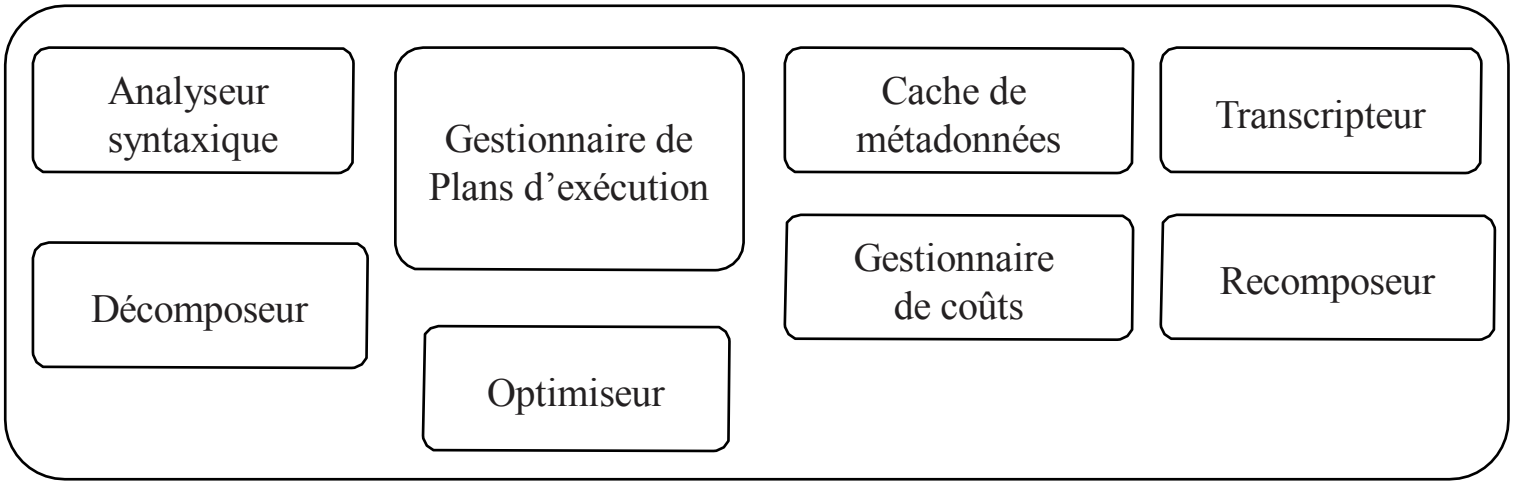
Objectif : traduire une requête déclarative en un programme impératif équivalent de coût minimal.

Le programme impératif est un plan d'exécution des requêtes : un arbre d'opérations dans une algèbre.

Notions de base en optimisation :

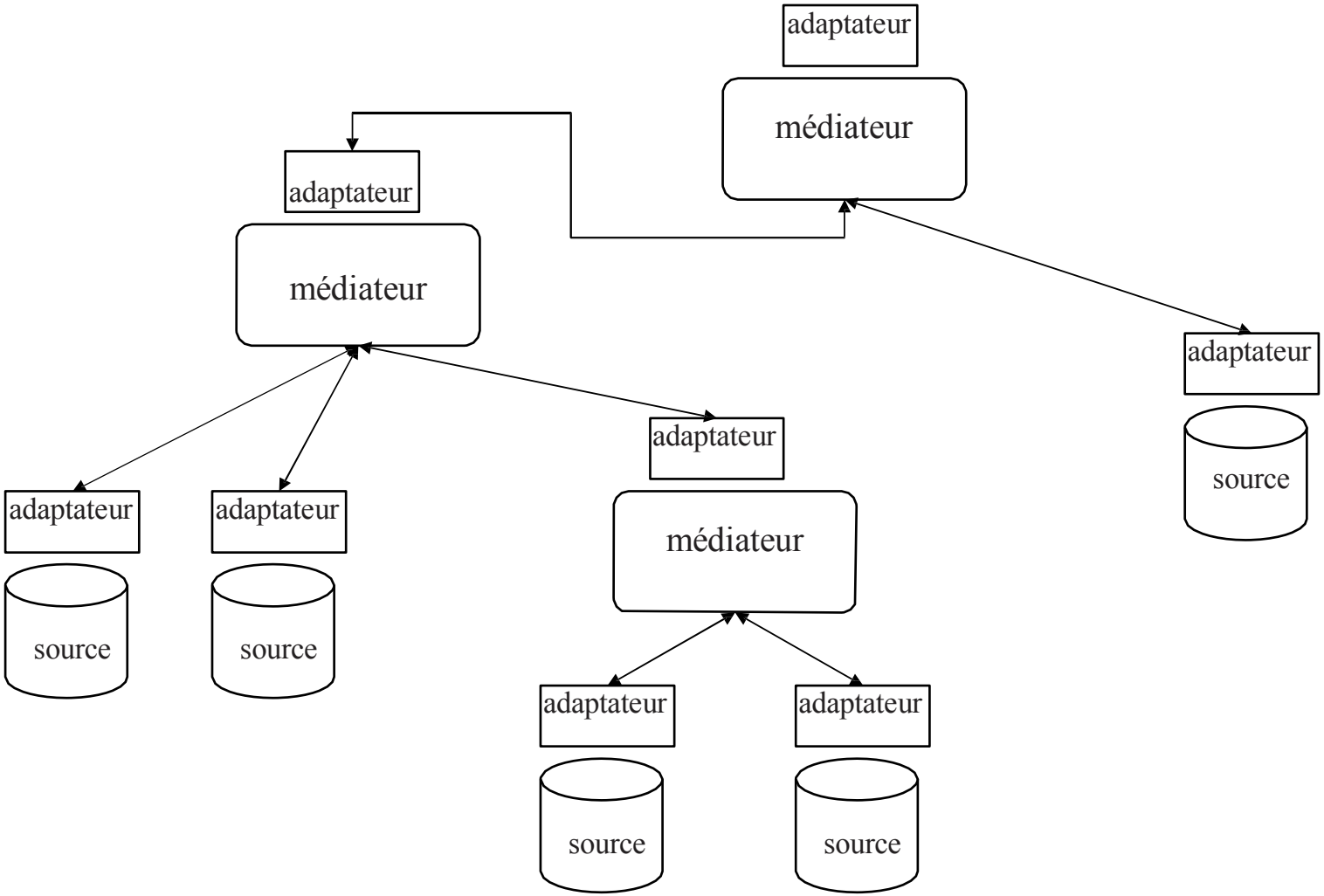
- modèle de coût
- espace de recherche
- stratégies de recherche

Architecture du médiateur



Analyseur : vérifie la validité et la syntaxe de la requête, la prépare pour le décomposeur.
Décomposeur : décompose la requête en sous-requêtes
Cache des métadonnées : conserve les schémas des sources, et la localisation des données
Gestionnaire de plans d'exécution : génère l'ensemble des plans pour une requête donnée
Gestionnaire de coûts : estime le coût d'exécution d'un plan
Optimiseur : détermine le plan optimal
Recomposeur : restructure et recompose les résultats des sources
Transcripteur : transforme la structure interne du résultat en un format lisible par l'utilisateur (XML)

Interconnexion de sources et médiateurs



Accès aux sources

- L'accès aux sources s'effectue via un adaptateur dédié, qui communique par XML/DBC.
- L'interface de communication permet :
  - Ouverture de session
  - Demande d'information sur les sources : informations sur les métadonnées, sur les capacités des sources, sur les formules et les statistiques des coût.
  - Exécution de requêtes XQuery
  - Récupération du résultat (en SAX ou DOM)
  - Fermeture de connexion

Capacités des sources

- Permet de savoir comment une source peut répondre à une requête.
- Chaque source possède une liste de règles de capacités, indiquant les opérations permises (ou interdites), sur chacun des objets.

Exemple :

Num.	Permission	Opération	Collection	Chemin	Op.	Collection2	chemin2
10	Allow	Scan					
20	Allow	Select	Personne				
30	Allow	Select	Voiture		=		
100	Allow	Select	Voiture	Age	<		
200	Allow	Project	Personne	Nom			
260	Allow	Project	Voiture				
261	Allow	Join	Personne	Id	=	Voiture	Id-conducteur
65635	deny						

Capacités des sources

- Les sources ont des capacités d'interrogation différentes.
- L'adaptateur peut pallier certaines déficiences des sources, et exécuter certaines opérations.
- Le médiateur fait le reste, càd, effectue les opérations qui n'ont pu être faites ni par la source, ni par l'adaptateur.

Plan d'exécution

- Normalisation
  - Suppression des affectations (clauses « let »)
- Canonisation
  - Désimbrication des requêtes imbriquées
- Atomisation
  - Séparation des collections
- Identification des sources
  - Identifier les sources gérant chaque collection
- Création du plan d'exécution
  - Transformation en un arbre algébrique
- Optimisation du plan d'exécution
  - Optimisation de l'arbre algébrique