



# Chapitre

## Bases de Données Réparties

### Partie 2

# Conception des BD Réparties

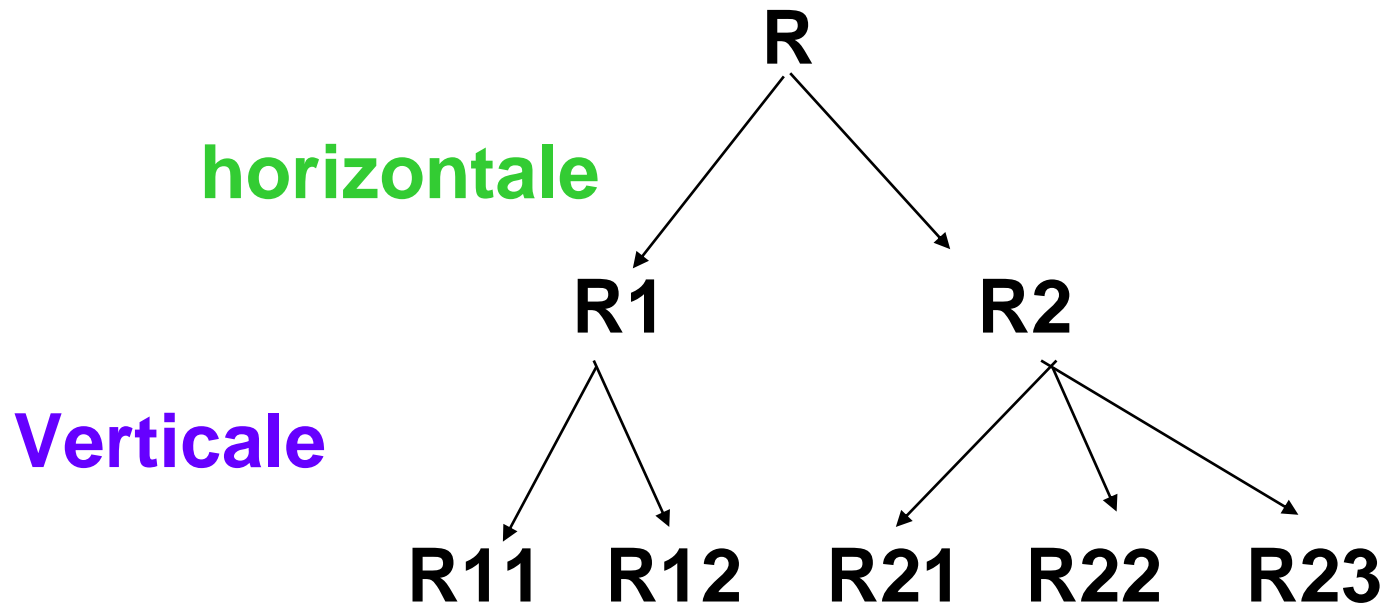
## ■ Objectifs :

- Maximiser les accès à un seul fragment.
- Minimiser les accès aux deux fragments.
  - $\text{Total RH} = \sum_{q_k \in \text{RH}} \sum_{\forall \text{site } s} \text{Ref}_s(q_k) * \text{Acc}_s(q_k) = 45.$
  - $\text{Total RB} = \sum_{q_k \in \text{RB}} \sum_{\forall \text{site } s} \text{Ref}_s(q_k) * \text{Acc}_s(q_k) = 75.$
  - $\text{Total Autres} = \sum_{q_k \in \text{autres}} \sum_{\forall \text{site } s} \text{Ref}_s(q_k) * \text{Acc}_s(q_k) = 8.$

# Conception des BD Réparties

## ■ Fragmentation Hybride

- Fragmentation horizontale suivit de la fragmentation verticale ou vice-versa



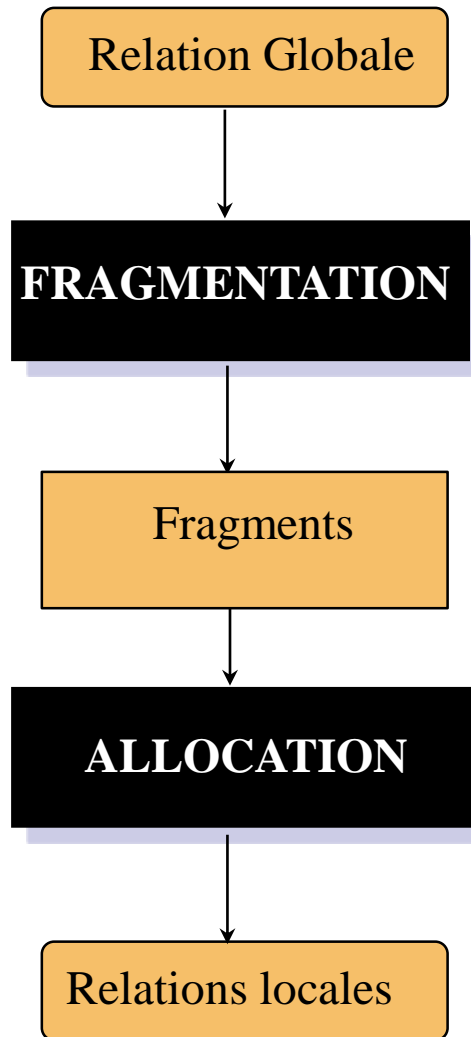
# Conception des BD Réparties

## ☐ Allocation

- Supposons qu'on dispose d'un ensemble de fragments  $F=\{F1, F2, \dots F_n\}$  et d'un réseau constitué d'un ensemble de sites  $S=\{S1, S2, \dots S_n\}$ .
- Une distribution optimale de  $F$  sur  $S$  est définie en considérant trois mesures:
  - ☐ Un coût minimal (où la fonction coût est une combinaison d'un ensemble de coûts:
    - Coût de stockage de chaque fragment  $F_i$  sur un site  $S_j$ .
    - Coût de modification de  $F_i$  sur un site  $S_j$ .
    - Coût d'interrogation de  $F_i$  sur un site  $S_j$ .
    - Coût de communication.
  - ☐ Une meilleure performance
  - ☐ Un temps de réponse minimum.

# Conception des BD Réparties

En Résumé



# Conception des BD Réparties

## Localisation des données

- Cette étape permet de traduire une requête en algèbre relationnelle exprimée par le schéma global de la base de données en une requête exprimée par les schémas locaux (l'ensemble des fragments)
- **Les requêtes utilisant des relations fragmentées horizontalement**

- **Exemple :** On considère que la relation employé (E) est fragmentée comme suit:

$$E_1 = \sigma_{\text{Enum} \leq 3}(E).$$

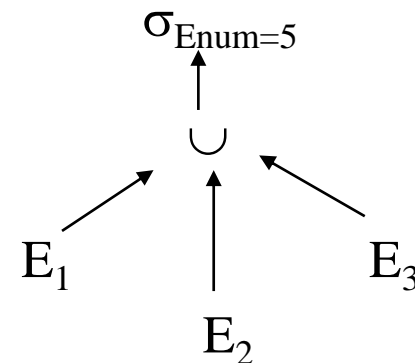
$$E_2 = \sigma_{3 < \text{Enum} < 6}(E).$$

$$E_3 = \sigma_{\text{Enum} > 6}(E).$$

- La relation E est obtenue par l'union  $E_1 \cup E_2 \cup E_3$

Soit la requête suivante:

```
SELECT *  
FROM   E  
WHERE  Enum=5.
```



# Conception des BD Réparties

## Localisation des données

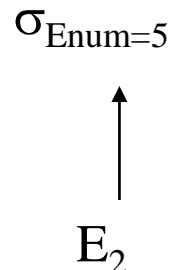
- Cette requête peut être simplifiée  $\Rightarrow$  requête réduite.
- Les règles de réduction permettent d'identifier les fragments générant des relations vides.
  - **Sélection:** on considère une relation  $R$  fragmentée horizontalement en  $R_1, R_2, \dots, R_n$  tel que  $R_j = \sigma_{P_j}(R)$ .

### Règle 1: Eliminer l'accès aux fragments inutiles

Exemple: on considère la requête:

SELECT \* FROM E WHERE Enum=5.

D'après la règle1, si on applique le prédicat de sélection « Enum=5 » Pour les fragments  $E_1$  et  $E_3$  on obtient des relation vides. Donc la requête est réduite comme suit



# Conception des BD Réparties

## Localisation des données

- **Jointure:** on peut détecter les jointures inutiles en se basant sur la règle suivante:

**Règle 2: distribuer les jointures par rapport aux unions et appliquer les réductions pour la fragmentation horizontale**

Exemple:

**pilote1 = pilote where adresse = Sousse;**

**pilote2 = pilote where adresse not=Sousse;**

**vol1 = vol where vol.numpil=pilote1.numpilote ;**

**vol2 = vol where vol.numpil=pilote2.numpilote ;**

**Requête : Select \* from pilote, vol**

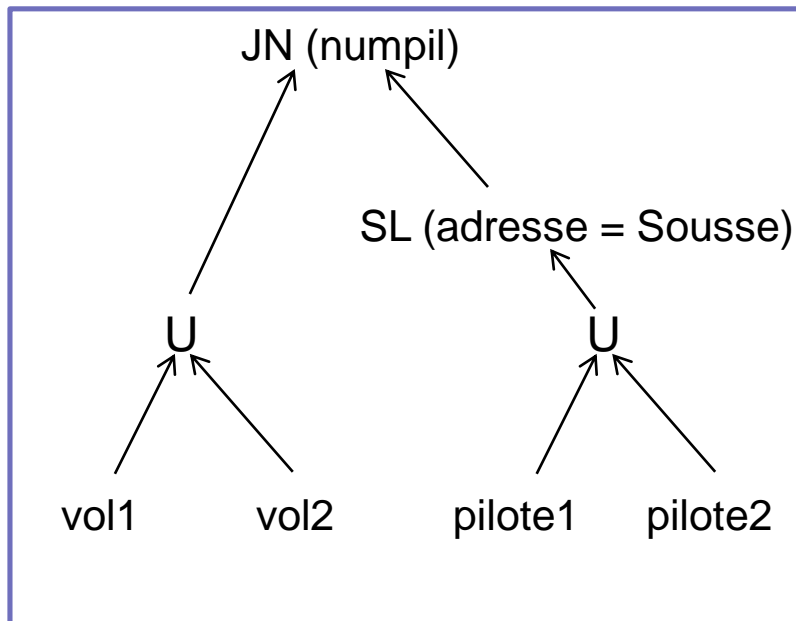
**where pilote.numpilote= vol.numpil and adresse=Sousse;**



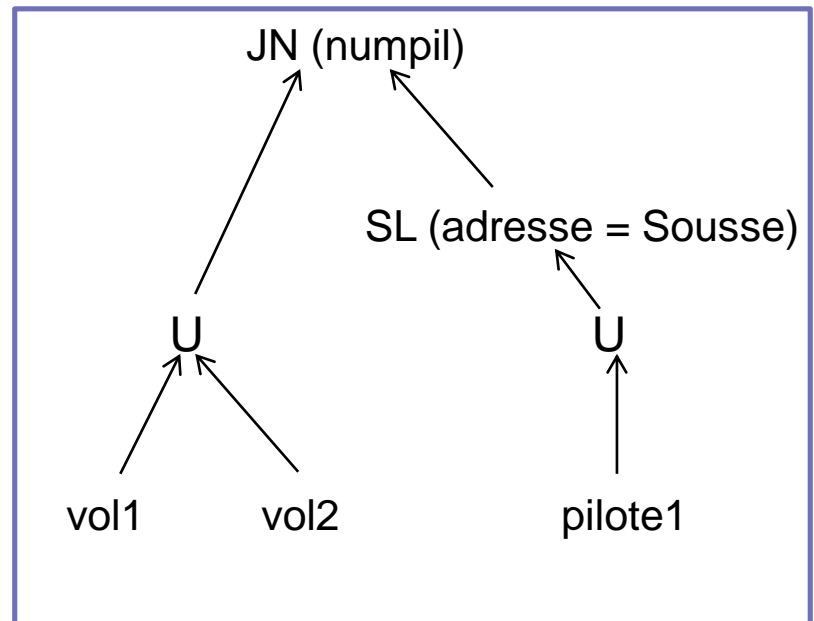
# Conception des BD Réparties

## Localisation des données

### Arbre Initial



### Première Réduction



**JN : Jointure**

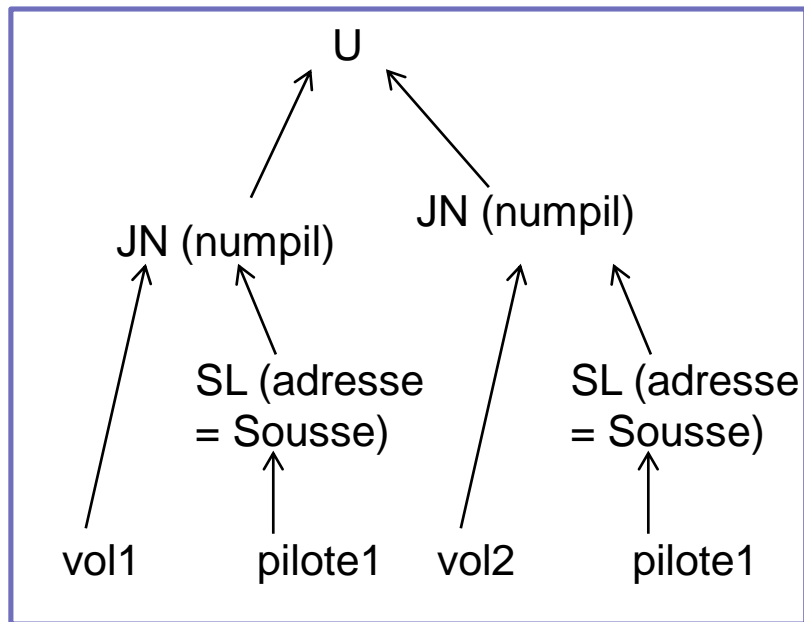
**SL : Sélection**

**Règle 1: Éliminer l'accès  
aux fragments inutiles**

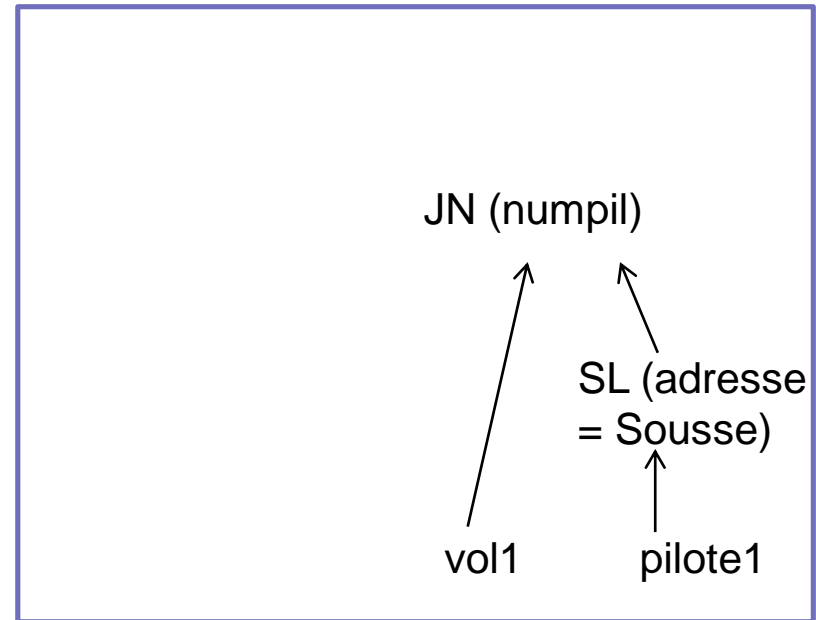
# Conception des BD Réparties

## Localisation des données

### Distribution de la jointure



### Élimination du sous arbre inutile



# Conception des BD Réparties

## Localisation des données

### ■ Les Requêtes utilisant des relations fragmentées verticalement

- On considère que la relation Employé E est fragmentée verticalement comme suit:

$$E_1 = \Pi_{\text{Enum}, \text{Enom}}(E)$$

$$E_2 = \Pi_{\text{Enum}, \text{Titre}}(E)$$

- La relation E est reconstruite par une opération de jointure  $E = E_1 \bowtie E_2$
- Les requêtes utilisant des relations fragmentées verticalement peuvent être réduites en éliminant les opérations inutiles. Pour ce faire, on utilise la règle suivante:

**Règle3: éliminer les accès aux relations de base qui n'ont pas d'attributs utiles pour le résultat final**

# Conception des BD Réparties

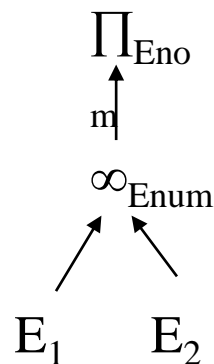
## Localisation des données

### Exemple 1:

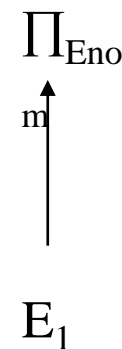
Soit la requête suivante:

```
SELECT Enom  
FROM E;
```

### Arbre Initial



### Arbre Réduit



# Conception des BD Réparties

## ■ Approche ascendante (Bottom-up)

### L'interopérabilité

#### □ Dans quel contexte ?

- Bases de données fédérées ou multi-bases
- Développement d'applications complexes : accès à des BD distantes
- Mise en œuvre d'un entrepôt de données

## ■ Objectifs :

- Accéder de manière transparente à plusieurs BD
- Permettre tout type de transaction
- Nécessité : **mécanismes d'intégration des sources d'informations**, langages de requêtes universels, mécanismes de gestion des transactions réparties

# Remarques :

## Architectures d'intégration

- Intégration matérialisée
  - Les données provenant des sources à intégrer sont stockées sur un support spécifique (entrepôt de données).
  - L'interrogation s'effectue comme sur une BD classique (relationnelle).
- Intégration virtuelle
  - Les données restent dans les sources
  - Les requêtes sont faites sur un schéma global, puis décomposées en sous-requêtes sur les sources. Les différents résultats des sources sont de la requête sont combinés pour former le résultat final.

# Conception des BD Réparties

## ■ Approche ascendante (Bottom-up)

### □ Problèmes :

- Concevoir le schéma global
- Offrir des outils permettant de traduire les opérations de base globale vers les bases locales

### □ Solutions :

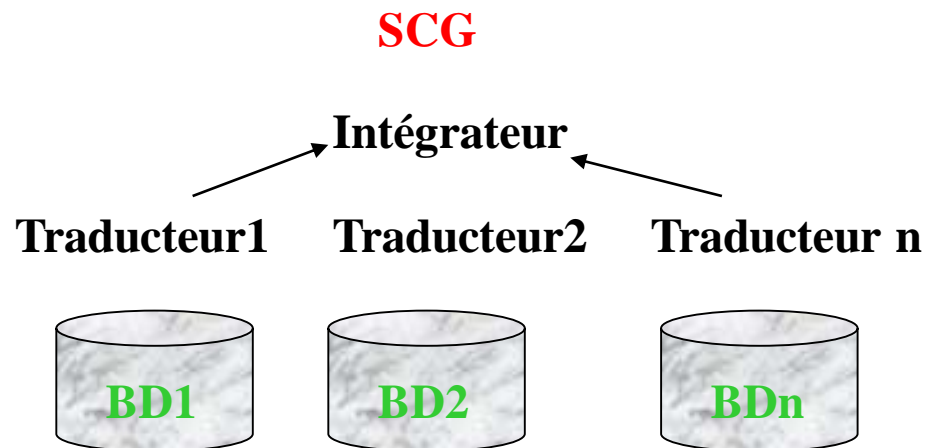
- **La médiation de schémas:** Approches et méthodes pour aboutir à un schéma global donnant une vue d'ensemble homogène des données stockées dans les différentes bases de données locales.
- Composants logiciels pour l'interopérabilité :
  - Adaptateur de sources (wrapper): Traduit les schémas du niveau local vers le niveau global et l'inverse
  - Médiateur: Offre une vision intégrée de l'ensemble des sources de données.
- Mécanismes étendus : évaluation et optimisation des requêtes, gestion des transactions

# Conception des BD Réparties

## ■ Approche ascendante (Bottom-up)

□ Les étapes de la médiation de schémas

1. Collecte des schémas
2. Traduction des schémas : en un formalisme commun
3. Intégration des schémas
4. Construction du schéma global





# Conception des BD Réparties

## ■ Collecte des schémas

- Description des schémas locaux : extraire la description des objets gérés par chacune des bases locales par la manipulation des méta-bases locales
- Identifier les règles de localisation des données présentes dans les bases locales

## ■ Traduction des schémas:

- Objectif : avoir un langage commun de description des données
- Résoudre le problème de différents modèles de données supportés par les SGBD
- Passer par un modèle pivot

## ■ Intégration des schémas

- Comparer les différents schémas
- Identifier les parties des schémas qui sont similaires
- Homogénéiser les schémas
- Choisir la meilleure représentation des concepts communs
- Produire des règles de traduction

## ■ Méthodes : binaire (les schémas locaux sont examinés et intégrés deux à deux) ou n-aire.

# Conception des BD Réparties

➡ L'intégration des bases de données peut être effectuée en trois étapes:

le traitement de l'hétérogénéité sémantique, la traduction des schémas et l'intégration des schémas.

## □ Traitement de l'hétérogénéité sémantique

- **Origine** : Résulte des conceptions indépendantes des différentes BD
- **Effet** : Désaccord sur la signification des données
- **Solution** : Analyse sémantique comparée des données préalable à la fédération

# Conception des BD Réparties

## □ Etape de Traduction

- Consiste à transformer le schéma local en un autre schéma.
- Exemple: transformer un schéma modèle réseau en un schéma modèle relationnel
- **Origine** : utilisation de modèles différents dans les BD composantes
- **Effet** : nécessite des traductions de tous les modèles
- **Solution** : traduction de tous les schémas dans un modèle commun (dit canonique ou pivot)
- Choix du modèle canonique :
  - Entité - Association et Relationnel
  - Objet

# Conception des BD Réparties

## □ Étape d'intégration

### ■ Pré-intégration

- Identification des éléments reliés (exple: domaines équivalents) et établissement des règles de conversion (exple 1 inch =2.54 cm).
- Comparaison
- Identification des conflits de noms (synonymes, homonymes) et des conflits structurels(clé, dépendances,...).

### ■ Conformance

- Résolution des conflits de noms (renommage) et des conflits structurels (changements des clés...)

### ■ Fusion et Restructuration

- Fusion des schémas intermédiaires et restructuration pour créer un schéma intégré optimal.

# Conception des BD Réparties

## □ Quelques cas de conflits

### ■ Conflits d'attributs

- Conflit de nom **Solution** : renommage
- Conflit de type **Solution** : conversion

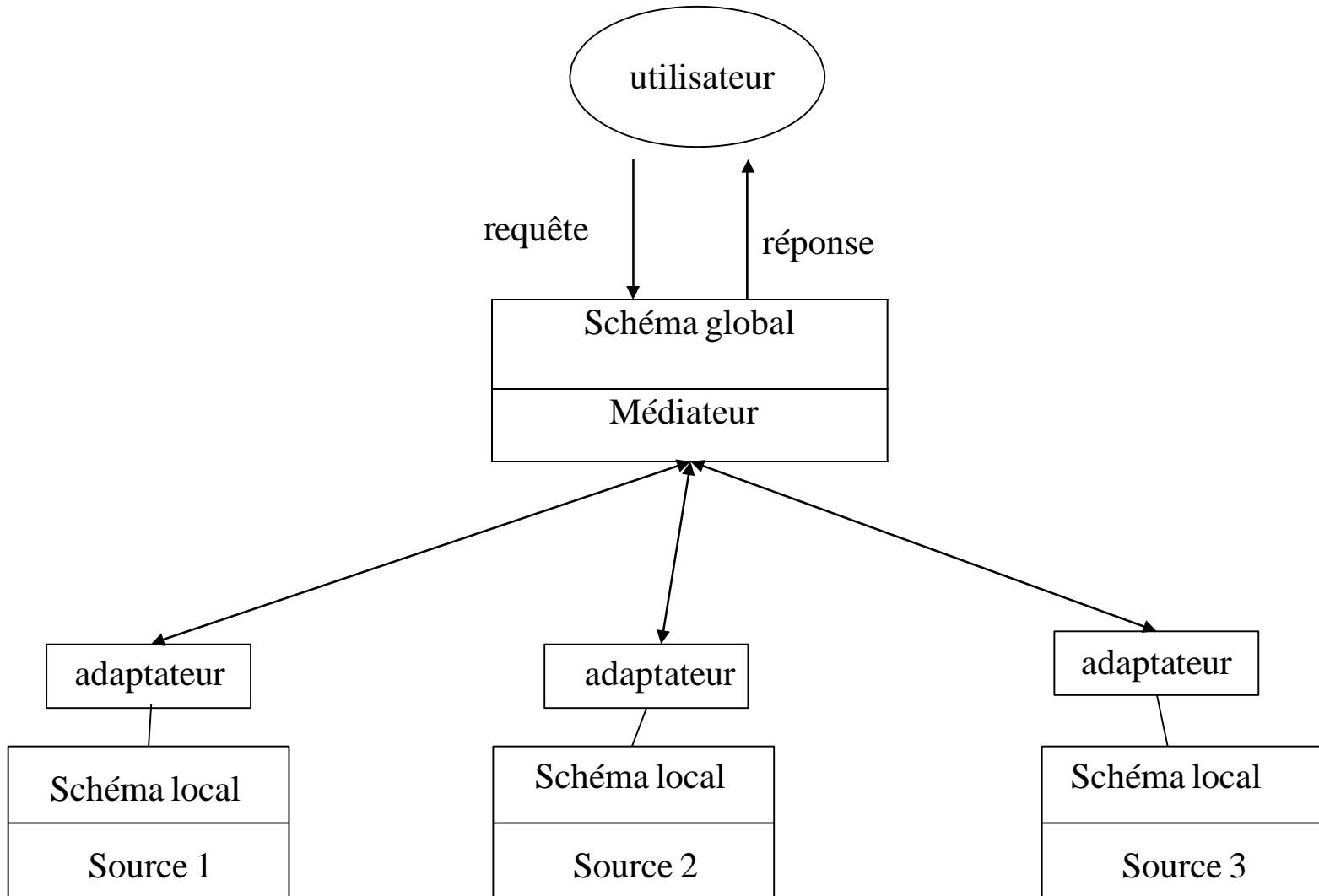
### ■ Attribut sans équivalent dans l'autre relation

- Attribut optionnel **Solution**: valeur nulle
- Attribut indispensable **Solution**: relation auxiliaire

### ■ Conflit de clé

- pas la même clé **Solution** : changement de clé
- la clé d'une des relations composantes n'est pas une clé générale **Solution**: génération d'une nouvelle clé par ajout d'un élément (ex. nom de commune pas déterminant au niveau national ajout du numéro de département au nom de la commune pour créer la nouvelle clé)

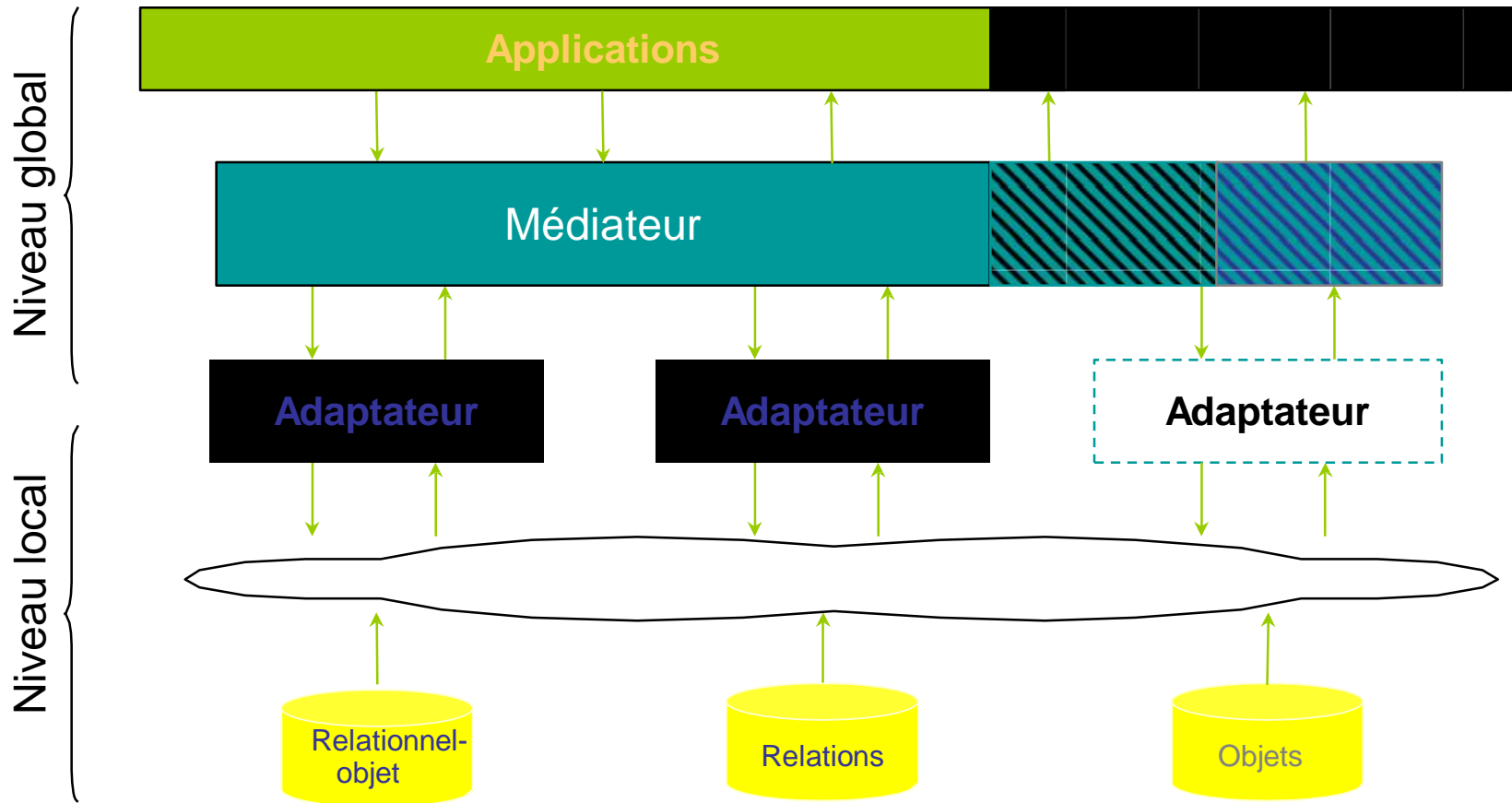
# Médiateur



# Médiateurs d'information

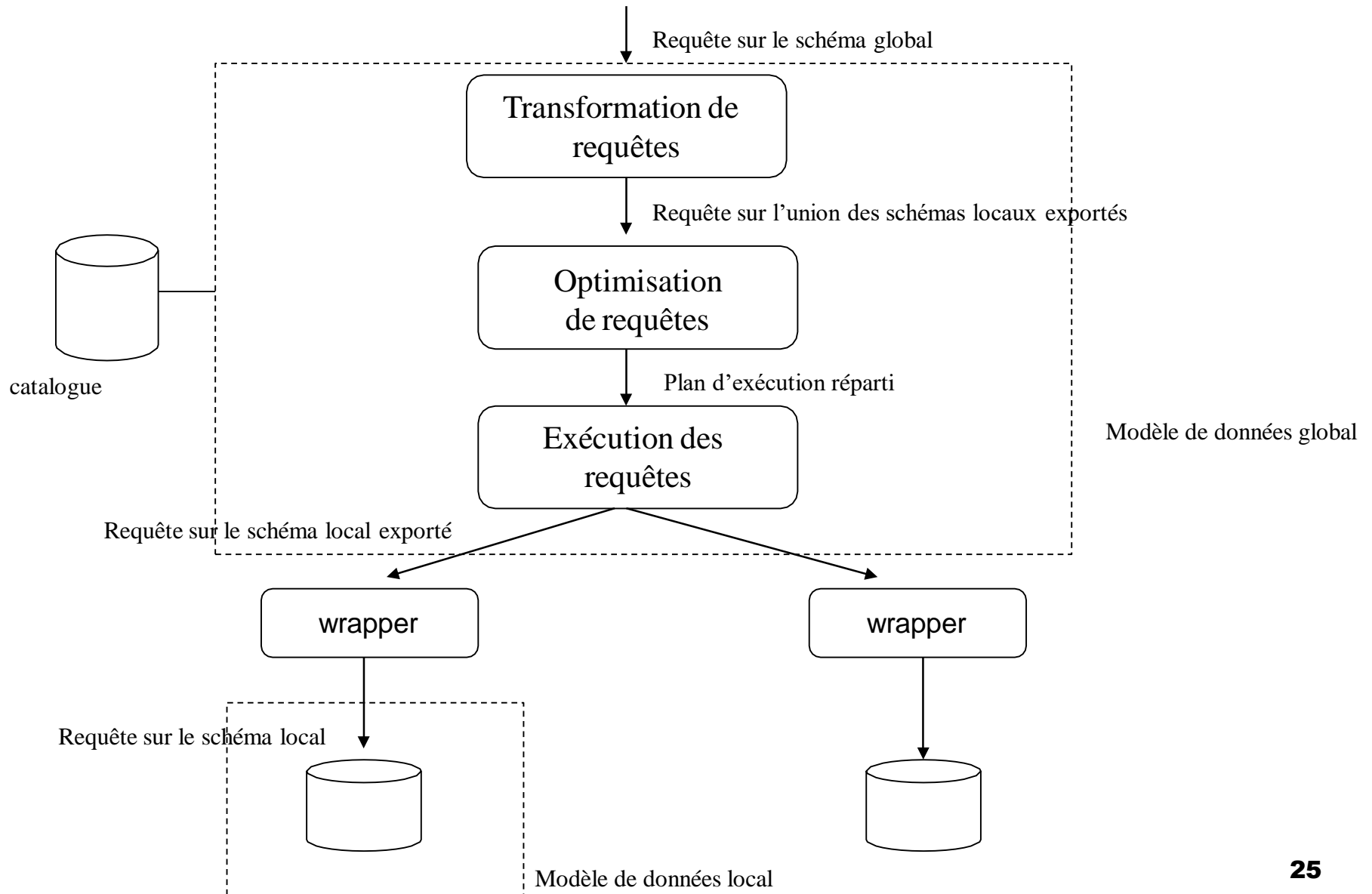
- Fonctions
  - catalogue global des données
  - intégration de schémas
  - génération d'adaptateurs
  - requêtes distribuées
- Bilan
  - point d'accès unique et uniforme
  - indépendance application/sources => évolution
  - les données sont toujours fraîches
  - traitement de requêtes peut être coûteux
  - performances

# Médiateurs





# Médiateur



# 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

**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).

# 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

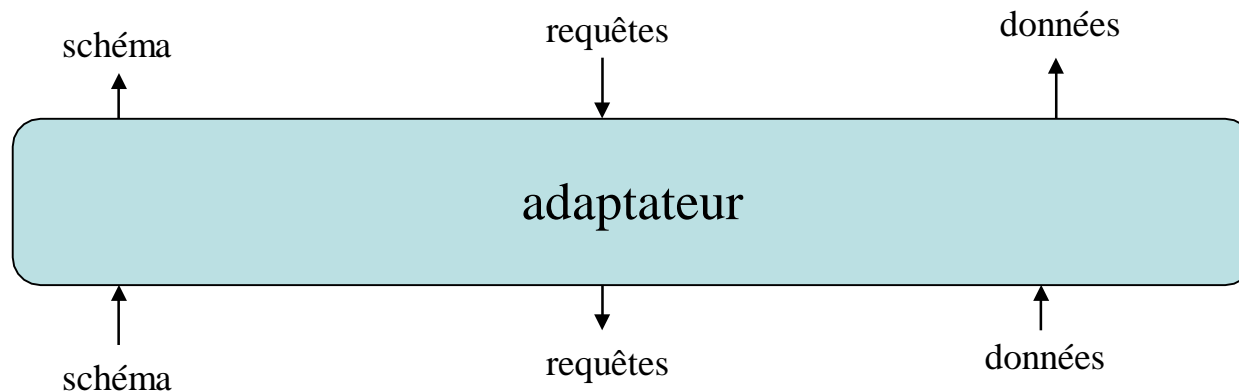
Trouver une requête R' utilisant seulement les données des relations sources, telle que

- 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



# 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
  - orienté-objet
- Plusieurs approches pour définir le schéma global :
  - Global as View
  - Local as View
  - 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 (dépliage 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

Projets(Pref, intitule, chef, annee, domaine)  
Rapport(label, mention)

Schéma de médiation

```
CREATE VIEW Projets (Pref, intitule, chef,  
année, domaine) as  
    SELECT num, titre, fondateur, annee,  
        NULL FROM S1.Projet  
UNION  
    SELECT refp, titre, directeur, NULL,  
        specialite FROM S2.Projet  
UNION SELECT * FROM S3.Projet
```

```
CREATE VIEW Rapport (label,  
mention) as SELECT  
    S1.Projet.titre,  
        S3.Avis.note FROM S1.Projet,  
    S3.Avis  
WHERE  
    S1Projet.num=S3.Avis.refpr  
.....
```

Projet (num, titre, fondateur,  
année)

Site1

Avis(refpr, note)

Projet(num, intitule, chef, année, domaine)

Site 3

Site 2

Projet(refp, titre, directeur, specialite )

# Exemple de requête en GAV

*Quelles sont les mentions des projets dirigés par Ali ?*

```
SELECT Projets.intitule, Rapport.mention FROM  
Projets, Rapport
```

```
WHERE Projets.chef = 'Ali' AND Projet.intitule=Rapport.label;
```

Traduction de la requête : par la technique de dépliement

```
SELECT S1.Projet.titre, S2.Projet.titre, S3.Pojet.intitule, S3.Avis.note  
FROM S1.Projet, S2.Projet, S3.Projet, S3.Avis  
WHERE (S1.fondateur='Ali' AND S1.Projet.nump=S3.Avis.refpr) OR  
(S2.Directeur='Ali' AND S2. Projet.refp=S3.Avis.refpr) OR  
(S3.chef='Ali' AND S3. Projet.nump=S3.Avis.refpr);
```