

Cours BD Avancées

Bases de Données Réparties

Djellali H. Master Réseau RSI

Fevrier 2021

Bases de Données Réparties

1. Définition
2. Architectures
3. Conception de BDR
4. Traitement des requêtes
5. Transaction répartie
6. Passerelles avec autres SGBD

1. Définition

➤ BD Répartie

- Ensemble de bases localisées sur différents sites, perçues par l'utilisateur comme une base unique

BD Répartie (*Distributed DB*)

un schéma global et une collection de BDs logiquement reliées et réparties entre plusieurs sites

- Chaque base possède **son schéma local**
- Le schéma de la base répartie constitue le **schéma global**
 - il n'est généralement pas matérialisé
 - il peut servir de support de conception
- Les données sont accédées via des vues intégrées
 - assurent la transparence à la localisation
 - permettent des recompositions de tables par union/jointure

Définition

- Une base de données répartie (distribuée) est une base de données logique dont les données sont distribuées sur plusieurs SGBD et visibles comme un tout.
- Si les données sont dupliquées, on parle plutôt de BD répliquée

Principe Fondamental

- « To the user, a distributed system should look exactly like a nondistributed system. » (C. Date, Introduction to Database Systems)
- Autonomie locale
- Égalité entre sites (pas de site « central »)
- Fonctionnement continu (pas d'interruption de service)
- Localisation transparente
- Fragmentation transparente
- Indépendance à la réplication
- Exécution de requêtes distribuées et Gestion de transactions réparties
- Indépendance vis-à-vis du matériel
- Indépendance vis-à-vis du Système d'Exploitation
- Indépendance vis-à-vis du réseau
- Indépendance vis-à-vis du SGBD

Principe Fondamental

➤ **Autonomie locale**

La BD locale est complète et autonome (intégrité, sécurité, gestion), elle peut évoluer indépendamment des autres (upgrades...)

➤ **Egalité entre sites**

Un site en panne ne doit pas empêcher le fonctionnement des autres sites (mais perturbations possibles)

➤ **Fonctionnement continu**

Distribution permet résistance aux fautes et aux pannes (en théorie)

Principe Fondamental

➤ **Localisation transparente**

Accès uniforme aux données quel que soit leur site de stockage

➤ **Fragmentation transparente**

Des données (d'une même table) éparpillées doivent être vues comme un tout

➤ **Indépendance à la réplication**

Les données répliquées doivent être maintenues en cohérence (délai possible)

Principe Fondamental

➤ **Indépendance vis-à-vis du SE**

Le SGBD fonctionne sur les différents SE...

Indépendance vis-à-vis du réseau

Le SGBD est accessible à travers les différents types de réseau utilisés

➤ **Indépendance vis-à-vis du SGBD**

La base peut être distribuée sur des SGBD hétérogènes

Principe Fondamental

➤ **Requêtes distribuées**

➤ L'exécution d'une requête peut être répartie (automatiquement) entre plusieurs sites (si les données sont réparties)

➤ **Transactions réparties**

Le mécanisme de transactions peut être réparti entre plusieurs sites (si ...)

➤ **Indépendance vis-à-vis du matériel**

Le SGBD fonctionne sur les différentes plateformes utilisées

Définition

- **Base de données fédérée** - a priori hétérogène (Federated DB) Plusieurs BD hétérogènes capables d'interopérer via une vue commune (modèle commun)

- **Multibase**

Plusieurs BD (hétérogènes ou non) capables d'interopérer sans une vue commune (absence de modèle commun)

- **SGBD réparti ou SGBD distribué** (Distributed DBMS)

Système gérant une collection de BD logiquement reliées, réparties sur différents sites en fournissant un moyen d'accès rendant la distribution transparente à l'utilisateur

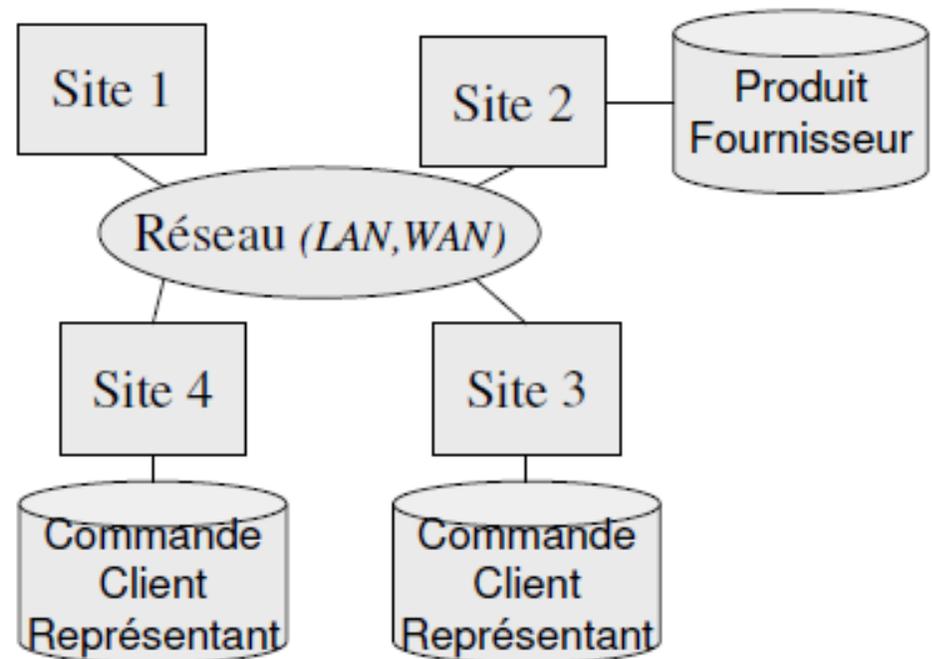
- BD distribuée ayant un schéma global : Les données sont réparties sur plusieurs sites, accessibles à partir du site central ou de tous les sites

SGBD Répartie

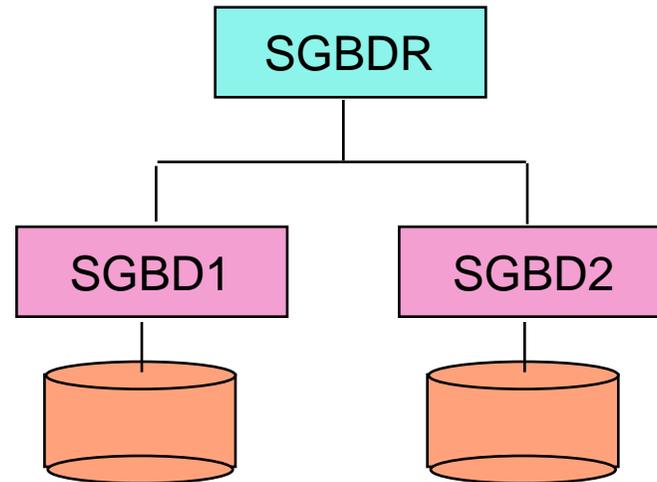
■ BD Répartie (*Distributed Database*)

- un schéma global
- une collection de BDs logiquement reliées et réparties entre plusieurs sites

Produit(NP, Designation, PrixUnit, NF)
Fournisseur(NF, Nom, Ville)
Client(NCL, Nom, Ville)
Représentant(NR, Nom, Ville)
Commande(NP, NCL, Date, Qte, NR)



SGBD réparti



➤ Rend la répartition (ou distribution) *transparente*

- dictionnaire des données réparties
- traitement des requêtes réparties
- gestion de transactions réparties
- gestion de la cohérence et de la sécurité

Evaluation de l'approche BDR

➤ avantages

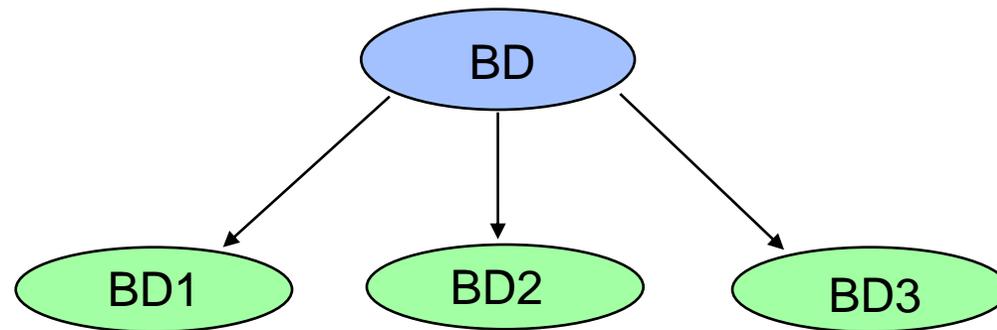
- extensibilité
- partage des données hétérogènes et réparties
- performances avec le parallélisme
- disponibilité avec la réplication

➤ inconvénients

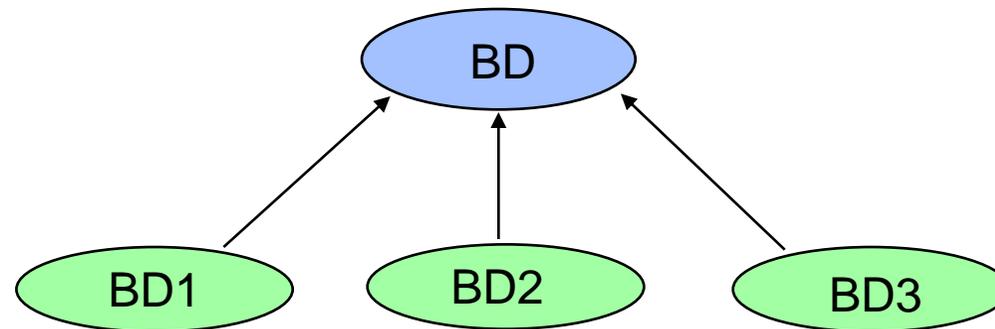
- administration complexe
- distribution du contrôle
- difficulté de migration

Migration vers une BDR

Décomposition en BD locales

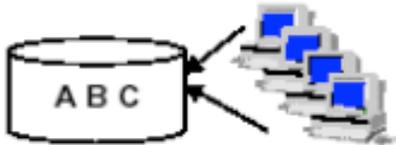


Intégration logique des BD locales existantes



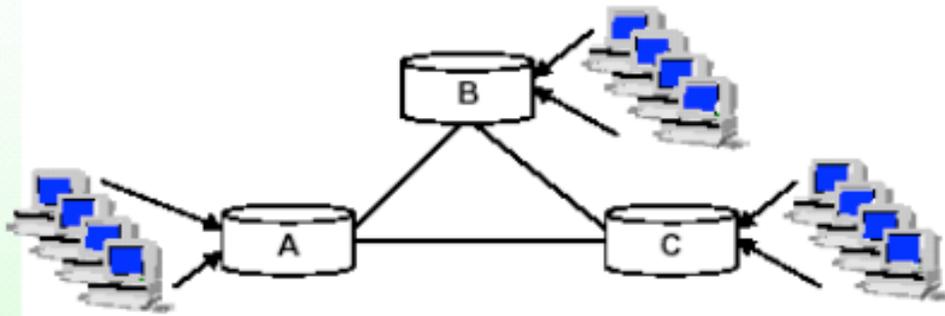
La Répartition Des Données [Bernard Espinasse: Introduction aux BD Réparties et Fédérés]

Centralisé



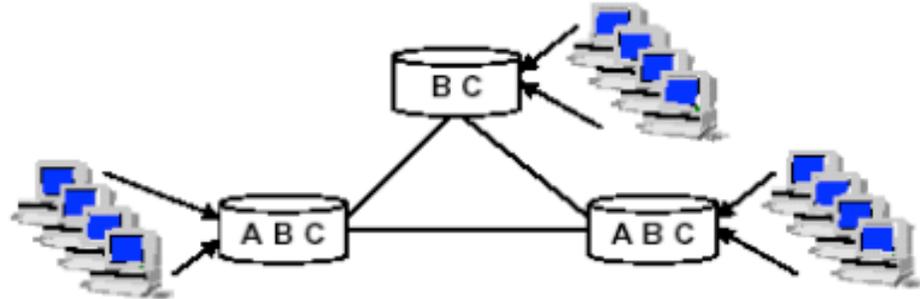
- + : Égalité des accès
Facilité de gestion
- : Contention sur la BD

Répartition



- + : Rapidité d'accès au données locales
Autonomie locale de chaque site,
Égalité des accès, facilité de gestion
Accès possible aux autres sites
- : Gestion globale de la BD

Duplication

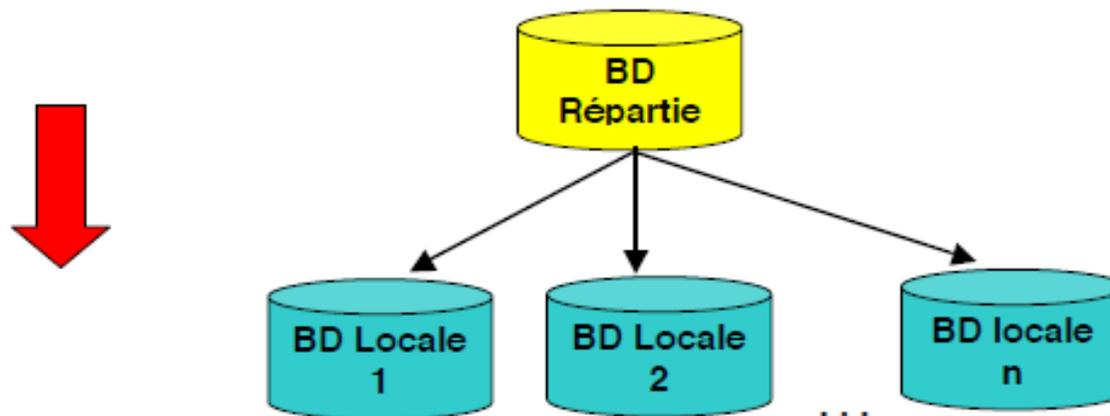


- + : Disponibilité des données
Rapidité d'accès aux données locales
- : Coordination de mises à jour

BD Répartis Intérêts et Problemes

BD Réparties

Approche descendante : **décomposition** en BD locales



Intérêts :

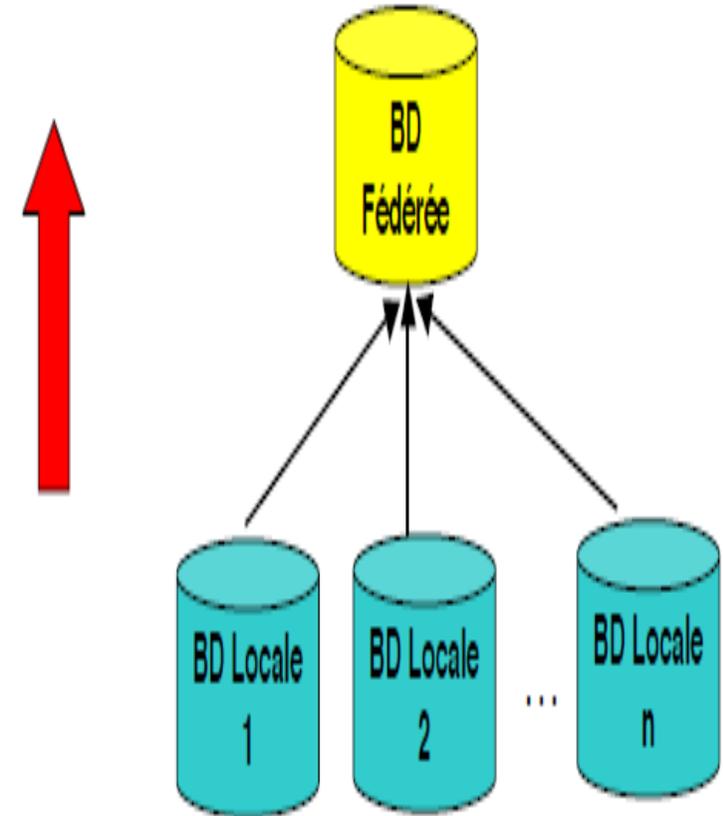
- **Amélioration des performances** (placer les traitements à l'endroit où se trouvent les données)
- **Disponibilité** en raison de l'existence de plusieurs copies
- **Maintien d'une vision unique** de la base de données malgré la répartition

Problèmes :

- **Complexité de la répartition** (fragmentation, duplication, placement)
- **Définition des schémas locaux** à partir du schéma global

Fédération De BD existantes: BD Fédérés

- **Intérêts :**
- • **Amélioration des performances** (placer les traitements où se trouvent les données)
- • Donner aux utilisateurs **une vue unique des données implémentées sur plusieurs systèmes à priori hétérogènes** (plates-formes et SGBD)
- • Cas typique rencontré lors de la concentration d'entreprises : faire cohabiter les différents systèmes tout en leur permettant d'interopérer
- **Problèmes :**
- • Hétérogénéité sémantique (BD) et syntaxique (SGBD, communications,...)
- • Intégration des schémas locaux pour créer un schéma global



Bases de Données Réparties

1. Définition

2. Architectures

3. Conception de BDR

4. Traitement des requêtes

5. Transaction répartie

6. Passerelles avec autres SGBD

2. Architecture de schémas

- indépendance applications/BDR
- schéma global lourd à gérer

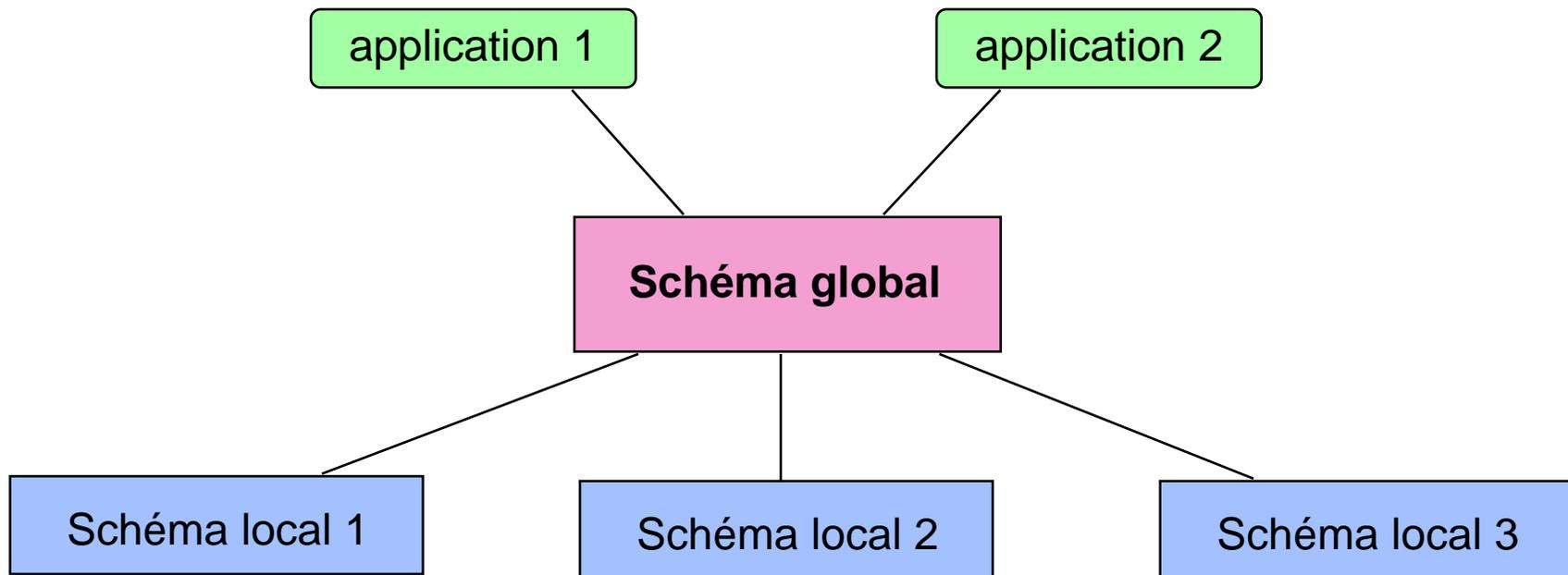


Schéma global

➤ schéma conceptuel global

- donne la description globale et unifiée de toutes les données de la BDR (e.g., des relations globales)
- indépendance à la répartition

➤ schéma de placement

- règles de correspondance avec les données locales
- indépendance à la localisation, la fragmentation et la duplication

➤ **Le schéma global fait partie du dictionnaire de la BDR et peut être conçu comme une BDR (dupliqué ou fragmenté)**

Exemple de schéma global

- Schéma conceptuel global

Client(nclient, nom, ville)

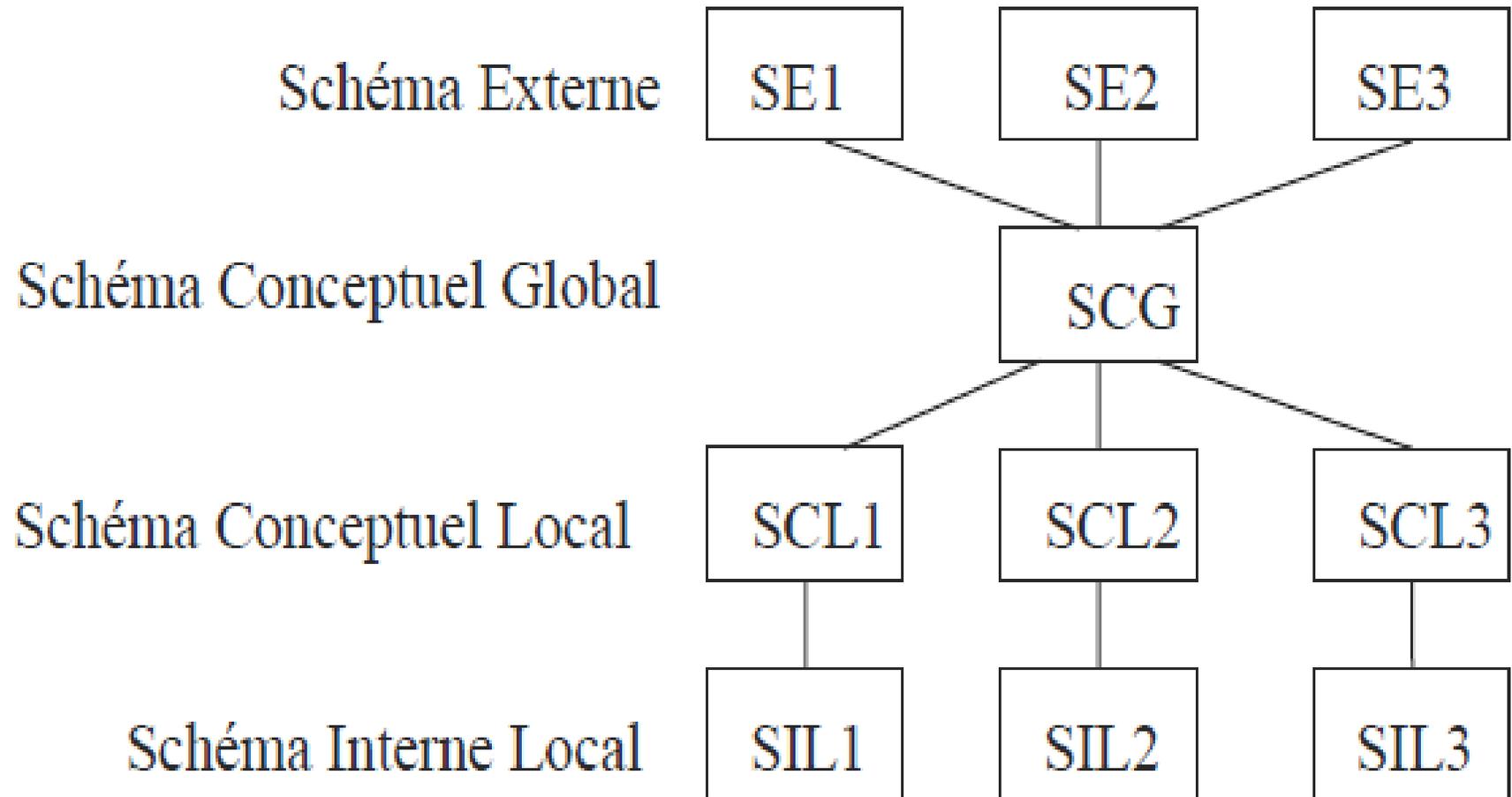
Cde (ncde, nclient, produit, qté)

- Schéma de placement

Client = Client1 @ Site1 U Client1 @ Site2

Cde = Cde @ Site3

Architecture de schémas BD Réparties



3. Conception des bases réparties

Conception de BD répartie

On ne met en place une BD répartie qu'en cas de réel besoin

Démarche de conception délicate

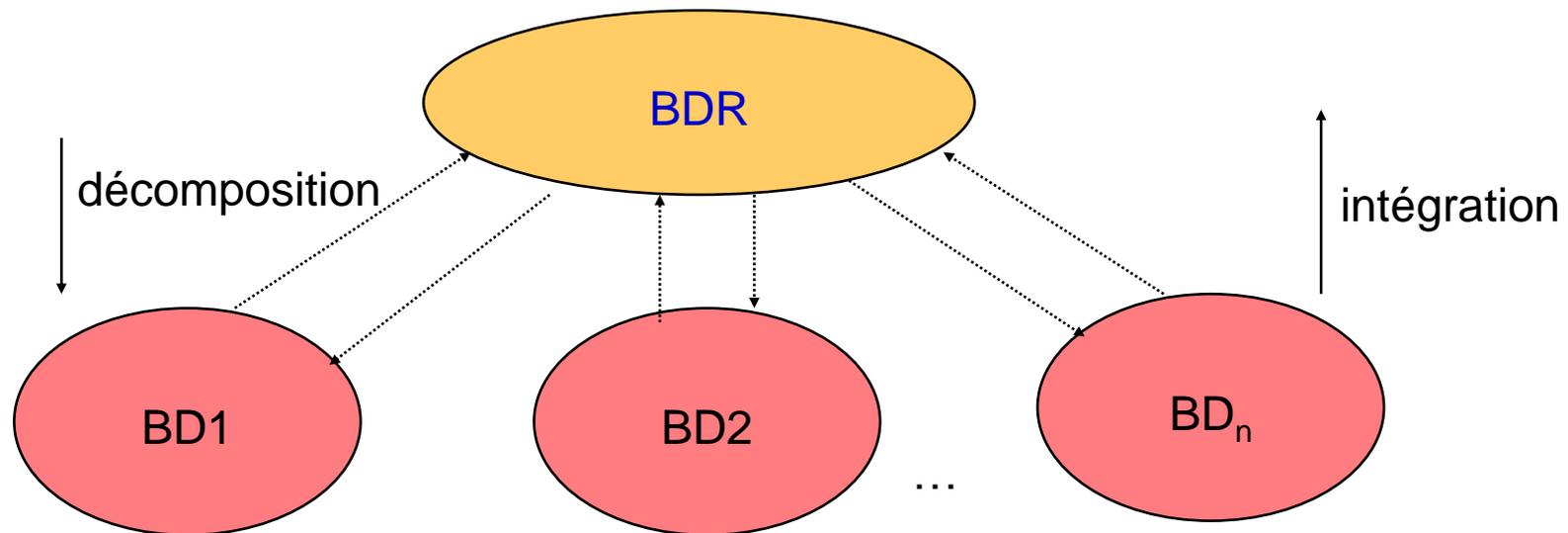
Gestion complexe

L'évolution du SI peut invalider la solution retenue...

Des raisons valables :

Volumes de données, sites distants, etc.

Fusions de SI



➤ **Conception ascendante**

- Part de l'existant
- Intègre bases locales dans schéma global

➤ ▪ **Conception descendante**

- On part du schéma global

On le scinde en schémas locaux

Conception descendante & ascendante

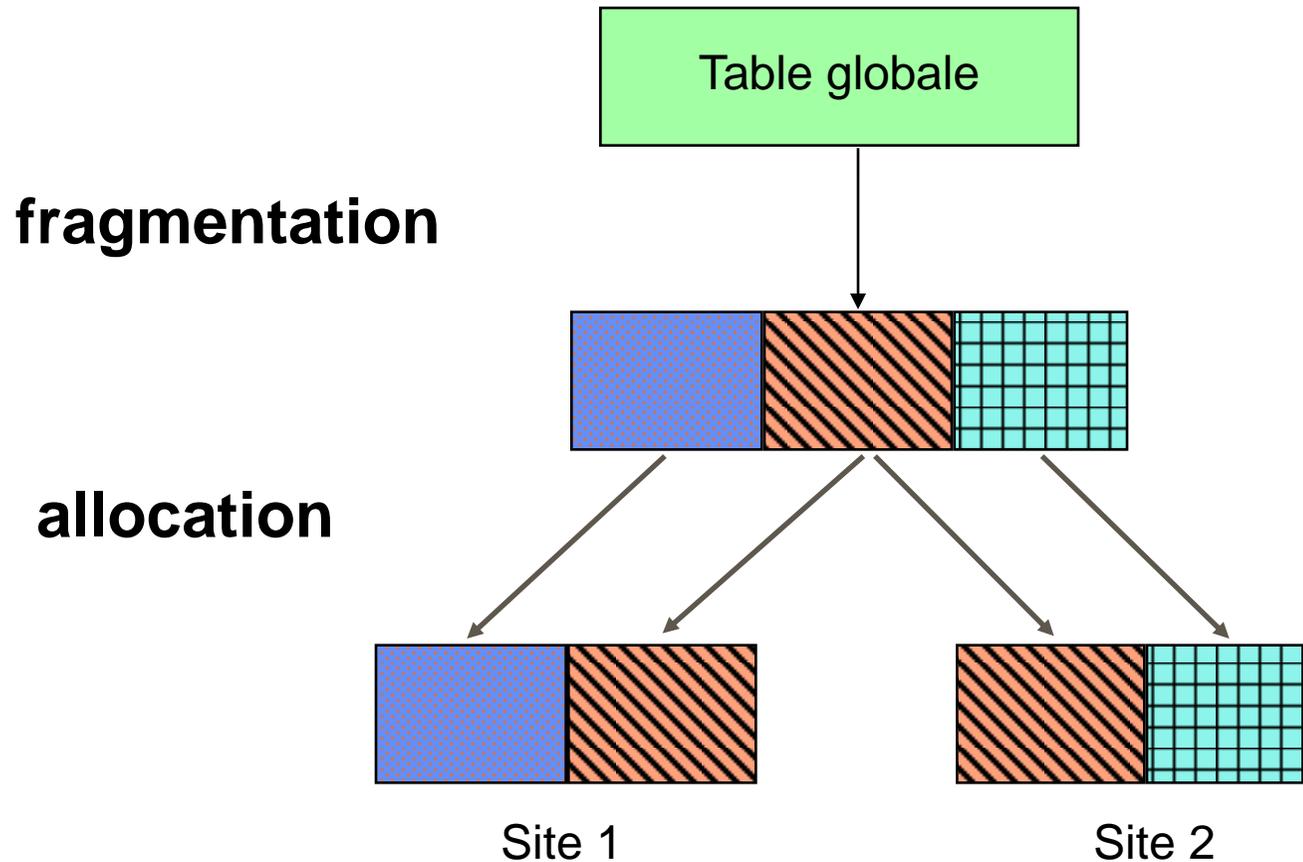
➤ Conception descendante

- On part de zéro (nouvelle base)
- Recherche de performance (pas forcément de répartition géographique)
- Assez peu fréquent

➤ · Conception ascendante

- Distribution pré-existante
- Nécessite consolidation, uniformisation (réconciliation sémantique)
- Identifier les données semblables
- Accorder leurs types, gérer leur cohérence...
- Interfacer ou adapter les SGBD...
- Ex : fusion,

Conception par décomposition



Objectifs de la décomposition

➤ **fragmentation**

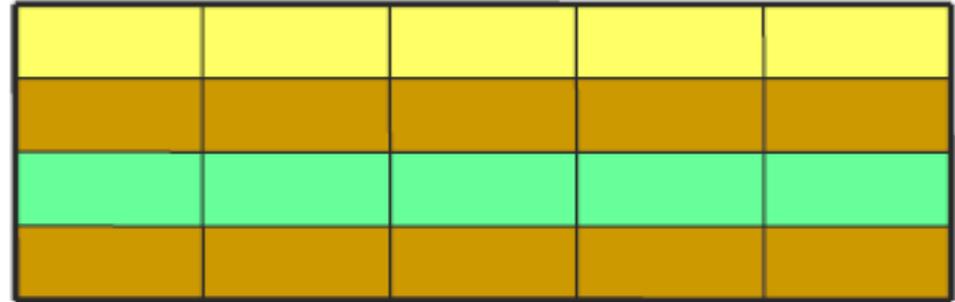
- trois types : horizontale, verticale, mixte
- performances en favorisant les accès locaux
- équilibrer la charge de travail entre les sites (parallélisme)

➤ **duplication (ou réplication)**

- favoriser les accès locaux
- augmenter la disponibilité des données

Fragmentation des données

- Fragmentation horizontale
 - Les tuples sont répartis



- Fragmentation verticale
 - Les tuples sont découpés et fragmentés
 - Nécessite colonne commune (clé ou unique) dupliquée



Fragmentation horizontale

➤ Fragments définis par sélection

- Client1 = Client where ville = "Paris"
- Client2 = Client where ville ≠ "Paris"

- Chaque fragment contient un sous-ensemble de **lignes**
- – Ex: comptes des clients de Paris sur le site de Paris.

Reconstruction

Client = Client1 U Client2

Client

nclient	nom	ville
C 1	Dupont	Paris
C 2	Martin	Lyon
C 3	Martin	Paris
C 4	Smith	Lille

Client1

nclient	nom	ville
C 1	Dupont	Paris
C 3	Martin	Paris

Client2

nclient	nom	ville
C 2	Martin	Lyon
C 4	Smith	Lille

Fragmentation horizontale dérivée

Fragments définis par jointure

Cde1 = Cde where

Cde.nclient = Client1.nclient

Cde2 = Cde where

Cde.nclient = Client2.nclient

Cde

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Reconstruction

Cde = Cde1 U Cde2

Cde1

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20

Cde2

ncde	nclient	produit	qté
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Fragmentation verticale

➤ Fragments définis par projection

- $Cde1 = Cde(ncde, nclient)$
- $Cde2 = Cde(ncde, produit, qté)$

➤ Reconstruction

- $Cde = [ncde, nclient, produit, qté]$ where $Cde1.ncde = Cde2.ncde$

Cde

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20
D 3	C 2	P 3	5
D 4	C 4	P 4	10

➤ Utile si forte affinité d'attributs

- **Ex:** la colonne des salaires sur le site de la comptabilité. Moins utilisé que la fragmentation horizontale

Cde1

ncde	nclient
D 1	C 1
D 2	C 1
D 3	C 2
D 4	C 4

Cde2

ncde	produit	qté
D 1	P 1	10
D 2	P 2	20
D 3	P 3	5
D 4	P 4	10

Allocation des fragments aux sites

➤ **Non-dupliquée**

- partitionnée : chaque fragment réside sur un seul site

➤ **Dupliquée**

- chaque fragment sur un ou plusieurs sites
- maintien de la cohérence des copies multiples

➤ **Règle intuitive:**

- si le ratio est [lectures/màj] > 1, la duplication est avantageuse

Exemple d'allocation de fragments

Client1

<i>nclient</i>	nom	ville
C 1	Dupont	Paris
C 3	Martin	Paris

Client2

<i>nclient</i>	nom	ville
C 2	Martin	Lyon
C 4	Smith	Lille

Cde1

<i>ncde</i>	<u>client</u>	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20

Site 1

Cde2

<i>ncde</i>	<u>client</u>	produit	qté
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Site 2

Mise en œuvre sous SQL (Assemblage)

- Frag. Horizontale
 - CREATE VIEW V1
AS SELECT Table1.cle, Table1.attr1
FROM Table1@site1
UNION
SELECT Table2.cle, Table2.attr1
FROM Table2@site2

Mise en œuvre sous SQL (Assemblage)

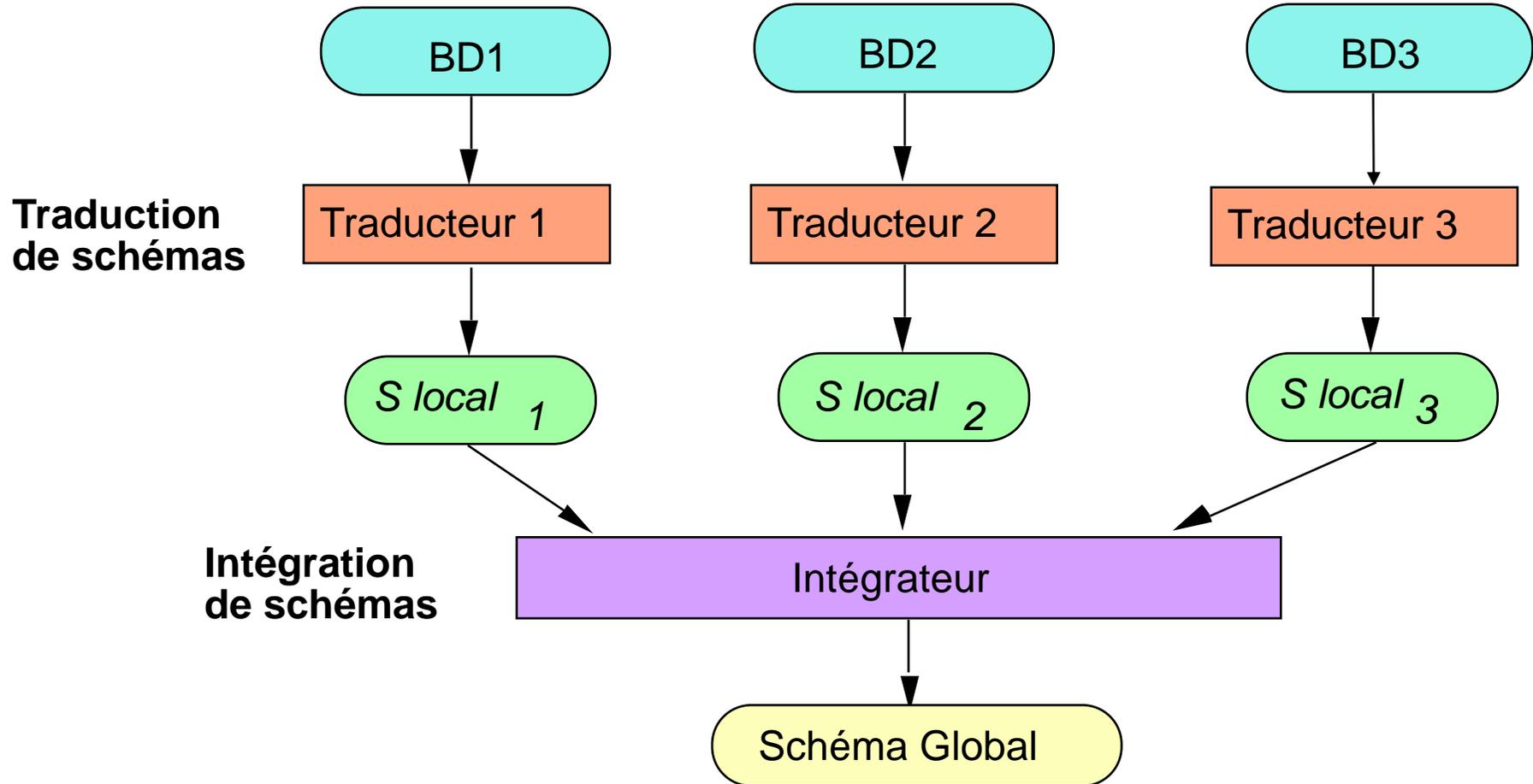
Frag. Verticale

- CREATE VIEW V1
AS SELECT Table1.cle, Table1.attr1,
Table2.attr2
FROM Table1@site1, Table2@site2
WHERE Table1.cle=Table2.cle

Remarque :

- l'attribut de fragmentation n'est pas forcément la clé primaire...
- En frag. verticale, il faut au néanmois que ce soit une clé

Conception BDR par intégration



4. Evaluation de requêtes réparties

Requête sur tables globales



Schéma
de fragmentation

A light blue oval with a black border containing the text "Schéma de fragmentation".

Requête sur fragments

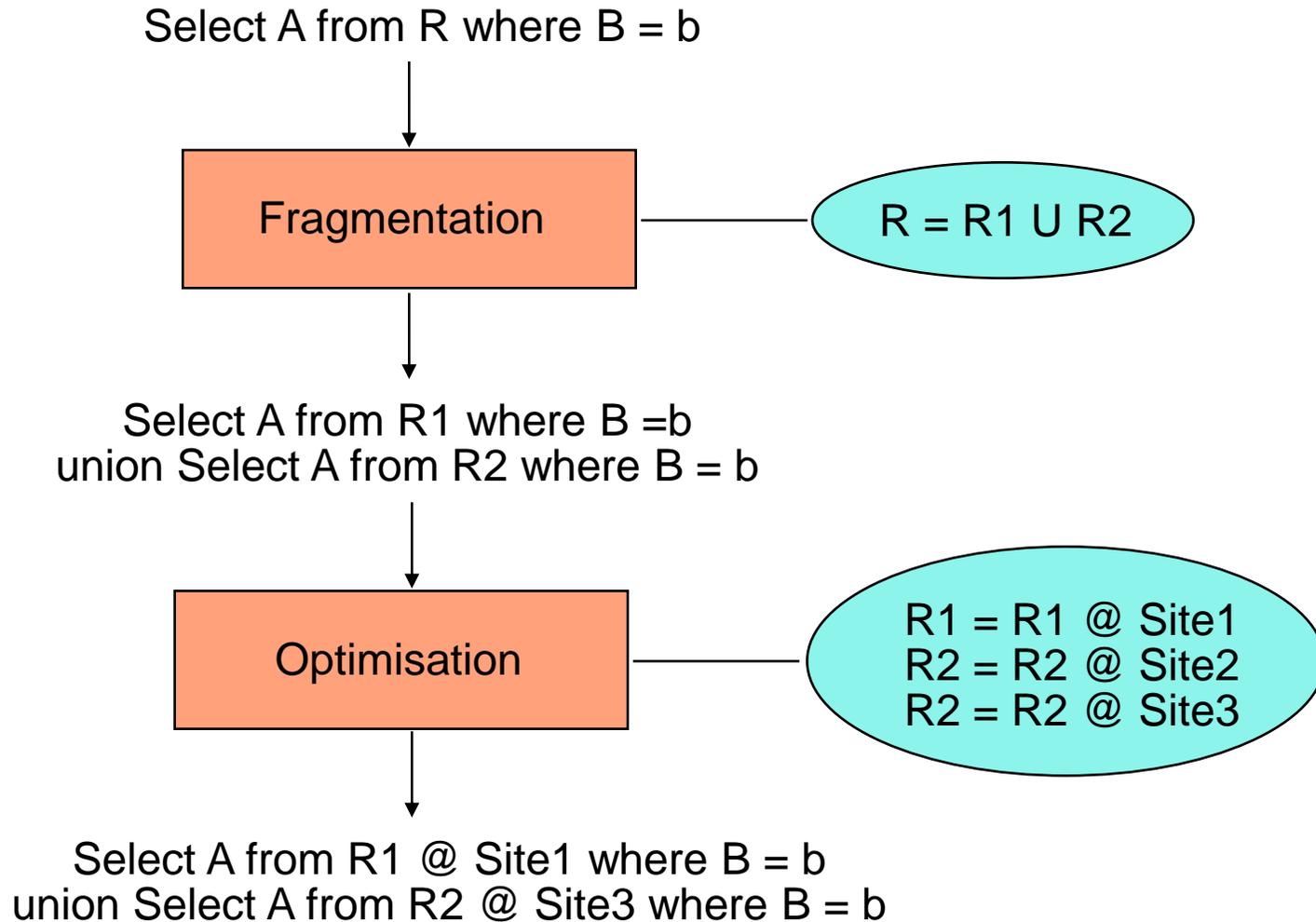


Schéma
d'allocation

A light blue oval with a black border containing the text "Schéma d'allocation".

Plan d'exécution réparti

Exemple d'évaluation simple



Communication Inter-sites

■ Chaque SGBD dispose d'un démon permettant les connexions distantes, sur un mode client - serveur

- *Listener (médiateur)*

Chaque SGBD dispose d'une table des BDs accessibles

- *Nom >> doit être unique !!!*

- *Adresse*

- *Protocole*

Cette approche permet aussi un équilibrage de charge transparent...

Exemple : Oracle

- Permet la distribution et la réplication
- Assure une bonne transparence à différents niveaux
- Système de nommage simple
 - sales.france.europe.computers
- Accès BD distante : LINK
 - CREATE DATABASE LINK sales...
- Accès table distante : schéma.table@base
 - svc_maint.emp@sales.france.europe.computers
- Lien public, lien privé

Caractéristiques

- Autonomie des sites
 - Gestion indépendante, upgrades localisés...
- Sécurité
 - Les utilisateurs et leurs rôles doivent être connus sur chaque site accédé
 - Possibilité d'utiliser un Security Server
 - Encryptage
- Administration globale : entreprise manager

Mise en oeuvre en SQL (Insertion avec les triggers Oracle)

- CREATE TRIGGER Tr1
INSTEAD OF INSERT on Table
BEGIN
IF :New.cle < 1000 THEN
 INSERT INTO Table1@site1(cle,attr1)
 VALUES(:New.cle,:New.attr);
ELSE
 INSERT INTO Table2@site2(cle,attr2)
VALUES (:New.cle,:New.attr);
END IF;
END;

Liens BD

3 types de liens :

- Connected user
 - Mode de base : même utilisateur en local et distant
 - Ce n'est pas nécessairement le créateur du lien
- Current user
 - Si lien utilisé à partir de procédure, identification en temps que propriétaire de la procédure
 - Intérêt ?
- Fixed user
 - Utilisateur fixe.
 - Intérêt ?

BD répartie avec Oracle (DATABASE LINKS)

- Permet à un usager local d'accéder aux tables d'une autre BD sans qu'il soit usager de cette BD
- **Syntaxe:**

```
CREATE [SHARED] [PUBLIC] DATABASE LINK nomLien  
CONNECT TO compteUsager IDENTIFY BY motDePasse  
USING nomService
```

- SHARED: permet de partager la connexion entre plusieurs usagers
- PUBLIC : rend le lien disponible à tous les usagers locaux
- nomService : doit être défini dans un fichier de configuration de la BD

BD répartie avec Oracle (DATABASE LINKS)

Exemple: accès au catalogue de produits en Grande-Bretagne

```
CREATE DATABASE LINK site.europe.uk  
CONNECT TO Comptel14 IDENTIFY BY "abc123"  
USING SID_siteUK  
  
SELECT * FROM Comptel14.Catalogue@site.europe.uk
```

Exemple

- `CREATE DATABASE LINK ventes.ulp.fr USING 'ventes';`
- `CREATE DATABASE LINK ventes.ulp.fr CONNECT TO scott IDENTIFIED BY tiger USING 'ventes';`
- `CREATE DATABASE LINK ventes.ulp.fr CONNECT TO CURRENT_USER USING 'ventes';`
- Remarque : USING -> nom du service!

Liens BD : notions avancées

- Visibilité :
 - Public
 - Pratique si utilisé par un grand nombre d'utilisateurs
 - Private (défaut)
 - + sécurisé, accessible uniquement au propriétaire et à ses sous-programmes
 - Global
 - Accessible de n'importe quelle base, mais suppose une gestion centrale (directory server)

Exemple

```
CREATE PUBLIC DATABASE LINK ventes.ulp.fr  
USING 'ventes';
```

```
CREATE PUBLIC DATABASE LINK ventes.ulp.fr  
CONNECT TO scott IDENTIFIED BY tiger  
USING 'ventes';
```

```
CREATE PUBLIC DATABASE LINK ventes.ulp.fr  
CONNECT TO CURRENT_USER  
USING 'ventes';
```

Différence avec privé ?

- La principale difficulté réside dans le fait qu'une mise à jour dans une relation du schéma global se traduit par plusieurs mises à jours dans différents fragments.
- Il faut donc identifier les fragments concernés par l'opération de mise à jour, puis décomposer en conséquence l'opération en un ensemble d'opération de mise à jour sur ces fragments.

Mise à jour de BD réparties

➤ ***Insertion***

- Retrouver le fragment horizontal concerné en utilisant les conditions qui définissent
- les fragments horizontaux, puis insertion du tuple dans tous les fragments verticaux correspondants.

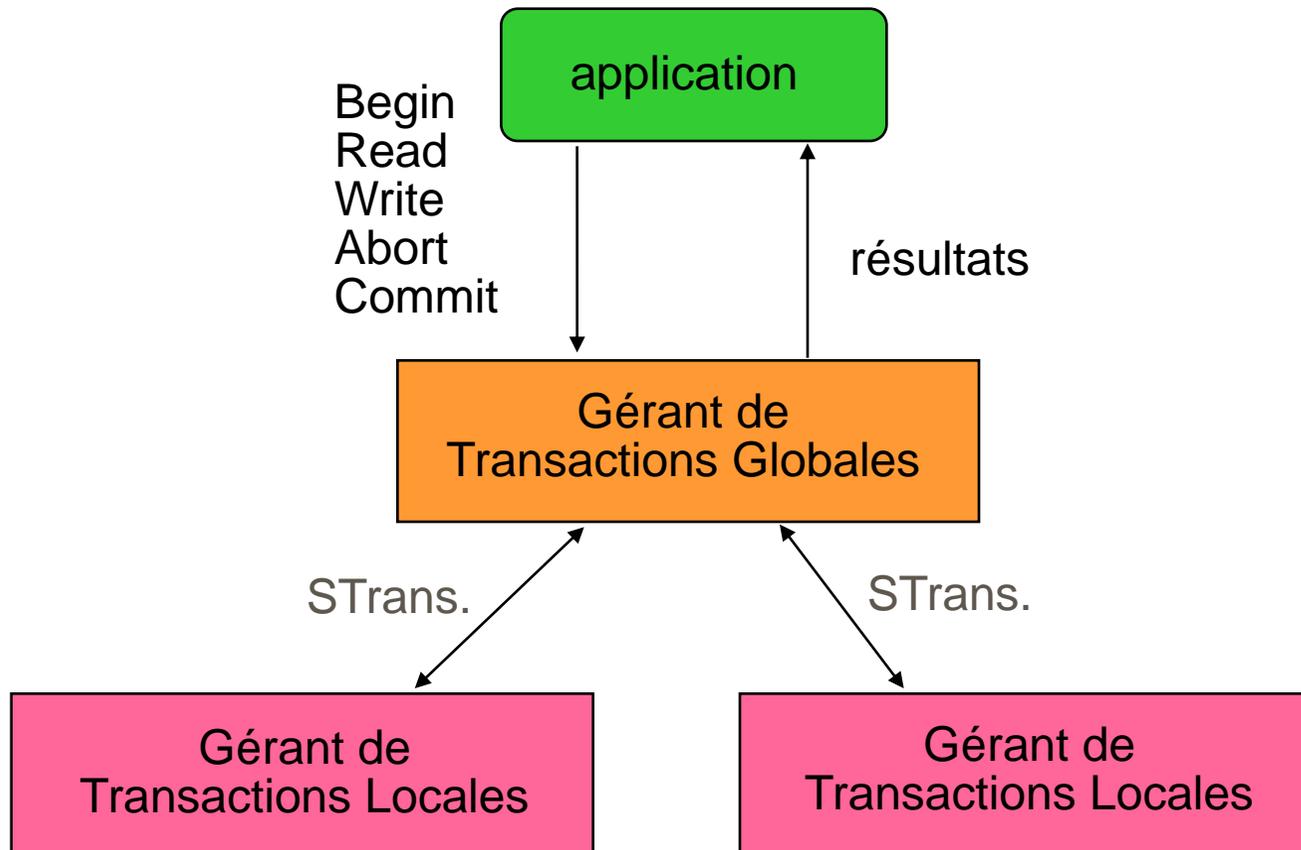
➤ ***Suppression***

- Rechercher le tuple dans les fragments qui sont susceptibles de contenir le tuple concerné, et supprimer les valeurs d'attribut du tuple dans tous les fragments verticaux.

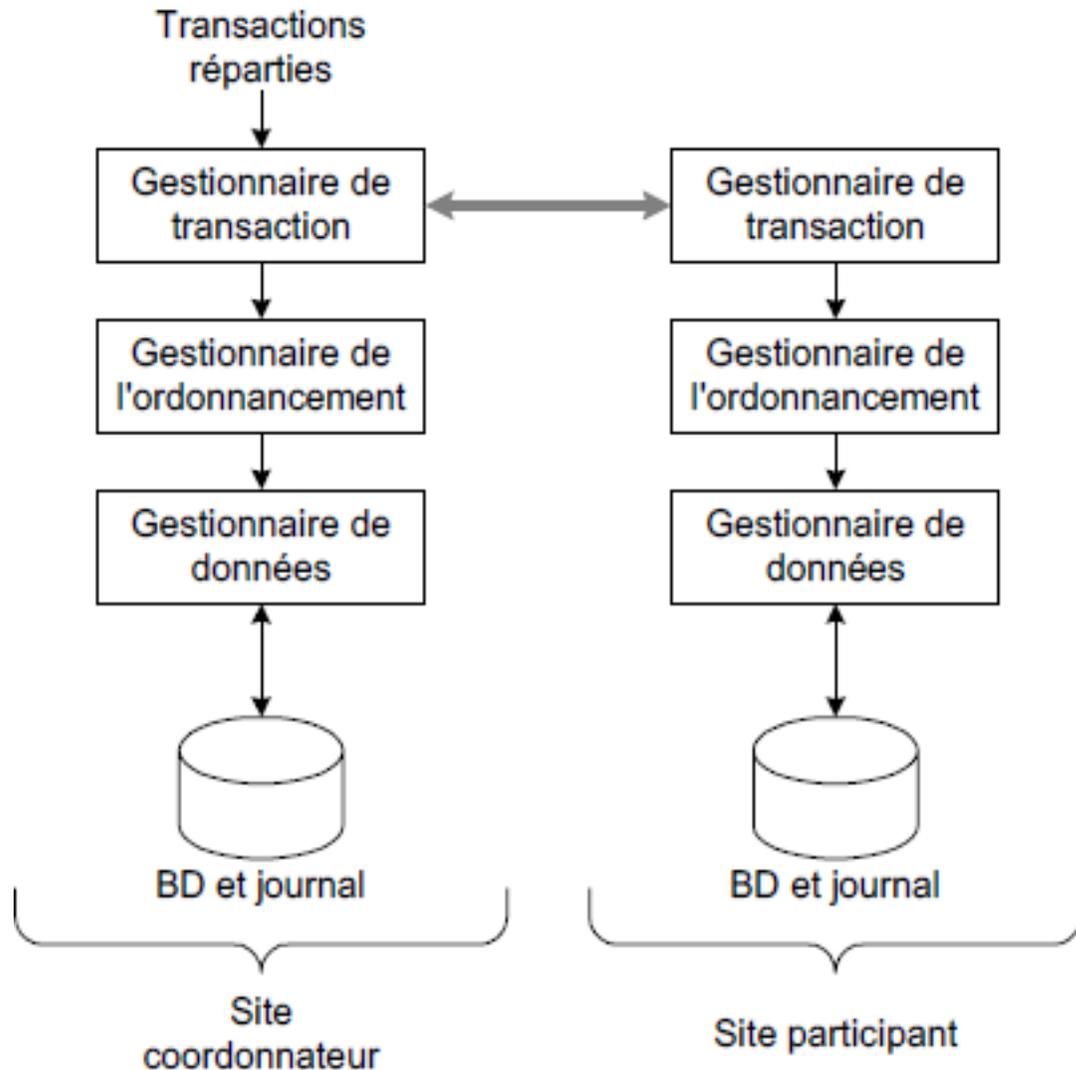
Mise à jour de BD réparties

- Modification
- Rechercher les tuples, les modifier et les déplacer vers les bons fragments si nécessaire.

5. Notion de Transaction Répartie



Gestion des Transactions Réparties



Site coordonnateur:

À l'origine de la transaction répartie

Site participant:

Coopère avec le gestionnaire de transaction du site coordonnateur

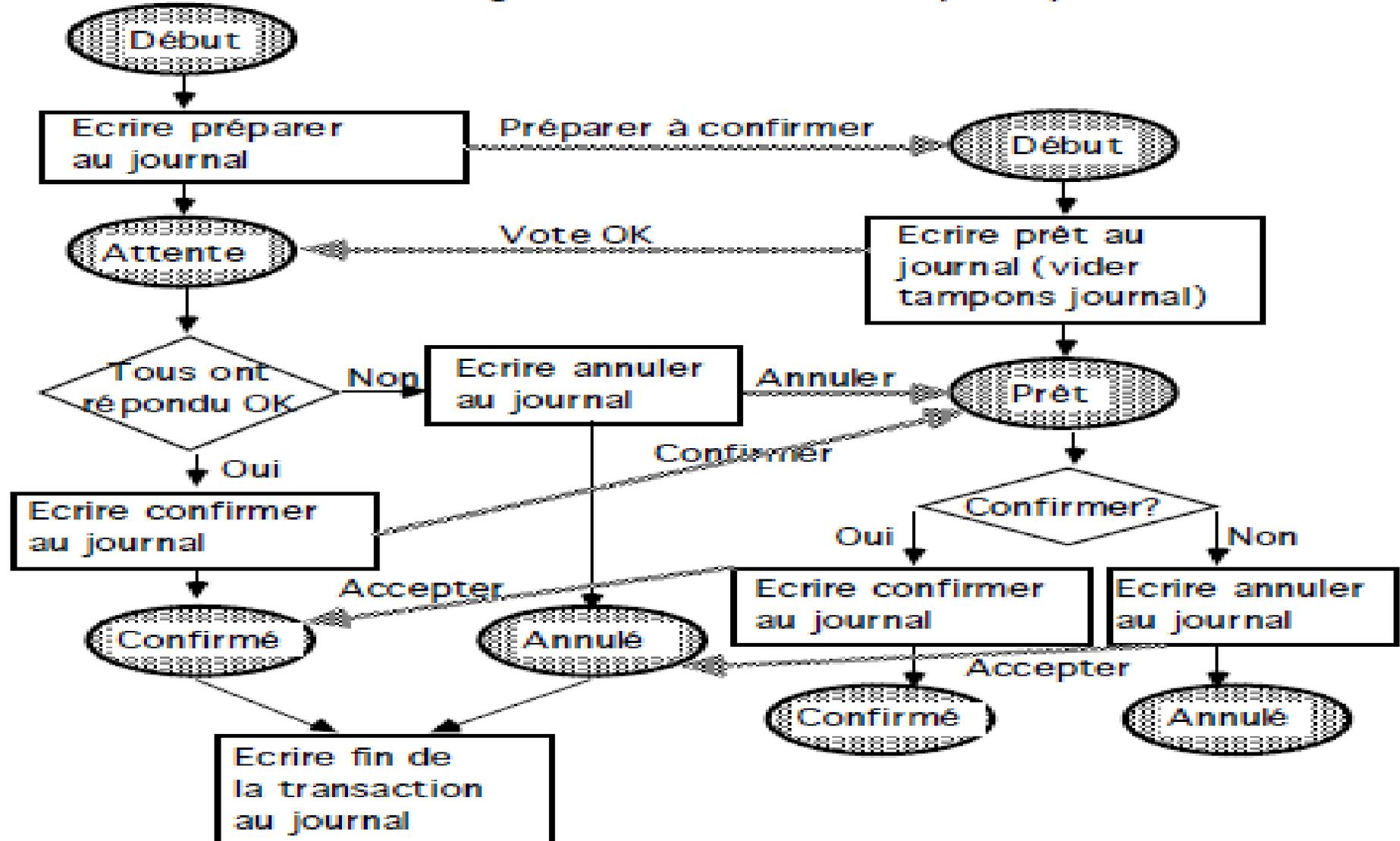
Protocole de validation en 2 étapes

- **Objectif** : Exécuter la commande COMMIT pour une transaction répartie
 - Phase 1 : Préparer à écrire les résultats des mises-à-jour dans la BD
 - Phase 2 : Ecrire ces résultats dans la BD
- **Coordinateur** : composant système d'un site qui applique le protocole
- **Participant** : composant système d'un autre site qui participe dans l'exécution de la transaction

Protocole de confirmation en 2 phases commit 2PC

Site coordonnateur (usager)

Site participant (données)



Protocole de confirmation en deux phases (Two-phase commit – 2PC)

Phase 1: demande de confirmation

- Le coordonnateur demande aux participants de confirmer ou infirmer le succès des transactions locales

Phase 2: confirmation

- Le coordonnateur reçoit les réponses des sites participants
- Si tous les participants ont confirmé le succès, le coordonnateur autorise les participants à compléter les transactions locales (écriture du commit dans leur journal)
- Sinon, le coordonnateur demande aux participants d'annuler les transactions locales

Note: les ressources sont bloquées (verrous) durant l'attente de la confirmation

Transparence de localisation (SYNONYM)

■ Synonyme

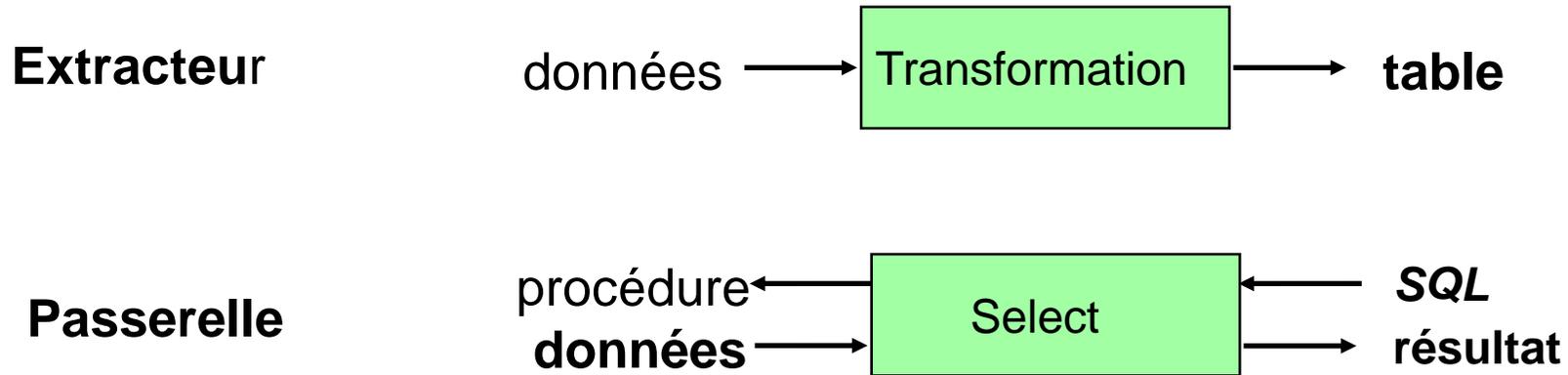
- Évite aux applications de devoir connaître la localisation des données (transparence de localisation)
- Permet de conserver les mêmes requêtes, même si le lien change

```
CREATE [PUBLIC] SYNONYM nomSynonyme  
FOR [nomSchéma].nomObjet[@lienBD]
```

■ Exemple (suite):

```
CREATE SYNONYM CatalogueEurope  
FOR comptel4.Catalogue@site.europe.uk  
  
SELECT * FROM CatalogueEurope  
WHERE ...
```

6. Extracteurs et Passerelles



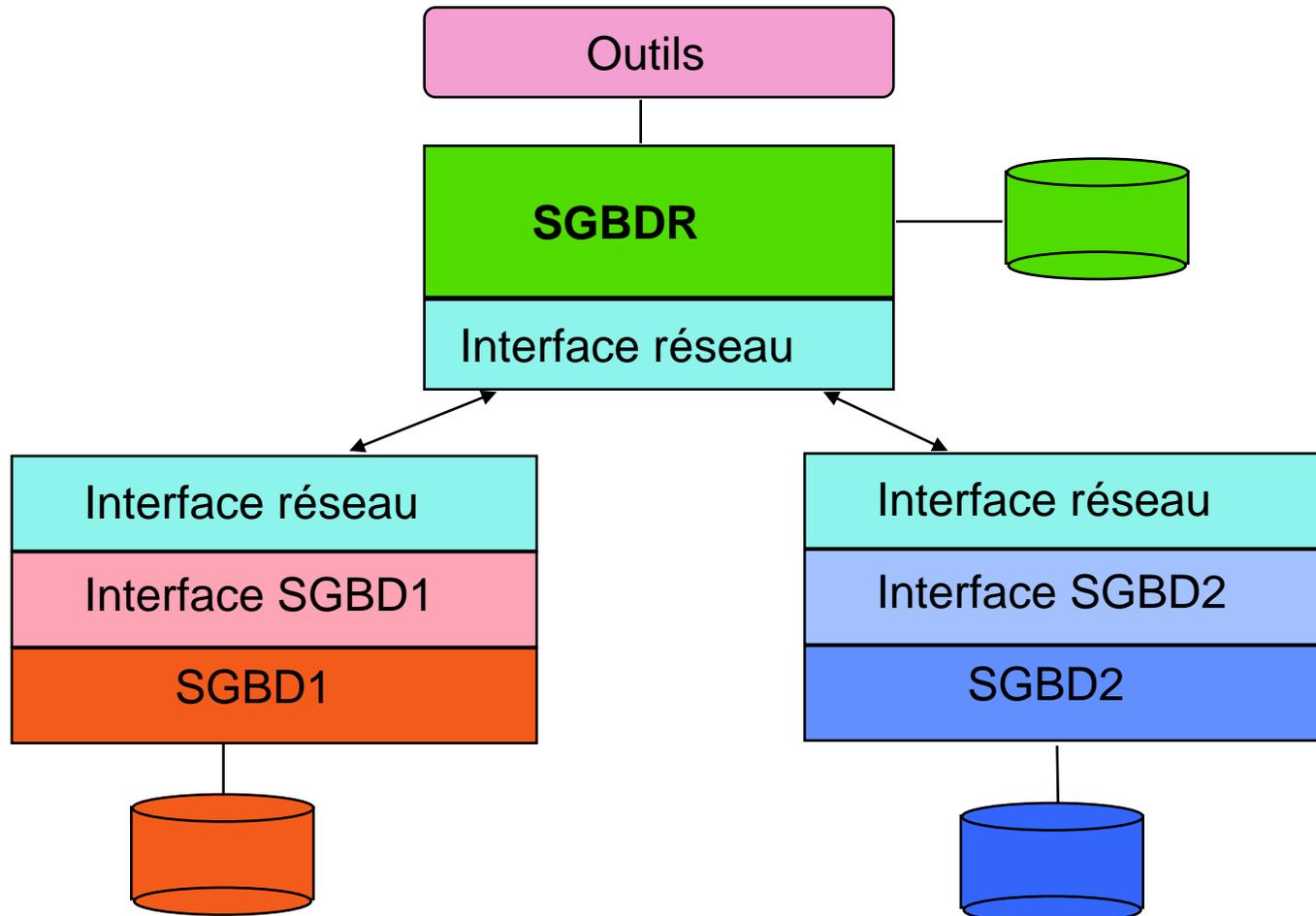
➤ Fournisseurs indépendants

- Outils ETL (ETI, Data Stage, Informatica, Sagent, Genio, Amadea, Sunopsis, ...).

➤ Editeurs de SGBD

- passerelles entre le SGBD et les données sources

SGBD réparti hétérogène



Produits

- SGBD relationnels
 - Oracle, DB2, SQL Server 2000, Sybase, Informix
- VirtualDB (Enterworks)
 - basé sur GemStone, vue objet des tables
- Open Database Exchange (B2Systems)

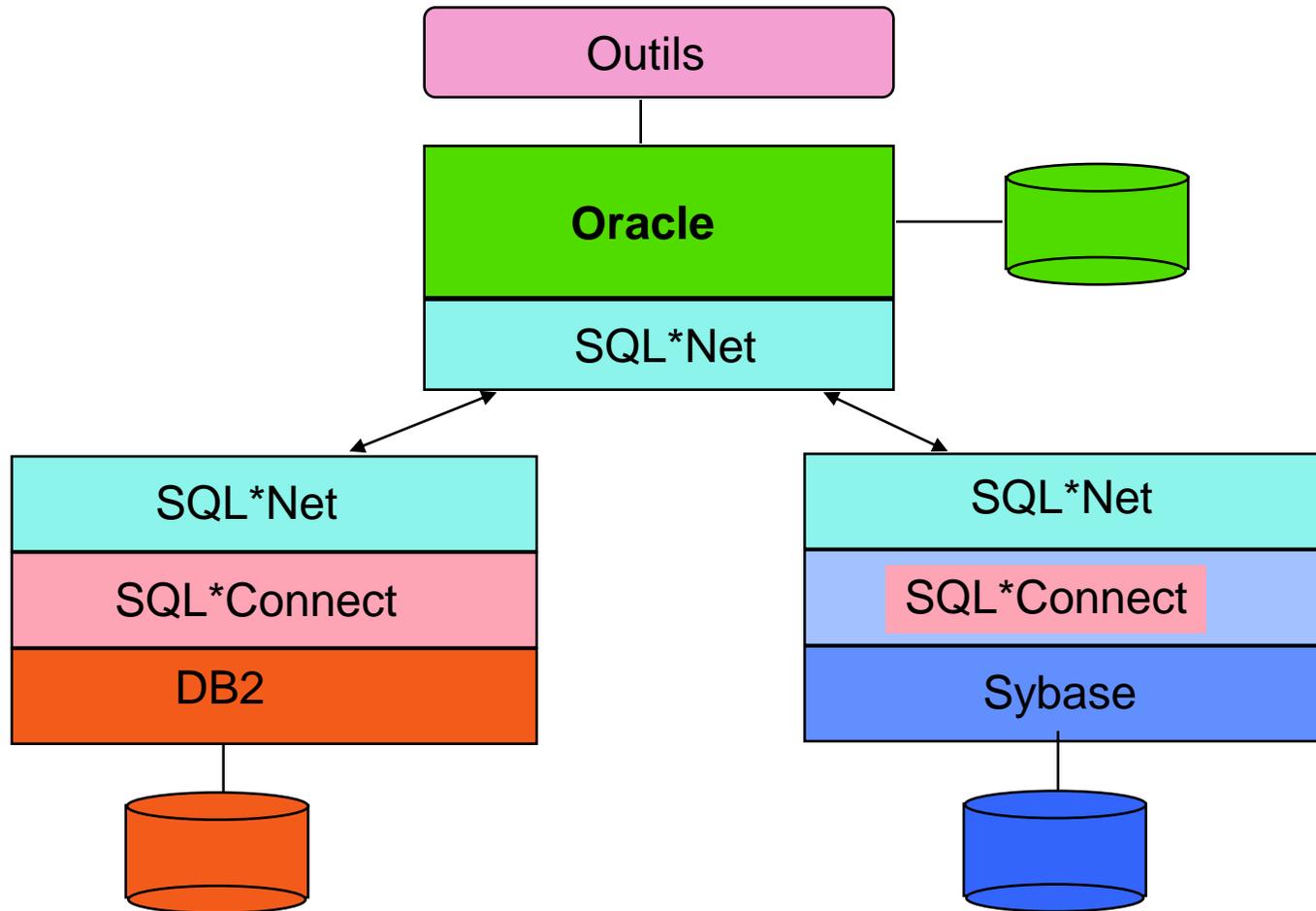
Oracle/Star

- SGBD Oracle
 - gestion du dictionnaire de la BDR
- SQL*Net
 - transparence au réseau
 - connexion client-serveur, login à distance automatique
 - évaluation de requêtes réparties
 - validation en deux étapes et réplication
- SQL*Connect : passerelle vers les bases non-Oracle

Database link

- Lien à une table dans une BD distante spécifié par :
 - nom de lien
 - nom de l'utilisateur et password
 - chaîne de connexion SQL*Net (protocole réseau, nom de site, options, etc...)
- Exemple
 - CREATE DATABASE LINK empParis
 - CONNECT TO patrick
 - IDENTIFIEDBY monPW
 - USING Paris.emp

Oracle/Star : architecture



Références

- Bernard Espinasse Introduction aux BD Réparties
 - Matthieu Exbrayat – BD Réparties
- ULP Strasbourg - Décembre 2007