

Introduction aux Bases de Données Réparties (BDR) **(emprunts G. Gardarin)**

Quelques définitions associées

Principes architecturaux

Conception de BDR

Vues et requêtes de consultation

Mise en œuvre Oracle (Solution DBLink)

Evaluation des approches réparties

- **Avantages**

- extensibilité
- partage de données hétérogènes et réparties
- performances
- disponibilité des données

- **Inconvénients**

- administration complexe
- distribution du contrôle

Constituants du 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

Exemple de schéma global

➤ Schéma conceptuel global

Client(nclient, nom, ville)

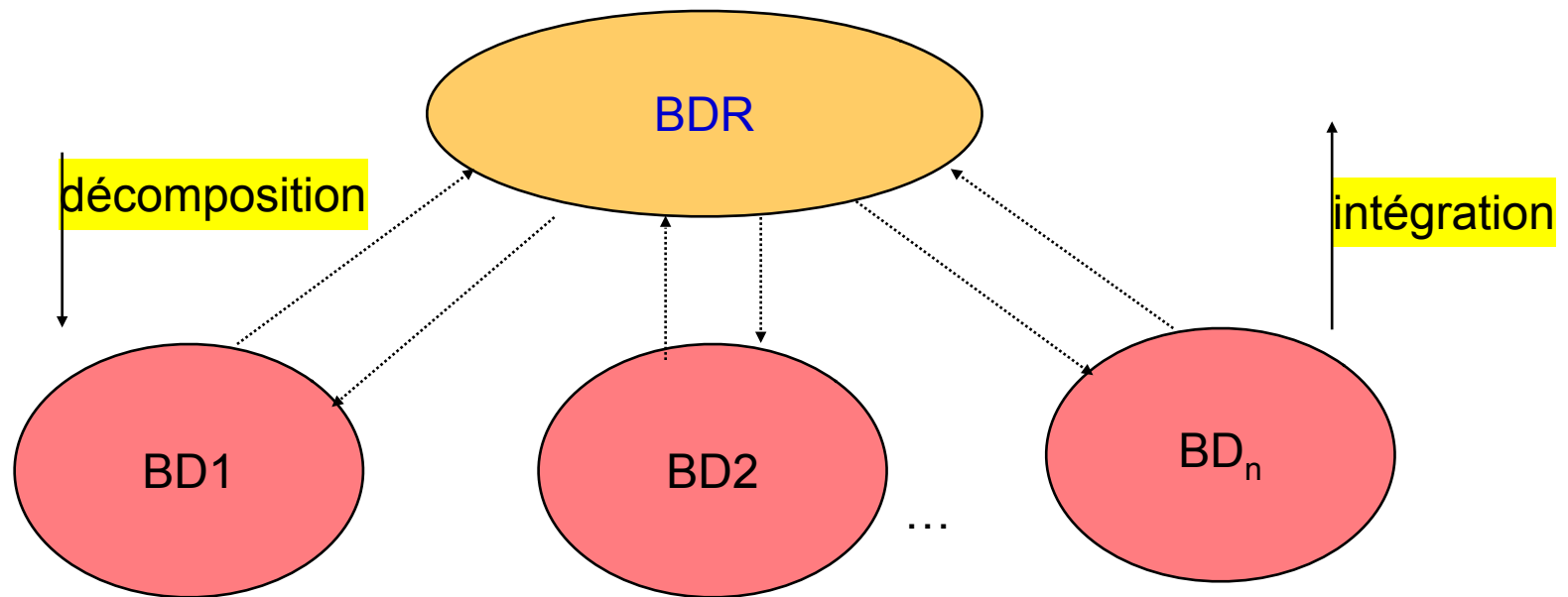
Cde (ncde, nclient, produit, qté)

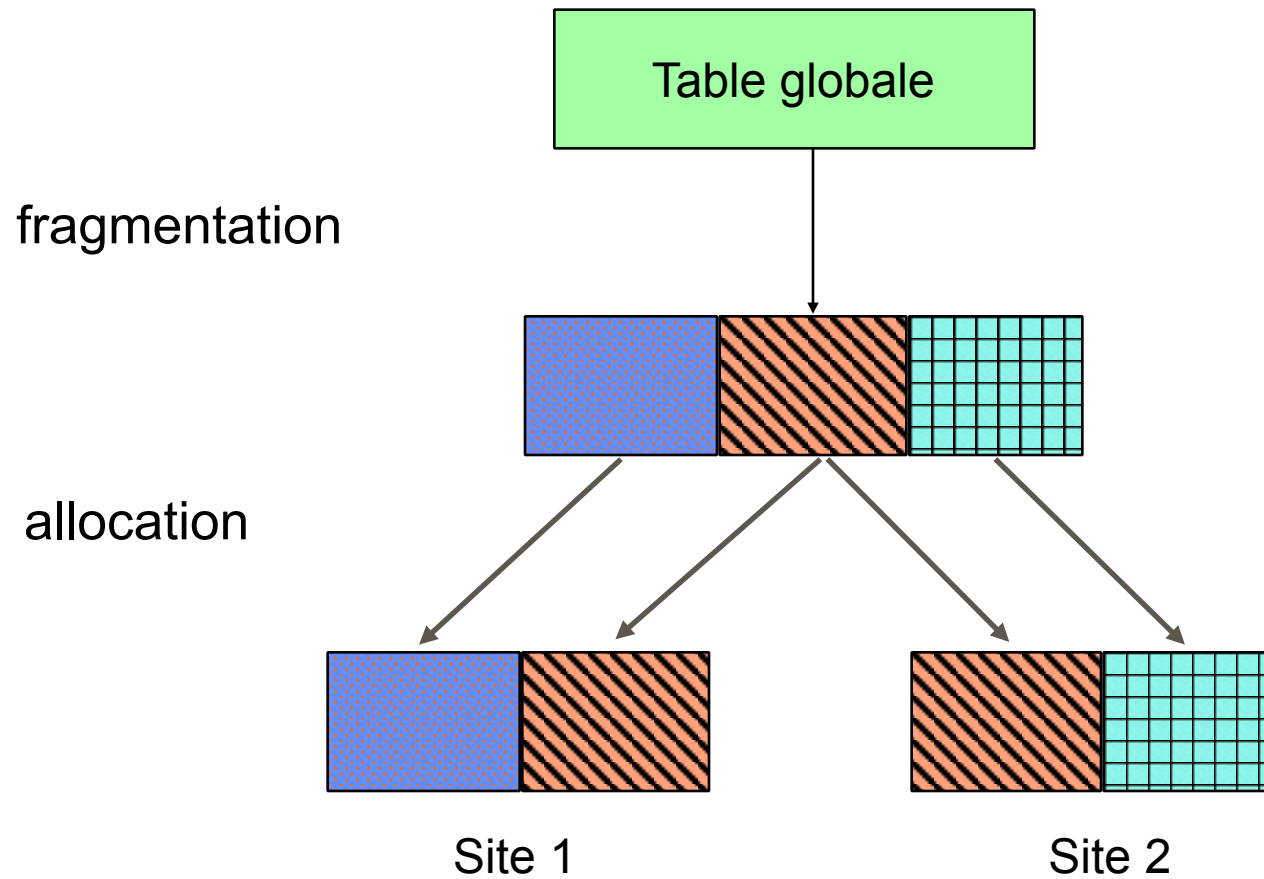
➤ Schéma de placement

Client = Client1 @ Site1 U Client1 @ Site2

Cde = Cde @ Site3

Principes clés de conception des BDR





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 horizontale

➤ Fragments définis par sélection

- Client1 = Client where ville = "Paris"
- Client2 = Client where ville ≠ "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 $Cde1.ncde = Cde2.ncde$

➤ Utile si forte affinité d'attributs

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

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 $[\text{lectures/màj}] > 1$, la duplication est avantageuse

Exemple d'allocation de fragments

Client1

nclient	nom	ville
C 1	Dupont	Paris
C 3	Martin	Paris

Client2

nclient	nom	ville
C 2	Martin	Lyon
C 4	Smith	Lille

Cde1

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

Site 1

Cde2

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

Site 2

Evaluation de requêtes réparties

Requête sur tables globales

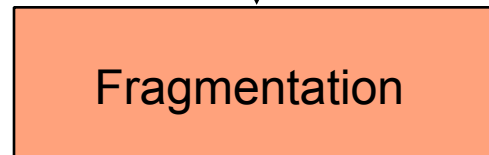


Schéma
de fragmentation

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

Requête sur fragments

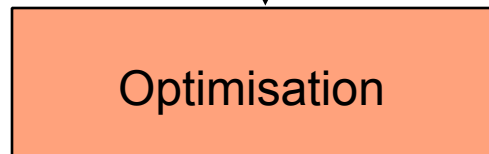
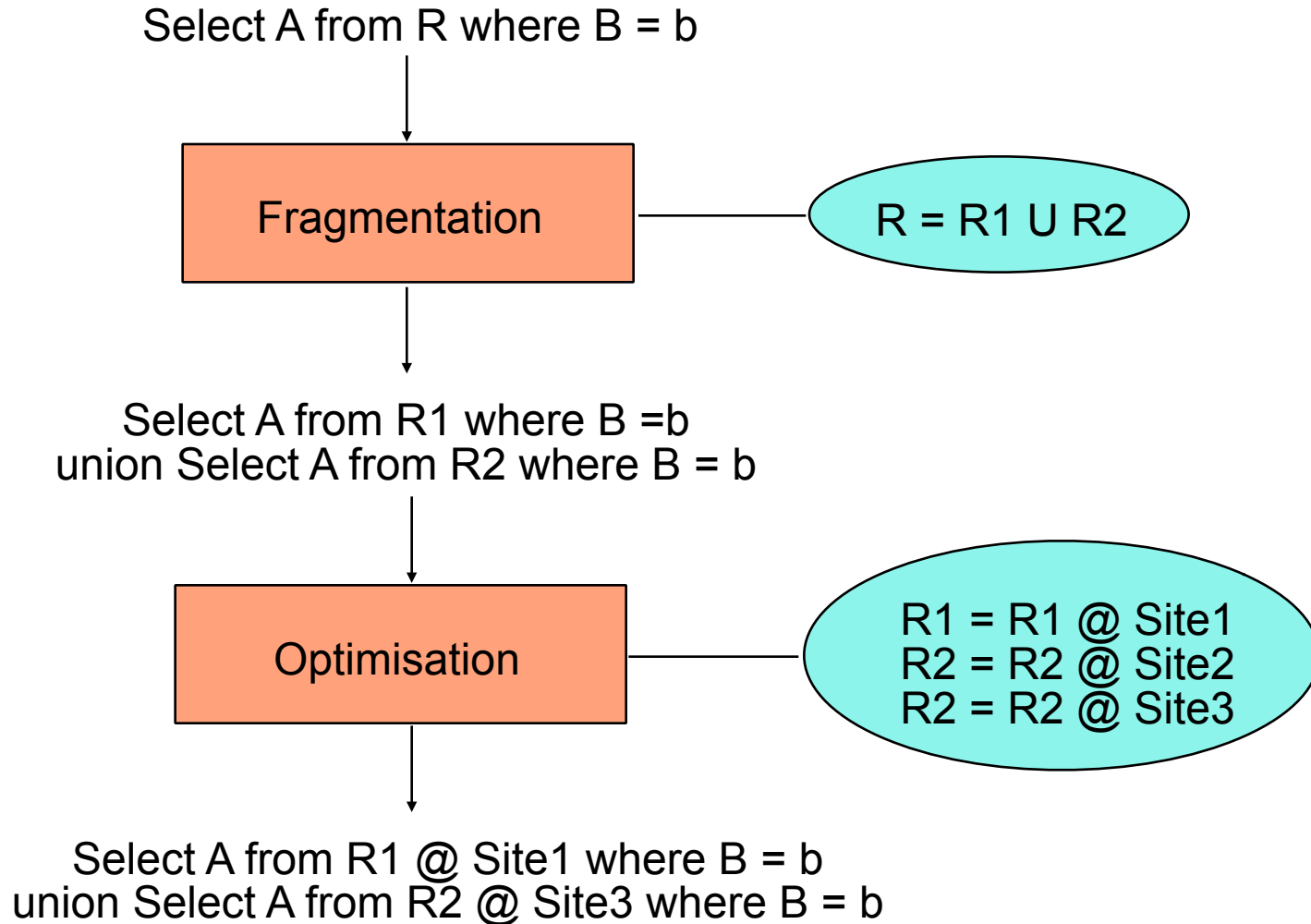


Schéma
d'allocation

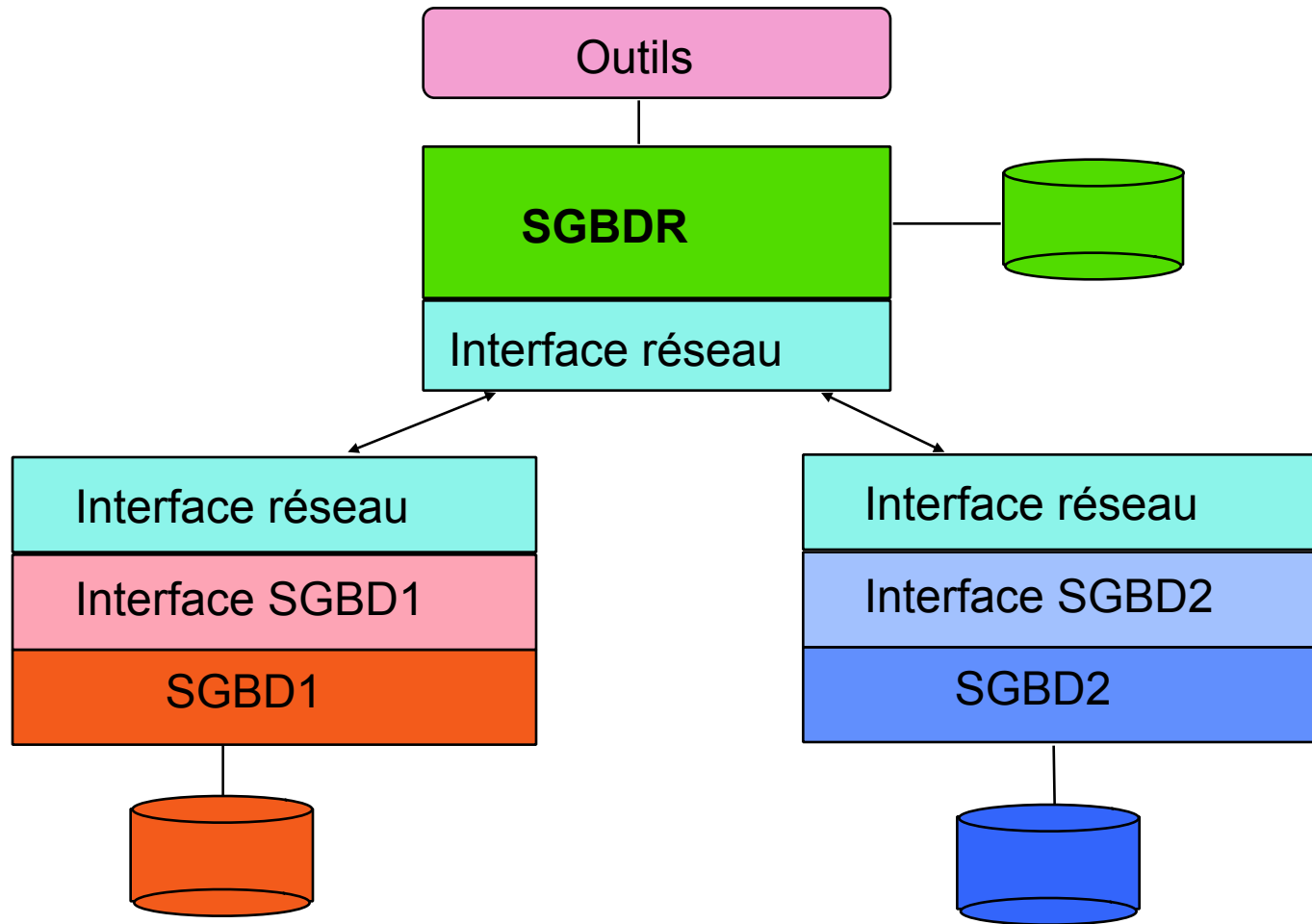
A light blue oval with a black border, containing the text "Schéma d'allocation" in black text.

Plan d'exécution réparti

Exemple d'évaluation simple



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

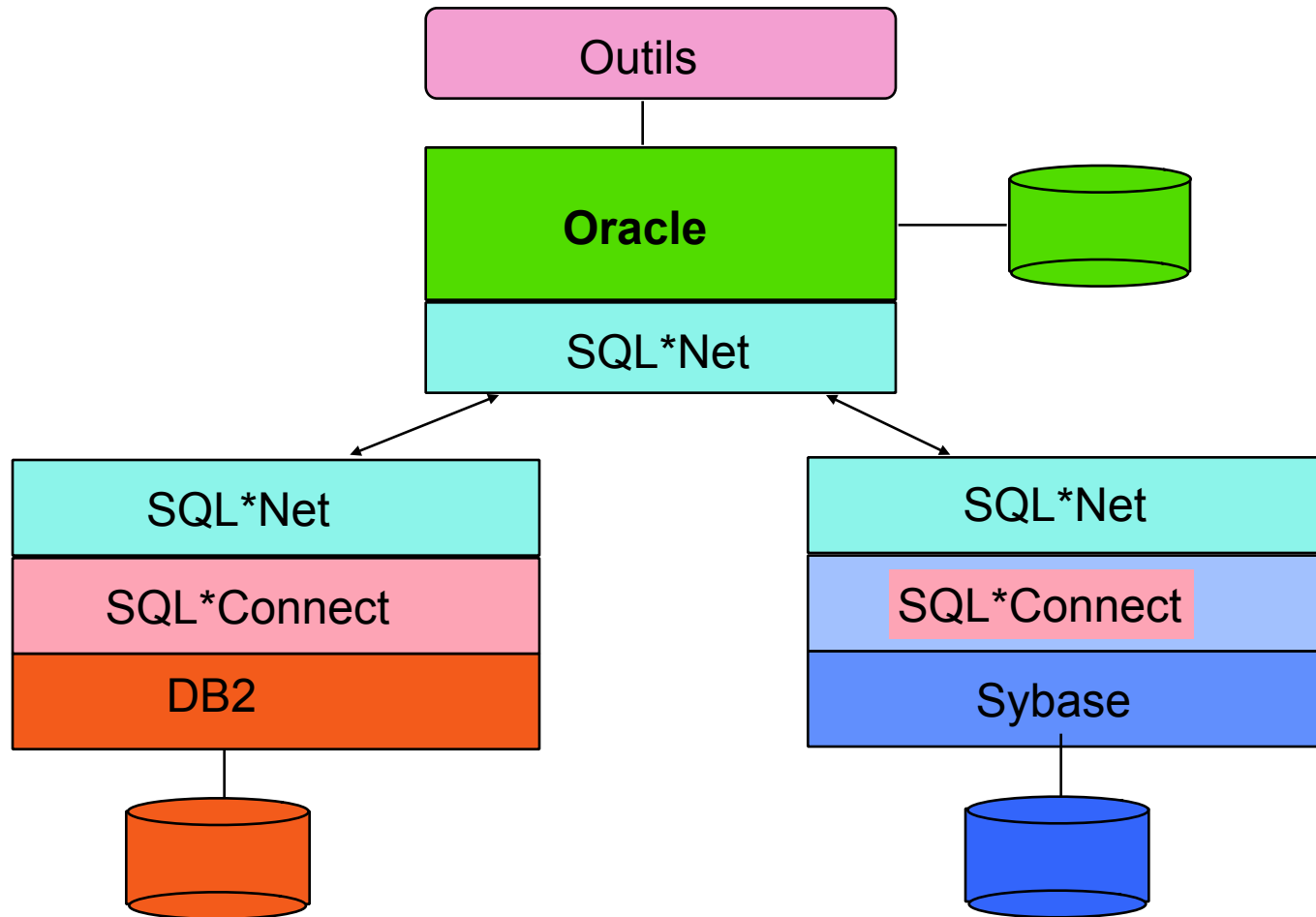
➤ Lien à une table dans une BD distante spécifié par :

- 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 empLien`
- `CONNECT TO user1`
- `IDENTIFIED BY mdp1`
- `USING 'master'`

Oracle/Star : architecture



Difficultés des bases réparties

➤ Choix et maintien des fragments

- En fonction des besoins des applications
- Heuristiques basées sur l'affinité d'attributs et le regroupement

➤ Disponibilité des données

- Dépend de la robustesse du protocole 2PC; implique une grande fiabilité du réseau et des participants

➤ Echelle

- Le nombre de sessions simultanées est limité par l'architecture 2-tiers; grande échelle nécessite un moniteur transactionnel

Bases de données répliquées

1. Intérêt de la réplication
2. Diffusion synchrone et asynchrone
3. Réplication asymétrique

Définitions

➤ Réplica ou copie de données

- Fragment horizontal ou vertical d'une table stockée dans une base de données qui est copiée et transféré vers une autre base de données
- L'original est appelé la copie primaire et les copies sont appelées copies secondaires

➤ Transparence

- Les applications clientes croient à l'existence d'une seule copie des données qu'ils manipulent :
 - soit « logique » dans le cas d'une vue
 - soit physique dans le cas de vues matérialisées

Les avantages de la réplication

➤ Amélioration des performances

- lecture de la copie la plus proche
- évitement du goulot d'étranglement du serveur unique

➤ Amélioration de la disponibilité

- lors d'une panne d'un serveur, on peut se replier sur l'autre
- Disponibilité = $1 - \text{probabilité_panne}^N$
 - probabilité de panne = 5% et 2 copies => disponibilité = 99.75%

➤ Meilleure tolérance aux pannes

- possibilité de détecter des pannes diffuses

Les problèmes de la réplication

➤ Convergence

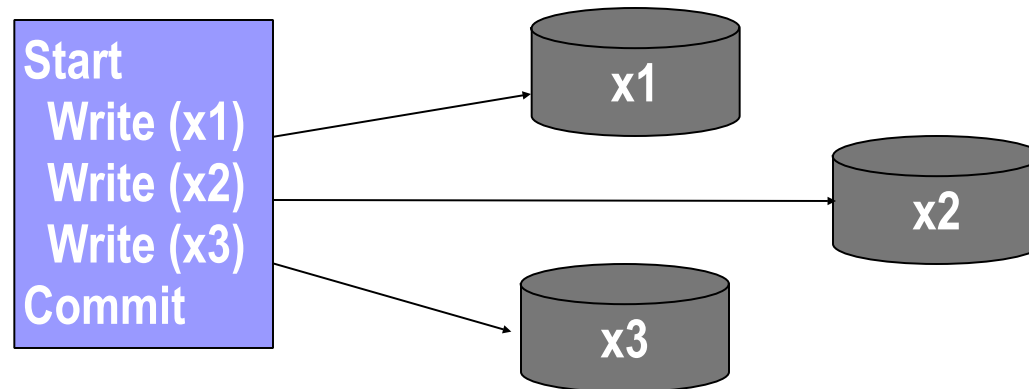
- les copies doivent être maintenues à jour
- à un instant donné, elles peuvent être différentes
- mais elles doivent converger vers un même état cohérent où toutes les mises à jour sont exécutées partout dans le même ordre

➤ Transparence : le SGBD doit assurer

- la diffusion et la réconciliation des mises à jour
- la résistance aux défaillances

Diffusion synchrone

- Une transaction met à jour toutes les copies de toutes les données qu'elle modifie.
 - + mise à jour en temps réel des données
 - trop coûteux pour la plupart des applications
 - pas de contrôle de l'instant de mise-à-jour



Diffusion asynchrone

- Chaque transaction met à jour une seule copie et la mise-à-jour des autres copies est différée (dans d'autres transactions)
- Réplication asymétrique : toutes les transactions mettent à jour la même copie
- Réplication symétrique : les transactions peuvent mettre à jour des copies différentes
 - + mise-à-jour en temps choisi des données
 - + accès aux versions anciennes puis nouvelles
 - l'accès à la dernière version n'est pas garanti