

Bases de données avancées

Introduction aux Bases de Données Réparties

Pr. Omar El Beqqali

Omar.el-beqqali@insa-lyon.fr

oelbeqqali@fsdmfes.ac.ma

<http://www710.univ-lyon1.fr/~obekkali>

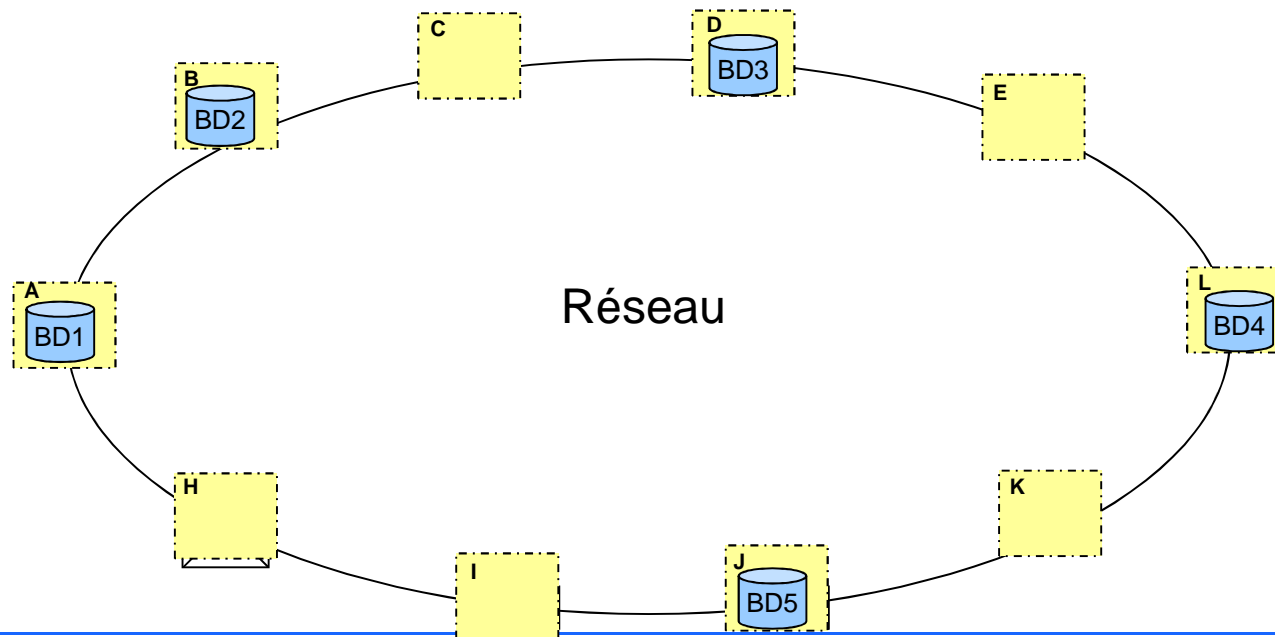
Plan

- Introduction**
- BDR : avantages**
- Fragmentation**
- Réplication: vues matérialisées**
- Mise en œuvre : TPs**

Bases de données distribuées : Définitions

(Distributed Database)

- **Ensemble de bases de données stockées sur plusieurs sites (machines) communiquant via un réseau.**
- **Chaque nœud est une BD gérée par un SGBD local ou serveur de base de donnée.**



Bases de données distribuées (suite)

(Distributed Database)

- ✓ La BD à laquelle un utilisateur est directement connecté, est la BD *locale*.
- ✓ Les autres BD accédées par l'utilisateur sont dites *distantes*.
- ✓ Apparence d'une unique BD
- ✓ Les BD peuvent être simultanément accédées et mises à jour

BD-R : Avantages (1)

- placement des données dans un site proche de leur utilisation: simplicité, transparence aux réseaux
- données locales traitées localement
- Fiabilité plus grande (plusieurs machines indépendantes)
- sécurité, performance (traitements effectués en parallèle)
- couplage de données appartenant à diverses institutions.

Avantages (2)

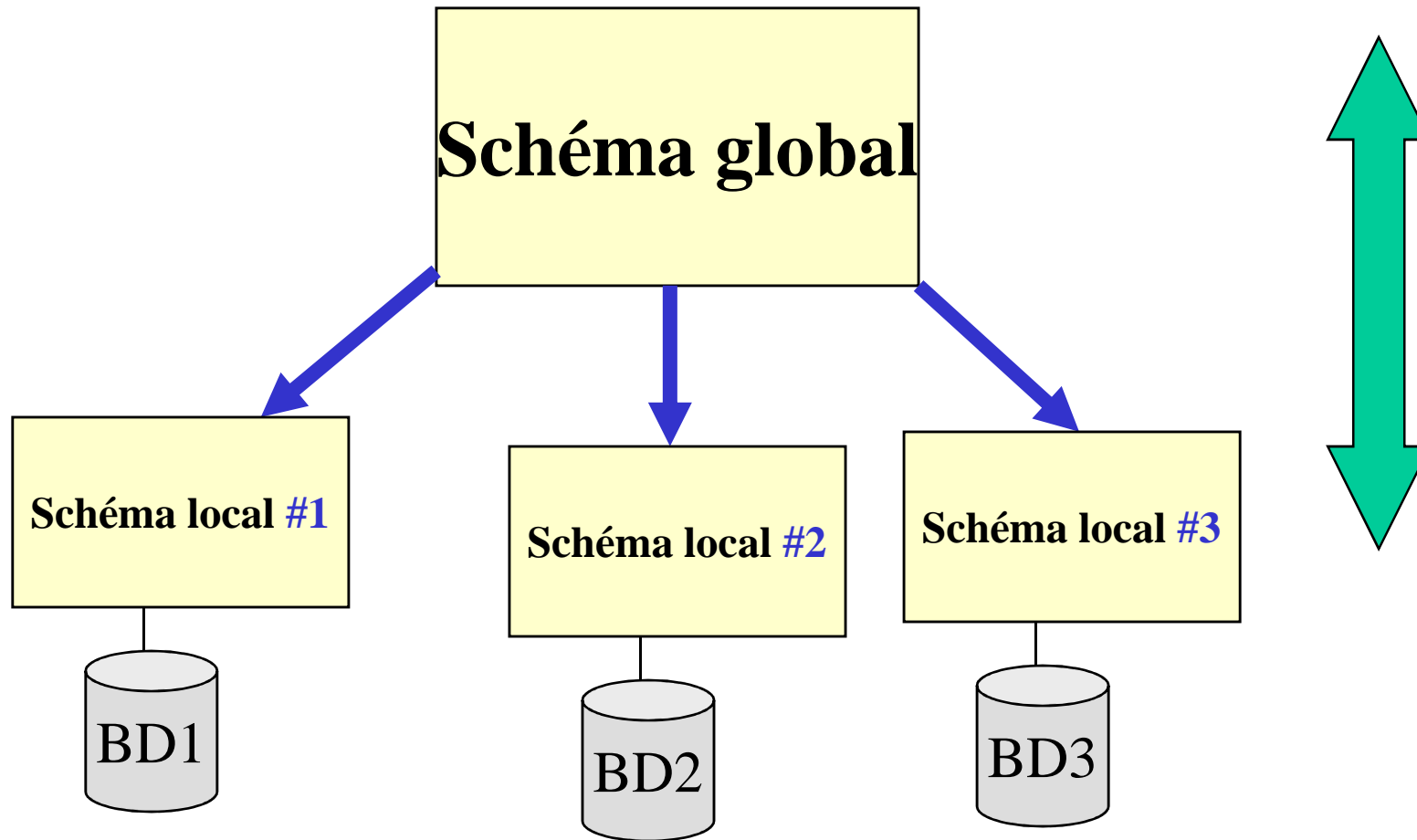
- Partage des responsabilités
 - Construction progressives, disponibilité accrue si duplication sur plusieurs sites.
-

Cas : Entreprises ou organismes ayant des agences géographiquement distribuées:

- *banques*
- *fabrication (plusieurs usines)*
- *systèmes de réservation de compagnie aériennes...*

Décomposition d'un schéma 1/2

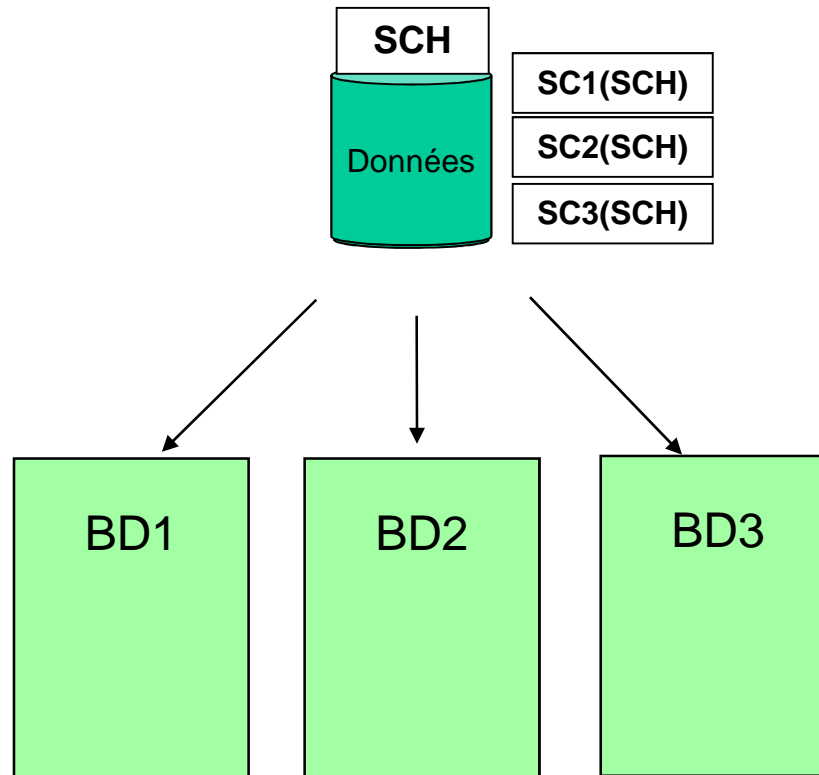
(bases de données distribuées)



CONCEPTION ASCENDANTE OU BOTTUM UP DESIGN : allocation fragments/sites

Décomposition d'un schéma 2/2

(bases de données distribuées)

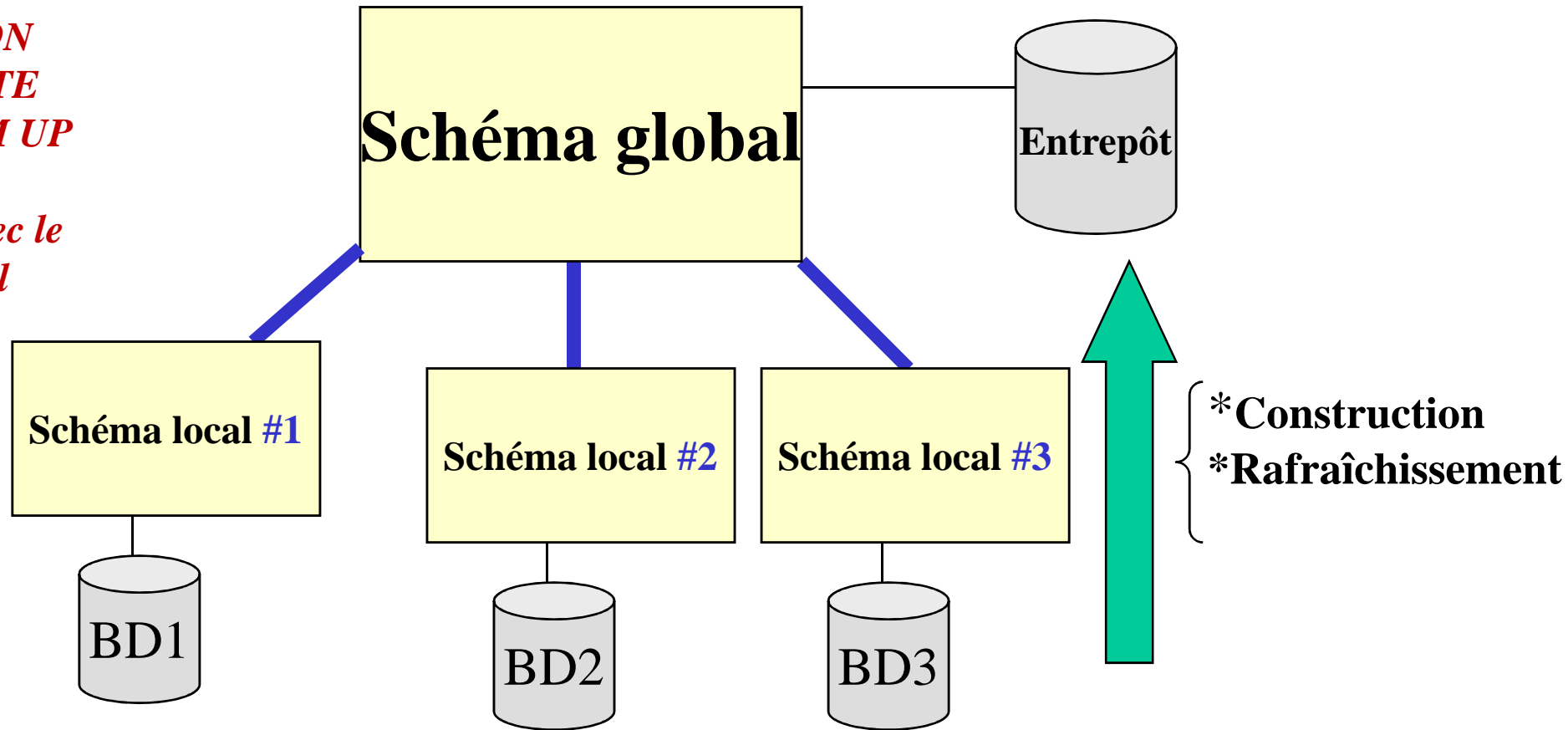


CONCEPTION DESCENDANTE OU TOP DOWN DESIGN : allocation fragments/sites

Data Warehouse :Entrepôt de données

(OLAP : On Line Analytical Processing)

**CONCEPTION
ASCENDANTE
OU BOTTUM UP
DESIGN :**
*cohérence avec le
schéma global*



***Data Warehouse : Collection de données historisées, intégrées à partir de BD, organisées
Pour offrir les informations nécessaires à la prise de décision***

Conception de bases de données réparties

Etapes

- 1. Normalisation**
- 2. Fragmentation horizontale et verticale**
- 3. Duplication (réplication)**

Fragmentation

Les données distribuées peuvent correspondre à une fragmentation (résultat d'une requête SQL)

- Fragmentation horizontale (F.H) d'une relation se fait par une opération de sélection qui place chacun de ses n-uplets dans un site. Par exemple une relation *Employés* peut être fragmentée selon la localisation de ses employés,

- Fragmentation verticale (F.V) divise une relation en fragments par projection sur ses attributs.

Exemple la relation *Employés* peut être fragmentée de la façon suivante:

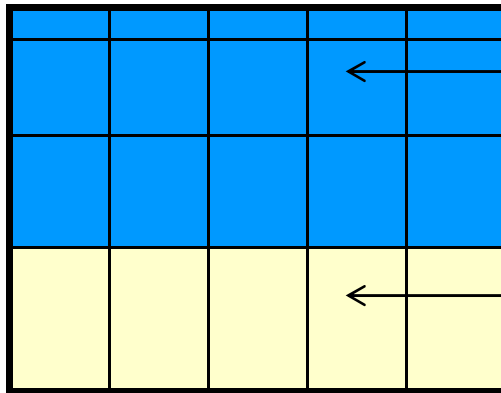
farg1 (numéro d'employé, le nom d'employé et son adresse)

frag2 (numéro d'employé, son salaire et son directeur).

- Fragmentation mixte (F.M) : FH & F.V

Fragmentation (suite)

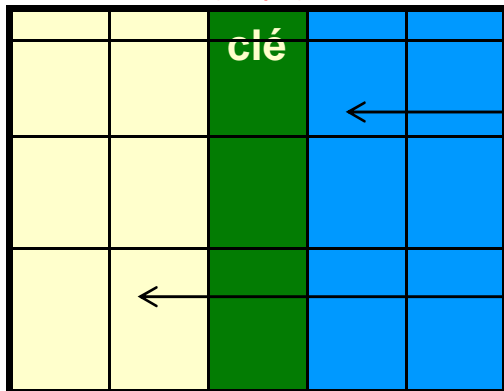
F.H.



Site A

Site B

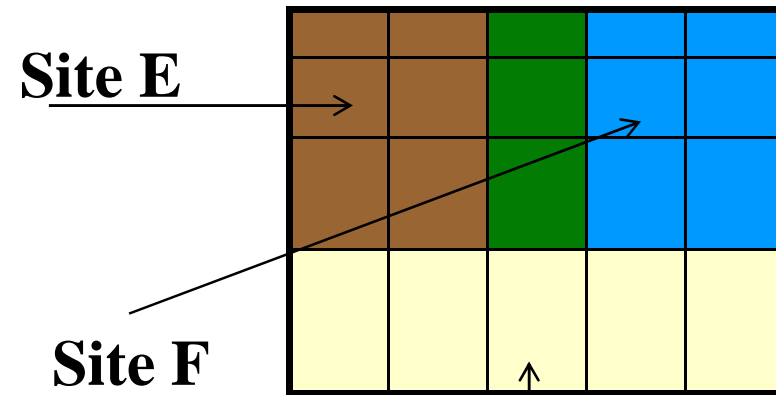
F.V.



Site C

Site D

F.M.



Site E

Site F

Site G

Les clés sont présentes sur les 2 sites

Réplication

La réplication permet de favoriser l'accessibilité aux données

Réplication => données dupliquées en plusieurs endroits.

Avantages :

- disponibilité, performance, coût
- organisation/adaptation/déconcentration
- multiplie l'accès local (disponibilité et temps de réponse)
- accroît la sûreté des données en autorisant :
 - des accès alternatifs, des sous-ensembles sélectionnés de données**
- réduit la charge du réseau
- décharge sites maîtres et sites de vues matérialisées maîtres

La réplication dans les bases de données réparties : Vue matérialisée

- Les copies ou **réplicats** (partiels ou totaux) sont appelées des *vues matérialisées / clichés / snapshots*.
- Les vues matérialisées peuvent être à leur tour répliquées. Les tables (respectivement vues matérialisées) origines sont appelées *tables maîtres* (respectivement *vues matérialisées maîtres*).

*Le SGBD doit assurer la **convergence des copies** vers un même état et offrir la **transparence de gestion** aux utilisateurs. Les clients doivent croire à l'existence d'une seule table.*

***En résumé** : le SGBD doit assurer la diffusion des mises à jour aux copies et le choix de la meilleure copie lors des accès (rafraîchissements de vue matérialisée)*

Nomination des BD et des objets

- Nom de service **SERVICE_NAMES** pouvant intégrer plusieurs instances et/ou nœuds égal par défaut au nom global de base de données
- Nom global de BD = nom de la BD + nom du domaine
[réseau] **DB_NAME + DB_DOMAIN**

(Descripteur de connexion)

- Nom global d'objet

Schéma.Objet@NomGlobBD

Ex : **Scott.Dept@Sales.ACME.com**

- Nom de service / identificateur de connexion *sales* :

CONNECT Scott/tiger@sales

DATABASE LINKs

Pour pouvoir transférer des données au travers d'une base de données distribuée, un administrateur doit créer tous les *DATABASE LINKs*

(liens de base de données nécessaires, ou chemins de communication d'une base de données à une autre base de données)

CREATE PUBLIC DATABASE LINK.....(voir TPs)

Accès aux liens BD créés :

SELECT * FROM USER_DB_LINKS;

Accès aux données distantes :

Schéma.Objet@NomGlobalBD

Ex : select * from scott.emp@bd1.miage.com

VUES MATERIALIZED SEES

Création

```
CREATE MATERIALIZED VIEW EmpCasa  
REFRESH FAST NEXT SYSDATE + 7  
AS SELECT * FROM Employee@bdCasa
```

Ex : `CREATE SNAPSHOT PersDepts REFRESH NEXT SYSDATE + 1/480`
`AS SELECT Nom, Prenom, NomDept`
`FROM Personnes@BD2 P, Departements@BD2 D`
`WHERE P.NoDept = D.NoDept AND NOT NomDept = 'AAA' ;`

REFRESH



spécifie comment et quand la vue matérialisée est rafraîchie

Transparence de l'accès :exemple

```
CREATE VIEW BLOCK AS
```

```
SELECT * FROM TATA.table1@bd1.gis1.com
```

```
UNION
```

```
SELECT * FROM TOTO.table2@bd2.gis2.com
```

Requête distribuée

SNAPSHOTS / Vues matérialisées (suite)

Mode de rafraîchissement:

- **Fast** : Snapshot log utilisé
- **Complete** : toute la table est transférée
- **Force** : rafraîchissement rapide tenté
sinon Complet est exécuté

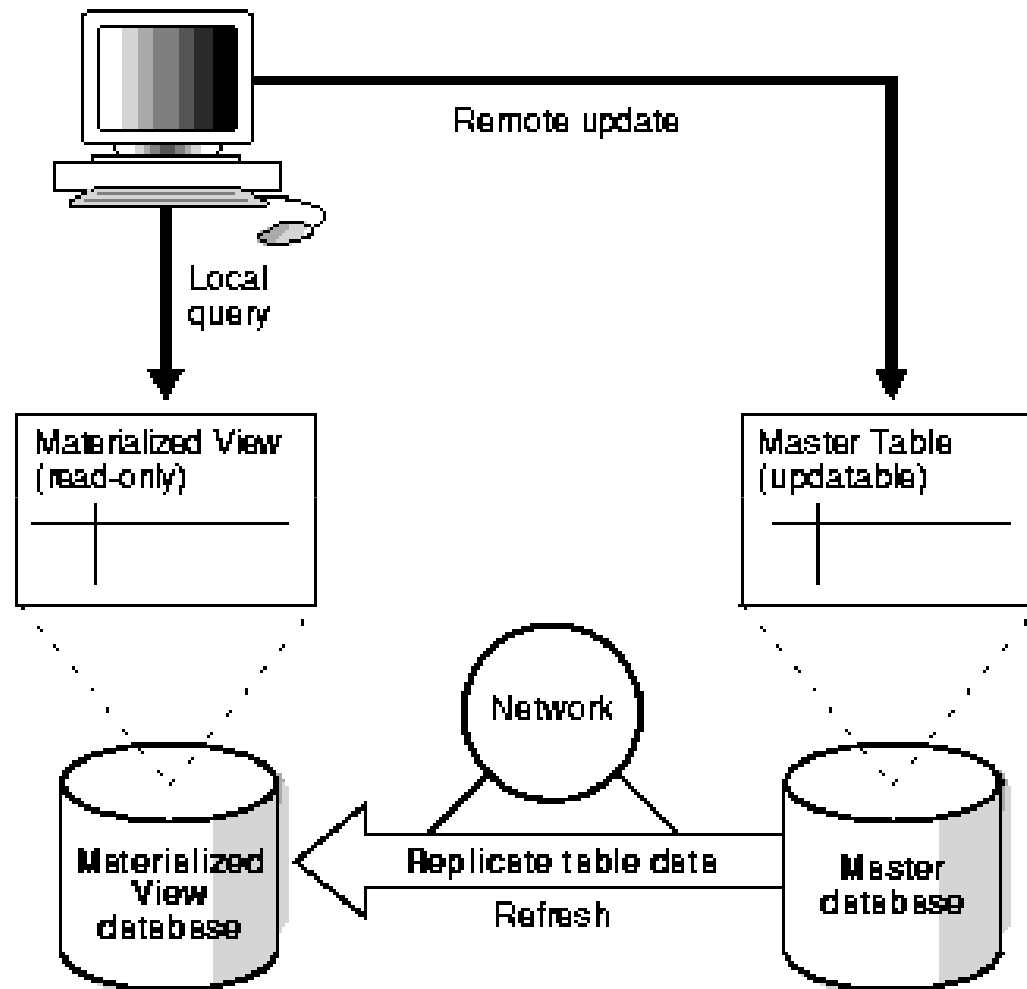
*A explorer les vues du DD utiles :

User_jobs[....failures....]

User_mviews user_db_links.....

Architecture

Client applications



TPs

- Gestion des transactions réparties (voir atelier en TP)**
- Conception de bases de données réparties**

Références :

- **G. Gardarin, "Bases de données : objet et relationnel", Eyrolles, 1999**
- **P. Valduriez, C. Oszu, "Principles of Distributed Database Systems", Prentice Hall,**
- **Web : <http://www.oracle.com>.....divers**