



UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE
FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE
LABORATOIRE DES SYSTÈMES INFORMATIQUES



Les Bases de Données Distribuées

Présenté par : Pr. Kamel Boukhalfa

Boukhalk@gmail.com
<http://boukhalfa.jimdo.com>

Plan

2

1. Introduction
2. Les bases de données distribuées
3. Le Système de Gestion de BD distribuées
4. Les techniques de répartition de données
5. Architecture de schémas d'une BDR
6. Fragmentation des données
7. Traitement des requêtes distribuées

INTRODUCTION

3

► Base de données

- ❑ Collection de données **cohérentes** et **structurées** (données **persistantes**) utilisées par des systèmes informatiques.
- ❑ **Collection** de données fortement **structurées** et **persistantes** dont la structure est définie dans un **schéma** au moyen d'un langage de définition de données.
- ❑ Les données et le schéma sont gérés à l'aide d'un logiciel appelé **système de gestion de bases de données (SGBD)**.

INTRODUCTION

4

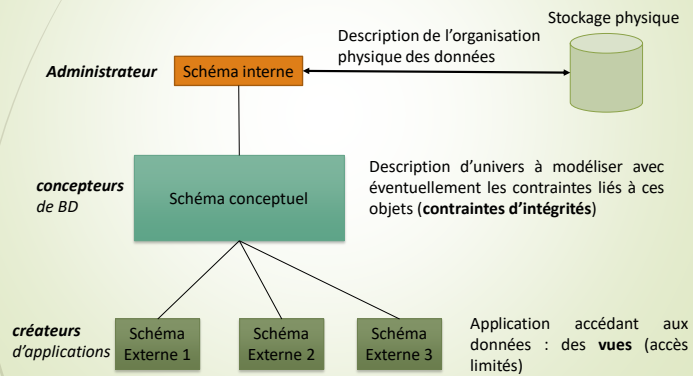
► Système de gestion de base de données (SGBD)

- ❑ Un ensemble de **logiciels informatiques** qui sert à la **manipulation** des bases de données.
- ❑ Il sert à effectuer des opérations ordinaires telles que **consulter, modifier, construire, organiser, transformer, copier, sauvegarder** ou **restaurer** des bases de données.
- ❑ Il est souvent utilisé par d'autres logiciels ainsi que les **administrateurs** ou les **développeurs**.

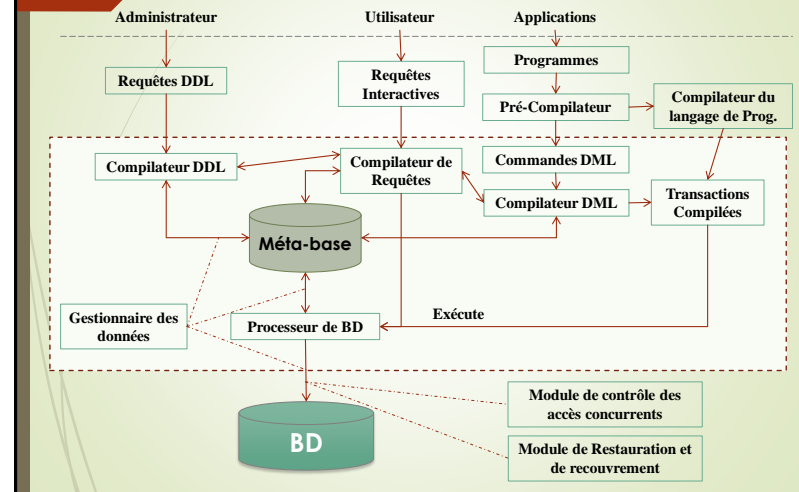
INTRODUCTION

5

- Conceptuel (Logique), Interne et Externe



Architecture interne d'un SGBD



Evolution des Architectures de BD

7

Architecture centralisée

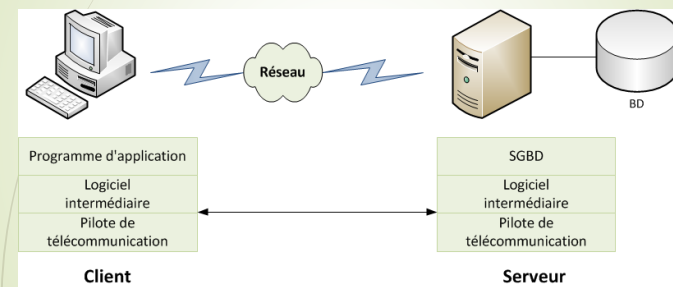
- ❑ Programme d'application et SGBD sur même machine (même site)
- ❑ Premiers systèmes

Architecture du type client-serveur

- ❑ Programme d'application = *client*
 - ❑ Interface (IHM) + traitement du domaine d'application
- ❑ SGBD = *serveur de données*
 - ❑ machines (sites) différentes

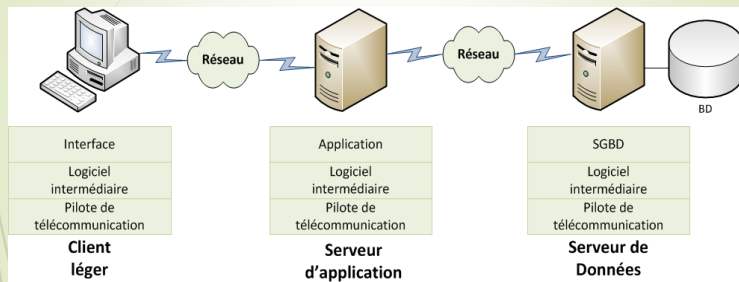
Architecture client-serveur

8



Architecture à trois tiers

9

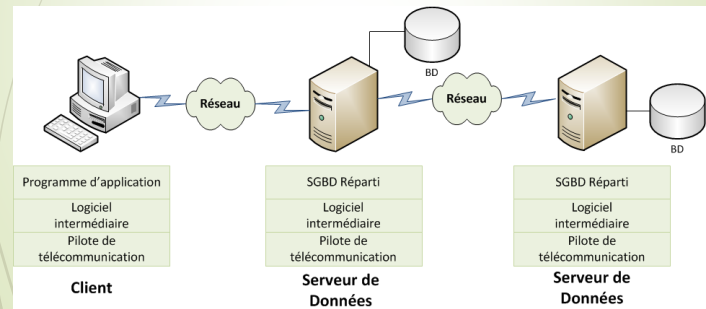


Base de données répartie, distribuée

10

▀ SGBD réparti

- ▣ Répartition des données de manière transparente
- ▣ Objet du module BDA (deuxième semestre)



11

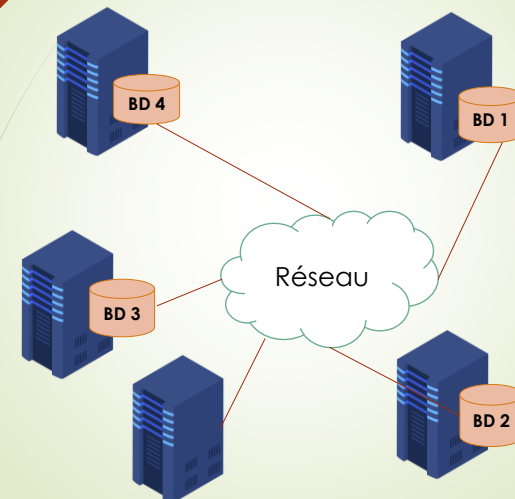
Bases de données distribuées

- Evolution de la technologie des bases de données
- Expérimentation de cette technologie sur un grand nombre d'applications
- Développement de la technologie réseaux :
- Connexion d'ordinateurs
- Échange de données
- Partage de ressources
- Intégration de la technologie réseau et bases de données

➔ **Bases de données distribuées**

12

Exemple



13

Définition [Gardarin]

- Une base de données distribuée est un **ensemble de bases de données localisées** et gérées par des **sites différents** et apparaissant à l'utilisateur comme **une base unique**
- Une base de données distribuée est une **collection** de **multiple bases** de données **distribuées** sur un réseau informatique logiquement **interreliées**
- **Un client** d'un système de gestion de base de données distribuée est **une application** qui **accède** aux **informations distribuées** par les **interfaces** du système.

14

SGBD Distribué

- Un système de gestion de base de données distribuées est un système qui **gère** des **collections de BD** logiquement reliées, distribuées sur un réseau, en fournissant **un mécanisme d'accès** qui rend **la répartition transparente** aux utilisateurs
- Un système de gestion de bases de données distribuées est **l'application** qui permet la **gestion** de la base de données distribuée et rend la **distribution transparente** aux utilisateurs

15

Bases de données distribuées

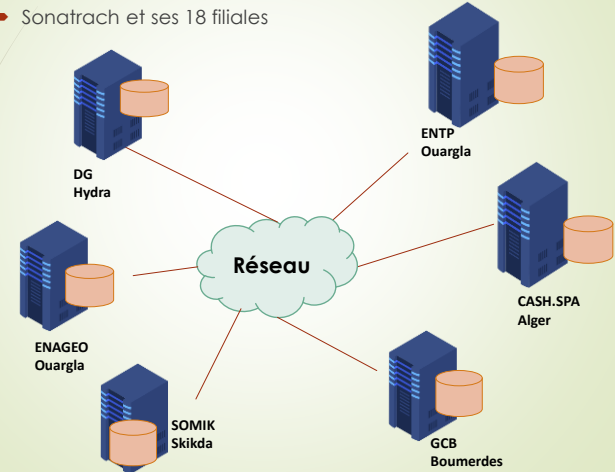
Différentes approches de la répartition

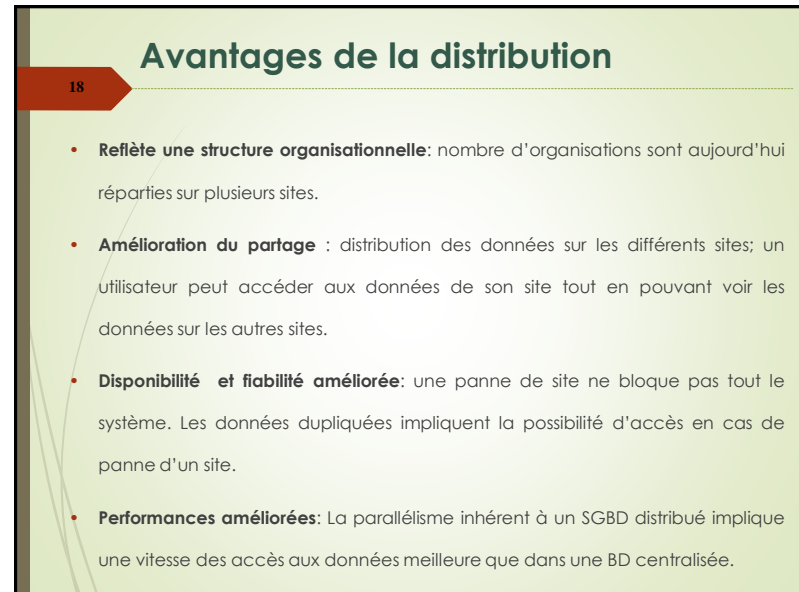
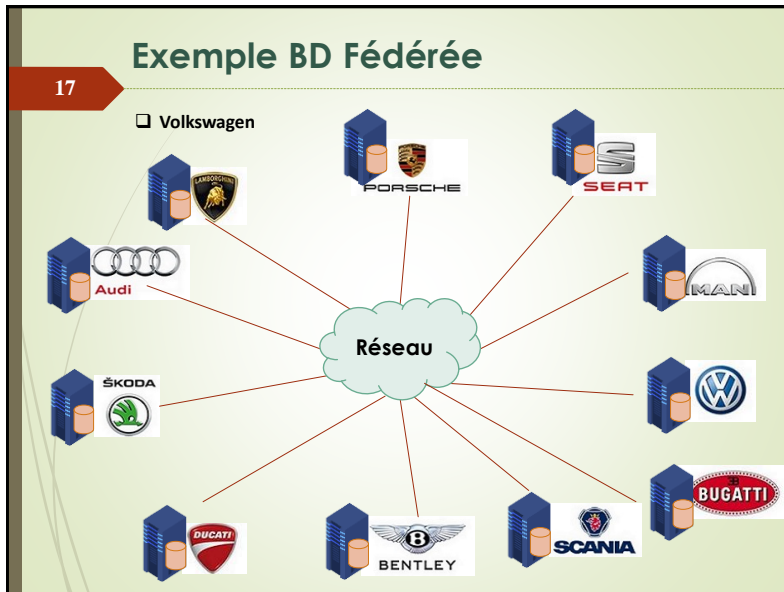
- Approche descendante
 - BD Distribuée
- Approche ascendante
 - BD intégrée
 - BD fédérée

16

Exemple BD Distribuée

➤ Sonatrach et ses 18 filiales





Avantages de la distribution

19

- **Flexibilité** : en ajoutant des sites au réseau le système réparti peut étendre la BD et enrichir les traitements pour de nouvelles applications
- **Autonomie** : la répartition permet aux différentes structures d'une même entreprise de conserver leurs spécificités (méthodes de travail, perceptions ...)
- **Economies**: La distribution est synonyme de petits équipements, donc la création de système informatique distribué est à coût inférieur si le même système était déployé sur un gros équipement.

INCONVENIENTS MAJEURS DE LA DISTRIBUTION

20

- Le manque d'expérience des équipes de développement et des utilisateurs finaux.
- Coûts de déploiement (de communication, d'adaptation des applications et des hommes).
- Conception et développement complexe : fragmentation, allocation des fragments, réplication etc.
- Contrôle d'intégrité plus difficile
- Sécurité: données dupliquées, réseaux

LE SYSTEME DE GESTION DE BD REPARTIES (SGBDR)

21

❑ Le SGBDR

- Un SGBDR est un SGBD assurant l'accès coordonné à des données hétérogènes stockées sur des sites différents via des requêtes.
- Une requête interrogeant une BD répartie est dite **requête répartie**. Elle se décompose en un ensemble de **sous-requêtes** sur les BD locales.

❑ Architecture fonctionnelle

3 composants majeurs :

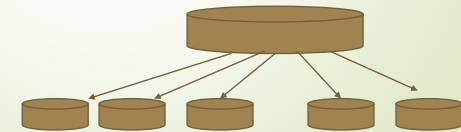
- SGBD local
- Gestionnaire de transactions
- Une composante de communication

CLASSIFICATION DES APPROCHES DE CONCEPTION D'UNE BDR

22

❑ Approche descendante

- ❑ Répartition d'une BD centralisée.
- ❑ Approche de décomposition descendante d'une BD centralisée en une collection de données appartenant **logiquement** au même système mais physiquement étalée sur plusieurs sites. La BDR est dite **homogène**.
- ❑ Définition d'une stratégie de duplication et fragmentation



CLASSIFICATION DES APPROCHES DE CONCEPTION D'UNE BDR

23

La démarche descendante comprend trois étapes

- la conception
- la fragmentation
- l'allocation
- La réplication

► Avantages

- favorise la croissance incrémentale en facilitant l'ajout d'un nouveau site dans le réseau et convient à une nouvelle BD

• Inconvénient

- une modification du schéma global peut entraîner la reprise tout le processus de conception

Les Techniques de fragmentation de données

24

□ Avantages de la fragmentation

- **Utile:** en général, les applications travaillent sur des vues qui sont des sous-ensembles de relations, qui peuvent être des unités de répartition.
- **Efficacité:** stockage de données à proximité du lieu de leur utilisation fréquente.
- **Parallélisme:** une transaction globale peut être décomposée en sous-transactions travaillant chacune sous un fragment, donc accroissement du parallélisme.

□ Inconvénients de la fragmentation

- Intégrité difficile puisque du fait de la fragmentation, les dépendances fonctionnelles sont aussi réparties sur les différents sites.

25

Les Techniques de répartition de données: la fragmentation

La fragmentation

- Fragmenter une relation R, c'est la diviser en un certain nombre de fragments r_1, r_2, \dots, r_n , dont chacun contient des données de R.

Règles de correction

- Complétude**: chaque élément de données appartenant à la relation globale appartient aussi à un ou plusieurs de ses fragments.
- Reconstruction**: reconstruire la relation globale à partir de ses fragments doit être toujours possible.
- Disjonction**: impose que les fragments sont disjoints dans le cas de la fragmentation horizontale.

26

La fragmentation de données

Définition

- Décomposer les objets de la BD (relation, index, vues) en un ensemble de morceaux appelés **Partitions**.

Fragmentation horizontale

- La table est fragmentée par rapport à ses instances en un ensemble de lignes.

Fragmentation verticale

- La table est fragmentée selon ses attributs en un ensemble de colonnes.

Fragmentation mixte

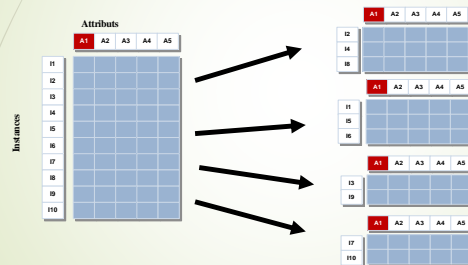
- La table est fragmentée horizontalement et verticalement.

Types de fragmentation

27

Fragmentation horizontale

- Décomposer les objets en un ensemble de lignes (instances)



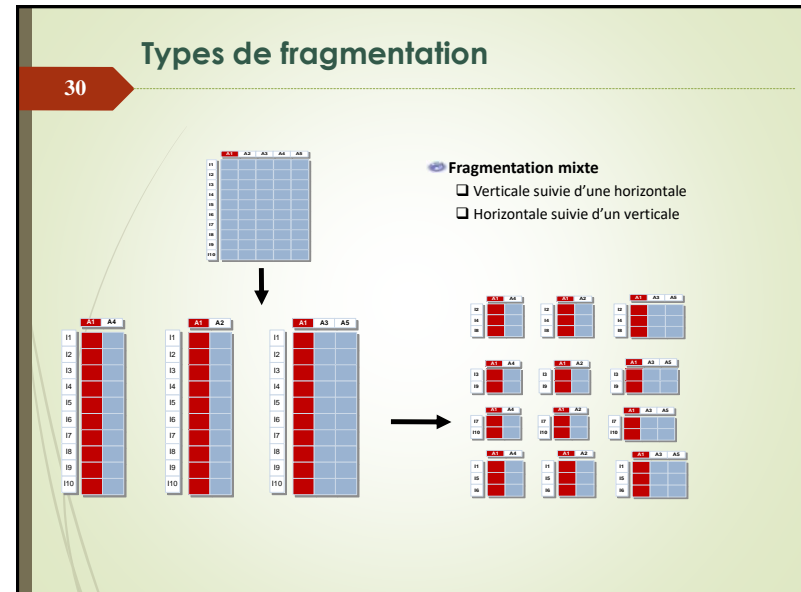
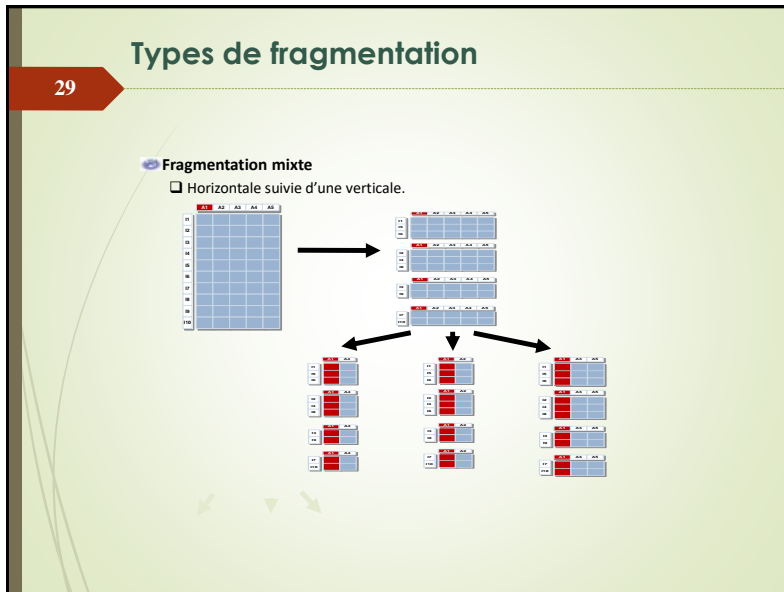
Types de fragmentation

28

Fragmentation verticale

- Décomposer les objets en un ensemble de colonnes (attributs).





Fragmentation Horizontale Primaire et dérivée (I)

31

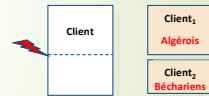
Fragmentation horizontale primaire (FHP)

- Fragmenter une table en utilisant les prédicats de sélection définis sur cette table

Prédicat : Attribut θ Valeur, $\theta \in \{=, <, \leq, >, \geq\}$ et valeur \in Domaine(Attribut).

- Exemple: Client (Client_id, Nom, Ville)

- Client₁ : $\sigma_{\text{Ville}=\text{Alger}}(\text{Client})$
- Client₂ : $\sigma_{\text{Ville}=\text{Béchar}}(\text{Client})$



Impact de la FHP sur les requêtes

- Optimisation des sélections



Fragmentation Horizontale Primaire et dérivée (II)

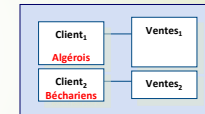
32

Fragmentation horizontale dérivée (FHD)

- Fragmenter une table (S) selon des attributs d'une autre table (T) : (existence de lien entre S et T)

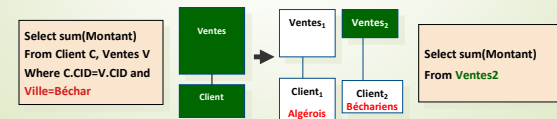
- Ventes(Client_id, Produit_id, Date, Montant)

- Ventes₁ = Ventes \times Client₁
- Ventes₂ = Ventes \times Client₂



Impact de la FHD sur les requêtes

- Optimisation de la jointure entre S et T



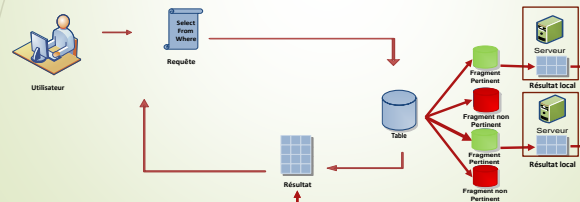
Intérêt de la fragmentation horizontale(I)

33

Améliorer la performance

1. Partition Elimination (Pruning)

- Elimination des partitions non pertinentes
- Possibilité d'exécution parallèle des sous requêtes



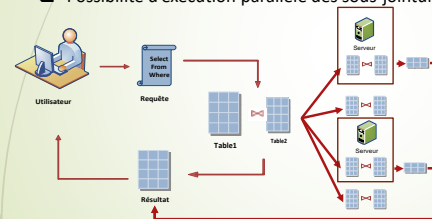
Intérêt de la fragmentation horizontale(II)

34

Amélioration de la performance

2. Partition Wise Joins

- ❑ Elimination des jointures non pertinentes
- ❑ Possibilité d'exécution parallèle des sous-jointures



Intérêt de la fragmentation horizontale(III)

35

Améliorer la managéabilité

- Fragmentation horizontale préserve le schéma logique
 - Toutes les opérations se font au niveau partition
 - Possibilité de manipulation individuelle ou collective des partitions
- Manipuler une partition à la fois.
 - Possibilité de définir des index locaux aux partitions
 - Existences de plusieurs Fonctions de manipulation des partitions

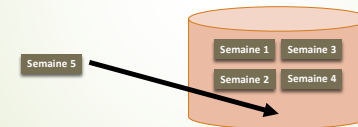
Fonction	Signification
ADD PARTITION	Ajouter une partition à une table déjà fragmentée
COLESCCE PARTITION	Redistribuer les tuples d'une partition dans les autres partitions
DROP PARTITION	Supprimer une partition ainsi que son contenu
EXCHANGE PARTITION	Convertir une table non fragmentée en une partition d'une autre table et inversement
MERGE PARTITION	Fusionner deux partitions dans une seule
SPUT PARTITION	Eclater une partition en deux partitions
TRUNCATE PARTITION	Vider une partition sans la supprimer

Exemple d'ajout de données

36

Alimentation hebdomadaire d'une table

- L'administrateur partitionne sa table par semaine
- L'alimentation de l'entrepôt consiste alors à ajouter une partition à la table
- Aucune autre partition ne sera touchée.

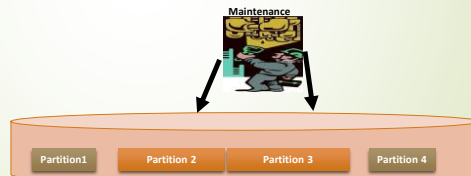


37

Intérêt de la fragmentation horizontale(IV)

Améliorer la maintenance et la disponibilité

- La manipulation au niveau partition permet
 - Cibler la maintenance sur certaines partitions
 - Minimiser le temps de maintenance
- La possibilité de stocker les partitions sur des emplacements différents permet
 - L'effet des pannes de disques est limité
 - Partitions saines toujours disponibles

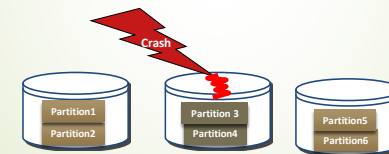


38

Améliorer la maintenance et la disponibilité

Améliorer la maintenance et la disponibilité

- La possibilité de stocker les partitions sur des emplacements différents permet
 - L'effet des pannes de disques est limité
 - Partitions saines toujours disponibles



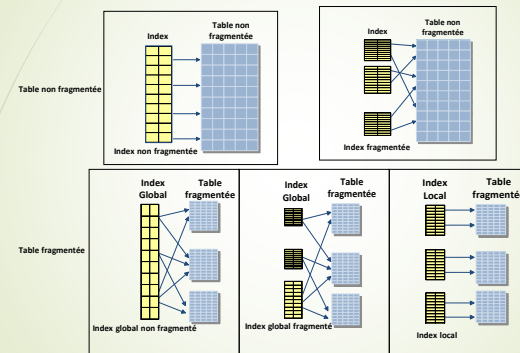
39

Fragmentation horizontale et index

- ☛ Un index peut être défini sur une table fragmentée ou non
- ☛ Un index défini sur une table fragmentée peut être fragmenté ou non.
- ☛ **Index global**
 - Peut être non fragmenté défini sur une table fragmentée
 - Peut être fragmenté où chaque partition de l'index référence plusieurs partitions de la table
- ☛ **Index local**
 - Equi-partitionné avec la table qu'il référence
 - Chaque partition de l'index référence une et une seule partition de la table

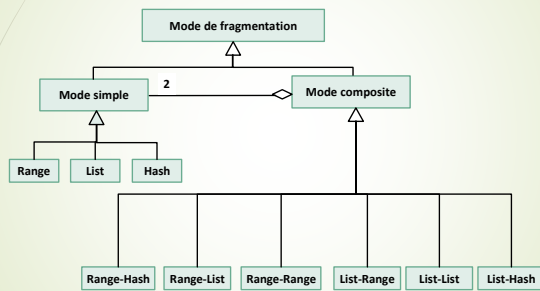
40

Fragmentation et index



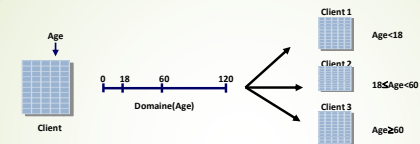
41

Modes de fragmentation



42

Fragmentation par intervalle « Range »

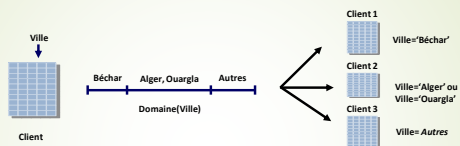


```

CREATE TABLE Client
(Client_id NUMBER(5),
Nom Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY RANGE (Age)
(
PARTITION Client_Moins_18 VALUES LESS THAN (18) TABLESPACE TBSMoins27,
PARTITION Client_18_59 VALUES LESS THAN (60) TABLESPACE TBS27-59,
PARTITION Client_60_et_Plus VALUES LESS THAN (MAXVALUE) TABLESPACE TBSPlus60
);
  
```

Fragmentation par liste « List »

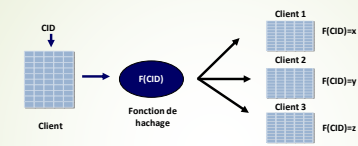
43



```
CREATE TABLE Client
(Client_id NUMBER(5),
Nom Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY List (Ville)
(
PARTITION Client_Béchar VALUES ('Béchar') TABLESPACE TBSBECHAR,
PARTITION Client_Alg_Oua VALUES ('Alger','Ouargla') TABLESPACE TBSALGOUA,
PARTITION Client_Autres VALUES (DEFAULT) TABLESPACE TBSAUTRES
);
```

Fragmentation par hachage « Hash »

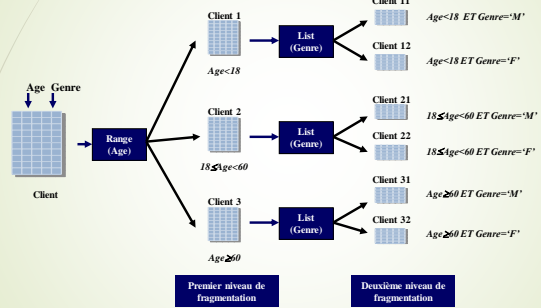
44



```
CREATE TABLE Client
(Client_id NUMBER(5),
Nom Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY Hash (CID)
(
PARTITIONS 3
STORE IN (TBS1, TBS2, TBS3)
);
```

Modes de fragmentation composite

45



Modes de fragmentation composite

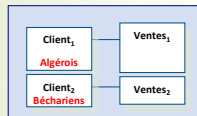
46

```
CREATE TABLE Client
(Client_id NUMBER(5),
Nom Varchar2(20),
Ville Varchar2(20),
Age Number(3),
Genre Varchar2(1))
PARTITION BY RANGE (Age)
SUBPARTITION BY LIST (Genre)
SUBPARTITION TEMPLATE(
SUBPARTITION Client1 VALUES ('M') TABLESPACE TBSMasculin,
SUBPARTITION Client2 VALUES ('F') TABLESPACE TBSFéminin )
(
PARTITION Client_Moins_18 VALUES LESS THAN (18),
PARTITION Client_18_59 VALUES LESS THAN (60),
PARTITION Client_60_Et_Plus VALUES LESS THAN (MAXVALUE)
);
```

Fragmentation dérivée par le mode Référence

47

- Fragmenter une table selon le schéma de fragmentation d'une autre table en utilisant le lien par clé étrangère.



```
CREATE TABLE Ventes
(Client_id NUMBER(5),
Time_id NUMBER(5),
Montant NUMBER(20),
CONSTRAINT order_items_fk
FOREIGN KEY(Client_id) REFERENCES Client(Client_id)
)
PARTITION BY REFERENCE(order_items_fk);
```

Allocation de données

48

Quatre stratégies

1. Allocation centralisée

Une seule base de données, un seul SGBD, sur un site et des utilisateurs répartis sur le réseau. Tous les sites sauf le site central doivent passer par le réseau pour tous les accès aux données.

→ Coûts de communication très élevés, fiabilité et disponibilité de données réduites

2. Allocation fragmentée ou partitionnée: partitionnement de la BD en fragments disjoints distribués chacun sur un site.

→ Coûts de stockage et de communication réduits, mais fiabilité et disponibilité de données réduites

2008/20
09

Allocation de données

49

3. Réplication complète

Une copie de la totalité de la BD sur chaque site.,

Fiabilité, disponibilité optimales, mais coûts de stockage et communication élevés à cause des mises à jour.

4. Réplication sélective

Combinaison de la centralisation, fragmentation, et réplication.

Tous les avantages de chacune des approches tout en évitant leur inconvénients;

2008/20
09

La réplication

50

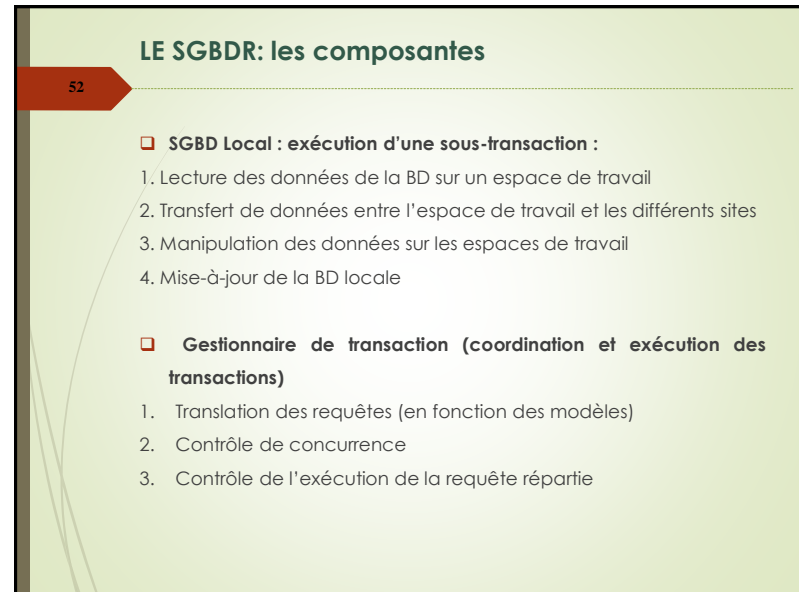
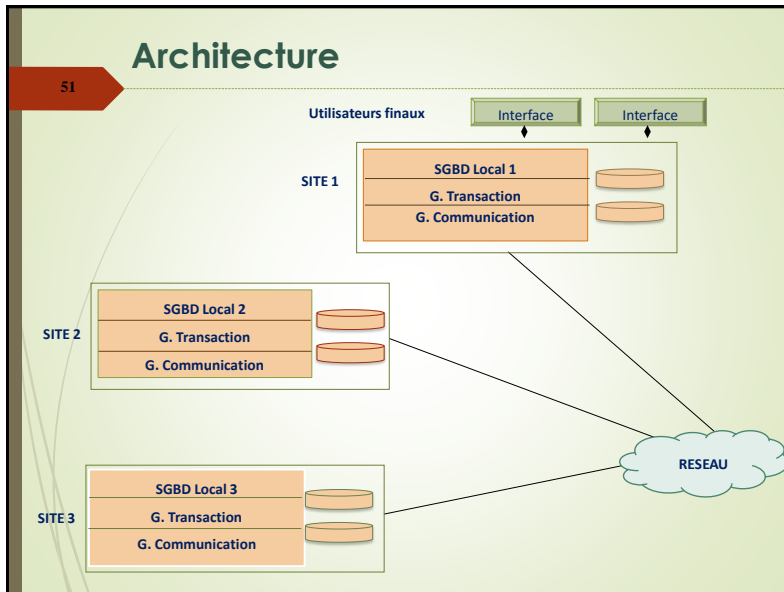
▣ Avantages :

- **Disponibilité des données** : une relation R peut être atteinte à partir de n'importe quel site, en cas d'avarie affectant un site donné.
- **Parallélisme des traitements** : lorsque la majorité des transactions relatives à une relation R ne mettent en jeu que des lectures, plusieurs sites peuvent travailler en parallèle sur cette relation.

▣ Inconvénients :

- Alourdissement des procédures de mise-à-jour : toute mise-à-jour doit être appliquée à l'ensemble des sites dépositaires de la relation pour garantir la consistance des répliques des relations.
- Augmentation du volume de la BD

2008/20
09



LE SGBDR: les composantes

53

□ Composante de communication (responsable de l'interconnexion de machines coopérantes)

1. Elle garantit la délivrance des messages entre les différentes machines sans perte d'information
2. Surveille l'atomicité des transactions et permet certains mécanismes de synchronisation.

LE SGBDR - Objectifs

54

1. Définition et manipulation des ensembles de données réparties sur différents sites comme un seul ensemble logique.
2. **Indépendance à la localisation**
 - Les données ne résident pas nécessairement sur le site utilisateur. Ce dernier ignore la localisation des données.
 - L'information concernant la localisation des données, est maintenue dans le *dictionnaire de données*, et consultée par le SGBD distribué pour déterminer la localisation des relations impliquées dans la requête des utilisateurs.

LE SGBDR - Objectifs

55

3. Extensibilité

- La capacité d'augmentation incrémentale, par introduction de nouveaux sites dans le réseau, avec un impact minimal sur les bases de données locales et les programmes d'applications existants.
- Le système de gestion de bases de données distribué doit être capable d'intégrer les nouveaux nœuds ajoutés au réseau.
- L'interface de l'utilisateur global doit être assez évolutive, pour permettre d'intégrer de nouvelles fonctionnalités sans détruire la gestion de l'interface existante.

LE SGBDR - Objectifs

56

4. Fiabilité et performance

- **Duplication transparente** de certaines données, de façon à permettre au système global de **router la requête** utilisateur à une **source alternative**, quand la source principale ne fonctionne plus.

5. Cohérence : le respect des contraintes d'intégrité

- Assurer la cohérence lors de la mise-à-jour des copies.
- Assurer la cohérence de la BD dans le cas de mise-à-jour concurrentes.
- Maintien de la cohérence de la BD en cas de panne.
- La cohérence est assurée grâce au concept de **transaction**: 3 types de transactions : **globale**, **sous-transaction** et **locale**

6. Sécurité : Contrôle d'accès et protection des données confidentielles.

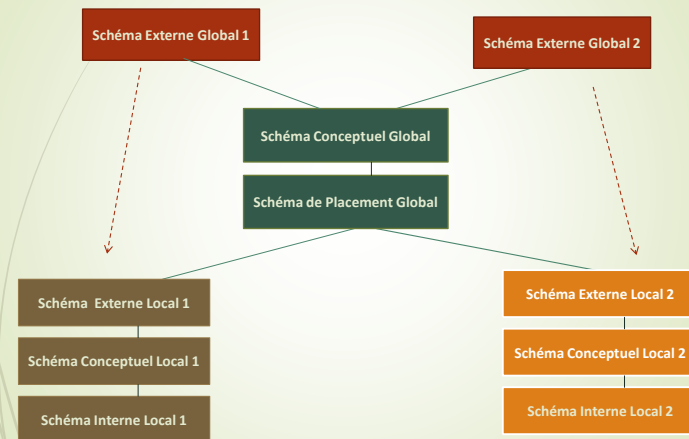
57

LE SGBDR - Les règles de transparence de Stonebraker

1. **Transparence de consultation** : une opération de consultation doit donner le même résultat quelque soit le site
2. **Transparence de mise-à-jour** : émise sur n'importe quel site, une mise-à-jour impliquant plusieurs sites, donnera toujours le même résultat.
3. **Transparence de schémas** : les schémas de tous les sites peuvent être rendus visibles sur n'importe quel site.
4. **Transparence de transaction** : exécution des transactions réparties concurrentes en maintenant la cohérence de la BD répartie.
5. **Transparence de copies** (indépendance à la duplication) : des copies nécessaires de données peuvent être effectuées sur des sites sans que cela ne soit visible à l'utilisateur.

58

Architecture de référence



Architecture de référence

59

❑ Schéma conceptuel global

- Description **unifiée** de la base de données répartie **indépendamment** de l'environnement réparti. Il est défini pour être la **synthèse** des besoins de l'ensemble des applications.

❑ Schéma de Placement

- **Dictionnaire de données** contenant les informations concernant la **localisation** provenant de la répartition (fragmentation et duplication).
- Capable de renseigner les emplacements de chaque donnée
- Dictionnaire **global** : contient des informations sur tous les sites.
- Dictionnaire **local** : contient des informations d'un seul site. Dans ce cas, le SGBD devra s'appuyer sur les dictionnaires des autres sites pour localiser une information.

Architecture de référence

60

❑ Schéma externe global

- **Vue** d'un groupe d'utilisateurs sur la BD répartie.

❑ Schéma interne local et schéma conceptuel local

- Identiques à ceux d'une base de données centralisées.

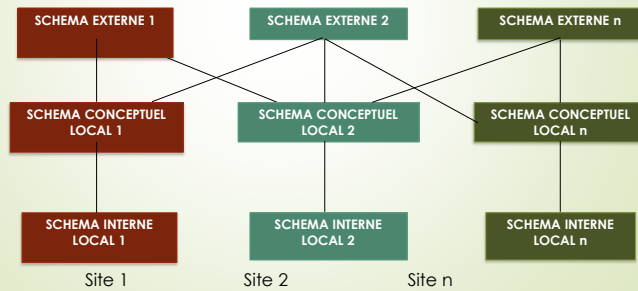
❑ Schéma local externe

- **Vue** de la base de données locales

Architecture de référence

61

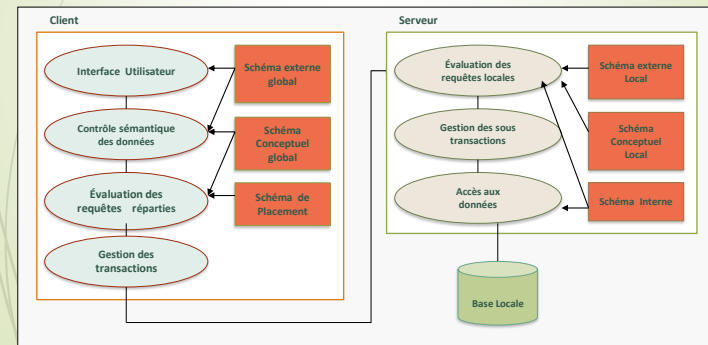
- L'existence de schéma conceptuel global permet de regrouper toutes les structures de données afin de les maintenir de façon centralisée.
- Architecture sans schéma global



ARCHITECTURE FONCTIONNELLE BASEE SUR UNE APPROCHE CLIENT SERVEUR

62

- Deux composants systèmes
 - Gérant d'applications sur le site (Client)
 - Gérant de données sur le site (Serveur)



ARCHITECTURE FONCTIONNELLE BASEE SUR UNE APPROCHE CLIENT SERVEUR

63

Site Client

☐ Interface utilisateur

- ▀ vérification de la conformité de la requête répartie avec le schéma externe.
- ▀ Restitution des résultats après exécution de la requête

☐ Contrôle sémantique des données

- ▀ vérification des contraintes d'intégrité sémantique
- ▀ contrôle d'autorisation

ARCHITECTURE FONCTIONNELLE BASEE SUR UNE APPROCHE CLIENT SERVEUR

64

☐ Évaluation de requêtes réparties

- ▀ A chaque requête répartie est construit un plan d'exécution réparti.
- ▀ Un plan d'exécution réparti est un programme découvrant la stratégie d'exécution optimisée d'une requête répartie. Une requête globale est décomposée en un ensemble de sous-requêtes locales, (génération d'un arbre syntaxique optimisé).
- ▀ **Optimisation** : choisir la meilleure localisation des copies de fragments dans le dictionnaire de données et ordonnancer les requêtes locales de façon optimisée.
- ▀ **Gestion de transactions**
 - ▀ assure le contrôle de concurrence réparti
 - ▀ validation des transactions
 - ▀ reprise de transactions

65

ARCHITECTURE FONCTIONNELLE BASEE SUR UNE APPROCHE CLIENT SERVEUR

Site serveur : Gérant des données

Évaluation de requêtes locales

- Vérification que les requêtes locales sont conformes au schéma externe local.
- Génération du plan d'exécution de la requête locale

Gestionnaire de sous-transactions

- Synchronise les sous-transactions d'une même transaction globale
- Validation des sous-transactions

Accès aux données

- fonctions normales de gestion de données : gestion de la mémoire et des méthodes d'accès aux données du site.

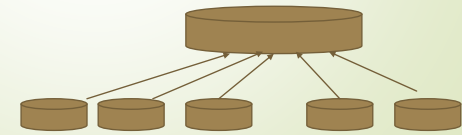
66

CLASSIFICATION DES APPROCHES DE CONCEPTION D'UNE BDR

BD virtuelle

Approche ascendante

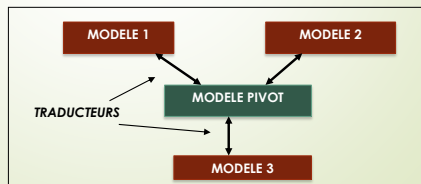
- Prise en compte d'un ensemble de BD existantes pour construire la BDR : intégration de BD en vue d'une seule dite **multibase** de données.
- La multibase est dite homogène si les SGBD gérant les BD locales sont les mêmes.
- La multibase est dite hétérogène dans le cas contraire.



CLASSIFICATION DES APPROCHES DE CONCEPTION D'UNE BDR

67

- ▀ La démarche ascendante comprend les étapes suivantes :
- ▀ sélection d'un modèle de données commun pour la description des schémas au niveau global appelé **modèle pivot** ou canonique.
- ▀ Transformation de chaque schéma local dans le modèle pivot.
- ▀ Intégration de l'ensemble des schémas locaux.



LES ETAPES DE DEVELOPPEMENT D'UNE FEDERATION

68

ETAPE 1 : Translation

- ▀ Construction du schéma composant à partir du schéma local en utilisant le modèle pivot.

ETAPE 2 : Définition

- ▀ Définition des schémas d'exportation à partir du schéma composant. C'est l'étape qui permet la construction des schémas d'exportation à partir des données que le gestionnaire de la fédération souhaite faire partager.

ETAPE 3 : Intégration

- ▀ Intégration des schémas d'exportation en un ou plusieurs schémas fédérés. C'est cette étape qui réalise la coopération entre les SGBD locaux.

ETAPE 4 : Attribution

- ▀ Définition des schémas externes pour chaque utilisateur ou chaque classe d'utilisateurs.

Synthèse

69

- Un système fédéré ne saurait s'envisager sans résoudre les problèmes de translation qui se posent lors du transfert de données d'un contexte local à un contexte global.
- Le principe général d'une telle translation repose sur la définition des caractéristiques intrinsèques des modèles et de développer ensuite des règles de correspondance permettant de générer un schéma composant dans le modèle pivot à partir d'un schéma source exprimé dans le modèle local.

MODELE COMMUN ET TRANSFORMATION DE SCHEMAS

70

Une BD fédérée supporte deux classes d'utilisateurs

- Les utilisateurs de la fédération qui manipulent les données par l'intermédiaire des schémas fédérés
- Les utilisateurs locaux qui accèdent uniquement aux données locales
- La conception de BDR fédérés hétérogènes est basée sur la notion de translation de schémas pour permettre la transformation depuis les structures de données depuis un modèle vers les structures de données d'un autre modèle appelé modèle commun.

CRITERES DE CHOIX DU MODELE COMMUN DE DONNEES

71

Objectifs d'un modèle commun

- Capturer les capacités structurelles et sémantiques des modèles locaux.
- Faciliter la communication entre utilisateurs des systèmes divers et parfois incompatibles.

Critères de choix

- **Complétude** : le modèle pivot doit permettre de décrire toutes les notions conceptuelles des autres modèles de données.
- **Facilité de traduction** : le modèle pivot doit pouvoir faciliter la translation entre le schéma local et le schéma composant.
- **Évolution du schéma** : Le modèle de données commun doit pouvoir assurer l'évolution d'un schéma avec un minimum de remise en cause de l'existant.
- **Expression de la répartition** : le modèle pivot doit permettre de décrire la répartition de données

TRANSFORMATION DE SCHEMAS

72

- **Principe général** : développement de langages de description permettant de définir les opérations à effectuer pour passer des valeurs d'une caractéristique à une autre valeur.

Premier modèle pivot : le modèle relationnel

➤ Les justificatifs étant les suivants :

- facilité de composition et de recomposition de relations (fragmentation facile)
- la représentation uniforme des attributs permet de retirer des informations sur la base de valeurs d'attributs communs.
- Les translations peuvent être exprimées dans les langages de manipulation rendant la décomposition de requêtes assez directe.



73

Evaluation des requêtes distribuées