Bases de Données Avancées

ECOLE NATIONALE POLYTECHNIQUE D'ORAN
Département de Génie des Systèmes Informatiques
Filière RT : 4ème année ingénieurs

M. SABRI
ram.sabri@gmail.com

Contenu du module

Partie I : Les Bases de Données Réparties

1. Besoins, Objectifs & Définitions
2. Conception d'une base de données répartie
3. Fragmentation
4. Schéma d'allocation
5. Réplication
6. Traitement & Optimisation de Requêtes Réparties
7. Gestion des Transactions Réparties
8. Les Architectures de Systèmes Parallèles

Partie II : Bases de données multimédia

1. Introduction
2. Gestion des bases de données multimédia
3. Espaces de représentation
4. Mesures de similarité
5. Évaluation
6. Description globale
7. Description locale
8. Techniques d'indexation
9. Recherche dans des espaces de grande dimension
10. Indexation Vidéo
11. Catégorisation des images
12. Indexation Audio

Bases de Données Avancées

Partie I : Les Bases de Données Réparties

1. BESOINS, OBJECTIFS & DEFINITIONS

L'évolution des techniques informatiques et des télécommunications permet d'adapter les outils informatiques et de télécommunications à l'organisation des entreprises (et non l'inverse).

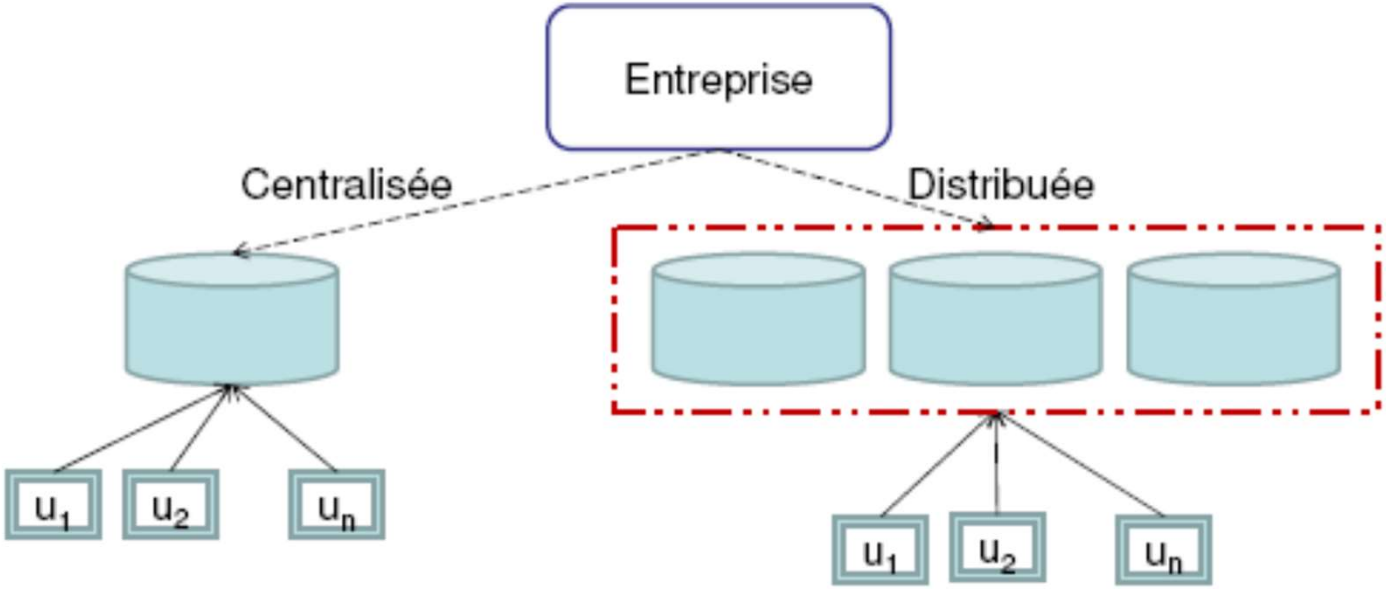
La puissance des micro-ordinateurs et des stations de travail,
la fiabilité et la souplesse des SGBD relationnels
et les performances des réseaux et des télécommunications

permettent d'envisager une répartition des ressources tout en préservant l'essentiel, c'est-à-dire la cohérence des bases de données.

1.1 Les pressions pour la distribution

- Augmentation du volume de l'information,
- Augmentation du volume des transactions.

→ Il devient impératif de décentraliser l'information.



1.2. Objectifs de la répartition

Les bases de données réparties ont une architecture plus adaptée à l'organisation des entreprises décentralisées.

→ Besoin de serveurs de BDD qui fournissent un bon temps de réponse sur des gros volumes de données.

- **Plus de fiabilité** : les bases de données réparties ont souvent des données répliquées. La panne d'un site n'est pas très importante pour l'utilisateur, qui s'adressera à autre site.
- **Meilleures performances** : réduire le trafic sur le réseau est une possibilité d'accroître les performances. Le but de la répartition des données est de les rapprocher de l'endroit où elles sont accédées. Répartir une base de données sur plusieurs sites permet de répartir la charge sur les processeurs et sur les entrées/sorties.
- **Faciliter l'accroissement** : l'accroissement se fait par l'ajout de machines sur le réseau.

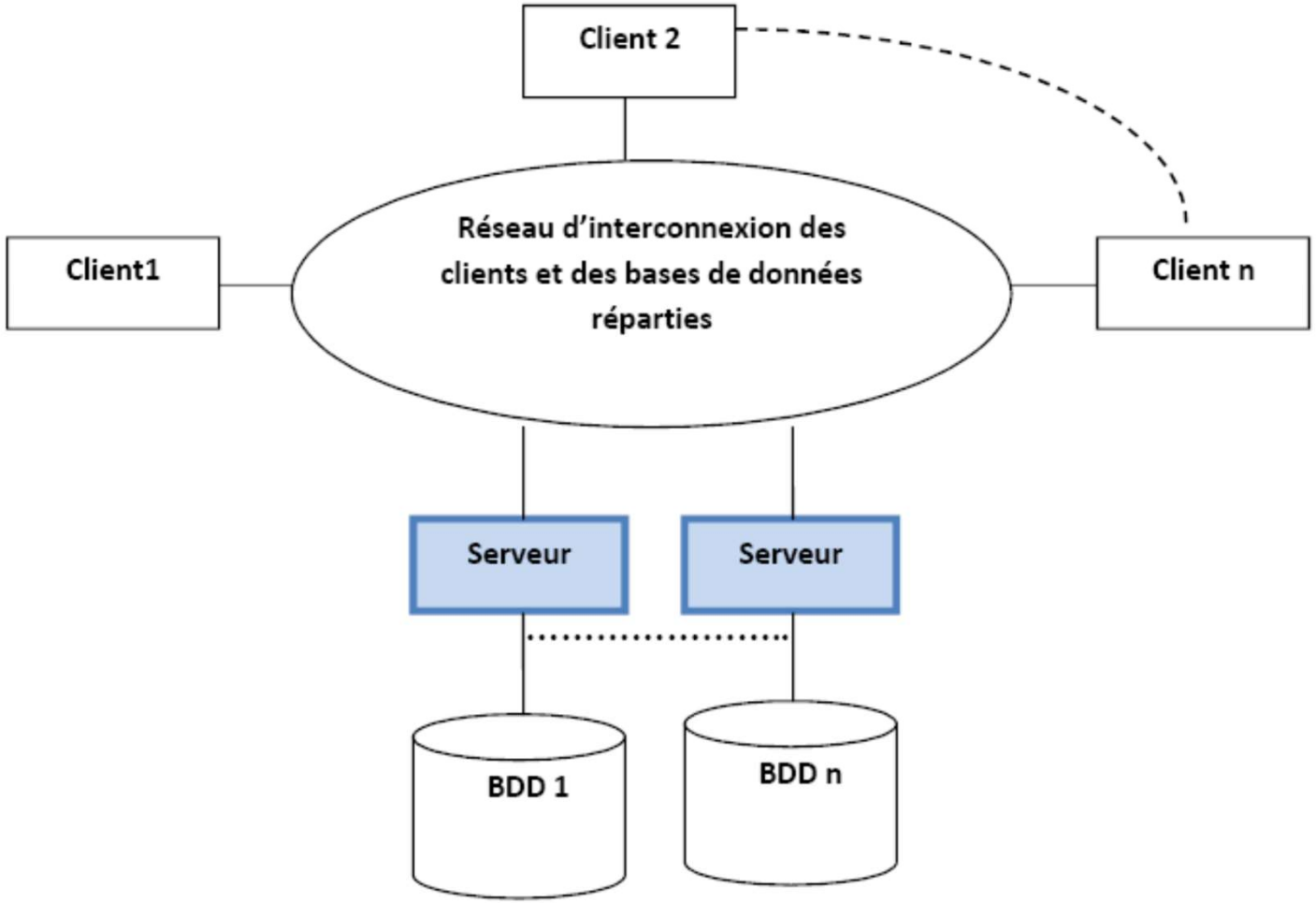
1.3. Définitions

Une base de données répartie (BDR) est un ensemble structuré et cohérent de données, **stocké sur des processeurs distincts**, et géré par un **système de gestion de bases de données réparties (SGBDR).**

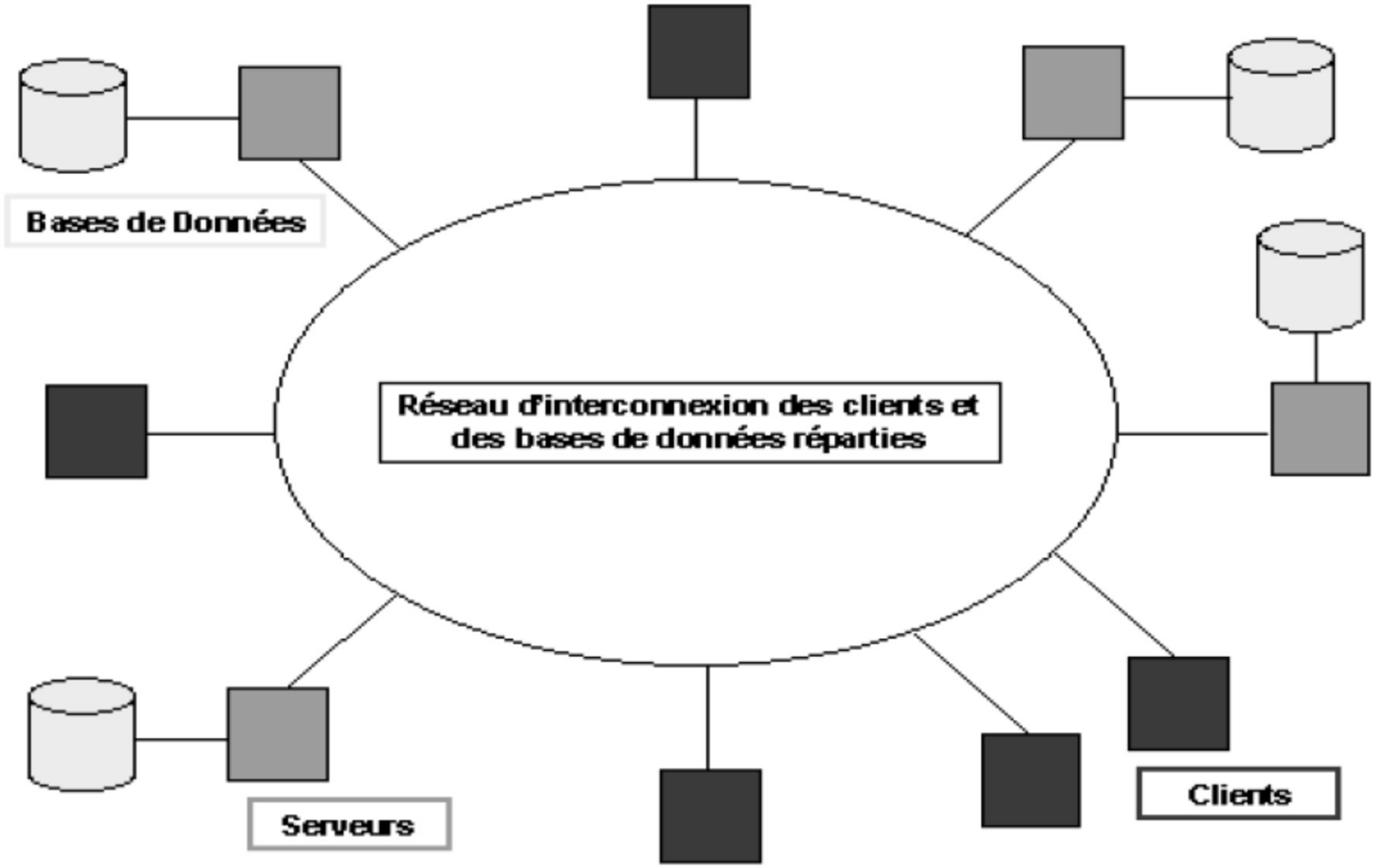
Le SGBDR repose sur un système informatique réparti qui est ***constitué d'un ensemble de processeurs autonomes appelés sites*** *(mini ou micro-ordinateurs, stations de travail, …)* ***reliés par un réseau de communication*** *(local ou public) qui leur permet d'échanger des données.*

Un SGBDR suppose en plus que les données soient stockées sur ***deux processeurs au moins****, ceux-ci étant dotés de leur SGBD propre. Ainsi, dans l'exemple d'une configuration constituée de trois processeurs interconnectés,* ***dont l'un est chargé de la gestion des données*** *(serveur base de données), la base de données n'est évidemment pas répartie bien que l'on soit en présence d'un système réparti.*

1.3. Définitions



1.3. Définitions



1.3. Définitions

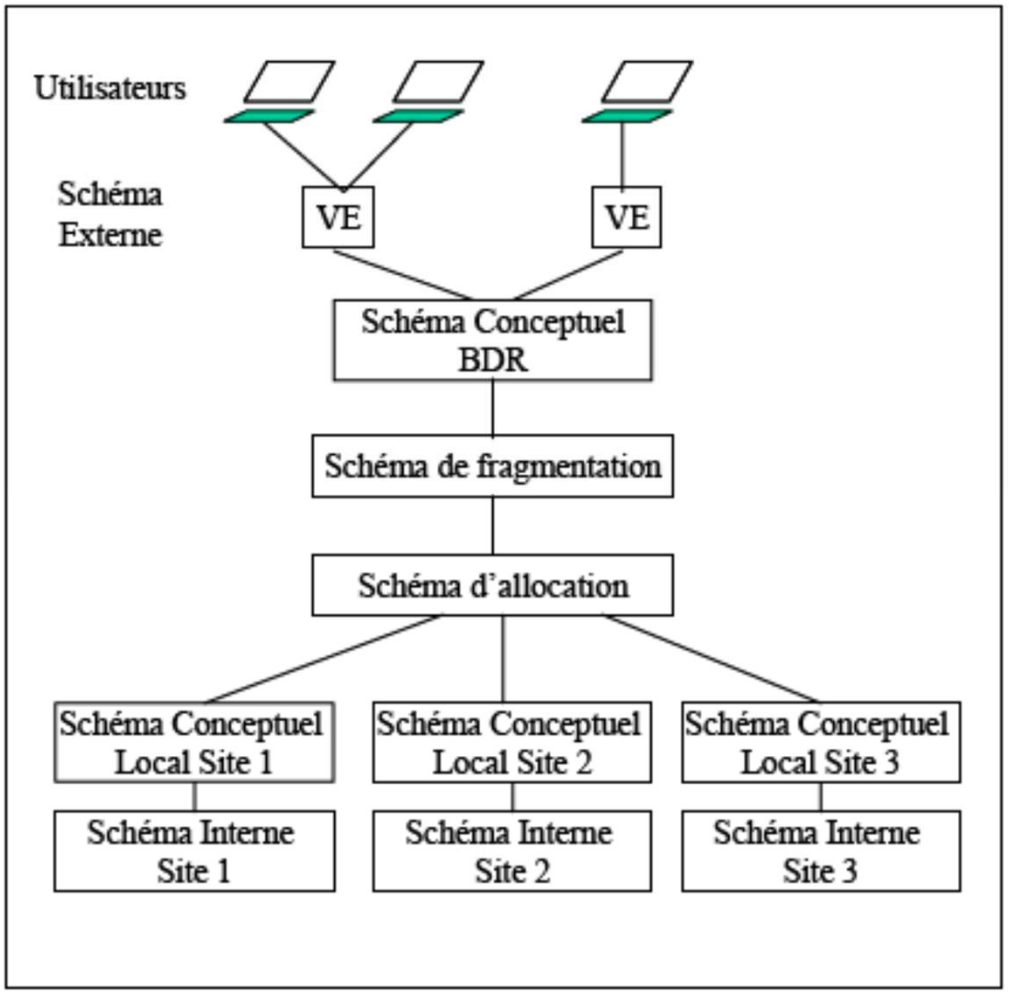
Une base de données répartie est une base de données **logique** dont les **données** sont **distribuées sur plusieurs SGBD et visibles comme un tout**.

*La BD répartie est décrite par **un schéma global qui contient la localisation** des données et qui permet ainsi d'aiguiller un traitement sur les processeurs détenant les données à traiter.*

*Dans les bases de données réparties, on peut distinguer entre la base de données **logique** et les bases de données **physiques** :*

Base logique : est la base de données vue par l'utilisateur comme une seule et même base de données.
Base physique : représente chacune des bases de données regroupées au sein d'une base logique.

1.3. Définitions



1.3. Définitions

Une base de données répartie ne doit donc pas être confondue avec un système dans lequel les bases de données sont accessibles à distance (selon le principe client-serveur).

Elle ne doit non plus être confondue avec un système multibase ou chaque utilisateur accède à différentes bases de données en spécifiant leur nom et adresse (plusieurs BDD hétérogènes), et le système se comporte alors comme un serveur de BDD et n'apporte aucune fonctionnalité particulière à la répartition.

Au contraire, un système de base de données répartie est suffisamment complet pour décharger les utilisateurs de tous les problèmes de concurrence, fiabilité, optimisation de requêtes ou transaction sur des données.

1.3. Définitions

Exemple :

Algérie Télécom possède des agences (Actel) à Alger, à Oran et dans toutes les villes d'Algérie ;

Dans le cas d'une BDD centralisée, le siège social d'Algérie Télécom gèrerait tous les comptes des abonnés à la téléphonie fixe et les agences devraient communiquer avec le siège social pour pouvoir accéder aux données des abonnés.

Par contre, dans le cas d'une BDD répartie, les informations sur les comptes sont distribuées dans les agences et celles-ci sont interconnectées (entièrement ou partiellement) afin qu'elles puissent avoir accès aux données externes.
Cependant, la répartition de la base de données d'Algérie Télécom est invisible aux agences en tant qu'utilisateurs et la seule conséquence directe pour elles est que l'accès à certaines données est beaucoup plus rapide.

1.4. Les règles des bases de données réparties

Quatre principales règles existent dans base de données réparties à savoir :

- Indépendance à la localisation
- Indépendance à la fragmentation/réplication
- Indépendance aux SGBD
- Autonomie des sites

1.4. Les règles des bases de données réparties

Indépendance à la localisation des données

Permettre d'écrire des programmes d'application sans connaitre la localisation physique des données ;
Les noms des objets de base de données doivent être indépendants de leur localisation.
Les requêtes des utilisateurs exprimées en SQL sont réécrites avec des requêtes locales de façon transparente.

Avantages ; Simplifier la vue utilisateur, la réécriture des requêtes, et particulièrement d'introduire la possibilité de déplacer les objets sans modifier les requêtes.

1.4. Les règles des bases de données réparties

Indépendance à la localisation des données

Les utilisateurs accèdent à la base soit directement par le schéma conceptuel. Mais en aucun cas ils n'ont les moyens d'accéder aux schémas locaux ni de préciser le site. C'est le principe de transparence de localisation.

C'est le système qui recherche le site ou sont mémorisées ces informations et non l'utilisateur qui doit l'indiquer.

1.4. Les règles des bases de données réparties

Indépendance à la fragmentation/réplication des données

Une relation dans une base de données répartie est constituée de différents fragments (parties), localisés dans des sites différents. La relation principale ne doit pas dépendre de la manière de sa décomposition et doit pouvoir être modifiée sans modifier les programmes.

Les utilisateurs ne doivent pas savoir si une telle information est fractionnée et ne doivent donc pas se préoccuper de la réunifier. C'est le système qui gère les partitionnements et les modifie en fonction de ses besoins et c'est donc lui qui doit rechercher toutes les partitions et les intégrer en une seule information logique présentée à l'utilisateur.

Les utilisateurs n'ont pas à savoir si plusieurs copies d'une même information sont disponibles. C'est le principe de transparence de duplication. La conséquence directe est que lors de la modification d'une information, c'est le système qui doit se préoccuper de mettre à jour toutes les copies.

1.4. Les règles des bases de données réparties

Indépendance aux SGBD

Une base de données répartie ne doit pas être dépendante des différents systèmes de gestion de bases de données.

La relation globale doit pouvoir être exprimée dans un langage normalisé indépendant des constructeurs.

1.4. Les règles des bases de données réparties

Autonomie des sites

Cette règle vise à garder une administration locale séparée et indépendante pour chaque serveur participant à la base de données répartie, et ainsi d'éviter la nécessite d'une administration centralisée.

La reprise après une panne et les mises à niveau des logiciels doivent être réalisées localement et ne doivent pas avoir d'impacts sur les autres sites.

Même si chaque base travaille étroitement avec les autres bases, les gestions de schémas doivent rester indépendantes. Chaque base conserve son dictionnaire local contenant les schémas locaux.

1.5. SGBD réparti

Une base de données centralisée est gérée par un seul SGBD, elle est stockée dans sa totalité à **un emplacement physique unique** et ses divers traitements sont confiés à **une seule et même unité de traitement**. Par opposition, une base de données réparties est gérée par **plusieurs processeurs**, **sites** ou SGBD.

Un système de bases de données réparties **ne doit** donc en aucun cas **être confondu** avec un système dans lequel les bases de données sont **accessibles à distance**. Il ne doit non plus être confondu avec une multibase ou une BD fédérée.

Dans une multibase, plusieurs BDD interopèrent avec une application via un langage commun et sans modèle commun.

Dans une BD fédérée, plusieurs BDD hétérogènes sont accédées comme une seule via une vue commune.

1.5. SGBD réparti

Un SGBD réparti doit disposer de :

- Dictionnaire de données réparties
- Traitement de requêtes réparties
- Gestion des transactions réparties
- Communication de données inter-sites
- Gestion de la cohérence et de la sécurité

1.5. SGBD réparti

Le SGBD réparti reçoit des requêtes référençant des tables d'une base de données réparties.

Il assure la réécriture des requêtes distribuées en plusieurs sous-requêtes locales envoyées à chaque site. La réécriture de requête est une décomposition qui prend en compte les règles de localisation.

Pour les mises à jour, le SGBD doit assurer la gestion des transactions réparties, en prenant en compte la vérification des règles d'intégrité multibases, le contrôle des accès concurrents et surtout la gestion de l'atomicité des transactions distribuées.

Le SGBD réparti peut utiliser les fonctions locales de gestion de transactions pour accomplir les fonctions globales.

1.6. Stockage de données réparties

Il existe deux façons de stocker des données sur différents sites. **Réplication** et **Fragmentation**

Réplication

L'ensemble de la relation est stocké de manière redondante sur 2 sites ou plus. Si l'ensemble de la base de données est disponible sur tous les sites, il s'agit d'une base de données entièrement redondante.

→ Les systèmes conservent des copies des données.

Avantage : elle augmente la disponibilité des données sur différents sites.

→ Possibilité traitements parallèles.

Certains inconvénients : Les données doivent être constamment mises à jour (sous peine d'entraîner une incohérence).

→ Représente beaucoup de frais généraux.

En outre, le contrôle de la simultanéité devient beaucoup plus complexe car l'accès simultané doit maintenant être vérifié sur plusieurs sites.

1.6. Stockage de données réparties

Fragmentation

Les relations sont fragmentées = divisées en plus petites parties.
Chaque fragment est stocké dans différents sites où il est nécessaire. Il faut s'assurer que les fragments sont tels qu'ils peuvent être utilisés pour reconstruire la relation originale (sans perte de données).

La fragmentation est avantageuse car elle ne crée pas de copies des données, la cohérence n'est pas un problème.

La fragmentation des relations peut se faire de deux manières :

- Fragmentation horizontale - Séparation par lignes - La relation est fragmentée en groupes de tuples de sorte que chaque tuple est attribué à au moins un fragment.
- Fragmentation verticale : séparation par colonnes. Le schéma de la relation est divisé en schémas plus petits. Chaque fragment doit contenir une clé candidate commune afin d'assurer une liaison sans perte.

Dans certains cas, une approche hybride de fragmentation et de réplication est utilisée.

1.7. Communication Inter-sites

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

- *Listener*

Chaque SGBD dispose d'une table des BDD accessibles

- *Nom >> doit être unique !!!*
- *Adresse*
- *Protocole*

Cette approche permet aussi un équilibrage de charge transparent…

2. CONCEPTION D'UNE BASE DE DONNEES REPARTIE

La définition du schéma de répartition est **la partie la plus délicate** de la phase de conception d'une BDD répartie car il n'existe pas de méthode miracle pour trouver la solution optimale.

L'administrateur doit donc prendre des décisions en fonction de critères techniques et organisationnels avec pour objectif de minimiser le nombre de transferts entre sites, les temps de transfert, le volume de données transférées, les temps moyens de traitement des requêtes, le nombre de copies de fragments, etc...

2.1. Conception descendante (*Top Down Design*)

On commence par définir un schéma conceptuel global de la BDD répartie, puis on distribue sur les différents sites en des schémas conceptuels locaux.

La répartition se fait donc en deux étapes, en première étape la **fragmentation**, et en deuxième étape l'**allocation** de ces fragments aux sites.

Cette méthode de conception est utilisée lors de la constitution de nouvelles bases de données.

Sa démarche consiste à partir du schéma global, de construire des schémas locaux.

Cette approche est généralement guidée par l'objectif de performances à obtenir par la mise a proximité des données aux utilisateurs potentiels.

2.2. Conception ascendante (*Bottom Up Design*)

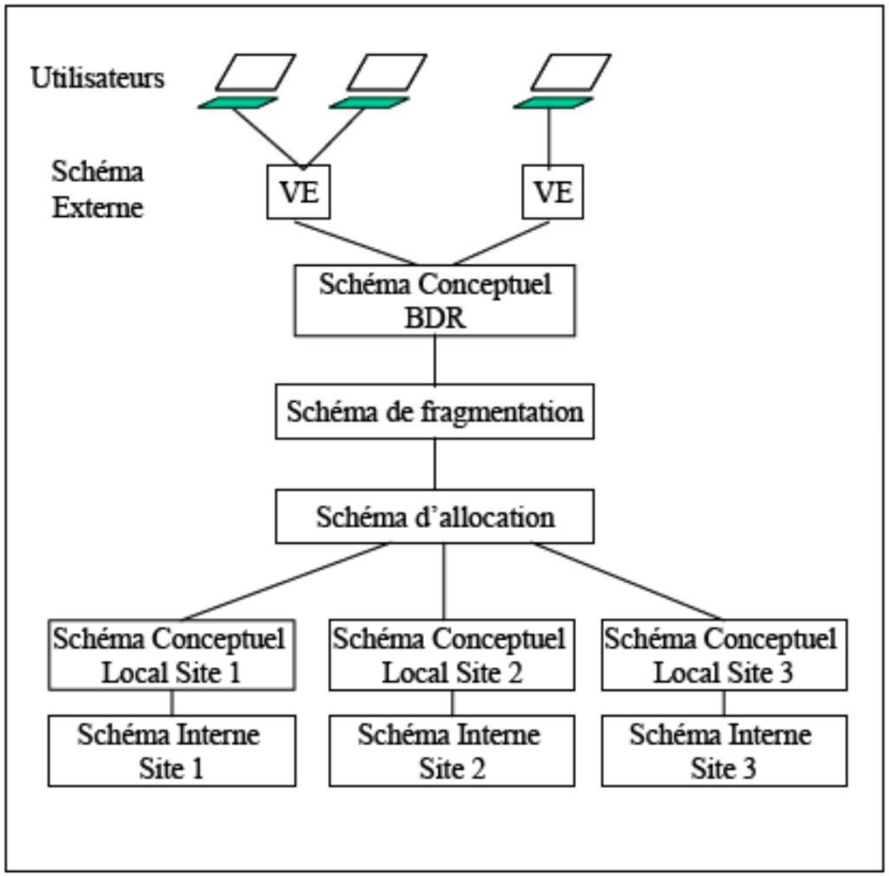
L'approche se base sur le fait que la répartition est déjà faite, mais il faut réussir à intégrer les différentes BDD existantes en une seule BD globale. En d'autres termes, les schémas conceptuels locaux existent et il faut réussir à les unifier dans un schéma conceptuel global.

La conception ascendante permet l'intégration de bases de données locales existantes dans une base répartie.

Il s'agit cette fois de construire un schéma global à partir de schémas locaux, généralement existants.

Cette démarche est la plus difficile puisqu'en plus des problèmes techniques identiques à ceux inhérents à une conception descendante, il faudra également résoudre des problèmes d'hétérogénéité des systèmes ou même sémantiques des informations.

2. CONCEPTION D'UNE BASE DE DONNEES REPARTIE



La répartition d'une base de données intervient dans les trois niveaux de son architecture en plus de la répartition physique des données :

Niveau externe :

Les vues sont distribuées sur les sites utilisateurs.

Niveau conceptuel :

Le schéma conceptuel des données est associé, par l'intermédiaire du schéma de répartition (schéma de fragmentation et schéma d'allocation), aux schémas locaux qui sont réparties sur plusieurs sites, les sites physiques.

Niveau interne :

Le schéma interne global n'a pas d'existence réelle mais fait place à des schémas internes locaux répartis sur différents sites.

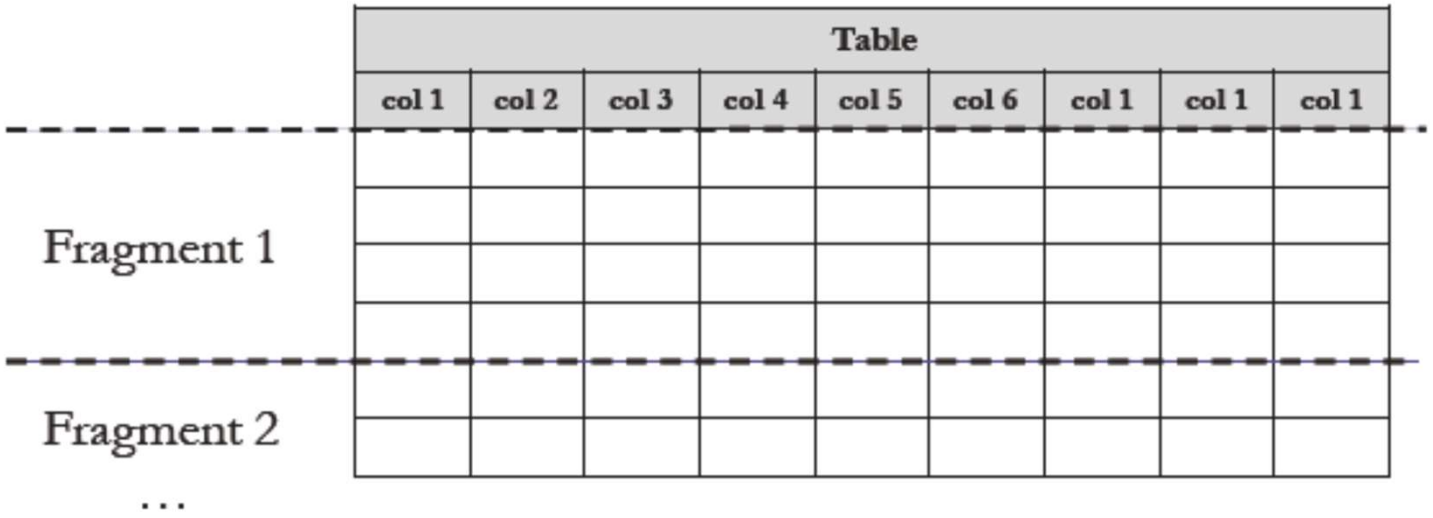
3. FRAGMENTATION

La fragmentation est le processus de décomposition d'une base de données en un ensemble de sous-bases de données. **Cette décomposition doit être sans perte d'information.**

Les **règles de fragmentation** sont les suivantes :

- **Complétude** : pour toute donnée d'une relation R, il existe un fragment Ri de la relation R qui possède cette donnée.
- **Reconstruction** : pour toute relation décomposée en un ensemble de fragments Ri, il existe une opération de reconstruction. Pour les fragmentations horizontales, l'opération de reconstruction est une union. Pour les fragmentations verticales c'est la jointure.
- **Disjonction** : une donnée n'est présente que dans un seul fragment, sauf dans le cas de la fragmentation verticale pour la clé primaire qui doit être présente dans l'ensemble des fragments issus d'une relation.

3.1. Répartition des occurrences – Fragmentation horizontale



La fragmentation horizontale est un découpage d'une table en sous-tables par utilisation de prédicats permettant de sélectionner les lignes appartenant à chaque fragment. La relation initiale sera obtenue par union des fragments.

3.1. Répartition des occurrences – Fragmentation horizontale

Les occurrences d'une même classe peuvent être réparties dans des fragments différents.

- L'opérateur de partitionnement est la **Sélection** (**σ**)
- L'opérateur de recomposition est l'**Union** (U)

Exemple

Relation Compte

Noclient	Agence	TypeCompte	Somme
174723	Oran	courant	123345
177498	Alger	courant	34564
201639	Oran	courant	45102
201639	Oran	dépôt	325100
203446	Alger	courant	274882

Relation Agence

Agence	Adresse
Oran	Rue du lac, 3. 1002 Oran
Alger	Avenue du Mont Blanc, 21. 1200 Alger

Relation Client

Noclient	NomClient	Prenom	Age
174723	LARBI	Nabil	29
177498	BAROUDI	Mounir	38
201639	YAGOUBI	Mohammed	51
203446	EDDINE	Kamel	36

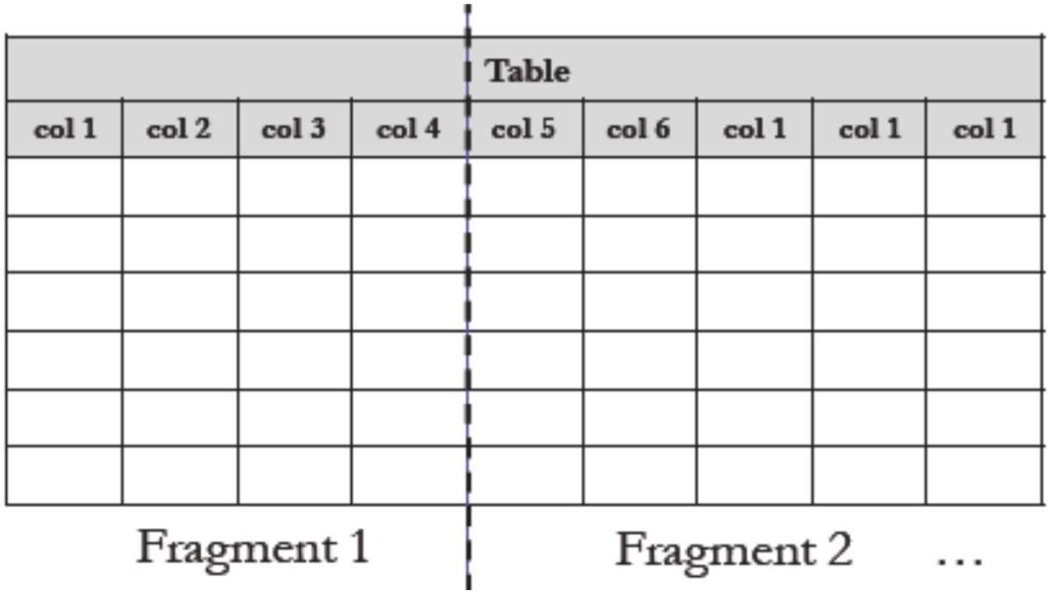
3.1. Répartition des occurrences – Fragmentation horizontale

Dans l'exemple précédent, la relation Compte peut être fractionnée en Compte1 et Compte2 avec la fragmentation suivante :

Compte1 = **σ**[TypeCompte = 'courant'] Compte
Compte2 = **σ**[TypeCompte = 'dépôt']Compte

La reconstruction de la table Compte est réalisée par :
Compte1 **U** Compte2

3.2. Répartition des attributs – Fragmentation verticale



Toutes les valeurs des occurrences pour un même attribut se trouvent dans le même fragment.

La fragmentation verticale est le découpage d'une table en sous-tables par des projections permettant de sélectionner les colonnes qui composent chaque fragment. Afin de ne pas perdre d'informations, la relation initiale doit pouvoir être recomposée par jointure des fragments.
Une fragmentation verticale est utile pour distribuer les parties des données sur le site ou chacune de ces parties est utilisée.

3.2. Répartition des attributs – Fragmentation verticale

- L'opérateur de partitionnement est la **Projection** (π)
- L'opérateur de recomposition est la **Jointure** ($\bowtie$)

Soit le partitionnement de la relation précédente Client en deux relations :

Cli1 = π [NoClient, NomClient] Client
Et Cli2 = π [Noclient, Prénom, Age] Client

La relation d'origine est obtenue avec la recomposition suivante : Client = Cli1 $\bowtie$ Cli2

3.3. Répartition des valeurs – Fragmentation hybride

C'est la combinaison des deux fragmentations précédentes, horizontale et verticale.

Les occurrences et les attributs peuvent donc être répartis dans des partitions différentes.

- L'opération de partitionnement est une combinaison de **Projections** et de **Sélections**.
- L'opération de recomposition est une combinaison de **Jointures** et d'**Unions**.

Exemple 1 :
Cli1 = π [NoClt, NomClt]σ[Age < 38]Client,
Cli2 = π [NoClt, NomClt]σ[Age ≥ 38]Client
Cli3 = π [NoClt, Prenom]Client
Cli4 = π [NoClt, Age]Client

La relation Client est obtenue avec : (Cli1 ∪ Cli2) ⋈ Cli3 ⋈ Cli4

3.3. Répartition des valeurs – Fragmentation hybride

Exemple :

Cli1 = π [NoClt, NomClt]σ[Age < 38]Client,
Cli2 = π [NoClt, NomClt]σ[Age ≥ 38]Client
Cli3 = π [NoClt, Prenom]Client
Cli4 = π [NoClt, Age]Client

La relation Client est obtenue avec : (Cli1 ∪ Cli2) ⋈ Cli3 ⋈ Cli4

3.4. Définition des fragments

Le principe est de baser la fragmentation sur l'ensemble des **requêtes** d'**interrogation** ou de **mise à jour**.

Pour cela, il faut extraire de ces requêtes toutes les conditions de **sélections**, les attributs **projetés** et les **jointures**.

Les opérations de sélection sont à la base des fragmentations horizontales
Les opérations de projection sont à la base des fragmentations verticales.

Pour éviter le risque d'émietter la base de données, on peut restreindre le nombre de requêtes prises en considération.

On ne s'intéresse alors qu'aux requêtes les plus fréquentes ou les plus sensibles (celles pour lesquelles le temps de réponse maximum est borné).

3.4.1. Définition des fragments horizontaux

Comme les fragments horizontaux doivent être exclusifs, on produit l'ensemble des 2^n conjonctions de conditions où chaque condition élémentaire est prise dans sa forme positive ou dans sa forme négative :

3.4.2. Définition des fragments verticaux

Les fragments verticaux sont exclusifs, sauf en ce qui concerne le (ou les) attribut(s) de jointure qui sont communs à tous les fragments, et ce pour que la décomposition soit sans perte d'information.