UNIVERSITÉ PRIVÉE DE MARRAKECH | UP | جامعة ابن زهر مراكش
UNIVERSITÉ RECONNUE PAR L'ÉTAT
جامعة معترف بها من طرف الدولة

Les bases de données réparties

INFO 3
2020-2021

Sommaire

1. Définitions
2. Pourquoi repartir?
3. Principe de répartition.
4. SGBD réparties.
5. Avantage de la répartition.
6. Inconvénient de la répartition.
7. Les BD fragmentées.
8. Conception.
9. Les types de fragmentation :
    1. Par relation (Par classes);
    2. Horizontale (Par occurrences ou par tuples);
    3. Verticale (par attributs);
    4. Mixte (Hybride, par valeurs).
10. Les bases de données répliquées.

Définitions

Architectures

- Architecture centralisée:

  Programme d'application et SGBD sur même machine(même site),

- Architecture client-serveur, notée2-tiers:

  Programme d'application sur le client et SGBD sur le serveur,

- Architecture 3-tiers:

  Interface utilisateur sur le client, programme d'application sur le serveur d'application et le SGBD sur le serveur de données.

Définitions

Bases de données

- Centralisées
- Réparties ou distribuées
- Fédérées
- Multi-bases

Définitions

Base de Données centralisée:

- Gérée par un seul SGDB,
- Stockée dans un emplacement physique unique,
- Les traitements sont confiés à une seule et même unité.

Définitions

Multi-base:

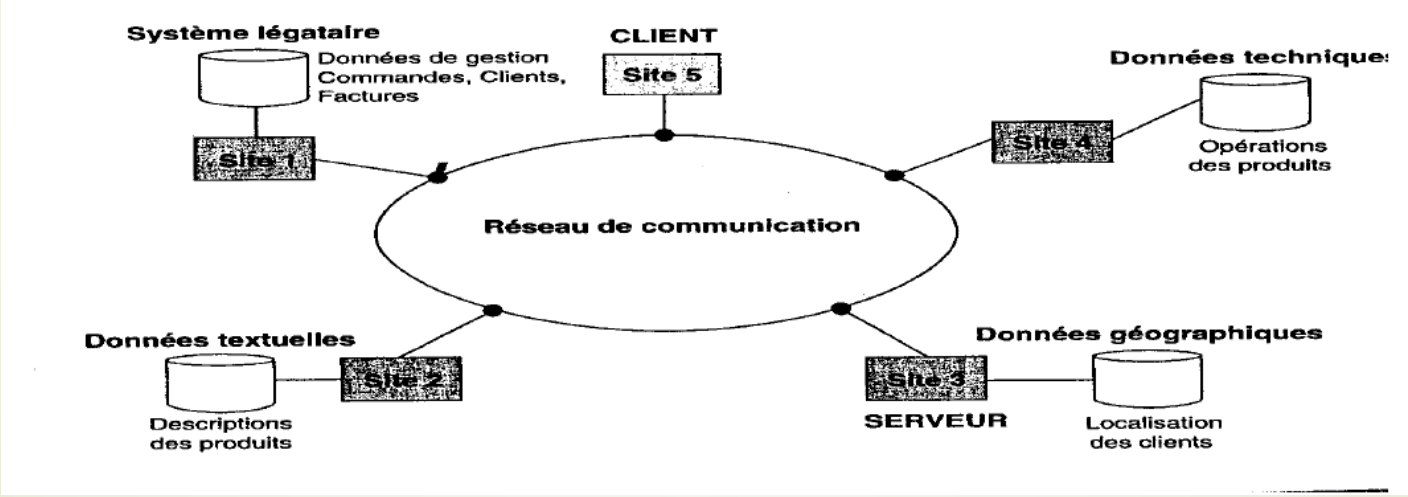
- Plusieurs SGBD, hétérogènes ou non, capables d'inter-opérer sans modèle commun(vue commune).

Définitions

Base de Données Fédérée:

- Plusieurs SGBD hétérogènes sont utilisés comme un seul via un modèle commun(vue commune)

Exemple : Utilisés dans la fusion ou acquisition des bases de données de compagnies.



Définitions

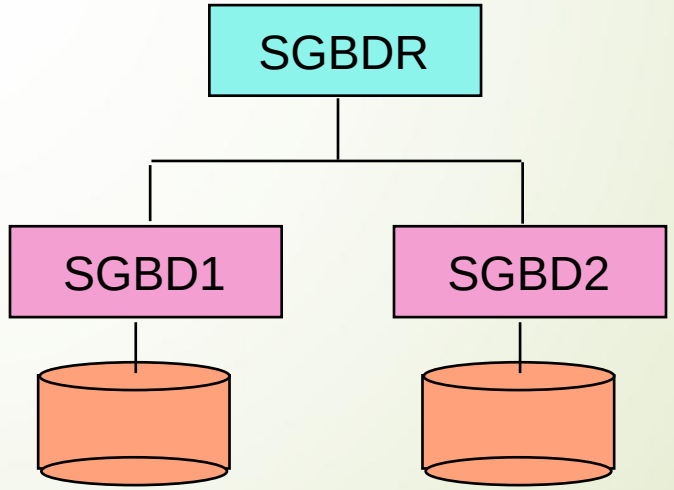
Base de Données Répartie:

- Un seul système gérant une collection de bases de données logiquement reliées qui sont réparties sur différents sites accessible via un modèle commun vue commune).

Définitions

Base de Données Répartie:

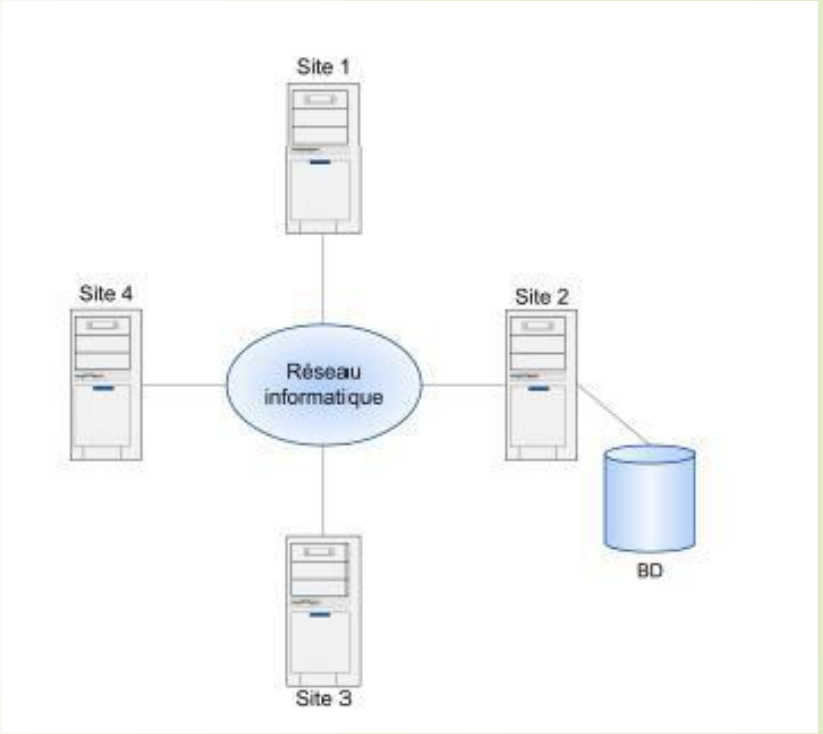
- Ensemble de Bases de Données stockées dans des serveurs distincts interconnecté par un réseau et géré par un SGDBR, mais apparaissent à l'utilisateur comme une Base de Données unique.



Définitions

Traitement distribué

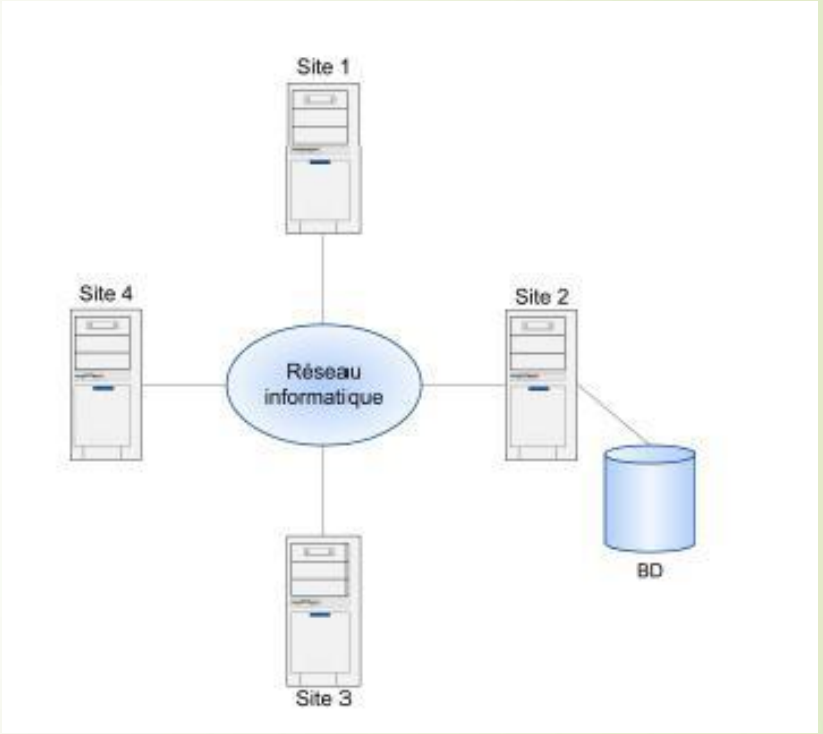
- Il est essentiel de distinguer un SGBD distribué du traitement distribué Traitement distribué : Une base de données centralisée accessible par le biais d'un réseau informatique.



Définitions

Traitement distribué

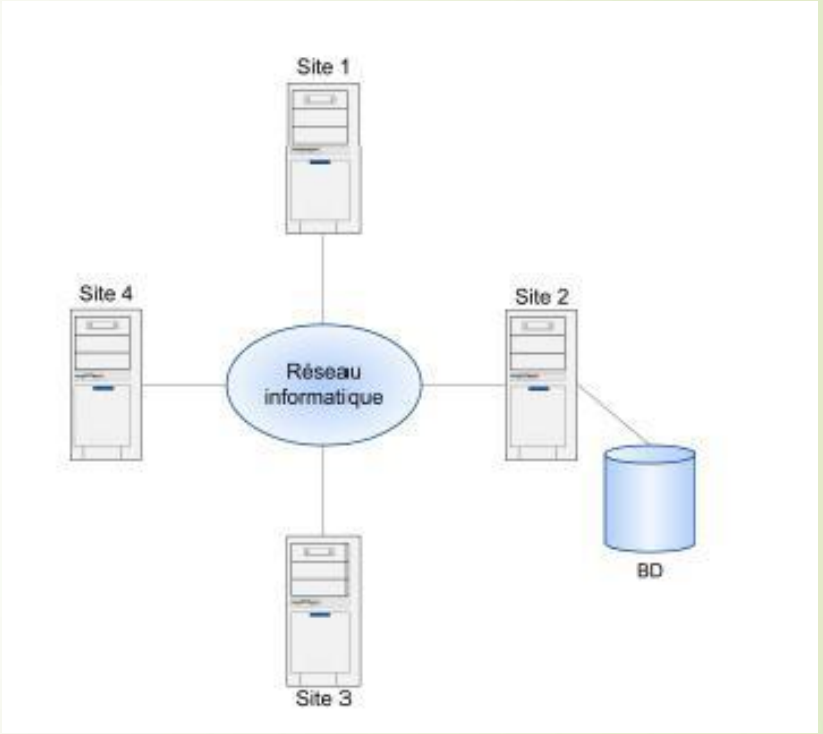
- Il est essentiel de distinguer un SGBD distribué du traitement distribué Traitement distribué : Une base de données centralisée accessible par le biais d'un réseau informatique.



Définitions

Traitement distribué

- Il est essentiel de distinguer un SGBD distribué du traitement distribué Traitement distribué : Une base de données centralisée accessible par le biais d'un réseau informatique.



La bases de données réparties et le modèle client-serveur

Client-Serveur	Bd Répartie
Plusieurs bases vues par le client	Une base logique vue par l'utilisateur
Plusieurs connexions	Une seule connexion
Localisation explicite des bases(« connection string » ou DSN)	Indépendance à la localisation
Règles de localisation dans l'application	Règles de localisation dans le dictionnaire
1 ordre SQL → 1 seule BD	1ordre SQL → plusieurs BD
n transactions monobase	1 transaction logique
Synchronisation des transactions dans l'application	Synchronisation automatique des n sous-transactions
Plusieurs COMMIT monobase	1 COMMIT généralisé

Pourquoi répartir?

- Besoin de décentralisation des organisations,
- Partage de données géographiquement reparties.

Principe de la répartition:

- Les données et traitements sont répartis sur différents sites interconnectés par un réseau de communication.
- La défaillance d'un site ne peut pas entraîner l'indisponibilité totale du système et sa probabilité peut être négligée grâce à la tolérance aux faute, assurée par la redondance des informations et des traitements.

Système de Gestion de Bases de Données Réparties:

- Constitué d'un ensemble de processeurs autonomes «appelés sites»(stations de travail, micro-ordinateurs,…)reliés par un réseau de communication qui leur permet d'échanger des données.
- Un SGBDRé suppose que les données soient stockées sur au moins deux sites.
- Chaque site est doté de son propre SGBD.

Avantages de répartir les données

Reflète une structure organisationnelle :

- Limiter le transfert d'information (en nombre et en volume) en répartissant les données là où elles sont le plus utilisées. Ceci est particulièrement important pour une base de données dont les différents utilisateurs sont géographiquement dispersés

Exemple: Des agences d'une banque

Avantages de répartir les données

Performances améliorées:

- La répartition de charge de travail sur plusieurs unités de traitement opérant en parallèle permet d'accroître les performances.

Avantages de répartir les données

- Augmenter la fiabilité et la disponibilité en dupliquant les données sur plusieurs sites

Exemple: Comptes bancaires

Avantages de répartir les données

Économie :

- l'ajout de stations de travail à un réseau est nettement moins coûteux que l'extension d'un gros système centralisé.

Scalability (évolutivité)

- Facilité d'accroissement du système

Inconvénients de répartir les données

Complexité:

- un SGBDD masque sa nature répartie aux yeux des utilisateurs et fournit un niveau acceptable de performances, de fiabilité et de disponibilité .

Inconvénients de répartir les données

Sécurité:

- Dans un système centralisé, l'accès aux données est d'un contrôle facile, tandis que dans un système distribué, non seulement il faut contrôler l'accès aux données dupliquées dans des emplacements multiples, mais le réseau lui-même doit être sécurisé.

Inconvénients de répartir les données

Contrôle d'intégrité de données plus difficile:

- L'intégrité de base de données fait référence à la validité et à la cohérence des données stockées.

Inconvénients de répartir les données

Complexité plus grande du design de bases de données :

- le design d'une base de données distribuée doit prendre en considération la fragmentation des données, l'allocation des fragments à des sites spécifiques et la réplication des données.

Inconvénients de répartir les données

Coût :

- La distribution entraîne des coûts supplémentaires en terme de communication, et en gestion des communications (hardware et software à installer pour gérer les communications et la distribution).

Exemple de bases de données distribuée

Une banque peut avoir des agences à Marrakech, à Casablanca et à Fès.

BD centralisée:

1. Le siège principal de la banque gérerait tous les comptes des clients,
2. Les agences devraient communiquer avec le siège social pour avoir accès aux données.

BD 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 bancaire est invisible aux agences entant qu'utilisateurs, et la seule conséquence directe pour elles est que l'accès à certaines données est beaucoup plus rapide.

Objectifs d'une base de données repartie

Les 12 principaux objectifs définis par C.J.Date sont:

1) Transparence pour l'utilisateur,
2) Autonomie de chaque site,
3) Absence de site privilégié,
4) Continuité de service,
5) Transparence vis-à-vis de la localisation de données,
6) Transparence vis-à-vis de la fragmentation,
7) Transparence vis-à-vis de la réplication,
8) Traitement réparti des requêtes,
9) Indépendance vis-à-vis du matériel,
10) Indépendance vis-à-vis du système d'exploitation,
11) Indépendance vis-à-vis du réseau,
12) Indépendance vis-à-vis du SGBD

Objectifs d'une base de données repartie

- Transparence de **localisation**:
  - Les utilisateurs accèdent au schéma conceptuel via des vues. Un utilisateur ne sait pas sur quel site se trouvent physiquement les données
    - Exemple: un client peut ouvrir un compte à Fès et effectuer régulièrement des opérations à Rabat → C'est le système qui recherche le site où sont mémorisées ses informations et non l'utilisateur qui doit l'indiquer.

Objectifs d'une base de données repartie

- Transparence de **partitionnement**:
  - l'utilisateur ne sait pas comment la base est partitionnée.

- Transparence de **duplication**:
  - l'utilisateur ne sait pas s'il existe plusieurs copies d'une même information ou non.
    - ➔Lors de la modification d'une information, c'est le système qui doit se préoccuper de mettre à jour toutes les copies.

      - Exemple: Un client possède des comptes (courant, épargne logement, ...) à Fès mais effectue régulièrement des retraits d'argent à Rabat, les informations le concernant soient réparties entre Fès à Rabat: l'historique des opérations du compte courant est conservé à Rabat, la gestion de ses autres comptes est assurée à Fès, et le solde du compte courant est dupliqué à Fès et Rabat.

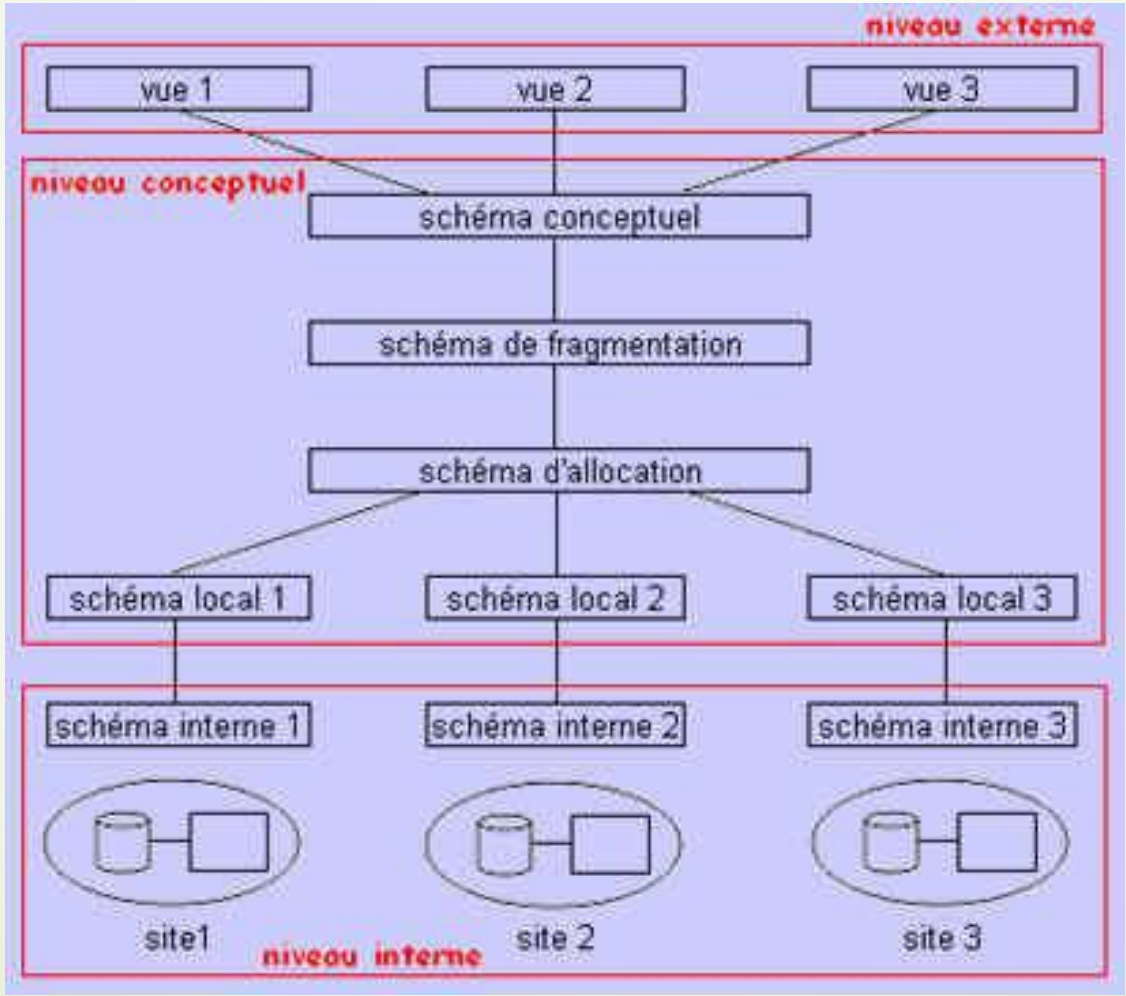
Objectifs d'une base de données repartie

- Indépendance vis-à-vis du SGBD:
  - Un système réparti ne doit pas être dépendant en aucun cas des différents SGBDs, la relation globale doit être exprimée dans un langage normalisé indépendant des constructeurs.
- Autonomie de chaque site:
  - Vise à garder une administration locale séparée et indépendante pour chaque serveur participant à la base de données répartie afin d'éviter une administration centralisée.
    - Toute manipulation sur un site (reprise après panne, mises à jour des logiciels) ne doit pas altérer le fonctionnement des autres sites.

Modélisation d'une base de données réparties

- Pour une base de données réparties, la répartition a lieu dans les trois niveaux:
  - Les vues des utilisateurs sont présentées sur leur site (site utilisateur); elles correspondent au niveau externe. Il y a donc répartition des vues.
  - Le schéma conceptuel (global) est associé aux schémas locaux des sites physiques via un schéma de fragmentation (la manière dont la base est découpée) et un schéma d'allocation (la manière dont les fragments sont répartis).
  - Il n'y a pas de schéma interne global, mais des schémas locaux internes.

Modélisation d'une base de données réparties



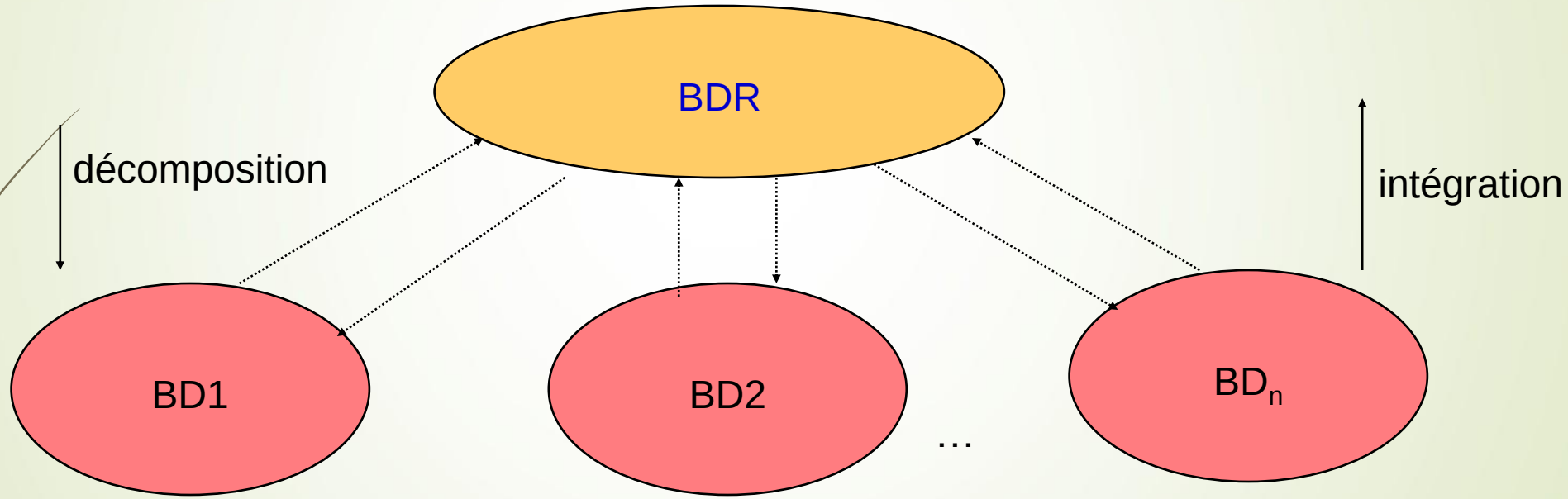
Conception d'une base de données réparties

- La définition du schéma de répartition est la partie la plus délicate de la phase de conception d'une base de données répartie:
  - Il n y'a pas de méthode standard → on se base sur les besoins,
  - L'administrateur (le concepteur) doit prendre des décisions sur comment la répartition doit être faite.
  - L'objectif est 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.

Conception d'une base de données réparties

- Méthodes de conception:
  - Deux approches fondamentales sont à l'origine de la conception des bases de données réparties :
    1. La conception descendante (Top down design),
    2. La conception ascendante (Bottom up design).
  - NB: Pour les deux approches il faut recueillir l'expression des besoins des utilisateurs afin de créer un schéma conceptuel unique et en déduire les vues externes à prévoir.

Conception d'une base de données réparties



Les types des bases de données réparties

Une base de données répartie ou distribuée peut de deux types :

- Base de données partitionnée ou fragmentée.
- Base de données répliquée.

les bases de données partitionnées ou fragmentées

Conception descendante Par Intégration Fédération

Conception d'une base de données réparties

Conception descendante / Par Intégration / Fédération

⇨ **Fédération** : Création d'un schéma unique partant de plusieurs schémas

- **Objectifs :**
  - **Donner aux utilisateurs une vue unique des données implémentées sur plusieurs systèmes *a 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'inter opérer**
  - **Procédure d'intégration :**
    - **Traitement de l'hétérogénéité sémantique**
    - **Traduction des schémas : traitement de l'hétérogénéité syntaxique**
    - **Intégration des schémas**

Conception d'une base de données réparties

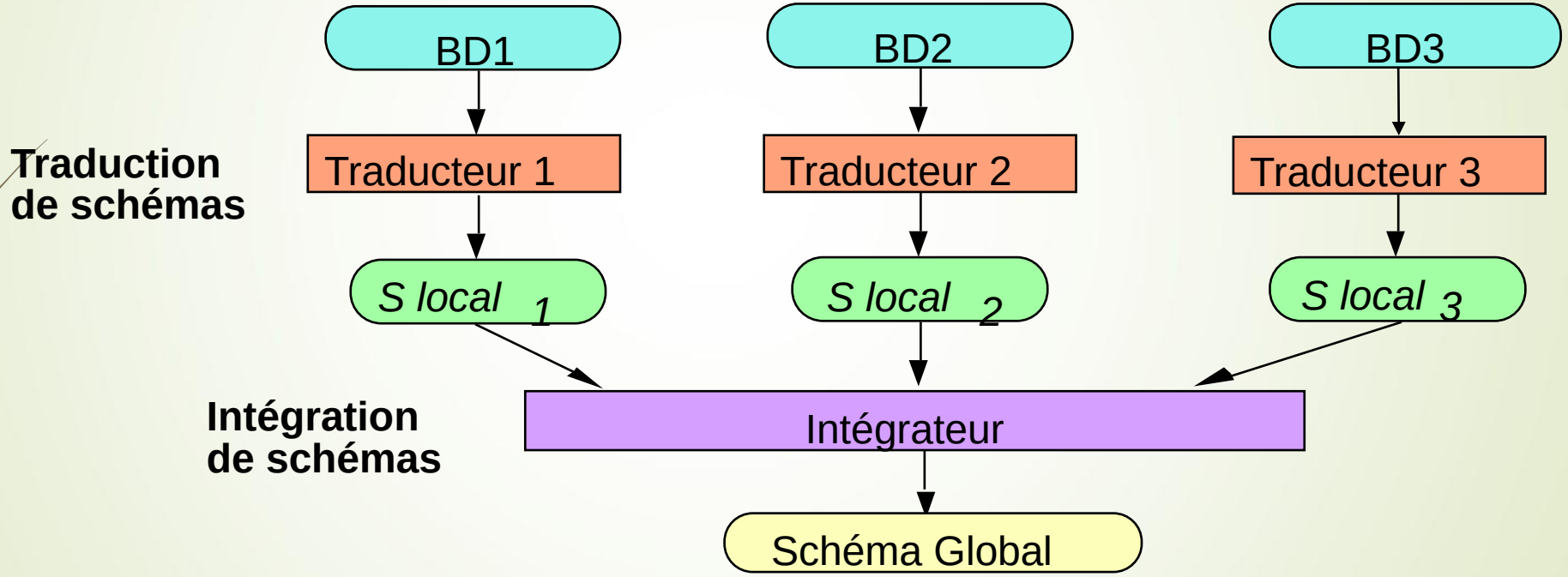
Conception descendante / Par Intégration / Fédération

- **Hétérogénéité sémantique**
  - **Origine : Résulte des conceptions indépendantes des différentes BD**
  - **Effet : Désaccord sur la signification des données**
  - **Solution : Analyse sémantique comparée des données préalable à la fédération souvent groupée avec la phase de traduction**

- **Traduction des schémas (résolution de l'hétérogénéité syntaxique)**
  - **Origine : utilisation de modèles différents dans les BD composantes**
  - **Effet : nécessite des traductions de tous les modèles vers tous les modèles**
  - **Solution : traduction de tous les schémas dans un modèle commun (dit canonique ou pivot)**
  - **Problématique :**
    - **Le modèle canonique doit avoir un pouvoir de modélisation ≥ à ceux des modèles des BD composantes**
    - **Nécessité de compléter sémantiquement des modèles de BD composantes qui seraient trop pauvres**
  - **Choix du modèle canonique :**
    - **Entité - Association et Relationnel**
    - **Objet**

Conception d'une base de données réparties

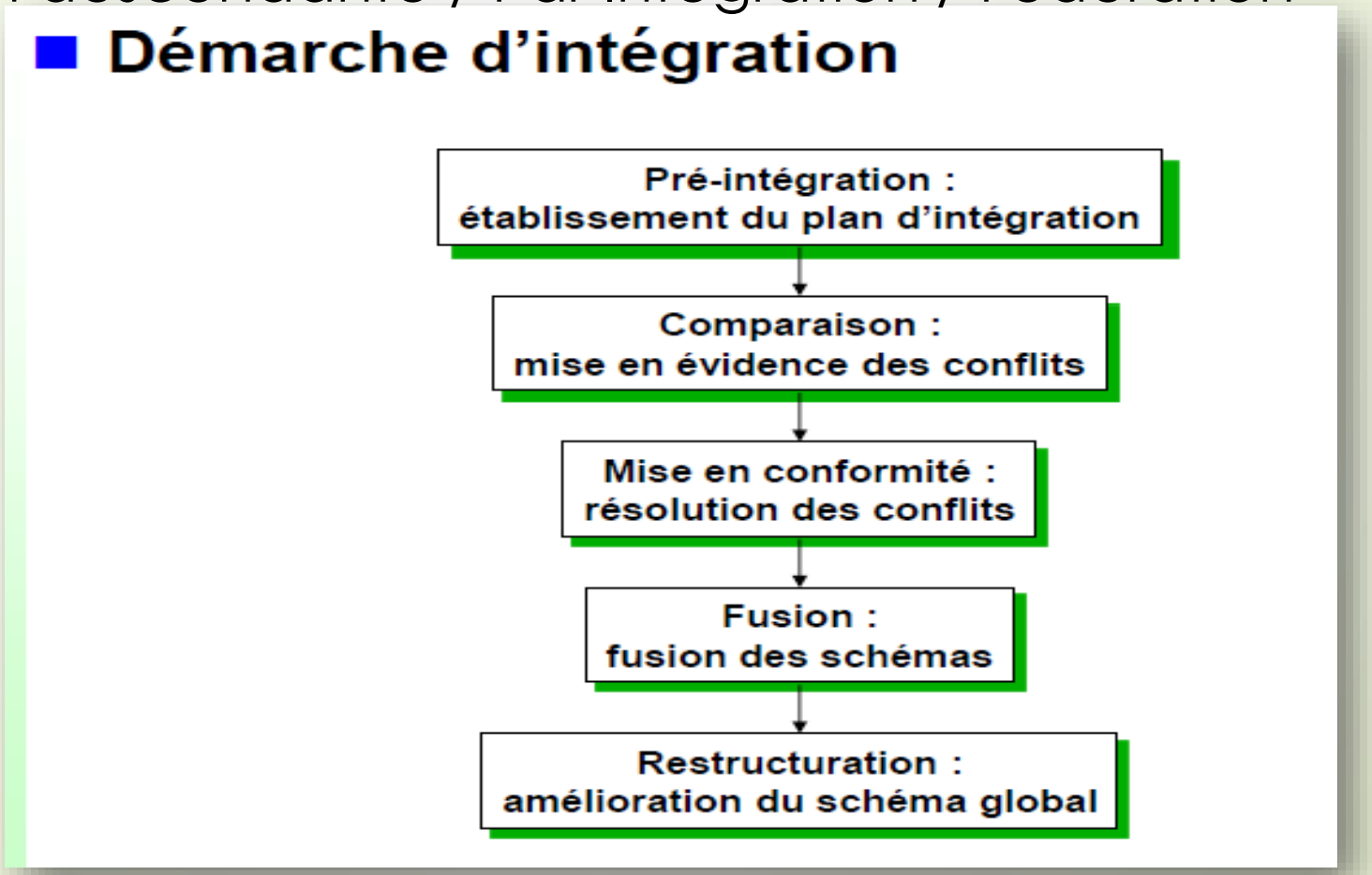
Conception descendante / Par Intégration / Fédération



- **Procédure :**
  - Identifier les éléments de base qui sont liés
  - Choisir la représentation la plus adéquate pour le schéma global
  - Intégrer les éléments des schémas intermédiaires

Conception d'une base de données réparties

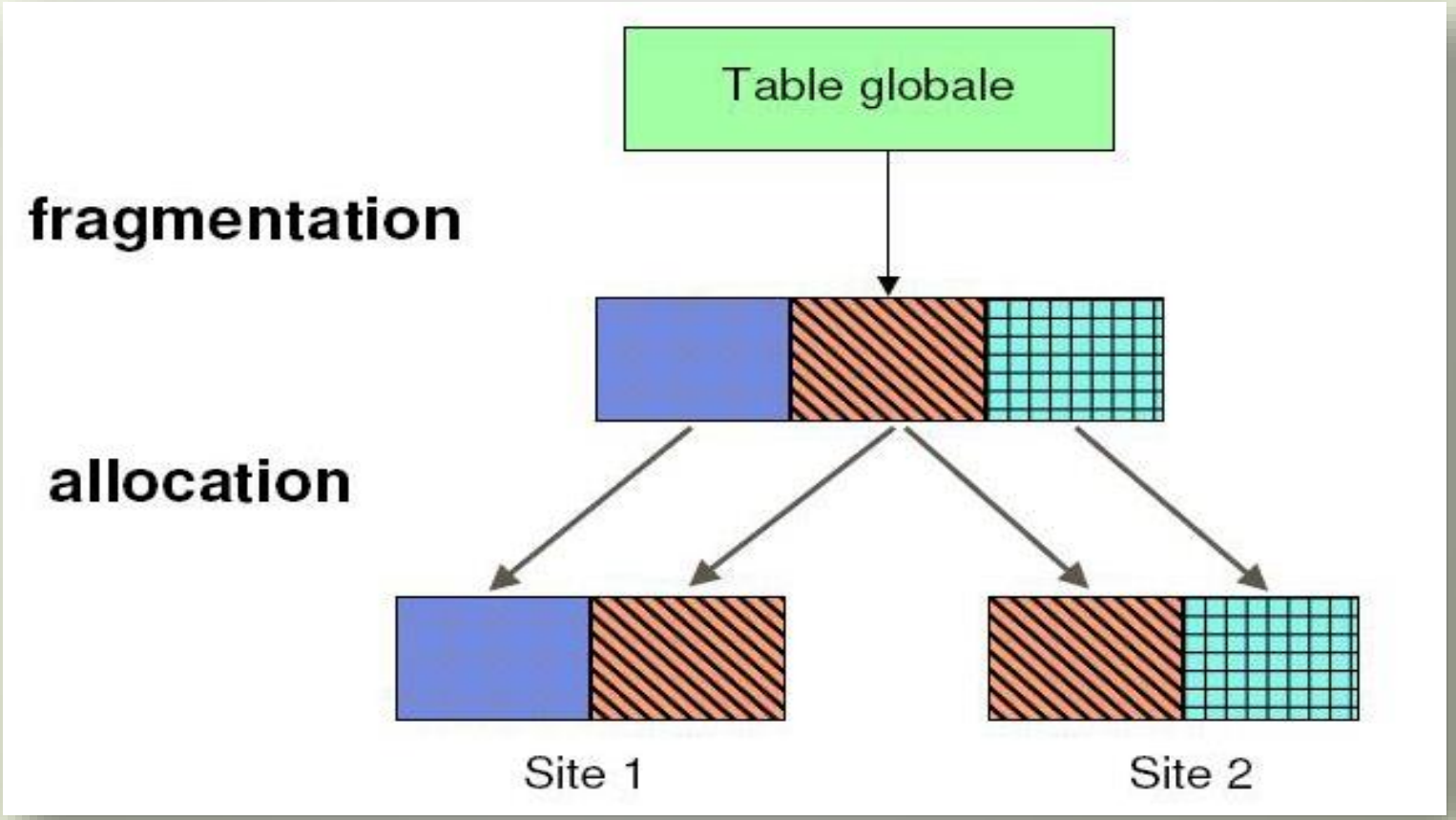
Conception descendante / Par Intégration / Fédération



Conception ascendante Par composition

Conception d'une base de données réparties

Conception ascendante / Par décomposition



Fragmentation

- Techniques de fragmentation
  - Il existe différentes façons pour fragmenter une base de données
  - L'unité de fragmentation détermine la nature de l'élément le plus petit dont les composants ne peuvent être fragmentés
    - On parle de granularité: taille du plus petit élément

    ➔ Une granularité fine donne de grandes possibilités pour la fragmentation et autorise une répartition flexible et efficace de la base de données, mais à l'inconvénient de provoquer une certaine lourdeur pour la recomposition des informations.

    ➔ A l'opposé, une granularité élevée permet une gestion simple de la fragmentation, mais fournit des possibilités très limitées pour la fragmentation.

    ➔ L'unité de fragmentation est généralement laissée au libre choix de l'administrateur.

Il y a quatre types de fragmentation : Par relation, horizontale, verticale, mixte

Fragmentation

Il y a quatre types de fragmentation :

- Par relation (Par classes);
- Horizontale (Par occurrences ou par tuples);
- Verticale (par attributs);
- Mixte (Hybride, par valeurs).

Fragmentation par relation

- Ce sont les classes (relation ou table dans le modèle relationnel, entité en Entité-Association, classe dans le modèle orienté-objet) qui peuvent être réparties dans des fragments différents.
  - → Les fragments sont définis à partir des "classes" de la base de données.
  - → Toutes les occurrences d'une même classe appartiennent au même fragment.

- L'opération de partitionnement est la définition de sous schémas.
- L'opération de recomposition est la réunion de sous schémas.

Fragmentation par relation

- Exemple: Banque
  - Trois tables: Agence, Clients, Compte

Compte

NoClient	Agence	TypeCompte	Somme
174 723	Lausanne	courant	123 345.89
177 498	Genève	courant	34 564.00
201 639	Lausanne	courant	45 102.50
201 639	Lausanne	dépôt	325 100.00
203 446	Genève	courant	274 882.95

Client

NoClient	NomClient	Prénom	Age
174 723	Villard	Jean	29
177 498	Cattell	Blaise	38
201 639	Tesllis	Alan	51
203 446	Kovalsky	Validmir	36

Agence

Agence	Adresse
Lausanne	Rue du Lac, 3, 1002 Lausanne
Genève	Avenue du Mont Blanc, 21, 1200 Genève

- Cette base peut être fragmentée en deux fragments: {Compte, Client} et {Agence}.

Fragmentation horizontale

- Appelée aussi fragmentation par occurrences ou par tuples,
- La fragmentation horizontale est basée sur un découpage (horizontal) des tuples des tables
  - Les occurrences d'une même table peuvent être réparties dans des fragments différents (avec tous les attributs)
  - Le découpage est effectué en utilisant la sélection,
  - La reconstruction est effectuée en utilisant l'opération d'union.

Fragmentation horizontale

- Fragments définis par sélection:

  Client1 = Client where ville = "Marrakech"
  Client2 = Client where ville ≠ "Marrakech"

- Reconstruction

  Client =Client1 U Client2

Client

nclient	nom	ville
C 1 C 2 C 3 C 4	Brahim Ahmed Amina Mustapha	Marrakech Rabat Marrakech Casablanca

Client1

nclient	nom	ville
C 1 C 3	Brahim Amina	Marrakech Marrakech

Client2

nclient	nom	ville
C 2 C 4	Ahmed Mustapha	Rabat Casablanca

Fragmentation horizontale dérivée

Fragments définis par jointure

Cde1 = Cde where

Cde.nclient = Client1.nclient

Cde2 = Cde where

Cde.nclient = Client2.nclient

Reconstruction

Cde = Cde1 U Cde2

Cde

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

Cde1

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

Cde2

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

Fragmentation verticale

- Les fragments sont construits à partir de quelques attributs d'une relation:
  - Ce sont les attributs (avec leur occurrences) d'une même relation qui peuvent être répartis dans des fragments différents.
  - Toutes les valeurs des occurrences pour un même attribut se trouvent dans le même fragment.
  - La répartition des attributs dans différents fragments ne peut être correcte que si l'identifiant (ou identité d'objet) est dupliqué dans chaque fragment → cet identifiant est utile pour la reconstruction (décomposition sans perte d'informations.
- Le découpage d'une relation en sous tables est effectuée utilisant la projection sur les colonnes composant chaque fragment.
- La reconstruction est effectuée par jointure des différents fragments.

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

Fragmentation Mixte

- Cette fragmentation combine la fragmentation horizontale (par occurrences) et verticale (par attributs).
  - Le partitionnement est effectué par une combinaison de projections et de sélections.
  - Les occurrences et les attributs peuvent donc être répartis dans des différents fragments.
- La reconstruction est effectuée par combinaison de jointures et d'unions des différents fragments.

Fragmentation Mixte

- Exemple: Banque
  - Compt1 = $\prod_{\text{NoClient, Somme}} (\sigma_{\text{TypeCompte = courant}}(\text{Compte}))$
  - Compt2 = $\prod_{\text{NoClient, Somme}} (\sigma_{\text{TypeCompte = dépôt}} (\text{Compte}))$
  - Compt3 = $\prod_{\text{NoClient, Agence}} \text{Compte})$
  - Compt4 = $\prod_{\text{Noclient, TypeCompte}} (\text{Compte})$

Compt1

NoClient	Somme
174 723	123 345.89
177 498	34 564.00
201 639	45 102.50
203 446	274 882.95

Compt2

NoClient	Somme
201 639	325 100.00

Compt3

NoClient	Agence
174 723	Lausanne
177 498	Genève
201 639	Lausanne
201 639	Lausanne
203 446	Genève

Compt4

NoClient	TypeCompte
174 723	courant
177 498	courant
201 639	courant
201 639	dépôt
203 446	courant

- La reconstruction

  Compte= (Compt1 ∪ Compt2) |x| Compt3 |x| Compt4

Conception d'une base de données réparties

- Conception descendante
  - Conception du schéma conceptuel global de la base de données répartie,
  - Création du schéma de fragmentation :
    - la base de données sera découpée en fragments distincts constituant une partition de la base ➔ l'intersection de ces fragments doit être vide et leur réunion doit redonner le schéma global.
  - Création du schéma d'allocation :
    - les fragments doivent être distribués entre les différents sites.
  - Création d'un schéma local pour chaque site, relatif aux fragments dévolus à ce site,
  - Création des schémas internes :
    - implémentation des données des fragments sur les supports physiques de stockage.

Les bases de données répliquées

Base de données répliquée

Dans les bases de données distribuées répliquées, la base de données entière est répliquée sur plusieurs serveurs. Il est évident que cette méthode réduit les coûts de communication et augmente les performances du système en éliminant le besoin pour la transmission de données à des nœuds différents. Malheureusement, cette méthode est très chère à cause de la mise à jour des données.

Qu'est-ce que la réplication des bases de données ?

- La réplication est un procédé qui consiste à recopier des données sur plusieurs serveurs.
- Les serveurs répliqués sont de deux types : le maître, qui contient les données de référence, et l'esclave, qui s'y approvisionne.

Qu'est-ce que la réplication des bases de données ?

- La réplication est composée d'un seul serveur maître et d'un ou plusieurs serveurs esclaves. Si en théorie il n'y a pas de limite au nombre d'esclaves, en pratique les contraintes sont d'ordre économique et physique.
- Toute modification de données doit s'effectuer sur le maître car l'information ne peut circuler que dans un seul sens, du maître vers l'esclave.

Qu'est ce qu'une réplication synchrone ?

Une réplication peut être synchrone : un serveur A envoie des données à un serveur B et doit attendre que ce dernier finisse son traitement et le prévienne en lui envoyant un accusé de réception, pour qu'il puisse poursuive. En cas de sinistre sur un site, la perte de données est donc infime voire nulle.

Qu'est ce qu'une réplication asynchrone ?

Une réplication peut également être asynchrone : dans ce cas, le serveur A envoie des données à un serveur B et il n'attend pas de réponse du serveur B pour continuer.

Comment la réplication peut améliorer la tolérance aux pannes ?

En cas de panne du serveur actif, l'application peut être basculée sur le serveur passif. La continuité de service est alors assurée.

Comment la réplication peut améliorer l'équilibrage de charge et un meilleur temps de réponse ?

C'est la possibilité d'assumer une augmentation de la charge des requêtes de lecture, en la répartissant sur plusieurs esclaves. Aucun bénéfice pour les requêtes d'écriture parce qu'elles doivent toutes s'effectuer sur le maître.

Principe

Le principe de la réplication, qui met en jeu au minimum deux SGBD, est assez simple et se déroule en trois temps :

1. La base maîtresse reçoit un ordre de mise à jour (INSERT, UPDATE ou DELETE).
2. Les modifications faites sur les données sont détectées et stockées en vue de leur propagation.
3. Un processus de réplication prend en charge la propagation des modifications à faire sur une seconde base dite esclave. Il peut bien entendu y avoir plus d'une base esclave.