

<b>Partie 6</b>	<b>SGBD distribués et réplication</b>	783
Chapitre 24	SGBD distribués - Concepts et conception	785
Chapitre 25	SGBD distribués - Concepts avancés	831
Chapitre 26	Réplication et bases de données mobiles	875

## **Chapitre 24**    **SGBD distribués – Concepts et conception**

### **objectifs du chapitre**

Dans ce chapitre, vous apprendrez:

- Le besoin de bases de données distribuées.
- Les différences entre les SGBD distribués, les traitements distribués et les SGBD parallèles.
- Les avantages et inconvénients des SGBD distribués.
- Les problèmes d'hétérogénéité dans un SGBD distribué.
- Concepts de base du réseau.
- Les fonctions qui devraient être fournies par un SGBD distribué.
- Une architecture pour un SGBD distribué.
- Les principaux problèmes liés à la conception de bases de données distribuées: fragmentation, réplication et allocation.
- Comment la fragmentation doit être effectuée.
- L'importance de l'allocation et de la réplication dans les bases de données distribuées.
- Les niveaux de transparence qui devraient être fournis par un SGBD distribué.
- Critères de comparaison pour les SGBD distribués.

La technologie des bases de données nous a fait sortir d'un paradigme de traitement des données dans lequel chaque application a défini et maintenu ses propres données pour une application dans laquelle les données sont défini et administré de manière centralisée. Ces derniers temps, nous avons vu le rapide les développements en matière de technologie de communication de réseau et de données, incarnés par le Internet, informatique mobile et sans fil, appareils intelligents et informatique en grille. Désormais, grâce à la combinaison de ces deux technologies, la technologie des bases de données distribuées ogie peut changer le mode de travail de centralisé à décentralisé. Cette technologie combinée est l'un des développements majeurs dans le domaine des systèmes de bases de données. Dans les chapitres précédents, nous nous sommes concentrés sur les systèmes de bases de données centralisées, c'est-à-dire des systèmes avec une seule base de données logique située sur un site sous le contrôle d'un SGBD unique. Dans ce chapitre, nous abordons les concepts et les problèmes du Système de gestion de base de données (DDBMS), qui permet aux utilisateurs d'accéder non seulement aux données sur leur propre site, mais également aux données stockées sur des sites distants. Il y a eu des réclamations que les SGBD centralisés deviendront à terme une «curiosité antique» en tant qu'organisations évoluer vers des SGBD distribués.

## 24.1 INTRODUCTION

Une motivation majeure derrière le développement de systèmes de bases de données est le désir d'intégrer les données opérationnelles d'une organisation et de fournir un accès contrôlé aux données. Bien que l'intégration et l'accès contrôlé puissent impliquer une centralisation, ce n'est pas l'intention. En effet, le développement des réseaux informatiques favorise un mode de travail décentralisé. Cette approche décentralisée reflète la structure organisationnelle de nombreuses entreprises, qui sont logiquement réparties en divisions, départements, projets, etc., et physiquement répartis dans les bureaux, les usines, où chaque unité conserve ses propres données opérationnelles (Date, 2000). Le partage des données et l'efficacité de l'accès aux données devraient être améliorés par le développement d'un système de base de données distribué reflétant cette organisation. structure, rend les données de toutes les unités accessibles et stocke les données à proximité endroit où il est le plus fréquemment utilisé.

**Les SGBD distribués** devraient aider à résoudre le problème des îlots d'informations. Les bases de données sont parfois considérées comme des îles électroniques qui sont des lieux distincts et généralement inaccessibles, comme des îles éloignées. Cela peut être le résultat d'une séparation géographique, architectures informatiques incompatibles, protocoles de communication incompatibles, etc. L'intégration des bases de données dans un tout logique peut empêcher cette façon de penser.

Pour commencer la discussion sur **les SGBD distribués**, nous donnons d'abord quelques définitions.

**Distribué base de données :** Une collection logiquement interdépendante de données partagées (et une description de ces données) physiquement réparties sur un réseau informatique.

**SGBD Distribuée:** Le système logiciel qui permet la gestion de la base de données distribuée et rend la distribution transparente pour les utilisateurs.

**Un système de gestion de base de données distribuée (SGBDD)** se compose d'une seule base de données logique qui est divisée en un certain nombre de fragments. Chaque fragment est stocké sur un ou plusieurs ordinateurs sous le contrôle d'un SGBD distinct, avec les ordinateurs connecté par un réseau de communication. Chaque site est capable de traiter les demandes des utilisateurs qui nécessitent un accès à des données locales (c'est-à-dire que chaque site a un certain degré d'autonomie locale) et est également capable de traiter des données stockées sur d'autres ordinateurs du réseau.

Les utilisateurs accèdent à la base de données distribuée via des applications, qui sont classées comme celles qui ne nécessitent pas de données provenant d'autres sites (applications locales) et celles qui le font nécessitent des données provenant d'autres sites (applications mondiales). Nous avons besoin d'un SGBDD pour avoir au moins une application globale. Un SGBDD présente donc les caractéristiques suivantes:

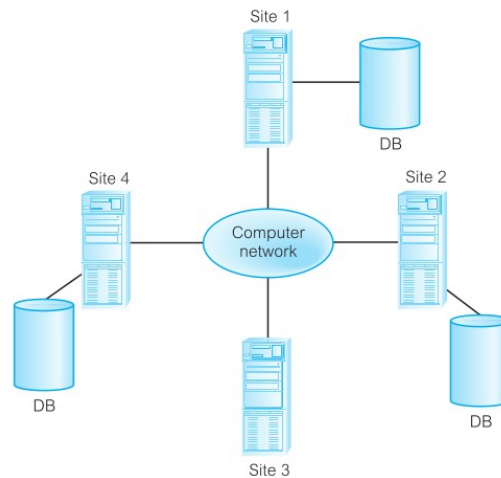
- une collection de données partagées logiquement liées;
- les données sont divisées en un certain nombre de fragments;
- les fragments peuvent être répliqués;
- les fragments / répliques sont attribués aux sites;
- les sites sont reliés par un réseau de communication;
- les données de chaque site sont sous le contrôle d'un SGBD;
- le SGBD de chaque site peut gérer les applications locales de manière autonome;
- chaque SGBD participe à au moins une application globale.

Il n'est pas nécessaire que chaque site du système dispose de sa propre base de données locale, car illustrée par la topologie du SGBDD illustrée à la figure 24.1.

### Exemple 24.1: DreamHome

En utilisant la technologie de base de données distribuée, DreamHome peut implémenter son système de base de données sur un certain nombre de systèmes informatiques distincts plutôt que sur un seul ordinateur central centralisé. Les systèmes informatiques peuvent être situés dans chaque succursale locale: par exemple, Londres, Aberdeen et Glasgow. Un réseau reliant les ordinateurs activera les succursales pour communiquer entre elles et un SGBDD leur permettra d'accéder aux données stockées dans une autre succursale. Ainsi, un client résidant à Glasgow peut se rendre au plus proche succursale pour savoir quelles propriétés sont disponibles à Londres, plutôt que d'avoir à téléphoner ou écrire à la succursale de Londres pour plus de détails.

Alternativement, si chaque succursale DreamHome dispose déjà de sa propre base de données (disparate), un SGBDD peut être utilisé pour intégrer les bases de données séparées dans une base de données logique unique, rendant à nouveau les données locales plus largement disponibles.



**Figure 24.1** Système de gestion de base de données distribuée.

A partir de la définition du SGBDD, le système est censé rendre la distribution transparente (invisible) pour l'utilisateur. Ainsi, le fait qu'une base de données distribuée soit divisée en fragments qui peuvent être stockés sur différents ordinateurs et peut-être répliqués, doit être caché à l'utilisateur. L'objectif de la transparence est de faire apparaître le système distribué comme un système centralisé. Ceci est parfois appelé le principe fondamental des SGBD distribués (Date, 1987b). Cette exigence fournit des fonctionnalités importantes pour l'utilisateur final mais malheureusement crée naturellement de nombreux problèmes supplémentaires qui doivent être traités par le SGBDD, comme nous le verrons dans la section 24.5.

#### Traitement distribué

Il est important de faire la distinction entre un **SGBD distribué** et un **traitement distribué**.

**Traitement distribué:** La base de données centralisée accessible via un réseau informatique.

Le point clé de la définition d'un **SGBD distribué** est que le système se compose de données qui sont physiquement réparties sur un certain nombre de sites du réseau. Si les données sont centralisées, même si d'autres utilisateurs peuvent accéder aux données via le réseau, nous ne considérons pas cela comme un SGBD distribué mais simplement distribué en traitement. Nous illustrons la topologie du traitement distribué dans la figure 24.2

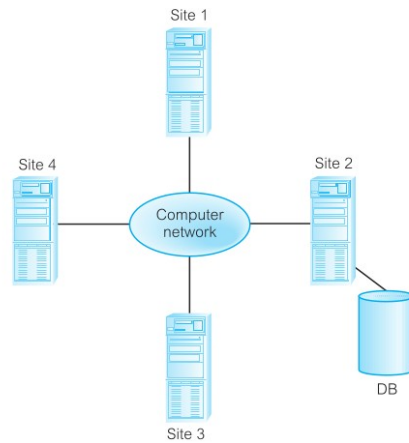


Figure 24.2: Traitement distribué.

### SGBD parallèles

Nous faisons également une distinction entre un SGBD distribué et un SGBD parallèle.

**SGBD parallèle:** Un SGBD fonctionnant sur plusieurs processeurs et disques conçu pour exécuter des opérations en parallèle, dans la mesure du possible, afin d'améliorer les performances.

**Les SGBD parallèles** reposent à nouveau sur le principe que les systèmes à processeur unique ne peuvent plus répondre aux exigences croissantes en matière d'évolutivité, de fiabilité et de performances rentables. Une alternative puissante et financièrement intéressante à un SGBD piloté par un seul processeur est un SGBD parallèle piloté par plusieurs processeurs. SGBD parallèles reliez plusieurs machines plus petites pour obtenir le même débit qu'un seul, plus grand machine, souvent avec une évolutivité et une fiabilité supérieures à celles des SGBD à processeur unique. Pour fournir à plusieurs processeurs un accès commun à une seule base de données, un SGBD parallèle doit permettre la gestion des ressources partagées. Les ressources partagées et la manière dont ces ressources partagées sont mises en œuvre affectent directement les performances et l'évolutivité du système, ce qui détermine à son tour sa pertinence pour une application / un environnement donné. Les trois principales architectures pour le parallèle Les SGBD, comme illustré dans la Figure 24.3, sont:

- la mémoire partagée;
- disque partagé;
- n'a rien partagé.

Figure 24.3 Architectures de bases de données parallèles: (a) mémoire partagée; (b) disque partagé; (c) n'a rien partagé.

La mémoire partagée est une architecture étroitement couplée dans laquelle plusieurs processeurs dans une seule mémoire système de partage de système. Connu sous le nom de multitraitement symétrique (SMP), cette approche est devenue populaire sur des plateformes allant du personnel stations de travail prenant en charge quelques microprocesseurs en parallèle, jusqu'aux grandes machines basées sur RISC (Reduced Instruction Set Computer), jusqu'aux plus grands mainframes. Cette architecture fournit un accès aux données à haut débit pour un nombre limité de processeurs, mais elle n'est pas évolutive au-delà d'environ 64 processeurs, auquel point le réseau d'interconnexion devient un goulot d'étranglement.

Le disque partagé est une architecture faiblement couplée optimisée pour les applications qui sont intrinsèquement centralisées et qui nécessitent une disponibilité et des performances élevées. Chaque processeur peut accéder directement à tous les disques, mais chacun a sa propre mémoire privée. Comme le partagé

### 24.1.2 Avantages et inconvénients des SGBDD

La distribution des données et des applications présente des avantages potentiels par rapport aux systèmes de bases de données centralisées. Malheureusement, il y a aussi des inconvénients. Dans cette section, nous passons en revue les avantages et les inconvénients du **SGBDD**.

#### Avantages

**Reflète la structure organisationnelle** De nombreuses organisations sont naturellement distribuées sur plusieurs emplacements. Par exemple, DreamHome a de nombreux bureaux dans différentes villes. Il est naturel que les bases de données utilisées dans une telle application soient distribuées sur ces emplacements. DreamHome peut conserver une base de données dans chaque succursale contenant des détails sur des éléments tels que le personnel qui travaille à cet endroit, les propriétés à louer et les clients qui possèdent ou souhaitent louer ces propriétés. Le personnel d'une succursale effectuera des recherches locales sur la base de données. Le siège de l'entreprise peut souhaiter effectuer des enquêtes globales impliquant l'accès à des données à tous ou à un certain nombre de succursales.

**Partage amélioré et autonomie locale** La répartition géographique d'une organisation peut se refléter dans la distribution des données; les utilisateurs d'un site peuvent accéder aux données stockées sur d'autres sites. Les données peuvent être placées sur le site à proximité des utilisateurs qui utilisent normalement ces données. De cette manière, les utilisateurs ont un contrôle local des données et ils peuvent par conséquent établir et appliquer des politiques locales concernant l'utilisation de ces données. Un DBA global est responsable de l'ensemble du système. En général, une partie de cette responsabilité est dévolue au niveau local, de sorte que le DBA local puisse gérer le SGBD local (voir section 10.15).

**Amélioration de la disponibilité** Dans un SGBD centralisé, une panne informatique met fin au opérations du SGBD. Cependant, une défaillance sur un site d'un SGBDD ou une défaillance d'une liaison de communication rendant certains sites inaccessibles ne rend pas l'ensemble du système inopérable. Les SGBD distribués sont conçus pour continuer à fonctionner malgré de telles défaillances. Si un seul nœud tombe en panne, le système peut être en mesure de rediriger les demandes du nœud ayant échoué vers un autre site.

**Fiabilité améliorée** Comme les données peuvent être répliquées de manière à exister sur plusieurs sites, la défaillance d'un nœud ou d'une liaison de communication n'entraîne pas nécessairement les données inaccessibles.

**Amélioration des performances** Les données étant situées à proximité du site "le plus sollicité", et étant donné le parallélisme inhérent aux SGBD distribués, la vitesse de la base de données l'accès peut être meilleur que celui réalisable à partir d'une base de données centralisée distante. De plus, étant donné que chaque site ne gère qu'une partie de la base de données entière, il peut ne pas y avoir le même conflit pour les services CPU et E / S caractérisés par un SGBD centralisé.

**Économie** Dans les années 1960, la puissance de calcul était calculée selon la carré des coûts de l'équipement: trois fois le coût fournirait neuf fois la puissance. C'était la loi de Grosch. Cependant, il est maintenant généralement admis qu'il coûte beaucoup moins cher de créer un système d'ordinateurs plus petits avec la puissance équivalente d'un seul gros ordinateur. Cela le rend plus rentable pour divisions et départements de l'entreprise pour obtenir des ordinateurs séparés. C'est

aussi beaucoup plus rentable d'ajouter des postes de travail à un réseau que de mettre à jour un mainframe système.

La deuxième économie de coûts potentielle se produit lorsque les bases de données sont géographiquement distantes et que les applications nécessitent un accès aux données distribuées. Dans de tels cas, en raison du coût relatif des données transmises sur le réseau par opposition à le coût de l'accès local, il peut être beaucoup plus économique de partitionner l'application et d'effectuer le traitement localement sur chaque site.

**Croissance modulaire** Dans un environnement distribué, il est beaucoup plus facile à gérer l'expansion. De nouveaux sites peuvent être ajoutés au réseau sans affecter les opérations d'autres sites. Cette flexibilité permet à une organisation de se développer relativement facilement. L'augmentation de la taille de la base de données peut généralement être gérée en ajoutant le traitement et le stockage au réseau. Dans un SGBD centralisé, la croissance peut impliquer des changements à la fois matériels (l'achat d'un système plus puissant) et logiciels (l'achat d'un SGBD plus puissant ou plus configurable). La transition vers le nouveau matériel / logiciel pourrait donner lieu à de nombreuses difficultés.

**Intégration** Au début de cette section, nous avons noté que l'intégration était un avantage clé. L'approche du SGBD, pas la centralisation. L'intégration des systèmes hérités est un exemple particulier qui montre comment certaines organisations sont forcées de s'appuyer sur un traitement de données distribué pour permettre à leurs anciens systèmes de coexister avec leurs systèmes plus modernes. Dans le même temps, aucun package ne peut fournir toutes les fonctionnalités dont une organisation a besoin de nos jours. Ainsi, il est important pour les organisations de pouvoir intégrer des composants logiciels de différents fournisseurs pour répondre à leurs besoins spécifiques.

**Rester compétitif** Il y a un certain nombre de développements relativement récents qui reposent fortement sur la technologie de base de données distribuée telle que le commerce électronique, prise en charge du travail collaboratif et de la gestion des flux de travail. De nombreuses entreprises ont dû réorganiser leurs activités et utiliser la technologie des bases de données distribuées pour rester compétitif. Par exemple, bien que plus de personnes ne loueront pas nécessairement propriétés simplement parce qu'Internet existe, DreamHome peut perdre une partie de son marché s'il n'autorise pas les clients à afficher les propriétés en ligne maintenant.

### Désavantages

**Complexité** Un SGBD distribué qui cache la nature distribuée à l'utilisateur et fournit un niveau acceptable de performances, de fiabilité et de disponibilité est intrinsèquement plus complexe qu'un SGBD centralisé. Le fait que les données puissent être répliquées ajoute également un niveau supplémentaire de complexité au SGBD distribué. Si le logiciel ne gère pas correctement la réplication des données, il y aura dégradation de la disponibilité, la fiabilité et les performances par rapport au système centralisé, et les avantages que nous avons cités plus tôt deviendront des inconvénients.

**Coût** Une complexité accrue signifie que nous pouvons nous attendre à ce que les coûts d'approvisionnement et de maintenance d'un DDBMS soient plus élevés que ceux d'un SGBD centralisé. De plus, un SGBD distribué nécessite du matériel supplémentaire pour établir un réseau travaillant entre les sites. Il y a des frais de communication continus liés à l'utilisation de ce réseau. Il existe également des coûts de main-d'œuvre supplémentaires pour gérer et entretenir les SGBD locaux et le réseau sous-jacent.

**Sécurité** Dans un système centralisé, l'accès aux données peut être facilement contrôlé. Cependant, dans un SGBD distribué, non seulement l'accès aux données répliquées doit être contrôlé dans

**Sécurité** plusieurs emplacements, mais le réseau lui-même doit être sécurisé. Dans le passé, les réseaux étaient considérés comme un moyen de communication peu sûr. Bien que ce soit encore partiellement Certes, des développements significatifs ont été réalisés pour rendre les réseaux plus sûrs.

**Contrôle d'intégrité plus difficile** L'intégrité de la base de données fait référence à la validité et à la cohérence des données stockées. L'intégrité s'exprime généralement en termes de contraintes, qui sont des règles de cohérence que la base de données n'est pas autorisée à enfreindre. Faire respecter l'intégrité contraintes nécessitent généralement l'accès à une grande quantité de données qui définissent la contrainte mais qui n'est pas impliqué dans l'opération de mise à jour proprement dite. Dans un SGBD distribuée, les coûts de communication et de traitement nécessaires pour faire respecter l'intégrité des contraintes peuvent être prohibitives. Nous revenons sur ce problème dans la section 25.4.5.

**Manque de normes** Bien que les SGBD distribués dépendent d'une communication efficace, nous commençons à peine à voir l'apparition d'une communication standard

**Tableau 24.1** Résumé des avantages et des inconvénients des DDBMS.

AVANTAGES	Inconvénients
Reflète la structure organisationnelle	Complexité
Partage amélioré et autonomie locale	Coût
Amélioration de la disponibilité	Sécurité
Fiabilité améliorée	Contrôle d'intégrité plus difficile
Amélioration des performances	Manque de normes
Économie	
Croissance modulaire	
Intégration	
Rester compétitif	

et les protocoles d'accès aux données. Ce manque de normes a considérablement limité le potentiel des SGBD distribués. Il n'existe pas non plus d'outils ou de méthodologies pour aider les utilisateurs à convertir un SGBD centralisé en SGBD distribué.

**Manque d'expérience** Les SGBD distribués à usage général n'ont pas été largement acceptés, bien que de nombreux protocoles et problèmes soient bien compris. Par conséquent, nous n'avons pas encore le même niveau d'expérience dans l'industrie que nous avons avec les SGBD centralisés. Pour un adoptant potentiel de cette technologie, cela peut être un élément dissuasif important.

**Conception de base de données plus complexe** Outre les difficultés normales de conception d'une base de données centralisée, la conception d'une base de données distribuée doit tenir compte de la fragmentation des données, de l'attribution de fragments à des sites spécifiques et de la réplication des données. Nous discutons de ces problèmes dans la section 24.4.

Les avantages et inconvénients des DDBMS sont résumés dans le tableau 24.1.



### 24.1.3 SGBDD homogènes et hétérogènes

Un **SGBDD** peut être classé comme homogène ou hétérogène. Dans un système homogène, tous les sites utilisent le même produit SGBD. Dans un système hétérogène, les sites peuvent exécuter différents produits de SGBD, qui n'ont pas besoin d'être basés sur le même modèle de données sous-jacent, et ainsi le système peut être composé de SGBD relationnels, réseau, hiérarchiques et orientés objet.

**Les systèmes homogènes** sont beaucoup plus faciles à concevoir et à gérer. Cette approche permet une croissance incrémentielle, facilitant l'ajout d'un nouveau site au SGBDD, et permet des performances accrues en exploitant la capacité de traitement parallèle de plusieurs sites.

Les systèmes hétérogènes se produisent généralement lorsque des sites individuels ont mis en œuvre leurs propres bases de données et que l'intégration est envisagée à un stade ultérieur. Dans un système hétérogène, des traductions sont nécessaires pour permettre la communication entre différents SGBD. Pour assurer la transparence du SGBD, les utilisateurs doivent pouvoir effectuer des requêtes dans la langue du SGBD sur leur site local. Le système a alors pour tâche de localiser les données et d'effectuer toute traduction nécessaire. Des données peuvent être requises à partir d'un autre site qui peut avoir:

- matériel différent;
- différents produits de SGBD;
- différents matériels et différents produits SGBD.

Si le matériel est différent mais que les produits SGBD sont identiques, la traduction est simple, impliquant le changement de codes et de longueurs de mots. Si les produits du SGBD sont différents, la traduction est compliquée impliquant le mappage des structures de données dans un modèle de données aux structures de données équivalentes dans un autre modèle de données. Par exemple, les relations dans le modèle de données relationnelles sont mappées aux enregistrements et aux ensembles dans le modèle de réseau. Il est également nécessaire de traduire le langage de requête utilisé (par exemple, les instructions SQL SELECT sont mappées aux instructions réseau FIND et GET). Si le matériel et le logiciel sont différents, ces deux types de traduction sont nécessaires. Cela rend le traitement extrêmement complexe.

Une complexité supplémentaire est la fourniture d'un schéma conceptuel commun, qui est formé à partir de l'intégration de schémas conceptuels locaux individuels. Comme vous l'avez déjà vu à l'étape 2.6 de la méthodologie de conception de base de données logique présentée au chapitre 17, l'intégration des modèles de données peut être très difficile en raison de l'hétérogénéité sémantique. Par exemple, les attributs portant le même nom dans deux schémas peuvent représenter des choses différentes. De même, des attributs avec des noms différents peuvent modéliser la même chose. Une discussion complète sur la détection et la résolution

l'hétérogénéité sémantique dépasse le cadre de ce livre. Le lecteur intéressé est renvoyé à l'article de Garcia-Solaco et al. (1996). La solution typique utilisée par certains systèmes relationnels qui font partie d'un DDBMS hétérogène consiste à utiliser des passerelles, qui convertissent le langage et le modèle de chaque SGBD différent en langage et modèle du système relationnel. Cependant, l'approche de la passerelle présente de sérieuses limitations. Premièrement, il peut ne pas prendre en charge la gestion des transactions, même pour une paire de systèmes; en d'autres termes, la passerelle entre deux systèmes peut n'être qu'un traducteur de requêtes. Par exemple, un système peut ne pas coordonner le contrôle d'accès concurrentiel et la récupération des transactions qui impliquent des mises à jour de la paire de bases de données. Deuxièmement, l'approche de la passerelle ne concerne que le problème de la traduction d'une requête exprimée dans une langue en une expression équivalente dans une autre langue. En tant que tel, il n'aborde généralement pas les problèmes d'homogénéisation des différences structurelles et de représentation entre les différents schémas.

**Accès aux bases de données ouvertes et interopérabilité** Le groupe ouvert a formé un groupe de travail sur les spécifications (SWG) pour répondre à un livre blanc sur l'accès ouvert aux bases de



données et l'interopérabilité (Gualtieri, 1996). L'objectif de ce groupe était de fournir des spécifications ou de s'assurer que les spécifications

existent ou sont en cours de développement qui créeront un environnement d'infrastructure de base de données où il y a :

- une API SQL commune et puissante qui permet d'écrire des applications clientes qui n'ont pas besoin de connaître le fournisseur du SGBD auquel elles accèdent;
- un protocole de base de données commun qui permet à un SGBD d'un fournisseur de communiquer directement avec un SGBD d'un autre fournisseur sans avoir besoin d'une passerelle;
- un protocole de réseau commun qui permet les communications entre différents SGBD.

L'objectif le plus ambitieux est de trouver un moyen de permettre à une transaction de s'étendre sur des bases de données gérées par des SGBD de différents fournisseurs sans utiliser de passerelle.

Cette

Le groupe de travail a évolué pour devenir le Consortium d'interopérabilité des bases de données (DBIOP), travaillant sur la version 3 de l'architecture de base de données relationnelle distribuée (DRDA), que nous abordons brièvement dans la section 24.5.2.

### Systèmes multi-bases de données

Avant de terminer cette section, nous discutons brièvement d'un type particulier de SGBD distribué connu sous le nom de système multidatabase.

#### Système multi-bases de données (MDBS)

Un SGBD distribué dans lequel chaque site maintient complet autonomie.

Ces dernières années, il y a eu un intérêt considérable pour les MDBS, qui tentent d'intégrer logiquement un certain nombre de DDBMS indépendants tout en permettant aux SGBD locaux de garder le contrôle complet de leurs opérations. Une des conséquences de l'autonomie complète est qu'il ne peut y avoir aucune modification logicielle des SGBD locaux. Ainsi, un MDBS nécessite une couche logicielle supplémentaire au-dessus des systèmes locaux pour fournir les fonctionnalités nécessaires.

Un MDBS permet aux utilisateurs d'accéder et de partager des données sans nécessiter une intégration complète du schéma de base de données. Cependant, il permet toujours aux utilisateurs d'administrer leurs propres bases de données sans contrôle centralisé, comme avec les vrais DDBMS. Le DBA d'un SGBD local peut autoriser l'accès à des parties particulières de sa base de données en spécifiant un schéma d'exportation, qui définit les parties de la base de données auxquelles les utilisateurs non locaux peuvent accéder. Il existe des MDBS non fédérés (là où il n'y a pas d'utilisateurs locaux) et fédérés. Un système fédéré est un croisement entre un SGBD distribué et un système centralisé

SGBD; c'est un système distribué pour les utilisateurs mondiaux et un système centralisé pour les utilisateurs locaux. La figure 24.4 illustre une taxonomie partielle des SGBD (voir également la figure 26.20).

Le lecteur intéressé est renvoyé à Sheth et Larson (1990) pour une taxonomie des SGBD distribués et à Bukhres et Elmagarmid (1996).

En termes simples, un MDBS est un SGBD qui réside de manière transparente au-dessus des bases de données et des systèmes de fichiers existants, et présente une base de données unique à ses utilisateurs. Un MDBS gère uniquement le schéma global par rapport auquel les utilisateurs émettent des requêtes et des mises à jour et les SGBD locaux eux-mêmes conservent toutes les données utilisateur. Le schéma global est construit en intégrant les schémas des bases de données locales. Le MDBS traduit d'abord les requêtes globales et les mises à jour en requêtes et mises à jour sur les SGBD locaux appropriés. Il fusionne ensuite les résultats locaux et génère le résultat global final

## 24.3 Fonctions et architectures d'un SGBDD

Au chapitre 2, nous avons examiné les fonctions, l'architecture et les composants d'un SGBD centralisé. Dans cette section, nous examinons comment la distribution affecte les fonctionnalités et l'architecture attendues.

### 24.3.1 Fonctions d'un SGBDD

Nous nous attendons à ce qu'un **SGBDD** ait au moins la fonctionnalité qu'un SGBD centralisé dont nous avons parlé au chapitre 2. De plus, nous nous attendons à ce qu'un **SGBDD** ait les fonctionnalités suivantes:

- des services de communication étendus pour donner accès à des sites distants et permettre le transfert de requêtes et de données entre les sites à l'aide d'un réseau;
- catalogue système étendu pour stocker les détails de distribution des données;
- traitement distribué des requêtes, y compris l'optimisation des requêtes et l'accès aux données à distance;
- un contrôle de sécurité étendu pour maintenir les privilèges d'autorisation / d'accès appropriés aux données distribuées;
- un contrôle d'accès concurrentiel étendu pour maintenir la cohérence des données distribuées et éventuellement répliquées;
- des services de reprise étendus pour prendre en compte les défaillances des sites individuels et les défaillances des liaisons de communication.

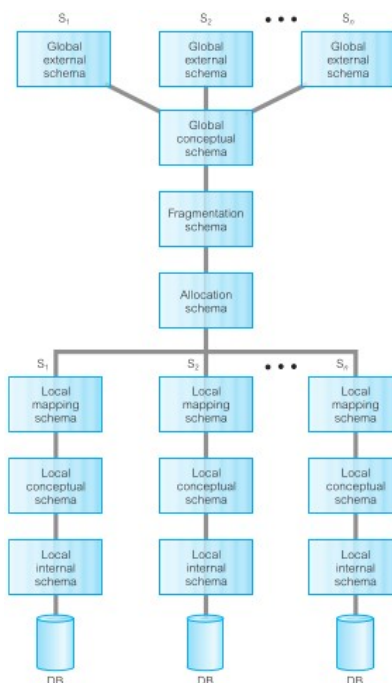
Nous aborderons ces questions plus en détail dans les sections ultérieures de ce chapitre et au chapitre 25.

### 24.3.2 Architecture de référence pour un SGBDD

L'architecture à trois niveaux ANSI-SPARC pour un SGBD présentée dans la section 2.1 fournit une architecture de référence pour un SGBD centralisé. En raison de la diversité des SGBD distribués, il est beaucoup plus difficile de présenter une architecture équivalente généralement applicable. Cependant, il peut être utile de présenter une architecture de référence possible qui traite de la distribution des données. L'architecture de référence illustrée à la figure 24.5 comprend les schémas suivants:

- un ensemble de schémas externes globaux;
- un schéma conceptuel global;
- un schéma de fragmentation et un schéma d'allocation;
- un ensemble de schémas pour chaque SGBD local conforme à l'architecture à trois niveaux ANSI-SPARC.

Les arêtes de cette figure représentent les mappages entre les différents schémas. Selon les niveaux de transparence pris en charge, certains niveaux peuvent être absents de l'architecture.



**Figure 24.5** Référence architecture pour un DDBMS.

### Schéma conceptuel global

Le schéma conceptuel global est une description logique de l'ensemble de la base de données, comme si elle n'était pas distribuée. Ce niveau correspond au niveau conceptuel de l'architecture ANSI-SPARC et contient des définitions d'entités, de relations, de contraintes, d'informations de sécurité et d'intégrité. Il fournit une indépendance des données physiques l'environnement distribué. Les schémas externes globaux fournissent une indépendance logique des données.

### Schémas de fragmentation et d'allocation

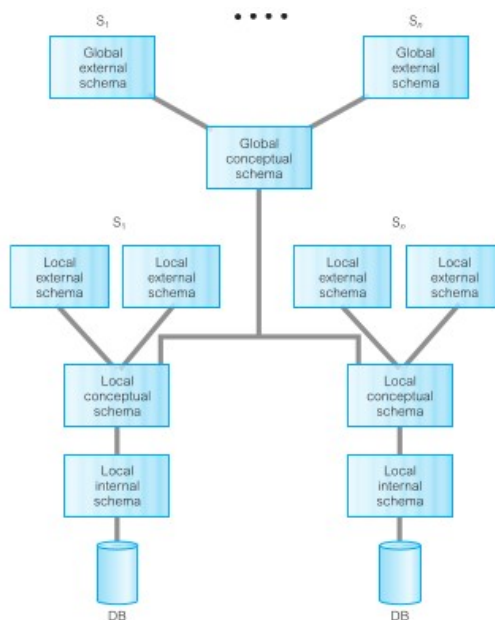
**Le schéma de fragmentation** est une description de la manière dont les données doivent être partitionnées logiquement. **Le schéma d'allocation** est une description de l'emplacement des données, en tenant compte de toute réplication.

### Schémas locaux

Chaque SGBD local possède son propre ensemble de schémas. Les schémas conceptuels locaux et internes locaux correspondent aux niveaux équivalents de l'architecture ANSI-SPARC. Le schéma de mappage local mappe les fragments du schéma d'allocation en objets externes dans la base de données locale. Il est indépendant du SGBD et constitue la base de la prise en charge des SGBD hétérogènes.

### 24.3.3 Architecture de référence pour un MDBS fédéré

Dans la section 24.1.3, nous avons brièvement discuté des systèmes fédérés multidatabase (FMDBS). Les systèmes fédérés diffèrent des SGBDD par le niveau d'autonomie locale fourni. Cette différence se reflète également dans l'architecture de référence. La figure 24.6 illustre une architecture de référence pour un FMDBS qui est étroitement couplé, c'est-à-dire qu'il a



**Figure 24.6** Architecture de référence pour un FMDBS étroitement couplé

un schéma conceptuel global (GCS). Dans un SGBDD, le GCS est l'union de tous les schémas conceptuels locaux. Dans un FMDBS, le GCS est un sous-ensemble des schémas conceptuels locaux, constitué des données que chaque système local accepte de partager. Le GCS d'un système étroitement couplé implique l'intégration soit des parties des schémas conceptuels locaux, soit des schémas externes locaux.

Il a été avancé qu'un FMDBS ne devrait pas avoir de GCS (Litwin, 1988), auquel cas le système est appelé faiblement couplé. Dans ce cas, les schémas externes consistent en un ou plusieurs schémas conceptuels locaux. Pour plus d'informations sur les MDBS, le lecteur intéressé est renvoyé à Litwin (1988) et Sheth et Larson (1990).

#### 24.3.4 Architecture des composants pour un SGBDD

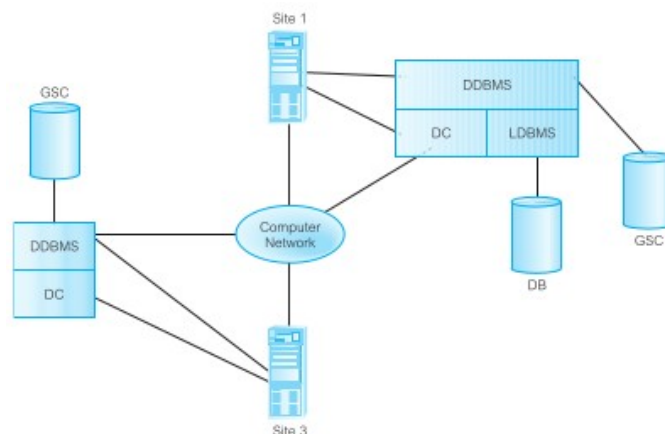
Indépendamment de l'architecture de référence, nous pouvons identifier une architecture de composants pour un SGBDD composé de quatre composants majeurs:

- composant de SGBD local (LDBMS);
- composante de communication de données (DC);
- catalogue système global (GSC);
- composant de SGBD distribué (SGBDD).

L'architecture des composants d'un SGBDD basée sur la Figure 24.1 est illustrée dans la Figure 24.7. Pour plus de clarté, nous avons omis le site 2 du diagramme, car il a la même structure que le site 1.

##### Composant SGBD local

Le composant LDBMS est un SGBD standard, chargé de contrôler les données locales sur chaque site disposant d'une base de données. Il possède son propre catalogue système local qui stocke des informations sur les données conservées sur ce site. Dans un système homogène, le LDBMS



**Figure 24.7** Composants d'un SGBDD.

composant est le même produit, répliqué sur chaque site. Dans un système hétérogène, il y aurait au moins deux sites avec des produits et / ou plates-formes de SGBD différents.

##### Composant de communication de données

Le composant DC est le logiciel qui permet à tous les sites de communiquer entre eux. Le composant DC contient des informations sur les sites et les liens.

##### Catalogue système global

Le GSC a la même fonctionnalité que le catalogue système d'un système centralisé. Le GSC détient des informations spécifiques à la nature distribuée du système, telles que les schémas de fragmentation, de réplication et d'allocation. Elle peut elle-même être gérée comme une base de données distribuée et ainsi être fragmentée et distribuée, entièrement répliquée ou centralisée, comme toute autre relation, comme nous le verrons bientôt. Un GSC entièrement répliqué compromet l'autonomie du site, car chaque modification du GSC doit être communiquée à tous les

autres sites. Un GSC centralisé compromet également l'autonomie du site et est vulnérable à la défaillance du site central.

L'approche adoptée dans le système distribué R \* surmonte ces défaillances (Williams et al., 1982). Dans R \*, il y a un catalogue local sur chaque site qui contient les métadonnées relatives aux données stockées sur ce site. Pour les relations créées sur un site (le site de naissance), il est de la responsabilité du catalogue local de ce site d'enregistrer la définition de chaque fragment et de chaque réplique de chaque fragment, et d'enregistrer l'emplacement de chaque fragment ou réplique. Chaque fois qu'un fragment ou une réplique est déplacé vers un autre emplacement, le catalogue local du site de naissance de la relation correspondante doit être mis à jour. Ainsi, pour localiser un fragment ou une réplique d'une relation, il faut accéder au catalogue du site de naissance de la relation. Le site de naissance de chaque relation globale est enregistré dans chaque GSC local. Nous revenons à la dénomination des objets lorsque nous discutons de la dénomination transparente dans la section 24.5.1.

### **Composant SGBD distribué**

Le composant DDBMS est l'unité de contrôle de l'ensemble du système. Nous avons brièvement énuméré les fonctionnalités de ce composant dans la section précédente et nous nous concentrons sur cette fonctionnalité dans la section 24.5 et dans le chapitre 25.