INTRODUCTION

1. QU'EST-CE QU'UNE BASE DE DONNÉES ?

Les bases de données ont pris aujourd'hui une place essentielle dans l'informatique, plus particulièrement en gestion. Au cours des trente dernières années, des concepts, méthodes et algorithmes ont été développés pour gérer des données sur mémoires secondaires ; ils constituent aujourd'hui l'essentiel de la discipline « Bases de Données » (BD). Cette discipline est utilisée dans de nombreuses applications. Il existe un grand nombre de Systèmes de Gestion de Bases de Données (SGBD) qui permettent de gérer efficacement de grandes bases de données. De plus, une théorie fondamentale sur les techniques de modélisation des données et les algorithmes de traitement a vu le jour. Les bases de données constituent donc une discipline s'appuyant sur une théorie solide et offrant de nombreux débouchés pratiques.

Vous avez sans doute une idée intuitive des bases de données. Prenez garde cependant, car ce mot est souvent utilisé pour désigner n'importe quel ensemble de données ; il s'agit là d'un abus de langage qu'il faut éviter. Une base de données est un ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique. Pour mériter le terme de base de données, un ensemble de données non indépendantes doit être interrogeable par le contenu, c'est-à-dire que l'on doit pouvoir retrouver tous les objets qui satisfont à un certain critère, par exemple tous les produits qui coûtent moins de 100 francs. Les données doivent être interrogeables selon n'importe quel critère. Il doit être possible aussi de retrouver leur structure, par exemple le fait qu'un produit possède un nom, un prix et une quantité.

Plutôt que de disserter longuement sur le concept de bases de données, précisons ce qu'est un SGBD. Un SGBD peut être perçu comme un ensemble de logiciels systèmes permettant aux utilisateurs d'insérer, de modifier et de rechercher efficacement des données spécifiques dans une grande masse d'informations (pouvant atteindre quelques milliards d'octets) partagée par de multiples utilisateurs. Les informations sont stockées

sur mémoires secondaires, en général des disques magnétiques. Les recherches peuvent être exécutée à partir de la valeur d'une donnée désignée par un nom dans un ensemble d'objets (par exemple, les produits de prix inférieur à 100 francs), mais aussi à partir de relations entre objets (par exemple, les produits commandés par un client habitant Paris). Les données sont partagées, aussi bien en interrogation qu'en mise à jour. Le SGBD rend transparent le partage, à savoir donne l'illusion à chaque utilisateur qu'il est seul à travailler avec les données.

En résumé, un SGBD peut donc apparaître comme un outil informatique permettant la sauvegarde, l'interrogation, la recherche et la mise en forme de données stockées sur mémoires secondaires. Ce sont là les fonctions premières, complétées par des fonctions souvent plus complexes, destinées par exemple à assurer le partage des données mais aussi à protéger les données contre tout incident et à obtenir des performances acceptables. Les SGBD se distinguent clairement des systèmes de fichiers par le fait qu'ils permettent la description des données (définition des types par des noms, des formats, des caractéristiques et parfois des opérations) de manière séparée de leur utilisation (mise à jour et recherche). Ils permettent aussi de retrouver les caractéristiques d'un type de données à partir de son nom (par exemple, comment est décrit un produit). Le système de fichiers est un composant de plus bas niveau ne prenant pas en compte la structure des données. La tendance est aujourd'hui à intégrer le système de fichiers dans le SGBD, construit au-dessus.

En conséquence, un SGBD se compose en première approximation de trois couches emboîtées de fonctions, depuis les mémoires secondaires vers les utilisateurs (voir figure I.1) :

- La gestion des récipients de données sur les mémoires secondaires constitue traditionnellement la première couche ; c'est le **gestionnaire de fichiers**, encore appelé système de gestion de fichiers. Celui-ci fournit aux couches supérieures des mémoires secondaires idéales adressables par objets et capables de recherches par le contenu des objets (mécanismes d'indexation notamment).

- La gestion des données stockées dans les fichiers, l'assemblage de ces données en objets, le placement de ces objets dans les fichiers, la gestion des liens entre objets et des structures permettant d'accélérer les accès aux objets constituent la deuxième couche ; c'est le système d'accès aux données ou **SGBD interne**. Celui-ci repose généralement sur un modèle de données internes, par exemple des tables reliées par des pointeurs.

- La fonction essentielle de la troisième couche consiste dans la mise en forme et la présentation des données aux programmes d'applications et aux utilisateurs interactifs. Ceux-ci expriment leurs critères de recherches à l'aide de langages basés sur des procédures de recherche progressives ou sur des assertions de logiques, en référençant des données dérivées de la base ; c'est le **SGBD externe** qui assure d'une part l'analyse et l'interprétation des requêtes utilisateurs en primitives internes, d'autre part la transformation des données extraites de la base en données échangées avec le monde extérieur.

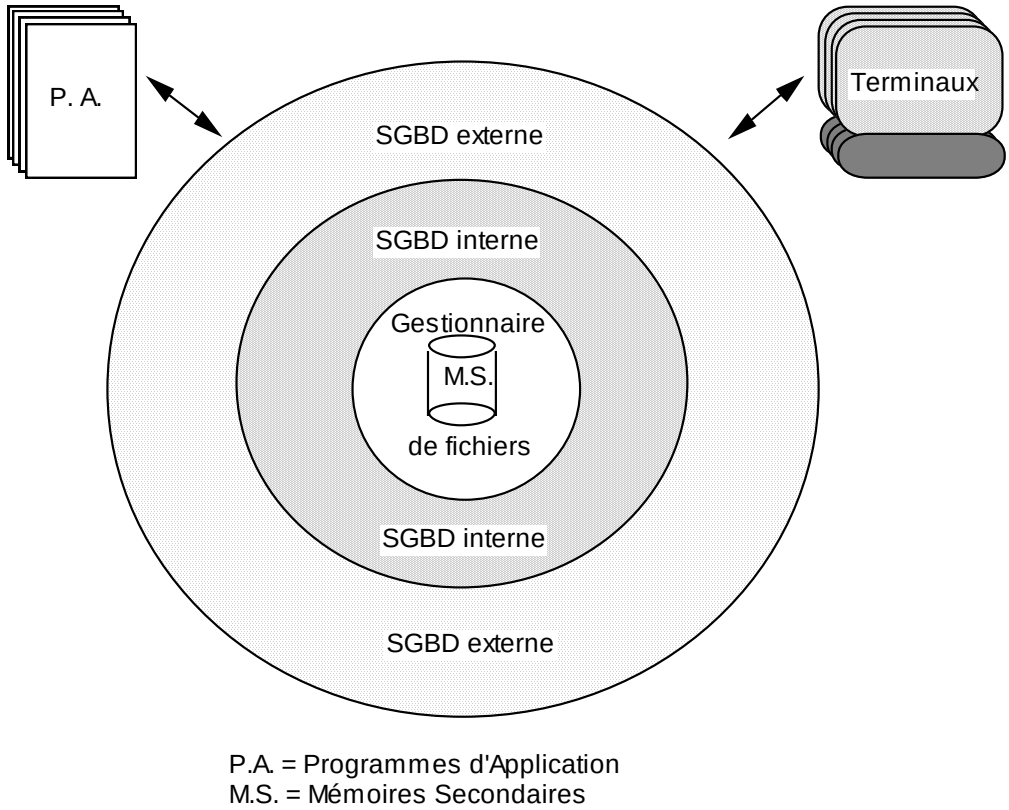

Figure I.1 — Première vue d'un SGBD

Ces couches de fonctions constituent sans doute seulement la moitié environ du code d'un SGBD. En effet, au-delà de ses fonctions de recherche, de rangement et de présentation, un SGBD gère des problèmes difficiles de partage et de cohérence de données. Il protège aussi les données contre les accès non autorisés. Ces fonctions qui peuvent paraître annexes sont souvent les plus difficiles à réaliser et nécessitent beaucoup de code.

Pour être complet, signalons qu'au-dessus des SGBD les systèmes d'informations intègrent aujourd'hui de plus en plus souvent des ateliers de génie logiciel permettant de modéliser les données d'une base de données et de représenter les traitements associés à l'aide de graphiques et de langages de spécifications. Ces outils d'aide à la conception, bien que non intégrés dans le SGBD, permettent de spécifier les descriptions des données. Ils s'appuient pour cela sur les modèles de données décrits dans cet ouvrage et supportés par les SGBD.

2. HISTORIQUE DES SGBD

Les SGBD ont bientôt quarante ans d'histoire. Les années 60 ont connu un premier développement des bases de données sous forme de fichiers reliés par des pointeurs. Les fichiers sont composés d'articles stockés les uns à la suite des autres et accessibles par des valeurs de données appelées clés. Les systèmes IDS.I et IMS.I développés respectivement à Honeywell et à IBM vers 1965 pour les programmes de conquête spatiale, notamment pour le programme APOLLO qui a permis d'envoyer un homme

sur la lune, sont les précurseurs des SGBD modernes. Ils permettent de constituer des chaînes d'articles entre fichiers et de parcourir ces chaînes.

Les premiers SGBD sont réellement apparus à la fin des années 60. La première génération de SGBD est marquée par la séparation de la description des données et de la manipulation par les programmes d'application. Elle coïncide aussi avec l'avènement des langages d'accès navigationnels, c'est-à-dire permettant de se déplacer dans des structures de type graphe et d'obtenir, un par un, des articles de fichiers. Cette première génération, dont l'aboutissement est marqué par les recommandations du CODASYL, est basée sur les modèles réseau ou hiérarchique, c'est-à-dire des modèles de données organisés autour de types d'articles constituant les nœuds d'un graphe, reliés par des types de pointeurs composant les arcs du graphe. Cette génération a été dominée par les SGBD TOTAL, IDMS, IDS 2 et IMS 2. Elle traite encore aujourd'hui une partie importante du volume de données gérées par des SGBD.

La deuxième génération de SGBD a grandi dans les laboratoires depuis 1970, à partir du modèle relationnel. Elle vise à enrichir mais aussi à simplifier le SGBD externe afin de faciliter l'accès aux données pour les utilisateurs. En effet, les données sont présentées aux utilisateurs sous forme de relations entre domaines de valeurs, simplement représentées par des tables. Les recherches et mises à jour sont effectuées à l'aide d'un langage non procédural standardisé appelé SQL (*Structured Query Language*). Celui-ci permet d'exprimer des requêtes traduisant directement des phrases simples du langage naturel et de spécifier les données que l'on souhaite obtenir sans dire comment les accéder. C'est le SGBD qui doit déterminer le meilleur plan d'accès possible pour évaluer une requête. Cette deuxième génération reprend, après les avoir faits évoluer et rendus plus souples, certains modèles d'accès de la première génération au niveau du SGBD interne, afin de mieux optimiser les accès. Les systèmes de deuxième génération sont commercialisés depuis 1980. Ils représentent aujourd'hui l'essentiel du marché des bases de données. Les principaux systèmes sont ORACLE, INGRES, SYBASE, INFORMIX, DB2 et SQL SERVER. Ils supportent en général une architecture répartie, au moins avec des stations clients transmettant leurs requêtes à de puissants serveurs gérant les bases.

La troisième génération a été développée dans les laboratoires depuis le début des années 80. Elle commence à apparaître fortement dans l'industrie avec les extensions objet des systèmes relationnels. Elle supporte des modèles de données extensibles intégrant le relationnel et l'objet**,** ainsi que des architectures mieux réparties, permettant une meilleure collaboration entre des utilisateurs concurrents. Cette troisième génération est donc influencée par les modèles à objets, intégrant une structuration conjointe des programmes et des données en types, avec des possibilités de définir des sous-types par héritage. Cependant, elle conserve les acquis du relationnel en permettant une vision tabulaire des objets et une interrogation via le langage SQL étendu aux objets. Elle intègre aussi le support de règles actives plus ou moins dérivées de la logique. Ces règles permettent de mieux maintenir la cohérence des données en répercutant des mises à jour d'un objet sur d'autres objets dépendants. Les systèmes objet-relationnels tels Oracle 8, DB2 Universal Database ou Informix Universal Server, ce dernier issu du système de recherche Illustra, sont les premiers représentants des systèmes de 3^{e} génération. Les systèmes à objets tels ObjectStore ou O2 constituent une voie plus novatrice vers la troisième génération. Tous ces systèmes tentent de répondre

aux besoins des nouvelles applications (multimédia, Web, CAO, bureautique, environnement, télécommunications, etc.).

Quant à la quatrième génération, elle est déjà en marche et devrait mieux supporter Internet et le Web, les informations mal structurées, les objets multimédias, l'aide à la prise de décisions et l'extraction de connaissances à partir des données. Certes, il devient de plus en plus dur de développer un nouvel SGBD. On peut donc penser que les recherches actuelles, par exemple sur l'interrogation par le contenu des objets multimédias distribués et sur l'extraction de connaissances (*data mining*) conduiront à une évolution des SGBD de 3^{e} génération plutôt qu'à une nouvelle révolution. Ce fut déjà le cas lors du passage de la 2^{e} à la 3^{e} génération, la révolution conduite par l'objet ayant en quelque sorte échouée : elle n'a pas réussi à renverser le relationnel, certes bousculé et adapté à l'objet. Finalement, l'évolution des SGBD peut être perçue comme celle d'un arbre, des branches nouvelles naissant mais se faisant généralement absorber par le tronc, qui grossit toujours d'avantage.

3. PLAN DE CET OUVRAGE

Ce livre traite donc de tous les aspects des bases de données relationnelles et objet, mais aussi objet-relationnel. Il est découpé en quatre parties autonomes, elles-mêmes divisées en chapitres indépendants, en principe de difficulté croissante.

La **première partie** comporte, après cette introduction, quatre chapitres fournissant les bases indispensables à une étude approfondie des SGBD.

Le chapitre II ébauche le cadre général de l'étude. Les techniques de modélisation de données sont tout d'abord introduites. Puis les objectifs et les fonctions des SGBD sont développés. Finalement, les architectures fonctionnelles puis opérationnelles des SGBD modernes sont discutées. L'ensemble du chapitre est une introduction aux techniques et problèmes essentiels de la gestion des bases de données, illustrées à partir d'un langage adapté aux entités et associations.

Le chapitre III se concentre sur la gestion des fichiers et les langages d'accès aux fichiers. Certains peuvent penser que la gestion de fichiers est aujourd'hui dépassée. Il n'en est rien, car un bon SGBD s'appuie avant tout sur de bonnes techniques d'accès par hachage et par index. Nous étudions en détail ces techniques, des plus anciennes aux plus modernes, basées sur les indexes multiples et les hachages dynamiques multi-attributs ou des *bitmaps*.

Le chapitre IV traite des modèles légués par les SGBD de première génération. Le modèle réseau tel qu'il est défini par le CODASYL et implanté dans le système IDS.II de Bull est développé. Des exemples sont donnés. Le modèle hiérarchique d'IMS est plus succinctement introduit.

Le chapitre V introduit les fondements logiques des bases de données, notamment relationnelles. Après un rappel succinct de la logique du premier ordre, la notion de bases de données logique est présentée et les calculs de tuples et domaines, à la base des langages relationnels, sont introduits.

La **deuxième partie** est consacrée au relationnel. Le modèle et les techniques de contrôle et d'optimisation associées sont approfondis.

Le chapitre VI introduit le modèle relationnel de Codd et l'algèbre relationnelle associée. Les concepts essentiels pour décrire les données tels qu'ils sont aujourd'hui supportés par de nombreux SGBD sont tout d'abord décrits. Les types de contraintes d'intégrité qui permettent d'assurer une meilleure cohérence des données entre elles sont précisés. Ensuite, les opérateurs de l'algèbre sont définis et illustrés par de nombreux exemples. Enfin, les extensions de l'algèbre sont résumées et illustrées.

Le chapitre VII est consacré à l'étude du langage standardisé des SGBD relationnels, le fameux langage SQL. Les différents aspects du standard, accepté en 1986 puis étendu en 1989 et 1992, sont tout d'abord présentés et illustrés par de nombreux exemples. La version actuelle du standard acceptée en 1992, connue sous le nom de SQL2, est décrite avec concision mais précision. Il s'agit là du langage aujourd'hui offert, avec quelques variantes, par tous les SGBD industriels.

Le chapitre VIII traite des règles d'intégrité et des bases de données actives. Le langage d'expression des contraintes d'intégrité et des déclencheurs intégré à SQL est étudié. Puis, les différentes méthodes pour contrôler l'intégrité sont présentées. Enfin, les notions de base de données active et les mécanismes d'exécution des déclencheurs sont analysés.

Le chapitre IX expose plus formellement le concept de vue, détaille le langage de définition et présente quelques exemples simples de vues. Sont successivement abordés : les mécanismes d'interrogation de vues, le problème de la mise à jour des vues, l'utilisation des vues concrètes notamment pour les applications décisionnelles et quelques autres extensions possibles du mécanisme de gestion des vues.

Le chapitre X présente d'abord plus précisément les objectifs de l'optimisation de requêtes et introduit les éléments de base. Une large part est consacrée à l'étude des principales méthodes d'optimisation logique puis physique. Les premières restructurent les requêtes alors que les secondes déterminent le meilleur plan d'exécution pour une requête donnée. L'optimisation physique nécessite un modèle de coût pour estimer le coût de chaque plan d'exécution afin de choisir le meilleur. Un tel modèle est décrit, puis les stratégies essentielles permettant de retrouver un plan d'exécution proche de l'optimal sont introduites.

La **troisième partie** développe les approches objet et objet-relationnel. Les problèmes fondamentaux posés par l'objet sont analysés en détail.

Le chapitre XI développe l'approche objet. Les principes de la modélisation de données orientée objet sont tout d'abord esquissés. Puis, les techniques plus spécifiques aux bases de données à objets, permettant d'assurer la persistance et le partage des objets, sont développées. Enfin, ce chapitre propose une extension de l'algèbre relationnelle pour manipuler des objets complexes.

Le chapitre XII présente le standard de l'ODMG, en l'illustrant par des exemples. Sont successivement étudiés : le contexte et l'architecture d'un SGBDO conforme à l'ODMG, le modèle abstrait et le langage ODL, un exemple de base et de schéma en ODL, le

langage OQL à travers des exemples et des syntaxes types de requêtes, l'intégration dans un langage de programmation comme Java et les limites du standard de l'ODMG.

Le chapitre XIII présente le modèle objet-relationnel, et son langage SQL3. Il définit les notions de base dérivées de l'objet et introduites pour étendre le relationnel. Il détaille le support des objets en SQL3 avec de nombreux exemples. Il résume les caractéristiques essentielles du langage de programmation de procédures et fonctions SQL/PSM, appoint essentiel à SQL pour assurer la complétude en tant que langage de programmation. Il souligne les points obscurs du modèle et du langage SQL3.

Le chapitre XIV présente une synthèse des techniques essentielles de l'optimisation des requêtes dans les bases de données objet, au-delà du relationnel. Les nombreuses techniques présentées sont issues de la recherche ; elles commencent aujourd'hui à être intégrées dans les SGBD objet-relationnel et objet. Une bonne compréhension des techniques introduites de parcours de chemins, d'évaluation de coût de requêtes, de placement par groupes, de prise en compte des règles sémantiques, permettra sans nul doute une meilleure optimisation des nouvelles applications.

La **dernière partie** traite trois aspects indépendants importants des bases de données : extensions pour la déduction, gestion de transactions et techniques de conception.

Le chapitre XV décrit les approches aux bases de données déductives. Plus précisément, il est montré comment une interprétation logique des bases de données permet de les étendre vers la déduction. Le langage de règles Datalog est présenté avec ses diverses extensions. Les techniques d'optimisation de règles récursives sont approfondies. L'intégration de règles aux objets est exemplifiée à l'aide de langages concrets implémentés dans des systèmes opérationnels.

Le chapitre XVI essaie de faire le point sur tous les aspects de la gestion de transactions dans les SGBD centralisés. Après quelques rappels de base, nous traitons d'abord les problèmes de concurrence. Nous étudions ensuite les principes de la gestion de transactions. Comme exemple de méthode de reprise intégrée, nous décrivons la méthode ARIES implémentée à IBM, la référence en matière de reprise. Nous terminons la partie sur les transactions proprement dites en présentant les principaux modèles de transactions étendus introduits dans la littérature. Pour terminer ce chapitre, nous traitons du problème un peu orthogonal de confidentialité.

Le chapitre XVII traite le problème de la conception des bases de données objet-relationnel. C'est l'occasion de présenter le langage de modélisation UML, plus précisément les constructions nécessaires à la modélisation de BD. Nous discutons aussi des techniques d'intégration de schémas. Le chapitre développe en outre les règles pour passer d'un schéma conceptuel UML à un schéma relationnel ou objet-relationnel. La théorie de la normalisation est intégrée pour affiner le processus de conception. Les principales techniques d'optimisation du schéma physique sont introduites.

Enfin, le chapitre XVIII couvre les directions nouvelles d'évolution des SGBD : datawarehouse, data mining, Web et multimédia. Ces directions nouvelles, sujets de nombreuses recherches actuellement, font l'objet d'un livre complémentaire du même auteur chez le même éditeur.

4. BIBLIOGRAPHIE

De nombreux ouvrages traitent des problèmes soulevés par les bases de données. Malheureusement, beaucoup sont en anglais. Vous trouverez à la fin de chaque chapitre du présent livre les références et une rapide caractérisation des articles qui nous ont semblé essentiels. Voici quelques références d'autres livres traitant de problèmes généraux des bases de données que nous avons pu consulter. Des livres plus spécialisés sont référencés dans le chapitre traitant du problème correspondant.

[Date90] Date C.J., *An Introduction to Database Systems*, 5e edition, The Systems Programming Series, volumes I (854 pages) et II (383 pages), Addison Wesley, 1990.

Ce livre écrit par un des inventeurs du relationnel est tourné vers l'utilisateur. Le volume I traite des principaux aspects des bases de données relationnelles, sans oublier les systèmes basés sur les modèles réseau et hiérarchique. Ce volume est divisé en six parties avec des appendices traitant de cas de systèmes. La partie I introduit les concepts de base. La partie II présente un système relationnel type, en fait une vue simplifiée de DB2, le SGBD d'IBM. La partie III approfondit le modèle et les langages de manipulation associés. La partie IV traite de l'environnement du SGBD. La partie V est consacrée à la conception des bases de données. La partie VI traite des nouvelles perspectives : répartition, déduction et systèmes à objets. Le volume II traite des problèmes d'intégrité, de concurrence et de sécurité. Il présente aussi les extensions du modèle relationnel proposées par Codd (et Date), ainsi qu'une vue d'ensemble des bases de données réparties et des machines bases de données.

[Delobel91] Delobel C., Lécluse Ch., Richard Ph., *Bases de Données : Des Systèmes Relationnels aux Systèmes à Objets*, 460 pages, InterEditions, Paris, 1991.

Une étude de l'évolution des SGBD, des systèmes relationnels aux systèmes objets, en passant par les systèmes extensibles. Un intérêt particulier est porté sur les langages de programmation de bases de données et le typage des données. Le livre décrit également en détail le système O2, son langage CO2 et les techniques d'implémentation sous-jacentes. Un livre en français.

[Gardarin97] Gardarin G., Gardarin O., *Le Client-Serveur*, 470 pages, Editions Eyrolles, 1997.

Ce livre traite des architectures client-serveur, des middlewares et des bases de données réparties. Les notions importantes du client-serveur sont dégagées et expliquées. Une part importante de l'ouvrage est consacrée aux middlewares et outils de développement objet. Les middlewares à objets distribués CORBA et DCOM sont analysés. Ce livre est un complément souhaitable au présent ouvrage, notamment sur les middlewares, les bases de données réparties et les techniques du client-serveur.

[Gray91] Gray J. Ed., *The Benchmark Handbook*, Morgan & Kaufman Pub., San Mateo, 1991.

Le livre de base sur les mesures de performances des SGBD. Composé de différents articles, il présente les principaux benchmarks de SGBD, en particulier le fameux benchmark TPC qui permet d'échantillonner les performances des SGBD en transactions par seconde. Les conditions exactes du benchmark définies par le « Transaction Processing Council » sont précisées. Les benchmarks de l'université du Madisson, AS3AP et Catell pour les bases de données objets sont aussi présentés.

[Korth97] Silberschatz A., Korth H., Sudarshan S., *Database System Concepts*, 819 pages, Mc Graw-Hill Editions, 3e edition, 1997.

Un livre orienté système et plutôt complet. Partant du modèle entité-association, les auteurs introduisent le modèle relationnel puis les langages des systèmes commercialisés. Ils se concentrent ensuite sur les contraintes et sur les techniques de conception de bases de données. Les deux chapitres qui suivent sont consacrés aux organisations et méthodes d'accès de fichiers. Les techniques des SGBD relationnels (reprises après pannes, contrôle de concurrence, gestion de transaction) sont ensuite exposées. Enfin, les extensions vers les systèmes objets, extensibles et distribués sont étudiées. Le dernier chapitre présente des études de cas de systèmes et deux annexes traitent des modèles réseaux et hiérarchiques. La nouvelle bible des SGBD en anglais.

[Maier83] Maier D., *The Theory of Relational Databases*, Computer Science Press, 1983.

Le livre synthétisant tous les développements théoriques sur les bases de données relationnelles menés au début des années 80. En 600 pages assez formelles, Maier fait le tour de la théorie des opérateurs relationnels, des dépendances fonctionnelles, multivaluées, algébriques et de la théorie de la normalisation.

[Parsaye89] Parsaye K., Chignell M., Khoshafian S., Wong H., *Intelligent Databases*, 478 pages, Wiley Editions, 1989.

Un livre sur les techniques avancées à la limite des SGBD et de l'intelligence artificielle : SGBD objets, systèmes experts, hypermédia, systèmes textuels, bases de données intelligentes. Le SGBD intelligent est à la convergence de toutes ces techniques et intégre règles et objets.

[Ullman88] Ullman J.D., *Principles of Database and Knowledge-base Systems*, volumes I (631 pages) et II (400 pages), Computer Science Press, 1988.

Deux volumes très complets sur les bases de données, avec une approche plutôt fondamentale. Jeffrey Ullman détaille tous les aspects des bases de données, des méthodes d'accès aux modèles objets en passant par le modèle logique. Les livres sont finalement très centrés sur une approche par la logique des bases de données. Les principaux algorithmes d'accès, d'optimisation de requêtes, de concurrence, de normalisation, etc. sont détaillés. A noter l'auteur traite dans un même chapitre les systèmes en réseau et les systèmes objets, qu'il considère de même nature.

[Valduriez99] Valduriez P., Ozsu T., *Principles of Distributed Database Systems*, 562 pages, Prentice Hall, 2e édition,1999.

Le livre fondamental sur les bases de données réparties. Après un rappel sur les SGBD et les réseaux, les auteurs présentent l'architecture type d'un SGBD réparti. Ils abordent ensuite en détail les différents problèmes de conception d'un SGBD réparti : distribution des données, contrôle sémantique des données, évaluation de questions réparties, gestion de transactions réparties, liens avec les systèmes opératoires et multibases. La nouvelle édition aborde aussi le parallélisme et les middlewares. Les nouvelles perspectives sont enfin évoquées.