MOZAIC FORMATIONS

# EXPLOITER UNE BASE DE DONNEES CLIENTS

## Support de cours

MOZAIC FORMATIONS
2 Allée du bois bleu
95450 LONGUESSE

Sarl AU CAPITAL 7 622.45 €
Siren 42909295000048
Code APE 8559A
Mise à jour 20/03/2019 V1 - Sourcelaurent-audibert.developpez.com

# Sommaire

Aujourd'hui, la disponibilité de systèmes de gestion de bases de données fiables permet aux organisations de toutes tailles de gérer des données efficacement, de déployer des applications utilisant ces données et de les stocker. Les bases de données sont actuellement au cœur du système d'information des entreprises.

Ce cours commence par s'intéresser à la problématique de la conception des bases de données. La deuxième partie est consacrée aux bases de données relationnelles, c'est-à-dire aux bases conçues suivant le modèle relationnel et manipulées en utilisant l'algèbre relationnelle. Il s'agit, à ce jour, de la méthode la plus courante pour organiser et accéder à des ensembles de données. La dernière partie constitue, enfin, une bonne introduction au langage SQL (Structured Query Language) qui peut être considéré comme le langage d'accès normalisé aux bases de données relationnelles. Le langage SQL est supporté par la plupart des systèmes de gestion de bases de données commerciaux (comme Oracle) et du domaine libre (comme PostgreSQL).

Il est difficile de donner une définition exacte de la notion de base de données. Une définition très générale pourrait être :

**Définition 1 -Base de données-** Un ensemble organisé d'informations avec un objectif commun.

Peu importe le support utilisé pour rassembler et stocker les données (papier, fichiers, etc.), dès lors que des données sont rassemblées et stockées d'une manière organisée dans un but spécifique, on parle de base de données.

Plus précisément, on appelle base de données un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données). Bien entendu, dans le cadre de ce cours, nous nous intéressons aux bases de données informatisées.

**Définition 2 -Base de données informatisée-** Une base de données informatisée est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, représentant des informations du monde réel et pouvant être interrogées et mises à jour par une communauté d'utilisateurs.

Le résultat de la conception d'une base de données informatisée est une description des données. Par description on entend définir les propriétés d'ensembles d'objets modélisés dans la base de données et non pas d'objets particuliers. Les objets particuliers sont créés par des programmes d'applications ou des langages de manipulation lors des insertions et des mises à jour des données.

Cette description des données est réalisée en utilisant un modèle de données(1). Ce dernier est un outil formel utilisé pour comprendre l'organisation logique des données.

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le *Système de gestion de base de données* (SGBD). Nous y reviendrons dans la section **1.2**. Un SGBD est caractérisé par le modèle de description des données qu'il supporte (hiérarchique, réseau, relationnel, objet : cf. section **1.1.2**). Les données sont décrites sous la forme de ce modèle, grâce à un Langage de Description des Données (LDD). Cette description est appelée *schéma*.

Une fois la base de données spécifiée, on peut y insérer des données, les récupérer, les modifier et les détruire. C'est ce qu'on appelle manipuler les données. Les données peuvent être manipulées non seulement par un Langage spécifique de Manipulation des Données (LMD), mais aussi par des langages de programmation classiques.

Les bases de données ont pris une place importante en informatique, et particulièrement dans le domaine de la gestion. L'étude des bases de données a conduit au développement de concepts, méthodes et algorithmes spécifiques, notamment pour gérer les données en mémoire secondaire (*i.e.* disques durs)(2). En effet, dès l'origine de la discipline, les informaticiens ont observé que la taille de la RAM ne permettait pas de charger l'ensemble d'une base de données en mémoire. Cette hypothèse est toujours vérifiée, car le volume des données ne cesse de s'accroître sous la poussée des nouvelles technologies du WEB.

Ainsi, les bases de données de demain devront être capables de gérer plusieurs dizaines de téra-octets de données, géographiquement distribuées à l'échelle d'Internet, par plusieurs dizaines de milliers d'utilisateurs dans un contexte d'exploitation changeant (on ne sait pas très bien maîtriser ou prédire les débits de communication entre sites) voire sur des nœuds volatiles. En physique des hautes énergies, on prédit qu'une seule expérience produira de l'ordre du pétaoctet de données par an.

Comme il est peu probable de disposer d'une technologie de disque permettant de stocker sur un unique disque cette quantité d'informations, les bases de données se sont orientées vers des architectures distribuées ce qui permet, par exemple, d'exécuter potentiellement plusieurs instructions d'entrée/sortie en même temps sur des disques différents et donc de diviser le temps total d'exécution par un ordre de grandeur.

Une base de données hiérarchique est une forme de système de gestion de base de données qui lie des enregistrements dans une structure arborescente de façon à ce que chaque enregistrement n'ait qu'un seul possesseur (par exemple, une paire de chaussures n'appartient qu'à une seule personne).

Les structures de données hiérarchiques ont été largement utilisées dans les premiers systèmes de gestion de bases de données conçus pour la gestion des données du programme Apollo de la NASA. Cependant, à cause de leurs limitations internes, elles ne peuvent pas souvent être utilisées pour décrire des structures existantes dans le monde réel.

Les liens hiérarchiques entre les différents types de données peuvent rendre très simple la réponse à certaines questions, mais très difficile la réponse à d'autres formes de questions. Si le principe de relation « 1 vers N » n'est pas respecté (par exemple, un malade peut avoir plusieurs médecins et un médecin a, *a priori*, plusieurs patients), alors la hiérarchie se transforme en un réseau.

Le modèle réseau est en mesure de lever de nombreuses difficultés du modèle hiérarchique grâce à la possibilité d'établir des liaisons de type *n-n*, les liens entre objets pouvant exister sans restriction. Pour retrouver une donnée dans une telle modélisation, il faut connaître le chemin d'accès (les liens) ce qui rend les programmes dépendants de la structure de données

Ce modèle de bases de données a été inventé par C.W. Bachman. Pour son modèle, il reçut en 1973 le prix Turing.

Une base de données relationnelle est une base de données structurée suivant les principes de l'algèbre relationnelle.

Le père des bases de données relationnelles est Edgar Frank Codd. Chercheur chez IBM à la fin des années 1960, il étudiait alors de nouvelles méthodes pour gérer de grandes quantités de données, car les modèles et les logiciels de l'époque ne le satisfaisaient pas. Mathématicien de formation, il était persuadé qu'il pourrait utiliser des branches spécifiques des mathématiques (la théorie des ensembles et la logique des prédicats du premier ordre) pour résoudre des difficultés telles que la redondance des données, l'intégrité des données ou l'indépendance de la structure de la base de données avec sa mise en œuvre physique.

En 1970, [8] publia un article où il proposait de stocker des données hétérogènes dans des tables, permettant d'établir des relations entre elles. De nos jours, ce modèle est extrêmement répandu, mais en 1970, cette idée était considérée comme une curiosité intellectuelle. On doutait que les tables puissent être jamais gérées de manière efficace par un ordinateur.

Ce scepticisme n'a cependant pas empêché Codd de poursuivre ses recherches. Un premier prototype de Système de gestion de bases de données relationnelles (SGBDR) a été construit dans les laboratoires d'IBM. Depuis les années 80, cette technologie a mûri et a été adoptée par l'industrie. En 1987, le langage SQL, qui étend l'algèbre relationnelle, a été standardisé.

C'est dans ce type de modèle que se situe ce cours de base de données.

La notion de *bases de données objet* ou *relationnel-objet* est plus récente et encore en phase de recherche et de développement. Elle sera très probablement ajoutée au modèle relationnel.

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le Système de gestion de base de données (SGBD). Un SGBD doit permettre l'ajout, la modification et la recherche de données. Un système de gestion de bases de données héberge généralement plusieurs bases de données, qui sont destinées à des logiciels ou des thématiques différents.

Actuellement, la plupart des SGBD fonctionnent selon un mode client/serveur. Le serveur (sous-entendu la machine qui stocke les données) reçoit des requêtes de plusieurs clients et ceci de manière concurrente. Le serveur analyse la requête, la traite et retourne le résultat au client. Le modèle client/serveur est assez souvent implémenté au moyen de l'interface des sockets (voir le cours de réseau) ; le réseau étant Internet.

Une variante de ce modèle est le modèle ASP (Application Service Provider). Dans ce modèle, le client s'adresse à un mandataire (broker) qui le met en relation avec un SGBD capable de résoudre la requête. La requête est ensuite directement envoyée au SGBD sélectionné qui résout et retourne le résultat directement au client.

Quel que soit le modèle, un des problèmes fondamentaux à prendre en compte est la cohérence des données. Par exemple, dans un environnement où plusieurs utilisateurs peuvent accéder concurremment à une colonne d'une table par exemple pour la lire ou pour l'écrire, il faut s'accorder sur la politique d'écriture. Cette politique peut être : les lectures concurrentes sont autorisées, mais dès qu'il y a une écriture dans une colonne, l'ensemble de la colonne est envoyé aux autres utilisateurs l'ayant lue pour qu'elle soit rafraîchie.

Des objectifs principaux ont été fixés aux SGBD dès l'origine de ceux-ci, et ce, afin de résoudre les problèmes causés par la démarche classique. Ces objectifs sont les suivants :

**Indépendance physique :**

- La façon dont les données sont définies doit être indépendante des structures de stockage utilisées.

**Indépendance logique :**

- Un même ensemble de données peut être vu différemment par des utilisateurs différents. Toutes ces visions personnelles des données doivent être intégrées dans une vision globale.

**Accès aux données :**

- L'accès aux données se fait par l'intermédiaire d'un Langage de Manipulation de Données (LMD). Il est crucial que ce langage permette d'obtenir des réponses aux requêtes en un temps « raisonnable ». Le LMD doit donc être optimisé, minimiser le nombre d'accès disques, et tout cela de façon totalement transparente pour l'utilisateur.

**Administration centralisée des données (intégration) :**

- Toutes les données doivent être centralisées dans un réservoir unique commun à toutes les applications. En effet, des visions différentes des données (entre autres) se résolvent plus facilement si les données sont administrées de façon centralisée.

**Non-redondance des données :**

- Afin d'éviter les problèmes lors des mises à jour, chaque donnée ne doit être présente qu'une seule fois dans la base.

**Cohérence des données :**

- Les données sont soumises à un certain nombre de contraintes d'intégrité qui définissent un état cohérent de la base. Elles doivent pouvoir être exprimées simplement et vérifiées

automatiquement à chaque insertion, modification ou suppression des données. Les contraintes d'intégrité sont décrites dans le Langage de Description de Données (LDD).

### Partage des données :

- Il s'agit de permettre à plusieurs utilisateurs d'accéder aux mêmes données au même moment de manière transparente. Si ce problème est simple à résoudre quand il s'agit uniquement d'interrogations, cela ne l'est plus quand il s'agit de modifications dans un contexte multiutilisateur, car il faut : permettre à deux (ou plus) utilisateurs de modifier la même donnée « en même temps » et assurer un résultat d'interrogation cohérent pour un utilisateur consultant une table pendant qu'un autre la modifie.

### Sécurité des données :

- Les données doivent pouvoir être protégées contre les accès non autorisés. Pour cela, il faut pouvoir associer à chaque utilisateur des droits d'accès aux données.

### Résistance aux pannes :

- Que se passe-t-il si une panne survient au milieu d'une modification, si certains fichiers contenant les données deviennent illisibles ? Il faut pouvoir récupérer une base dans un état « sain ». Ainsi, après une panne intervenant au milieu d'une modification deux solutions sont possibles : soit récupérer les données dans l'état dans lequel elles étaient avant la modification, soit terminer l'opération interrompue.

Pour atteindre certains de ces objectifs (surtout les deux premiers), trois niveaux de description des données ont été définis par la norme ANSI/SPARC.

### Le niveau externe

- correspond à la perception de tout ou partie de la base par un groupe donné d'utilisateurs, indépendamment des autres. On appelle cette description le *schéma externe* ou *vue*. Il peut exister plusieurs schémas externes représentant différentes vues sur la base de données avec des possibilités de recouvrement. Le niveau externe assure l'analyse et l'interprétation des requêtes en primitives de plus bas niveau et se charge également de convertir éventuellement les données brutes, issues de la réponse à la requête, dans un format souhaité par l'utilisateur.

### Le niveau conceptuel

- décrit la structure de toutes les données de la base, leurs propriétés (*i.e.* les relations qui existent entre elles : leur sémantique inhérente), sans se soucier de l'implémentation physique ni de la façon dont chaque groupe de travail voudra s'en servir. Dans le cas des SGBD relationnels, il s'agit d'une vision tabulaire où la sémantique de l'information est exprimée en utilisant les concepts de relation, attributs et de contraintes d'intégrité. On appelle cette description le *schéma conceptuel*.

### Le niveau interne ou physique

- s'appuie sur un système de gestion de fichiers pour définir la politique de stockage ainsi que le placement des données. Le niveau physique est donc responsable du choix de l'organisation

physique des fichiers ainsi que de l'utilisation de telle ou telle méthode d'accès en fonction de la requête. On appelle cette description le *schéma interne*.

Il existe de nombreux systèmes de gestion de bases de données, en voici une liste non exhaustive :

**PostgreSQL:**

- **http://www.postgresql.org/** - dans le domaine public ;

**MySQL :**

- **http://www.mysql.org/** - dans le domaine public ;

**Oracle :**

- **http://www.oracle.com/** - de Oracle Corporation ;

**IBM DB2 :**

- **http://www-306.ibm.com/software/data/db2/**

**Microsoft SQL :**

- **http://www.microsoft.com/sql/**

**Sybase :**

- **http://www.sybase.com/linux**

**Informix :**

- **http://www-306.ibm.com/software/data/informix/**

(1)
cf. section **1.1.2** pour une présentation générale de plusieurs modèles de données. Le modèle entités-associations est présenté dans la section **2** et le modèle relationnel dans la section **3.1**

(2)
Il faut savoir que les temps d'accès à des disques durs sont d'un ordre de grandeur supérieur (disons 1000 fois supérieur) aux temps d'accès à la mémoire RAM. Tout gestionnaire de base de données doit donc traiter de manière particulière les accès aux disques.