

INFORMATIQUE DE GESTION INFORMATIQUE DE GESTION

PETIT COURS DE MERISE

Tutoriaux qui vous permettront d'apprendre merise facilement sous divers angles !!!

Source : http://www.supinfo-projects.com/fr/2004/concept_bdd/1/

Document réalisé par Mr Souleymane, Elève ingénieur en informatique de gestion

00225- 08281648

sangsoulinter@yahoo.fr - sangsoulci@hotmail.com

Introduction

Objectifs

A l'heure où le e-commerce se développe, de plus en plus d'entreprises utilisent des bases de données pour gérer leur activité. La base de données est en effet un moyen très efficace pour stocker physiquement une masse d'informations et gérer cet ensemble d'une façon très rapide. L'entreprise gagne ainsi un temps considérable car elle n'a pas à faire intervenir d'opérateurs pour répondre aux attentes de ses différents clients.

Concevoir une base de données ne pose pas de problème en soit, mais il faut utiliser une certaine méthodologie qu'il est impératif de respecter pour parvenir à ses fins. Cette synthèse aura donc pour objectif de vous présenter les différentes phases de conception d'une base de données, afin d'arriver à un système qui sera fiable pendant de nombreuses années ...

1. Définitions

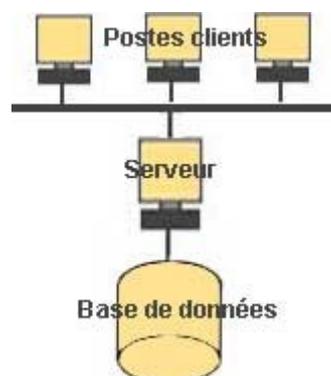
1.1. Base de données

Chaque individu effectue sans le savoir parfois, un stockage d'informations dans un domaine bien précis. Par exemple : un particulier peut écrire dans un classeur ses différentes recettes de cuisine, effectuer un carnet d'adresse, faire une liste de sa vidéothèque, des dépenses prévues ... Un bibliothécaire maintient une liste de ses membres, de ses livres ...

Ces informations sont appelées des données. Elles peuvent être stockées sur différents types de support et format comme par exemple : des fichiers papiers, des fichiers informatiques ou alors une base de données.

Une base de données est donc un ensemble d'informations structurées portant sur un thème commun. Par exemple, une entreprise pourra se constituer une base de données qui regroupera les données de ses différents produits, clients, ou fournisseurs ...

Pour administrer efficacement une base de données, il est impératif de posséder un Système de Gestion de Base de Données (SGBD), qui va permettre de gérer cette base.



Exemple d'une configuration Base de données

1.2. Système de Gestions de Base de Données

Pour pouvoir gérer les multiples données d'une base, permettre la gestion de ces dernières par de nombreux utilisateurs, il est nécessaire de posséder un système de gestion. Cette gestion s'effectue grâce à un Système de Gestions de Base de Données (SGBD).

C'est le SGBD choisit qui va permettre de gérer :

- L'accès aux données
- Autoriser et superviser l'accès aux informations par de multiples utilisateurs
- Manipuler les données (ajout, insertion, suppression, modification ...)
- Sécuriser les données

Il est donc nécessaire de bien choisir son système de Gestion, car tout le bon fonctionnement de la base de données reposera sur lui ...

2. Analyse de la conception

Afin de réaliser un projet informatique, il faut pratiquer une analyse informatique. Cette analyse consiste à comprendre et modéliser le système d'information (SI) sur lequel nous travaillons. Un système d'information regroupe toutes les informations d'un domaine précis.

La méthode d'analyse MERISE est une méthode de conception et de développement de système d'information. C'est cette méthode qui va nous permettre de réaliser le MCD (modèle conceptuel des données : diagramme entité-relation permettant de modéliser le système d'information sans prendre en compte les détails liées à sa mise en oeuvre physique), et le MLD (modèle logique des données : qui est issue du MCD et qui va permettre de prendre en compte la structuration technique propre au stockage informatisé). Si le MCD et le MLD sont très bien réalisés, la mise en oeuvre physique de la base de données sera très facile à implémenter.

L'efficacité et la validité de cette analyse reposent sur la qualité de la communication entre les utilisateurs (maîtrise d'Ouvrage) et les informaticiens (maîtrise d'Oeuvre).

2.1. Modèle Conceptuel des Données

Pour vous expliquer comment concevoir un MCD, prenons l'un des SI que j'ai eu l'occasion de développer pendant mon stage:

Un client, identifié par son numéro de client, son nom, son prénom, son adresse, son téléphone, passe une commande identifiée par son numéro de commande et sa date. Cette commande est constituée d'articles identifiés par un numéro d'articles, une désignation d'articles, et un prix unitaire hors taxe. Cette article pourra être archivée dans la base de données selon s'il est encore vendu ou non. Cette commande sera par la suite facturée au client par l'intermédiaire d'une facture (n° facture, nom de facture, adresse de facture qui peut différer de celle du client, montant de la facture et mode de règlement de cette facture).

Comment les données circulent elles ?

- Le client passe une commande
- La commande est facturée au client

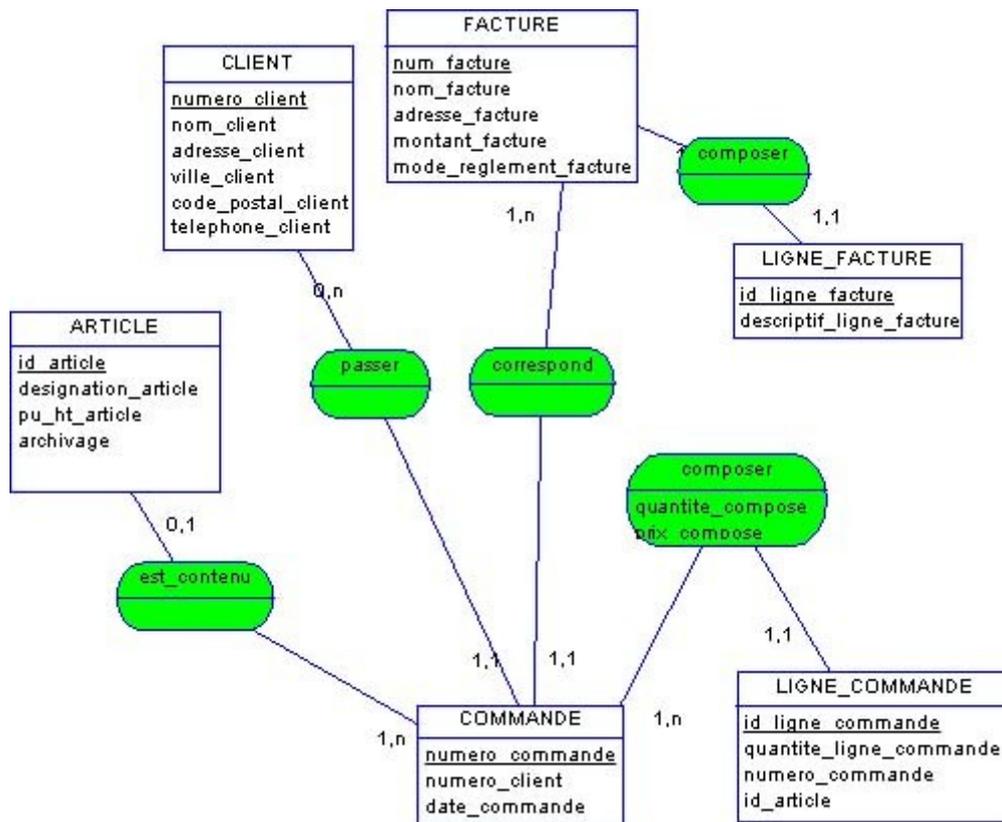
Quelles sont les données ?

- Le client : n° client, nom, prénom, adresse, téléphone
- La commande : n° commande, date_commande
- La facture : n° facture, nom de facture, adresse de facture, montant de la facture, mode de règlement de la facture

Comment sont elles liées entre elles ?

- Une commande n'appartient qu'à un seul client
- Un client peut passer plusieurs commandes
- Une commande est constituée de produits
- Une facture ne se rapporte qu'à une seule commande ...

Une fois que la maîtrise d'ouvrage a donnée l'ensemble de ses informations, que celles ci ont été transcrites, le passage au MCD se fait donc de manière naturelle :



Exemple d'un Modèle conceptuel de données

Ce modèle est appelé : **Modèle Entité - Relation**. Les entités sont représentées par les rectangles, et les relations par des ovales. On appelle attributs les différentes composantes d'une entité. Par exemple : numero_commande, numero_client, date_commande sont les attributs de l'entité Commande.

2.2. Modèle Logique de Données

Dès lors que le modèle conceptuel de données est effectué, il est temps de réaliser le modèle logique de données. Pour se faire, il suffit de déterminer quel sera le type des différents attributs contenu dans les entités. Il faudra aussi déterminer les différentes contraintes d'intégrité qu'il pourrait y avoir sur ses attributs.

En reprenant l'exemple précédent, nous aurions un MLD de cette forme :

ENTITE	ATTRIBUT	TYPE	CONTRAINTES(S)
COMMANDE	numero_commande	Entier	Clé primaire / Non Null
	numero_client	Entier	Non Null
	date_commande	Date	

ENTITE	ATTRIBUT	TYPE	CONTRAINTES(S)
CLIENT	numero_client	Entier	Clé primaire / Non Null
	nom_client	Chaîne de 20 Caractères	
	adresse_client	Chaîne de 100 Caractères	
	ville_client	Chaîne de 20 Caractères	
	code_postal_client	Entier	
	telephone_client	Chaîne de 50 Caractères	

Grâce à la conception de ces 2 modèles, la mise en oeuvre au niveau physique devient très facile.

3. Choix des différents Système de Gestion

De nombreux SGBD sont actuellement disponible sur le marché. Pourquoi choisir tel ou tel SGBD ? Cette page a pour but de vous présenter les avantages et inconvénients des différents SGBD que l'on connaît : Interbase, MySQL, Oracle, SQLServer et Access. Ainsi, en fonction de ce que vous souhaitez réaliser, de votre budget, de vos compétences informatiques, de l'évolution de votre projet ... vous serez à même de choisir le SGBD qui vous convient le mieux.

Il existe bien entendu d'autres système de gestions, mais cette synthèse étant destinée à un large public, seul les principaux SGBD sont présentés car le support qui est offert en cas de problème ne peut pas poser de problèmes ...

3.1. Interbase



InterBase possède beaucoup de qualités : ce SGBD offre une administration facile qui est de plus auto optimisée et auto administrée. De nombreux outils sont disponibles pour gérer InterBase. Ce SGBD est fiable et robuste et supporte de nombreuses instructions avancées. Enfin, l'un de ses principaux points forts est sa disponibilité dans une version OpenSource. Il existe des versions payantes de ce produit, qui propose des améliorations, mais la version OpenSource offre déjà beaucoup de possibilités. Dans la mesure où Interbase est un produit Borland, son interaction avec C++Builder ou Delphi ne pose aucun problèmes.

InterBase est vraisemblablement le meilleur choix pour gérer des bases de données, de petite taille comme de moyenne taille.

Site de l'éditeur : <http://www.borland.com/interbase/>

3.2. MySql



MySQL est une solution très courante en ce qui concerne l'hébergement public de site Internet. MySQL offre en effet une très bonne intégration dans l'environnement PHP/Apache et une bonne rapidité. Il est très stable et ceux même avec un grand nombre d'enregistrements. Un des gros avantages est donc une grande communauté d'utilisateurs, ce qui en facilite le support en cas de problèmes. De plus, MySQL est gratuit, sauf pour certaines utilisations.

En revanche, MySQL ne supporte qu'une partie du standard SQL-92. Par exemple, MySQL ne permet pas d'effectuer des sous requêtes, des procédures stockées et des transactions natives.

MySQL sera donc un très bon choix si vous souhaitez l'utiliser en relation avec PHP, en utilisant des requêtes simples. Sachez néanmoins que de nombreuses fonctions ne pourront être utilisées avec ce SGBD.

Site de l'éditeur : <http://www.mysql.com/>

3.3. Oracle



Oracle est incontestablement l'un des systèmes de gestion figurant parmi les plus performants du marché. Il est multi plateformes : les premières versions d'Oracle tournaient sur des systèmes Unix, et depuis, l'éditeur a développé des versions pour des plateformes de type BSD / Windows NT-2000-XP, et plus récemment Linux.

Dans ses versions actuelles, Oracle est un système réservé aux grandes entités car il nécessite de fortes contraintes de disponibilités. Oracle n'est pas fait pour optimiser des petites bases de données : sur de petits volumes de traitements, avec peu d'utilisateurs, MySQL offre des performances quasi comparables à Oracle. Par contre, dès que le volume de données devient très important, qu'il y a un grand nombre d'utilisateurs, les écarts de performances sont très visibles et tournent à l'avantage d'Oracle

Les principaux atouts d'Oracle sont :

- Une très grande stabilité de noyau, ce qui est indispensable pour des entreprises qui ne peuvent pas se permettre d'avoir une interruption de service.
- Il permet de paramétrer tout ce que vous désirez pour votre entreprise.
- Les procédures stockées en PL/SQL (langage propriétaire Oracle), ou en Java, permettent de rendre la tâche beaucoup plus facile aux équipes de développement.
- La pérennité de l'éditeur : il est important pour une entreprise de savoir que la solution qu'elle adopte ne va pas disparaître du jour au lendemain. De plus, Oracle offre une qualité de support technique irréprochable (24h/24 et 7jrs/7). En plus de cela, il existe sur Internet une grande communauté de DBA/Développeurs.

Pour les points négatifs d'Oracle :

- Le prix est bien évidemment un obstacle car il cible une clientèle bien définie.
- Il est très gourmand en ressources, ce qui, outre le coût des licences serveur, implique un investissement matériel non négligeable. Mais c'est cette "gourmandise" qui permet un paramétrage du système aussi puissant.
- Il nécessite la présence d'un ou plusieurs DBA pour maintenir le système et le faire évoluer. Oracle manque cruellement d'outil qui permette son auto administration (tel qu'on pourrait les trouver dans SQL Serveur, Interbase et autres ...).
- Le langage SQL utilisé est très loin des normes SQL2 et SQL3

Pour résumer : Oracle est vraiment un poids lourd des SGBD. Moyennant un investissement important et la nécessité d'un administrateur, Oracle se révélera un excellent SGBD pour gérer de très grosses bases de données.

Site de l'éditeur : <http://www.oracle.com/>

3.4. SQLServer



SQL Server est un SGBD très rapide qui a l'avantage de proposer un langage procédural facilement exploitable grâce à la simplicité du langage Transact SQL. Le Langage SQL de ce SGBD est assez complet et respecte en général les différentes normes. SqlServer offre la possibilité de faire des sous requêtes dans la clause FROM, ce qui permet de réaliser assez facilement des requêtes assez complexes, directement exploitable.

Sans être simple, l'administration de SQLServer n'atteint pas des sommets de complexité. Il a en effet l'avantage d'être très visuelle, comme la majorité des programmes Microsoft.

On peut cependant y voir les inconvénients suivants : il est mono plateforme, de sorte qu'il pourra n'être utilisé que sur un système Windows, il n'offre pas la possibilité d'utiliser le trigger BEFORE. Enfin, il imbrique des composants Microsoft, ce qui augmente l'insécurité, même si des correctifs sont régulièrement disponibles.

SQL Server est donc un bon choix pour des bases de données de moyenne taille.

Site de l'éditeur : <http://www.microsoft.com/sql/default.asp>

3.5. Access



Access est un outil professionnel qui est adapté à un certain type de besoins. Il est assez performant en tant que SGBD et il intègre un outil de développement, ce qui en facilite l'utilisation. Une personne n'ayant pas forcément beaucoup de connaissances en informatique peut très facilement utiliser Access et se créer une base de données complète. Sa parfaite intégration dans Microsoft Office permet l'utilisation des nombreux assistants en cas de problèmes

D'un autre côté, Access n'est pas un SGBD client/serveur. Il s'agit simplement d'un SGBD fichier. Par conséquent, le trafic sur le réseau peut être perturbé. De surcroît, les performances chutent très rapidement lorsque plusieurs utilisateurs sont connectés en même temps, ou quand la base dépasse les 100000 lignes.

L'interface est plutôt lourde, ce qui fait que les temps de réponses peuvent augmenter.

Access est donc un excellent choix si vous désirez avoir une petite base de données qui pourra très facilement être gérée.

Site de l'éditeur : <http://www.microsoft.com/france/office/access/prodinfo/default.asp>

Je vous invite à visiter ce site : http://sqlpro.developpez.com/SQL_AZ_F.htm, qui vous permettra d'avoir une liste des fonctions disponibles en SQL, et vérifier leur compatibilité avec les différents SGBD.

4. Création de la Base de Données

4.1. Développement interne de la base

Une fois que vous aurez choisi le système de gestion le mieux adapté à votre entreprise, que vous aurez finalisé la conception de votre base de données sous forme de schémas logiques que nous avons étudié précédemment (Modèle conceptuel et modèle logique), il sera alors grand temps de développer la base de données ...

Pour se faire, il vous suffira de transcrire en langage informatique vos différents modèles. Si ceux-ci sont bien réalisés, cela ne devrait pas poser de problèmes particuliers ...

En général, le langage utilisé pour les bases de données est le SQL (Structural Query Language). De nombreux sites pourront vous aider à apprendre ce langage (ex : <http://www.sqlfacile.com>). Je vous invite également à consulter la documentation du SGBD que vous choisirez en cas de problème. En effet, les standards SQL ne sont pas forcément respectés entre les différents SGBD.

Reprenons l'exemple de la partie 2 :

Nous avons réalisé le modèle logique de données des différentes entités. En ce qui concerne les clients, nous avons :

ENTITE	ATTRIBUT	TYPE	CONTRAINTES(S)
CLIENT	numero_client	Entier	Clé primaire / Non Null
	nom_client	Chaîne de 20 Caractères	
	adresse_client	Chaîne de 100 Caractères	
	ville_client	Chaîne de 20 Caractères	
	code_postal_client	Entier	
	telephone_client	Chaîne de 50 Caractères	

Afin de créer la table Client dans la base de données : le code SQL sera le suivant :

```
CREATE TABLE CLIENT
```

```
(numero_client INT NOT NULL)
```

```
PRIMARY KEY
```

```
,nom_client VARCHAR(20)
```

```
,adresse_client VARCHAR(100)
```

```
,ville_client VARCHAR(20)
```

```
,code_postal_client INT
```

```
,telephone_client VARCHAR(50));
```

Il est à noter que ce code est valable pour une base de données sous **Interbase**. On retrouve bien évidemment le nom des différents attributs, leur type, leur taille, ainsi que les différentes contraintes d'intégrité.

Si l'analyse est donc bien réalisée, la création des différentes tables de la base de données ne devrait pas poser de problème.

Il ne vous restera plus qu'à réaliser vos fonctions/procédures stockées, éventuellement vos triggers et tout ce dont vous pourriez avoir besoin pour exploiter au maximum votre base de données.

Cette méthodologie s'appuie donc sur une phase d'analyse assez poussée entre la maîtrise d'ouvrage et la maîtrise d'oeuvre. A l'issue de cette analyse, un cahier des charges pourra être réalisé. Il est vivement conseillé de l'effectuer car il permettra de baliser votre travail. Ainsi en le faisant lire à vos responsables ou à des personnes tierces, elles pourront facilement rendre compte des problèmes de votre solution, la compléter, ou l'affirmer; et cela vous permettra de concevoir votre base d'un seul jet, sans être obligé de recommencer L'entreprise ne peut pas se permettre de perdre de temps ...

4.2. Interaction avec la base

La base de données étant maintenant prête. Il est temps que vos utilisateurs puissent l'exploiter. (Faire des sélections dans les tables, des insertions, des suppressions, des modifications ...).

Vous avez à votre disposition plusieurs solutions pour faire interagir la base de données. La plus connue étant le PHP. Ainsi, via une adresse internet, vos utilisateurs peuvent se connecter à la base de données et réaliser leur tâche.

Java, Microsoft .NET, C++Builder ... vous permettront également de faire interagir votre base de données.

Ici encore, il faudra réaliser une petite étude car tous les systèmes de gestion ne sont pas compatibles avec toutes les solutions d'exploitation. Ce choix devra également être fait en fonction des compétences informatiques de vos utilisateurs, de votre matériel ...

5. Sécurisation de la Base de données

Afin d'éviter des catastrophes, il est primordial de sécuriser la base de données ainsi que le réseau sur lequel elle se trouve.

Il faudra donc :

- Mettre en place des comptes utilisateurs avec différents niveaux d'accès au niveau de la base. Ainsi chaque utilisateur pourra accéder à telles ou telles parties de la base de données, à l'aide d'un login et d'un mot de passe.
- L'accès aux tables système ne devra être possible que par un administrateur. Ainsi les utilisateurs ne pourront pas mettre en demeure le bon fonctionnement de la base de données. Un opérateur pourra éventuellement veiller au contrôle des données au sein de la base.
- Développer une stratégie de Sauvegarde de la Base de données. Ainsi, en cas de problème sur cette dernière, la restitution des données pourra se faire aussi rapidement que possible, et l'entreprise n'aura rien perdu en productivité.
- Sécuriser le réseau local de l'entreprise : les données des entreprises étant confidentielles, il est nécessaire que des personnes malveillantes n'y accèdent pas. Ainsi, il est nécessaire de mettre en place un Firewall et un Antivirus. Si l'entreprise a des commerciaux qui doivent se connecter à la base de données de l'extérieur (via internet par exemple), il faudra mettre en place une sécurité accrue pour éviter les failles de sécurité. (citons par exemple le ver Slammer de Janvier 2003 qui a attaqué les serveur SQL Server car ils avaient autorisé les connexions via Internet).
- Mettre en place des sécurités au niveau des utilisateurs du réseau local. La standardiste n'a pas forcément besoin d'accéder aux données confidentielles de la base de données ... Ceci se fait via la configuration de l'Active Directory et de la configuration des différentes stratégies des comptes utilisateurs ...

Si vous désirez de plus amples informations sur la sécurisation d'un réseau d'entreprise : je vous invite à visiter ce site : <http://www.devparadise.com/technoweb/hardware/net/D59.asp>

Conclusion

Les bases de données sont de plus en plus utiles de nos jours pour exploiter une quantité importante d'informations avec un maximum d'efficacité. Pour les entreprises, il y a en effet des contraintes en moins, car la base de données va permettre de centraliser des informations importantes tout en faisant intervenir un nombre réduit d'opérateurs.

Afin de réaliser une base de données qui sera fonctionnelle, il est primordial de respecter une méthodologie bien précise :

- Etude du projet dans les moindres détails : analyse du projet avec ses différentes contraintes ...
- Schéma de la conception sous forme de modèle conceptuel de données.
- Choix du système de gestion en fonction des différentes contraintes de l'entreprise, de l'importance de la base de données et des limites des différents système de gestion.
- Développement proprement dit de la base de données.

En respectant ce schéma, il sera alors possible de concevoir une base de données fiable, qui respectera pleinement le cahier des charges initiale. L'entreprise sera ainsi soulagé d'un poids important et pourra ainsi pleinement gérer ses différentes activités ...

Nous espérons que ce cours vous a permis de mieux comprendre Merise.

Si vous avez des commentaires ou des suggestions à faire, veuillez nous contacter :

Sangaré Souleymane

00225-08281648

sangsoulinter@yahoo.fr - sangsoulci@hotmail.com