Conception et Exploitation d'une base de données

Sous la direction de Christophe MONTAGNE

GEII
Année universitaire
2014/2015

[redacted]

Table des matières

Table des figures

Introduction

Aujourd'hui, les différentes institutions investissent énormément dans les Systèmes d'Informations. Le besoin de collecter, regrouper, classifier, traiter et diffuser un ensemble d'information permet d'être organiser dans la recherche et le traitement d'information et ainsi d'atteindre ses objectifs. Afin de pouvoir manipuler efficacement ces informations, on utilise généralement des Bases de Données.

Pour cela, on répondra à cette problématique :
Comment peut-on créer une Base de Données efficace, robuste, fiable et sécurisé ?

Ce rapport traitera de la notion de base de données ainsi que les concepts autour de : la création, la gestion et l'exploitation grâce aux différentes recherches bibliographique effectuées par les membres du projet.

Architecture des Bases de Données

Afin de construire et exploiter efficacement une Base de donnée, il faut définir par qui elle sera accessible, où et comment sera-t-elle accessible.
L'architecture opérationnelle d'une Base de Données est le nom donné au modèle physique réel de celle-ci. On la définit par les Systèmes Informatiques et les logiciels utilisés et l'emplacement du matériel par rapport à l'application qui interroge la Base de Données.

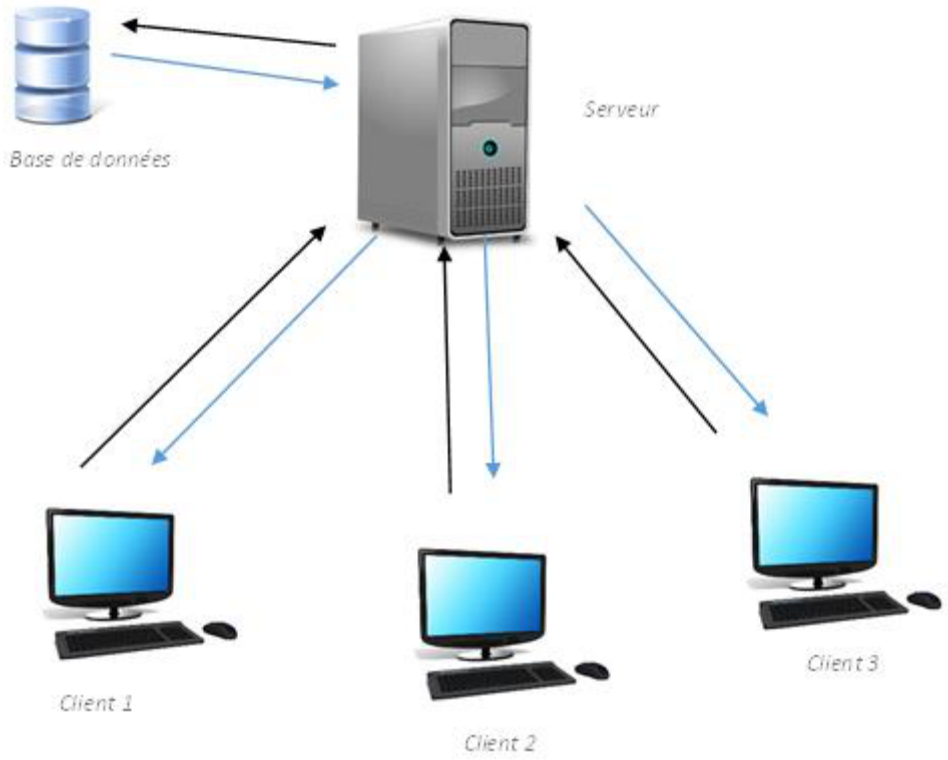


Figure 1: Architecture client/serveur

L'architecture la plus classique est celle dite du client / serveur :
On considère qu'une application informatique comporte trois types de traitement :

- La gestion des données (recherche, mise à jour).
- La logique applicative qui permet de calculer et sélectionner les résultats.
- La présentation des résultats. (WATTIAU 2003) (NAVATHE 2011)

<u>Rôle du serveur et du client</u>

Le serveur est un poste sur lequel la Base de Données est installée ou reliée. Il permet de la gérer directement et de transmettre les informations demandées au client. Le serveur a généralement une interface d'utilisation très complexe et réservé aux professionnels.
Il applique aussi une logique applicative : ce sont les filtres issus du client par rapport à l'ensemble des données. (WATTIAU 2003)
Le poste client permet d'avoir un affichage réduit et informationnel : le client voit ce qu'il demande.

La faiblesse de ce type d'architecture réside dans le nombre de connexion. En effet, si les données sont trop volumineuses et le nombre de client important, il y a un risque de dysfonctionnement du support de communication; mais encore, la logique applicative peut mettre énormément de temps à s'effectuer car il y a trop d'utilisateurs à servir simultanément.
Pour pallier aux inconvénients énoncés ci-dessus, les ingénieurs ont mis en place l'architecture trois-tiers.

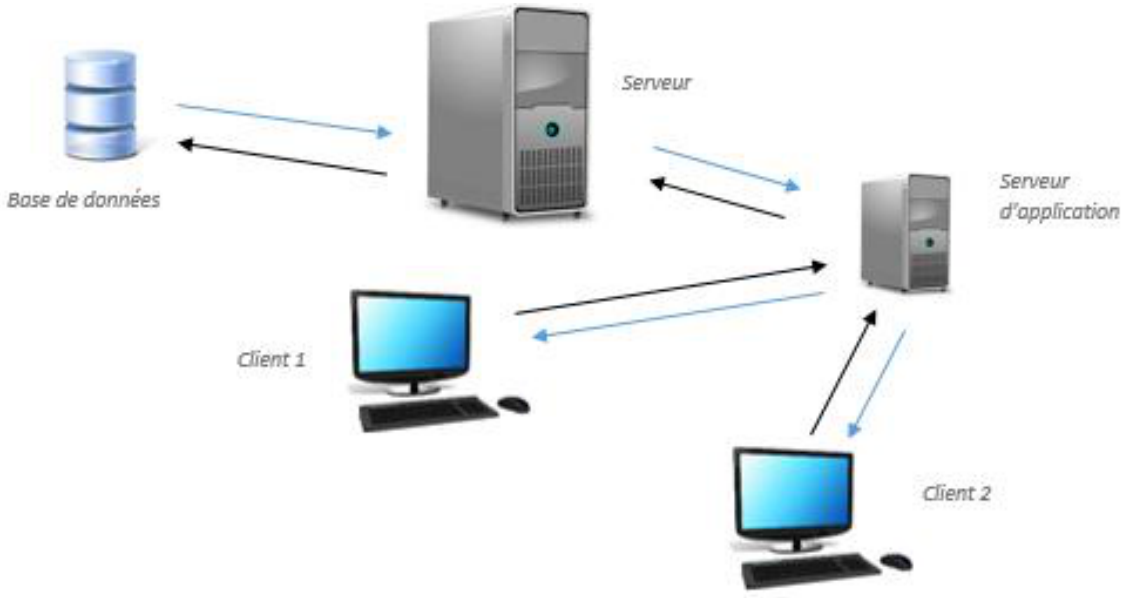


Figure 2 : Architecture trois-tiers.

Par rapport à l'architecture client/serveur, une nouvelle machine fait son apparition : le serveur d'application. Le but de ce serveur est d'appliquer la logique applicative demandée par le client. Ainsi, elle permet au serveur de se concentrer essentiellement sur son activité : traiter les informations sur la Base de Données et répondre aux requêtes issues du serveur d'application.
Toutefois, le serveur d'application peut très bien être décomposé en plusieurs serveurs reliés entre eux : on parle dans ce cas de serveurs n-tiers. Ce genre d'architecture est utilisée par les grandes Bases de Données (WATTIAU 2011) (exemple : les Data Centers de la Société Générale).

Ces différentes architectures ont une utilisation bien précise : selon l'importance et le niveau d'utilisation, il faudra sélectionner dans les plus grands des cas une architecture n-tiers. La plus minimale étant une architecture client/serveur (HAINAUT 2012).

Après avoir défini l'architecture d'une Base de Données, nous pouvons réfléchir à la conception de cette dernière.

Conception d'une Base de Données

Un Système d'Information est la représentation opérationnelle de la réalité afin d'assurer la gestion, le traitement, le transport et la diffusion des informations mais aussi le stockage des données. L'utilisation de ces dernières rencontre plusieurs problèmes tels que le manque de sécurité ou bien encore la lourdeur d'accès aux données. Afin d'y remédier, il est nécessaire de

recourir à une Base de Données prenant en charge les fonctionnalités de protection, de sécurité et de fournir l'interface nécessaire à l'accès aux données.
Dans cette partie, nous verrons dans un premier temps le processus de conception d'une Base de Données et puis dans un second temps les principales méthodes d'analyse.

Processus de conception d'une Base de Données (LARROUSSE, 2006)

Tout d'abord, il faut effectuer l'analyse du système du monde réel afin de pouvoir le modéliser. En effet, la Base de Données concerne des utilisateurs cibles, c'est-à-dire ceux qui utiliseront les données de la base. Il est donc nécessaire de dialoguer avec notre client qui effectue le lien avec ces utilisateurs, qui connaissent les besoins réels, liés à la réalité actuelle et à la réalité souhaitée mais aussi qui expliquent la sémantique des données.
C'est pourquoi, l'étude de l'existant et des besoins nous facilitera cette modélisation. Cette analyse est essentielle afin d'éviter tout problème de compréhension tout au long de la conception.
Lors de cette phase d'analyse, on détermine les objectifs du Système d'Information à concevoir et on identifie tous les éléments à prendre en compte dans le système; ce sont les champs qui contiendront les données. Un ensemble de champs peut constituer un objet du monde réel. Enfin, il faut identifier les liens à modéliser entre ces objets ainsi que les éléments caractéristiques de ces liens.
Toute cette modélisation du réel aboutira à un cahier des charges représenté en un schéma conceptuel qui servira à la description générale du Système d'Information. Ce schéma permet de distinguer les entités qui constituent la Base de Données, et les associations entre ces entités. Il est souvent réalisé à l'aide du modèle Entité-Association.
Ensuite, ce modèle précédent doit être transformé en modèle relationnel pour le rendre acceptable par le Système de Gestion de Bases de Données pour stocker, manipuler et retrouver les données. Ce modèle est utilisé pour décrire les méthodes d'organisation et d'accès aux données de la base.

En résumé, après l'analyse du monde réel, le processus de conception d'une Base de Données se réalise en trois principales étapes qui correspondent à trois niveaux différents comme on peut le voir sur cette illustration :

- Niveau conceptuel (modèle entité-association).
- Niveau logique (modèle relationnel).
- Niveau physique (implémentation sur le SGBD).

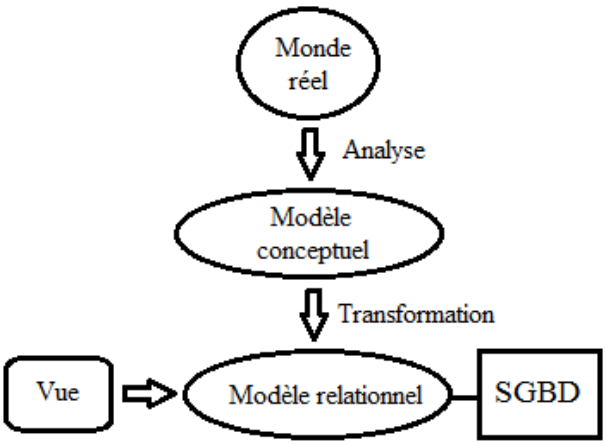


Figure 3: Processus de conception d'une base de données.

Principales méthodes d'analyse

Il existe différentes méthodes pour la conception d'une base de données. L'objectif est de guider le travail d'analyse et d'aider à la réalisation d'un modèle de données le plus juste possible. Parmi celles-ci, il y a la méthode MERISE ou encore UML que nous détaillerons dans les prochaines parties.
En premier lieu, nous expliquerons la méthode MERISE et en second lieu le langage de modélisation UML.

MERISE (Cours 1 / 2, SADI, 2014) (http://www.lsis.org/kleinerm/files/AM/SGBD/CM1.pdf)

Cette méthode française, développée initialement par Hubert Tardieu, est née à la demande du Ministère de l'Industrie qui souhaitait une méthode de conception des Systèmes d'Information dans les années 80.

Cette démarche se décompose en trois niveaux de modélisation.
Le premier qui est le niveau conceptuel. Il faut répondre aux questions telles que :
Quoi ? Que fait l'entreprise ? Avec quelles données ?
Le second qui est le niveau logique ou organisationnel. Il faut répondre à ces questions :
Qui fait quoi ? Quand ? Où ?
Et enfin, le niveau physique ou opérationnel qui permet de répondre aux questions :
Comment ? Quels sont les moyens de le faire ?

1.1) Niveau conceptuel

(http://tecfaetu.unige.ch/staf/staf-h/tassini/staf2x/Heidi/last_bd.htm)

Au niveau conceptuel, il faut effectuer une représentation graphique qui correspond au Modèle Conceptuel des Données. Ce dernier est basé sur le Modèle Entité-Association :

Définitions :

Entité : Définit un objet identifiable ou un individu.

Attribut : Représente la caractéristique d'une entité ou d'une association.

Association : Représente le lien entre les entités par l'utilisation d'un verbe à l'infinitif.

Identifiant : C'est un ou plusieurs attributs de l'entité permettant de déterminer de manière unique toutes les autres entités. Il est inscrit et souligné en tête de l'entité correspondante.
Chaque entité possède au moins un identifiant. Ce dernier est généralement une clé primaire dans le cas d'une relation unique et une clé étrangère dans une autre relation.

Cardinalités : Obligatoires pour le modèle afin de préciser le lien entre l'entité et l'association.
Il en existe différents types :

- Aucune à un : 0,1.
- Un à un : 1,1.

- Aucune à plusieurs : 0, N.
- Un à plusieurs : 1, N.

Représentation graphique :

Les entités sont représentées par des rectangles contenant leurs attributs.
Les associations sont représentées par des cercles.
Les liaisons sont représentées par des traits.
Les cardinalités sont représentées dans des parenthèses sur les liaisons entre l'entité et l'association.

Schéma final explicatif :

Toutes ces règles vues précédemment sont représentées dans ce schéma explicatif ci-dessous :

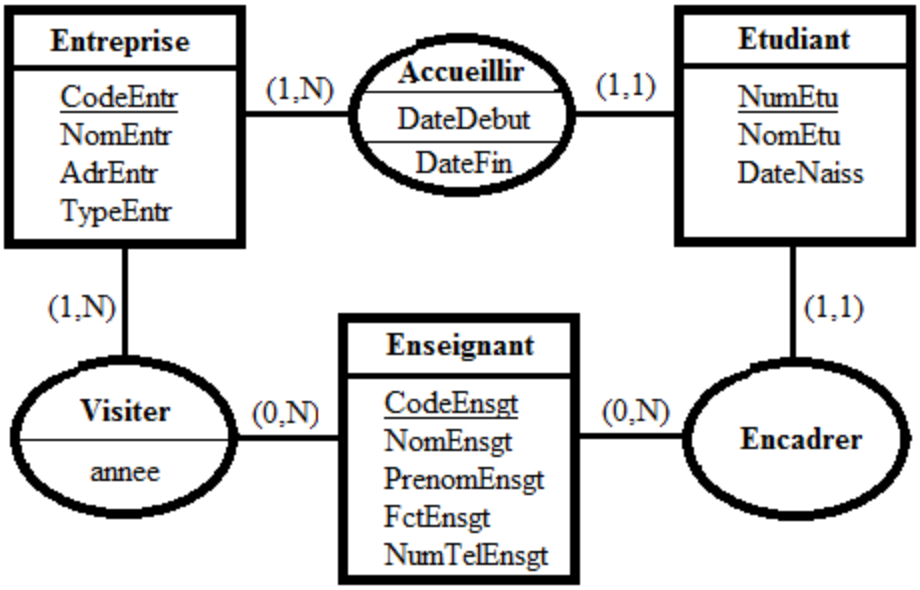


Figure 4: Exemple de Modèle Conceptuel de Données.

Explications :
Comme on peut le voir sur la figure 2, chaque entité contient un identifiant souligné, par exemple l'identifiant de l'entité *Entreprise* est CodeEntr.
De plus, l'association *Accueillir* lie les entités *Entreprise* et *Etudiant*. Cette relation crée des cardinalités telles que celle entre l'entité *Entreprise* et l'association *Accueillir* : (1, N). Cette cardinalité traduit le fait que l'entreprise accueille un ou plusieurs *Etudiant*.

1.2) Niveau logique

(http://www.labri.fr/perso/zemmari/m1dfac/c3.pdf)

Au niveau logique, à partir du modèle conceptuel, il faut établir le modèle relationnel.
Ce modèle a été introduit par Ted CODD en 1970. Il matérialise le schéma conceptuel par un ensemble de relations. Dans ce modèle, les entités du schéma conceptuel sont transformées en

tableaux afin de percevoir les données sous une forme plus simple. La manipulation de ces données s'effectue à l'aide d'un ensemble de règles mais aussi d'opérateurs algébriques que nous évoquerons.

Définitions :

Domaine : C'est un ensemble de valeurs distinctes caractérisé par un nom.

Relation : Correspond au sous-ensemble du produit cartésien d'une liste de domaines.

Attribut : Correspond à la colonne d'une relation.

N-uplet : Correspond à la ligne d'une relation.

Règles de passage d'un Modèle Entité-Association vers un schéma relationnel :

Règle n°1 : Chaque entité est traduite par une relation :
- La clé de la table est l'identifiant de l'entité.
- Les attributs sont les propriétés de l'entité.

Règle n°2 : Une association avec cardinalité 1 à plusieurs se réécrit en :
- Notant dans la relation fille, la clé primaire de la relation mère
- Ainsi la clé devient une clé étrangère.

Règle n°3 : Une association avec cardinalités plusieurs à plusieurs, a les caractéristiques suivantes :
- La création d'une relation ayant comme attributs les identifiants de l'association.
- Ces attributs sont la clé primaire de la relation.

Formes normales :

Ce sont les règles à appliquer pour s'assurer de la cohérence. Il en existe trois types :
- *1FN* : Éviter les groupes répétitifs.
- *2FN* : - La relation est en 1FN.
  - Les attributs dépendent de la totalité de la clé primaire.
- *3FN* : - La relation est en 2FN.
  - Toutes les dépendances sont directes.

Opérateurs algébriques :

- *Union* : Cet opérateur porte sur deux relations qui sont définies dans le même domaine et fournit une nouvelle relation qui a tous les n-uplets des deux relations initiales.

- *Intersection* : Pour deux relations ayant le même domaine, cet opérateur fournit une nouvelle relation qui a les n-uplets communs aux deux relations initiales.

- *Différence* : Pour deux relations ayant le même domaine, cet opérateur fournit une nouvelle relation qui a les n-uplets de la première relation, ne se trouvant pas dans la deuxième relation.

- *Projection* : Cet opérateur porte sur une relation et fournit une nouvelle relation définie par des attributs spécifiés.

- *Sélection* : Pour une relation ayant le même domaine, cet opérateur fournit une nouvelle relation qui retient que les n-uplets qui respectent une condition exprimée à l'aide d'opérateurs arithmétiques ou logiques.

- *Jointure* : Cet opérateur porte sur deux relations qui sont définies dans le même domaine et fournit une nouvelle relation avec les n-uplets des 2 relations initiales ayant même valeur pour un attribut commun.

Avec toutes ces règles précédemment expliquées, on peut maintenant transformer le Modèle Conceptuel de Données de la figure 2 en Modèle Logique de Données :

Entreprise (CodeEntr, NomEntr, AdrEntr, TypeEntr)
Etudiant (NumEtu, NomEtu, DateNaiss, DateDebut, DateFin, #CodeEntr, #CodeEnsgt)
Enseignant (CodeEnsgt, NomEnsgt, PrenomEnsgt, FctEnsgt, NumTelEnsgt)
Visiter (#CodeEntr, #CodeEnsgt, annee)

UML

(MIRANDA, 2002) (https://www.lri.fr/~longuet/Enseignements/12-13/Inge3-UML/Inge3-DiagClasses.pdf) (http://uml.free.fr/cours/i-p14.html)

UML (*Unified Modeling Language)* est un autre langage de modélisation plus récent qui est né en 1994 à partir de différentes méthodes lorsque la programmation par objets a pris de l'importance au début des années 1990.
Ce langage permet de représenter un système selon différents diagrammes :

- De classes.
- D'objets.
- D'états.
- De cas d'utilisation.
- De collaboration.
- De séquence.
- D'activités.
- De composants.
- De déploiement.

Nous expliquerons seulement le diagramme de classes parce qu'il est le plus important de la modélisation orientée-objet. C'est celui qui nous sera utile lors de la phase de production du projet.

Définitions :

Classe : C'est un regroupement d'objets, caractérisé par des attributs et des opérations.

Objet : Décrit par des attributs et des opérations.

Attribut : Correspond à la propriété partagée par tous les objets de la classe et associée à une valeur.

Opération : Effectuée à tout objet de la classe.

Association : Correspond à la relation entre classes.

Lien : Correspond à la relation entre objets.

Rôle : Définit la fonction d'une classe pour une association.

Dépendance : Une classe peut dépendre d'une autre.

Agrégation : Correspond à une association particulière entre classes. Il en existe deux types :
- *Agrégation faible* : Indique qu'une entité est faite d'une autre entité.
- *Composition* : Indique qu'une entité n'existe qu'en fonction d'une autre entité.

Héritage : Regroupe et organise les classes ayant des attributs et des opérations.

Cardinalités : Il en existe différents types : (a et b étant des entiers naturels)
- Exactement à : a.
- De a à b : a..b.
- Plusieurs : *.
- a ou plus : a..*.

Représentation graphique :

Les classes et les objets sont représentés par des rectangles contenant leurs attributs et leurs opérations.
Les associations et les liens sont représentés par un trait avec leurs noms.
Le rôle est écrit à l'extrémité de l'association.
La dépendance est représentée par une flèche en pointillés.
L'agrégation faible est représentée par un losange vide alors que la composition, par un losange plein.
L'héritage est représenté par un triangle.

Après la transformation du modèle conceptuel en un modèle logique, il faut passer au niveau physique dans ce cas il faut traiter l'exploitation.

Retro-ingénierie

La retro-ingénierie (ou reverse engineering) est une façon d'étudier un objet afin de déterminer son fonctionnement interne et le procédé de conception.
Dans le cadre d'une base de données existante, il faudrait reconstruire son modèle logique puis reconstruire le schéma physique. En effet, il faudra extraire le modèle physique construit par SQL, analyser le journal des modifications et le contenu de la base puis dés-optimiser le modèle extrait. [Hainaut 2012]

Après la transformation du modèle conceptuel à partir d'un modèle réelle ou d'une base de données existante en un modèle logique, il faut passer au niveau physique. Ce niveau fait partie du concept de l'exploitation d'une base de données. Cette partie traitera de la création du schéma de la base de données

L'exploitation d'une base de données

Notion de table

Dans les bases de données relationnelles, les tables représentent un ensemble de données organisées dans un tableau où les colonnes représentent les catégories d'information caractérisant un objet concerné. Dans une table, on doit mettre des espaces entre les mots qu'on y entre, par exemple. Une table contient des attributs de type integer, varchar, date, time, décimal, text, char, datetime, bigit, money... Pour écrire un attribut dans une table on doit mettre des espaces entre les mots qu'on écrit ; le logiciel ne reconnaît pas.
Il existe des tables courantes du genre avec des colonnes et des tables associatives qui assurent la liaison entre les tables associatives si elles existent.

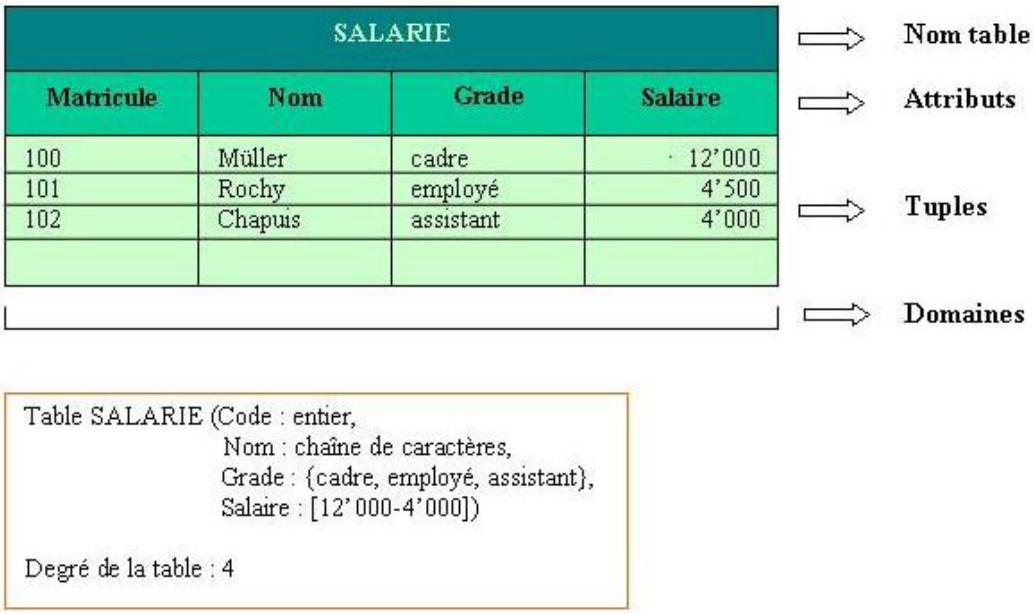


Figure 5: *extraite du cours de gestion de base de données de Luiz Gonzales sur le site d'unige.fr*
http://tecfaetu.unige.ch/staf/staf-h/tassini/staf2x/Heidi/last_bd.htm

La table SALARIE présente ci-dessus est une table à deux dimensions, où on peut distinguer très clairement, les attributs, les lignes correspondant aux informations suivant chaque individu.

Quand on crée des tables, et conformément aux bases de données relationnelles, on peut créer des contraintes d'intégrité qui sont entre autres les différents types de clés (primaire, étrangère).

Une clé primaire est une contrainte d'unicité, composée d'une ou plusieurs colonnes, et permet d'identifier de manière unique une ligne de cette table. C'est ainsi dans l'exemple suivant de base de données, nous avons une table avec une clé primaire qui s'auto incrémente afin que chaque ligne à sa clé unique qui l'identifie. Elle est Unique et non nul (NOT NULL).

Une clé étrangère est une contrainte qui permet de gérer les relations deux ou plusieurs tables, si nous sommes bien dans une base de données avec plusieurs tables.

Un index par contre est un objet complémentaire et non indispensable dans une table, entretenue dans un SGBD pour lui permettre de retrouver rapidement des données. L'emploi des index accélère les opérations de recherche, de tri effectué par le SGBD.

L'image ci-dessus montre la structure et le contenu d'une table.
Contenu montre les lignes qui compose la table. Ces lignes sont nommées uplet, tuplet ou n-uplet.

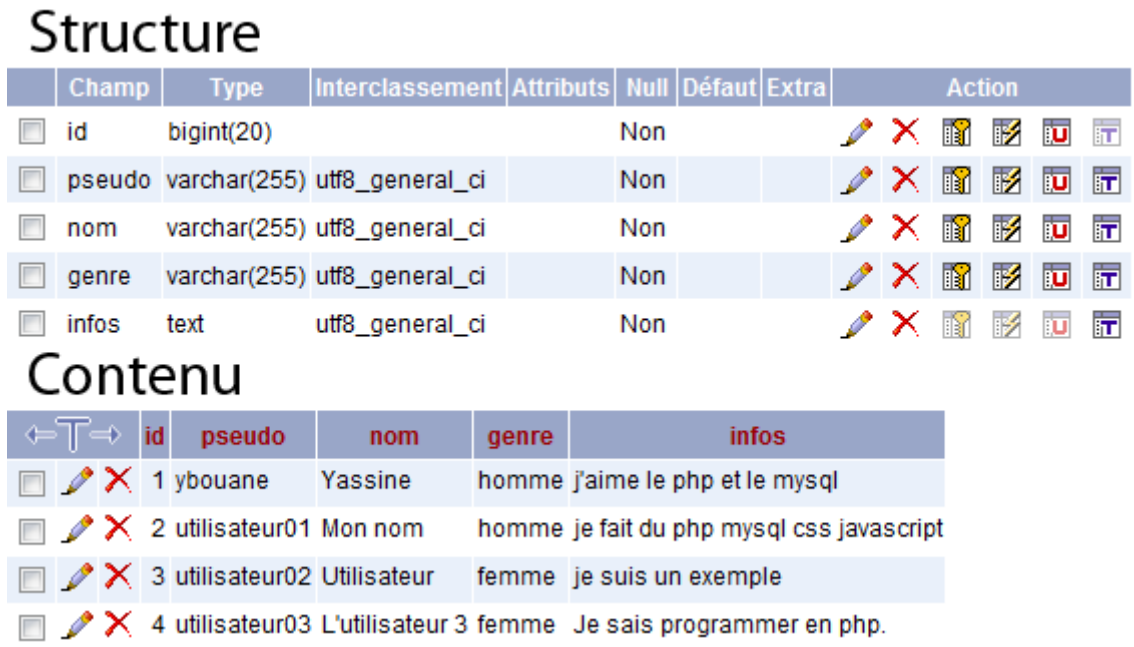


Figure 6: récupérée du site http://www.supportduweb.com/scripts_tutoriaux-code-source-82-mysql-recherche-dans-la-base-de-donnee-like-moteur-de-recherche-internet.html

Le langage SQL

SQL signifie en anglais Structured Query Langage (langage structurée de requête en français). C'est le langage le plus utilisé pour l'exploitation des bases de données relationnelles.
SQL a été normalisé à sept reprises : en 1986 (SQL 86-ANSI), en 1989 (ISO et ANSI), en 1992 (SQL2-ISO et ANSI), en 1999(SQL3), en 2003 (SQL 2003), en 2008 (SQL 2008) et enfin en 2011 (SQL 2011).
SQL se différencie en trois langages pour son exploitation : le DDL, DML et le SQL AVANCE. (Jean-Luc Hainaut, 2012)

LE LANGAGE SQL DDL

DDL signifie Data Définition Language, ce qui signifie que ce langage traite de la création des tables, de la modification des données. Il utilise des commandes spécifiques pour ces opérations précitées.
La fonction CREATE SCHEMA CLICOM permet de créer le schéma. Une fois le schéma défini on peut ensuite définir les conditions d'autorisation d'accès qui, seront utiles pour la connexion entre la base et le serveur ; et définir les composants de cette base.
La fonction CREATE TABLE permet de créer une table dans la base et on peut y intégrer entre parenthèses les différents attributs avec les contraintes si elles existent. Les différents types d'attributs cités dans la partie notion de tables sont : smallint (entiers signés courts); Integer(ou Int sont des entiers signés long) ; floats et decimal qui sont utilisés pour les nombres décimaux.
Les contraintes d'intégrité peuvent être définies lors de la création de la table en complément c'est-à-dire à la fin de la déclaration des attributs.
Pour supprimer une table, on utilise la fonction DROP TABLE. Pour ajouter une colonne à une table on fait : ALTER TABLE nom. Table add column nom.colonne Int. Pour supprimer une colonne dans une table, on fait ALTER TABLE nom. Table drop COLUMN nom.colonne.
Pour les contraintes, on peut aussi les ajouter ou les supprimer d'une table. On utilise la fonction ALTER TABLE pour identifier la table en question et Add pour ajouter les différents types de clés, et Drop pour supprimer chaque contrainte. (Jean-Luc Hainaut, 2012)

LE LANGAGE SQL DML

DML signifie Data Manipulation Language, qui traite les principes d'extraction des données d'une table. On extrait mais on modifie aussi les données avec les commandes de ce langage.
La commande SELECT permet d'extraire les données d'une ou plusieurs tables, précise les valeurs qui constituent chaque ligne du résultat ; la commande FROM indique la table où on récupère les valeurs ; et la commande WHERE spécifie la condition la condition de sélection que doivent satisfaire les lignes qui fournissent le résultat. Le résultat de ces 3 types de requêtes sera affiché à l'écran comme une table fictive. Si l'on souhaite les utiliser dans une application, celle-ci devra traiter les différentes réponses du serveur et y extraire les résultats.

Nous avons quatre types de tables qui obéissent à des règles contraires :
- tables de base : stockées de manière permanente et accessible à tout utilisateur ;
- tables fictives ou résultats des 3 requêtes ;
- tables dérivées qui sont des résultats de requêtes (INSERT INTO) stockés, et sont temporaires et pas accessible à tout le monde ;

-Vues : c'est des tables virtuelles dont les contenus sont les résultats d'une requête SFW qui leur est attachée.

Dans le DML on traite aussi l'extraction de données de plusieurs tables grâce aux jointures, les traitements de structure de données cycliques, les opérateurs de modification de donnée. Pour faire a jointure des tables il faut qu'il ait des conditions de jointure telles ligne1.table1.=ligne2.table3. La jointure des tables exclut les lignes qui n'ont pas de correspondants dans l'autre table. On utilise beaucoup plus les commandes Union pour la jointure et Union all pour un ensemble, Intersect pour l'intersection, Except pour la différence, Group by pour les groupes.
Cyclique est une structure de données qui fait directement ou non référence à elle-même. (Jean-Luc Hainaut, 2012)

LE LANGAGE SQL AVANCE

Le SQL AVANCE est utilisable pour ceux qui sont désireux de comprendre le SQL3, et on y définit les fonctions de contrôle d'accès et les vues, les formes puissantes de la requête SFW. On a vu que les données d'une base ne sont pas accessibles à tous. Du coup pour certaines bases, il faut des fonctions de règlementation de droits d'accès sous forme de privilège.
Un privilège est l'autorisation accordée à un utilisateur pour effectuer des opérations sur un objet. Les principales opérations admises sur le contenu des tables sont :
-SELECT : pour l'extraction des données ;
-INSERT : pour l'ajout de lignes ;
-DELETE : pour la suppression de lignes ;
-UPDATE : pour la modification des valeurs de certaines colonnes.
La commande GRANT permet la transmission d'un privilège par un destinataire à d'autres utilisateurs.
La commande REVOKE permet de la suppression d'un privilège accordé au préalable. (Jean-Luc Hainaut, 2012)

Le langage SQL est avant tout le plus basique et le plus courant des langages de base de données. Il n'est pas le seul puisqu'il existe entre autre le NoSQL, langage assez récent qui commence à faire ses preuves.

Une fois la base de données crée et exploitée, il se peut qu'il y ait des mises à jour des données pour éviter une obsolescence de la base. La partie suivante traitera des mises à jour et de la cohérence de la base de données.

Mise à jour d'une base de données et contrôle de l'intégrité

Le but de cette partie est de faire des recherches et de collecter le maximum d'informations afin de mettre à jour les données d'une base existante et de vérifier la cohérence.
La mise à jour de données est un processus de l'entrée de nouvelles données, de création et d'intégration des nouveaux champs, de collecte d'informations nécessaires et de faire un renseignement et une études sur les champs à mettre à jour (Christian Soutou 2013).

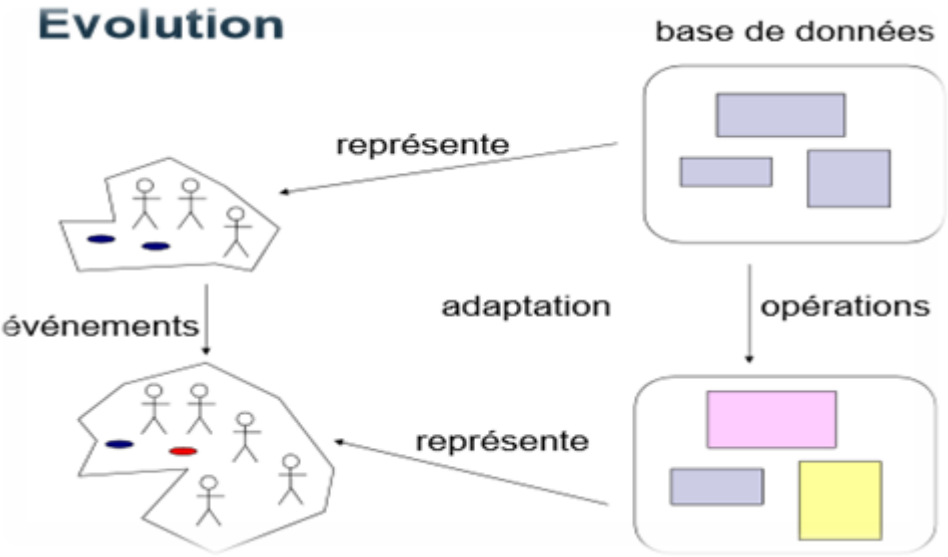


Figure 7: Simple exemple d'application d'une mise à jour d'une base de données extrait d'un article paru (G.Falquet et C.Métal, 2003)

Les données représentent une réalité, cette réalité change au cours du temps. Il faudra donc adapter ces données qui « définissent » une réalité.
Par exemple pour la création d'un emploi du temps, il est important de garantir le respect des règles d'intégrité pour éviter des chevauchements d'horaires.

Dans notre recherche, nous avons trouvé que parmi les méthodes existantes (Rigaux 2009) qu'il existe trois manières principales pour modifier une base de données : l'insertion, la destruction et la modification c'est le cas dans (Elmasri 2005) ou les auteurs utilisent ces trois principales commandes :

L'insertion s'effectue avec la commande INSERT qui sert à rajouter un tuple à une relation.
Il faut donc fournir le nom d'une relation et une liste des valeurs pour le tuple. Les valeurs doivent être écrites dans le même ordre que celui dans lequel ont été spécifiés les attributs.

La destruction s'effectue avec la commande DELETE dont la syntaxe est défini comme suite DELETE FROM table WHERE condition, *table* étant bien entendu le nom de la table, et condition est tout conditions, ou liste de conditions, valide pour une clause WHERE. En d'autre termes, si l'effectue, avant la destruction, on modifier la requête par SELECT*FROM table WHERE condition.
On obtient l'ensemble des lignes qui seront détruites par DELETE. On procède de cette manière est un des moyens de s'assurer que l'on va bien détruire ce que l'on souhaite.

La modification : La commande UPDATE sert à modifier les valeurs des attributs d'un ou plusieurs tuples sélectionnées. La syntaxe est la même que celle de la commande DELETE

UPDATE table A1=X1, A2=X2, … A(n)=X(n) WHERE ….
Les Ai sont les attributs, les Xi les nouvelles valeurs, WHERE permet de différencier les données à mettre à jour dans la base.
Par exemple dans (Wattiau 2001) la manipulation des données est mise en œuvre à l'aide d'un Langage de manipulation de Données (Data Manipulation Langage en anglais ou DML)

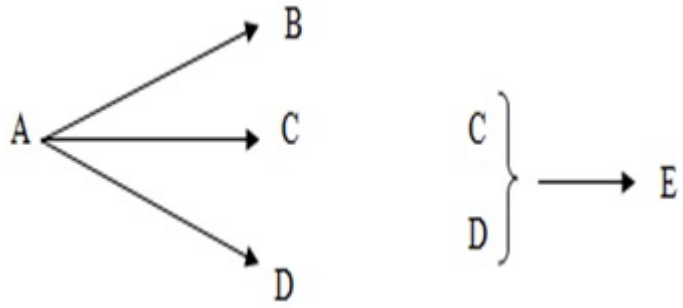
Anomalies de mise à jour et redondance

Nous avons constaté qu'un mauvais schéma défini lors de la phase de conception peut conduire à un certain nombre d'anomalies pendant la phase d'exploitation de la base (RIGAUX 2009) a décrit qu'une anomalie de mise à jour à lieu lorsqu'on fait une modification de la base. Pour éviter ce genre de problèmes, des contraintes et des mécanismes de contrôle sont intégrés aux SGBDR.
Mais cela nous pose :

- Une perte de temps dans la gestion de la base
- Certains contrôles pouvant être assez lourds
- Une implémentation rigoureuse de toutes les contraintes par le concepteur de la base

Le compromis est alors atteint en faisant l'hypothèse suivante: le concepteur n'implémente que des clés étrangères. Le contrôle automatique de ces contraintes est peu coûteux par le SGBD, et leur implémentation est toujours intégrée dans les SGBDR. Ainsi, on considère que toute mise à jour respecte ces clés.

Exemple
Supposons le schéma de relation :
ETUDIANT (NUMETUD -- A; NOM -- B; VILLE -- C; CP -- D; DPT -- E) muni de l'ensemble de la dépendance fonctionnelle (DF)



La clé minimale de la relation est donc A (Toutes les autres clés sont des sous-ensembles de A). Supposons qu'un tuple (ligne) soit inséré pour un nouvel étudiant, avec une ville et un CP déjà présent mais un département différent. La clé ne sera pas violée (pas de doublon sur A) mais la DF de C et D → E ne sera plus satisfaite. Puisqu'un tel cas de figure est possible, la relation ETUDIANT possède une anomalie de mise à jour.

Une suppression ne peut entraîner aucune anomalie proprement dite ; toutefois, elle peut engendrer une perte involontaire d'information, lié à la redondance dans les données. Dans l'exemple précédent, si on supprime tous les étudiants de l'université d'Evry, on aura

également perdu le code postal et leurs départements de même si on souhaitait garder cette information.

Redondance des données

La notion de redondance est une autre façon de considérer les problèmes de mises à jour. Elle se définit sur les relations, alors les problèmes de mise à jour portent sur des schémas.

Exemple :

Soit le schéma de relation ***FOURNISSEUR (Nom_Fournisseur, Adresse, Produit, Prix).***
Une relation (table) correspondant à ce schéma pourra contenir plusieurs produits pour un même fournisseur. Dans ce cas, l'adresse du fournisseur sera dupliquée dans chaque n-uplet (redondance). Si on souhaite modifier l'adresse d'un fournisseur, il faudra rechercher **et mettre à jour** tous les n-uplets correspondant à ce fournisseur. Si on insère un nouveau produit pour un fournisseur déjà référencé, il faudra vérifier que l'adresse est identique.
Si on veut supprimer un fournisseur, il faudra retrouver et supprimer tous les n-uplets correspondant à ce fournisseur (pour différents produits) dans la table.

Ces anomalies n'apparaitront pas si on décompose le schéma initial de base de données.

La mise à jour des BD est très importante car elle représente la garantie et la longévité d'une base de données efficace. Cependant une base de données peut rencontrer plusieurs problèmes tels que le manque de sécurité ou bien encore la lourdeur d'accès aux données. Afin d'y remédier, il est nécessaire de recourir à un SGBD fournissant une interface de compréhension.

Gérer une base de données avec un SGBD

Définition
Un système de gestion d'une base de données (SGBD) est un ensemble de programme permettant de gérer le stockage et l'accès d'une base de données (http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es). Ce système assure le bon fonctionnement c'est-à-dire garantir la qualité, la durabilité et la confidentialité des informations (http://sql.sh/sgbd). De plus il permet le partage de données entre les différents utilisateurs.

Il existe 5 types de modèle de SGBD :

Hiérarchique :
Première génération de logiciels crée en 1960. Les données sont classées de manière hiérarchique, dans une arborescence descendante où chaque nœud est une structure de données, appelée segment et dont les branches sont des systèmes de pointeurs reliant ces structures. (WATTIAU 2003)

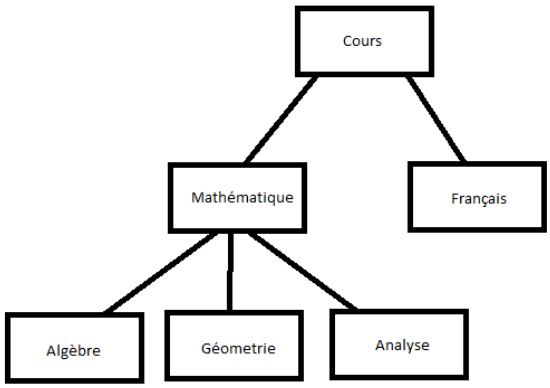


Figure 8: Schéma du modèle hiérarchique

Réseau :
En 1970, un nouveau SGBD voit le jour. Celui ce base sur le principe que le modèle hiérarchique mais offre des beaucoup plus élargies. C'est-à-dire que l'arborescence n'est plus seulement descendante. (WATTIAU 2003)

Relationnel :
Les données sont représentées sous forme de table. Ce modèle de SGBD met en pratique la méthode Merise (entité-association). (WATTIAU 2003)

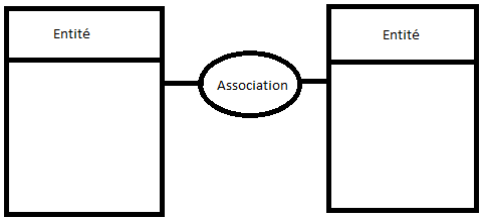


Figure 9: Schéma du modèle relationnel

Déductif :
Identique au modèle relationnel mais la manipulation sa fait par un calcul de prédicats. (http://www.commentcamarche.net/contents/105-les-modeles-de-sgbd)

Objet :
Les données sont stockées sous la forme d'instance (objet) de classes hiérarchisé qui possèdent leur propre méthode d'exploitation. (WATTIAU 2003)

Fonctionnement d'un système de gestion de base de données

Les SGBD sont des logiciels qui se situent entre utilisateurs et la base de données. (Voir ci-dessous)

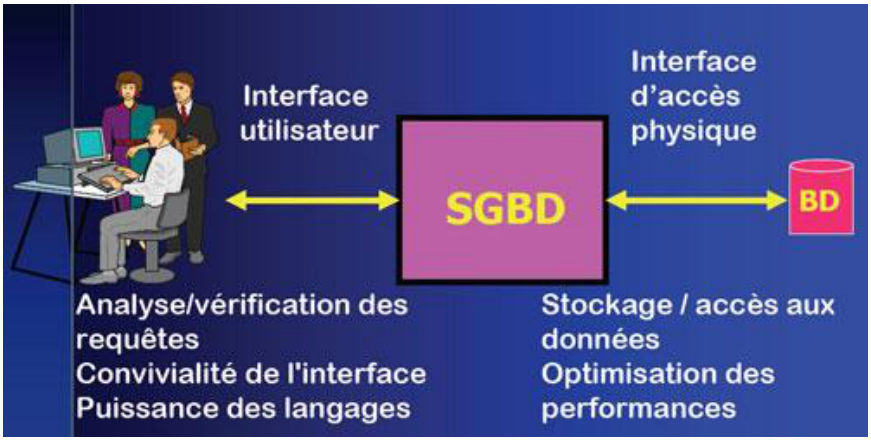


Figure 10: Schéma représentant un utilisateur et sa base de données (aiservice.fr année 2013)

Il sert d'interface entre les programmes et les fichiers de données. Sa fonction est d'enregistrer des informations, puis les rechercher ainsi que les modifier. Il permet de spécifier le type de données, les structures des informations et la cohérence des données. Ce qui octroie une liberté aux programmeurs et aux utilisateurs sur la connaissance du stockage des données dans un SGBD.

Le SGBD doit remplir ses conditions pour garantir la fiabilité de notre base de données :

La qualité des données : éviter toutes les redondances des différentes données.
La concurrence des accès : gérer la gestion d'accès lorsque plusieurs utilisateurs désirent accéder en même temps aux données
Confidentialité des données : gérer les droits (création, modification, permission et consultation) des différents utilisateurs.
Sécurité des données : assurer la sécurité des données contre les différents problèmes matériels ou logiciels.

Le SGBD assure le suivi des manœuvres en fournissant des statistiques sur les utilisations de la base et du service de gestion.

Processus

Les SGBD suit un processus afin d'exécuter les différentes requêtes exiger par les utilisateurs. Ce processus est reparti en quatre étapes qui sont :

- Utilisateurs émet une requête (requête en langage SQL).
- SGBD reçoit la requête et l'analyse.
- SGBD inspecte le schéma externe cet utilisateur, le lien externe/conceptuel correspondant, le schéma conceptuel, le lien conceptuel/interne et la définition de la base de données mémoire.
- SGBD exécute les opérations nécessaires sur la base de données.
- La gestion de données

Le rôle de du système est de gérer les différentes données inclus dans la base de données. Pour cela le SGBD doit parfaitement connaitre l'ensemble des informations contenu dans sa base. Il doit pouvoir manœuvrer sa base de données c'est-à-dire ajouter, supprimer, modifier les informations. Cette manipulation se fait avec le langage SQL qui est actuellement le plus répandu dans le monde.
De plus le SGBD fournit un dictionnaire de données qui liste l'ensemble des données ainsi que leurs caractéristiques. Ces caractéristiques représentent leur utilisation, leur représentation physique leur propriétaire et les autorisations. Cet outil est primordial pour la gestion. Les utilisateurs et les programmeurs peuvent connaitre les spécificités des différentes données.

De nos jours, il existe énormément de SGBD dont deux qui sont utilisés dans notre projet Access et MySQL.

Access/MySQL : Présentation

Access

Figure 11: Logo du logiciel Access (Wikipédia)

Un logiciel de système de gestion de base de données crée par la société américaine Microsoft qui a pour spécification une utilisation d'une interface graphique pour saisir les requêtes et un langage de programmation applicable sur les bases de donnée axé sur ce logiciel. (http://sql.sh/sgbd/microsoft-access)

MySQL

Figure 12: Logo du logiciel MySQL (Wikipédia)

MySQL est un serveur de base de données qui fonction sous plusieurs système d'exploitation (Windows, mac, linux) qui a pour objectif d'être rapide résistant et facile d'utilisation.

MySQL a été développé par la société suédoise TcX qui devait répondre au attente de la base de donnée SQL qui gérer des grandes bases de données de manière rapide et efficace. (http://www.linux-france.org/article/cel/SICOMOR/SGBDR/html/Rapport_7-9V10.html)

De même qu'Access, MySQL est un logiciel de gestion qui est le plus utilisé au monde. Il est très connu des développeurs car il fait partie du célèbre quatuor :

- WAMP (window, apache, MySQL, PHP)
- LAMP (Linux, apache, MySQL, PHP)
- MAMP (Mac, apache, MySQL, PHP)

Caractéristique :

- Produit sous forme de code source
- Mémoire importante
- Utilisation de différent langage (C, C++, java, PHP, Perl, Python)
- Information sur les procédures
- Optimisation des performances
- Bonne intégration avec le serveur Apache et le langage PHP

Choix du SGBD

Il existe beaucoup de SGBD pour gérer notre base de données. Ce choix peut être parfois difficile. Notre SGBD doit répondre à certaines caractéristiques qui sont :

- La taille de la base donnée
- Le nombre d'utilisateurs
- Le temps
- Son intégration
- Disponibilité de fonctions spécifique
- Le coût

La taille de la base de données correspond au nombre d'information nécessaire à l'ensemble des données d'une organisation. Cette taille est définit selon deux paramètres la mémoire qu'on dispose pour les données et le type de SGBD qu'on utilise.

Le nombre de personnes pouvant entrer en même temps dans la base de données est aussi un critère important car les utilisateurs ne doivent pas être sanctionnés c'est-à-dire une exclusion de la base si trop de personnes sont connectées en même temps.

La SGBD doit être aussi évolutif car la base la taille de la base et le nombre d'utilisateur peut augmenter.

Le temps joue un rôle très important car l'accès doit est rapide ainsi que le temps de réponse selon les requêtes envoyées par les utilisateurs, c'est-à-dire de l'ordre de la milliseconde.

Le SGBD doit être capable d'intégrer des informations provenant d'autres bases par des processus tel que l'importation et l'exportation. Sa prise en main doit être facile et non complexe.

Le SGBD a un rôle très important à jouer pour l'organisation des BD. Il facilite la liaison entre les utilisateurs et leur BD en utilisant uniquement envoi de requêtes. Ce qu'ils permettent de gagner du temps dans leurs recherches dans la recherche d'information pour les entreprises. (http://www.christian.braesch.fr/page/choix-dun-sgbd)

Les BD facilitent la gestion de donnée de beaucoup d'entreprise et des particuliers mais celle-ci doit assurer la sécurité des informations qu'elle contient.

Sécurité des bases de données.

Dans le cadre de la loi

Dans le cadre de la loi, les personnes doivent être informées des données enregistrées dans la base de données. (Article 32 de la loi informatique et liberté). La base de données doit être déclarée au CNIL (Commission Nationale de l'Informatique et des Libertés) sauf ci-celle-ci rejoint les critères établie par la commission afin d'obtenir une protection juridique.

Du point de vue informatique

Les bases de données sont une cible prisée des hackers, en effet, la revente d'information confidentielle est monnayée par différentes organismes à effet lucratif. Les gestionnaires de bases de données utilisent en moyenne un SGBD qui possède une sécurité minimale et ayant les critères suivants :

- Identification et authentification des utilisateurs
- Contrôle d'accès aux données
- Contrôle d'accès aux services d'administration
- Protection des communications réseau
- Intégrité du logiciel
- Tolérance aux erreurs
- Modification du code source

Selon le SGBD utilisé, il existe des fonctions natives qui faudrait activer. Les SGBD libres présente un net avantage en termes de sécurité par rapport aux SGBD commerciaux.

La sécurité de la base de données est primordiale pour garantir celle de ses différents utilisateurs.

Conclusion

Cette recherche bibliographique a permis à notre groupe de posséder des connaissances sur les bases de données et de les renforcer.

Nous avons ainsi les informations essentielles sur la création, la gestion et l'exploitation d'une base de données.

On notera que la phase la plus importante dans notre projet sera la création du modèle conceptuelle de la base de données qui se doit d'être réutilisable et robuste.

Pour cela il est important de rester en contact avec notre client afin de définir un cahier des charges extrêmement précis et claire.

Enfin ce travail a permis de mieux rapprocher notre groupe et de mieux gérer nos temps de travail.

Remerciements

Nous tenons à remercier Mme Lydie NOUVELIERE pour tous ses précieux conseils et son expertise sur les recherches bibliographiques, M. Christophe MONTAGNE pour ses conseils avisés et sa tutelle ainsi que tous les intervenants ERD pour les réponses à nos différentes questions.

Références

Ouvrage :

Serge MIRANDA, Bases de Données : Architectures, modèles relationnels et objets, SQL 3. France : DUNOD, 2002, 458 pages. (Collection InfoPro).

Yasmina SADI, Cours n°1 de Base de Données. Analyse et conception des Systèmes d'Information : Evry, IUP d'Evry, licence, 2014, 32 pages.

Yasmina SADI, Cours n°2 de Base de Données. Le modèle logique (Relationnel) de données : Evry, IUP d'Evry, licence, 2014, 26 pages.

Jean-Luc HAINAUT, Cours de bases données, Dunod : France, 2012, 702 pages.

Patrice BOURSIER, Cours de Base de Données variable, structure de contrôle interaction avec la base mise à jour et extraction : La Rochelle, Université de la rochelle, licence, 2002, 33 pages

Christian SOUTOU, Programmer avec MySQL, Eyrolles : France, 2013, 502 pages

Philippe RIGAUX, Pratique MySQL et PHP : Création d'une base MySQL, Dunod : France, 2009, 533 pages

Ramez ELMASRI et Shamkant NAVATHE, Conception et architecture de base de données, Pearson Education : France, 2005, 700 pages

Ramez ELMASRI et Shamkant B. NAVATHE, Database Systems, Pearson, 6 Edition: France, 2011, 1155 pages

NACER Boudijlida, Architecture et administration de SGBD : éléments de méthode, Gestion et administration des bases de données, Dunod : France, 2003, p 238

Chris J.Date, une architecture pour les bases de données, Introduction aux Bases De Données, Vuibert: Massachusetts, Etats-Unis, 2004, p 44

I.Comyn-Wattiau et J.Akoka, les générations de SGBD, Les Bases De Données, PUF : Paris, France, 2003, p27-35

Web

Mathias KLEINER. SGBD : Conception d'une base de données [en ligne]
http://www.lsis.org/kleinerm/files/AM/SGBD/CM1.pdf
(Consulté le 12.01.15)

Dana TORRES, Luis GONZALEZ, Sara TASSINI. Cours de Bases de Données [en ligne]
http://tecfaetu.unige.ch/staf/staf-h/tassini/staf2x/Heidi/last_bd.htm
(Consulté le 16.01.15)

A. ZEMMARI. Modèle relationnel [en ligne]
http://www.labri.fr/perso/zemmari/m1dfac/c3.pdf
(Consulté le 16.01.15)

Delphine LONGUET. UML : Diagrammes de classes [en ligne]
https://www.lri.fr/~longuet/Enseignements/12-13/Inge3-UML/Inge3-DiagClasses.pdf
(Consulté le 17.01.15)

Laurent PIECHOCKI. Les vues statiques d'UML : Diagramme de classes [en ligne]
http://uml.free.fr/cours/i-p14.htm
(Consulté le 17.01.15).

Fabien DEMARCHI. Conception des bases de données relationnelles: Cours de BD [en ligne]
http://liris.cnrs.fr/fabien.demarchi/PageWebIF10/
(Consulté le 12.01.2014)

Wikipédia - système de gestion de base de données [en ligne]
http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es
(Consulté le 1.12.2014).

SQL - système de gestion de base de données [en ligne]
http://sql.sh/sgbd
(Consulté le 1.12.2014)

Comment ça marche – modèle de SGBD [en ligne]
http://www.commentcamarche.net/contents/105-les-modeles-de-sgbd
(Consulté le 02.12.2014)

Christian BRAESCH - Système de gestion de base de données [en ligne]
http://www.christian.braesch.fr/page/choix-dun-sgbd
(Consulté le 15.01.2015)

SQL- Access [en ligne]
http://sql.sh/sgbd/microsoft-access
(Consulté 16.01.2015)

Linux-France – Caractéristiques générales de MySQL [en ligne]
http://www.linux-france.org/article/cel/SICOMOR/SGBDR/html/Rapport_7-9V10.html
(Consulté le 16.01.2015)

Listes des sigles

BD : base de données
SGBD : Système de gestion de base de données
SQL : Structured Query Language