

# MERISE : MCD & MPD

## APPROCHE THEORIQUE

# 1 - MCD : Modèle Conceptuel de Donnée

## A - Méthodologie de construction du MCD

La méthode décrite ici est basée sur MERISE, méthode française qui a plus de 20 ans. Elle consiste à concevoir un Modèle Conceptuel de Données (MCD), qui sera transposé en Modèle Logique de Données Relationnelles (MLDR), lequel générera le Modèle Physique de Données (MPD) correspondant à la Base choisie. C'est la plus répandue des techniques d'analyse de Base de Donnée. Elle intègre un formalisme graphique très simple.

Nous étudierons plus particulièrement dans cet article la construction du Modèle Conceptuel de Données et de ses 5 caractéristiques : Entités, Propriétés, Identifiants, Associations, Cardinalités.

Il est intéressant de remarquer qu'à ce stade de la conception, toute la méthodologie est strictement indépendante du matériel, du système d'exploitation, ainsi que du système de base de données.

## B - Décrire le système d'information

Durant la phase d'apprentissage, il est conseillé de transformer ce que l'on veut analyser en mots simples. L'écriture de cette petite rédaction permet à elle seule de bien comprendre ce que l'on va modéliser, ce que l'on appelle le Système d'Information (S.I.). L'analyste passera directement à la phase de schématisation une fois le processus bien assimilé.

Il s'agit aussi à ce stade d'établir un lien entre l'informaticien et les utilisateurs, il ne faut donc pas hésiter à faire relire votre petit texte et à poser toutes les questions qui vous viennent à l'esprit, afin de bien analyser l'existant. La difficulté principale est d'arriver à faire abstraction des habitudes de programmation : à ce stade, nous sommes totalement indépendants du matériel et du logiciel. Il ne faut pas penser en terme de tables, mais en terme d'entités.

Prenons l'exemple très simple d'un logiciel (ou partie de site Web) ayant pour but de gérer les envois de NewsLetters aux abonnés d'un site ayant plusieurs rubriques. Le service marketing veut aussi savoir quelle raison a poussé l'abonné à s'inscrire en lui proposant plusieurs choix de motivations lors de son inscription.

Le système d'information se décrit ainsi : *"Un abonné est inscrit à une ou plusieurs rubrique. Chaque rubrique envoie une Newsletter chaque semaine, aux abonnés de la rubrique correspondante. Un abonné a une motivation d'inscription parmi plusieurs possibles."*

Ces quelques phrases, si elles sont exactes et validées par le client, sont suffisantes pour modéliser notre premier modèle. Elles contiennent en effet toutes les informations nécessaires.

## La méthode de modélisation comprend 5 étapes simples :

### 1 - Identifier les entités présentes

L'entité ABONNES représente l'ensemble des abonnés.

L'entité RUBRIQUES l'ensemble des rubriques auxquelles l'abonné peut s'inscrire.

L'entité NEWSLETTERS représente les newsletters envoyées,

MOTIVATIONS l'ensemble des motivations d'inscriptions des abonnés.

D'où les 4 entités :



Une légère analyse grammaticale suffit bien souvent à identifier les entités présentes : ce sont les verbes et les compléments d'objets de l'analyse du système d'information.

Généralement, une entité est créée dans le système d'information si elle possède au moins 2 occurrences. Chaque élément d'une entité est appelé une occurrence de l'entité.

### 2 - Lister les propriétés des entités

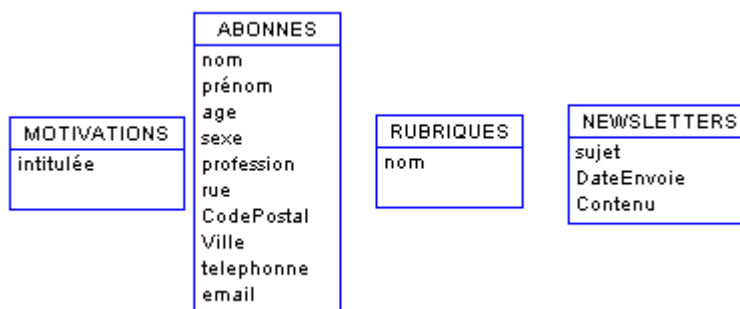
Un Abonné est caractérisé par son nom, son prénom, son âge, son sexe, sa profession, sa rue, son code postal, sa ville, son pays, son téléphone et son email.

Une Newsletter est caractérisée par son sujet, sa date d'envoi et son contenu.

Une Motivation est caractérisée par son intitulé.

Une Rubrique est caractérisée par son nom.

Les 4 entités deviennent :



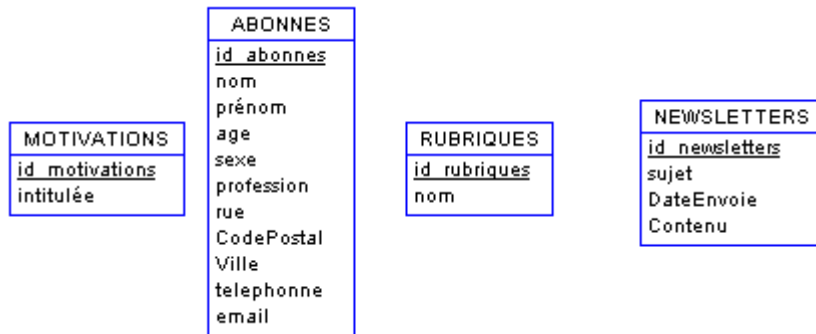
Afin de ne pas en avoir trop, on se limite généralement aux propriétés nécessaires au développement. Chaque propriété doit avoir une seule valeur possible pour chaque occurrence, sinon il s'agit d'une entité. Elle doit de plus être élémentaire et non-décomposable. Par exemple, l'adresse n'est pas une propriété élémentaire : elle comporte une rue, un Code Postal et une ville qui elles, sont 3 propriétés élémentaires.

### 3 - Identifier de manière unique chaque occurrence

Imaginons que nous ayons deux abonnés qui s'appellent DUPOND : il est nécessaire de les distinguer sous peine de les confondre. On rajoute alors une propriété qui permettra d'identifier de manière unique chaque occurrence. Cette propriété est appelée l'identifiant de l'entité. Cela peut être une référence interne, un code, ou plus généralement un nombre entier.

Cette propriété est soulignée afin de mettre en évidence son rôle d'identifiant.

Les 4 entités sont finalement :



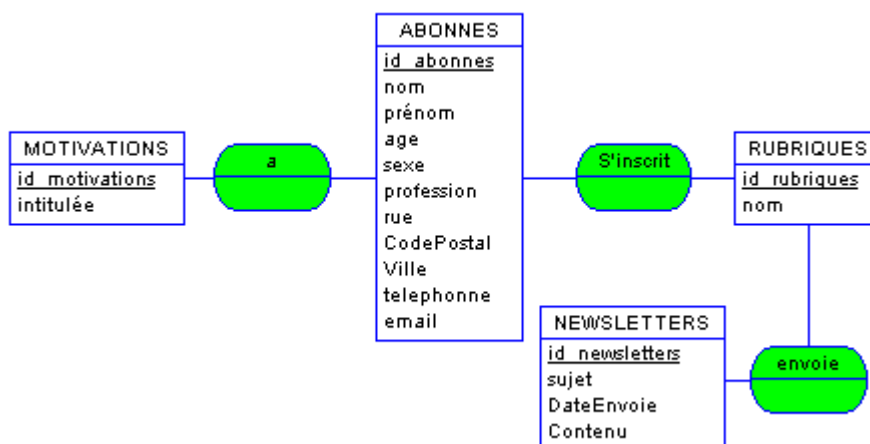
### 4 - Etablir les relations entre les différentes entités

maintenant, il s'agit d'identifier les relations entre les entités. Généralement, la simple transposition du texte suffit, les sujets et compléments d'objets étant les entités, et les verbes les relations.

Reprenons notre texte initial :

"Un Abonné *a* une Motivation. Un Abonné *s'inscrit* à une ou plusieurs Rubriques. Chaque Rubrique *envoie* une NewsLetter."

Les verbes sont en rouge et relient les entités. Il suffit de les intégrer au schéma :



## 5 - Identifier les cardinalités

Il faut maintenant établir le nombre possible d'interactions entre les entités.

Il s'agit d'un couple d'entiers de type (a ; b).

a est la cardinalité minimum, et est égal à 0 ou 1.

b est la cardinalité maximum, et est égal à 1 ou n, n étant plus grand que 1.

Continuons notre exemple :

Un Abonné a ici une et une seule Motivation d'inscription, le marketing ayant imposé un champ obligatoire afin d'avoir cette valeur. On a donc 1 minimum, et 1 maximum. D'où la cardinalité (1 ; 1).

Une Motivation donnée concerne 0 ou plusieurs Abonnés. On a donc 0 minimum, et n en maximum. D'où la cardinalité (0 ; n).

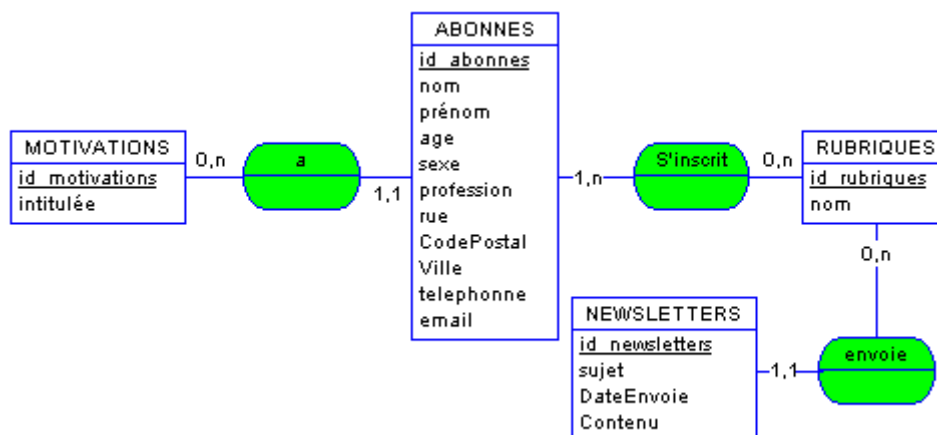
De même, un Abonné s'inscrit à une ou plusieurs Rubriques : (1 ; n),

Et une Rubrique possède 0 ou plusieurs Abonnés : (0 ; n).

Enfin, une Rubrique envoie 0 ou plusieurs Newsletters : (0 ; n),

Et une Newsletter appartient à une et une seule Newsletter : (1 ; 1).

Il suffit maintenant de marquer ces couples sur le schéma, et nous avons notre Modèle Conceptuel de Donnée (MCD) :



### Et pour finir : Valider le Modèle avec le client

A ce stade, il est aisé d'aller voir encore une fois les utilisateurs du logiciel final, afin de discuter le MCD avec eux. Cela vous permettra d'entériner les propriétés qu'ils désirent utiliser, d'être bien certain des cardinalités, et de valider avec eux cette partie de votre travail. Un MCD doit pouvoir s'expliquer avec des phrases simples et être compréhensible par tout le monde. Il ne s'agit ni plus ni moins que de modéliser l'existant. Ainsi, vous serez certain de faire le développement demandé, et cela vous permettra de vous protéger par la suite en cas de nouvelles demandes ou de modifications du cahier des charges.

Il est important de bien réaliser que jusqu'à ce stade, toute cette analyse s'est déroulée totalement indépendamment de la machine ou de toute contrainte logicielle.

Ces règles fonctionnent toujours, même s'il peut y avoir parfois plusieurs solutions pour chaque modèle. Le processus de modélisation, après quelques tentatives, est très simple à acquérir. Enfin, une fois le Modèle Conceptuel de Donnée établi, vous aurez fait le plus difficile. La conception de la base qui en découle est mécanique, et repose sur 6 règles strictes, nécessaires et suffisantes. Transformer un MCD en Modèle Logique, puis Physique est tellement standardisé que certains logiciels le font automatiquement...

## 2°) MPD : Modèle Physique de Donnée

Transformer un Modèle Conceptuel de Données (MCD) en Modèle Physique de Données (MPD) repose sur **6 règles simples**, nécessaires et suffisantes. Que cette transformation soit manuelle ou assistée par un logiciel de modélisation, il est utile de posséder les connaissances théoriques sur le sujet. Auparavant, il faudra bien sûr acquérir les notions de clé primaire, de clé étrangère, ainsi que quelques termes élémentaires.

### Préliminaires :

Un peu de vocabulaire : Les **données** sont stockées dans des **relations**. Une **relation** est un ensemble de **T-uple**, et un **T-uple** est défini par un ou plusieurs **attributs**. Dans la pratique, la **relation** est en fait la table, un **T-uple** est une **ligne** (ou **enregistrement**), et les **attributs** sont les **colonnes**. Les **attributs** sont aussi appelés **champs**.

### Exemple d'une table, appelée NEWSLETTER :

<u>id_newsletter</u>	Sujet	DateEnvoie	Contenu	id_rubrique
25	Newsletter N°25	11/01/2001	Texte 25	10
26	Newsletter N°26	21/01/2001	Texte 26	20
...	... ..	... ..	... ..	...

Cette table est décrite ainsi :

**NEWSLETTER** (id\_newsletter, Sujet, DateEnvoie, Contenu, #id\_rubrique)

### Exemple d'une autre table, appelée RUBRIQUE :

<u>id_rubrique</u>	Nom
5	Rubrique 5
10	Rubrique 10
...	... ..

Cette table, elle, est décrite ainsi :

**RUBRIQUE** (id\_rubrique, Nom)

Chaque **enregistrement** doit être identifié de manière unique (voir la notion d'identifiant abordée dans l'article précédent). L'**attribut** qui permet d'identifier de façon unique chaque ligne est appelée la **Clé Primaire**. Elle peut être composée, c'est-à-dire comprendre plusieurs **attributs**. Pour **NEWSLETTER**, il s'agit de l'**attribut id\_newsletter**. Pour **RUBRIQUE**, il s'agit de l'**attribut id\_rubrique**.

La **table NEWSLETTER** comprend un **attribut** provenant de la **table RUBRIQUES**, l'**attribut id\_rubrique**. Cet **attribut** est appelé **Clé Etrangère**. Ici, il sert par exemple à retrouver les Newsletters correspondantes à une rubrique donnée.

Dans le formalisme, la **clé primaire** est soulignée, et la **clé étrangère** est précédée du signe #.

D'où l'écriture définitive :

**MATABLE** (Cle\_Primaire, Colonne1, Colonne2, #Cle\_Etrangere)

Rappel de notre exemple :

**NEWSLETTER** (id\_newsletter, Sujet, DateEnvoie, Contenu, #id\_rubrique)

**RUBRIQUE** (id\_rubrique, Nom)

Ici, **id\_rubrique** est la **Clé Primaire** de la table **RUBRIQUE**, et est une **Clé Etrangère** dans la table **NEWSLETTER**.

Ce formalisme est ce que l'on appelle le **Modèle Logique de Données** (MLD). Il s'agit du passage entre le Modèle Conceptuel de Données et l'implémentation physique de la base. Le MLD est lui aussi indépendant du matériel et du logiciel, il ne fait que prendre en compte l'organisation des données. C'est d'ailleurs le point primordial de la modélisation : si l'organisation des données est relationnelle (si elles sont "liées" entre elles), alors le MLD est relationnel et devient le MLDR, ou Modèle Logique de Données Relationnelles. Pour la petite histoire, le MLDR a été inventé par Codd en 1970, et repose sur la théorie ensembliste...

Une fois assimilée ces notions de **clés primaires** et de **clés étrangères**, nous pouvons maintenant énoncer les 6 règles suivantes :

### **1 : Une entité se transforme en une relation (table)**

Toute entité du MCD devient une relation du MLDR, et donc une table de la base de données. Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante. L'identifiant de l'entité devient la **Clé Primaire** de la relation (elle est donc soulignée), et donc la **Clé Primaire** de la table correspondante.

Exemple :

Modèle Conceptuel de Données (MCD) :

CLIENT
<u>id_client</u>
Nom_Client
Tel_Client

Modèle Logique de Données Relationnelles (MLDR) :

CLIENT (id\_client, Nom\_Client, Tel\_client)

Modèle Physique de Données (MPD), ou schéma de base :

CLIENT
<u>id_client</u>
Nom_Client
Tel_Client

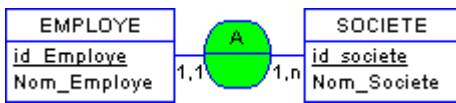
### **2 : Relation binaire aux cardinalités (X,1) - (X,n), X=0 ou X=1**

La **Clé Primaire** de la table à la cardinalité (X,n) devient une **Clé Etrangère** dans la table à la cardinalité (X,1). Si la relation est porteuse de données, celle-ci devient un attribut dans la table à la cardinalité (X,1) :

Exemple de Système d'Information (SI) :

Un employé a une et une seule société. Une société a 1 ou n employés.

Modèle Conceptuel de Données (MCD) :

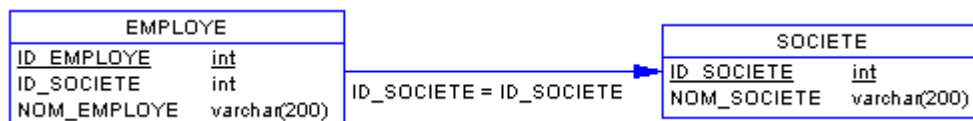


Modèle Logique de Données Relationnelles (MLDR) :

EMPLOYE (id\_Employe, Nom\_Employe, #id\_Societe)

SOCIETE (id\_Societe, Nom\_Societe)

Modèle Physique de Données (MPD), ou schéma de base :



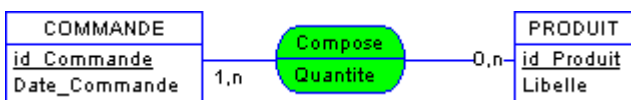
**3 : Relation binaire aux cardinalités (X,n) - (X,n), X=0 ou X=1**

Il y a création d'une table supplémentaire ayant comme **Clé Primaire** une clé composée des **identifiant** des 2 entités. On dit que la **Clé Primaire** de la nouvelle table est la **concaténation** des **Clés Primaires** des deux autres tables. Si la relation est porteuse de données, celles-ci deviennent des attributs pour la nouvelle table.

SI :

Une commande est composée de 1 ou n produits distincts en certaine quantité. Un produit est présent dans 0 ou n commandes en certaine quantité.

MCD :



MLDR :

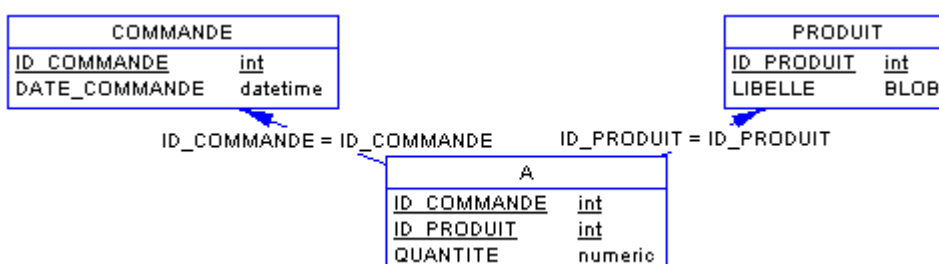
COMMANDE (id\_Commande, Date\_commande)

PRODUIT (id\_Produit, libelle)

COMPOSE (#id\_Commande, #id\_Produit, quantité)

La table COMPOSE servira en pratique à retrouver tous les produits composants chaque commande. De même, pour chaque produit, on pourra retrouver toutes les commandes où il figure.

MPD :



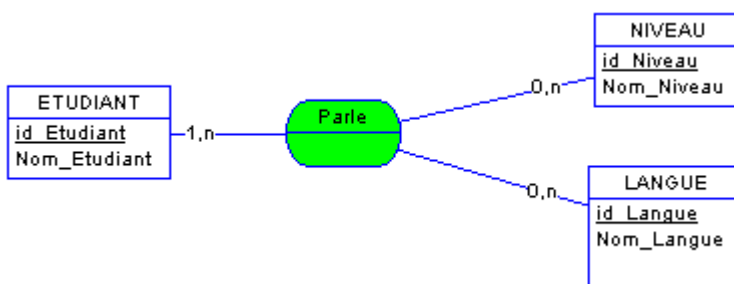
#### 4 : Relation n-aire (quelles que soient les cardinalités)

Il y a création d'une table supplémentaire ayant comme **Clé Primaire** la **concaténation** des **identifiants** des **entités** participant à la **relation**. Si la relation est porteuse de données, celles-ci deviennent des **attributs** pour la nouvelle table.

SI :

Un étudiant parle une ou plusieurs langues avec un niveau. Chaque langue est donc parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue.

MCD :



MLDR :

ETUDIANT (id\_Etudiant, Nom\_Etudiant)

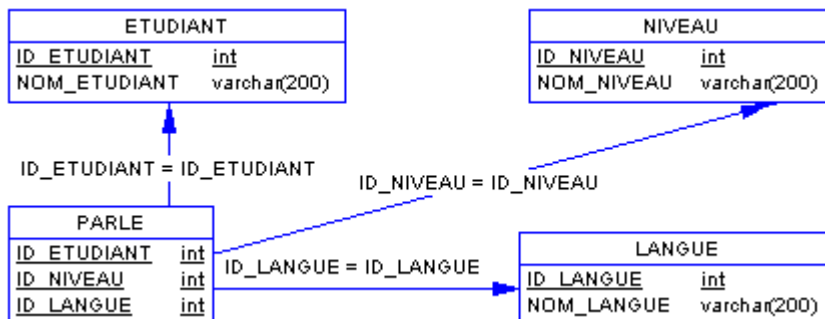
NIVEAU (id\_Niveau, Nom\_Niveau)

LANGUE (id\_Langue, Nom\_Langue)

PARLE (#id\_Etudiant, #id\_Niveau, #id\_Langue)

Chaque enregistrement de la table PARLE a comme clés primaires la concaténation de id\_Etudiant, id\_Niveau et id\_Langue. Chaque enregistrement est donc forcément unique. Ainsi, pour chaque Langue, il est possible de retrouver tous les Etudiants ayant ou non un Niveau précis, de manière unique et distincte.

MPD :





## 5 : Association Réflexive

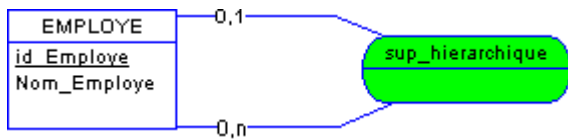
\* Premier cas : cardinalité (X,1) - (X,n), avec X=0 ou X=1

La **Clé Primaire** de l'entité se dédouble et devient une **Clé Etrangère** dans la relation qui devient elle-même une nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,1) - (X,n) (Cf règle 2).

SI :

Prenons l'exemple d'une société organisée de manière pyramidale : chaque employé a 0 ou 1 supérieur hiérarchique direct. Simultanément, chaque employé est le supérieur hiérarchique direct de 0 ou plusieurs employés.

MCD :



MLDR :

EMPLOYE (id\_Employe, Nom\_Employe, #id\_Sup\_Hierarchique)

#id\_Sup\_Hierarchique est l'identifiant (id\_Employe) du supérieur hiérarchique direct de l'employé considéré.

MPD :



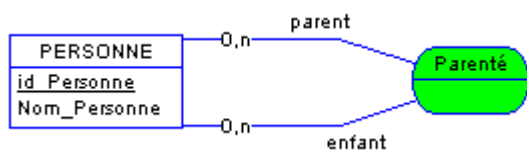
\* Deuxième cas : cardinalité (X,n) - (X,n), avec X=0 ou X=1

De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,n) - (X,n) (Cf règle 3). Il y a donc création d'une nouvelle table.

SI :

Prenons cette fois l'exemple d'une organisation de type familiale : chaque personne a 0 ou n ascendants directs (parents), et a aussi 0 ou n descendants directs (enfants).

MCD :



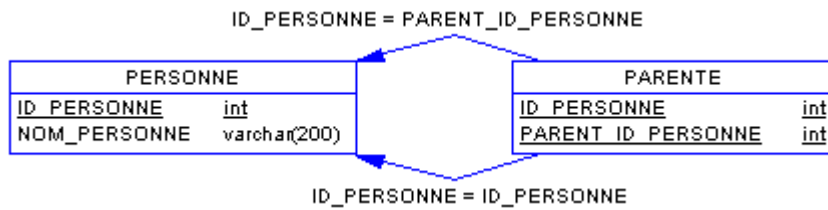
MLDR :

PERSONNE (id\_Personne, Nom\_Personne)

PARENTE (#id\_Parent, #id\_Enfant)

#id\_Parent est l'identifiant (id\_Personne) d'un ascendant direct de la personne. #id\_Enfant est l'identifiant (id\_Personne) d'un descendant direct de la personne. La table PARENTE sera en fait l'ensemble des couples (parents-enfants) présent dans cette famille.

MPD :



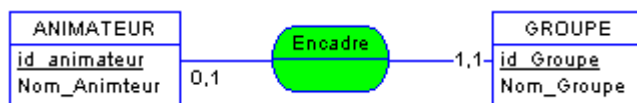
## 6 : Relation binaire aux cardinalités (0,1) - (1,1)

La **Clé Primaire** de la table à la cardinalité (0,1) devient une **Clé Etrangère** de la table à cardinalité (1,1) :

SI :

Dans ce centre de vacances, chaque animateur encadre en solo 0 ou 1 groupe, chaque groupe étant encadré par un et un seul animateur.

MCD :

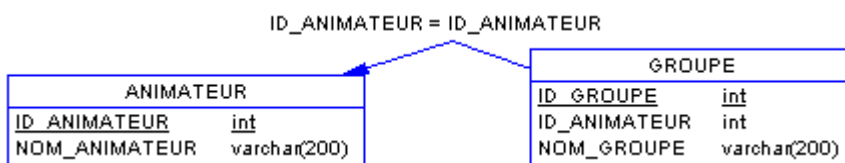


MLDR :

ANIMATEUR (id\_Animateur, Nom\_Animateur)

GROUPE (id\_Groupe, Nom\_Groupe, #id\_animateur)

MPD :



## CONCLUSION :

Ces 6 règles représentent tous les cas existant. Il ne faut surtout pas se laisser impressionner par le nombre de schémas, ni se laisser intimider par le côté inhabituel du processus de modélisation. Il est très simple à acquérir. Après, la majorité du travail restant ne sera plus qu'une question de requêtes SQL, de mise en forme et d'ergonomie, avec une bonne gestion d'Entrée/Sortie de l'information...

En fait, au bout de quelques modélisations et d'un ou deux développements, l'analyste réalise bien souvent que finalement, tout ceci est très logique et d'une évidence rare.

De plus, écrire le MCD, le valider avec le client, puis en déduire le MLDR et donc le Modèle

Physique est un excellent moyen de rentrer complètement dans le chantier. Et surtout, la base de données correspondra exactement au Système d'Information décrit dans le cahier des charges. Les risques d'être hors sujet deviennent minimales, et les temps de développement se trouvent considérablement accélérés.

## **UNE BASE DE DONNEE COHERENTE**

Un tel modèle est dit **cohérent**, c'est à dire que pour chaque donnée fournie, il permet de retrouver toutes les informations s'y rattachant.

Pour chaque client donné, il sera possible d'avoir toutes ses factures, et donc tous les articles qu'il a achetés et en quelle quantité. Pour chaque facture, il sera possible de retrouver le client correspondant, ainsi que la liste des articles et leurs quantités respectives. Enfin, pour chaque article, il sera aisé de retrouver les quantités vendues, à quelle date et à quels clients.

La base ne contient aucune **redondance**, c'est à dire qu'aucune information présente ne peut être déduite d'autres informations présentes. Ceci évite grandement le risque de **corruption**, ou présence de donnée aberrante.

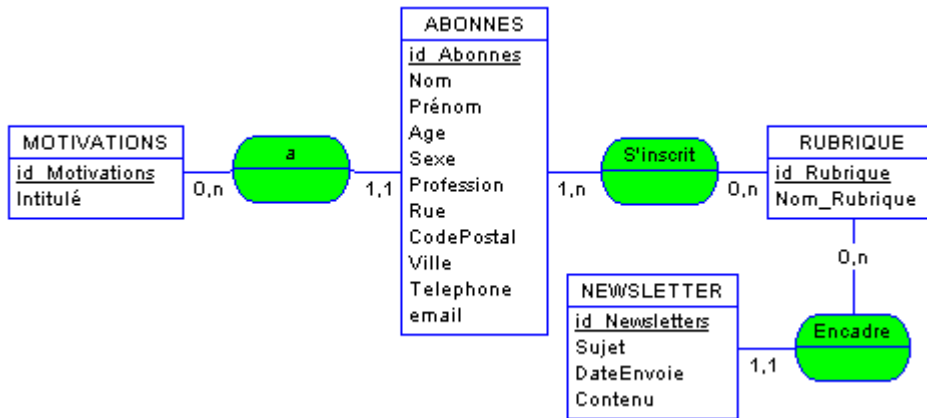
Les prix intermédiaires, la somme totale et autres résultats sont des **traitements**, c'est à dire qu'ils sont calculés par rapport aux informations contenues dans la base. Ainsi, toute erreur sera forcément une erreur de calcul, et non pas une erreur de stockage. Il est plus facile de vérifier un programme d'extraction de donnée que de vérifier la cohérence du contenu d'une base.

## **LES AVANTAGES D'UN TEL RESULTAT**

Le développement se réduit maintenant à une interface d'administration (BackOffice) ou l'utilisateur pourra, par exemple, ajouter/modifier/supprimer ses **ARTICLES**, rentrer ses **FACTURES**, et consulter son fichier **CLIENT**, c'est à dire gérer les données de la base. Le FrontOffice du Site/Logiciel se limitant bien souvent à de l'extraction et de l'affichage de donnée.

## APPLICATION :

Pour terminer, voici la fin de l'analyse de la Newsletter :



Entraîne le MLDR suivant :

MOTIVATIONS (id\_Motivation, Intitule)

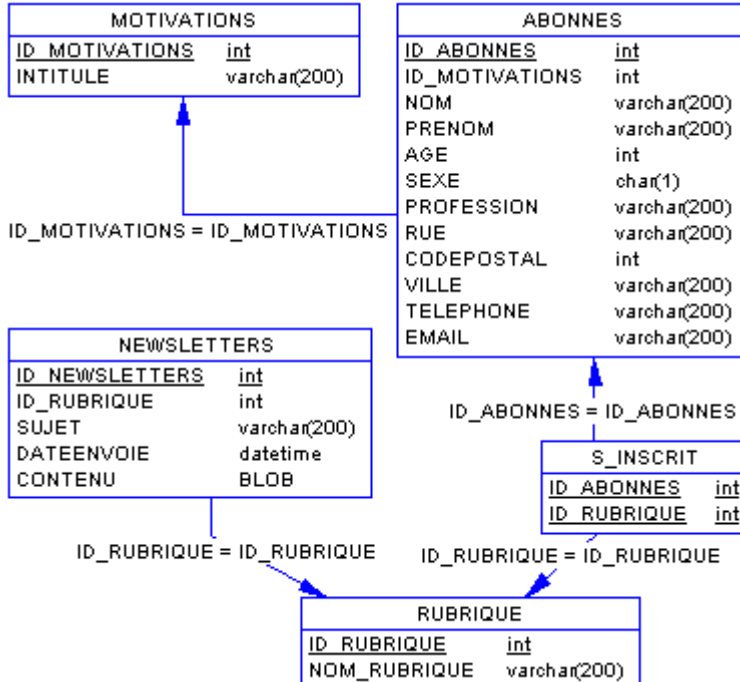
ABONNES (id\_Abonne, #id\_Motivation, Nom, Prenom, Age, Sexe, Profession, Rue, CodePostal, Ville, Telephone, Email)

S\_INSCRIT (id\_Abonne, id\_Rubrique)

RUBRIQUES (id\_Rubrique, Nom\_Rubrique)

NEWSLETTERS (id\_Newsletters, #id\_Rubrique, Sujet, DateEnvoie, Contenu)

Qui nous mène au Modèle Physique de Données (MPD) ou schéma de la Base :



# MEMENTO

**MCD** : Modèle Conceptuel de Donnée. Modélisation abstraite servant à analyser le Système d'Information.

**MLDR** : Modèle Logique De Donnée. Transformation du MCD en modèle indépendant du logiciel de base de donnée choisie.

**MPD** : Modèle Physique De Donnée. Schéma physique du modèle des tables utilisés, prenant en compte le type de stockage des données.

**Clé Primaire** : Combinaison d'une ou de plusieurs valeurs de colonne d'une table qui rendent unique une ligne de donnée de la table. L'attribut qui permet d'identifier de manière unique chaque ligne est appelé la Clé Primaire.

**Clé étrangère** : Combinaison d'une ou de plusieurs valeurs de colonne d'une table qui référence une clé primaire d'une autre table.

**Client / Serveur** : Combinaison d'un ordinateur principal, appelé *serveur*, et d'un ou de plusieurs ordinateurs individuels appelés *clients*, reliés en réseau au *serveur*.

## BIBLIO & WEBOGRAPHIE

### **Ouvrages :**

Conception de bases de données – Ryan K. Stephens & Ronald R. Plew – Campuss Press  
SQL – Frédéric Brouard – Campuss Press

### **Forums & Newsgroup :**

[fr.comp.applications.sgbd](http://fr.comp.applications.sgbd)

### **Sites Web :**

<http://sqlpro.multimania.com> => le site de Frédéric Brouard

<http://dev.nexen.net/docs/mysql/chargement.html> => la doc de MySql en Français.

### **Articles ayants servis à cette rédaction :**

Un peu d'histoire des Bases de Données :

<http://www.informit.fr/article/article.asp?id=667&f=0&idom=24&isdom=0&issdom=0>

Modèle physique de données

<http://www.informit.fr/article/article.asp?id=714&f=0&idom=24&isdom=46&issdom=0>

Conception du modèle conceptuel de données

<http://www.informit.fr/article/article.asp?id=684&f=0&idom=24&isdom=44&issdom=0>

Systemes de Gestion de Base de Donnée

<http://www.informit.fr/article/article.asp?id=832&f=0&idom=21&isdom=51&issdom=0>

*Parus chez PearsonEducation (CampussPress) via le site [www.informit.fr](http://www.informit.fr) courant 2001.*

*Parus précédements chez SAM-MAGASINE ([www.sam-mag.com](http://www.sam-mag.com))*