



<b>1 INTRODUCTION AU MODELE LOGIQUE ET AU MODELE LOGIQUE RELATIONNEL .....</b>	<b>2</b>
<b>2 QUELQUES DEFINITIONS .....</b>	<b>2</b>
<b>3 REGLES DE TRANSFORMATION DU MODELE LOGIQUE EN MODELE RELATIONNEL .....</b>	<b>3</b>
3.1 ENTITE .....	3
3.2 RELATION BINAIRE AUX CARDINALITES $(X,1) - (X,N)$ , $X=0$ OU $X=1$ .....	4
3.3 RELATION BINAIRE AUX CARDINALITES $(X,N) - (X,N)$ , $X=0$ OU $X=1$ .....	4
3.4 RELATION N-AIRE (QUELLES QUE SOIENT LES CARDINALITES). ....	5
3.5 ASSOCIATION REFLEXIVE. ....	7
3.5.1 <i>Premier cas : cardinalité <math>(X,1) - (X,n)</math>, avec <math>X=0</math> ou <math>X=1</math>.</i> .....	7
3.5.2 <i>Deuxième cas : cardinalité <math>(X,n) - (X,n)</math>, avec <math>X=0</math> ou <math>X=1</math>.</i> .....	7
3.6 RELATION BINAIRE AUX CARDINALITES $(0,1) - (1,1)$ . ....	8

## 1 Introduction au modèle logique et au modèle logique relationnel

Après avoir conçu le Modèle Conceptuel de Donnée (MCD), il est maintenant temps de le transposer en Modèle Logique de Données Relationnelles (MLDR). Ce MLDR est en fait le dernier pas vers le Modèle Physique de donnée (MPD), c'est à dire la description de la base qui va être créée. Et là, deux solutions s'ouvrent à vous : soit vous laissez à un programme le soin de transformer votre MCD, soit vous le faites vous-même. Dans les deux cas, il est utile d'avoir un minimum de connaissance théorique sur le sujet. Après avoir défini les notions de clé primaire et de clé étrangère, nous étudierons les 6 règles strictes, nécessaires et suffisantes pour passer d'un MCD à un MLDR.

## 2 Quelques définitions

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 Relationnel. Le MLDR a été inventé par Codd en 1970, et repose sur la Théorie Ensembliste...

### Un peu de vocabulaire :

Les données sont stockées dans des **relations**. Dans la pratique, une **relation** est une **table**.

Une relation est un ensemble de **T-uple** ou **n-uplets** (ligne ou enregistrement)

Un T-uple est défini par un ou plusieurs **attributs** (colonnes de la table).

Un **Domaine** est un ensemble de valeur caractérisé par un nom et qui correspond à un type élémentaire. C'est l'ensemble de valeurs que peut prendre un attribut. Par exemple, on trouve les entiers, les réels, les chaînes de caractères, les dates, etc.

Une information placée dans une colonne d'une table possède forcément un domaine construit sur un type de base.

Le domaine peut être directement identifié par le nom d'un type : entier, réel, chaîne, date, etc. Il peut être défini précisément :

- Nom : chaîne de caractère de longueur < 30 caractères ;
- Année : entier  $\in [1920 \text{ à } 2000]$  ;
- Note : réel  $\in [0.0 \text{ à } 20.0]$  ;
- Cours : {Algo, BD, C, Système}

### Clé primaire - Clé étrangère

Représentons une table COURS et une table SALLE:

Id_cours	Désignation	Id_salle
25	Espagnol débutant	A05
26	Anglais perfectionnement	A20

Id_salle	Etage	Capacité
A05	3	20
A20	1	16

La table COURS est décrite par : COURS (id\_cours, Désignation, #id\_salle)

Chaque enregistrement doit être identifié de manière unique. 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. Ici, il s'agit de l'attribut **id\_cours**.

La table COURS comprend un attribut provenant de la table **SALLE**, l'attribut **id\_salle**. Cet attribut est appelé **Clé Etrangère** et permet de relier les données des 2 tables.

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\_Etrangère)

Dans notre exemple :

SALLE (id\_salle, Etage, Capacité)

COURS (\_, Désignation, #id\_salle)

Ici, **id\_salle** est la **Clé Primaire** de la table SALLE, et est une **Clé Etrangère** dans la table COURS.

Une fois assimilée ces notions de **clés primaires** et de **clés étrangères**, nous pouvons maintenant énoncer les règles de transformation du modèle logique en modèle relationnel

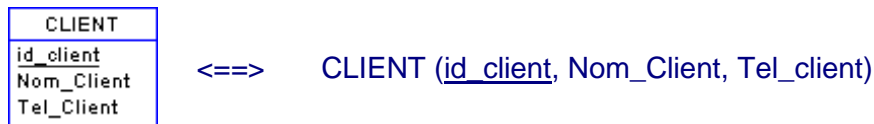
## 3 Règles de transformation du modèle logique en modèle relationnel

### 3.1 Entité

**Règle: 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 donc 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.



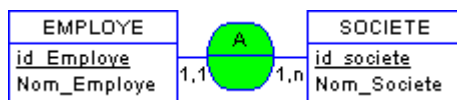
### 3.2 Relation binaire aux cardinalités (X,1) - (X,n), X=0 ou X=1

Règle: La Clé Primaire de la table à la cardinalité (X,n) devient une Clé Etrangère 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) :

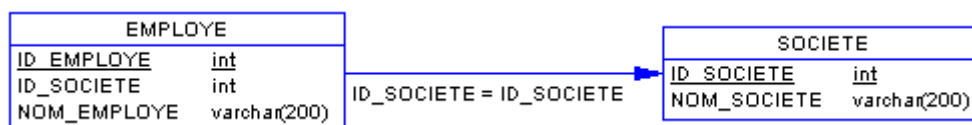


Modèle Logique de Donnée Relationnelle (MLDR) :

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

SOCIETE (id\_Societe, Nom\_Societe)

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



### 3.3 Relation binaire aux cardinalités (X,n) - (X,n), X=0 ou X=1

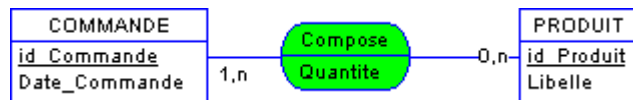
Règle: Il y a création d'une table supplémentaire ayant comme Clé Primaire une clé composée des identifiants des 2 entités. On dit que la Clé Primaire de la nouvelle table est la concaténation des Clés Primaires des deux autres tables.

Si la relation est porteuse de données, celles ci deviennent des attributs pour la nouvelle table.

### Système d'informations :

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 :



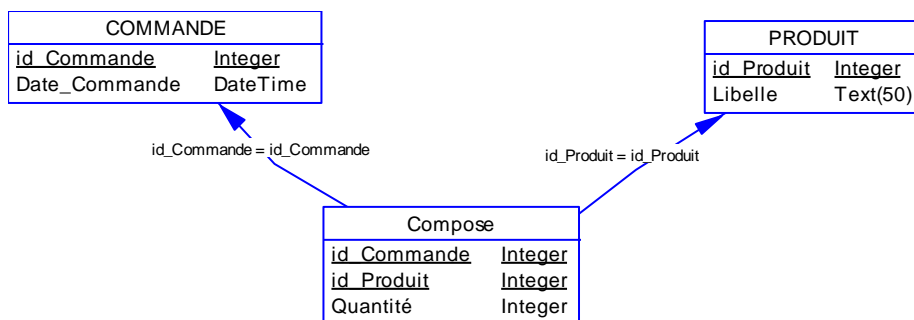
### MLDR :

COMMANDE (id\_Commande, Date\_commande)

PRODUIT (id\_Produit, libelle)

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

### MPD:



### **3.4 Relation n-aire (quelles que soient les cardinalités).**

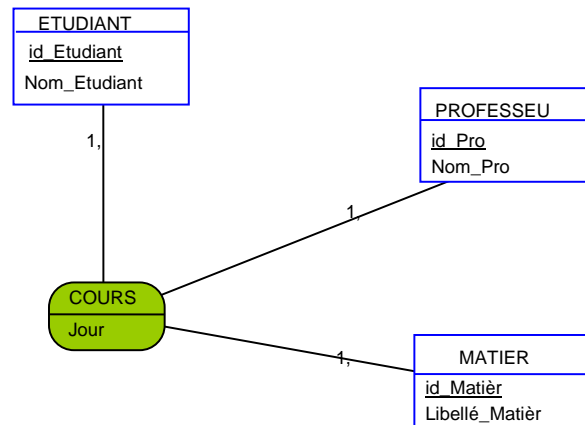
**Règle:** Il y a création d'une table supplémentaire ayant comme Clé Primaire la concaténation des identifiants des entités participant à la relation.

Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.

### Système d'informations :

Un étudiant suit des cours dans plusieurs matières. Chaque matière est donc suivie par 1 ou n étudiants avec un niveau. Une matière peut être enseignée par plusieurs professeurs.

MCD :



MLDR:

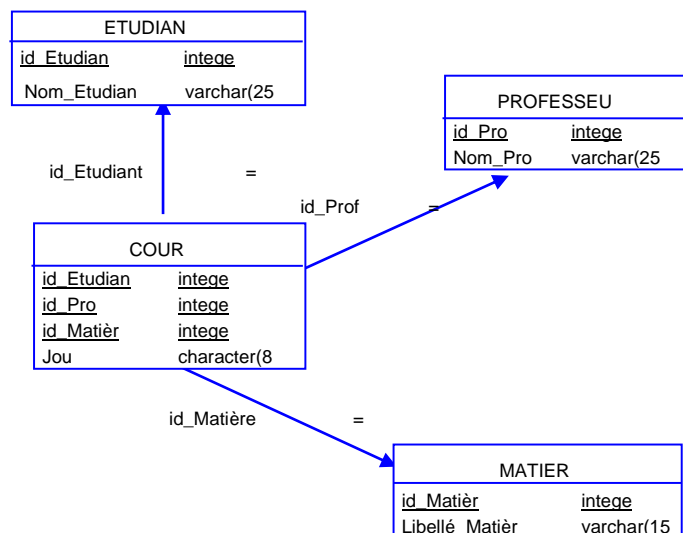
ETUDIANT (id\_Etudiant, Nom\_Etudiant)

PROFESSEUR (id\_Prof, Nom\_Prof)

MATIERE (id\_Matière, Libellé\_Matière)

COURS (#id\_Etudiant, #id\_Prof, #id\_Matière, Jour)

MPD :



### 3.5 Association Réflexive.

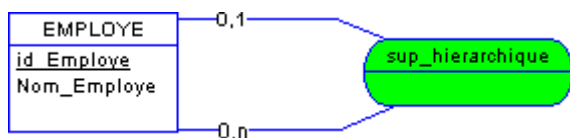
3.5.1 Premier cas : cardinalité (X,1) - (X,n), avec X=0 ou X=1.

**Règle:** La Clé Primaire de l'entité se dédouble et devient une Clé Etrangère dans la relation ou nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,1) - (X,n).

Système d'informations :

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 :

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

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

MPD :



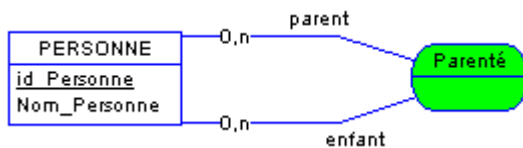
3.5.2 Deuxième cas : cardinalité (X,n) - (X,n), avec X=0 ou X=1.

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

Système d'informations :

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

MCD :



MLDR :

PERSONNE (id\_Personne, Nom\_Personne)

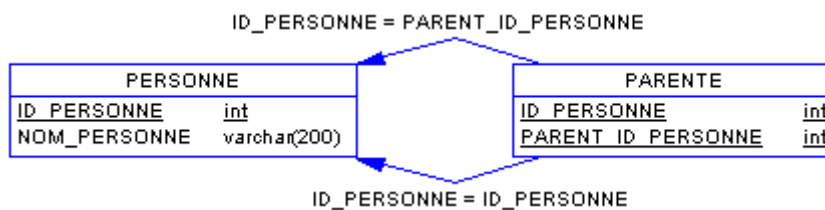
PARENTE (id\_Parent, id\_Enfant)

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 :



### 3.6 Relation binaire aux cardinalités (0,1) - (1,1).

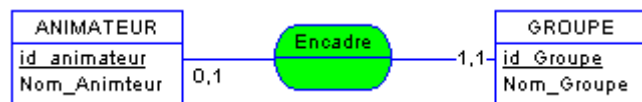
**Règle:** La Clé Primaire de la table à la cardinalité (0,1) devient une Clé Etrangère dans la table à la cardinalité (1,1) .

Système informations. :

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 :

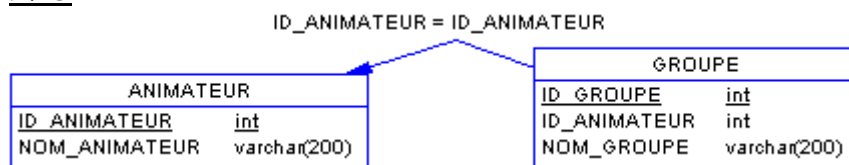


MLDR :

ANIMATEUR (id\_Animateur, Nom\_Animateur)

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

MPD :



## CONCLUSION

Ces 6 règles représentent TOUS les cas que vous pourrez rencontrer. 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. En fait, au bout de quelques modélisations et d'un ou deux développements, vous vous rendrez compte que finalement tout ceci est très logique et d'une évidence rare... Et surtout, surtout, votre base de donnée correspondra EXACTEMENT au système d'information décrit dans le cahier des charges. De plus, écrire le MCD, le valider avec votre client, puis en déduire le MLDR et donc le Modèle Physique vous fera rentrer complètement dans le chantier. Vous irez ensuite beaucoup plus vite, avec très peu de risque d'être hors sujet. Après, la majorité du travail restant ne sera plus qu'une question de requêtes, de mise en forme et d'ergonomie, avec une bonne gestion d'Entrée/Sortie de l'information...