

Système d'Information et Merise

Introduction :

01. L'informatique de Gestion est une discipline du management regroupant l'ensemble des connaissances, des technologies, et des outils assurant la gestion de données et plus généralement l'organisation du système d'information.

L'utilisation de moyens informatiques, électroniques et la télécommunication permettent d'automatiser et de dématérialiser les opérations telles que les procédures de virement et de retrait. Ils sont aujourd'hui largement utilisés en lieu des moyens classiques tels que les formulaires sur papier et le téléphone et cette transformation est à l'origine de la notion de **système d'information**.

02. L'entreprise est un système complexe dans lequel transitent de très nombreux flux d'informations. Sans un dispositif de maîtrise de ces flux, l'entreprise (banque) peut très vite être dépassée et ne plus fonctionner avec une qualité de service satisfaisante. L'enjeu de toute entreprise qu'elle soit de négoce (commerce), industrielle ou de services consiste donc à mettre en place un système destiné à **collecter, mémoriser, traiter et distribuer** l'information (avec un temps de réponse suffisamment bref).

1. Définition :

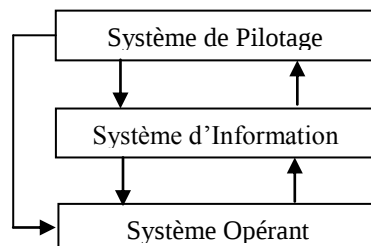
Un système d'Information (SI) représente l'ensemble des éléments participant à la gestion, au traitement, au transport et à la diffusion de l'information au sein de l'organisation.

Le périmètre du terme Système d'Information peut être très différent d'une organisation à une autre et peut recouvrir selon les cas tout ou partie des éléments suivants :

- **Bases de données** de l'entreprise,
- **Progiciel** de gestion intégré (ERP),
- **Outil** de gestion de la relation client (Customer Relationship Management),

Le système d'information est le véhicule de la communication dans l'organisation. Sa structure est constituée de l'ensemble des ressources (les hommes, le matériel, les logiciels) organisées pour : collecter, stocker, traiter et communiquer les informations. Le système d'information coordonne ainsi grâce à l'information les activités de l'organisation et lui permet ainsi d'atteindre ses objectifs.

Ce système d'information assurera le lien entre deux autres systèmes de l'entreprise : le système **opérant** et le système de **pilotage**.



- Le système de pilotage : **décide des actions à conduire** sur le système opérant en fonction des objectifs et des politiques de l'entreprise,
- Le système opérant : **englobe** toutes les fonctions liées à l'activité propre de l'entreprise : facturer les clients, régler les salariés.

2. Système d'information et système informatique

Parmi les informations qui appartiennent au système d'information, certaines doivent ou peuvent faire l'objet d'un traitement automatisé grâce aux outils informatiques. La méthode **Merise** (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise) propose une démarche d'informatisation.

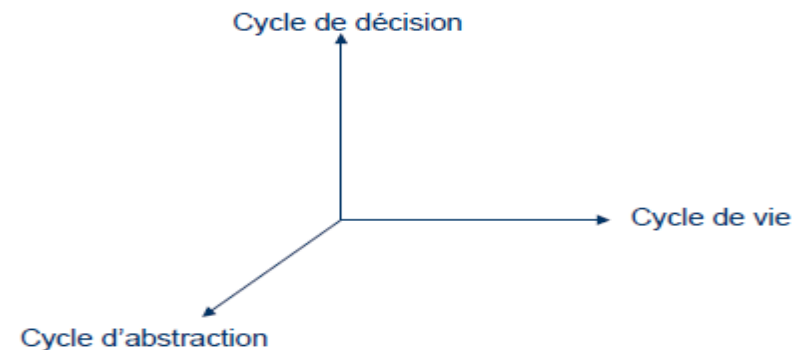
2.1 Définition :

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques, elle vise à recenser la totalité des informations dont un organisme a besoin pour assurer tout ou partie de ses activités fondamentales. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la **séparation des données** et des **traitements** à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité (Longue durée de vie) au modèle.

2.2 Les cycles de Merise :

La méthode Merise pour assurer la démarche de développement d'un système d'information intègre trois axes fondamentaux :

- Cycle de vie.
- Cycle de décision.
- Cycle d'abstraction

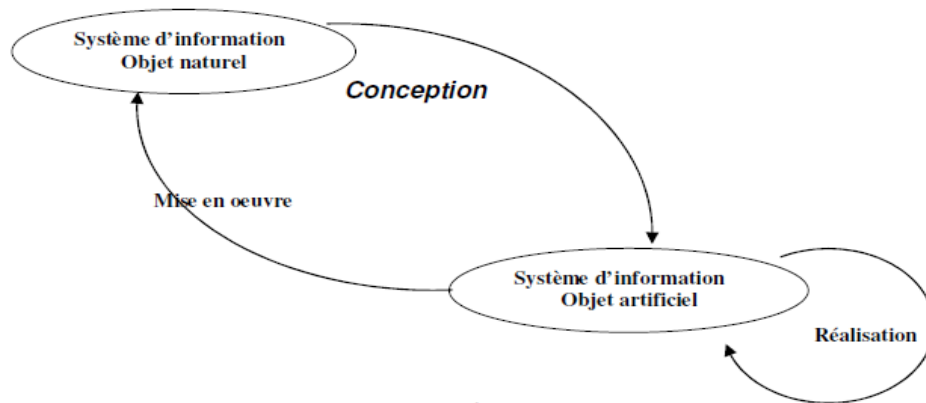


A. Cycle de vie :

Ce cycle se situe à l'échelle du temps au niveau de la vie d'un système de son départ jusqu'à son exploitation c'est le parcours qui va de l'étude de l'objet naturel jusqu'à l'intégration du système artificiel.

La méthode Merise découpe ce cycle en trois parties qui sont :

- **Conception du système** : c'est la description détaillée des spécifications fonctionnelles et techniques du système.
- **Réalisation des programmes** : c'est la production correspondante aux spécifications détaillées.
- **Maintenance du système** : elle tend à adapter le système à l'évolution de son environnement



B. Cycle de décision :

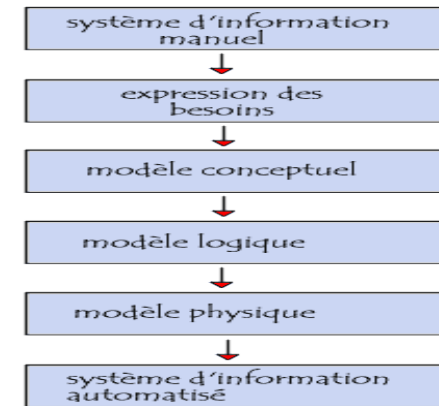
Il représente des choix qui doivent être fait durant le déroulement du cycle de vie. Ce cycle permet à l'organisation d'assurer que le système correspond à ses objectifs. Ces décisions peuvent être de gestion ou organisationnelles et techniques, elles fixent également les priorités de développement, et veillent sur la mise en place des plannings d'avancement de l'organisme.

C. Cycle d'abstraction :

La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance (l'accord) entre données et traitements afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues.

L'expression des besoins est une étape consistant à définir ce que l'on attend du système d'information automatisé, il faut pour cela :

- faire l'inventaire des éléments nécessaires au système d'information
- délimiter le système en s'informant auprès des futurs utilisateurs



Cela va permettre de créer le **MCC** (Modèle conceptuel de la communication) qui définit les flux d'informations à prendre en compte.

L'étape suivante consiste à mettre au point le **MCD** (Modèle conceptuel des données) et le **MCT** (Modèle conceptuel des traitements) décrivant les règles et les contraintes à prendre en compte.

Le modèle organisationnel consiste à définir le **MOT** (Modèle organisationnel des traitements) décrivant les contraintes dues à l'environnement (organisationnel, spatial et temporel).

Le modèle logique représente un choix logiciel pour le système d'information

2.3 Les Niveaux de l'approche :

Il existe 03 niveaux en cette approche :

a). Niveau conceptuel: Définition des finalités de l'entreprise. Détermination de l'ensemble des règles de gestion traduisant les objectifs et les contraintes de l'entreprise. Le SI doit les intégrer. C'est le niveau le plus stable.

b). Niveau organisationnel: Organisation à mettre en place pour atteindre les objectifs visés : postes de travail, chronologie des opérations, nature des traitements.

c). Niveau technique: Moyens techniques nécessaires au projet (matériels/logiciels). Il est soumis à de fréquents changements.

Niveau	Données	Traitements
Conceptuel	Modèle Conceptuel des Données (MCD)	Modèle Conceptuel des Traitements (MCT)
Organisationnel	Modèle Logique des Données (MLD)	Modèle Organisationnel des Traitements (MOT)
Technique (Physique)	Modèle Physique des Données (MPD)	Modèle Opérationnel des traitements (MOPt)

2.3 La démarche d'informatisation :

La démarche d'informatisation comportant les étapes suivantes :

1) **Le schéma directeur** : dont le rôle est de définir, de manière globale, la politique d'organisation et d'automatisation du système d'information. Pour ce faire, il est nécessaire de répertorier l'ensemble des applications informatiques existantes à modifier et à développer. Pour rendre contrôlable et modulable ce développement, il est nécessaire de découper le système d'information en sous-ensembles homogènes et relativement indépendant. Ces sous-ensembles sont appelés domaines.

2) **L'étude préalable par domaine**: qui doit aboutir à une présentation générale du futur système de gestion (modèles des données et des traitements) en indiquant les principales novations (nouvelles) par rapport au système actuel, les moyens matériels à mettre en œuvre, les bilans coût – avantage. Cette étude est réalisée en 4 phases :

- ✓ une **phase de recueil** qui a pour objectif d'analyser l'existant afin de cerner les dysfonctionnements et les obsolescences les plus frappantes du système actuel.
- ✓ une **phase de conception** qui a pour objectif de formaliser et hiérarchiser les orientations nouvelles en fonction des critiques formulées sur le système actuel et d'autre part des politiques et des objectifs de la direction générale. Cela revient à modéliser le futur système avec une vue pertinente de l'ensemble.
- ✓ une **phase d'organisation** dont l'objectif est de définir le système futur au niveau organisationnel: qui fait quoi ?
- ✓ une **phase d'appréciation** dont le rôle est d'établir les coûts et les délais des solutions définies ainsi que d'organiser la mise en œuvre de la réalisation. A cet effet un découpage en projets est effectué.

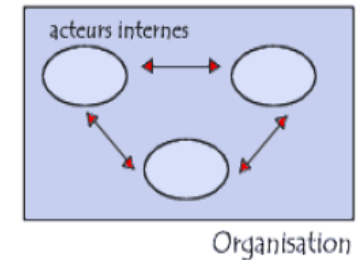
3) **L'étude détaillée par projet** qui consiste d'une part à affiner les solutions conçues lors de l'étude préalable et d'autre part à rédiger, pour chaque procédure à mettre en œuvre, un dossier de spécifications détaillé décrivant les supports (maquettes d'états ou d'écran) ainsi que les algorithmes associés aux règles de gestion... A l'issue de cette étude, il est possible de définir le cahier des charges utilisateurs qui constitue la base de

l'engagement que prend le concepteur vis à vis des utilisateurs. Le fonctionnement détaillé du futur système, du point de vue de l'utilisateur, y est entièrement spécifié.

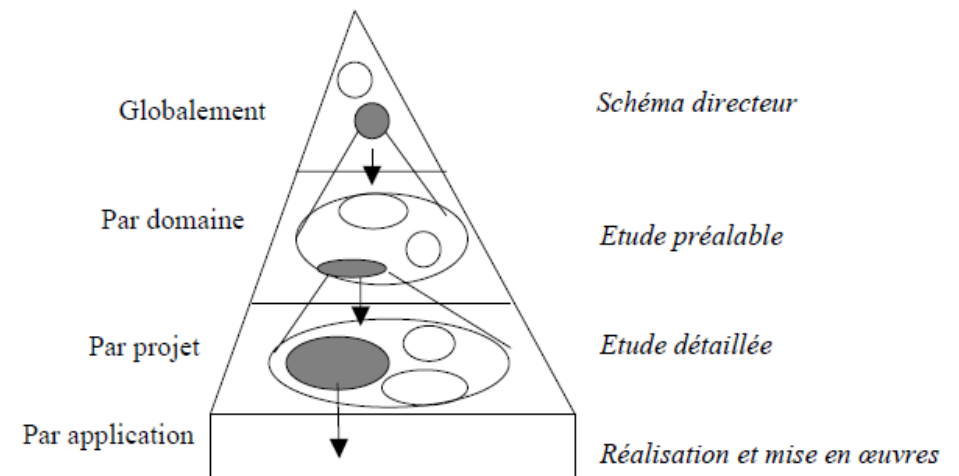
4) **La réalisation** dont l'objectif est l'obtention des programmes fonctionnant sur un jeu d'essais approuvés par les utilisateurs.

5) **La mise en œuvre** qui se traduit par un changement de responsabilité : l'équipe de réalisation va en effet transférer la responsabilité du produit à l'utilisateur. Cette étape intègre en particulier la formation des utilisateurs. Après une période d'exploitation de quelques mois, la recette définitive de l'application est prononcée.

- **La maintenance** qui consiste à faire évoluer les applications en fonction des besoins des utilisateurs, de l'environnement et des progrès technologiques.



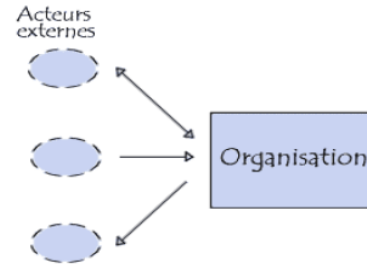
Le schéma suivant reprend les étapes décrites ci-dessus :



2.4 Modèle conceptuel de la communication (MCC) :

1) Définition de l'organisation

La **première** étape de ce modèle est d'arriver à isoler le système en le délimitant. Il s'agit donc de définir le système et les éléments externes avec lesquels il échange des flux d'information. Ces éléments extérieurs sont appelés **acteurs externes** (ou partenaires).



La **seconde** étape consiste à découper l'organisation en entités appelées *acteurs internes* (ou domaines).

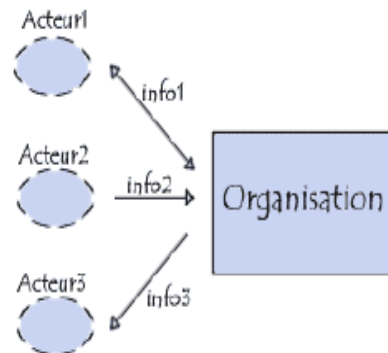
Lorsque les domaines d'une organisation sont trop importants, ils peuvent être décomposés eux-mêmes en *sous-domaines*

La **dernière** étape est l'analyse des flux d'information, c'est-à-dire la définition des processus (enchaînement de faits).

2) Diagramme de contexte :

Le diagramme de contexte a pour but de représenter les flux d'informations entre l'organisation et les acteurs externes selon une représentation standard dans laquelle chaque objet porte un nom :

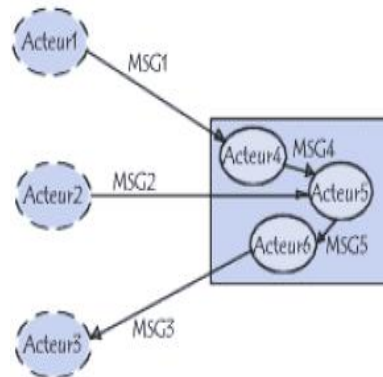
- l'organisation est représentée par un **rectangle**
- les acteurs externes sont représentés par des **ellipses en pointillés**
- les flux d'information sont représentés par des flèches dont l'orientation désigne le sens du flux d'information



3) Diagramme conceptuel de flux :

Ce diagramme (appelé aussi **modèle conceptuel de la communication**) permet de compléter le diagramme de contexte en décomposant l'organisation en une série d'acteurs internes. Dans ce diagramme la représentation standard est la suivante :

- Les acteurs internes sont représentés par des **ellipses**,
- Les messages internes sont représentés par des **flèches**



Modèle Conceptuel de Données (MCD)

1. Définition

Le modèle conceptuel des données est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique. Il a pour but d'écrire de les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données, facilement compréhensible. Cet aspect recouvre les mots qui décrivent le système ainsi que les liens existants entre ces mots. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « entité-association ».

2. Concepts de base :

2.1. La propriété (ou attribut ou rubrique) :

La propriété est une information **élémentaire**, c'est-à-dire non déductible d'autres informations, qui présente un intérêt pour le domaine étudié. Par **exemple**, si l'on considère le domaine de gestion des comptes d'une banque les données : « N° Client », « Nom », « Prénom », « Adresse » sont des propriétés pertinentes pour ce domaine (Client).

- Chaque valeur prise par une propriété est appelée occurrence.
- Une propriété est dite simple ou encore atomique si chacune des valeurs qu'elle regroupe n'est pas décomposable. La propriété « Adresse », dont des exemples d'occurrences sont donnés ci-dessous, n'est pas élémentaire car elle peut être décomposée en trois propriétés : la rue, le code postal et la ville.

Adresse
08, rue 01 Novembre 1954, 24000 Guelma
15, Cité Hassiba Benbouali , 16000 Alger

- Le principe de non-redondance impose que chaque propriété, correctement identifiée, n'apparaisse qu'une seule fois dans le modèle.

2.2. L'entité ou individu-type :

Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire, comme on peut dire que c'est un regroupement bien pensé, donc sensé, de plusieurs propriétés.

On appelle **classe d'entité** un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même. Le classement des entités au sein d'une classe s'appelle **classification** (ou *abstraction*). Une entité est une **instanciation** de la classe.

Les classes d'entités sont représentées par un rectangle. Ce rectangle est séparé en deux champs:



- le champ du haut contient le **libellé** (ensemble des termes). Ce libellé est généralement une abréviation pour une raison de simplification de l'écriture. Il s'agit par contre de vérifier qu'à chaque classe d'entité correspond un et un seul libellé, et réciproquement.
- le champ du bas contient la **liste des propriétés** de la classe d'entité

Exemple, on considère l'entité **client** qui regroupe les propriétés : N°Client, Nom, Prénom, Adresse.

Client
- N° Client
- Nom
- Prénom
- Adresse

Entité : Client

2.3. Les identifiants :

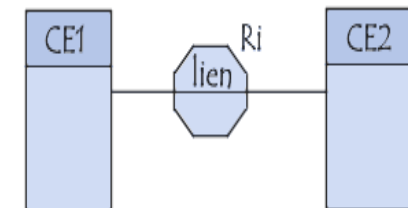
Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité. La définition originale est la suivante: **L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur.**

Les attributs d'une classe d'entité permettant de désigner de façon unique chaque instance de cette entité sont appelé identifiant absolu. Le modèle conceptuel des données propose de souligner les identifiants (parfois de les faire précéder d'un #).

2.4. L'association (ou relation-type) :

a) Définition :

Une association (appelée aussi parfois *relation*) est un lien sémantique entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité).



Une classe de relation peut lier plus de deux classes d'entité. Voici les dénominations des classes de relation selon le nombre d'intervenants:

- une classe de relation **récursive** (ou *réflexive*) relie la même classe d'entité
- une classe de relation **binaire** relie deux classes d'entité
- une classe de relation **ternaire** relie trois classes d'entité

Une classe de relation **n-aire** relie n classes d'entité Les classes de relations sont représentées par des hexagones (parfois des ellipses) dont l'intitulé décrit le type de

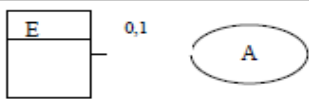
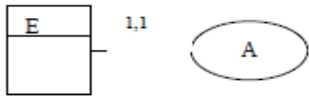
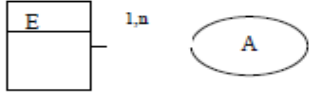
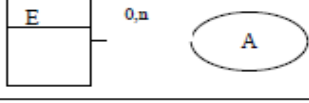
relation qui relie les classes d'entité (généralement un verbe). On définit pour chaque classe de relation un identificateur de la forme Ri permettant de désigner de façon unique la classe de relation à laquelle il est associé. On peut éventuellement ajouter des propriétés aux classes de relation.

b) Cardinalité

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composé d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur:

- **la borne minimale** (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- **la borne maximale** (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation.

Un couple de cardinalités placé entre une entité E et une association A représente le nombre minimal et maximal d'occurrences de l'association A. Le tableau ci-après récapitule les valeurs que peut prendre ce couple.

	Pour chaque occurrence de E, le modèle admet : - soit l'absence de lien - soit la présence d'un seul lien
	Pour chaque occurrence de E le modèle admet la présence d'un et un seul lien
	Pour chaque occurrence de E le modèle admet la présence d'un seul ou de plusieurs liens
	Pour chaque occurrence de E le modèle admet : - soit l'absence de lien - soit la présence de plusieurs liens

3. Le dictionnaire de données

Un dictionnaire de données est un tableau qui regroupe toutes les données du SI. C'est la liste précise de chacune des données manipulées, représentée par une mnémonique (un identificateur) et une définition précise de la donnée reconnue au sein de l'organisation.

La constitution du dictionnaire de données consiste à établir la liste exhaustive des données élémentaires à introduire dans la base de données.

N°	Mnémonique (Nom de rubrique)	Signification	Type et Taille	Nature	Remarque
Numéro d'ordre	Nom attribué	Ce qui signifie la rubrique	Type et taille	Elémentaire, Calculée, non calculée, incrémentale, etc.	Contrainte d'intégrité/ formule de calcul / Règle de gestion, format spécifique (ex, date « jj/mm/aa »)

Le dictionnaire de donnée est un tableau qui regroupe toutes les données du SI, pour chaque donnée identifiée il faut préciser :

- Une **mnémonique** (un nom de rubrique) : NumClient, NumFact, NomClient, ...
- Sa **signification** : numéro client, numéro de facture, nom de client,
- Son **type** et sa **taille** (numérique (entier, réel, date) ou alphanumérique)

N pour numérique. Exemple : N(6) numérique longueur 6, N(6,2) réel de partie entière de longueur 6 et partie fractionnaire de 2.
A pour Alphabétique,
AN pour Alphanumérique,
D pour date avec spécification du format "jj/mm/AA".
B pour booléen (logique)

- Sa **nature** : donnée calculée (**C**) ou non calculée (**NC**), (**I**) Incrémentale.
- Des remarques :
 - ✓ Des contraintes d'intégrité (**C.I**) : exemples : Quantité > 0, Prix unitaire > 0, 0 <= note matière <= 20
 - ✓ Règle de calcul (**R.C**): Montant = Somme (note * coefficient)/Somme(Coefficient)
 - ✓ Règle de gestion (**R.G**).

4. Les Règles de Gestion

Une règle de gestion est une contrainte qui s'applique à une action, à une activité ou encore à un processus de l'entreprise.

Exemple de règles de gestion d'un système d'information (SI) des ventes d'un grossiste de produits de consommation ménagère :

- Pour chaque **Client**, on doit mémoriser son identification, son adresse, son téléphone, adresse e-mail et son numéro de registre de commerce.
- Pour chaque **Produit**, on doit connaître sa désignation, sa date de production et sa date péremption.
- Pour une **Commande**, on doit connaître son numéro, sa date et le client ayant passé la commande.
- Une **Commande** doit contenir **un** ou **plusieurs produits**. Pour chaque produit de la commande on doit connaître la quantité commandée.
- Un **client** peut passer **une ou plusieurs commandes**. Une Commande est passée par **un seul Client**.
- Pour une **Facture**, on doit connaître, son numéro, sa date et la commande qui lui correspond.
- Une **Facture** doit contenir **un** ou **plusieurs produits**. Pour chaque produit de la facture on doit connaître la quantité facturée, son prix unitaire.
- A une **Commande** peut correspondre **une ou plusieurs Factures**. Une Facture correspond à une **un seul** Commande.
- Le même **Taux de Tva** s'applique sur les produits d'une même facture.

Ces règles sont parfois données par l'utilisateur mais, généralement, c'est le concepteur qui doit les établir lui-même.

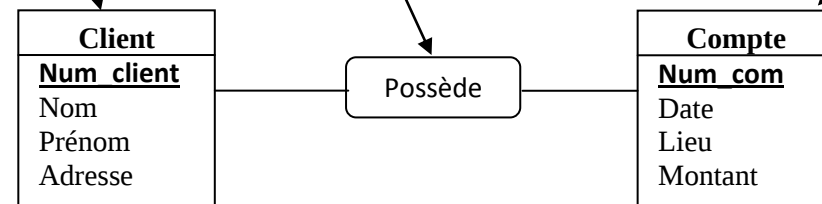
3. La modélisation directe :

Elle consiste à identifier, à partir d'une description exprimée en langage naturel, les entités et les associations en appliquant les règles suivantes :

- les **noms** deviennent des **entités**
- les **verbes** deviennent des **associations**

L'exemple suivant qui illustre ce propos est bien trop simple pour que cette méthode conduise à des résultats satisfaisants sur un système d'information de taille plus importante

- Le client d'une banque possède plusieurs comptes bancaires. Les noms sont : « **Compte** », « **Client** ». Le verbe est : « **Possède** », ce qui donne la modélisation suivante



Remarque :

Le modèle obtenu par cette méthode est **très loin de la représentation optimale** et il sera nécessaire d'appliquer une phase de validation et de normalisation (élimination des situations qui induisent des redondances) pour aboutir à une solution satisfaisante.

Modèle Logique de données

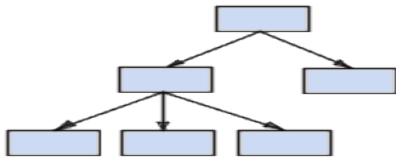
1. Qu'est ce qu'un SGBD :

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le Système de gestion de base de données (**SGBD**, en anglais *DBMS*).

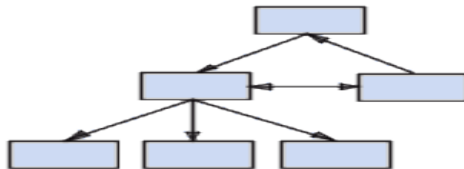
Un SGBD est un ensemble qui sert à la manipulation des bases de données. Il sert à effectuer des opérations ordinaires telles que consulter, modifier, construire, organiser, transformer, copier, sauvegarder ou restaurer des bases de données. Il est souvent utilisé par d'autres logiciels ainsi que les administrateurs ou les développeurs.

Il existe plusieurs modèles de SGBD, différenciés selon la représentation des données qu'elle contient :

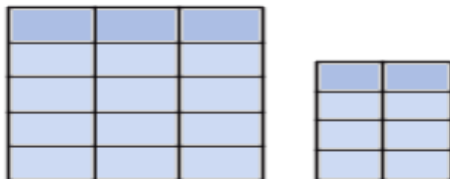
- **le modèle hiérarchique** : les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD.



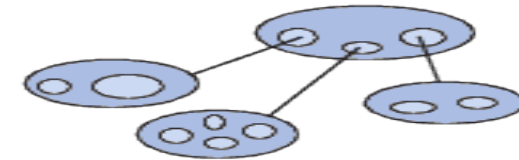
- **le modèle réseau** : comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant.



- **le modèle relationnel** (SGBDR, *Système de gestion de bases de données relationnelles*) : les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes).



- **le modèle objet** (SGBDO, *Système de gestion de bases de données objet*) : les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées *classes* présentant des données membres. Les champs sont des instances de ces classes.



Remarque :

A la fin des années 90 les bases relationnelles sont les bases de données les plus répandues (environ trois quarts des bases de données).

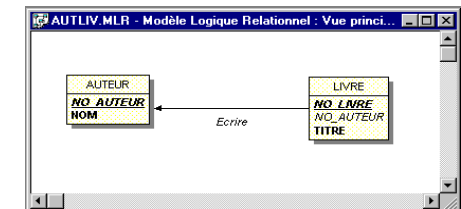
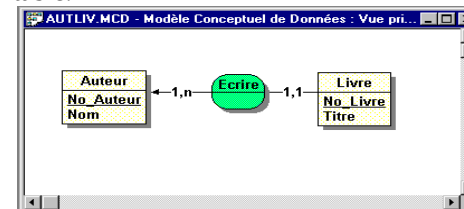
2. Modèle Logique de données :

2.1 Définition :

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements

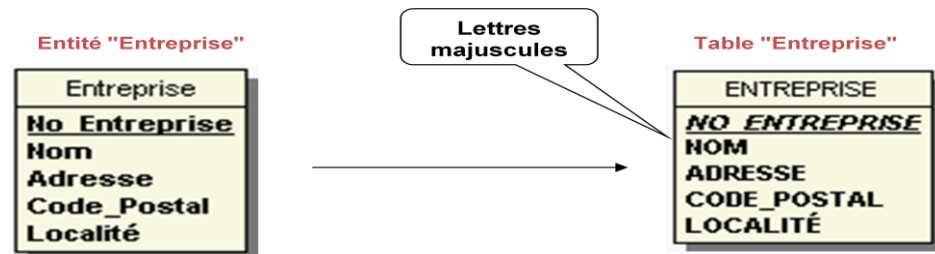
2.2 Règles de transformation du MCD au MLD

La transcription d'un MCD en **modèle relationnel** s'effectue selon quelques règles simples qui consistent d'abord à transformer toute entité en table, avec l'identifiant comme clé primaire, puis à observer les valeurs prises par les cardinalités maximum de chaque association pour représenter celle-ci soit (ex : card. max 1-n ou 0-n) par l'ajout d'une clé étrangère dans une table existante, soit (ex : card. max n-n) par la création d'une nouvelle table.



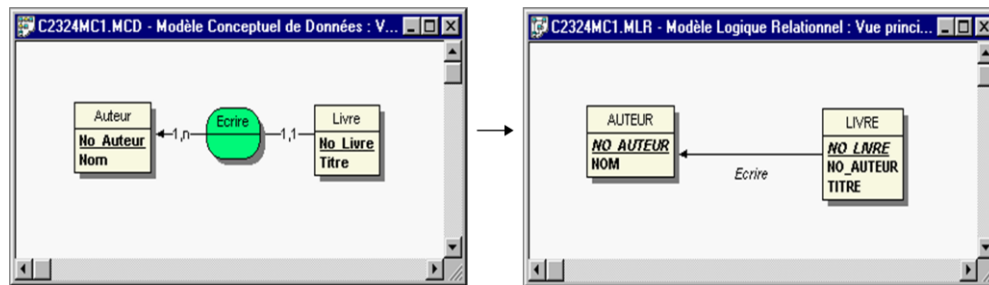
A. Transformation des entités :

Toute entité est transformée en table. Les propriétés de l'entité deviennent les attributs de la table. L'identifiant de l'entité devient la clé primaire de la table.



B. Transformation des relations binaires du type $(x,n) - (x,1)$

Afin de représenter la relation, on duplique la clé primaire de la table basée sur l'entité à cardinalité (x,n) dans la table basée sur l'entité à cardinalité $(x,1)$. Cet attribut est appelé clé étrangère. Les deux tables sont liées par une flèche nommée selon la relation, qui pointe de la table à clé étrangère vers la table qui contient la clé primaire correspondante.



L'attribut *No_Auteur* qui est clé primaire de la table *Auteur*, devient clé étrangère dans la table *Livre*.

C. Transformation des relations binaires du type $(x,1) - (x,1)$

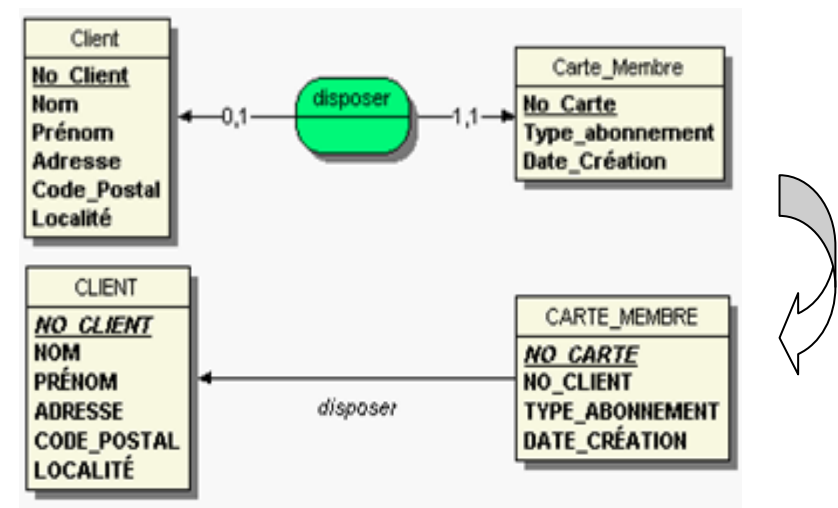
Nous devons distinguer plusieurs cas. Sachant qu'une relation binaire du type $(1,1)-(1,1)$ ne doit pas exister il nous reste les 2 cas suivants:

Relation binaire $(0,1)-(1,1)$

Relation binaire $(0,1)-(0,1)$

✓ **Relation binaire $(0,1)-(1,1)$:**

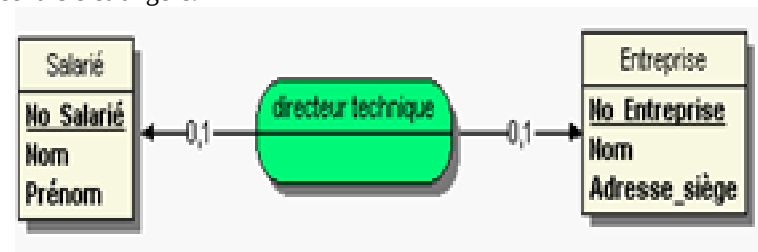
On duplique la clé de la table basée sur l'entité à cardinalité $(0,1)$ dans la table basée sur l'entité à cardinalité $(1,1)$.

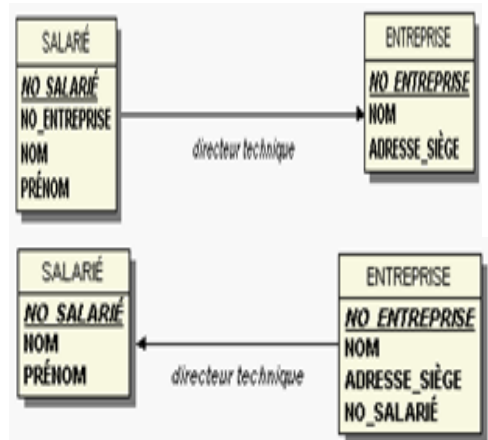


Le *No_Client*, qui est clé primaire de la table *Client*, devient clé étrangère dans la table *Carte_Membre*.

✓ **Relation binaire $(0,1)-(0,1)$**

On duplique la clé d'une des tables dans l'autre. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent également attributs de la table dans laquelle a été ajoutée la clé étrangère.

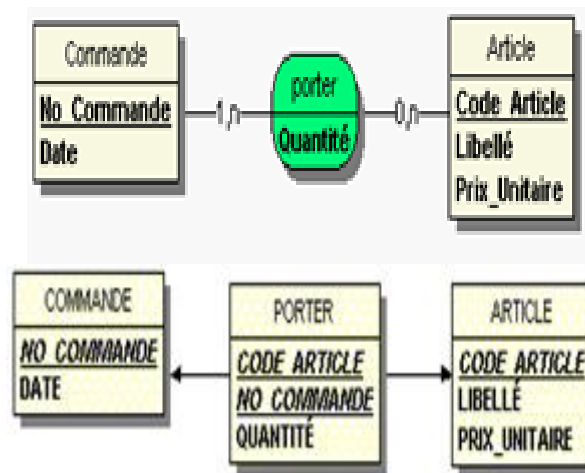




Soit on migre la clé primaire de la table *Entreprise* dans la table *Salarié*, soit on fait l'inverse

D. Transformation des relations binaires du type $(x,n) - (x,n)$

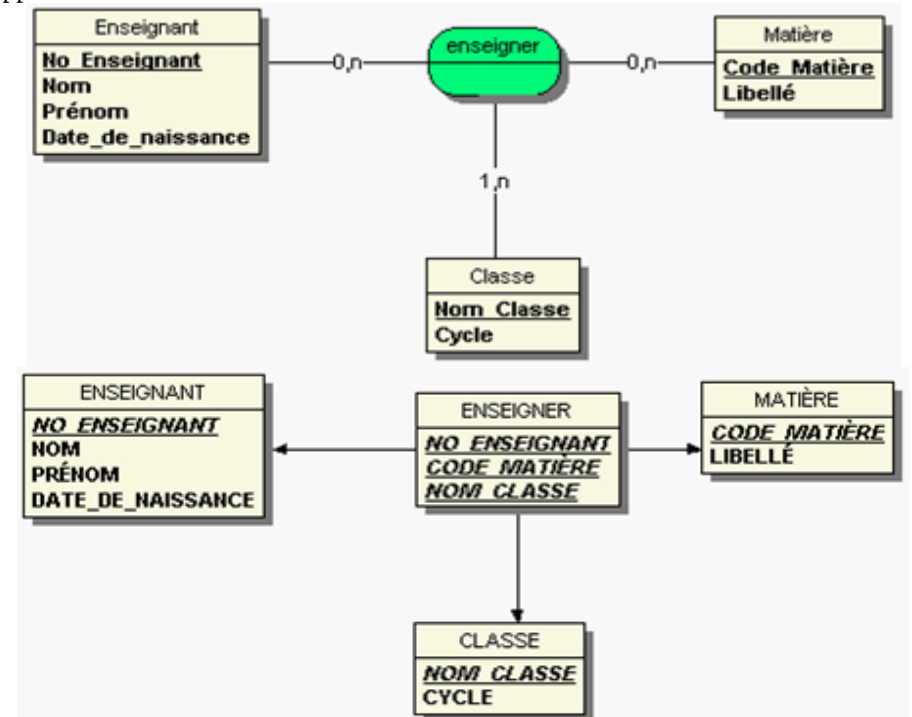
On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires des 2 tables. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.



On crée une table *Porter*, qui contient comme clé primaire une clé composée de *No-Commande* et *Code_Article*. Elle contient également la propriété *Quantité* issue de la relation *Porter*

E. Transformation des relations ternaires

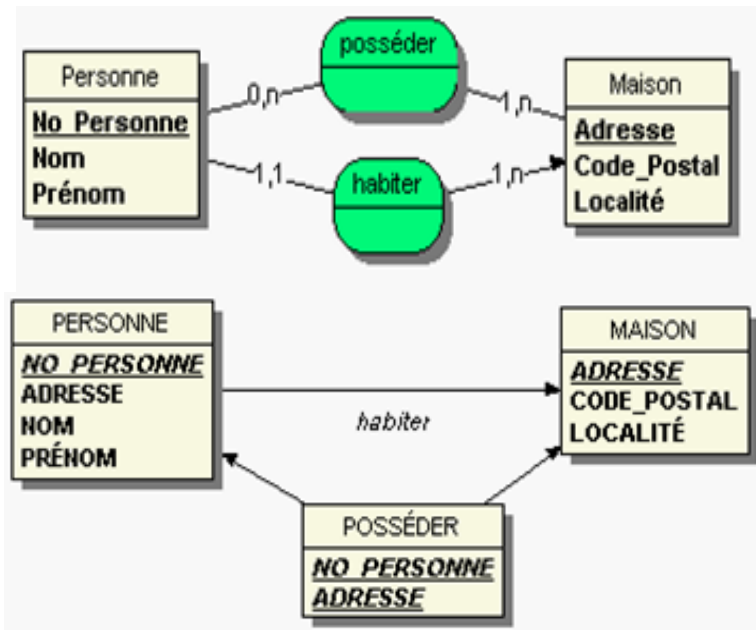
On crée une table supplémentaire ayant comme clé primaire une clé composée des clés primaires de toutes les tables reliées. Cette règle s'applique de façon indépendante des différentes cardinalités. Lorsque la relation contient elle-même des propriétés, celles-ci deviennent attributs de la table supplémentaire. Une propriété de la relation qui est soulignée devra appartenir à la clé primaire composée de la table supplémentaire.



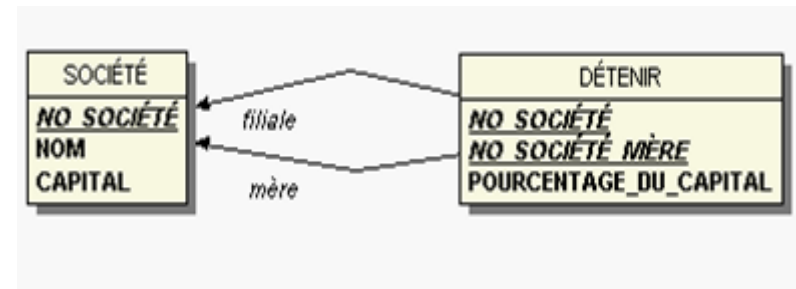
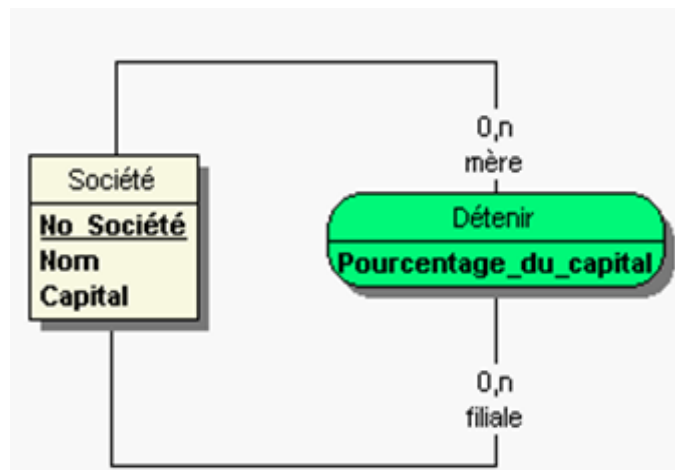
La table *Enseigner* contient une clé composée de *No_Enseignant*, *Code_Matière* et *Nom_Classe*.

F. Transformation de plusieurs relations entre 2 entités

Les règles générales s'appliquent



G. Transformation des relations réflexives :



Nous appliquons les règles générales avec la seule différence que la relation est 2 fois reliée à la même entité

3. Algèbre Relationnelle :

3.1 Définition :

Ensemble d'opérations permettant de manipuler des relations (schéma relationnel) et produisant comme résultat de nouvelles relations. Son principe repose sur la création de nouvelles tables (tables résultantes).

3.2 Les requête :

Une requête est destinée à obtenir des informations précises et ordonnées sur la base de données. On peut également par le biais des requêtes :

- Effectuer des calculs.
- Obtenir des statistiques.
- Modifier les tables.
- Créer de nouvelles tables.
- Gérer des utilisateurs...

Trois façons de créer des requêtes :

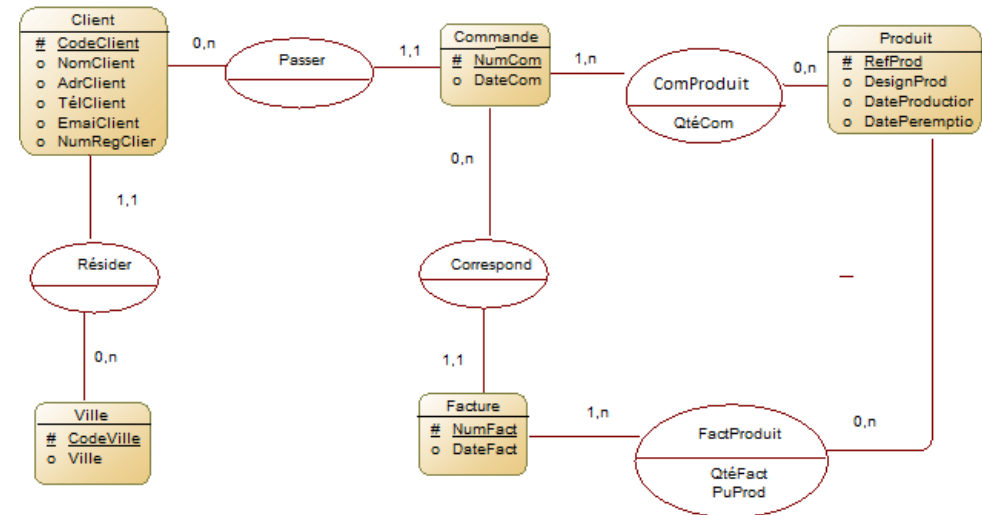
- L'algèbre relationnelle qui permet de préparer une requête en utilisant un langage algébrique.
- Le QBE (Query By Exemple), qui est un mode graphique où l'on clique sur les éléments choisis pour construire la requête. C'est un moyen interactif et puissant utilisé par Access.
- Le SQL (Structured Query Language) qui est le langage des requêtes, qui est plus puissant que le QBE et plus universel.

Exemple 2 :

1. Dictionnaire de données épuré

N°	Rubrique	Désignation	Type	Nature	Observation
1	NumClient	Numéro client	N(5)	E, Inc	
2	NomClient	Nom client	A(50)	E	
3	AdrClient	Adresse client	A(100)	E	
4	CodeVille		N(3)	E, Inc	
5	NomVille	Ville de l'adresse	A(50)	E	
6	TélClient	Tél client	N(10)	E	Format: ##### ## ## ##
7	EmailClient	Email client	AN(50)	E	Format: aaaa@aaaa
8	NumRegClient	N° de Registre	AN(20)	E	
9	NumCom	Numéro de commande	N(5)	E, Inc	
10	DateCom	Date de la commande	D (10)	E	JJ/MM/AAAA
11	RefProd	Référence produit	AN(4)	E	
12	DésignProd	Désignation produit	AN(20)	E	
13	DatePropduct	Date de production	D (10)	E	JJ/MM/AAAA
14	DatePerempt	Date de péremption	D (10)	E	JJ/MM/AAAA
15	QtéCom	Quantité de la commande	N(6,2)	E	CI: >0
16	NumFact	Numéro facture	N(5)	E, Inc	
17	DateFact	Date facture	D (10)	E	JJ/MM/AAAA
18	PUProd	Prix unitaire produit	AN	E	CI: >0
19	QtéFact	Quantité de la facture	N(6,2)	E	CI: >0
20	TauxTVA	Montant produit	N(2,2)	E	

2. le MCD obtenu :



3. Le MLD

Le modèle logique de données du SI Ventres :

Client (**NumClient**, NomClient, AdrClient, CodeVille, TélClient, EmailClient, NumRegClient, #CodeVille)

Ville (CodeVille, NomVille)

Commande (NumCom, DateCom, #NumClient)

Produit (RefProd, DésignProd, DateProduct, DatePerempt)

Facture (NumFact, DateFact, #NumCom)

ComComporter (NumCom, RefProd, QtéCom)

FactComporter (NumFact, RefProd , PUProd, QtéFact)