

# CHAPITRE 2

## **Objectif global**

Ce cours a pour objectif principal d'initier les étudiants à la conception des applications informatiques de façon systématique (méthodique) et reproductible (rééritable) ; en les incitant à rechercher et établir les fonctionnalités d'une application, et à les modéliser sous forme de modèles. Pour permettre à l'étudiant une immersion immédiate, une étude de cas détaillée et guidée sera mise en exergue afin de faciliter l'assimilation de cette méthode.

## **Objectifs Spécifiques**

Dans la partie sur la méthode Merise, vous découvrirez comment :

- ✓ Participer à la conception et à l'évolution des systèmes d'information dans une organisation ;
- ✓ Représenter schématiquement les flux d'information, les données et les traitements d'un système d'informatisation avec les extensions Merise/2 ;
- ✓ Comparer certains modèles Merise à certains diagrammes UML.

# METHODE D'ANALYSE ET DE CONCEPTION DES SI : MERISE

## Introduction

Les méthodes d'analyse et de conception fournissent une méthodologie et des notations standards qui aident à concevoir des systèmes d'information de qualité. Il existe différentes manières pour classer ces méthodes : **les méthodes fonctionnelles**<sup>1</sup>(*cartésiennes et systémique*) et **l'approche orientée objet**<sup>2</sup>. Dans le cadre de notre cours, nous nous focaliserons sur une méthode classique et systémique à savoir la **Méthode MERISE**.

## I. Présentation générale de la méthode Merise

Pour modéliser un système d'information, différents métiers (*utilisateurs, experts, organisateurs, informaticiens, ...*) interviennent ensemble dans un processus de développement, constitué d'activités distinctes qui sont exercées dans un environnement organisationnel et basé sur différents outils et techniques. Leur compréhension passe par la maîtrise de certaines notions clés.

### 1.1 Notions : Méthode, Modèle, Langage de modélisation et diagramme

Dans le petit Robert, **une méthode** peut-être "*un ensemble de démarches*

---

<sup>1</sup> **Les méthodes fonctionnelles** mettent en évidence les fonctions à assurer et proposent une approche hiérarchique descendante et modulaire.

<sup>2</sup> **L'approche objet** considère un système orienté objet (une entreprise, société, logiciel ou système informatique) comme une collection d'objets qui coopèrent.

*raisonnées, suivies pour parvenir à un but*". Dans le contexte de conception des systèmes d'information elle est définie comme un ensemble composé d'un langage, présenté sous forme d'un ensemble de schémas ou figures associés, ainsi que des préconisations sur la façon d'utiliser ces modèles. Le concept de modèle est un concept essentiel et fondateur du monde des Systèmes d'Information (SI). **Un modèle** est une représentation abstraite et simplifiée de tout ou partie d'un SI existant ou futur, mettant en évidence certains aspects essentiels.

Les modèles permettent d'analyser l'organisation et de découvrir ses besoins, d'assister dans la conception et le développement de produit, de maîtriser la complexité d'un système d'entreprise et enfin, d'envisager l'évolution des SI.

Un **langage de modélisation** est un ensemble de règles et de concepts qui permettant de construire des modèles. Un **diagramme** est la représentation graphique d'un modèle.

La modélisation comporte deux étapes : **l'analyse** et **la conception**. Dans chacune de ces deux parties, les concepteurs pourront être assistés éventuellement d'outils logiciels (AGL) adaptés, facilitant la conception et la documentation.

## 1.2 Historique de la méthode Merise

**MERISE (Méthode d'Etude et de Réalisation Informatique des Systèmes d'Entreprise)** est une méthode systémique d'analyse et de conception de système d'information. Proposée en 1978, elle préconise un développement logiciel basé sur le modèle en cascade<sup>3</sup>. L'utilisateur intervient pendant le développement du système

---

<sup>3</sup> On appelle modèle en cascade un modèle de gestion séquentiel permettant de représenter les développements à travers des phases successives.

mais sa participation se situe uniquement au début lors de la phase d'analyse du système d'information. Rigoureuse et surtout détaillée, elle permet notamment de concevoir un système informatique d'une façon standardisée et méthodique

Depuis 1980, la méthode Merise a connu des développements et des enrichissements remarquables, dont les principaux sont les suivants : extension du formalisme entité-relation (types et sous-types, de contraintes d'intégrité, ...), extension des niveaux d'abstraction et de modèles (modèle logique de traitements, modèle organisationnel de données), développement d'ateliers de génie logiciel (AMC, MEGA, SILVERRUN, WIN'DESIGN, ...).

### 1.3 Les trois composantes de Merise

La mise en œuvre de la méthode Merise doit se repérer par rapport au trois dimensions suivantes

#### a) La démarche ou cycle de vie

La méthode Merise propose, pour le déroulement du cycle de vie, le découpage en trois périodes (conception, réalisation et maintenance). Chaque période est découpée en différentes étapes.

#### b) Le raisonnement ou cycles d'abstraction

Merise a quatre niveaux d'abstraction : **Le niveau conceptuel, le niveau organisationnel, le niveau logique et le niveau physique.** Ces niveaux sont adaptés à la conception de deux systèmes d'information : le (SIO), et le (SII). Le niveau conceptuel qui définit les choix de gestion en termes d'informations et des activités et le niveau organisationnel qui définit les choix d'organisation des ressources humaines et matérielles sont classes dans le système d'information organisationnel ; tandis que le niveau logique qui détermine les choix de moyens et de ressources informatiques et le niveau physique porte sur les choix techniques retrouvent dans le système

d'information informatisé (SII).

### ➤ Les modèles de Merise

A chaque niveau d'abstraction, et pour chaque volet (communication<sup>4</sup>, données<sup>5</sup>, traitements<sup>6</sup>), le système d'information est représenté par un modèle. Tout modèle est exprimé dans un formalisme utilisant des concepts adaptés, comme le présente le tableau suivant :

Tableau 1: Les modèles de Merise

<b>Préoccupation</b> <b>Abstraction</b>	Communications	Données	Traitements
Conceptuel	MCC	MCD	MCT
Organisationnel	MOC	MOD	MOT
Logique	MLC	MLD	MLT
Physique	MPC	MPD	MPT

### c) La maîtrise ou cycle de décision

Dans la pratique, le cycle de décision est intégré dans le cycle de vie. Cela se traduit par des résultats types à l'issue de chaque étape et par des décisions attendues.

À la vue des insuffisances relevés dans son utilisation, des extensions ont donc été apportées à Merise I qui forment la méthode Merise 2. Les principales évolutions par rapport à Merise sont :

- L'utilisation des MFC avec des techniques de perfectionnement ;
- La modification des modèles de traitement (MCT-> MCTA) ;
- L'extension du MCD.

## II. Utilisation de la méthode MERISE 2

### 2.1 Les modèles conceptuels

La description conceptuelle permet de représenter la finalité du système et sa

<sup>4</sup> Echange d'informations entre acteurs (porteurs de données)

<sup>5</sup> Nature, organisation, relations entre les données

<sup>6</sup> Les actions réalisées sur les données

raison d'être, en s'appuyant sur ses objectifs et les réalités externes qui le contraignent. Le niveau conceptuel traite des événements et fournit des résultats sans se soucier de la manière dont sont acquises et restituées les informations portées par ces événements et résultats. Autrement dit, Le premier niveau, dit « conceptuel », décrit le « QUOI ? », c'est-à-dire ce que fait le système d'information, sa raison d'être (quoi faire ? avec quelles données ?). Les fonctions sont décrites sans tenir compte des contraintes matérielles ni des aspects d'organisation du travail dans l'entreprise.

Les différents modèles proposés sont : Le Modèle de Contexte (MC) et le Modèle de Flux Conceptuels (MFC), Le Modèle Conceptuel des Données (MCD), Le Modèle Conceptuel de Traitements (MCT), le Modèle Conceptuel des Traitements Analytique (MCTA).

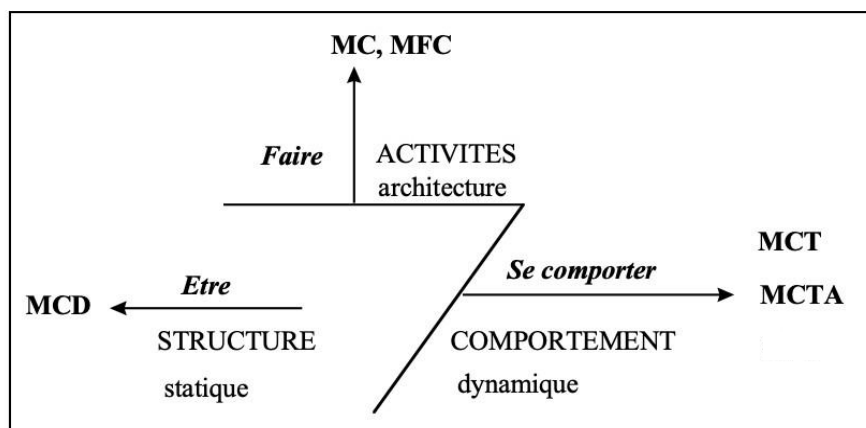


Figure 1: Les modèles conceptuels

#### A. Le modèle de contexte (MC) et le modèle de flux conceptuels (MFC)

Dans Merise, les modèles conceptuels de communication (modèles conceptuels de flux) sont des modèles fonctionnels. Ils répondent à la question : **Que fait le système ?** L'objectif de ce modèle est de représenter les mouvements des données à l'intérieur d'un Système d'Information et entre le SI et l'environnement. Le concept fondamental ici est le message : il contribue à décrire un flux d'activités et symbolise les rapports

entre les systèmes fonctionnels (internes et externes) à l'entreprise. On distingue deux niveaux de représentation : le modèle (ou diagramme) de contexte et le modèle (ou diagramme) de flux conceptuel qui affinent par étape le modèle de contexte.

### 1) Les concepts

Plusieurs concepts sont liés à la modélisation conceptuelle et sont utiles à sa compréhension et à sa matérialisation. Il s'agit du domaine d'étude, du domaine connexe, de l'acteur et enfin du Flux d'information.

- **Le domaine d'étude**

Une des premières étapes dans Merise consiste à délimiter le domaine de l'étude. Le domaine d'étude regroupe l'ensemble des processus ou traitements homogènes contenu dans le système à étudier. Il est un sous ensemble de l'organisation dont on étudie séparément le SI. Autrement dit, c'est le domaine sur lequel porte l'analyse à réaliser. Chaque domaine d'étude est considéré comme « quasi-autonome » avec son propre SP, SI et SO.

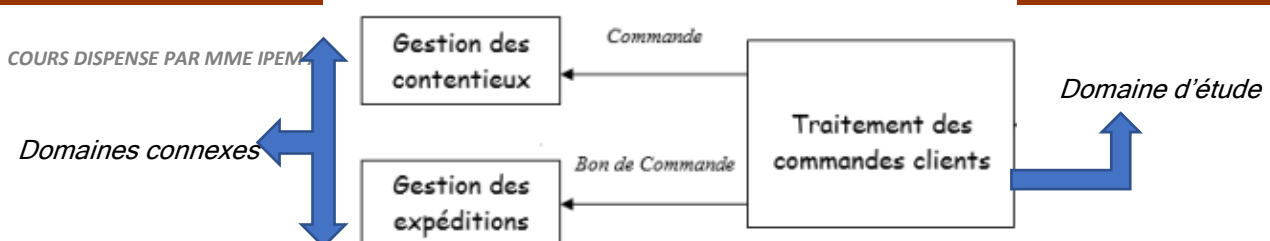
**Exemple :** *Traitement des commandes / Traitement des règlements des clients, gestion des absences, gestion des activités commerciales...*

- **Le domaine connexe**

Un domaine connexe appartient à l'entreprise, il interagit avec le domaine d'étude par échange d'informations mais n'en fait pas partie. Il correspond aux autres domaines d'activités de l'entreprise. Ils ne sont représentés que s'ils échangent des informations avec le domaine d'étude.

#### Remarque

*Dans un diagramme de contexte, on ne fait pas apparaître les flux de données entre acteurs externes et domaines connexes, ou entre les domaines connexes. Les domaines (étude ou connexe) sont symbolisés par un rectangle à l'intérieur duquel est noté le nom*



du domaine.

**Exemple :**

Figure 2: Représentation d'un domaine connexe en interaction avec le domaine d'étude

- **L'acteur**

Un **acteur** est un émetteur ou un récepteur d'un flux d'information lié à une activité au sein du système d'information. Autrement dit, Un acteur est **une entité, humaine ou matérielle**, intervenant dans le système d'information.

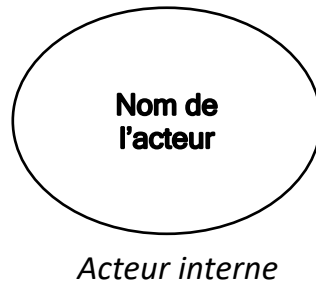
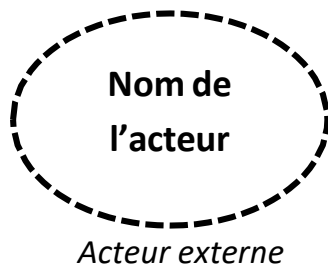
Un acteur reçoit une information qui lui permet d'agir en la transformant en une ou plusieurs autres informations destinées à d'autres acteurs. Les acteurs sont représentés par leur **rôle** (*fonction, service ou direction...*) dans le domaine étudié. Autrement dit, il est une **unité active : il fait quelque chose**. Les acteurs peuvent être : **des personnes** (*le client, le comptable...*), **des services** (*secrétariat, service vente, la banque...*), **des appareils** (*un lecteur de badge qui fait office de contrôle d'entrée...*)

Nous distinguons deux types d'acteurs :

- **L'acteur interne** qui fait partie du domaine d'étude ; (*Ex : service vente, service comptable, Service de la scolarité...*). Il est représenté par une **ellipse** ou **un cercle au tracé continu** ;



- **L'acteur externe** qui est située hors des limites du système et qui fournit des entrées de données ou accepte des sorties de données (*client, fournisseur...*) dans le cadre de l'activité étudiée. Il est considéré comme un partenaire au système. Sa représentation graphique est une ellipse ou un cercle **au tracé en pointillé**



Dans la notation que nous retiendrons, le nom de l'acteur est placé à l'intérieur du cercle. Afin de faciliter la modélisation, nous pouvons utiliser un tableau appelé le dictionnaire des acteurs qui est un tableau qui récence l'ensemble des acteurs du système étudié ainsi que leur type et leurs caractéristiques. Tous les acteurs répertoriés ici devront figurer dans la matrice (ou tableau) des flux

Tableau 2: Tableau des acteurs

Acteur	Description	Type	Caractéristiques
Client	Un client	Externe	Personne
Accueil	L'accueil téléphonique du service commercial	Externe	Service
Spécialiste	Un commercial spécialiste	Interne	Personne

- **Flux d'information**

Un **flux d'information** désigne un transfert d'information entre des acteurs du SI. Il part d'un acteur source (**émetteur**) pour aboutir à un acteur but (**ou récepteur**). En d'autres termes, c'est l'échange d'informations entre acteur du domaine d'étude et une composante extérieure ou entre deux activités du domaine (flux internes et externes). Il est symbolisé par une flèche dont l'orientation désigne le sens du flux d'information.

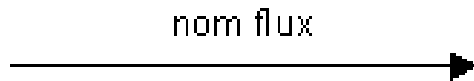


Figure 3: Représentation graphique d'un flux

Les flux peuvent intervenir dans un ordre déterminé qu'on peut noter pour faciliter la lecture. Cet ordre chronologique n'est pas nécessairement systématique et n'exclut pas la simultanéité : certains flux peuvent être émis en même temps et porter le même numéro d'ordre. Un flux peut être **conditionnel** dans le sens où il n'a lieu que lorsqu'une condition est remplie. Dans ce cas on peut noter cette condition entre guillemets « ».

Tout comme avec les acteurs, un dictionnaire de flux entre les acteurs peut être schématisé pour une meilleure compréhension de l'activité.

### Exemple

*Repérer dans ce système les flux entre les acteurs en les ordonnant si c'est utile à une meilleure compréhension de l'activité.*

*Le client pose sa question (1) auprès l'accueil. Cette dernière peut répondre à la plupart des questions courantes (2), elle assure une assistance de premier niveau*

Tableau 3: Dictionnaire des flux

Flux	N°	Description : contenu, émetteur/récepteur
Question	1	Question du client posée l'assistante N1
Réponse	2	Réponse de niveau 1, de l'assistante au client

### Remarques

Tous les flux répertoriés devront figurer dans la matrice des flux. Ce dictionnaire et cette matrice des flux ne sont pas obligatoire

- Le flux ne doit pas être réflexif ;
- Pas de flux entre des acteurs externes.

## 2) Étapes de construction du MCC

Le respect scrupuleux de ces étapes est primordial à l'élaboration ou schématisation de ce dernier

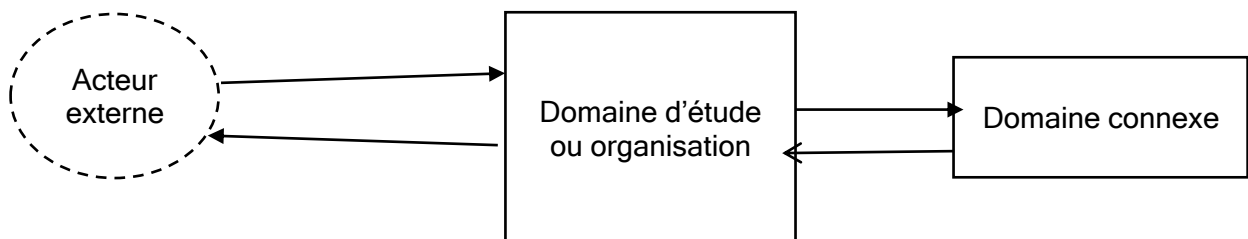
1. Trouver le domaine d'étude
2. Lister tous les acteurs (internes, externes ainsi que le(s) domaine(s) connexe(s) existant(s)) qui sont amenés à agir sur le système
3. Etablir la liste des flux (Toutes les informations échangées)
4. Réalisation du MCC (acteur + n° des flux)

### 1) Formalisme du MCC

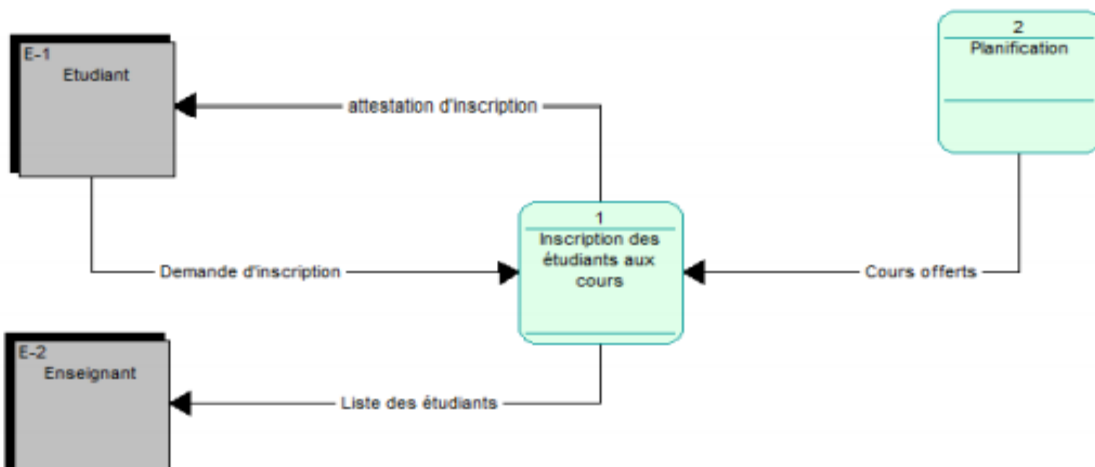
Il repose sur la compréhension des deux modèles qui le compose à savoir : **Le modèle de contexte (MC)** et **modèle conceptuel de flux (MCF)**.

#### a) Le modèle de contexte (MC)

C'est un modèle qui modélise les interactions entre le domaine d'étude et l'environnement, et entre le domaine d'étude et les éventuels domaines connexes. Le domaine d'étude y est représenté comme une boîte noire (processus). En d'autres mots, c'est le schéma qui présente tous les flux échangés entre le système et ses partenaires (acteur externe et domaine connexe). Il fait abstraction du détail des différents acteurs internes.



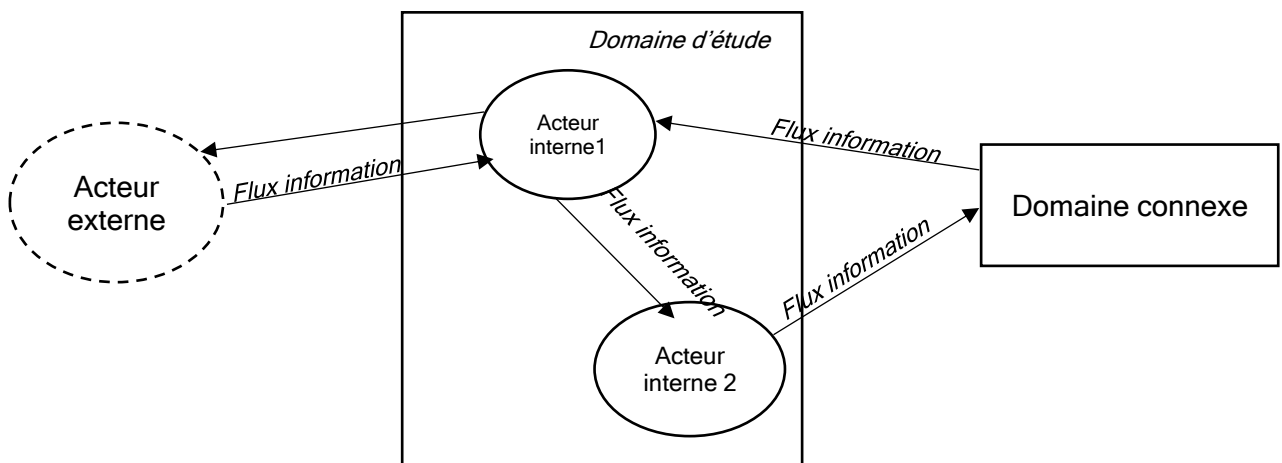
#### Exemple



L'exemple montre un diagramme de contexte d'un système d'inscription à des cours qui interagit avec deux agents externes : enseignant, étudiant et un domaine connexe « planification des cours » qui offre la liste des cours programmés.

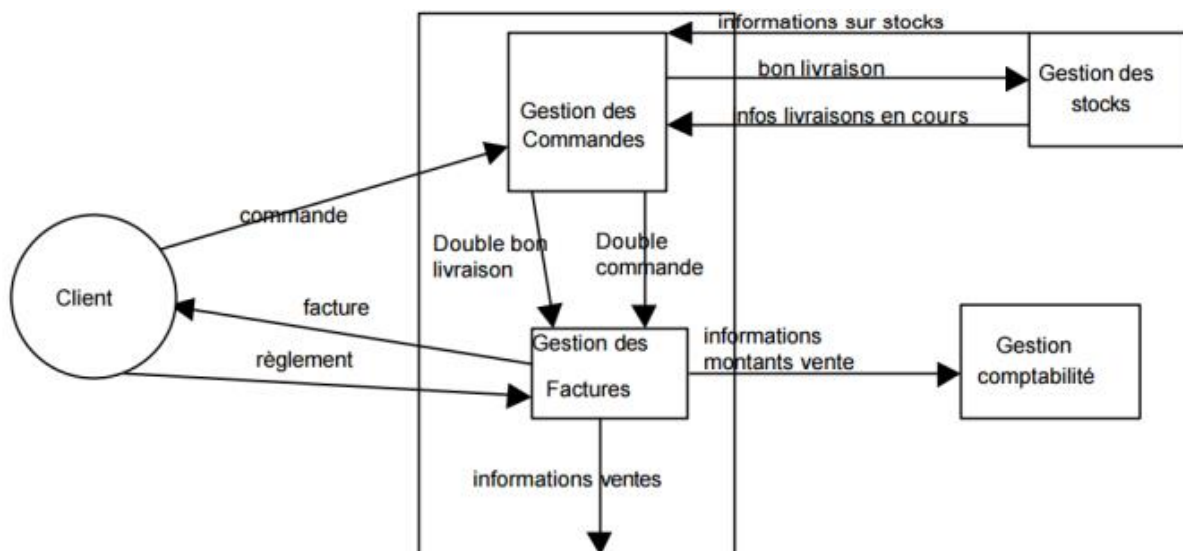
### b) Le Modèle de flux conceptuel (MFC)

Ce modèle permet de compléter celui de contexte en décomposant l'organisation en une série d'acteurs internes. Il schématise l'ensemble des acteurs internes au domaine étudié et l'ensemble des flux échangés entre eux. De plus, il est un point d'appui pour affiner l'analyse et trouver d'autres acteurs et d'autres flux utiles au bon déroulement de l'activité. Par conséquent, il peut permettre de détecter les redondances ou des incohérences dans la circulation de l'information.



#### Exemple

L'exemple ci-dessous montre une représentation d'un modèle de flux conceptuel



**Étude de cas applicatif**

Au supermarché "CARREFOUR", quand un client s'adresse à un vendeur et lui dit qu'il souhaite acheter un téléviseur (par exemple), celui-ci vérifie sur ordinateur si l'article est disponible. Si ce n'est pas le cas, le vendeur informe le client de l'indisponibilité du matériel souhaité ; sinon le vendeur remplit un bon avec les références de l'article et le prix à payer. Le client doit alors se présenter à la caisse, muni de ce bon. Une fois le montant à payer réglé, la caissière lui remet une facture et un bon de sortie de stock. Un double de bon de sortie est envoyé à l'entrepôt afin que soit apporté au magasin (au rayon "retrait des articles ») l'article acheté. Le client se présente alors au rayon "retrait des articles", juste à côté de la caisse et présente son bon de sortie de stock. Dès que l'article est arrivé, le magasinier tamponne le bon de sortie : "article livré" et remet l'article acheté au client.

**Travail à faire**

- a) Trouvez le domaine d'étude de ce système
- b) Dressez la liste des acteurs internes et externes et des domaines connexes au domaine d'étude ;

- c) Élaborez le modèle de contexte ;
- d) Élaborez le MFC correspondant.

### 3) Matrice des flux

Le DF peut être représenté par une matrice carrée appelée matrice des flux. C'est une matrice qui présente en colonnes et en lignes les acteurs intervenant dans le diagramme ; les flux sont représentés à l'intersection des lignes et des colonnes. Elle peut être schématisée comme suit :

Émetteur / Récepteur	Acteur 1	Acteur2	Acteur3
Acteur 1		Flux 1	Flux 3
Acteur 2	Flux 2		
Acteur 3		Flux 4	

#### Exemple d'application

Reprenez l'étude de cas applicatif et schématisez sa matrice de flux.

## A. Le Modèle Conceptuel de Données (MCD) ou Modèle Entité-Association (MAE)

Le **Modèle Conceptuel de Données (MCD)** est une représentation statique du système d'information de l'entreprise. Il a pour but de représenter de façon structurée les données qui seront utilisées par le système d'information. Le MCD décrit **la sémantique** c'est à dire le sens attaché à ces données et à leurs rapports. C'est aussi un ensemble de concepts et de règles permettant de définir comment représenter des informations dans un système informatique. Il est basé sur deux notions principales : les **entités** et les **associations**, d'où sa seconde appellation de **Modèle Entité-**

**Association (MEA).** Nous commencerons par décrire le MCD originel, puis les extensions proposées dans Merise 2.

## 1. Les concepts de bases utilisés

### a) Les entités ou objets

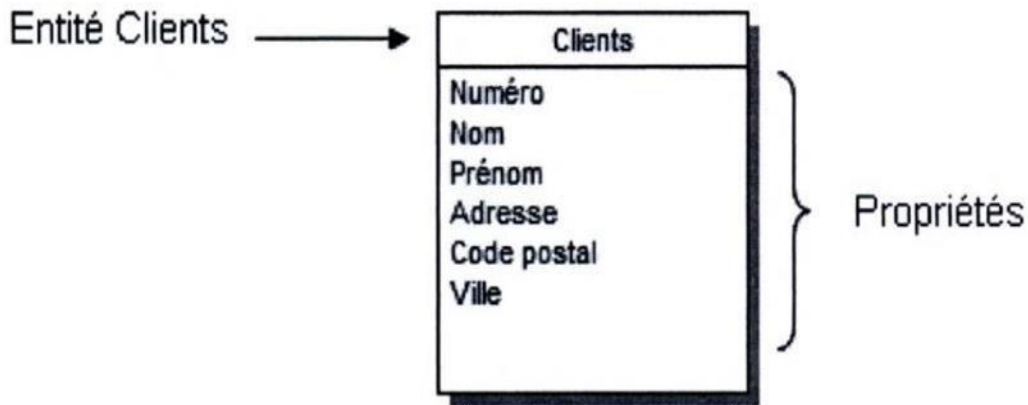
Une **entité** est un élément concret ou abstrait qui peut être reconnu distinctement (personne, produit, service, pays, facture...) et qui est caractérisé par sa typicité c'est-à-dire un ensemble d'objets de même nature. Il s'agit en fait, d'un objet **identifiable** et **pertinent** pour l'entreprise ou encore un ensemble d'objets ayant les mêmes caractéristiques. Dotée d'une existence propre (autonome, libre.), elle est décrite par un **identifiant** et une liste de caractéristiques qui lui sont spécifiques. Une entité est représentée par un rectangle à cartouche. Le nom de l'entité est placé dans la cartouche et est écrit en **MAJUSCULE** : il est recommandé de choisir un nom commun décrivant l'entité (**exemple : ETUDIANT, ENSEIGNANT, MATIERE**). Les attributs prendront place dans la partie basse.

**Notation :**

Nom du type d'entité

### b) Attribut ou propriété

Un **attribut** (ou une **propriété**) est une information de base du SI. Elle est une caractéristique associée à une entité ou à une association. Au niveau de l'entité ou de l'association, chaque attribut possède un **type** qui définit l'ensemble des valeurs possibles qui peuvent être choisies pour lui (**entier, alphanumérique, numérique, booléen, réel, Date...**). Un attribut ne peut en aucun cas être partagé par plusieurs entités ou association.

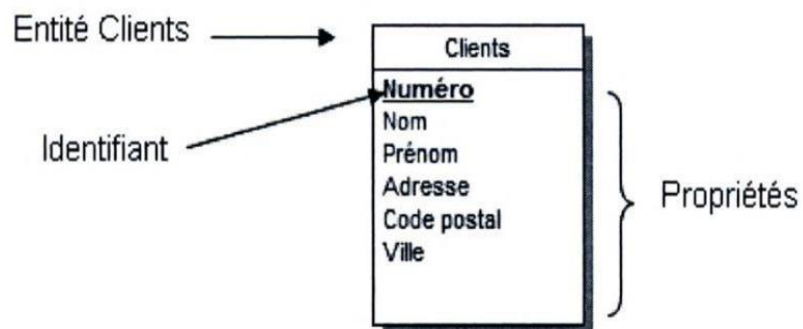


### c) Identifiant ou clé primaire

Un **identifiant** d'une entité ou d'une association est constitué par un de ses permet de connaître de façon sûre et unique l'ensemble des propriétés qui participent à l'entité. Il est donc impossible que l'attribut constituant l'identifiant d'une entité ou d'une association prenne la même valeur pour deux entités ou deux associations distinctes. Néanmoins, certains cas particuliers peuvent exister notamment celui d'une entité ne contenant qu'un seul attribut. Dans ce cas, cet attribut est donc forcément l'identifiant.

**NB :** la notation graphique et manuelle d'un identifiant demande de le placer en tête de liste, de le souligner et de le faire précéder d'un #. Celle avec l'applicatif se note comme ci-dessous.

#### Notation

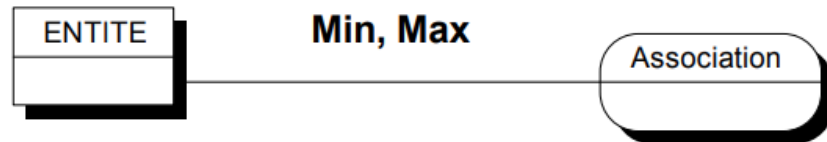


### d) Cardinalités

Les cardinalités sont des couples de valeurs que l'on trouve entre chaque entité



et ses associations liées et expriment le nombre de fois ou l'occurrence(possibilité) qu'une entité participe aux occurrences de la relation. Donc, pour une association de 2 entités, il y a 4 cardinalités à indiquer (2 de chaque côté). Les cardinalités traduisent des **règles de gestion**. Ce sont des règles propres à l'organisation étudiée, qui sont décidées par les gestionnaires et décideurs. Ces règles expriment des contraintes sur le modèle.



➤ **Cardinalité minimale**

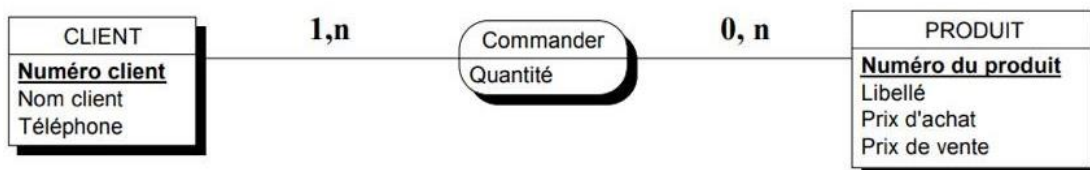
Elle est exprimée presque toujours par l'une des deux valeurs 0 ou 1. Elle traduit combien de fois au minimum une occurrence de l'entité participe à l'association.

➤ **Cardinalité maximale**

La cardinalité maximale exprime le nombre maximum de fois qu'une occurrence d'une entité participe à une relation.

En fait, dans la grande majorité des cas, on n'utilise que 4 combinaisons de valeurs pour les cardinalités à savoir :

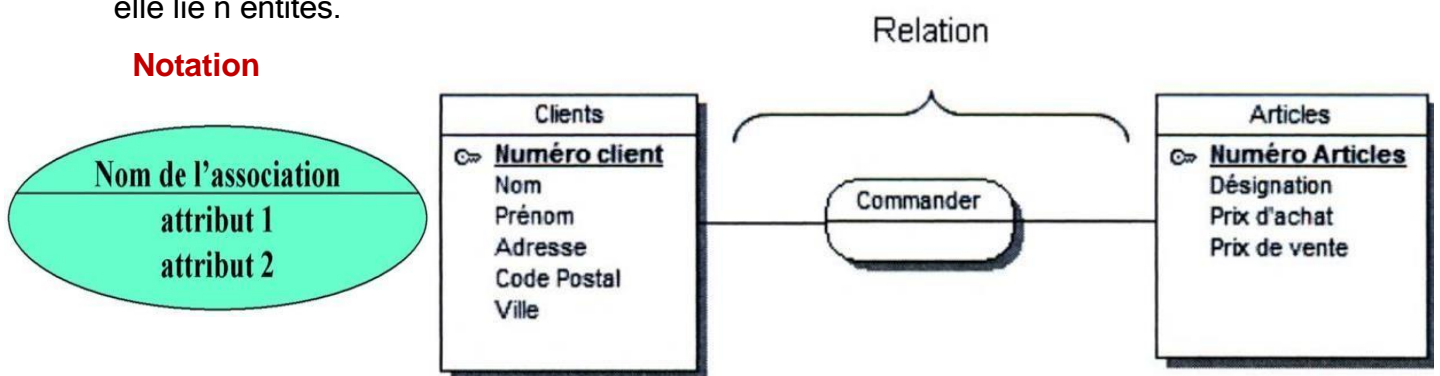
- **(0,1) au plus un(e)**
- **(1,1) un(e) et un(e) seul(e)**
- **(1, n) un(e) ou plusieurs**
- **(0, n) zéro ou plusieurs**



### c) Association ou relation

Une **association (ou une relation)** est une attache entre une ou plusieurs entités. Une association est représentée graphiquement par **une ellipse à cartouche** dans laquelle figure son nom, souvent **un verbe à l'infinitif**, à la forme passive ou bien accompagné d'un adverbe qui caractérise le type de relation entre les entités. Notez que l'initiale du nom de l'association est en **MAJUSCULE**. Une association possède parfois des propriétés. Une association est dite binaire si elle lie 2 entités et n-aire si elle lie n entités.

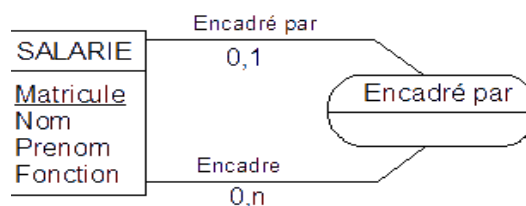
#### Notation



#### ✓ Dimension d'une association

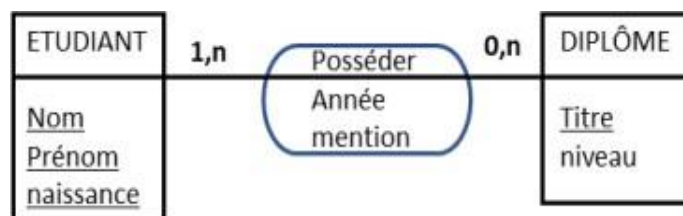
La dimension d'une relation est le nombre d'objets participant à l'association. Elle est dite : **réflexive** lorsqu'elle relie la même entité

#### Exemple

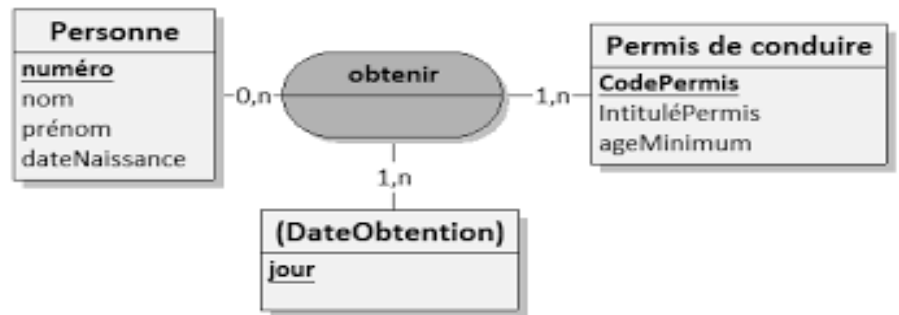


**Binaire** relie deux entités

#### Exemple

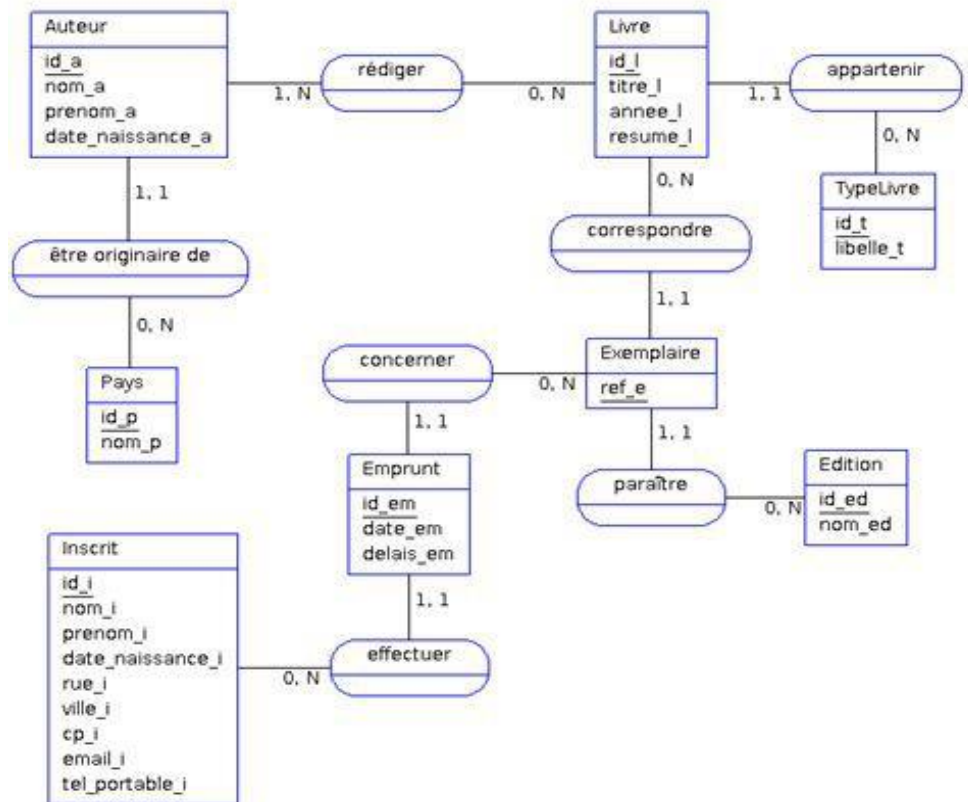


Ternaire relie trois entités,



*Exemple*

N -aires relie n entités.

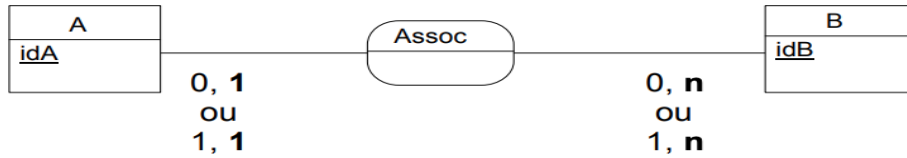


### ✓ Les différents types d'associations

On différencie deux types d'associations : hiérarchiques et non hiérarchiques.

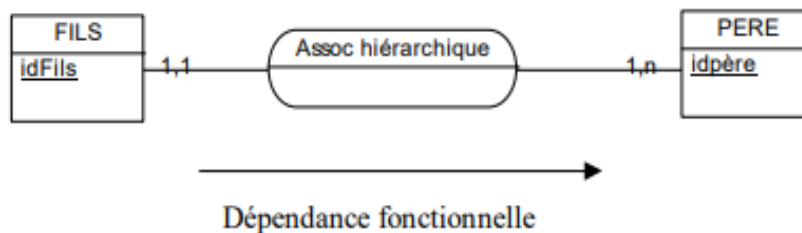
**Associations hiérarchiques** : associations où d'un côté de la branche la cardinalité maximale est à 1 et de l'autre côté la cardinalité maximale est à N.

Cela signifie qu'une occurrence de A est reliée au plus à une seule occurrence de B. C'est-à-dire si on connaît une occurrence de A alors on saura forcément quelle est la seule occurrence de B qui correspond (si elle existe). On dit que **A détermine B**. C'est



un lien de dépendance fonctionnelle. B dépend fonctionnellement de A.

L'entité qui correspond à la branche du côté du 1 est parfois appelée entité fils et l'entité correspondant à la branche du côté n est parfois appelée entité père. Cette appellation découle de l'analogie : un fils n'a qu'un seul père, et un père peut avoir plusieurs fils.

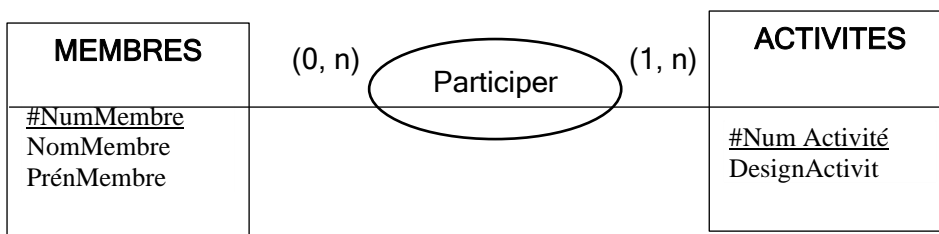


Ce type d'association n'a jamais de propriété portée. Si vous êtes tentés de mettre une propriété à l'intérieur de l'association, c'est soit que vos cardinalités sont fausses, soit que la propriété devrait être dans une des entités.

Le lien qui associe l'entité fils à l'entité père est de même nature que le lien qui lie les propriétés à l'entité fils. C'est une dépendance fonctionnelle. On dit que l'entité Père est en dépendance fonctionnelle de l'entité fils.

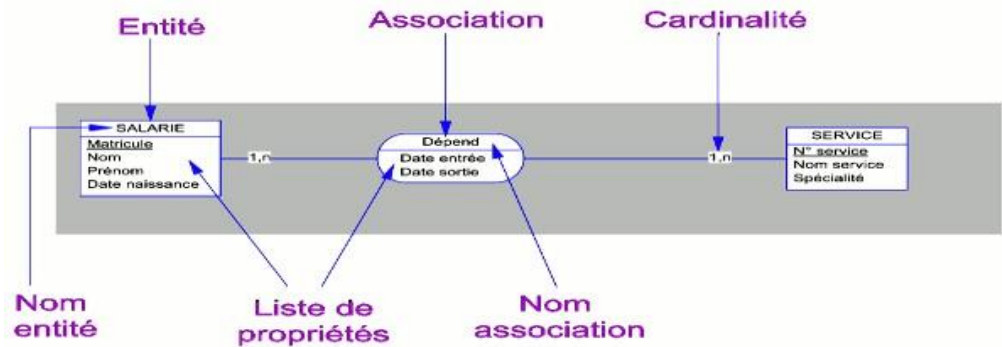
### Associations non hiérarchiques

Lorsque les cardinalités maximales de chaque côté des associations sont supérieures à 1, l'association est dite non hiérarchique car la connaissance d'une occurrence d'une entité ne permet pas de connaître une mais plusieurs occurrences d'une autre entité.



Un membre peut ne participer à aucune des activités. Il peut participer à plusieurs.

En résumé :

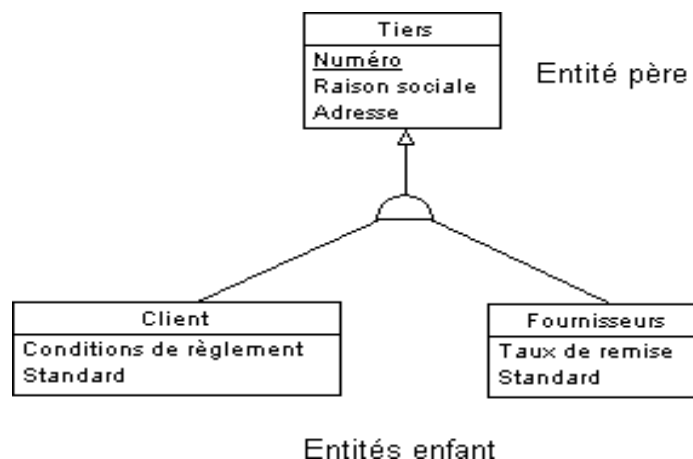


### c) La notion d'héritage dans un MCD

L'**héritage** permet de définir une entité spécialisée par rapport à une entité générale. Dans un héritage, les entités ont beaucoup de caractéristiques communes mais sont toutefois différentes. L'entité générale est appelée entité **sur type** ou **père**, elle est dotée de toutes les caractéristiques communes aux entités spécialisées qui elles, sont appelées **sous-type**,  **fils** ou **enfant**. Les entités sous-types ne disposent que des caractéristiques qui leur sont propres.

Vous pouvez établir un lien d'héritage entre une entité générale et des entités spécialisées. Dans un lien d'héritage, une ou plusieurs entités enfant héritent, au niveau physique, de tout ou partie des attributs d'une entité père.

**Exemple :** Soit le cas d'une entreprise gérant sa comptabilité où tous les partenaires avec lesquels elle commerce sont considérés comme des tiers. Parmi ces tiers, on souhaite distinguer deux sous-types : les clients et les fournisseurs. En tant que tiers, clients et fournisseurs ont des caractéristiques communes, mais ils ont aussi des caractéristiques spécifiques telles que les conditions de règlement ou les taux de remises. Dans un héritage, les entités Clients et Fournisseurs constituent des types spécialisés de l'entité père Compte. Graphiquement, vous obtiendrez le résultat suivant :



## 2. La réalisation d'un MCD

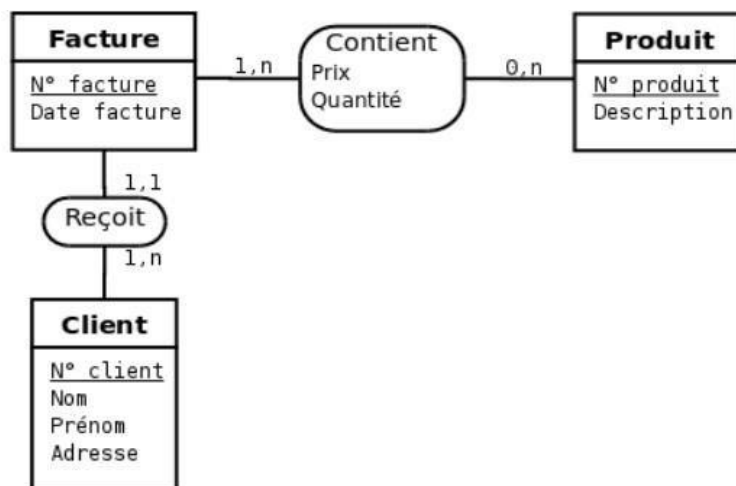
**Principe :** Réaliser un MCD demande de passer par un ensemble d'étapes bien définies. A partir d'un cahier des charges, concevoir de manière visuelle les différents liens qui existent entre les différentes données. Ces dernières peuvent en deux catégories : la modélisation directe et la modélisation par analyse (Ou indirecte)

### a) 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**

*Un client reçoit une facture contenant des produits Ce qui donne la modélisation suivante :*



Le modèle obtenu par cette méthode comme le montre l'exemple ci-dessus, 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.

### b) La modélisation par analyse ou indirecte

Ici, certaines étapes doivent être respectées, mais pour un souci de compréhension, nous les résumerons en cinq (05) étapes fondamentales. Il s'agit de :

**Étape 1 : L'identification en premier lieu de toutes les propriétés du système d'information à analyser.**

L'identification des propriétés du SI à étudier passe par l'établissement du **dictionnaire de données** et de **son épuration**. Le **dictionnaire de données** est un document essentiel à la construction du MCD car il a pour but de recenser et collecter l'ensemble des données élémentaires (ou attributs) manipulées par le système et se rapportant strictement au domaine étudié.

✓ **Établissement du dictionnaire de données**

C'est une étape intermédiaire qui peut avoir son importance, surtout si vous êtes plusieurs à travailler sur une même base de données, d'un volume conséquent.

Concevoir le dictionnaire de données revient à extraire des documents en notre possession et des entretiens effectués, l'ensemble des informations strictement indispensables à la gestion du domaine d'activité à informatiser. Il faut faire la liste exhaustive de toutes les données qui sont utilisées dans le cadre d'un système d'information et leur attribuer un nom différent pour chaque champ. Toutes les données sont recensées dans un tableau en sachant que l'on distingue 3 types de données :

- **Données élémentaires** : Elles ne sont pas obtenues par calcul à partir d'autres données.

**Exemple** : On donne la quantité, le prix de l'article, calculer le coût total... La quantité et le prix sont des données élémentaires

- **Données calculées** : Elles résultent d'un calcul effectué à partir d'autres données.

**Exemple** : Le coût total est une donnée calculée (= quantité \* prix unitaire).

- **Les données concaténées** : Elles résultent de plusieurs propriétés

**Exemple** : Étudiant est une donnée concaténée Étudiant (= nom, prénom, âge...)

Le dictionnaire des données est un document/tableau qui regroupe toutes les données que vous aurez à conserver dans votre base (et qui figureront donc dans le MCD). Pour chaque donnée, il indique :

- Le **nom de la donnée/propriété** indiquant son nom intégral

- Le **code mnémonique** : il s'agit d'un libellé ou une abréviation désignant une donnée (par exemple «*titre\_1*» pour le titre d'un livre)
- La **désignation** : il s'agit d'une mention décrivant ce à quoi la donnée correspond (par exemple «titre du livre»)
- La **nature de la donnée** :
  - **A ou Alphabétique** : lorsque la donnée est uniquement composée de caractères alphabétiques (de 'A' à 'Z' et de 'a' à 'z') ;
  - **N ou Numérique** : lorsque la donnée est composée uniquement de nombres (entiers ou réels) ;
  - **AN ou Alphanumérique** : lorsque la donnée peut être composée à la fois de caractères alphabétiques et numériques ;
    - **Date** : lorsque la donnée est une date (au format AAAA-MM-JJ) ;
    - **Booléen** : Vrai ou Faux'
    - **Paramètre** : la valeur de la donnée est invariable
- La **taille** : elle s'exprime en nombre de caractères ou de chiffres. Dans le cas d'une date au format AAAA-JJ-MM, on compte également le nombre de caractères, soit 10 caractères. Pour ce qui est du type booléen, nul besoin de préciser la taille (ceci dépend de l'implémentation du SGBDR).
- Et parfois des **remarques** ou **observations** complémentaires (par exemple si une donnée est strictement supérieure à 0, etc.).

N°	Nom de la donnée(Propriété)	Code ou Mnémonique	Nature	Type	Longueur	Commentaires/ Remarques ou observations
01	<b><i>Identifiant du personnel</i></b>	Id_Pers	Elémentaire(E)	AN	20	
02	<b><i>Total facture</i></b>	Tot_Fact	Calculée (Ca)	N	15	

**NB**



Les données qui figurent dans le MCD (et donc dans le dictionnaire des données) doivent être, dans la plupart des cas, élémentaires :

- Elles ne doivent pas être calculées : les données calculées doivent être obtenues, par le calcul, à partir de données élémentaires qui, elles, sont conservées en base. Cependant, il existe quelques cas où il s'avère pertinent de conserver, pour des raisons d'optimisation, une donnée calculée. Par exemple le montant d'une commande par exemple. On ne conservera cependant pas les données calculées intermédiaires sauf en cas d'obligation légale (c'est le cas pour un montant HT par exemple, où les composantes peuvent d'ailleurs avoir un prix variable dans le temps). En effet, cela évite de refaire les calculs plusieurs fois pour un résultat qui restera fixe.
- Elles ne doivent pas être composées : les données composées doivent être obtenues par la concaténation de données élémentaires conservées en base. Par exemple une adresse est obtenue à partir d'une rue, d'une ville et d'un code postal : ce sont ces trois dernières données qui sont conservées et donc qui figureront dans le MCD (et dans le dictionnaire des données).

### ✓ Épuration du dictionnaire de données

Le but de l'épuration est de nettoyer l'ensemble des attributs obtenus à l'étape précédente, en vérifiant que chaque donnée correspond à un "fragment" d'informations, indivisible et indépendant des autres données.

### **NB : IL NE FAUT AUCUNE REDONDANCE (DIRECTE OU INDIRECTE) DANS UN DDD ÉPURÉ.**

Pour ce faire il faut passer en revue, l'un après l'autre et sans ordre préétabli, chacun des attributs du DDD et vérifier les points suivants :

- **Vérifier les synonymes** : la même donnée utilisée sous deux termes différents par deux acteurs différents ;
- **Vérifier les homonymes** : deux données différentes utilisées sous le même terme par deux acteurs différents ;
- **Vérifier les dépendances directes** : une donnée qui peut être obtenue à partir d'autres données (exemple : prix unitaire HT, TVA, prix unitaire TTC : une de ces données doit être épurée), ;
- **Vérifier les dépendances indirectes et les données calculées** : une donnée obtenue comme totalisation ou comptage d'autres données (exemple : nombre de

factures, ou chiffre d'affaires d'un client).

## Étape 2 : Détermination des dépendances fonctionnelles

Une **dépendance fonctionnelle** est une interrelation, un lien, une association, entre deux données ou deux groupes de données. Soit **A** et **B** des attributs, une information B dépend fonctionnellement d'une information A **si et seulement si à une seule valeur de A, il n'est possible d'associer qu'une et qu'une seule valeur de B.** On parle de **Dépendances fonctionnelles directes** ou **élémentaires**. Autrement dit, si on connaît la valeur de **A**, on peut en déduire **une seule valeur de B**. Mais la réciproque n'est pas vraie (si on connaît b, on ne peut pas en déduire a). La première donnée est dite **source (en ligne)**, et le second but (en **colonne**). On note

**Source**  $\longrightarrow$  **But** soit  $A \longrightarrow B$ .

Une DF doit être :

- **Élémentaire** : C'est l'intégralité de la source (partie gauche) qui doit déterminer le but (partie droite) d'une DF. Exemple : Si  $A \rightarrow C$  alors  $A, B \rightarrow C$  n'est pas élémentaire.

- **Directe** : La DF ne doit pas être obtenue par transitivité. Par exemple, si  $A \rightarrow B$  et  $B \rightarrow C$  alors  $A \rightarrow C$  a été obtenue par transitivité et n'est donc pas directe.

**NB:** En général, Toutes les propriétés d'une entité « dépendent fonctionnellement » de l'**identifiant**

### Exemple

*La connaissance d'un numéro de matricule détermine un seul nom de famille, celui du titulaire de ce numéro. Par contre, un prénom ne détermine rien, car plusieurs personnes peuvent avoir le même prénom. On peut représenter cette DF de la façon suivante :*  
**numMat  $\rightarrow$  nomPers**

Lorsqu'une information dépend fonctionnellement de plusieurs autres informations, la dépendance fonctionnelle est dite **composée**. On note par un **+** cette concaténation. **Exemple** : Si l'on souhaite connaître la note obtenue par un étudiant à un examen, il est nécessaire de connaître le numéro de l'étudiant, le numéro du module, et la session qu'il passait.

Ce qui se représente : **numEtudiant + numModule + numSession  $\rightarrow$  valeurNote**

Dans le cas où l'information ne possède pas de dépendance, alors l'information dépend fonctionnellement d'elle-même.

La détermination des dépendances fonctionnelles dans un MCD, se fait par la réalisation d'une part de **la matrice des dépendances fonctionnelles** et d'autre part, par celui du **graphe des dépendances fonctionnelles**.

### a) La matrice des dépendances fonctionnelles

Une **matrice des DF** est un tableau faisant apparaître horizontalement et verticalement toutes les données recensées dans le DDD.

Soit l'exemple ci-dessous représentant une des factures établies par une entreprise du nom de "Au Bon Beurre".

<b>Entreprise AU BON BEURRE</b> <b>7, rue de la vache joyeuse</b> <b>34200 SÈTE</b>				
<div style="border: 1px solid black; padding: 5px; display: inline-block;">FACTURE</div>				
<b>N° client : 152</b> <b>Nom client : Casino</b> 7 montée St Clair 34200 SÈTE			<b>N° facture : 1258</b> <b>Date facture : 14/08/1999</b>	
Référence	Désignation	Prix unitaire	Quantité	Montant brut
X25	Plaquette beurre 250 g	4,50	500	2 250,00
X32	Crème UHT 125 g	3	400	1 200,00
			<b>Total HT</b>	3 450,00
			<b>Total TVA</b>	189,75
			<b>TTC</b>	3 639,75

La sélection des données a permis de ressortir le DDD suivant :

N°	Nom de la donnée (Propriété)	Code ou Mnémonique	Nature	Type	Longueur	Commentaires/ Remarques ou observations
1	Nom entreprise	Nom_ent	A	P	21	
2	Adresse entreprise	Adr_ent	AN	P	50	
3	Code postal entreprise	CP_ent	AN	P	15	
4	Ville entreprise	Vil_ent	AN	P	30	
5	Numero Client	Num_Clt	N	E	20	
6	Nom Client	Nom_Clt				
7	Adresse Client	Adr_Clt	AN	E	30	
8	Code postal client	CP_Clt	AN	E	25	
8	Ville client	Vil_Clt	AN	E	32	
10	Numero facture	Num_Fact	N	E	10	
11	Date facture	Dat_Fact	D	E	8	
12	Reference	Ref	AN	E	25	
13	Designation	Des	AN	E	15	
14	Prix unitaire	P_Unit	N	E	50	
15	Quantite	Qte	N	E	32	
16	Montant brut	Mont_Brut	N	C	45	
17	Total HT	Tot_HT	N	C		
18	Montant TVA	Mont_TVA	N	C		
19	TTC	TTC	N	C		

### Explication sur le choix des données du DDD

1. La donnée "Nom entreprise" ne prendra pour seule valeur que la valeur "Au Bon

Beurre'', raison pour laquelle elle est de type paramètre. Il en est de même pour les données adresse entreprise, code postal entreprise et ville entreprise ;

2. Numéro client et Numéro facture sont des données élémentaires car elles sont calculées sur elles-mêmes. En effet, le nouveau numéro client ou d'une facture sera égal à l'ancien numéro client ou facture + 1. C'est donc des données élémentaires ;
3. Date Facture, Reference, Prix unitaire, sont des données élémentaires car ne peuvent être obtenu à partir d'un calcul ;
4. Montant brut est une donnée calculée car résulte de la "règle de calcul" entre la quantité et le prix unitaire (**Prix unitaire x Quantité**) ;
5. Montant TVA est une donnée calculée, qui ajoute une autre celle de taux TVA ;
6. TTC est une donnée calculée sans problème particulier ;
7. Taux TVA est un paramètre car nous considérons que tous les produits vendus dans cette entreprise sont vendus à un taux fixe.

Après épuración, il en ressort le dictionnaire final ci-dessous :

N°	Nom de la donnée (Propriété)	Code ou Mnémonique	Nature	Type	Longueur	Commentaires/ Remarques ou observations
1	Numéro Client	Num_Clt	N	E	20	
2	Nom Client	Nom_Clt	A	E	30	
3	Adresse Client	Adr_Clt	AN	E	30	
4	Code postal client	CP_Clt	AN	E	25	
5	Ville client	Vil_Clt	AN	E	32	
6	Numéro facture	Num_Fact	N	E	10	
7	Date facture	Dat_Fact	D	E	8	
8	Reference	Ref	AN	E	25	
9	Désignation	Des	AN	E	15	
10	Prix unitaire	P_Unit	N	E	50	
	Quantité	Qte	N	E	32	

Pour trouver les dépendances fonctionnelles, l'astuce est de trouver les identifiants (clés) des entités à partir de la matrice des dépendances fonctionnelles.

### ✚ Création

Pour créer cette matrice, nous nous occuperons uniquement des **données élémentaires**. Il s'agit d'un tableau à 2 entrées : En ligne et en colonnes, on inscrit les données issues du dictionnaire de données (**données élémentaires uniquement**).

### ✚ Méthode

Pour remplir ce tableau, on considère chaque colonne de données. On pose une question pour chaque colonne de donnée : **Pour une valeur de cette donnée, existe-t-il une seule et unique valeur de la donnée située en ligne ?** Dans l'affirmative, on inscrit le chiffre 1 à l'intersection et dans la négative, on n'inscrit rien du tout.

La matrice des dépendances fonctionnelles de notre précédent exemple est le suivant :

	N° client	Nom client	Adresse client	CP client	Ville client	N° facture	Date facture	Référence	Désignation	Prix unitaire	Quantité
N° client						1					
Nom client	1					1					
Adresse client	1					1					
Code postal client	1					1					
Ville client	1					1					
N° facture											
Date facture						1					
Référence											
Désignation								1			
Prix unitaire								1			
Quantité											

La règle de remplissage de cette matrice est la suivante :

**Première colonne**

**Question** : pour un n° de client existe-t-il un seul et unique nom de client ?

**Réponse** : OUI. Chaque client possède un numéro différent ; Idem pour ses coordonnées (ville, prénom etc. ...)

**Seconde colonne**

**Question** : Pour un nom de client, existe-t-il un seul et unique n° de client :

**Réponse** : NON. Pour un nom de client, je peux avoir plusieurs n° de client

*Exemple : Atangana Etienne possède le n° 1000, Atangana Annie en possède un également : 1005. Les 2 personnes ont le même nom (Atangana).*

On réalise cette opération pour toutes les données situées en colonnes. Après le remplissage de toutes intersections de ladite matrice, il faut maintenant la simplifier pour une meilleure manipulation des données.

**b) Simplification de la matrice des dépendances fonctionnelles**

On ne conserve pas dans le tableau les colonnes **vides** (suppression des colonnes qui ne contiennent qu'une seule fois le chiffre 1)



	N° client	Nom client	Adresse client	CP client	Ville client	N° facture	Date facture	Référence	Désignation	Prix unitaire	Quantité
N° client						1					
Nom client	1					1					
Adresse client	1					1					
Code postal client	1					1					
Ville client	1					1					
N° facture											
Date facture						1					
Référence											
Désignation								1			
Prix unitaire								1			
Quantité											

Et nous ne gardons que les colonnes qui possèdent le chiffre 1 plus d'une fois.

Ce qui nous conduit au tableau simplifié suivant :

	N° client	N° facture	Référence
N° client		1	
Nom client	1	1	
Adresse client	1	1	
Code postal client	1	1	
Ville client	1	1	
N° facture			
Date facture		1	
Référence			
Désignation			1
Prix unitaire			1
Quantité			



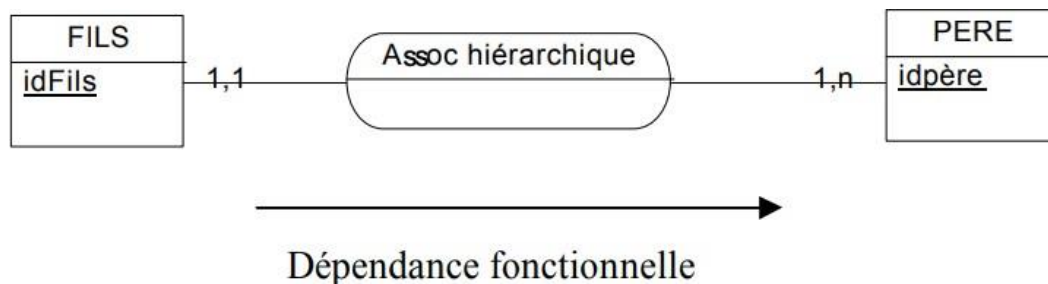
Dans l'exemple, la donnée **Quantité** n'est pas reliée aux données têtes de colonnes ; il s'agit donc d'une donnée qui va dépendre de plusieurs données têtes de colonnes (dépendance fonctionnelle composée).

- **Une dépendance fonctionnelle composée** est de la forme : A, B à C, elle se lit : Pour une valeur de A et une valeur de B, on a une seule valeur de C. La donnée Quantité : Pour un N° de facture et une référence, il y a une seule quantité. Elle se note : **N° facture, référence → Quantité.**

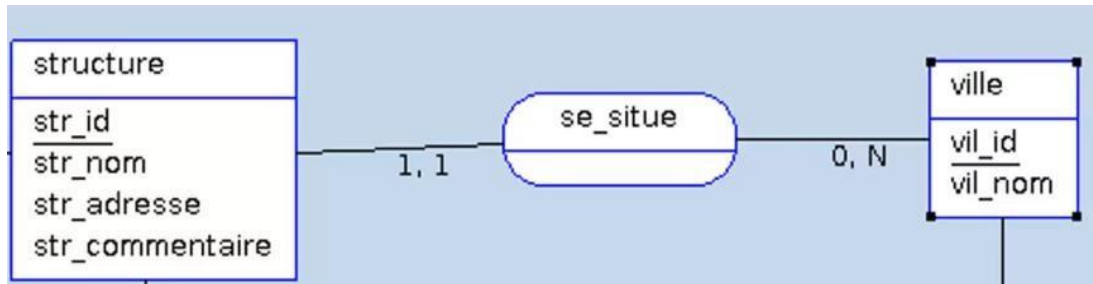
- **Les contraintes d'intégrité fonctionnelles (CIF)**

Dans la matrice des dépendances fonctionnelles, la présence de 1 dans la ligne d'une donnée en tête de colonne traduit la présence d'une contrainte d'intégrité fonctionnelle entre ces deux données. Autrement dit, Une **CIF est un type d'association entre 2 entités. Elle se caractérise par un 1 en cardinalité supérieure (0,1 ou 1,1) sur une des pattes de la relation.** On dit alors que la relation est porteuse d'une dépendance fonctionnelle. Une CIF indique donc une dépendance.

**Exemple :** L'entité qui correspond à la branche du côté du 1 est parfois appelée entité fils et l'entité correspondant à la branche du côté n est parfois appelée entité père. Cette appellation découle de l'analogie : un fils n'a qu'un seul père, et un père peut avoir plusieurs fils.



**NB :** Ce type d'association n'a jamais de propriété portée. Si vous êtes tentés de mettre une propriété à l'intérieur de l'association, c'est soit que vos cardinalités sont fausses, soit que la propriété devrait être dans une des entités.

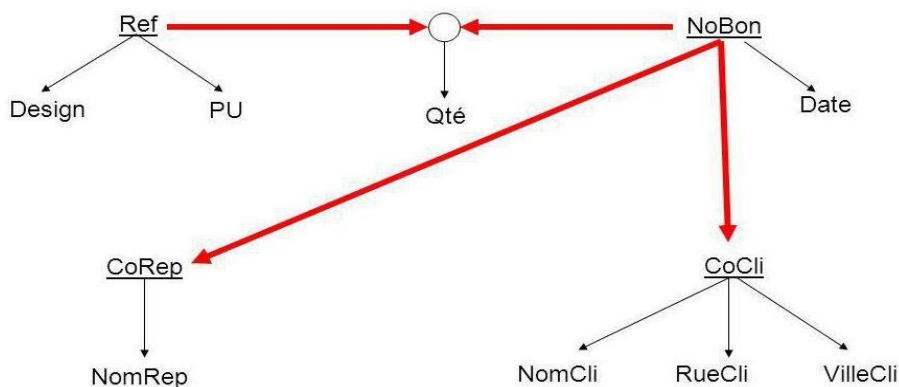


La cardinalité 1,1 sur la gauche de la relation fait de la relation une CIF

### c) Le graphe de dépendance fonctionnelle (GDF)

On peut également représenter une dépendance fonctionnelle à l'aide d'un Graphe de Dépendances Fonctionnelles (GDF). Ce graphe est une représentation graphique de l'ensemble des DF unissant les propriétés dans un domaine d'activité du système d'information. Ces propriétés sont obtenues à partir du dictionnaire de données du domaine.

D'après notre exemple ci-dessus, nous aurons le graphe suivant :



### Étape 3 : Détermination des entités (mise en évidence des objets ou regroupements des attributs en entités)

Pour déterminer les entités, il faut tenir compte du graphe des dépendances fonctionnelles. En effet, il s'agira ici, de mettre en évidence les objets ou de regrouper les différents attributs en famille. Dans un sens général, les identifiants ou clés retrouvés dans le GDF permettent de déduire les entités.

Dans notre exemple, chaque colonne induit la création d'une entité dont la clé primaire est la tête de colonne.

Ce qui nous donne 3 entités : **CLIENT, PRODUIT, FACTURE**

Les clés primaires :

Entité client : N° client (un numéro est unique)

Entité produit : référence

Entité facture : N° facture

CLIENT
# <u>Num_Clt</u>

PRODUIT
# <u>Ref</u>

FACTURE
# <u>Num_Fact</u>

#### **Étape 4 : Détermination des associations**

Elles sont déduites suivant les objets qu'elles lient ou sont construites à partir des dépendances fonctionnelles composées. Toute dépendance fonctionnelle composée induit la création d'une association dont les branches sont reliées aux entités contenant les différentes données clés primaires de l'association

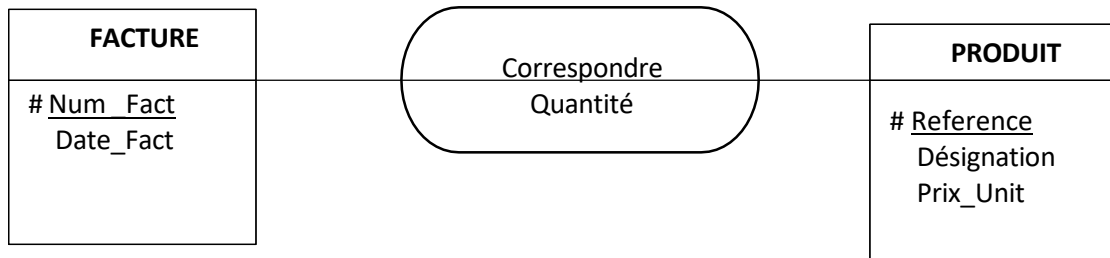
**Ex :** dans notre exemple, N° facture, —→ Quantité est une association dont les branches sont reliées aux entités Facture et Produit, dont les clés primaires sont N° facture et référence.

#### **Transformation du GDF en MCD Règles de transformation**

**R1 :** les données sources d'au moins une DF (celles qui sont soulignées sur le GDF) représentent les identifiants des entités dont les attributs sont les cibles de ces DF.

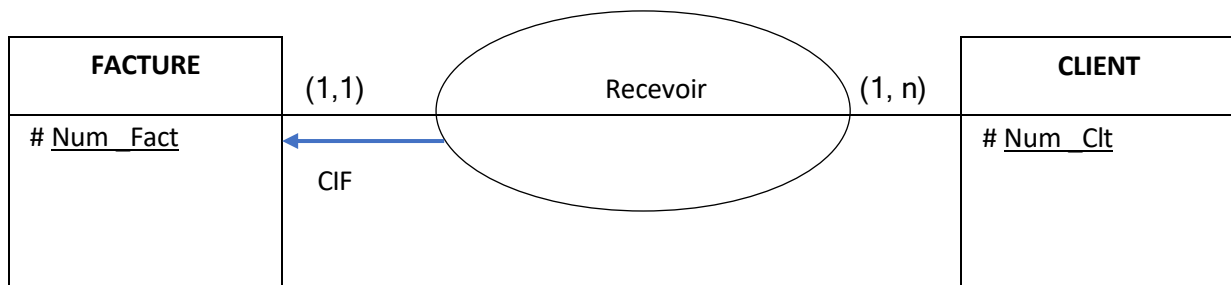
**R2 :** Les flèches restantes deviennent des associations. Les données déterminées par une DF conjointe deviennent des attributs portés par l'association.

**R3** : Les règles de gestion doivent permettre de trouver les cardinalités.



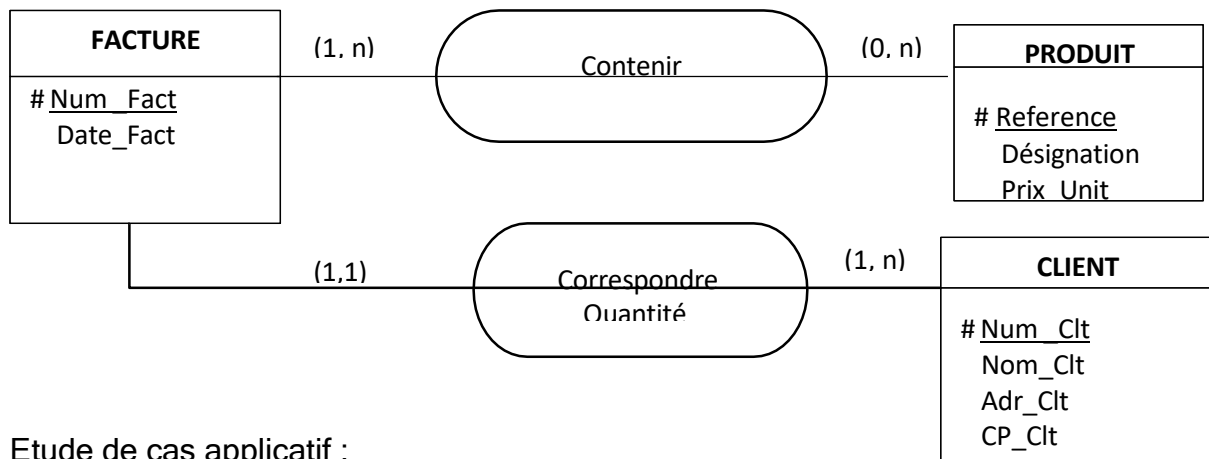
Dans l'exemple, la ligne N° client (qui est également une donnée en tête de colonne), contient un 1 dans la colonne N° facture. Ceci se traduit par l'existence d'une CIF entre N° Facture et N° Client Cette CIF se lit de la façon suivante : POUR UNE FACTURE, IL Y A UN CLIENT.

Représentation de la CIF :



### Étape 5 : Détermination des cardinalités à partir des règles de gestion établies

Le modèle conceptuel finale est le suivant :



Etude de cas applicatif :

TAF : Représentez le MCD correspondant de notre étude applicative de ce cours en suivant les étapes de sa construction