Chapitre 3 :

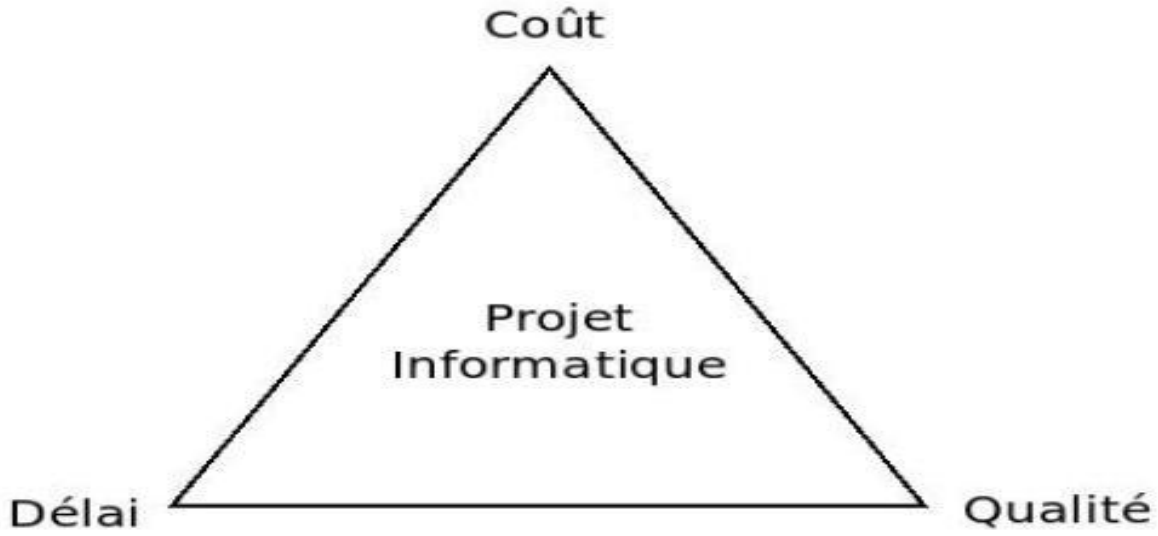
Méthodologie de développement du SI

I)- Introduction

Un logiciel de manière générale et les systèmes d'information en particulier sont des projets d'ingénierie. C'est-à-dire, un projet qui

- Vise à résoudre un problème réel et bien défini,
- Repose sur un ensemble d'outils et de méthodes,
- Est un travail d'équipe, chaque membre d'équipe se charge d'une mission claire et joue un rôle bien défini.
- Passe par plusieurs étapes, chaque étape se compose de plusieurs taches,
- Donne comme résultat un "livrable".

La réalisation d'un projet informatique est soumise à des exigences contradictoires et difficilement conciliables :



- Délai : le projet doit respecter les délais fixes,
- Cout : le projet ne doit pas dépasser le budget attribué,
- Qualité : le projet doit aboutir a un livrable de qualité.

Il est très difficile de respecter toutes les trois exigences.

Tenant compte de ces notions et pour pouvoir mener à bien la réalisation d'un Système d'Information, il faut faire appel à des méthodes (ou approches) d'analyse et de conception conçues spécialement pour ce type de projets.

Analyse : Etude, examen d'un objet, d'une situation pour en comprendre le fonctionnement dans un but d'amélioration, examen d'un objet existant.

Conception : Création d'un objet, d'un système : action qui donne naissance à quelque chose qui n'existe pas.

II- les méthodes de développement d'un SI :

L'objectif de ces méthodes est de présenter une démarche (ensemble d'étapes) et un ensemble de modèles permettant de mettre en place un nouveau système cohérent, pertinent, fiable et homogène. Elle est nécessaire pour formuler le problème informationnel et maîtriser la résolution.

Toute méthode met en œuvre 4 composants indissociables:

- **Des modèles :** Ensemble de concepts et règles destinés soit à expliquer et représenter les phénomènes organisationnels, soit à expliquer et à représenter les éléments qui composent le SI et leurs relations.
- **Un langage :** Ensemble de constructions qui permettent de décrire formellement les images du SI élaborées aux différents stades du processus de conception, éventuellement en faisant appel à des méthodes.
- **Une démarche** : C'est un processus opératoire par lequel s'effectue le travail de modélisation, de description, d'évaluation et de réalisation du S1.
- **Des outils** : Ce sont les outils logiciels supportant la démarche (outils de
- documentation, d'évaluation, de simulation, d'aide à la conception ou à la réalisation.

II-1- Une méthode pourquoi ?

+ **Maîtriser le développement des systèmes**
  - Contrôler les coûts, et les délais
  - Répondre aux besoins nés de l'évolution de l'entreprise
  - Intégrer les innovations techniques
  - Augmenter le service rendu par la structure.

+**Faciliter la communication**
  - Normaliser le vocabulaire
  - Homogénéiser la présentation de solutions
  - Améliorer l'expression des besoins

+**Assurer la pérennité des systèmes**
  - Apporter des aides à la maintenance
  - Améliorer les performances des systèmes.

II-2- Familles de méthodes :

II-2-1 les méthodes cartésiennes (fonctionnelles ou analytiques) :

elle associe au **paradigme cartésien , une approche fonctionnelle** de conception

le paradigme cartésien : met en œuvre le principe de Descartes (Diviser pour mieux résoudre) qui conduit l'analyse à décomposer l'ensemble de procédures de gestion de l'entreprise en applications indépendantes qui peuvent être étudiées séparément sans tenir compte des autres applications (c'est une démarche descendante top-down)

Exemples :

- Décomposition de l'université en facultés
- Dans une entreprise commerciale :
  - Gestion du personnel.
  - Gestion des clients.
  - Gestion des fournisseurs.
  - Gestion des stocks.

l'approche fonctionnelle : analyser et concevoir le système en se basant sur ses fonctions (des méthodes orientée traitements)

A-Avantages :

1. Simplicité de mise en œuvre.
2. Possibilité du traitement des applications en parallèle.
3. Facilité de maintenance.
4. Facilité d'estimation des coûts de fonctionnement.

B-Inconvénients :

1. Difficulté de mettre en pratique des entités indépendantes
2. Peut augmenter les coûts de développement
3. Problème d'arrêt de la décomposition
4. Pas de modification de la structure alors qu'elle peut être source de dysfonctionnement

parmi ces méthodes **:** ***la méthode SASD :Structured Analysis Structured Design***

II-2-2- les méthodes systémiques :

Constat : Une organisation ne peut pas toujours se décomposer en applications indépendantes.

La résolution des différents sous-problèmes indépendants n'implique pas forcément la résolution du problème global.

- l'approche systémique consiste donc à considérer les sous-systèmes aussi indépendants que possible et à les traiter en tenant compte **de leur interaction**.

Exemple :

Dans une faculté on peut considérer : La bibliothèque, la gestion du personnel, les départements…etc.

a- Comme des sources indépendantes (approche cartésienne)
b- Comme des sources en interaction (approche systémique).

Les méthodes systémiques sont entièrement centrées sur **la modélisation des données**. Elles considèrent l'entreprise comme un système à part entière. Qui se compose d'un système de pilotage, d'un système d'information et d'un système opérant. En outre, ces méthodes se caractérisent par la description des relations entre informations, une modélisation du domaine concerné de l'entreprise, une circulation des informations

Exemple de méthodes : Merise , Remora

A-Avantages

1. Meilleur prise en compte de la réalité
2. possibilité de remise en question de l'organisation existante
3. solution intégrée et coopérative

B-Inconvénients :

1. Plus complexe à mettre en œuvre
2. Plus difficile de traiter en parallèle.

II-2-3 méthodes orientées objets (OMT, Grady Booch, UML, …):

Un SI est considéré comme une collection d'objets qui se communiquent par envoi de message. Ces méthodes visent essentiellement à :

- identifier les objets du système, cette identification porte à la fois sur les attributs (données) décrivant les objets, sur les opérations (ou services) que chaque objet peut exécuter et sur la composition des objets
- identifier la communication entre les objets, cette identification consistent essentiellement à reconnaitre quelles opérations d'un objet sont utilisées

III- la Méthode Merise :

« **Méthode pour Rassembler les Idées Sans Effort** » ou « **Méthode d'Etude et de Réalisation Informatique de Systèmes d'Entreprise** » :
c'est une méthode systémique créée en 1979 par **Peter Chen et Hubert Tardieu** à l'issue d'un projet lancé par le ministère de l'industrie français dans le but de doter les entreprises et les administrations d'une méthode de conception qui leur permettre de réussir leurs projets dans les couts et les délais prévus.

MERISE admet une vision globale de l'entreprise, elle se caractérise par :

- **une séparation entre les données et les traitements**
- **traitement en profondeur de l'aspect organisationnel**
- **elle offre deux approches complémentaires de développement : approche par étapes et approche par niveaux**

III-1- Approche par étapes : on peut distinguer 6 étapes :

1- Le schéma directeur : Permet de définir d'une manière globale la politique d'organisation et d'automatisation du SI :
découper le SI en sous-ensembles homogènes appelés domaines. Le résultat de cette étape est **une définition précise des domaines** et **une planification de développement des applications qu'on doit réaliser pour chaque domaine**

2- Etude préalable ou étude d'opportunité (par domaine): Elle doit aboutir à une représentation du futur système (modèle de données et de traitements) .cette étape est réalisée en quatre phases :

- **Phase de recueil** : analyser l'existant (étude de procédures de travail, des moyens utilisés : documents, discussion avec les personnes concernées) afin de cerner les dysfonctionnements du système actuel.

- **Phase de conception** : modéliser le futur système en formalisant les orientations nouvelles en fonction des critiques formulées sur le système actuel

- **Phase d'organisation** : définir le futur système au niveau organisationnel (qui fait quoi)

- **phase d'appréciation** : établir les couts et les délais

3-L'étude détaillée (analyse fonctionnelle) : concevoir le nouveau système c'est à-dire les nouveaux documents, les structures de données, les traitements à effectuer, proposer plusieurs solutions pour la réalisation et faire un choix en fonction de possibilités de l'entreprise

4- **la réalisation** : consiste à détailler l'analyse fonctionnelle et l'adapter au matériel disponible pour que la programmation soit possible.
C'est :

- **la rédaction des programmes**
- **le test et la mise au point** : les programmes ne fournissent pas toujours les résultats espérés, il faut parfois reconsidérer certaines solutions définies au niveau de l'analyse organique, c'est la phase du test.

5- **Mise en œuvre** : exécuter toutes les actions qui permettent de lancer le système auprès de l'utilisateur :
formation, installation et initialisation des données

6- la maintenance : elle permet de modifier les applications et les mettre à niveau jusqu'à leur mort

III-2) Approche par niveaux d'abstraction : MERISE distingue trois niveaux de description d'un SI. à chaque niveau est associé un couple de modèles (données- traitements) avec un formalisme de représentation pour chaque modèle

- **le niveau conceptuel** : consiste à concevoir le SI sans envisager les notions liées à l'organisation, il consiste à se poser la question **QUOI** : quoi faire et avec quelles données.

les deux modèles associés à ce niveau :

le MCD : modèle conceptuel de données

le MCT : modèle conceptuel des traitements

- **le niveau logique (ou organisationnel)** : consiste à intégrer à l'analyse les critères liés à l'organisation (**notion de lieu, temps et d'acteurs : on se pose les questions : qui, où, quand**) .

Les deux modèles associés :

Le MLD : modèle logique des données
Le MOT : modèle organisationnel des traitements

- **le niveau physique** : consiste à définir les choix techniques du système (mode de stockage pour les données, découpage des programmes).

Il consiste à se poser **la question : comment ?**
les deux modèles associés sont :

le MPD : modèle physique des données
le MOPT : modèle opérationnel des traitements

NIVEAU	**DONNEES**	**TRAITEMENTS**	**CHOIX PRIX EN COMPTE**
Conceptuel	Modèle Conceptuel des Données (MCD)	Modèle Conceptuel des Traitements (MCT)	Choix de gestion Quoi ?
Organisationnel	(MLD) Modèle Logique des Données	Modèle Organisationnel des Traitements (MOT)	Choix d'organisation Qui ?, Où ?, Quand ?
Physique	Modèle Physique des Données (MPD)	Modèle Physique des traitements (MPT)	Choix techniques Comment ?

Les modèles de la méthode MERISE

IV-Le modèle conceptuel des données(MCD) :

Permet de décrire les données et de les représenter, ainsi que les liens qui existent entre elles, grâce à un formalisme simple et efficace :

Le formalisme « entité-association ».

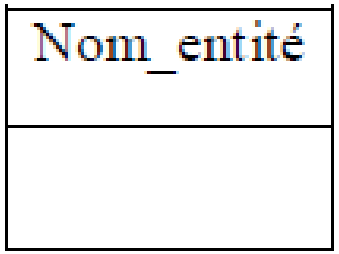
IV- 1- Concepts de base du formalisme 'Entité-Association' : il a été élaboré par CHEN en 1976 . C'est une représentation naturelle et efficace du monde réel . il est bâti autour de trois concepts :

entité, association et propriété

1-L'entité : une entité est une représentation dans un SI d'un objet matériel ou immatériel pourvue d'une existence propre,

exemple : étudiant, client, produit….

Formalisme :



2-Association : une association représente un lien entre les entités. elle est dépourvue d'existence propre. son existence est liée à l'existence des entités qu'elle met en interaction

exemple : un étudiant **appartient** à une seule section

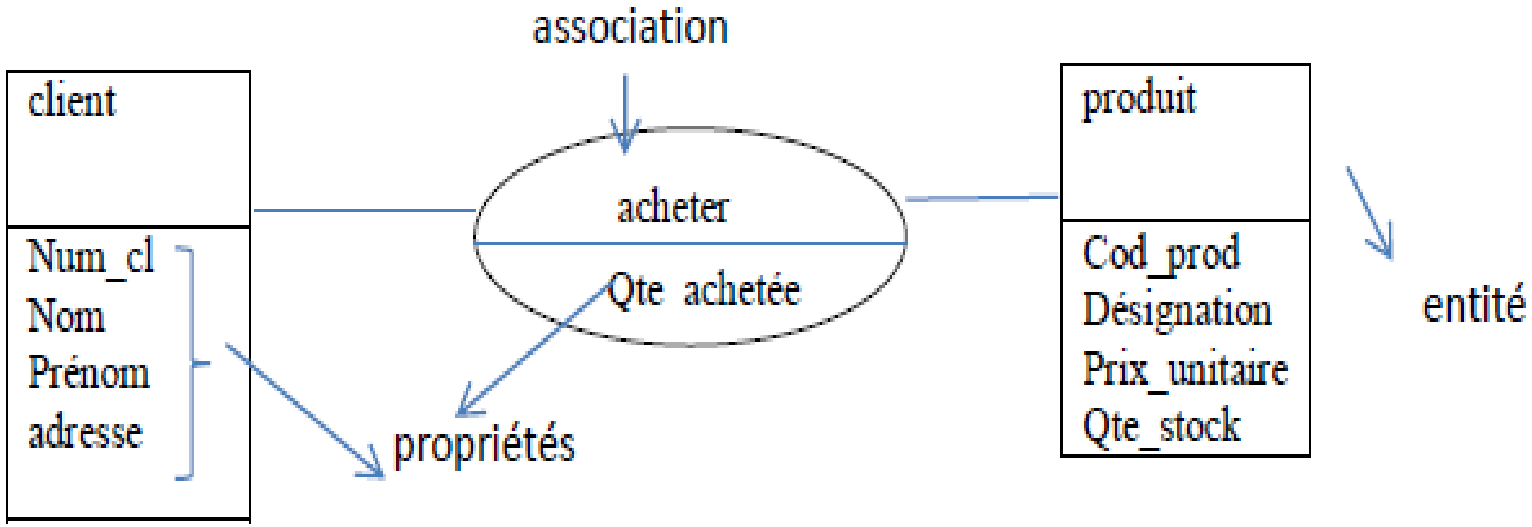
3- **Propriété** : une propriété est une donnée élémentaire qui caractérise une entité ou association

exemple : cod-etudiant, prenom, dat-nais

Types de propriétés : une propriété peut être :

Un code : information représentatif de l'objet
Un Libellé : donnée alphanumérique qualitative : nom, prénom,….
Un montant : donnée numérique quantitative QTE, prix, âge

Formalisme graphique :

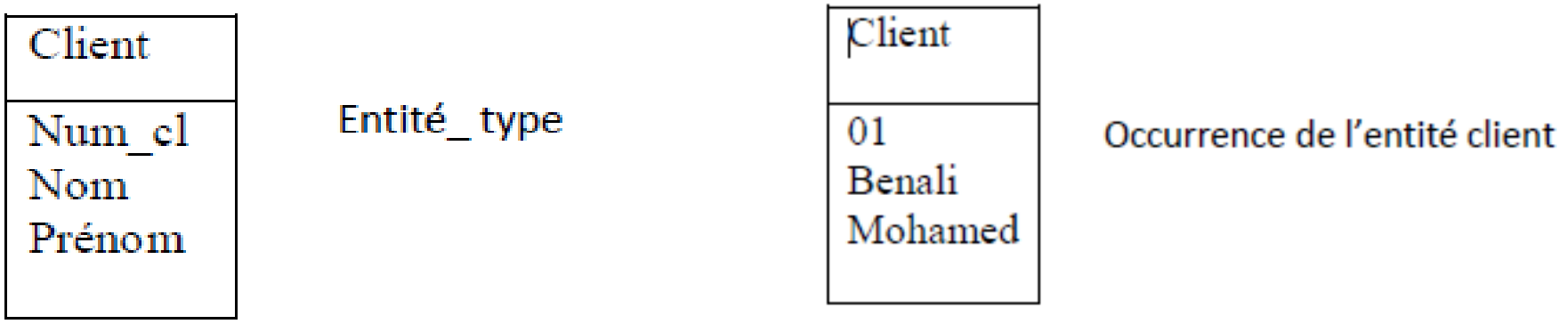


4-Types et occurrence

Type : un ensemble d'éléments qui ont les mêmes caractéristiques

Occurrence d'un type : est un élément particulier appartenant à cet ensemble

Exemple :



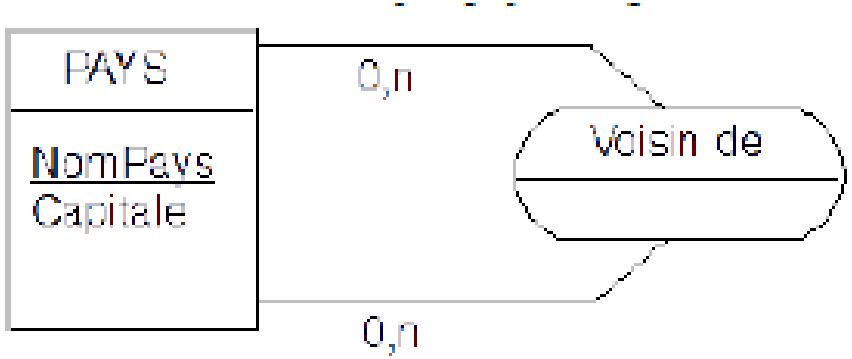
5-Caractéristique d’une association :

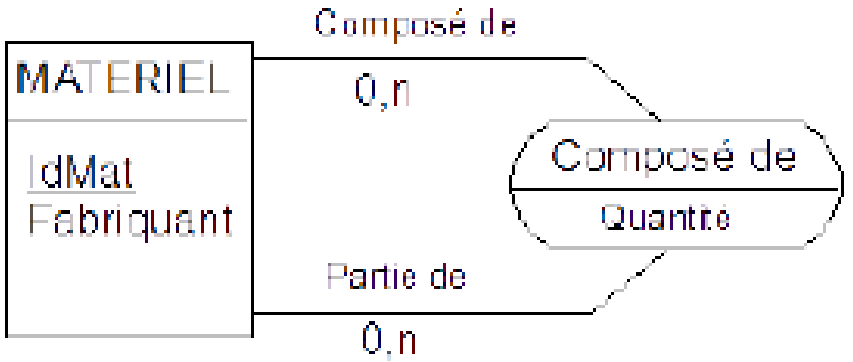
5-1-Dimension d’une association : c’est le nombre d’entités participant à cette association

Exemple précédent : dimension= 2

-une association qui lie **une entité** à elle-même est dit **unaire (réflexive)**

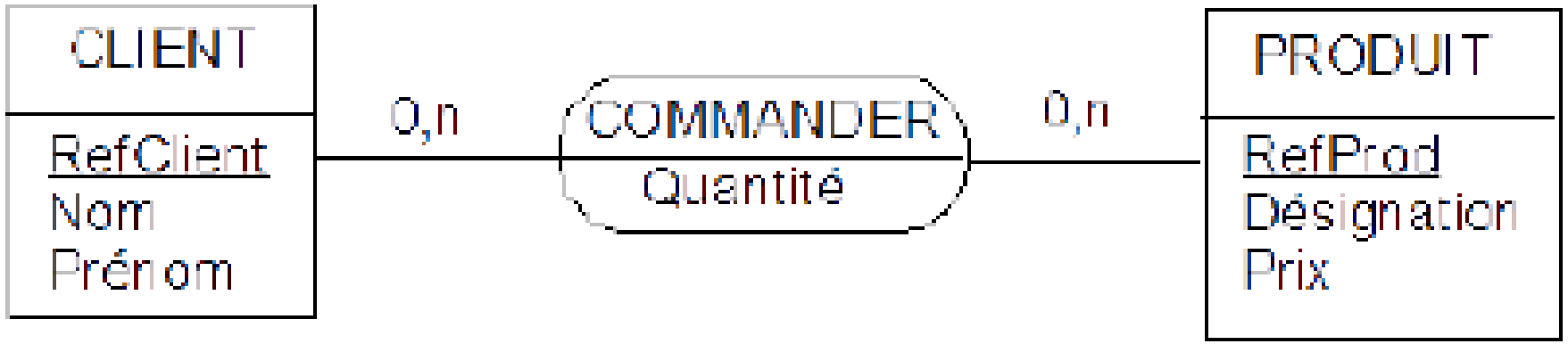
Exemples :





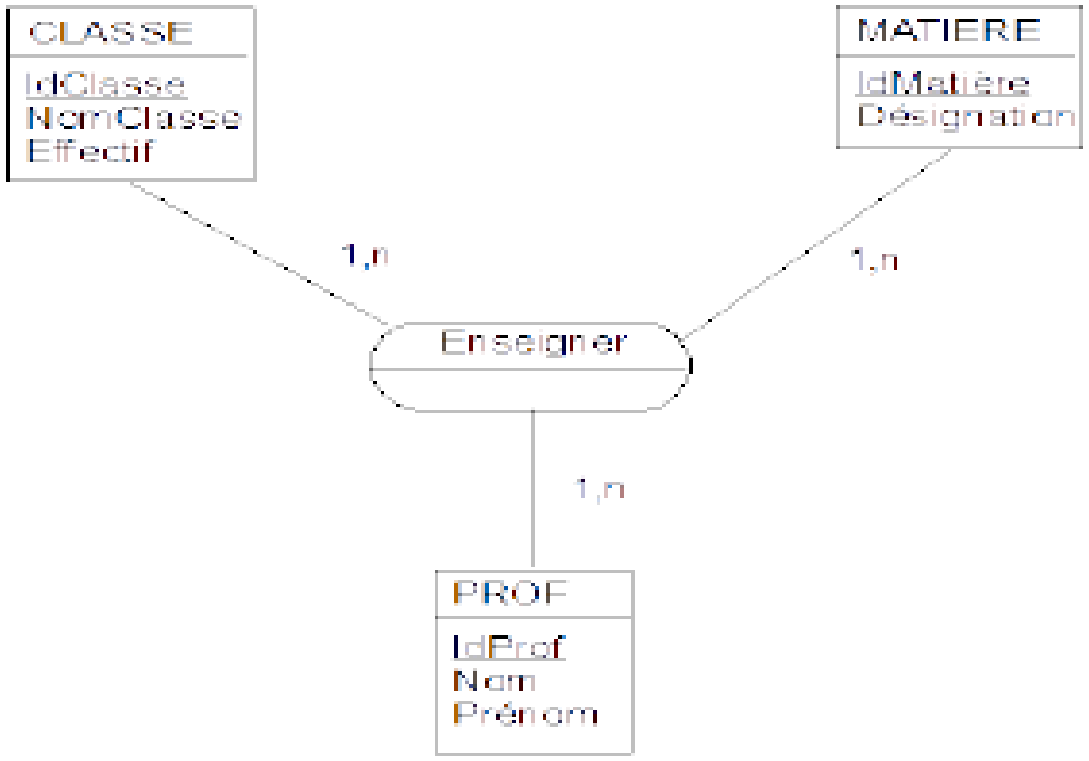
- **une association qui lie deux entités est dite binaire**

Exemple :



- **une association qui lie trois entités est dite ternaire**

Exemple :



5-2-les Cardinalités : les cardinalités d'une entité E1 par rapport à une association avec une entité E2 expriment le nombre d'occurrences de E2 qu'on peut associer à une occurrence d'E1.

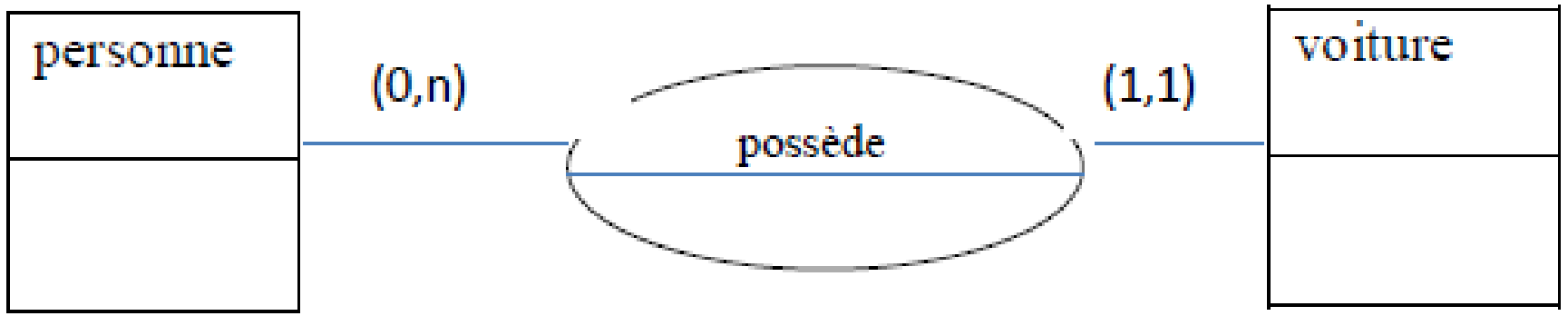
Elles sont exprimées par un couple de valeur (X,Y) tel que :

X est le nombre minimum d'occurrence de E2 que l'on peut associer à une occurrence de E1 : ***Cardinalité minimale***

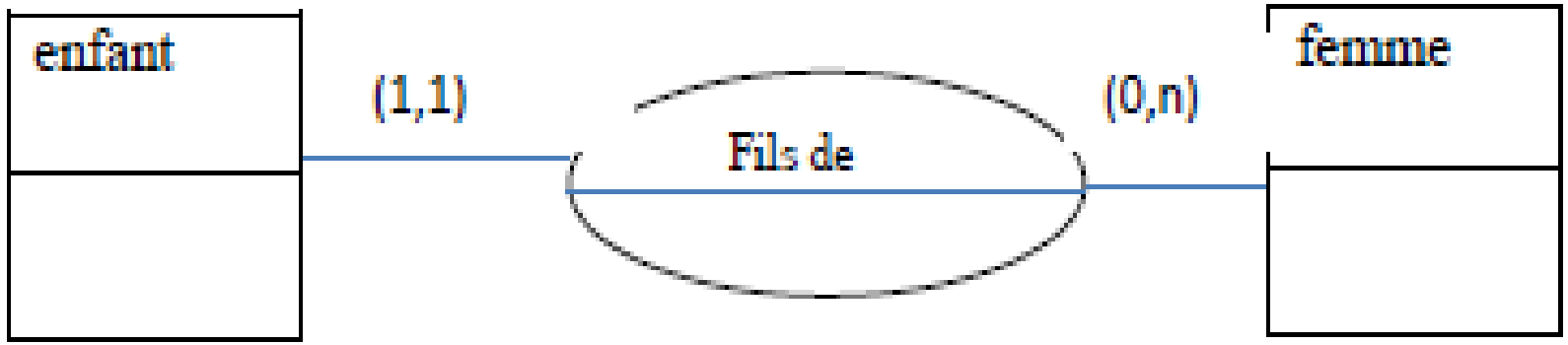
Y est le nombre maximum d'occurrences d'E2 que l'on peut associer à une occurrence d'E1 : ***Cardinalité maximale*.**

Les valeurs possibles des cardinalités : ***(0,1),(0,n),(1,n),(1,1)***

Exemple1



Exemple2



6-Identifiant d'une entité : l'identifiant d'une entité est une propriété particulière de l'entité qui permet d'identifier chaque occurrence de cette entité de **manière unique**

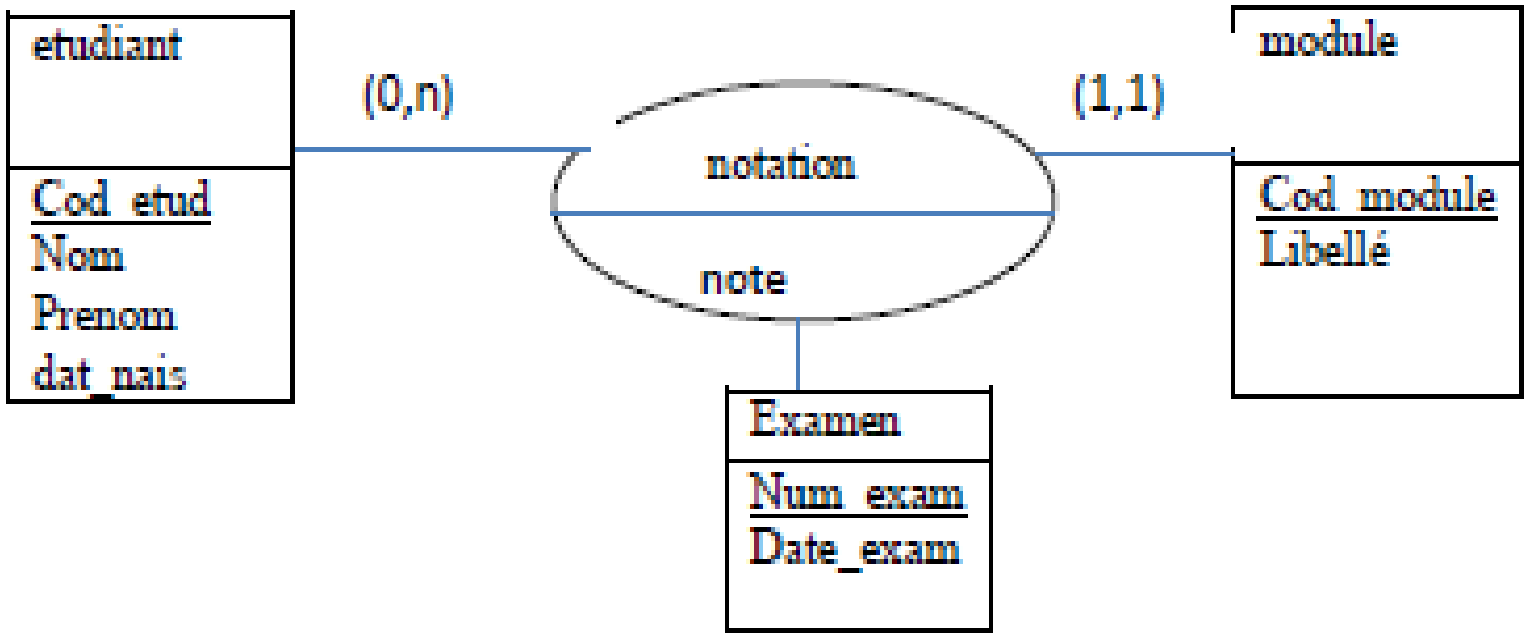
Exemple : pour l'entité Etudiant : l'identifiant c'est : **num_etudiant**

Remarques :

- l'identifiant peut être formé de plusieurs propriétés
-Chaque entité doit posséder un identifiant, si aucun identifiant n'existe pas dans la liste des propriétés , il faudra le créer .

7- Identifiant d'une association :L'identifiant d'une association est obtenu par la concaténation des identifiants des entités qui participent à cette association

Exemple :



L'identifiant de l'association 'notation' est :

cod_etud+cod_module+num_exam

IV-2 -Règles relatives au modèle de données (au MCD):

1-Les dépendances fonctionnelles

1-1 Définition

Les propriétés **A** et **B** sont reliées par une **dépendance fonctionnelle** (df):

A → **B**, si la connaissance de la valeur de a détermine une et une seule valeur de b.

Exemple : **Num_Etud** → **Nom_Etud**. (on dit **Num_Etud** **détermine** **Nom_Etud** **ou** **dépend fonctionnellement de** **Num_Etud**)

Mais l'inverse nest pas juste :

Nom_Etud ne détermine pas Num_Etud

Remarque:

La dépendance fonctionnelle peut porter sur la concaténation de plusieurs propriétés. **Exemple** :

NumCommande + **RéfProduit** → **QtéCommandée.**

1-2 Dépendance fonctionnelle élémentaire

Il y a une dépendance fonctionnelle **élémentaire** entre les propriétés A et B si

A → B et aucune partie de A ne détermine B

Exemple :

Num_Etud + Nom_Etud → Adr_Etud n'est pas élémentaire puisque la connaissance de Num_Etud suffit pour déterminer Adr_Etud.

Par contre, Num_Etud → Adr_ Etud est élémentaire.

-3 Dépendance fonctionnelle directe

Les propriétés A et B sont reliées par une **dépendance fonctionnelle directe si :**

A → B et s'il n'existe pas une propriété c telle que a → c et c → b

Autrement dit, on élimine toute transitivité.

Exemple :

NumProf → CodeMatière

CodeMatière → NomMatière

NumProf → NomMatière.

Les deux premières dépendances fonctionnelles sont **directes**, mais la troisième ne l'est pas : c'est une dépendance transitive

NumProf → CodeMatière → NomMatière

-2-*Règles de normalisation des entités et des associations :*

La normalisation du modèle permet d'enlever toute anomalie qui peut rendre le modèle incorrect (difficile, ou même impossible, à implémenter et à gérer). L'anomalie la plus courante est la redondance de l'information.

La redondance dans le modèle peut engendrer :

- Des erreurs lors de mise a jour : puisque l'information est répétée, il est possible qu'au moment de mise a jour, quelques copies soient mises a jour tandis que d'autres ne le seront pas (garde l'ancienne valeur).
- Lectures incohérentes : si des erreurs de mise à jour sont survenues, il est possible de lire des valeurs différentes (incohérentes) pour la même information.

- **R1:** Toutes les propriétés des entités ou des associations doivent être **élémentaire** (non décomposable a d'autre propriétés) et il existe un **identifiant** pour chaque entité

Exemple : si on considère la propriété **Adresse** composée de : nom_ville + nom rue+ num rue alors : **Adresse** n'est pas élémentaire et on doit la remplacer par ses propriétés : nom_ville, nomrue, num_rue

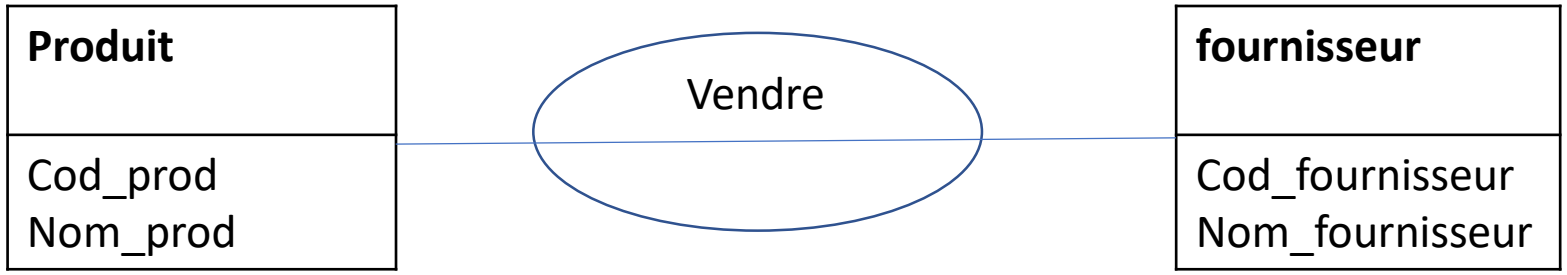
- **R2:** dans une entité, les propriétés doivent dépendre de l'identifiant par une dépendance fonctionnelle **élémentaire et directe** : **pas de transitivité** et **aucune partie de l'identifiant ne détermine les autres propriétés**

Exemple1

Dans l'entité suivante la dépendance cod_prod → nom_fournisseur **n'est** pas directe car il y a la DF code_fournisseur → nom fournisseur

donc il faut corriger le modèle comme suit

produit
Code_prod Designation Code_fournisseur Nom_fournisseur



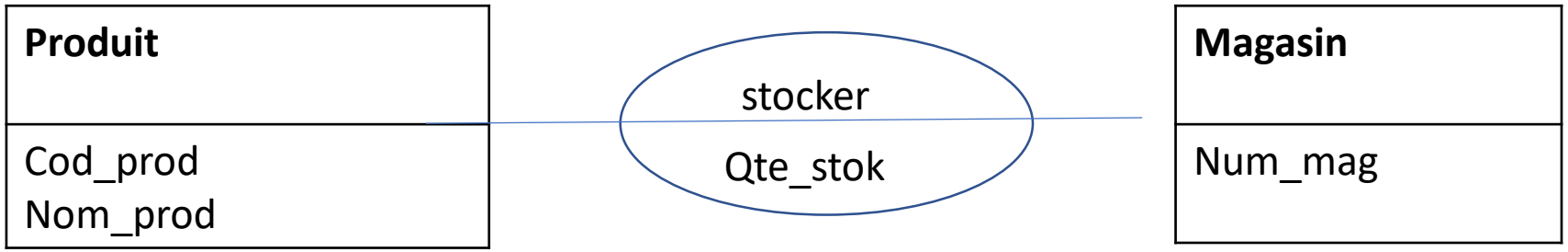
Exemple2

Dans l'entité suivante la dépendance cod_prod,num_mag → nom_prod

n'est pas élémentaire car il y a la DF code_prod → nom_prod

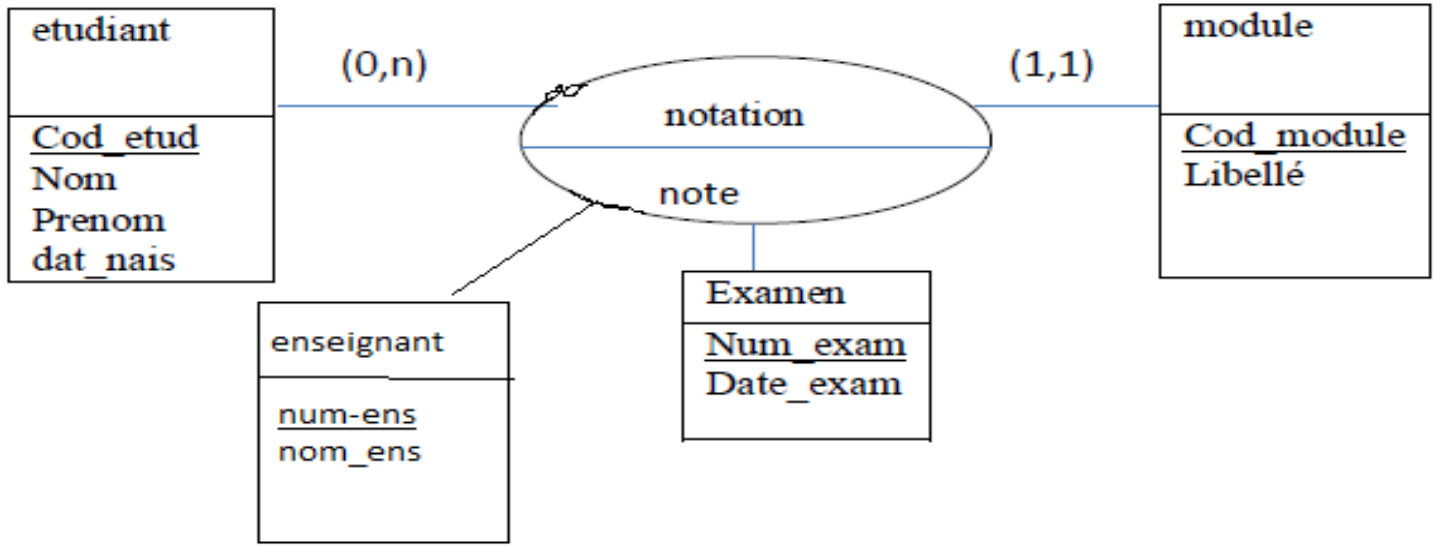
donc il faut corriger le modèle comme suit

Produit
Code_prod Num_mag Nom_prod Qte_stok

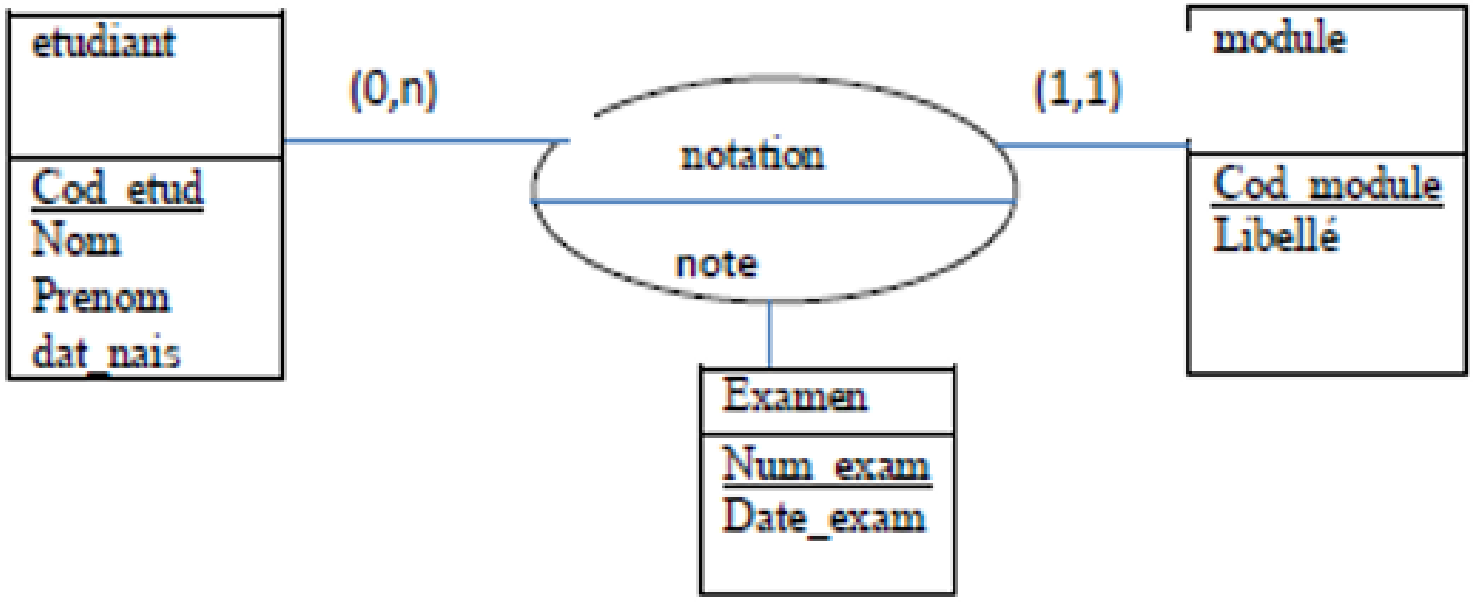


- R3: Les propriétés d'une association doivent dépendre fonctionnellement de tous les identifiants des entités concernées par l'association et non d'un sous-ensemble de ces identifiants.
- **Exemple :** dans le modèle suivant**:**

La note dépend seulement de code_etud, num_exam, cod_mod et ne dépend pas de l'enseignant (num_ens) donc il faut supprimer l'entité enseignant de l'association notation



Donc le modele correct

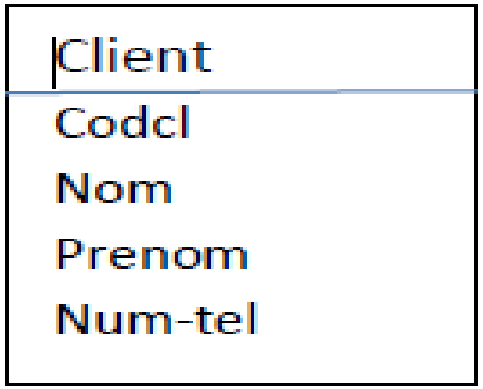


-2- règles de vérification des entités et des associations

1- Dans toute occurrence d'entité ou d'association, il ne doit y avoir qu'une seule valeur de chaque propriété. Une propriété ne doit pas avoir plusieurs valeurs pour une même occurrence de l'entité ou de l'association : pas de propriétés répétitives .

Exemples

1) Soit l'entité suivante

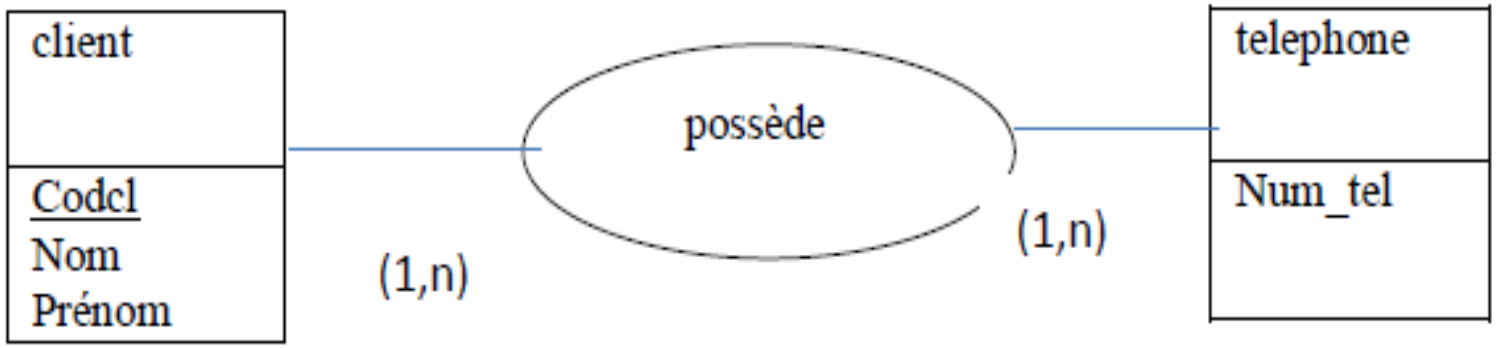


si on a plusieurs numéros de téléphone pour un client on dira que Num_tel est une propriété répétitive . pour corriger la représentation on a deux possibilité

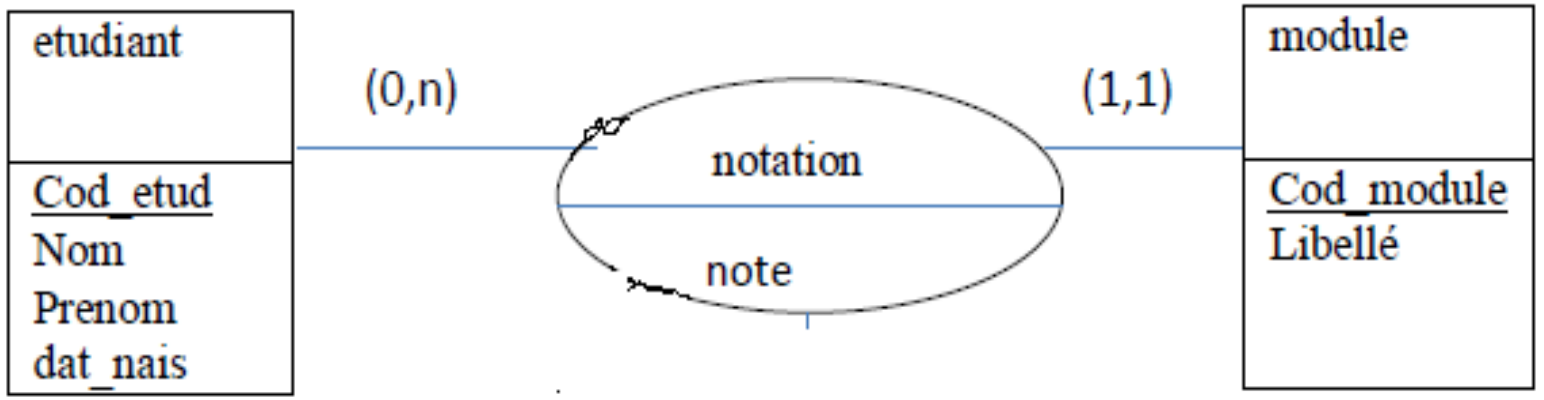
1 er cas : le nombre de Num_tel est connu (par exemple = 2)
on doit les représenter par deux propriété Num_tel1 et num_tel2

Client
Codcl
Nom
Prénom
Num_tel1
Num_tel2

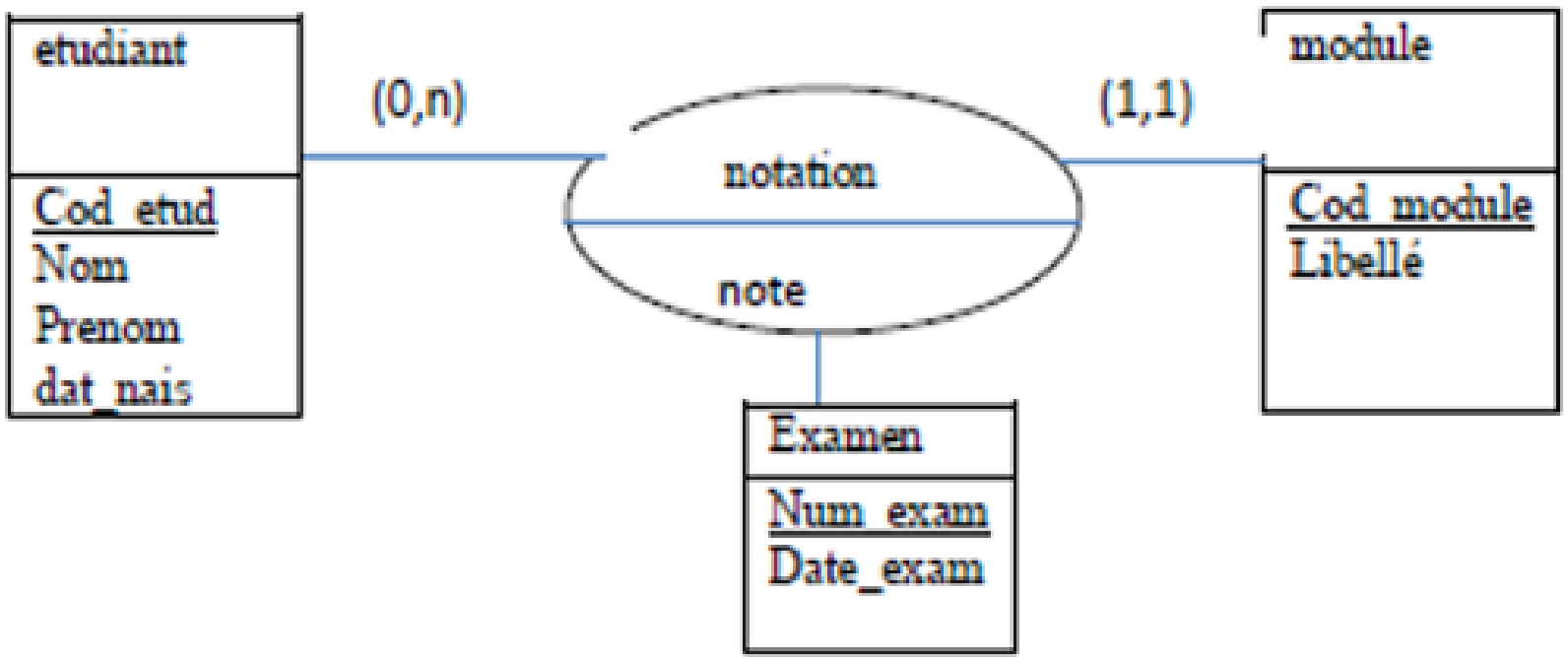
2eme cas : le nombre de Num_tel est variable : le modele est corrigé comme suit



Exemple 2 soit le modèle suivant:



pour le meme etudiant et pour le meme module on a plusieurs valeurs de la propriete note donc
Ce modele est faux , il faut ajouter une troisième entité qui nous permet de determiner une seule valeur de la note c'est l'entité examen


etudiant
Cod_etud
Nom
Prenom
dat_nais
(0,n)
notation
note
(1,1)
module
Cod_module
Libellé
Examen
Num_exam
Date_exam

-3 Règles de décomposition des relations :

La décomposition consiste à remplacer une relation de dimension $n > 2$ en plusieurs relations de dimensions plus petites en utilisant les dépendances fonctionnelles présentes sur la relation

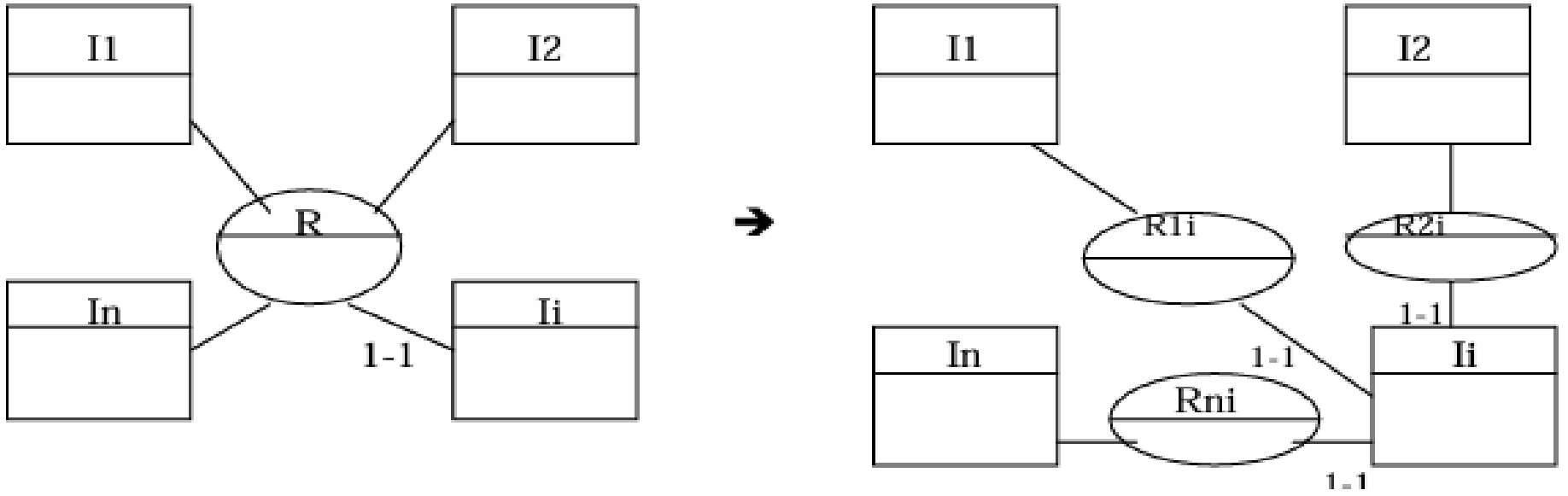
La décomposition est un processus réversible qui vise à simplifier le modèle conceptuel sans en modifier le sens. Elle permet de diminuer la dimension d'une relation-type par l'utilisation des contraintes d'intégrité définies sur les relations-type.

On distingue deux sortes de décomposition : endogène et exogène.

1.La décomposition endogène :

1.1- en utilisant les cardinalités individuelles : soit une relation R de dimension n et de collection (E1, E2 , …., En) et une entité-type Ei de cette collection telle que sa cardinalité individuelle dans R soit de (1-1). La décomposition endogène en utilisant la cardinalité individuelle (de Ei) permettra de transformer la relation R de dimension n en (n – 1) relations:

Exemple :

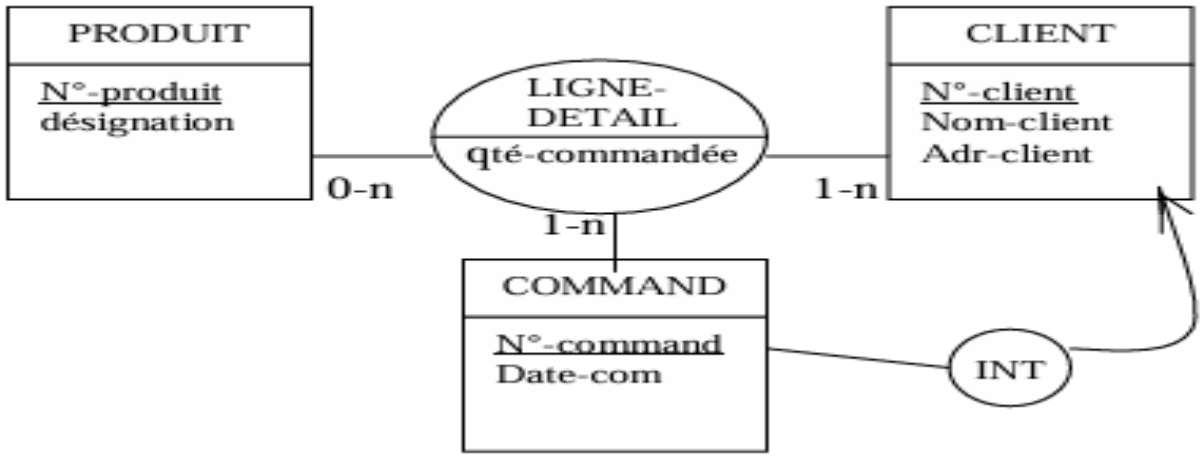


- en utilisant les CIFs :

contraintes d'intégrité fonctionnelles (CIF) : dans une relation-type, la CIF indiqiue l'existence d'une dépendance fonctionnelle entre un sous-ensemble de la collection et un individu-type de la collection

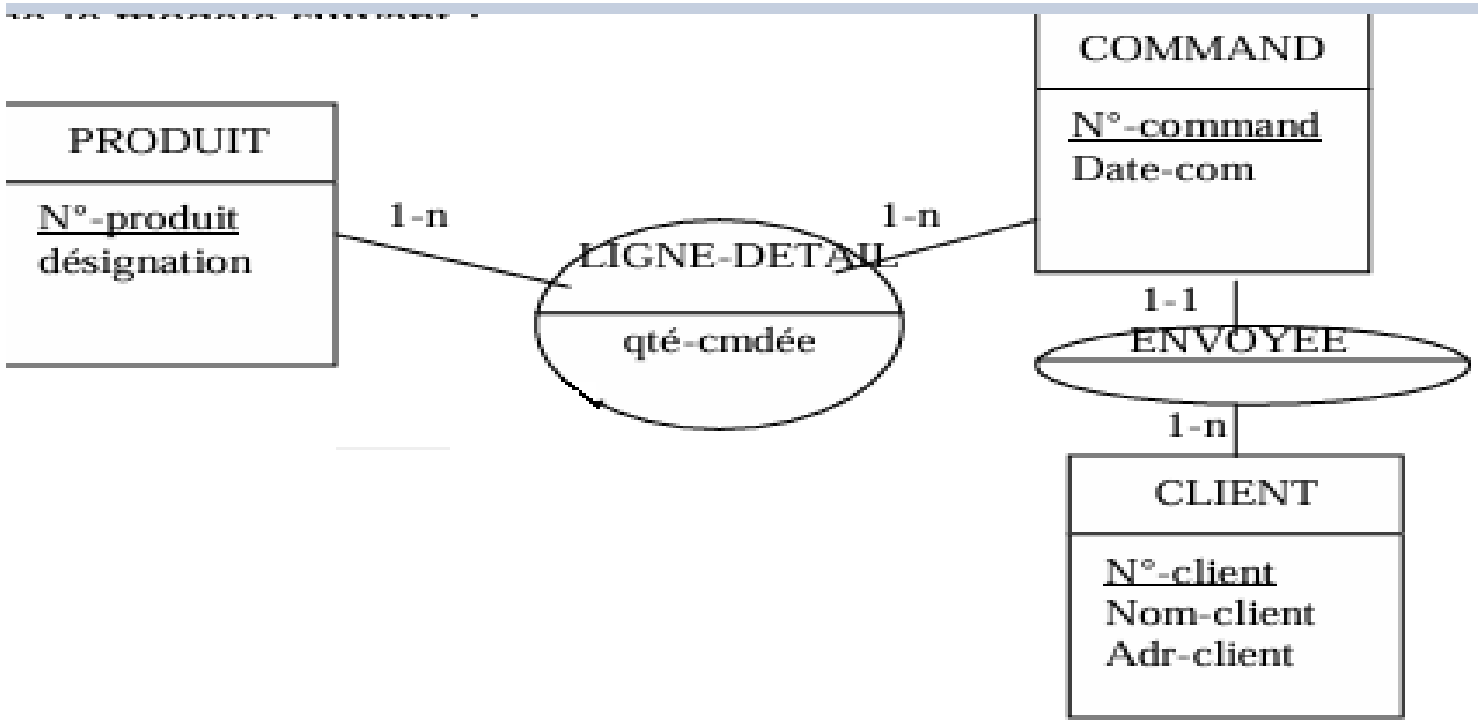
Soit une relation-type R de dimension n, de collection c et une CIF « INT » définie sur R de sous-collection c' avec dimension de c' = m. La décomposition endogène de R en utilisant « INT » permet d'obtenir une relation type R' de dimension (n-1) et une relation-type R'' de dimension (m+1).

Exemple :



La CIF « INT » de COMMANDE vers CLIENT exprime qu'une commande ne peut être envoyée qu'à un seul client.

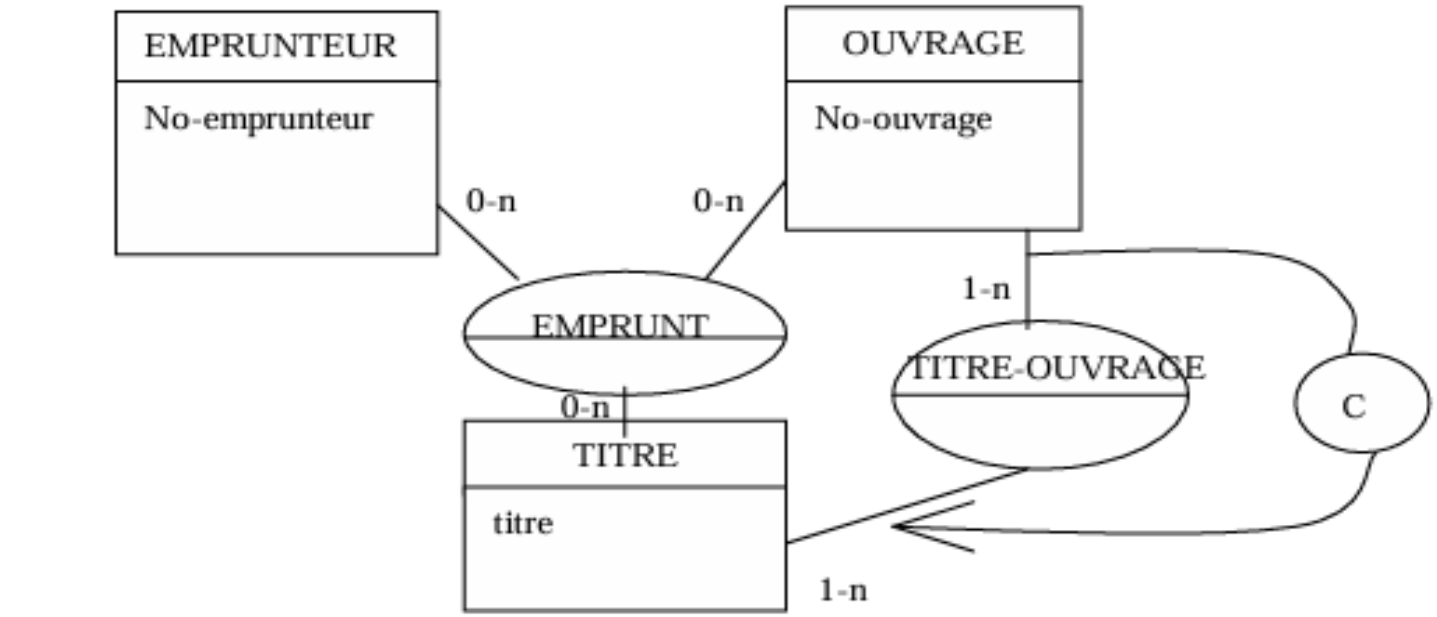
Après la décompositions le modèle devient:



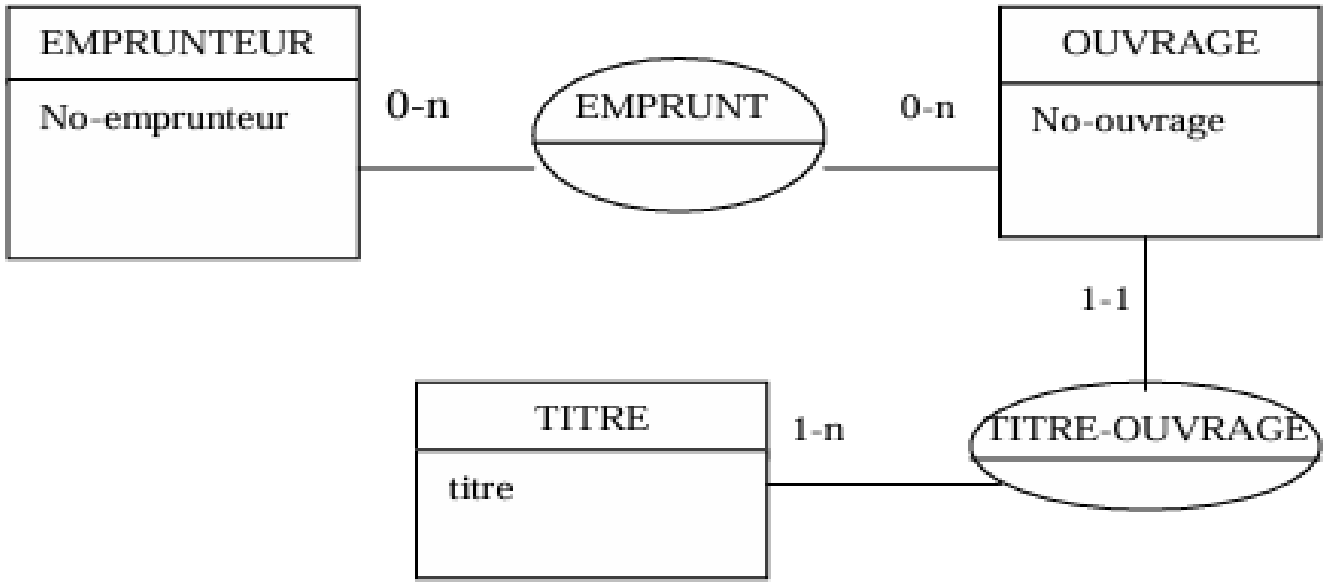
La décomposition exogène : d'une relation-type R permet de diminuer la dimension de R en utilisant des CIFs définies sur une relation-type R' différente de R à condition que :

- la CIF porte sur entités communs à R et R', et qu'elle porte sur les mêmes occurrences

Exemple :



La CIF C exprime qu’un ouvrage ne possède qu’un et un seul titre.
Le modèle décomposé sera :



IV-3- Démarche de construction du MCD :

1. Établir la liste des données recensées : cette liste est appelée ***dictionnaire de données***. Le formalisme du DD est le suivant :

Code	Désignation	Type	Taille	Observation
Code_F	Numéro du fournisseur	N	3	
Nom_F	Nom du fournisseur	A	20	
Adr_F	…	..	..	
Date_Cde	Date de la commande	Date	..	
. . . .	…	..	..	

2. **Épurer le dictionnaire de données** : ne garder que les données utiles pour le SI. De ce fait,

Les synonymes, les polysémies, les données calculées et les données concaténées doivent être supprimées.

3 Établir le graphe des dépendances fonctionnelles : à partir des règles de gestion.

a. Il ne faut garder que les dépendances fonctionnelles élémentaires directes,

b. S'il reste des propriétés isolées, proposer des identifiants qui permettent de les déterminer

4. Construire le MCD :

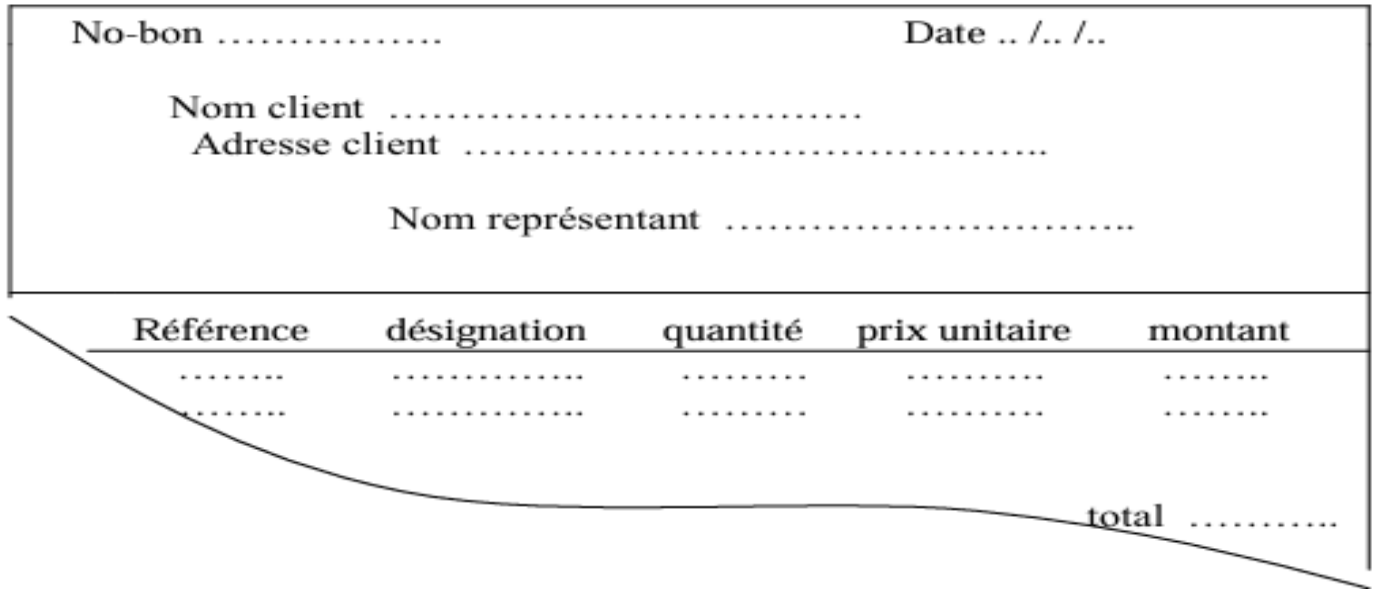
a. Les arcs terminaux d'une même origine constituent une Entité

b. Les arcs restants qui relient deux ou plusieurs entités traduisent des relations entre
ces entités,

c. Tenir compte des règles de gestion qui ne sont pas encore couvertes (celles qui n'expriment pas de dépendances fonctionnelles),

d. Normaliser, vérifier et décomposer le MCD brut

Soit l'exemple suivant :



a) recueil des informations

rassembler les exemplaires de tous les documents etles regles de gestion;

R.G.1 : un client peut passer une, plusieurs ou aucune commande ;

R.G.2 : une commande peut concerner un ou plusieurs produits ;

R.G.3 : une commande est passée à un représentant qui n'est pas toujours le même pour un client donné.

b) constitution du dictionnaire de données

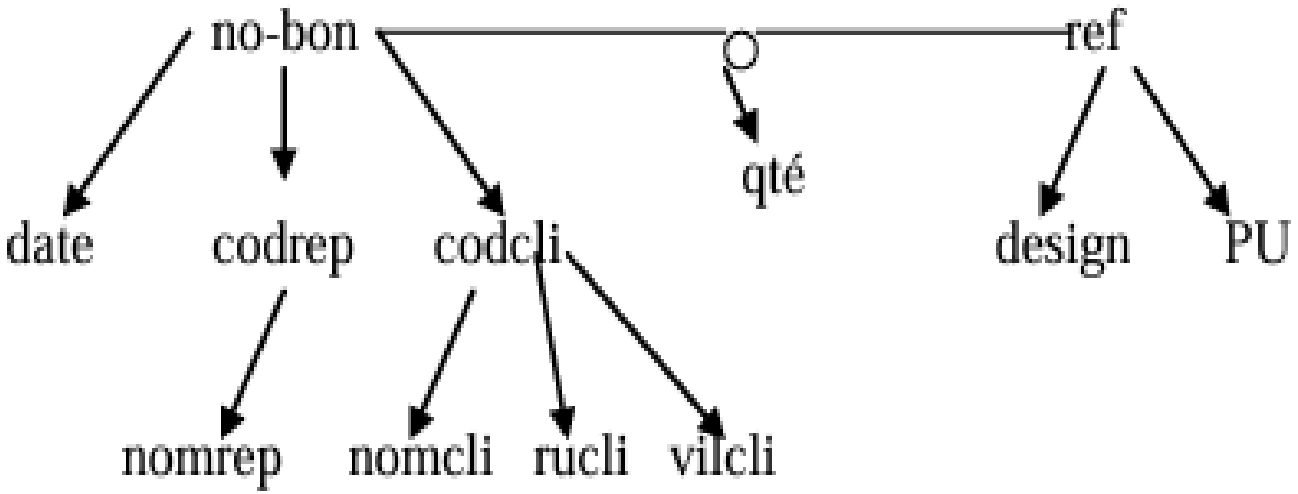
NOM	SIGNIFICATION	TYPE	LONGUEUR	NATURE		REGLE DE CALCUL OU INTEGRITE
				E/CO/CA	M/SIG/SIT	
No-bon	N° de bon de cmde	N	4	E	M	
Date	Date cmde	N	6	E	M	JJMMAA
Cocli	Code client	?	?	E	SIG	A créer
Nomcli	Nom client	A	30	E	SIG	
Adresse	Adresse client	AN	60	CO	SIG	Rue+ville
montant	Montant ligne	N	8	CA	M	P.U x QTE
…..	…….	…..	……	…….		……

c- épuration du dictionnaire de données

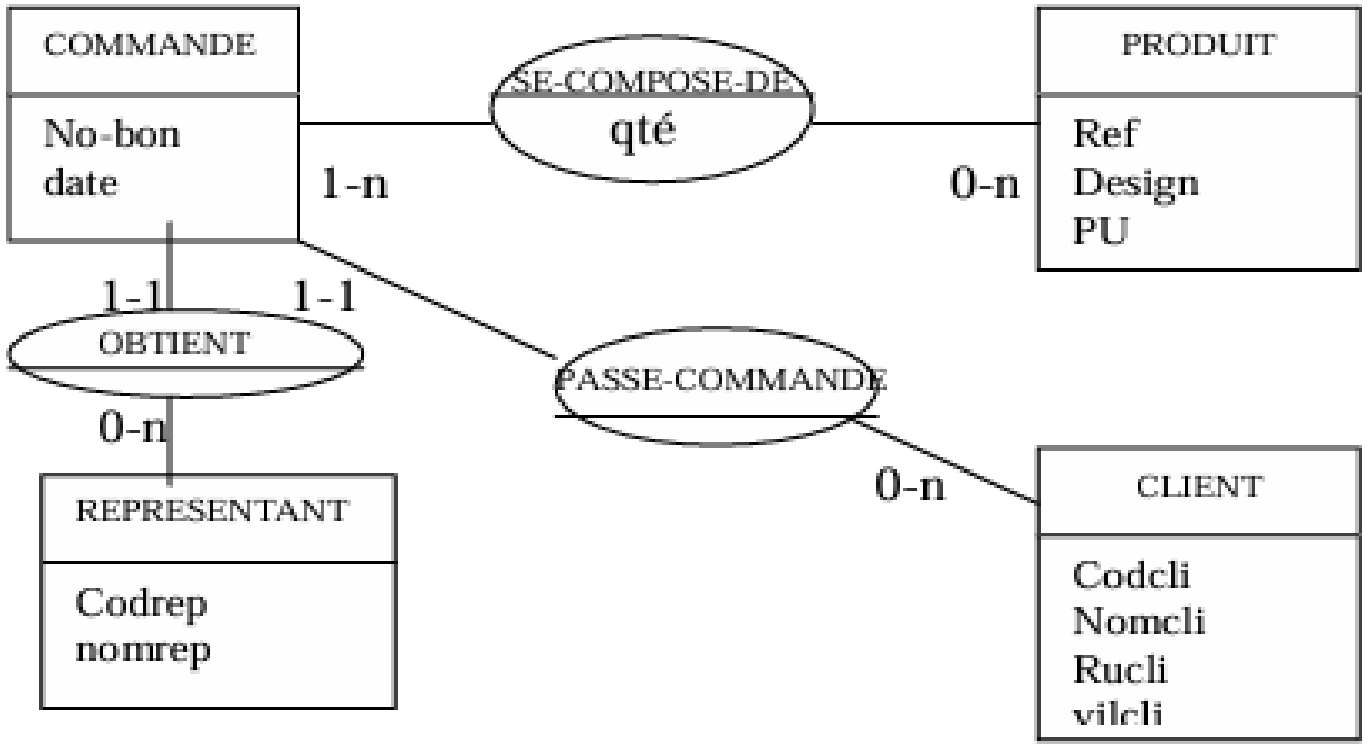
Il faut alors éviter les cas suivants :

- Synonymes : deux signifiants pour un même signifié, exemple : n°-client et code-client.
- polysèmes : un signifiant pour deux signifiés, exemple : nom pour nom du client et nom du fournisseur.

D- graphe de dépendances fonctionnelles



e) Etablissement du MCD



V- Le MCT : modèle conceptuel des traitement

La modélisation des traitements consiste à décrire tous les traitements qui sont effectués sur les données, et par quoi sont-ils déclenchés ? et que produisent comme résultats. On aboutira ainsi à un modèle de traitement. il décrit les traitements sans prendre en considération les aspects organisationnels
(qui fait le travail?, Où s'effectue ? quand et comment ?) .
Ce modèle répond à la question : Quoi ? (quoi faire ?)

V.1 Exemple

Procedure: gestion de promotion

Dans une grande administration les demandes de promotion sont traitées selon les règles suivantes :

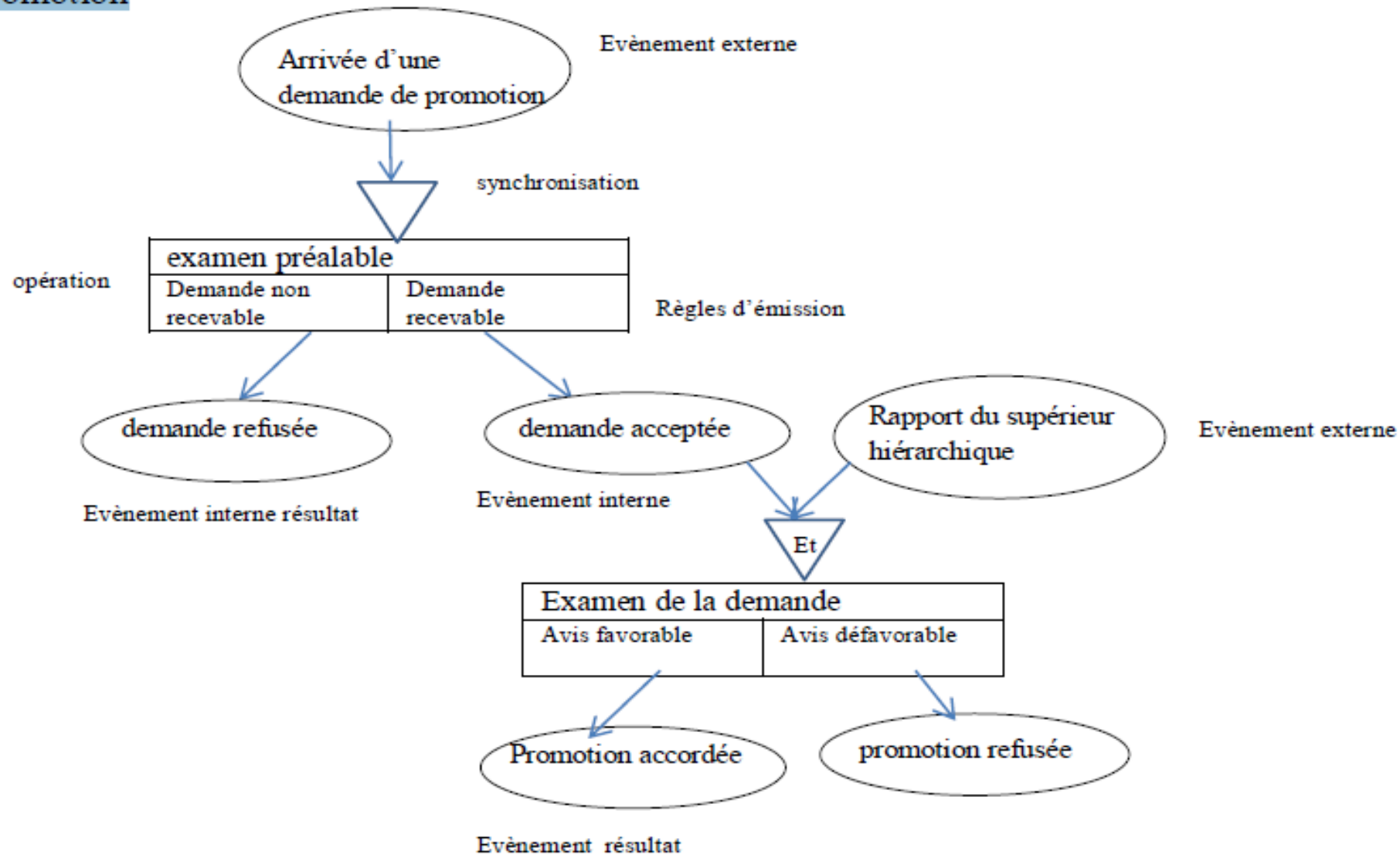
R1 : toute demande de promotion doit subir un examen préalable permettant de déterminer si elle est recevable ou non

R2 : l'examen du dossier d'une demande recevable ne peut se faire qu'après rapport du supérieur hiérarchique

R3 : après examen du dossier par l'autorité compétente la promotion est accordée ou refusée

Le MCT correspondant :

Processus : promotion



V-2 Les concepts de base :

1- ***L'évènement*** : c'est un fait qui génère une réaction de la part du SI sous forme d'opération. L'évènement peut être externe au domaine étudié ou interne au SI. Il peut être temporel : c'est-à-dire lié à des dates qui rythment l'exécution de certains traitements.

2- ***L'opération*** : c'est un ensemble d'actions accomplies par le SI en réaction d'un évènement ou à une conjonction d'évènements et non interruptibles par un évènement externe

3- ***Le résultat*** : un résultat peut être un document, un message externe, un nouvel état du SI créé par une opération qui peut lui-même jouer le rôle d'évènement

4- ***La synchronisation*** : c'est une condition booléenne (et/ou) traduisant les règles de gestion que doivent respecter les événements pour déclencher une opération.

5- ***Règles d'émission*** : expriment les conditions de sortie des résultats .on peut utiliser l'opérateur NON pour exprimer la négation d'une condition.
6- ***Le Processus*** : c'est un enchainement synchronisé d'opérations au sein d'un même domaine, généralement déclenché par un evt externe.

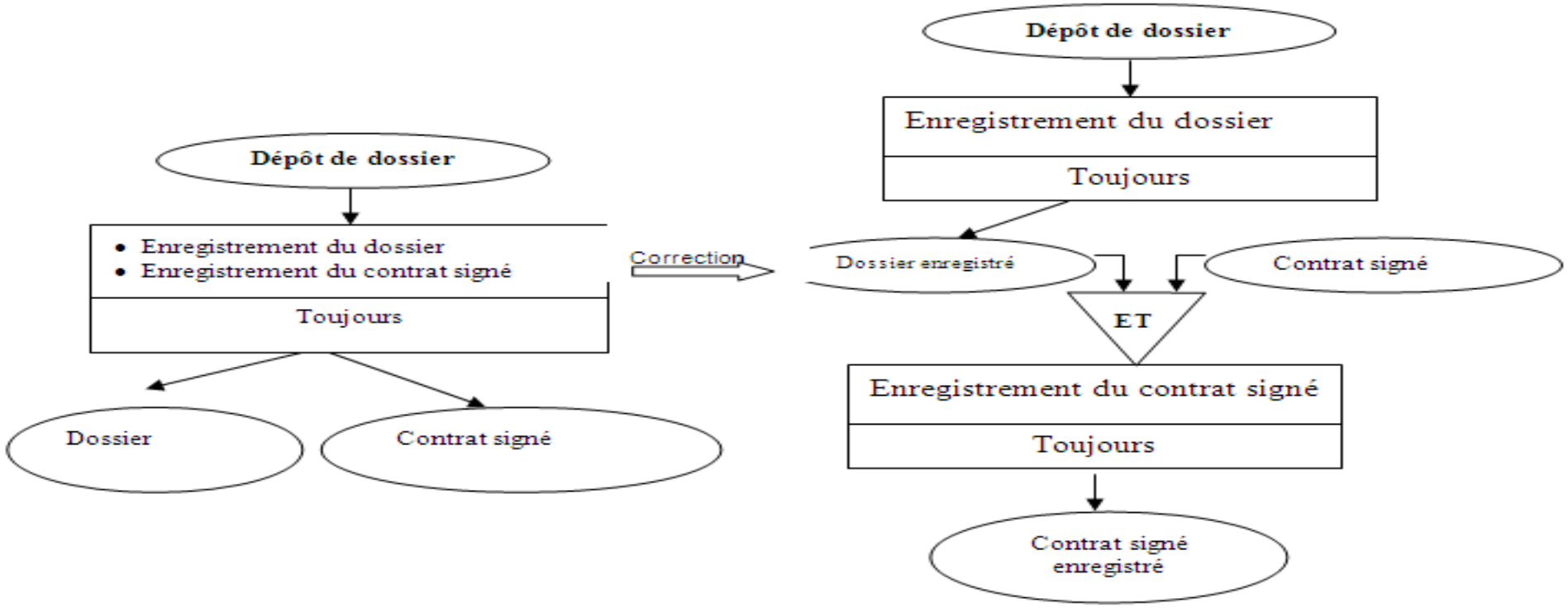
V-3 - Etablissement d'un MCT :

1- Réaliser le graphe de flux permettant de déterminer les évènements
2- pour chaque évènement, recenser les opérations déclenchées et les évènements internes produits.
3- Regrouper dans une même opération tous les traitements qui ont les mêmes déclencheurs dans une unité de temps avec la même synchronisation.

V-4-Règles relatives au MCT : pendant la construction du MCT, il faut respecter les règles suivantes :

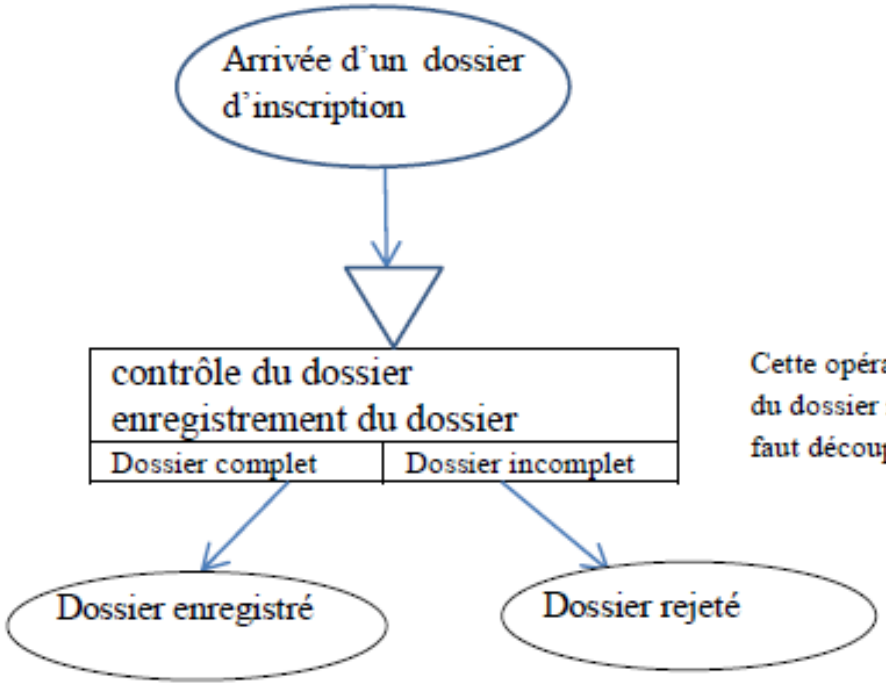
R1 : ***La non-interruption*** : une opération est une suite non interrompue de traitements. Toute interruption due à un évènement externe de processus provoque un découpage de l'opération

Exemple:



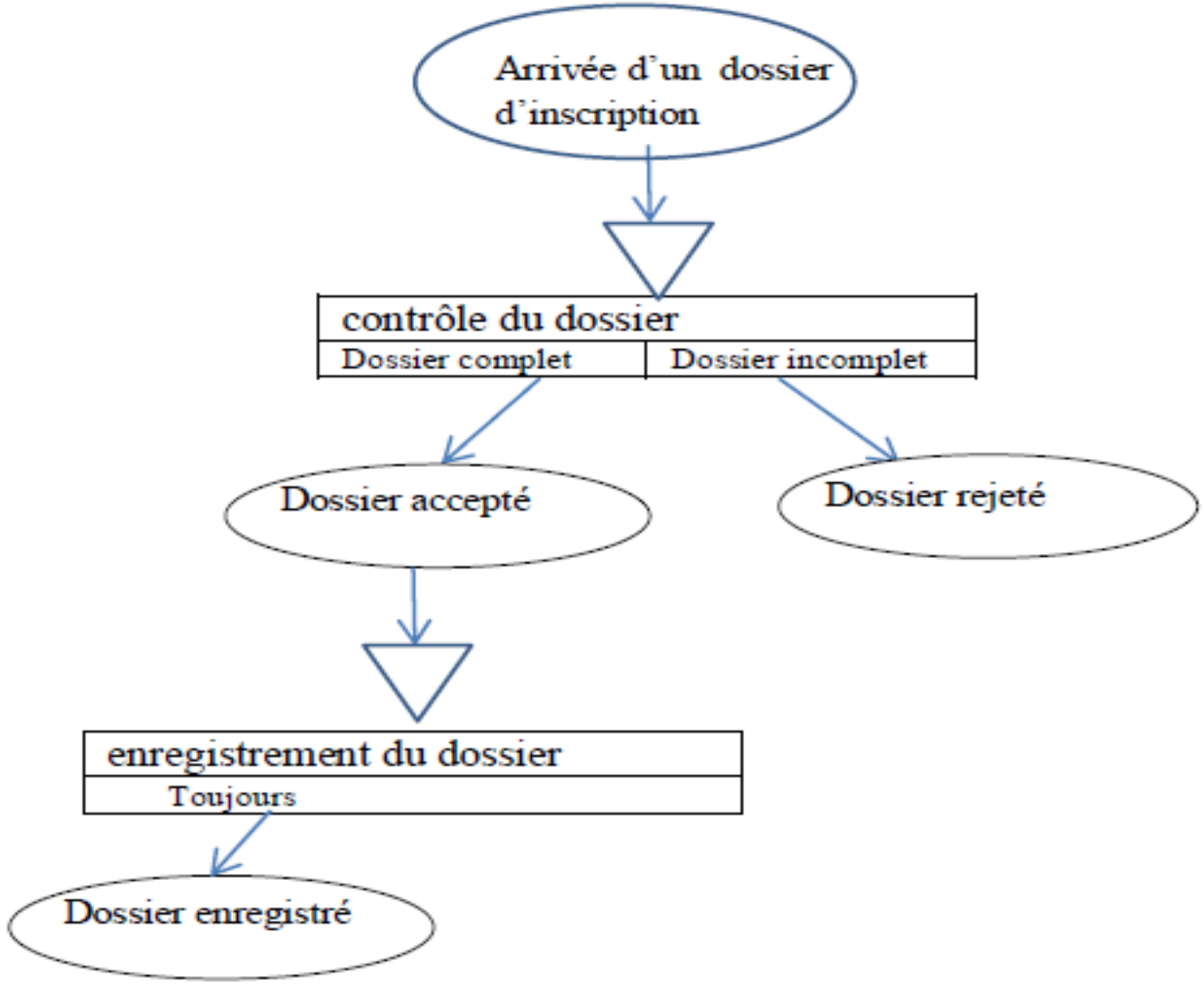
R2 : *Homogénéité des opérations* : à l'intérieur d'une opération il ne doit pas apparaitre des résultats pouvant
conditionner la suite du déroulement des opérations, si tel est le cas , il faut découper l'opération en question.

Exemple :



Cette opération est non homogène car l'action 2 :enregistrement du dossier ne peut se faire que si le dossier est complet donc il faut découper cette opération en 2 operations :

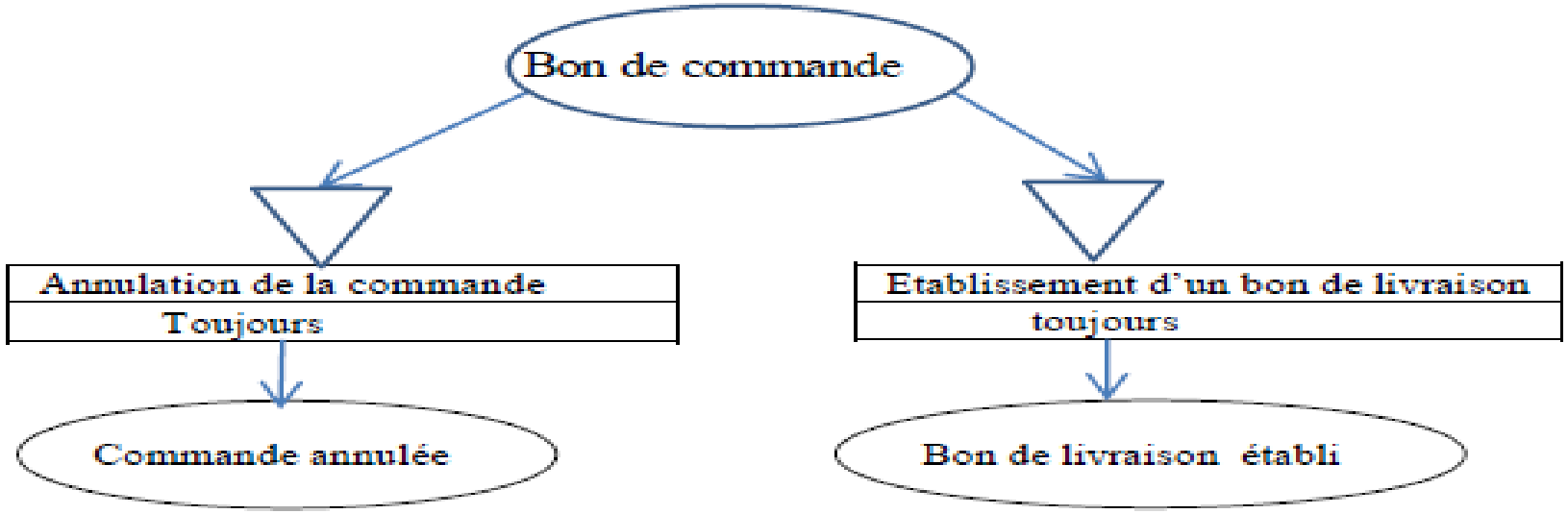
Le MCT devient :


Arrivée d'un dossier d'inscription
contrôle du dossier
Dossier complet
Dossier incomplet
Dossier accepté
Dossier rejeté
enregistrement du dossier
Toujours
Dossier enregistré

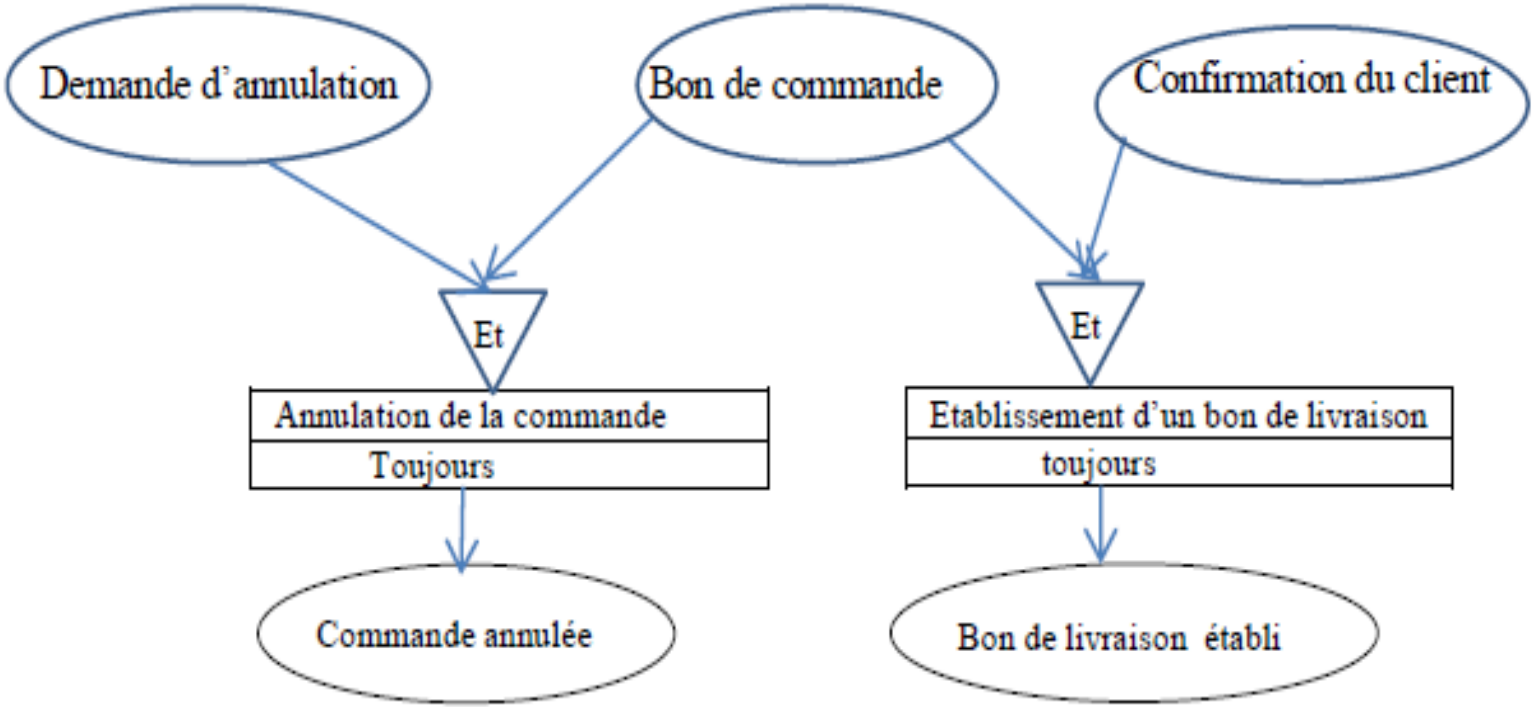
R3 : ***La consommation d'un évènement*** : un même evenementt ne peut pas être le déclencheur unique de deux opérations différentes si tel est le cas ,deux erreurs sont possibles :

- Il peut manquer des évènements dans l'une ou l'autre des opérations.
- Les deux opérations ne font qu'une seule opération.

Exemple : soit le MCT :

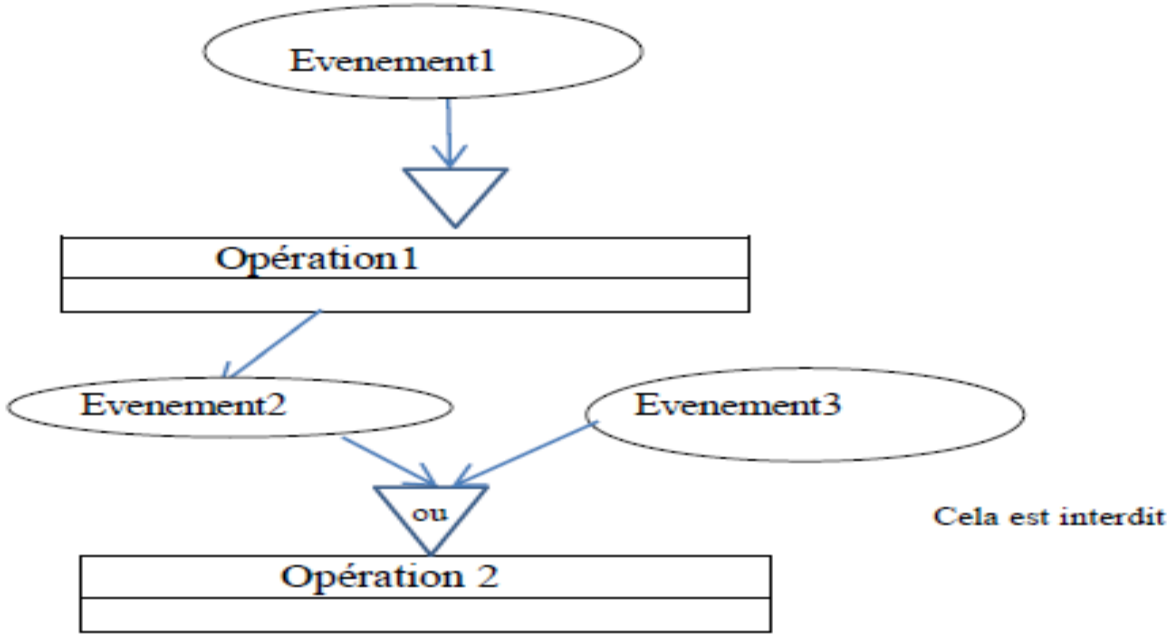


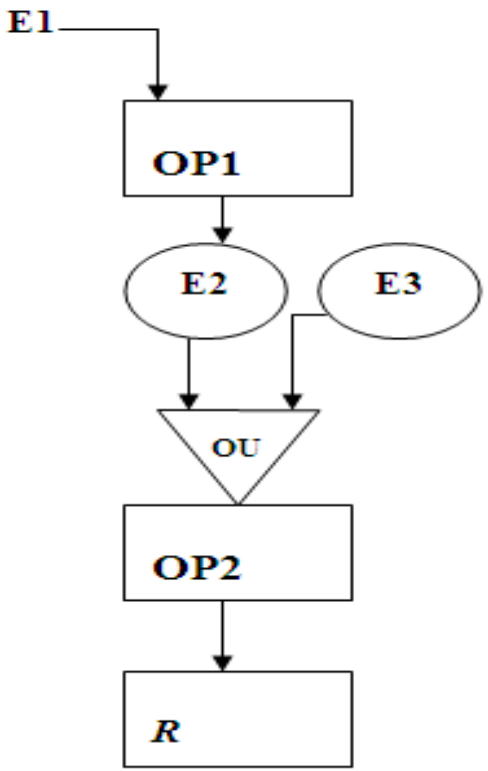
Dans cet exemple, l'évènement bon de commande ne peut pas être le déclencheur unique des deux opérations : annulation de la commande et établissement d'un bon de livraison , il faut ajouter les évènements manquants :

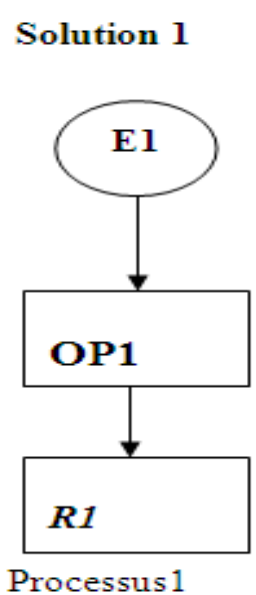


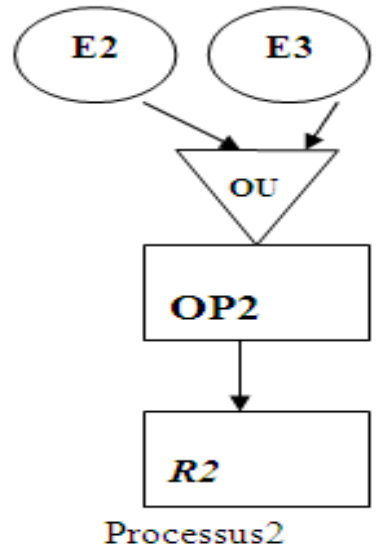
R4 : *La non-redondance des opérations* : une opération ne doit pas figurer plus d'une fois dans un processus du MCT. si tel est le cas il faut regrouper les opérations similaires dans une même opération et synchroniser leurs évènements déclencheurs

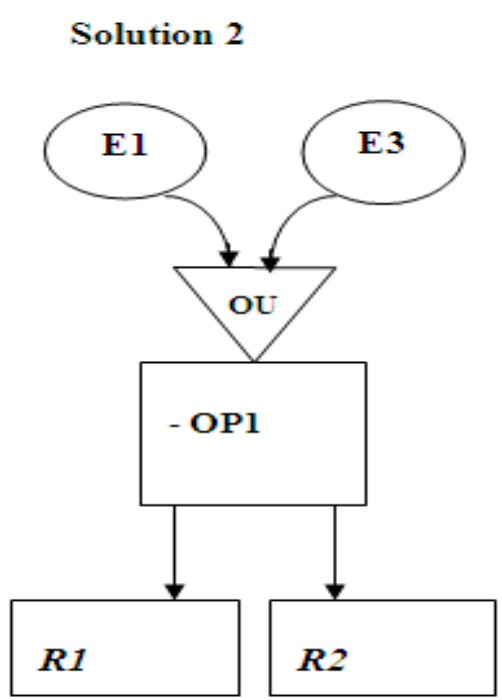
R5 :*La synchronisation 'ou' entre un évènement interne et un évènement externe* ne doit pas figurer dans un MCT




E1
OP1
E2
E3
OU
OP2
R


Solution 1
E1
OP1
R1
Processus1


E2
E3
OU
OP2
R2
Processus2


Solution 2
E1
E3
OU
- OP1
R1
R2

VI. Le Modèle Logique des de Données (MLD)

1. Introduction

La description conceptuelle a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser. Mais cette représentation ne peut pas être directement manipulée et acceptée par un système informatique, car le MCD est une représentation des données dans un formalisme compris par les concepteurs et pas par la machine. Il est donc nécessaire de passer du niveau conceptuel à un niveau plus proche des capacités des systèmes informatiques. Ce niveau, appelé niveau logique, consiste à choisir l'un des trois modèles suivants :

- modèle hiérarchique
- modèle réseau,

ou modèle relationnel

Dans ce cours l'accent sera mis sur le **modèle relationnel.**

2. Les concepts relationnels

- ***Domaines*** **:** ensemble de valeurs. Chaque domaine est caractérisé par un nom.
- Exemple :

Le domaine « nom » : ensemble de mots (suite de lettres)
Le domaine « sexe » : {masculin, féminin}.

Attributs **:** sous-ensemble d'un domaine caractérisé par un nom. Il correspond à la donnée élémentaire
Exemple : l'attribut « nom »

- ***Relations*** **:** sous-ensemble d'un produit cartésien de domaines, muni d'un nom.

Exemple : la relation « ETUDIANT » est un sous-ensemble de :
« nom » x « prénom » x « no-carte » x « sexe ».

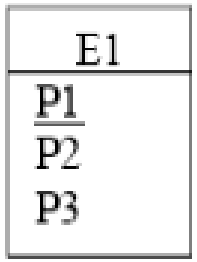
Etudiant (nom, prenom, nocarte, sexe)

Remarque : l'association du formalisme entité-association est différente de la relation du modèle relationnel.

Clé de la relation : ensemble minimum d'attributs dont la connaissance des valeurs identifie un tuple de façon unique.

3. Règles de passage du MCD au MLD

- **Propriété :** une propriété du modèle entité-association devient un attribut du modèle relationnel.
- **Entité** : devient une relation du modèle relationnel. L'identifiant de l'individu devient la clé primaire de la relation correspondante.



Entité

E1(P1,P2,P3, …)

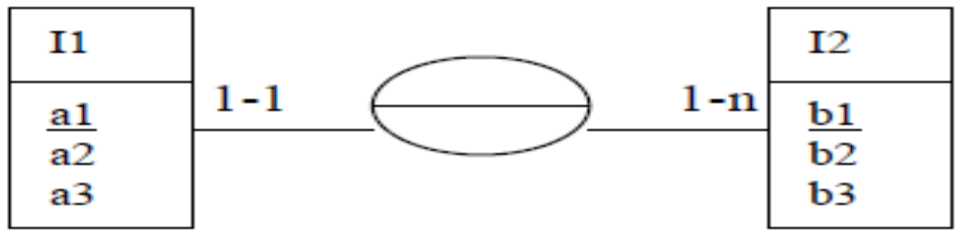
Relation

- **Association**

Sans propriétés propres

Si elle est binaire fonctionnelle ===> la relation-type disparaît

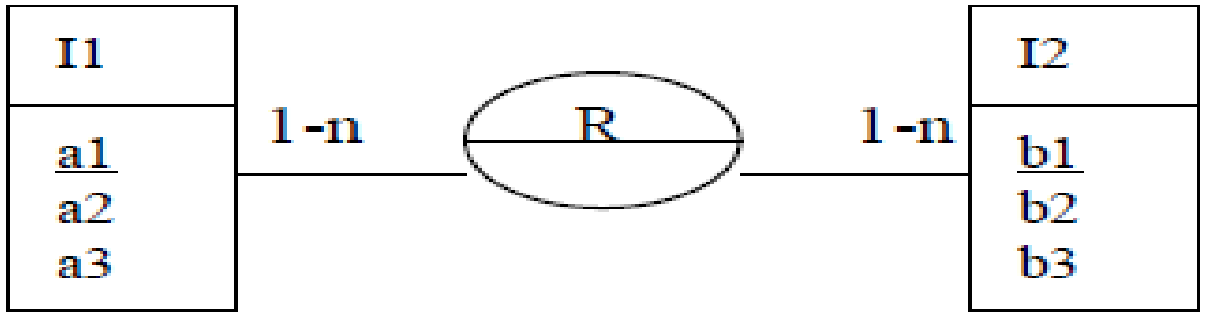
Exemple :



on obtient le modèle relationnel suivant : I1(a1, a2, a3, b1) ; I2(b1, b2, b3)

- Si elle est binaire non-fonctionnelle ===> la relation-type du MI devient une relation relationnelle

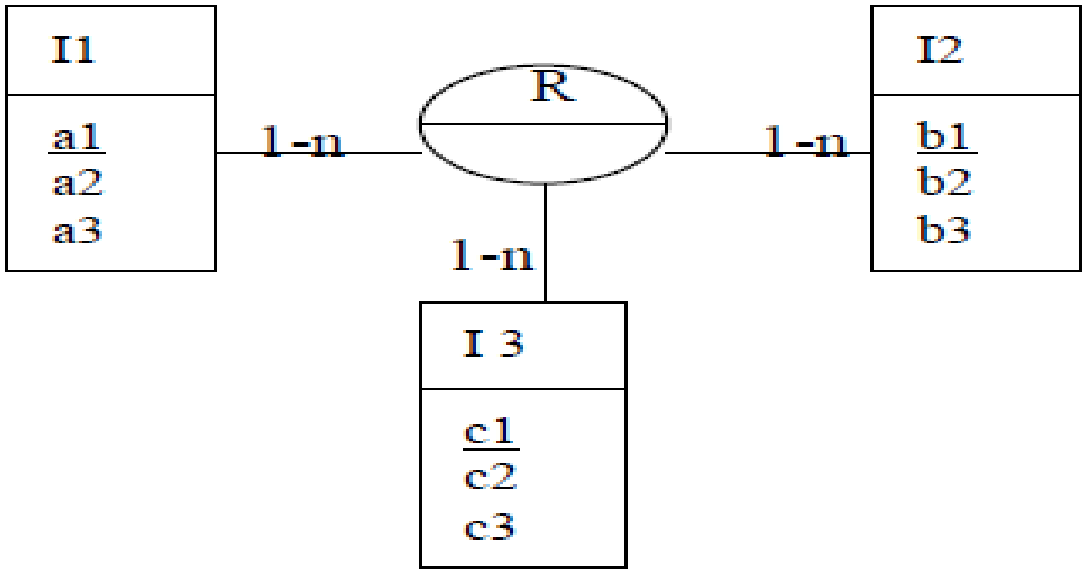
Exemple :



On obtient le modèle suivant relationnel suivant :
I1(a1, a2, a3) ; I2(b1, b2, b3) ; R(a1, b1)

- Si elle est n-aire ===> elle devient une relation relationnelle

Exemple :

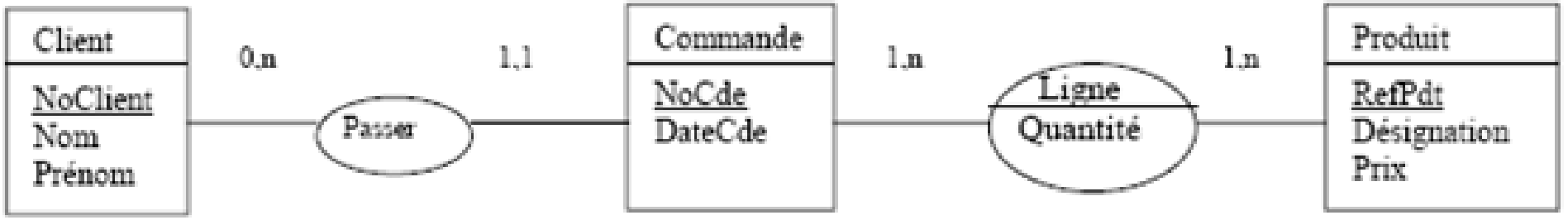


On obtient le modèle relationnel suivant :

I1(a1, a2, a3) ; I2(b1, b2, b3) ; I3(c1, c2, c3) ; R(a1, b1,c1)

Avec propriétés propres : l'association est équivalente à une relation relationnelle, les propriétés propres deviennent des attributs de la relation.

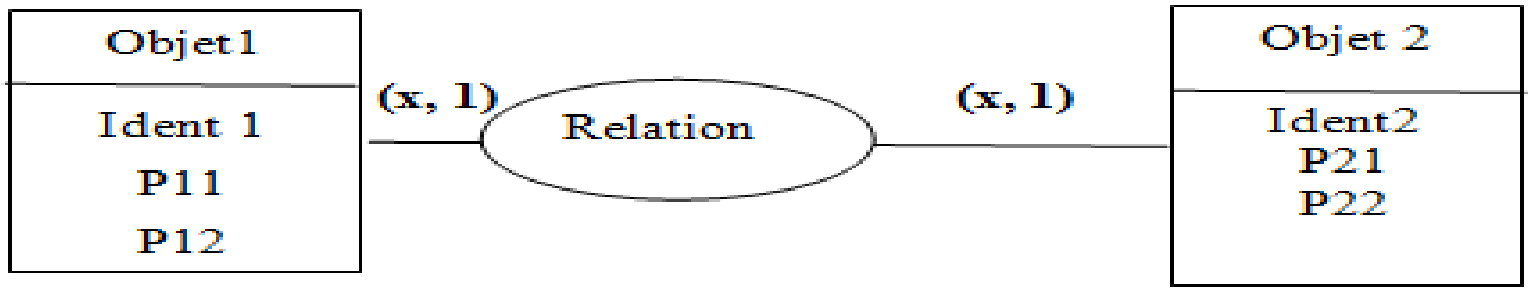
Exemple



Modèle logique de données MLD relationnel ´équivalent est le suivant :

Client(NoClient, Nom, Prénom)
Commande (NoCde, DateCde, NoClient#)
Produit(RefPdt, désignation, Prix)
Ligne(NoCde#, RefPdt#, Quantité)

Cas particuliers : Cas de cardinalité (x,1),(x,1) : x peut prendre 0,1 indifféremment.

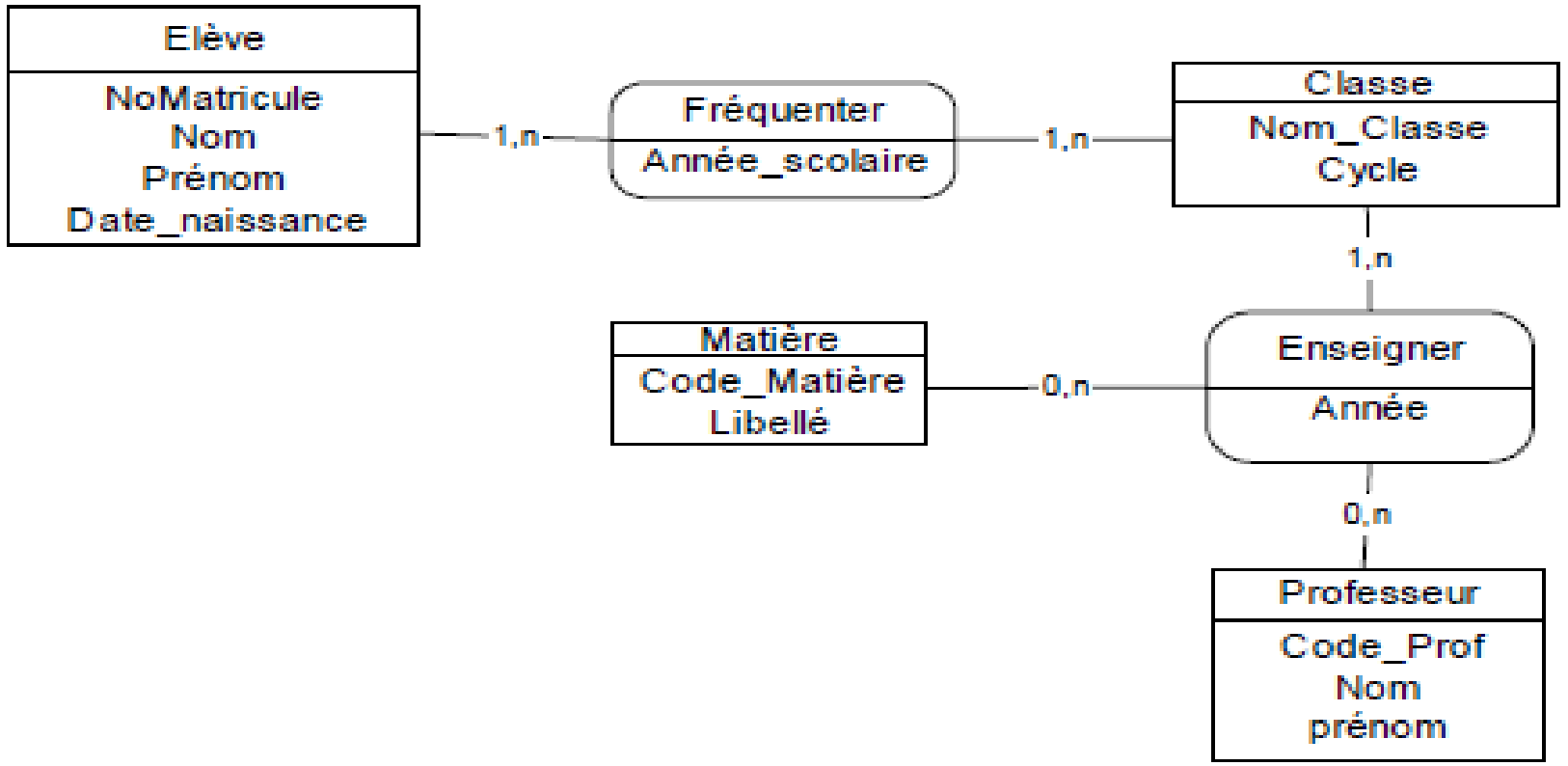


Plusieurs manières de transformations existent nous retenons la suivantes :
Dans le MLD relationnel l'identifiant ident1 devient clé dans la table associée `a la relation (table) associée `a l'objet 2. Il sera de même pour l'identifiant ident2 qui sera clé ´étrangère dans la relation (table) associée `a l'objet1 donc le modèle logique est le suivant:

Objet1 (ident1,p11,p12,ident2*)
Objet2 (ident2,p21,p22,ident1*)

Exercice1 (Gestion d'école)

Transformez le MCD suivant, qui représente «la gestion d'une école », en un MLD en respectant toutes les règles du passage MCD à MLD.



Exercice 2

Donner le MCD correspondant au MLDR suivant, préciser les cardinalités et les identifiants des entités :

CANDIDAT (n°candidat, nom candidat, prénom candidat, date-naissance)

EPREUVE (n°épreuve,libellé-épreuve,date rédaction,date épreuve,coefficient,Code examen#)

EXAMEN (Code examen, libellé-examen)

ENSEIGNANT (n°enseignant, nom-enseignant, prénom enseignant)

PASSER (n°candidat#, n°épreuve#, note)

REDIGER (n°enseignant#, n°épreuve#)

INSCRIRE (Code examen#, n°candidat#, appréciation)