

Matière : Systèmes d'Information

Chapitre 5.2 : MERISE Modèle conceptuel des données (MCD) (SUITE)

5.1 Introduction

MERISE (Méthode d'Etude et de Réalisation Informatique de Systèmes d'Entreprise), prévoit une **conception par niveaux**, et définit pour cela 3 niveaux essentiels :

- 1. Le niveau conceptuel**, qui se base directement sur l'analyse, décrit l'ensemble des données et des traitements du système d'information, sans tenir compte de l'implémentation informatique de ces données ni des détails organisationnels et techniques des traitements (quoi?). Ce niveau se traduit par deux modèles conceptuels que nous appelons : **Modèle conceptuel des données (MCD)** et **Modèle conceptuel des traitements (MCT)**.
- 2. Le niveau logique**, qui se base sur le modèle conceptuel, prend en considération la technique d'organisation de données et les détails organisationnels des traitements (qui, où et quand?). Ce niveau est représenté par les : **Modèle logique des données (MLD)** et **Modèle organisationnel des traitements (MOT)**.
- 3. Le niveau physique**, qui se base sur les modèles du niveau précédent, contient finalement tous les détails d'implémentation du système de données et des traitements (comment?). Ce niveau est représenté par les : **Modèle physique des données (MPD)** et **Modèle opérationnel des traitements (MOpT)**.

La modélisation des données de SI consiste à *décrire* et à *représenter* l'ensemble des données manipulées par celui-ci grâce à un formalisme de représentation ainsi que des *liens existants* entre ces données. Pour représenter fidèlement les données du SI nous avons besoins de connaître :

- Ses besoins en information
- Les règles qui régissent son fonctionnement,
- Les données manipulées par le SI (dictionnaire de données).

5.2 Le Dictionnaire de donnée

Le dictionnaire de donnée est un outil nécessaire pour la construction de MCD.

5.2.1 Construction du dictionnaire de donnée

Prenant l'exemple du SI lié au *service de vente*. Nous allons extraire toutes les données contenues dans les deux (2) documents commande et facture.

Le dictionnaire de donnée est un tableau qui regroupe toutes les données du SI, pour chaque donnée il faut préciser : *Code de la donnée, désignation, type, taille (longueur), une observation* si cela est nécessaire.

Commande
 N° commande :
 Date commande :
 N° client :
 Nom client :
 Adresse client :

Ref	Désignation	PU	Quantité

Facture
 N° facture :
 Code commande :
 Date facture :
 Matricule client :

Ref	Désignation	PU	Quantité	Montant

Montant total :

Le dictionnaire de donnée correspondant est le tableau suivant :

Code de la donnée	Désignation	Type	Taille	Observation
Num-C	Numéro de la commande	N	4	
Date-C	Date de la commande	Date	10	JJ/MM/AAAA
Num-CL	Numéro client	N	4	
Nom-CL	Nom client	A	15	
Adr-CL	Adresse client	A N(c)	40	
Ref	Référence produit	C	10	
Désignation	Désignation produit	C	20	
PU	Prix unitaire produit	N	8	
Quantité	Quantité produit	C	6	
Num-F	Numéro facture	N	4	
Date-F	Date facture	D	10	
Code-C	Code commande	N	4	
Mat-C	Matricule client	N	4	
Ref	Référence produit	C	10	
Desgn	Désignation produit	C	20	
PU-P	Prix unitaire produit	C	6	
Quantité	Quantité produit	N		
Montant	Montant produit	N		
Mont-T	Montant total	N		

Certain données doivent être *éliminées* de ce dictionnaire.

- **Synonymes** : des propriétés qui désigne la même donnée] : Les donnée **Num-C** et **Code-C** désigne la même donnée : "Numéro de la commande"

Num-CL et **Mat-CL** désigne la même donnée : "Numéro client" Donc Num-CL et Mat-CL sont des synonymes.

Il faut *éliminer tous les synonymes* et ne garder qu'une seule donnée, dans ce cas en garde Num-C et Mat-CL.

- **Polysémie** : La donnée qte désigne une quantité mais il y a deux quantités différentes :

Quantité-C et Quantité-F. Qte est une donnée qui a deux sens donc il est polysémie, il faut le supprimé et crée deux données Qte-fac, Qte-com.

- **Valeur calculé** : La donnée montant-P = PU * Qte-fact

La donnée Mont-Total = \sum Mont-P = PU _ Qte-fact. Les données calculées doivent être supprimé du dictionnaire de données.

- **Les données concaténées** : La donnée Adr-CL est une donnée (composée) qu'il faut décomposer comme suit : Ville-CL et Rue-CL.

Le dictionnaire de données **épuré** est comme suit :

Code de la donnée	Désignation	Type	Taille	Observation
Num-C				
Date-C				
Nom-CL				
Rue-CL				
Ville-CI				
Réf-Prod				
Des-Prod				
PU-p				
Num-F				
Date-F				
Num-C				
Qte-C				
Qte-f				

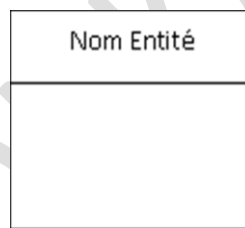
5.3 Le modèle Entité/Association

Le modèle E/A a été élaboré par CHEN en 1976 pour la modélisation des données et des liens existants entre elles avec des concepts simples et efficaces, c'est une représentation naturelle du monde réel du SI à étudier.

5.3.1 Concepts du modèle

5.3.1.1 Entité :

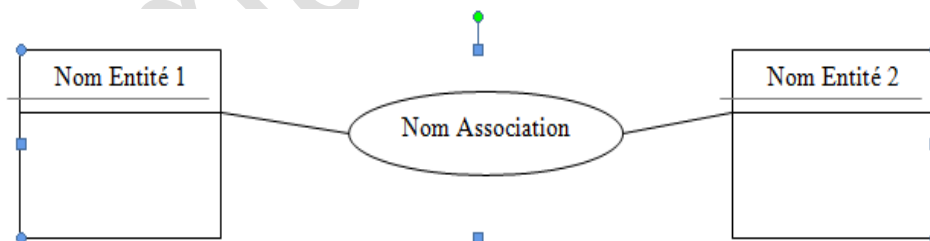
C'est une représentation dans un système d'information d'un objet matériel ou immatériel pourvu d'une existence propre. L'entité est schématisée comme suit :



5.3.1.2 Association :

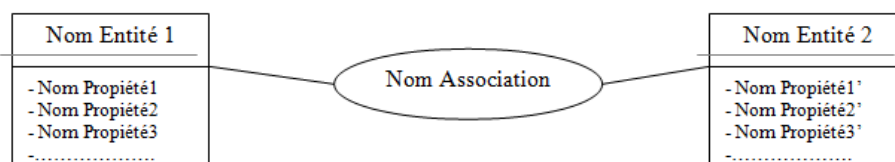
C'est une relation qui représente un lien entre les entités, elle est dépourvue de l'existence propre, son existence est liée à l'existence des entités qu'elle met en relation.

Exemple : Soit, la règle de gestion suivante : Un étudiant appartient à une section, on déduit l'association suivante : appartenir entre les entités Etudiant et section.



5.3.1.3 Propriété (attribut) :

C'est une donnée élémentaire qui caractérise une entité ou une association.

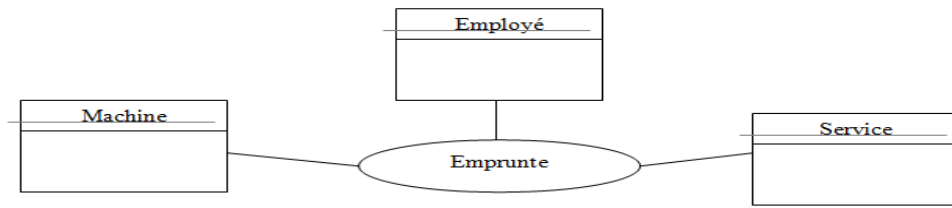


5.3.1.4 Dimension d'une Association :

La dimension d'une association désigne le nombre d'entité qui participe à cette association.

Exemples :

1. Soit la relation emprunter qui met en interaction 3 entités. Elle est dite ternaire.



Remarque :

- Une association qui lie une entité à elle-même est dite unaire (réflexive)
- Une association qui lie deux entités est dite binaire
- Une association qui lie trois entités est dite ternaire.

5.3.1.5 les cardinalités :

La cardinalité entre l'entité X par rapport à une association avec l'entité Y, exprime le nombre d'occurrence de Y, que l'on peut associer à une occurrence de l'entité X.

La cardinalité est exprimée par un couple (X,Y) tel que :

X est le nombre minimum d'occurrence de Y que l'on peut associer à une occurrence de X.

Y est le nombre maximum d'occurrence de Y que l'on peut associer à X.

Les cas possibles des cardinalité sont : (0,1), (0,n), (1,1), (1,n).

5.3.1.6 Occurrence d'une entité

: On considère les deux étudiants suivant :

Numéro : 001 Nom : <u>Aloui</u> Prénom : <u>Ali</u> Date de N : 09/12/80	Numéro : 002 Nom : <u>Mokhtar</u> Prénom : <u>Salaha</u> Date de N : 05/02/81
---	--

Ces deux étudiants sont deux individus particuliers de l'ensemble des étudiants, ils appartiennent à l'entité Etudiant donc ils sont appelés occurrence de l'entité étudiant.

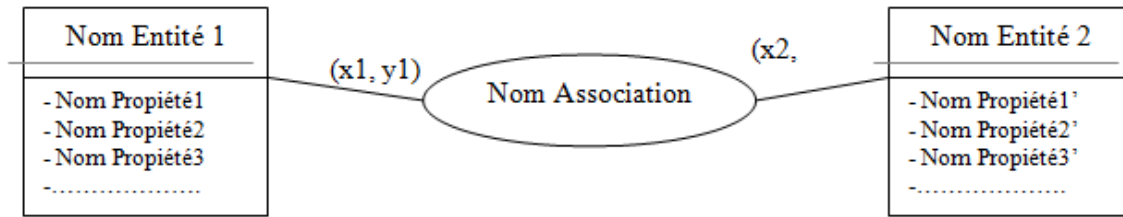
Par définition une occurrence de l'entité est un élément individualisé appartenant à cette entité, elle s'obtient par l'attribution de valeurs aux différentes propriétés qui caractérise cette entité.

5.3.1.7 Occurrence d'une association

: On considère la relation " enseigner " entre enseignant et module, l'association qui lie une occurrence de l'entité enseignant à une occurrence de l'entité Module est appelée : Occurrence de l'association Enseigner.

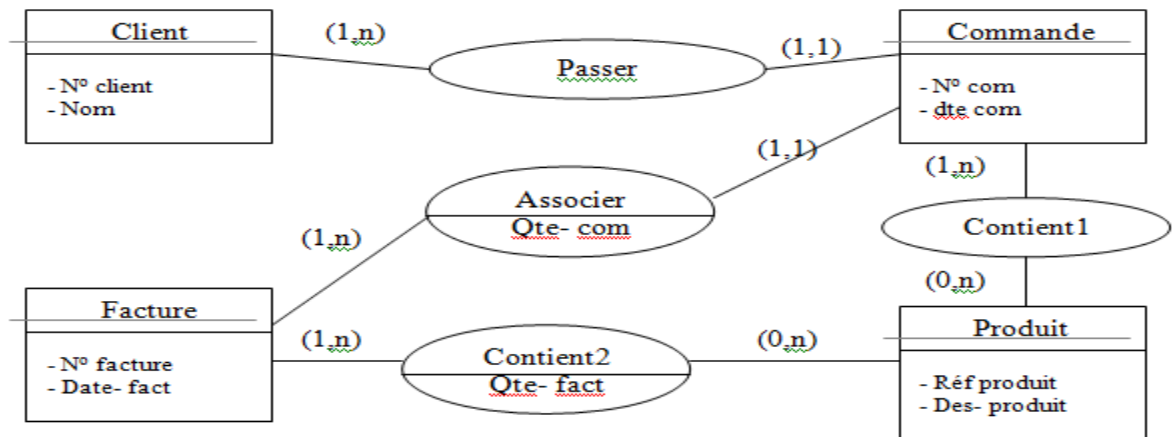
Par définition, une occurrence d'une association est une association individuelle entre une et une seule occurrence de chaque entité participant à l'association.

5.3.1.8 Représentation graphique



Exemple : Etablir le MCD correspondant aux règles de gestion suivantes :

- Un client peut passer une ou plusieurs commandes
- Une commande est passée par un et un seul client
- Une commande contient un ou plusieurs produits
- Un produit peut figurer sur plusieurs commandes, comme il peut ne pas être commandé.
 - A chaque commande est associée une et une seule facture.



Remarque :

- Il est interdit, de donner le même nom à deux entités ou (à deux associations différentes),
- Il est interdit, de donner le même nom à une propriété qui intervient plusieurs fois dans une même entité (ou une association) ou dans des entités (ou association) différentes.

5.3.1.9 Identifiant d'une entité

: L'identifiant d'une entité est une propriété particulière de l'entité, permettant d'identifier chaque occurrence de cette entité de manière unique.

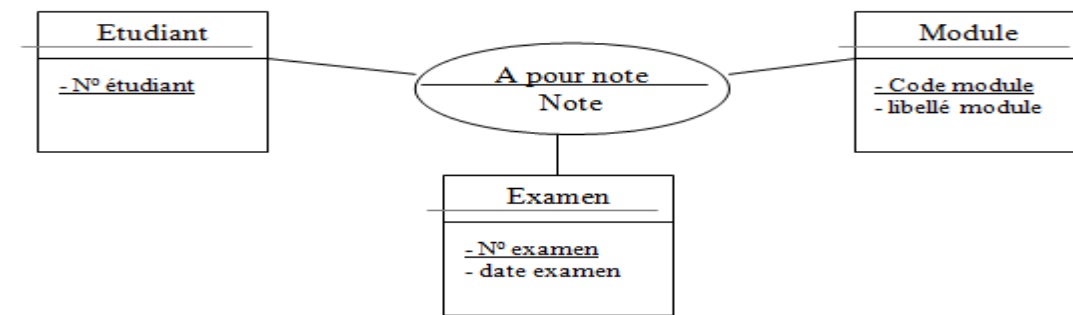
Remarque :

- L'identifiant peut être formé de plusieurs propriétés.
- Si l'entité possède plusieurs identifiants, il faudra en retenir celui qui répond le plus aux besoins de gestion et du domaine étudié.

5.3.1.10 Identifiant d'une association :

L'identifiant d'une association est obtenue par la concaténation des identifiants des entités qui participent à cette association. Cet identifiant est appelé identifiant implicite de l'association et il n'est pas représenté graphiquement.

Exemple : L'identifiant implicite de l'association " **a pour note** " est :



No étudiant + code module + no examen.

En plus de l'identifiant implicite, une association peut avoir son propre identifiant qui lui sera représenté graphiquement.

5.4 Règles à respecter lors de l'élaboration du modèle E/A :

Règle1 : (règle d'identification) Il doit exister un identifiant pour chaque entité.

Règle2 : Les propriétés des entités doivent être élémentaires (type non-décomposé)

Règle3 : (règles de vérification des entités) Soit l'entité (assuré)

Si un assuré peut avoir plusieurs véhicules, alors La propriété, no, type, marque de véhicule ne peuvent Appartenir à l'entité assurée, donc on doit faire appel à une autre modélisation.

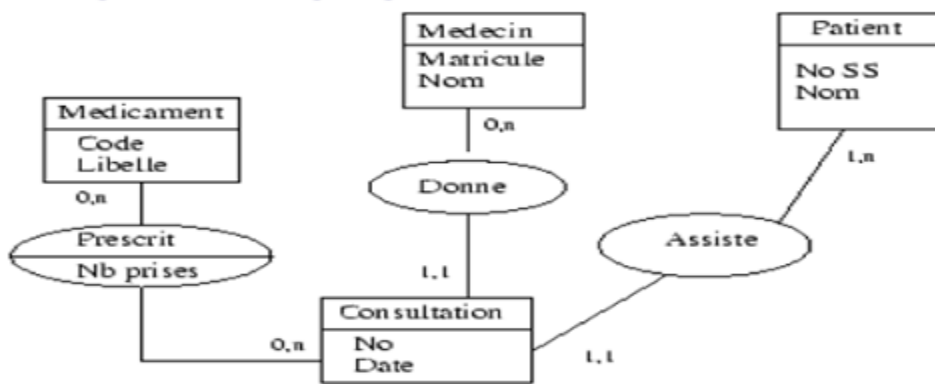
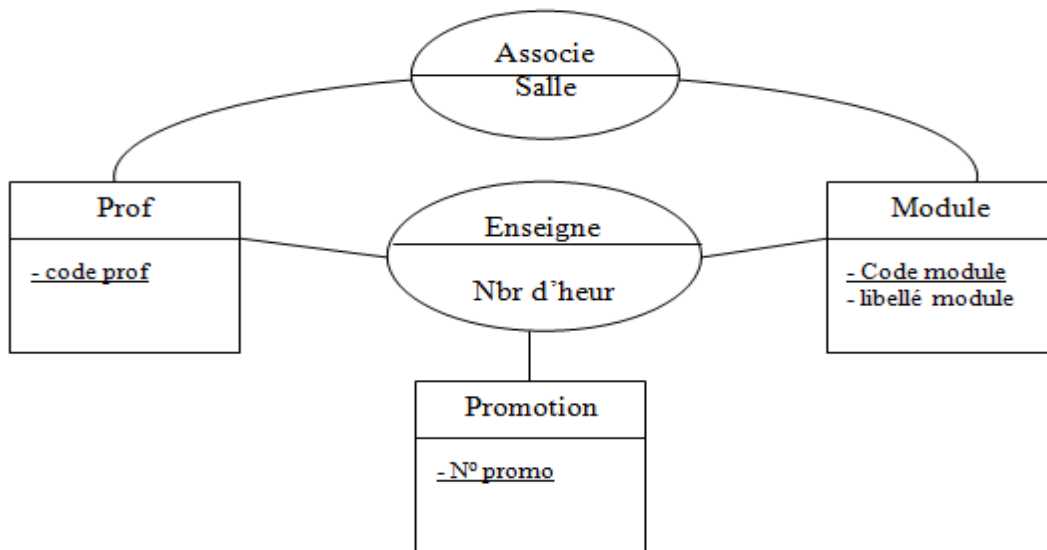
Règle 4 : Il faut éviter des propriétés vides pour certaine occurrence de l'entité.

Règle 5 : (règle de normalisation d'une association) A une occurrence d'une association, il ne doit pas y avoir qu'une seule valeur pour chacune des propriétés rattachées à cette association. En plus, il ne doit y avoir qu'une et une seule occurrence de chaque entité participant à cette relation.

Règle 6 : Toute propriété portée par l'entité, doit dépendre de l'identifiant de cette entité et de tout l'identifiant, pas d'une partie seulement.

Règle 7 : Dépendance pleine des propriétés des associations de toutes les entités concernées. Si l'on suppose que l'enseignant, enseigne un module dans la même salle.

Donc code prof + code module ! Salle



5.5 Normalisation du MCD :

Normaliser un MCD revient à appliquer un ensemble de règles sur ses objets et ses relations.

a. Normalisation des objets : Se fait suivant 3 étapes

1. Normalisation en première forme normale (1ere FN).
2. Normalisation en deuxième forme normale (2eme FN).
3. Normalisation en troisième forme normale (3eme FN).

a.1 La première forme normale (1ere FN) : un objet est dit en 1ere FN s'il vérifie les 2 règles suivantes :

- **Règle1 :** (règle d'identification) Il doit exister un identifiant pour chaque entité. (étudiant sans mat)
- **Règle2 :** Les propriétés des entités doivent être élémentaires (type non-décomposé). (adresse=rue, ville, code_postal)

a.2 La deuxième forme normale (2eme FN) : un objet est dit en 2eme FN, s'il est en 1ere FN et en plus, vérifie la règle suivante :

- **Règle 6 :** Toute propriété portée par l'entité, doit *dépendre de l'identifiant* de cette entité et de *tout l'identifiant*, pas d'une partie seulement. (l'identifiant de étudiant : mat, date_inscription)

a.3 La troisième forme normale (3eme FN) : un objet est dit en 3eme FN, s'il est en 2eme FN et en plus, vérifie la règle suivante :

- **Règle 8 :** Dans l'objet toute propriété doit dépendre de l'identifiant par une dépendance fonctionnelle élémentaire et directe. Exp : produit (code produit, nom_produit, code_categorie, nom_categorie) on a $\text{code_produit} \rightarrow \text{nom_categorie}$ n'est pas direct car $\text{nom_categorie} \rightarrow \text{nom_categorie}$;

la solution : produit (code produit, nom_produit, code_categorie) et categorie(code_categorie, nom_categorie)

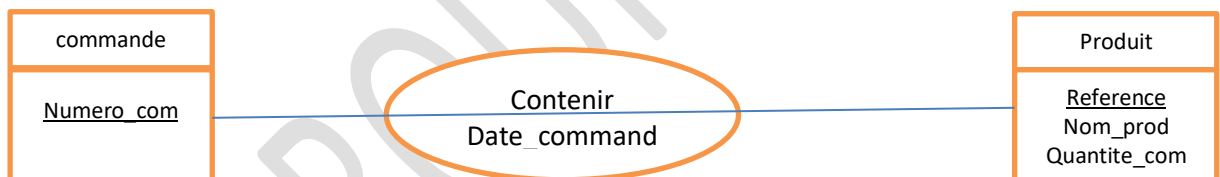
etudiant (matricule, nom, age, club, salle) ; $\text{matricule} \rightarrow \text{club}$ mais $\text{matricule} \rightarrow \text{salle}$ (n'est pas correcte) et $\text{club} \rightarrow \text{salle} \Rightarrow \text{salle}$ ne dépend pas de l'identifiant n'est pas 3emeFN

\Rightarrow solution : etudiant (matricule, nom, age, club) et activite(club, salle)

b. Normalisation des relations : la règle suivante doit être appliquée :

- **Règle 7 :** Dépendance pleine des propriétés des associations de toutes les entités concernées (toute propriété de la relation (association) doit dépendre pleinement de l'ensemble des identifiants des objets qui participent à la relation, mais pas d'aucun sous-ensemble de cet ensemble).

Exp :



La relation contenir n'est pas normalisée car la date de commande dépend du numéro_com et référence du produit d'où : $\text{numero_com, reference} \rightarrow \text{Date_command}$. (n'est pas élémentaire)
C'est faux car la date_command dépend uniquement de la commande d'où ; $\text{numero_com} \rightarrow \text{Date_command}$.

5.6 Vérification du MCD :

a. Vérification des objets : La vérification des objets revient à appliquer les règles suivantes :

- **Règle9 :** pour chaque occurrence d'un objet, chaque propriété ne peut prendre qu'une et une seule valeur (on ne peut avoir de valeurs répétitives pour une même propriété).
- **Règle10 :** toutes les propriétés d'un objet doivent être significatives (il ne doit pas y avoir d'occurrence d'objet ne possédant pas de valeur pour une propriété donnée).

b. Vérification des relations :

- **Règle 11 :** 2 occurrences d'un objet ne peuvent participer à une même occurrence de relation.
- **Règle 12 :** pour une occurrence de la relation, il ne doit pas y avoir de participation optionnelle de l'un de ses objets.

5.7 Contraintes d'intégrités et dépendances fonctionnelles :

Un autre outil qui aide à la construction du MCD, c'est le graphe des dépendances fonctionnelles, appelé aussi structure d'accès théorique (SAT). Mais avant cela, nous devons tout d'abord expliquer certains concepts qui s'y rapportent qui sont : *règle de gestion, contrainte d'intégrité fonctionnelle, dépendance fonctionnelle*.

- a. **Les règles de gestion** : expriment les règles auxquelles obéit le système à modéliser donc déduire des relations entre objets et déterminer les cardinalités du MCD.

Exemple : une commande contient au moins un produit.

Un étudiant appartient à une et une seule spécialité.

- b. **Les contraintes d'intégrité fonctionnelles** : Une contrainte d'intégrité est une règle qui définit la cohérence d'une donnée ou d'un ensemble de données.

Exemple : sur les propriétés : un nom est de type alphabétique, la note de l'étudiant entre 0 et 20, la date d'inscription d'un étudiant doit être > date de naissance.

- c. **Les dépendances fonctionnelles** : une dépendance est un lien déterminant entre éléments(on dit que 2 propriétés **A** et **B** sont reliées par une dépendance fonctionnelle, si la connaissance de la valeur de **A** déterminé une et une seule valeur de **B**) et on note $A \rightarrow B$. **A** peut-être une ou *plusieurs propriétés*.

Les dépendances fonctionnelles doivent être **élémentaires** ($A \rightarrow B$ élémentaire si aucune partie de **A** ne détermine **B**) et directes ($A \rightarrow B$ est directe s'il n'existe pas une propriété **C** telle que $A \rightarrow C$ et $C \rightarrow B$).

Exemple : mat-etud \rightarrow nom-et, prenom-etud.

Reference_produit,num_commaBnde \rightarrow quatite_command.

Mat \rightarrow spécialité et spécialité \rightarrow faculté et mat \rightarrow faculté . \Rightarrow mat \rightarrow faculté sera éliminé.