

- **A-Introduction à Merise**
 1. Le système d'information de gestion
 2. La modélisation

- **B-Introduction au Modèle conceptuel des données**
 1. Recueil des données
 2. Présentation des concepts et du formalisme du MCD
 3. Règles de base pour la construction d'un MCD
 4. Introduction aux cardinalités

- **C-Les cardinalités et les différents types d'associations**
 1. Les cardinalités
 2. Les différents types d'associations
 3. Représentation des cardinalités
 4. Association ternaire
 5. Associations réflexives

- **D-Passage MCD \Leftrightarrow modèle relationnel**
 1. Rappel sur le modèle relationnel
 2. Passage du MCD au MR

- **E-Normalisation du modèle relationnel**
 1. Dépendance fonctionnelle
 2. Intérêt de la normalisation
 3. Les 3 formes normales

- **F-Les diagrammes de flux**
 1. Vocabulaire associé aux modèles de flux
 2. Modèle de contexte
 3. Modèle de flux conceptuel ou diagramme de flux de données

- **G-Les Modèles de traitements (MCT et MCTA)**
 1. MCT
 2. Intégration des données dans le MCT (à MCTA)

SANGARE SOULEYMANE

00225-08281648

sangsoulinter@yahoo.fr

sangsoulci@hotmail.com

A-Introduction à Merise

I-Système d'information de gestion

Sommaire

- [1 Qu'est ce qu'un système ?](#)
- [2 Qu'est ce qu'un système d'information de gestion](#)
- [3 Les fonctions du système d'information](#)

Le champ d'action d'un informaticien de gestion s'appelle système d'information de gestion (SIG)

1-Qu'est ce qu'un système ?

Un système est un tout constitué d'éléments unis par des relations, leurs propriétés et les valeurs que peuvent prendre ces dernières, ainsi que son activité et l'organisation qui en découle.

L'entreprise peut être vue comme un système, composé d'éléments tels que des « employés », des « services », des « produits », etc. Les propriétés décrivant ces éléments peuvent être « le matricule de l'employé », son « nom », la « référence » du produit, sa « désignation », etc. Entre ces éléments, on trouve des relations, telles la relation « est rattachée » entre un employé et son service, la relation « est stockée » entre un produit et son dépôt de stockage... Les propriétés de ces relations seront du type « date d'entrée dans le service », « quantité stockée » ...

2-Qu'est ce qu'un système d'information de gestion

Le système d'information d'une entreprise est l'ensemble des informations qui y circulent ainsi que l'ensemble des moyens mis en oeuvre pour les gérer. L'objectif d'un système d'information est de restituer l'information à la personne concernée, sous la forme appropriée et en temps opportun pour prendre une décision ou effectuer un travail.

Toutes les informations qui circulent dans l'organisation, quelle que soit leur forme, font partie du SI. Mais seules les informations formalisées nous intéressent, car elles sont les seules à être vraiment utilisables en informatique. L'ensemble de ces informations formalisées constituent ce qu'on appelle le système d'information automatisé.

Les systèmes d'information préexistent à l'informatique. Avant d'utiliser les ordinateurs, les informations concernant les clients, les commandes, les stocks, etc, étaient mémorisées sur papier, sous forme de fiches, formulaires, regroupés dans des dossiers. Et il existait des procédures manuelles pour traiter ces informations. Le système informatisé doit être au service du système d'information mis en place par les dirigeants de l'entreprise (et non l'inverse comme cela existe trop souvent).

3-Les fonctions du système d'information

Le système d'information de l'entreprise reçoit de son environnement des informations qu'il doit traiter. Ce sont par exemple des commandes de clients qui doivent être traitées jusqu'à leur aboutissement, ou l'arrivée des factures des fournisseurs. Le SI reçoit et traite aussi des informations internes à l'organisation, comme par exemple les documents comptables, ou les chiffres de production.

- On peut distinguer 4 fonctions principales du système d'information :

- 1- Recueillir l'information (saisie)
- 2- Mémoriser l'information (stockage dans des fichiers ou bases de données)
- 3- Exploiter l'information (traitement)
 - a. Consulter
 - b. Organiser
 - c. Mettre à jour
 - d. Produire de nouvelles informations par des calculs
- 4- Diffuser l'information (édition)

II-Merise:Modélisation

Sommaire

- [1 Qu'est qu'un modèle](#)
- [2 La méthode Merise](#)
 - [2.1 Les composantes de Merise](#)
 - [2.2 La démarche Merise](#)
 - [2.3 Les niveaux d'abstraction](#)
 - [2.4 La courbe du soleil](#)

Les méthodes d'étude des systèmes d'information ont pour objectif de décrire ces systèmes à l'aide de modèles, puis de réaliser les systèmes informatisés qui en découlent. MERISE est une de ces méthodes, la plus répandue en France actuellement.

1-Qu'est qu'un modèle

Un modèle peut être défini comme étant une image de la réalité. Toute réalité complexe a besoin d'être représentée pour être comprise et maîtrisée.

- Exemple :

Une carte routière représente, à l'aide de symboles, tout ou partie d'un territoire et son infrastructure routière. Le plan établi par un architecte permet à chaque corps de métier d'intervenir efficacement dans la construction d'un immeuble.

Ainsi, l'informaticien peut être vu comme l'architecte du système informatisé, qui en dresse le plan détaillé. Un modèle doit permettre de communiquer dans l'ambiguïté. Donc il faut utiliser un formalisme normalisé. Merise propose plusieurs modèles pour représenter le [SIG](#).

2-La méthode Merise

Merise est une méthode qui permet de construire un système d'information automatisé qui soit efficace, flexible et adapté à l'entreprise. C'est une méthode née en 1976 suite à un projet du Ministère de l'Industrie.

2.1-Les composantes de Merise

Comme toute méthode d'analyse et de conception des SI, [Merise](#) regroupe :

- Des modèles (concepts, règles de représentation)
- Un langage (vocabulaire, règles de syntaxe)
- Une démarche
- Des outils (des logiciels tels que AMC Designor ou Win Design)

Elle permet de représenter les composantes d'un [SIG](#)

- Les acteurs
- Les données
- Les traitements
- Les procédures
- Les postes de travail, etc.

Ceci afin de concevoir, réaliser et mettre en place une nouvelle informatisation.

2.2-La démarche Merise

1-Approche globale du SI	Schéma directeur
2- Etude des différentes solutions possible puis choix	Etude préalable
3- Complément des spécifications du domaine	Etude détaillée
4- Spécifications techniques complètes	Etude technique
5- Ecriture des programmes	Production
6- Tests, essais, formation utilisateur	Mise en oeuvre
7- Corrections et adaptations du logiciel	Maintenance
8- Contrôle de la conformité des résultats par rapport aux demandes	Contrôle de qualité, recette

2.3-Les niveaux d'abstraction

La nécessité d'aborder différents types de préoccupations (décrire l'activité, définir les règles de gestion, définir les informations, répartir les traitements entre l'homme et la machine, organiser physiquement les fichiers, choisir le matériel, répartir les responsabilités, etc.) a conduit à proposer différents niveaux d'abstraction.

Les voici :

NIVEAU	QUESTION	CONTENU	DOMAINE
Conceptuel	Que faire ?	- données manipulées - règles de gestion - enchaînement des traitements	Gestion
Organisationnel	Qui fait quoi ? Quand ? Où ?	- partage des tâches - mode de traitement - répartition géographique des traitements - organisation des données	Organisation
Physique	Comment ?	- programmes - logiciels - matériels	Technique

Merise (du moins dans sa première version) est une méthode qui préconise de séparer l'étude des données et les traitements.

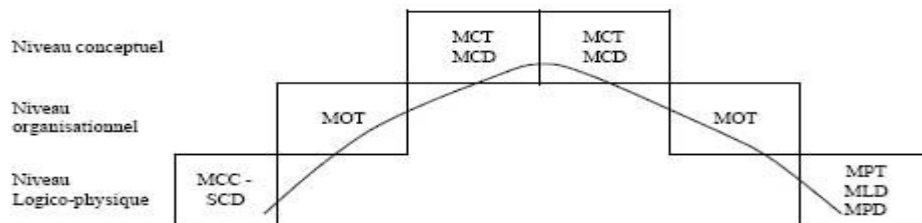
Chaque modèle de Merise concerne soit les données, soit les traitements, à un niveau d'abstraction donné.

Voici la présentation de ces différents modèles selon cette typologie :

NIVEAU	DONNEES	TRAITEMENT
Conceptuel	MCD Modèle Conceptuel des Données	MCT Modèle Conceptuel des Traitements
Organisationnel	MLD Modèle logique des données	MOT Modèle Organisationnel des Traitements
Physique	MPD Modèle Physique des Données	MOPT Modèle Opérationnel des Traitements

2.4-La courbe du soleil

Merise est utilisé dans le cas où l'on veut mettre en place ou modifier un système informatisé. Dans les deux cas, il convient d'analyser et de critiquer le système existant afin de créer un nouveau système adapté à l'organisation. Pour cela, la démarche consiste à suivre la « courbe du soleil ».



L'analyse du système existant conduit à construire :

- Au niveau logico-physique :
 - Le modèle conceptuel des communications ([MCC](#)) qui représente les échanges de flux d'informations entre les différents acteurs du SI et les acteurs extérieurs
 -
 - Le schéma de circulation des documents ([SCD](#)) qui représente les échanges d'informations entre les acteurs du SI ainsi que les tâches qui produisent les documents.
- Au niveau organisationnel :
 - Le modèle organisationnel des traitements qui permet de préciser par rapport à l'étape précédente si les tâches sont automatisées ou manuelles, les événements déclencheurs et les conditions d'émission des objets externes.
- Au niveau conceptuel
 - Le [MCD](#) (modèle conceptuel des données) et le [MCT](#) (modèle conceptuel des traitements)

Le passage de la modélisation du système existant à la modélisation du système futur implique une critique du système existant, une réflexion sur les évolutions possibles en tenant compte des nouveaux objectifs assignés au système futur. L'analyse du système futur conduit à construire :

- Au niveau conceptuel :
 - Le [MCD](#) et le [MCT](#) découlant de la critique de l'existant

- Au niveau organisationnel :
 - Le [MOT](#) du système futur mettant en valeur la nouvelle organisation (y compris en terme de ressources humaines et de nouveaux postes de travail)
- Au niveau logico-physique
 - Le [MLD](#) modèle logique des données et le [MPD](#) modèle physique des données obtenus à partir du [MCD](#) adapté aux choix effectués dans le [MOT](#) et aux besoins d'informations complémentaires
 - Le [MPT](#) modèle physique des traitements (algorithmes, structure des programmes...)

B- Introduction au Modèle conceptuel des données

Définition

Le modèle conceptuel des données (MCD) a pour but de représenter de façon structurée les données qui seront utilisées par le [système d'information](#). Le modèle conceptuel des données décrit la sémantique c'est à dire le sens attaché à ces données et à leurs rapports et non à l'utilisation qui peut en être faite.

On établit le **MCD** après avoir recensé et donné un nom à l'ensemble des données du domaine étudié. Ensuite on étudie les relations existantes entre ces données ([dépendances fonctionnelles](#)), pour aboutir au MCD.

I-MCD:Recueil des données

Exemple

Voilà plusieurs phrases qui décrivent une seule et même personne, prononcées par des personnes différentes.

« Odile Martin est une personne convenable. Et je ne dis pas ça parce c'est une cliente qui m'en prend pour 100F toutes les semaines ! » Le boulanger « MARTIN Odile, habitant 6 rue des prés, Saint Amour (Jura), née le 13/02/73 à Béziers (Hérault) ... » Un inspecteur de police judiciaire « Odile Martin, voilà une personne qui n'a jamais commis aucun péché mortel, qui a fait preuve de beaucoup de générosité. Elle ira au paradis » Saint Pierre Différents interlocuteurs, différents points de vue. Lequel est le plus pertinent ?

Ca dépend du domaine étudié. Mais on ne recense que les informations objectives, formelles

Quand on analyse les données sur un objet (abstrait ou concret) du réel, il est nécessaire de faire le tri entre ce qui est nécessaire pour le système d'information et ce qui ne l'est pas.

- 2° représentation

Odile sera représentée par au moins ces informations :

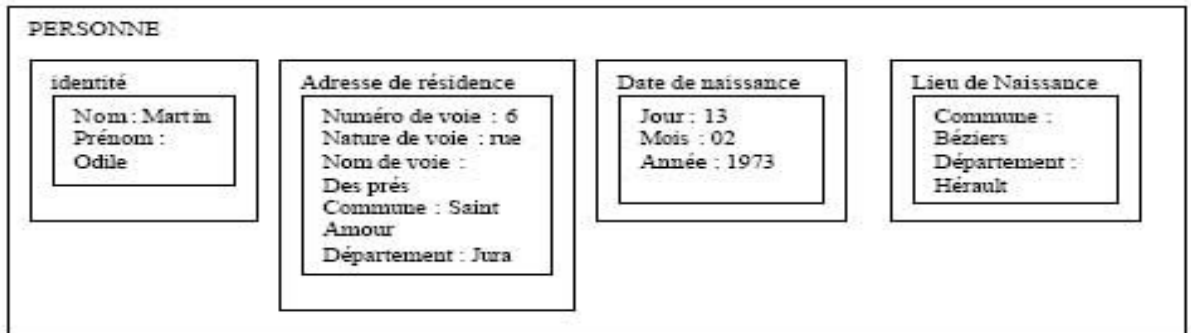
Nom
Prénom
Numéro de la voie
Nature de la voie
Nom de la voie
Commune de résidence
Département de résidence
Jour de naissance
Mois de naissance
Année de naissance

Commune de naissance
Département de naissance

Peut-on décomposer encore ? Par exemple la commune de résidence est composée de 2 mots, de même que le nom de la voie ... va-t-on les décomposer ? Non, ces mots représentent ensemble une information, ne veulent rien dire à eux seuls.

Mais n'avons-nous pas trop décomposé ? Il existe par exemple des liens entre les informations 3, 4 et 5 qui représentent l'adresse d'Odile, entre les informations 8 et 9 qui représentent sa date de naissance... On va alors regrouper ensemble ces informations reliées.

On remarque que toutes les personnes (et Odile est une personne, on dit que c'est une **OCCURRENCE** de personne) de la manière suivante :



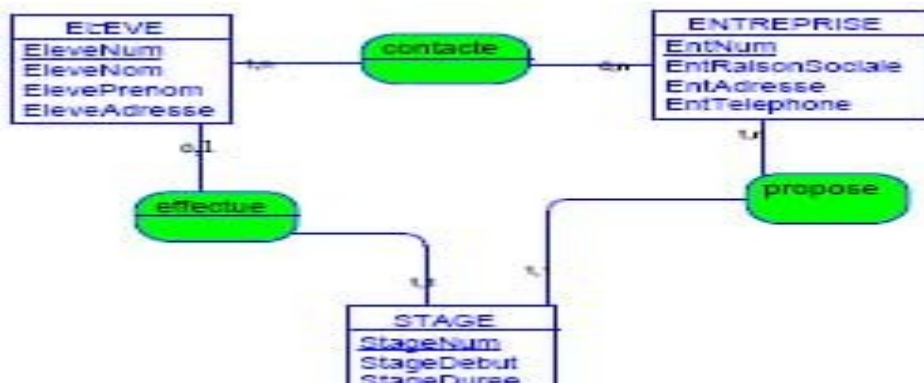
- on peut dire qu'on a créé une entité (un ensemble) PERSONNE dont Odile est une occurrence (ou instance).

Les données de base sont appelées **PROPRIETES** ou **ATTRIBUTS**.

II-MCD:Concepts et du formalisme

Sommaire

- [1 Les concepts de base](#)
 - [1.1 Entité](#)
 - [1.2 Association](#)
 - [1.3 Cardinalités](#)
- [2 La notion d'occurrence](#)
- Exemple



1-Les concepts de base

1.1-Entité

Une entité représente un objet du **SI** (acteur, document, concept, ...), ou plus exactement un ensemble d'objets ayant les mêmes caractéristiques.

Dans une entité, on met les informations nécessaires et suffisantes pour caractériser cette entité. Ces informations sont appelées propriétés. Les propriétés sont collectées lors de l'établissement du dictionnaire des données (voire partie suivante). Les propriétés prennent des valeurs pour chaque occurrence d'une entité.

Une propriété particulière, appelée identifiant, permet de distinguer sans ambiguïté toutes les occurrences de l'entité. L'identifiant est toujours souligné. L'identifiant est une propriété qui ne peut pas changer au cours du temps pour une occurrence.

1.2-Association

C'est un lien entre deux entités (ou plus). On doit lui donner un nom, souvent un verbe, qui caractérise le type de relation entre les **entités**.

Une association possède parfois des **propriétés**.

1.3-Cardinalités

Ce sont des expressions qui permettent d'indiquer combien de fois au minimum et au maximum le lien entre 2 entités peut se produire. Pour une association de 2 entités, il y a 4 cardinalités à indiquer.

Il y a trois valeurs typiques : 0, 1 et N (plusieurs).

Les **cardinalités** traduisent des règles de gestion. Ce sont des règles propre au SI étudié, qui expriment des contraintes sur le modèle.

2-La notion d'occurrence

Il ne faut pas confondre

- ENTITE et OCCURRENCE d'entité,
- ni PROPRIETE et VALEUR de propriété
- ni ASSOCIATION et OCCURRENCE d'association

Reprenons ces concepts :

Une ENTITE est une FAMILLE d'objets ayant les mêmes caractéristiques, appelées propriétés.

Un MEMBRE de la famille est appelé OCCURRENCE d'entité.

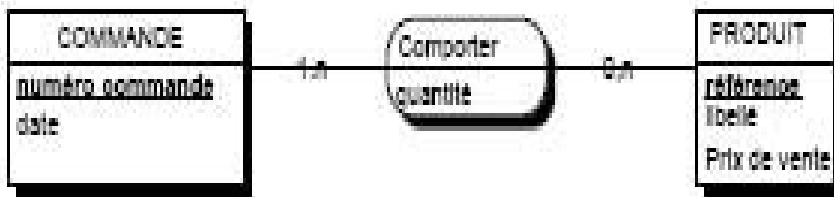
Une entité représente un ENSEMBLE d'OCCURRENCES.

Une PROPRIETE est une information élémentaire qui permet de décrire une entité ou une association. Une propriété peut prendre une VALEUR (c'est l'équivalent d'une variable). (On peut dire qu'une VALEUR est une OCCURRENCE de PROPRIETE)

De même, une ASSOCIATION est un LIEN entre 2 entités ou plus, et une OCCURRENCE d'association est un lien entre 2 OCCURRENCES d'entités.

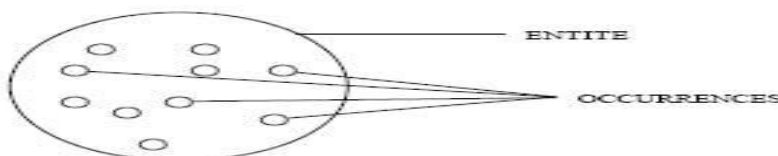
- Exemple

Considérons le schéma suivant :

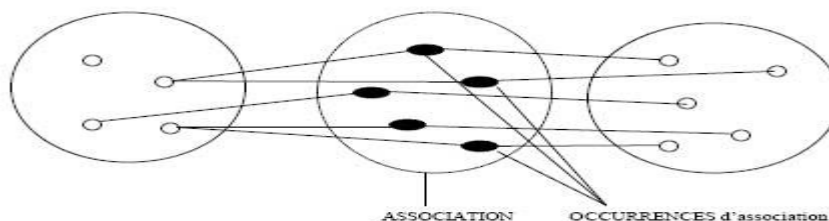


PROPRIETE	VALEUR de PROPRIETE
Référence :	456
Désignation :	Manteau
Prix unitaire :	100€
ENTITE	OCCURRENCE d'entité
PRODUIT	Manteau de référence 456 à 100€
COMMANDE	numéro 123, du 08/10/02
ASSOCIATION	OCCURRENCE d'association
CONCERNER	lien entre la commande 123 et le manteau 456 pour une quantité de 3 unités

- On peut représenter une entité comme un ensemble où les éléments sont les occurrences.



- Une association peut être vue comme l'ensemble des relations entre les occurrences des entités associées



Attention à ne pas confondre occurrences et propriétés

En effet, on peut dire à la fois qu'une entité est un ensemble de propriétés et un ensemble d'occurrences.

Une propriété est une rubrique, un élément d'information qui permet de décrire une entité.

Une entité est décrite par plusieurs propriétés.

Une occurrence est un exemplaire, un élément particulier de la famille représentée par l'entité.

Les propriétés d'une entité prennent des valeurs pour chaque occurrence.

Chaque occurrence d'entité est identifiée de manière unique par un identifiant, qui est une propriété particulière telle que 2 occurrences de l'entité ne peuvent pas avoir la même valeur pour cette propriété.

On représente souvent les occurrences d'une entité sous forme d'un tableau. Les lignes correspondent aux occurrences, et les colonnes correspondent aux propriétés.

- Exemple

numero	nom	premier	sexe	annee_naiss
1	Aloin	Michel	M	81
2	Bouchard	Antoine	M	83
3	Garnier	Sophie	F	82
4	Delabre	Etienne	M	80

III-MCD: Règles de construction

Sommaire

- [1 Règles concernant les propriétés](#)
- [2 Règles concernant les entités](#)
- [3 Règles concernant les associations](#)

1-Règles concernant les propriétés

- Pour qu'il n'y ait pas d'ambiguïté possible, une propriété ne peut pas apparaître plusieurs fois dans un [MCD](#).

Autrement dit, toutes les propriétés d'un [MCD](#) doivent être différentes les unes des autres. Exemples : Une propriété Date ne peut pas apparaître simultanément dans une entité COMMANDE et une entité FACTURE. Si c'était le cas, il faudrait remplacer Date par Date_commande et Date_facture par exemple.

- Toute propriété ne doit avoir à un moment donné qu'une seule valeur pour une occurrence. Par conséquent, un nom de propriété doit toujours être au singulier.

Exemple : Dans la propriété note d'une entité ELEVE, on ne peut avoir qu'une seule note. Si on veut la note de plusieurs matières, il faut mettre autant de propriétés qu'il y a de matières, par exemple noteALSI, noteAMSI, noteDAIGL, etc.

2-Règles concernant les entités

- Toute entité doit comporter un identifiant qui permet de distinguer entre elles toutes les occurrences d'une même entité. L'identifiant est placé en tête des propriétés et il est souligné.
- Pour chaque occurrence d'une entité, il ne doit y avoir qu'une seule valeur pour chacune des propriétés à un instant donné. Cette valeur peut changer au cours du temps, mais à un instant donné, il n'y en a qu'une seule.

3-Règles concernant les associations

- L'identifiant d'une association est implicitement formé par la concaténation des identifiants des entités liées.

On ne représente pas cet identifiant au niveau du [MCD](#). Deux occurrences d'association ne peuvent pas avoir le même identifiant. Pour une occurrence, l'identifiant ne doit jamais changer de valeur (l'identifiant est une propriété constante).

- Une association peut avoir des propriétés, mais ce n'est pas obligatoire.

IV-MCD:Introduction aux cardinalités

Sommaire

- [1 Formalisme](#)
- [2 La cardinalité minimale](#)
 - [2.1 Exemples](#)
 - [2.2 Application](#)
- [3 La cardinalité maximale](#)
 - [3.1 Exemple 1](#)
 - [3.2 Exemple 2](#)
- [4 Conclusion](#)
- [5 Exemple](#)
- [6 Représentation des cardinalités](#)

Les cardinalités d'une entité dans une association exprime le nombre de fois qu'une occurrence de cette entité est impliquée dans l'association, au minimum et au maximum.

1-Formalisme



2-La cardinalité minimale

Elle est exprimée presque toujours par l'une des deux valeurs 0 ou 1.

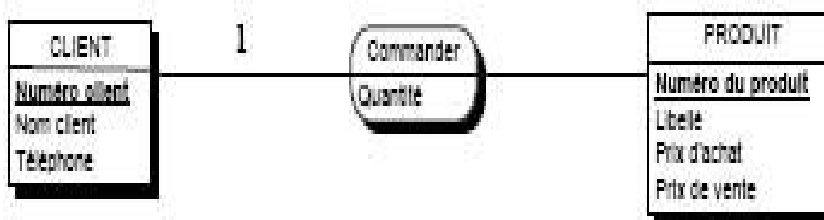
Elle traduit combien de fois au minimum une occurrence de l'entité participe à l'association.

2.1--Exemples

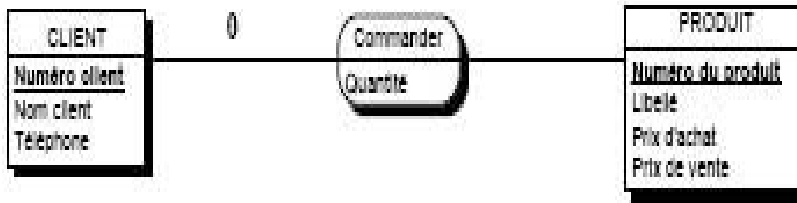
Pour la cardinalité mini entre client et commander il faut se poser la question :

Pour un client donné, combien de fois au minimum il commande ?

Si la réponse est « tout client doit passer au moins une commande sinon ce n'est pas un client » on met la cardinalité mini à 1



Mais on peut très bien imaginer que l'entreprise veut aussi mémoriser les clients potentiels (prospects), qui n'ont encore rien commandé. Dans ce cas, un client peut très bien ne pas avoir encore commandé, et on met la cardinalité mini à 0.



2.2-Application

1. Trouver la question qui permet de trouver la cardinalité minimale pour l'entité produit et l'association commander.
2. Faites deux hypothèses de règle de gestion concernant ce lien et trouver les cardinalités minimales correspondantes.

3-La cardinalité maximale

Elle traduit combien de fois au maximum l'entité peut être en relation avec l'association. Cela peut être plusieurs fois (si c'est un nombre indéterminé, on indique la valeur n) ou une seule fois.

On répond à la question : Combien au maximum l'entité peut participer à l'association ?

Si la réponse est « au plus une fois », la cardinalité maximale prend pour valeur 1.

Si la réponse est « plusieurs », la cardinalité maximale prend la valeur N.

3.1-Exemple 1

- REGLES DE GESTION

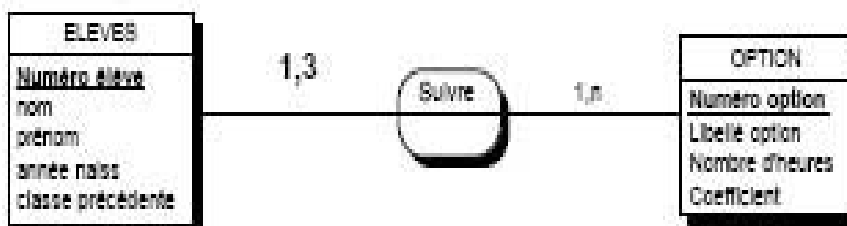
Un salarié est affecté au plus à un seul service. Dans un service sont affectés plusieurs salariés



Il arrive (mais c'est rare) qu'une cardinalité maximale ait une valeur limitée.

3.2-Exemple 2

- REGLE DE GESTION : Un élève doit suivre au minimum une option et au maximum 3 options.

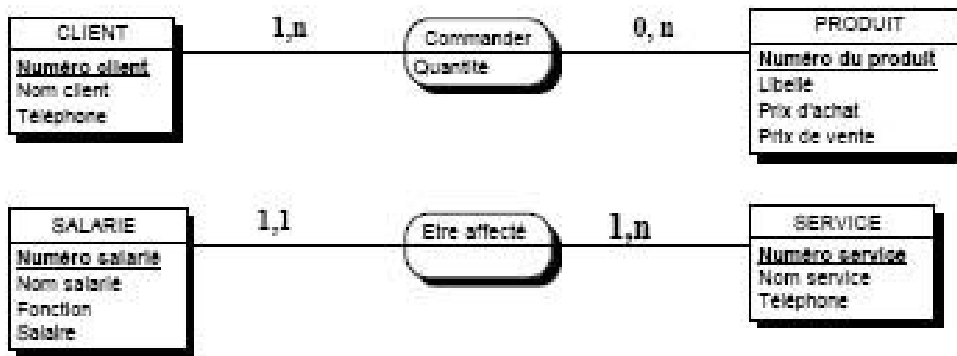


4-Conclusion

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

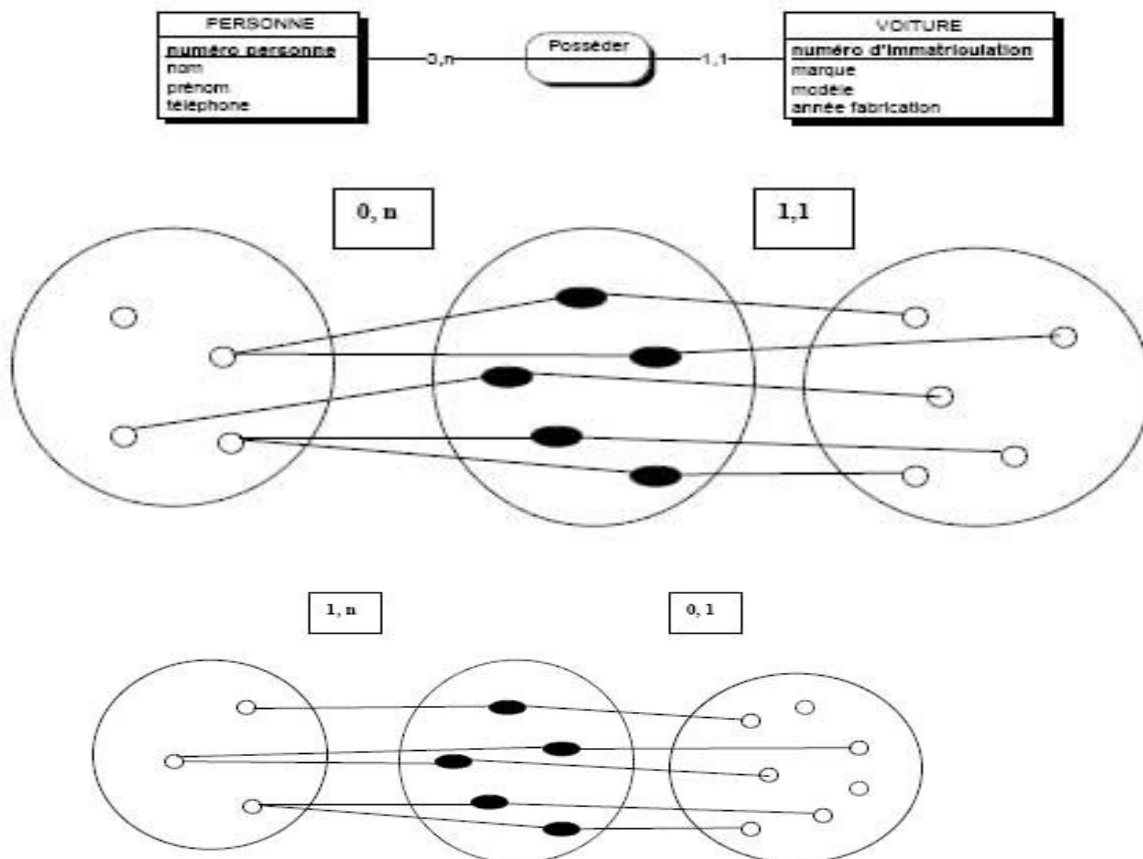
- 0,1 au plus un(e)
- 1,1 un(e) et un(e) seul(e)
- 1, n un(e) ou plusieurs
- 0, n zéro ou plusieurs

5-Exemple

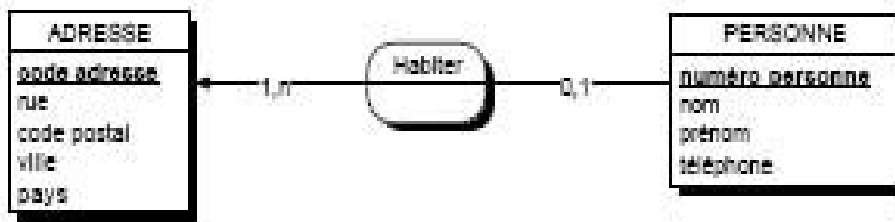


6-Représentation des cardinalités

Une voiture est possédée par une seule personne. Une personne peut posséder de 0 à plusieurs voitures



Une personne a une adresse ou est sans domicile. A une adresse, il y a une ou plusieurs personnes qui y habitent.



C-Les cardinalités et les différents types d'associations

I-MCD:Cardinalités

Sommaire

- [1 Définition et formalisme](#)
- [2 La cardinalité minimale](#)
 - [2.1 Exemple](#)
 - [2.2 Application](#)
- [3 La cardinalité maximale](#)
 - [3.1 Exemple](#)
- [4 Récapitulatif](#)
 - [4.1 Exemples complets](#)

1-Définition et formalisme

Les **cardinalités** sont des **couples de valeur** que l'on trouve entre chaque entité et ses [associations](#) liées.

Donc, **pour une association de 2 entités, il y a 4 cardinalités** à indiquer (2 de chaque côté). Il y a trois valeurs typiques : 0, 1 et N (plusieurs).

Pour les [associations](#) à 2 entités, ce sont des valeurs qui permettent d'indiquer combien de fois au minimum et au maximum une occurrence d'entité peut être liée à une autre occurrence d'entité.

De manière plus générale, les cardinalités d'une entité dans une [association](#) expriment le nombre de fois qu'une occurrence de cette entité peut être impliquée dans une occurrence de l'association, au minimum et au maximum.

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.



2-La 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](#), autrement dit, si une occurrence est obligatoirement associée à une autre ou pas.

2.1-Exemple

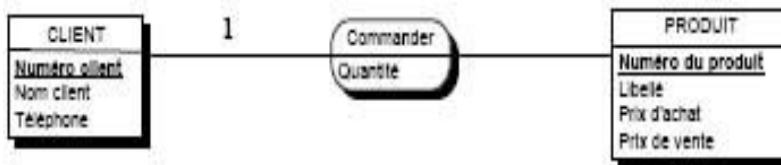
Pour la cardinalité minimale entre client et commander, il faut se poser la question :
Pour un client donné, combien de fois au minimum il commande ?

ou encore mieux

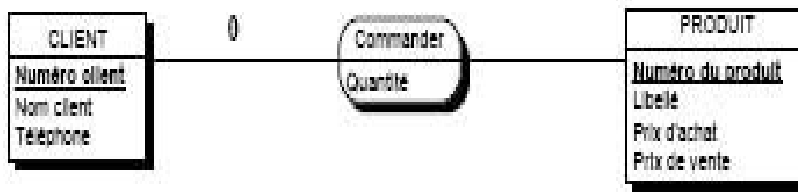
Est-il obligatoire qu'un client effectue une commande de produit ?

Cela dépend des REGLES DE GESTION de l'entreprise.

Si la règle de gestion est « tout client doit passer au moins une commande sinon ce n'est pas un client » on met la cardinalité mini à 1



Mais on peut très bien imaginer que l'entreprise veut aussi mémoriser les clients potentiels (prospects), qui n'ont encore rien commandé. Dans ce cas, un client peut très bien ne pas avoir encore commandé, et on met la cardinalité mini à 0.



En fait, les cardinalités dépendent des règles de gestion propres à l'organisation étudiée. La règle peut très bien être « On ne mémorise que les clients qui ont déjà commandé » ou bien « On veut mémoriser les prospects ».

2.2-Application

- Trouver la question qui permet de trouver la cardinalité minimale pour l'entité produit et l'association commander.

Réponse : Est-il obligatoire qu'un produit ait été commandé pour qu'il soit mémorisé dans la base de donnée? ou bien Peut-il exister dans la base de donnée des produits qui n'ont jamais été commandés?

- Faites deux hypothèses de règle de gestion concernant ce lien et trouver les cardinalités minimales correspondantes.

Réponse :

H1 : l'entreprise peut proposer de nouveaux produits, sur catalogue par exemple, qu'elle mémorise dans sa base de données, mais qui n'ont jamais encore été commandés => cardinalité minimale à 0 (c'est plutôt le cas pour les entreprises de vente)

H2 : tout produit inscrit dans la base de donnée est forcément commandé (par exemple parce que l'entreprise ne propose pas de catalogue, mais fabrique seulement à la commande) => cardinalité minimale à 1 (c'est plutôt le cas pour les entreprises de production)

3-La cardinalité maximale

Elle traduit combien de fois au maximum une occurrence d'entité peut être en relation avec une occurrence de l'association. Cela peut être plusieurs fois (si c'est un nombre indéterminé, on indique la valeur n) ou une seule fois.

Cette cardinalité répond à la question :

- la participation d'une occurrence doit-elle être unique ou bien peut-elle être multiple ? ou bien
- combien de fois au maximum une occurrence est elle impliquée dans l'association ?

Si l'association est binaire (relie seulement deux entités), la question peut être aussi :

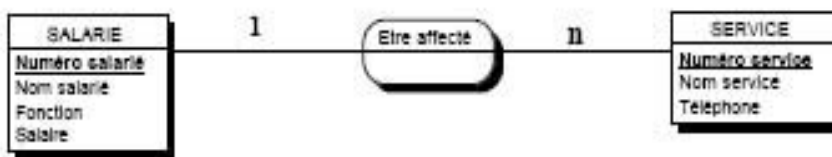
- Une occurrence de l'entité peut-elle être reliée à plusieurs occurrences de l'autre entité ou bien ne peut-elle être reliée qu'à une seule autre occurrence au plus?

Si la réponse est « au plus une fois » (participation unique), la cardinalité maximale prend pour valeur 1.

Si la réponse est « plusieurs » (participation multiple), la cardinalité maximale prend la valeur N.

3.1-Exemple

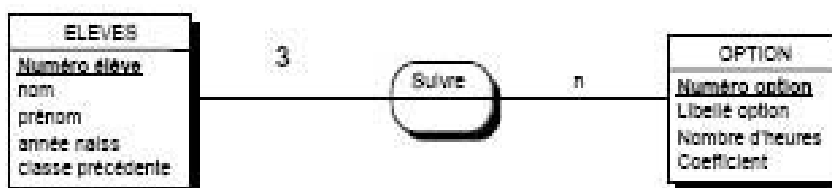
- RG (règles de gestion)
 1. Un salarié est affecté au plus à un seul service.
 2. Dans un service sont affectés plusieurs salariés



Il arrive (mais c'est rare) qu'une cardinalité maximale ait une valeur limitée.

- Exemple :

RG : Un élève peut suivre au maximum 3 options.

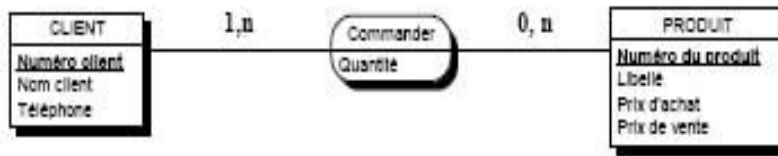


4-Récapitulatif

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

- 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

4.1-Exemples complets



RG : un client commande au moins 1 produit (sous entendu ou plusieurs) et un produit peut ne pas encore avoir été commandé, comme il peut l'avoir été plusieurs fois.



RG : Un salarié est obligatoirement affecté à 1 et 1 seul service. Un service pour exister doit avoir au moins un salarié affecté (sous-entendu, il peut bien sur en avoir plusieurs).

II-MCD:Associations

Sommaire

1 Les associations binaires concernant 2 entités

1.1 Les associations hiérarchiques [1,n]

1.2 Les associations non hiérarchiques [n, n]

1.3 Les associations de type [1, 1]

2 Les autres types d'association

2.1 Les associations réflexives

2.1.1 Exemple d'association réflexive hiérarchique

2.1.2 Exemples d'associations réflexives non hiérarchiques

2.2 Les associations de dimension 3 ou plus (ternaires ou plus)

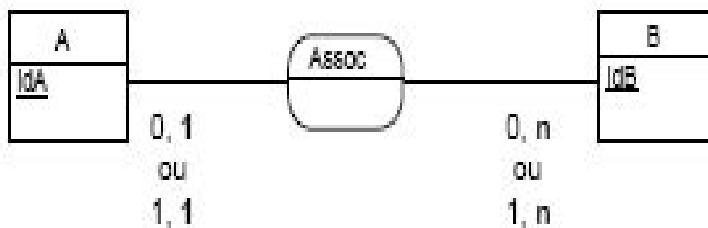
1-Les associations binaires concernant 2 entités

On distingue trois catégories d'associations en fonction des cardinalités maximales de ses branches:

- Les associations hiérarchiques encore appelées associations [1, n] ou associations fonctionnelles
- Les associations non hiérarchiques, encore appelées associations [n, n] ou non fonctionnelles
- Les association [1, 1], les 2 branches ont pour cardinalité maximale 1. Ce cas est rare.

1.1-Les associations hiérarchiques [1, n]

Ce sont les associations où d'un côté 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.



1.2-Les associations non hiérarchiques [n, n]

1.3-Les associations de type [1, 1]

Cas exceptionnel

Ex : un employé travaille sur un ordinateur (p 99 de Modéliser les données)
La facture correspondant à une commande

2-Les autres types d'association

Le plus souvent, une association relie 2 entités avec 2 branches. Ce type d'association est appelé association binaire. Nous avons déjà étudié en détail les différents types d'associations binaires.

Mais une association peut aussi relier 3 ou très rarement 4 entités (quasiment jamais plus). On parle d'association ternaire, quaternaire (n-aire).

Enfin, une association peut aussi relier une entité à elle-même, c'est-à-dire que des occurrences de la même entité sont reliées entre elles. L'association correspondante est qualifiée de réflexive.

Ce sont ces autres types d'association que nous allons étudier maintenant.

2.1-Les associations réflexives

Une association réflexive est une association reliant des occurrences de la même entité.

Ces associations sont quasiment toujours binaire (2 branches).

Pour lire une association réflexive, il est faut connaître le **rôle attribué à chaque branche de l'association**.

Il existe comme pour les autres associations, des associations réflexives hiérarchique et des associations réflexives non hiérarchique

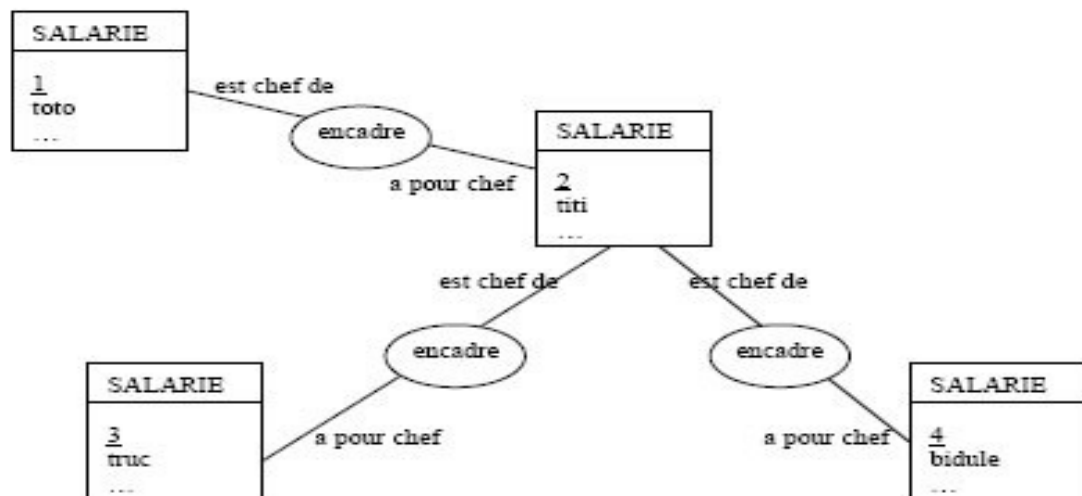
2.1.1-Exemple d'association réflexive hiérarchique

Dans une entreprise bureaucratique, chaque salarié a un seul chef (supérieur hiérarchique direct) sauf le patron et chaque chef a au moins un salarié sous ses ordres. Evidemment, certains salariés ne sont chefs de personnes.



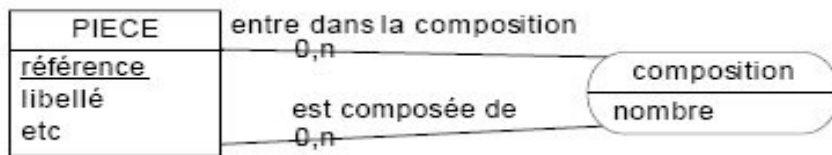
Exemples d'occurrences : diagramme d'occurrences

Remarquez l'importance du rôle dans le diagramme d'occurrence.



2.1.2--Exemples d'associations réflexives non hiérarchiques

Pièces composants, composé = réflexive asymétrique



Personnes frères = réflexive symétrique (rôle pas indispensable car a pour frère = être frère de)

Précision de vocabulaire : dimension d'une association

La dimension d'une association correspond au nombre de branches de l'association. Dans plus de 80% des cas, les associations sont binaires

Dimension 2 : binaire

Dimension 3 : ternaire (rare)

Dimension supérieure : aire (exceptionnel)

Lorsqu'il n'y a pas de réflexivité, le nombre de branches est égal au nombre d'entités reliées.

La dimension est dans ce cas le nombre d'entités concernées par l'association.

Mais la réflexivité existe et donc on peut avoir plusieurs branches de la même association sur la même entité.

Dans ce cas alors, la dimension est supérieure au nombre d'entités concernées.

Dimension = nombre d'entité liées

Une association réflexive (1 entité) est quasiment toujours binaire (2 branches).

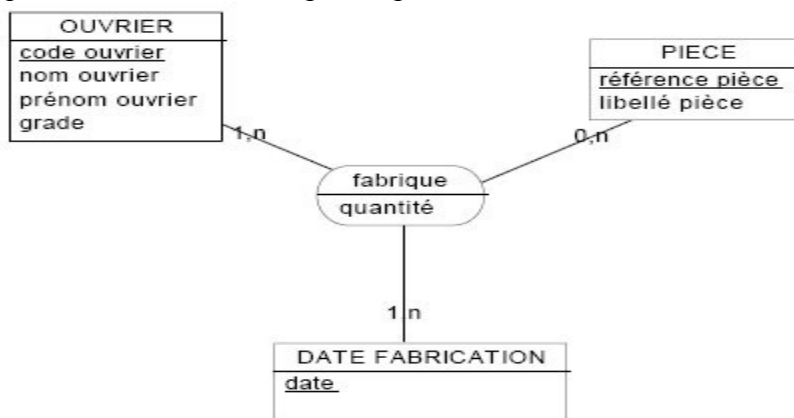
On pourrait aussi avoir une association ternaire impliquant seulement deux entités.

2.2-Les associations de dimension 3 ou plus (ternaires ou plus)

Une association peut relier plus de 2 entités ensemble, le plus souvent trois. On parle alors d'association ternaire. On utilise une association ternaire quand on a besoin de connaître une occurrence de chaque entité pour avoir une information.

Exemple :

Pour connaître la quantité de chacune des pièces fabriquée par chaque ouvrier à une date donnée, on utilise une association ternaire entre OUVRIER, PIECE et DATE. La quantité est une donnée portée par cette association.



Une occurrence de l'association fabrique implique une seule occurrence de chaque entité.

Exemple :

Ouvrier	Pièce	Date	quantité
Dupont	1	17/02	50
Dubois	2	17/02	100
Martin	1	17/02	40
Dupont	3	17/02	55
Dubois	3	17/02	20
Dupont	1	16/02	40
Martin	1	16/02	40

L'association ternaire implique aussi que :

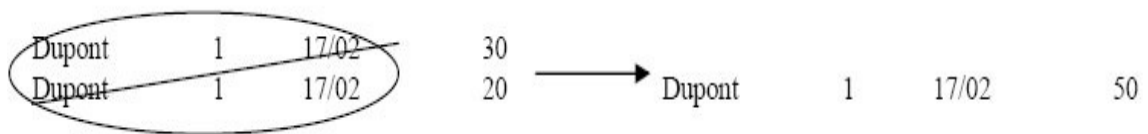
Pour un ouvrier, on peut avoir plusieurs pièces différentes à la même date (il peut fabriquer plusieurs types de pièces le même jour).

Une pièce peut être fabriquée par plusieurs ouvriers différents le même jour.

A des dates différentes, un même ouvrier peut fabriquer les mêmes pièces.

L'identifiant d'une association ternaire est formée de la concaténation (juxtaposition) des identifiants des 3 entités reliées.

Dans notre exemple, on ne peut donc pas avoir plusieurs occurrences de "fabrique" qui concernent Dupont, la pièce n°1 à la date du 17/02.



III-MCD:Représentation des cardinalités

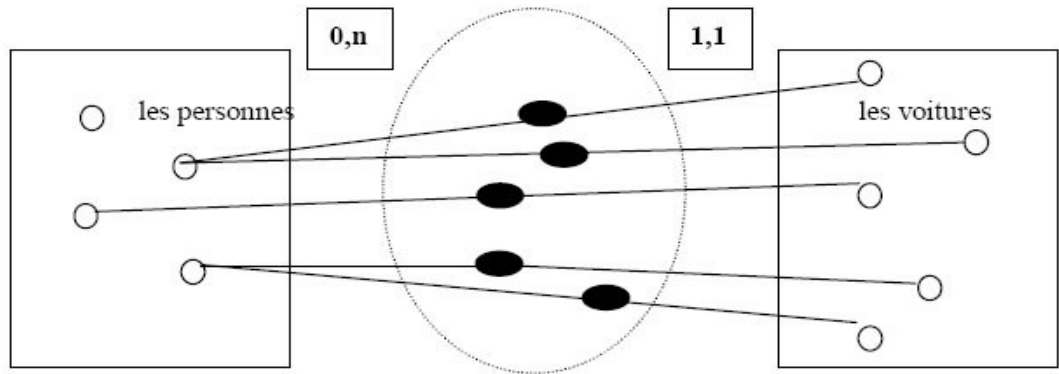
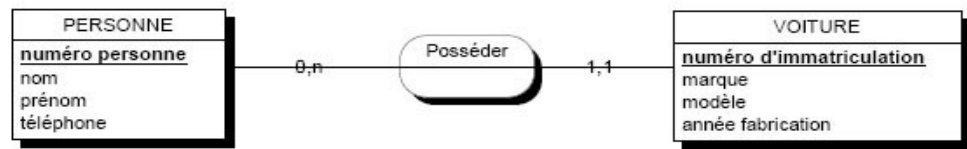
occurrence d'entité



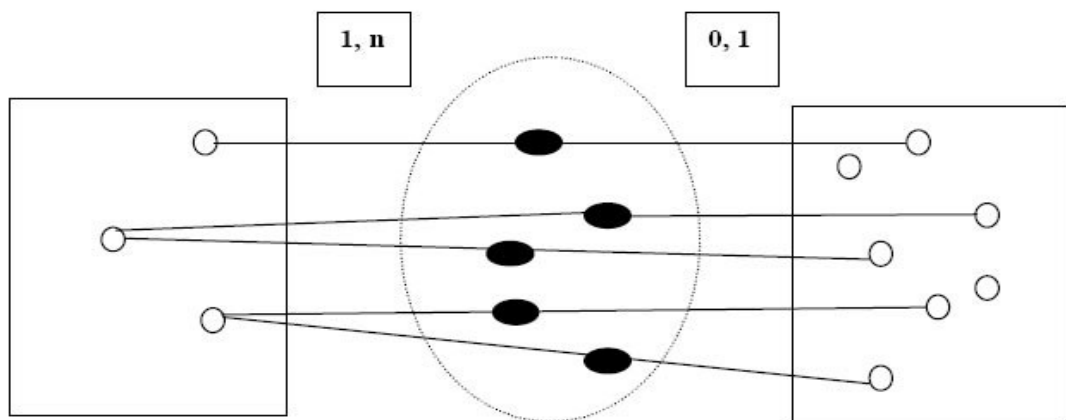
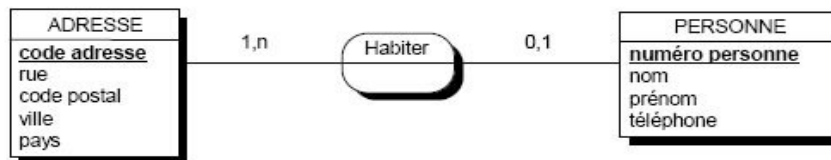
occurrence d'association



- Une voiture est possédée par une seule personne. Une personne peut posséder de 0 à plusieurs voitures.



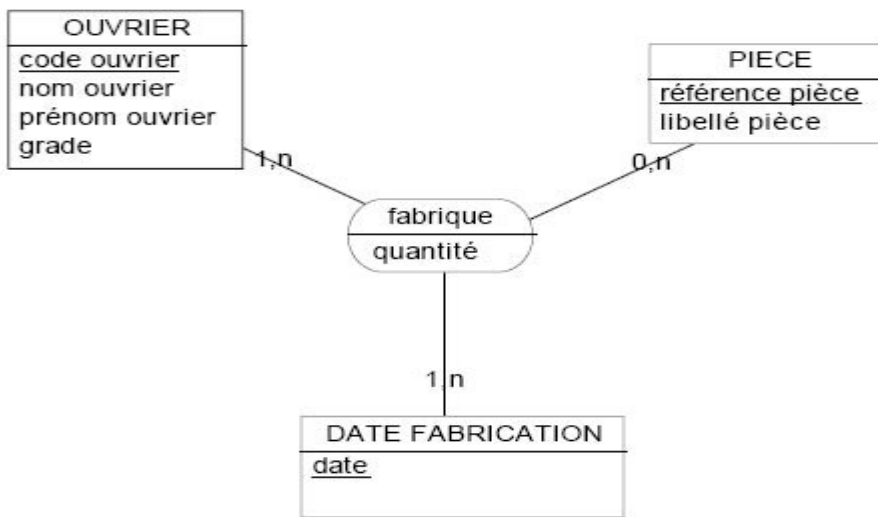
- Une personne a une adresse ou est sans domicile. A une adresse, il y a une ou plusieurs personnes qui y habitent.



IV-MCD: Association ternaire

- Une association peut relier plus de 2 entités ensemble, le plus souvent trois. On parle alors d'association ternaire (dimension 3). On utilise une association ternaire quand on a besoin de connaître une occurrence de chaque entité pour avoir une information.
- Ex :

Pour connaître la quantité de chacune des pièces fabriquée par chaque ouvrier à une date donnée, on utilise une association ternaire entre OUVRIER, PIECE et DATE. La quantité est une donnée portée par cette association.



Une occurrence de l'association fabrique implique une seule occurrence de chaque entité.
Exemple :

Ouvrier	Pièce	Date	quantité
Dupont	1	17/02	50
Dubois	2	17/02	100
Martin	1	17/02	40
Dupont	3	17/02	55
Dubois	3	17/02	20
Dupont	1	16/02	40
Martin	1	16/02	40
...			

L'association ternaire implique aussi que :

1. Pour un ouvrier, on peut avoir plusieurs pièces différentes à la même date (il peut fabriquer plusieurs types de pièces le même jour).
2. Une pièce peut être fabriquée par plusieurs ouvriers différents le même jour.
3. A des dates différentes, un même ouvrier peut fabriquer les mêmes pièces.

L'identifiant d'une association ternaire est formée de la concaténation (juxtaposition) des identifiants des 3 entités reliées.

Dans notre exemple, on ne peut donc pas avoir plusieurs occurrences de "fabrique" qui concernent Dupont, la pièce n°1 à la date du 17/02.



V-MCD:Associations réflexives

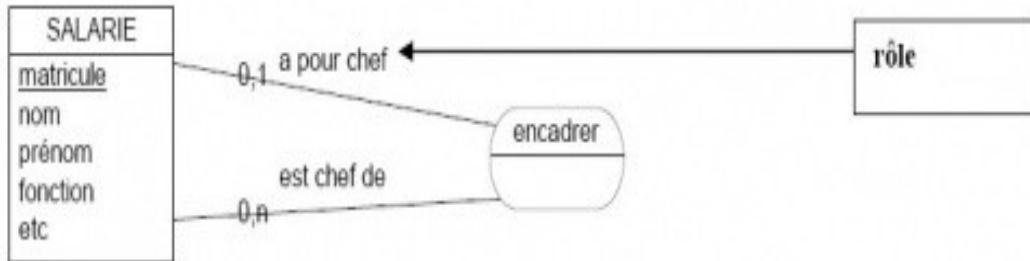
Une association réflexive est une association reliant des occurrences de la même entité.
Ces associations sont quasiment toujours binaire (2 branches).

Pour lire une association réflexive, il est faut connaître le rôle attribué à chaque branche de l'association, c'est à dire le rôle de chaque occurrence dans le lien.

Il existe comme pour les autres associations, des associations réflexives hiérarchique (une des cardinalités maximales à 1) et des association réflexives non hiérarchique (les deux cardinalités maximales à n).

1-Exemple d'association réflexive hiérarchique

Dans une entreprise bureaucratique, chaque salarié a un seul chef (supérieur hiérarchique direct) sauf le patron et chaque chef a au moins un salarié sous ses ordres. Evidemment, certains salariés ne sont chefs de personnes.



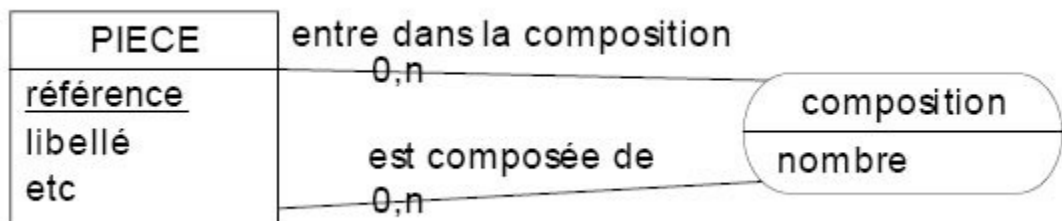
Un même salarié peut à la fois être chef et avoir un chef

Tutu peut être chef de titi et de toto et d'autres salariés, mais il n'a qu'un seul chef au maximum.

toto peut être le chef de coco et avoir pour chef tutu.

2-Exemples d'associations réflexives non hiérarchiques

Pièces composants, composé = réflexive asymétrique



Une pièce peut entrer dans la composition de plusieurs autres pièces (ex : un boulon). Une pièce peut aussi être composée de plusieurs autres pièces (ex : un moteur). Une même pièce peut être composée d'autres pièces et entrer dans la composition d'une ou plusieurs autres pièces (ex : une carte réseau).

Précision de vocabulaire : dimension d'une association

La dimension d'une association correspond au nombre de branches de l'association. Dans plus de 80% des cas, les associations sont binaires

Dimension 2 : binaire

Dimension 3 : ternaire (rare)

Dimension supérieure : n-aire (exceptionnel)

Lorsqu'il n'y a pas de réflexivité, le nombre de branches est égal au nombre d'entités reliées. La dimension est dans ce cas le nombre d'entités concernées par l'association.

Mais la réflexivité existe et donc on peut avoir plusieurs branches de la même association sur la même entité. Dans ce cas alors, la dimension est supérieure au nombre d'entités concernées.

Dimension = nombre d'entité liées

Une association réflexive (1 entité) est quasiment toujours binaire (2 branches).
On pourrait aussi avoir une association ternaire impliquant seulement deux entités.

D-Passage MCD <=> modèle relationnel

I-MR: Rappel sur le modèle relationnel

C'est un modèle LOGIQUE de donnée, celui qui correspond à l'organisation des données dans les bases de données relationnelles (il existe d'autres organisations de bases de données : hiérarchique, réseau, objet,...).

Les SGBD actuels les plus courants sont relationnels (Oracle, SQL Server, Access, MySQL, ...)

Un modèle relationnel est composé de **relations**, encore appelée **tables**. Ces tables sont décrites par des **attributs** ou **champs** (noms de colonnes). Pour décrire une relation, on indique tout simplement son nom en majuscule, suivi du nom de ses attributs entre parenthèses.

L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant **une clé étrangère**, qui correspond à la **clé primaire** de la relation référencée.

Il n'y a pas de **notation officielle** pour repérer les clés primaires et étrangères. C'est à vous d'en adopter une et de l'expliquer en légende. Toutefois, une notation s'est peu à peu répandue :

- on souligne la clé primaire d'un seul trait
- on fait précéder (ou suivre) les clés étrangères du symbole #

Chaque ligne (**tuple ou enregistrement**) d'une table représente une **occurrence** de l'entité ou de l'association correspondante.

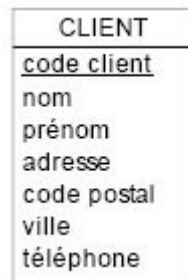
II-MR: Passage du MCD au MR

Sommaire

- 1 Règle 1
- 2 Règle 2
- 3 Exception à la règle 1
- 4 Cas particulier des associations réflexives
 - 4.1 Réflexive hiérarchique
 - 4.2 Réflexive non hiérarchique

1-Règle 1

Toute entité devient une relation ayant pour clé primaire son identifiant.

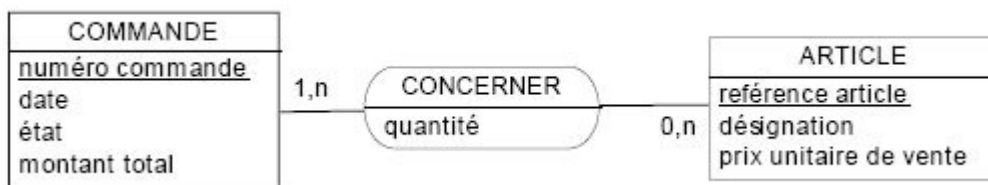


Chaque propriété se transforme en attribut. CLIENT (code_client, nom, prénom, adresse, code_postal, ville, téléphone)

Remarque : contrairement aux propriétés, les attributs ne doivent pas comporter d'espaces.

2-Règle 2

Toute association non hiérarchique (de type [n, n] ou de dimension > 2) **devient une relation**. La clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

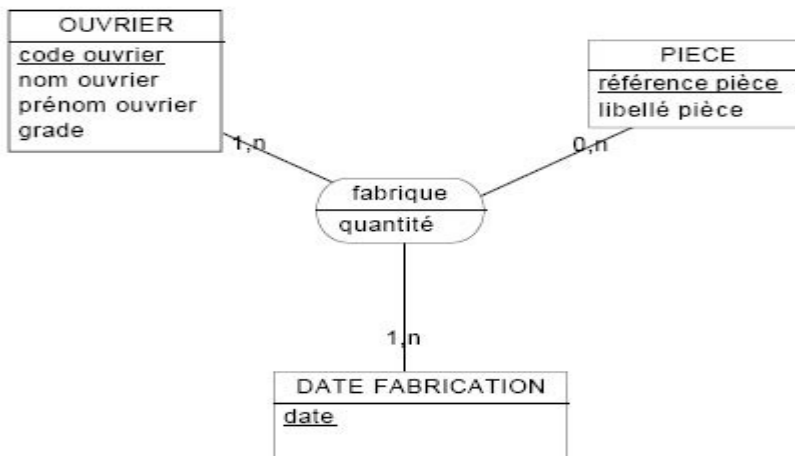


CONCERNER (#numéro_commande, #référence_article, quantité)

Cette règle est valable pour toutes les [associations ternaires](#) (ou quaternaires) qui sont forcément non hiérarchiques ([cardinalités](#) maximales toutes égales à n).

3-Exception à la règle 1

Les entités n'ayant que leur identifiant comme attribut ne deviennent pas des relations, mais des attributs dans les autres relations liées.



Avec ce modèle, on mémorise chaque jour pour chaque ouvrier les pièces qu'il a fabriqué et en quelle quantité. Quand on passe au modèle relationnel, l'entité DATE FABRICATION ne

devient pas une relation, mais un attribut clé dans la relation FABRIQUE issue de l'association.

FABRIQUE (#code_ouvrier, #référence_pièce, date, quantité). Date ici fait partie de la clé primaire, mais n'est pas clé étrangère

4-Cas particulier des associations réflexives

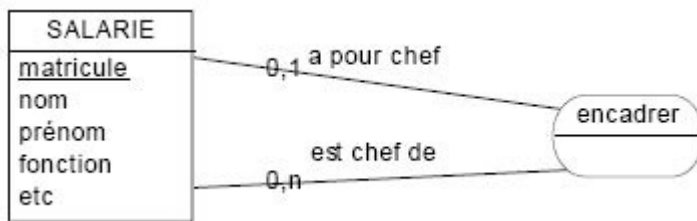
Les **associations réflexives** suivent les règles 2 ou 3 selon les [cardinalités](#) mais posent un problème particulier : une même propriété va se retrouver deux fois en attribut dans la même relation. Il faut alors donner un nom différent et significatif aux deux attributs correspondants.

Dans les réflexives, il est conseillé de nommer les branches par des **rôles** pour pouvoir lire dans le bon sens l'association. Les rôles aident à nommer les attributs correspondant à l'association.

4.1-Réflexive hiérarchique

(Une branche à la cardinalité maxi à 1 et l'autre à n)

Règle n° 2



Lecture de l'association :

Règle n°1: l'identifiant de SALARIE va devenir clé primaire et les autres propriétés des attributs

Règle n°2 : pour traduire l'association [1, n] encadrer, l'identifiant de l'entité SALARIE devient clé étrangère

l'identifiant de SALARIE matricule se retrouve deux fois dans la relation : comme clé primaire et comme clé étrangère

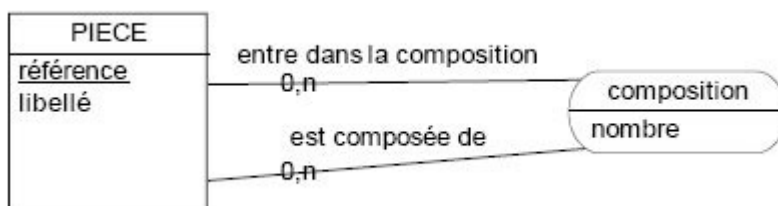
On va donc donner un nom différent et significatif à ces deux matricules, par exemple Un salarié a pour chef 0 ou un seul autre salarié. Un salarié est chef de 0 à n autre(s) salarié.

Traduction en modèle relationnel :

SALARIE (matricule, nom, prénom, fonction,..., #matricule_chef)

4.2-Réflexive non hiérarchique

Règle n°3



- Lecture de l'association

Une pièce entre dans la composition de 0 à plusieurs autres pièces. Une pièce peut être composée de plusieurs autres pièces. Une pièce entre dans la composition d'une autre un certain nombre de fois.

Ex : La pièce "voiture" est composée de 4 pièces "roue". La pièce "roue" est elle-même composée d'une pièce "pneu" et d'une pièce "jante".

Une pièce entrant dans la composition d'une autre est appelée composant. Une pièce composée d'autres pièces est appelée composé. Une roue est à la fois un composant (de voiture) et un composé (de pneu et jante)

- Traduction en modèle relationnel

PIECE (référence, libellé)

COMPOSITION(#référence_composé, #référence_composant, nombre)

E-Normalisation du modèle relationnel

I-Dépendance fonctionnelle

Sommaire

[1 Introduction](#)

[2 Rappels sur la notion de dépendance fonctionnelle](#)

[2.1 Définition](#)

[2.2 Terminologie](#)

[2.3 DF à partir de propriétés concaténées](#)

[2.4 Propriétés des dépendances fonctionnelles](#)

[2.4.1 Union](#)

[2.4.2 Transitivité](#)

1-Introduction

L'objectif de la normalisation est de construire un schéma de [base de données](#) cohérent. Un mauvais schéma logique peut conduire à un certain nombre d'anomalies pendant la phase d'exploitation de la base de donnée. Nous allons voir ces anomalies dans une première partie. Pour qu'un modèle relationnel soit normalisé, il faut qu'il respecte certaines contraintes appelées les formes normales. Les formes normales s'appuient sur les dépendances fonctionnelles entre attributs.

2-Rappels sur la notion de dépendance fonctionnelle

La construction du MCD mais également du modèle relationnel correspondant, repose presque entièrement sur le concept de dépendance fonctionnelle. C'est ce concept qui permet

de passer d'un ensemble de propriétés non structuré à un modèle conceptuel des données formé d'entités et d'associations et au modèle relationnel correspondant.

2.1-Définition

On dit que b est en dépendance fonctionnelle (DF) de a si à une valeur quelconque de la propriété a, on ne peut faire correspondre qu'une seule valeur au plus de la propriété b.

On note $a \longrightarrow b$
(Source) \rightarrow (but)

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).

- Exemple :

Num client \longrightarrow Nom client

Il existe une DF entre num client et Nom client, car si on connaît une valeur de la propriété num client (ex : 4553), il ne peut lui correspondre qu'une seule valeur de la propriété nom (ex : Duval).

La réciproque est fautive :

Nom client $\not\longrightarrow$ Num client n'est pas une DF

Si l'on connaît la valeur de la propriété Nom client, on ne peut pas en déduire la propriété Num client, car il peut y avoir des homonymes.

2.2-Terminologie

Si on a une dépendance fonctionnelle $a \longrightarrow b$, on peut employer les expressions suivantes de façon équivalente :

- Il y a une dépendance fonctionnelle de a vers b
- b est en dépendance fonctionnelle de a
- b dépend fonctionnellement de a
- a est la source et b est le but (ou la cible) de la dépendance fonctionnelle

2.3-DF à partir de propriétés concaténées

(Partie gauche composée de plusieurs attributs)

Il peut exister des dépendances fonctionnelles à partir de propriétés concaténées, c'est-à-dire qui forment un tout indissociable, comme si elles étaient soudées.

On note par un + cette concaténation.

Exemple : Considérons une commande qui comporte plusieurs produits Num_Commande + Ref_Produit \longrightarrow quantité commandée

Si on n'a seulement le numéro de la commande, on ne peut pas en déduire la quantité commandée, car il faut aussi savoir de quel produit. De même, on ne peut pas savoir la quantité commandée d'un produit si on ne sait pas de quelle commande. Il faut bien connaître à la fois la commande et le produit (leurs identifiants respectifs) pour en déduire la quantité commandée.

Les dépendances fonctionnelles dont la source est formée de plusieurs propriétés doivent être **élémentaires**, c'est-à-dire ne pas être créées artificiellement.

Ex : Num Commande + Num Client date commande
n'est pas une DF élémentaire car on n'a pas besoin du numéro de client pour connaître la date de commande, il suffit de connaître le numéro de la commande. La propriété Num Client ne

2.4-Propriétés des dépendances fonctionnelles

Les dépendances fonctionnelles ont les propriétés suivantes :

2.4.1-Union

Si on a deux DF ayant la même source, on peut les rassembler en une seule, en séparant les cibles par une virgule. Si $a \twoheadrightarrow b$ et $a \twoheadrightarrow c$

Ex : Référence \twoheadrightarrow Désignation et Référence \twoheadrightarrow Prix de vente unitaire

Alors par union on a : Référence \twoheadrightarrow Désignation, Prix de vente unitaire

Lors du tracé du graphe des dépendances fonctionnelles, l'union permet de regrouper sur une seule ligne toutes les dépendances fonctionnelles ayant la même source.

2.4.2-Transitivité

Si $a \twoheadrightarrow b$ et $b \twoheadrightarrow c$ alors on a $a \twoheadrightarrow c$

Ex : Num Médecin \twoheadrightarrow Code Service et Code Service \twoheadrightarrow Num Hopital

Alors on a Num Médecin \twoheadrightarrow Num Hopital

Les DF qui peuvent être déduites par transitivité de deux autres DF (qui ne sont pas directes) doivent être éliminées car elles sont alors redondantes.

Il ne reste alors que les **DF directes**, c'est-à-dire celles qui ne peuvent pas être retrouvées par transitivité.

\twoheadrightarrow Attention toutefois à la signification des dépendances. Une dépendance fonctionnelle qu'on peut retrouver par transitivité ne doit pas être supprimée si elle n'a pas le même sens que la transitivité des deux autres, car il y aurait perte d'information.

II-MR:Intérêt de la normalisation

Exemple

Pour vous montrer l'intérêt de la normalisation d'une base de donnée relationnelle, voyons les problèmes que peuvent poser l'utilisation d'une base de donnée basée sur un modèle relationnel non normalisé.

Soit le schéma de relation :

FOURNISSEUR (NomFournisseur, AdresseFournisseur, Produits, Prix)

Modèle en extension

NomFournisseur	AdresseFournisseur	Produit	Prix
Lebras	10, Rue des Gras - Clermont	Chaise	20
		table	35
Dupont	86, Rue de la République - Moulins	Bureau	60
Lajoie	26, Rue des Dômes - Vichy	Lit	50
Dupont	39, Rue des Buttes - Moulins	Lampe	18
		Table de chevet	25

- **1°problème :**

Il n'y a pas de clé primaire : on ne sait pas si les deux Dupont sont différents ou pas (si c'est le même Dupont, il y a une des deux adresses qui est fausse).

- **2°problème :**

L'adresse n'est pas décomposée. Si on veut par exemple rechercher tous les fournisseurs qui habitent la même ville, ça ne va pas être possible

- **3°problème :**

Une relation (table) correspondant à ce schéma pourra éventuellement contenir plusieurs produits pour un même fournisseur.

Dans ce cas, il faudra faire face à un certain nombre de problèmes :

- l'adresse du fournisseur sera dupliquée dans chaque n-uplet (redondance),
- si on souhaite modifier l'adresse d'un fournisseur, il faudra rechercher et mettre à jour tous les n-uplets correspondant à ce fournisseur,
- si on insère un nouveau produit pour un fournisseur déjà référencé, il faudra vérifier que l'adresse est identique,
- si on veut supprimer un fournisseur, il faudra retrouver et supprimer tous les n-uplets correspondant à ce fournisseur (pour différents produits) dans la table.
- **La normalisation élimine les redondances**, ce qui permet :
 - une diminution de la taille de la base de donnée sur le disque
 - une diminution des risques d'incohérence
 - d'éviter une mise à jour multiple des mêmes données

III-MR: 3 formes normales

Les 3 formes normales du modèle relationnel

Sommaire

[1 1er forme normale](#)

[2 2ème forme normale](#)

[3 3ème forme normale](#)

[4 Application des règles](#)

[5 Résumé](#)

1-1er forme normale

Une relation est normalisée en première forme normale si :

- Elle possède une clé identifiant de manière unique et stable chaque ligne
- Chaque attribut est monovalué (ne peut avoir qu'une seule valeur par ligne)
- Aucun attribut n'est décomposable en plusieurs attributs significatifs

Contre-exemple :

EMPLOYE (Nom, Prénom, Enfants, Diplômes)

NOM	DIPLOMES		ENFANTS	
	Nature	Année	Prénom	Année de naissance
DUPONT	Bac	1975	Sophie	1985
	Licence	1980	Simon	1993
			Lucie	1996

Cette relation n'est pas en première forme normale

Un employé peut avoir plusieurs enfants et plusieurs diplômes. En outre, ces attributs sont décomposables : diplôme est décomposable en Nature et Année, et Enfants est décomposable en Prénom et Année de Naissance.

2-2ème forme normale

Une relation R est en deuxième forme normale si et seulement si :

- Elle est en 1FN
- Et tout attribut non clé est totalement dépendant de toute la clé.

Autrement dit, aucun des attributs ne dépend que d'une partie de la clé.

La 2FN n'est à vérifier que pour les relations ayant une clé composée. Une relation en 1FN n'ayant qu'un seul attribut clé est toujours en 2FN

Contre-exemple :

Cette relation est en première forme normale (existence d'une clé valide et aucun attribut n'est décomposable)

MAIS elle n'est pas en 2° forme normale car on a DésignationProd ne dépend pas de toute la clé mais seulement de référenceProd:

RéférenceProd \longrightarrow DésignationProd

Pour connaître l'attribut désignationProd, on n'a pas besoin de connaître le numéro de commande.

3-3ème forme normale

Une relation est en 3° forme normale si et seulement si :

Elle est en 2° forme normale et tout attribut doit dépendre **directement** de la clé, c'est-à-dire qu'aucun attribut ne doit dépendre de la clé par transitivité.

Autrement dit, aucun attribut ne doit dépendre d'un autre attribut non clé.

Contre-exemple :

CLIENT (Num_client, Nom_client, code_categ, nom_categ) Cette relation n'est pas en 3FN car num_client \longrightarrow nom_categ n'est pas une dépendance directe.

En effet, on a aussi num_client \longrightarrow num_categ \longrightarrow nom_categ

4-Application des règles

Si l'une des 3 règles n'est pas vérifiée, cela indique une erreur dans le modèle relationnel et il faut alors modifier pour que les 3 règles soient vérifiées pour toutes les relations.

On vérifie les règles dans l'ordre. Si la première forme normale n'est pas respectée, pas la peine de vérifier la 2FN. Et si la 2FN n'est pas vérifiée, inutile de vérifier la 3FN.

5-Résumé

Modèle normalisé = relations avec:

- Une clé, qui permet de distinguer chaque occurrence
- Des attributs élémentaires (1FN)
- En dépendance de TOUTE la clé (2FN),
- Et RIEN QUE de la clé (3FN)

On parle aussi de normalisation pour un **MCD**. Un **MCD** qui donne un **MR** normalisé est qualifié aussi de normalisé.

F-Merise:Les diagrammes de flux

Les diagrammes de flux répondent à la question : Que fait le système ?

En ce sens, ce sont des modèles FONCTIONNELS (qui décrivent les fonctions)

Il existe 2 types principaux de diagrammes de flux :

1. Le modèle de contexte (**MC**) où le domaine d'étude est vu comme une boîte noire. On ne représente que les flux extérieurs au domaine.
2. Le modèle de flux de données (**DFD**) ou encore modèle de flux conceptuels (**MFC**) où l'on détaille les activités du domaine d'étude. On représente aussi les flux internes au domaine.

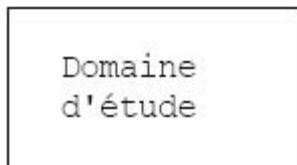
I-Merise: Vocabulaire associé aux modèles de flux

Sommaire

- [1 Domaine d'étude](#)
- [2 Acteur externe](#)
- [3 Domaine connexe](#)
- [4 Flux de données](#)

1-Domaine d'étude

Le domaine d'étude est un **sous-ensemble cohérent** de l'entreprise ou de l'organisme, bien **délimité** et **formant le contenu du sujet à étudier**.



Dans les modèles de flux, le domaine d'étude est représenté par un rectangle à trait plein. Le nom du domaine est placé à l'intérieur du rectangle.

2-Acteur externe

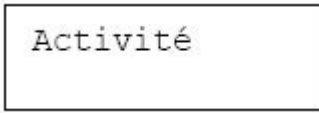
Un **acteur externe** est un **élément émetteur ou récepteur de données, situé hors du système d'information étudié**.



Dans les modèles de flux, un acteur externe est représenté par un cercle plein. Le nom de l'acteur est placé à l'intérieur du cercle.

3-Domaine connexe

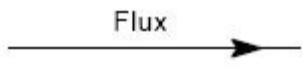
Un **domaine connexe** est un **composant du système d'information interagissant avec le domaine d'étude**. C'est un acteur interne à l'entreprise, mais externe au domaine d'étude.



Dans les modèles de flux, une activité est représentée graphiquement par un rectangle. Le nom de l'activité est placé à l'intérieur du rectangle.

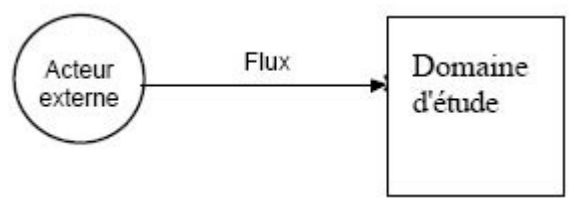
4-Flux de données

Un flux est un transfert d'informations entre composants du système. Le composant peut être un domaine, une activité ou un acteur externe.



Dans les modèles de flux, un flux de données est représenté graphiquement par une flèche orientée du composant émetteur du flux vers le composant récepteur. Le libellé du flux est inscrit en regard de la flèche tracée.

Formalisme graphique illustrant par exemple un échange entre un acteur externe et le domaine d'étude :



II-Merise:Modèle de contexte

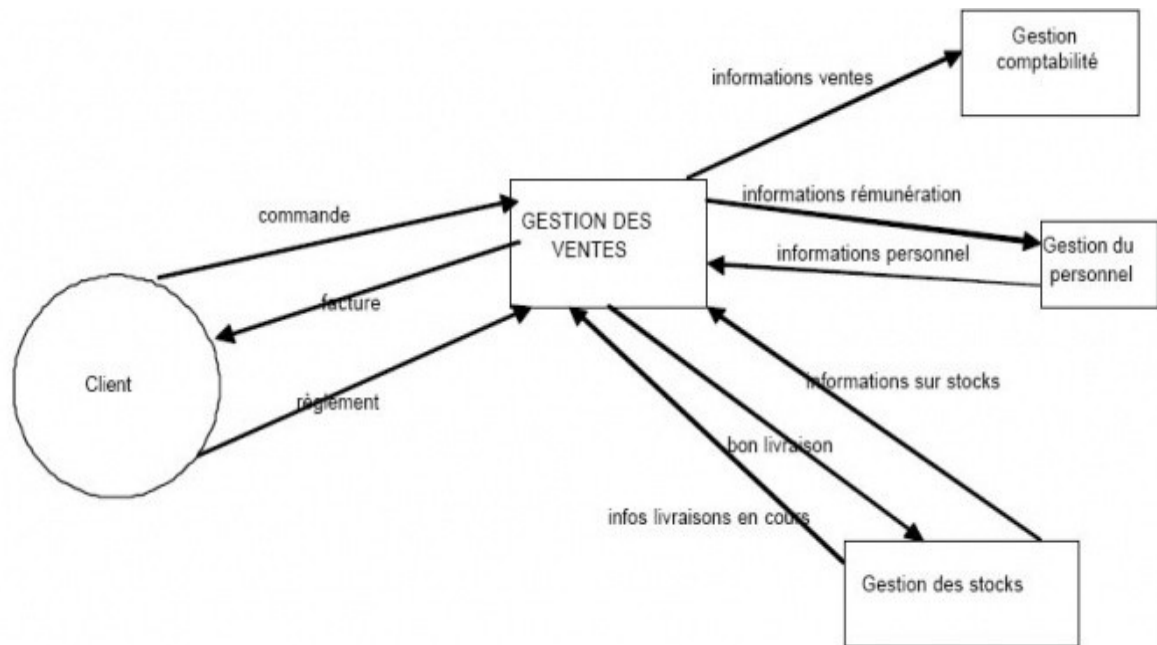
Le modèle de contexte sert à représenter les interactions entre le domaine d'étude et l'environnement, et entre le domaine d'étude et les éventuels domaines connexes.

Le domaine d'étude y est représenté comme une boîte noire.

Le modèle de contexte utilise les concepts suivants :

- le domaine d'étude
- les acteurs externes
- les flux de données
- les domaines connexes

Exemple de modèle de contexte : au sein d'une société commerciale, on étudie le domaine « gestion des ventes ».



Remarque

- on ne fait pas apparaître les flux entre acteurs externes et domaines connexes, ou entre les domaines connexes.

Exemple : le flux correspondant à la livraison de marchandises n'apparaît pas.

II-Merise:MFC ou DFD

Sommaire

- [1 Modèle de flux conceptuel\(MFC\) ou diagramme de flux de données\(DFD\)](#)
 - [1.1 Définition](#)
 - [1.2 Exemple](#)
 - [1.3 Règles de décomposition des activités](#)

1-Modèle de flux conceptuel (MFC) ou diagramme de flux de données (DFD)

1.1-Définition

Ce modèle permet de décider quelles activités, inter-reliées de quelle manière, permettront de résoudre au mieux le problème posé, et cette réflexion est menée sans s'encombrer dans un premier temps du comportement du système (ordonnancement, règles d'émission, synchronisations...).

Les modèles de flux conceptuels permettent de décomposer le domaine d'étude en activités. **Il n'y a pas ici de notion d'organisation mais d'objectifs à réaliser.** On représente les flux entre activités et avec l'environnement.

Pour analyser les communications et les activités, on procède par « zooms » successifs sur le domaine étudié pour élaborer des modèles de plus en plus détaillés qui permettront d'avoir une cartographie détaillée du système et de préparer le passage au modèle conceptuel de représentation des traitements.

Le modèle de contexte est également appelé le diagramme de flux de données de niveau 0.

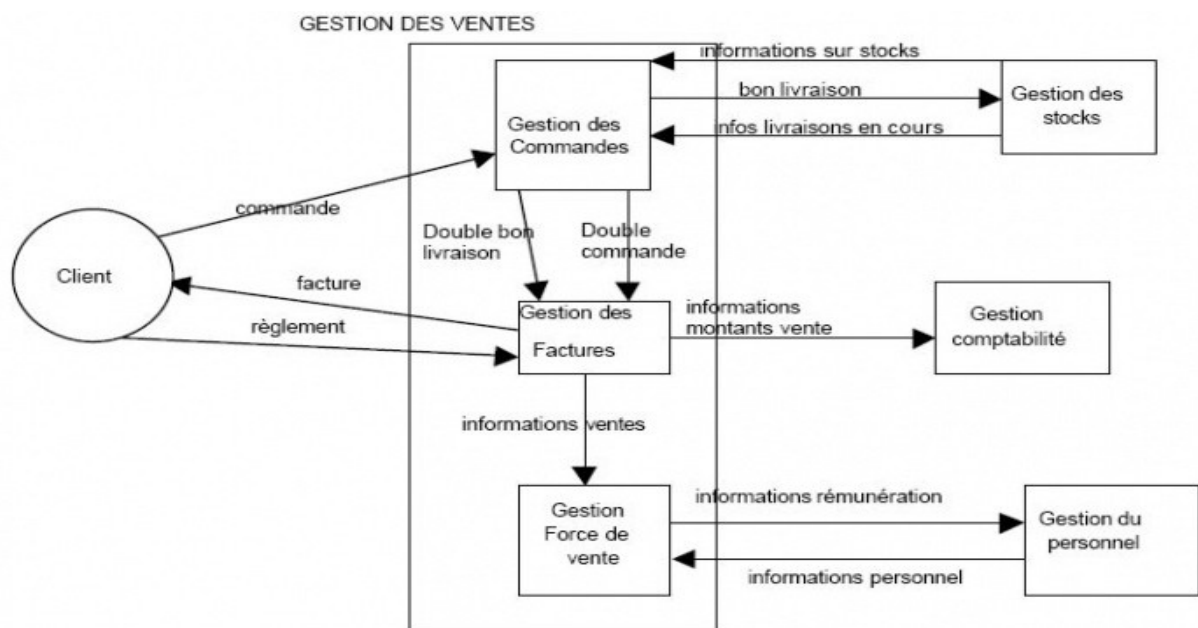
Nous obtenons ensuite des diagrammes de premier, deuxième, troisième, ... niveau, par éclatements successifs des activités à chacun de ces niveaux.

La décomposition d'un domaine ou d'une activité en plusieurs activités peut faire apparaître de nouveaux flux dus :

1. à l'échange d'informations entre activités
2. à la décomposition d'un flux présent au niveau n en plusieurs flux au niveau n+1.

1.2-Exemple

Si on reprend le modèle de contexte précédent, on s'aperçoit que le domaine des ventes peut-être éclaté en trois activités. Nous obtenons ainsi le diagramme de flux de données de niveau 1 :



Remarque

- On peut décomposer le modèle de flux de niveau 1 en un modèle de flux de niveau 2 et etc. ... jusqu'à arriver à un modèle où l'activité correspond à une opération au sens Merise (règle d'ininterruption).

Exemple : l'activité « gestion des factures » peut être encore décomposée en activités « facturation » et « Suivi des règlements ».

1.3-Règles de décomposition des activités

- Pourquoi décomposer ?
 1. Pour désagréger les groupes de flux, pour arriver à la définition des flux-types du domaine étudié
 2. Pour préparer l'étude dynamique du système d'information, pour arriver à l'identification des processus et des opérations conceptuelles
- Comment décomposer ?

1. Identifier les groupes de données entrant et sortant du domaine d'étude pour construire le modèle de contexte
 2. Identifier les activités générant ou traitant les flux de données pour construire le DFD de niveau 1 (approche par les données) ou identifier une activité de niveau 1 comme un ensemble d'activités participant à une même finalité (approche par les objectifs)
- Jusqu'où décomposer ?

Lorsqu'une activité a atteint le niveau d'interruptibilité (dès lors que le traitement est déclenché, il se déroule sans attente de ressources complémentaires extérieures). L'activité est alors une opération conceptuelle qui sera décrite lors de l'étude dynamique du SI dans le [MCTA](#).

G-Les Modèles de traitements (MCT et MCTA)

I-Merise:MCT

Sommaire

- [1 Modélisation des Traitements](#)
- [2 Le MCT](#)
- [3 Les concepts du MCT](#)
 - [3.1 L'événement](#)
 - [3.2 L'opération](#)
 - [3.3 Le résultat](#)
 - [3.4 La synchronisation](#)
 - [3.5 Règle d'émission](#)
 - [3.6 Rôle des règles de gestion](#)
 - [3.7 Le processus](#)
- [4 Méthode d'établissement d'un MCT](#)
 - [4.1 Règles de validation](#)
 - [4.2 Cas particuliers](#)
 - [4.2.1 Cycles](#)
 - [4.2.2 Les événements consommables](#)

1-Modélisation des Traitements

La modélisation des traitements pour but de faire la représentation dynamique du système d'information, c'est-à-dire de représenter l'enchaînement des traitements réalisés.

Cette modélisation s'effectue à deux niveaux :

1. le niveau conceptuel où l'on s'intéresse aux opérations en dehors de toute mise en oeuvre organisationnelle.
2. le niveau organisationnel, où se pose les questions du qui, où, quand

2-Le MCT

L'objectif du MCT est de répondre à la question QUOI faire par rapport à un événement. C'est la chronologie qui importe. Autrement dit, le MCT est une représentation de la

succession des règles de gestion dont l'entreprise veut se doter pour répondre aux événements auxquels elle doit faire face, du fait de son activité et de son environnement.

3-Les concepts du MCT

3.1-L'événement

C'est une sollicitation du système d'information qui génère une réaction de la part de celui-ci.

Un événement peut être externe au domaine étudié (ex : commande client) ou interne au SI, souvent le résultat d'un processus antérieur (ex : ordre de préparation). Un événement peut-être aussi temporel, c'est-à-dire lié à des dates qui rythment l'exécution de certains traitements (délai de maintenance, relances)

3.2-L'opération

C'est un ensemble d'actions accomplies par le système d'information en réaction à un événement ou à une conjonction d'événements et non interruptibles par un événement externe.

Remarque : Une opération déclenche au moins un résultat. Une opération est représentée par un verbe ou mieux un substantif (ex : Préparer la commande ou préparation de la commande)

3.4-Le résultat

Un résultat peut-être un document, un message externe, un nouvel état du SI (nouvelle situation, nouvelles données), créé par une opération, qui peut lui-même jouer le rôle d'événement.

Un résultat externe représente une information envoyée à l'extérieur du SI (ex : facture)

Un résultat interne est un nouvel état du système d'information (ex : ordre de préparation)

3.5-La synchronisation

C'est une condition booléenne (ET / OU) traduisant les règles de gestion que doivent respecter les événements pour déclencher une opération. Dans le cas ET, elle marque qu'un événement déjà là doit en attendre un ou plusieurs autres.

Remarque : Pour qu'il soit question de synchronisation, il faut la présence de plusieurs événements déclencheurs ; aussi, le symbole de synchronisation est laissé à blanc dans le cas d'un événement unique.

Si tous les événements entrants sont liés par le même opérateur, on peut seulement faire figurer l'opérateur dans le symbole de synchronisation sinon il faut numéroter les événements (a, b, c ...) et constituer l'expression à l'aide des événements et des opérateurs. (ex : (a ET b) OU c) Règle d'émission

Condition, traduisant les règles de gestion, qui permet d'exprimer des conditions de sortie des résultats.

Remarques : L'expression d'une règle d'émission peut être composée de plusieurs conditions élémentaires reliées par les opérateurs ET, OU. On peut également utiliser l'opérateur NON pour exprimer la négation d'une condition.

3.6-Rôle des règles de gestion

Recensées lors de l'étude de l'existant ou définies pour le futur SI, elles décrivent les enchaînements d'opérations. Elles rendent possible le regroupement des actions au sein d'une seule opération non interruptible au niveau conceptuel.

3.7-Le processus

C'est un enchaînement synchronisé d'opérations au sein d'un même domaine, généralement déclenché par un événement externe (externe au domaine ou au SI tout entier).

Un MCT est la représentation de l'enchaînement des opérations d'un processus.

4-Méthode d'établissement d'un MCT

1. Réaliser le graphe des flux (représentation des acteurs, événements et résultats externes
2. Pour chaque événement, recenser les opérations déclenchées, et/ou les événements 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

4.1-Règles de validation

Une opération ne peut pas être interrompue par l'attente d'un événement externe.

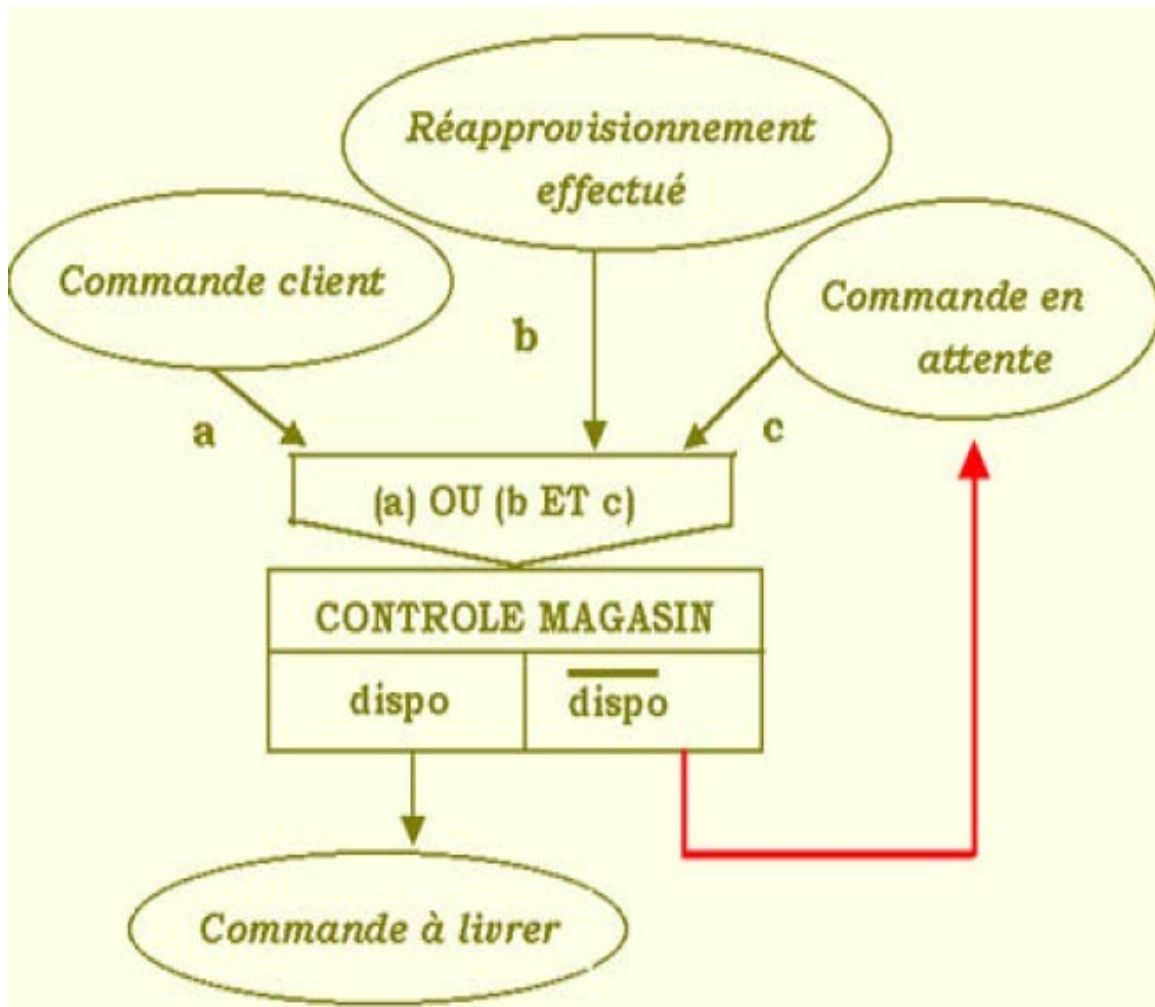
Si tel est le cas, il faut décrire une seconde opération déclenchée par cet événement en attente.

4.2-Cas particuliers

4.2.1-Cycles

Pour une même opération, l'événement déclencheur et l'événement résultat sont identiques.

Une commande de client est servie si les marchandises sont disponibles, sinon cette commande est mise en attente. Les commandes en attente sont réexaminées lors de chaque réapprovisionnement afin de les livrer.



Il faut s'assurer que le cycle a bien un début et une fin : ici, c'est l'événement "Réapprovisionnement effectué" qui permet d'éviter que le cycle ne se déroule sans fin sur les commandes en attente.

Noter l'importance du concept d'occurrences d'événement :

-une occurrence de Réappro est-elle autorisée à consommer toutes les occurrences de Commande en attente ?

-si le réapprovisionnement est insuffisant, de nouvelles occurrences de Commande en attente doivent-elles être créées ?

4.2.2-Les événements consommables

Un client non satisfait peut retourner un article. On lui fait un "avoir". Il peut alors le présenter en caisse pour être remboursé de son achat. Il peut encore remplacer l'article défectueux par un autre, auquel cas on annule l'avoir.



Ce qui pose question :

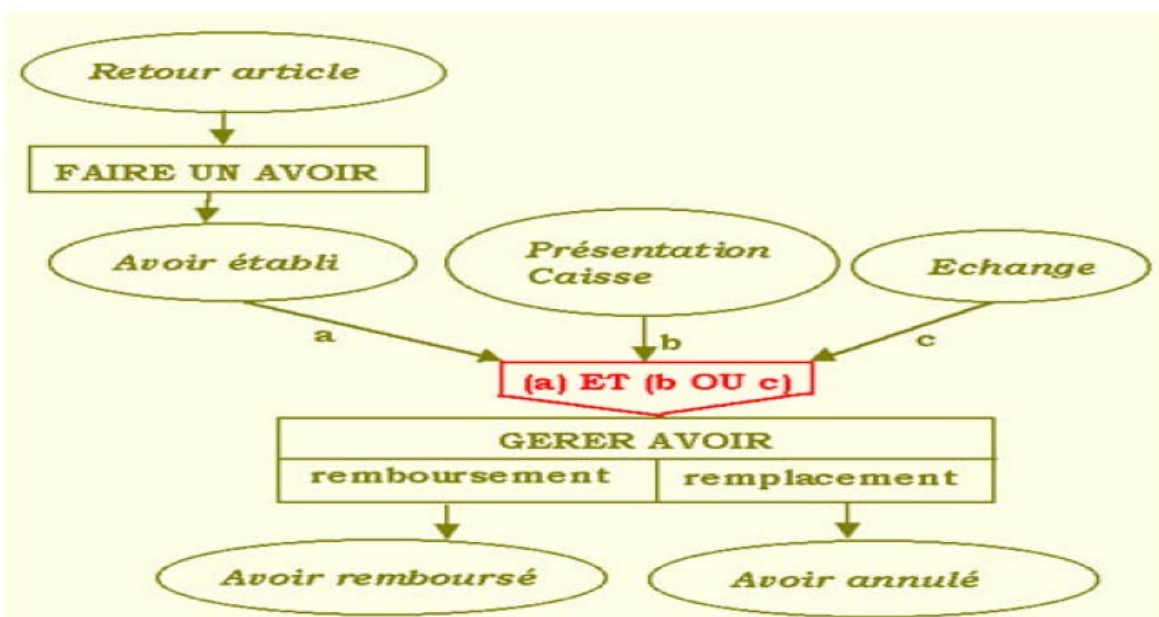
UNE occurrence de l'événement "avoir établi" ne peut être consommée par 2 opérations (rembourser et imputer).

Dit autrement : "avoir établi" ne sait pas vers quelle opération se diriger. On pourrait se dire qu'il est implicite que c'est le premier événement qui survient qui gagne : si Echange survient en premier, alors l'avoir est annulé.

Cependant, que se passe-t-il si les 2 événements externes se produisent simultanément ? Si on implémente tel quel ce schéma dans un automate, il est vraisemblable que le client pourra avoir le beurre et l'argent du beurre :

se présenter en caisse et faire l'échange pour un seul retour. Il est donc préférable d'explicitier (cf. ci-après).

Une autre solution consiste à indiquer explicitement que cet événement est consommable, c'est-à-dire que la première opération qui se déclenche annule l'événement pour l'autre opération.



II-Merise: Intégration des données dans le MCT

Sommaire

- [1 DEFINITION](#)
- [2 FORMALISME DU MCTA](#)
- [3 REPRESENTATION DES ACTIONS SUR LES DONNEES](#)
- [4 COMPLEMENTS](#)
 - [4.1 Condition de déclenchement](#)
 - [4.2 Action collective](#)
 - [4.3 Cohérence de la base de données](#)
 - [4.4 REGLES DE VALIDATION D'UN MCTA](#)
 - [4.5 Exemple de MCTA](#)

1-DEFINITION

Le MCTA est une représentation synthétique des traitements qui met en évidence les réactions du système aux événements et l'interaction avec les données.

Il vise à

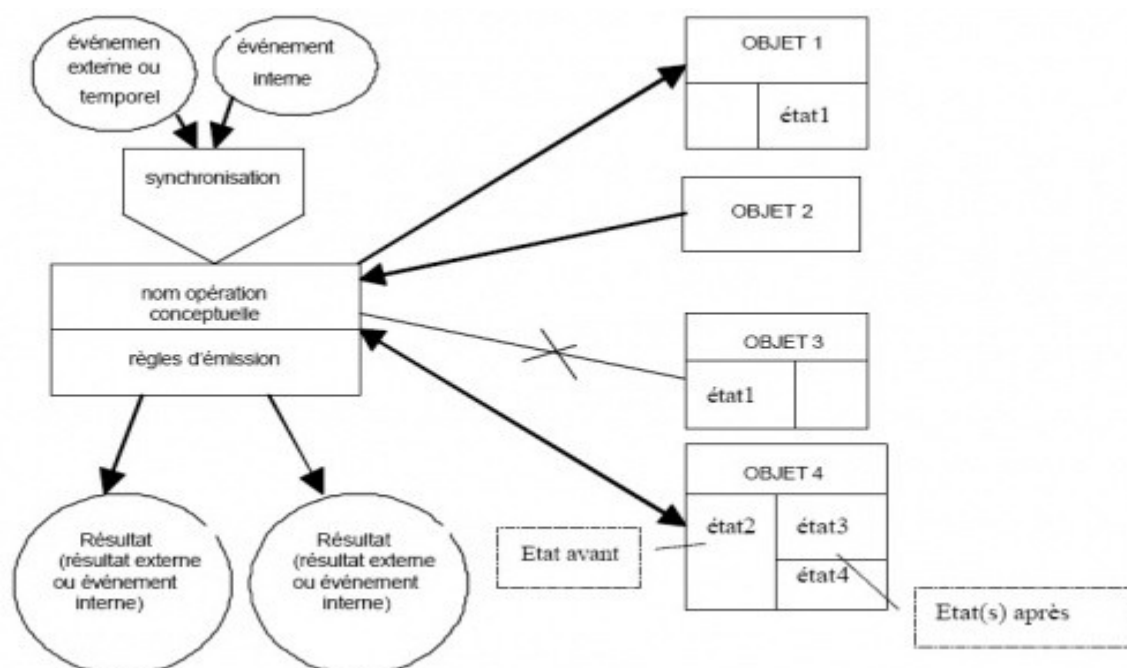
Décrire le fonctionnement du système indépendamment de l'organisation

Donner une vision des traitements et de la coordination des événements déclencheurs

Rapprocher les données et les traitements

2-FORMALISME DU MCTA

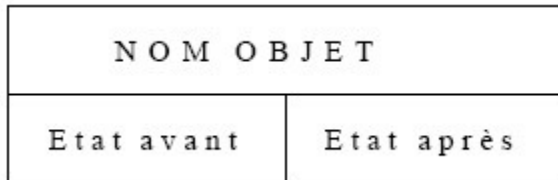
Pour mettre en évidence l'articulation entre les données et les traitements dans le [MCT](#), on ajoute à droite de chaque opération les entités ou associations qui sont consultées ou qui subissent un changement d'état (création, mise à jour, suppression, ...)



3-REPRESENTATION DES ACTIONS SUR LES DONNEES

Un objet correspond à une entité ou une association du MCD correspondant au champ d'étude du SI. Une action est une manipulation (création, consultation, modification ou suppression) d'un objet **du système d'information**. Elle est symbolisée par un trait entre l'opération et l'objet manipulé. Ce trait est non orienté s'il s'agit d'une suppression et il est orienté pour les autres types de manipulation, le sens de la flèche dépend du type de manipulation.

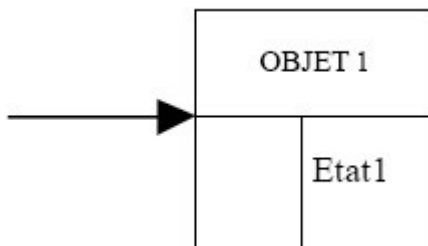
Un état d'objet est un stade transitoire par lequel passe un objet (entité ou association) au cours de sa « vie ».



Dans un MCTA, une opération est composée d'un ensemble d'actions sur les données.

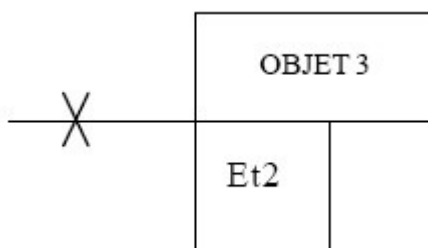
Les actions sur les données sont de 4 types :

Création d'une occurrence :



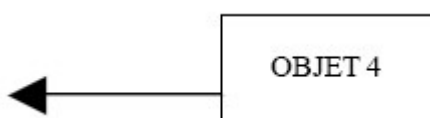
Création d'une occurrence de l'objet 1 dans l'état 1.

Suppression d'une occurrence :



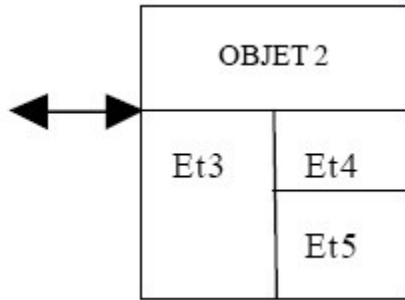
Suppression d'une occurrence de l'objet 3 se trouvant dans l'état 2.

Consultation d'une occurrence :



Consultation d'une occurrence de l'objet 4 dans l'état 6. L'état de l'objet n'est pas indiqué mais sera éventuellement testé avant réalisation de la (ou des) action (s) suivante (s).

Modification d'une occurrence :



Modification d'une occurrence de l'objet 2 si elle est dans l'état 3 : elle passe à l'état 4 ou à l'état 5.

L'action de modification inclut implicitement la ou les consultations préalables

On ne doit pas séparer les consultations qui précèdent les modifications des modifications elles-mêmes

Remarque générale L'ordre des actions d'une opération sera si possible présenté de haut en bas.

4-COMPLEMENTS

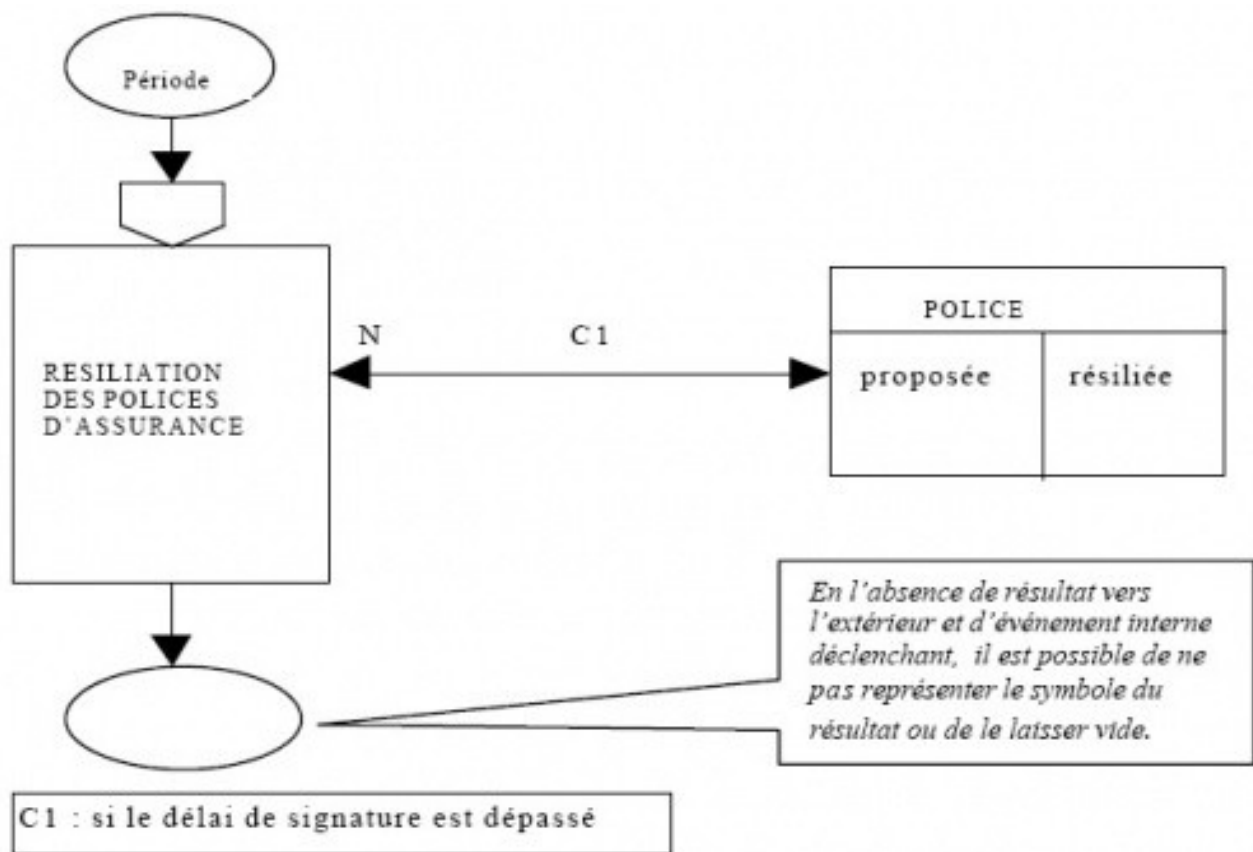
4.1-Condition de déclenchement

L'action élémentaire peut être accompagnée d'une condition de déclenchement (à indiquer au niveau du trait ou à côté de l'objet). Cette condition peut porter sur l'état de la structure des données à un moment précis et/ou sur le type d'événement constaté à l'entrée de l'opération.

4.2-Action collective

Elle permet d'indiquer que l'action intervient sur un ensemble d'occurrences de l'objet concerné (toutes s'il n'y a pas de condition de déclenchement associée limitant le nombre d'occurrences manipulées).

Exemple : en fin de mois, une compagnie d'assurances résilie les polices qu'elle a proposées dès que le délai de signature de celles-ci est dépassé.



C1 : si le délai de signature est dépassé

4.3-Cohérence de la base de données

Une BD a cohérent si elle ne comporte pas de contradictions (par rapport aux règles de gestion en particulier)

Une action peut ne pas laisser la structure de données dans un état cohérent.

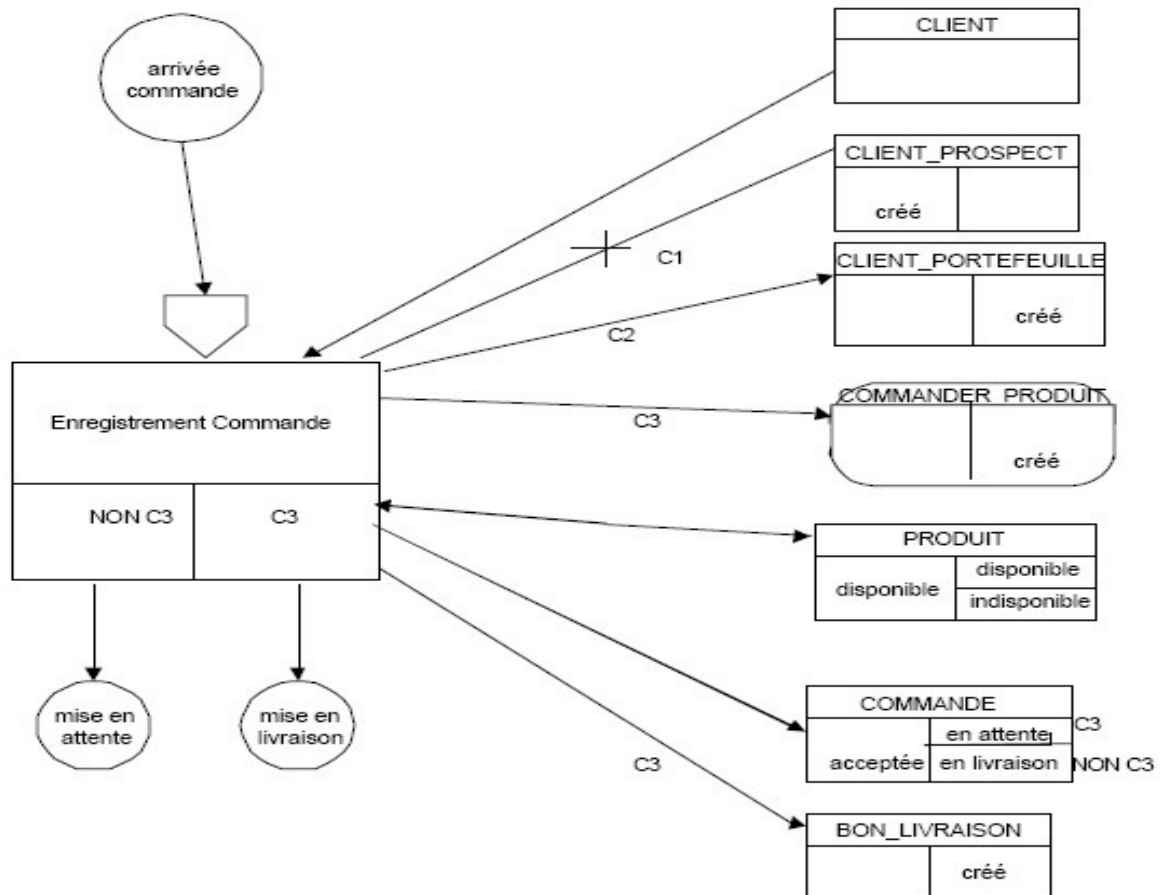
En revanche, la cohérence doit toujours être assurée au niveau d'une opération.

4.4-REGLES DE VALIDATION D'UN MCTA

Une opération conceptuelle ne doit pas être interrompue par l'attente d'un événement externe

Une opération conceptuelle prend la structure de données manipulées dans un état cohérent et la rend dans un état cohérent.

4.5-Exemple de MCTA



C1 : si le client était précédemment un client prospect (c'est-à-dire client potentiel).

C2 : s'il s'agit d'un nouveau client ou si le client était précédemment un client prospect, le client est mis en portefeuille (c'est-à-dire qu'il est réellement client)

C3 : si disponibilité de tous les produits (quantité en stock suffisante pour assurer la livraison)

Remarque : on considère ici qu'il n'y a pas de livraison partielle