

Chapitre 3

MERISE COMME METHODE D'ANALYSE ET DE CONCEPTION

I. MERISE comme méthode d'analyse : Etude de l'existant

1. Objectifs

- Prendre connaissance du domaine étudié en détail,
- Recenser l'ensemble (exhaustif) des objectifs poursuivis par le domaine,
- Collecter les documents traités,
- Et déduire après critiques, les objectifs de l'automatisation.

2. Fonctions

- Recueil de l'existant,
- Synthèse,
- Critiques de l'existant.

a. Recueil de l'existant

Il se résume en la collecte des informations comme

- Informations générales,
- Objectifs de l'organisation,
- Organigramme de l'organisation,
- Délimitation du champ d'étude,
- Calendrier de visite.

La collecte des informations s'effectue par :

- L'interview,
- L'observation,
- L'enquête.

La collecte des informations permet la production :

- ✓ Des objectifs et des contraintes,
- ✓ De la liste des règles de gestion, d'organisation et techniques,
- ✓ Du dictionnaire des données.

3. Démarche

a. L'interview

❖ L'interview de direction

L'interview de direction consiste à :

- Prendre connaissance du problème posé,
- Recenser les objectifs poursuivis par le demandeur,
- Localiser les postes de travail concernés,
- Délimiter le champ de l'étude,
- Décrire les interfaces avec d'autres projets.

❖ L'interview de poste de travail

Elle consiste en un entretien avec les responsables de chaque poste. Elle aura pour objectifs :

- De recenser et de décrire les tâches exécutées,
- D'observer la circulation des informations entre les différents postes de travail,
- D'inventorier les ressources utilisées,
- Chercher à connaître les supports des informations échangées,
- Prendre en compte les problèmes rencontrés,
- Apprendre le langage de travail des utilisateurs.

- Le déroulement de l'interview

- La préparation de l'interview

Il faut

- Bien choisir son interlocuteur,
- Prendre rendez-vous au moins 2 jours à l'avance,
- Préparer la liste des points à aborder,

- Lancement de l'interview

- Arriver toujours à l'heure,
- Avoir une tenue correcte,
- Être à deux si possible,
- Toujours se présenter,
- Rappeler le but de l'interview,
- Préciser la prise de note avec compte rendu transmis à l'utilisateur.

- Pendant l'interview

- Savoir poser les questions,
- Vérifier que votre interlocuteur ne sorte pas du cadre du sujet,
- Reformuler pour montrer que vous avez compris,
- Ne pas dépasser le temps imparti,
- Demander les documents concernant les thèmes abordés.

- La fin de l'interview

- Conclure,
- Remercier votre interlocuteur,
- Prendre un prochain rendez-vous.

b. L'observation

Nous distinguons deux modes d'observation :

❖ L'observation par événement

L'événement est suivi depuis son entrée dans le système d'information jusqu'à sa sortie en déterminant par station les tâches effectuées.

❖ L'observation par poste de travail

Chaque poste de travail est étudié. Il faut alors faire le tour de tous les postes concernés pour construire le diagramme de circulation des informations.

c. L'enquête

Elaborer des questionnaires sur les informations que l'on souhaiterait obtenir, les faire remplir par les utilisateurs puis procéder au dépouillage.

d. La synthèse

❖ La synthèse des traitements

- Inventorier les procédures des traitements,
- Dresser les diagrammes de circulation des flux de chaque procédure,
- Déduire le MCT (Modèle Conceptuel des Traitements) à partir du modèle organisationnel en faisant abstraction de l'organisation en place.

❖ La synthèse des données

- Lister les rubriques contenues dans les documents en prenant soin de supprimer les synonymes et les polysèmes,
- Constituer le dictionnaire des données,
- Déduire le modèle conceptuel des données.

4. Analyse

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 :

- Faire une critique objective du système existant,
- Proposer des objectifs afin d'améliorer la situation actuelle,
- Modéliser éventuellement ces objectifs afin d'améliorer la situation actuelle.

Une réflexion sur les évolutions possibles en tenant compte des nouveaux objectifs assignés débouche sur un nouveau système (ou l'ancien amélioré) appelé système futur dont l'analyse conduit à construire :

Niveau	Données	Traitements	
Conceptuel	<u>MCD</u> -Entité -Relation -Propriétés	<u>MCT</u> -Evènement -Opération -Résultat	Indépendant du système ; Quoi ?
Logique ou Organisationnel	<u>MLD</u> Représentation des données suivant un modèle	<u>MOT</u> -procédure -manuel / automatique -temps réel /différé	Choix du SGBD Qui, quand, où ?
Physique	<u>MPD</u> (modèle physique de données) -implantation des données dans l'environnement informatique	<u>MOpT</u> (mode opérationnel de traitements) Logiciel réalisé dans l'environnement informatique	Hautes connaissances du SGBD Comment ?

II. Le Modèle Conceptuel des Données (MCD)

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 (les dépendances fonctionnelles), pour aboutir au MCD.

1. Les concepts de base

a. Entité

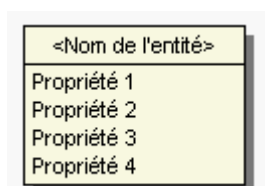
Une entité représente un objet du SI (acteur, document, concept, ...), ou plus exactement un ensemble d'objets ayant les mêmes caractéristiques.

Dans une entité, on met les informations nécessaires et suffisantes pour caractériser cette entité. Ces informations sont appelées **propriétés**.



Une entité est caractérisée par son nom et ses propriétés.

Représentation graphique:



Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.

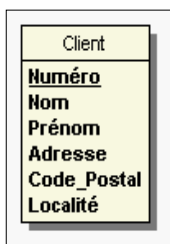
Exemple : Paul, mon vélo, l'Allemagne

Une entité type est la représentation commune que l'on adopte pour les entités possédant les mêmes caractéristiques.

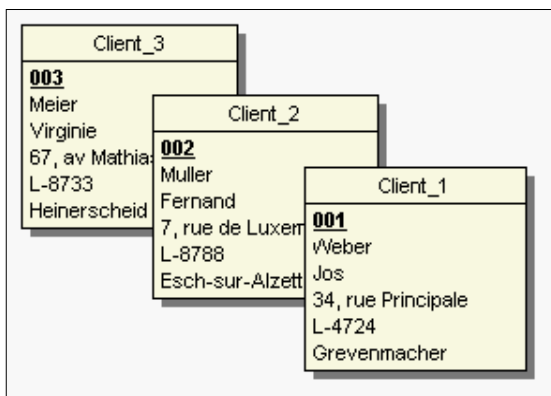
Exemple : Personne, voiture, Pays.

Une entité est donc une occurrence d'une entité type (ou instance).

Prenons par exemple une entité *Client*:



Voici quelques exemples de clients:



 Chacun de ces clients représente une occurrence de l'entité *Client*.

b. Propriété

La propriété est une information élémentaire, c'est-à-dire non déductible d'autres informations, qui présente un intérêt pour le domaine étudié.

Chaque valeur prise par une propriété est appelée occurrence.

Une propriété est dite simple ou encore atomique si chacune des valeurs qu'elle regroupe n'est pas décomposable.

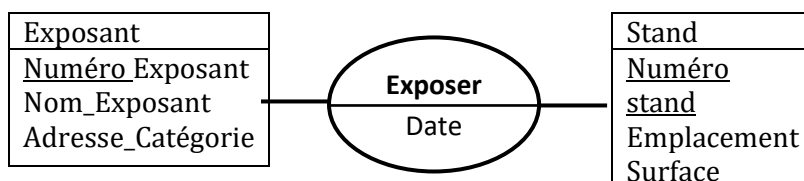
c. Relation ou association

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.

A Une relation décrit un lien entre deux ou plusieurs entités.
 Chaque relation possède un nom, généralement un verbe à l'infinitif.
 Bien qu'une relation n'ait pas d'identifiant propre, elle est implicitement identifiée par les identifiants des entités auxquelles elle est liée.

A Nous distinguons deux types de relations:

- Les relations binaires, qui sont liées à 2 entités;
- Les relations ternaires, qui sont liées à 3 entités.



d. Cardinalité

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

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

Un couple de cardinalités placé entre une entité E et une association A, représente le nombre minimal et maximal d'occurrences de l'association A qui peuvent être « ancrées » à une occurrence de l'association E. Le tableau ci-après récapitule les valeurs que peut prendre ce couple.

E 0,1 A	Pour chaque occurrence de E, le modèle admet : - Soit l'absence de lien - Soit la présence d'un seul lien
E 1,1 A	Pour chaque occurrence de E, le modèle admet la présence d'un et d'un seul lien
E 1,n A	Pour chaque occurrence de E, le modèle admet la présence d'un ou de plusieurs liens
E 0,n A	Pour chaque occurrence de E le modèle admet - Soit l'absence de lien - Soit la présence de plusieurs

Remarque : dans certaines situations, la lettre n peut être remplacée par une valeur discrète si celle-ci est connue.

 De façon générale, on peut dire:

La cardinalité minimale exprime le nombre minimum de fois qu'une occurrence d'une entité participe à une relation. Cette cardinalité est généralement 0 ou 1.

- Cardinalité minimale = 0 : Certaines occurrences de l'entité ne participent pas à la relation
- Cardinalité minimale = 1 : Chaque occurrence de l'entité participe au moins une fois à la relation

La cardinalité maximale exprime le nombre maximum de fois qu'une occurrence d'une entité participe à une relation. Cette cardinalité vaut souvent 1 ou n, avec n indiquant une valeur >1 mais pas connue à priori.

- Cardinalité maximale = 1 : Chaque occurrence de l'entité participe au maximum une seule fois à la relation.
- Cardinalité maximale = n : Chaque occurrence de l'entité peut participer plusieurs fois à la relation.

Exemples : (1,1 – 1,n) (1,1 – 0,n)

 On dit qu'une entité est indépendante par rapport à une relation lorsque sa cardinalité minimale vaut 0, et dépendante par rapport à une relation lorsque sa cardinalité minimale vaut 1.

 Une relation ne peut pas être liée uniquement à des entités dépendantes ayant en plus une cardinalité maximale de 1 !!!

e. Notion d'identifiant

 Afin de pouvoir distinguer les différentes occurrences d'une même entité, l'entité doit être dotée d'un identifiant. L'identifiant est composé d'une ou de plusieurs propriétés de l'entité. Chaque occurrence d'une entité doit avoir une valeur différente pour l'identifiant. Le choix d'un identifiant correcte est très important pour la modélisation:

Comme choix pour l'identifiant d'une entité nous distinguons généralement 3 possibilités :

1. Une propriété **naturelle**

Exemple: Le nom d'un pays pour une entité *Pays*

2. Une propriété **artificielle** qui est inventée par le créateur du MCD

Exemple: Le numéro d'un client pour une entité *Client*

3. Une propriété **composée** d'autres propriétés naturelles

Exemple: Le nom et la localité pour une entité *Entreprise*

 **Représentation graphique de l'identifiant d'une entité:**
La ou les propriétés qui constituent l'identifiant d'une entité sont soulignées.

L'identifiant est une propriété qui ne peut pas changer au cours du temps pour une occurrence.

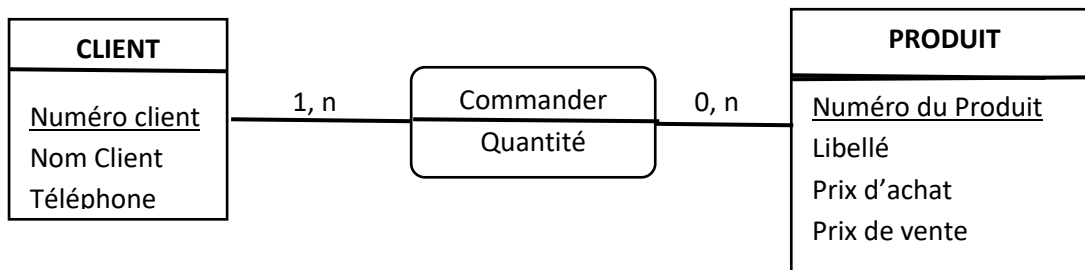
f. Dimension d'une association

Le nombre d'occurrences d'entités concernées par une occurrence de l'association est appelé sa dimension.

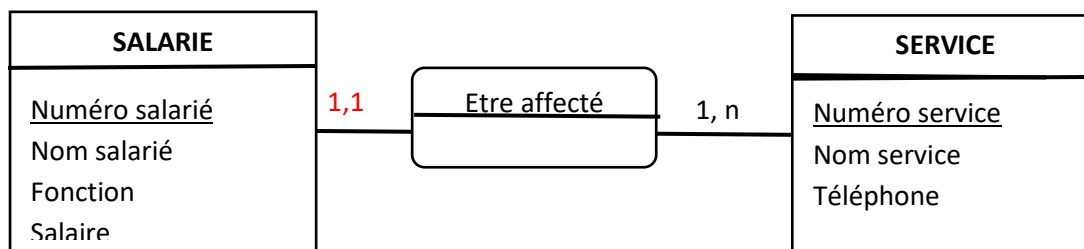
EXEMPLES DE MCD :

Association binaires

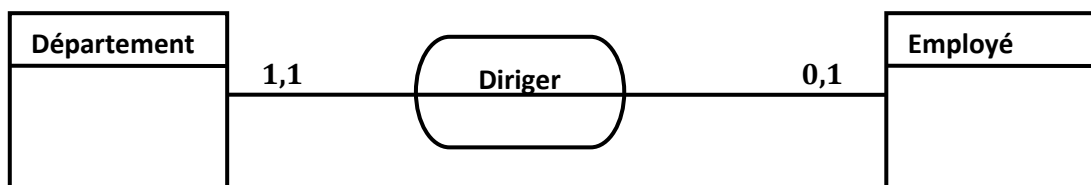
- Lien maillé



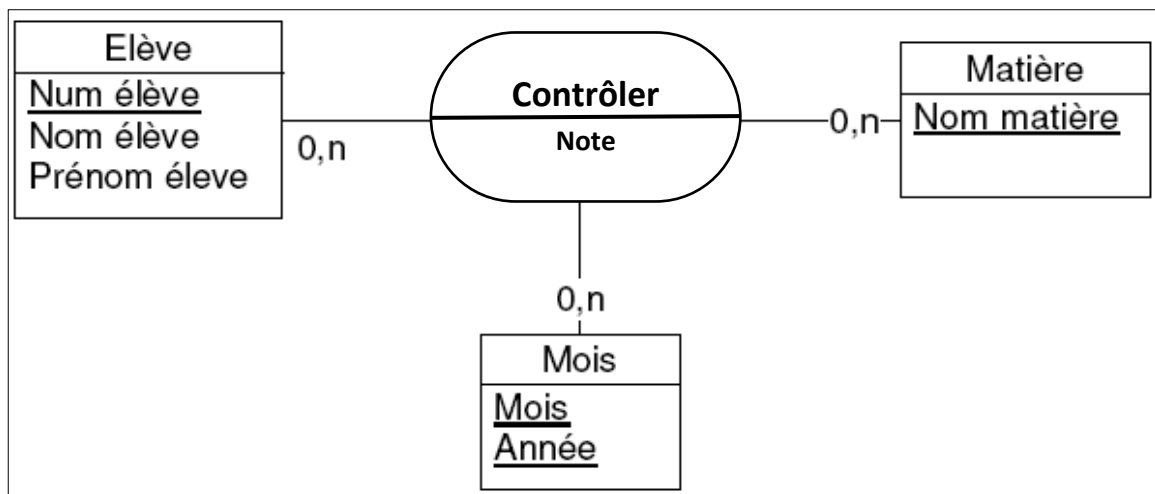
- Lien hiérarchique



- Lien fonctionnel



Association ternaire ou n-aire



Une relation peut généralement être porteuse de propriétés.



Attention: Cette propriété peut même devenir une partie de l'identifiant. Dans ce cas, elle doit être soulignée.



Attention: Une relation à cardinalité (1,1) n'est jamais porteuse de propriétés. Dans ce cas, les propriétés migrent dans l'entité portant cette cardinalité (1,1).

g. Les règles de gestion

Elles précisent les contraintes qui doivent être respectées par le modèle.

Elles expriment les contraintes d'intégrité du modèle.

On distingue :

- Des règles de calcul
- Des règles d'action
- Des règles administratives
- Du délai d'attente

De ces règles, sont tirées les cardinalités.

Il existe une règle générale de conception qui dit:



Aucune propriété qui peut être calculée à partir d'autres propriétés existantes, ne devra être stockée dans le MCD.

Ex : le prix d'un article (qui est égal à son PU * Qté)

Exemples de règles de gestion :

RG1 : Un client peut avoir plusieurs adresses donc nous avons une cardinalité de 1, n.

RG2 : Un carnet d'adresses est lié à un seul client donc nous avons 1,1.

RG3 : Un carnet d'adresses peut contenir plusieurs adresses professionnelles donc 1, n

RG4 : Une adresse professionnelle est liée à un seul carnet d'adresse 1,1.

RG5 : Un client peut passer une ou plusieurs commandes donc nous avons 1, n.

RG6 : Une commande ne concerne qu'un seul client donc nous avons 1,1.

RG7 : Une facture ne concerne qu'une seule commande donc 1,1.

RG8 : Il doit y avoir une commande pour créer une facture et il peut y avoir plusieurs commandes a facturé donc 1, n.

RG9 : Une commande est reliée à un ou plusieurs historiques de statut donc 1, n.

RG10 : L'historique de statut peut être lié à une seule commande donc 0,1.

Exercice d'application : Etude des cardinalités - Mariages

Traduire à l'aide du modèle entité-association les différentes situations maritales évoquées ci-dessous :

- **Option classique** : un homme est marié avec au plus une femme et une femme avec au plus un homme.
- **Option nouvelle tendance** : une personne est mariée avec au plus une autre personne.
- **Polyandrie** : tout homme est marié avec une et une seule femme et une femme avec plusieurs hommes.
- **Option hippie** : toute personne est mariée avec une autre personne au moins.
- **Option harem** : un homme est marié avec aucune, une ou plusieurs femmes, mais une femme avec au plus un homme.

h. Le dictionnaire des données

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

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

- ✧ Le **code mnémonique** : il s'agit d'un libellé désignant une donnée (par exemple «*titre_liv*» pour le titre d'un livre)
- ✧ La **désignation** : il s'agit d'une mention décrivant ce à quoi la donnée correspond (par exemple «*titre du livre*»)
- ✧ Le **type de donnée** :
 - ✧ **A** ou **Alphabétique** : lorsque la donnée est uniquement composée de caractères alphabétiques (de 'A' à 'Z' et de 'a' à 'z')
 - ✧ **N** ou **Numérique** : lorsque la donnée est composée uniquement de nombres (entiers ou réels)
 - ✧ **AN** ou **Alphanumérique** : lorsque la donnée peut être composée à la fois de caractères alphabétiques et numériques
 - ✧ **Date** : lorsque la donnée est une date (au format JJ-MM-AAAA)
 - ✧ **Booléen** : Vrai ou Faux
- ✧ La **taille** : elle s'exprime en nombre de caractères ou de chiffres. Dans le cas d'une date au format AAAA-JJ-MM, on compte également le nombre de caractères, soit 10 caractères.

Pour ce qui est du type booléen, nul besoin de préciser la taille (ceci dépend de l'implémentation du SGBDR).

- ✧ Et parfois des **remarques** ou **observations** complémentaires (par exemple si une donnée est strictement supérieure à 0, etc)

N°	Code	Signification	Nature	Type	Longueur	Observation
1	CodeCli	Code du client	E	AN	8	Identifiant
2	Nom	Nom du client	E	A	25	Obligatoire
3	Prenom	Prénoms du client	E	A	30	
4	Adress	Adresse du client	Co	AN	150	
5	NumCpte	Numéro de compte	E	N	1	Identifiant
6	Solde	Solde du compte	E	N	14	Obligatoire
7	Num_op	Numéro de l'opération	E	N	4	Identifiant
8	Type	Etat actuel de l'opération	E	A	6	Dépôt/Retrait
9	Date	Date de l'opération	E	D	10	JJ/MM/AAAA
10	Montant	Montant de l'opération	E	N	14	Obligatoire

A = Alphabétique
 AN = Alpha – numérique
 D = Date
 E =Elémentaire
 N = Numérique
 Co = Composé ou Concaténé
 Ca = Calculé
 Pa = Paramètre

Remarque :

- ✧ Les données qui figurent dans le MCD (et donc dans le dictionnaire des données) doivent être, dans la plupart des cas, **élémentaires** :
 - ✧ Elles ne doivent pas être **calculées** : les données calculées doivent être obtenues, par le calcul, à partir de données élémentaires qui, elles, sont conservées en base. Cependant, il existe quelques cas où il s'avère pertinent de conserver, pour des raisons d'optimisation, une donnée calculée, le montant d'une commande par exemple. On ne conservera cependant pas les données calculées intermédiaires sauf en cas d'obligation légale (c'est le cas pour un montant HT par exemple, où les

composantes peuvent d'ailleurs avoir un prix variable dans le temps). En effet, cela évite de refaire les calculs plusieurs fois pour un résultat qui restera fixe.

- ⤴ Elles ne doivent pas être **composées** : les données composées doivent être obtenues par la concaténation de données élémentaires conservées en base. Par exemple une adresse est obtenue à partir d'une rue, d'une ville et d'un code postal : ce sont ces trois dernières données qui sont conservées et donc qui figureront dans le MCD (et dans le dictionnaire des données).
- ⤴ Lorsque l'on n'effectue jamais de calcul sur une donnée numérique, celle-ci doit être de type AN (c'est le cas par exemple pour un numéro de téléphone).

N.B :

Données paramètres est une donnée qui ne prend qu'une unique valeur. Exemple : L'entreprise s'appelle PVF. La donnée nom de l'entreprise est une donnée qui ne prend qu'une seule valeur : PVF. Il s'agit donc d'une donnée paramétrable.

i. La dépendance fonctionnelle

■ Définition.

Une donnée 2 est en dépendance fonctionnelle d'une donnée 1 quand la connaissance d'une valeur de la donnée 1 permet de déterminer la connaissance d'au maximum une et une seule valeur de la donnée 2. La donnée 1 est appelée la source et la donnée 2 est appelée le but.

Question.

La question fondamentale à se poser est : "Connaissant une valeur de la source, peut-on connaître une valeur unique du but ?".

Quand la réponse est affirmative, on a l'habitude de représenter cette dépendance comme suit :

SOURCE =====> BUT

Dépendance fonctionnelle à partie gauche (source) composée.

Il peut arriver que ce soit la combinaison de plusieurs attributs (en source) qui permettent de connaître une valeur unique du but. Exemple : un numéro de facture + un code produit nous donne la quantité facturée.

(numéro de facture, code produit) =====> quantité facturée

On parlera de dépendance fonctionnelle à partie gauche composée (DFPGC).

■ Dépendance non fonctionnelle.

Deux rubriques sont en dépendance non fonctionnelle si la connaissance d'une valeur de la première

- ⤴ ne permet de connaître aucune des valeurs de la seconde (pas de rapport entre les deux)
- ⤴ détermine la connaissance de plusieurs valeurs de la seconde

exemples :

- ⤴ la connaissance d'un numéro de facture permet de connaître plusieurs références d'articles
- ⤴ la connaissance d'une date de naissance ne permet pas de connaître une adresse

■ Dépendance fonctionnelle élémentaire.

Une dépendance fonctionnelle donnée 1 =====> donnée 2 est élémentaire s'il n'existe pas une donnée 3, sous-ensemble de donnée1, qui assure elle-même une dépendance fonctionnelle donnée 3 =====> donnée 2.

Exemples :

Référence article =====> nom article
(Numéro facture, référence article) =====> quantité facturée
(Numéro facture, référence article) =====> nom article

La première est élémentaire car la référence article permet directement de connaître son nom.
La deuxième est élémentaire car le numéro de facture seul ou la référence article seule ne permet pas de connaître la quantité facturée.

La troisième n'est pas élémentaire car on peut se passer du numéro de facture pour trouver le nom de l'article (la partie référence article du source suffit pour retrouver le nom d'article).

■ Dépendance fonctionnelle directe.

Une dépendance fonctionnelle donnée 1 =====> donnée 2 est directe s'il n'existe pas une donnée 3 (ou une collection de rubriques) qui engendrerait une dépendance fonctionnelle transitive de telle sorte que l'on pourrait écrire :

Donnée 1 =====> donnée 3 =====> donnée 2

Exemple :

Numéro facture =====> numéro représentant

Numéro représentant =====> nom représentant

Numéro facture =====> nom représentant

On ne dessinera pas cette dernière car sa représentation est superflue.

■ Dépendance fonctionnelle élémentaire et directe. (DFED).

Les dépendances fonctionnelles que nous allons nous efforcer de trouver dans le système d'information sont celles qui sont à la fois élémentaires et directes. Cette notion de DFED que nous noterons dorénavant DF (en oubliant tout ce qui précédait) est sans doute la partie la plus importante du cours car elle est le fruit de la réflexion de l'analyste alors que pour les étapes suivantes, on pourra utiliser un certain nombre de règles de passage. Il convient donc de bien maîtriser ces notions avant d'aborder la suite de la démarche.

4.2. Démarche de recherche des DF.

A partir du dictionnaire des données élémentaires, il faudra

- ⤴ rechercher les DF à deux rubriques élémentaires et directes
- ⤴ rechercher les DF à partie gauche composée.

■ Recherche des DF à deux rubriques.

On commence par rechercher les DF à deux rubriques en commençant par les plus évidentes du genre

Numéro de client =====> nom de client

Si on a une DF du type Numéro client =====> adresse, il faudra la décomposer en

Numéro de client =====> rue

Numéro de client =====> code postal

Numéro de client =====> localité

.....

Parfois, pour gagner du temps et de la place, on conserve la première version.

Parfois, la DF est symétrique. Numéro état civil =====> numéro de sécurité sociale.

Dans ce cas, on supprime une des deux pour garder la plus fréquemment utilisée.

Recherche des DF à partie gauche composée.

Voici la marche à suivre pour les trouver (en respectant l'ordre suivant) :

- ⤴ considérer l'ensemble des rubriques sources de DF simples (S)
- ⤴ considérer l'ensemble des rubriques non encore concernées par au moins une DF simple (ni source ni but d'au moins une DF simple) (N)
- ⤴ chercher une DF à partie gauche composée dont la source est composée de plusieurs rubriques de l'ensemble S ou de l'ensemble N, le but étant une rubrique de l'ensemble N.
- ⤴ quand toutes les rubriques de l'ensemble N sont concernées par au moins une DF (aussi bien dans le but que dans la source) ou bien quand on ne peut vraiment pas trouver de DF intégrant toutes les rubriques de N : rechercher des DFPGC dont le but serait une rubrique de S et la source des rubriques de S ou de N. Il ne doit beaucoup de rubriques non concernées par des DF.

■ Précautions relatives aux DFPGC.

Quand on traite des DFPGC, il faut toujours se poser les deux questions suivantes, si on a une DFPGC du type A, B, C =====> D

- * n'y aurait-il pas des DF du style D =====> A ou D =====> B ?
exemple : (date commande, n° client) =====> n° commande. On préférera pourtant n° commande =====> n° client et n° commande =====> date commande
- * n'y aurait-il pas, entre A, B, C et D une ou des DFPGC de moins de rubriques que celle citée, du type D,A =====> B ? Dans ce cas, il faut la privilégier.

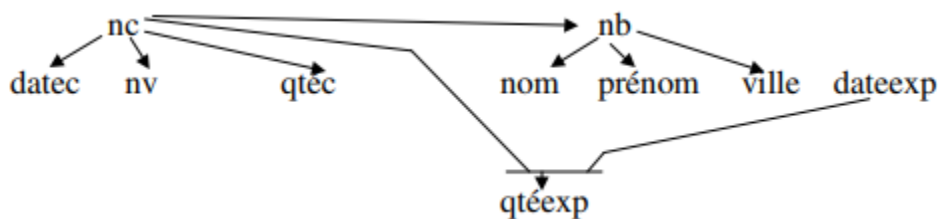
Exemple : (jour, heure, classe, salle) =====> professeur où jour donne lundi , mardi, ...; heure nous donne 1ère heure, 2ème heure, ... ; salle nous donne son n° et classe 1ère info, ...
On préférera : (jour, heure, prof) =====> classe, etc

■ Les modes de représentation des DF.

Il existe deux méthodes pour représenter les dépendances fonctionnelles ; vous déterminerez celle qui vous convient le mieux en pensant à la manière qui est la plus parlante pour vous.

• Le graphe des DF

Il s'agit de représenter les DF par des flèches.



Dans le cas des DFPGC, on utilise un nœud représenté par un cercle :

Exemples

Livraison

Nous souhaitons gérer une chaîne d'approvisionnement. Nous tenons une liste de produits et une liste de fournisseurs. Certains produits peuvent être proposés par plusieurs fournisseurs, à des prix qui peuvent être différents. Nous garderons l'historique des livraisons en mémoire, avec pour chacune d'entre elle, le fournisseur qui l'a effectué, la date de livraison ainsi que le détail des produits livrés.

Secrétariat Pédagogique

Nous souhaitons gérer un secrétariat pédagogique. Nous recensons les étudiants en utilisant leurs noms, prénoms et adresse. Des formations sont organisées en modules, qui eux-mêmes sont répartis sur des semestres. Un étudiant ne peut être inscrit que dans une formation à la fois, et un module est affecté à un seul semestre dans une formation. On tiendra compte du fait qu'un étudiant peut redoubler ou suspendre sa formation en prenant des congés.

CMS

Nous voulons gérer un CMS (Content Management System). Un CMS est un logiciel permettant de gérer le contenu d'un ou plusieurs sites web. Chaque site porte un nom, est caractérisé par une URL, et est découpé en catégories, imbriquables les unes dans les autres. Des utilisateurs sont répertoriés dans le CMS, et chacun peut avoir le droit (ou non) d'écrire dans un site web. Chaque utilisateur doit avoir la possibilité de publier des articles dans une catégorie d'un site web (pourvu qu'il dispose de droits suffisants sur le site). Un article, en plus de son titre et de son texte d'introduction, est constitué de chapitres portant chacun un nom et contenant un texte. Il

va de soi qu'on doit avoir la possibilité pour chaque site de conserver l'historique de quel article a été publié par quel utilisateur.

Covoiturage

Nous allons étudier la modélisation d'une plate-forme de co-voiturage.

Des utilisateurs peuvent être :

- des conducteurs
- des passagers
- des modérateurs de la société de co-voiturage.
- ou encore prendre plusieurs rôles à la fois...

On mémorisera leur nom, prénom, mail, téléphone, coordonnées bancaires.

Les utilisateurs peuvent en tant que conducteurs s'inscrire et donner des informations sur leurs véhicules (marque, modèle, immatriculation).

Les conducteurs peuvent créer des trajets (point et date de départ, d'arrivée, véhicule, nombre de places disponibles, étapes).

Les points de départ, d'arrivée, et les étapes, sont :

- soit des points covoiturage prédéfinis,
- soit des adresses utilisant des villes prédéfinies.

Le conducteur doit indiquer pour chaque étape :

- l'heure à laquelle il doit passer
- le coût de prise en charge d'un passager

Les utilisateurs, en tant que passagers peuvent :

- rechercher des trajets,
- contacter le conducteur avec une messagerie interne,
- faire une réservation sur un trajet ou une partie du trajet.
- annuler leur réservation
- valider leur trajet, ce qui déclenche le paiement.

Une fois un trajet validé, les passagers peuvent noter le conducteur et le conducteur peut noter ses passagers. Une note est constituée :

- d'un nombre d'étoiles (1 à 5)
- d'un commentaire

Les commentaires doivent être validés par un modérateur. Un usager ne peut pas voir un commentaire le concernant tant qu'il n'a pas lui-même rédigé un commentaire. Si au bout de 15 jours il n'a pas répondu, alors le commentaire est publié et l'utilisateur concerné ne peut pas répondre.

La note moyenne d'un utilisateur est toujours publiée sur son profil.

Bibliothèque

Nous souhaitons gérer une bibliothèque simple. La base de données devra recenser une liste d'ouvrages, avec pour chacun le titre et l'auteur (éventuellement plusieurs). Vous tiendrez compte du fait que des ouvrages peuvent exister en plusieurs exemplaires, certains pouvant ne pas être disponibles pour le prêt (consultables uniquement sur place, ou détruits). Une liste d'adhérents devra être tenue à jour, les adhésions devant se renouveler une fois par an. Il devra être possible de vérifier qu'une adhésion est à jour (c'est-à-dire qu'elle a été renouvelée il y a moins d'un an), mais il ne sera pas nécessaire de connaître l'historique des renouvellements. Les adhérents peuvent emprunter jusqu'à 5 livres simultanément et chaque livre emprunté doit être retourné dans les deux semaines. On devra conserver un historique permettant de savoir quel abonné a emprunté quels livres, les dates d'emprunts et de retours.

- **la matrice des DF**

La matrice des dépendances fonctionnelles est le tableau qui nous permet de décrire les dépendances fonctionnelles et les contraintes liés à une base de données. C'est un tableau à deux dimensions où les colonnes contiennent les sources de dépendances fonctionnelles et les lignes contiennent toutes les propriétés du dictionnaire des données (les buts). Quand une propriété en ligne est but de dépendance fonctionnelle élémentaire, on indique cette dépendance par le chiffre (1) à l'intersection de la ligne et de la colonne concernée. Les dépendances réflexives sont représentées par le symbole (*) à l'intersection de la ligne et de la colonne. Les contraintes d'intégrité sont représentées par le symbole (©) toujours à l'intersection de la ligne et de la colonne concernée.

La matrice des dépendances fonctionnelles

La matrice des dépendances fonctionnelles est une représentation graphique de l'ensemble des dépendances fonctionnelles entre données. Elle permet de faire une étude exhaustive des relations entre chaque donnée.

Elle se présente sous forme d'une table carrée ayant pour entrées l'ensemble des données du dictionnaire.

- Les entêtes de lignes sont les données sources
- Les entêtes de colonnes sont les données buts

Comme de bien entendu une donnée est en dépendance fonctionnelle avec elle-même, et réciproquement. On traduira cette règle par une série '*' sur la diagonale.

On placera pour chaque dépendance fonctionnelle déterminée un '1' dans la case correspondante, pour cela il va falloir parcourir pour chaque données sources l'ensemble des données buts. Le parcours va se faire colonnes par colonnes, et pour chaque colonne ligne par ligne.

On se pose à chaque étape la question suivante : la donnée source est-elle en dépendance fonctionnelle avec la donnée but.

- si OUI on place un '1'
- si NON on passe à la ligne suivante.

Définition d'identifiant

Les données sources de dépendances fonctionnelles sont appelées **identifiants** des futures entités.

Graphiquement ce sont les colonnes qui comprennent des '1'

Exemple

Code Epreuve

Numéro Candidat

Les dépendances fonctionnelles composées

La matrice des dépendances fonctionnelles ne doit pas faire apparaître de lignes sans '1', si c'est le cas cela veut dire que la donnée n'est but d'aucune dépendance fonctionnelle.

- soit la donnée est source d'une dépendance fonctionnelle (*donc un identifiant*),
- soit la donnée n'a rien à faire dans notre système d'information, il faut donc l'éliminer,
- soit la donnée est en dépendance fonctionnelle avec plusieurs sources (*on est en présence d'une dépendance fonctionnelle composées*).

Exemple

Une note dépend d'un candidat et d'une épreuve

A Numéro candidat, Code Epreuve « **est associé** » Note

En français : à un numéro de candidat et un code épreuve est associée une et une seule note.

Suppression des transitivités entre données

L'objectif principal de la conception d'une base de données est la non redondance des informations. Cela veut dire qu'une donnée 'non Identifiant' ne doit être le but que d'une seule dépendance fonctionnelle.

Graphiquement cela se traduit par un seul '1' par ligne, il va falloir effectuer des choix entre les dépendances fonctionnelles dans le cas contraire.

Exemple

Code Epreuve « **est associé** » libellé examen

Code Epreuve « **est associé** » Code Examen

Code Examen « **est associé** » libellé examen

En conclusion

A Code Epreuve « **est associé** » libellé examen

		S O U R C E S															
Buts		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	date naissance	*															
2	coefficient épreuve		*														
3	date épreuve			*													
4	nom enseignant				*												
5	prénom enseignant					*											
6	date rédaction						*										
7	note élève							*									
8	libellé examen								*								
9	Code épreuve									*							
10	numéro candidat										*						
11	nom candidat											*					
12	prénom candidat												*				
13	code examen													*			
14	Appréciation														*		
15	libellé épreuve															*	
16	numéro enseignant																*

Dépendances fonctionnelles entre identifiants

Sur une ligne d'une donnée 'identifiant' la présence de plusieurs '1' indique soit :

- une transitivité de dépendance fonctionnelle (*qu'il faudra donc éliminer*)
ou
- la mise en évidence d'une association hiérarchique (*appelée souvent Contrainte d'Intégrité Fonctionnelle C.I.F et pas toujours de façon judicieuse*)

Exemple d'une association hiérarchique : Code Epreuve « **est associé** » Code Examen

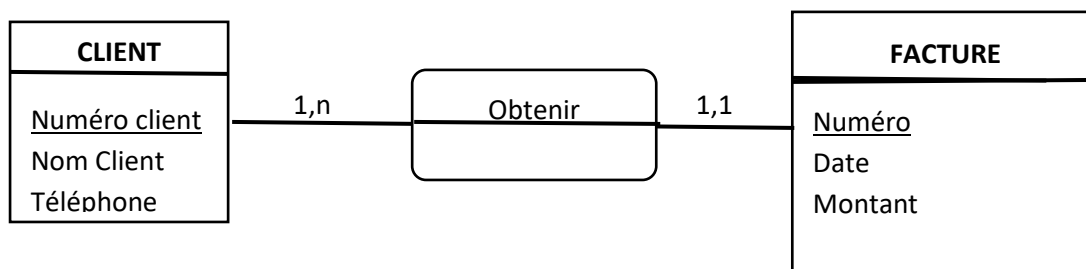
Exemple de matrice des D.F. finale

		S O U R C E S					
Buts		9	10	13	16	9+10	10+13
1	date naissance		1				
2	coefficient épreuve	1					
3	date épreuve	1					
4	nom enseignant				1		
5	prénom enseignant				1		
6	date rédaction	1					

7	note élève					1	
8	libellé examen			1			
11	nom candidat		1				
12	prénom candidat		1				
13	code examen	1		*			
14	Appréciation						1
15	libellé épreuve	1					

j. Les contraintes d'intégrité fonctionnelle (CIF)

 Quand on détermine entre une relation et une entité une cardinalité qui présente les valeurs 0,1 ou 1,1, alors cette relation est particulière et on dit qu'elle représente une Contrainte d'Intégrité Fonctionnelle (CIF).



La relation *Obtenir* représente une CIF.

Une CIF indique que l'une des entités est totalement déterminée par la connaissance de l'autre. Dans notre exemple on peut dire que connaissant une facture bien précise, on connaît avec certitude le client correspondant.

Comment est-ce qu'on représente une CIF dans un MCD ?

Une CIF est représentée par un **trait plein** (comme toute autre relation) sur la patte opposée à celle ayant une cardinalité 0,1 ou 1,1. L'entité qui est attachée à cette patte est appelée entité **cible** de la CIF, tandis que l'autre entité constitue l'entité **émettrice** de la CIF.

Application 1 :

Une banque désire posséder un SGBD pour suivre ses clients. Elle désire ainsi stocker les coordonnées de chaque client (nom, prénom adresse), et les comptes dont elle dispose ainsi que leur solde (sachant par ailleurs que certains comptes ont plusieurs bénéficiaires). On stockera également les opérations relatives à ces comptes (retrait et dépôt, avec leur date et le montant). Vous disposez du dictionnaire des données suivant en annexe.

Dictionnaire des données

N°	Code	Signification	Nature	Type	Longueur	Observations
1	CodeCli	Code du client	E	AN	8	Identifiant
2	Nom	Nom du client	E	A	25	Obligatoire
3	Prenom	Prénoms du client	E	A	30	
4	Adress	Adresse du client	Co	AN	150	
5	NumCpte	Numéro de compte	E	N	1	Identifiant
6	Solde	Solde du compte	E	N	14	Obligatoire
7	Num_op	Numéro de l'opération	E	N	4	Identifiant
8	Type	Etat actuel de l'opération	E	A	6	Dépôt/Retrait
9	Date	Date de l'opération	E	D	10	JJ/MM/AAAA

A = Alphanumérique
 AN = Alpha – numérique
 D = Date
 E =Elémentaire
 N = Numérique
 Co = Composé ou Concaténé
 Ca = Calculé
 Pa = Paramètre

10	Montant	Montant de l'opération	E	N	14	Obligatoire
----	---------	------------------------	---	---	----	-------------

TAF : Présenter le MCD.

Application 2 : Gestion des commandes

Votre société veut créer une petite base de données de gestion des commandes pour contrôler ses stocks et ainsi pouvoir alimenter ses clients de manière ininterrompue, et ce qu'elle que soit la demande.

Chaque client est une société avec une raison sociale et un numéro de client interne à votre entreprise. Pour des raisons pratiques, vous possédez le nom d'un contact et sa fonction au sein de l'entreprise. Votre société, pour des raisons d'organisation, ne veut avoir à faire qu'à un et un seul contact. De plus, vous connaissez évidemment le numéro de téléphone, le fax et l'adresse de votre client.

Vous possédez une gamme de produits en catalogue, parmi lesquels les clients peuvent commander. Un produit possède une référence, un nom et un prix unitaire. Il entre dans une

catégorie codée. Chaque catégorie a en plus d'un code, un nom et une description. Elle référence logiquement plusieurs produits.

Vous recevez ainsi des commandes d'un client, chacune d'entre elles pouvant contenir un ou plusieurs produits. Cette commande est numérotée et datée. Le ou Les produits qui la composent sont vendus à un prix de vente qui peut différer du prix unitaire, puisque votre société décide de sa marge selon ses clients.

TRAVAIL A FAIRE :

A partir de ce texte, construire :

- Un dictionnaire des données.
- Une table (Matrice) et un graphe des dépendances fonctionnelles.
- Un Modèle conceptuel des données (MCD).
- Un modèle logique des données (MLD).

Application 3 : Vente sur commande.

Le but est de construire un système permettant de gérer un magasin de vente de produit à des particuliers.

Les produits du magasin possèdent une référence (un code), un libelle et un prix unitaire.

Les clients ont une identité (nom, prénom, adresse).

Les clients passent des commandes de produits. On mémorise la date de la commande.

Pour chaque commande, le client précise une adresse de livraison.

La commande concerne un certain nombre de produits, en une quantité spécifiée pour chaque produit. Le prix facturé pour chaque article lors d'une commande diffère d'un client à un autre.

Le montant total de la commande est mémorisé.

Une commande passée fait objet de livraison(s); le numéro du bon de livraison et la date de livraison sont enregistrés. On désire connaître la quantité livrée pour chaque produit commandé car une livraison peut être partielle ou totale.

TRAVAIL A FAIRE :

Présenter :

- Un dictionnaire des données.
- Une table (Matrice) et un graphe des dépendances fonctionnelles.
- Un Modèle conceptuel des données (MCD).
- Un modèle logique des données (MLD).

k. Association Réflexive.

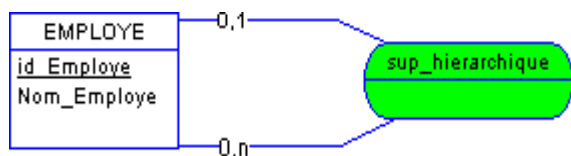
- Premier cas : cardinalité $(X,1) - (X,n)$, avec $X=0$ ou $X=1$.

La **Clé Primaire** de l'entité se dédouble et devient une **Clé Etrangère** dans la relation ou nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire $(X,1) - (X,n)$.

S.I. :

Prenons l'exemple d'une société organisée de manière pyramidale : chaque employé a 0 ou 1 supérieur hiérarchique direct. Simultanément, chaque employé est le supérieur hiérarchique direct de 0 ou plusieurs employés.

MCD :



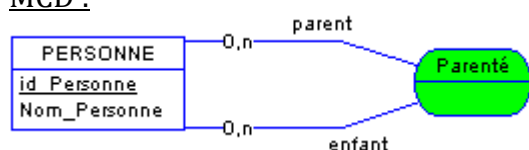
- Deuxième cas : cardinalité (X,n) - (X,n), avec X=0 ou X=1.

De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,n) - (X,n) (Cf règle 3). Il y a donc création d'une nouvelle table.

S.I. :

Prenons cette fois l'exemple d'une organisation de type familiale : chaque personne a 0 ou n descendants directs (enfants), et a aussi 0 ou n ascendants directs (parents).

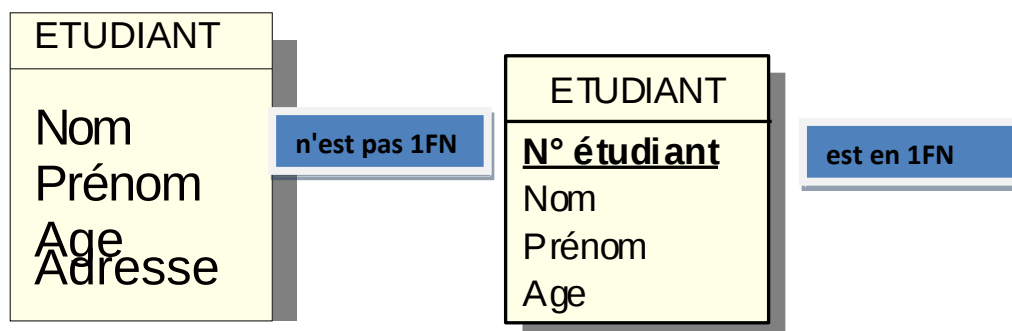
MCD :



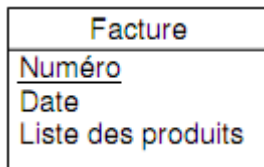
1. Normalisation du MCD : Les formes normales

1ère Forme Normale (1FN)

- Toutes les entités et les associations possèdent un identifiant
- Aucune propriété n'est à valeurs multiples (propriétés atomiques)

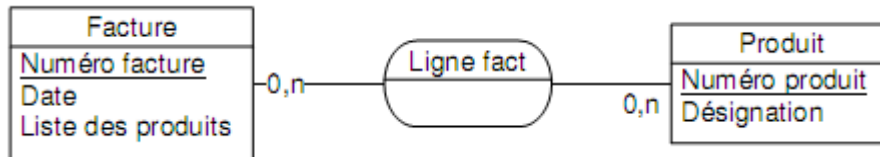


1)



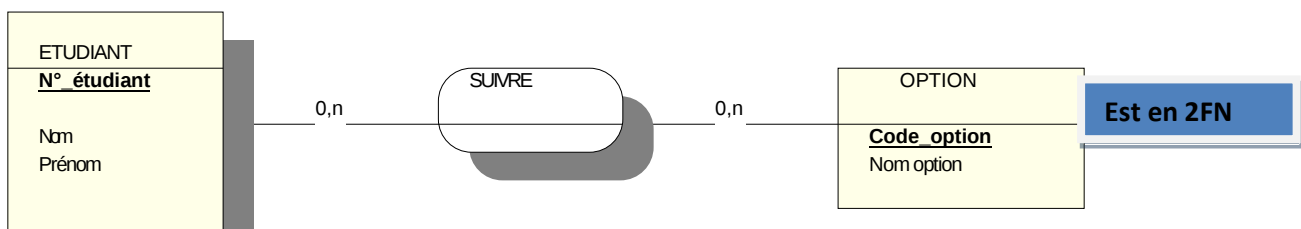
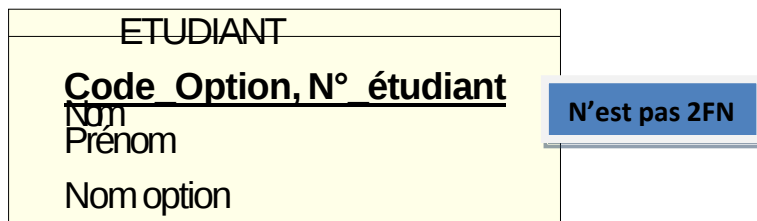
n'est pas en 1FN

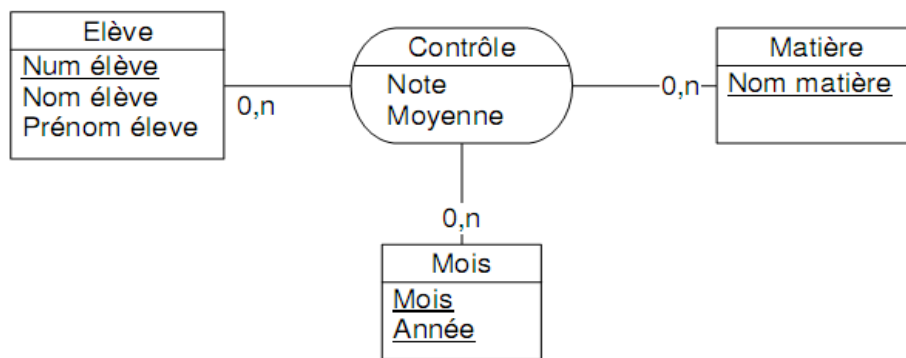
- Ici, « liste des produits » n'est pas atomique, c'est une liste
ou élémentaire
X



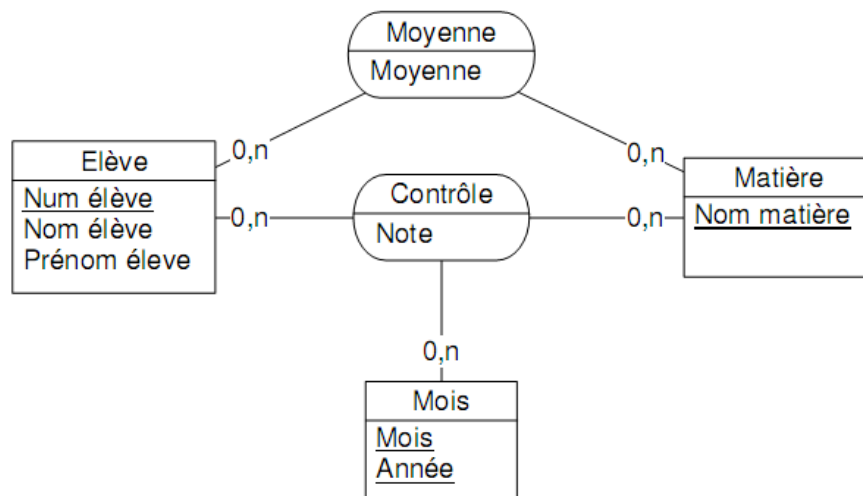
2e Forme Normale (2FN)

- Le modèle est en 1FN
- Toutes les DF entre les propriétés sont élémentaires
- Toute propriété n'appartenant pas à une clé ne dépend pas seulement d'une partie de son identifiant
- Les propriétés d'une entité ne doivent dépendre que de l'identifiant de l'entité et non d'une partie de cet identifiant



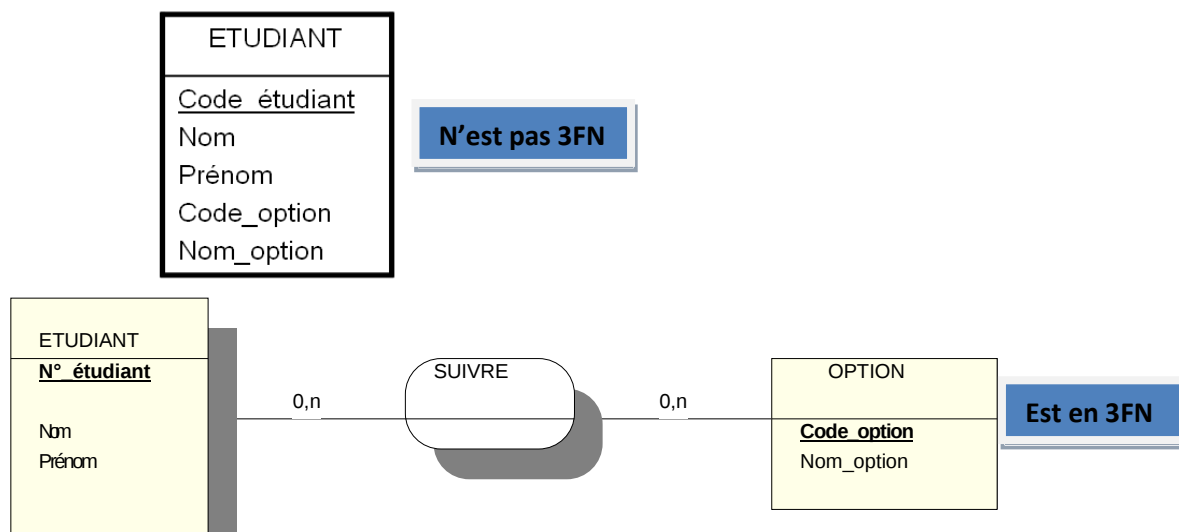


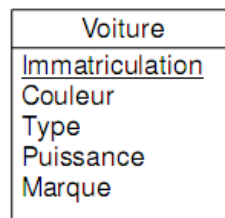
- Ici, d'après le schéma, NumEleve, NomMatière, Mois, Année → Moyenne
- Or, NumEleve et NomMatière suffisent



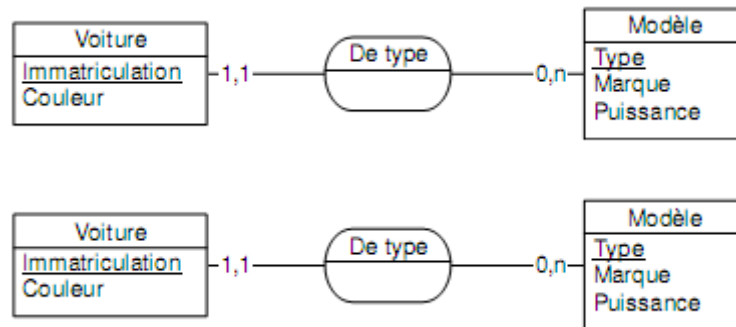
3e Forme Normale (3FN)

- Le modèle est en 2FN
- Toutes les DF entre les propriétés sont directes
- Les propriétés d'une entité doivent dépendre de l'identifiant de l'entité de manière directe
- Toute propriété n'appartenant pas à un identifiant ne dépend pas d'un attribut non identifiant

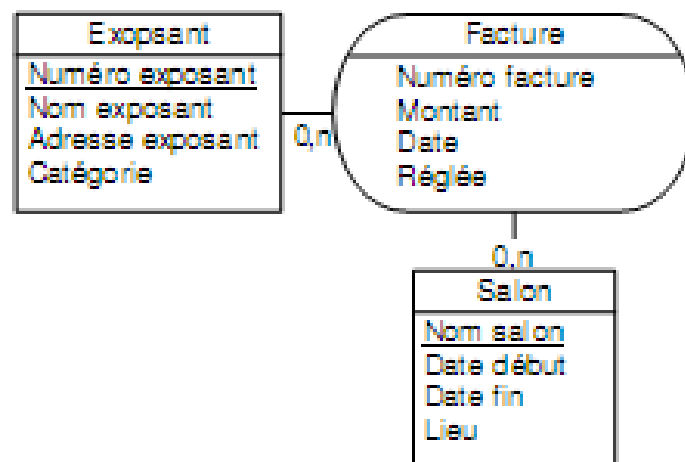




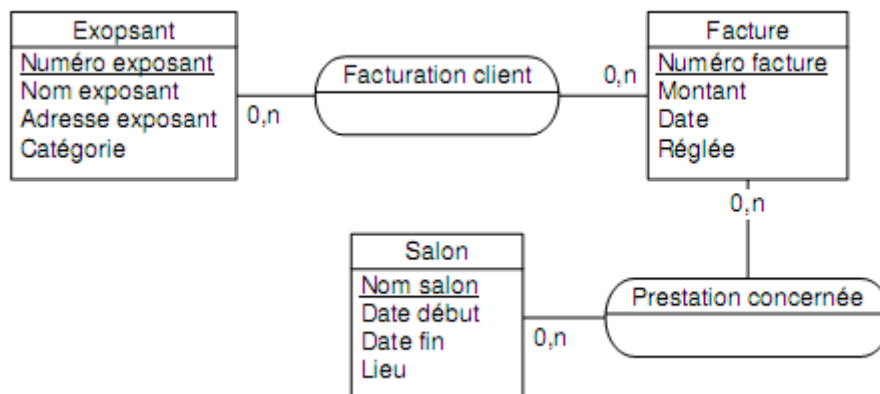
– Or, Type → Marque, Puissance alors que Type n'est pas un identifiant



– Très bien mais si on voulait rajouter un numéro de facture...



– Or, NumFact → Montant, Date, Réglée alors que NumFact n'est pas un identifiant



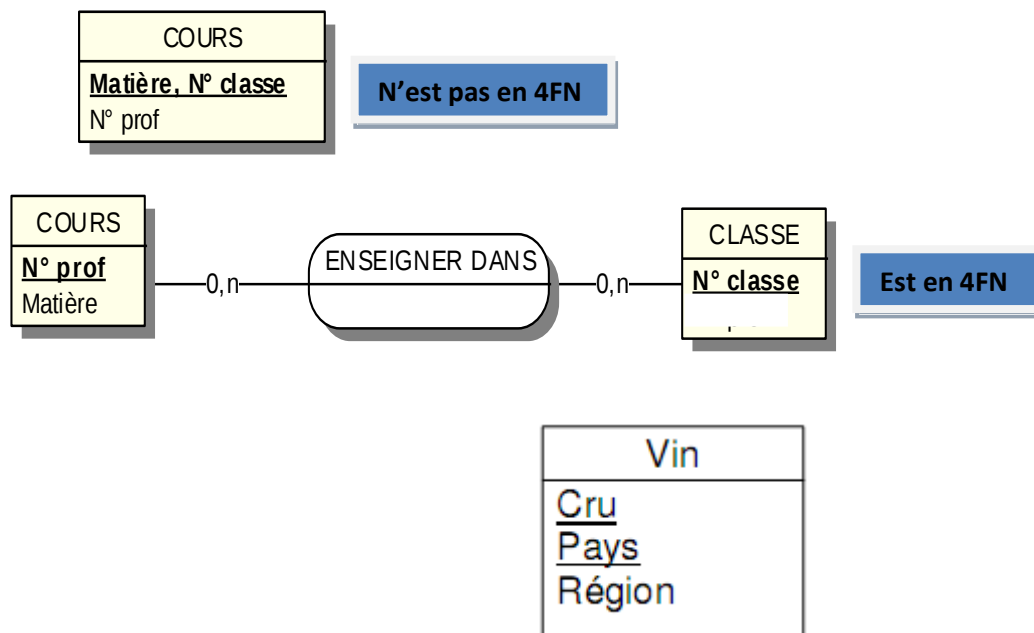
Forme normale de Boyce-Codd (BCNF)

- Le modèle est en 3FN
- Les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles un identifiant détermine une propriété
- Pour les identifiants composés de plusieurs propriétés, ces dernières ne doivent pas être dépendantes d'une autre propriété de l'entité (pour éviter les cycles de DF)

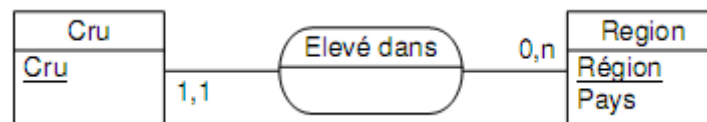
Exemple :

RG1 : tout prof enseigne une et une seule matière.

RG2 : toute classe n'a qu'un seul prof par matière.



– Or, Région → Pays



- On a alors, Cru → Region et Region → Pays
 - Attention : Même si elle peut être retrouvée par jointure, on a perdu la dépendance Cru, Pays → Région
- Un MCD ne doit pas nécessairement être en BCNF, il faut peser le pour et le contre avant de perdre des dépendances fonctionnelles

En résumé :

1FN : élémentarité des attributs et existence de l'identifiant.

2FN : DF élémentaire de l'identifiant.

3FN : DF directe de l'identifiant.

4FN : DF complète de l'identifiant :
Si l'identifiant est concaténé, un composant ne doit pas être en DF avec un autre attribut.

III. Règles de base pour la construction d'un MCD

A. Règles concernant les propriétés

1) 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.

2) 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 noteATM, noteMSI, noteAGL, etc.

B. Règles concernant les entités

3) 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.

4) 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.

C. Règles concernant les associations

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

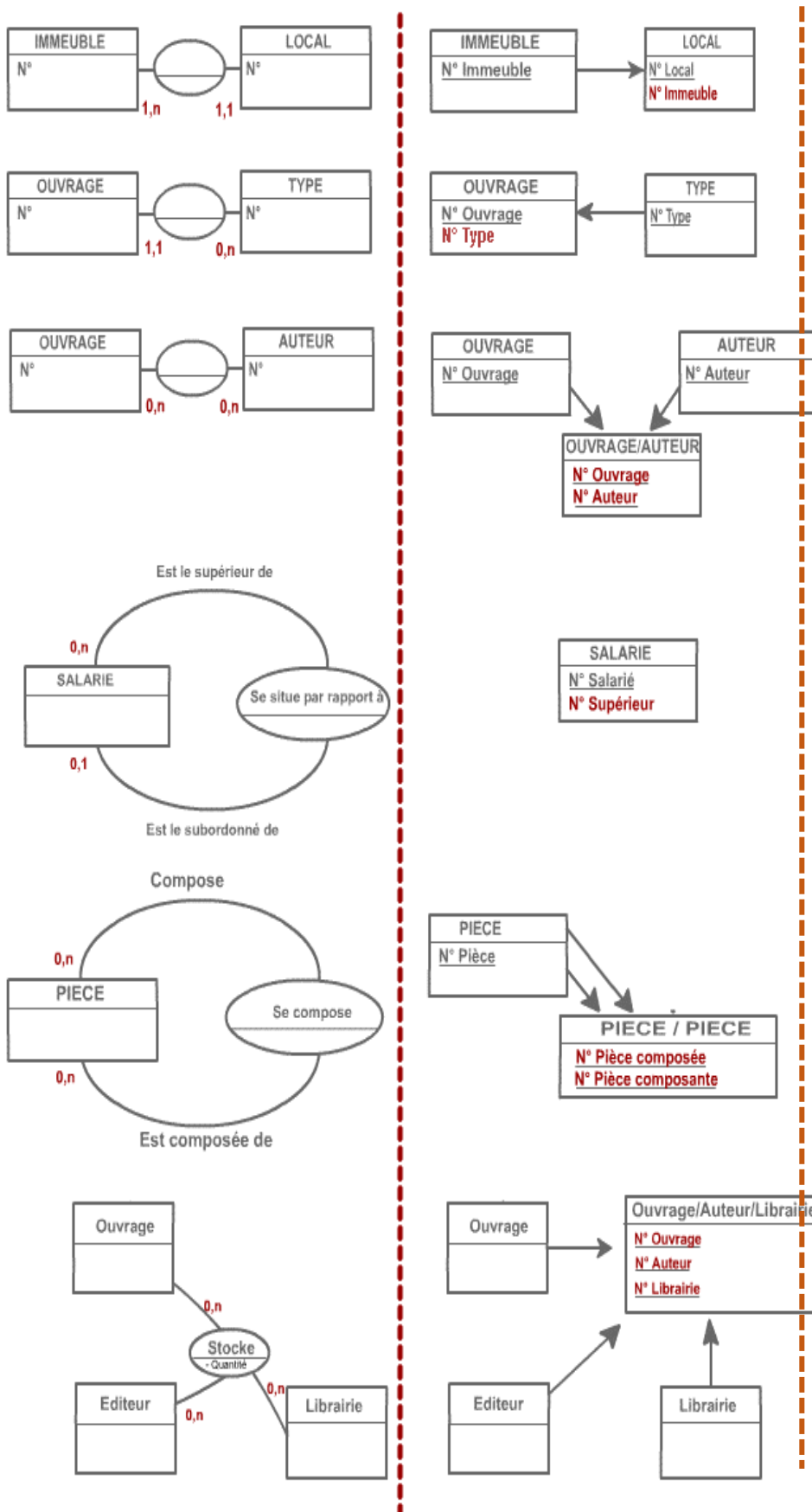
On ne représente pas cet identifiant au niveau du MCD. Deux occurrences d'association ne peuvent pas avoir le même identifiant. Pour une occurrence, l'identifiant ne doit jamais changer de valeur (l'identifiant est une propriété constante).

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

IV. MLD (Modèle Logique des Données)

C'est grâce à toutes les opérations précédentes que l'ensemble des tables de la base de donnée vont pouvoir être structurées de manière simple et très rapide : Le M.C.D. ayant été corrigé à la fin de l'étape du M.O.T., ce sont les cardinalités maximales qui vont jouer le rôle essentiel.

- Les entités deviennent des tables (sauf des cas particuliers comme les "dates")
- Si l'une des cardinalités maximales est à "1" et l'autre à "n", l'association disparaît et l'identifiant de l'entité marquée "n" vient s'ajouter à la liste des propriétés de l'entité marquée "1".
Remarque : Une telle association n'a pas lieu d'être porteuse de propriété, par conformité avec la "2ème Forme Normale".
- Si toutes les cardinalités maximales sont à "n", l'association se transforme en une table qui permettra la correspondance entre les enregistrements des tables reliées (tout en pouvant comporter ses propres propriétés).
- Ces règles s'appliquent aussi bien pour les associations "réflexives". Pour les associations de dimension "3" ou plus, elles sont toujours transformées en table.



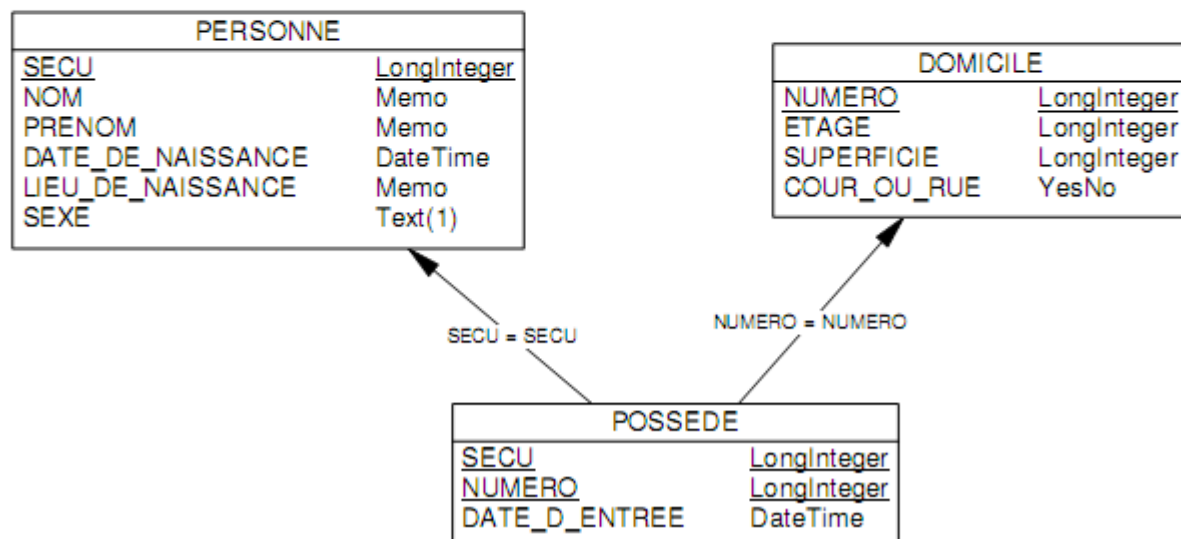
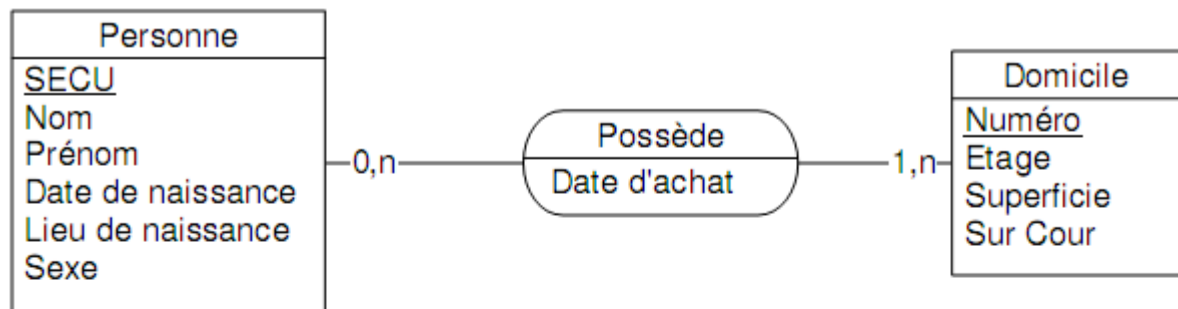
TRANSFORMATION DES ASSOCIATIONS

V. Modèle Physique des Données (MPD)

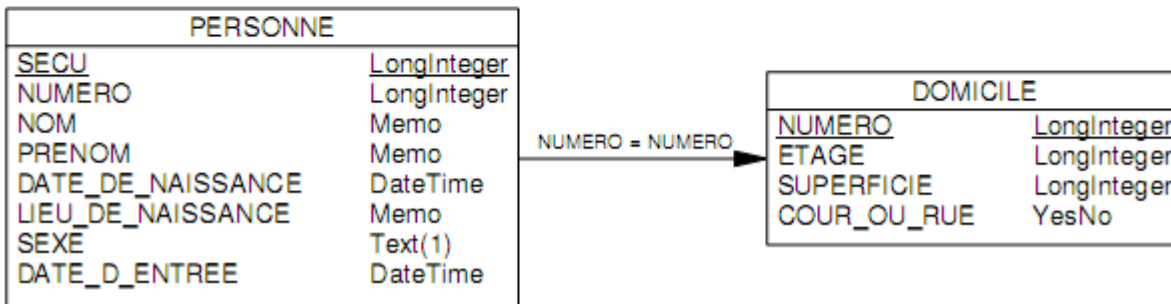
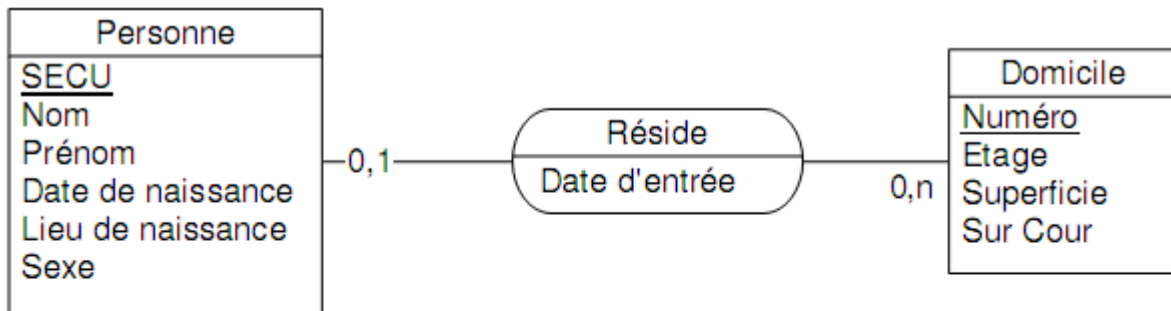
Transformation en tables des :

- Entités et associations si on produit le MPD directement à partir du MCD
- Schémas relationnels si on produit un MLD
- Dépend de la base de données cible
- Types de données
- Domaines des propriétés
- Les attributs qui permettent d'indexer les tables sont des clés primaires
- Les attributs (non clés primaires) qui font référence aux clés primaires d'autres tables sont des attributs secondaires.

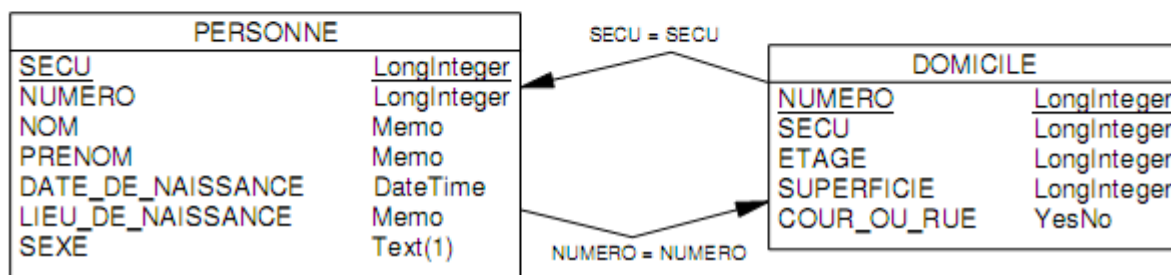
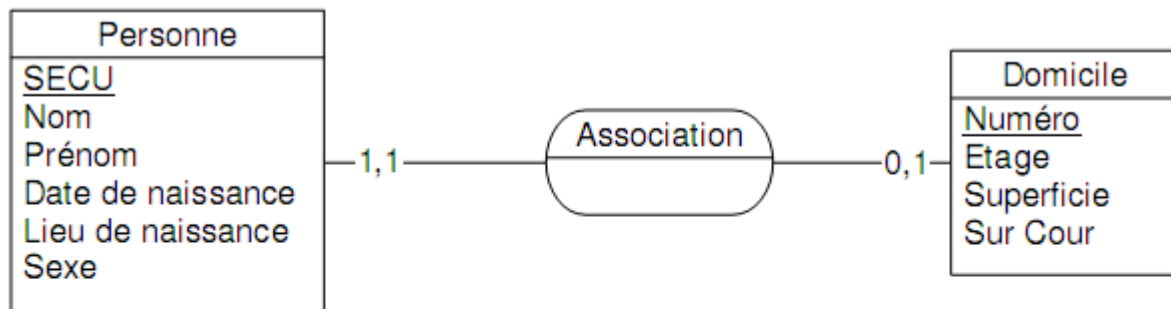
Associations n:m



Associations 1:n



Associations 1:1



VI. Les concepts étendus

Le modèle entité-association retenu par la méthode Merise date des années 70. Or les concepts de ce modèle peuvent s'avérer insuffisants pour modéliser certaines situations ou contraintes et l'on est obligé dans ce cas d'ajouter des commentaires pour en faire mention. Les extensions au modèle individuel remédient aux faiblesses du formalisme de base.

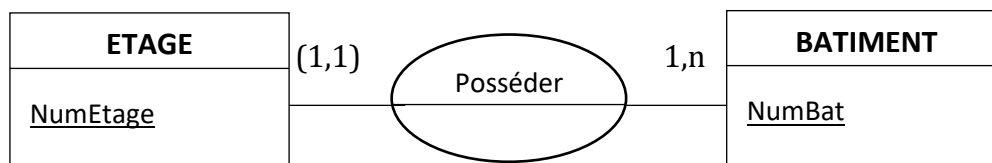
1. L'identification relative (entités dépendantes ou entités faibles)

Si un identifiant ne comporte que des propriétés de son entité, on le nomme "identifiant absolu". Les identifiants absolus se rencontrent dans le cas d'entités définies indépendamment les unes des autres.

D'autres entités sont identifiées par l'intermédiaire d'une ou plusieurs autres entités. Cela s'appelle l'identification relative" ou encore "agrégation".

- L'entité permettant l'identification est nommée "entité agrégeant".
- L'entité identifiée se nomme "entité agrégée".
 - Une entité faible ne peut être identifiée que par rapport à une autre entité forte à laquelle elle se réfère ;
 - Son identifiant sera son identifiant partiel plus l'identifiant de l'entité forte

L'identification relative se note de la manière suivante :



Remarque : l'identification relative n'existe que si les cardinalités exprimant l'identification relative sont $(1,1)$ et s'il y a stabilité dans le temps. (Un étage ne peut pas changer de bâtiment).

MLD relationnel correspondant :

Les règles de passage au schéma relationnel s'appliquent pour obtenir :

BATIMENT (NumBat)

ETAGE (NumEtage, #NumBat)

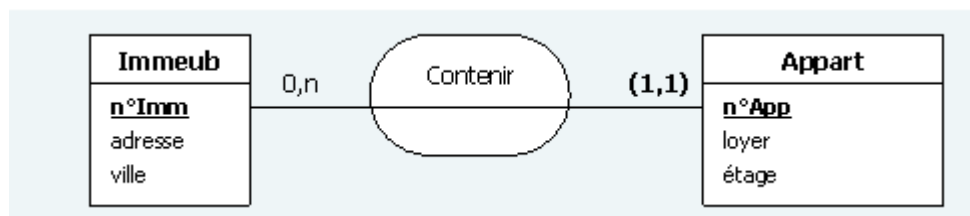
Exemple 2:

Immeuble et appartement : un appartement a un n° dans l'immeuble, si l'immeuble n'existe pas, l'appartement non plus.

L'identifiant de cette entité faible est dit identifiant relatif et nécessite la concaténation de celui de l'entité maîtresse.

L'appartement aura comme identifiant le n° de l'immeuble + celui de l'appartement dans l'immeuble dans le schéma relationnel.

Modélisation :



Le schéma relationnel de cette association est :

Immeub (n°Imm, ...)

Clé primaire : n°Imm

Appart (n°Imm, n°App, ...)

Clé primaire : n°Imm, n°App

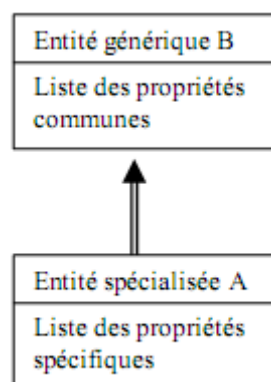
Clé étrangère : n°Imm référence n°Imm dans Immeub

La clé primaire de la relation dépendante est la concaténation de l'attribut de la relation déterminante suivi de l'attribut qui était l'identifiant de l'entité faible.

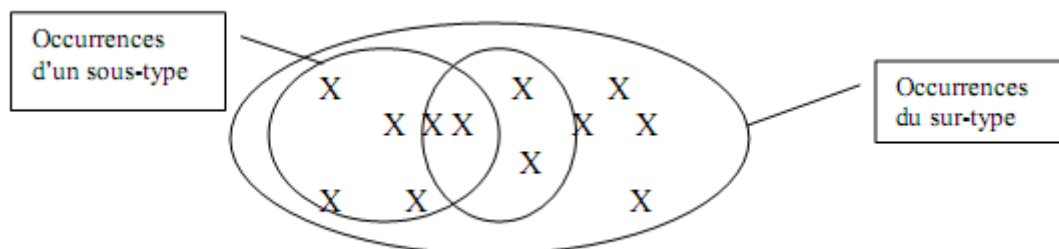
L'attribut de la relation déterminante est aussi clé étrangère.

2. Le concept d'héritage

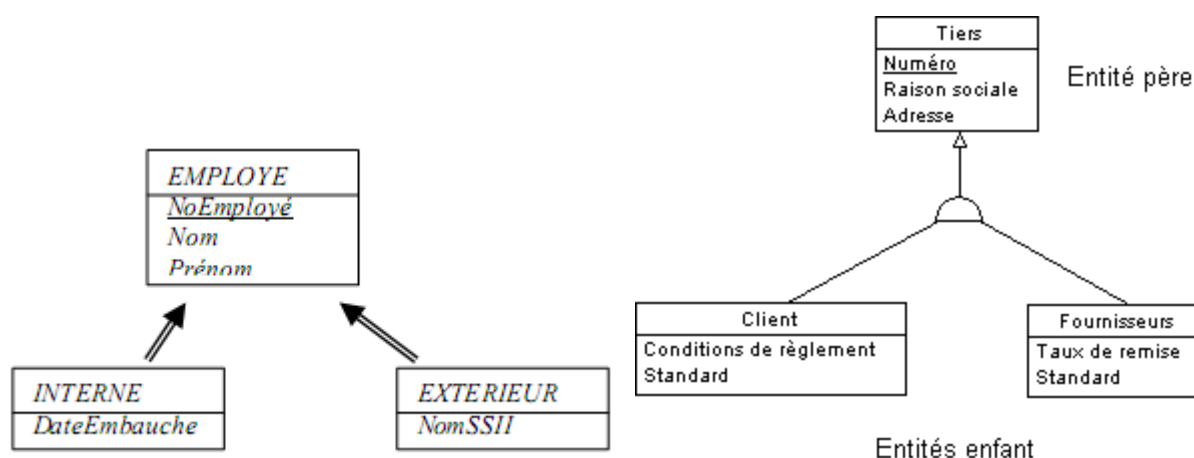
Quand le concepteur s'aperçoit que plusieurs entités, proches mais distinctes, partagent un ensemble de caractéristiques, il doit mettre en œuvre un processus de création d'entités génériques (ou entités sur-types) et d'entités spécialisées (ou entités sous-types) appelé «héritage». Ce concept qui permet de représenter le lien « est-un » ou « IS-A » entre deux entités A et B (une occurrence de A est une occurrence de B) est représenté graphiquement par une flèche double allant de A vers B.



On dit qu'il y a héritage simple quand un sous-type n'a qu'un seul sur-type. Dans ce cas, toutes les occurrences du sous-type sont en même temps des occurrences de son sur-type. Cela n'implique pas que toutes les occurrences du sur-type soient des occurrences de l'un des sous-types. Le schéma suivant illustre l'inclusion des ensembles d'occurrences des sous-types dans l'ensemble des occurrences du sur-type.



Le sous-type hérite de toutes les propriétés de son sur-type y compris de son identifiant.



Le sous typage est une orientation vers le monde "objet".

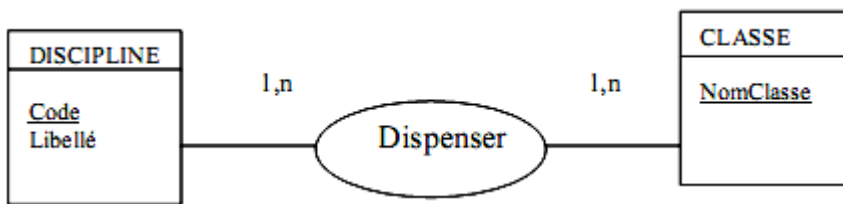
Application

On souhaite représenter le système d'information relatif à la composition du personnel intervenant dans un lycée de Lomé. Cette description devra fournir pour chaque personnel du lycée : administratif, professeur, surveillant, etc, un numéro qui permet de l'identifier (NUMEN) sans ambiguïté ainsi que les données signalétiques suivantes :

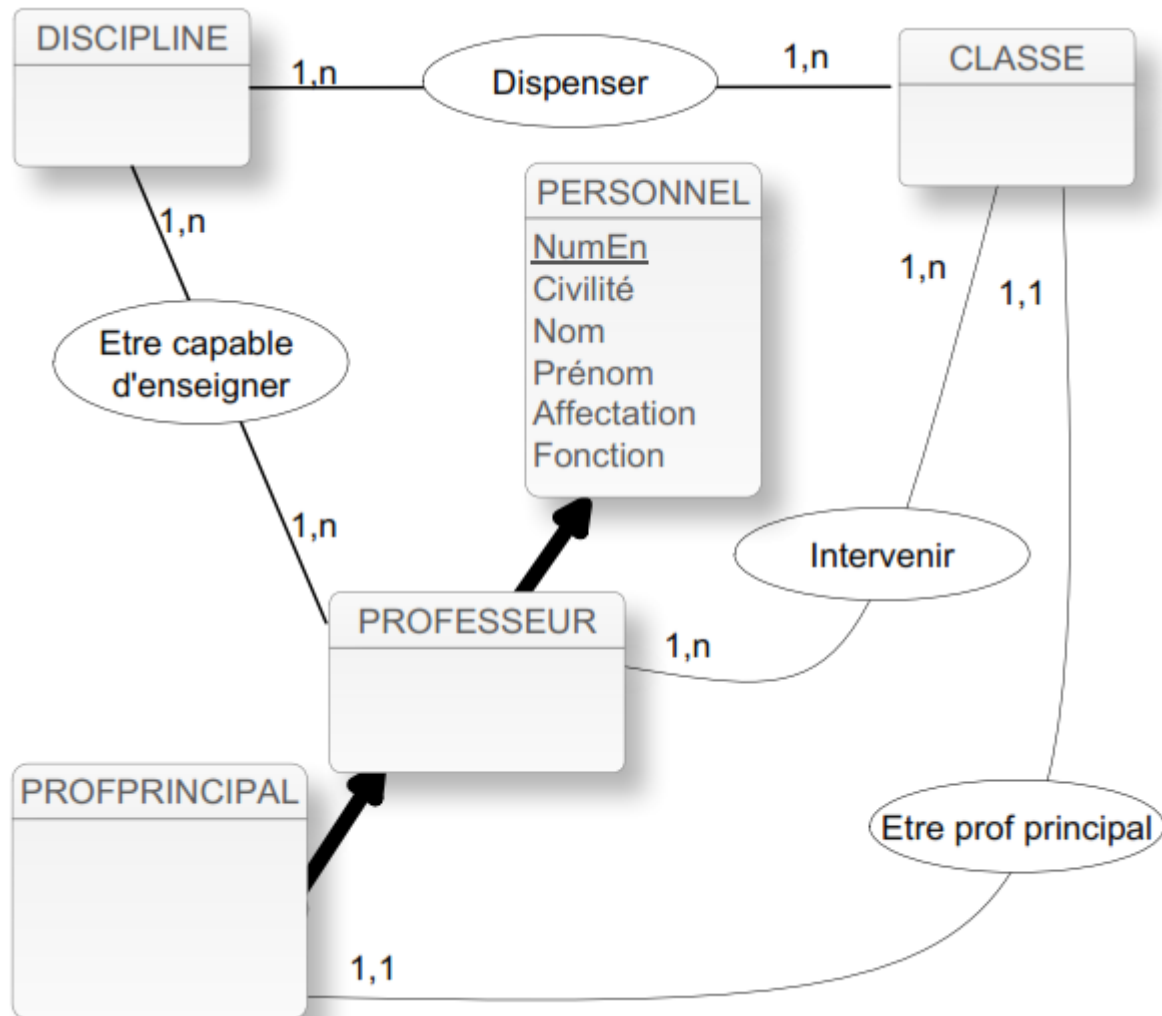
- civilité,
- nom,
- prénom,
- date d'affectation dans le lycée,
- fonction : professeur, proviseur, surveillant, ...

Parmi ces personnels, la représentation devra donner des renseignements supplémentaires sur les professeurs notamment la ou les discipline(s) qu'ils sont capables d'enseigner ainsi que les classes qu'ils ont en charge. La notion de professeur principal devra être aussi modélisée.

Compléter le modèle conceptuel des données suivant et mentionner sous forme de commentaires les contraintes qui sont non exprimables par les concepts de base.



Solution



Contraintes:

- On ne sait pas ce que le professeur enseigne à sa classe.
- Le professeur doit enseigner à la classe dont il est principal.

3. L'agrégation

Elle consiste à considérer globalement des entités et une association comme un objet unique. Cet objet peut ensuite être lié à d'autres entités ou agrégations via des relations. On peut également parler d' "association d'associations".

L'agrégation peut être représentée de deux manières différentes.

Un exemple d'agrégation est le suivant :



Passage au MLD relationnel

Les règles de passage au schéma relationnel s'appliquent pour obtenir :

JOCKEY (numJoc, nomJoc);

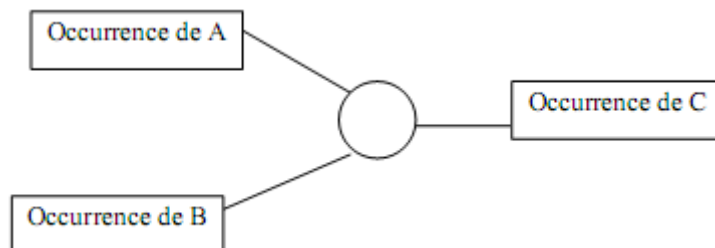
COURSE (numCourse, nomCourse, dateCourse);

CHEVAL (NumChev, NomChev, SexeChev);

EFFECTUER (#numJoc, #numCourse, #numChev);

4. Les contraintes ensemblistes

Ces formalismes vont permettre d'exprimer des contraintes sur des ensembles d'occurrences d'entités ou d'associations. On peut considérer que c'est le plus grand apport de Merise 2. Si la notion d'ensemble d'occurrences d'une entité ne pose aucun problème, il convient d'apporter certaines précisions sur l'ensemble des occurrences d'une association. Considérons pour cela une association ternaire reliant trois entités A, B et C. Une occurrence de cette association est un lien « tri-pattes » qui relie une occurrence de l'entité A, une occurrence de l'entité B et une occurrence de l'entité C.



En désignant par a1, b1 et c1 les valeurs des rubriques identifiantes de ces 3 occurrences d'entités, on peut matérialiser l'occurrence de l'association par le triplet : (a1, b1, c1). L'ensemble des occurrences d'une association peut donc être représenté par l'ensemble des triplets issus de la présence de liens entre les occurrences des entités A, B et C. Le tableau ci-dessous présente des exemples d'occurrences des associations « Travailler » et « Habiter ».

Modèle conceptuel des données	Occurrences de l'association TRAVAILLER	Occurrences de l'association HABITER
<i>Remarque : on suppose qu'il n'y a pas deux professeurs qui portent le même nom</i>	(« Dupont »,16) (« Dubois »,16) (« Durand »,17) (« Laforet »,17)	(« Dupont »,16) (« Dubois »,17) (« Durand »,16) (« Laforet »,86)

A partir de l'ensemble des occurrences d'une association, il est possible de construire d'autres ensembles en supprimant, dans chaque n-uplets, la participation d'une ou plusieurs entités. Dans l'exemple ci-dessus, si l'on considère uniquement l'entité DEPT (et donc on ne se préoccupe plus de l'entité PROF), l'ensemble des occurrences de l'association Travailler, limité à l'entité DPT, devient composé des éléments 16 et 17.

(« Dupont »,16)

(« Dubois »,16)

(« Durand »,17)

(« Laforet »,17)

On appelle pivot la (ou les) entité(s) que l'on retient pour former ces nouveaux ensembles. Pour simplifier les futures définitions, l'ensemble des occurrences d'un pivot participant à une association sera désormais référencé Ens(Pivot,Association). En gardant la même logique, la

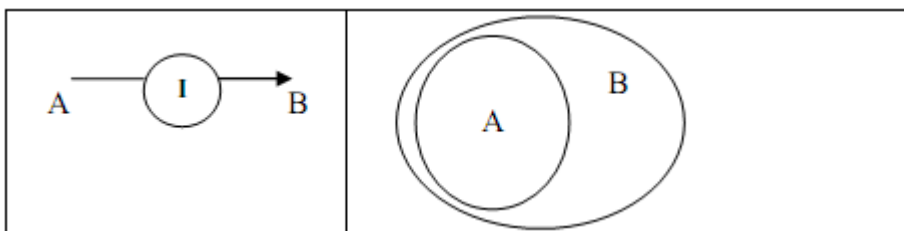
référence $\text{Ens}(\text{Entité})$ désignera l'ensemble des occurrences de l'entité citée et la référence $\text{Ens}(\text{Association})$ l'ensemble des occurrences de l'association.

Pour les associations réflexives, il est possible de définir, à partir de l'ensemble des occurrences, deux nouveaux ensembles, notés $\text{Ens}(\text{Rôle}, \text{Association})$ issus chacun des rôles portés par les deux segments

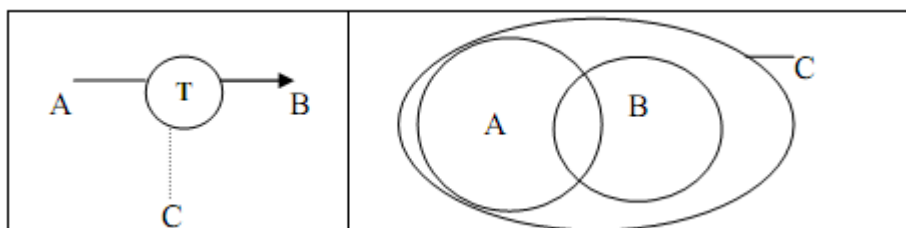
Quelle que soit leur origine, les ensembles sur lesquels portent une contrainte ensembliste doivent être compatibles c'est-à-dire que leurs éléments doivent avoir des structures (en terme de rubriques) identiques. Par exemple on peut définir une contrainte ensembliste entre $\text{Ens}(\text{TRAVAILLER})$ et $\text{Ens}(\text{HABITER})$. Par contre, il n'est pas possible de définir une contrainte ensembliste entre $\text{Ens}(\text{PROF})$ et $\text{Ens}(\text{TRAVAILLER})$ car leurs éléments n'ont pas la même structure.

Les contraintes ensemblistes qu'il est possible d'exprimer au moyen des concepts étendus sont présentées de façon générale ci-dessous :

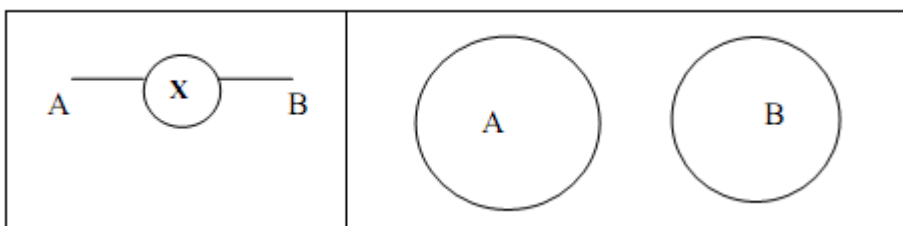
- La contrainte d'inclusion (notée I) entre deux ensembles A et B impose que l'ensemble A soit inclus dans l'ensemble B. L'inclusion n'étant pas symétrique, le formalisme associé à cette contrainte met en valeur, parmi les deux ensembles, celui qui doit être inclus dans l'autre.



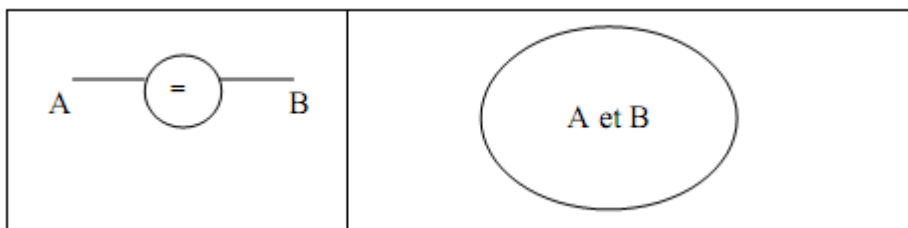
- La contrainte de totalité (notée T) porte sur trois ensembles A, B et C et impose que l'union de A et de B soit égale à C. Les trois ensembles ne jouant pas le même rôle, le formalisme prévoit un moyen de discerner les ensembles sur lesquels l'union sera réalisée et l'ensemble qui doit être égal à cette union.



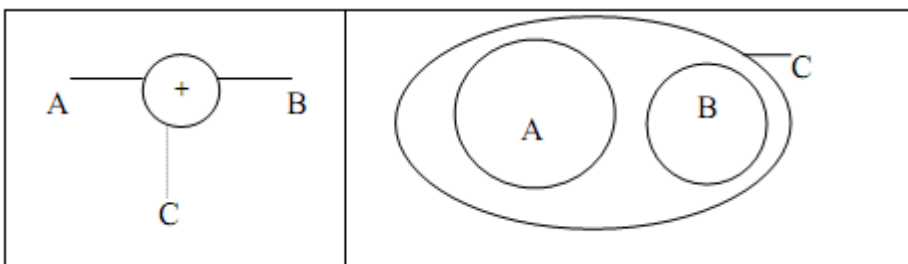
- La contrainte d'exclusion (notée X) entre deux ensembles A et B impose que l'intersection entre A et B soit vide.



- La contrainte d'égalité (notée =) entre deux ensemble A et B impose que l'ensemble A soit égal à l'ensemble B. Elle traduit en fait les deux inclusions suivantes : A inclus dans B et B inclus dans A.



- La contrainte du ou exclusif (notée +) porte sur trois ensembles et est la combinaison d'une exclusion et d'une totalité. Elle traduit le fait que l'intersection entre A et B est vide et que l'union de A et B est égale à l'ensemble C.



Les contraintes d'égalité et du « ou exclusif » étant dérivées respectivement des contraintes d'inclusion, de totalité et d'exclusion, seules ces trois dernières font l'objet d'une description plus précise dans les paragraphes qui suivent.

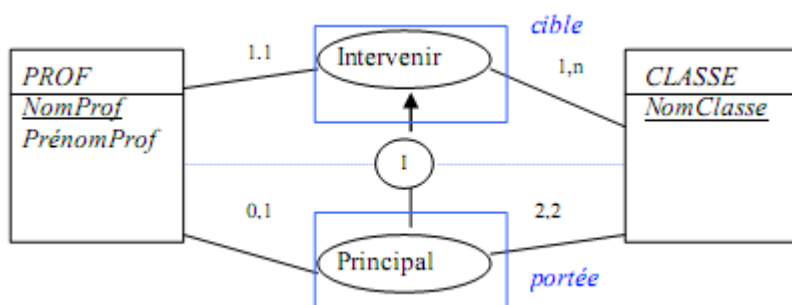
a) La contrainte d'inclusion

La contrainte d'inclusion s'exprime entre une ou plusieurs associations sources appelées portée et une seule association destination appelée cible. Si le pivot n'est pas précisé graphiquement (au moyen de traits en pointillés), il est nécessaire de le déterminer en appliquant la règle suivante : il est constitué des entités communes aux associations de la portée et de la cible. La contrainte d'inclusion garantit alors que $\text{Ens}(\text{Pivot}, \text{Portée})$ est inclus dans $\text{Ens}(\text{Pivot}, \text{Cible})$. Dans l'exemple suivant le pivot est implicite et correspond aux entités ELEVE et CLASSE. On a donc les deux égalités suivantes :

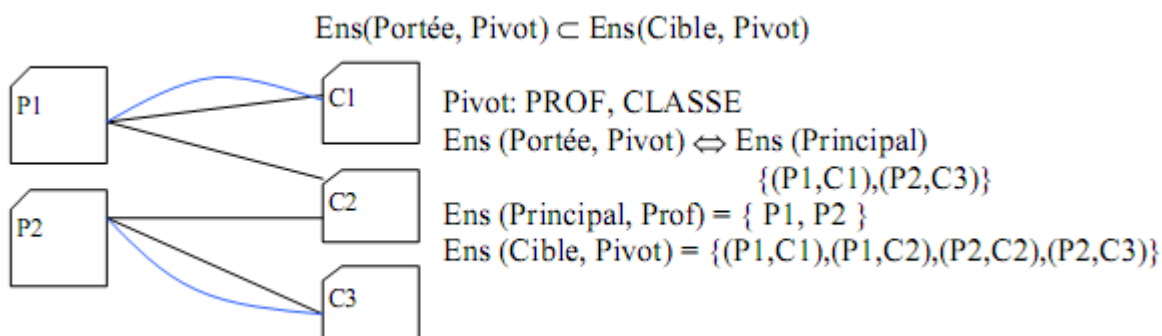
$$\text{Ens}(\text{Pivot}, \text{Portée}) = \text{Ens}(\text{Principal})$$

$$\text{Ens}(\text{Pivot}, \text{Cible}) = \text{Ens}(\text{Intervenir})$$

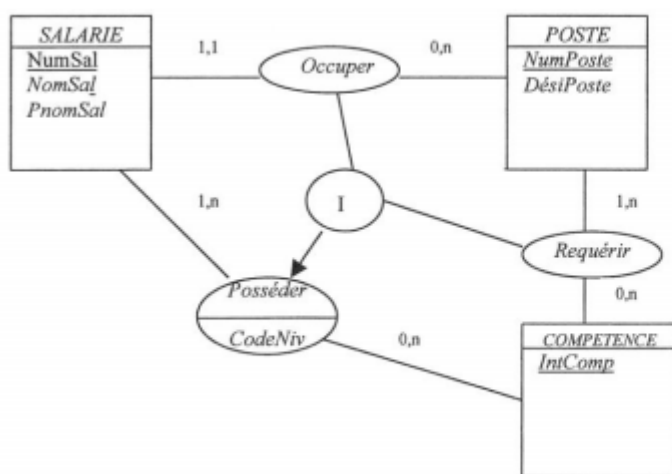
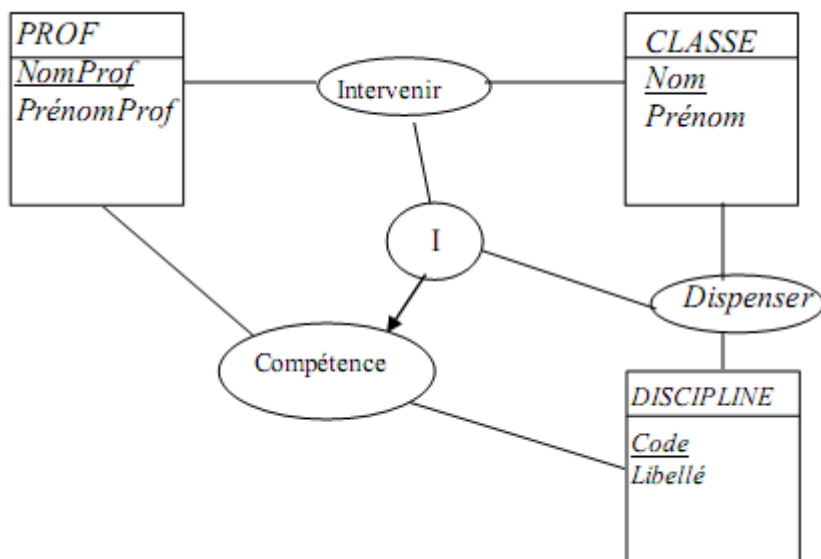
La contrainte d'inclusion assure que tout professeur principal d'une classe intervient dans la classe.



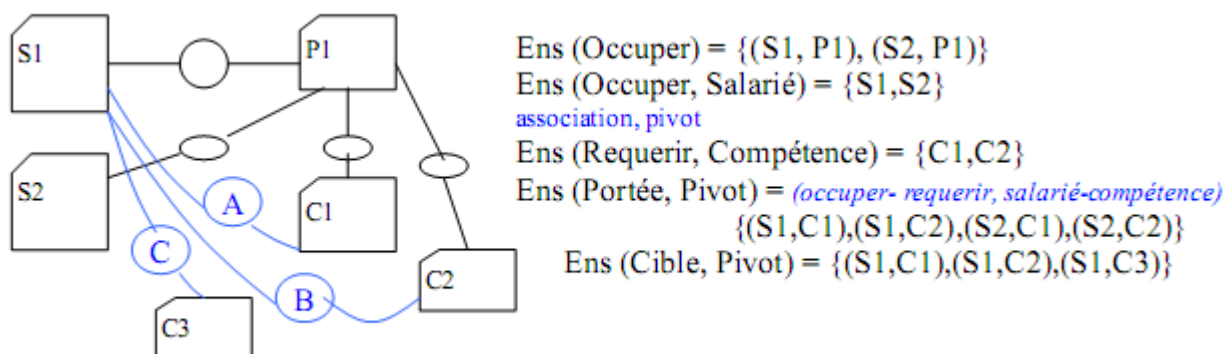
On note le pivot explicite par des pointillés, alors que le pivot implicite est formé par les entités communes à la portée et à la cible).



La cible est toujours simple, mais la portée peut être composée. Dans le cas où la portée est composée (c'est-à-dire formée de plusieurs associations), la détermination de $\text{Ens}(\text{Pivot}, \text{Portée})$ impose des opérations de jointure comme le montre les exemples suivants :



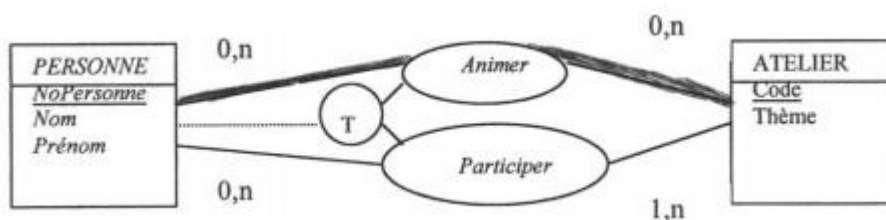
Le pivot implicite est formé par les entités : SALARIE et COMPETENCE. La portée est formée par les deux associations : Occuper et Requérir. Pour trouver l'ensemble $\text{Ens}(\text{Pivot}, \text{Portée})$, il faut pour chaque Salarié, récupérer les compétences qu'il requiert. Tous les couples (Salarié, Compétence) trouvés en empruntant ce chemin forment l'ensemble $\text{Ens}(\text{Pivot}, \text{Portée})$ qui doit être inclus dans l'ensemble $\text{Ens}(\text{Posséder})$.



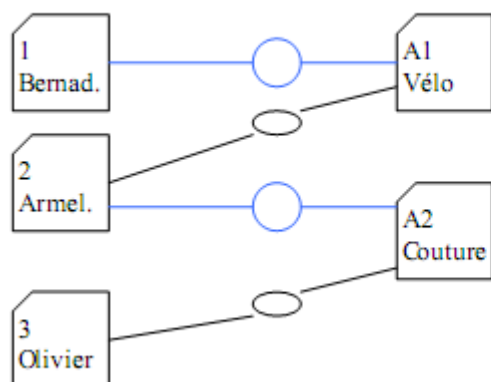
b) La contrainte de totalité

➤ La contrainte de totalité entre associations

Cette contrainte s'exprime entre n associations ($n \geq 2$) et impose que le pivot implicite ou explicite soit composé d'une seule entité. La contrainte garantit que l'union de tous les ensembles $\text{Ens}(\text{Pivot}, \text{Association } i)$ (avec i compris entre 1 et n) soit égal à $\text{Ens}(\text{Pivot})$. Dans l'exemple qui suit la contrainte de totalité impose que toute personne est soit animateur d'un atelier, soit participant soit les deux.



$$\text{Ens}(\text{AssociationA}, \text{Pivot}) \cup \text{Ens}(\text{AssociationB}, \text{Pivot}) = \text{Ens}(\text{Entité Pivot})$$



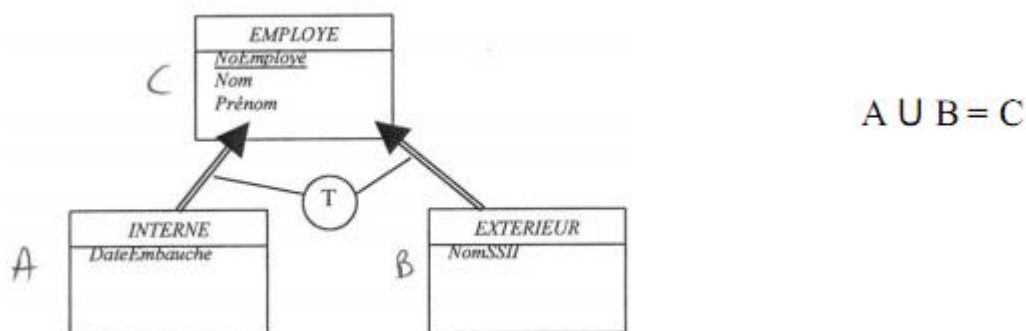
$$\begin{aligned} \text{Ens (Animer)} &= \{(1, A1), (2, A2)\} \\ \text{Ens (Animer, Personne)} &= \{1, 2\} \\ \text{Ens (Participer, Personne)} &= \{2, 3\} \end{aligned}$$

$$\text{Ens (Participer, Personne)} \cup \text{Ens (Animer, Personne)} = \{1, 2, 3\}$$

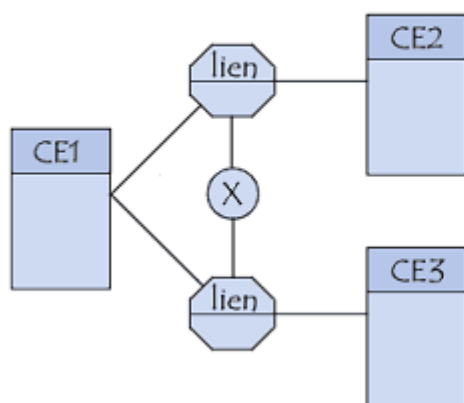
➤ La contrainte de totalité entre sous-types

Cette contrainte s'exprime entre n sous-types ($n \geq 2$). Elle traduit le fait que l'union des ensembles $\text{Ens}(\text{SousType } i)$ (avec i compris entre 1 et n) soit égale à $\text{Ens}(\text{SurType})$. Toute occurrence du sur-type correspond donc à une occurrence d'un ou de plusieurs sous-types.

Par exemple pour modéliser le fait que tous les employés du service informatique du CENETI sont soit des internes soit des extérieurs on placerait une contrainte de totalité entre les deux sous-types.



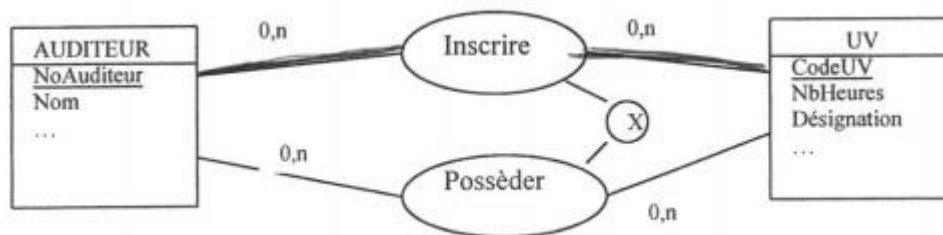
➤ La contrainte d'exclusion entre associations



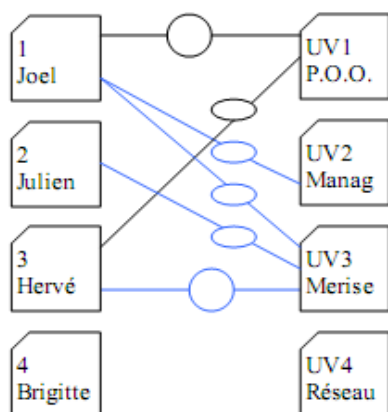
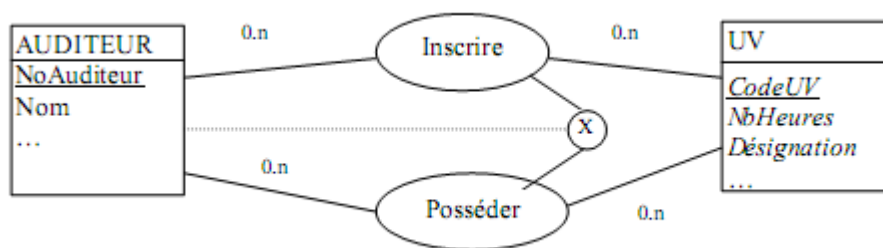
La contrainte d'exclusion sur relation exprime le fait que **deux occurrences de classes d'entité ne peuvent pas participer simultanément à une même classe de relation.**

Elle est représentée par un "X" reliant deux classes de relation.

Cette contrainte s'exprime entre n associations ($n \geq 2$). Si le pivot n'est pas précisé graphiquement le pivot implicite est composé des entités communes aux différentes associations en jeu. La contrainte d'exclusion garantit que l'intersection des Ens (Pivot, Association i) (avec i compris entre 1 et n) est vide. Dans le modèle suivant, la contrainte d'exclusion permet d'exprimer qu'un auditeur ne peut, à la fois, être inscrit à une UV et la posséder. Le pivot implicite est composé des entités AUDITEUR et UV.



Le modèle suivant, qui fait apparaître un pivot explicite, interdit qu'un auditeur apparaisse à la fois dans une occurrence de l'association *Inscrire* et dans une occurrence de l'association *Posséder*. Cela signifie que tout auditeur doit soit être inscrit à un ensemble d'UV soit posséder un ensemble d'UV mais pas les deux à la fois. Une telle modélisation n'a bien évidemment pas de sens.



Pivot: Auditeur et UV

Ens (Posséder, Pivot) = {(1,UV1), (3,UV1)}

Ens (Inscrire, Pivot) = {(1,UV2), (1,UV3), (2,UV3), (3,UV3)}

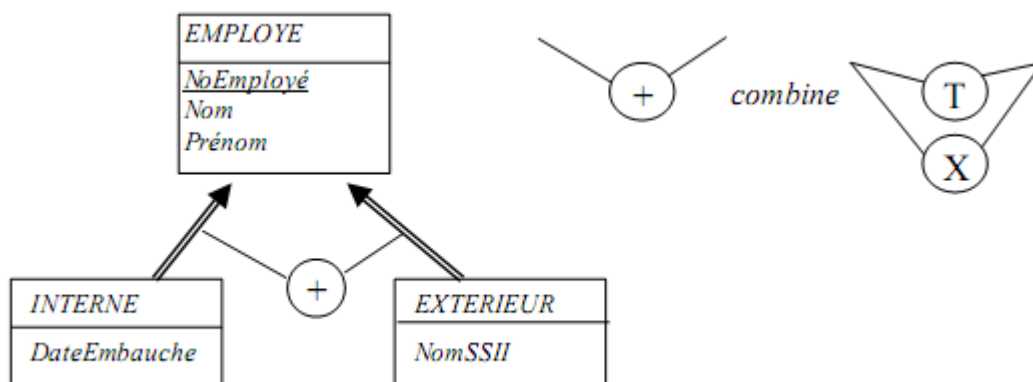
Pivot: Auditeur

Ens (Inscrire, Pivot) = { 1, 2, 3 }

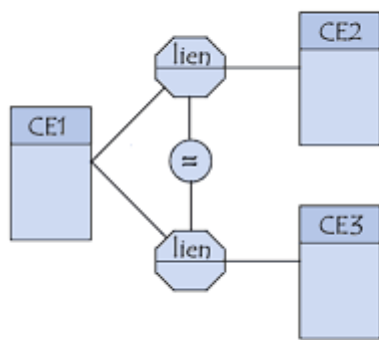
Ens (Posséder, Pivot) = { 1, 3 }

➤ **La contrainte d'exclusion entre sous-types**

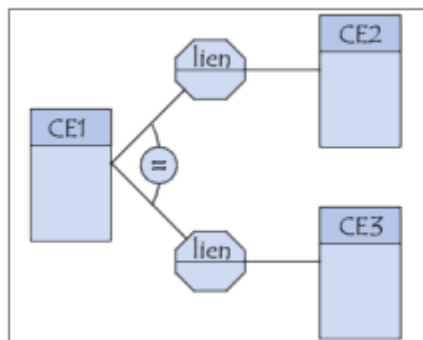
Cette contrainte s'exprime entre n sous-types ($n \geq 2$) et traduit le fait que l'intersection des $\text{Ens}(\text{SousTypes}_i)$ (avec i compris entre 1 et n) est vide ce qui revient à dire que toute occurrence de l'entité sur-type ne peut participer qu'à l'un ou l'autre (ou aucun) des sous-types de la contrainte. Dans l'exemple relatif à la modélisation du personnel du service informatique, un employé est soit un extérieur soit un salarié interne mais il ne peut pas être les deux à la fois.



d) La contrainte d'égalité

➤ La contrainte d'égalité entre associations

La contrainte d'égalité entre associations exprime le fait qu'une occurrence de classe d'entité participant à une classe d'association, participe obligatoirement à l'autre classe d'association, et réciproquement. Il s'agit donc d'une contrainte de sous-ensemble bidirectionnelle. Elle est représentée par un signe "=" reliant deux classes d'association. Cette contrainte peut faire intervenir plusieurs occurrences de classes d'entité, auquel cas une occurrence de classe d'entité participant à une classe de relation doit participer aux n classes d'association.

➤ La contrainte d'égalité entre sous-types

La contrainte d'égalité entre sous-types exprime le fait qu'une entité participant à une classe d'association, participe obligatoirement à l'autre association, et réciproquement. Il s'agit donc d'une contrainte de sous-ensemble bidirectionnelle.

Elle est représentée par un signe "=" reliant deux classes d'entités. Cette contrainte peut faire intervenir plusieurs associations, auquel cas une entité participant à une association doit participer aux n associations.

VII. MODELE ORGANISATIONNEL DES DONNEES

Il est facile de décrire la méthode MERISE de l'analyse organisationnelle, encore que son application exige à coup sûr savoir et pratique.

La modélisation organisationnelle des données va prendre en compte des éléments relevant de l'utilisation des ressources de mémorisation :

- Choix des informations à mémoriser informatiquement.
- Quantification des informations à mémoriser (volume et durée de vie).
- Répartition des données informatisée entre unités opérationnelles.

1. Choix des informations à mémoriser

Il s'agit de distinguer, à partir des informations formalisées sur le MCD, celles qui devront être mémorisées informatiquement dans le système d'information informatisé (SII), et les autres.

2. Quantification des informations à mémoriser

La quantification prend en compte deux notions :

- Le volume : taille et nombre de chaque élément.
- La durée de vie : statistiques sur le nombre minimum, maximum et moyen d'occurrences concrètes pour chaque entité et chaque association.

Cycle de vie

Pour analyser le cycle de vie des informations, on part du MCT, et on regarde, pour chaque opération, quelles sont les données qui sont créées et quelles sont celles qui sont modifiées.

Tableau de quantification

Pour chaque entité et pour chaque association, on calcule le volume théorique d'une occurrence, à partir du volume théorique d'une occurrence d'un attribut.

Pour toutes les entités et les associations, on détermine le nombre minimum, maximum et moyen d'occurrences. On regroupe l'ensemble des informations dans un tableau.

3. Répartition des données et droits d'accès

On va analyser au niveau du MOD la répartition concrète des données entre les unités opérationnelles de l'entreprise.

Dans le cas des données non informatisées, il faudra préciser leur localisation.

Dans le cas des données informatisées, on va préciser les droits des différents utilisateurs (les acteurs du MOT).

Ces droits peuvent être :

- Lecture
- Écriture
- Création
- Suppression

Chacun de ces droits s'appliquant aux entités, aux attributs, aux associations et à leurs occurrences.

VIII. Modèle Conceptuel de la Communication (MCC)

A. Diagramme des flux

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

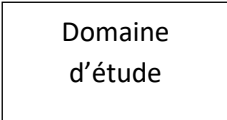
En ce sens, ce sont des modèles FONCTIONNELS (qui décrivent les fonctions) et peut être présenté processus par processus.

1. Vocabulaire associé aux modèles de flux

a. 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.



Domaine
d'étude

b. 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 le modèle de flux, un domaine connexe est représenté par un rectangle (ou un rond). Le nom du domaine connexe est placé à l'intérieur du rectangle.

c. activité

L'activité est un ensemble de traitements homogènes qui transforment ou manipulent des données. Une activité peut souvent être vue comme un sous -domaine d'étude, un morceau du domaine d'étude.

Chaque activité peut être éclatée. Cet éclatement se traduit alors par l'élaboration d'un nouveau diagramme qui décompose ce processus éclaté en plusieurs processus plus élémentaires.

d. Acteurs

– Représenté par un cercle ou ellipse en trait plein (acteur interne) ou en trait discontinu (acteur externe) dans lequel est inscrit le nom de l'acteur le nom de l'acteur

– L'acteur représente une unité active intervenant dans le fonctionnement d'un système opérant.

Il peut

– Etre stimulé par des flux d'information

– Transformer et émettre des flux d'information

– Un acteur « fait quelque chose », il est actif

– Ex : Service comptabilité, Guichet ...

– Un acteur est un rôle plutôt qu'une personne physique (« Direction » et pas « M. Jean-Claude »)

– Il peut être pertinent de modéliser séparément deux fonctions assumées par une même personne physique

On distingue les acteurs internes et externes :

➤ Acteur externe

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

➤ Acteur interne

- Acteurs faisant partie du système d'information étudié
- Ex : guichet, service informatique...
- Si le système est complexe, on peut considérer un acteur interne comme un sous-domaine et détailler ce sous-domaine dans un nouveau MCC

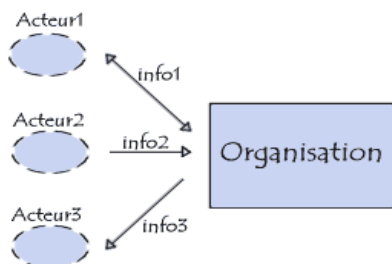
e. Flux de données

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.

Il existe 2 types principaux de diagrammes de flux :

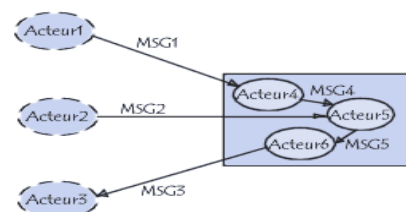
- 1) Le modèle de contexte (MC) où le domaine d'étude est vu comme une boîte noire. On ne représente que les flux extérieurs au domaine.



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

- 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.

Ce diagramme (appelé aussi **modèle conceptuel de la communication**) permet de compléter le diagramme de contexte en décomposant l'organisation en une série d'acteurs internes.



B. Graphe des flux

Il est obtenu à partir du diagramme de flux en prenant soins de gommer tout ce qui est de nature organisationnel puis en ordonnant bien les flux, dans leur ordre d'apparition.

- Graphe des flux bruts (GFB) : est issu du diagramme des flux.
- Graphes des flux nets GFN) : c'est le graphe des flux brut dans lequel on a exclu les flux de nature organisationnel.

IX. Modèle Conceptuel des Traitements (MCT)

Le Modèle Conceptuel des Traitements (MCT) décrit les traitements et plus précisément toutes les activités découlant des échanges entre le domaine étudié et le monde extérieur. Il exprime donc ce que fait le domaine sans se poser le problème de savoir qui le fait, quand et comment. Dans les MCT, on représente comment un acteur de l'organisation réagit quand il reçoit ce message et quelle opération il effectue.

A. Les concepts de base

2. L'acteur

Un acteur est une personne morale ou physique capable d'émettre ou de recevoir des informations.

3. L'événement

L'événement matérialise un fait, qui en se produisant, doit déclencher une réaction du système.

4. L'opération

La réponse à l'arrivée d'un événement est le déclenchement d'un ensemble de traitements appelé opération qui est un ensemble d'actions élémentaires. Lors de son exécution une opération ne peut pas être interrompue par l'attente d'un événement externe.

5. Le résultat

Il est produit par une opération et peut être un événement pour une autre opération.

6. La synchronisation

La synchronisation d'une opération est composée de deux éléments :

- d'une part la liste des événements (internes ou externes) qui doivent être arrivés avant de déclencher l'opération.
- et d'autre part la règle sous forme d'une proposition logique qui précise de quelle manière les événements participent au déclenchement de l'opération.

7. Les règles d'émission

Ce sont des conditions (généralement des propositions logiques) traduisant les règles de gestion auxquelles est soumise l'émission des résultats d'une opération.

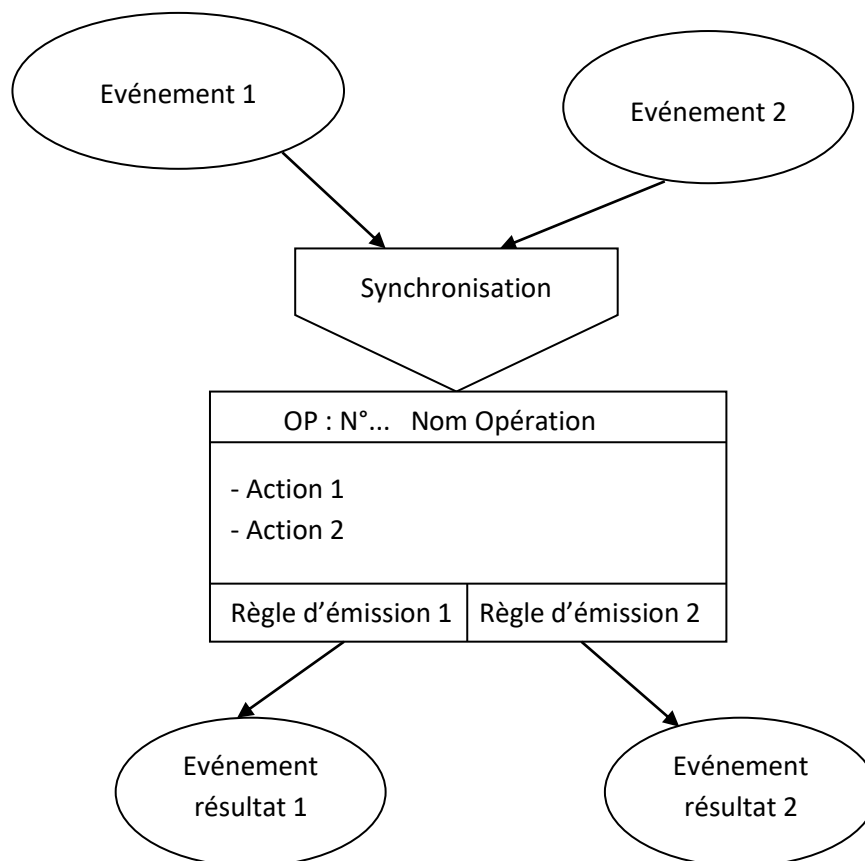
8. Processus

Il est un sous – ensemble cohérent d'un domaine incluant un ensemble structuré d'événements, d'opérations et de résultats consécutifs qui concourent à un même objectif.

9. Règles de gestion

Ce sont générales à toute l'organisation s'appliquant à une ou plusieurs opérations. Elles déterminent les conditions de fonctionnement dans le domaine étudié en faisant abstraction des contraintes d'organisation.

B. La représentation graphique



Remarque: Le MCT conditionne complètement l'interface graphique du S.I. et est généralement présenté processus par processus.

Application 1

Gestion des ventes immédiates des pièces détachées dans une société.

Le client se présente au comptoir et demande auprès du vendeur les renseignements (disponibilité, quantité, prix) sur les pièces à acheter. Celui-ci vérifie auprès du magasinier la disponibilité de ces pièces, puis il communique au client les quantités disponibles des pièces demandées ainsi que leurs prix respectifs. Le client communique au vendeur ses choix d'achat. Ce dernier lui établit une facture qu'il doit payer à la caisse. Le caissier en retour lui délivre un reçu qu'il doit présenter au vendeur pour retirer les pièces achetées.

TAF :

- ❖ Faire le diagramme des flux.
- ❖ Déduire le graphe des flux brut et net.
- ❖ Lister les règles de gestion.
- ❖ Réaliser le MCT.

Application2 : Gestion des commandes de clients

Les commandes des clients sont saisies au niveau des comptoirs d'accueils régionaux et sont transmises chaque soir au siège central de la société.

Tous les matins, le service magasin du siège traite les commandes reçues la veille : il édite pour chaque commande retenue des bons de réquisition.

Ces bons sont transmis aux magasiniers qui sortent les articles des stocks, pointent manuellement les bons, puis apportent les articles au service emballage, afin qu'ils soient conditionnés le jour même.

Dès le lendemain, le service emballage expédie les colis aux clients puis avise le centre de traitement informatique par une copie du bon de livraison.

En fin de semaine, le centre de traitement informatique du siège édite et envoie aux clients les factures qui correspondent aux commandes de la semaine.

TAF :

- ❖ Faire le diagramme des flux.
- ❖ Déduire le graphe des flux brut et net.
- ❖ Lister les règles de gestion.
- ❖ Réaliser le MCT.

X. M.O.T. ("Modèle organisationnel des traitements")

Représentation des "opérations" du M.C.T. sous une forme détaillée, puisque l'**organisation interne** de l'entreprise est ici prise en compte.

Cette représentation équivaut, en quelque sorte, à un "zoom avant" sur le M.C.T. Dans la pratique, il s'agit d'une étape longue et minutieuse.

Chaque "opération" du M.C.T. se trouve donc subdivisée ici en "procédures fonctionnelles" ("P.F.") qui sont, elles aussi, interruptibles.

Remarque : Certains auteurs remplacent "procédures fonctionnelles" par le terme de "phase".

MOT = MCT + lieu + moment + nature

- Lieu
 - Qui exécute ? Acteurs (MCC)
- Moment
 - Quand exécute-t-on l'opération ?
 - Agencement temporel
- Nature
 - Manuelle
 - Automatique
 - Interactive

■ Procédures fonctionnelles

Une PF est un ensemble d'actions d'une opération conceptuelle affecté à un poste de travail. Une PF est interruptible.

Il est fréquent d'affecter les actions d'une opération à plusieurs postes de travail. Pour simplifier l'organisation, on découpe chaque opération en procédures fonctionnelles (PF) pour lesquelles on précise le poste de travail, la nature et le déroulement.

Tableau des PF : Processus							
PF	Déroulement		Actions	Nature	Poste de travail		
	Début	Durée (max)			Lieu	Responsable	Ressources
PF1	09H	30min	Editer des propositions de commandes	A	SA	Adjoint Chef SA	PC
PF2	09H30	1H30	Analyser les propositions	M	SA	Chef SA	Liste propositions
PF3	11H	30min	Valider les propositions	I	SA	Chef SA	PC + Adjoint Chef SA
.			.				
.			.				
.			.				

Du MCT au MOT

1. Importer la liste des acteurs du MCC
2. Importer le MCT
 - Répartir les opérations en les acteurs
 - Diviser les traitements répartis entre plusieurs acteurs
 - Définir les périodes de traitement
 - Diviser les traitements répartis sur plusieurs périodes
 - Définir les types de traitements

Application

Confer l'application des MCC et MCT.

TAF

1. Décrire les PF
2. Etablir le MOT

Conclusion :

Même si la méthode MERISE étant, avant tout, une méthode de conception de systèmes d'information, et non de systèmes informatiques, il apparaît aujourd'hui que les systèmes d'information sont largement gérés par des applications informatiques. Les modèles MERISE doivent donc être utilisés pour faciliter le développement de ces applications en s'appuyant sur les technologies logicielles actuelles telles que les bases de données relationnelles et/ou l'architecture client-serveur.