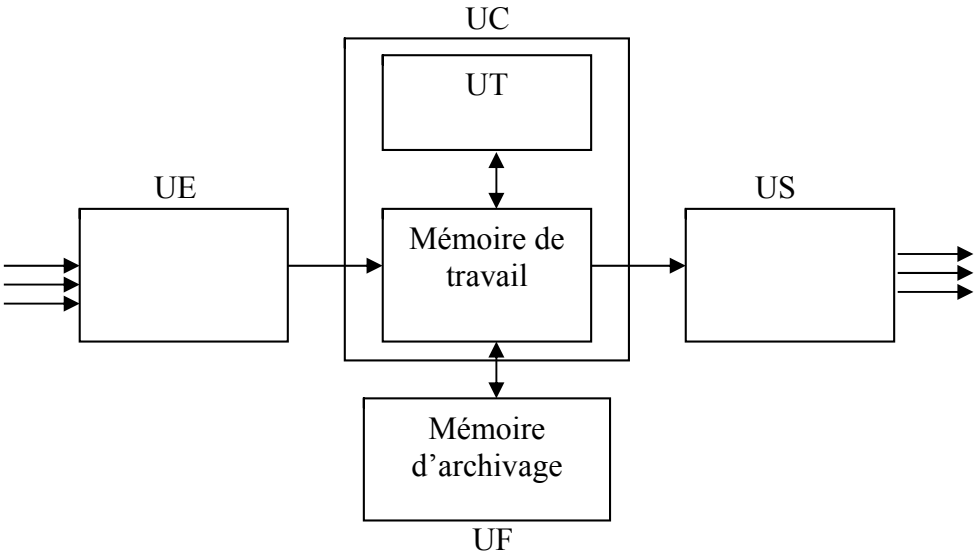
Analyse

Extraits concernant l'analyse des données

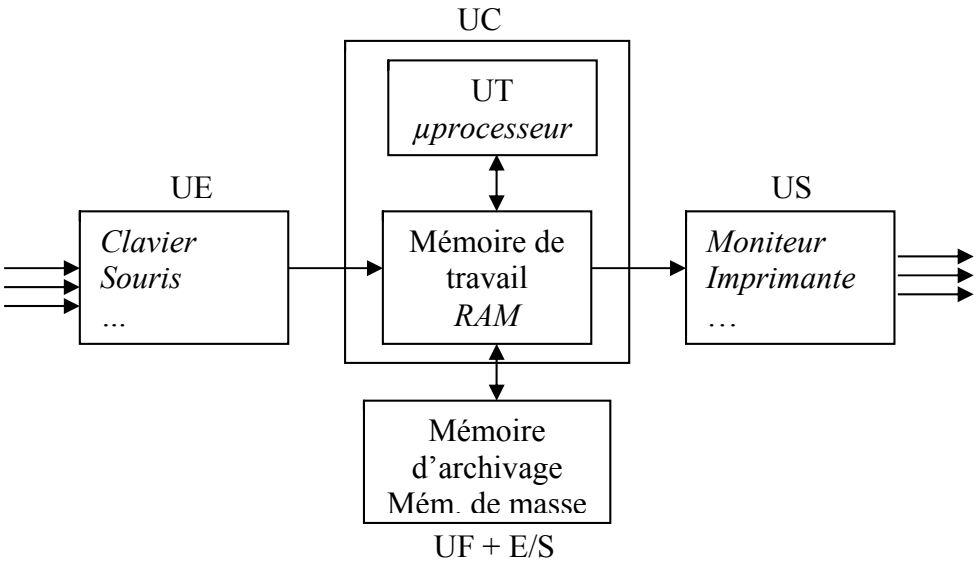
1. Pour commencer

1.1. Le schéma du traitement rationnel de l'information

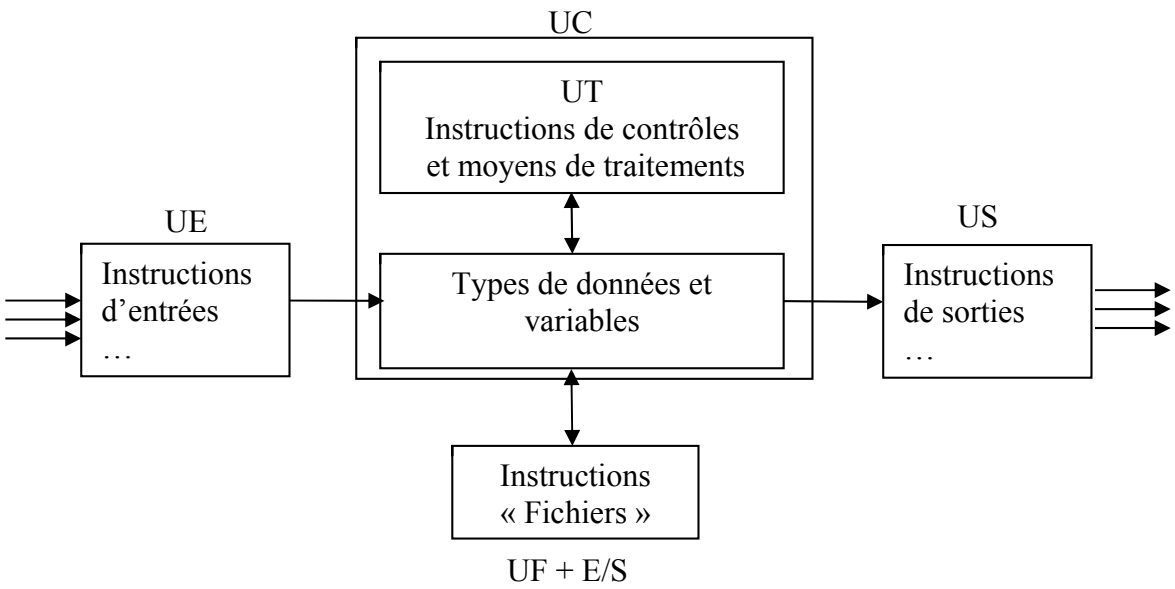
(inspiré de von Neuman, mathématicien, 1903-1957)



1.1.1. Application du schéma au système informatique



1.1.2. Application du schéma aux langages informatiques



1.2. Le questionnement primordial

Les programmes sont idéalement structurés conformément au schéma du cycle de traitement rationnel de l'information. Les résultats issus d'un traitement conforme à ce schéma sont soit des résultats finis (si fin de traitement), soit des résultats destinés à être introduit dans un nouveau traitement également selon le schéma.
En pratique, il est rare de pouvoir schématiser une application comme étant l'effet d'un seul passage dans le système de traitement. Le plus souvent, un traitement complet se représente comme étant l'effet de nombreux passages partiels ou complets dans le système.

L'approche d'un problème informatique se fait en répondant aux trois questions suivantes, au sens le plus large d'abord, et de façon de plus en plus détaillée ensuite (dérivé de la méthode Corig) :

Que faire ? (définition des résultats attendus)
Avec quoi ? (définition des données initiales disponibles)
Comment ? (définition des traitements par lesquels les données vont fournir les résultats)

Exemple :

Que faire ?
Calculer et afficher la surface d'un triangle
Que faire ?
Afficher *La surface du triangle est :* et le résultat du traitement
Avec quoi ?
La chaîne : **La surface du triangle est :**
Le résultat du traitement
Comment ?
Exécuter l'instruction de sortie pour envoyer la chaîne à l'écran
Exécuter l'instruction de sortie pour envoyer le résultat de B*H/2 à l'écran

Avec quoi ?
Les base et hauteur du triangle
Que faire ?
Afficher les questions adéquates pour obtenir les valeurs nécessaires
Avec quoi ?
La chaîne : **Introduisez la base du triangle :**
La chaîne : **Introduisez la hauteur du triangle :**
Comment ?
Exécuter l'instruction de sortie pour envoyer la 1ère chaîne à l'écran
Exécuter l'instruction d'entrée pour stocker dans B la valeur obtenue
Exécuter l'instruction de sortie pour envoyer la 2ème chaîne à l'écran
Exécuter l'instruction d'entrée pour stocker dans H la valeur obtenue

Comment ?
Par l'application de la formule S = B x H / 2
Que faire ?
Calculer B*H/2
Avec quoi ?
Les variables B et H en mémoire
Comment ?
Exécution directe dans l'instruction de sortie

La méthode des 3 questions est fastidieuse et rarement conduite jusqu'au détail de la programmation. Toutefois, l'appliquer au premier niveau de questionnement permet de réduire l'énoncé d'un problème à l'essentiel et de distinguer les sorties, les entrées et les traitements.

Concevoir un programme c'est définir le mode opératoire détaillé de la résolution d'un problème, c'est décrire son algorithme. Concevoir une application, c'est définir l'ensemble de ses fonctions, compte tenu des informations à produire et des informations disponibles.

2. Concepts de base

2.1. Observation d'un document

L'INFORMATIQUE FACILE
Rue Xyz, 14
1300 WAVRE
TVA 415 450 648
RCN 63789
Tél. 010/22.22.24
Fax 010/22.22.26

Client 985073
TVA BE 438,415,650

FACTURE 97/389

Monsieur MONCLIENT P.

Rue du Pigeon, 11

1348 Louvain-la-Neuve

Wavre, le 8/9/1997

DOIT :

Qtés	Libellés	Remise	P.U.	Total
1	Unité centrale P200		29500	29500
4	CD Rom 24 vit.		4200	16800
5	Clavier Mitsumi FR	15%	750	3188
1	HDD 2 Go. WD IDE		9850	9850
2	Moniteur CTX 15"		9745	19490
4	FDD 1,44 Mo.	10%	920	3312
		Total		**82140**
		TVA	**21%**	**17249**
		A PAYER		**99389**

En cas de payement dans les 8 jours, veuillez décompter 2% d'escompte, soit : 1643

Tout ce qui est visible sur un document est information. Les traits, soulignés, et autres surcharges sont des informations au même titre que les textes. Nous les percevons (sans même y prendre attention) et elles nous aident dans la compréhension du contenu par la structuration et les mises en évidences qu'elles induisent.
Pour l'informaticien, un même document porte des informations différentes selon qu'il est examiné du côté de l'émetteur ou du côté du récepteur.
Le document ci-dessus est une facture qui peut selon le cas être considérée comme une facture d'entrée ou comme une facture de sortie.
Les informations pertinentes du document ne seront pas toutes identiques selon le point de vue adopté. De plus, certaines informations n'auront d'intérêts que dans des cas tout à fait exceptionnels. Ainsi par exemple, les traits et éléments de mise en page ne présenteront jamais d'intérêt au niveau de l'analyse. Dans la plupart des cas, des informations écrites telles que le mot client (avant le numéro de client) ou le mot Wavre (avant la date) ne seront pas prises en considération. Ceci ne signifie pas que ces informations auraient pu être omises. Le mot client nous informe sur la signification du numéro 985073 qui le suit, et ce dernier est une information qu'il sera souvent nécessaire de gérer.
De même, si les mots Qtés, Libellés, Remise, P.U. et Total ne doivent pratiquement jamais être gérés, ils nous informent sur la signification des informations écrites en-dessous, et ces dernières seront presque toujours gérées par l'informatique. Ces mots sont des étiquettes nous renseignant sur les types d'informations qu'elles désignent.
Enfin, selon qu'on doive adopter le point de vue de l'émetteur ou celui du récepteur, certaines informations cessent d'être pertinentes alors que d'autres le deviennent. En général, les informations d'en-tête du document (informations souvent pré-imprimées reprenant le signalétique de l'émetteur) ne sont pas pertinentes pour l'émetteur tandis que certaines le sont pour le récepteur. En revanche, les informations concernant le destinataire du document ne sont pas pertinentes pour le récepteur tandis qu'elles le sont pour l'émetteur.
L'observation d'un document doit commencer par le repérage des informations utiles compte tenu du point de vue à adopter (émetteur ou récepteur), et continuer par la distinction des informations 'étiquettes' de celles à gérer.
Ces informations 'étiquettes' seront appelées attributs, propriétés, ou encore 'types d'informations', selon les méthodes et/ou phases d'analyse. Mais peu importe ! Il est seulement important de faire la distinction entre l'étiquette d'un conteneur et son contenu.

2.2. Dictionnaire des données

Le dictionnaire des données se constitue au départ des interviews et des documents reçus. Chaque document fait l'objet d'un recensement des informations qu'il contient (attention au point de vue : émetteur ou récepteur) avec une description de leurs principales caractéristiques (connues ou prévisibles).
Si une propriété sert à des utilisations différentes, il faut considérer qu'il s'agit de propriétés distinctes (ex. : Date de commande et date de facture pour DATE).

L'épuration du dictionnaire se fait par la suppression :
- des synonymes : deux noms différents pour la même propriété;
- des polysèmes : deux propriétés différentes avec le même nom.

Le dictionnaire est avantageusement complété d'informations non présentes, mais reconnues (parfois seulement pressenties) comme indispensables à la suite de l'analyse, telles que des identifiants et le repérage de groupes (et sous groupes) répétitifs (plusieurs valeurs sous les mêmes 'étiquettes', sous les *étiquettes du groupe*). Ces informations ajoutées doivent être présentées comme telles par une typographie différente (ex. italique) ou par une mention en remarque. Enfin, un lexique des abréviations et sigles employés doit compléter le dictionnaire.

Voici pour exemple, un dictionnaire des données de la facture de la page précédente, du point de vue de l'émetteur du document :

Inventaire	Signification	Type	Taille	Nature (si connu)	Type entité (si connu)	Remarque	
NumFS	Numéro de facture de sortie	N	Entier	Auto	Mvt		
NumCli	Numéro du client	N	Entier	Co	Sig		
NomCli	Nom client	A	30	Co	Sig		
AdrCli	Rue et N°	A	35	Co	Sig		
CPCli	Code postal	A	6	Co	Sig		
LocCli	Localité	A	30	Co	Sig		
TvaCli	Numéro TVA	A	12	Co	Sig		
DateFact	Date d'émission de la facture	D	12	Auto	Mvt		
CodArt	*Code article*	N	Entier	Auto	Sig	A ajouter	GR
QteArt	Quantité	N	Entier	Enc	?		
LibArt	Libellé	A	30	Co	Sig		
Rem	Remise accordée	N	Réel	Enc	?		
PU	Prix unitaire	N	Réel	Co	?		
TotLig	Total de la ligne	N	Réel	Calc	?		
TotHTVA	Total hors TVA de la facture	N	Réel	Calc			
TauxTVA	Taux de TVA applicable	N	Réel	Enc ou Co			
TotTVA	Montant de TVA	N	Réel	Calc			
TotTTC	Total TTC de la facture	N	Réel	Calc			
TauxEsc	Taux d'escompte accordé	N	Réel	Enc			
TotEsc	Montant de l'escompte	N	Réel	Calc			

Sigles	Significations
GR	Groupe répétitif
N	Numérique
D	Date
A	Alphanumérique
Auto	Automatique (choisi et imposé par le système)
Co	Conversationnel (choisi par l'utilisateur, parmi une liste par ex.)
Enc	Encodage (nouvelle donnée obligatoirement encodée)
Calc	Calculé (calculé et imposé par le système)
Mvt	Mouvement
Sig	Signalétique

L'exemple précédent est excessif s'il s'agit d'un premier dictionnaire de données. Le plus souvent, dans un dictionnaire de début d'analyse, les désignations des Type et Taille sont moins détaillées et les seules valeurs possibles pour Nature sont : Elémentaire et Calculée. Dans cet exemple, les natures Auto, Co et Enc sont élémentaires.

2.3. Entité

L'entité représente un objet matériel ou immatériel de l'univers extérieur dans le système informatique. Elle est un ensemble de données qui dépendent de l'une d'entre elle appelée identifiant.
Dans une entité correctement conçue, ces données sont en dépendance fonctionnelle élémentaire directe (dfed) avec l'identifiant (*cf.* « Dépendances fonctionnelles »). Ce sont les techniques de normalisation des données qui permettent la définition correcte des entités.

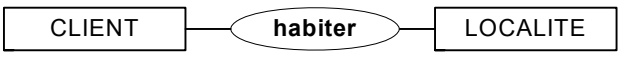
Il est possible de distinguer les entités permanentes et les entités de type mouvements.

Les entités permanentes sont conservées en permanence dans la base d'information et par là elles sont stables, mais peuvent être mises à jour à tout moment. Elles contiennent essentiellement les propriétés signalétiques et celles de situation. Par exemple, une entité Client conserve en permanence les informations des clients. Ses propriétés signalétiques sont notamment NomCli, TVACli, AdrCli, TelCli et certaines peuvent être mises à jour si nécessaire : la propriété AdrCli doit être modifiée si le client déménage. L'entité peut aussi contenir des propriétés de situation, comme un chiffre d'affaire annuel du client qui est mis à jour à l'occasion de chaque opération de vente.

Les entités mouvements sont les images des événements qui ont porté ces mouvements. Il existe un moment où il devient impossible de mettre à jour une entité mouvement. Par exemple, une entité Commande issue d'un événement « commande reçue » ne peut plus être modifiée pour une commande qui a été livrée.

2.4. Relation

Une relation est un lien unissant une ou plusieurs entités.
Comme pour les entités, il faut distinguer les relations permanentes et les relations de type mouvements.
Les relations permanentes sont celles qui lient des entités permanentes, qui sont donc conservées en permanence, et dont on peut modifier les propriétés à tout moment. Elles correspondent à des associations structurelles. Une relation liant une entité Client à une entité Localité est une relation permanente. Le client peut déménager, mais il sera toujours situé dans une localité.



Les relations de type mouvements sont des relations entre entités permanentes et entités mouvements. Elles représentent le souvenir d'un événement. Une relation liant une entité Client à une entité Commande est une relation mouvement qui conserve le souvenir de la passation de chaque commande.

2.5. Cardinalités

2.5.1. Cardinalités des entités

Les cardinalités spécifient le nombre de participations (occurrences) d'une entité (entité-type) à une relation. Elles s'expriment en termes de cardinalité minimale et de cardinalité maximale.
Les cardinalités minimales peuvent être :

0 : occurrence non obligatoire.
1 : occurrence d'entité-type ne peut exister sans participer à une occurrence de la relation.
N : occurrence d'entité-type participe obligatoirement à N occurrences de la relation.

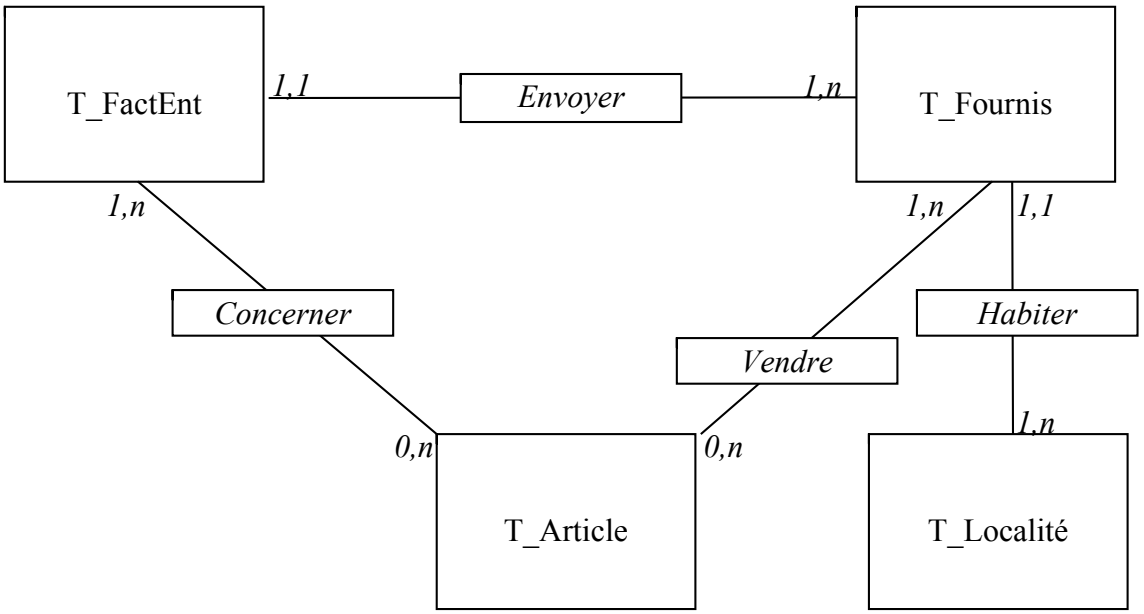
Les cardinalités maximales peuvent être :

1 : occurrence d'entité-type ne peut participer qu'à une occurrence, au plus, de la relation.
N : occurrence d'entité-type peut être impliquée dans un maximum de N occurrences de la relation.

Les cardinalités découlent des règles de gestions qui décrivent avec précision les relations qui lient les différentes entités. Ces règles expriment la façon dont le demandeur de l'application gère (ou souhaite gérer) ses informations. Les règles de gestion d'une facture d'entrée peuvent donc varier d'une entreprise à l'autre.
Exemples de règles de gestion pour des factures d'entrée :

- Un fournisseur émet de une à plusieurs factures.
- Une facture est envoyée par un et un seul fournisseur.
- Un fournisseur vend de un à plusieurs articles.
- Un article est vendu par un ou plusieurs fournisseurs, mais certains ne proviennent d'aucun fournisseur car nous les produisons nous-mêmes.
- Une facture concerne au moins un article, parfois plusieurs.
- Un article peut figurer sur plusieurs factures d'entrée, mais certains ne s'y trouvent jamais car nous les produisons nous-mêmes.

- Un fournisseur habite une et une seule localité.
- Plusieurs fournisseurs peuvent habiter une même localité.

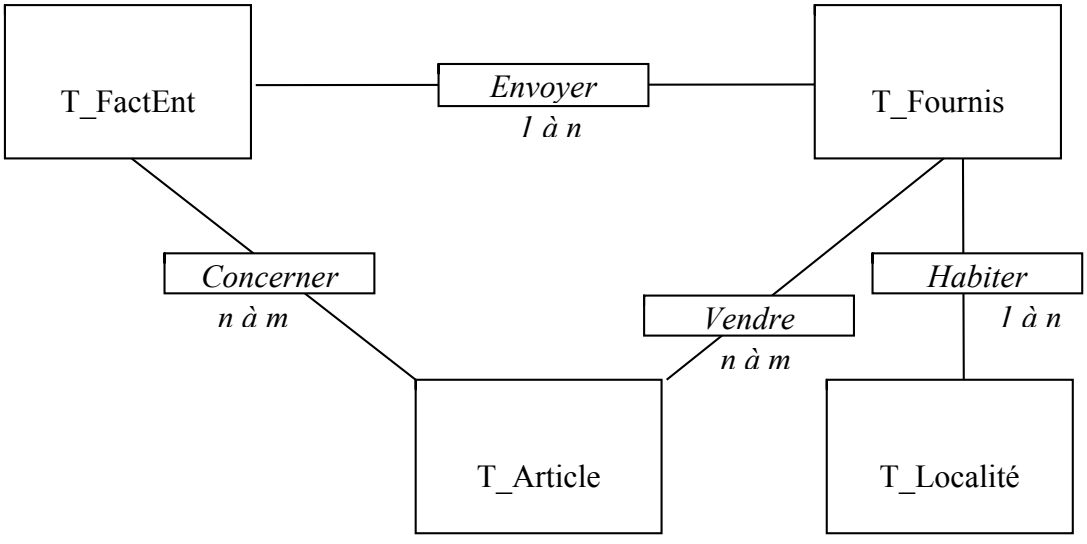


L'observation des cardinalités peut révéler des incohérences dans les règles de gestion exprimées. Par exemple, selon les règles de gestion de factures d'entrées énoncées ci-dessus, il serait incohérent de trouver une cardinalité 1,n pour l'entité T_Article vis-à-vis de l'entité T_Fournis et une cardinalité 0,n pour l'entité T_Article vis-à-vis de l'entité T_FactEnt. Il est en effet peu probable qu'on puisse considérer qu'un article provienne obligatoirement d'un fournisseur et en même temps, admettre qu'un article puisse ne pas être facturé. Si telles étaient les cardinalités dégagées des règles de gestion, il faudrait revoir ces dernières avec le demandeur et les faire corriger ou compléter.

2.5.2. Cardinalités des relations

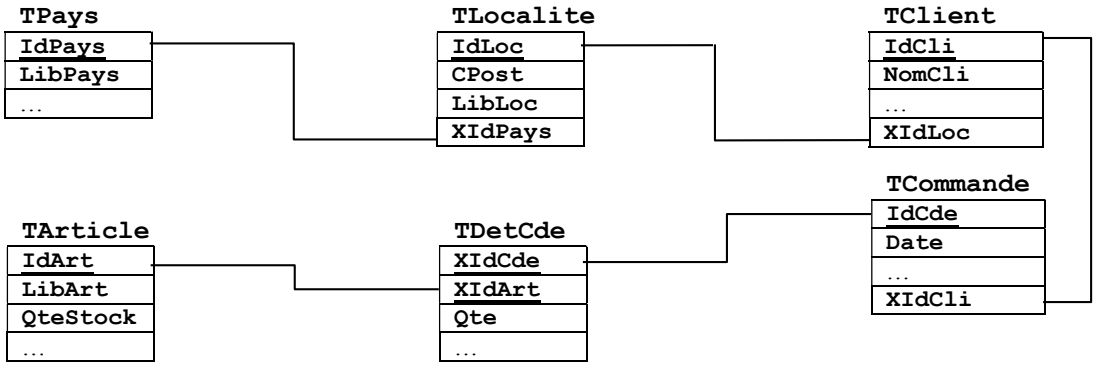
Les cardinalités des relations expriment les cardinalités maximales des entités qui interviennent dans la relation considérée. Elles servent à détecter les relations de type N à M qui ne sont pas gérables et qui dès lors, doivent être transformées ou abandonnées.

Ces cardinalités peuvent faire apparaître aussi des relations 1 à 1. Il faut dans ce cas observer si elles proviennent de cardinalités d'entité 0,1 et 1,1 ou 1,1 et 1,1.
Des cardinalités d'entité 0,1 et 1,1 désignent des informations isolées et facultatives.
Des cardinalités d'entité 1,1 et 1,1 désignent une relation strictement 1 à 1 qui doit être supprimée par la réunion des deux entités en une seule.



Dans le cas de figure illustré ici, les relations Concerner et Vendre ont des cardinalités de type N à M. Ce sont les techniques de normalisation des données qui désignent celle de ces deux relations qui est indispensable à la gestion demandée (l'autre étant abandonnée) et fournissent le moyen de sa transformation.

2.6. Dépendances fonctionnelles



2.6.1. Notion de dépendance

Il existe une dépendance entre deux attributs si la connaissance de l'un permet l'accès à une ou plusieurs propriétés de l'autre (en d'autres termes, il n'existe pas de dépendance entre deux attributs si la connaissance de l'un ne permet pas toujours l'accès à au moins une propriété de l'autre). Attention, que A donne l'accès à B n'implique pas que B donne accès à A (les relations de dépendances ne sont pas réciproques).

Exemples :

Le libellé d'un article permet-il de connaître le code postal d'une localité ?

Il est possible de trouver les localités des clients qui commandent cet article, mais il n'est pas possible de savoir si un des codes postaux associés aux localités trouvées est celui de la localité donnée. De plus, si une localité ne contient aucun client, le libellé de l'article ne permet pas de trouver cette localité, ni son code postal.

➔ Il n'y a pas de dépendance entre le libellé d'un article et le code postal d'une localité.

Le libellé d'une localité permet-il de connaître son pays ?

Il est possible de trouver le pays d'une localité à partir du libellé de cette dernière. Mais si un libellé de localité existe dans plusieurs pays comme Waterloo (3 occurrences aux USA, 2 au Canada, 1 en UK, 1 en Australie, 1 en Sierra Leone, 1 en Belgique), Courcelles (1 occurrence en Belgique, 1 au Canada, 10 en France), Belgrade (1 occurrence en Belgique et capitale de la Serbie), une localité donne accès à plusieurs pays.

➔ Il existe une dépendance entre le libellé d'une localité et des pays.

2.6.2. Dépendance fonctionnelle (df)

Il existe une dépendance fonctionnelle entre deux attributs si la connaissance de l'un permet l'accès à une et une seule propriété de l'autre.

Exemple :

L'identifiant d'une localité permet-il de connaître son pays ?

Il est possible de trouver le pays d'une localité à partir de l'identifiant de cette localité. En outre, l'identifiant d'une localité donne accès à un seul pays, celui de cette localité.

➔ Il existe une dépendance fonctionnelle entre l'identifiant d'une localité et son pays.

2.6.3. Dépendance fonctionnelle élémentaire (dfe)

Il existe une dépendance fonctionnelle élémentaire entre un ou plusieurs attributs et un autre donné si seule la connaissance de tous ces attributs ensemble permet l'accès à une et une seule propriété de l'attribut donné.
Lorsque plusieurs attributs sont considérés ensemble pour déterminer la dépendance fonctionnelle, celle-ci est plus justement qualifiée de « composée », plutôt que de « élémentaire ».

Exemples :

L'identifiant d'un article permet-il de connaître sa quantité commandée ?

Sur base de son identifiant, il est possible de trouver toutes les quantités commandées d'un article, mais pas une en particulier.

➔ Il existe bien une dépendance entre l'identifiant d'un article et les quantités commandées, mais il ne s'agit même pas d'une dépendance fonctionnelle élémentaire, ce n'est même pas une dépendance fonctionnelle.

L'identifiant d'un article et celui d'une commande permettent-ils de connaître la quantité commandée pour cette commande ?

Il est possible de trouver la quantité commandée d'un article sur base de son identifiant, pour une commande donnée également par son identifiant. En outre, les deux identifiants fournissent ensemble une et une seule quantité.

➔ Il existe une dépendance fonctionnelle élémentaire entre les identifiants d'un article et d'une commande et la quantité commandée de cet article sur cette commande.

2.6.4. Dépendance fonctionnelle élémentaire directe (dfed)

Il existe une dépendance fonctionnelle élémentaire directe entre deux attributs si la connaissance de l'un permet l'accès à une et une seule propriété de l'autre sans qu'il soit nécessaire de prendre en compte la valeur d'un troisième attribut.

Exemples :

L'identifiant d'une commande permet-il de connaître le nom du client ?

Il est possible de trouver le nom du client sur base de l'identifiant de la commande car celle-ci contient l'identifiant du client. Il s'agit d'une dépendance fonctionnelle élémentaire. Elle n'est toutefois pas directe puisqu'il faut d'abord prendre en compte la valeur d'un autre attribut de la commande pour accéder au nom du client. A noter qu'entre l'identifiant de la commande et l'identifiant du client, il existe une dépendance fonctionnelle élémentaire directe, mais elle n'est pas la réponse à cette question.

➔ Il existe une dépendance fonctionnelle élémentaire entre l'identifiant d'une commande et le nom du client, mais elle n'est pas directe.

L'identifiant d'un client repris sur une commande permet-il de connaître le nom du client ?

L'identifiant d'un client repris sur une commande donne accès aux informations de ce client et donc aussi à son nom. Il s'agit d'une dépendance fonctionnelle élémentaire. Elle n'est toutefois pas directe puisqu'il faut d'abord rechercher l'identifiant du client dans l'entité qui contient ses informations signalétiques.

➔ Il existe une dépendance fonctionnelle élémentaire entre l'identifiant du client mentionné sur une commande et le nom du client, mais elle n'est pas directe.

L'identifiant d'un article repris sur une liste de stock permet-il de connaître le libellé d'un article ?

L'identifiant d'un article repris sur une liste de stock donne directement accès aux informations de cet article et donc aussi à son libellé.

➔ Il existe une dépendance fonctionnelle élémentaire directe entre l'identifiant d'un article repris sur une liste de stock et le libellé de cet article.

2.7. Formes normales

La normalisation répond, par le choix de certaines associations, à la question qui se pose de savoir qu'elle est la meilleure façon de représenter le monde réel. Elle est un processus qui consiste à remplacer une relation complexe par des séries de relations de plus en plus simples et de structures de plus en plus proches. Les objectifs de la normalisation sont :

- rendre possible la représentation de n'importe quelle relation dans une base de données;
- obtenir de puissants algorithmes de recouvrements basés sur une série d'opérations relationnelles simples;
- réduire les besoins de restructuration des relations si de nouveaux types de données sont ajoutés;
- rendre les relations indifférentes à la nature des requêtes qui peut changer dans le temps;
- prévenir les redondances de données.

2.7.1. Les différents niveaux de normalisation

Les différents niveaux de normalisation correspondent à différents degrés de dépendances entre les données regroupées dans une entité. Ces degrés de dépendances (notés ci-après *df*, *dfe* et *dfed*) sont étudiés plus loin dans ces pages sous le titre « Dépendances fonctionnelles ».

2.7.1.1. Première forme normale (1FN)

L'entité E est en 1FN si toutes les propriétés sont basées sur un domaine simple c.-à-d. si et seulement si, chaque propriété n'a qu'une seule valeur (pas de groupes répétitifs) et s'il y a un identifiant (valeur unique permettant de retrouver toutes les propriétés d'une entité).

2.7.1.2. Deuxième forme normale (2FN)

L'entité E est en 2FN si et seulement si E est en 1FN et si toutes les propriétés non principales de E sont totalement dépendantes de tout l'identifiant de E (l'identifiant doit être indivis c.-à-d. être en dfe avec toutes les propriétés de son entité) et ne peuvent se déduire d'un sous-ensemble de l'identifiant.

2.7.1.3. Troisième forme normale (3FN)

L'entité E est en 3FN si et seulement si E est en 2FN et si aucune propriété non principale de E n'est transitivement dépendante de la clé de E (identifiant doit être en dfed avec toutes les propriétés de son entité) c-à-d. qu'une valeur de propriété n'appartenant pas à la clé ne dépend pas d'une propriété non clé.

2.7.1.4. Forme normale BOYCE-CODD (BCFN)

Une entité E est en BCFN si E est en 3FN et s'il n'existe pas une quelconque propriété en df avec une partie de l'identifiant. Le procédé de mise en forme normale décrit ci-après résout la BCFN lors du traitement de la 2FN.

2.7.1.5. Quatrième forme normale (4FN, approche PHR)

L'entité E est en 4FN si et seulement si E est en BCNF et s'il n'existe aucune propriété isolée facultative, ni de propriété dont le jeu de valeurs possibles est répété dans l'ensemble de l'entité. Cette 4FN doit toujours être justifiée par un gain d'espace mémoire.

2.7.1.6. En résumé

1FN ➔ un identifiant et pas de groupes répétitifs (df) (*cf.* « Dépendances fonctionnelles »)
2FN ➔ 1FN et identifiant indivis (dfe).
3FN ➔ 2FN et pas de transitivité (dfed).
BCFN ➔ être en 3FN et pas de propriété en df avec partie de l'identifiant.
4FN ➔ BCFN et pas de propriété isolée facultative ou dont les valeurs sont répétées, et économie d'octets.

2.7.2. Méthode pratique

L'analyse des données d'une application se fait idéalement support par support en considérant tous les supports d'informations entrantes dans le système et tous les supports d'informations sortantes. Ces supports (ou lots) d'informations ne se présentent pas forcément sous forme de documents en papier, mais tous peuvent être esquissés. Par exemple, un document exclusivement destiné à une consultation par ordinateur peut être dessiné sur une feuille de papier avec ses lignes, ses colonnes, ses libellés et un échantillonnage de données. C'est sur base de cette esquisse que sont normalisées les informations.

A l'issue de la normalisation des données de tous ces supports, les entités (tables ou fichiers) obtenues sont regroupées et certaines ôtées de sorte à ne conserver qu'une seule entité par type d'information complétée des données de ses homologues. Par exemple, l'analyse d'une commande client et celle d'une facture de sortie vont toutes deux produire des entités Clients et Articles : une seule de chaque doit être conservée, mais elle doit contenir toutes les données (sans redondance) des deux entités issues de la normalisation (*cf.* dernier point de la méthode : l'optimalisation).

Attention :

1. *Le texte suivant emploie les mots groupe, ensemble et sous-ensemble. Ces trois mots sont strictement synonymes. L'emploi du mot ensemble a pour seul but d'éviter la répétition lassante du mot groupe. L'emploi du préfixe sous- est destiné à mettre en évidence qu'un groupe donné provient du traitement d'un autre.*
2. *A chaque étape de la normalisation, il est nécessaire de répéter les opérations des étapes précédentes sur chaque ensemble généré et ce, jusqu'à ce qu'il ne soit plus possible de générer un seul sous-ensemble.*

3. *Il est prudent de n'extraire à la fois qu'un seul sous-ensemble, mais aussi large que possible (Ex. En 3ème FN, extraire d'abord les infos signalétiques de l'émetteur d'une facture, et extraire ensuite du sous-ensemble obtenu les informations signalétiques de la localité).*
4. *Ne pas créer d'ensemble dont les données ne seraient que quantitatives (mesures) ou temporelles (dates).*
5. *Nombreux sont les identifiants nécessaires qui n'existent pas dans le dictionnaire des données. Si un identifiant n'existe pas, il faut l'ajouter.*

2.7.2.1. Inventaire des informations

Sur base d'un dictionnaire des données d'un support d'informations, il faut :

- énumérer les attributs (= les types d'information);
- souligner la clef identifiant l'ensemble de ces informations;
- repérer les groupes répétitifs éventuels ainsi que leur clef respective.

2.7.2.2. Mise en 1ère forme normale

Traiter l'inventaire précédent en séparant les groupes répétitifs du reste des attributs de sorte que :

- chaque groupe répétitif devienne un ensemble d'attributs dont la clef est constituée d'une copie de la clef du document source à laquelle est ajoutée la clef du groupe répétitif traité (clef composée ou concaténée);
- le reste des attributs constitue un ensemble dont la clef est celle de l'ensemble des informations du support.

2.7.2.3. Mise en 2ème forme normale

Traiter chacun des ensembles issus des groupes répétitifs en y recherchant des attributs qui ne dépendent pas de l'ensemble de la clef composée, mais seulement d'une partie (la partie qui est la clef du groupe répétitif de départ) :

- ces attributs ne dépendant que d'une partie de la clef composée sont regroupés en sous-ensembles où ils reçoivent comme identifiant, la partie de la clef composée dont ils dépendent;
- une copie de cette clef est conservée dans la clef composée du groupe répétitif où elle est nommée "clef étrangère".

2.7.2.4. Mise en 3ème forme normale

Traiter chacun des sous-ensembles existant en y recherchant des attributs non clefs dépendants de l'un d'entre eux ou interdépendants :

- regrouper ces attributs dans des sous-ensembles, chacun recevant un identifiant;
- une copie de cet identifiant est conservée dans le sous-ensemble d'origine où elle est nommée "clef étrangère".

2.7.2.5. Mise en 4ème forme normale

Traiter chacun des sous-ensembles existant en y recherchant les attributs isolés dont le nombre de valeurs possibles est limité (par exemple, les noms de jours, de mois, de pays, de langue, …) et les attributs isolés dont les valeurs sont facultatives mais gourmandes en espace mémoire lorsqu'elles sont affectées (par exemple, les chaînes de caractères) :

- créer un sous-ensemble, avec identifiant (à créer), pour chacun des attributs isolés dont le nombre de valeurs possibles est relativement limité (par rapport au nombre d'enregistrements prévisible de l'entité), et placer une copie de l'identifiant dans le sous-ensemble d'origine où elle est nommée "clef étrangère" (à n'utiliser que lorsque les valeurs possibles des attributs représentent plus de 2 fois le nombre d'octets de l'identifiant créé);
- créer un sous-ensemble, avec identifiant à créer selon une des deux manières suivantes, pour chacun des attributs isolés dont les valeurs sont facultatives (voir ci-après : Variations en 4èmeFN) :
  1. créer un nouvel identifiant et en placer une copie dans le sous-ensemble d'origine où elle est nommée "clef étrangère";
  2. créer l'identifiant du nouveau sous-ensemble comme étant une copie de la clef du sous-ensemble d'origine.

2.7.2.6. Optimalisation

- vérifier si les ensembles définis existent déjà dans la base de données auquel cas il ne faudra plus les y créer. Par exemple, l'analyse d'une commande client et celle d'une facture de sortie vont toutes deux produire des ensembles Clients et Articles, mais ne contenant pas obligatoirement les mêmes informations (par exemple, il est peu probable que la facture de sortie contienne le numéro de téléphone du client, alors qu'il est sûrement présent sur la commande). Il convient de ne conserver qu'un seul ensemble Clients et un seul ensemble Articles, mais en ayant soin de les compléter de sorte qu'ils contiennent toutes les informations nécessaires tant à la commande qu'à la facture;
- supprimer les informations qui peuvent être recalculées (rapidement) lors de l'édition. Par exemple, si on dispose du prix unitaire et de la quantité, le prix total peut toujours être recalculé et il est donc superflu d'alourdir un fichier ou une table avec cette information;
- conserver les informations dont le temps de calcul serait trop long pour une édition conviviale (le chiffre d'affaire d'un client, par exemple).

2.7.2.7. Illustration de la méthode : Fiche d'inscription d'étudiant

Nom :..... Prénom : .....Date d'inscription : .....
Adresse :..... ....
Code postal : ..... Localité : .....
Date de naissance : ..... Jour de naissance :.....
Profession principale : .....
Langue maternelle : ..... Sexe (M/F) :.....
Cours :
1 : Libellé..... Année.....
2 : Libellé..... Année.....
3 : Libellé..... Année.....
4 : Libellé..... Année.....
5 : Libellé..... Année.....
… … … …
Commentaires de la direction :.....
.....
.....
.....

Extrait du dictionnaire des données :

NrOrdre	Numéro d'ordre	…	Identifiant de l'ensemble
Nom			
…			
Localité			
Cpost			
…			
Cours Libellé Année			Créer IdCours
Commentaire			

Informations complémentaires issues de l'interview :

- Les fiches reçoivent un numéro d'ordre manuscrit dans le coin supérieur droit.
- Les commentaires de la direction sont facultatifs et concernent l'inscription sur le plan administratif et non l'étudiant, ni les cours.

Inventaire	**1ère FN**	**2ème FN**	**3ème FN** (a)	**3ème FN** (b)	**4ème FN**
NrOrdre Nom Prénom Adresse Localité Cpost DateNais JourNais DateInscrip Profession LangueMatern Sexe *Cours* Libellé Année Commentaire	***IdInscription*** Nom Prénom Adresse Localité Cpost DateNais JourNais DateInscrip Profession LangueMatern Sexe Commentaire ***xIdInscription*** ***IdCours*** Libellé Année	***IdInscription*** Nom Prénom Adresse Localité Cpost DateNais JourNais DateInscrip Profession LangueMatern Sexe Commentaire ***xIdInscription*** ***xIdCours*** Année ***IdCours*** Libellé	***IdInscription*** ***xIdEtudiant*** DateInscrip Commentaire ***IdEtudiant*** Nom Prénom Adresse Localité Cpost DateNais JourNais Profession LangueMatern Sexe ***xIdInscription*** ***xIdCours*** Année ***IdCours*** Libellé	***IdInscription*** ***xIdEtudiant*** DateInscrip Commentaire ***IdEtudiant*** Nom Prénom Adresse ***xIdLocalité*** DateNais JourNais Profession LangueMatern Sexe ***IdLocalité*** Localité Cpost ***xIdInscription*** ***xIdCours*** Année ***IdCours*** Libellé	***IdInscription*** ***xIdEtudiant*** DateInscrip ***xIdComment*** ***IdComment*** Commentaire ***IdEtudiant*** Nom Prénom Adresse ***xIdLocalité*** DateNais ***xIdJour*** ***xIdProfession*** ***xIdLangue*** Sexe ***IdJour*** Jour ***IdProfession*** Profession ***IdLangue*** Langue ***IdLocalité*** Localité Cpost ***xIdInscription*** ***xIdCours*** Année ***IdCours*** Libellé

Explications :

La méthode est suivie pas à pas.

Inventaire : Sont reprises toutes les informations visibles de la fiche d'inscription. Le type d'information « Cours » désigne un groupe répétitif, puisqu'il désigne des types d'informations plusieurs fois présents sur le formulaire. La clef *NrOrdre* est repérée.

1ère FN : Le groupe répétitif est séparé du reste des infos et reçoit une clef composée de la clef du document source (*NrOrdre* renommée *IdInscription*) à laquelle est ajoutée celle du groupe répétitif (l'attribut « Cours » devenu la clef *IdCours*).

2ème FN : L'ensemble issu du groupe répétitif est étudié et un attribut (Libellé) dépendant de la clef du groupe répétitif (*IdCours*) sans dépendre de la clef du document source (*IdInscription*) est trouvé. Cet attribut est extrait de son groupe pour en former un nouveau dont la clef est celle du groupe répétitif (*IdCours*). En effet, le libellé d'un cours ne change pas d'une inscription à l'autre, tandis que l'année change selon que l'étudiant s'inscrit en 1ère, 2ème ou 3ème année, et est donc liée à l'inscription.

3ème FN(a) : Chaque ensemble existant est étudié. Il s'agit de trouver des attributs interdépendants non clefs et de les regrouper en sous-ensemble aussi large que possible. Ainsi sont trouvés les attributs signalétiques propres à l'étudiant (Nom, Prénom, ..., Sexe). Sont également trouvés CPost et Localité, mais dans le but de constituer un ensemble aussi large que possible, et compte tenu que ces attributs concernent bien l'étudiant, ils sont inclus dans les informations signalétiques de l'étudiant. L'ensemble de ces attributs interdépendants est donc extrait pour constituer un sous-ensemble dont la clef (créée dans ce cas : *IdEtudiant*) est gardée dans l'ensemble d'origine sous forme de clef étrangère ou externe *xIdEtudiant*.

3ème FN(b) : Si les 1ère et 2ème FN n'ont pu modifier les ensembles définis à leur niveau, une nouvelle normalisation des ensembles issus de la 3ème FN permet de nouveau un traitement de 3ème FN. En effet, les attributs interdépendants CPost et Localité sont trouvés. Ils sont donc extraits selon les règles de la 3ème FN.

4ème FN : (à utiliser dans les cas où les valeurs possibles des attributs représentent plus de 2 fois le nombre d'octets de l'attribut clef)
Tous les ensembles obtenus sont étudiés et les attributs Commentaire, JourNais, Profession, LangueMatern et Sexe sont épinglés. Chacun de ces attributs est considéré seul et ne concerne qu'une seule valeur possible par formulaire d'inscription (si ce n'était le cas pour l'un d'eux, il désignerait un groupe répétitif et aurait donc du être traité aux niveaux des 1ère et 2ème FN).

Commentaire : Attribut dont la valeur est facultative mais constituée de plusieurs lignes de texte lorsqu'elle existe. Il est opportun de l'extraire et de le remplacer dans l'ensemble d'origine par une clef externe. Toutes les inscriptions ne portant pas de commentaires pourront avoir la même clef pointant vers le commentaire « *Sans commentaire* ».
JourNais : La taille du nom des jours varie de 5 à 8 caractères. Il faut donc réserver un espace de 8 caractères pour pouvoir enregistrer n'importe quel nom de jour. Si la clef de l'ensemble regroupant le nom des jours occupe moins de 4 caractères (1 seul peut suffire dans ce cas), il est opportun d'extraire l'attribut JourNais et de constituer l'ensemble Jour. L'ensemble Jour pourra recevoir toutes ses valeurs en une seule session d'encodage.
Profession : La taille du nom des professions varie de quelques caractères jusqu'à la longueur du champ qu'on veut bien lui accorder. Il est courant de réserver 20 à 40 caractères pour enregistrer un nom de profession. La somme des tailles de la clef du groupe Profession et de sa copie dans l'ensemble d'origine sera toujours largement inférieure à celle d'un champ Profession. Il est donc opportun d'extraire cet attribut et de constituer un nouvel ensemble. De plus, ce regroupement pourra servir une éventuelle extension de l'application qui prendrait en compte les multiples professions d'un même étudiant.
LangueMatern : Raisonnement identique au précédent.
Sexe : Raisonnement identique à celui appliqué à JourNais. Mais dans ce cas, les valeurs possibles de l'attribut sont **M** et **F**. Aucun gain de place ne sera dégagé du traitement en 4ème FN de cet attribut. Il est donc convenable de ne pas créer de sous-ensemble Sexe.

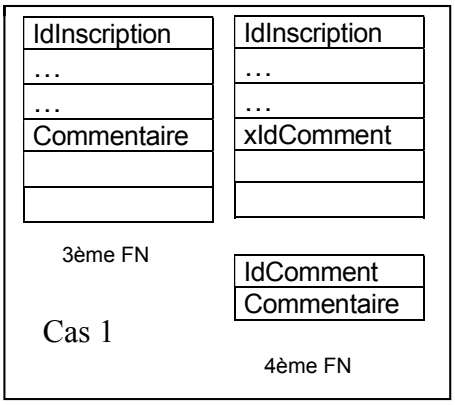
2.7.2.8. Variations en 4ème FN

La méthode de mise en 4FN propose deux manières de résoudre la normalisation des données facultatives. Il convient d'examiner ce qui se passe dans un cas et dans l'autre.

La manière de faire du premier cas est la première proposée par la méthode et aussi celle qui a été employée lors de la mise en 4FN de la fiche d'inscription d'étudiant.

Elle présente l'inconvénient de devoir référencer un commentaire dont le texte est « Sans commentaire », ou « Néant », pour chaque fiche ne comportant aucun commentaire.

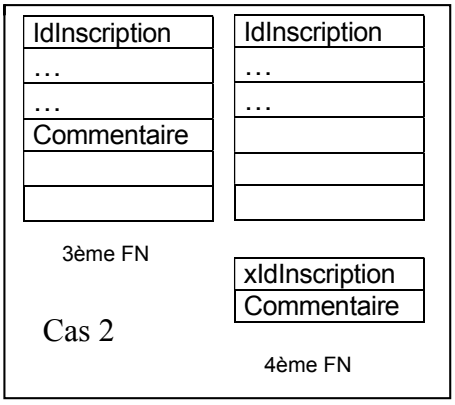
Elle présente l'avantage que s'il faut enregistrer les commentaires d'autres documents (fiche d'embauche de professeur, par exemple), cela peut être fait dans la même entité.



Quant à l'autre manière de faire ….

Elle présente l'avantage de ne pas référencer de commentaire dans l'entité liée et de n'être utile qu'à l'enregistrement qui en a besoin. Il n'y a pas de référence « inutile » à des commentaires tels que « Sans commentaire », ou « Néant ».

Elle présente l'inconvénient de rendre l'entité « Commentaire » inexploitable (sauf acrobaties risquées) par d'autres documents. Par exemple, s'il fallait aussi enregistrer les commentaires des fiches d'embauche de professeur, il faudrait s'assurer que les différents systèmes de fiches comportent des numérotations différentes (ce qui ne peut être garanti) faute de quoi des doublons apparaîtraient parmi les identifiants des commentaires (ce qui ne peut être toléré).



3. Modèle Conceptuel des Données

Le MCD (modèle conceptuel des données) consiste en une description statique des informations du système, indépendante de toute contrainte d'organisation. Il repose sur les notions d'entité et de relations. Il représente la structure du système d'information, du point de vue des données, c'est-à-dire les dépendances ou relations entre les différentes données du système d'information (par exemple : le client, la commande, la ligne de commande).

3.1. Construction du MCD

Deux approches, qui nécessitent la connaissance des données en jeu et de leurs règles de gestions, permettent la construction d'un MCD :

1. le schéma bloc (approche empirique, c'est-à-dire qu'elle procède davantage de l'expérience que de techniques);
2. le graphe des dépendances fonctionnelles (approche technique).

3.1.1. Le schéma bloc

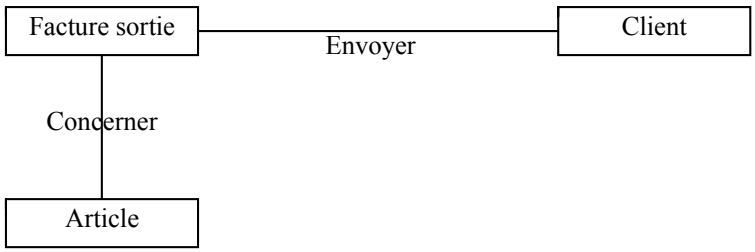
Il faut identifier les lots d'informations évidentes, ou entités naturelles, en repérer les principaux compte tenu de la gestion envisagée, et les représenter avec les relations qui les lient.

Par exemple, la facture de sortie, qui a illustré la constitution d'un dictionnaire des données, ou l'usage de la matrice des dépendances fonctionnelles, est un lot d'informations évidentes : c'est une facture de sortie. Une facture de sortie est destinée à un client suite à une vente d'articles, ou pour des services rendus.
Facture de sortie, Client, Article, …, sont des entités naturelles : il ne faut pas être analyste, ni comptable, ni grand maître es n'importe quoi, pour avoir une idée plutôt exacte de ce que recouvre ces termes et de l'usage qui en est ordinairement fait (les règles de gestion permettront d'en préciser l'usage quand ce sera nécessaire).

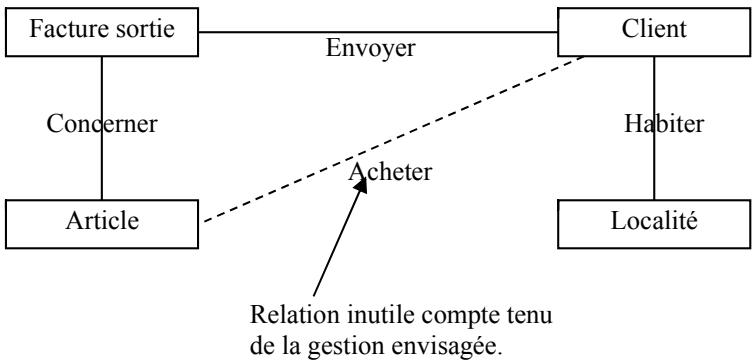
Inventaire des entités naturelles (les principales) : Facture de sortie, Client, Article.
Gestion envisagée : les factures de sortie.
Entité principale : Facture de sortie.
Relations qui lient les entités (au moins les relations qui impliquent l'entité principale) :

- la facture de sortie est envoyée chez le client;
- la facture de sortie concerne des articles.

Représentation graphique :



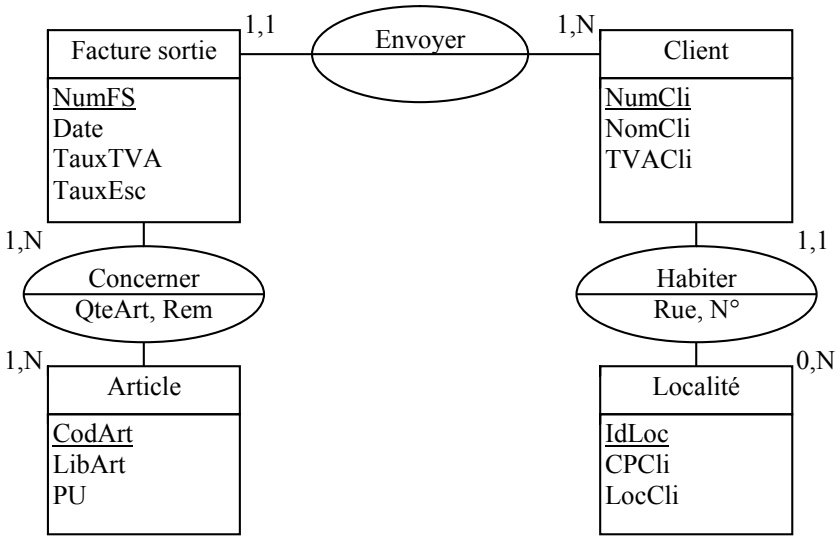
Une approche empirique impose que la solution obtenue soit revue et éventuellement étoffée sur base de l'expérience de l'analyste. Il faut cependant se garder d'en faire trop car des erreurs (rectifiables par la suite, c'est-à-dire travail inutile) s'immisceront immanquablement dans les schémas.
Dans cet exemple, l'expérience conduit vite l'analyste à considérer que le client habite une localité et que cette dernière peut être considérée comme une entité naturelle.
Mais l'analyste qui perdrait de vue que la gestion envisagée est une gestion de facture de sortie et non une gestion de statistiques commerciales, pourrait tracer une relation supplémentaire (et inutile) : le client achète des articles.
Le schéma devient ainsi :



3.1.1.1. Passage du schéma bloc au MCD

La transformation du schéma bloc en MCD s'opère en :

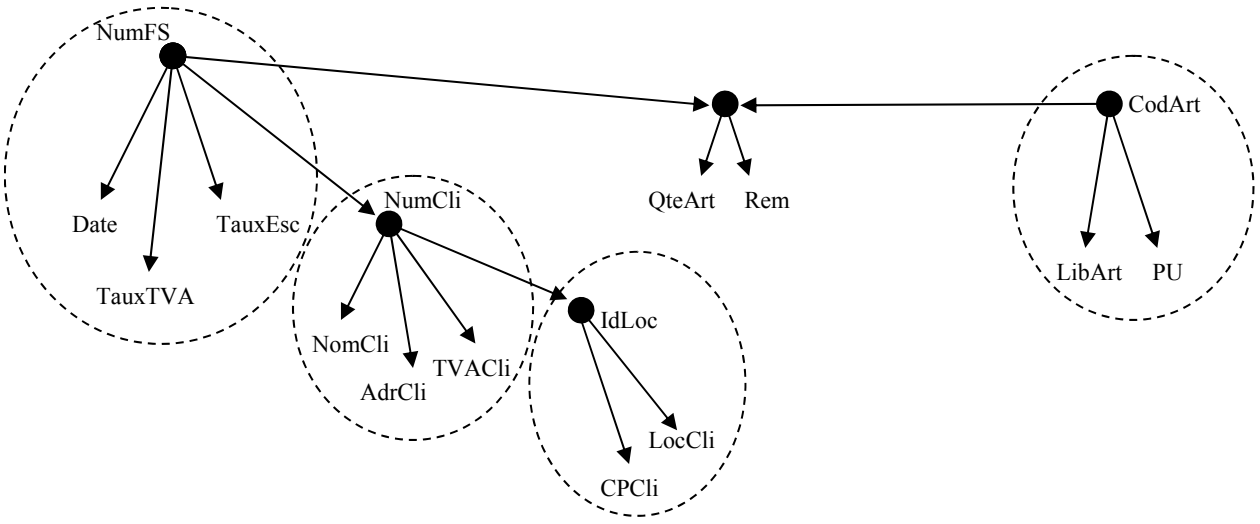
- nommant chaque entité;
- complétant chaque entité de son identifiant souligné;
- complétant chaque entité avec ses informations indépendantes des relations qui les lient;
- complétant les relations de leurs informations distinctives (souvent des informations quantitatives et temporelles, mais d'autres aussi : la rue et le numéro distinguent un client d'un autre dans la même localité);
- écrivant des cardinalités des entités conformément aux règles de gestion.



3.1.2. Le graphe des dépendances fonctionnelles

Le graphe des dépendances fonctionnelles doit être construit comme indiqué précédemment (par l'usage de la matrice, sur base du dictionnaire des données) et éventuellement complété (sur base de l'expérience de l'analyste) pour atteindre un niveau de normalisation supérieur.

Par exemple, le graphe des dépendances fonctionnelles établi pour la gestion de la facture de sortie peut être complété pour séparer les attributs CPCli et LocCli et mettre ainsi l'entité Client en 3ème FN.



3.1.2.1. Passage du SAT au MCD

La transformation du SAT en MCD s'opère comme suit :

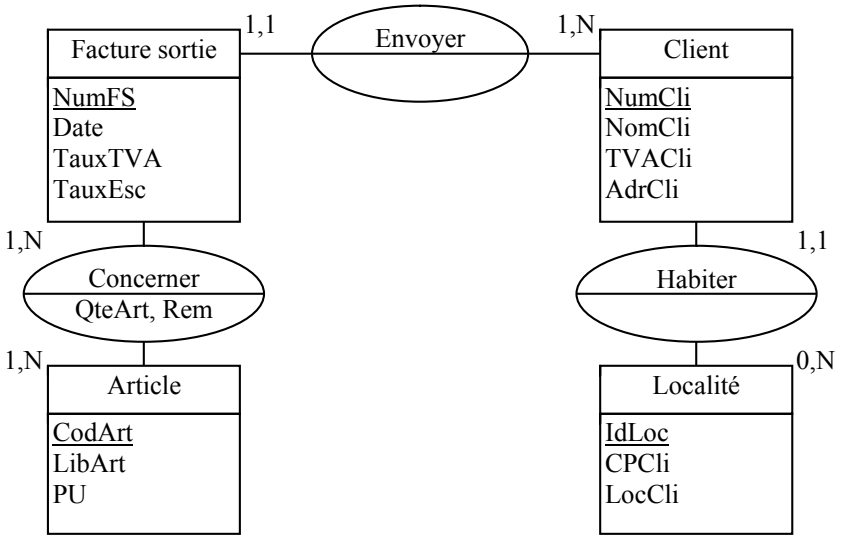
- chaque point d'entrée et racine de sous-arbre fournit l'identifiant d'une entité (et donc aussi l'entité);
- chaque point terminal qui n'est pas une racine de sous-arbre est une donnée de l'entité;
- les données en dépendance composée sont les informations distinctives de la relation à laquelle elles appartiennent.

Comme le suggèrent les ovales en traits interrompus sur le graphe précédent, les entités peuvent aisément être définies graphiquement sur le SAT.

Soit, en poursuivant l'exemple en cours :

- Points d'entrées, racines et points terminaux non racine :
  - NumFs désigne l'entité Facture de sortie, identifiant : NumFs;
    ➔ points terminaux non racine : Date, TauxTVA, TauxEsc.
  - NumCli désigne l'entité Client, identifiant : NumCli;
    ➔ points terminaux non racine : NomCli, TVACli, AdrCli.
  - IdLoc désigne l'entité Localité, identifiant : IdLoc;
    ➔ points terminaux non racine : CPCli, LocCli.
  - CodArt désigne l'entité Article, identifiant : CodArt;
    ➔ points terminaux non racine : LibArt, PU.
- Les données en dépendance composée QteArt et Rem, dépendantes de NumFs et de CodArt (les identifiants de Facture de sortie et Article), sont les données distinctives de la relation existant entre Facture de sortie et Article.

Une telle description est bien celle du MCD à dessiner :



Les deux approches fournissent des MCD légèrement différents. En effet, alors que l'approche par schéma bloc a permis de trouver des données distinctives (par expérience) de la relation Habiter, ça n'a pas été possible avec l'approche par le graphe des dépendances fonctionnelles. C'est qu'on ne s'est pas donné la peine de poser toutes les bonnes questions lors de la réalisation de la matrice des dépendances fonctionnelles.

Y a-t-il une erreur ? Non, il n'y a pas d'erreur. Chacun de ces MCD conduit bien à la même normalisation des données. Chacun de ces MCD est la base d'un même modèle logique des données (MLD). Il faut d'ailleurs noter que, dans la mesure où AdrCli et Rue + N° désignent la même information, chacun de ces MCD reprend bien les mêmes données. Simplement, elles sont présentées un peu différemment.

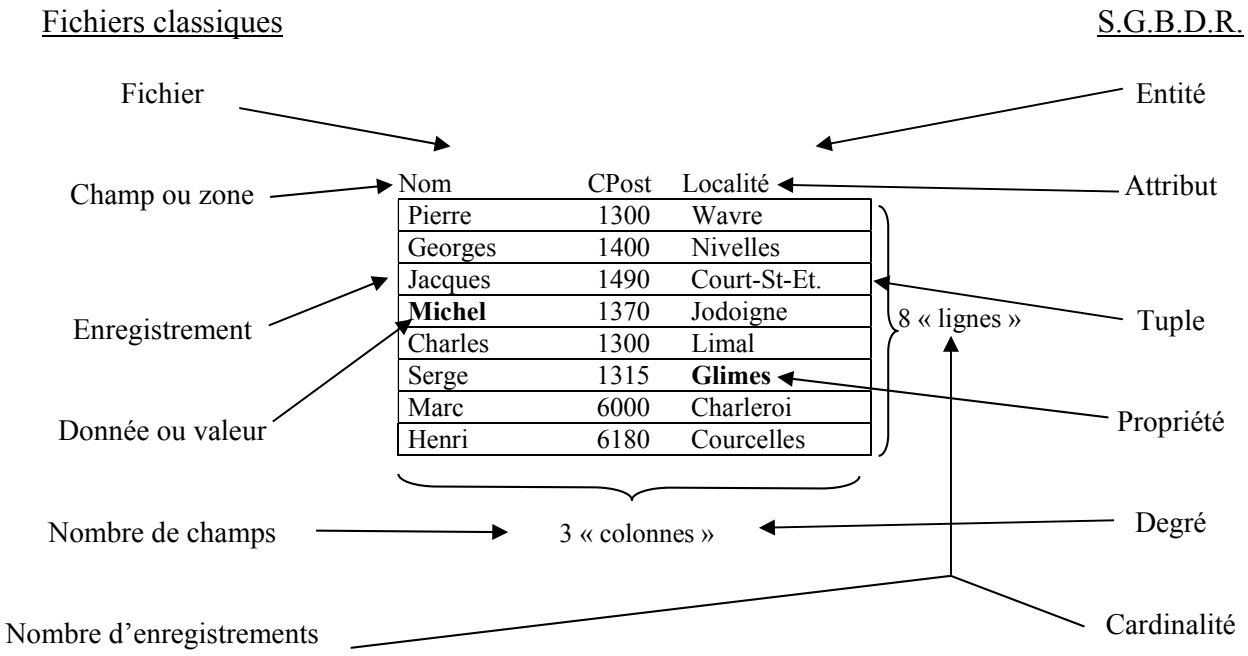
4. Modèle Logique des Données

Le MLD (modèle logique des données) est la traduction d'un modèle conceptuel *validé* (cf. plus loin : Lectures complémentaires) dans une représentation assurant la persistance (conservation) de l'information.
Le MCD peut être traduit en différents systèmes logiques, de la gestion de fichiers dits « à plats » aux systèmes de bases de données orientées objet, sans oublier les SGBDR (Système de gestion de bases de données relationnelles) auquel s'intéresse ce cours.

4.1. A propos des systèmes logiques

Avant l'apparition des systèmes de gestion de base de données (SGBD ou DBMS pour Data Base Management System en anglais), les données étaient stockées dans des fichiers binaires et gérées par des programmes exécutables (Basic, Cobol, Pascal ou Dbase par exemple). La maintenance des programmes (en cas de modification de la structure des données par exemple) était très problématique.
Sont alors apparus les SGBD hiérarchiques dans lesquels les données sont organisées en arbre (IMSDL1 d'IBM par exemple), puis les SGBD réseaux (Codasyl) dans lesquels les données sont organisées selon un graphe plus général (IDS2 de Bull par exemple). Ces deux types de SGBD sont dits navigationnels car on peut retrouver l'information à condition d'en connaître le chemin d'accès.
Aujourd'hui, ils sont largement remplacés par les SGBD relationnels (SGBDR) avec lesquels l'information peut être obtenue par une requête formulée dans un langage quasiment naturel. Ce sont les SGBD les plus répandus (Oracle et DB2 d'IBM par exemple).
Plus récemment, sont apparus des SGBD orientés objets qui offrent à la fois un langage de requête et une navigation hypertexte. Un modèle logique de données orienté objet permet par exemple de concevoir des classes Java s'appuyant sur une base de données relationnelle et permettant le développement d'une application web.

4.1.1. Le vocabulaire des SGBD comparativement à celui des fichiers



4.2. Construction du MLD

Plusieurs procédés conduisent à l'établissement d'un MLD.
1. dans la méthode Merise, c'est la traduction du MCD par l'application de 5 règles qui produit le MLD;
2. la méthode « maître-esclave » est une variante originale de la précédente;
3. le graphe des dépendances fonctionnelles permet la définition visuelle (graphique) du MLD (tout comme il permet la définition des entités du MCD);
4. la méthode pratique de mise en 4FN expliquée précédemment, produit directement un MLD.

4.2.1. Remarques préalables

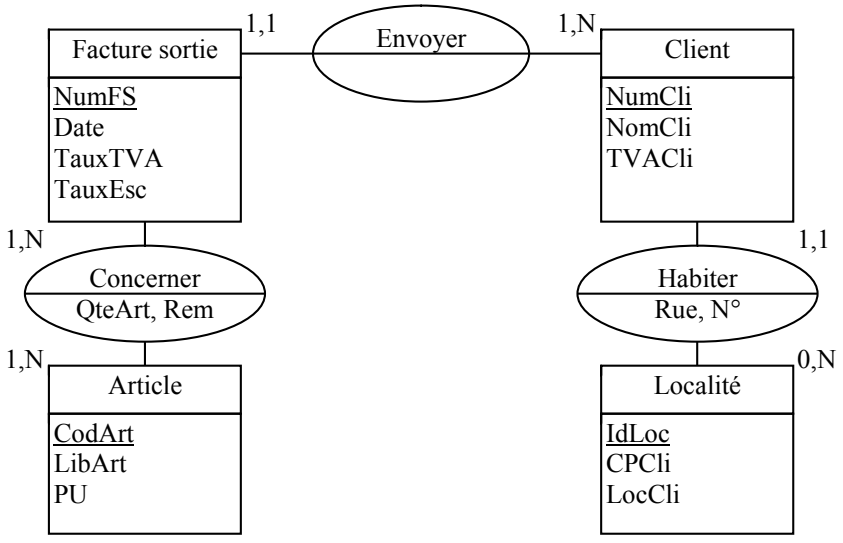
- les clés primaires doivent être soulignées;
- les clés étrangères sont marquées d'un dièse (#), d'un astérisque (*) ou d'un X;
- une table ne peut avoir qu'une seule clé primaire, mais celle-ci peut être une clé composée (concaténation de plusieurs);
- une table peut avoir plusieurs clés étrangères;
- une clé étrangère peut aussi être primaire;
- une clé étrangère peut être composée.

4.2.2. Traduction d'un MCD en MLD

Pour traduire un MCD en troisième forme normale en un MLD, il suffit d'appliquer les cinq règles suivantes :

1. toute entité devient une table dans laquelle les attributs deviennent des champs (colonnes). L'identifiant de l'entité constitue la clé primaire de la table;
2. dans le cas de deux entités reliées par une relation de type 1 à n, l'identifiant de l'entité côté 0,n ou 1,n (maître) devient une clé étrangère dans la table côté 0,1 ou 1,1 (esclave). Les attributs de la relation glissent vers la table côté 0,1 ou 1,1;
3. une relation entre deux entités et de type n à m est traduite par une table supplémentaire (esclave) dont la clé primaire est composée de deux clés étrangères vers les clés primaires des deux tables (maîtres) en relation. Les attributs de la relation deviennent des champs de cette table;
4. une relation entre trois entités ou plus est traduite par une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères que d'entités. Les attributs de la relation deviennent des champs de cette table;
5. dans le cas (particulier) de deux entités reliées par une relation de type 1 à 1, il faut observer si cette relation provient de cardinalités d'entité 0,1 et 1,1 ou 1,1 et 1,1 :
   - des cardinalités d'entité 0,1 et 1,1 désignent des informations isolées et facultatives, à traiter comme décrit dans « Variation en 4ème FN »;
   - des cardinalités d'entité 1,1 et 1,1 désignent une relation strictement 1 à 1 qui doit être supprimée par la réunion des deux entités en une seule table.

Exemples sur base du MCD de la facture de sortie :



Par l'application de la règle 1 :

L'entité Localité devient la table TLocalite et son identifiant devient la clé primaire :

➔ TLocalite(IdLoc, CPCli, LocCli)

L'entité Client devient la table TClient et son identifiant devient la clé primaire :

➔ TClient(NumCli, NomCli, TVACli)

Par l'application de la règle 2 :

La table TClient reçoit une copie la clé de TLocalite, ainsi que les attributs de la relation Habiter :

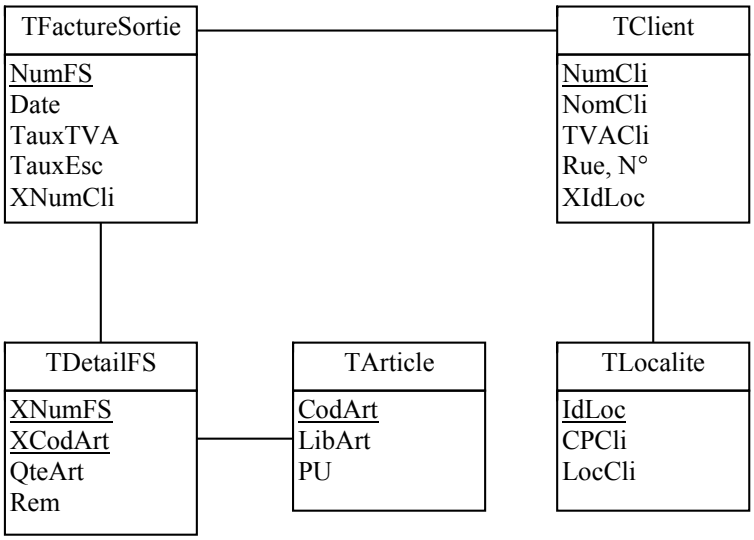
➔ TClient(NumCli, NomCli, TVACli, Rue, N°, XIdLoc)

Par l'application de la règle 3 :

Une table TDetailFS est créée et sa clé primaire est composée des copies des clés des tables TFactureSortie et TArticle liées par la relation Concerne. Les attributs de cette relation devient des champs de cette nouvelle table.

➔ TDetailFS(XNumFS, XCodArt, QteArt, Rem)

4.2.2.1. Représentation graphique du MLD



4.2.3. Méthode « Maître – Esclave »

Avertissement : l'extrême cruauté des propos suivants ne peut être reprochée à l'auteur car il n'existe pas d'autre manière pour relater les pratiques décrites.

Il est bien connu que dans l'antiquité, et plus récemment encore, des personnages nantis avaient le pouvoir d'asservir des humains pour en faire des bêtes de somme, des esclaves. Ces esclaves étaient plus ou moins bien traités, de la même manière que l'était le bétail en général. C'est ainsi notamment, qu'à l'instar des chevaux et des bovins, les esclaves étaient marqués (le plus souvent au fer rouge) du signe de la maison de leur maître. Il arrivait aussi que plusieurs maîtres s'achètent des esclaves en copropriété. Ces esclaves étaient marqués du signe de chacun des maîtres !

Il faut maintenant voir le MCD comme étant le schéma indiquant qui fait travailler qui, qui est maître de qui. L'énoncé des 5 règles suggère bien les relations entre maîtres et esclaves. La méthode consiste presque exclusivement à marquer les esclaves.

Il faut savoir qu'un maître peut posséder de nombreux esclaves et qu'un esclave ne possède qu'un seul maître (à l'exception des cas de copropriétés, qui sont traités dans un second temps). Dès lors, le maître est l'entité dont la cardinalité est N et l'esclave est l'entité dont la cardinalité est 1. Dans les relations 0,1 à 1,1, l'esclave est du côté 1,1.

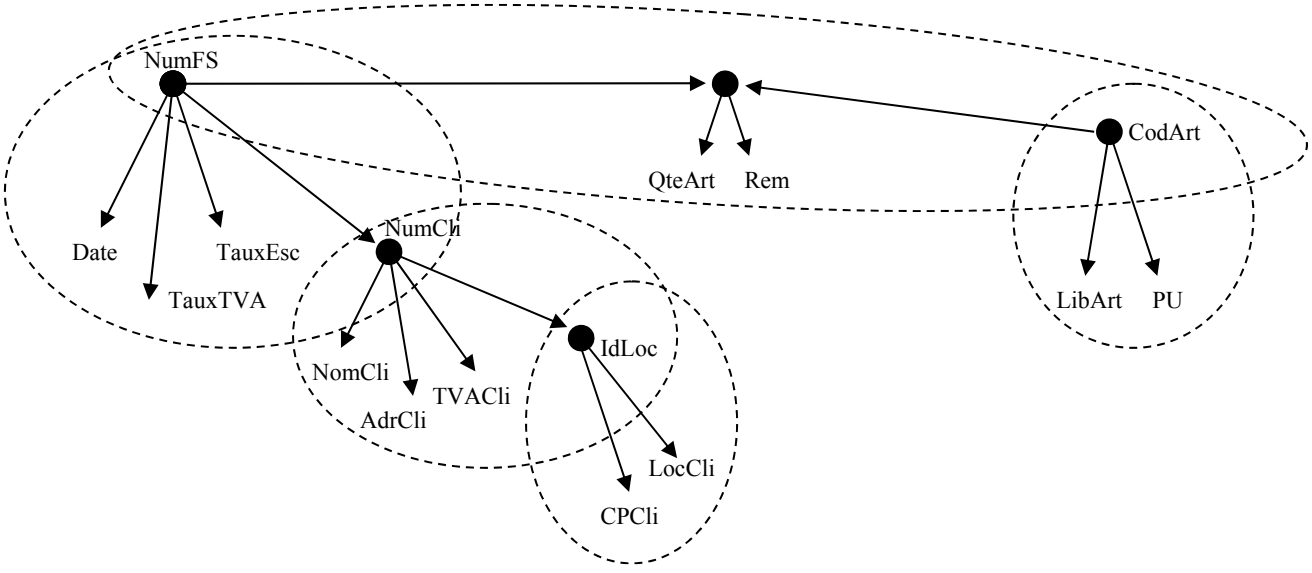
- Le marquage de l'esclave se fait avec l'identifiant du maître.
- S'il y a des attributs dans la relation, ils suivent le marquage.

Quand toutes les relations N à 1 ont été traitées, il peut subsister des relations N à M. C'est le cas où plusieurs maîtres s'approprient un esclave en copropriété.

- La relation devient un esclave.
- Le marquage de l'esclave se fait avec les identifiants de chaque maître.
- S'il y a des attributs dans la relation, ils deviennent des champs de la nouvelle table.

4.2.4. Définir un MLD à partir d'un SAT

La méthode visuelle de définition du MLD à partir d'un graphe des dépendances fonctionnelles est extrêmement semblable à celle qui permet la définition d'un MCD.



La transformation du SAT en MLD s'opère comme suit :

- chaque point d'entrée et racine de sous-arbre fournit la clé primaire d'une table (et donc aussi la table);
- chaque point terminal (y compris les racines de sous-arbre) constituent les champs de la table;
- chaque point terminal qui est racine d'un sous-arbre devient la clé de la table représentée par ce sous-arbre et aussi une clé étrangère dans sa table d'origine;
- les données en dépendance composée deviennent des tables dont les clés primaires sont composées des copies des clés des tables liées.

Comme le suggèrent les ovales en traits interrompus sur le graphe précédent, les entités peuvent aisément être définies graphiquement.