



Merise I

MODELE DE DONNEES CONCEPTUEL LOGIQUE PHYSIQUE

(document d'autoformation)

MERISE

I PRELIMINAIRES AU MODELE DE DONNEES

Elaboration d'un modèle représentatif pour la structuration de données d'un système d'informations.

On cherche à trouver un modèle représentatif des données que nous fournit un univers d'information, pour lequel on mettra en place ultérieurement des traitements automatiques.

Le domaine d'étude concerne la gestion du suivi des propriétaires successifs de véhicules.

Recherche d'une méthode et d'un modèle de représentation

Première approche : repérage des entités

On peut considérer que l'on dispose a priori de fiches:

VEHICULES renseignés par marque, type, couleur ;

PERSONNES renseignées par nom, prénom, adresse.

Deuxième approche : repérage des relations :

Pour lier un véhicule à une personne il est nécessaire de créer une fiche (appelée relation):

PROPRIETE renseignée par date d'achat, lieu d'achat.

Troisième approche : identification des entités

Pour assurer l'unicité de chaque entité, on leur affectera un numéro unique (appelé identifiant).

Ainsi on ajoutera aux entités :

VEHICULE le numéro de série (symbole noser)

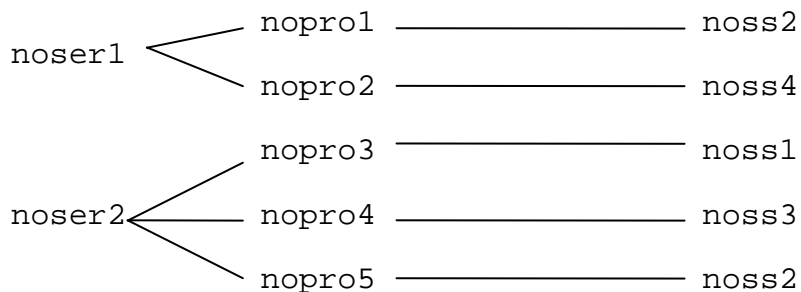
PERSONNE le numéro de sécurité sociale (symbole noss)

Quatrième approche : mode de représentation graphique de la structure de données

1 - Tableau extensif

VEHICULE	PERSONNE
noser1	noss1
noser2	noss2
noser3	noss3
noser4	noss4
noser5	noss5
"	"
"	"
"	"

2 - Schéma arborescent



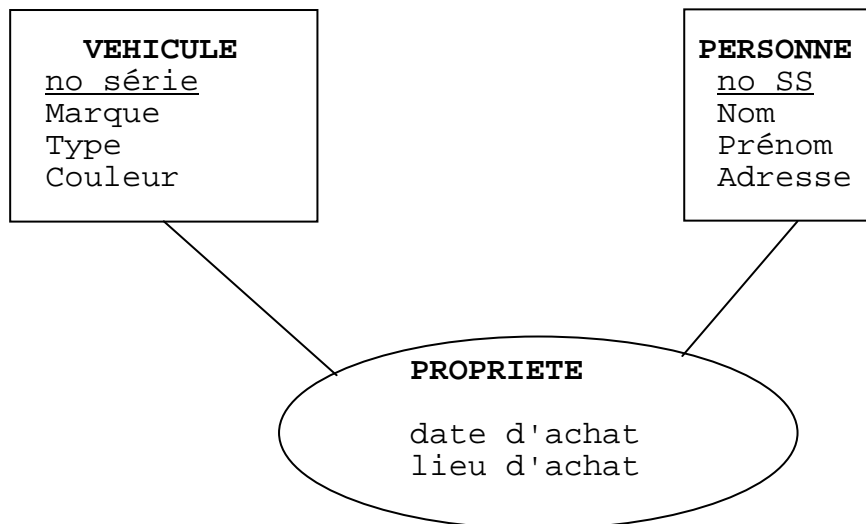
3 - Schéma conceptuel et schéma relationnel

Les schémas précédents présentent l'inconvénient de lister exhaustivement tous les identifiants et de décrire toutes les relations possibles (définition en extension).

Par ailleurs, elles ne décrivent que la relation dans un sens (de véhicule vers propriétaire) : il faudrait élaborer les schémas réciproques pour établir les différentes propriétés d'un véhicule pour chacune des personnes.

Ces modèles ne sont pas assez synthétiques pour donner une idée précise de l'organisation des données.

On préfère donc le schéma conceptuel suivant, suffisant pour représenter la structure, indépendamment des contenus (c'est-à-dire des valeurs que peuvent prendre les identifiants).



Ce modèle permet de repérer sans ambiguïté :

- les entités objets (ex: VEHICULES)
- les entités relation (ex: PROPRIETE)
- les identifiants (ex: no-série) soulignés
- les rubriques de renseignements (ex: Marque, Type, Couleur)
- les liaisons entre entités, par des traits.

On devra noter que les normes Merise ne sont pas tout à fait respectées (l'objectif de cette introduction étant simplement de voir où l'on « veut en venir »).

Conclusion

La représentation graphique ci-dessus correspond à un *schéma conceptuel* des données du système d'information.

Il permet de spécifier de façon précise et visuellement simple la structure des données du système étudié. Il est utilisable par un analyste et peut être commenté, critiqué, modifié avec les utilisateurs.

Une autre expression peut être donnée par le *schéma relationnel* suivant :

VEHICULE (<u>no-série</u> , Marque, Type, Couleur)
PERSONNE (<u>no-SS</u> , Nom, Prénom, Adresse)
PROPRIETE(Dateachat, Lieuachat)

Il permet de définir la structure des données en vue de sa mise en place sur un matériel informatique disposant d'un SGBD (Système de Gestion de Base de Données). Il est utilisable par un informaticien ou programmeur tourné vers le dispositif (matériel et logiciel) de traitement informatique.

Exercice de synthèse 0

I/ Etant donné des fiches :

FILMS renseignés avec titre, date de sortie, durée.

ACTEURS renseignés avec nom, prénom, nationalité, âge.

Concevoir un tableau extensif, un schéma arborescent, un schéma conceptuel représentant la structure des données, après avoir créé l'entité relationnelle manquante (appelée **ROLE**) permettant de lier les acteurs aux films dans lesquels ils ont joué (un acteur ne joue qu'un seul rôle dans un film).

II/ On ajoute les fiches :

PRODUCTEURS renseignés avec nom, raison sociale.

Etablir le schéma conceptuel complet, sachant qu'un producteur finance plusieurs films et qu'un film peut être financé par plusieurs producteurs. Pour chaque film, on connaît le montant de financement d'un producteur qui y participe.

III/ Définir le schéma relationnel.

MERISE

II MODELE CONCEPTUEL DE DONNEES CONSTRUCTION

Le système d'information est connu à travers ce qu'il est convenu d'appeler **l'univers du discours** appelé aussi **base de connaissance**.

En fait il s'agit des informations recueillies sur le système dans le cahier des charges (par exemple) suite à :

- interview des acteurs du système, repris et rédigé précisément
- une collection de documents actuels ou futurs (entrant et sortant des postes de travail)

Exemple :

On dispose du cahier des charges d'une entreprise de distribution qui fait de la vente sur stock à partir des commandes effectuées par les clients.

On va considérer le domaine d'étude "suivi des commandes clients" qui est un "sous univers du discours" (ou partie de la base de connaissance) de cette entreprise.

Le rédactionnel de ce domaine nous fait savoir que :

L'entreprise reçoit les commandes par téléphone ou par courrier. Dans tous les cas, elles doivent préciser quel est le client payeur (en général le siège de la société) et le client destinataire de la livraison (lorsque la société dispose de plusieurs dépôts), avec pour chacun son numéro, son nom, son adresse.

Pour chaque produit commandé, on doit connaître le code et le nom de la famille à laquelle il appartient ainsi que son numéro, son libellé et la quantité demandée.

Le réceptionnaire de la commande notifie la date d'arrivée de la commande et lui affecte un numéro d'ordre (incrémenté de 1 à chaque fois).

Par ailleurs, on dispose d'une maquette des commandes à livrer dont le format est donné dans la page suivante. Cette maquette fait partie de la collection des documents recueillis (dans l'existant) ou élaborés (dans l'expression des besoins) lors du cahier des charges.

```

COMMANDE A LIVRER

      Client destinataire                                client payeur

No Client : -----
Nom       : -----
Adresse   : -----
           -----
-----
NO COMMANDE : -----  DATE COMMANDE : -----

```

client payeur

No Client : -----

Nom : - - - - -

Adresse : -----

NO COMMANDE : ----- DATE COMMANDE : -----

CO FAMILLE	NOM FAMILLE	NO PRODUIT	LIBELLE	QTE COMMANDEE

La méthode MERISE se propose :

De donner une **représentation des données** de cette base de connaissance (partielle dans le cas de notre exemple) qui soit indépendante des traitements mis en jeu pour l'obtention de ces commandes.

Pour ce faire :

Etape 1 : On établira un **dictionnaire des données** utiles à la gestion de ces commandes.

Si on ne dispose que du rédactionnel, on soulignera chaque mot du vocabulaire utilisé dans la gestion des commandes et on en dressera la liste.

Si on ne dispose que d'une maquette ou d'un document, des rubriques qui donne le même résultat.

Si on dispose des deux, c'est encore mieux du fait que le rédactionnel complète la maquette en terme de meilleure compréhension du système et permet de découvrir des incohérences, s'il y en a. D'autre part, on peut obtenir dans ce cas un dictionnaire plus riche et donc plus proche de la réalité. Par exemple, on pourrait imaginer que la maquette fasse figurer la date de livraison souhaitée, alors que le rédactionnel n'en parle pas, ce qui permet de contrôler des omissions. On remarque d'ailleurs que la notion de client destinataire et payeur a été explicitée dans le rédactionnel.

Ici, la constitution du dictionnaire nous a amené à la liste suivante :

- 1 numéro client destinataire
- 2 nom client destinataire
- 3 adresse client destinataire
- 4 numéro client payeur
- 5 nom client payeur
- 6 adresse client payeur
- 7 numéro de la commande
- 8 date de la commande
- 9 code famille
- 10 nom famille
- 11 numéro produit
- 12 libellé produit
- 13 quantité commandée.

Etape 2 : On établit un regroupement des mots qui ont trait, de façon invariante, à une même entité. Ces entités seront appelées des **objets**.

On constate ici que les données :

1,2,3 sont associées à un objet CLIENT DESTINATAIRE,
4,5,6 à un objet CLIENT PAYEUR,
7,8 à un objet COMMANDE,
9,10 à un objet FAMILLE,
11,12 à un objet PRODUIT.

Ces données sont des **propriétés** des objets auxquels elles se rapportent.

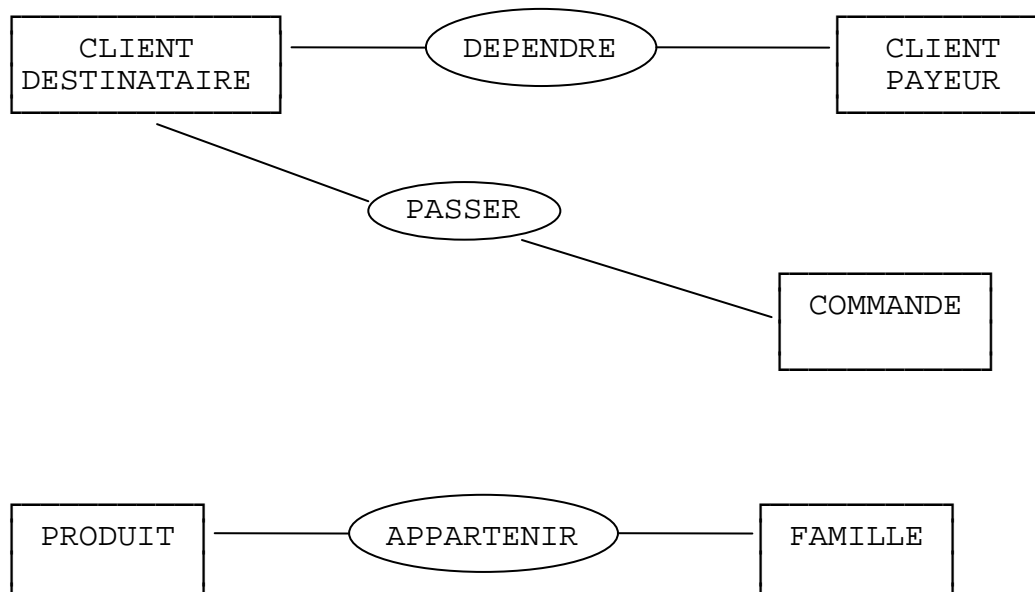
Mais qu'en est-il de la donnée 13 : quantité commandée? Elle n'est pas liée invariablement à l'objet produit. Nous traiterons son cas à l'étape 4/.

Etape 3 : On va décrire maintenant les rapports qui existent entre certains objets. On les appellera des **relations** entre objets. On les déduit de la **sémantique** suivante :

A client destinataire dépend de client payeur,
B client destinataire passé des commandes,
C produit appartient à famille.

la relation A sera nommée DEPENDRE
B PASSER
C APPARTENIR

Enfin on formalisera la structure des données (objets et relations) par le schéma suivant :



Le modèle n'est pas terminé (il manque des relations) mais en attendant de les mettre en place à l'étape 4, on vous propose deux exercices intermédiaires.

Exercice intermédiaire 1 (étape 1-2-3)

Une société de vente d'équipements électroménagers fait effectuer ses livraisons par des installateurs.

Tous les matins, on délivre à chaque installateur une liasse de bons de livraison correspondant aux clients qui lui incombent (c.à.d. relevant de son secteur).

Après installation chez le client, le bon de livraison doit être signé :

No installateur : -----		Nom installateur : -----	
VENDETOU		BON DE LIVRAISON	
Référence équipement : -----		Type : -----	
Numéro de série : -----		No garantie : -----	
No client : -----			
Nom : -----			
Adresse : ----- -----			
Code Postal : -----		Ville : -----	
Date de livraison : -----		Heure de livraison : -----	
signature client :			

- 1 Elaborer le dictionnaire des données sous forme de liste,
- 2 Déterminer les objets qui apparaissent, avec leurs propriétés,
- 3 Trouver des phrases qui permettent d'associer des objets entre eux, et en déduire les relations,
- 4 Formaliser par un schéma les objets et les relations.

Exercice intermédiaire 2 (étape 1-2-3)

Une société de vente d'équipements électroménagers fait effectuer des tournées de livraison par des installateurs.

Tous les matins, on délivre à chaque installateur une liasse de bons de livraison correspondant aux tournées qui lui incombent (c'est-à-dire relevant de son secteur). Après installation chez le client, le bon de livraison doit être signé :

No installateur : ----		Nom installateur : -----	
No tournée : -----			
Date tournée : -----		Heure départ tournée : -----	
Capacité tournée : -----			
VENDETOU		BON DE LIVRAISON	
Référence équipement : -----		Type : -----	
Numéro de série : -----		No garantie : -----	
No client : -----			
Nom : -----			
Adresse : ----- -----			
Code Postal : -----		Ville : -----	
Date de livraison : -----		Heure de livraison : -----	
signature client :			

1 Réitérer les opérations de l'exercice précédent avec la nouvelle formulation du cas et le nouveau document qui sont présentés.

2 Formaliser le nouveau schéma des objets et des relations.

Etape 4 : Revenons à l'exemple traité dans les étapes 1, 2 et 3 et au problème que posent certaines données qui ne peuvent être affectées à des objets.

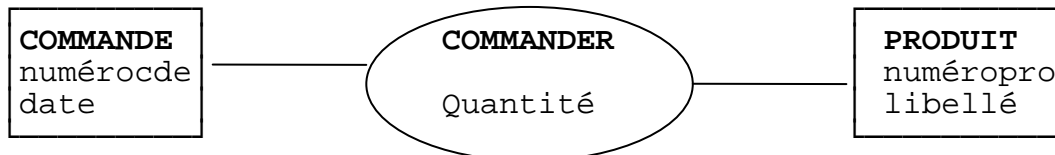
C'est ainsi que la quantité commandée (rubrique 13) dépend à la fois de la commande et du produit. En effet, pour toute quantité commandée, s'il n'y a pas de commande, on ne saura pas à quelle commande elle se réfère et s'il n'y a pas de produit, on ne saura pas à quoi s'adresse cette quantité commandée.

Autrement dit, cette donnée n'a de sens que par l'existence de ce rapport, qu'on a nommé relation, entre commande et produit.

On nommera cette relation "Commander" en référence à la sémantique suivante : une **commande** sert à **commander** des **produits**.

La donnée « Quantité » est une **propriété** de la relation "Commander". On dit que c'est une **relation porteuse de données**, par opposition aux relations précédentes qui étaient **non porteuses de données**, dites aussi **relations vides**.

Par ailleurs, on formalisera cette sémantique selon le schéma suivant, où l'on a inclus les propriétés des objets aussi bien que des relations.



On remarque que chaque objet dispose d'un **identifiant** (ici **numérocd** identifiant de COMMANDE, **numéropro** identifiant de PRODUIT).

Etape 5 : On se pose la question de savoir si on peut représenter des règles de gestion qui expriment le fait que ces relations peuvent exister(une ou plusieurs fois), ou ne pas exister(0 fois).

C'est ainsi que l'on mettra en place les **cardinalités**.

Par exemple :

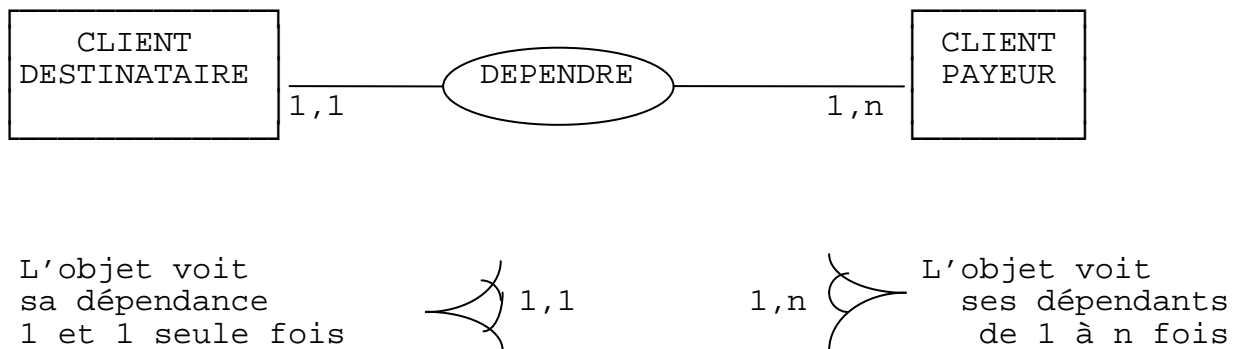
Un client payeur voit 1 client destinataire au minimum (il a au moins un dépôt chez lui ou à l'extérieur) et n clients destinataires au maximum (s'il a n dépôts) : ce sera une cardinalité **1,n** (minimum 1, maximum n).

Un client destinataire ne dépend que d'un client payeur et d'un seul : ce sera une cardinalité **1,1** (minimum 1, maximum 1).

Un produit peut être commandé par 0 commande au minimum (cas où il existe en stock mais non commandé) et n commandes au plus (cas où il existe en stock et on le voit apparaître dans plusieurs bons de commande), d'où la cardinalité **0,n**.

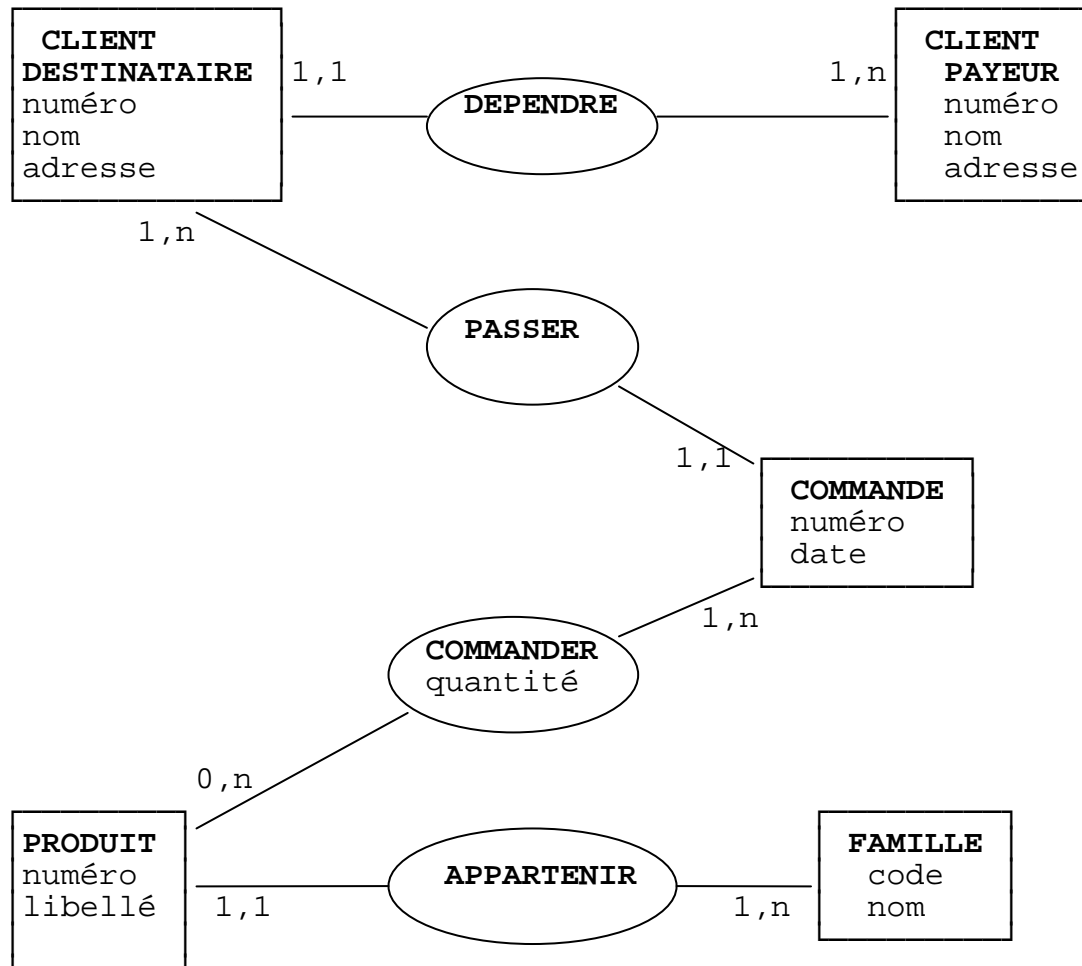
De même, une commande sert à commander au moins un produit (1), sinon plusieurs (n) : ce sera une cardinalité **1,n** etc.

Ces cardinalités seront notifiées sur le modèle comme suit :



Etape 6 : Enfin on établira le **modèle conceptuel des données** définitif, où l'on reporte pour chaque **objet** et chaque **relation** ses **propriétés**, c'est-à-dire les données qui lui sont propres et les **cardinalités** de chaque objet qui voit une relation. Les identifiants des relations seront analysés et reportés dans le chapitre suivant.

MODELE CONCEPTUEL DE DONNEES
MCD
 domaine "suivi des commandes clients"



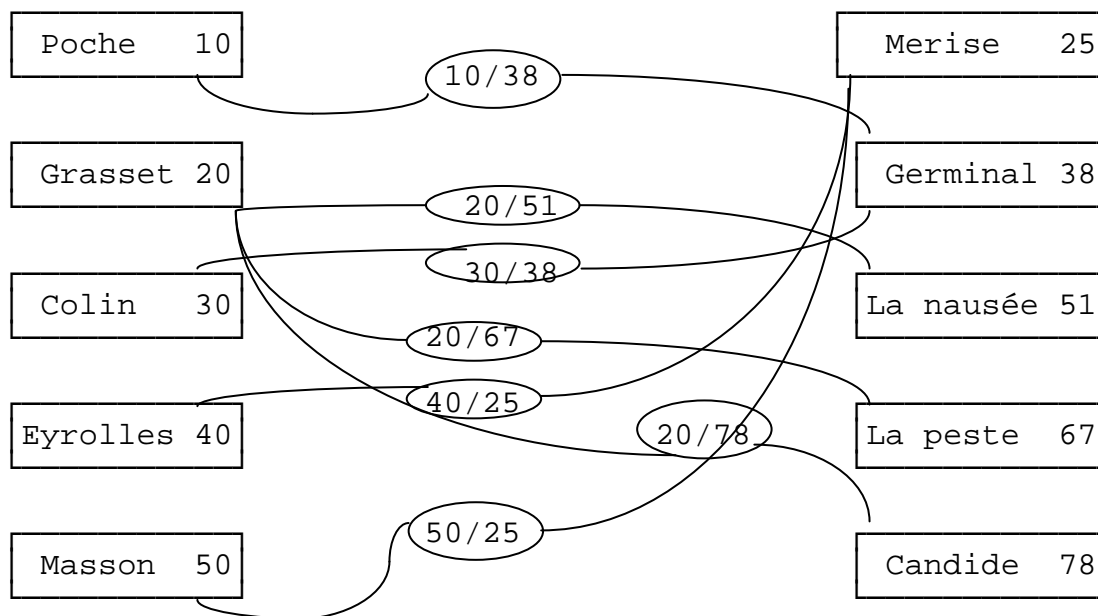
Aide à la conception :

1/ Dans un premier temps, afin de visualiser les objets en relation avec leur cardinalité et leurs occurrences et, afin de lever quelques doutes sur l'image que l'on se fait du modèle en cours d'élaboration, on pourra réaliser une sorte de "jeu d'essai" appelé **diagramme d'occurrence (D.O.)**, qui n'est pas une représentation normalisée, mais simplement une visualisation en "extension" aidant à mieux comprendre ce qui se passe.

Ainsi, à titre d'exemple, nous présentons le D.O. relatif à la relation PARAÎTRE entre EDITEUR et LIVRE :



Un diagramme d'occurrence pourrait être :



2/ Il est utile de noter que pour qu'un **objet** existe il faut qu'il corresponde à une **collection** d'informations homogènes que l'on veut gérer dans notre système d'information. Le choix de ces objets dépend des choix de gestion (c'est-à-dire ce que l'on veut conserver, manipuler, utiliser dans notre système).

Par ailleurs, il faut savoir que tout objet qui n'aurait qu'une occurrence n'est pas pris en compte (en général). C'est le cas par exemple de « l'entreprise » concernée par le S.I. ou de son "directeur", etc. On ne modélisera jamais :

ENTREPRISE acheter PRODUIT
ou
DIRECTEUR organiser REUNION

Exercice de synthèse 1

Dans une entreprise, on s'est limité au domaine "suivi des commandes de réapprovisionnement auprès des fournisseurs". On a obtenu la liste de données suivantes classées alphabétiquement :

- 1 adresse du fournisseur,
- 2 conditions particulières de la commande,
- 3 correspondant fournisseur de la commande,
- 4 date de commande,
- 5 date de livraison prévue de la commande,
- 6 libellé du produit,
- 7 montant global de la commande,
- 8 nom du fournisseur,
- 9 numéro de la commande,
- 10 numéro de téléphone du fournisseur,
- 11 prix unitaire du produit fixé par le fournisseur,
- 12 quantité produit commandée,
- 13 quantité produit en stock,
- 14 prix unitaire de vente (HT) du produit.
- 15 accusé de réception de la commande par le fournisseur

On sait que chaque produit est proposé par différents fournisseurs à un prix qu'ils ont établi et communiqué.

Déterminer les objets, les relations, les propriétés attachées à ces entités, et établir le MCD avec ses cardinalités en suivant les étapes 1-2-3-4-5

Exercice de synthèse 2

Soit la base de connaissance suivante :

"L'Institut privé est un établissement qui offre un certain nombre de stages auxquels peuvent s'inscrire des stagiaires.

Chaque stage est identifié par un code, est caractérisé par un libellé(remise à niveau scientifique, perfectionnement en anglais, espagnol débutant,...) et une date de début.

A chaque stagiaire est affecté un numéro d'identification. En outre, on enregistre son nom, son adresse et la date d'inscription au stage. On sait qu'un stagiaire peut s'inscrire à plusieurs stages différents mais ne peut s'inscrire qu'une fois à un stage donné.

Chaque stage est composé de matières connues par leur code et leur libellé(grammaire espagnole, mathématiques, mécanique,...)

Une matière peut être suivie dans le cadre de plusieurs stages et elle est enseignée par un seul professeur. Ce professeur est connu par son matricule et son nom et prénom."

Etablir le modèle conceptuel de données correspondant en suivant les étapes 1-2-3-4-5.

MERISE

III

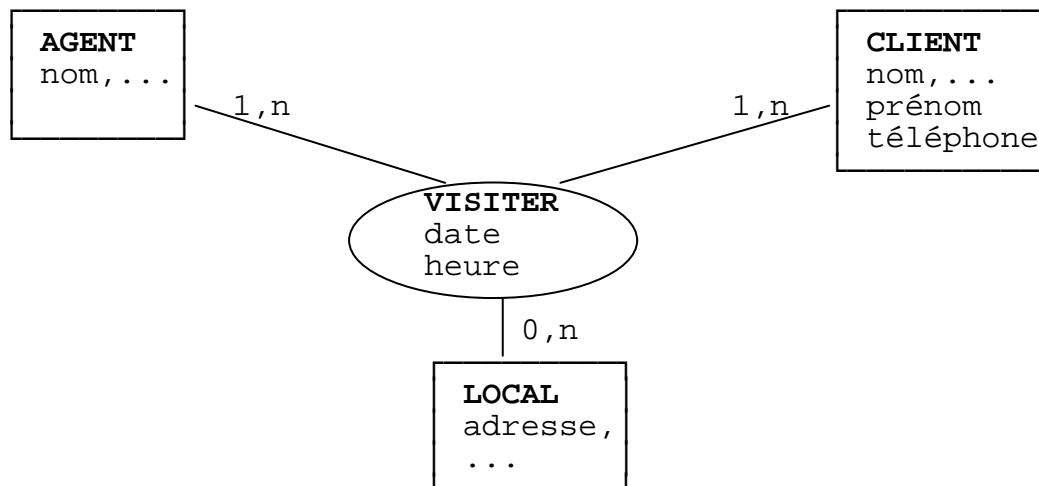
MODELE CONCEPTUEL DE DONNEES CAS PARTICULIERS DE RELATIONS

Relations n-aires

Nous avons mis en évidence des relations entre deux objets, elles se nomment **relations binaires**, mais il peut en exister entre trois objets ou plus. Dans le descriptif de gestion d'une agence immobilière, par exemple, on note :

"L'agent immobilier prend des rendez-vous pour visiter des locaux (appartements, villas,...) avec des clients potentiels ; il note pour chaque visite la date et l'heure et pour le client son nom, prénom, numéro de téléphone".

Il apparaît que l'existence d'une visite dépend des trois objets : AGENT, CLIENT, LOCAL et que l'absence d'un seul de ces objets rend nulle et non avenue la mise en place de cette relation VISITE. On dit que c'est une **relation ternaire** et on la modélise ainsi :



Les cardinalités consistent à dénombrer le minimum et maximum d'occurrences possibles de la relation vue depuis une occurrence d'un objet. Ainsi, pour un agent donné il organise de une à n visites, soit (1,n). Pour un client donné, il effectue de une à n visites, soit (1,n). Enfin, un local est sujet à 0 ou n visites, soit (0,n).

Remarque 1 :

Il est intéressant de remarquer que pour mettre en évidence une relation n-aire, on pourra le faire par une approche dite "relationnelle" telle que :

pour VISITER, il faut connaître

{
 l'AGENT
 le CLIENT
 le LOCAL

Remarque 2 :

Pour des relations binaires, on a plutôt tendance à utiliser une approche "sémantique" telle que :

client "PASSER" commande

que l'on découvre soit de façon explicite dans la base de connaissances, soit de façon implicite en formulant soi-même la phrase la mieux adaptée.

Exercice intermédiaire sur relations n-aires

Mettre en évidence les relations n-aires dans les bases de connaissances partielles suivantes :

1 - "On veut gérer les livraisons de véhicules (connues par la marque, le type, le numéro de série) effectuées par les concessionnaires (connus par le nom et la ville) à qui on confie la charge des administrations clientes (repérées par nom, adresse, ville)".

2 - "Les appareillages sont commandés au fournisseur le plus proche de la succursale de notre société".

3 - "On veut contrôler les tournées commanditées par les distributeurs auprès de leurs magasins. Chaque distributeur choisit les transporteurs qu'il désire et leur assigne les magasins qu'ils devront livrer dans leur tournée, en sorte que des tournées différentes ne devront pas passer dans un même magasin".

4 - "La société doit gérer de plus près ses stocks de vêtements (blue-jean Lewis, jupe droite pied de poule...) pour chaque famille (pantalon, robe, jupe, chemisier...) pour chaque catégorie (homme, femme, enfant...) et chaque taille (38,40,42,...)".

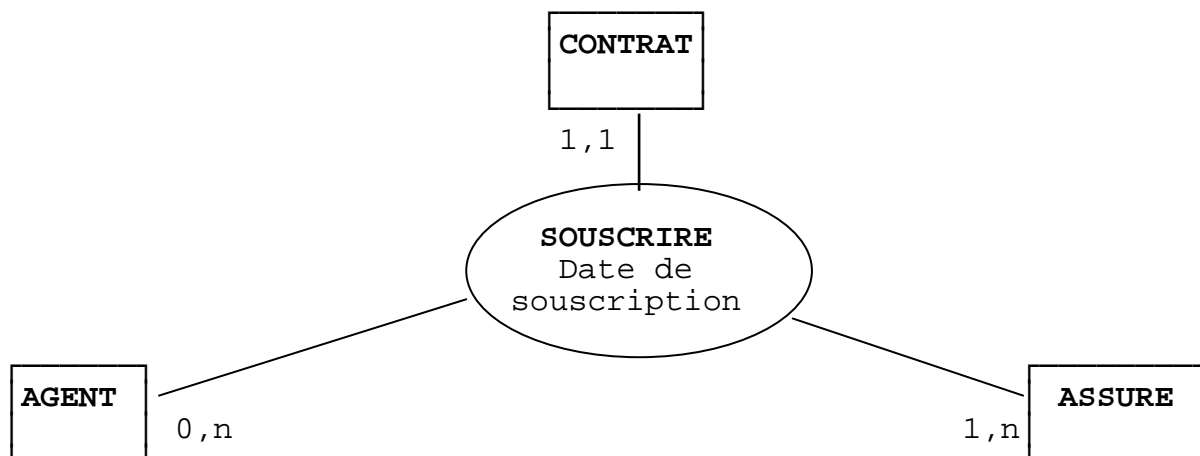
5 - "On voudrait suivre les résultats des chevaux dans leur participation à n'importe quel type de concours (complet, cross, dressage,...) dans chaque ville où ils ont lieu, à des dates données."

Contraintes d'intégrité fonctionnelles

Le problème que pose les relations n-aires c'est qu'elles seront difficiles à gérer (plus tard dans les modèles logiques et physiques). On va chercher, à chaque fois qu'on le pourra, à les réduire, par exemple à décomposer une relation ternaire en plusieurs relations binaires.

Du texte partiel suivant :

"Un agent d'assurance fait souscrire, à une certaine date, un contrat à son assuré...", on a déduit le modèle restreint suivant :



On constate qu'il y a une **dépendance fonctionnelle** entre un agent et un contrat, ce qui signifie que :

Si je connais un contrat, je connais obligatoirement **un et un seul** agent qui en a la responsabilité.

(La réciproque n'étant pas vraie car si je connais un agent, je ne connais que **les** contrats dont il est responsable).

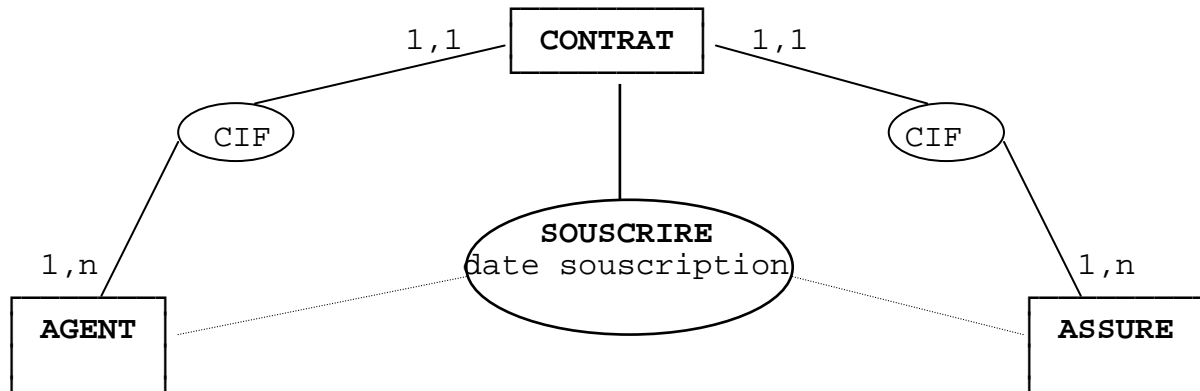
Cela montre qu'il y a une **relation obligatoire** (unique) entre un contrat et son agent. Elle est appelée **Contrainte d'Intégrité Fonctionnelle (CIF)**. Une CIF est toujours une relation de type **x,1 - x,n** (x valant 0 ou 1) :



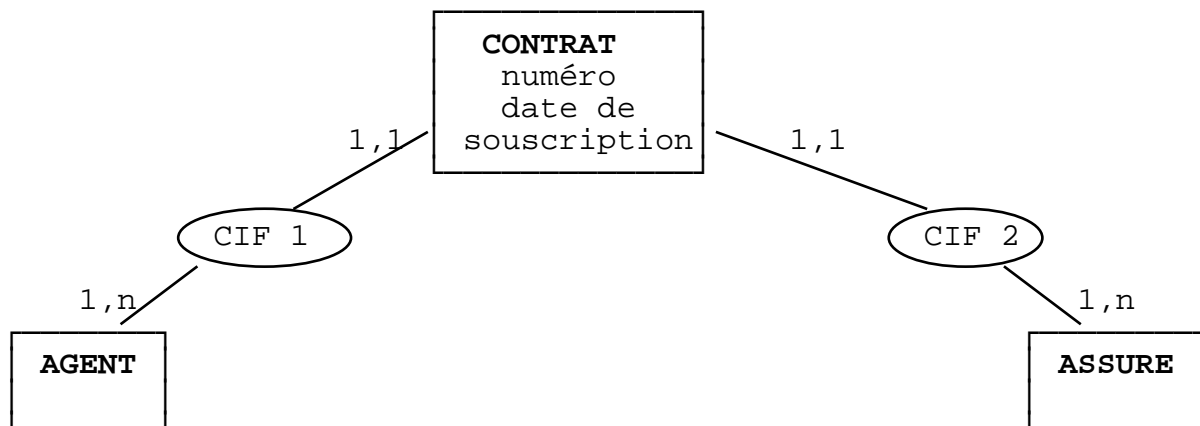
De même, comme un contrat ne peut être souscrit que par **un et un seul** assuré, il existe une contrainte qui associe l'objet contrat à l'objet assuré, donc une autre CIF (encore de type **x,1 - x,n**).

Attention : une relation de type **x,n - x,n** n'est pas une CIF.

Le schéma partiel précédent devient alors :



Si on fait "**glisser**" la propriété "date de souscription" au niveau de l'objet **CONTRAT**, on obtient un schéma, où la relation ternaire peut être complètement éliminée :



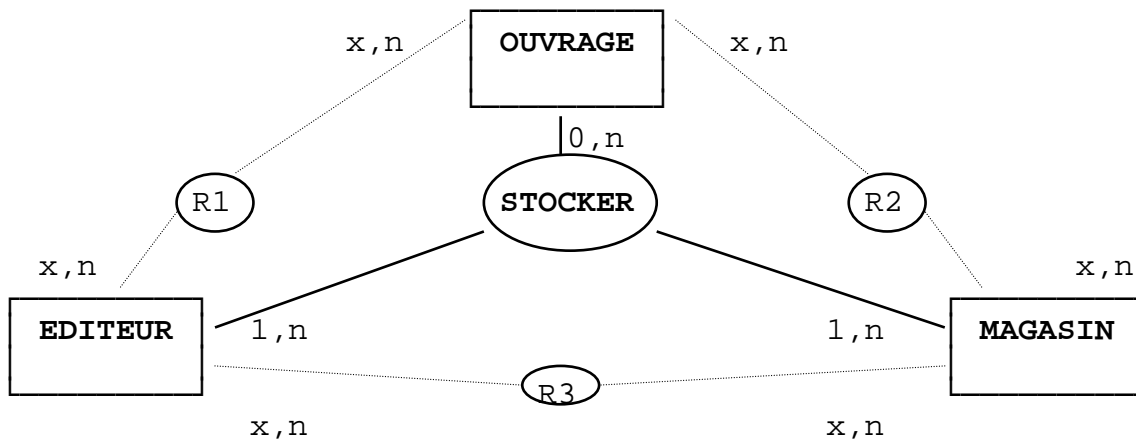
Il est utile ensuite de trouver une **dénomination** aux **CIF** ainsi mises à jour :

pour CIF 1 : agent SUIVRE contrat,
pour CIF 2 : assuré SIGNER contrat.

Remarque 1 : Il faut bien noter que les CIF ne peuvent être trouvés que pour des relations de cardinalités $x,1 - x,n$ tel que



Par exemple dans le schéma qui suit, on ne pourra effectuer aucune réduction :



En effet les cardinalités directes pour les relations:

- R1 entre OUVRAGE et EDITEUR sont de type $x,n - x,n$
- R2 entre OUVRAGE et MAGASIN sont aussi de type $x,n - x,n$
- R3 entre EDITEUR et MAGASIN sont aussi de type $x,n - x,n$

Remarque 2 : on notera aussi qu'une façon de mettre en évidence des relations d'ordre supérieur à 2, c'est de les formuler de la manière suivante, par exemple dans le cas précédent :

- Pour STOCKER, il faut connaître
 - l'OUVRAGE
 - l'EDITEUR
 - le MAGASIN

et on s'aperçoit après construction du modèle que l'on n'obtient aucune réduction,

- Par contre lorsque l'on énonce que :

pour SOUSCRIRE, il faut connaître

- l'AGENT
- le CONTRAT
- l'ASSURE

Le modèle construit plus haut avec ses cardinalités nous a montré qu'il était réductible.

Exercice de synthèse 3

1 - Construire les modèles conceptuels de données suivants et les réduire par mise en place des CIF s'il y a lieu, préciser avec quelles hypothèses de gestion ces CIF existent ou n'existent pas :

a - pour gérer
le RECRUTEMENT, il faut connaître { le CANDIDAT (qui postule)
le POSTE (PROFIL proposé)
la SUCCURSALE (qui offre)

b - Une chaîne hôtelière fait réserver les chambres de ses différents hôtels par l'intermédiaire d'agences qui reçoivent les demandes des clients :

pour RESERVER,
il faut connaître { le CLIENT (qui recherche des chambres)
la CHAMBRE (disponible dans un hôtel)
l'HÔTEL (qui nous a confié ses réservations)
l'AGENCE (qui fait la réservation)

c - "dans un institut privé de langues, un stagiaire peut s'inscrire à une certaine date pour un cours (anglais, russe,...) à la session qui lui convient (printemps, été, automne, hiver)".

d - "une société veut savoir à quel service (administration, commercial,...) et à quelle succursale (Bordeaux, Toulouse, Nice,...) appartiennent ses employés".

2 - montrer que la formulation par une approche relationnelle telle que :

A/ pour COMMANDER, il faut connaître { le CLIENT
la COMMANDE
le PRODUIT

donne après réduction le même modèle que dans une approche sémantique telle que :

B/ Client PASSER Commande pour COMMANDER Produit.

Exercice de synthèse 4

1 - Le service de formation d'une entreprise souhaite gérer ses actions de formation. Afin d'adresser des convocations aux employés, on enregistre les agents avec leur numéro, leur nom, prénom ainsi que l'établissement auquel ils appartiennent (l'entreprise est implantée dans différents lieux géographiques). De cet établissement on connaît le code, le nom et l'adresse.

Un certain nombre de cours sont offerts aux employés qui peuvent s'y inscrire. Ces cours sont connus dans un catalogue, où ils figurent avec un code et un intitulé. Tous les cours sont animés par des employés internes à l'entreprise. On s'assurera des cours que les personnes affectées à l'animation ne sont pas sujettes à une inscription à ce même cours. Personne dans cette entreprise n'ayant le don d'ubiquité.

On désire établir des convocations, un mois avant le début des cours, aussi bien pour le personnel inscrit que pour le personnel enseignant. A ce sujet, il faut savoir que tout cours (initiation informatique, anglais débutant, espagnol perfectionnement...) est planifié plusieurs fois par an et que l'on parle plus volontiers de session. Une session est repérée par un numéro et a un intitulé (printemps, été...). La planification consistera à décider quels cours seront offerts dans une session, et pour chacun d'entre eux quelle en sera la date et la durée, puis de les inscrire au catalogue. Les inscriptions ou animations ou convocations se font par référence aux cours et aux sessions.

Faire le MCD correspondant.

MERISE

IV

MODELE CONCEPTUEL DE DONNEES RELATIONS REFLEXIVES ET RELATIONS A ROLE

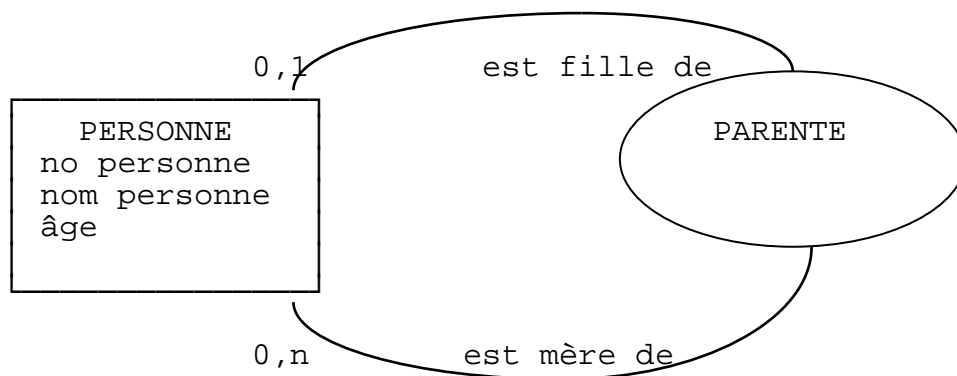
Relations réflexives

Jusqu'à maintenant nous avons établi des relations entre deux ou plusieurs objets différents.

Pourtant il peut exister des relations sur un seul objet (c'est-à-dire de l'objet sur lui-même). Par exemple on veut gérer un ensemble de personnel féminin (connu avec leur numéro, leur nom, leur âge) et on souhaite faire apparaître les relations parentales les concernant, de façon à repérer :

- si une personne est la mère d'une autre,
- si une personne est la fille d'une autre.

On créera l'objet PERSONNE et une relation PARENTE selon la sémantique "une personne est parente d'une personne". Ceci est une **relation réflexive** schématisée ainsi :



En ce qui concerne les cardinalités, on voit que :

- la relation dans le sens "**est la mère de**" est (0, n) car une femme peut être mère de 0 à n filles.
- la relation dans le sens "**est la fille de**" est (0, 1) car une femme a toujours une mère mais celle-ci n'est pas forcément répertoriée dans le système.

Exercice relation réflexive 1

Le service du personnel d'une entreprise souhaite gérer les employés. Ces derniers sont identifiés par leur numéro et on connaît leur nom, salaire de base, qualification, date d'embauche.

De plus on souhaite repérer le numéro d'employé du chef de service qui dirige chacun d'entre eux.

Enfin on désire savoir à quel service (connu avec son numéro, nom, localisation) ils appartiennent.

Etablir le MCD de cette base de connaissance.

Exercice relation réflexive 2

On veut gérer des pièces et ces mêmes pièces sont composantes d'autres pièces (puisque le cadre, les roues, le guidon,... sont composants de bicyclette, etc.).

Etablir le MCD avec la relation réflexive qui correspond.

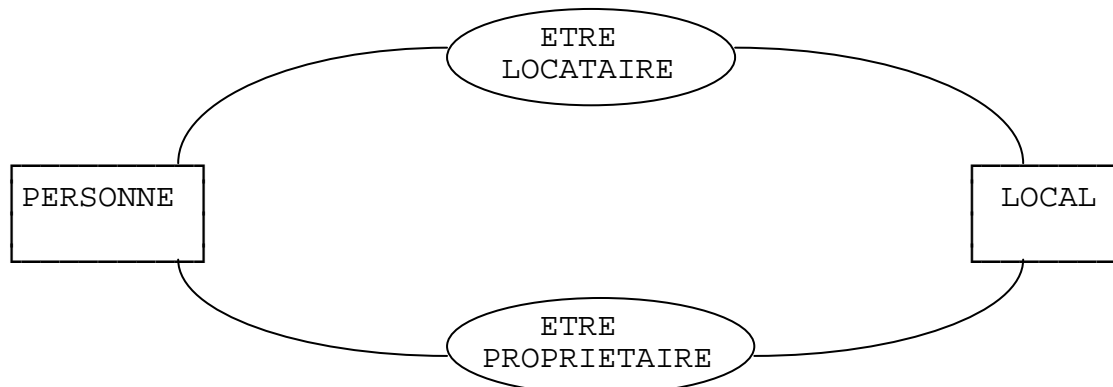
Relations à rôle:

Imaginons que nous ayons modélisé un système où il est question de gérer des personnes et les locaux où elles habitent, dans le cadre par exemple d'une agence immobilière ou d'un syndic d'immeuble. On a obtenu le schéma suivant :



On va rapidement constater que cette relation est insuffisante parce que les propriétés de "HABITER" vont différer s'il s'agit d'une personne propriétaire ou locataire. En effet, le loyer mensuel (pour un locataire) est affecté à la relation HABITER, de même que le prix de vente (pour un propriétaire). Il en est ainsi pour d'autres propriétés (taxe foncière, date d'achat, date de premier loyer...), en sorte que la relation entre Personne et Local ne sera pas la même selon le rôle joué par la personne vis-à-vis du local.

On mettra en place alors deux relations à rôle, l'une s'appellera "ETRE LOCATAIRE", l'autre "ETRE PROPRIETAIRE" et formalisées comme suit :



On notera qu'il n'est pas toujours possible de trouver un verbe simple correspondant à la relation. On dérogera à la recommandation le moins souvent possible. On peut s'imposer comme autre règle de nommer les relations uniquement avec des substantifs (habitation, inscription, enseignant...). Dans les deux options, évidemment, on ne réussira pas toujours à trouver le mot convenable ; on cherchera simplement à s'y conformer au mieux.

Exercice relation à rôle 1

"Les commandes en provenance des grossistes sont sujettes à une date de livraison liée à la date de tournée et ne sont prises en compte qu'au-dessus d'une somme minimum. Par contre, s'il s'agit d'un détaillant, la commande de ce client sera établie au comptoir et servie sur place."

Etablir le modèle qui permet de lier les commandes à ces clients.

Exercice relation à rôle 2

"Une compagnie d'aviation propose des vols (auxquels elle attribue un numéro, un type d'avion, un matricule de pilote, un nombre de passager) au départ de certaines villes (connues par leur nom, la dénomination de l'aéroport et le pays) et à destination d'autres villes. Les départs ont certaines caractéristiques (heure d'enregistrement, heure de départ, hall et porte de départ...), les arrivées en ont de différentes (heure d'arrivée, hall de récupération des bagages...)"

Faire le MCD représentatif.

Exercice relation à rôle 3

"Afin de suivre le cursus honorum de chiens de race, on veut savoir chez quels propriétaires ils sont nés, puis ceux chez qui ils sont décédés, sachant qu'entre la naissance et la mort ils ont été vendus plusieurs fois"

Faire le MCD représentatif.

MERISE

V

MODELE CONCEPTUEL DE DONNEES RECAPITULATIF DE LA DEMARCHE DE MODELISATION

Pour construire un modèle conceptuel de données on suivra la démarche par étapes suivante, où l'on a rappelé les principales définitions :

1 Analyser l'existant en constituant un catalogue exhaustif de toutes les données utilisées par l'organisation, que l'on appelle dictionnaire des données. On pourra s'aider pour cela des documents véhiculés dans le système.

Dégager la signification précise de ces données, c'est-à-dire épurer le catalogue des polysèmes et synonymes.

2 Mettre en évidence les objets naturels:

OBJET : reflet d'une entité manipulée par l'organisme, dotée d'une existence propre, dont chaque occurrence est identifiable par une donnée particulière.

3 Rattacher à ces objets leurs propriétés:

PROPRIETES : plus petit élément logique d'information, qui a un sens en lui-même et dont la valeur caractérise partiellement une occurrence d'objet ou de relation.

4 Recenser les relations vides (non porteuses de données) en exprimant les dépendances entre objets.

RELATION : entité dont l'existence des occurrences dépend de l'existence des objets qu'elle relie.

5 Etudier les relations porteuses de données en analysant les propriétés non attachées à des objets.

6 Relever les identifiants des objets.

IDENTIFIANT : c'est une propriété d'un objet, choisie pour qu'à chacune de ses valeurs, corresponde une et une seule occurrence de l'objet.

7 Evaluer les cardinalités de chaque couple objet-relation.

CARDINALITE : nombre minimum et maximum d'occurrences d'une relation vue d'une seule occurrence d'un objet.

8 Simplifier le modèle à l'aide des contraintes d'intégrité fonctionnelles (CIF)

CIF : relation obligatoire entre deux objets qui n'est pas du type $x, n - x, n$.

Exercice de synthese 5

Dans le cadre de la réorganisation du système d'information d'une agence immobilière de la région parisienne, une première liste des données a été établie. Elle concerne essentiellement le sous univers "offre de ventes et de locations". Construire le modèle conceptuel des données correspondant.

Les données sont recueillies par ordre alphabétique :

- 1 adresse du client,
- 2 adresse du propriétaire,
- 3 agent immobilier,
- 4 année de construction,
- 5 charges trimestrielles du local,
- 6 commission de l'agent immobilier (en pourcentage),
- 7 date prévue de libération des lieux,
- 8 date et lieu de rendez-vous,
- 9 date et lieu de visite,
- 10 département de la localité,
- 11 durée du mandat,
- 12 étage du local (si appartement),
- 13 état des lieux,
- 14 garage,
- 15 gare SNCF, métro ou bus la plus proche,
- 16 nature du mandat(vente ou location),
- 17 nom du client,
- 18 nom du propriétaire,
- 19 nombre de pièces,
- 20 numéro (de l'adresse du local),
- 21 prix du loyer mensuel,
- 22 prix de vente,
- 23 quartier du local,
- 24 rue du local.

Exercice de synthèse 5 (suite)

- 25 surface habitable,
- 26 surface de terrain (si pavillon),
- 27 téléphone personnel du client,
- 28 téléphone personnel du propriétaire,
- 29 téléphone professionnel du client,
- 30 téléphone professionnel du propriétaire,
- 31 type du local (appartement ou pavillon),
- 32 ville du local.
- 33 standing moyen du quartier (codifié S1, S2, S3)
- 34 nombre d'étage du pavillon
- 35 nombre d'étage de l'immeuble

Travail à réaliser : construire le MCD.

Exercice de synthèse 6

Une Société de location de téléviseurs veut gérer les tournées de ses installateurs pour la livraison ou le retrait des appareils sous contrat. On a recueilli les explications suivantes :

A partir des dates de début de contrat (dates à livrer) et des dates de fin de contrat (dates à retirer) la société NALPLUS établit une fiche de tournée journalière selon le modèle ci-après.

Les tournées, identifiées par un numéro de tournée et une date de tournée, sont confiées à des installateurs connus par leur numéro et leur nom. Chaque intervention comporte le numéro de contrat, le nom et l'adresse du client, puis le type (NB = noir et blanc, CL = couleur...) et la référence du téléviseur, ainsi que le type d'intervention (à livrer ou à retirer). Enfin, si l'intervention a pu s'effectuer, on note sa réalisation (OK), sinon on la reporte dans une tournée ultérieure.

On a obtenu un exemplaire de document délivré chaque matin aux installateurs :

<u>NALPLUS</u>							
No de tournée : T504				Date : 20/04/89			
No installateur : I12				Nom : DUCHENE ET FILS			
Numero contrat	Nom/No client	Adresse Client	TYP	Référence Télévis.	Livraison Retrait	Effect	
2006700H	Dupont G. C432	3,rue du du Bois BALMA	NB	1052HF	Livrer	OK	
4007800J	Storm P. C435	6,rue d' Ulm BALMA	CL	30007HJ	Retirer		
5009600J	Ecele M. C447	12,Av. Beltre L UNION	NB	3008IJ	Livrer	OK	

Travail à réaliser : construire le MCD.

MERISE
VI
MODELE LOGIQUE DE DONNEES
MISE EN PLACE

Objectifs

Le modèle logique des données est une représentation du modèle conceptuel en termes d'organisation des données : il se réfère à l'état de l'art technologique c'est-à-dire des possibilités du moment en matière de logiciels et de matériels.

Généralités

Actuellement, on assiste à une mutation des logiciels de gestion de bases de données dites hiérarchiques ou navigationnelles vers des systèmes relationnels.

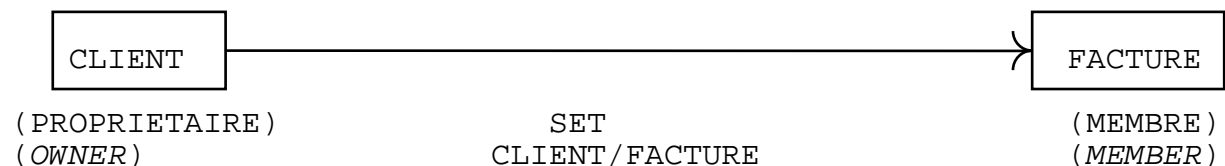
Le formalisme proposé pour schématiser le modèle logique est un appauvrissement de celui utilisé au niveau conceptuel ; il coïncide avec la notation de Backmann adoptée par CODASYL (CONFérence on Data SYstems Language) qui permet de conserver l'indépendance des traitements par rapport aux données et à leur implantation physique. Ce langage de description des données s'appuie sur des définitions et un type de diagramme que nous n'utiliserons que temporairement parce que notre propos est ici de déboucher sur **le modèle relationnel** que nous avons évoqué dans le chapitre "préliminaire à Merise".

Première approche

Soit **le MCD** suivant :



Le **diagramme de Backmann** correspondant sera :



CLIENT est propriétaire de 0 à n FACTURE.

FACTURE est membre de 1 et 1 seul CLIENT.

(Nous justifierons plus tard la disparition de la relation RECEVOIR)

L'ensemble s'appelle le set CLIENT/FACTURE

Pour la suite, nous serons amenés à utiliser des **définitions** que nous allons comprendre en nous appuyant sur l'exemple précédent :

RECORD : collection de données (ou Segment)
exemple : record FACTURE

DATA : donnée (ou propriété)
exemple : data montant, nom client,...

SET : ensemble de 2 RECORDS (ou ensemble PM) qualifiés de :
l'un "OWNER" (P=Propriétaire)
et l'autre "MEMBER" (M=Membre).
exemple : set CLIENT/FACTURE
record owner : CLIENT
record member : FACTURE

On notera que, pour une occurrence du RECORD OWNER, il peut exister de 0 à n occurrences du RECORD MEMBER, alors que, pour une occurrence du RECORD MEMBER, il ne peut exister que 0 ou 1 occurrence du RECORD OWNER.

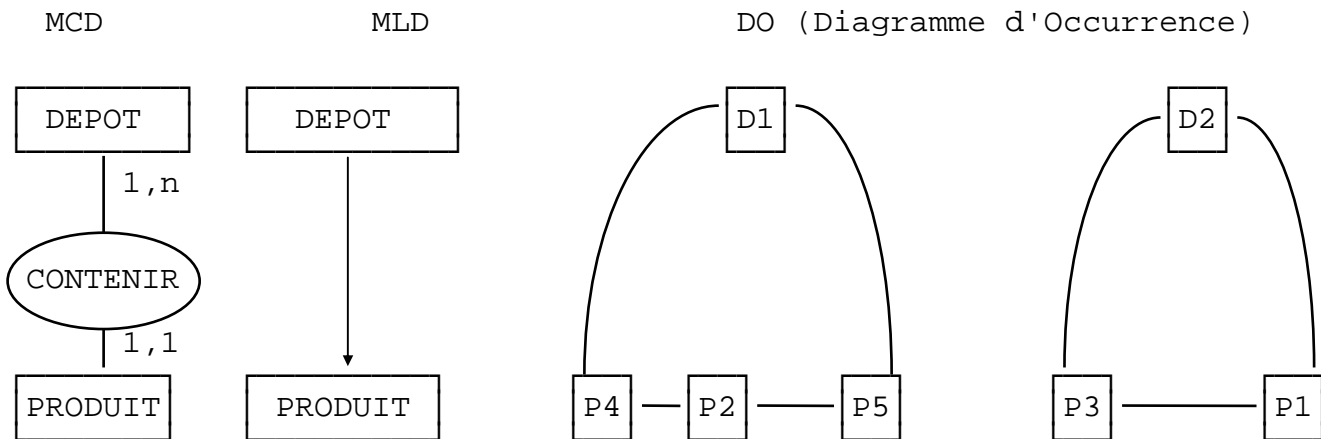
Nous élaborerons, dans un premier temps seulement, le diagramme de Backmann (que nous appellerons désormais notre MLD) mais nous allons l'accompagner des diagrammes d'occurrence simplement pour imager son formalisme.

Le diagramme d'occurrences (DO) permet de mieux comprendre la **représentation logique (ou organisationnelle)** proposée par le MLD dans la mesure où le nombre de structures illustrées reste limité (après quoi le passage au modèle relationnel sera presque une simple formalité).

C'est cette représentation de Backmann que nous appellerons le **Modèle Logique de Données (ou MLD)**

Voici donc trois structures classiques qui vont éclairer ce passage du MCD au MLD :

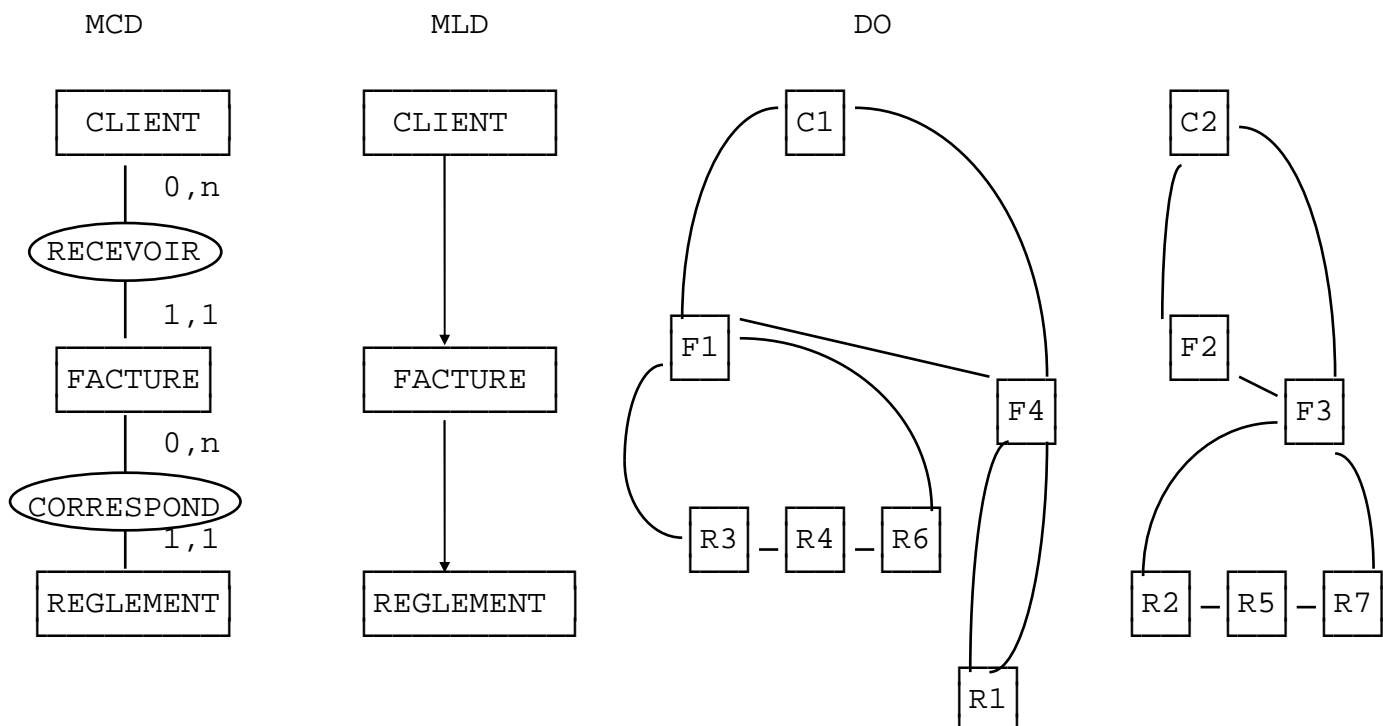
I Hiérarchie à un niveau: un RECORD est propriétaire d'un autre RECORD, par exemple :



Il apparaît que, par exemple, les produits P4, P2, P5 sont gérés dans le dépôt D1 ,et P3, P1 par le dépôt D2.

II Hiérarchie à plusieurs niveaux :

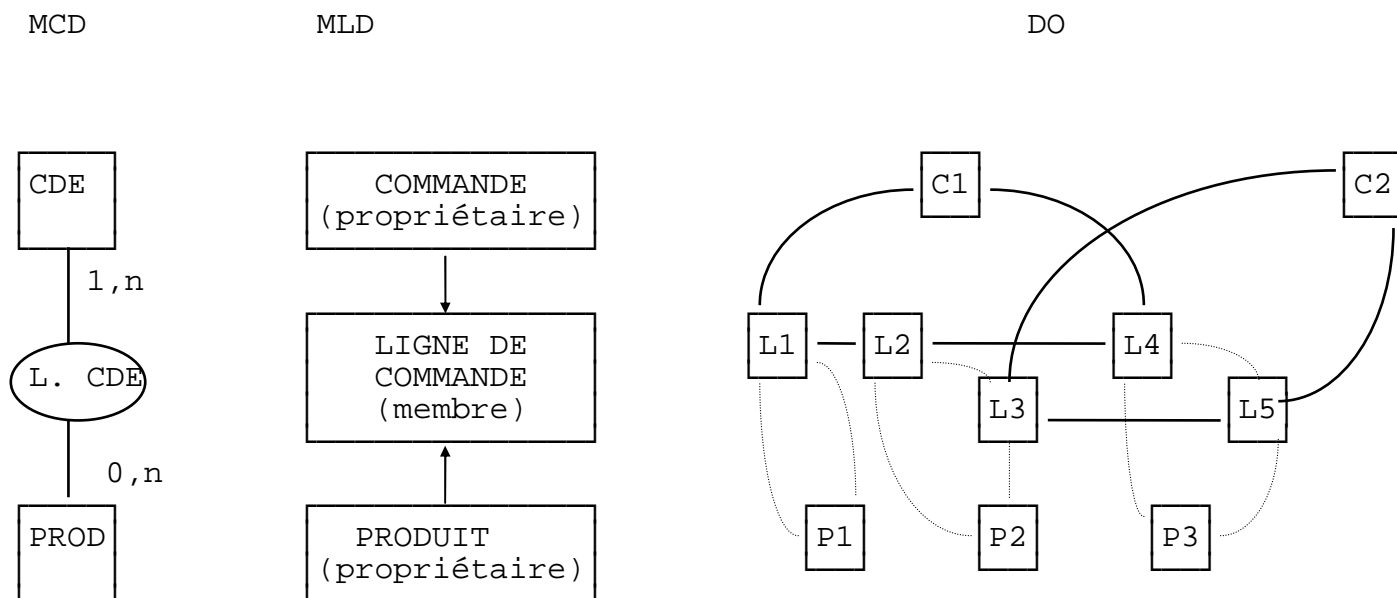
dans l'exemple qui suit le RECORD intermédiaire(FACTURE) se trouve membre du premier SET(CLIENT/FACTURE) mais propriétaire dans le second SET(FACTURE/REGLEMENT) :



Ici est représenté le fait que le règlement R4 est un règlement partiel de la facture F1 du client C1 ; pour le client C2, on voit qu'il est possible qu'une facture F2 n'ait pas encore été réglée.

III Réseau

Un membre appartient à plusieurs RECORDS



Dans cet exemple, on peut constater que le produit P2 est commandé dans la deuxième ligne (L2) de C1 et dans la première ligne (L3) de C2.

On remarque que le passage du modèle conceptuel au modèle logique est relativement simple. Ce modèle logique constitue une étape intermédiaire avant le modèle physique qui, lui, sera dépendant des logiciels (par exemple Oracle, Ingres, SQL-Server...)

Remarque : Dans le MCD qui précède, on notera que la relation "commander" a été appelée "Ligne de commande" : c'est une autorisation que l'on se donnera assez souvent vu le poids de ce vocable en gestion. On retrouvera parfois cela quand il s'agit de lignes de livraison, de lignes de facture...). Il est naturel que ces dénominations se retrouvent dans le MLD.

On va maintenant lister toutes les règles de passage du MCD au MLD et nous donnerons un exemple pour chacune d'entre elles.

Règles de passage au niveau logique

La transformation d'un modèle conceptuel en un modèle logique est entièrement algorithmique (c'est la raison pour laquelle il existe des progiciels-outils de MERISE réalisant cette phase automatiquement et proposant la gestion d'un dictionnaire des données).

Les règles de passage sont énoncées ci-après et chacune est illustrée d'un **exemple dans les pages suivantes** :

1 - Les **propriétés** deviennent des DATA ITEM.

2 - Tout **objet** donne un RECORD.

3 - **Toute relation binaire de cardinalité $x,1 - x,n$**
(x valant 0 ou 1), se transforme en un SET dont :

le propriétaire correspond à la cardinalité x,n
le membre à la cardinalité $x,1$

Quand la relation est porteuse de données, celles-ci "glissent" dans les DATA ITEMS du membre. De plus l'identifiant du RECORD propriétaire est reporté dans le RECORD membre, où il deviendra une "clé étrangère" (clé de communication entre RECORD : voir chapitre suivant).

4 - **Toute relation binaire de cardinalité $x,n - x,n$** donne lieu à la création d'un RECORD membre qui appartiendra par deux SETS à

- _ un premier RECORD propriétaire (voyant une cardinalité x,n)
- un deuxième RECORD propriétaire (voyant une cardinalité x,n)

L'identifiant du nouveau RECORD membre sera la concaténation des identifiants des RECORDS propriétaires.

5 - **Toute relation n-aire**, par extension, donnera un RECORD membre et n SETS avec les n RECORDS correspondant aux objets. La même règle de concaténation des identifiants s'applique.

6 - **Une relation réflexive** va donner :

un pseudo-RECORD membre dans deux SETS du même RECORD propriétaire correspondant à l'objet.

7 - **Cas particulier** : une relation binaire de cardinalité $0,1 - 1,1$ donne un SET dont :

- le RECORD propriétaire a la cardinalité $0,1$.
- le RECORD membre a la cardinalité $1,1$.

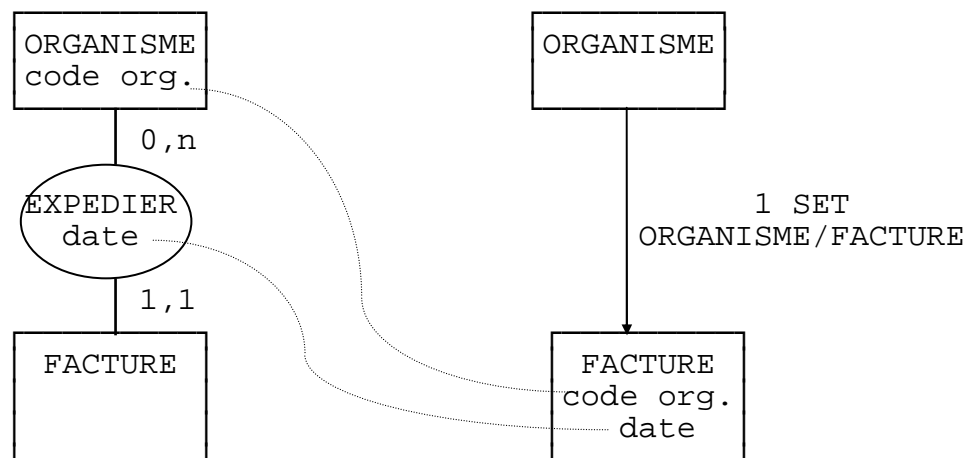
REGLE 1

Les propriétés de l'objet PRODUIT, Nom, Prix Unitaire, Quantité en Stock deviennent des Data Items (c'est-à-dire des données codifiables) : NOM (20 caractères), PU (8 numériques dont 2 décimales), QTE STOCK (6 numériques).

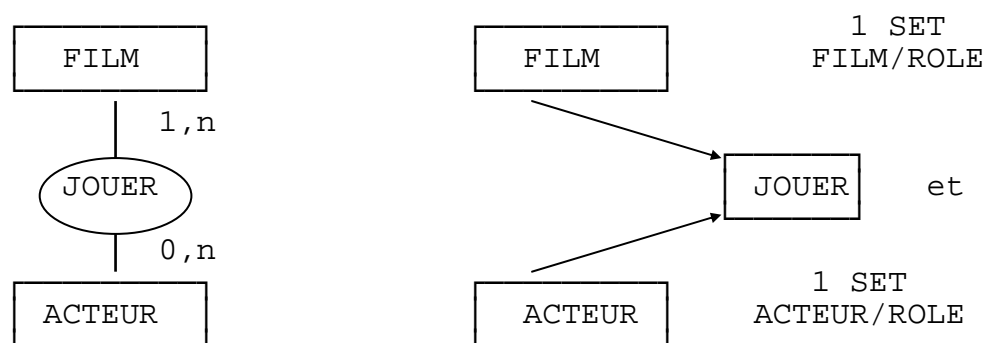
REGLE 2

L'objet FACTURE devient le record FACTURE.

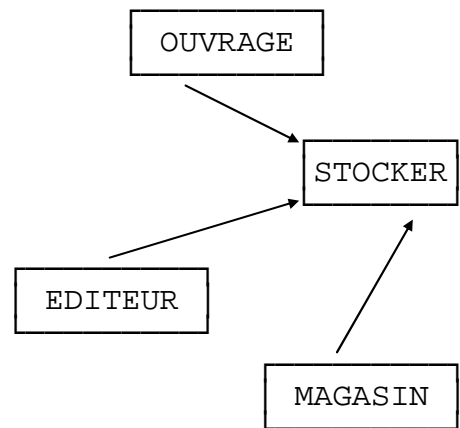
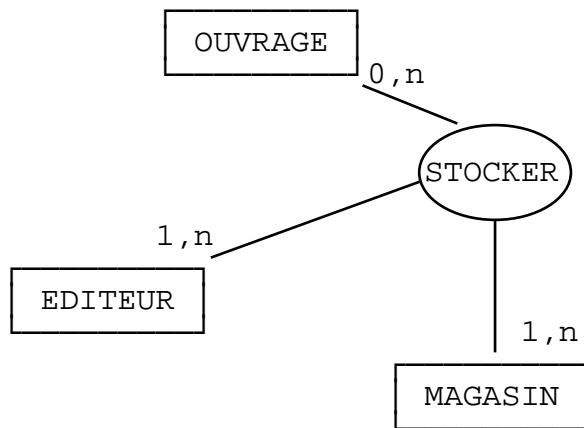
REGLE 3



REGLE 4

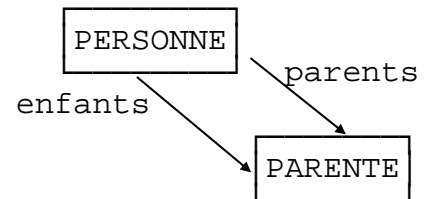
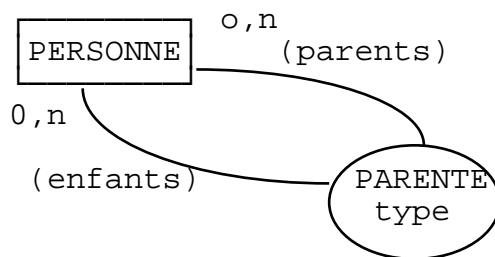


REGLE 5



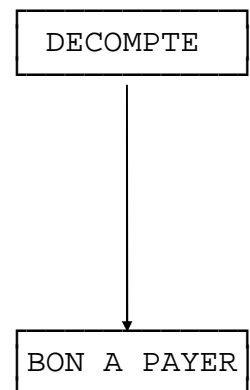
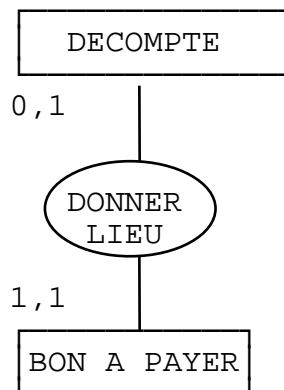
3 SET
 OUVRAGE/STOCK
 EDITEUR/STOCK
 MAGASIN/STOCK

REGLE 6



2 PSEUDO SET

REGLE 7



1 SET DECOMPTE/BON A PAYER

Exercice de synthèse 7 (sur MLD)

Le texte qui suit reprend l'exercice de synthèse 4, mais enrichi de plusieurs paragraphes que vous modéliserez afin de modifier le MCD d'origine.

Puis vous en déduirez le MLD

1 - Le service de formation d'une entreprise souhaite gérer ses actions de formation. Afin d'adresser des convocations aux employés, on enregistre les agents avec leur numéro, leur nom, prénom ainsi que l'établissement auquel ils appartiennent (l'entreprise est implantée dans différents lieux géographiques). De cet établissement on connaît le code, le nom et l'adresse.

Un certain nombre de cours sont offerts aux employés qui peuvent s'y inscrire. Ces cours sont connus dans un catalogue, où ils figurent avec un code et un intitulé. Tous les cours sont animés par des employés internes à l'entreprise. On s'assurera que les personnes affectées à l'animation de cours ne sont pas sujettes à une inscription à ce même cours (Personne dans cette entreprise n'ayant le don d'ubiquité).

On désire établir des convocations aussi bien pour le personnel inscrit que pour le personnel enseignant. A ce sujet, il faut savoir que tout cours est planifié plusieurs fois par an et que l'on parle plus volontiers de session. Une session est repérée par un numéro et a un intitulé (printemps, été...). La planification consistera à décider quels cours seront offerts dans une session, et pour chacun d'entre eux quelle en sera la date et la durée, puis de les inscrire au catalogue. Par ailleurs, c'est à ce moment là que l'on réservera les salles (dont on connaît le numéro et le nombre de place). Les inscriptions ou animations ou convocations ainsi que la planification se font par référence aux cours et aux sessions.

Lors de la convocation, on fait savoir la liste des ouvrages (dont on donne le numéro, la référence et l'intitulé) qu'il est conseillé de consulter pour un cours donné.

On sait aussi que certains cours nécessitent d'avoir suivi un cours de niveau inférieur au préalable. On refusera les inscriptions des personnes qui n'auraient pas ces cours en prérequis.

Ultérieurement, on veut s'assurer que les personnes inscrites et qui ont été convoquées ont été présentes ou absentes à la date convenue.

Autres exercices de synthèse sur le MLD

Reprendre les MCD des exercices de synthèse 1, 2 et 6 et construire les MLD correspondants.

MERISE
VII
MODELE PHYSIQUE DE DONNEES

Objectifs

Il s'agit maintenant de représenter les données sous une forme que puisse assimiler un logiciel informatique dans sa syntaxe propre, et que ce logiciel se charge d'implanter les données physiquement sur un ordinateur.

C'est ce que nous appellerons le passage du MLD au MPD (Modèle Physique des Données)

Nous retiendrons ici les logiciels de type SGBD relationnel tels que Ingres, Oracle, SQL-Server. Mais il est évident que la méthode Merise permet de transposer le MLD vers toutes sortes de logiciels.

Le Schéma relationnel

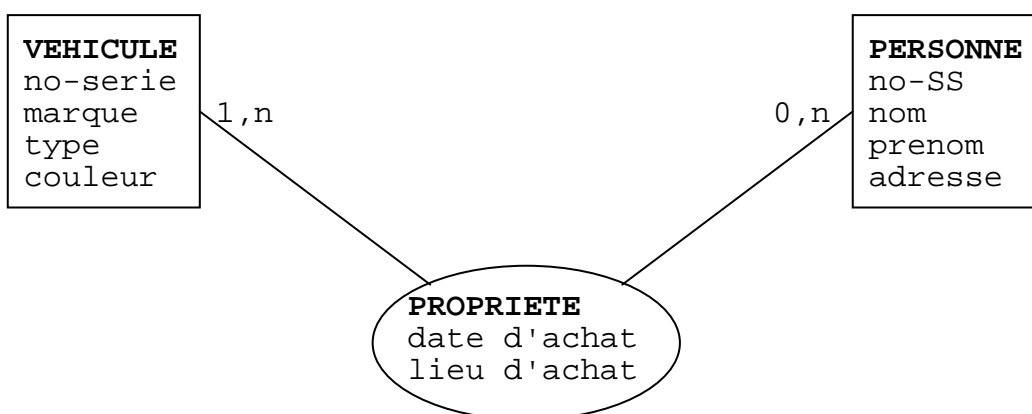
Nous supposons que ce schéma standardisé est connu du lecteur et a fait l'objet d'un apprentissage extérieur à Merise. Toutefois, afin de ne pas pénaliser les lecteurs oublieux ou ignorant ce standard, nous rappelons les notions principales avec des exemples dans un chapitre VIII joint en annexe.

De toute façon, nous nous bornerons aux notions les plus élémentaires dans un premier temps, à savoir celles que nous avons utilisées dans le chapitre I et qui nous avaient permis d'anticiper sur les finalités de la méthode.

Nous utiliserons pour l'instant les termes de : relation, attribut, clé primaire, clé étrangère.

D'ailleurs, nous allons reprendre l'exemple des personnes qui sont propriétaires de véhicules, pour revoir tout le cheminement du MCD au MLD, puis au MPD.

Le MCD représentant ce système d'information est le suivant :



Le MLD que nous en déduisons va tenir compte de ce que :

nous avons une relation PROPRIETE de type x,n - x,n qui va donner

le segment PROPRIETE avec son identifiant concaténé no-serie+no-SS

et deux sets VEHICULE/PROPRIETE et PERSONNE/PROPRIETE en sorte que notre MLD sera le suivant :



Le passage au schéma relationnel sera extrêmement simple et nous donnera 3 tables :

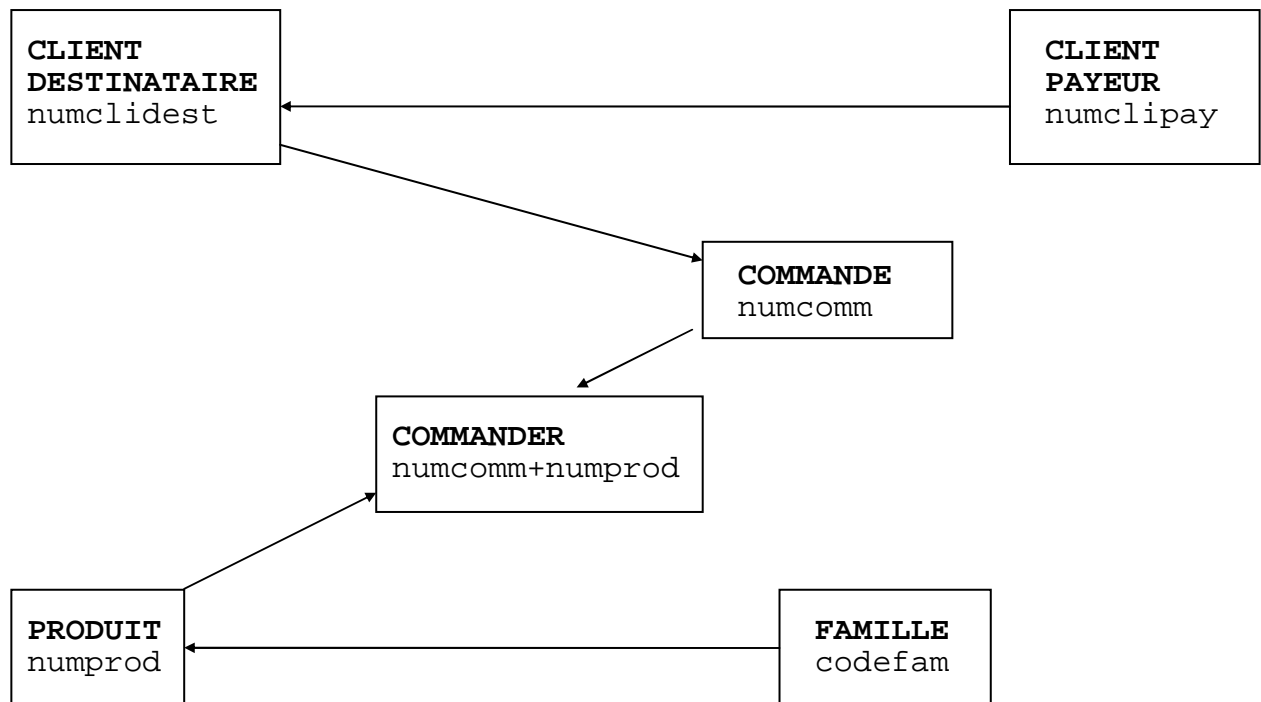
VEHICULE (<u>no-serie</u> , marque, type, couleur)
PERSONNE (<u>no-SS</u> , nom, prénom, adresse)
PROPRIETE <u>no-serie*</u> , <u>no-SS*</u> , dateachat, lieuachat)

⇒ les **attributs** sont: no-serie, marque, type, couleur ...

⇒ les **attributs soulignés** sont des **clés primaires** et ne sont autres que les identifiants dans le MCD.

⇒ les **attributs étoilés** sont des **clés étrangères** qui permettent de faire la liaison entre une relation et une autre.

Si nous reprenons le MCD du chapitre II intitulé "suivi des commandes clients", nous obtiendrons le MLD suivant :



Nous en déduisons les tables :

CLIENTPAYEUR (numclipayeur, nom, adresse)

CLIENTDESTIN (numclidest, nom, adresse, numclipayeur*)

COMMANDE (numcomm, date, numclidest*)

COMMANDER (numcomm*, numprod*, quantité)

PRODUIT(numprod, libellé, codefam*)

FAMILLE(codefam, nom).

Le MPD définitif prendra en compte la taille et le type des attributs de chaque table et les spécificités du SGBD relationnel choisi, ainsi pour l'exemple précédent, sous ORACLE :

```
CLIENTPAYEUR(numclipayeur number(4) not null,  
             nom char(20),  
             adresse char(100)  
             primary key(numclipayeur) )  
  
CLIENTDESTIN(numclidest number(4) not null ,  
             nom char(20),  
             adresse char(100),  
             numclipayeur number(4) not null,  
             foreign key (numclipayeur),  
             primary key(numclidest) )  
  
COMMANDE(numcomm number(3) not null,  
         date date,  
         numclidest number(4) not null  
         foreign key (numclidest),  
         primary key(numcomm) )  
  
COMMANDER(numcomm number(3) not null,  
          numprod number(5) not null ,  
          quantite number(8,2),  
          primary key(numcomm,numprod),  
          foreign key(numcom),  
          foreign key(numprod) )  
  
PRODUIT(numprod number(5) not null,  
        libelle char(20),  
        codefam char(4),  
        primary key(numprod) ,  
        foreign key (codefam))  
FAMILLE(codefam char(4),  
        nom char(20),  
        primary key(codefam))
```

Exercice de synthèse sur MPD

reprendre les MLD des exercices de synthèse 1, 2, 4, 6
et en déduire les schémas relationnels avec les clés primaires
et les clés étrangères

MERISE

VIII ANNEXE

Définitions du Modèle Relationnel

Domaine D_i : ensemble des valeurs possibles d'une propriété élémentaire.

exemple 1 : couleur = (blanc, bleu, vert)

2 : avion = (Airbus, B707, ATR42)

3 : salaire compris entre 3000 et 20000

Produit cartésien d'un ensemble de 2 domaines D_1 et D_2 :

c'est l'ensemble de tous les couples obtenus en combinant les valeurs prises dans D_1 avec celles prises dans D_2

exemple 4 : produit cartésien des domaines avion et couleur.

(Airbus, blanc)
(Airbus, bleu)
(Airbus, vert)
(B707, blanc)
(B707, bleu)
(B707, vert)
(ATR42, blanc)
(ATR42, bleu)
(ATR42, vert)

Généralisation : le produit cartésien de n domaines D_i est l'ensemble des $(d_1, d_2, d_3, \dots, d_n)$ de toutes les combinaisons des valeurs prises dans les domaines $D_1, D_2, D_3 \dots D_n$.

Relation R : sous-ensemble du produit cartésien de n domaines D_i , n est le degré de la relation.

exemple 5.0 :

FLOTTE (avion, couleur) est une relation de degré 2 dans le produit cartésien de l'exemple 4 : la relation est restreinte à la flotte des avions dont je dispose dans mon système d'information, à savoir :

(Airbus, bleu)
(Airbus, vert)
(B707, blanc)
(Atr42, vert)

exemple 5.1 :

EMPLOYE(matricule, nom, prenom, salaire)

de degré 4 dans le produit cartésien des domaines :

Numéro de 1 à 10000

Patronyme de 1 à 10 caractères

Prénom de 1 à 10 caractères

Salaire entre 3000f et 30000f

Attribut : rôle joué par un domaine dans une relation

exemple 6 :

VOL (Ville-départ, Ville-arrivée, Heure-départ, Heure-arrivée)

est une relation de degré 4 dans le sous-ensemble du produit cartésien : Ville, Ville, Heure, Heure

La relation Vol a été définie sur les domaines :
Ville (D1) pour la ville de départ et d'arrivée
Heure (D2) pour l'heure de départ et d'arrivée.

Les colonnes Ville-départ, Ville-arrivée, Heure-départ, Heure-arrivée ont une fonction précise dans la relation VOL et représentent ses attributs.

Cardinal d'un domaine : c'est le nombre d'éléments de ce domaine
(couleur a un cardinal de 3, avion de 3)

n-tuple : réalisation des valeurs d'attributs dans une relation

exemple 7 : (4580, dupont, georges, 8000) est un n-tuple de la relation EMPLOYE.

clé primaire : attribut ou combinaison d'attributs permettant d'identifier de façon unique chaque n-tuple.

exemple 8 : matricule est la clé primaire de la relation EMPLOYE.

Clé candidate : lorsque plusieurs attributs peuvent être choisis comme clé primaire, ceux qui ne sont pas retenus sont dits clés candidates.

exemple 9 : Nom + Prénom est une clé candidate de la relation EMPLOYE.

Table : synonyme de relation

Clé étrangère : lorsque un attribut est une clé primaire d'une autre relation, on dit qu'elle est étrangère. C'est par les clés étrangères qu'on établit les liaisons entre relations.

exemple 10 : soit les relations :

SERVICE (noserv, nomserv, chefserv)
EMPLOYE(matricule, nosecu, nom, prenom, noserv, salaire)

- ⇒ noserv est clé primaire de SERVICE
- ⇒ matricule est clé primaire de EMPLOYE
- ⇒ nosecu est clé candidate de EMPLOYE
- ⇒ nom + prenom est clé candidate de EMPLOYE
- ⇒ **noserv dans la relation EMPLOYE est la clé étrangère** qui permet de faire la liaison entre EMPLOYE et SERVICE

soit le n-tuple :

EMPLOYE(4580, 145027512002, DUPONT, Georges, 'S12', 8450)

grâce à la clé étrangère noserv de valeur 'S12', on peut rechercher l'occurrence correspondante dans la relation SERVICE et obtenir le n-tuple SERVICE ('S12', Commercial, JACKUBOVICH) et savoir ainsi que monsieur DUPONT appartient au service Commercial et que son chef s'appelle JACKUBOVICH.

Dépendance fonctionnelle : on dit que l'attribut B dépend fonctionnellement de l'attribut A si, quand on connaît A on connaît B. Bien noter que la réciproque n'est pas forcément vraie. Dans une relation, tous les attributs dépendent fonctionnellement de la clé primaire.

exemple 11 : dans la relation EMPLOYE, la connaissance du matricule implique la connaissance de son salaire.

Schéma relationnel : ensemble de plusieurs relations sémantiquement liées par leurs domaines de définitions.

Cela revient à dire qu'elles sont associées par leurs clés dans la limite des valeurs possibles du domaine de définition de ces clés, sachant que la valeur d'une clé étrangère (dans une relation R1) permet d'accéder à la valeur correspondante d'une clé primaire (dans une autre relation R2).

On conçoit aisément, dans l'exemple 10 précédent, que si 'S12' n'était pas une valeur possible du domaine de définition de l'attribut noserv, la liaison n'aurait pas pu se faire.