

MERISE PARTIE 2 MCD

DESCRIPTION STATIQUE DU SYSTEME D'INFORMATION MODELE CONCEPTUEL DE DONNEES

Le **modèle conceptuel des données** est une représentation statique du système d'information de l'entreprise qui met en évidence sa sémantique.

Il a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information.

Il s'agit donc d'une représentation des données, facilement compréhensible. Cet aspect recouvre les mots qui décrivent le système ainsi que les liens existants entre ces mots.

Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « **entité-association** ».

I. Les concepts de base

1.1) La propriété (ou attribut ou rubrique)

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

Par exemple, si l'on considère le domaine de gestion des commandes d'une société de vente par correspondance,

Les données :

« référence article »,

« désignation article »,

« prix unitaire HT »,

« taux de TVA » sont des propriétés pertinentes pour ce domaine.

La donnée « prix unitaire TTC » n'est, d'après la définition, pas une propriété car ses valeurs peuvent être retrouvées à partir des propriétés «prix unitaire HT » et « taux de TVA ».

- Chaque valeur prise par une propriété est appelée occurrence.

Des occurrences de la rubrique « désignation article » sont par exemple : « râteau », « bêche », « scie », ...

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

La propriété « **Adresse** », dont des exemples d'occurrences sont donnés ci-dessous, n'est pas élémentaire car elle peut être décomposée en trois propriétés : la rue, le code postal et la ville.

<i>Adresse</i>
<i>310, rue de la gare 16000 Angoulême</i>
<i>45, avenue de la Plage 17000 La Rochelle</i>

La **décomposition** d'une propriété en propriétés plus simples ne doit pas être systématique et doit surtout tenir compte de son l'exploitation dans le système.

Si cette exploitation est toujours **globale**, **l'atomisation n'est pas nécessaire**, dans les autres cas il faut procéder à l'isolement de chacune des composantes de la propriété et donc introduire de nouvelles propriétés.

Propriété	Occurrences
Prénom	J-Philippe, Laurent, Jean

- Une propriété **paramètre** est une propriété qui, à un instant donné, contient une **seule valeur**.
Un des exemples les plus classiques pour illustrer les paramètres est la rubrique « ValeurEuro ».

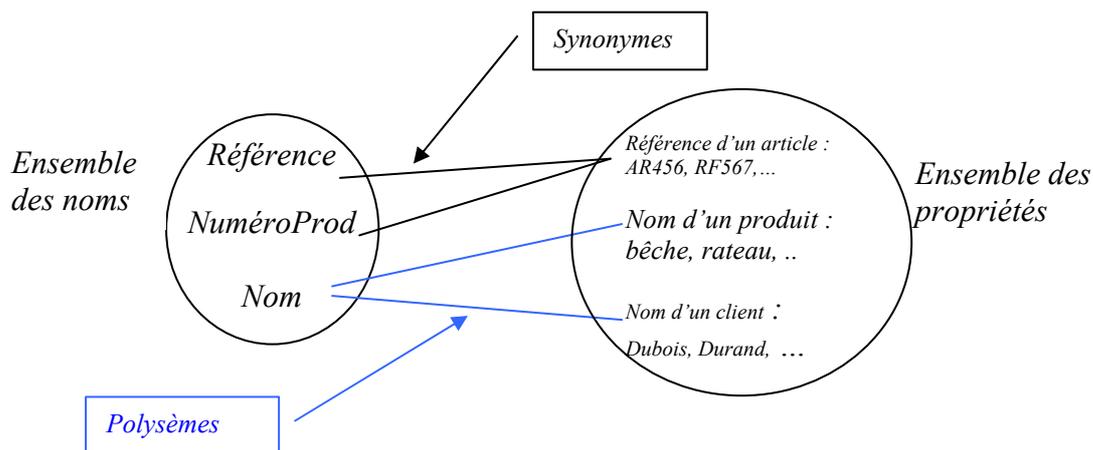
Dans le modèle conceptuel des données figurent toutes les propriétés, identifiées par un nom, qui présentent un intérêt pour le domaine à étudier.

Ce nom doit être **le plus explicite possible**.

En outre, l'identification de chaque propriété consiste à garantir une **bijection** entre l'ensemble des **noms** et l'ensemble des **propriétés** à gérer.

On devra donc **exclure**

- les **synonymes** qui correspondent à **deux noms différents** pour identifier la **même propriété**.
- les **polysèmes** qui représentent **deux propriétés différentes ayant le même nom**.



Enfin, le **principe de non-redondance** impose que chaque propriété, correctement identifiée, n'apparaisse qu'une **seule fois** dans le modèle.

1.2) L'entité ou individu-type

a) Définition

L'ENTITE est un élément **concret ou abstrait** qui a une **existence propre** et sur lequel nous souhaitons enregistrer des **informations** qui lui sont spécifiques. En général, l'entité est exprimée par un substantif (nom).

On appelle **classe d'entité** un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même. Le classement des entités au sein d'une classe s'appelle *classification* (ou *abstraction*).

Une entité est une *instanciation* de la classe.

Chaque entité est composée de propriétés, données élémentaires permettant de la décrire.

Prenons par exemple une Ford fiesta, une Renault Laguna et une Peugeot 306.

*Il s'agit de 3 entités faisant partie d'une **classe d'entité** que l'on pourrait appeler **voiture**.*

*La **Ford Fiesta** est donc une **instanciation** de la classe **voiture**.*

Chaque entité peut posséder les propriétés couleur, année et modèle.

On peut définir l'entité comme étant un **regroupement bien pensé**, de plusieurs propriétés.

*Par exemple, on considère l'entité **ARTICLE** qui regroupe les propriétés : Référence, Désignation et PrixUnitaireHT.*

Le **droit d'entrée** d'une propriété dans une entité est soumis à d'autres facteurs que le bon sens, et ce sont ces facteurs que l'on va étudier.



Les classes d'entités sont représentées par un rectangle.

Ce rectangle est séparé en deux champs:

- le champ du haut contient le libellé.

Ce libellé est généralement une abréviation pour une raison de simplification de l'écriture.

Il s'agit par contre de vérifier qu'à chaque classe d'entité correspond un

et un seul libellé, et réciproquement.

- le champ du bas contient la liste des propriétés de la classe d'entité.

Considérons deux propriétés P1 et P2.

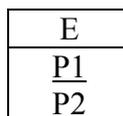
La création d'une entité E regroupant ces deux seules propriétés n'est envisageable que si l'une des deux conditions suivantes est satisfaite :

- à toute valeur de la propriété P1 doit correspondre au plus une valeur de la propriété P2.

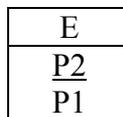
Ce fait traduit l'existence d'une dépendance fonctionnelle monovaluée entre P1 et P2

notée : $P1 \rightarrow P2$. On dit encore que P1 détermine P2.

P1 est alors rubrique identifiante de l'entité E. La représentation graphique de l'entité E a la forme suivante :



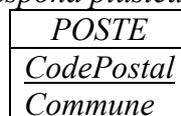
- ou à toute valeur de la rubrique P2 doit correspondre au plus une valeur de la rubrique P1. P2 est alors en dépendance fonctionnelle avec P1 et l'entité E doit être représentée ainsi :



Contre exemple :

L'entité suivante, qui peut être considérée comme un regroupement sensé, *n'est pas correcte* car il n'y a *pas dépendance fonctionnelle* entre la rubrique « CodePostal » et la rubrique « Commune ».

Ainsi au code postal « 16600 » correspond plusieurs communes : « Mornac », « Magnac »,



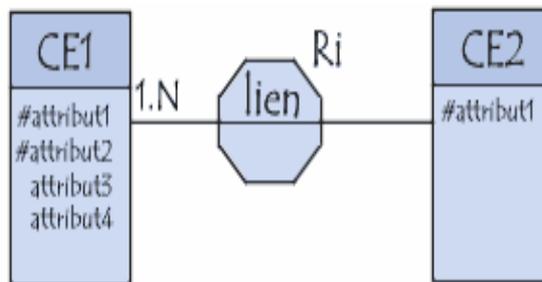
b) Les identifiants

Un **identifiant** est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité.

La définition originale est la suivante: **L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur.**

Les attributs d'une classe d'entité permettant de désigner de façon unique chaque instance de cette entité sont appelé **identifiant absolu**.

Le modèle conceptuel des données propose de **souligner les identifiants** (parfois de les faire précéder d'un #).



Ainsi, chaque classe d'entité doit posséder au moins un **attribut identifiant**, et l'ensemble de ses attributs identifiants doivent être renseignés à la création de l'entité.

c) Occurrence d'entité ou individu

D'après la définition d'une entité, on sait que la connaissance d'une valeur de la rubrique identifiante détermine la connaissance des valeurs des autres rubriques de l'entité.

L'ensemble de ces valeurs est appelé **occurrence d'entité**.

Le tableau suivant présente des exemples d'occurrences de l'entité ARTICLE.

<i>ARTICLE</i>		
<i>Référence</i>	<i>134ER</i>	<i>354TY</i>
<i>Désignation</i>	<i>Rateau</i>	<i>Bèche</i>
<i>PrixUnitaireHT</i>	<i>150 F</i>	<i>68,50 F</i>
	<i>452GT</i>	
	<i>Scie</i>	
	<i>45 F</i>	

d) Notion de dépendance fonctionnelle directe

Considérons l'entité suivante et quelques une de ses occurrences :

ARTICLE			
Référence	134ER	354TY	452GT
Désignation	Râteau	Bêche	Scie
PrixUnitaireHT	150 F	68,50 F	45F
NoCatégorie	A	A	B
LibelléCatégorie	Jardinage	Jardinage	Bricolage

Cette entité est juste mais elle implique une redondance d'information relative à la catégorie.

L'association entre le numéro de la catégorie et son libellé est en effet répétée dans chaque occurrence de l'entité ARTICLE.

Pour supprimer de telles redondances, on devra veiller à ce que toute dépendance fonctionnelle entre la propriété identifiante de l'entité et une propriété non identifiante de l'entité soit directe.

Une dépendance fonctionnelle monovaluée $x \rightarrow y$ est directe s'il n'existe pas de propriété z telle que : $x \rightarrow z$ et $z \rightarrow y$.

Dans l'exemple précédent la dépendance fonctionnelle Référence \rightarrow LibelléCatégorie n'est pas directe car il existe la propriété NoCatégorie telle que :

Référence \rightarrow NoCatégorie *et* NoCatégorie \rightarrow LibelléCatégorie

Exercice n°1 (Acquis : Propriété et Entité)

La société Azur-Hebdo consacre l'essentiel de son activité à l'édition et à la distribution d'un journal spécialisé dans les petites annonces et la publicité dans le Sud-est de la France. La parution du journal est hebdomadaire et sa distribution est assurée uniquement dans les départements des Alpes maritimes et du Var.

La tarification d'une annonce est fournie ci-dessous :

- Première semaine de parution : tarif pour 5 lignes au plus :

Rubrique de l'annonce	Prix
Emploi	50 F
Bourse aux affaires	40 F
Tout ce qui roule	55 F
Immobilier	55 F
Contacts	75 F
Loisirs	50 F

- Options

Prix de la ligne supplémentaire : 50 F

Domiciliation (pour préserver l'anonymat du client) : 80 F

Semaines supplémentaires : des réductions sont accordées selon le tableau ci-dessous :

Période	Pourcentage de réduction *
2^{ème} semaine	20%
3^{ème} semaine et suivantes	40 %

*Ces réductions sont applicables au prix de base de la première semaine.

Le tableau ci-dessous répertorie un ensemble de données qui se rapporte à la gestion des annonces (La liste est triée sur le nom de la donnée).

Nom	Signification
CP	Code postal du client qui dépose l'annonce
DateRedac	Date de dépôt de l'annonce
Dom	Domiciliation de l'annonce déposée(oui/non)
NbLignes	Nombre de lignes de l'annonce déposée
NbSem	Nombre de semaines de parution de l'annonce déposée
Nom	Nom du client qui dépose l'annonce
NumAnn	Numéro qui identifie chaque annonce déposée
NumCli	Numéro qui identifie chaque client qui dépose une annonce
Prénom	Prénom du client qui dépose l'annonce
Prix	Prix de l'annonce déposée

Rub	Rubrique de l'annonce déposée : Emploi, Contacts
Rue	Première partie de l'adresse du client qui dépose une annonce
TarifDom	Tarif de la domiciliation
TarifSup	Tarif de la ligne supplémentaire
TarifPrem	Tarif de la première semaine
Texte	Texte de l'annonce
Ville	Ville de l'adresse du client qui dépose l'annonce

1. Indiquer, parmi les données ci-dessus, celles qui peuvent être qualifiées de propriétés.

Parmi la liste des données, on recherche les informations non déductibles d'autres informations et qui ont un intérêt pour le domaine étudié. Or, parmi cette liste, **deux** données seulement **ne peuvent pas être considérées comme des propriétés**:

- NbLignes: donnée qui peut être déduite de la propriété Texte et éventuellement d'une propriété paramètre fournissant le nombre de caractères par ligne,
- **Prix** : donnée calculée à partir des propriétés TarifDom, TarifSup, TarifPrem.

2. En vous basant sur les éléments de tarification, fournir l'ensemble des occurrences des propriétés : TarifPrem et TarifSup. Parmi ces deux propriétés laquelle est une propriété paramètre ?



3. On considère l'ébauche du modèle conceptuel de données suivante :

ANNONCE
<u>NumAnn</u>
Texte
NbLignes
Dom
NbSem
DateRédac
Rub
TarifPrem

CLIENT
<u>NumCli</u>
Nom
Prénom
Rue
CP
Ville

3.1 On considère l'événement suivant :

Figeac Claire qui habite : 72, Avenue de la gare 05000 NICE dépose le 14/02/2001 l'annonce ci-après :

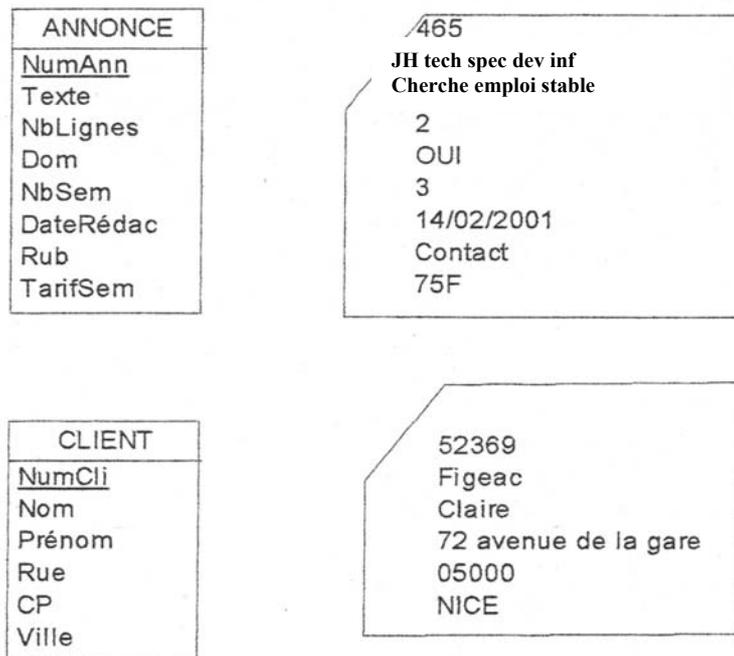
Annnonce n° 465

JH tech spec dev inf
cherche emploi stable

Ecrire sous référence 52369 au journal

(Nombre de semaines de parution : 3)

En limitant le système d'information à cette annonce, fournir les occurrences de l'entité CLIENT et ANNONCE.



3.2 Les dépendances fonctionnelles issues de la conception de l'entité ANNONCE sont-elles toutes directes ? justifier votre réponse.

La dépendance fonctionnelle NumAnn ---> TarifPrem n'est pas une dépendance fonctionnelle directe.

Elle peut, en effet, être retrouvée par transitivité grâce aux deux dépendances fonctionnelles élémentaires suivantes: NumAnn --->Rub et Rub--->TarifSem

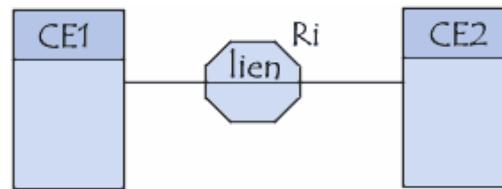
Pour corriger ce **MCD**, il faut supprimer la dépendance fonctionnelle directe tout en maintenant les deux dépendances fonctionnelles NumAnn ---> Rub et Rub ---> TarifPrem.

L'idée est de créer une **nouvelle entité RUBRIQUE** dont Rub est la propriété identifiante. Pour garder la dépendance fonctionnelle NumAnn ---> Rub il faut relier les **entités Rubrique et Annonce par des associations**.

1.3) L'association (ou relation-type)

a) Définition

Une association (appelée aussi parfois *relation*) est un lien sémantique entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité).



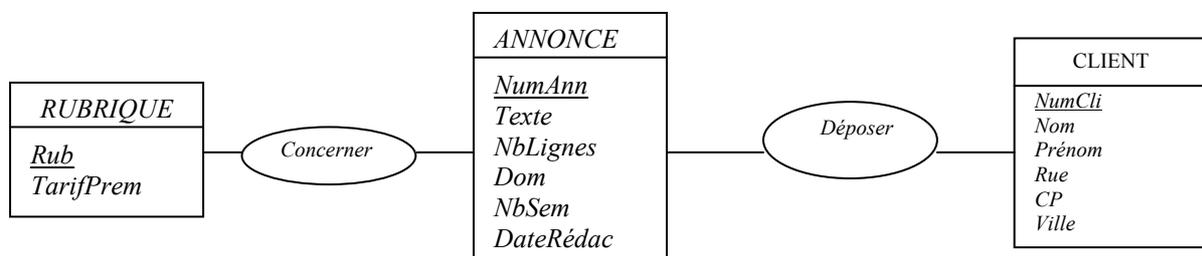
Une classe de relation peut lier plus de deux classes d'entité. Voici les dénominations des classes de relation selon le nombre d'intervenants:

- une classe de **relation récursive** (ou *réflexive*) relie la **même classe** d'entité
- une classe de **relation binaire** relie **deux classes** d'entité
- une classe de relation **ternaire** relie **trois classes** d'entité

Une classe de relation **n-aire** relie **n classes** d'entité. Les classes de relations sont représentées par des hexagones (parfois des ellipses) dont l'intitulé décrit le type de relation qui relie les classes d'entité (**généralement un verbe**).

On définit pour chaque classe de relation un **identificateur** de la forme R_i permettant de désigner de façon unique la classe de relation à laquelle il est associé. On peut éventuellement ajouter des **propriétés** aux classes de relation.

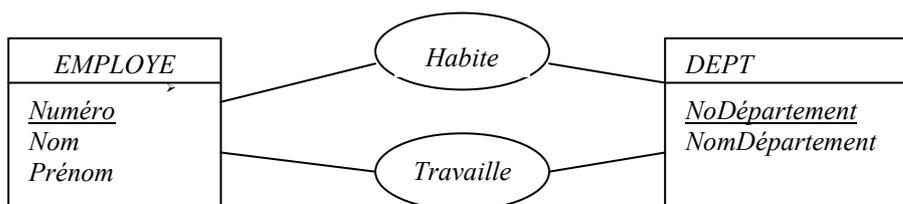
Par exemple, dans le modèle conceptuel de données relatif à la gestion des annonces on introduit les deux associations « Déposer » et « Concerner » afin d'exprimer les réalités suivantes : un client dépose une annonce et une annonce concerne une rubrique.



Il peut y avoir, entre deux mêmes entités, plusieurs associations qui représentent chacune des réalités différentes.

Dans le modèle conceptuel des données ci-dessous, l'association Habite indique le département dans lequel l'employé habite et l'association Travaille fournit le département dans lequel il travaille.

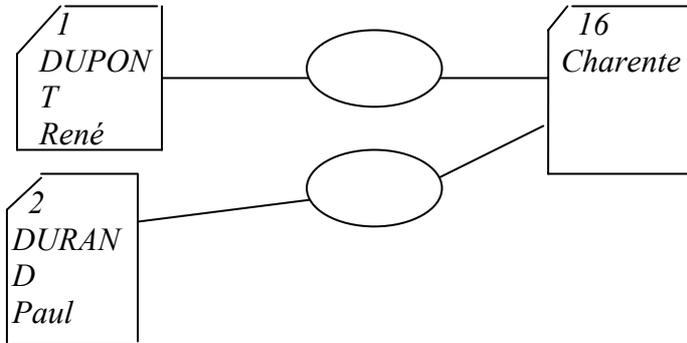
Pour certains employés ces deux départements peuvent être identiques.



L'ensemble des entités participant à une association est appelé collection de cette association. *La collection de l'association HABITE est formée des entités : EMPLOYE et DEPT.*

b) Occurrence d'association

Une occurrence d'association est un lien particulier qui relie deux occurrences d'entités. *Le schéma ci-dessous présente deux exemples d'occurrences de l'association « Habite ».*



Remarque : certains auteurs définissent l'identifiant d'une association comme étant la concaténation des identifiants des entités qui participent à l'association.

c) Cardinalité

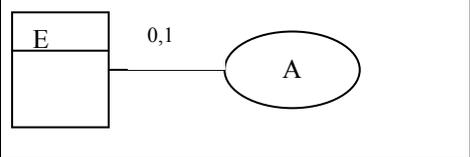
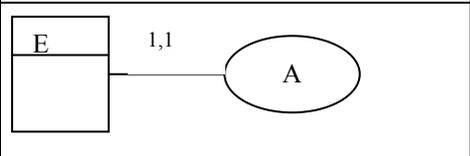
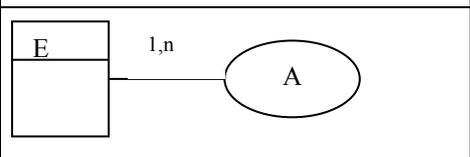
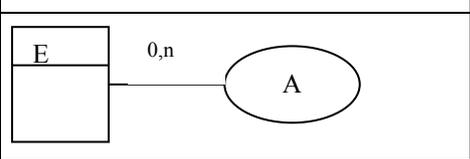
Les **cardinalités** permettent de caractériser le **lien** qui existe entre une **entité et la relation** à laquelle elle est reliée.

La cardinalité d'une relation est composé d'un **couple** comportant une **borne maximale** et une **borne minimale**, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur:

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

Un couple de cardinalités placé entre une entité E et une association A représente le nombre minimal et maximal d'occurrences de l'association A qui peuvent être « attachées » à une occurrence de l'association E.

Le tableau ci-après récapitule les valeurs que peut prendre ce couple.

	<p>Pour chaque occurrence de E, le modèle admet :</p> <ul style="list-style-type: none"> - soit l'absence de lien - soit la présence d'un seul lien
	<p>Pour chaque occurrence de E le modèle admet la présence d'un et un seul lien</p>
	<p>Pour chaque occurrence de E le modèle admet la présence d'un seul ou de plusieurs liens</p>
	<p>Pour chaque occurrence de E le modèle admet :</p> <ul style="list-style-type: none"> - soit l'absence de lien - soit la présence de plusieurs liens

Remarque : dans certaines situations, la lettre n peut être remplacée par une valeur.

Exercice n°2 (Acquis : propriété, entité et association)

Le système d'information étudié concerne l'activité de gestion des locations saisonnières d'une agence immobilière. Une analyse de l'existant a permis de dégager les entités suivantes :

Entité	Objectif	Propriétés
PROPRIETAIRE	Regroupe toutes les informations relatives aux propriétaires d'appartements	NumPropriétaire Nom Prénom Adresse1 Adresse2 CodePostal Ville NumTel1 NumTel2 E-mail CAcumulé
APPARTEMENT	Regroupe toutes les informations des appartements meublés mis à la location	NumLocation Catégorie : 1, 2, ou 3 étoiles Type : T2, T3, T4 NbPersonnes AdresseLocation Photo Equipements
LOCATAIRE	Regroupe toutes les informations sur les locataires qui ont effectué au moins une location par l'intermédiaire de l'agence	NumLocataire NomLocataire PrénomLocataire Adresse1Locataire Adresse2Locataire CodePostalLocataire VilleLocataire NumTel1Locataire NumTel2Locataire E-mailLocataire
CONTRAT	Regroupe toutes les informations relatives à une location qui va avoir lieu ou qui a actuellement lieu. Une location s'étend éventuellement sur plusieurs semaines consécutives.	NumContrat Etat : réservé, confirmé, soldé DateCréation DateDébut DateFin
TARIF	Regroupe les informations liées à la tarification	CodeTarif PrixSemHS (prix semaine haute saison) PrixSemBS (prix semaine basse saison)

1. Pourquoi l'information CAcumulé de l'entité PROPRIETAIRE est-elle une propriété ?

Bien que pouvant être une donnée calculée par le système d'information, il est préférable d'avoir le chiffre d'affaire cumulé en propriété de manière à pouvoir accéder plus rapidement à

l'information (allègement des traitements). C'est une "dénormalisation".

2. La propriété Equipements est destinée à décrire les principaux équipements de l'appartement : téléviseur, lave-vaisselle, ... Quels sont les inconvénients liés à une telle propriété ?

Le fait de regrouper toutes les caractéristiques des équipements dans une même propriété ne permettra pas de faire des traitements précis sur ces derniers. *Par exemple, il sera impossible de lister tous les appartements de type T2 qui possède un lave-vaisselle.*

L'isolement des différents composants peut être réalisé de deux manières;

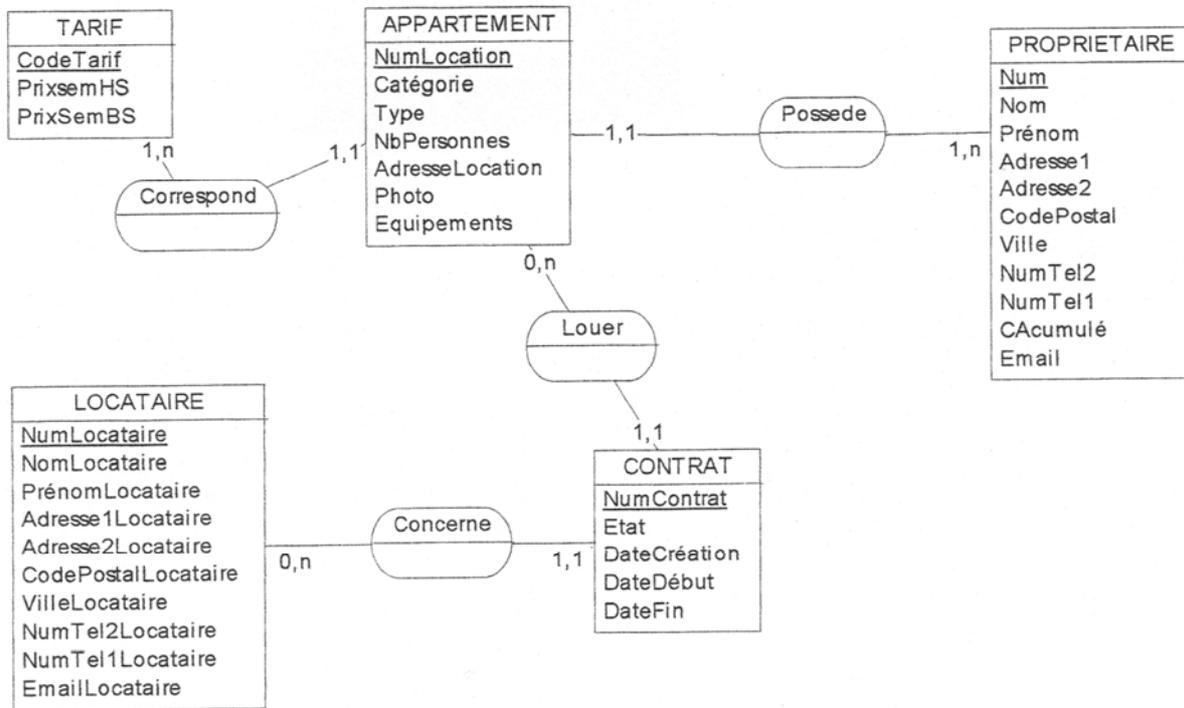
- Introduire dans l'entité APPARTEMENT des propriétés booléennes telles que Téléviseur, Lave-vaisselle etc.
- Créer une entité TYPE-EQUIPEMENT et mettre en place une association (m-n) entre l'entité APPARTEMENT et l'entité TYPE-EQUIPEMENT.

3. Présenter le modèle conceptuel des données décrivant ce système d'information en tenant compte des règles de gestion suivantes :

- La notion de co-propriété ne doit pas être prise en compte ce qui revient à dire que tout appartement appartient à un et un seul propriétaire.

- A tout appartement correspond un code tarif

Seules les noms des entités figureront sur le modèle.



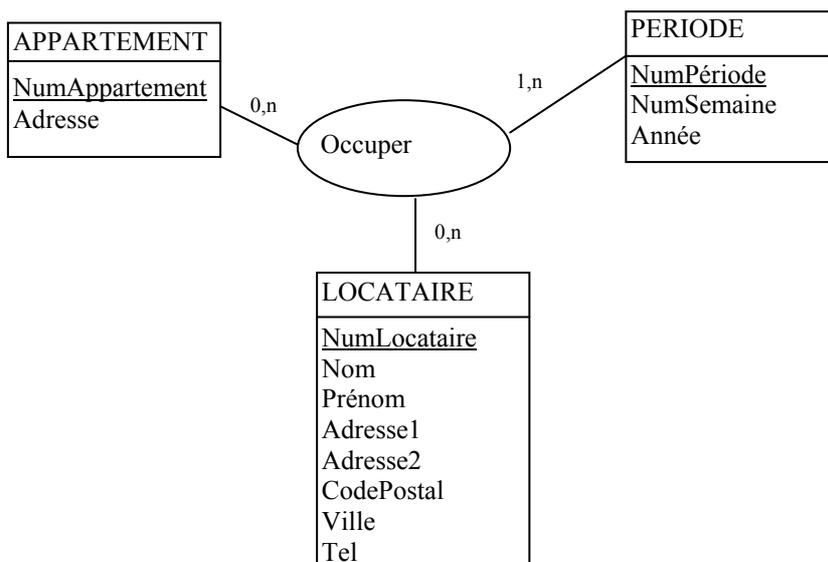
4. On restreint le domaine étudié à la gestion des locations des appartements possédés par M. X. Les entités recensées sont données ci-dessous :

Entité	Objectif	Propriétés
APPARTEMENT	Regroupe toutes les informations relatives aux appartements de M. X	<u>NumAppartement</u> Adresse
PERIODE	Cette entité admet une occurrence par semaine réservée ou occupée	<u>NumPériode</u> NumSemaine Année
LOCATAIRE	Regroupe toutes les informations sur le locataire	NumLocataire Nom Prénom Adresse1 Adresse2 CodePostal Ville Tel

Pour une semaine donnée, un appartement de M. X peut être :

- soit réservé ou occupé par un locataire
- soit libre
- soit indisponible (ce cas correspond à l'occupation de l'appartement par M.X)

Discuter la proposition de modélisation suivante qui est destinée à représenter l'occupation des appartements de M. X :



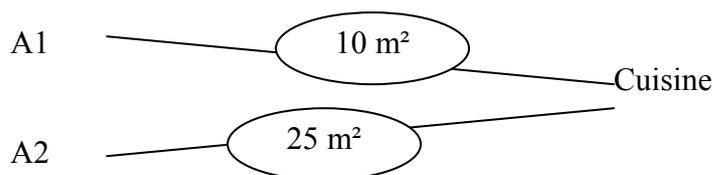
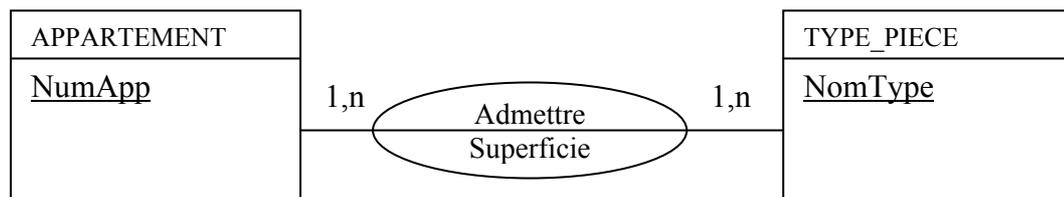
Cette représentation **n'est pas satisfaisante** car elle admet que le locataire 2205 occupe l'appartement n°1 pour la période 3 et que le locataire 1903 occupe également l'appartement n°1 pour la période 3.

Le système d'information n'accepte pas qu'un appartement soit loué pour la même période à deux locataires différents.

Il faut pouvoir représenter le fait que pour un couple d'occurrence APPARTEMENT-PERIODE, on n'admette qu'une et une seule occurrence de LOCATAIRE.

5. On souhaite décrire pour chaque appartement les différentes pièces qui le composent ainsi que leur superficie. *Par exemple : l'appartement n° 345 possède une kitchenette de 4 m2, une salle de bains de 4 m2, un séjour de 20 m2 et une terrasse de 5m2.*

Enrichir le modèle conceptuel afin de représenter une telle réalité



d) Caractéristiques d'une association

➤ La dimension d'une association

La dimension d'une association indique le nombre d'entités participant à l'association. Les dimensions les plus courantes sont 2 (association binaire) et 3 (association ternaire) :

- L'association binaire exprime la présence de liens sémantiques entre les occurrences d'une entité A et les

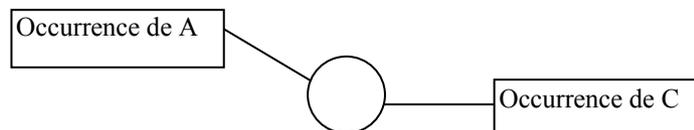
occurrences d'une entité B. *L'association « Habite » entre les EMPLOYE et HABITE est binaire.*

- L'association ternaire exprime la présence de liens sémantiques entre les occurrences de 3 entités.

Remarques :

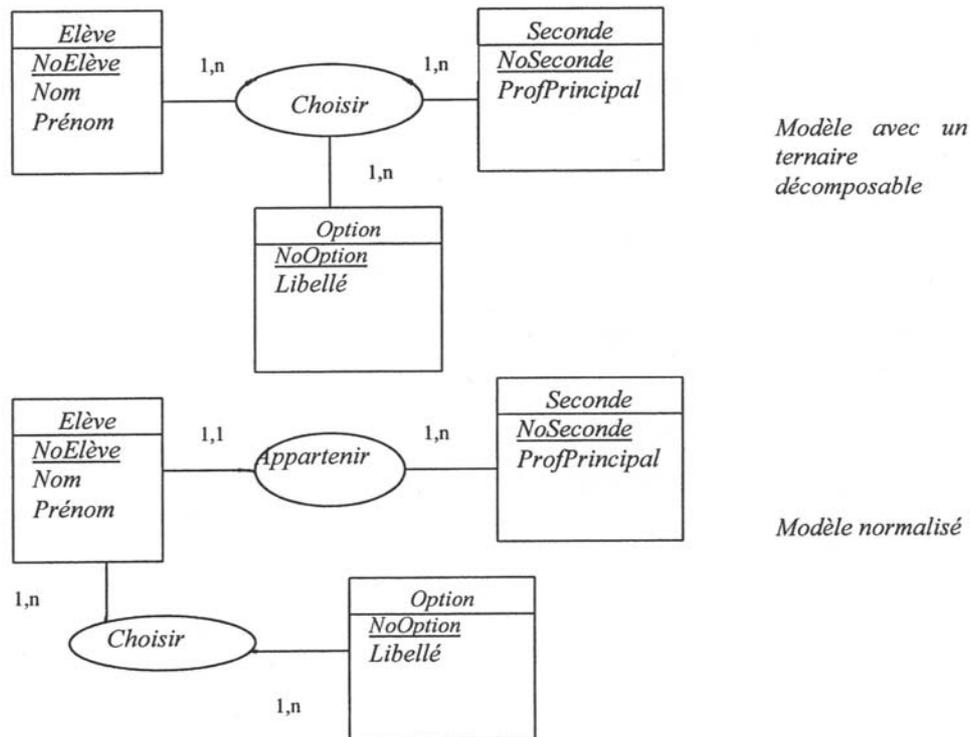
- Toute occurrence d'une association de dimension n doit être reliée à n occurrences d'entités.

Par exemple, pour une association ternaire dans laquelle participent trois entités « A », « B » et « C », toute occurrence doit être reliées à 3 occurrences des entités respectives A, B et C. On ne peut donc pas avoir une occurrence à 2 pattes de la forme ci-dessous.



- L'opération de décomposition consiste à éclater une relation de dimension n en une ou plusieurs associations de dimension moindre sans perte de sémantique.

Dans l'exemple ci-dessous , qui se rapporte à la gestion des options en classe de seconde, l'association ternaire du premier modèle doit être est éclatée en deux associations binaires afin d'aboutir au second modèle.



Les associations figurant dans le modèle conceptuel des données devront être non décomposables. Pour ce faire on recherchera en priorité les associations binaires puis les ternaires.

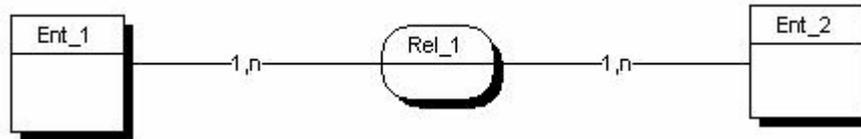
➤ La fonctionnalité d'une association binaire

Les différents types de fonctionnalités d'une association binaire A définie entre deux entités E1 et E2 sont les suivants :

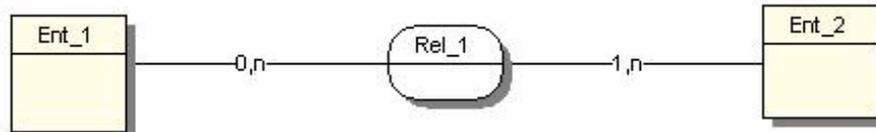
- un à un (**1 – 1**) caractérisé par la cardinalité **maximum** égale à 1 sur les deux segments « E1 -- A » et « E2 – A »,
- un à plusieurs (**1 – n**) caractérisé par la cardinalité **maximum** égale à 1 sur l'un des segments et la cardinalité maximum égale à n sur l'autre,
- plusieurs à plusieurs (**m-n**) caractérisé par la cardinalité **maximum** égale à n sur les deux segments « E1 – A » et « E2 – A ».

➤ Les notions de partialité et de totalité

Une association binaire A définie entre les entités E1 et E2 est **totale** si elle est caractérisée par la cardinalité **minimale** égale à **un** sur les **deux segments** « E1 – A » et « E2 – A ».



Une association binaire A définie entre les entités E1 et E2 est **partielle** si elle est caractérisée par la cardinalité **minimum** égale à 0 sur l'un des segments.



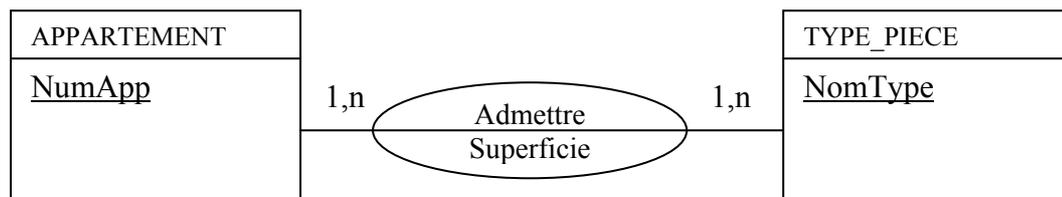
e) L'association porteuse

Les propriétés qui dépendent fonctionnellement de plusieurs identifiants d'entités sont portées par les associations entre ces entités.

C'est une dépendance fonctionnelle multi attributs au niveau de la source.

Dans le système d'information de l'exercice n° 2 la propriété Superficie est portée par l'association qui relie l'entité APPARTEMENT à l'entité TYPE_PIECE car pour un appartement et une pièce on a une et une seule superficie.

Par exemple pour l'appartement 134 et la pièce de type kitchenette on a la superficie : 3 m².



Le droit d'entrée d'une propriété P dans une association reliant n entités est donc soumis à l'existence de la dépendance fonctionnelle suivante : $I_1, \dots, I_n \rightarrow R$

où I_1, I_2, \dots représentent l'identifiant de chacune des entités qui participent à l'association.

Pour éviter toute redondance, on s'assurera en outre que la dépendance fonctionnelle est élémentaire.

Les associations porteuses sont donc toujours de type **(m-n)**.

On dit qu'une propriété est en dépendance fonctionnelle élémentaire avec une liste de rubriques LR :

- si elle est fonctionnellement dépendante de LR,
- si elle n'est pas fonctionnellement dépendante d'une sous-liste de LR.

La dépendance fonctionnelle suivante :

NoAppart, NomType, NumPropriétaire → Superficie n'est pas élémentaire

car il existe la sous-liste NumAppart, NomType telle que :

NumAppart, NomType → Superficie.

Auto-évaluation n° 1:

Pour faciliter les contacts entre ses différents employés, la société X a confié à son service informatique la réalisation d'un annuaire électronique.

L'une des spécifications de cette application précise qu'elle devra être accessible à l'ensemble du personnel par l'intermédiaire de l'INTRANET de l'entreprise.

Un extrait de l'entretien avec le demandeur de l'application M. D est reproduit ci-dessous :

L'analyste : Comment sont identifiés les salariés de la société ?

M. D : Chaque employé de la société est identifié par un numéro et appartient à un service caractérisé par un nom. Comme exemples je peux vous citer le service comptabilité ou le service production.

L'analyste : Peut-il y avoir deux services qui portent le même nom ?

M. D : Non.

L'analyste : Quelles sont les données relatives à un salarié que vous souhaitez déposer sur l'annuaire

M.D : son nom, son prénom, ses coordonnées téléphonique et INTERNET ainsi que le service et la section auquel il est rattaché. Je précise que certains services sont découpés en sections, elles aussi identifiées par un nom. Par exemple, le service informatique comprend les sections études et production. Le service comptabilité n'admet pas de section.

L'analyste : Tout salarié a-t-il un poste téléphonique ?

M.D : non, certains employés n'ont pas encore de poste téléphonique. Pour les autres ils en ont un et un seul. C'est un numéro interne composé de 4 chiffres. Le mien est par exemple 48 14.

L'analyste : Je suppose que tous les salariés n'ont pas une adresse électronique.

M.D : Sur notre INTRANET tous les employés ont une adresse professionnelle qui leur permet d'envoyer ou de recevoir du courrier électronique relatif à leur activité.

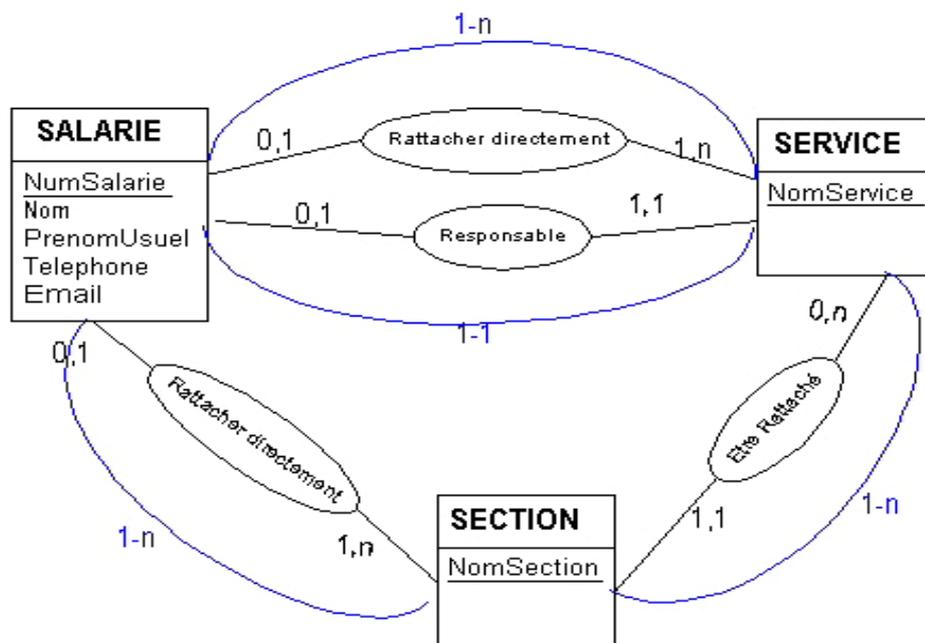
L'analyste : Revenons sur les sections. Une section peut-elle concerner plusieurs services ?

M. D : Non, une section concerne un et un seul service.

L'analyste : En terme de traitements quelles sont vos attentes ?

M.D : Ma première attente serait bien évidemment de retrouver très rapidement les coordonnées d'un employé. Dans le cas où l'employé n'a pas de numéro de téléphone, il serait souhaitable d'afficher le nom ainsi que le numéro de téléphone du responsable du service auquel il appartient. Je précise qu'il y a un seul responsable par service.

1. Proposer un modèle conceptuel des données pour la conception de cette application
2. Indiquer pour chacune des associations son type : 1 - 1, 1-n ou m-n



Notons qu'une association est partielle si une des cardinalité est à 0. Donc ici, elles sont toutes partielles.

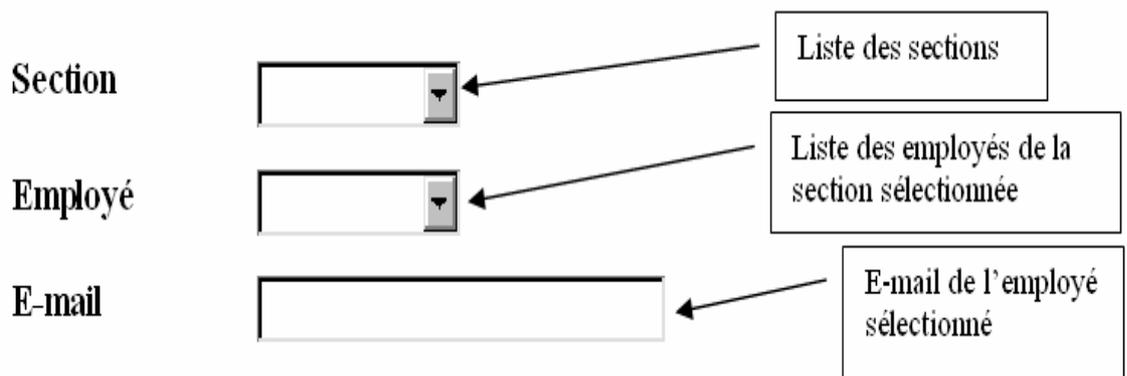
3. Lister toutes les incohérences acceptées par votre MCD

- Un salarié peut n'appartenir à rien (ni section, ni service),
- Un salarié peut être responsable d'un autre service que celui dont il fait partie.

Il faut donc mettre en place des contraintes applicatives:

- Un responsable d'un service doit être "Rattaché directement" au même service,
- Tout salarié est rattaché soit à une section, soit à un service.

4. Faire apparaître sur le MCD le chemin emprunté lors de la mise œuvre du traitement suivant :

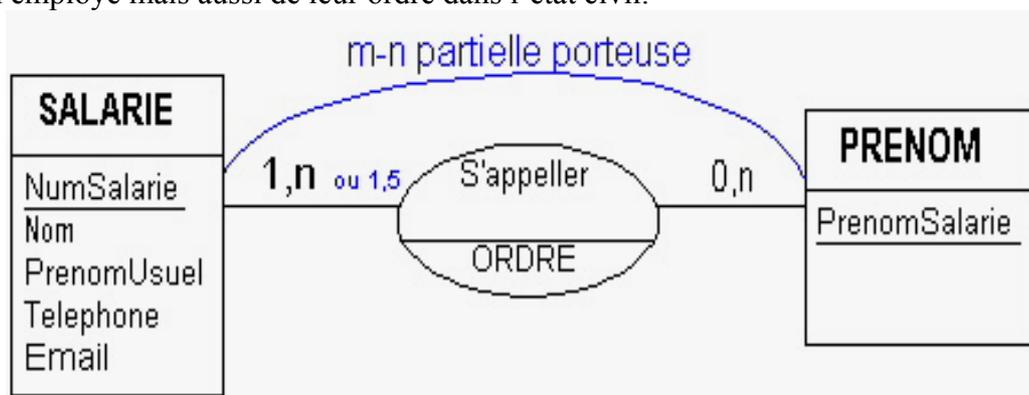


Section: On prend toutes les occurrences de l'entité SECTION,

Employé: Cardinalité 1,n pour avoir les occurrences de SALARIE concernées,

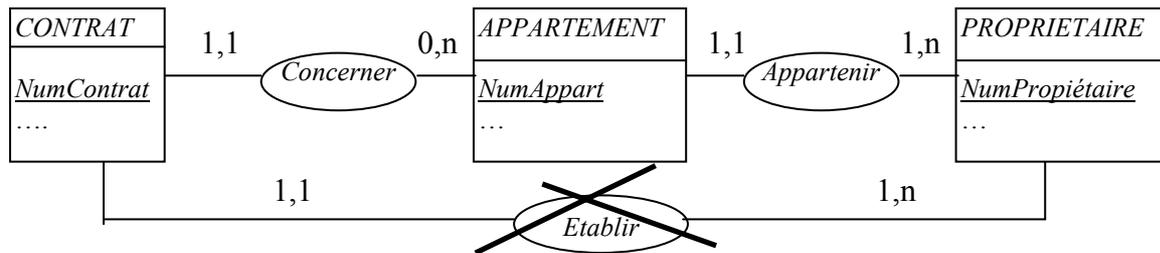
E-mail: Cardinalité 0,1 pour avoir la propriété du SALARIE concerné.

5. On souhaite modifier la modélisation de façon à permettre la mémorisation de tous les prénoms d'un employé. Certains, aux parents forts imaginatifs, admettent jusqu'à 5 prénoms ! Proposer une solution qui permet l'enregistrement non seulement des prénoms d'un employé mais aussi de leur ordre dans l'état civil.



f) Les associations transitives

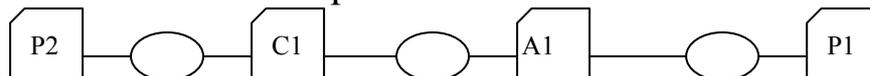
Considérons le modèle suivant :



L'association binaire qui relie l'entité « **CONTRAT** » et l'entité « **PROPRIETAIRE** » doit être ôtée du modèle car on peut retrouver le *propriétaire* à partir des associations « *Concerner* » et « *Appartenir* ».

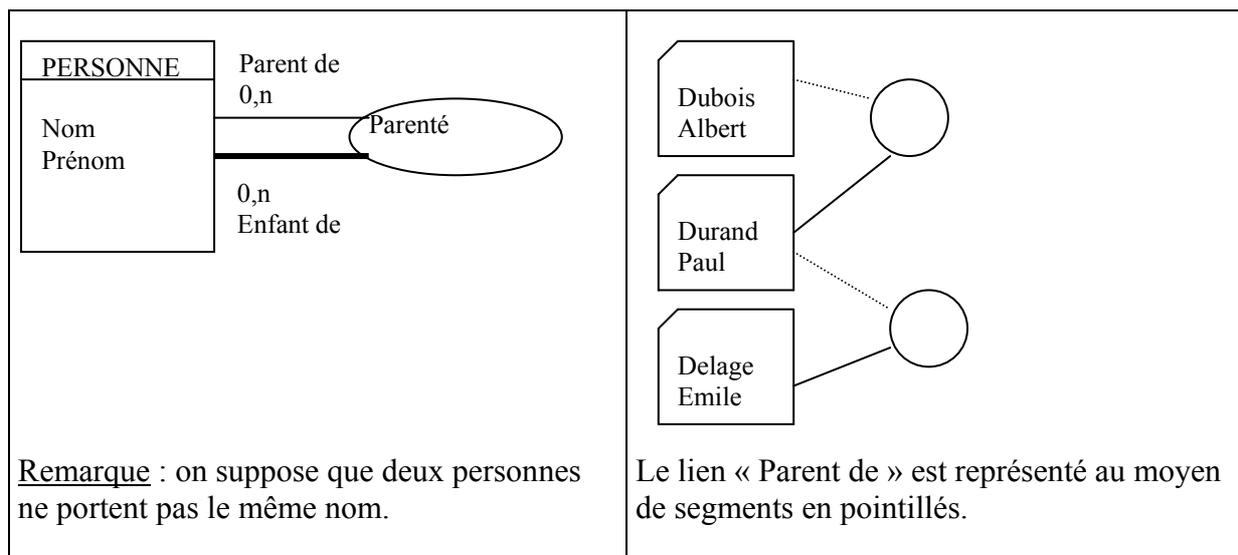
Il s'agit d'une association transitive. Seules les dépendances fonctionnelles directes entre identifiants d'entités devront donner lieu à des association binaires de type (1-n).

On veillera à supprimer les dépendances transitives pour ne pas surcharger le MCD car on risque de faire des incohérences comme:



g) Les associations réflexives

L'association réflexive est une association binaire qui relie une entité à elle-même. Une occurrence de l'association établit donc un lien entre une occurrence de l'entité et une autre occurrence de cette même entité. Dans le cas d'une association non symétrique, on doit faire porter le rôle sur chacun des segments, *comme l'illustre l'exemple suivant*.



Exercice n°3 (Acquis : Concepts de base uniquement)

On considère l'univers du discours suivant :

« Le personnel du service informatique de la MAAF peut être réparti en

deux catégories :

- le personnel interne qui regroupe les employés rémunérés par la MAAF,
- le personnel extérieur qui englobe des salariés de différentes SSII

Excepté le directeur du service, chaque employé (extérieur ou interne) est

encadré par un supérieur hiérarchique qui est obligatoirement un salarié interne.»

Relativement à la gestion du personnel du service informatique, on retient

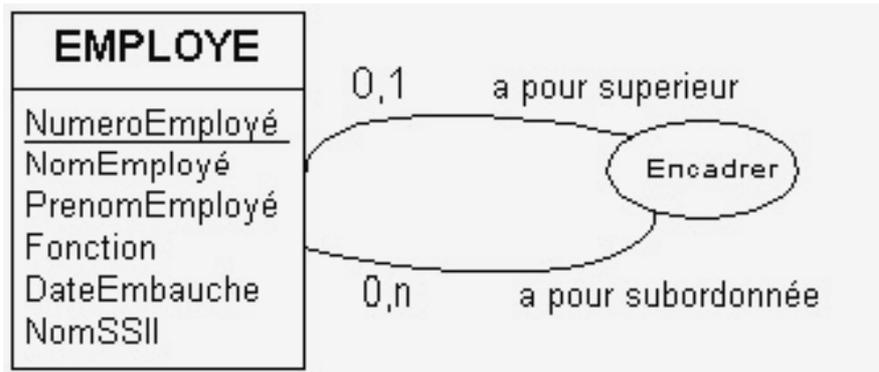
la liste des données suivantes :

NuméroEmployé	numéro qui identifie chaque employé travaillant au service informatique
NomEmployé	nom de l'employé
PrénomEmployé	prénom de l'employé
Fonction	fonction occupée par l'employé : Analyste-programmeur, chef de projet, ...
DateEmbauche	date de l'embauche. Propriété uniquement définie pour les salariés internes
NomSSII	nom de la société de services. Cette propriété est uniquement définie pour les salariés extérieurs

Proposer un modèle conceptuel de données décrivant cette réalité et permettant

de répondre à des requêtes telles que :

- Quel est le nom du supérieur hiérarchique de M.X?
- Quelle est la liste des salariés internes?
- Quelle est la liste des employés ayant une fonction d'encadrement?



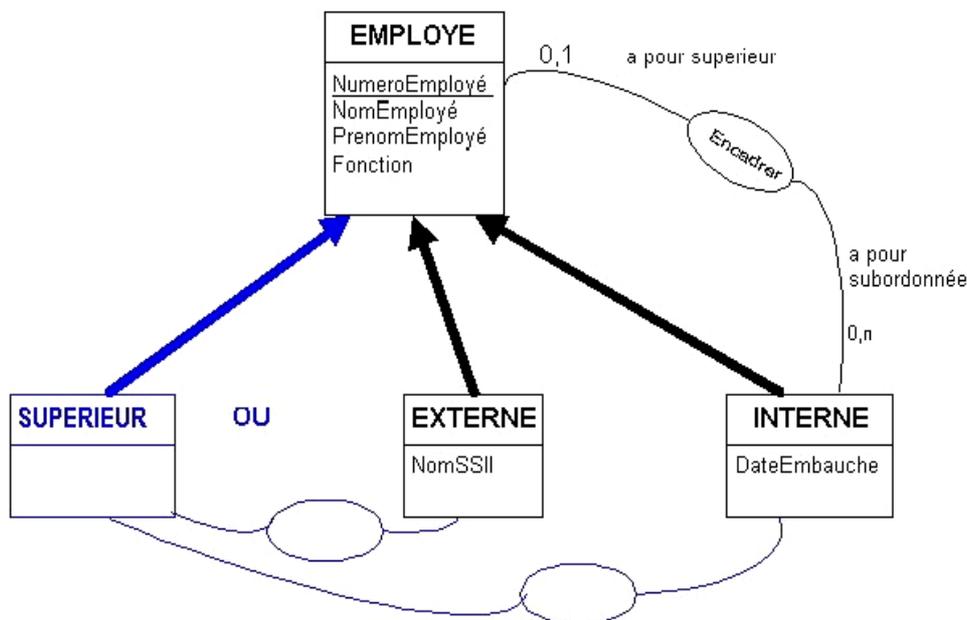
Contrainte applicative: Un employé doit avoir un supérieur "interne".

Contrainte d'exclusion: sur DateEmbauche et NomSSII.

Niveau sémantique: la date d'embauche n'a pas de sens pour un employé externe (idem pour NomSSII et un interne).

Considération physique: optimisation mémoire si NomSSII vide.

Solution par l'utilisation de Merise 2:



Auto-évaluation n° 2:

Dans le cadre de l'automatisation du suivi annuel de formation du personnel de la société x, le responsable des ressources humaines a défini pour chaque poste de l'entreprise, les compétences requises.

Par exemple, le poste « infographiste » requiert les compétences: créativité, connaissances techniques et aptitudes relationnelles. Outre cette définition de poste, chaque salarié réalise en fin d'année un bilan de compétences destiné à connaître pour chaque compétence exigée par son poste son niveau actuel. Le tableau suivant récapitule les compétences du salarié n°1325.

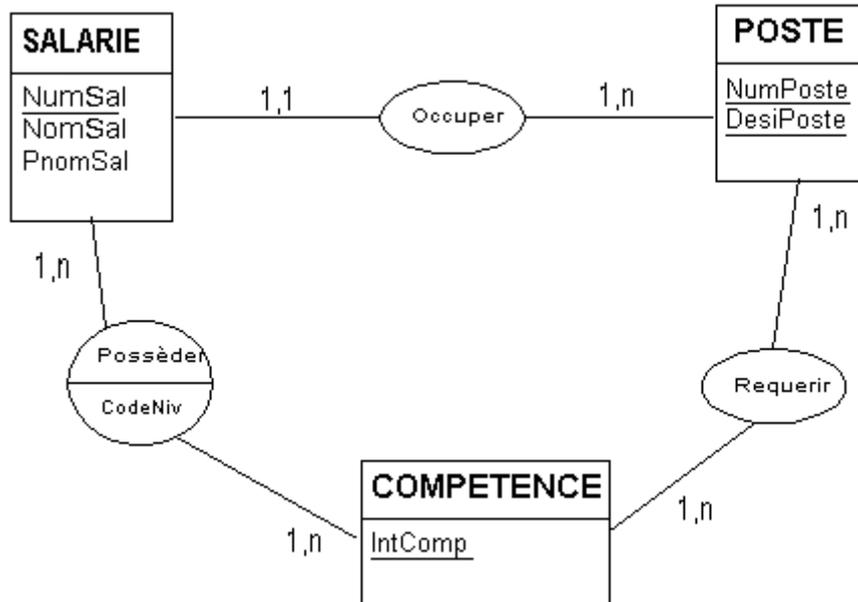
N° salarié: 1325	Poste: infographiste		
Nom: Dubois	Prénom: Michel		
Intitulé des compétences	Niveau actuel		
	A	B	C
Connaissances techniques	X		
Créativité		X	
Aptitudes relationnelles		X	

Les niveaux de compétence sont codés ainsi:

- A : compétence confirmée
- B : compétence à renforcer
- C : compétence en cours d'acquisition

1. Proposer un modèle conceptuel des données permettant de structurer les propriétés figurant dans le tableau suivant:

Nom propriété	Définition
NumSal	Numéro du salarié
NomSal	Nom du salarié
PnomSal	Prénom du salarié
NumPoste	Numéro du poste
DésiPoste	Désignation du poste
<u>IntComp</u>	Intitulé de la compétence
<u>CodeNiv</u>	Code du niveau: A, B ou C



Notons que le couple (NumSal, IntComp) détermine un et un seul CodeNiv.

2. Indiquer les incohérences éventuellement admises par la modélisation proposée.

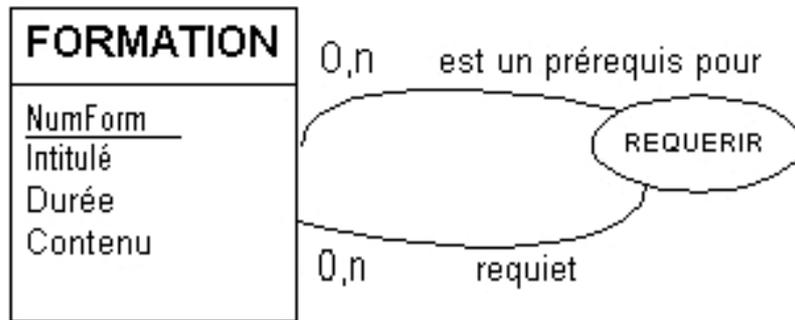
Un salarié peut occuper un poste sans posséder une compétence requise alors qu'il devrait l'avoir avec un certain niveau.

3. Afin de permettre à chaque employé d'évoluer dans son poste ou sur un autre poste, l'entreprise X propose différentes formations. Un extrait du catalogue est présenté ci-dessous:

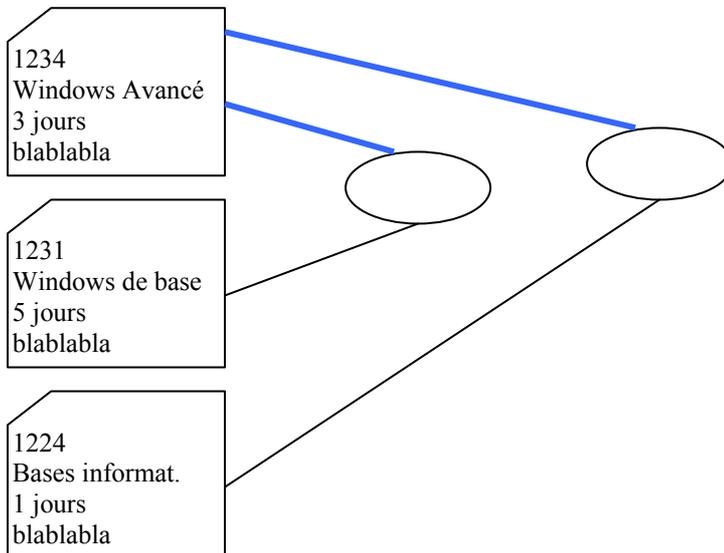
Formation n° 1231
 Intitulé : WINDOWS de base
 Durée: 5 jours
 Contenu :
 Formation(s) prérequis :

Formation n° 1234
 Intitulé: WINDOWS Avancé
 Durée: 3 jours
 Contenu:
 Formations pré-requis: 1231, 1224

Proposer un modèle conceptuel des données permettant d'accueillir l'ensemble des données contenues dans le catalogue de formations.



Exemple:



II. Les concepts étendus (MERISE 2)

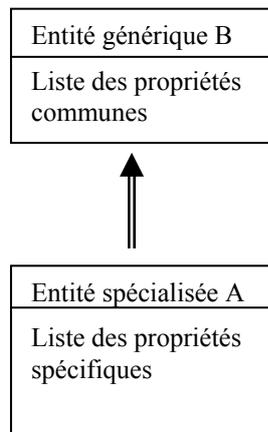
Le modèle entité association retenu par la méthode Merise date des années 70.

Or les concepts de ce modèle peuvent s'avérer insuffisants pour modéliser certaines situations ou contraintes et l'on est obligé dans ce cas d'ajouter des commentaires pour en faire mention. Les extensions au modèle individuel remédient aux faiblesses du formalisme de base.

2.1) Le concept d'héritage

Quand le concepteur s'aperçoit que plusieurs entités, proches mais distinctes, partagent un ensemble de caractéristiques, il doit mettre en œuvre un processus de création d'entités **génériques** (ou entités **sur-types**) et d'entités **spécialisées** (ou entités **sous-types**) appelé «**héritage**».

Ce concept qui permet de représenter le lien «**est-un**» ou «**IS-A**» entre deux entités A et B (une occurrence de A est une occurrence de B) est représenté graphiquement par une flèche double allant de A vers B.

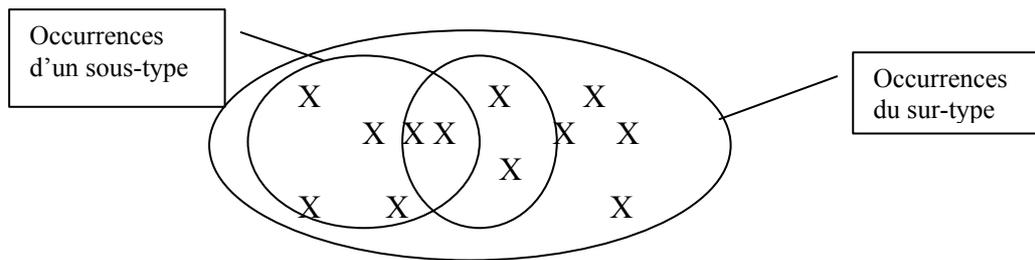


On dit qu'il y a **héritage simple** quand un sous-type n'a qu'un seul sur-type.

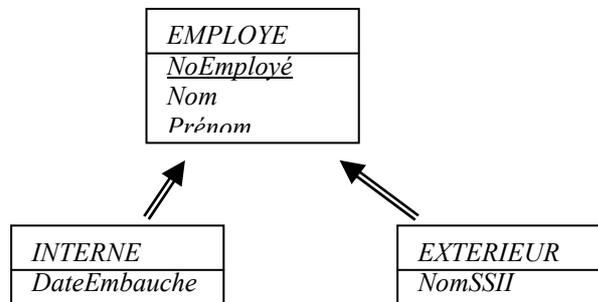
Dans ce cas, toutes les occurrences du sous-type sont en même temps des occurrences de son sur-type.

Cela n'implique pas que toutes les occurrences du sur-type soient des occurrences de l'un des sous-types.

Le schéma suivant illustre l'inclusion des ensembles d'occurrences des sous-types dans l'ensemble des occurrences du sur-type.



Le sous-type hérite de toutes les propriétés de son sur-type y compris de son identifiant. *Ce mécanisme qui correspond à l'héritage par spécialisation est utilisé dans le modèle suivant pour décrire partiellement l'univers du discours de l'exercice n° 3.*



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

Exercice n° 5 (Acquis : concepts de base + sous-typage)

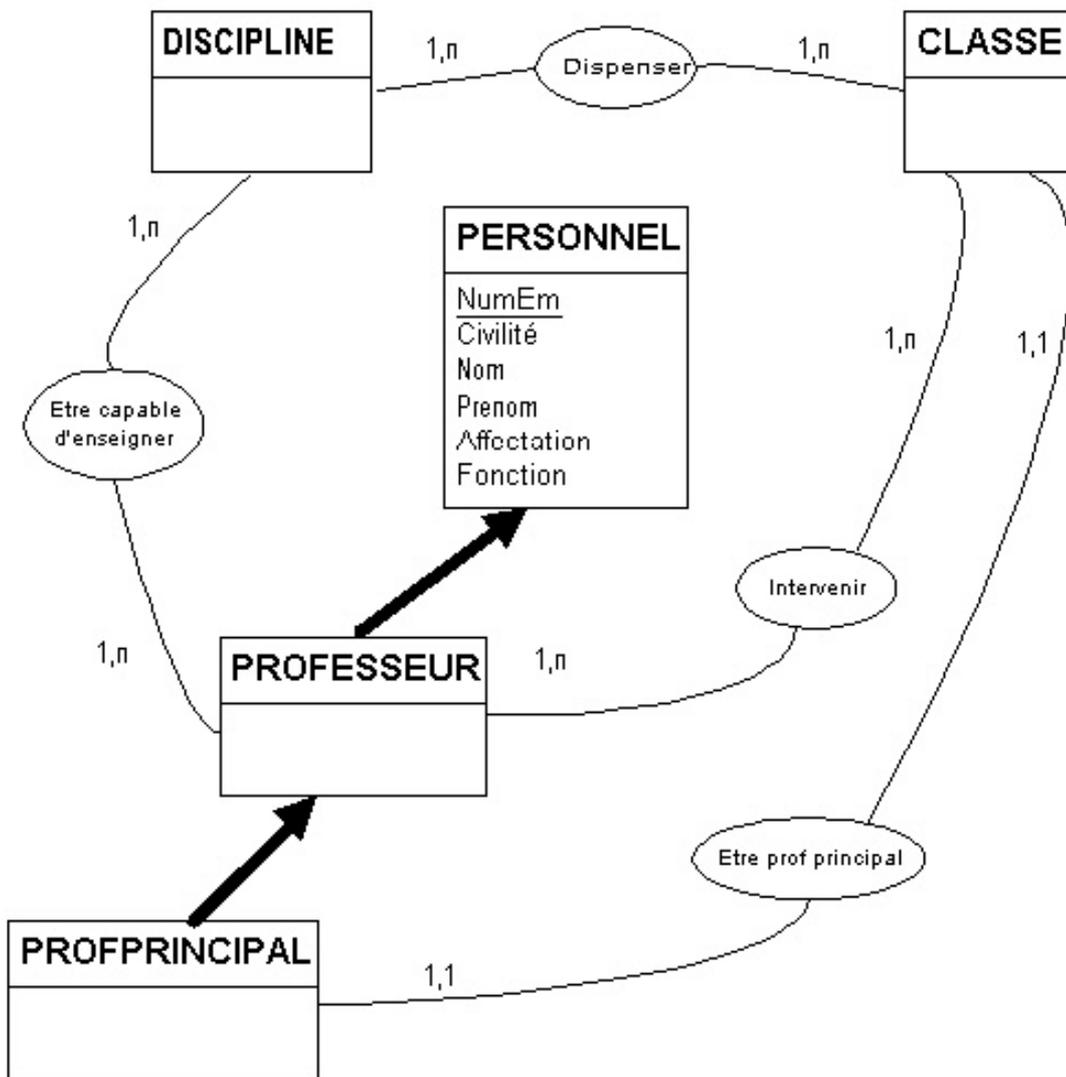
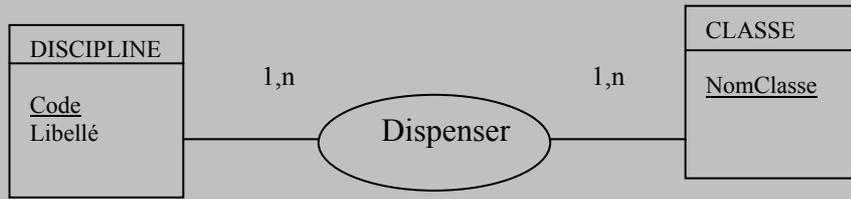
On souhaite représenter le système d'information relatif à la composition du personnel intervenant dans un lycée. Cette description devra fournir pour chaque personnel du lycée : administratif, professeur, surveillant, etc. un numéro qui permet de l'identifier (NUMEN ?) sans ambiguïté ainsi que les données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

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

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

La notion de professeur principal devra être aussi modélisée.

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



Contraintes:

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

On remarquera qu'ici, Ens(Classe) est incompatible avec Ens(Intervenir).

Par contre, Ens(ProfPrincipal) est compatible avec Ens(Intervenir).

2.2) Les contraintes ensemblistes

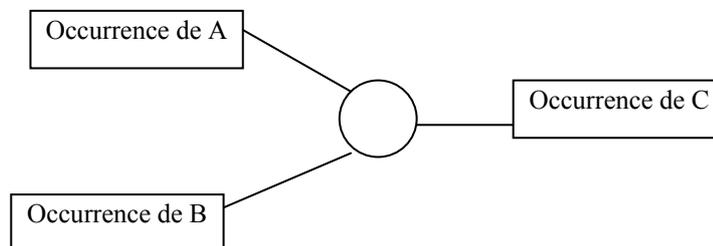
Ces formalismes vont permettre d'exprimer des contraintes sur des ensembles d'occurrences d'entités ou d'associations.

On peut considérer que c'est le plus grand apport de **Merise 2**.

Si la notion d'ensemble d'occurrences d'une entité ne pose aucun problème, il convient d'apporter certaines précisions sur l'ensemble des occurrences d'une association.

Considérons pour cela une association ternaire reliant trois entités A, B et C.

Une occurrence de cette association est un lien « tri-pattes » qui relie une occurrence de l'entité A, une occurrence de l'entité B et une occurrence de l'entité C .



En désignant par a1, b1 et c1 les valeurs des rubriques identifiantes de ces 3 occurrences d'entités, on peut matérialiser l'occurrence de l'association par le **triplet : (a1, b1, c1)**.

L'ensemble des occurrences d'une association peut donc être représenté par l'ensemble des triplets issus de la présence de liens entre les occurrences des entités A, B et C.

Le tableau ci-dessous présente des exemples d'occurrences des associations « Travailler » et « Habiter ».

<i>Modèle conceptuel des données</i>	<i>Occurrences de l'association TRAVAILLER</i>	<i>Occurrences de l'association HABITER</i>
<p style="margin-top: 10px;"><i>Remarque : on suppose qu'il n'y a pas deux professeurs qui portent le même nom</i></p>	<p>(« Dupont »,16) (« Dubois »,16) (« Durand »,17) (« Laforet »,17)</p>	<p>(« Dupont »,16) (« Dubois »,17) (« Durand »,16) (« Laforet »,86)</p>

A partir de l'ensemble des occurrences d'une association, il est possible de construire d'autres ensembles en supprimant, dans chaque « nuplet », la participation d'une ou plusieurs entités.

Dans l'exemple ci-dessus, si l'on considère uniquement l'entité *DEPT* (et donc on ne se préoccupe plus de l'entité *PROF*),

l'ensemble des occurrences de l'association *Travailler*, limité à l'entité *DPT*, devient composé des éléments 16 et 17.

- (« ~~Dupont~~ »,16)
- (« ~~Dubois~~ »,16)
- (« ~~Durand~~ »,17)
- (« ~~Laforet~~ »,17)

...

On appelle pivot la (ou les) entité(s) que l'on retient pour former ces nouveaux ensembles. Pour simplifier les futures définitions,

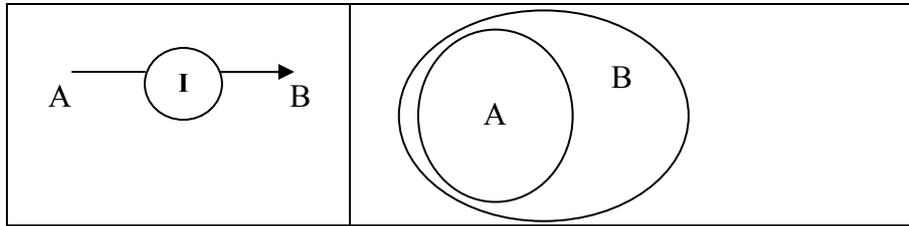
l'ensemble des occurrences d'un pivot participant à une association sera désormais référencé **Ens(Pivot,Association)**. En gardant la même logique, la référence **Ens(Entité)** désignera l'ensemble des occurrences de l'entité citée et la référence **Ens(Association)** l'ensemble des occurrences de l'association.

Pour les associations réflexives, il est possible de définir, à partir de l'ensemble des occurrences, deux nouveaux ensembles, notés **Ens(Rôle, Association)** issus chacun des rôles portés par les deux segments. En reprenant l'exemple fournit au paragraphe 1.3-g, les occurrences de l'association *Parenté* sont : (« Dubois », « Durand ») et (« Durand », « Delage »). Le premier composant du couple désigne le père, le second l'enfant. L'ensemble **Ens(Enfant, Parenté)** regroupe donc les valeurs « Durand », « Delage ».

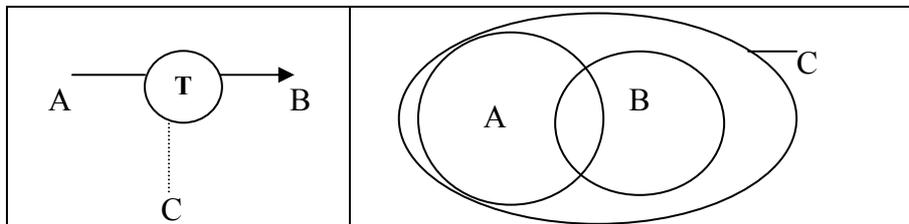
Quelque soit leur origine, les ensembles sur lesquels portent une contrainte ensembliste doivent être compatibles c'est-à-dire que leurs éléments doivent avoir des structures (en terme de rubriques) identiques. Par exemple on peut définir une contrainte ensembliste entre **Ens(TRAVAILLER)** et **Ens(HABITER)**.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

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

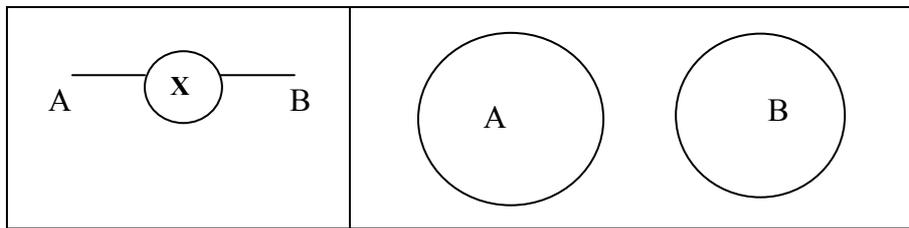
- **La contrainte d'inclusion** (notée **I**) entre deux ensembles A et B impose que l'ensemble A soit inclus dans l'ensemble B. L'inclusion n'étant pas symétrique, le formalisme associé à cette contrainte met en valeur, parmi les deux ensembles, celui qui doit être inclus dans l'autre.



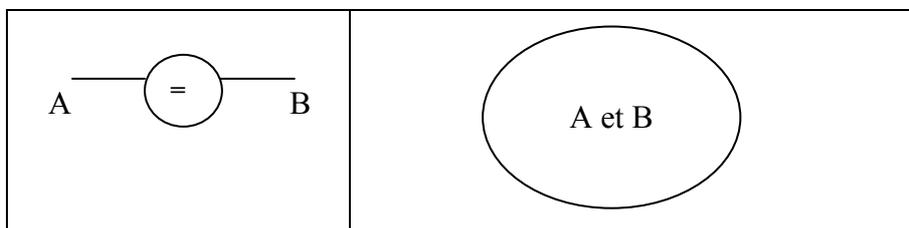
- **La contrainte de totalité** (notée **T**) porte sur trois ensembles A, B et C et impose que l'union de A et de B soit égale à C. Les trois ensembles ne jouant pas le même rôle, le formalisme prévoit un moyen de discerner les ensembles sur lesquels l'union sera réalisée et l'ensemble qui doit être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



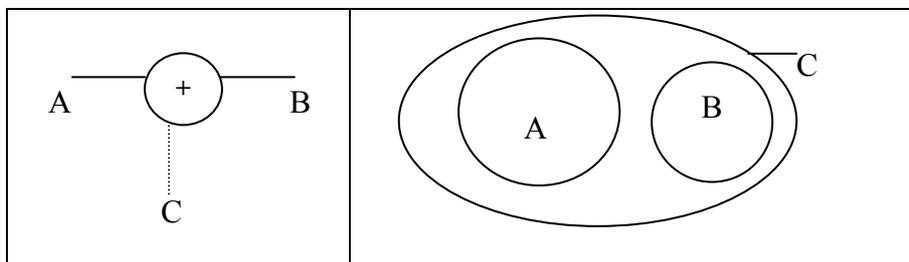
- **La contrainte d'exclusion** (notée X) entre deux ensembles A et B impose que l'intersection entre A et B soit vide.



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



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



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

a) La contrainte d'inclusion

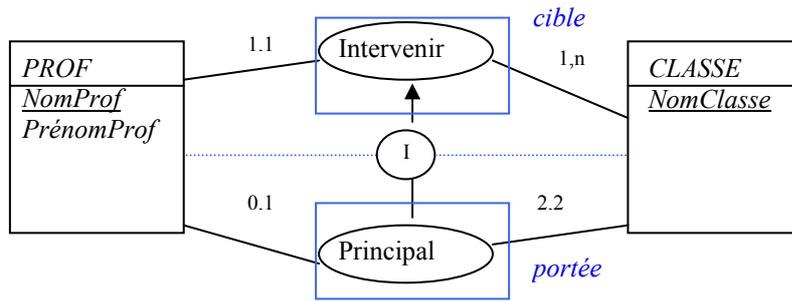
La contrainte d'inclusion s'exprime entre une ou plusieurs associations sources appelées **portée** et une seule association destination appelée **cible**. Si le pivot n'est pas précisé graphiquement (au moyen de traits en pointillés), il est nécessaire de le déterminer en appliquant la règle suivante :

il est constitué des entités communes aux associations de la portée et de la cible. La contrainte d'inclusion garantit alors que $Ens(Pivot, Portée)$ est inclus dans $Ens(Pivot, Cible)$. Dans l'exemple suivant le pivot est implicite et correspond aux entités *ELEVE* et *CLASSE*. On a donc les deux égalités suivantes :

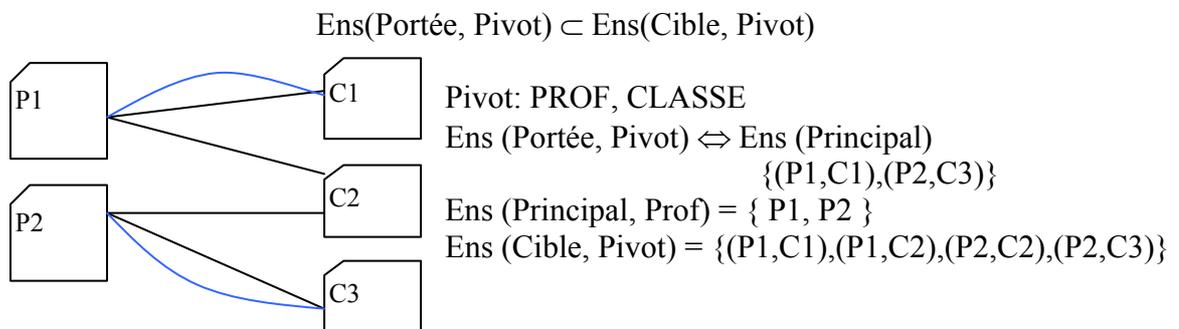
$$Ens(Pivot, Portée) = Ens(Principal)$$

$$Ens(Pivot, Cible) = Ens(Intervenir)$$

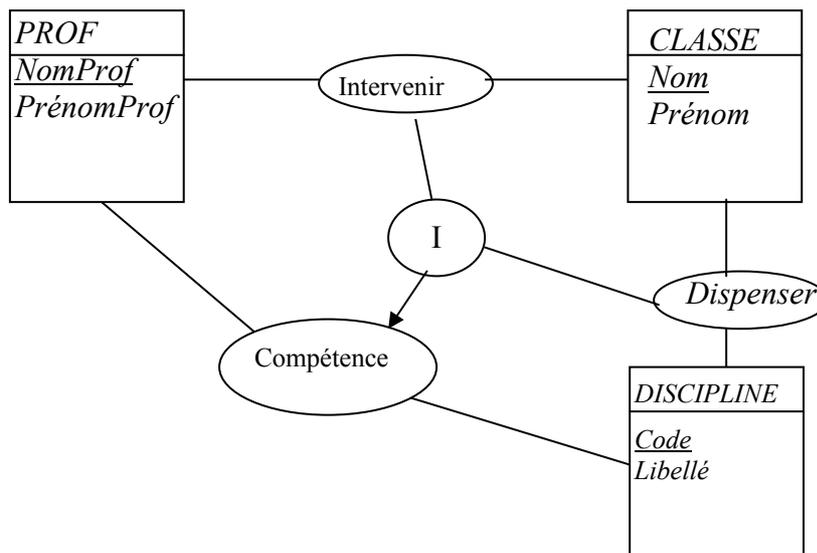
La contrainte d'inclusion assure que tout professeur principal d'une classe intervient dans la classe.

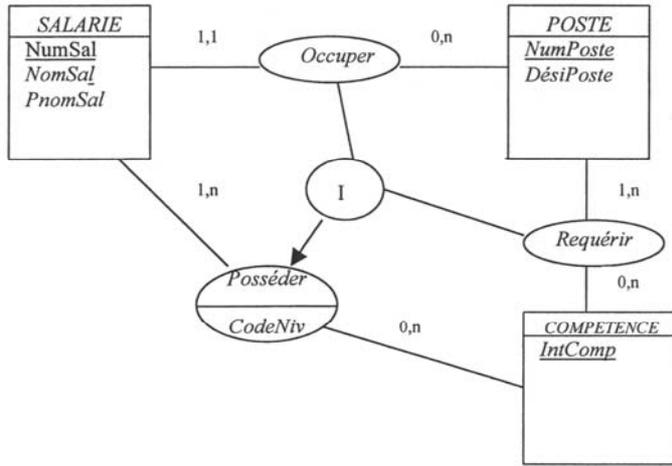


On note le pivot explicite par des pointillés, alors que le pivot implicite est formé par les entités communes à la portée et à la cible).

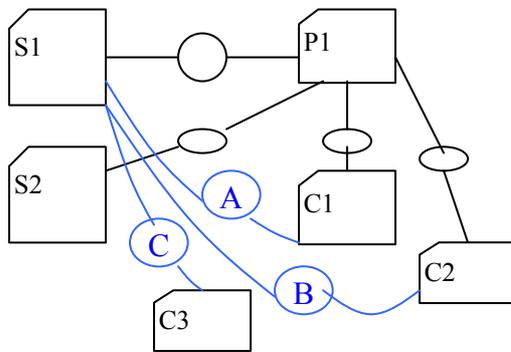


La cible est toujours simple, mais la portée peut être composée. Dans le cas où la portée est composée (c'est-à-dire formée de plusieurs associations), la détermination de $Ens(\text{Pivot}, \text{Portée})$ impose des opérations de jointure comme le montre les exemples suivants :





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

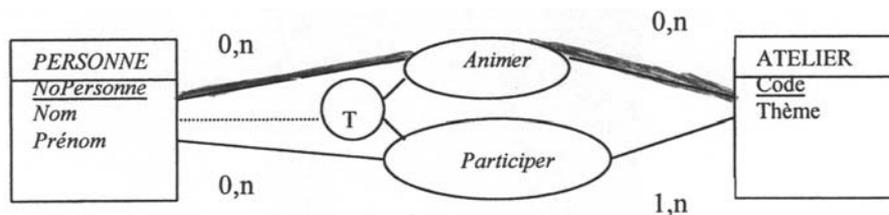


$Ens(Occuper) = \{(S1, P1), (S2, P1)\}$
 $Ens(Occuper, Salarié) = \{S1, S2\}$
 association, pivot
 $Ens(Requierir, Compétence) = \{C1, C2\}$
 $Ens(Portée, Pivot) = (occuper-requierir, salarié-compétence)$
 $\{(S1, C1), (S1, C2), (S2, C1), (S2, C2)\}$
 $Ens(Cible, Pivot) = \{(S1, C1), (S1, C2), (S1, C3)\}$

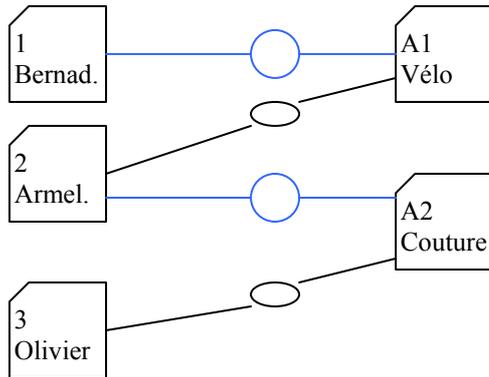
b) La contrainte de totalité

➤ La contrainte de totalité entre associations

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



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

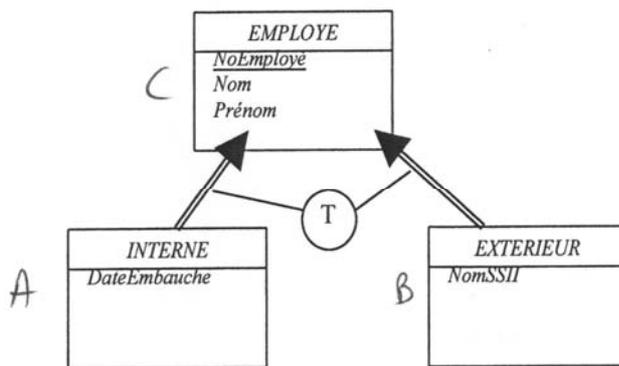


$Ens(Animer) = \{(1,A1), (2,A2)\}$
 $Ens(Animer, Personne) = \{1,2\}$
 $Ens(Participer, Personne) = \{2,3\}$

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

➤ **La contrainte de totalité entre sous-types**

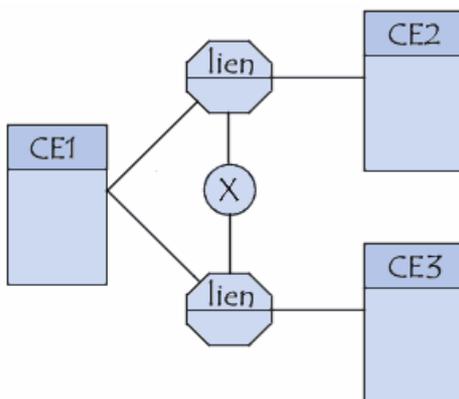
Cette contrainte s'exprime entre n sous-types ($n \geq 2$). Elle traduit le fait que l'union des ensembles $Ens(SousType_i)$ (avec i compris entre 1 et n) soit égale à $Ens(SurType)$. Toute occurrence du sur-type correspond donc à une occurrence d'un ou de plusieurs sous-types. Par exemple pour modéliser le fait que tous les employés du service informatique de la MAAF sont soit des internes soit des extérieurs on placerait une contrainte de totalité entre les deux sous-types.



Il est important de remarquer que cette contrainte n'interdit pas qu'un salarié interne soit aussi un extérieur. Pour refuser cette possibilité il sera nécessaire de renforcer cette contrainte par l'exclusion.

c) **La contrainte d'exclusion**

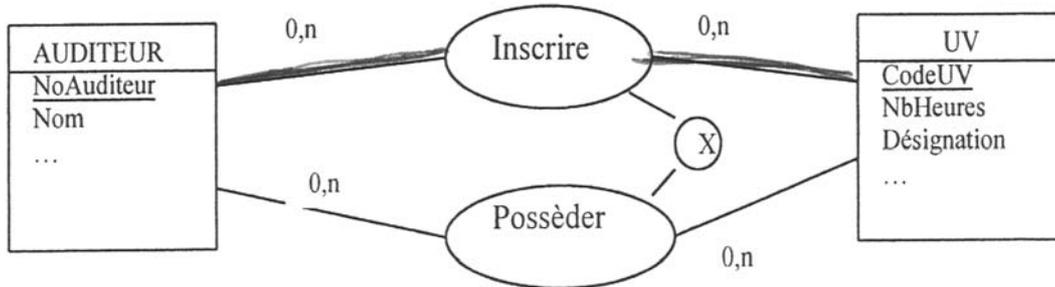
➤ **La contrainte d'exclusion entre associations**



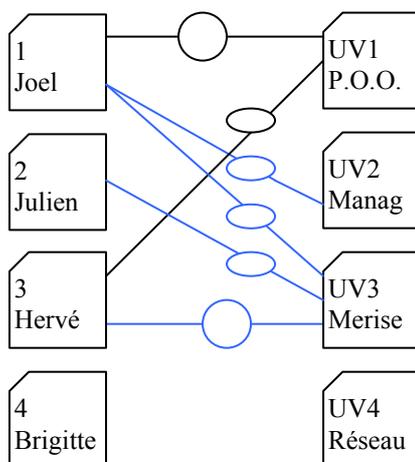
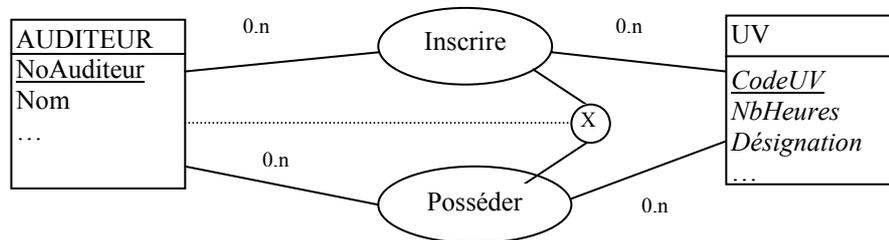
La contrainte d'exclusion sur relation exprime le fait que **deux occurrences de classes d'entité ne peuvent pas participer simultanément à une même classe de relation.**

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

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



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



Pivot: Auditeur et UV

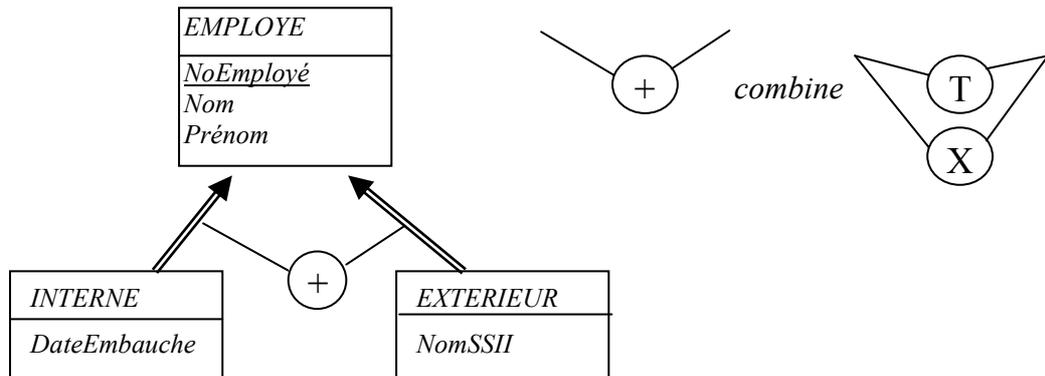
Ens (Posséder, Pivot) = {(1,UV1), (3,UV1)}
Ens(Inscrire,Pivot) = {(1,UV2),(1,UV3),(2,UV3),(3,UV3)}

Pivot: Auditeur

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

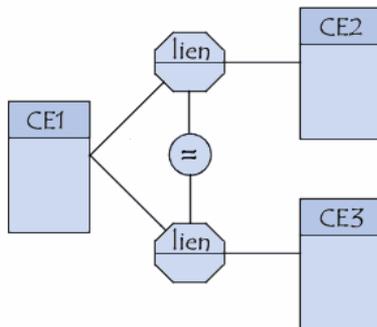
➤ La contrainte d'exclusion entre sous-types

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



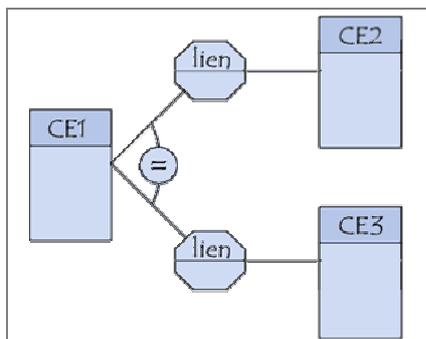
d) La contrainte d'égalité

➤ La contrainte d'égalité entre associations



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

➤ La contrainte d'égalité entre sous-types



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

Exercice n° 6 (Acquis : base + sous-typage + contraintes ensemblistes)

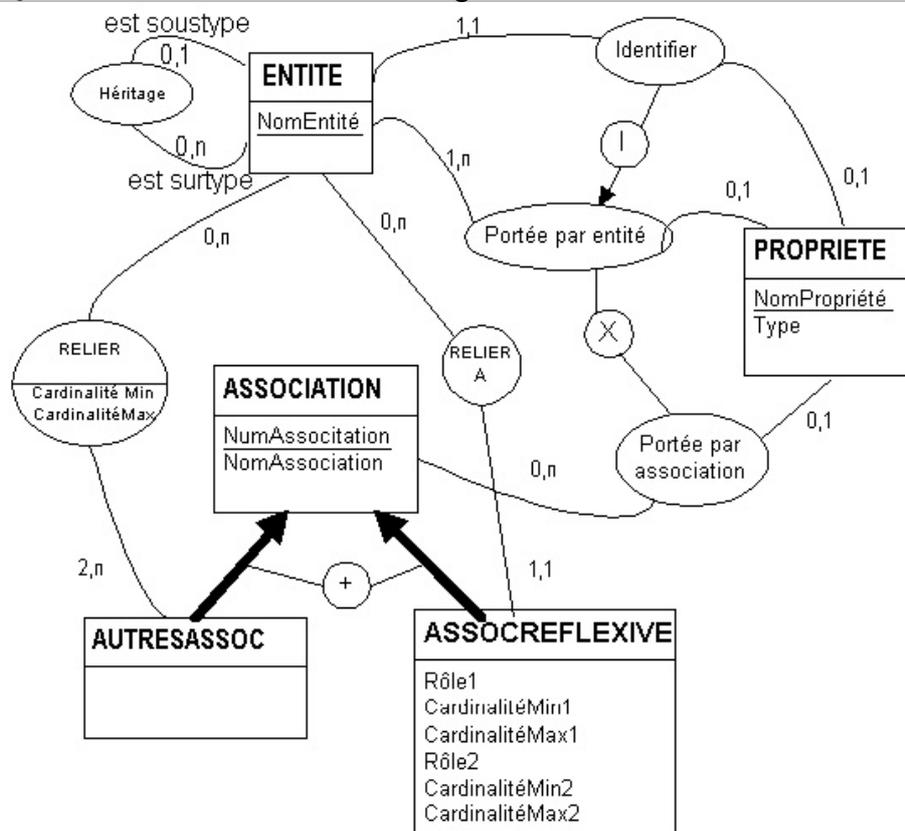
Cet exercice consiste à représenter toutes les notions qui se rapporte au modèle conceptuel des données au moyen des formalismes de ce modèle. On parle de méta-modèle. Les propriétés à structurer sont présentées dans la liste qui suit :

Propriétés	Définition
NomPropriété	Nom qui identifie une propriété
TypePropriété	Type de la propriété : alphabétique, numérique, ..
NumAssociation	Numéro qui identifie chaque association
NomAssociation	Nom de l'association
NomEntité	Nom qui identifie chaque entité
CardinalitéMin	Valeur minimale d'une cardinalité
CardinalitéMax	Valeur maximale d'une cardinalité
Rôle	Rôle porté par un segment d'une association réflexive

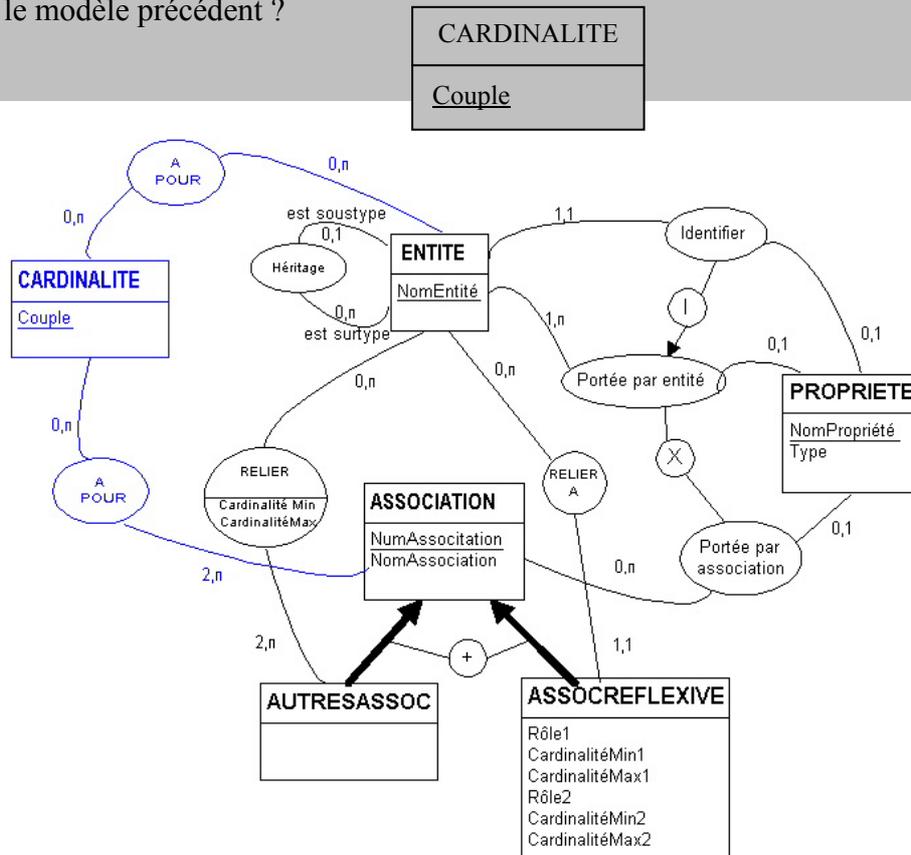
Remarque : la notion de contrainte n'est pas modélisée

1. Concevoir un MCD qui permet de répondre aux requêtes suivantes :

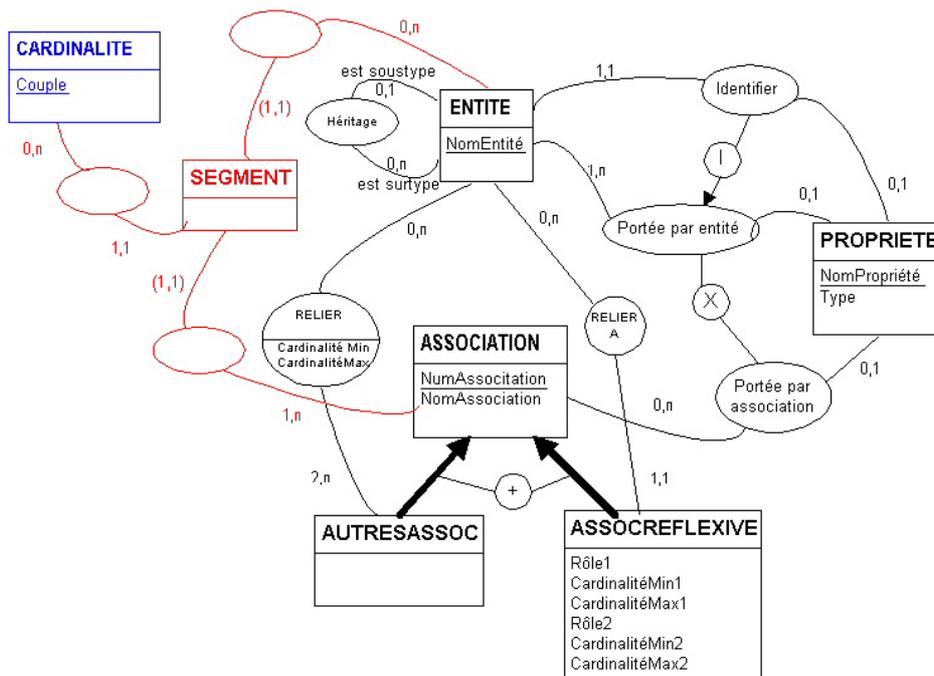
- Quelle est la liste des propriétés d'une entité?
- Quel est le nom de la propriété identifiante d'une entité?
- Quelle est la liste des entités sous-types d'une entité?
- Quelle est la liste des entités, participant à une association, avec pour chaque segment, reliant une association à une entité, le couple de cardinalités?
- Quelle est la liste des associations avec éventuellement les propriétés portées?
- Quels sont les rôles associés aux segments d'une association réflexive?



2. On introduit l'entité **CARDINALITE** ci-dessous destinée à mémoriser les 4 couples possibles de cardinalités : (0,1), (1,1), (0,n) et (1,n). Quel problème pose l'insertion de cette entité dans le modèle précédent ?



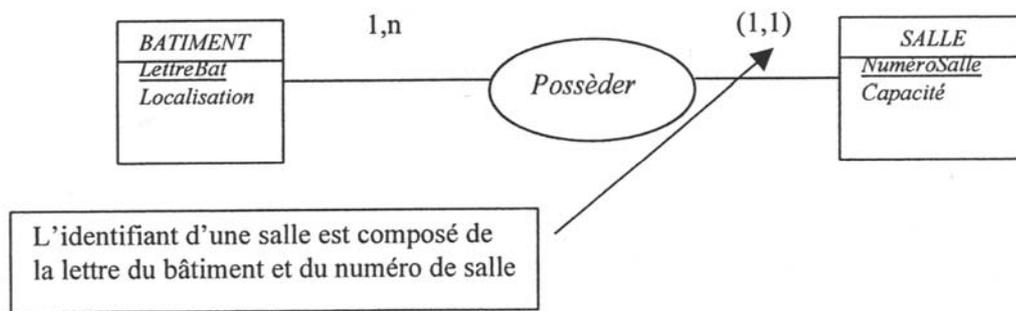
Le couple entité-relation n'est plus respecté mais on évite ainsi des cardinalités absurdes (comme 0,0 ou 20,10). On peut introduire aussi une pseudo entité **SEGMENT** avec un identifiant relatif (1,1) qui est la traduction du couple (NomEntité, NumAssociation).



2.3) L'identification des occurrences d'entités

Les extensions à l'identification des entités permettent d'accepter plusieurs façon d'identifier une entité et suppriment ainsi les identifiants artificiels, introduits uniquement pour respecter la définition d'une entité. Merise 2 admet les deux types d'identifiants suivants :

- **l'identifiant absolu** constitué de une ou plusieurs propriétés de l'entité. Dans la représentation graphique celles-ci sont soulignées. *On pourra ainsi identifier une personne au moyen des trois propriétés : nom, prénom et date de naissance.*
- **l'identifiant relatif** constitué de propriétés de l'entité et/ou au moins de l'identifiant d'une autre entité reliée par une association 1-n ou 1-1. *L'exemple suivant illustre une telle notion et introduit sa représentation graphique.*

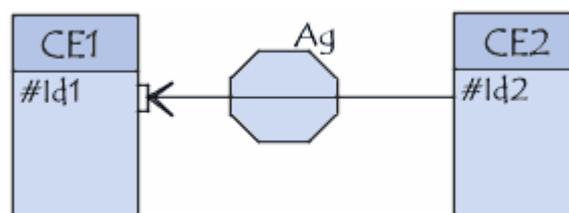


Remarque : l'association entre les deux entités doit être stable, c'est-à-dire qu'une fois un lien établi entre deux occurrences, celui-ci ne doit plus être modifié dans le temps.

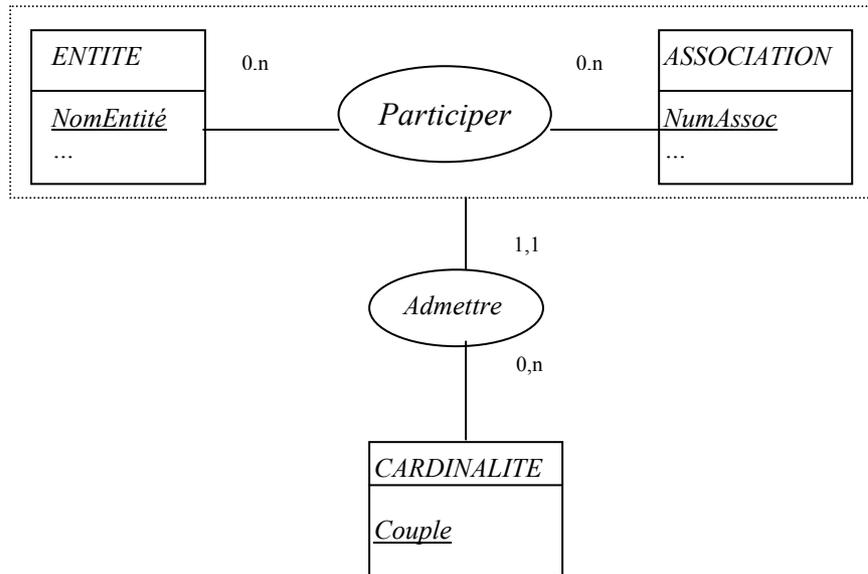
La notion d'identifiant relatif permet aussi d'exprimer un lien entre une association et une ou plusieurs entités. Certains auteurs appellent une telle association pseudo-entité ou agrégation. Lorsqu'un identifiant est constitué uniquement d'attributs intrinsèques à une entité, c'est-à-dire ne faisant référence à aucune autre entité, on le nomme **identifiant absolu**. Les entités comportant des identifiants absolus peuvent être définis indépendamment des autres occurrences d'entités, on dit que ces entités sont indépendantes. Certaines entités ne peuvent toutefois être identifiées que par l'intermédiaire d'autres entités, c'est la raison pour laquelle on parle d'**identification relative**. On parlera par exemple de *la 4^{ème} porte au 2^{ème} étage du bâtiment B* au lieu de dire la porte n°3451... Ainsi, l'**agrégation** (appelée aussi identification relative) permet de spécifier qu'une entité est nécessaire pour en identifier une autre.

- la classe d'entité permettant d'identifier est appelé classe d'entité agrégeante
- la classe d'entité identifiée est appelée classe d'entité agrégée

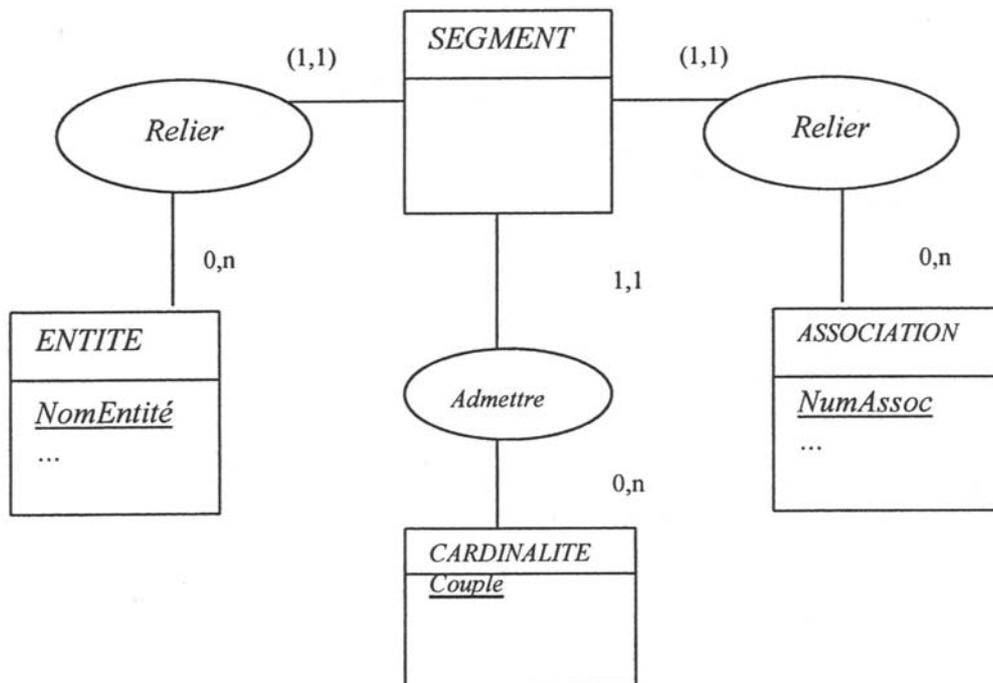
La représentation de ce type de relation est la suivante:



Dans l'exemple qui suit on souhaite établir un lien entre l'association participer et l'entité CARDINALITE.



Pour se dispenser de cette pseudo-entité, il suffit de transformer l'association en une entité et de lui associer un identifiant relatif composé des identifiants des entités reliées par des association de fonctionnalité (1-n) ou (1-1). L'exemple ci-dessous pourrait donc être transformé ainsi :



III. Règles de construction d'un M.C.D.

Pour la construction du modèle conceptuel, beaucoup de méthodes ont été mises en place mais aucune ne donne réellement satisfaction. On peut cependant les répartir en deux catégories :

3.1) modélisation directe

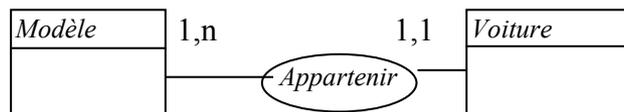
Elle consiste à identifier, à partir d'une description exprimée en langage naturel, les entités et les associations en appliquant les règles suivantes :

- les noms deviennent des entités
- les verbes deviennent des associations

L'exemple suivant qui illustre ce propos est bien trop simple pour que cette méthode conduise à des résultats satisfaisants sur un système d'information de taille plus importante.

Une voiture appartient à un modèle particulier. Les noms sont : « voiture », « modèle ». Le verbe est : « appartient à »

Ce qui donne la modélisation



Le modèle obtenu par cette méthode est très loin de la représentation optimale et il sera nécessaire d'appliquer une phase de validation et de normalisation (élimination des situations qui induisent des redondances) pour aboutir à une solution satisfaisante.

3.2) modélisation par analyse des dép. fonct^{lles}

Cette méthode consiste à identifier en premier lieu toutes les propriétés du système d'information à analyser.

Cette étape aboutit au dictionnaire des données épuré qui devra comporter ni synonyme, ni polysème, ni donnée calculée.

Pour faciliter la conception ultérieure des bases de données, il est recommandé de définir pour chaque donnée du dictionnaire son domaine. Le domaine d'une donnée est l'ensemble des valeurs que peut prendre cette donnée. Il peut être :

- étendu: il correspond alors au type d'une donnée : Numérique, alphabétique, etc.
- restreint: on l'exprime alors au moyen d'une liste ou d'un intervalle. Par exemple, pour la rubrique « Sexe », le domaine sera la liste de valeurs « F », « M ».

La seconde étape réside dans la recherche des dépendances fonctionnelles entre les propriétés recensées à la première étape. Pour mener de façon méthodique ce travail, on construit une matrice des dépendances fonctionnelles admettant une ligne et une colonne par propriétés du dictionnaire. Un « 1 » placé à l'intersection de la ligne i et de la colonne j indique la présence d'une dépendance fonctionnelle entre la propriété Pj et Pi (Pj → Pi).

L'exemple suivant illustre cette technique sur l'exemple classique de gestion des commandes d'une entreprise.

	1	2	3	4	5	6	7	8
1 NoCommande	1							
2 DateCommande	1	1						
3 NoClient	1		1					
4 NomClient	1		1	1				
5 RefProduit					1			
6 Designation					1	1		
7 PrixUnitaire					1		1	
8 QtéCommandée								1

NoClient → NomClient

Cette première phase met en évidence deux types de propriétés :

- les **propriétés identifiantes** repérées dans la matrice par des colonnes qui comportent au moins deux « 1 ». *Dans l'exemple ci-dessus les propriétés 1, 3 et 5 sont sources de dépendances fonctionnelles et joueront donc le rôle d'identifiant d'entités dans le modèle conceptuel.*
- les **propriétés** qui ne sont **destination d'aucune dépendance** fonctionnelle et qui ont donc, dans la matrice des dépendances fonctionnelles, leur ligne vide. *Dans l'exemple ci-dessus, les propriétés TauxTVA et QtéCommandée ont ces caractéristiques.* Parmi ces propriétés il convient alors de distinguer :
 - les **propriétés paramètres** telles que le *taux de TVA*
 - les **autres propriétés** pour lesquelles on doit rechercher les dépendances fonctionnelles ayant des sources multi-attributs qui permettent de les atteindre.

La source de ces dépendances sera constituée d'un sous-ensemble des rubriques identifiantes repérées à l'étape précédente. Ainsi, la propriété QtéCommandée, est déterminée à partir d'un numéro de commande et d'une référence produit ce qui revient à écrire la dépendance fonctionnelle suivante : **RefProduit, NoCommande → QtéCommandée**

Ces nouvelles dépendances fonctionnelles sont alors ajoutées à la matrice des dépendances fonctionnelles comme le montre la figure suivante.

	<u>1</u>	2	<u>3</u>	4	<u>5</u>	6	7	8	1,5
1 NoCommande	1								
2 DateCommande	1	1							
3 NoClient	1		1						
4 NomClient	1		1	1					
5 RefProduit					1				
6 Designation					1	1			
7 PrixUnitaire					1		1		
8 QtéCommandée								1	1

Certaines des dépendances fonctionnelles mentionnées dans cette matrice sont « parasites » car elles peuvent être déduites d'autres dépendances fonctionnelles par application des propriétés remarquables telles que la réflexivité ou la transitivité. Il faut donc les éliminer pour obtenir l'ensemble minimal des dépendances fonctionnelles qui représente la même information. Si l'on désigne par F l'ensemble initial des dépendances fonctionnelles, l'ensemble obtenu, noté F[^], après élimination des dépendances parasites est appelé couverture minimale de F et peut être obtenu par application d'un algorithme.

C'est encore un procédé algorithmique qui permet d'aboutir, à partir de la matrice dépouillée de toute dépendance inutile, au modèle conceptuel des données. Ainsi toute propriété identifiante donne naissance à une entité dont le contenu sera formée des propriétés avec lesquelles elle est en dépendance. Les propriétés atteintes par des dépendances fonctionnelles multi-attributs seront intégrées à des associations porteuses (m-n) reliant les entités dont les identifiants sont spécifiées dans la source.

Enfin, les dépendances entre identifiants se matérialiseront par la présence d'une association (1-n). Le modèle ci-dessous résulte de l'application de ces différentes règles. Ce dernier devra alors être complété notamment au niveau des cardinalités minimales afin de prendre en compte toutes les règles de gestion. Les associations non porteuses de type m-n ...



Cette méthode est très lourde dans sa mise en œuvre dès que le nombre de propriétés devient important. De plus, tout lien sémantique devenant une dépendance fonctionnelle, elle est très réductrice d'un point de vue sémantique, et ne permet pas de mettre en évidence des situations telles que plusieurs associations portant des réalités différentes entre deux mêmes entités.

Il semble que la bonne approche de construction d'un modèle conceptuel des données soit un compromis entre la méthode directe, qui laisse une large part à l'intuition et la méthode basée sur l'étude des dépendances fonctionnelles. Quelle que soit la technique utilisée, le modèle doit être vérifié, normalisé et enrichi de toutes les concepts étendus pour représenter le plus fidèlement possible l'univers du discours.