

INSIA – SIGL 2

La méthode MERISE

MCD - 2

Bertrand LIAUDET

SOMMAIRE

SOMMAIRE	1
COMPLEMENTS DE MODELISATION	3
0.1 Bilan de la modélisation relationnelle : les 7 types de clés primaires	3
Clé primaire simple : les tables noms et les tables types	3
Clé primaire simple et étrangère : les tables espèces	3
Clé primaire concaténée avec un identifiant relatif : les tables de composition	4
Clé primaire concaténée avec uniquement des clés étrangères : les tables de liaison	4
Clé primaire concaténée avec une date : les tables d'historique	5
Historique d'une table de composition	5
Historique d'une table de liaison	5
Synthèse	6
0.2 Bilan des bases de la modélisation entité-association	7
1. Identification relative et notion de composition	7
Présentation	7
Formalisme MEA	7
Traduction en MR	8
2. Historique	9
Présentation	9
Historique d'un attribut	9
Historique d'entité	10
Historique d'association hiérarchique	10
Utilisation des CIF dans le MEA	11
3. Héritage - Généralisation	16
Héritage : l'inclusion d'un ensemble dans un autre	16
Les différents cas d'héritage et le passage du MEA-OO au MR	17
Les contraintes sur la généralisation	19
Vocabulaire	22
Représentation ensembliste du MEA	23

4. Les entités complexes	25
Présentation	25
Entité complexe	26
Autre formalisme : association d'association	28
5. Autres contraintes	30
Les contraintes de stabilité	30
Les contraintes sur associations	31
La spécialisation des associations	36
6. Les données isolées	37
Présentation	37

Première édition : mai 2008

Deuxième édition : décembre 2008

COMPLEMENTS DE MODELISATION

PRINCIPALES NOTIONS

Identification relative	Composition
Historique	Héritage
Représentation ensembliste	Couverture
Disjonction	CIF
Association d'association	Entité complexe

0.1 Bilan de la modélisation relationnelle : les 7 types de clés primaires

Clé primaire simple : les tables noms et les tables types

Exemples

- 1 : Les employés et les départements.
- 2 : Les livres de la bibliothèque.
- 3 : Les avions et leurs types

Solutions

Employés (NE, nom, fonction, salaire, #ND)
Départements (ND, nom, ville)

Livres (NL, éditeur, dateAchat, #NO)
Oeuvres (NO, titre, auteur, dateCréation)

Avions (NA, année, couleur, propriétaire, #typeAvion)
TypeAvion(typeAvion, nombre places, année, moteur)

Principe

En général, une table noms correspond à une réalité physique : les employés, les départements, les exemplaires physiques des livres.

Elles peuvent aussi correspondre à des types de la réalité physique : c'est le cas des « TypeAvion », par exemple, le A320, ou des « œuvres » qui peuvent être considérées comme un type de « livres », le livre comme l'avion étant les exemplaires physiques.

Clé primaire simple et étrangère : les tables espèces

Exemple

On gère des personnes. Certaines sont étudiantes et suivent des études : année, domaine, spécialisation. D'autres sont salariés et ont une fonction, un salaire et une date d'embauche.

Solution

Personnes (NP, nom, prénom, adresse, téléphone)

Etudiants (#NP, domaine, spécialisation, année)

Salariés (#NP, fonction, salaire, datemb)

Principe

La table espèce correspond à une spécialisation d'une table-genre. La clé primaire de la table espèce est constituée par celle de la table genre et est donc clé étrangère en même temps.

Clé primaire concaténée avec un identifiant relatif : les tables de composition

Exemple

On gère des projets qui ont un nom, une date de début, une date de fin et un budget. Les projets sont composés d'étapes en nombres variables. Une étape est définie par son numéro d'ordre dans le projet (de 1 à N), par une date de début et une date de fin, un nom d'étape et un budget d'étape.

Solution

Projets (NP, nom, début, fin, budget)

Etapes (#NP, NE, nom, début, fin, budget)

Principe

Le numéro d'étape est relatif : de 1 à N. Il y a donc plusieurs étapes qui ont le même numéro d'étape. C'est le couple «NP, NE » qui est unique.

L'étape est un composant du projet : elle disparaît nécessairement avec le projet (elle n'a pas d'existence indépendamment du projet).

Clé primaire concaténée avec uniquement des clés étrangères : les tables de liaison

Exemple

On envoie des courriers en nombre à des clients. Un courrier est caractérisé par un libellé et une date. La date d'envoi correspond à la date du courrier. On veut savoir quel client à reçu quel courrier

Solution

Courriers (NCO, libellé, date)

Clients (NCL, nom, adresse)

Envoyer (#NCL, #NCO)

Principe

La clé primaire ne contient que des clés étrangères.

Elle peut en contenir plus que 2 si elle relie 3 ou plus tables entre elles.

Les clés étrangères peuvent faire référence à tous les types des clés primaires.

Une table de liaison peut aussi avoir des attributs en plus de sa clé primaire.

Clé primaire concaténée avec une date : les tables d'historique

Exemple

- 1 : les emprunts à la bibliothèques.
- 2 : l'historique des adresses des adhérents de la bibliothèque

Solution

Livres (NL, éditeur, dateAchat, #NO)
Oeuvres (NO, titre, auteur, dateCréation)
Adhérents (NA, nom)
Emprunter (#NL, datEmp, dureeMax, dateRet, #NA)
HistoAdressesAdherents (#NA, date, adresse)

Principe

Dès qu'une clé primaire contient une date, c'est un historique.
Dans le cas de l'adresse, on a sorti l'attribut adresse de la table « Adhérents ».
A noter que l'historique est un cas particulier de la composition : les adresses sont des composants des adhérents.

Historique d'une table de composition

Exemple

On gère des projets qui ont un nom, une date de début, une date de fin et un budget. Les projets sont composés d'étapes en nombres variables. Une étape est définie par son numéro d'ordre dans le projet (de 1 à N), par une date de début et une date de fin, un nom d'étape et un budget d'étape.

Le budget des étapes peut varier. On veut garder l'historique.

Solution

Projets (NP, nom, début, fin, budget)
Etapes (#NP, NE, nom, début, fin)
HistoBudgetEtapes (#(NP, NE), date, budget)

Principe

C'est le même principe qu'un historique simple.
On crée une nouvelle table pour l'historique. Sa clé primaire est constituée par la concaténation d'un attribut date et de la clé primaire de la table contenant l'attribut dont on veut faire l'historique.

Historique d'une table de liaison

Exemple

On envoie des courriers en nombre à des clients. Un courrier est caractérisé par un libellé et une date. Un même courrier peut être envoyé plusieurs fois à la même personne. On veut savoir quel client à reçu quel courrier

Solution

Courriers (NCO, libellé, date)

Clients (NCL, nom, adresse)

Envoyer (#NCL, #NCO, date)

Principe

Même principe que pour toutes les tables d'historique

Synthèse

Principe de la clé primaire	Type de table	Principe de la clé primaire	
		Non historique	Historique
CP simple	Table-nom et table-type	CP	Tables historiques
Héritage	Table-espèce	#CP	#CP, date
Identifiant relatif	Table de composition	#CP, n°	#CP, n°, date
Liaison	Table de liaison (table verbe)	#CP1, #CP2	#CP1, #CP2, date

En grisé (jaune) : les clés primaires concaténées.

Les tables de liaison peuvent avoir plus de 2 clés étrangères.

Clés primaires imbriquées : clés étrangères complexes

Les clés étrangères peuvent toujours faire référence à n'importe qu'elle type de clé primaire. Elles peuvent donc toujours être concaténées.

Par exemple : on peut imaginer une table de liaison historique qui relie une table de liaison historique avec une table de composition, ce qui donnerait comme clé primaire :

#(CP, n°), #(CP1, CP2, date), date

Principe de la découverte de la totalité des clés primaires

A partir d'une clé primaire simple, on peut avoir 3 types de relations :

- Des relations d'héritage
- Des relations de composition
- Des relations liaison simple

Ces trois types de relations sont celles qu'on retrouvera en UML dans les diagrammes de cas d'utilisation et dans les diagrammes de classes.

A cela s'ajoute dans tous les cas :

la possibilité d'un historique. A noter que l'historique est un cas particulier de la

0.2 Bilan des bases de la modélisation entité-association

Les bases de la modélisation entité-association ont permis de prendre en compte :

- Les tables noms et les tables types
- Les tables de liaison
- Certaines tables historiques

Il reste à voir la façon de représenter :

- Les tables de composition
- Les tables espèces
- Toutes les tables historiques
- Les clés étrangères complexes.

1. Identification relative et notion de composition

Présentation

L'identification absolue est le principe de base de la modélisation entité-association : la clé primaire d'une entité type est ce qui définit l'identifiant absolu de chaque entité.

Quand une entité type a une existence dépendante d'une autre entité type, on a recours à une identification relative.

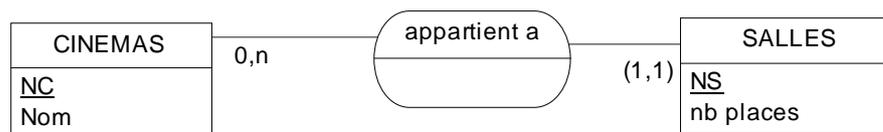
Cette dépendance traduit une relation de composition entre les entités : l'entité (concrète) dépendante est un composant de l'entité dont elle dépend.

Cette dépendance se traduit par le fait que la destruction de l'entité composée implique la destruction des entités composantes.

Exemples

- Un cinéma est composé de salles de cinéma : chaque salle de cinéma a un numéro qui est un identifiant relatif au cinéma auquel elle appartient.
- Un projet est composé de tranches : chaque tranche a un numéro qui est un identifiant relatif au cinéma auquel elle appartient.

Formalisme MEA



L'identification relative correspond toujours à une association hiérarchique.

La cardinalité (1.1) est mise entre parenthèses.

Traduction en MR

CINEMAS (NC, nom)

SALLES (#NC, NS, nbPlaces)

Règle de passage du MEA au MR

La clé primaire d'une entité composante est constituée par la concaténation de la clé primaire de l'entité composante et de la clé étrangère issue de l'association hiérarchique correspondant à l'identification relative.

2. Historique

Présentation

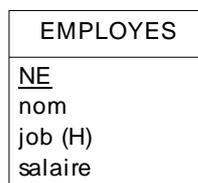
Gérer un historique consiste à garder la trace des modifications apportées aux valeurs de certains attributs.

Historique d'un attribut

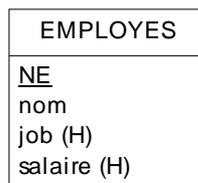
Formalisme MEA

On veut conserver l'historique d'un attribut, par exemple le job des employés.

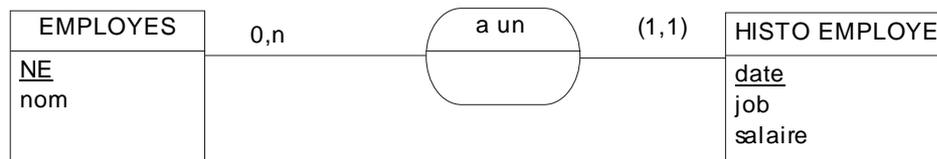
On marque l'attribut avec un « H » entre parenthèses :



L'historique peut aussi concerner plusieurs attributs :



On peut aussi écrire :



On voit ici que l'historique est un composant de l'employé.

La première méthode est préférable car elle évite de surcharger le MEA.

Traduction en MR

Historique du job :

EMPLOYES (NE, nom, salaire)

JOBS_EMPLOYES (#NE, dateDébut, job)

Historique du job et du salaire :

EMPLOYES (NE, nom)

JOBS_EMPLOYES (#NE, dateDébut, job, salaire)

Règle de passage du MEA au MR

Un attribut avec historique conduit à la création d'une table historique. La clé primaire de la table d'origine devient clé étrangère dans la table historique. Cette clé étrangère, concaténée à un attribut dateDébut, devient forme la clé primaire de la table historique. L'attribut avec historique devient attribut dans la table historique et n'est plus attribut dans la table issue de son entité d'origine.

Historique d'entité

Principe

Une entité « historisée » porte un attribut « date » qui est clé primaire. L'entité historisée est associée en tant que composant à l'entité à laquelle elle correspond.

Formalisme MEA

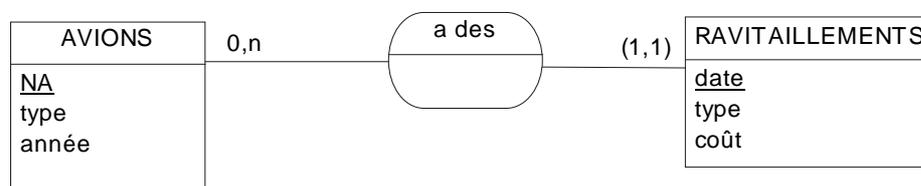
Association hiérarchique avec identifiant relatif. Attribut « date » comme clé primaire de l'entité « historisée ».

Traduction en MR

La traduction en MR correspond à la traduction des identifications relatives.

Exemple

On veut garder l'historique des ravitaillement d'essence pour des avions. Un ravitaillement est caractérisé par un volume et un coût.



On évite de créer un numéro absolu de ravitaillement.

Historique d'association hiérarchique

Principe

Une association hiérarchique « historisée » se transforme en association non hiérarchique et porte un attribut « date ».

Formalisme MEA

Association non hiérarchique avec historique.

Traduction en MR

La traduction en MR correspond à la traduction classique des associations non hiérarchiques.

Exemple 1

On « historicise » le département de l'employé :



Si on veut « historiciser » le job, on écrira :



ou :



ou :

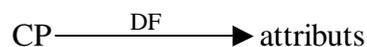


Cette dernière version à l'avantage qu'elle permet un passage automatique au MR.
Elle a le défaut de surcharger le MEA.

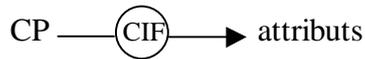
Utilisation des CIF dans le MEA

La notion de CIF : contrainte d'intégrité fonctionnelle

Une CIF entre deux attributs traduit une dépendance fonctionnelle entre ces attributs.



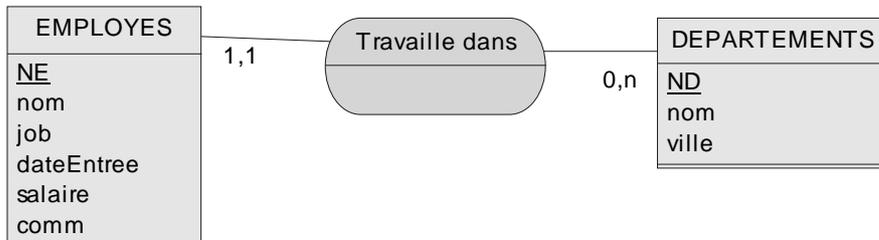
Est équivalent à :



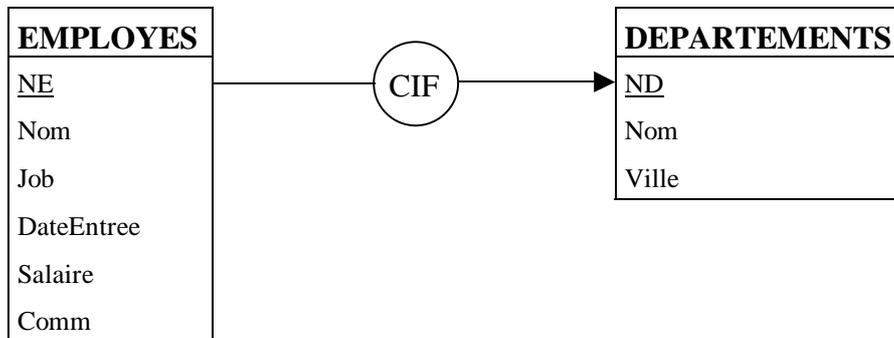
La CIF signifie (comme la DF) : si la CP existe, alors les attributs existent et sont déterminés, même si on ne connaît pas leurs valeurs.

CIF et association hiérarchique

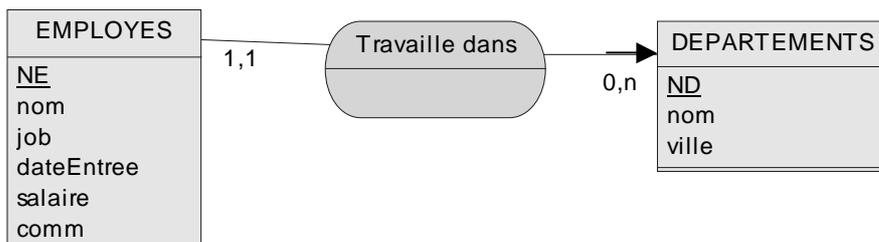
L'association hiérarchique suivante :



est parfois représentée ainsi :



On peut aussi représenter l'association hiérarchique ainsi :



Dans ce cas, la CIF est inutile : l'association hiérarchique la représente très bien.

La CIF traduit la dépendance fonctionnelle entre la clé étrangère qui sera dans l'entité 1.1 et la clé primaire de l'entité 0.N :



soit :



Usage des CIF dans le MEA – exemple 1

On reprend l'historique des employés et on ajoute la CIF et le ou les attributs de l'association participant à la clé primaire :



Cette représentation évite de surcharger le MEA.

Elle permet un passage automatique au MR.

TRAVAILLE(#NE, dateDeb, job, #ND)

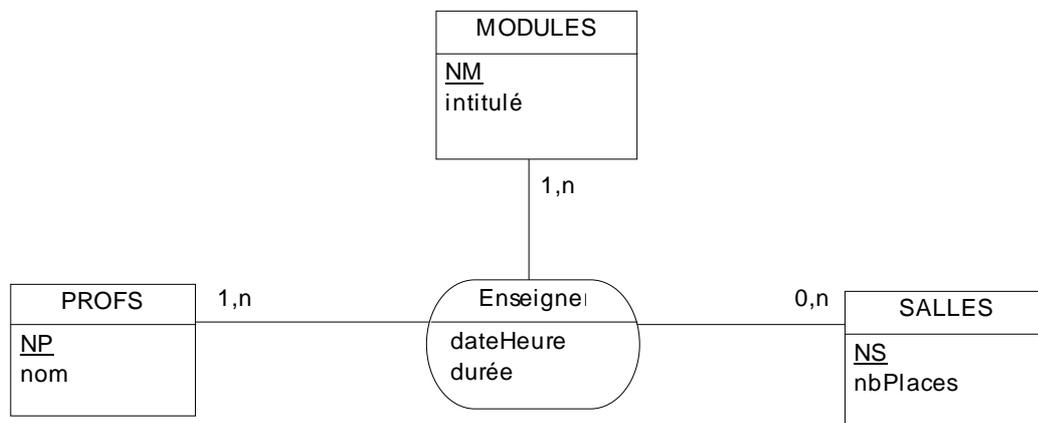
Règles de passage du MEA au MR

On modifie seulement la règle 3 :

Règle 3 – Association non hiérarchique : Une association non hiérarchique devient une table-verbe. Les clés primaires des entités associées deviennent clés étrangères dans cette table. Les attributs de l'association deviennent attributs de la table. **La clé primaire de la table-verbe est constituée par la concaténation des clés primaires des entités associées qui ne sont pas pointées par une CIF et des attributs clé-primaire (soulignés) de l'association.**

Usage des CIF dans le MEA – exemple 2

Soit le MEA suivant :



Ce MEA décrit la situation suivante : un professeur enseigne 1 ou N modules. Un module est enseigné par 1 ou N enseignants. Un module est enseigné dans 1 ou N salles. Un professeur enseigne dans 1 ou N salle. A une date et heure données, un professeur enseigne un module dans une salle.

Le MEA donne le MR suivant :

P(NP, nom)

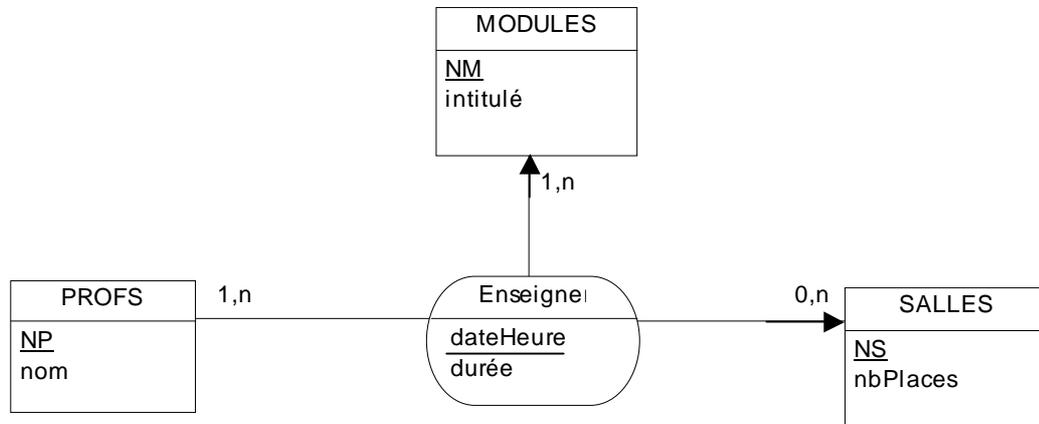
M(NM, intitulé)

S(NS, nbPlaces)

ENSEIGNER(#NP, dateHeur, durée, #NS, #NM)

La clé primaire de la table issue de l'association hiérarchique est déterminée sémantiquement.

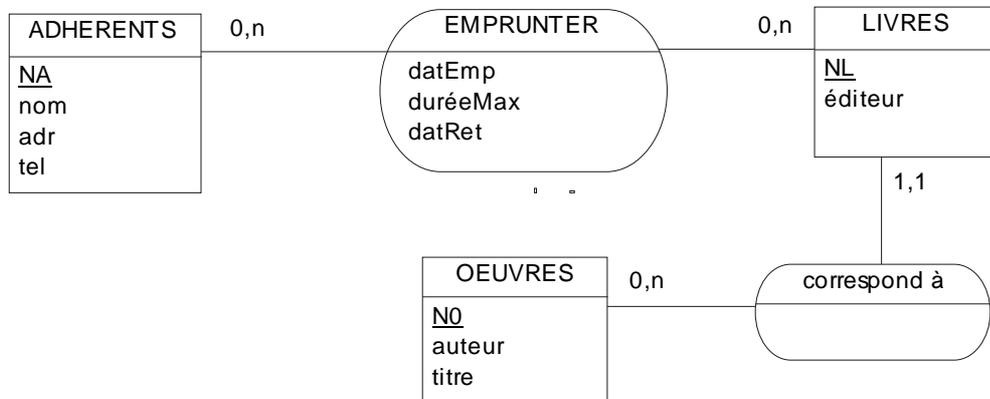
Représentation du MEA avec des CIF :



Les pattes d'association sont fléchées dans le sens des CIF. On souligne aussi les attributs de l'association qui participeront à la clé primaire.

Usage des CIF dans le MEA – exemple 3

Soit le MEA suivant :



Le MEA donne le MR suivant :

ADH(NA, nom, adr, tél)

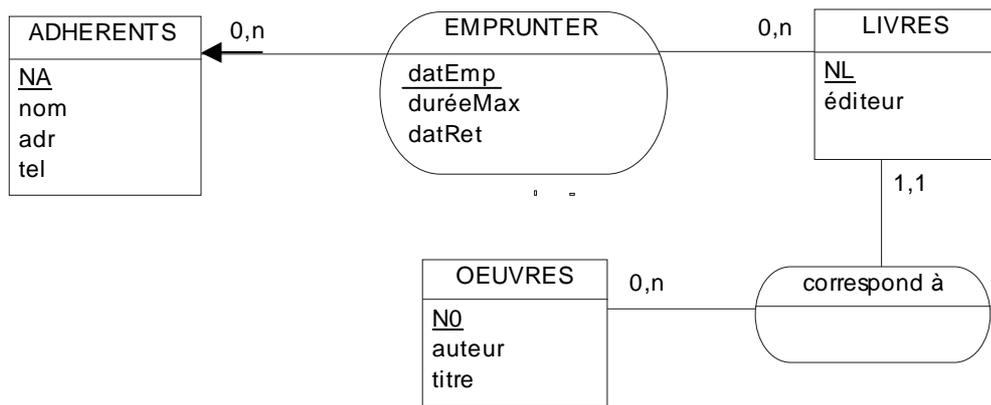
LIV(NL, éditeur, #NO)

OEUV(NO, titre, auteur)

EMP(#NL, datemp, duréeMax, datret, #NA)

La clé primaire de la table issue de l'association hiérarchique est déterminée sémantiquement.

Représentation du MEA avec des CIF :



3. Héritage - Généralisation

Héritage : l'inclusion d'un ensemble dans un autre

Héritage, spécialisation, généralisation

La représentation ensembliste permet de concevoir aisément la notion héritage : il suffit de concevoir l'inclusion d'un ensemble dans un autre.

L'ensemble inférieur hérite des propriétés et des associations de l'ensemble supérieur : l'espèce hérite des attributs du genre.

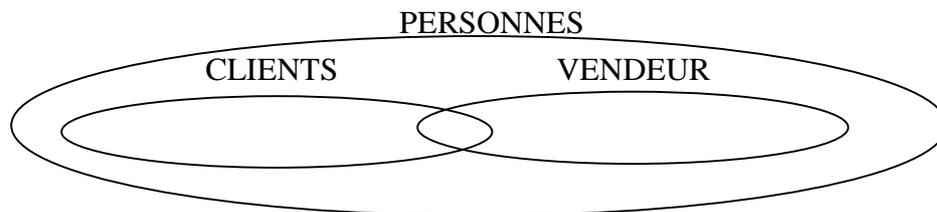
Un ensemble fait hériter ses attributs à tous les ensembles de niveaux inférieurs qu'il contient. Le genre fait hériter à toutes ses espèces.

Cette organisation forme une hiérarchie, c'est la hiérarchie de spécialisation/généralisation, encore appelée hiérarchie « is-a », « est-un ».

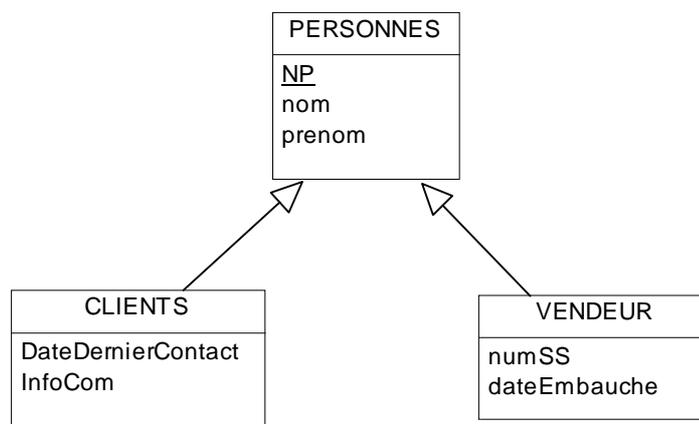
Ces trois notions : **spécialisation, généralisation, héritage** sont les notions clés du modèle « orienté objet ».

Dans le formalisme objet, on parle d'objet à la place de tuple et de classe à la place d'entité.

Formalisme ensembliste



Formalisme MEA-OO



En général, on utilise la flèche avec une pointe en triangle creux pour symboliser l'héritage.

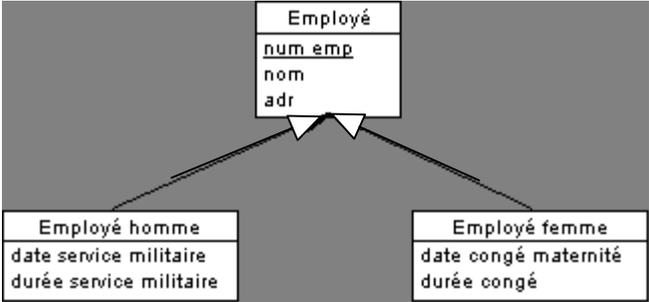
Il n'est pas nécessaire de préciser les cardinalités puisque l'héritage correspond toujours à une association 0..1-1 : une personne est ou n'est pas un client. Un client est toujours une personne.

Les différents cas d'héritage et le passage du MEA-OO au MR

Généralisation de base

À plusieurs entités-espèces correspond un genre commun. C'est le cas le plus courant de la généralisation.

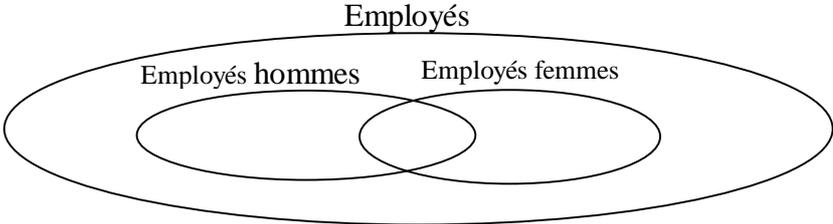
➤ **MEA**



L'entité « Employé » est une entité comme les autres.

Les entités « Employé homme » et « Employé femme » sont des entités spécialisées. Elles sont reliées à l'entité « Employé » par un lien fléché. **Elles ne contiennent pas de clés primaires.**

➤ **Représentation ensembliste**



➤ **MR**

Emp(numEmp, nom, adr)
EmpH(#numEmp, datSM, durSM)
EmpF(#numEmp, datCM, durCM)

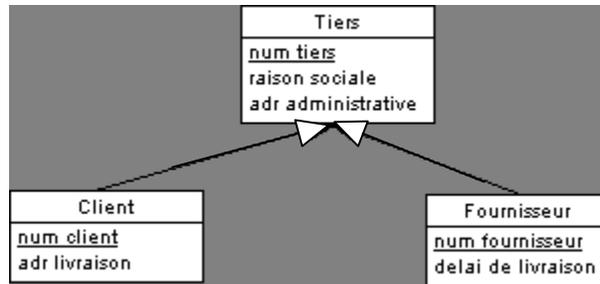
➤ **Règle de passage du MEA au MR**

La clé primaire de l'entité genre devient clé primaire et étrangère dans les entités espèces.

Généralisation d'entité dont on conserve la clé primaire

Cette généralisation suit le même principe que la généralisation de base, mais les entités spécialisées ont une clé primaire.

➤ MEA



➤ MR

Clients (NC, (#NT), adrLiv)

Fournisseurs (NF, (#NT), delaiLiv)

Tiers (NT, raiSoc, adrAdm)

#NT est présenté entre parenthèse et juste après la clé primaire car c'est une clé secondaire (attribut UNIQUE). Ces deux attributs se justifient uniquement si l'information qu'ils représentent est une information du système d'information, indépendamment de la BD.

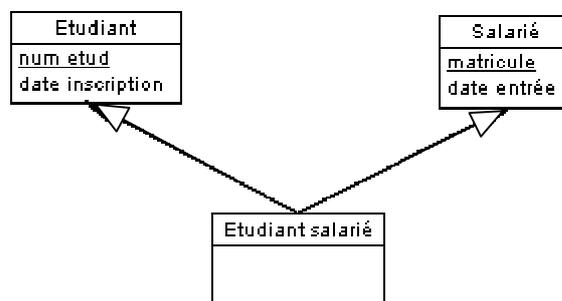
➤ Règle de passage du MEA au MR

La généralisation avec clé primaire dans l'entité spécialisée est équivalente à une association hiérarchique : elle se traduit par la création d'une clé étrangère. A cela s'ajoute que la clé étrangère est UNIQUE.

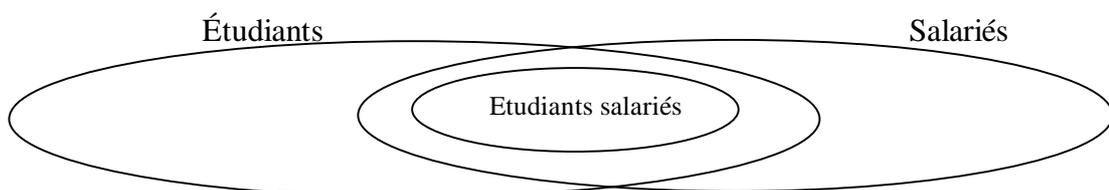
Intersection de 2 entités

On part de deux entités. On veut spécialiser leur intersection.

➤ MEA



➤ Représentation ensembliste



➤ **MLD**

Etud(numEtud, datIns)

Sal(numSal, datEnt)

EtudSal(#numEtud, (#numSal))

Ou EtudSal(#numSal, (#numEtud))

Le deuxième attribut est mis entre parenthèses : c'est une clé secondaire (attribut UNIQUE).

Remarque :

Chaque clé étrangère doit être unique (en plus du fait que le couple des clés étrangères est unique).

Les étudiants salariés pourraient porter des attributs spécifiques.

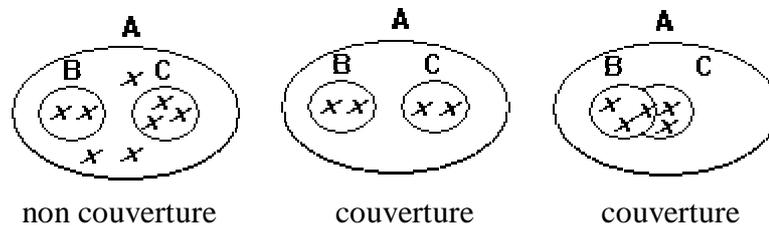
➤ **Règle de passage du MEA au MR**

La table issue de l'entité espèce hérite des clés primaires des entités genres qui deviennent clés étrangères. Ces attributs sont tous uniques. L'un d'entre eux, au choix, peut être clé primaire.

Les contraintes sur la généralisation

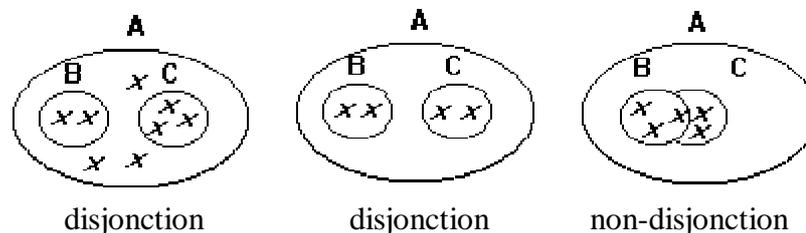
La notion de couverture

Il y a couverture si toute occurrence de A (sur-type) appartient au moins à l'un des sous types (B et C) :



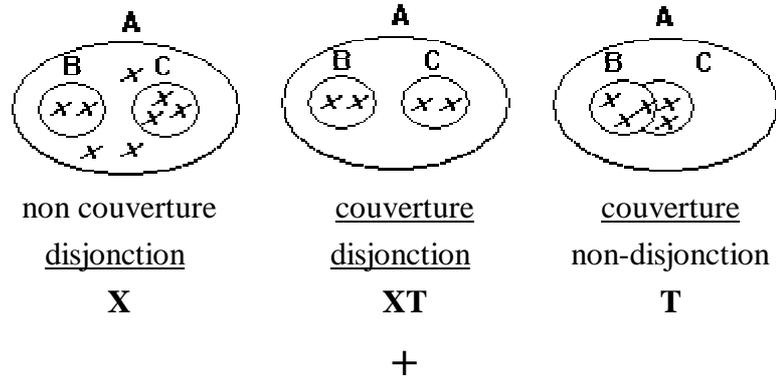
La notion de disjonction

Il y a disjonction quand toute occurrence du genre (A) appartient au plus à une seule espèce (B et C) :



Les 4 cas de spécialisation

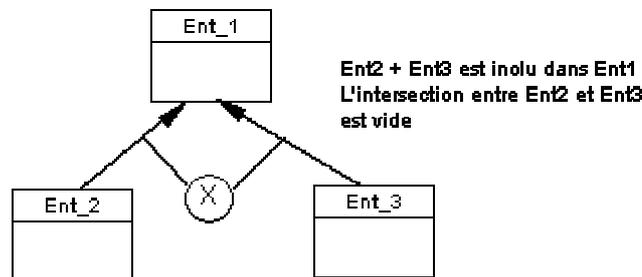
Les notions de couverture et de disjonction font apparaître 4 situations possibles pour une généralisation.



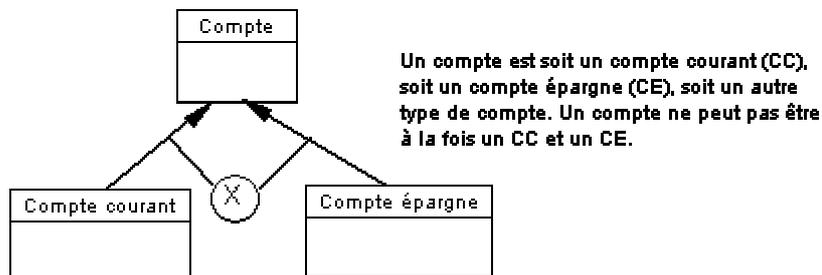
∅ : non-disjonction et non-couverture

Remarque : Si nous avons non-disjonction + non-couverture, il n'y a plus de contraintes à représenter.

X : disjonction et non-couverture (exclusivité)

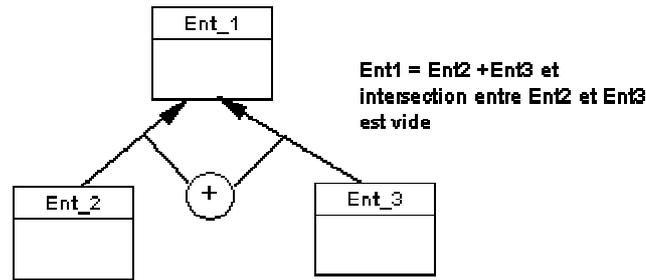


Exemple :

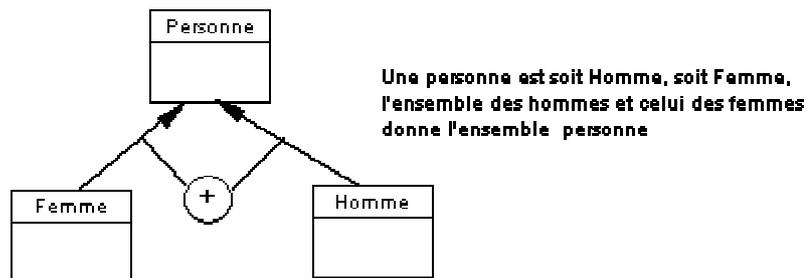


Dans ce cas, les occurrences de la table des comptes (C) correspondent aux comptes qui ne sont ni courant (CC), ni épargne (CE). Ces occurrences ne font référence ni au CC ni au CE, de même que ni les CC, ni les CE ne font référence aux C.

XT ou + : Disjonction et couverture (partition)

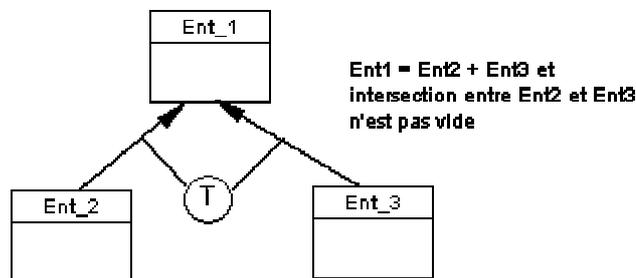


Exemple :

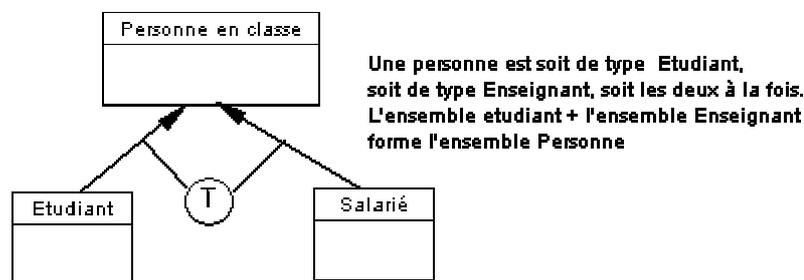


Dans ce cas, le système d'information doit empêcher la création d'occurrences dans la table des personnes.

T : non-disjonction et couverture (totalité)



Exemple :



Dans ce cas le système d'information doit imposer que les occurrences dans la table des personnes en classes fassent référence à un étudiant et à un salarié : ces occurrences correspondent aux personnes qui sont à la fois étudiant et salarié.

Ensemble : entité, table

On peut considérer les entités comme des ensembles. Les attributs de l'entité sont les attributs de l'ensemble. Chaque tuple est un élément de l'ensemble.

Sous-ensemble : espèce, entité-enfant

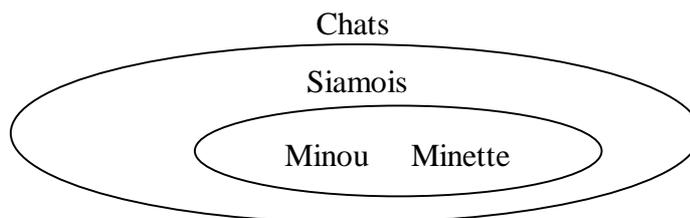
L'espèce est un sous-ensemble de l'ensemble dont on parle : le siamois est une espèce de chat. Quand on parle d'un sous-ensemble, on dit aussi : « entité-enfant » ou « entité-fille ».

Sur-ensemble : genre, entité-parent

Le genre est un « sur-ensemble » de l'ensemble dont on parle : le chat est le genre du siamois. Quand on parle d'un « sur-ensemble », on dit aussi : « entité-parent » ou « entité-mère ».

Elément

Chaque élément d'un ensemble est aussi élément des ensembles qui le contiennent : « Minou », c'est mon chat siamois : il appartient à l'ensemble des Siamois, mais aussi à l'ensemble des Chats. Minette c'est le chat siamois de la voisine.



Abstrait / Concret - Abstraction - Abstraite

Tous les noms d'ensemble sont abstraits : ce sont des abstractions.

Les seules choses concrètes, ce sont les éléments de l'ensemble, c'est-à-dire les tuples (Minou, mon chat siamois).

Abstraire, c'est remonter du concret à l'abstrait, donc de tuples à un ensemble qui les englobe. Par exemple de Minou et Minette au Siamois, ou de Minou et Minette au Chat.

Abstraire, c'est aussi remonter de l'espèce au genre. Par exemple du Siamois et de l'Angora au Chat.

Abstraire consiste à trouver des attributs communs à plusieurs ensembles ou à plusieurs choses concrètes pour définir un ensemble qui portera ces attributs et qui inclura les ensembles ou les choses concrètes en question.

Représentation ensembliste du MEA

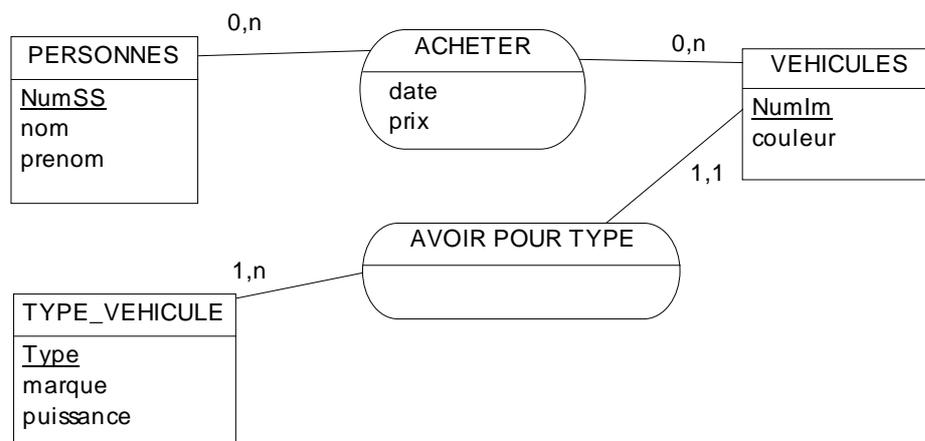
Présentation

On peut représenter le MEA avec une représentation ensembliste :

- Les ensembles correspondent à des entités.
- Les éléments des ensembles correspondent à des tuples.
- Les relations entre les ensembles correspondent aux associations.

MEA

Exemple des voitures avec les propriétaires :



MR

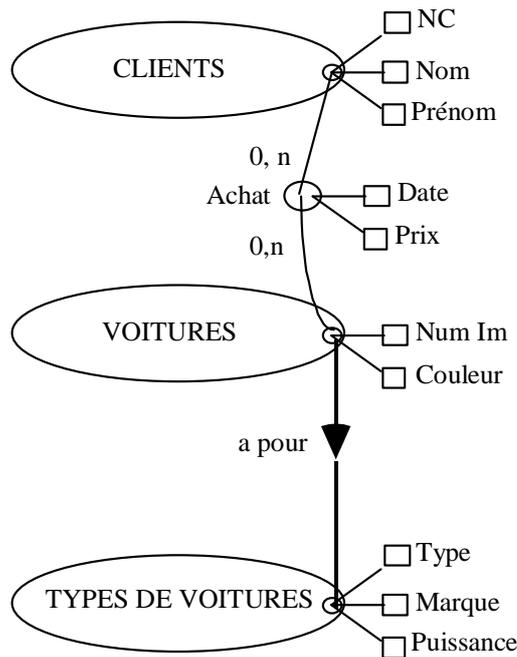
Client (**NC**, Nom, Prénom)

Véhicules (**NumIm**, Coul, #Type)

TypeVéhicule (**Type**, Marque, Puissance)

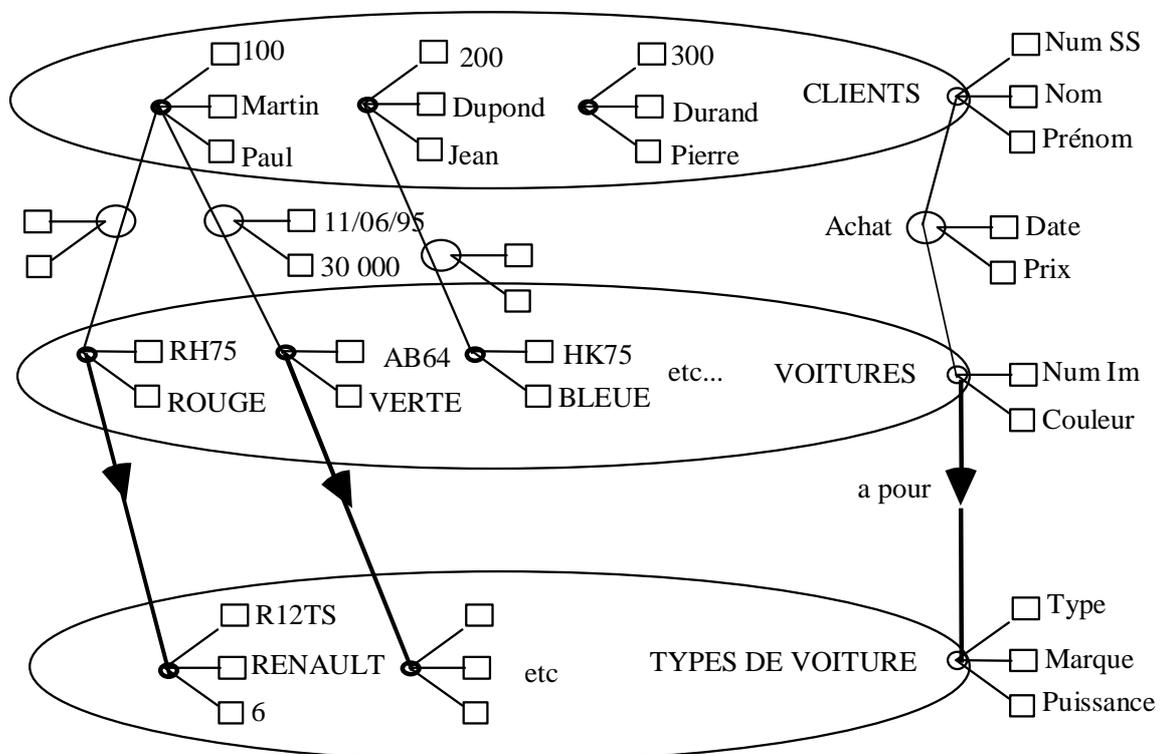
Ventes (#**NumIm**, #**NumSS**, Date, Prix)

Représentation ensembliste



Les attributs sont représentés par des petits carrés accrochés soit aux ensembles, soit aux relations entre les ensembles. Les relations hiérarchiques sont représentées par un lien rectiligne gras avec une flèche de 1 vers n.

Représentation des éléments des ensembles



Les éléments des ensembles sont représentés. Ils portent des attributs qui ont une valeur particulière. De même pour les relations entre les éléments.

4. Les entités complexes

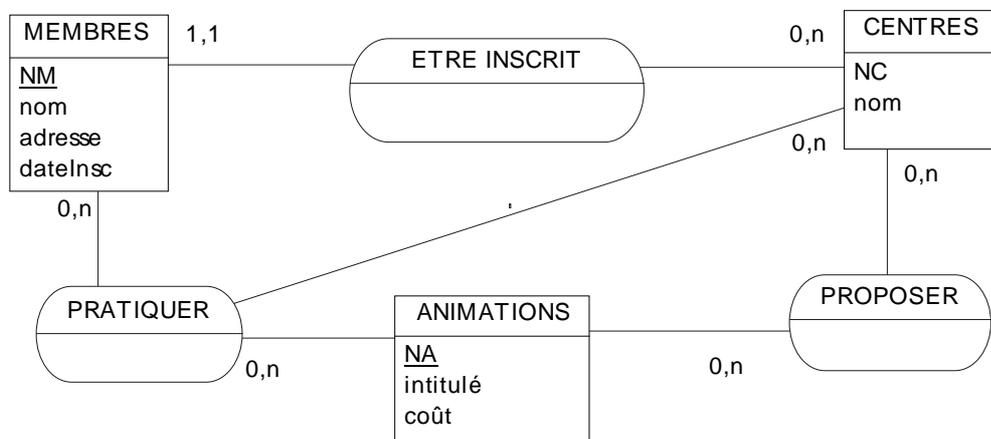
Présentation

Les entités complexes vont permettre de traduire les clés primaires complexes du modèle relationnel.

Exemple

La Mairie de Paris veut gérer les membres de ses centres d'animations et les animations qu'ils pratiquent. La Mairie possède plusieurs centres. Ils ont un nom, une adresse, un arrondissement. Chaque centre propose plusieurs animations. Une animation peut être proposée par plusieurs centres. Les animations ont un intitulé, un coût semestriel et un public (enfants, adulte, tout public). Un membre ne peut s'inscrire que dans un centre. Un membre peut pratiquer plusieurs activités dans plusieurs centres différents.

MEA



MR

MEMBRES (NM, nom, ad, datInsc, #NC)
CENTRES (NC, nom)
ANIMATIONS (NA, intitulé, coût)
PROPOSER (#NC, #NA)
PRATIQUER (#NM, #NC, #NA)

Critique

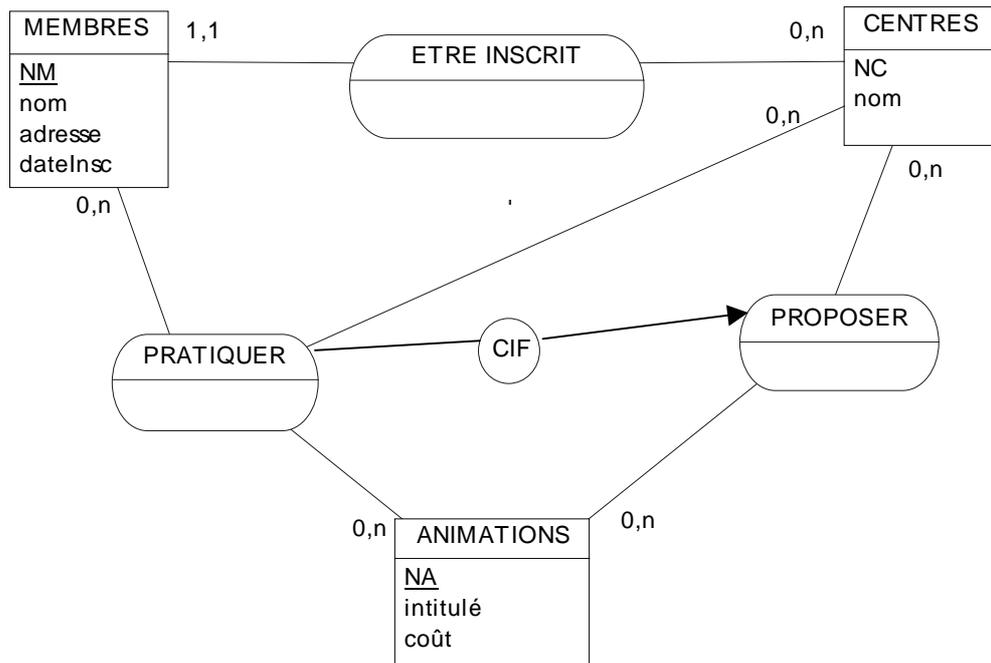
Le défaut du modèle est qu'il permet de pratiquer une activité qui n'est pas proposée !

Correction

PRATIQUER (#NM, # (NC, NA))

Le couple (NC, NA) est clé étrangère faisant référence à la clé primaire de PROPOSER.

MEA avec CIF



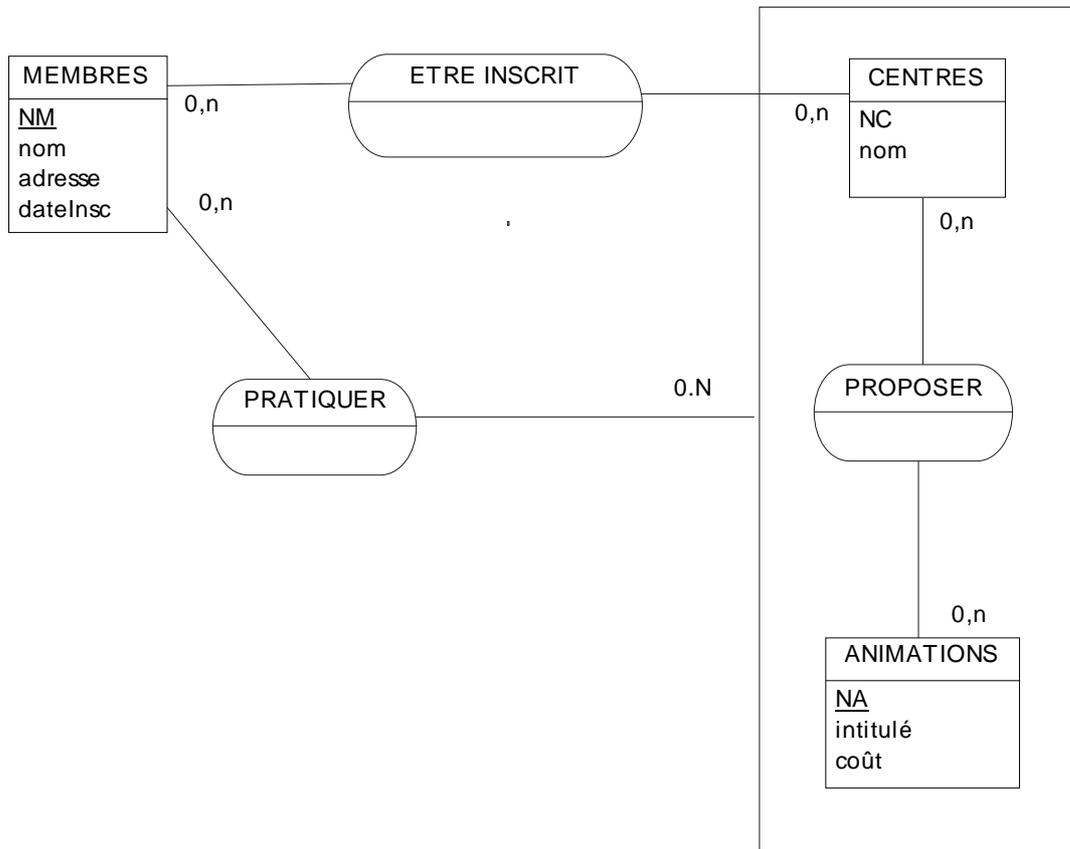
Entité complexe

Présentation

Une entité complexe est une entité qui regroupe une association non hiérarchique et les entités qu'elle associe.

Une entité complexe peut être associée à une autre entité.

Exemple



Interprétation

Les membres pratiquent les animations proposées par les centres.

MR

On obtient bien :

PROPOSER (#NC, #NA)

PRATIQUER (#NM, # (NC, NA))

Règles de passage du MEA avec entités complexes au MR

Avec un tel modèle, on peut appliquer les 4 règles de transformation avec quelques mises à jour et ainsi la règle 3 peut devenir entièrement mécanique. On ajoute une 5^{ème} règle pour les entités complexes.

Règle 1 - Entité : Chaque entité non abstraite devient une table-nom. Chaque attribut de l'entité devient un attribut de cette table.

Règle 2 – Association hiérarchique : Dans le cas d'une association hiérarchique, la clé primaire de la table issue de l'entité supérieure (simple ou complexe) devient attribut clé étrangère dans la table issue de l'entité inférieure (simple ou complexe).

Dans le cas d'une association hiérarchique réflexive, ce nouvel attribut doit être renommé et ajouté dans le dictionnaire des attributs.

Règle 3 – Association non hiérarchique : Une association non hiérarchique devient une table-verbe. Les clés primaires des tables issue des entités associées (simples ou complexes)

deviennent clés étrangères dans cette table. Les attributs de l'association deviennent attributs de la table. La clé primaire de la table-verbe est constituée par la concaténation des clés primaires des tables issues des entités associées (simples ou complexes).

Règle 4 – Association semi-hiérarchique : Dans le cas des associations semi-hiérarchiques : si elles portent des attributs, elles seront nécessairement traitées comme des associations non-hiérarchiques. Si elles ne portent pas d'attributs, on peut les traiter comme des associations hiérarchiques.

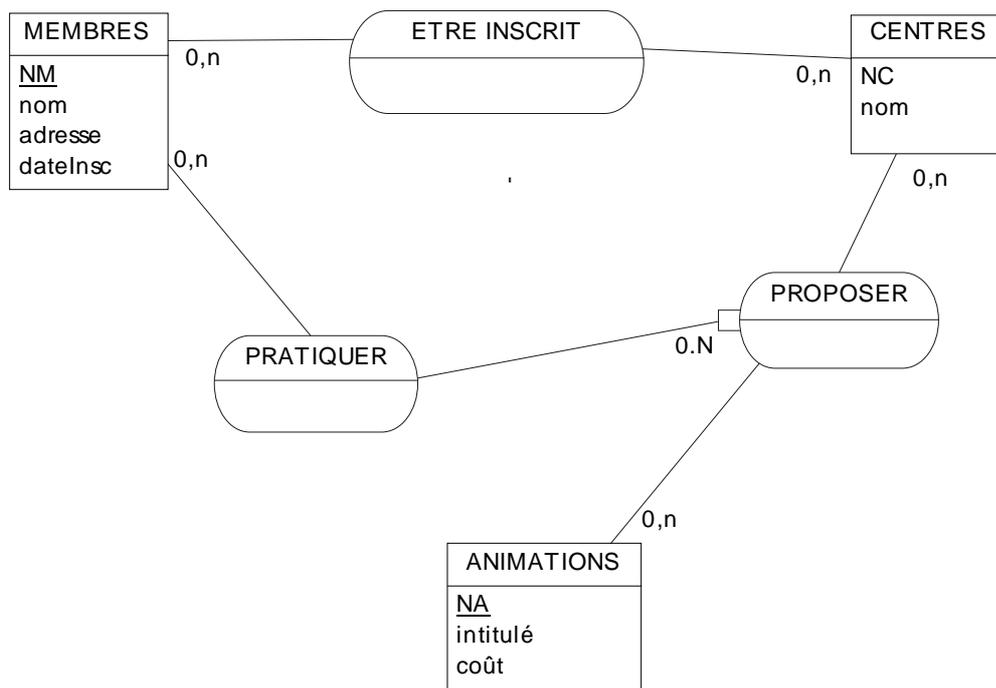
Règle 5 – Entité complexe : la table issue d'une entité complexe, c'est la table de l'association participant à l'entité complexe. L'entité complexe intervient ensuite dans les associations auxquelles elles participent (règles 3 et 4).

Autre formalisme : association d'association

Présentation

Une entité complexe peut être considérée comme une généralisation de la notion d'association. En effet, l'entité complexe est une façon de permettre aux associations de relier les entités mais aussi les associations, d'où la notion d'association d'associations.

Exemple



Interprétation

Dans ce MEA, on trouve une association qui relie une entité et une association. Pour une telle association, la patte qui la relie à une autre association doit se terminer par un petit carré pour qu'on sache dans quel sens lire le schéma (autrement quelle est l'association qui fait office d'entité complexe et quelle est celle qui joue le rôle d'association).

Règles de passage du MEA avec association d'association au MR

Avec un tel modèle, on peut appliquer les 4 règles de transformation avec quelques mises à jour et ainsi la règle 3 peut devenir entièrement mécanique.

Règle 1 - Entité : Chaque entité non abstraite devient une table-nom. Chaque attribut de l'entité devient un attribut de cette table.

Règle 2 – Association hiérarchique : Dans le cas d'une association hiérarchique, la clé primaire de l'entité ou de l'association supérieure devient attribut **clé étrangère** dans la table issue de l'entité ou de l'association inférieure.

Dans le cas d'une association hiérarchique réflexive, ce nouvel attribut doit être renommé et ajouté dans le dictionnaire des attributs.

Règle 3 – Association non hiérarchique : Une association non hiérarchique devient une table-verbe. Les clés primaires des tables issues des entités ou des associations associées deviennent clés étrangères dans cette table. Les attributs de l'association deviennent attributs de la table. La clé primaire de la table-verbe est constituée par la concaténation des clés primaires des tables issues des entités ou des associations associées

Règle 4 – Association semi-hiérarchique : Dans le cas des associations semi-hiérarchiques : si elles portent des attributs, elles seront nécessairement traitées comme des associations non-hiérarchiques. Si elles ne portent pas d'attributs, on peut les traiter comme des associations hiérarchiques.

5. Autres contraintes

Les contraintes de stabilité

Présentation

Les contraintes de stabilité concernent le cycle de vie des tuples.

Attribut stable (S)

Un attribut est stable si la première valeur qu'il reçoit ne peut pas être ultérieurement modifiée.

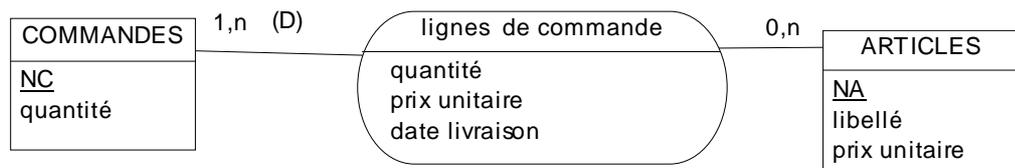
La stabilité d'un attribut correspond à la contrainte d'intégrité : non-modifiable.

Dans un MEA, on ajoute « (S) » derrière le nom de l'attribut.

Patte d'association définitive (D)

Une patte d'association est définitive si la suppression d'une occurrence d'une association implique la suppression de l'occurrence de l'entité associée et donc de toutes les occurrences des associations avec cette entité.

➤ Exemple



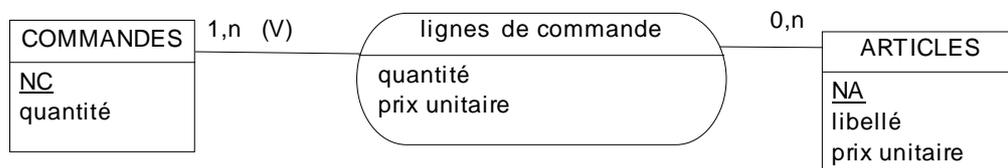
Une fois la ligne de commande ajoutée pour une commande donnée, on ne doit pas le supprimer, sauf à supprimer la commande et toutes les lignes de commande associées.

Mais la ligne de commande reste modifiable : on la mettra à jour à l'occasion de la livraison.

Patte d'association verrouillée (V)

Une patte d'association est verrouillée si pour l'entité concernée, toutes les associations sont créées en même temps et si elles sont stables et définitives.

➤ Exemple

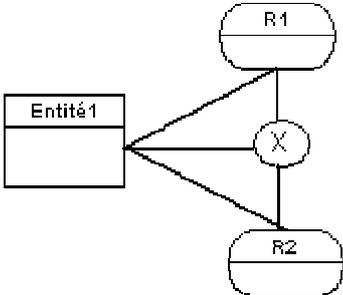


Dans cet exemple, la date de livraison ne fait pas partie de la ligne de commande. De ce fait, la ligne de commande peut être verrouillée : elle n'est plus modifiable.

Les contraintes sur associations

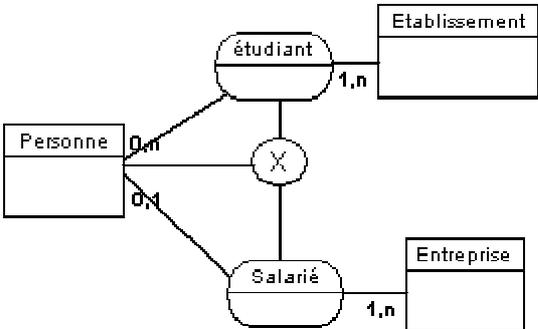
Ces contraintes concernent l'existence même des occurrences des associations. Elles sont à vérifier à chaque fois qu'on crée une nouvelle occurrence d'association.

X : la contrainte d'exclusion : non-couverture + disjonction



Si une occurrence de l'entité1 participe à la relation R1, elle ne doit pas participer à la relation R2, et réciproquement

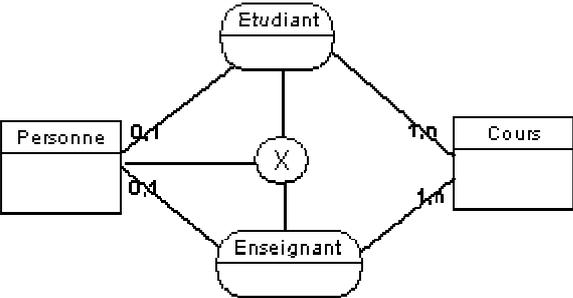
➤ *Exemple 1 :*



Une personne ne peut pas être à la fois étudiant et salarié, certaines personnes sont ni l'un ni l'autre.

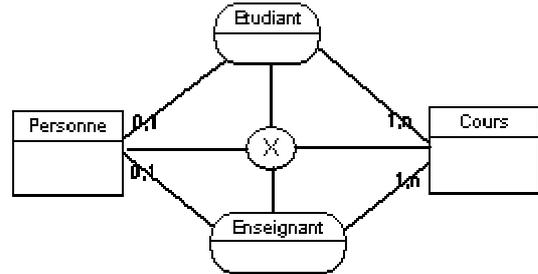
Dans ce cas, par exemple, le système vérifie quand on veut créer une occurrence de l'association salarié, que la personne concernée n'est pas étudiante.

➤ *Exemple 2 :*



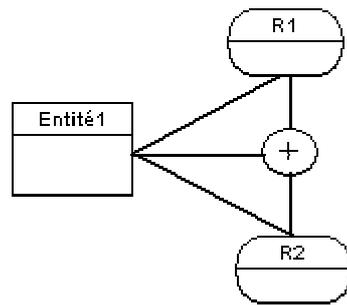
Une personne ne peut pas être à la fois étudiant et enseignant, même dans des cours différents

➤ *Exemple 3 :*



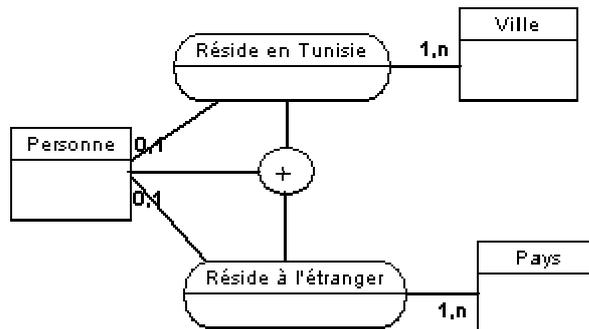
Une personne ne peut pas être à la fois étudiant et enseignant dans un même cours, mais il peut l'être dans des cours différents

XT ou + : la contrainte de partition : couverture + disjonction



Toute occurrence de l'entité1 participe au moins à l'une des relations R1 et R2 mais pas aux deux à la fois

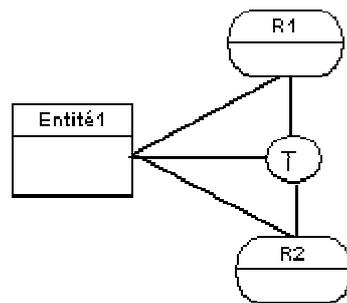
➤ *Exemple :*



Toute personne est soit résidente en Tunisie, soit résidente à l'étranger mais non pas aux deux à la fois

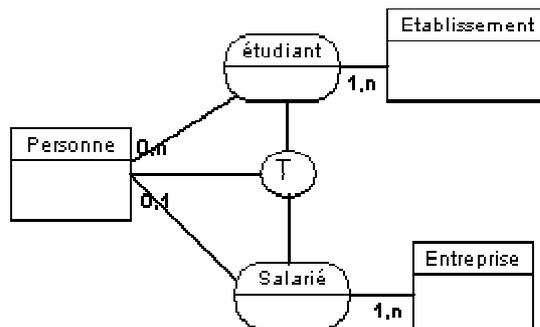
La contrainte de partition diffère de la contrainte d'exclusion par le fait que la personne est forcément associée une fois : la somme des cardinalités des associations sur lesquelles on pose la contrainte de partition (0,1) et (0,1) vaut (1, 1) avec la contrainte.

T : La contrainte de totalité : couverture + non-disjonction



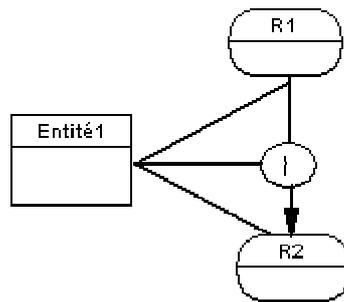
Toute occurrence de l'entité1 participe au moins à l'une des relations R1 et R2 ou les deux à la fois

➤ *Exemple :*



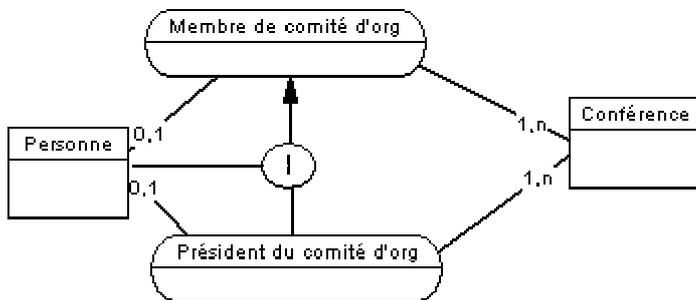
Toute personne est étudiant dans un établissement ou salarié dans une entreprise, ou les deux à la fois

I : la contrainte d'inclusion



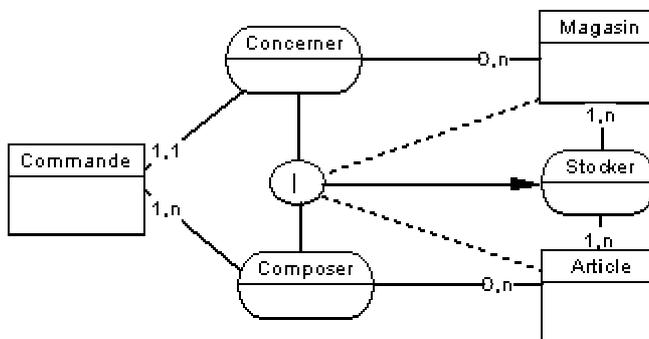
Si une occurrence de l'entité1 participe à R1 elle participe aussi à R2, mais pas réciproquement

➤ Exemple 1 :



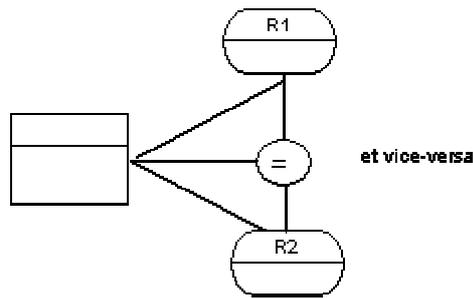
Le président de la comité d'organisation d'une conférence doit être un membre de ce comité d'organisation

➤ Exemple 2 :

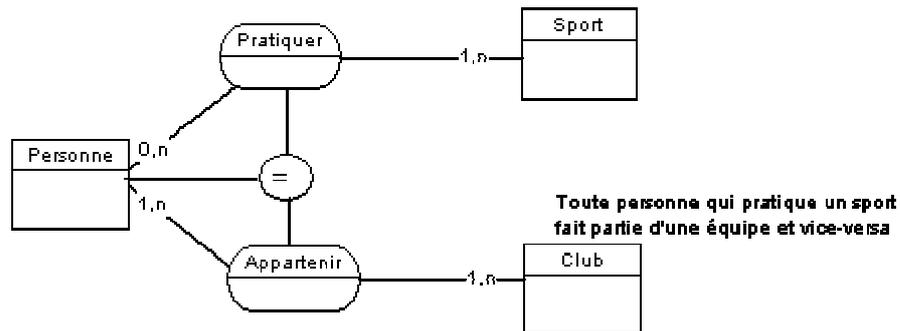


Un article qui entre dans la composition d'une commande, et celle ci concerne

= : la contrainte d'égalité



Exemple



Dans ce cas, on peut aussi considérer qu'on devrait créer une association entre les trois entités, ce qui nous permettrait de savoir dans quel club les personnes pratiquent les sports, et ce qui réglerait l'obligation de faire en sorte que toute personne qui pratique un sport fasse partie d'une équipe.

Dans l'exemple, on a :

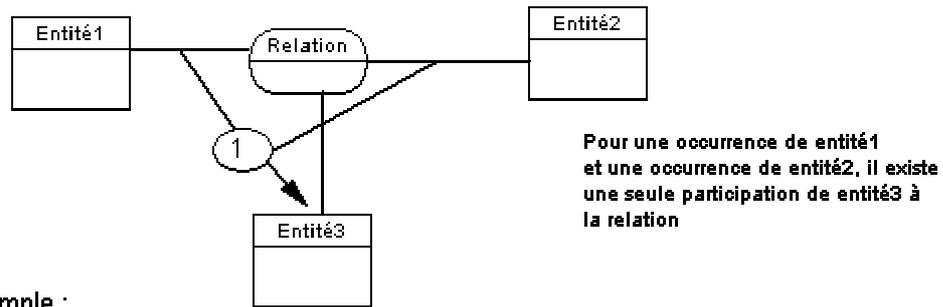
$P(P, Px) ; C(C, Cx) ; S(S, Sx) ; App(\#C, \#P) ; Prat(\#P, \#S)$

La contrainte d'égalité dit que quand je crée une occurrence App (cx, p1) je dois créer un Prat (p1, sx).

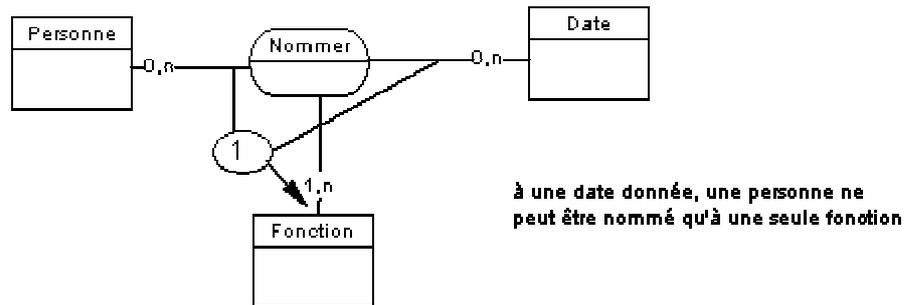
L'autre solution de modélisation consisterait à proposer le schéma suivant :

$P(P, Px) ; C(C, Cx) ; S(S, Sx) ; PratiqueDansUnClub(\#C, \#P, \#C)$

1 : la contrainte d'unicité



Exemple :

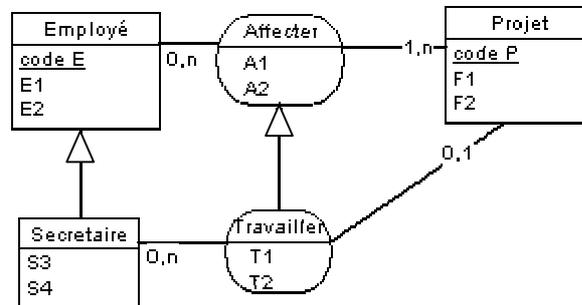


Dans ce cas, c'est comme si on avait une association hiérarchique entre l'association « nommer » et l'entité « fonction ».

La spécialisation des associations

Les types et sous type de relations concernent des restrictions de la relation à des sous types d'entités.

MEA



On suppose qu'il y a au plus une secrétaire qui est affectée au projet.

La relation travailler est une relation sous type et hérite de toutes les propriétés de la relation sur type affecter, et peut comporter des propriétés propres.

MR

Employés (NE, E1, E2)

Projets (NP, P1, P2)

Affecter (#NE, #NP, A1, A2)

Secrétaires (#NE, S3, S4)

Travailler (#(NE, NP), T1, T2)

Règle de passage du MEA au MR

La clé primaire de la table issue d'une association spécialisée non-hiérarchique fait référence à la clé primaire de la table issue de l'association-genre non-hiérarchique correspondante.

Remarque

Dans la table « Travailler », c'est le couple (NE, NP) qui est clé étrangère (et primaire). Il fait référence au couple (NE, NP) de « Affecter ».

6. Les données isolées

Présentation

Les données isolées sont des données qui ne sont pas reliées aux autres données de la BD.

Exemple

On pourrait avoir une entité « Entreprises » qui ne contiendra qu'une seule entité : celle de l'entreprise concernée, et qui permettra de conserver dans la BD toutes les informations relatives à l'entreprise : nom, adresse, n° Siret, etc.