

Petit guide d'analyse des données

Méthode MERISE

Niveau : débutant

L'analyse des données constitue le point de passage obligé de toute conception d'application mettant en œuvre un SGBDR (système de gestion de base de données relationnelle). La méthode MERISE, basée sur le modèle entité-association, est un outil simple et efficace, très répandue chez les développeurs français. La plupart des bases de données micro pour PC (dBase, Paradox, Foxpro, Access...) sont imprégnées de cette technique pour montrer les relations entre les tables au sein d'une base. Dans cet article, nous allons découvrir les bases et principes fondamentaux de MERISE à l'aide d'exemples et de cas concrets.

SOMMAIRE

1.	<i>Les travaux de Peter CHEN et d'Hubert TARDIEU</i>	<i>3</i>
2.	<i>Du vocabulaire et comment on le représente graphiquement.....</i>	<i>4</i>
3.	<i>La forme des liens</i>	<i>11</i>
4.	<i>Attributs d'associations, pour aller plus loin</i>	<i>12</i>
5.	<i>De la théorie à la pratique</i>	<i>14</i>
5.1.	<i>Transformation des entités (passer de l'entité à la table)</i>	<i>14</i>
5.2.	<i>Où placer les attributs d'association ?</i>	<i>16</i>
6.	<i>Conseils divers (et généreux pour aller plus loin)</i>	<i>18</i>
6.1.	<i>Généralisation (héritage) (ramifier les espèces)</i>	<i>18</i>
6.2.	<i>Personnalisation (ou les sous-modèles).....</i>	<i>19</i>
6.3.	<i>Regroupement d'entités (un truc à connaître pour éviter la redondance</i>	<i>19</i>
7.	<i>Exemples de MCD pour mieux comprendre</i>	<i>20</i>
7.1.	<i>Agence de location de films vidéo</i>	<i>20</i>
7.2.	<i>Location d'appartements pour une agence immobilière.....</i>	<i>21</i>
8.	<i>Divers</i>	<i>22</i>
8.1.	<i>Bibliographie</i>	<i>22</i>
8.2.	<i>Outils de modélisation</i>	<i>22</i>

1. Les travaux de Peter CHEN et d'Hubert TARDIEU

Le modèle entité-association est un outil (et une technique) d'analyse permettant de construire des schémas théoriques de raisonnement sur des applications tournant avec les bases de données relationnelles. Pour mettre en œuvre cette technique, en connaître les possibilités et les limites, il n'est pas nécessaire de posséder une base de donnée.

Bien connaître les règles simples des schémas entités-associations (aussi appelé entités-relations) permet d'affiner petit à petit une application apparemment simple, sans avoir besoin de la programmer, et par conséquent d'économiser du temps de conception tout en obtenant une plus grande souplesse au niveau de l'analyse.

Il existe des logiciels permettant de construire des schémas entités-associations et d'en analyser les conséquences logiques, puis de construire les tables associées aux modèles de manière entièrement automatique. Ces logiciels sont appelés AGL (atelier de génie logiciel) ou CASE suivant leur puissance. Les logiciels TRAMIS, AMC*Designor, SELECT... en sont des exemples.

Le modèle entité-association est apparu dans les travaux des chercheurs, entre 1972 et 1975 lors des travaux du français MOULIN puis de TARDIEU, TEBOUL... etc. Il a été rendu célèbre dans le monde entier par l'américain Peter CHEN, à la suite d'une publication intitulée "The Entity-Relationship Model" (ACM, Transaction on Database Systems, 1976).

A ce jour tous les spécialistes français et/ou latins du domaine de l'analyse orientée base de données se servent de ce modèle comme outil de communication des applications SGBDR. Il est présent de manière transparente ou plus visible, dans la plupart des logiciels de construction d'applications de bases de données comme ACCESS, PARADOX, ORACLE, SQL Server, Informix, Ingres, Sybase...

Il n'est en revanche pas adapté aux bases de données purement objet comme O2 de Ardent Software..., même si l'on admet la nouvelle dérive de MERISE orientée objet !

2. Du vocabulaire et comment on le représente graphiquement

Le modèle entité-association est constitué de deux éléments de base :

- Les entités, qui sont des regroupements d'informations, et possèdent des attributs (caractéristiques)
- Les associations qui sont les liens logiques entre les entités (et sont quantifiées par des cardinalités)

2.1. Les entités (des ensembles)

Ce sont des regroupements d'informations. Les informations contenues dans les entités (informations que l'on appelle "attributs") doivent être des informations variables, mais communes à une même classe d'objets.

Par exemple, si l'on considère l'entité "être humain" les informations communes aux être humains peuvent être :

- le nom
- le ou les prénoms
- la date de naissance
- le lieu de naissance
- le sexe
- l'adresse
- etc.

On considère souvent qu'il s'agit de "classes" d'entités.

Une entité donnée peut elle-même être constituée de sous-classes.

Par exemple, un être humain donné peut habiter au même endroit qu'un autre (si deux personnes vivent sous le même toit parce qu'ils sont mariés). Dans ce cas, l'adresse constitue une sous-classe de l'entité "être humain", c'est à dire une nouvelle entité à part entière.

D'un autre côté, il arrive souvent que plusieurs personnes résident au même endroit, sans même se connaître (cas d'un immeuble collectif par exemple). Dans ce cas on peut considérer l'adresse, comme une entité et la décrire de la manière suivante :

- Pays
- Région
- Département
- Rue
- etc.

On schématise une entité par un rectangle.

Exemple :



2.2. Les attributs (des caractéristiques)

Les attributs sont les caractéristiques décrivant les entités et doivent être représentés comme une liste de mots, la plus simple possible, dans le cadre de l'entité correspondante. On devra préciser le type des données attendues pour chaque attribut.

Exemple :

ÊTRE HUMAIN	
Nom	A32
Prénom	A25
Date de naissance	D
Lieu de naissance	A32
Sexe	BL

ADRESSE	
Ligne adresse 1	A32
Ligne adresse 1	A32
Ligne adresse 1	A32
Code Postal	A7
Ville	A32
Pays	A32

Les types associés aux attributs sont les suivants :

D	<i>Date</i>
Annn	<i>Caractères de longueur nnn</i>
BL	<i>Booléen (vrai / faux)</i>
T	<i>Temps</i>
DT	<i>Date Temps</i>
N	<i>Nombre</i>
S	<i>(Smallint) entier court</i>
S	<i>(Integer) entier</i>
Etc.	

2.3. Les associations (ou relations)

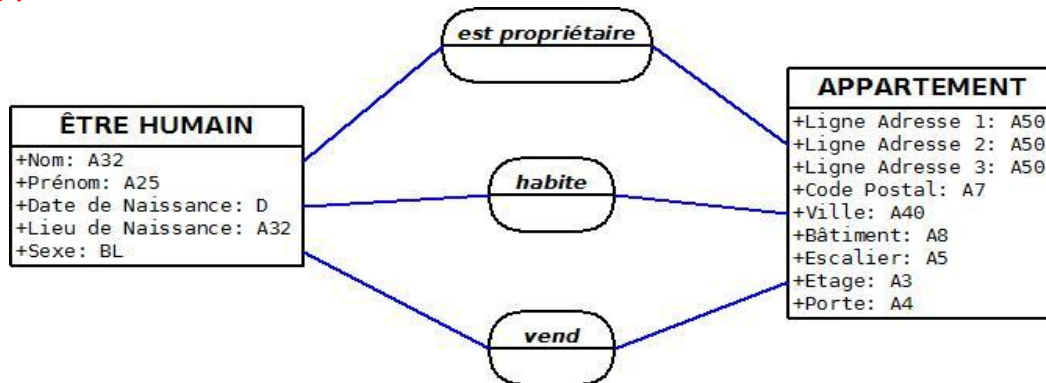
Ce sont des liaisons logiques entre les entités.

Elles peuvent être de nature factuelle, ou de nature dynamique.

Par exemple, une personne peut acheter un objet (action d'acheter), mais si l'on considère qu'une personne est propriétaire d'un objet, alors l'association entre l'objet et cette personne, est purement factuelle.

Les associations se représentent dans une ellipse (ou un rectangle aux extrémités rondes), reliée par des traits aux entités qu'elles lient logiquement.

Exemple :



2.4. Les cardinalités (ou "combien" ?)

Les cardinalités, au sens arithmétique du terme, permettent de dénombrer les éléments de l'entité d'arrivée en relation avec un élément de l'entité de départ, et vice versa.

Exemple :

Considérons le cas de l'association "habite" et les deux entités "être humain" et "appartement" du schéma précédent, les cardinalités minimales et maximales sont les suivantes :

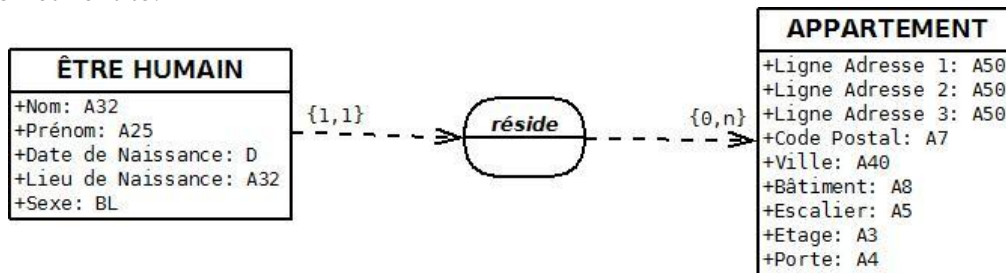
- sens "être humain" vers "appartement" : 1 (minimum) et 1 (maximum)
- sens "appartement" vers "être humain" : 0 (minimum) et n (maximum)

Ce qui signifie que dans cette modélisation un être humain réside dans un appartement et un seul à la fois, mais qu'un appartement peut se trouver vide ou être pourvu de plusieurs résidents.

Bien entendu tout être humain ne réside pas forcément dans un appartement, ce peut être dans une maison, à l'hôtel ou à la belle étoile (SDF...). Mais il convient de se concentrer sur ce que l'on doit modéliser et non sur l'univers entier.

En outre nous avons convenu qu'un même être humain ne pouvait résider dans plusieurs appartements à la fois (notion de « résidence principale » par exemple).

On note les cardinalités de chaque côté de l'association, sur les traits faisant la liaison entre l'association et l'entité.



Dès cet instant, on peut en déduire le type de relation parmi les types suivants :

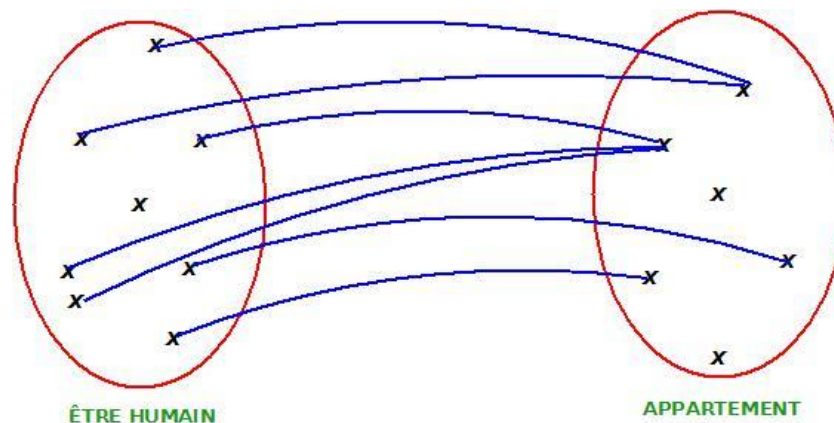
1/1, 1/n ou n/1 ou encore n/m.

Ici c'est une relation de type 1,n.

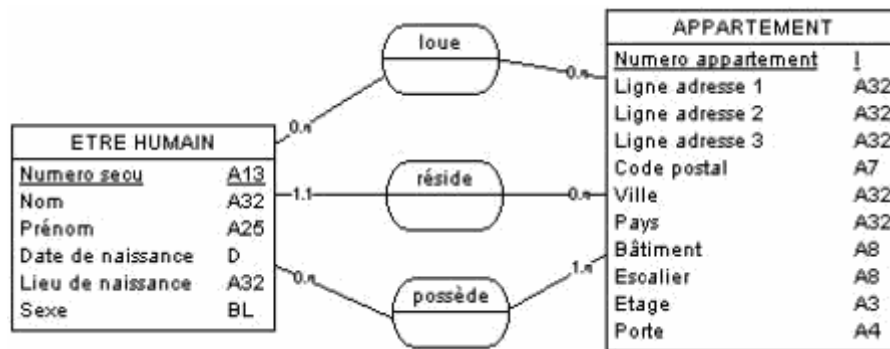
Pour déduire le type d'une relation, il suffit de récupérer les cardinalités maximales des deux côtés de l'association, sans tenir compte de leur valeur exacte.

On peut s'y aider à l'aide des diagrammes de Wen et des notions de mathématiques ensemblistes, bien connus de ceux qui ont pratiqués les maths modernes dès leur plus jeune âge...

Entrent alors en jeu les notions de bijection, interjection, surjection...



Attention ! Des relations différentes entre mêmes entités peuvent posséder des cardinalités différentes; c'est même souvent le cas.

Exemple :

La relation « loue » est de type n:m.

La relation « réside » est de type 1:n.

La relation « possède » est de type n:m.

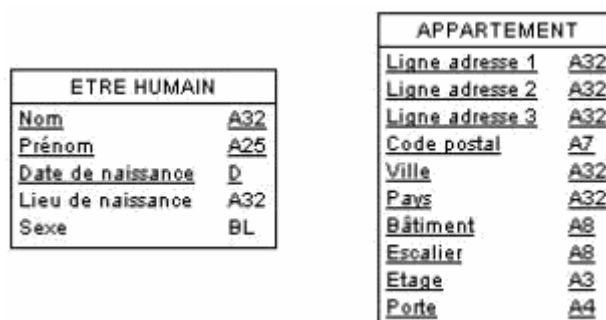
2.5. Clef d'une entité (la base de la relation)

C'est un attribut (ou un ensemble d'attributs) qui permet (tent) de distinguer un élément de l'entité de manière unique et sans aucune ambiguïté par rapport à l'ensemble des autres éléments, et à l'univers de tous les éléments qui peuvent entrer un jour ou l'autre dans cette entité.

Par exemple la clef de l'entité "être humain" pourrait être le nom. Mais comme le cas d'homonymie est assez fréquent, surtout lorsque l'on manipule des fichiers volumineux, alors cet attribut constitue une mauvaise clef en général.

En revanche, il n'est pas impossible que la clef d'une entité soit composée de plusieurs attributs. Par exemple, la clef de l'entité "être humain" pourrait être le nom et le prénom. Cependant il n'est toujours pas impossible d'avoir deux personnes dont le nom et le prénom soient identiques...

On note qu'un attribut est une clef en le soulignant dans le schéma entité association. Si c'est une clef composée, alors plusieurs entités seront soulignées.

Exemple :

On voit ici que dans le cas de l'entité "appartement" tous les attributs sont utilisés pour composer la clef. Cette clef naturelle n'étant pas pratique, il est plus judicieux de créer un nouvel attribut qui servira expressément de clef à l'association.

Pour l'entité "être humain", on pourrait se servir du numéro de sécurité sociale (plus exactement du numéro INSEE), comme clef de l'entité. En revanche, pour ce qui est de l'entité "appartement" il est conseillé de créer un nouvel attribut clef qui serait, par exemple, un numéro.

Exemple :

ETRE HUMAIN	
<u>Numero secu</u>	A13
Nom	A32
Prénom	A25
Date de naissance	D
Lieu de naissance	A32
Sexe	BL

APPARTEMENT	
<u>Numero appartement</u>	I
Ligne adresse 1	A32
Ligne adresse 2	A32
Ligne adresse 3	A32
Code postal	A7
Ville	A32
Pays	A32
Bâtiment	A8
Escalier	A8
Etage	A3
Porte	A4

Pour une entité de type « Voiture » il pourrait être fait usage de l'immatriculation du véhicule comme clef de l'entité.

MAIS... lisez attentivement ce qui suit !**2.5.1. Discussion sur la qualité d'une clef**

De manière générale il convient de limiter les clefs composées. **Chaque fois que l'on aura le choix entre la création d'une clef numérique, et une clef naturelle mais composée, il sera préférable de créer une clef numérique, à de très rares exceptions près.**

En effet les SGBDR sont plus « à l'aise » lorsqu'ils ont à manipuler des clefs purement numériques. De plus une clef est un concept purement informatique.

Par exemple le n° de sécurité sociale est une mauvaise clef en générale : un individu étranger qui arrive sur le sol français est doté d'une immatriculation provisoire et ne connaît son numéro définitif que plusieurs mois après avoir rempli les conditions requises par l'administration.

Par exemple l'immatriculation d'un véhicule est une mauvaise clef : en effet, du fait de la fiscalité sur les véhicules à moteur (et en particulier les vignettes), les sociétés n'hésitent pas à faire immatriculer leur parc de véhicules dans le département où les taxes sont les moins élevées (le 51). Cette immatriculation peut donc être amenée à changer.

Or toute clef évolutive est un danger pour le système informatique : si la valeur de la clef change, nous verrons qu'il faut la modifier dans tous les fichiers dans laquelle elle est référencée. Il est vrai que certains SGBDR autorisent automatiquement ce genre de manœuvre, mais cette automatisation très contraignante pénalise lourdement le fonctionnement du SGBDR.

On veillera donc à prendre une clef totalement indépendante des attributs ordinaires de l'entité.

De plus une clef a intérêt à être la plus courte possible afin que son stockage soit limité à quelques octets dans les fichiers et donc le temps de recherche le plus rapide possible.

Le plus simple consiste donc à introduire dans le descriptif de l'entité une clef strictement « informatique » qui se résumera en général à un numéro (entier long) que l'on pourra incrémenter automatiquement.

Parfois, il arrive qu'une clef ait été créée spécifiquement pour les besoins des systèmes informatiques. C'est en particulier le cas des indicatifs téléphoniques (33 pour la France) ou des codes de pays (F pour France). Dans ce dernier cas on devra alors utiliser la clef commune, comme clef de l'entité.

2.5.2. La technique de la double clef

Une technique éprouvée consiste à introduire une double clef dans toutes les tables : la clef « informatique » et une clef « utilisateur ».

La clef informatique est l'index primaire de la table et doit posséder les caractéristiques suivantes :

- purement numérique (par exemple un entier long)
- unique bien entendu
- obligatoire
- sans mise à jour en cascade
- avec intégrité référentielle
- générée automatiquement
- invisible pour l'utilisateur

La clef utilisateur doit être assez « souple », c'est à dire posséder dans la mesure du possible, les caractéristiques suivantes :

Index secondaire unique

Obligatoire

Utiliser un jeu de caractère réduit s'il s'agit d'un format alpha (par exemple les 26 lettres majuscules de l'alphabet et les chiffres de 0 à 9)

Limité à une faible taille (16/32 octets - 16 caractères maximum)

La présentation des données devant toujours s'effectuer suivant l'ordre établis par la clef utilisateur à défaut d'autre spécification.

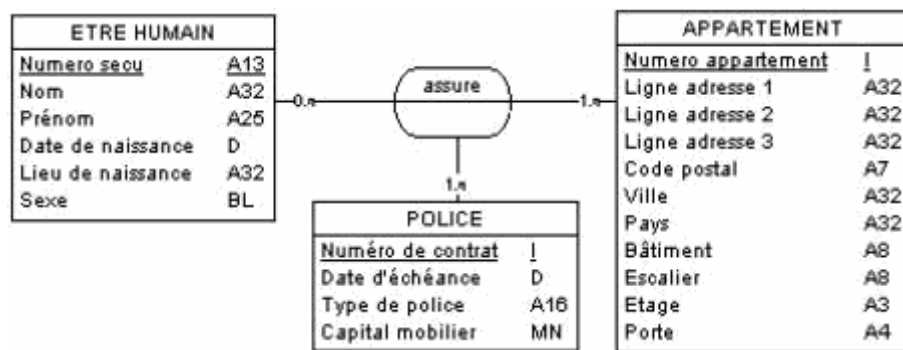
3. La forme des liens

La plupart des associations sont de nature binaire, c'est à dire composées de deux entités mise en relation par une ou plusieurs associations.

C'est le cas par exemple de l'association "est propriétaire" mettant en relation "être humain" et "appartement".

Cependant il arrive qu'une association concerne plus de deux entités (on dit alors qu'il s'agit d'association "n-aires").

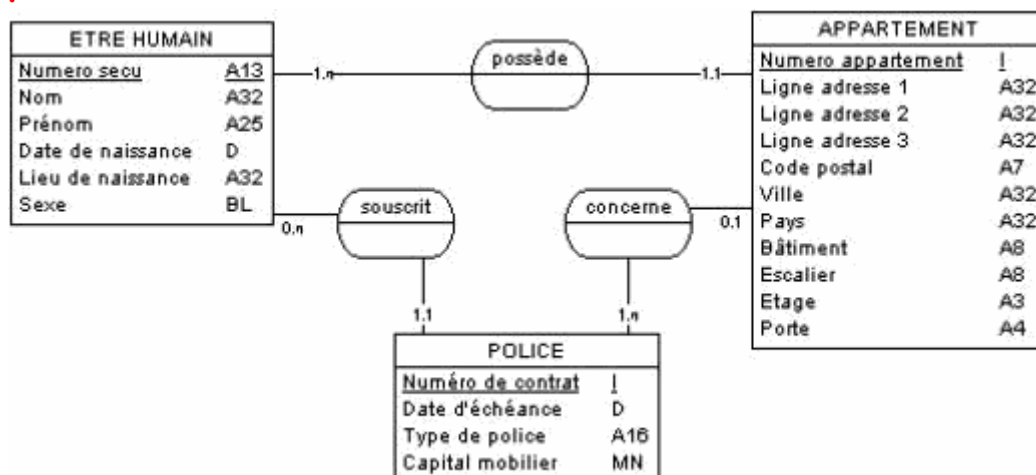
Exemple :



Mais dans ce cas il y a de grandes difficultés pour exprimer les cardinalités.

On aura tout intérêt à essayer de transformer le schéma de manière à n'obtenir que des associations binaires.

Exemple :

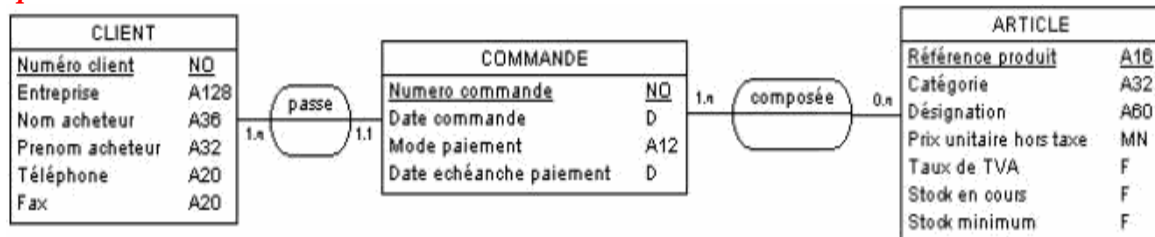


4. Attributs d'associations, pour aller plus loin

Il arrive parfois que l'on soit obligé de munir d'attributs des associations.

Considérons par exemple, que nous voulons modéliser les relations existant entre les entités "client", "commande" et "article" :

Exemple :

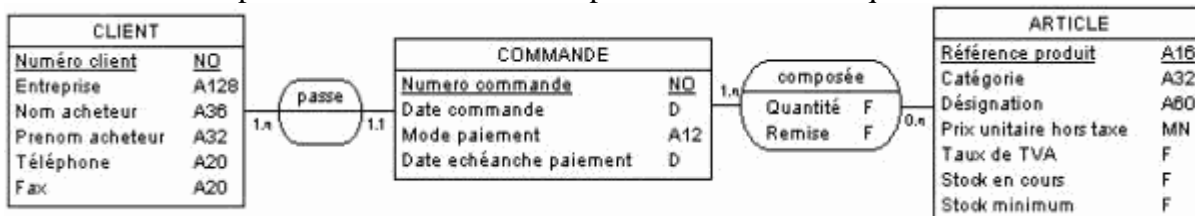


Mais comment dans ce schéma introduire l'attribut "quantité" et plus encore l'attribut "réduction" dont on voudrait qu'il puisse s'appliquer à chacun des articles d'une commande de manière différente ?

En effet si l'on introduit l'attribut quantité à l'entité COMMANDE, chaque ligne de la commande se verra dotée de la même quantité...

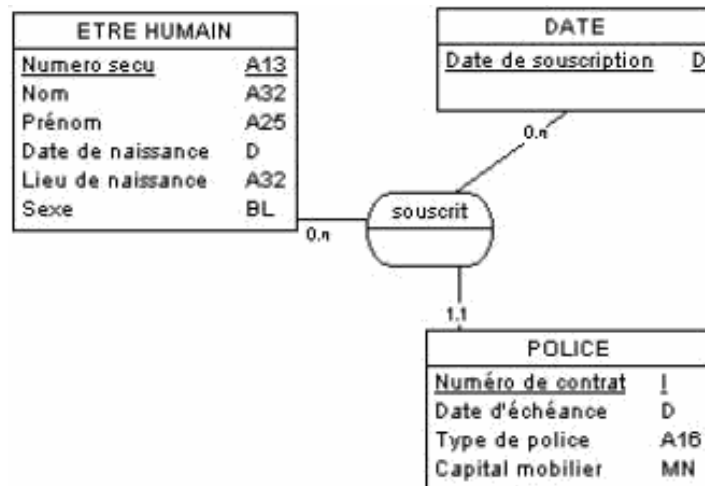
D'autre part si l'on introduit l'attribut quantité à l'entité ARTICLE alors chacun des article se verra doté de la même quantité quelque soit la commande...

La solution est de pourvoir l'association "composée" des attributs "quantité" et "réduction" :



Il arrive dans certains cas que l'attribut "date" soit d'une importance capitale, notamment dans les applications SGBDR portant sur la signature de contrats à échéance ou dans la durée (assurance par exemple).

Il n'est pas rare alors que le seul attribut "date" constitue à lui seul une entité.

Exemple :

On appelle alors cela une entité temporelle. Une entité temporelle possède souvent un seul attribut, mais dans le cas où elle possède plusieurs attributs (année, mois, jour, heure, minute, seconde...), l'ensemble de ces attributs constitue alors la clef de l'entité.

Mais dans ce cas on peut aussi retirer cette entité et introduire la date en tant qu'attribut de l'association "souscrit".

5. De la théorie à la pratique

Ce que nous venons de voir concerne l'analyse conceptuelle des données, c'est à dire un niveau d'analyse qui s'affranchi de toutes les contraintes de la base de données sur lequel va reposer l'application. Une fois décrit sous forme graphique, ce modèle est couramment appelé MCD pour "Modèle Conceptuel des Données".

Dès lors, tout MCD peut être transformé en un MPD ("Modèle Physique des Données") c'est à dire un modèle directement exploitable par la base de données que vous voulez utiliser...

Tout l'intérêt de cet outil d'analyse est de permettre de modéliser plus aisément les relations existant entre les entités et d'automatiser le passage du schéma muni d'attributs aux tables de la base de données pourvues de leurs champs.

Voici maintenant les règles de base nécessaire à une bonne automatisation du passage du MCD au MPD :

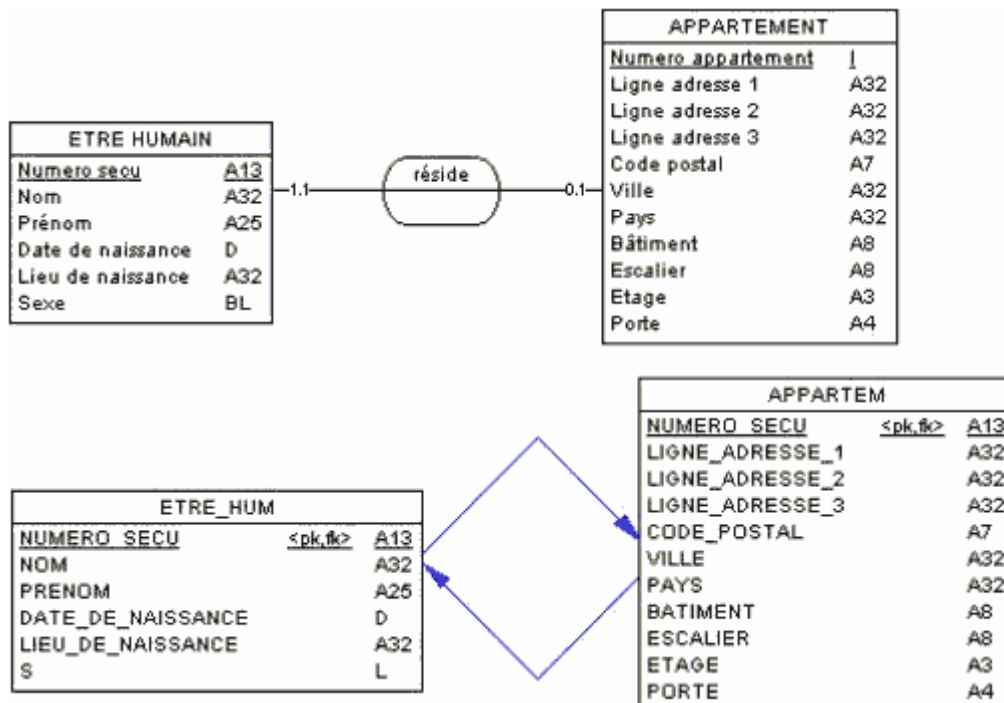
5.1. Transformation des entités (passer de l'entité à la table)

Règle n°1 : toute entité doit être représentée par une table.

5.1.1. Relation de type 1:1 (la voix de la simplicité)

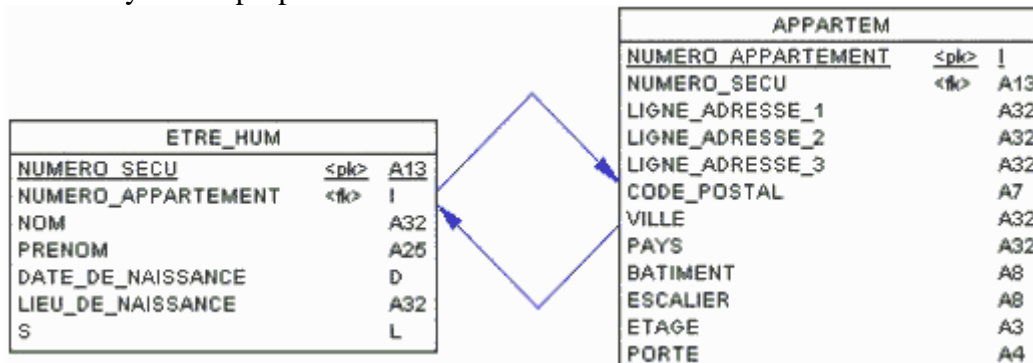
Règle n°2 : Dans le cas d'entités reliées par des associations de type 1:1, les tables doivent avoir la même clef.

Exemple :



NOTA : on aurait pu choisir l'autre clef comme clef commune, à savoir le n° d'appartement.

Bien que certains systèmes proposent une autre solution :

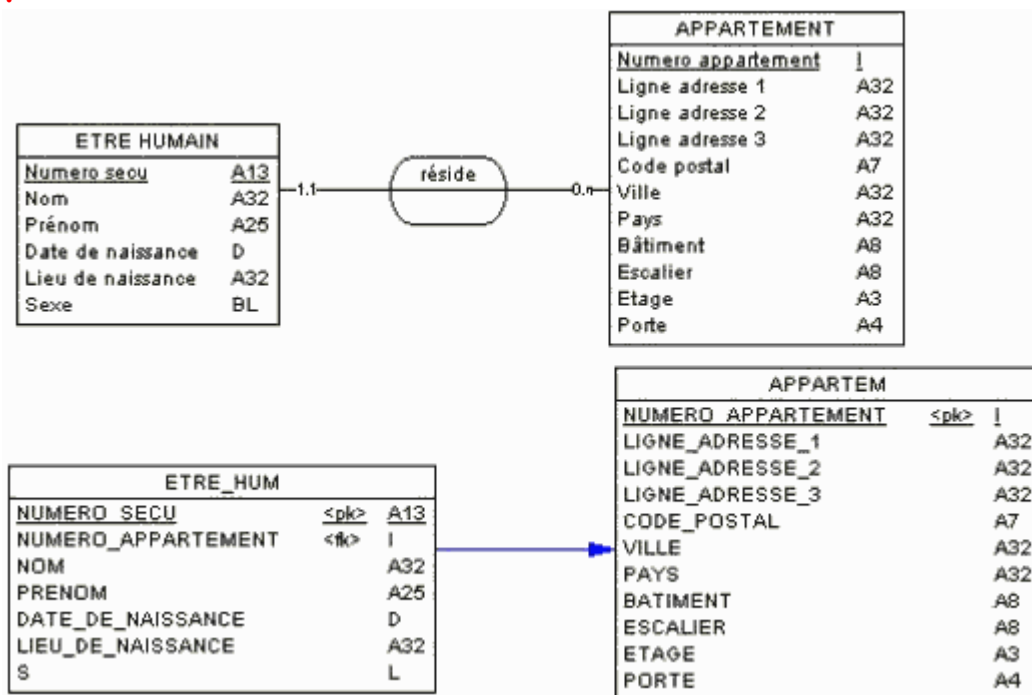


Dans ce cas une étude approfondie de la solution à adopter est nécessaire, mais ce type de relation est en général assez rare et peu performante...

5.1.2. Relation de type 1:n (maître et esclave)

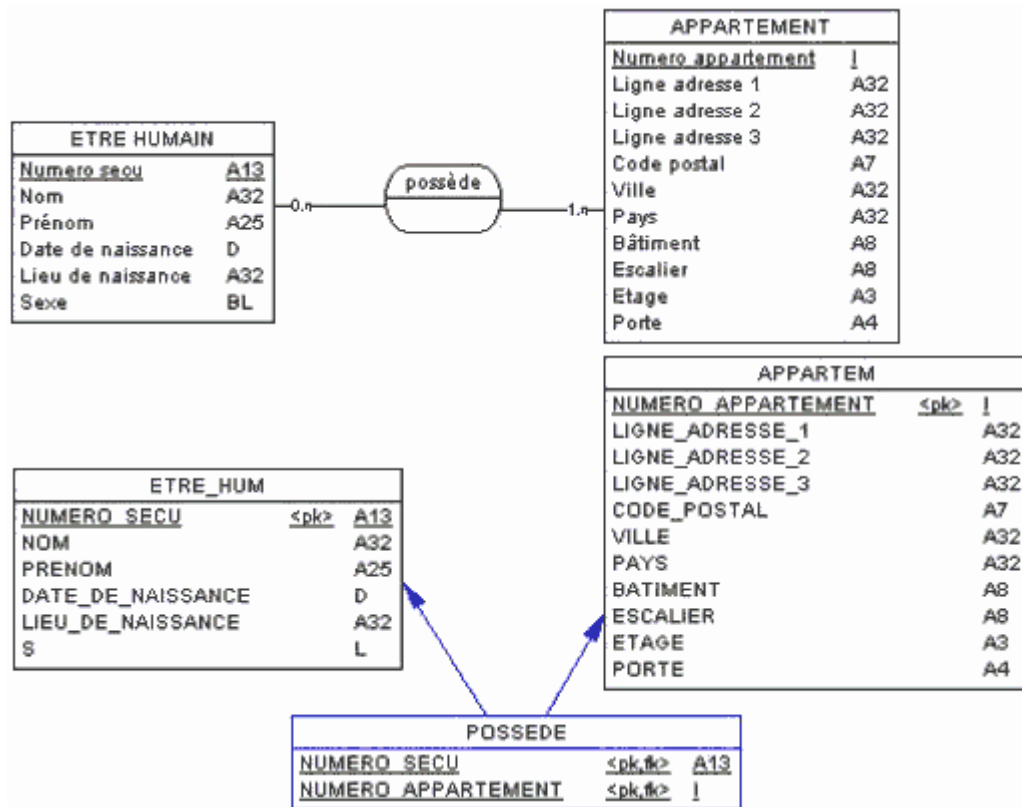
Règle n°3 : Dans le cas d'entités reliées par des associations de type 1:n, chaque table possède sa propre clef, mais la clef de l'entité côté 0,n (ou 1,n) migre vers la table côté 0,1 (ou 1,1) et devient une clef étrangère (index secondaire).

Exemple :



5.1.3. Relation de type n:m (plusieurs à plusieurs)

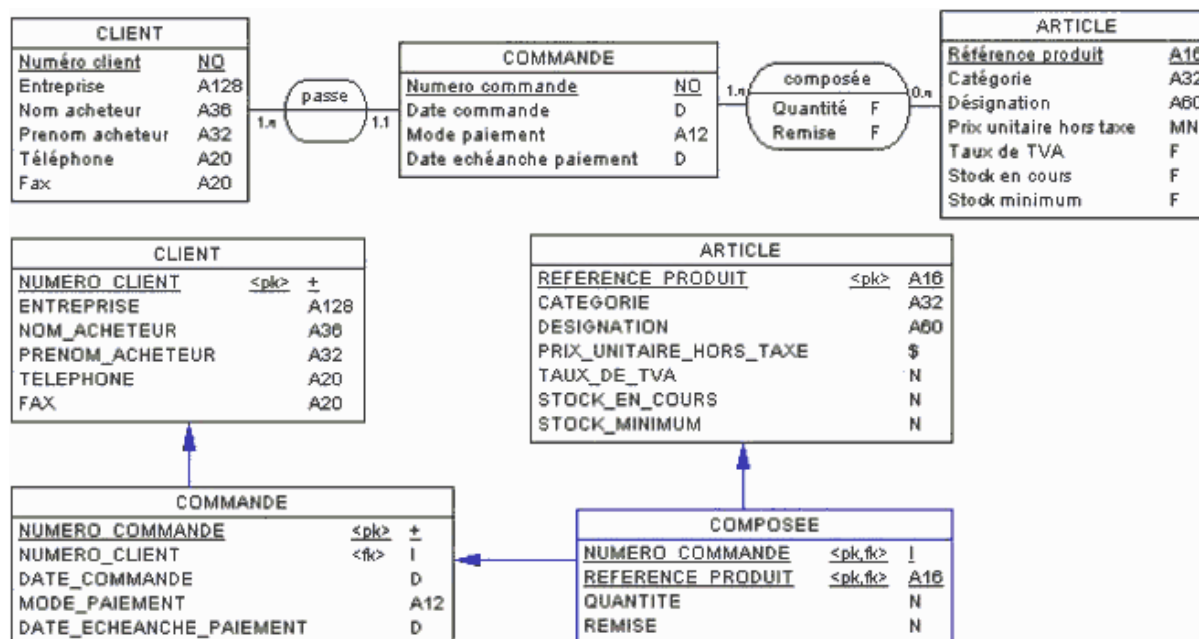
Règle n°4 : Dans le cas d'entités reliées par des associations de type n:m, une table intermédiaire dite table de jointure, doit être créée, et doit posséder comme clef primaire une conjonction des clefs primaires des deux tables pour lesquelles elle sert de jointure.

Exemple :**5.2. Ou placer les attributs d'association ?****Règle n°5 : Cas des associations pourvues d'au moins un attribut :**

- ✚ si le type de relation est n:m, alors les attributs de l'association deviennent des attributs de la table de jointure.
- ✚ si le type de relation est 1:n, il convient de faire glisser les attributs vers l'entité pourvue des cardinalités 1:1.
- ✚ si le type de relation est 1:1, il convient de faire glisser les attributs vers l'une ou l'autre des entités.

Pour synthétiser toutes ces règles, voici un exemple de modélisation d'une application. En l'occurrence il s'agit d'un service commercial désirant modéliser les commandes de ses clients.

Exemple :



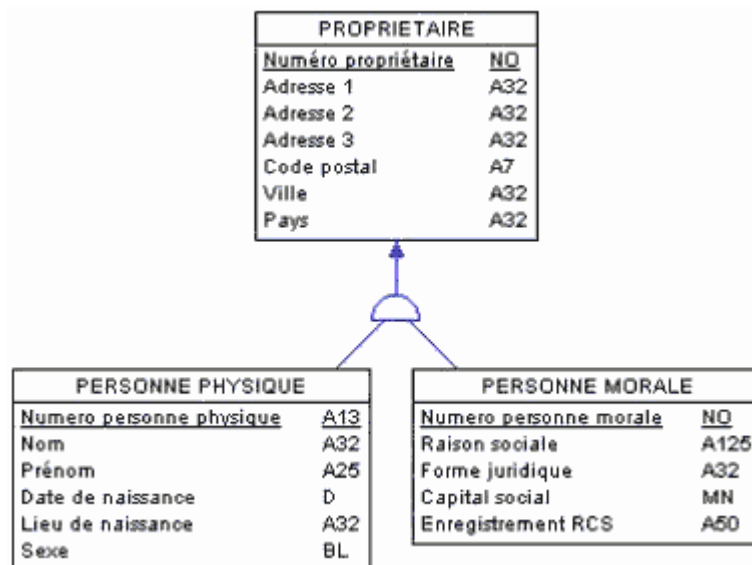
6. Conseils divers (et généreux pour aller plus loin)

6.1. Généralisation (héritage) (ramifier les espèces)

Dans le schéma ci-dessous, les entités "Personne physique" (des êtres humains) et "Personne morale" (des sociétés, associations, collectivités, organisations...) sont généralisées dans l'entité "Propriétaire".

On dit aussi que l'entité "Propriétaire" est une entité parente et que les entités "Personne morale" et "Personne physique" sont des entités enfants, car il y a une notion d'héritage...

Exemple :

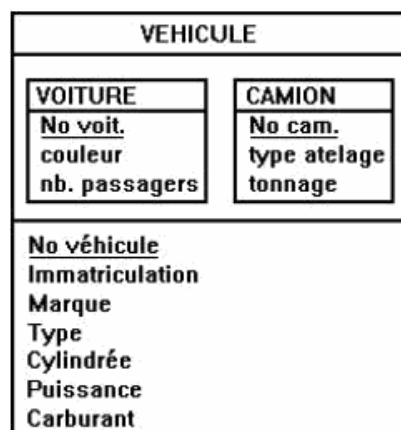


Par exemple une entité "Etre humain" est une généralisation pour toute entité faisant appel à une personne, comme les entités "Etudiant", "Client", "Artiste", "Souscripteur", "Patient", "Assujetti"...

On les appelle aussi "entités-génériques".

Certains ateliers de modélisation représentant les données sous la forme d'entités « encapsulées ».

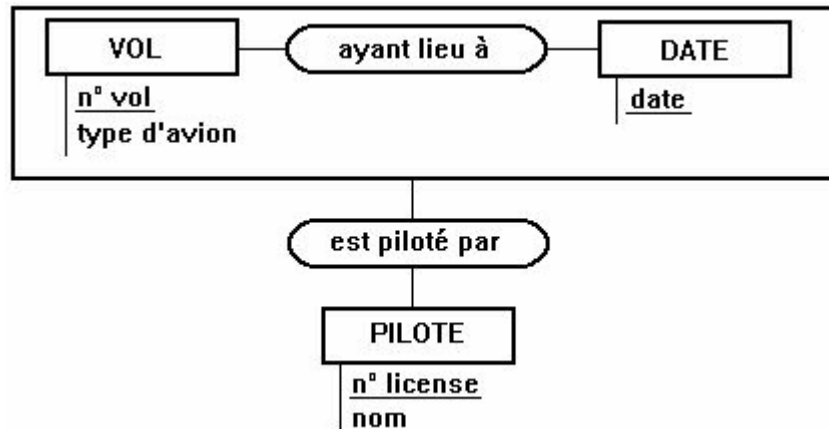
Exemple :



6.2. Personnalisation (ou les sous-modèles)

Une personnalisation est un regroupement dans une super entité de plusieurs entités munies d'une ou de plusieurs associations.

Par exemple, une compagnie d'aviation proposant des vols peut modéliser le planning des pilotes par le schéma suivant :



6.3. Regroupement d'entités (un truc à connaître pour éviter la redondance)

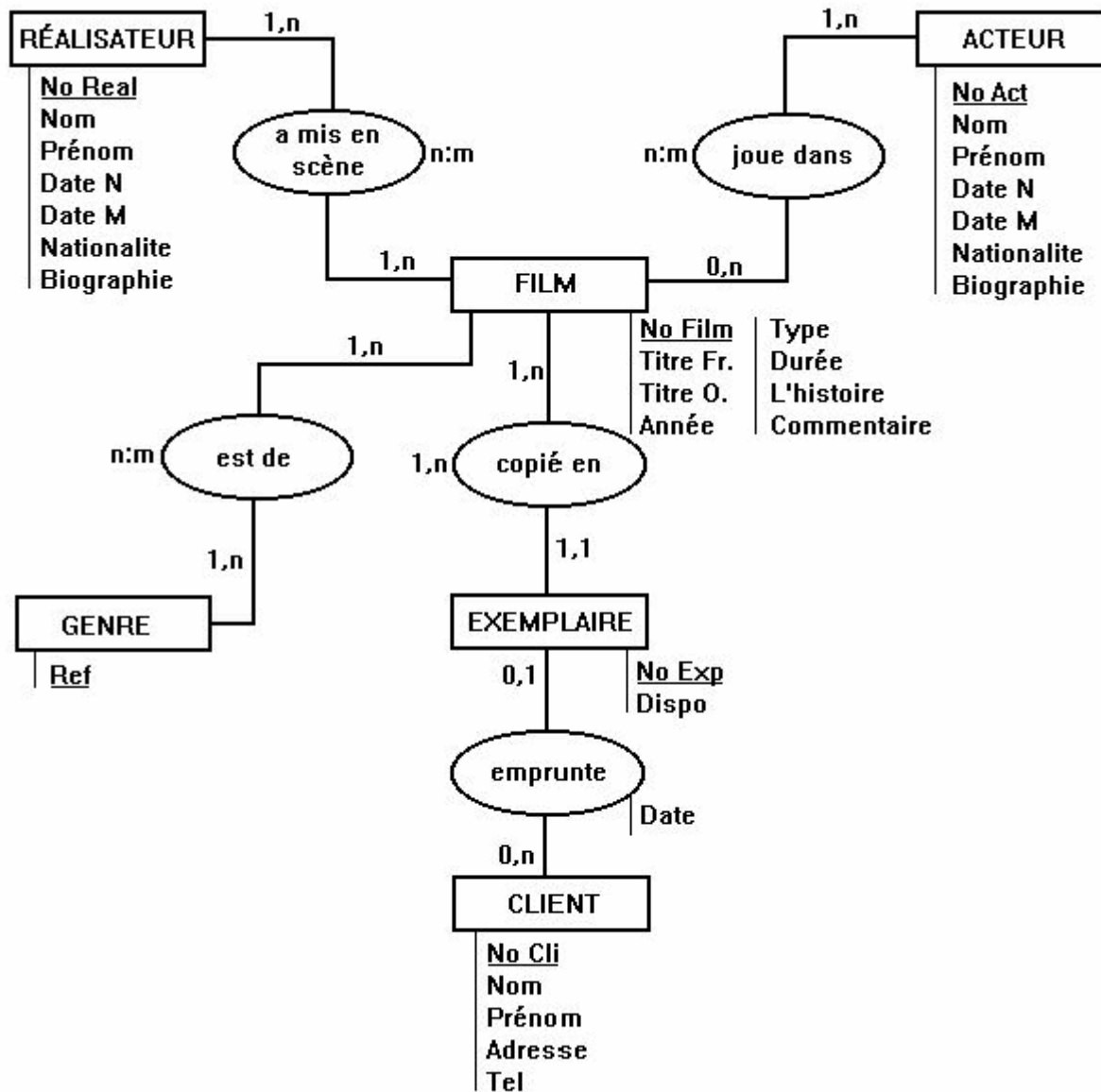
Comme toute technique, le schéma entité-association possède des limites et des contraintes que seuls l'expérience et le bon sens peuvent permettre d'éliminer.

Il arrive parfois que certaines entités apparaissent comme redondantes. Dans ce cas, et pour gagner de la place en matière de stockage de l'information, il convient de regrouper ces entités dans une seule et même table du SGBDR en ajoutant un champ supplémentaire à cette table de manière à permettre de distinguer les entités du schéma théorique.

Par exemple si l'on désire modéliser une gestion de compact-disc on peut créer une entité "Compositeur" et une entité "Interprète". Mais on constate qu'une grande majorité de compositeurs sont leurs propres interprètes, ce qui signifie qu'une même personne peut se trouver présente dans les deux entités. Pour résoudre ce problème il suffit de construire une seule table pour les deux entités (par exemple une table "MUSICIEN") et d'y ajouter un champ permettant de distinguer le type de "musicien" : compositeur ou interprète ou les deux.

7. Exemples de MCD pour mieux comprendre

7.1. Agence de location de films vidéo

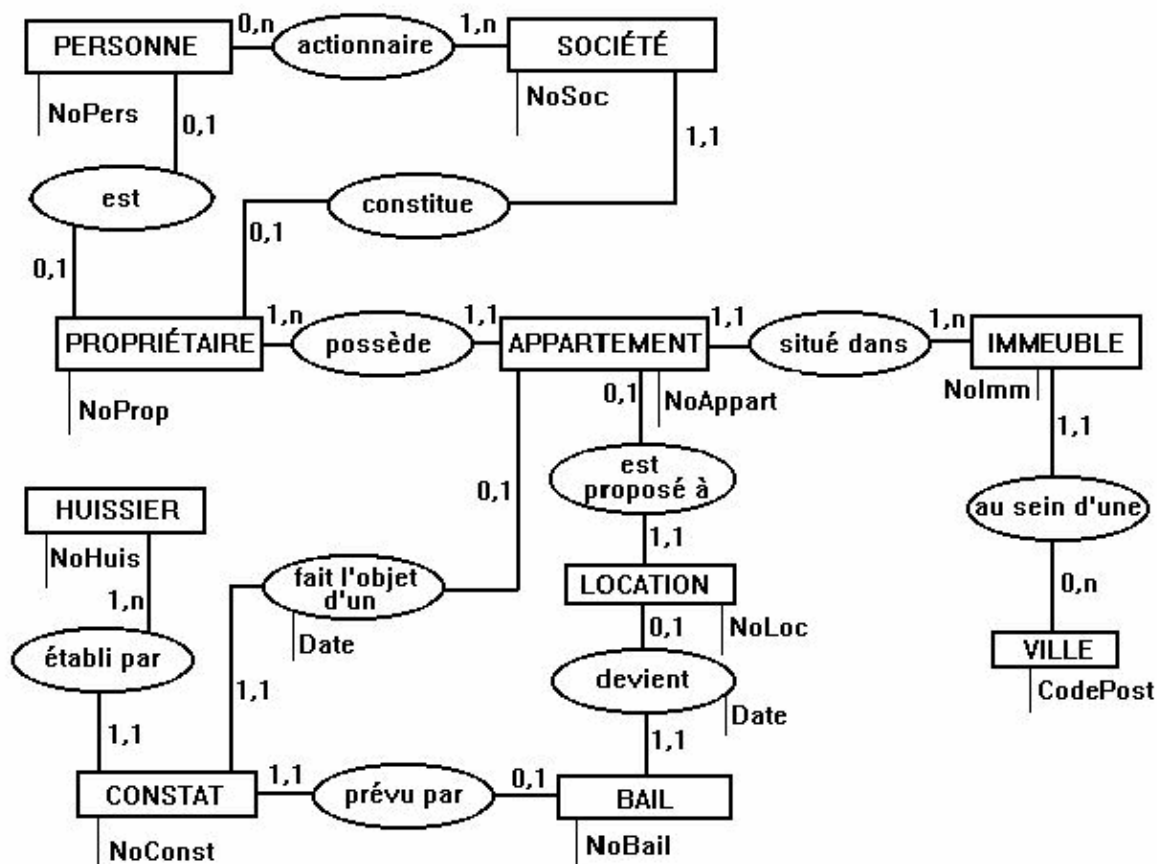


Lors de la réalisation de la base de données, les entités RÉALISATEUR et ACTEUR peuvent être regroupées en une seule table car il y a un nombre non négligeable de réalisateurs qui sont acteurs, et vice-versa. Dans ce cas, un champ d'un seul caractère permettra de faire la différence entre un réalisateur pur, un acteur pur et un acteur réalisateur.

Notez aussi les cardinalités entre les entités ACTEUR et FILM, en effet, un film d'animation ne possède aucun acteur.

Dans l'entité EXEMPLAIRE figure un attribut "dispo" permettant de savoir si l'exemplaire n° X d'un film est disponible ou en cours d'emprunt.

7.2. Location d'appartements pour une agence immobilière



NOTA : pour simplifier l'écriture, ne figurent dans ce schéma que les attributs clefs ou les attributs d'associations.

8. Divers

8.1. Bibliographie

- La méthode MERISE, principes et outils (Tome 1 & 2) - TARDIEU, ROCHFELD, COLLETTI - Les éditions d'organisation 1986
- Conception de bases de données : du schéma conceptuel au schéma physique - GALACSI - Dunod 1989
- Modélisation dans la conception des systèmes d'information - ACSIOME - Masson 1990
- Apprendre et pratiquer MERISE - J. GABAY - Masson 1993
- Maîtriser les bases de données – Georges GARDARIN – Eyrolles 1993
- Concepts fondamentaux de l'informatique – Alfred AHO, Jeffrey ULLMAN – Dunod 1993
- Bases de Données et Modèles de Calcul - Outils et Méthodes pour l'Utilisateur - Jean-Luc HAINAUT - InterEditions 1994
- MERISE, vers une modélisation orientée objet – José MOREJON – Les Editions d'Organisation 1994
- AMC*Designer, mise en œuvre de MERISE – Gilles GUEDJ – Eyrolles 1996
- Introduction aux bases de données, 6e édition– Chris J. DATE – Thomson International Publishing 1998
- De UML à SQL - Christian SOUTOU - Eyrolles 2002

8.2. Outils de modélisation

<i>DataBase Design Studio de Chili Source Software</i>	http://www.chillisource.com/
<i>DeZign for DataBases de Datanamic</i>	http://www.datanamic.com/
<i>ERWin de Computer Associates</i>	http://www3.ca.com/Solutions/Product.asp?ID=260
<i>Mega Suite de Mega International</i>	http://www.mega.com/
<i>PowerDesigner de PowerSoft (ex AMC*Designer)</i>	http://www.sybase.com/products/enterprisemodeling/powe
<i>RoboCase de DB Logic Inc</i>	http://www.dblogic.com/
<i>Visible Analyst de Visible Systems Corporation</i>	http://www.visible.com/Products/Analyst/overview.html
<i>Win'Design de Cecima</i>	http://www.win-design.com/fr/index.htm
<i>xCase de Resolution</i>	http://www.xcase.co.il/
<i>DataArchitect de theKOMPANYcom</i>	http://www.thekompany.com/products/dataarchitect/
<i>Rational Rose de Rational Software</i>	http://www.rational.com/
<i>Studio 2 de CaseStudio</i>	http://www.casestudio.com
<i>ER Studio de Embarcadero</i>	http://www.embarcadero.com/
<i>Visio de Microsoft (ex infomodeler de Synactics)</i>	http://www.microsoft.com/office/visio/
<i>Silverrun de Magna solutions</i>	http://www.silverrun.com/
<i>System Architect de Popkin</i>	http://www.popkin.com/products/sa2001/systemarchitect.h
<i>Designer 2000 d'Oracle (mono base)</i>	http://www.oracle.com/ (???)
<i>Casewise</i>	http://www.casewise.com/
<i>QuickUML de Excel (Linux et Windows) de Excel Software</i>	http://excelsoftware.com/quickumllinuxnews100.html
<i>MacA&D / WinA&D de Excel Software</i>	http://excelsoftware.com/richdatamodel.html
<i>Select d'Aonix</i>	http://www.aonix.com/