

## PRELIMINAIRES

L'objectif est d'élaborer un modèle pour structurer les données d'un système d'informations. On cherche donc à trouver un modèle représentatif des données que nous fournit un univers d'information, pour lequel on mettra en place ultérieurement des traitements automatiques.

Supposons que nous voulions formaliser les données qui concerne la gestion de véhicules et le suivi de leurs propriétaires successifs. On peut considérer, dans cet exemple, que l'on dispose à priori des fiches:

- **VEHICULE** contenant : num de série, marque, type, couleur
- **PERSONNE** contenant : num de Sécu, nom, prénom, email

Dans notre exemple, pour lier un véhicule à une personne il est nécessaire de créer une fiche (appelée entité relationnelle, fiche d'association ou encore table « pivot ») permettant de connaître qui est (ou a été) propriétaire de tel ou tel véhicule. Cette fiche d'association pourrait être :

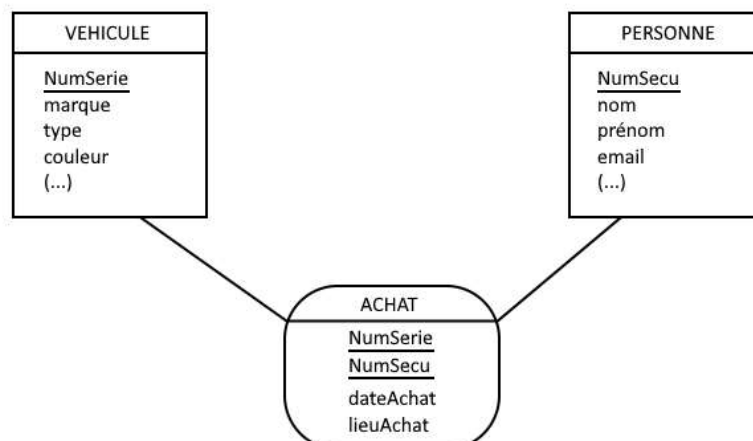
- **ACHAT** avec : n° de série (véhicule), n° de Sécu (personne), date d'achat, lieu d'achat

*NB : La conjonction de ces 2 numéros fournit un identifiant composé de deux rubriques, qui assure l'unicité de chaque fiche propriété. En effet un véhicule est la propriété d'une seule personne pour chaque association.*

### Comment modéliser la persistance des données pour cet exemple très simple ?

Plusieurs modes de représentation sont possibles (tableaux extensifs ou schémas arborescents) mais ils présentent le désavantage de lister exhaustivement tous les identifiants et de décrire toutes les relations possibles : En cas de volume trop important de données, leurs élaborations seraient très lourdes à effectuer. Par ailleurs, ils ne décrivent la relation que dans un seul sens (de véhicule vers propriétaire ou de propriétaire vers véhicule). Il faudrait alors élaborer les schémas réciproques pour visualiser les propriétés d'un véhicule pour chacune des personnes...

On préférera donc le **schéma conceptuel** suivant qui est suffisant pour représenter la structure des données **indépendamment** des contenus (càd des valeurs que peuvent prendre les identifiants et les rubriques associées).



Ce modèle permet de repérer sans ambiguïté :

- les objets de type entité (VEHICULE, PERSONNE)
- les objets de type association (ACHAT)
- les identifiants (NumSerie, NumSecu) qui sont toujours soulignés (c'est une règle d'écriture dans ce formalisme),
- les rubriques de renseignements (marque, type, couleur, nom ...),
- les relations entre les entités: par des traits,
- les rubriques de liaison (rubriques soulignées dans l'association ACHAT).

**On se rappellera que l'identifiant de l'association est constitué par la concaténation des identifiants des entités participantes.**

Il nous suffira juste de rajouter, ici, des **cardinalités** à cette représentation pour se rapprocher plus fidèlement d'un véritable modèle conceptuel aux normes Merise. Cette représentation graphique permet de spécifier de façon précise et visuellement simple la structure des données du système étudié. Il est utilisable par un analyste et peut être commenté, critiqué, modifié avec les utilisateurs.

## Exercice 0

1. Etant donné des fiches:

- FILM renseignés avec titre, date de sortie, durée
- ACTEUR renseignés avec nom, prénom, nationalité, âge

Concevoir un schéma conceptuel représentant la structure des données, après avoir créé l'entité relationnelle ROLE, qui permet de lier les acteurs aux films dans lesquels ils ont joué (un acteur ne joue ici qu'un seul rôle dans un film).

2. On y ajoute les fiches

- PRODUCTEUR renseignées avec nom, raison sociale

Enrichir votre schéma conceptuel, sachant qu'un producteur finance plusieurs films et qu'un film peut être financé par plusieurs producteurs. Pour chaque film on connaît le montant du financement de chaque producteur.

## I. PRINCIPES GENERAUX DU MODELE CONCEPTUEL DE DONNEES

Le système d'information est connu à travers ce qu'il est convenu d'appeler **l'univers du discours** appelé aussi **base de connaissances**. En fait il s'agit des informations recueillies sur le système et par exemple consignées dans le cahier des charges suite à l'interview des acteurs du système (précisément rédigée) et/ou une collection de documents actuels (ou futurs).

La démarche, stratégique pour la conception de votre application, peut se décomposer en **cinq étapes indispensables**:

- 1) Etablissement d'un **dictionnaire des données**
- 2) Repérage des **entités**
- 3) Description des **relations** entre les entités
- 4) Définition des **cardinalités** entre entités et relations
- 5) **Formalisation définitive** du modèle (et gestion des cas particuliers)

### 1. Etablir le dictionnaire des données

Il s'agit de relever toutes les **données utiles à la gestion** quotidienne **du système** ou du processus étudié : dans notre cas, on dispose d'un rédactionnel (le cahier des charges fonctionnel ou un rapport d'audit SI, par exemple). Une technique consiste à souligner chaque mot du vocabulaire utilisé dans la gestion des commandes et à en dresser la liste.

Lorsqu'on dispose d'une maquette d'un document (imprimé, fiche de liaison interservices), on peut également dresser la liste des rubriques à partir de celui-ci (on obtient généralement le même résultat), mais on dispose le plus souvent des deux. C'est ainsi que le rédactionnel complète les documents (et vice versa) en termes de compréhension du système ; cela permet même de découvrir les éventuelles incohérences d'un processus...

Exemple :

*« L'entreprise reçoit ses commandes clients par téléphone, par email ou par courrier. Dans tous les cas il est précisé quel est le client payeur (en général le siège de la société) et le client destinataire de la livraison (lorsque la société dispose de plusieurs dépôts) avec pour chacun son numéro, son nom, son email et son adresse. Pour chaque produit commandé on connaît le code et le nom de la famille à laquelle il appartient ainsi que son numéro, son libellé et la quantité commandée. Le réceptionnaire de la commande notifie la date d'arrivée de la commande et lui affecte un numéro d'ordre (incrémenté de 1 à chaque fois) ».*

Ce qui nous amène à établir la liste suivante :

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

## 2. Repérer les entités « objets »

Il s'agit d'effectuer un regroupement des mots qui ont traits, de façon invariante, à une même entité. Ces entités seront appelées des **objets** et n'existeront que parce qu'ils correspondent à une collection d'informations homogènes gérées au sein de notre système d'information.

Dans notre exemple, on constate 5 objets distincts :

- CLIENT DESTINATAIRE (1,2,3,4)
- CLIENT PAYEUR (5,6,7,8)
- FAMILLE (DE PRODUIT) (9,10)
- PRODUIT (11,12)
- COMMANDE (14,15)

Les données de la 1<sup>ère</sup> étape, qui vont « enrichir » ces entités objets, sont appelées des **propriétés**. On remarquera que la n°13 (la Quantité commandée) n'est liée à aucun objet en particulier ; nous verrons ce que nous en ferons un peu plus tard...

### 3. Décrire les relations entre les entités

On va maintenant décrire les rapports qui existent entre les objets repérés, et on utilise pour cela un verbe simple. Dans notre exemple, on en déduit la **sémantique** suivante :

A - client destinataire **dépend** de client payeur,

B - client destinataire **pass** des commandes,

C - produit **appartient** à famille.

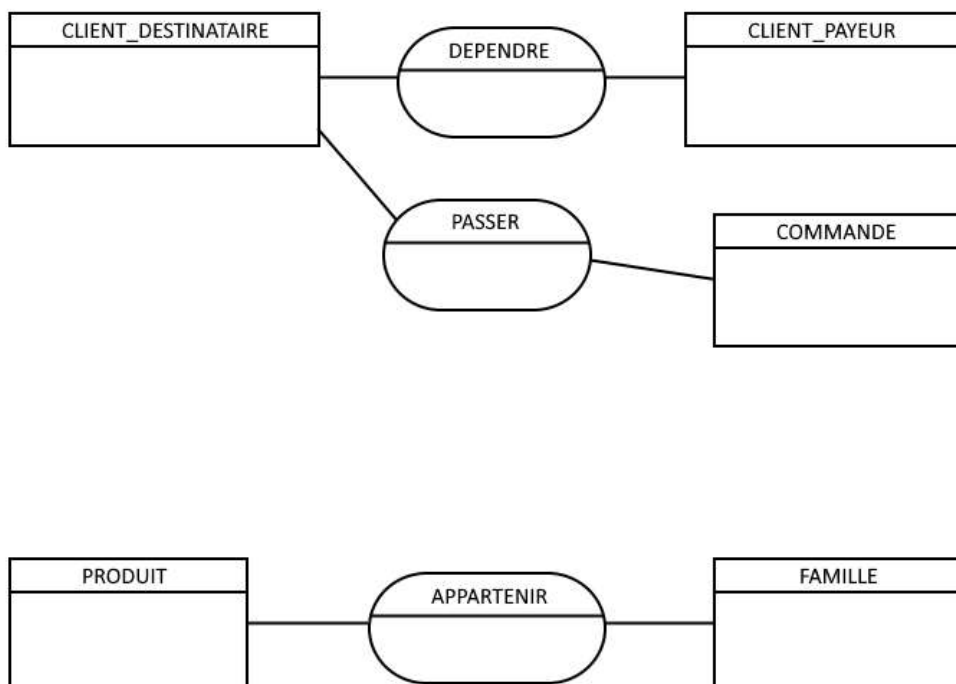
La relation A sera nommée DEPENDRE

La relation B sera nommée PASSER

La relation C sera nommée APPARTENIR

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

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



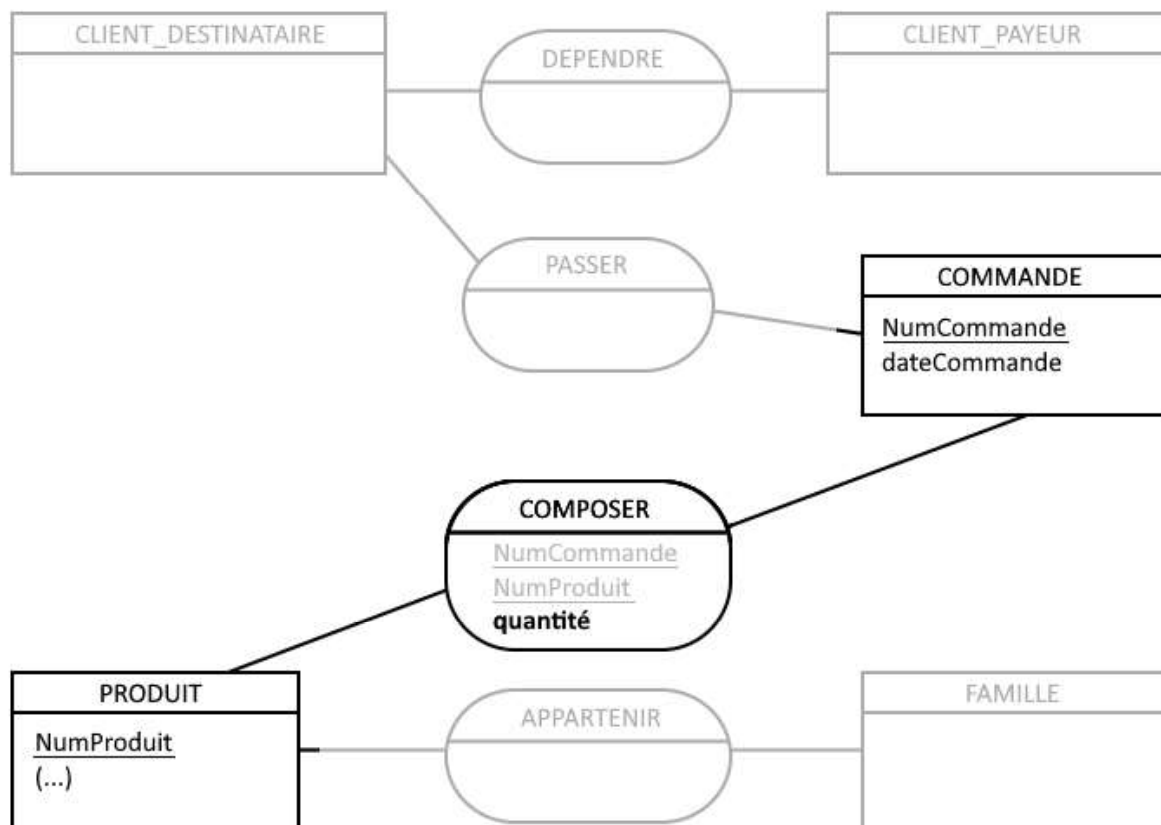
Dans notre exemple, la quantité commandée (rubrique 13) dépend **à la fois** de la commande **et** du produit. En effet une quantité commandée doit correspondre à un produit spécifique pour une commande spécifique. Autrement dit, cette donnée n'aura du sens que par l'existence d'une relation entre commande et produit.

Par ailleurs on pourra remarquer que la version du schéma conceptuel élaboré en étape 3 présente deux "sous ensembles". Cette disjonction indique également que le système de données n'est pas complètement formalisé.

On nommera donc COMPOSER l'association entre les entités PRODUIT et COMMANDE en référence à la sémantique suivante : "une commande est COMPOSEE de produits".

La donnée n°13 « quantité commandée » est une **propriété de l'association COMPOSER**. On dira que cette **association** est **porteuse de données**, par opposition aux associations précédentes (DEPENDRE, PASSER, APPARTENIR) qui étaient non porteuses de données (vides).

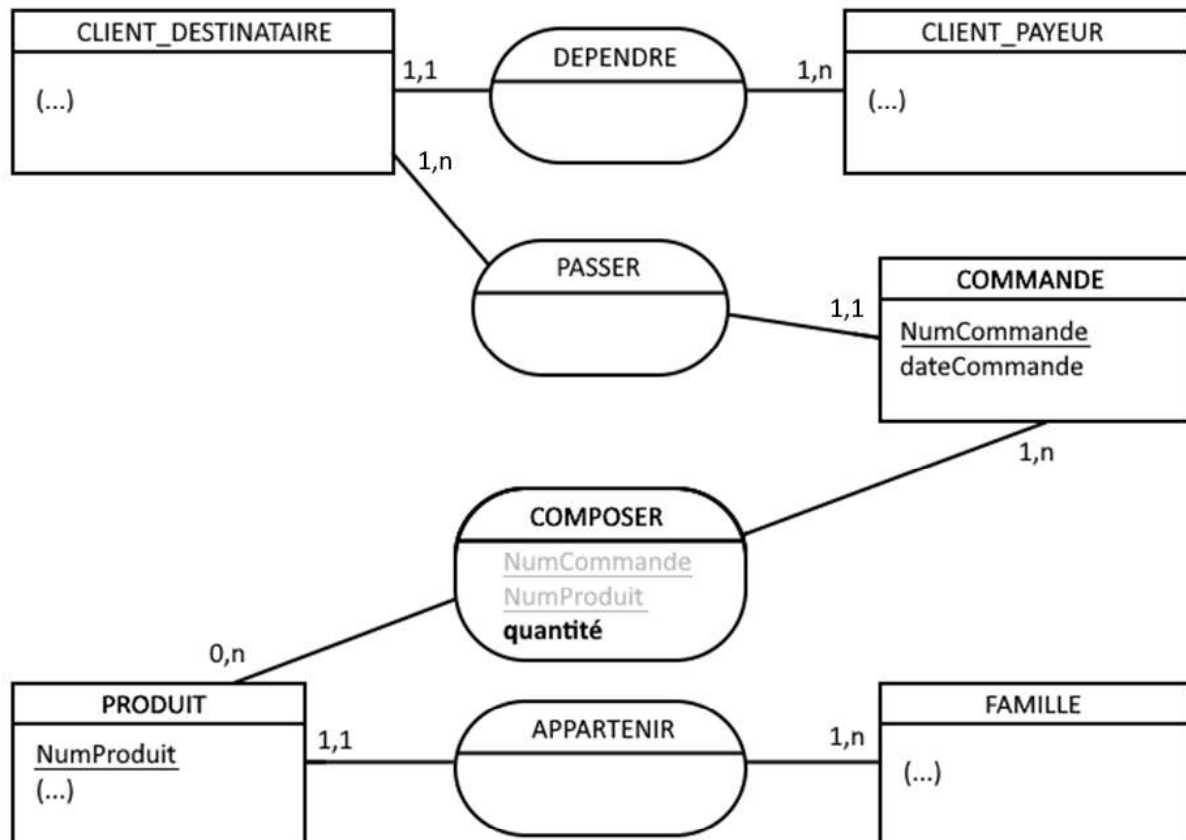
On formalisera donc cette sémantique selon le schéma suivant, où l'on a inclu les propriétés des entités aussi bien que des associations, ainsi que 2 propriétés supplémentaires (numéro commande et numéro produit) pour l'association COMPOSER



#### 4. Définir les cardinalités de parts et d'autres d'une relation

Les **cardinalités** (minimales et maximales, séparées par une virgule) portent sur les liens entre entités et relations. On se pose la question de savoir si on peut représenter des règles de gestion qui expriment le fait que ces liens peuvent :

EXISTER (une ou plusieurs fois) **ou** ne PAS exister (0 fois)

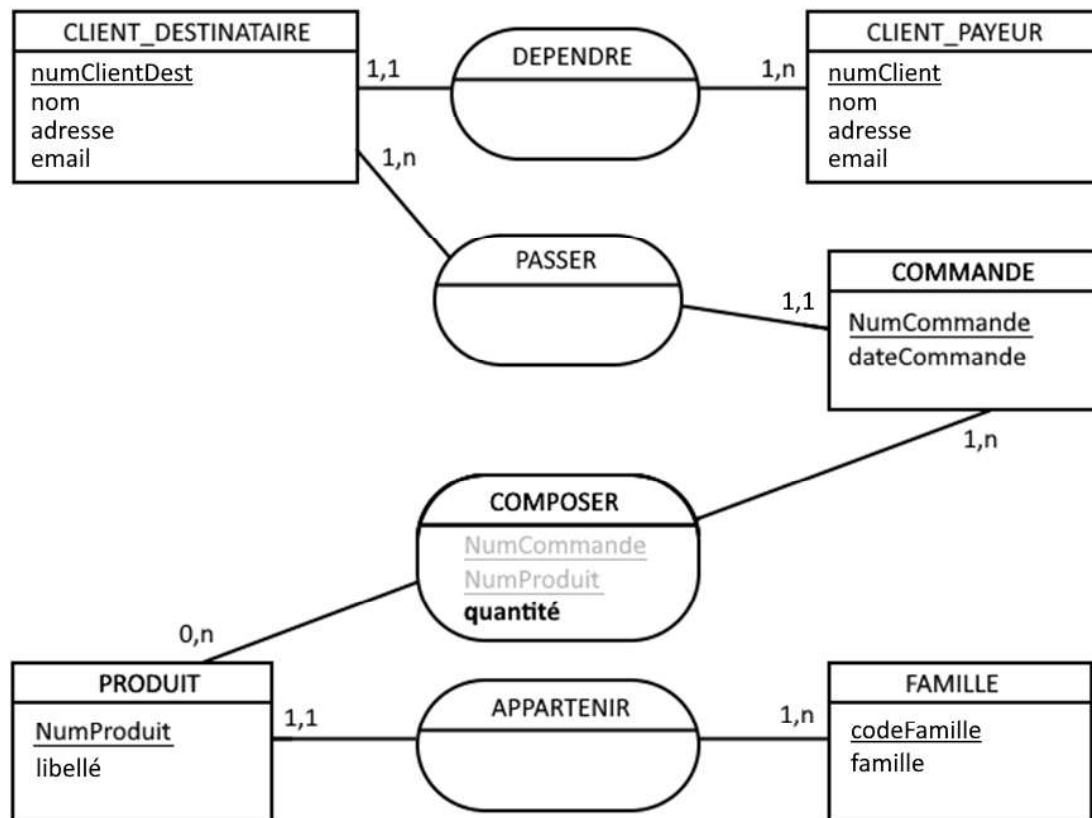


#### 5. Formalisation définitive du modèle

On peut enfin établir le modèle complet en respectant les règles d'écriture suivantes :

- 1) Les entités sont représentées par des rectangles
- 2) Les associations sont représentées par des "ronds"
- 3) Les propriétés de chaque objet sont indiquées (il s'agit des données qui lui sont propres)
- 4) Les identifiants des entités sont soulignés
- 5) Les identifiants des associations ne sont pas indiqués (d'où le grisé dans « COMPOSER »)
- 6) Les cardinalités sont portées sur les liens entre entités et associations

Dans notre exemple commenté, le modèle définitif serait donc :



### Exercice de Synthèse 1

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

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

NB : chaque produit est proposé par différents fournisseurs à un prix qu'ils ont établi et communiqué.

**Vous êtes chargé d'établir le MCD avec ses cardinalités en suivant les étapes 1-2-3-4-5**



## Exercice de Synthèse 2

Soit la base de connaissances suivante :

"Notre institut privé est un établissement qui offre un certain nombre de stages auxquels peuvent s'inscrire des stagiaires. Chaque **stage** est identifié par un code, est caractérisé par un **libellé** (remise à niveau scientifique, perfectionnement en anglais, espagnol débutant,...) et une **date de début**. A chaque **stagiaire** est affecté un **numéro d'identification**. En outre, on enregistre son nom, son adresse et la date d'inscription au stage."

"On sait qu'un stagiaire peut s'inscrire à plusieurs stages différents mais ne peut s'inscrire qu'une fois à un stage donné. Chaque **stage** est composé de **matières** connues par leur **code** et leur **libellé** (grammaire espagnole, mathématiques, mécanique,...) Une matière peut être suivie dans le cadre de plusieurs stages et elle est enseignée par **un seul professeur**. Ce professeur est connu par son matricule et son nom et prénom."

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

## II. CAS PARTICULIERS DU MODELE CONCEPTUEL DE DONNEES

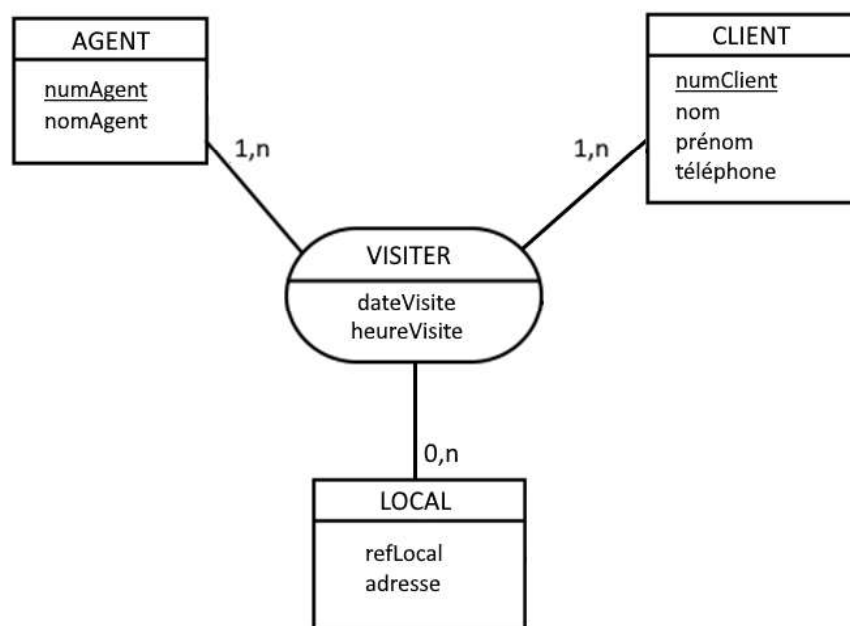
Les cas particuliers qui vont suivre sont fréquemment rencontrés d'un projet de conception à un autre : maintenant que vous maîtrisez les bases de la réalisation d'un MCD, il est important pour vous de parfaitement les appréhender ; )

### 1. Relations n-aires et C.I.F.

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

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

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

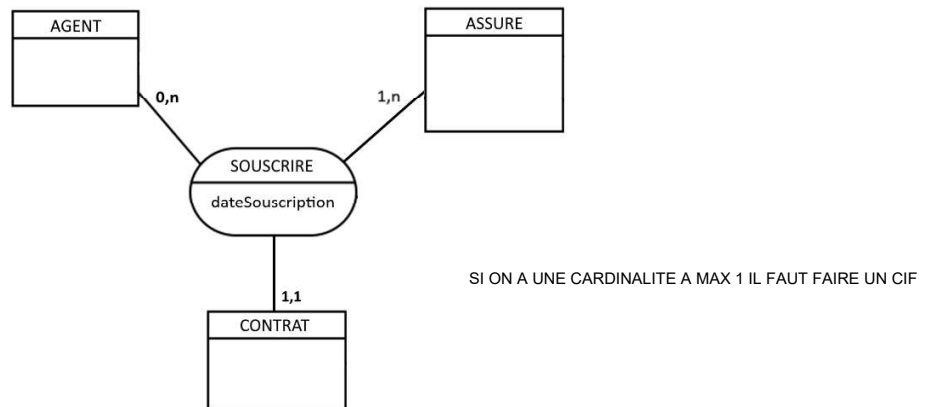


Les cardinalités s'évaluent toujours par le dénombrement d'occurrences minimum et maximum possibles de la relation vue depuis une occurrence d'une entité. Ainsi, un agent donné organise de une à n visites, soit (1,n) ; un client donné effectue de une à n visites, soit (1,n) ; enfin un local est sujet à 0 ou n visites, soit (0,n).

## Contraintes d'Intégrité Fonctionnelles

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

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

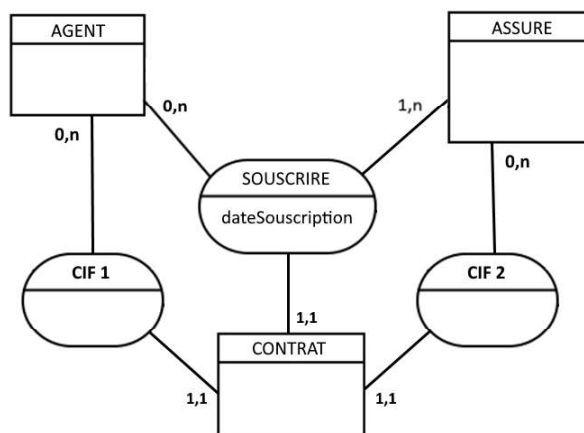


De cet exemple on peut déduire que de la connaissance d'un contrat, on a obligatoirement connaissance du seul agent qui en est responsable (cardinalité 1,1). La réciproque n'est pas vraie car si je connais un agent, je ne connais que les contrats dont il est responsable (cardinalité 0,n).

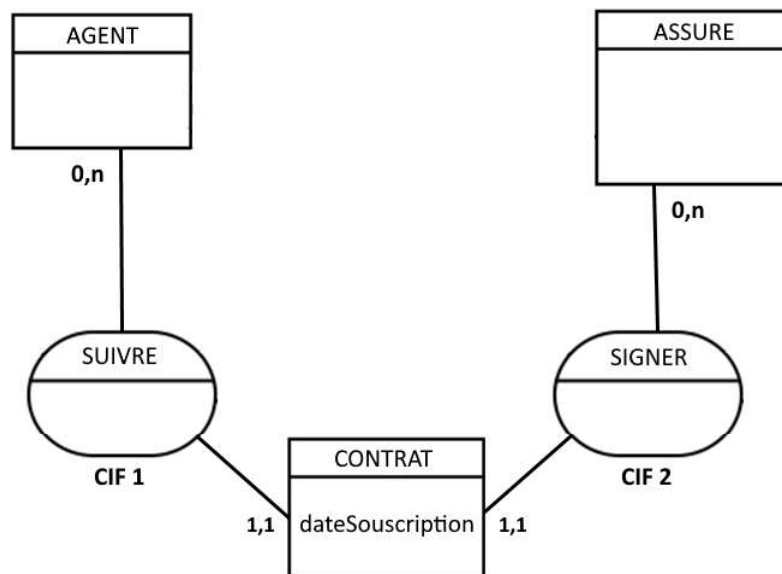
Ce schéma de données montre qu'il y a une **relation spécifique** entre contrat et agent : un contrat est mis en souscription par un agent et un seul, alors qu'il peut en faire souscrire de 0 à plusieurs. Ce type de relation (toujours de cardinalités x,1- x,n) est appelée **Contrainte d'Intégrité Fonctionnelle (CIF)**.

Suivant le même raisonnement, comme un contrat ne peut être souscrit que par un assuré et un seul (cardinalité 1,1), bien que un assuré peut souscrire de un à plusieurs contrats (cardinalité 1,n), on se trouve également en présence d'une CIF entre CONTRAT et ASSURE.

Le modèle précédent peut donc être complété de la manière suivante :



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



### Exercice de Synthèse 3

Le service ressources humaines d'une entreprise qui souhaite gérer ses actions de formation a établi le rédactionnel suivant :

"Afin d'adresser des convocations aux employés, on enregistre les agents avec leur numéro, leur nom, prénom ainsi que l'établissement auquel ils appartiennent (l'entreprise est implantée dans différents lieux géographiques). De cet établissement on connaît le code, le nom et l'adresse."

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

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

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

**A partir de ce rédactionnel :**

1. Etablir le dictionnaire des données
2. Faire le MCD correspondant

## 2. Relations réflexives

Jusqu'à maintenant nous avons établi des relations entre deux ou plusieurs objets différents. Pourtant il peut exister des relations sur un seul objet (c'est à dire de l'objet sur lui-même), c'est ce que nous appelons une **relation réflexive**.

On veut ici gérer un ensemble de personnel féminin (connu avec leur numéro, leur nom, leur âge) et on souhaite faire apparaître les relations parentales les concernant. On créera l'objet PERSONNE et une relation PARENTE selon la sémantique « une personne est parente d'une personne ». Cette relation réflexive est schématisée ainsi :



En ce qui concerne les cardinalités on voit que :

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

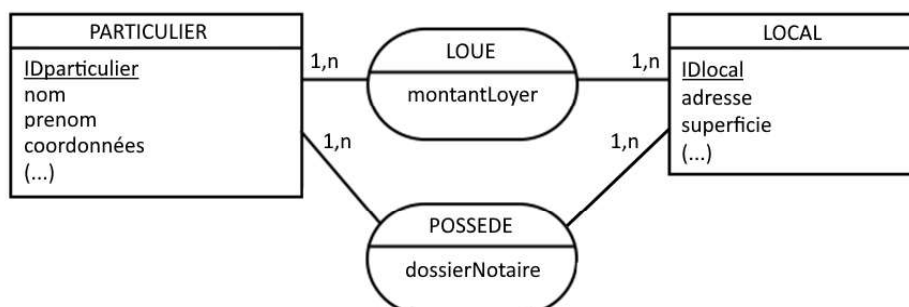
(\* : dans le cas où toutes les mères seraient répertoriées, la cardinalité serait alors 1,1)

Les relations réflexives sont utiles pour modéliser tout ce qui ressemble de près ou de loin à une **arborescence** (arbre généalogique, nomenclature de produit industriel, phylogénie...).

## 3. Relations à rôle

On peut avoir besoin d'exprimer plusieurs relations distinctes entre les **mêmes entités**. Chaque relation a une signification sémantique différente des autres : on parlera de **relation à rôle** (chacune ayant un rôle différent).

Dans le cadre de la gestion d'une agence immobilière, imaginons que nous ayons modélisé un système où il est question de gérer des personnes (particuliers), ainsi que les locaux qu'elles possèdent (propriétaires) ou qu'elles habitent (locataires). On obtient alors le schéma suivant :

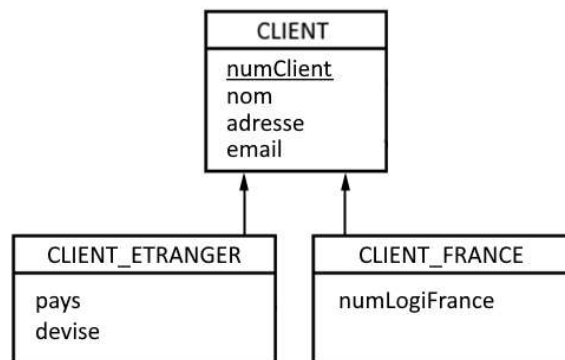


#### 4. Spécialisations et généralisations

C'est un concept qui vous sera particulièrement utile lorsque vous développerez votre future application en programmation objet, lorsque vous utiliserez le concept « d'héritage ».

Prenons le cas d'une société de logistique ayant des relations commerciales à l'exportation, où il peut être intéressant (voire indispensable) de faire apparaître la spécificité des clients étrangers et des clients français. Le concepteur peut ainsi distinguer deux sous-divisions du concept de client : en tant que clients, les "étrangers" et les "français" ont des caractéristiques **communes** (nom, adresse et email). Ils ont aussi des caractéristiques **spécifiques** : pays et devise pour les étrangers, numéro de licence « LogiFrance <sup>TM</sup> » pour les français.

La **spécialisation** permet de formaliser ce type de cas, mais uniquement sur des entités ; en aucun cas sur des relations ! Dans notre exemple, elle consiste tout d'abord à modéliser un individu CLIENT, décrit par les caractéristiques communes. Ensuite de considérer les deux individus comme deux spécialisations particulières de l'individu CLIENT : CLIENT\_ETRANGER et CLIENT\_FRANCE sont alors appelés « individus sous-classe » de l'individu CLIENT. La représentation graphique est la suivante :



L'introduction de sous-classes spécialisées permet de traduire des **réalités complémentaires** dont l'intérêt est :

- a) faire apparaître les propriétés communes d'une classe d'entité
- b) faire corolairement apparaître les propriétés spécifiques de chaque sous-classe
- c) de mettre en évidence des règles de gestion spécifiques pour chaque cas\*

(\* : Notamment avec les relations du reste de votre MCD)

**Si au moins 2 de ces intérêts n'ont pas à être mis en évidence dans la formalisation, il n'y a alors pas lieu de créer de spécialisation.**

## **Exercice de Synthèse 4**

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

Les données sont recueillies par ordre alphabétique :

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

**Travail à réaliser : construire le MCD**