

Chapitre 3. Le modèle Entité-Relation

3.1. INTRODUCTION.

Ce chapitre passe en revue les concepts principaux (section 3.2) et avancés (3.3.) introduits par le modèle Entité-Relation, avant d'étudier la construction d'un modèle conceptuel E/R (section 3.4.) Tous les concepts ne seront cependant pas développés ici ; on laissera notamment de côté, provisoirement, l'étude des dépendances fonctionnelles ; celles-ci sont spécifiques au modèle relationnel. La simplification (ou normalisation) du modèle, qui utilise les propriétés des dépendances fonctionnelles, sera donc seulement évoquée de façon pratique au cours de la section 3.4.

3.2. CONCEPTS FONDAMENTAUX : LES ELEMENTS DU MODELE.

3.2.1. Notion de type ou de classe.

Un type d'entités est un sous-ensemble regroupant des entités de même nature, c'est-à-dire présentant des caractéristiques communes.

Exemple : { DIOUF B., FIANYO E., GAYE E. }

Caractéristiques : amies, collègues, africaines, au premier rang de la classe, ne portent pas de lunettes.

Au regard de certaines de ces caractéristiques, elles sont donc semblables.

3.2.2. Entité (ou objet, ou individu).

{ DIOUF, Boury, rang 1 } = entité au regard d'un objectif :

"nommer les gens et leur place".

Une entité apparaît comme un paquet de valeurs sur des caractéristiques significatives par rapport à un objectif. On constate que l'ensemble est une "chose" qui existe et que l'on peut distinguer d'une autre.

Exemples : { DAKAR, 14-11-92, 25 degrés, 23 degrés, Dégage } est une entité descriptive de la météorologie ;

{ TF1, 14-11-92, 14H25, 105 mn, "Les enquêtes du Commissaire Maigret", Série } est une entité descriptive de programmes de télévision.

3.2.3. Entité-type.

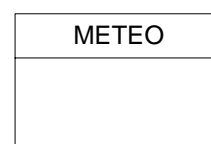
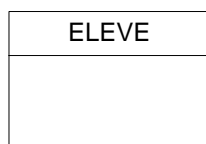
La description complète des entités n'est pas pratique à l'étape de la modélisation. Elle constituerait d'ailleurs le fonctionnement lui-même du système d'information. On cherchera donc à utiliser la notion de type en opérant les regroupements d'entités par classe.

Exemple de classes ou Entités-type : ELEVE, METEO, PROGRAMME-TV.

ELEVE est la classe des entités suivantes : { BADIANE, DIEYE, DIOUF, FAYE, FIANYO, SAYE, KANDE, MBOW, NDOUR, SY, THIAW }

Une entité-type est une classe d'entités particulières ayant des caractéristiques communes au regard d'un objectif d'utilisation.

On représente les entités-type par un rectangle en plaçant le nom attribué à l'entité-type dans la partie haute.



Rectangles nommés représentant des objets.

3.2.4. Propriétés d'une entité.

Les valeurs énoncées dans les exemples précédents constituent les propriétés des entités.

Exemples : "GAYE" est une propriété,

"Aminata" en est une autre,

"14-11-92" en est une troisième.

On notera qu'une propriété n'a pas de sens en soi, mais seulement rattachée à une entité.

Les valeurs caractéristiques sont elles-mêmes puisées dans des ensembles de valeurs.

Exemple : "Aminata" provient de l'ensemble des valeurs suivantes : {Abdou, Albert, Alfred, Aliou, Aminata, Amédée, Arthur}.

3.2.5. Propriété-type.

Le même problème de l'énumération intégrale des valeurs des propriétés des entités se pose au modélisateur. On regroupe alors l'ensemble des valeurs possibles d'une propriété pour en constituer une classe. En attribuant un nom à cette classe, on aura défini une propriété-type.

Exemple : "Aminata", appartenant à l'ensemble précédent, fait partie d'un ensemble plus grand, celui de tous les prénoms. prénom est une propriété-type caractéristique de l'entité-type ELEVE.

De même, Date du jour, Température maximum, Titre de l'émission sont des propriétés-type appartenant à des entités-type.

On rattache donc "naturellement" les propriétés-type aux entités-type.

Une propriété-type est une classe générique de propriétés distinguant une entité-type d'une autre.

Exemple :

Les propriétés-type {Nom, Prénom, Place} distinguent ELEVE de METEO caractérisée par {Ville, Date, Température max., Température min., Ciel}. Certaines entités-type peuvent cependant posséder des propriétés-type communes : la propriété-type "Ville" peut très bien appartenir aussi à l'entité-type ELEVE (dans un autre modèle) s'il est nécessaire de connaître des éléments d'adresse ou de lieu de naissance (toujours au regard d'un objectif de "gestion").

Représentation graphique des propriétés-type :

ELEVE
Nom
Prénom
Place

METEO
Ville
Date
Temperature_Max
Temperature_Min
Ciel

Représentation graphique classique des propriétés dans les objets.

Une autre représentation existe :

ELEVE
Nom
Prénom
Place

METEO
Ville
Date
Température_Max
Température_Min
Ciel

Autre représentation d'objets.

3.2.6. Identifiant et identifiant-type.

Nom dans l'objet-type ELEVE semble pouvoir identifier chaque élève de la classe Elèves-Ingénieurs 1ère année. L'identifiant est ainsi une propriété-type choisie parmi les propriétés-type de l'objet-type.

Définition :

L'identifiant-type est une propriété-type qui détermine de manière univoque chaque exemplaire d'une population (même si la population change par mise à jour).

Attention : le Nom n'est pas, en général, un identifiant-type efficace à cause des homonymies possibles (la classe comporte deux élèves dont le nom est DIOUF).

On représente graphiquement l'identifiant-type en soulignant la propriété-type choisie.

3.2.7. Occurrences.

L'occurrence d'une entité-type est l'élément individualisé appartenant à ce type. La somme des occurrences d'une entité-type représente donc le cardinal de la population.

Exemple :

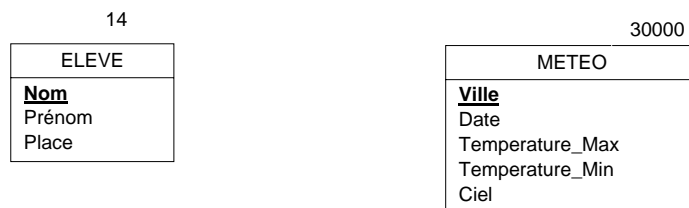
nombre d'occurrences de ELEVE dans la classe Elèves Ingénieurs 1ère Année : 14.

Le cardinal |ELEVE| = 14.

nombre d'occurrences de METEO dans la base météorologique : 30 000.

Le cardinal |METEO| = 30 000.

Représentation graphique :



Représentation du cardinal et de l'identifiant-type.

3.2.8. Association entre des entités. Association-type ou relation-type.

Exemple : Soit les objets-type FOURNISSEUR et PRODUIT. Une association entre ces deux objets-type est le constat du regroupement existant entre deux entités.

"Peugeot vend 309GR" : Ce regroupement existe bien sûr au regard d'objectifs déterminés, pour un système d'information donné.

Définition 1 :

Une association-type a pour objet de décrire une réalité perçue dans l'organisation.

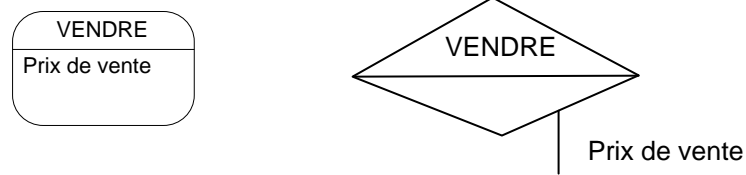
Contre-exemple : "Peugeot vend XM" n'est pas une association, car le rapprochement n'existe pas (ou n'est pas vérifié).

Définition 2 :

Une association-type est un sous-ensemble du produit cartésien des entités-type qu'elle relie.

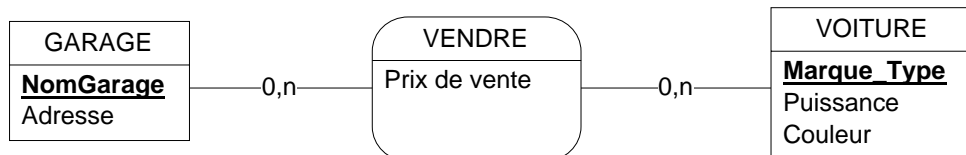
$Vendre = \{x,y \text{ de } FOURNISSEUR, PRODUIT \mid (x,y) = \text{VRAI} \}$

Représentation graphique :



Deux façons de représenter une relation-type.

Puisque la relation-type associe des objets-type, un ensemble de traits relie chaque objets-type à la relation. On appelle patte chaque trait (il y a autant de pattes que d'objets associés).

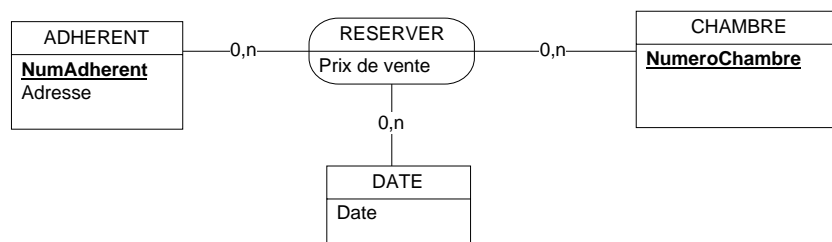


Modèle conceptuel de données à deux objets-type et une relation-type.

3.2.9. Occurrences d'une relation-type.

Tous les objets d'une population ne sont pas obligatoirement associés dans toutes les relations-type descriptives du réel.

Exemple : Soit une mutuelle comptant 1 000 000 d'adhérents et mettant à leur disposition 640 chambres dans des centres de soins, de cure et de retraite. Un adhérent peut ne jamais réserver de chambre, il peut aussi en réserver plusieurs. Tous les adhérents ne sont donc pas concernés par la relation "réserver".



Combien d'occurrences de la relation-type ?

Définition :

Une relation-type est totale si chaque occurrence des objets-type associés participe à la relation-type.

Elle est partielle si certaines occurrences des objets-type peuvent ne pas y participer.

3.2.10. Cardinalités.

La cardinalité d'un objet-type dans une relation-type est le couple du nombre minimum et du nombre maximum d'occurrences de la relation-type pouvant exister pour une occurrence de l'objet-type.

La cardinalité est aussi appelée Cardinalité minimum et Cardinalité maximum.

La cardinalité minimum est représentée par un chiffre : 0 ou 1.

La cardinalité maximum est représentée par le chiffre 1 ou la lettre n.

3.2.11. Dimension d'une relation-type.

Définition 1 :

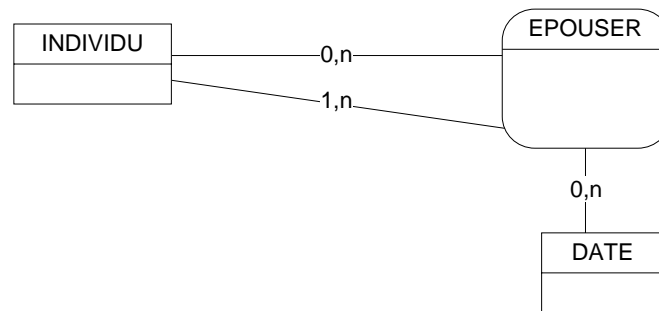
C'est le nombre d'objets-type (pas forcément distincts) intervenant dans la relation-type.

Exemple 1 : La relation-type "RESERVER" relie les objets-type ADHERENT, CHAMBRE et DATE ; elle est de dimension 3.

Exemple 2 : La relation-type "EPOUSER " relie les objets-type INDIVIDU et DATE ; elle est également de dimension 3.

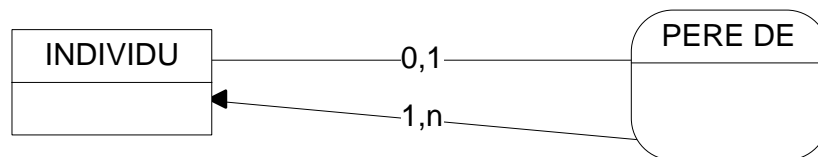
Définition 2 :

La dimension d'une relation-type est le nombre de pattes que comporte cette relation-type.



Relation-type de dimension 3 avec 2 objets-type.

Ainsi, une relation binaire peut-elle ne faire intervenir qu'un seul objet-type :



Relation-type binaire avec 1 seul objet-type.

Cependant, pour une occurrence de la relation-type, ce sont bien deux occurrences de l'objet-type qui interviendront. D'où la définition suivante :

Définition 3 :

La dimension d'une relation-type est le nombre d'occurrences d'entités-type concernées par une occurrence de la relation-type.

3.2.12. Identifiant-type d'une relation-type.

Il n'y a pas d'identifiant-type en soi d'une relation-type, car la relation-type n'a pas d'existence propre. Cependant, on identifie une relation-type en concaténant les identifiants-type de chaque objet-type participant à celle-ci.

Définition :

L'identifiant-type d'une relation-type est la concaténation des identifiants-type des objets-type associés.

3.3. AUTRES CONCEPTS.

On aborde ici la généralisation et la spécialisation, qui sont deux concepts symétriques. On doit les comprendre d'abord d'un point de vue dynamique comme des actions : l'action qui consiste à généraliser, l'action qui consiste à spécialiser. On les appréhendera ensuite comme des résultats, indépendamment du processus y menant : la généralisation mène à un objet générique à partir d'objets d'origine. La spécialisation mène à des objets spécialisés à partir d'objet généraux.

3.3.1. Généralisation.

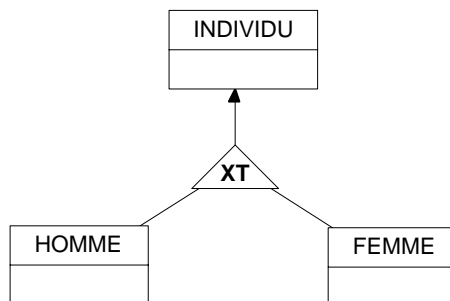
La généralisation consiste à créer un objet-type à partir de deux objets-type ou plus qui possèdent des propriétés communes.

Exemple :

HOMME est-un INDIVIDU ;

FEMME est-un INDIVIDU ;

INDIVIDU constitue un type nouveau dont les deux éléments sont des objets-type HOMME et FEMME.



Généralisation d'objets-type.

Remarques :

La relation entre l'objet origine et l'objet généralisé (l'objet générique) est de type "EST-UN" (en notation anglo-saxonne "IS-A"). Il s'agit donc d'un lien hiérarchique ou encore d'une inclusion (on parle de réseau sémantique). Cette inclusion peut disposer de propriétés particulières :

- On a une disjonction entre CAMION et VOITURE (un objet est l'un ou l'autre, mais pas les deux).
- On peut avoir une couverture $PERSONNE = ETUDIANT \cup ENSEIGNANT$ (un étudiant pouvant aussi être un enseignant)

Comme on part d'objets-type pour réaliser une généralisation, ces objets-type disposent d'identifiants-type. L'objet-type généralisé doit aussi disposer d'un identifiant-type. La question concernera donc à déterminer s'il y a une différence réelle entre les identifiants-type des objets de départ (voir l'héritage et la spécialisation).

3.3.1.1. Héritage des propriétés.

L'objet-type générique transmet toutes ses propriétés aux objets-type d'origine. Les objets-type "héritent" donc des propriétés de l'objet générique. L'objectif de l'héritage est de réaliser la non-redondance dans la description des entités.

3.3.1.2. Propriétés propres des objets origine.

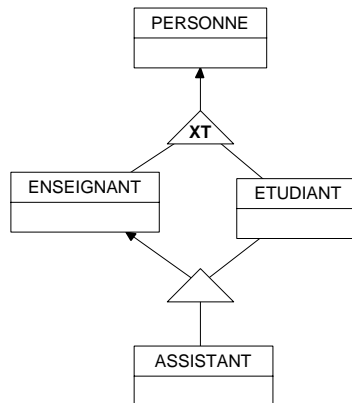
Ces propriétés sont celles dont l'objet générique ne peut hériter.

Exemple : ETUDIANT possède la propriété Numéro-d-étudiant ; tous les INDIVIDUS ne sont pas ETUDIANT. INDIVIDU ne peut donc pas hériter de la propriété Numéro-d-étudiant.

3.3.1.3. Héritage multiple.

Exemple :

ASSISTANT *est-un* ENSEIGNANT
 ASSISTANT *est-un* ETUDIANT
 ENSEIGNANT *est-un* PERSONNE
 ETUDIANT *est-un* PERSONNE



Héritage multiple.

Nombre d'heures de cours est une propriété héritée par ASSISTANT, en provenance de deux objets génériques. Or, pour une même occurrence d'ASSISTANT, ces propriétés ne prennent pas les mêmes valeurs. La solution consiste à différencier la propriété en conflit par le nom complet ou à changer son nom.

3.3.1.4. L'exclusion ou disjonction.

La généralisation/spécialisation est un processus de modélisation qui permet le cas échéant de classer les occurrences d'un objet-type en groupes exclusifs.

- On peut avoir une partition INDIVIDU = HOMME \cup FEMME qui réalise les conditions de la partition et de la couverture ($\text{HOMME} \cap \text{FEMME} = \emptyset$).

3.3.2. Spécialisation ou extraction d'entités-type.

Le dual ou le symétrique de la généralisation est une spécialisation ou extraction. Une entité-type est découpée en plusieurs sous-ensembles possédant des propriétés propres (ou des valeurs spéciales).

Exemple : Soit l'objet-type EMPLOYE. Si le système d'information a besoin de connaître les noms-de-jeune-fille de ses employés de sexe féminin, une extraction sera nécessaire, puisqu'une employée peut être célibataire (problème de normalisation de l'objet-type : toutes les propriétés d'un objet-type doivent posséder une valeur).

Remarque sur les propriétés de l'héritage :

Les sous-ensembles créés héritent des propriétés de l'objet générique (cf. 3.3.1.).

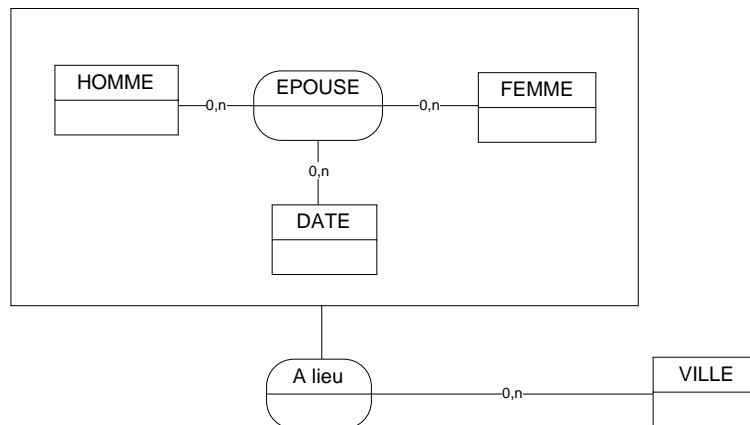
Exemple :

EMPLOYE-MARIEE hérite du Numéro-d-employé. (Attention : cela signifie que le numéro-d-employé n'a pas à figurer dans le sous-ensemble ; toute occurrence d'employée mariée récupère pourtant bien la valeur de numéro d'employé).

Les sous-ensembles extraits ne constituent (donc) pas toujours des objets-type. Si les sous-ensembles sont des objets-type, ils seront reliés à leur objet générique.

3.3.3. Personnalisation de relation-type / agrégation.

L'agrégation est le regroupement d'objets-type et de la relation-type existant entre eux par un "super-objet-type".



Exemple d'agrégation.

Les occurrences de ce "super-objet-type" sont les occurrences de la relation-type d'origine.

3.3.4. Remarques "Orientées-Objet".

On note un intérêt croissant pour l'approche dite "orientée-objet", du fait des insuffisances du modèle Entité-Relation (difficulté à prendre en compte les contraintes sur les objets-type). Il convient de distinguer deux types d'orientations objet :

1- Du point de vue de l'utilisateur.

L'évolution technologique des ordinateurs permet de plus en plus la prise en compte d'objets nouveaux, c'est-à-dire de types données particuliers : photographies, son etc. Il s'agit de données dont la taille unitaire est très grande (une photographie peut prendre 0,5 à 1Mo sur disque). On parle de BLOB (binary large object).

Par exemple, le SGBD ORACLE (à partir de la version 5) accepte des champs de taille maximale de 65536 octets, alors que INGRES dispose d'une capacité de champ à 2 Mo.

2- Du point de vue de l'informaticien.

L'orientation objet dans la conception des systèmes d'information consiste à utiliser les connaissances dégagées par la programmation orientée-objet ; il s'agit donc ni plus ni moins que de chercher à réunifier les données et les traitements, en d'autres termes revenir sur la séparation données/traitements préconisée par le rapport ANSI/SPARC de 1975.

Dans cette optique, l'objet est défini à la fois par sa structure de données et par les actions élémentaires que l'on peut exécuter sur lui, par les événements qu'il peut déclencher.

L'objet est, dans cette conception, équivalent à l'objet-type du modèle E/R : "collection d'éléments de données structurées identifiées par une référence unique" (GARDARIN).

La classe représente "un groupe d'objets ayant les mêmes propriétés, caractérisé par une collection d'opérations qui s'applique aux objets de la classe en cachant la structure [interne CC]." (GARDARIN)

Les approches orientées objet utilisent les concepts évoqués en 3.3.1. et 3.3.2. :

- la généralisation/spécialisation,
- l'agrégation,

avec 3 types de liens sémantiques principaux :

- est-un,
- est-un-composant-de,
- est-instance-de.

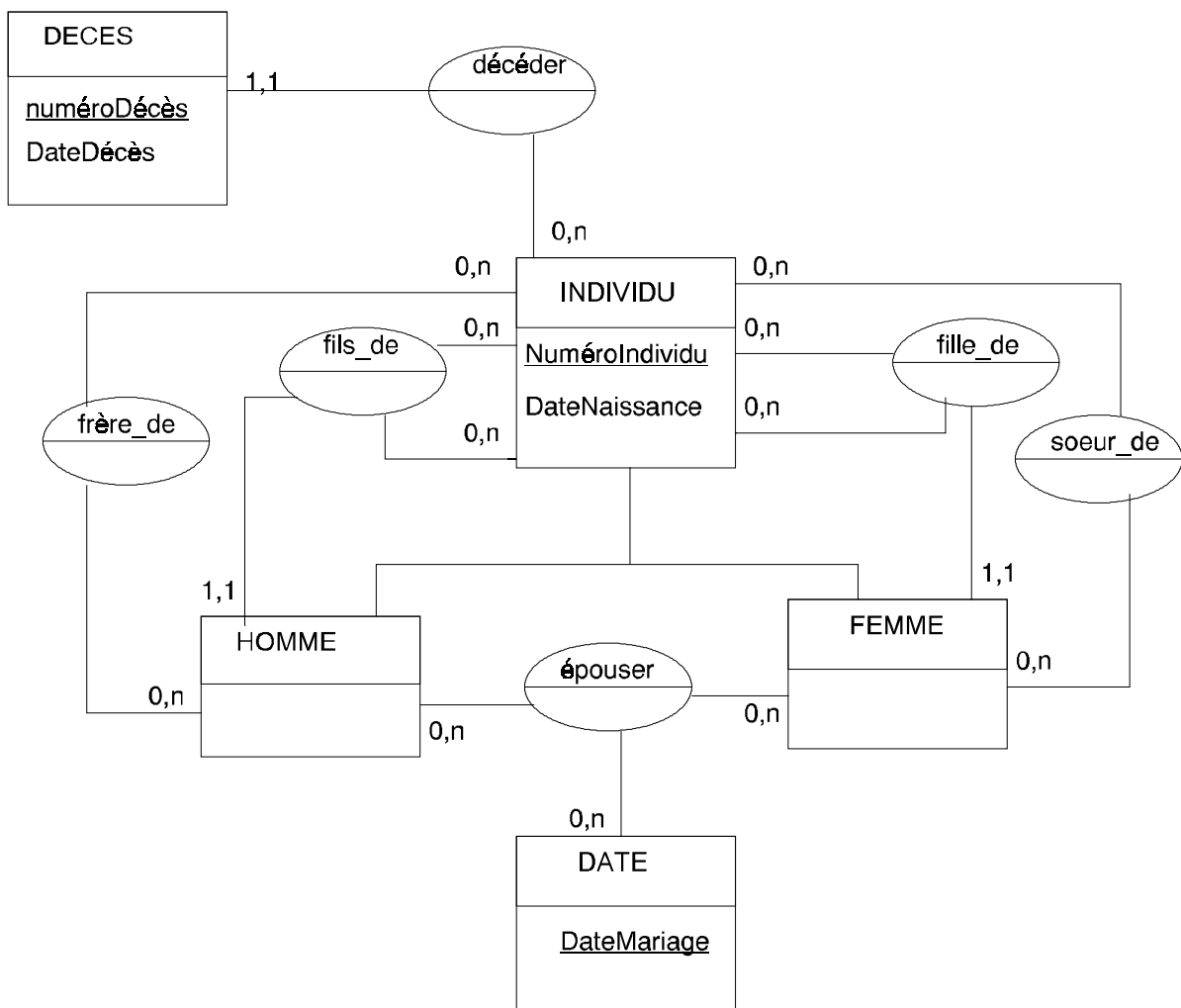
L'encapsulation, par laquelle on cache certains attributs ou encore le détail de l'implémentation des opérations, en n'accordant de visibilité que sur les seules interfaces.

Une méthode est une opération permise sur une classe ; elle retourne l'état d'un objet ou d'une partie d'objet de la classe.

La signature d'une méthode est sa vue externe, c'est-à-dire son nom, la classe sur laquelle elle intervient, le type des arguments (ou de la classe) utilisée et en général le résultat. L'utilisateur s'intéresse donc essentiellement à la signature d'une méthode.

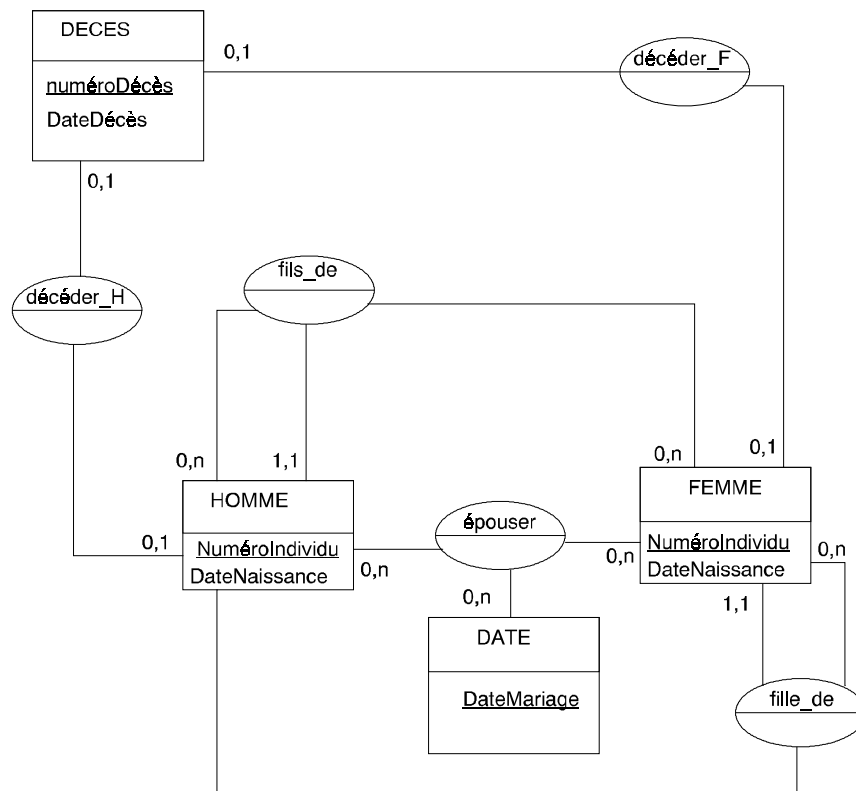
Les méthodes peuvent être publiques - elles sont alors visibles de toutes les classes et tout le monde peut les utiliser ; elles peuvent être privées - elles ne sont visibles qu'à l'intérieur de leur classe (elles sont spécifiques à leur classe).

Voici un modèle conceptuel de données exprimant la généalogie ; il intègre une généralisation/spécialisation. Il peut se justifier si les spécialisations HOMME et FEMME disposent de propriétés particulières non indiquées ici.



Un modèle avec spécialisation.

On peut cependant préférer un modèle plus classique, sans généralisation/spécialisation :



Un modèle sans spécialisation.

3.4. CONSTRUCTION DU MODELE CONCEPTUEL DES DONNEES E/R .

3.4.1. Mode opératoire de la construction du MCD : la méthode directe.

3.4.1.1. Recenser le réel de l'organisation à modéliser.

L'intérêt de la modélisation conceptuelle est de parvenir le plus rapidement possible à des applications cohérentes.

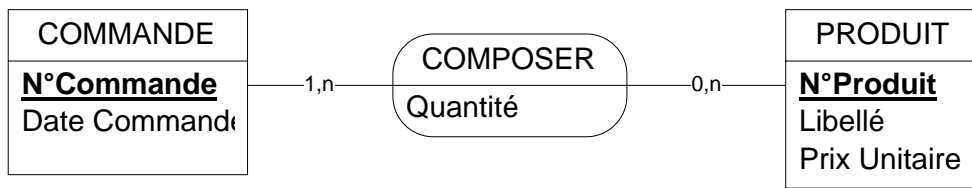
La démarche de mise au point du MCD est progressive :

- rechercher les informations de l'organisation identifiant et décrivant les objets, les propriétés, les contraintes sur les données ; ce repérage se fait auprès des responsables de l'entreprise ou du secteur concerné par l'application.
- préciser les cardinalités, propriétés, dimension des relations. L'étude précise des documents en circulation, le recueil des interviews seront nécessaires.

3.4.1.2. Repérer les objets-type, les relations-type et les propriétés-type.

- Un objet ne peut plus être une relation ;
- une relation existe obligatoirement entre deux ou plusieurs objets ; il n'existe pas de relation sans lien (patte) avec au moins deux objets (mais avec au moins un objet-type, dans le cas de relations-type binaire sur un seul objet-type) ;
- une propriété-type placée dans un objet-type ne peut se retrouver dans un autre objet-type ou dans la relation-type associant ces objets-type;
- l'organisation qu'il faut modéliser ne fait pas partie du modèle (elle n'a pas à y figurer en tant qu'objet-type ou relation-type ou propriété-type) ;
- un objet-type peut être représenté comme tel si l'on en repère plus d'une occurrence possible ;

Exemple :



Le repérage des entités du modèle.

Comment faut-il faire ?

Il suffit de répondre à la question suivante : connaissant le numéro du bon de commande, puis-je connaître de manière univoque la quantité qui compose la commande ; en d'autres termes, pour une occurrence de la relation-type, il existe une seule occurrence de la propriété-type Quantité;

- le temps peut être pris en compte dans le MCD.

On passe alors d'un point de vue synchronique à un point de vue diachronique ou historique.

Première représentation du temps : par l'objet-type DATE.

Le schéma suivant est dit "synchronique" :

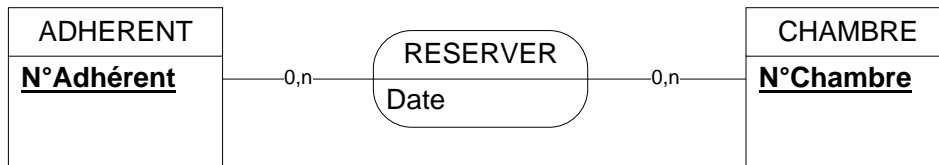


Schéma synchronique ou "à plat".

Le modèle suivant à trois objets est quant à lui "diachronique".

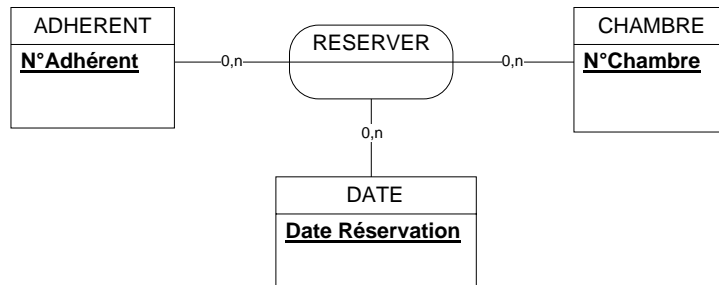


Schéma diachronique.

Le point de vue diachronique correspond à la réalité où un même adhérent peut, pour un même numéro de chambre, prendre plusieurs réservations. En effet, selon les règles du modèle conceptuel E/R, il ne peut exister qu'une seule occurrence de relation-type reliant les mêmes occurrences d'objets. Ainsi ne peut-on pas avoir les occurrences suivantes :

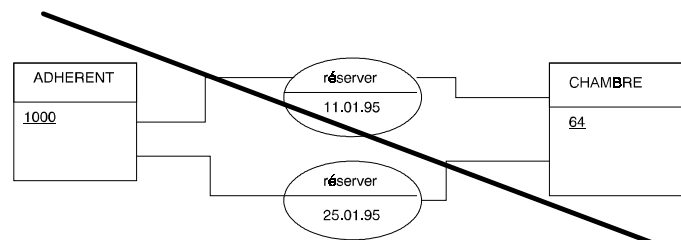


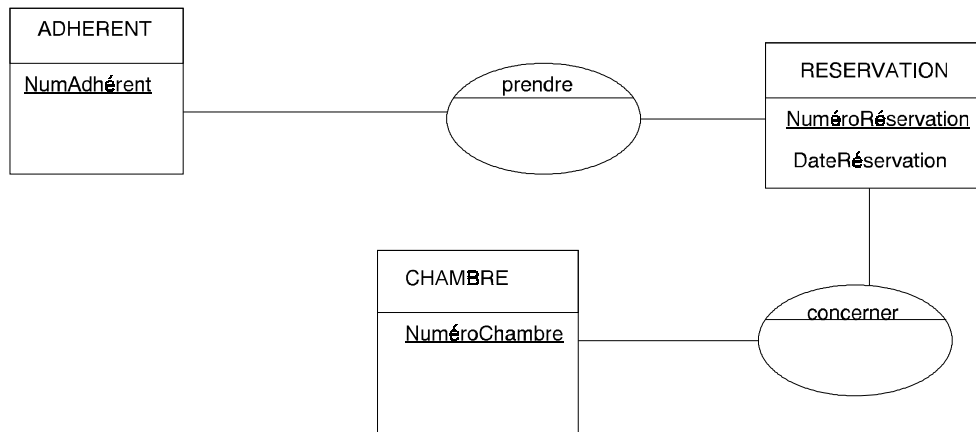
Schéma incorrect.

L'objet-type DATE est particulier :

- il n'a en général qu'une seule propriété-type, son identifiant est la date elle-même (la date et éventuellement l'heure) ;
- il entre dans une relation de dimension 3 ou supérieure.

Deuxième représentation du temps :

par la création d'un objet remplaçant la relation et une ou deux relations entre les objets anciens et le nouvel objet :



Modèle modifié en diminuant la dimension de la relation.

Dans tous les cas, la propriété de l'événement "Réserver", ici la date, est alors intégrée à l'objet-type créé : RESERVATION, car elle lui est "naturellement" rattachée.

3.4.1.3. Reconnaître les fonctionnalités, la totalité/partialité des relations.

Exemple 1 : COMMANDE "composer " PRODUIT ; la fonctionnalité de cette relation-type est N-M, c'est-à-dire plusieurs à plusieurs.

Exemple 2 : ELEVE "suivre-cours " MATIERE ; la fonctionnalité est ici N-M, c'est-à-dire plusieurs à plusieurs.

Exemple 3 : ENFANT "avoir " MERE ; la fonctionnalité est bien sûr 1-N, mais on dira de plus qu'ici, la relation est totale, car chaque occurrence d'objet-type entre dans la relation-type. La totalité exprime donc le caractère obligatoire de la participation de toutes les occurrences d'objets-type dans la relation.

Cette réflexion en termes de possibilité ou d'obligation de participation permet de déterminer les cardinalités (cf. 3.2.10) :

- Cardinalité minimum = 0 → possibilité de participation ;
- Cardinalité minimum = 1 → obligation de participation.

3.4.2. Les règles de validation du MCD E/R.

La validation du MCD s'obtient par la mise en cohérence de tous les éléments le constituant. Un modèle bien formé s'obtient en outre grâce au respect de certaines règles pratiques et formelles.

3.4.2.1. Règles pratiques.

3.4.2.1.1. Règles pratiques concernant les objets-type.

Exemple 1 :

Si l'on repère une relation "réserver une chambre ", avec une propriété "nombre de jours, il devra y avoir au moins deux objets-type, CLIENT et CHAMBRE.

Exemple 2 :

En première approche, un objet-type existe si un identifiant-type "naturel" existe au préalable (ou s'il n'a pas besoin d'être créé). Il en va ainsi de Numéro-de-Facture, qui induit l'existence d'un objet-type FACTURE.

3.4.2.1.2. Règles pratiques concernant les relations-type.

On peut transformer une relation-type en un objet-type si la relation-type dispose d'un identifiant-type et si la réalité est mieux décrite par cette autonomisation de la relation-type sous forme d'un objet-type (cf. la transformation de "réserver" en RESERVATION).

3.4.2.2. Règles formelles de validation.

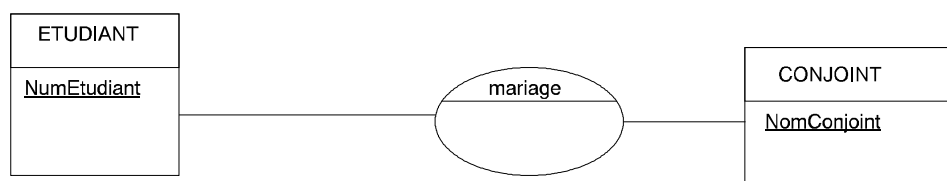
3.4.2.2.1. Unicité de valeur pour les propriétés-type.

Pour chaque occurrence d'objet-type, chaque propriété-type ne doit accepter qu'une seule valeur. La répétitivité est donc interdite.

Exemple : Soit l'objet-type ELEVE, muni des propriétés Nom, Prénoms. Si prénom peut prendre plusieurs valeurs pour un même ELEVE, il conviendra de créer des propriétés prénom1, prénom2, prénom3, etc. ou de créer un objet-type PRENOM relié à l'objet ELEVE par une relation-type.

Pour chaque propriété-type il doit y avoir une valeur. Si ce n'est pas le cas, on créera aussi un nouvel objet-type en relation à l'objet-type précédent.

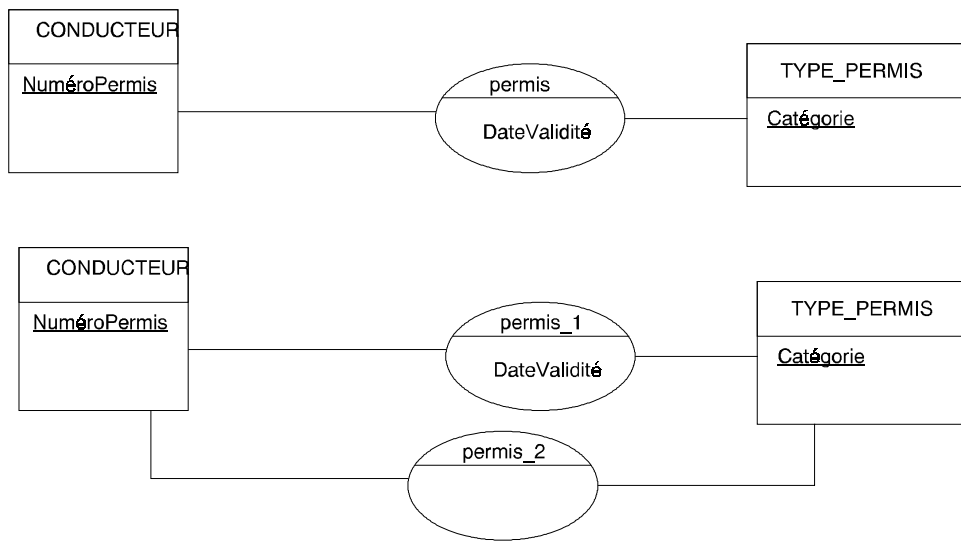
Exemple : Soit l'objet-type ETUDIANT pour lequel on a besoin de connaître l'identité du conjoint. Puisque tous les étudiants ne sont pas forcément mariés, il faut créer la relation-type "mariage " reliant ETUDIANT et CONJOINT.



Création d'un nouvel objet-type relié.

Si le problème de l'unicité et de la présence de valeur concerne une propriété portée par une relation-type, on créera deux relations-type.

Exemple :



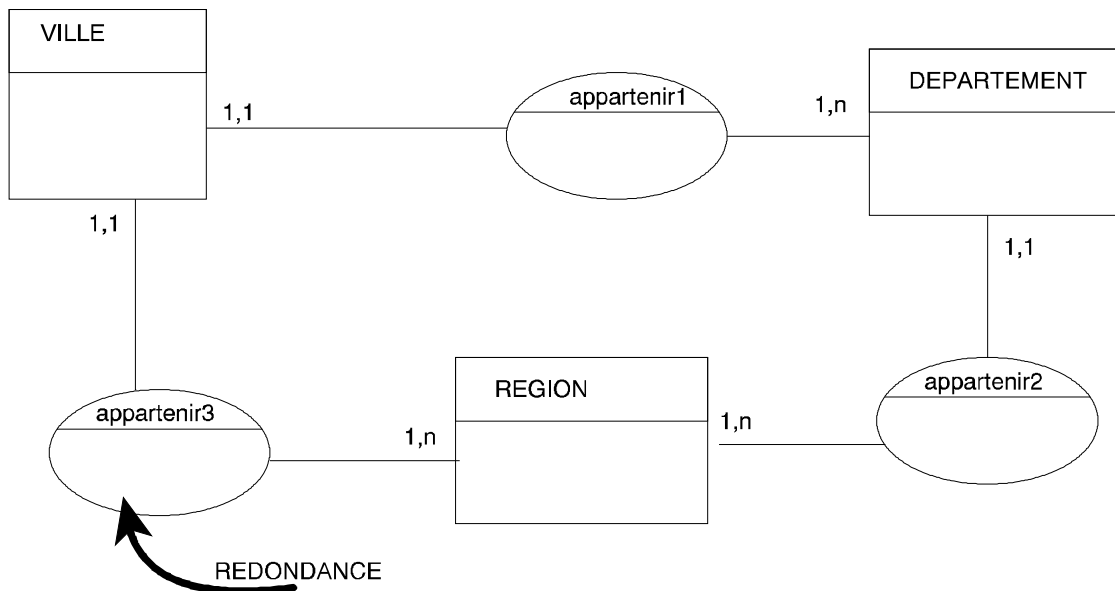
Respect de l'obligation de valeur de propriété dans une relation-type.

3.4.2.2.2. Normalisation des relations-type.

La propriété-type portée par une relation-type doit dépendre de l'identifiant-type de cette relation-type dans son intégralité. Si tel n'était pas le cas, cette propriété-type devrait se retrouver dans l'un des objets de la relation-type, ou bien portée par une autre relation-type de dimension inférieure.

3.4.2.2.3. Suppression de relations-type redondantes.

Exemple : "appartenir3" est redondante, car elle exprime une réalité déjà connue grâce à "appartenir1" et "appartenir2". Au niveau conceptuel, on supprimera une telle relation-type.

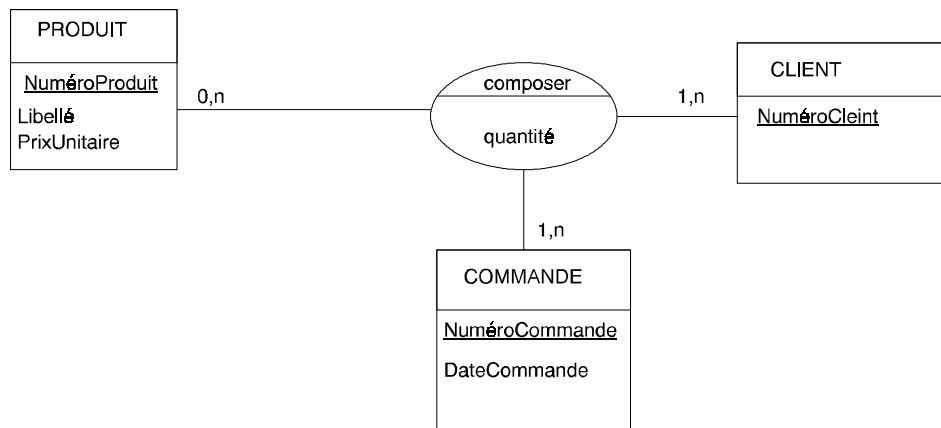


Repérage et élimination des redondances.

3.4.2.2.4. Décomposition de relations-type en relations-type de dimensions inférieures.

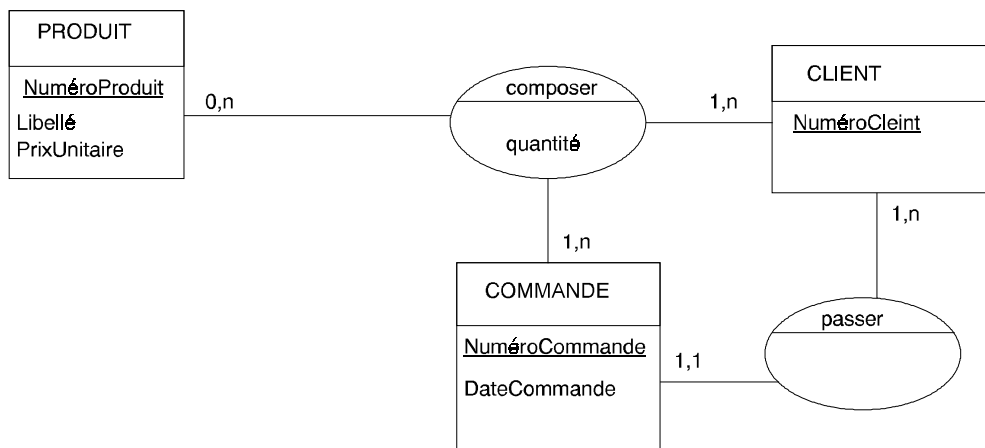
La décomposition est possible et souhaitable si l'on peut ainsi décrire la réalité sans perte d'information.

Exemple :



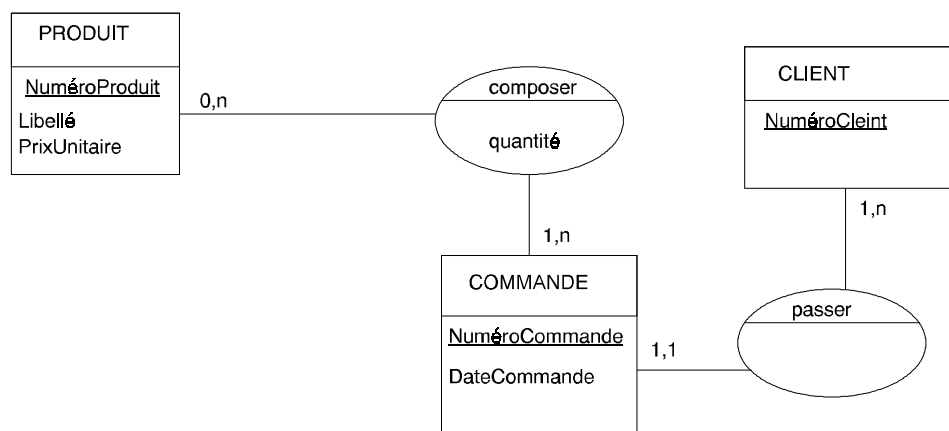
Une relation à décomposer.

On dispose d'une connaissance supplémentaire sur la réalité : une commande est passée par un client et un seul :



Utilisation de contraintes connues pour décomposer.

Du modèle précédent, on déduit le suivant, équivalent et sans perte d'information. Il est en outre plus facile à interpréter :



Une relation parfaitement décomposée.