

MERISE2

OBJECTIFS MERISE 2 ET NOTIONS DE BASE

LES MODÈLES

MODÈLE CONCEPTUEL DE DONNÉES "ÉTENDU"

LES MODÈLES DE FLUX

MODÈLE CONCEPTUEL DES TRAITEMENTS ANALYTIQUE

LES CYCLES DE VIE DES OBJETS

LES OBJECTIFS DE MERISE 2

- **fournir** un ensemble complet de démarches, modèles et méthodes pour la conception et le développement de SI
- **conserver** une compatibilité totale avec Merise
- **amélioration de certains points clés :**
 - technique de raffinement
 - intégration des diagrammes de flux de données
 - enrichissement de la sémantique du modèle de données
 - représentation du cycle de vie du système et des objets
 - meilleure distinction entre les niveaux organisationnel et logique
- **extension vers des aspects techniques**
 - prise en compte des nouvelles architectures d'applications
 - prise en compte des nouvelles interfaces
- **outillage** par des logiciels d'aide à la conception et au développement

Notions de base

→ Les systèmes

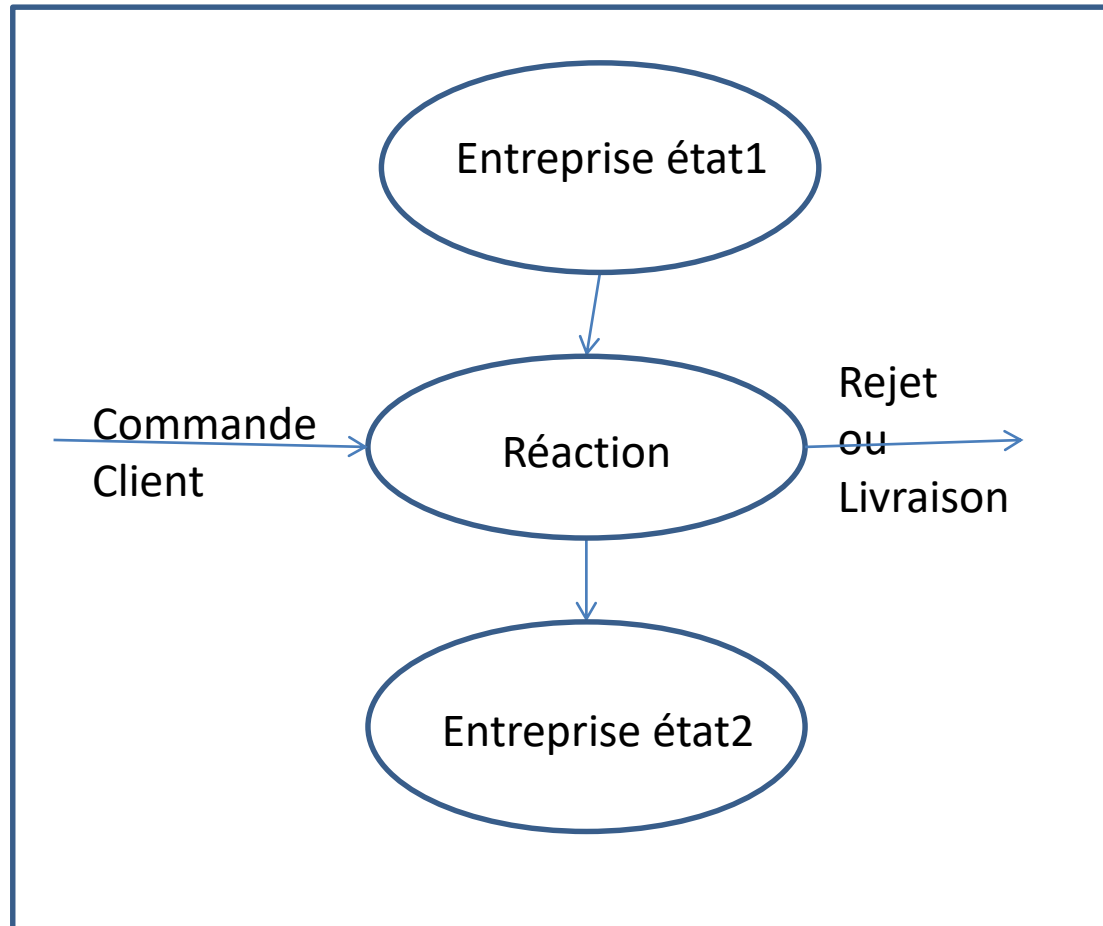
Système : "un tout constitué d'éléments unis par des relations, ces éléments et ces relations étant munis de propriétés"

Exemple : le système "d'entreprise" est composé d'éléments tels que des employés, des services, des articles, des emplacements de stockage... Les propriétés décrivant ces éléments sont le matricule de l'employé, son nom, la référence de l'article.. Entre ces éléments on trouve des relations telles que "est rattaché" entre un employé et un service, "est stocké" entre un article et un emplacement de stockage... Les propriétés de ces relations sont la date d'entrée dans le service, la quantité stockée...

Etat d'un système : l'ensemble des valeurs, à un moment donné, des propriétés des éléments et des relations du système.

Un système vit dans un **environnement**. Il subit de cet environnement des stimuli qui viennent le perturber et l'obligent à réagir c.a.d à **déclencher des activités** qui vont le faire évoluer vers un nouvel état.

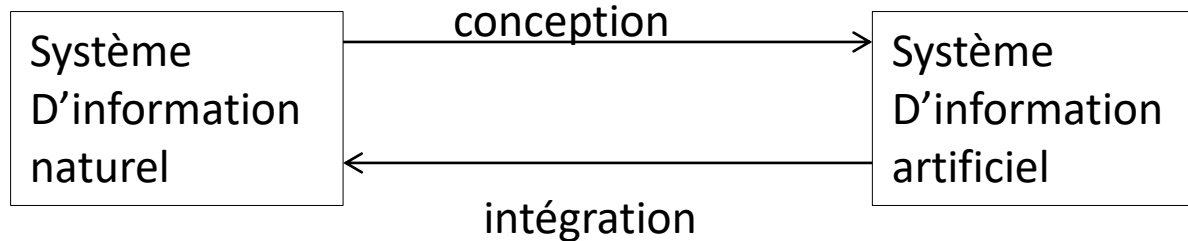
Système : "un tout constitué d'éléments unis par des relations, doté d'une activité et répondant à des stimuli".



→ Le système d'information

Le concept de SI d'une organisation recouvre deux notions:

- l'organisation réelle se transformant, agissant, communiquant et mémorisant des informations, notion qui apparente le SI à un objet naturel
- le système construit par l'homme pour représenter les actions, la communication et la mémorisation de l'information, notion qui apparente le SI à objet artificiel.



→ Conception d'un SI

Cycle de vie

Tout projet est mené dans le cadre *d'une démarche par étapes*, appelée *cycle de développement ou cycle de vie*.

Ce cycle se situe sur une échelle de temps qui part de *l'étude de l'objet naturel* à *l'intégration du système artificiel* dans l'objet naturel.

La partie cycle de vie prise en compte par Merise est découpée en quatre principales périodes :

- **la conception** (descriptions détaillées des spécifications fonctionnelles),
- **la réalisation** (description logique et physique des données, production des programmes et des consignes d'utilisation...),
- **la mise en œuvre** (mise en place effective du système dans son environnement réel)
- **la maintenance du système** (adaptation du système aux évolutions de l'entreprise).

Plus finement, on distingue :

- **le Schéma Directeur**
- **l'Etude Préalable**
- **l'étude détaillée**
- **la réalisation**
- **Préparation de la Mise en Oeuvre et Mise en Oeuvre**
- **Maintenance**

• **le Schéma Directeur** : définition des domaines d'étude, planification du développement de chaque domaine. Il fixe les moyens en personne, machine... S'il n'y a pas de schéma directeur récent une **Etude d'Opportunité** est nécessaire.

• **l'Etude Préalable** : elle permet de déterminer *le domaine sur lequel porte le projet*, les *structures concernées et l'organisation des postes de travail*. Elle décrit les circuits de l'information et *les procédures*, en précisant pour chacune d'elle sa nature et son degré d'automatisation.

Elle définit également *les moyens informatiques à mettre en œuvre, les coûts, les délais* des différentes étapes de *l'étude détaillée*.

L'objectif est ici de définir la mission, d'établir un diagnostic de l'existant et de proposer de nouvelles orientations de gestion, d'organisation et technique.

L'étude préalable peut être décomposée *en trois étapes* :

- un *recueil* se soldant par un bilan de l'existant,
- une *conception* permettant de préciser les nouvelles orientations de gestion, d'organisation et technique
- une *appréciation* permettant en particulier de planifier la suite du projet.

• **l'étude détaillée** : elle détermine les spécifications fonctionnelles en respectant les solutions retenues à l'issue de l'étude préalable. Elle se scinde en deux étapes : la conception globale ou **Conception d'Ensemble (Merise 2)** et la conception détaillée ou **Conception Fonctionnelle Détaillée (Merise 2)**.

• **la réalisation**. Elle se scinde en deux étapes : **Conception Technique Détaillée (Merise 2)** qui détermine les descriptions logique et physique des données, l'architecture des programmes, etc., la **Réalisation** proprement dite qui aboutit à la production des programmes et comporte les tests unitaires et les tests d'intégration.

- **Préparation de la Mise en Oeuvre et Mise en Oeuvre** : concerne la préparation de la nouvelle organisation en particulier la conduite du changement et les supports de formation. La mise en œuvre proprement dite concerne davantage le basculement des données et les changements de logiciels.

- **Maintenance** : adaptation du système aux évolutions de l'entreprise

Les trois premières étapes correspondent à la conception du système, les suivantes concernent sa réalisation et son lancement.

conception du système

- le Schéma Directeur
- l'Etude Préalable
- l'étude détaillée

réalisation et lancement.

- la réalisation
- Préparation de la Mise en Oeuvre et Mise en Oeuvre
- Maintenance

→ Différences principales entre Merise et Merise 2

Le cycle de référence de Merise 2 propose un découpage légèrement plus fin que le découpage initial de Merise.

- Il sépare en deux étapes distinctes la *Conception d'Ensemble* et la *Conception Fonctionnelle détaillée*. Dans Merise ces deux étapes étaient appelées *Etude Détaillée*.
- Il propose une nouvelle phase la *Conception Technique Générale*, définie lors de la Conception d'Ensemble. En effet, la plupart des choix techniques sont généralement faits à l'issus de cette étape.
- Il sépare *la Préparation de la Mise en Œuvre* de *la Mise en Œuvre* proprement dite.

Les niveaux d'abstraction

La conception d'un SI se fait en utilisant des modèles permettant de représenter les données (aspects statiques) et les traitements (aspects dynamiques) du futur système.

Les niveaux d'abstraction ont pour but de permettre *une modélisation progressive*, par niveaux de préoccupation. Ces niveaux sont au nombre de 4.

- Niveau conceptuel
- Niveau organisationnel
- Niveau logique
- Niveau physique

→ Niveau conceptuel

La description conceptuelle du système permet de représenter sa raison d'être et sa finalité en s'appuyant sur ses objectifs et les réalités qui le contraignent. Il s'agit dans un premier temps de décrire les **règles de gestion** qui permettront l'élaboration des modèles conceptuels de données et de traitements. **Une règle de gestion traduit un objectif prioritaire sans se soucier de la manière de le mettre en oeuvre.**

Exemple intuitif : considérons les carrefours d'un système routier

Règles de gestion : - le système assure une circulation fluide des véhicules
- il s'agit d'éviter au mieux les collisions

Ces règles traduisent les deux objectifs primordiaux (fluidité de la circulation et limitation des collisions).

→ Niveau organisationnel

La description organisationnelle du système représente l'organisation permettant d'atteindre les objectifs définis au niveau conceptuel. Il s'agit donc de décrire le fonctionnement du SI dans le cadre d'une organisation cible. Les descriptions du niveau organisationnel pour les traitements et les données ne préfigurent pas des moyens à mettre en œuvre pour y parvenir.

La description organisationnelle permet de décrire les vues partielles du système pour chaque type d'acteur par site de l'organisation. Il s'agit de décrire **D'OU** sont visibles les données et les traitements, **QUI** fait quoi en matière de données et de traitement, **QUAND** réalise-t-on les traitements et manipule-t-on les données.

Exemple intuitif :

Règles d'organisation : le système alterne les flots de circulation, en autorisant un passage en séquence dans une durée limitée des véhicules des différents axes.

→ Niveau logique

Le niveau logique concerne la conception du logiciel correspondant aux parties à automatiser du système. Il inclue une description logique des données c'est à dire une description dans un formalisme compatible avec l'état de l'art (modèle relationnel, modèle objet, fichiers, etc.) mais encore portable par rapport à des choix techniques précis.

Il inclue également des modèles logiques de traitements décrivant le guidage fonctionnel, les boîtes de dialogue, l'arborescence des fenêtres...

Exemple intuitif : L'alternance sera assurée à l'aide de signaux lumineux, placés sur chaque axe de circulation...

→ Niveau physique

le niveau physique tient compte des préoccupations et des choix techniques nécessaires à l'implantation physique des données et à la mise en place des traitements : langage de programmation, choix du SGBD, taille mémoire, etc.

Chaque niveau d'abstraction offrent une panoplie de modèles. Chaque étape du cycle de vie d'un projet nécessite de s'intéresser :

- à un ou plusieurs niveaux d'abstraction,
- en utilisant un ou plusieurs modèles offerts par chaque niveau d'abstraction.

Parcours au sein des niveaux d'abstraction

Pour un projet donné il s'agit de fixer :

- **La démarche.** Par exemple, certaines étapes du cycle de vie peuvent être "fusionnées" dans le cas de "petits projet".
- **Le choix des modèles de différents niveaux d'abstraction** qui seront utilisés lors de chaque étape.

Le parcours au sein de ces niveaux d'abstraction lors des différentes étapes d'un projet d'informatisation est appelé **cycle d'abstraction**

Le processus est itératif, de plus suivant l'étape de la démarche dans laquelle on se trouve, la description d'un niveau d'abstraction peut être :

- plus ou moins **détaillée (technique d'affinement)**
- sur une couverture **plus ou moins large du domaine**

Le niveau de détail augmente au fur et à mesure que l'on progresse dans les étapes. ^{14/3}

LES MODELES

Un modèle est *une représentation abstraite de la réalité*. Par exemple une carte routière est une représentation abstraite de routes, un plan d'architecture est une représentation abstraite d'un bâtiment, etc.

Un modèle doit :

- correctement représenter la pensée du modéliseur,
- permettre de communiquer sans ambiguïté

Pour ce faire il s'agit d'offrir des formalismes précis et normalisés pour modéliser les différents éléments (données et traitements) des systèmes d'information et ceci aux différents niveaux d'abstraction.

Par abus de langage nous utiliserons le mot "modèle" pour parler aussi bien des formalismes que de leur utilisation. C'est le contexte qui précise le sens. Lorsque nous parlerons par exemple, de modèles de données il s'agit du formalisme utilisé pour modéliser des données. Quand nous parlerons du modèle de donnée du domaine vente. Il s'agit de son utilisation pour modéliser les données concernées par le domaine "vente".

Les trois axes de modélisation de Merise 2

Merise 2 propose une démarche globale de modélisation basée sur 3 axes de modélisation :

- l'axe d'architecture ou fonctionnel qui permet de décrire ce que **fait le système (les activités)** ;
- l'axe statique qui permet de décrire ce qu'**est le système (les données)** ;
- l'axe dynamique ou comportemental qui permet de décrire comment **se comporte le système** (les processus et la succession de transformations effectuées sur les données).

Axes de modélisation et niveaux d'abstraction

Merise 2 offre différents modèles permettant de représenter les aspects **statique, fonctionnel** (architecture) et **dynamique** d'un système et ceci à différents niveaux d'abstraction : **conceptuel, organisationnel et logique**.

Les modèles physiques ne font pas partie de la méthode.

Par rapport à Merise, Merise 2 :

- propose *de nouveaux modèles*

- établit une distinction* beaucoup plus claire *entre les modèles organisationnels et les modèles logiques*.

les modèles de Merise 2

QUOI

D'où
Qui, quand

quand

INTERFACE	APPLICATION		
	STATIQUE	DYNAMIQUE	ARCHITECTURE
MC	MCD	MCTA CVO	MFC
MOT	MOD	MOTA CVO	MFO
Maquettes IHM	MLD MLDr	MLT MLTr	SALMI, SAL SALR

est

Se comporte

fait

- Modèle de Contexte (MC)
- Modèle Conc. des Données (MCD)
- Modèle Conceptuel des Traitements Analytique (MCTA)
- Cycle de Vie des Objets (CVO)
- Modèle de Flux Conceptuel (MFC)
- + Règles de Gestion

- Modèle Org. des Traitements (MOT)
- Modèle Org. des Données (MOD)
- Modèle Org. des Traitements Analytique (MOTA)
- Modèle de Flux Organisationnel (MFO)
- + Règles d'Organisation

- Modèle Logique des Données (MLD)
- Modèle Logique des Données Réparties (MLDr)
- Modèle Logique des Traitements (MLT)
- Modèle Logique des Traitements Répartis (MLTR)
- + Primitives

Ces modèles sont complétés pour les études d'architectures techniques complexes par les modèles suivants :

- Schéma d'Architecture Logique des Moyens Informatiques (SALMI)
- Schéma d'Architecture Logique (SAL)
- Schéma d'Architecture Logique Répartie (SALr)

Suivant le type de projet et l'étape du cycle de vie, il convient de sélectionner une partie de ces modèles.

Exemple : Dans le cas d'un "petit projet" (quelques hommes/mois, pas de répartition, logiciel classique) on ne développera que les MC, MFC, MCD, MCTA, CVO, MLD, MLT, maquettes.

Rappel : Ces modèles peuvent être élaborés à différents niveaux de détail, sur une couverture partielle ou totale du domaine d'étude et s'inscrivent dans une démarche itérative.

Modèles Conceptuels

La description conceptuelle permet de représenter la finalité du système et sa raison d'être, en s'appuyant sur ses objectifs et les réalités externes qui le contraignent.

Elle s'appuie sur un ensemble de **Règles de Gestion** qui décrivent le "quoi" de l'entreprise.

Une Règle de Gestion est une traduction conceptuelle des objectifs choisis et des contraintes acceptées par l'entreprise. La plupart du temps il s'agit de règles d'actions liées aux traitements ou de règles de calcul liées aux données.

Exemples :

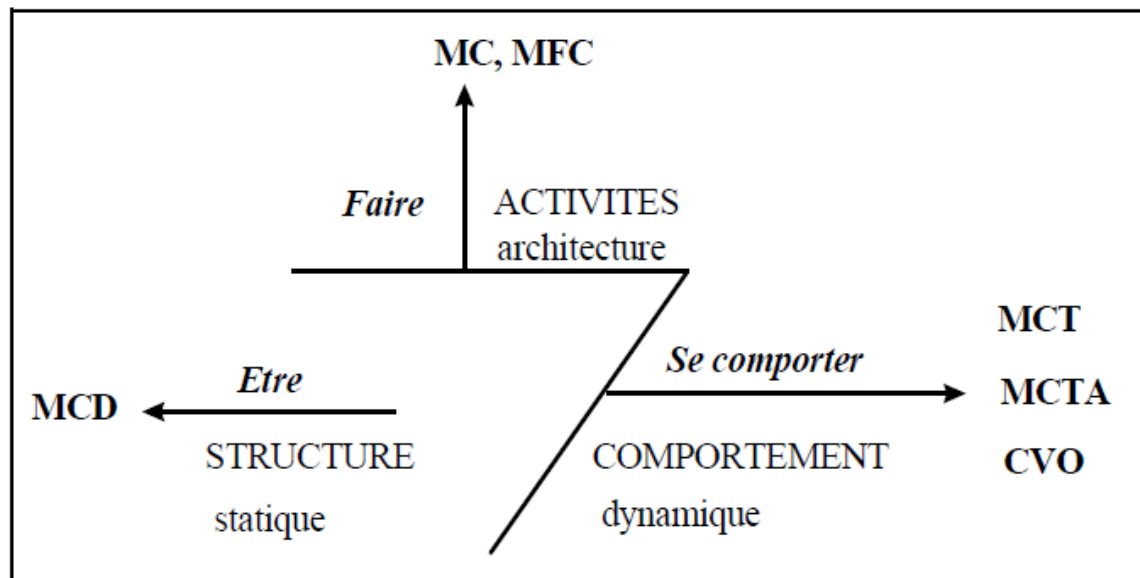
- "un inventaire doit être dressé périodiquement"*
- " tout produit livré sera entré en stock"*
- "la centrale d'achat sera libre d'imposer des jours de commandes"*
- "le salaire de base est égal à l'indice multiplié par la valeur du point "*

Attention : l'analyste n'a aucune initiative sur les règles de gestion, son unique rôle est de les trouver (interview), les faire valider puis les utiliser pour élaborer les différents modèles conceptuels.

Le niveau conceptuel traite des événements et fournit des résultats sans se soucier de la manière dont sont acquises et restituées les informations portées par ces événements et résultats.

Les différents modèles proposés sont :

- le Modèle de Contexte (MC) et les Modèles de Flux Conceptuels (MFC)
- le Modèle Conceptuel des Données (MCD)
- le Modèle Conceptuel de Traitements (MCT), le Modèle Conceptuel des Traitements Analytique (MCTA) et les Cycles de Vie des Objets (CVO)



Les principales évolutions par rapport à Merise sont :

- l'utilisation des MFC avec des techniques d'affinement (*refinement*),
- la modification des modèles de traitement (MCT-> MCTA),
- l'extension du MCD,
- l'introduction des CVO.

Modèle Conceptuel de Données "étendu"

Le Modèle Conceptuel de Données est une représentation statique du système d'information de l'entreprise. Au niveau conceptuel il s'agit de modéliser les données fondamentales de l'entreprise (les invariants décrits par des règles de gestion). Il ne doit être fait aucune hypothèse sur l'utilisation ultérieure de ces données.

Le Modèle Conceptuel de Données "étendu" (Merise 2) apporte des extensions au formalisme individuel adopté par Merise.

Ces extensions ont pour objectifs :

- de préciser et d'enrichir la description des objets en mettant en évidence des propriétés, des contraintes d'intégrité supplémentaires et des objets historiques,
- d'aider la validation du modèle de traitement (MCTA),
- de permettre à différents utilisateurs de partager le même modèle (accès à différents niveaux de décomposition ou de spécialisation),
- de permettre une relative stabilité par rapport aux évolutions de l'entreprise ou de l'application (une modification de détail laissant les niveaux supérieurs inchangés).

Commençons par *rappeler le MCD originel*, avant de détailler *les extensions proposées* dans Merise 2.

Rappel de Vocabulaire

Propriété

information élémentaire, conforme au choix de gestion de l'entreprise (adresse d'un client, référence d'un article, prix d'achat..).

Objet type (ou individu ou entité)

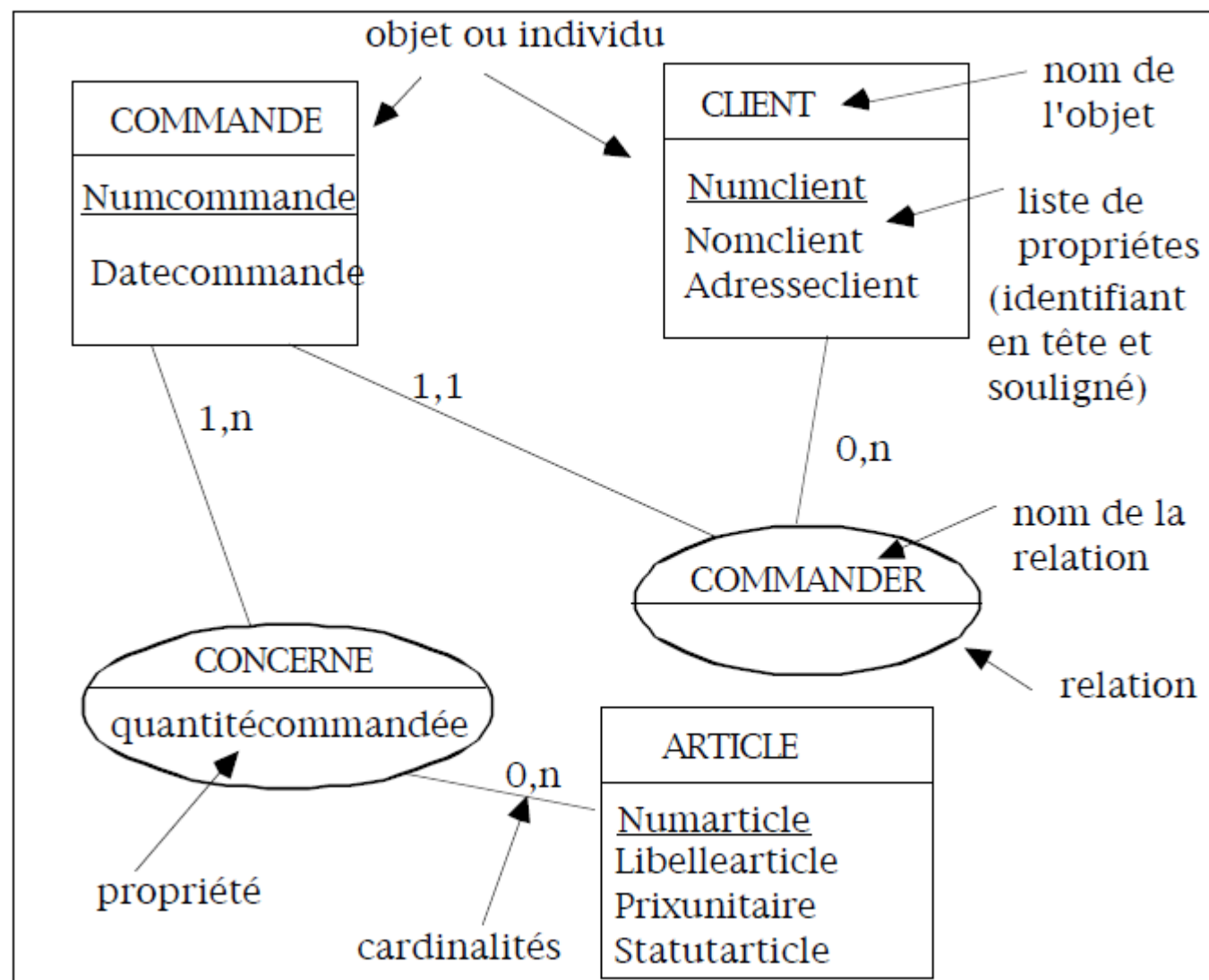
regroupement de propriétés, reflet d'une entité présentant un intérêt pour le système étudié, *doté d'une existence propre et identifiable* (par exemple Article doté des propriétés libellé, prix unitaire, nature... et identifié par sa référence).

Relation type (ou association)

représente une association entre plusieurs objets; *son existence est conditionnée par celles des objets mis en relation* (par exemple la relation "commander" entre client et commande).

Cardinalités

indiquent pour chaque couple objet-relation les nombres minimum et maximum de valeurs de la relation pouvant exister pour chaque valeur d'objet.



➤ La Propriété :

La propriété correspond à la plus petite partie logique d'information manipulée dans l'entreprise, il peut s'agir d'une propriété simple (un nom), mais aussi d'une propriété composée (une adresse).

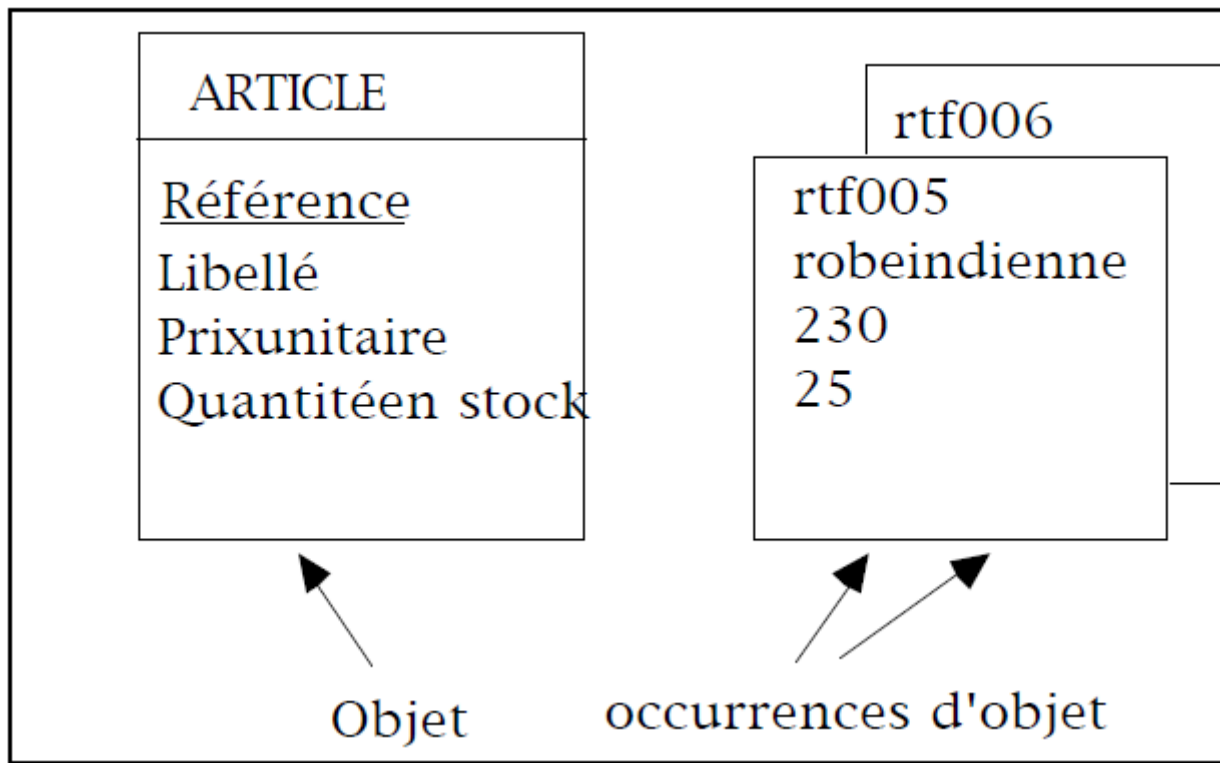
Il faut autant que possible :

- éviter les propriétés **calculables et redondantes**,
- faire la chasse **aux synonymies** (référence article et num de produit) et **aux polysèmes** (préciser s'il s'agit de l'adresse du client ou celle du fournisseur).

➤ Objet, Occurrence d'objet

L'objet est un *regroupement de propriétés (ou attributs)*. Il est le reflet d'une entité de l'organisme *dotée d'une existence propre*. Il est identifiable et ne doit représenter qu'un seul concept sémantique.

Chaque propriété d'un objet peut être assimilée à une variable. Les valeurs qui lui seront affectées représentent *les occurrences de cette propriété*. Si l'on affecte une valeur à chacune des propriétés composant un objet, on obtient une occurrence de celui-ci.



Dans un objet toute propriété dépend fonctionnellement de l'identifiant.

- Une propriété non identifiant ne dépend pas d'une autre propriété non identifiant.
- Une propriété non identifiant ne dépend pas que d'une partie de l'identifiant.

➤ Relation

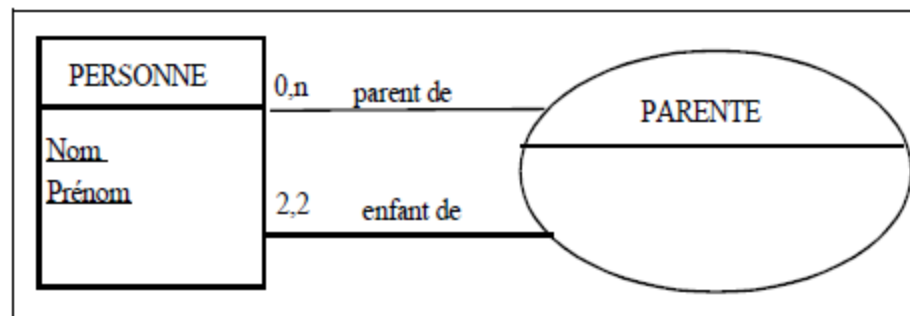
La relation peut être vue comme une *association entre divers objets du modèle*. Elle est *la traduction des verbes du langage de l'entreprise*.

La **dimension** de la relation est égale au *nombre d'objets* de sa collection. Elle est dit n-aire (binaire, ternaire..) ou de dimension n (2, 3..).

Une relation *peut ou non être porteuse de propriétés*.

Contrairement aux objets, une association *n'a pas d'existence propre*. Son existence est conditionnée à celle des objets qui la composent.

Relation réflexive : Elle relie un objet à lui même. Une occurrence de la relation associe une occurrence de l'objet à une autre occurrence de ce même objet (exemple PARENTE).



➤ Cardinalités

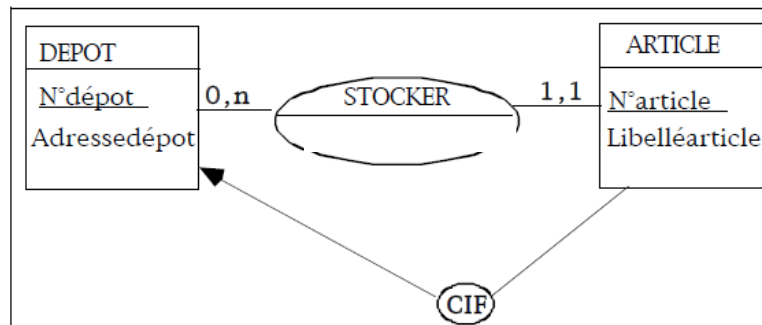
Les cardinalités représentent pour chaque couple objet-relation les nombres minimum et maximum d'occurrences de la relation pouvant exister pour une occurrence d'objet.

Elles sont *des traductions de règles de gestion*.

➤ Contrainte d'Intégrité Fonctionnelle

Une Contrainte d'Intégrité Fonctionnelle (CIF) définie sur une relation représente le fait que l'un des objets de sa collection est identifié sans aucun doute par la connaissance d'un ou plusieurs autres.

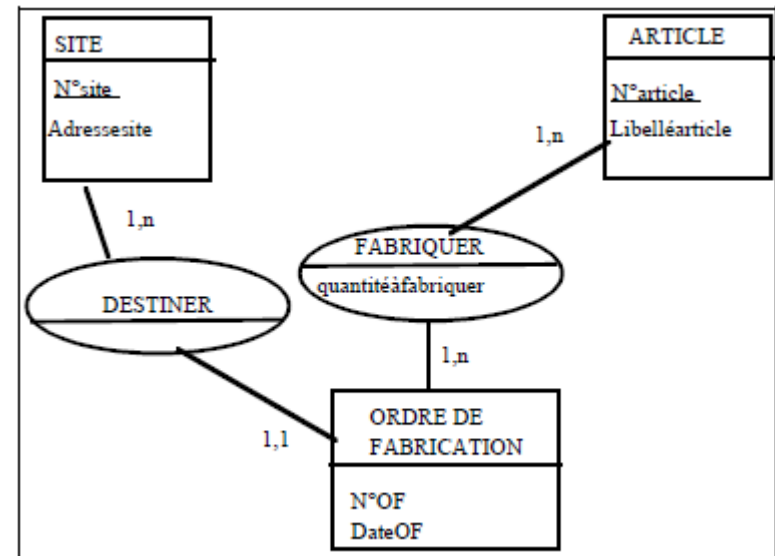
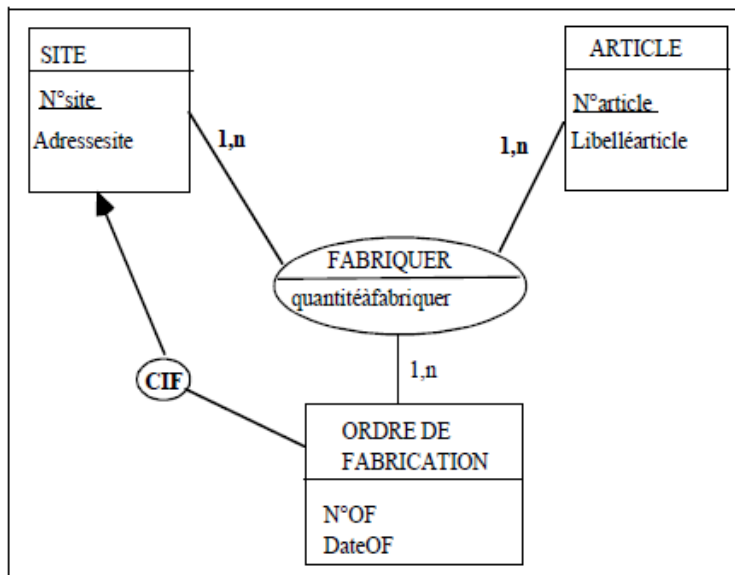
Une relation binaire ayant des cardinalités (0,1) ou (1,1) exprime une CIF.



Les CIF sont utilisées pour décomposer les relations. Elles permettent donc de simplifier des relations de dimensions supérieure à 2.

Exemple : dans une entreprise industrielle, les ordres de fabrication sont établis par le siège en direction de divers sites de production. On suppose la règle de gestion suivante:

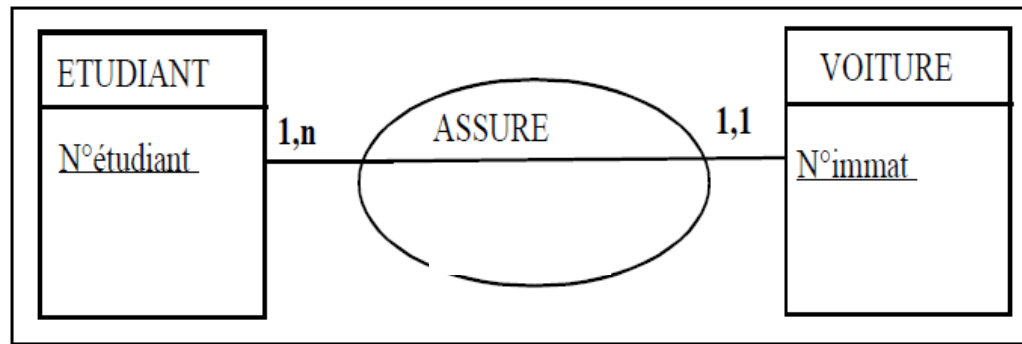
- un ordre de fabrication ne concerne qu'un seul site.



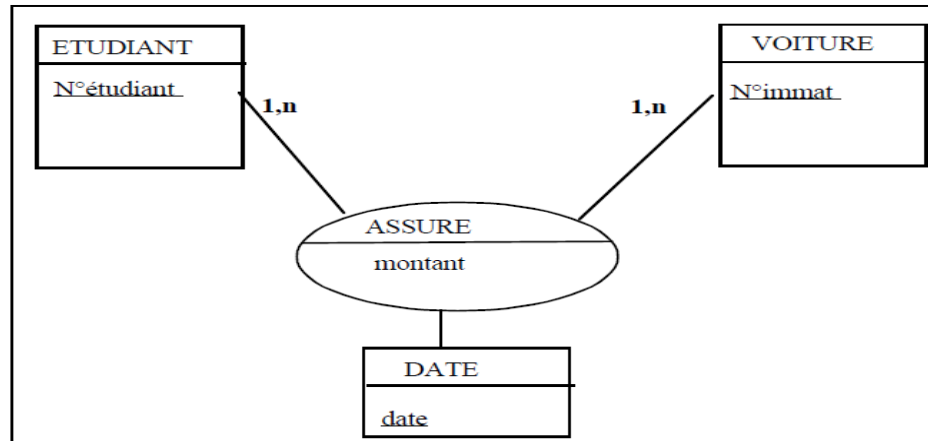
➤ Le problème de la représentation du temps

Il existe deux points de vue concernant la représentation du temps.

Représentation synchronique : Le temps n'intervient pas comme élément discriminant. Elle correspond à une vision instantanée de la réalité modélisée.



Représentation diachronique (ou historique) : Le temps intervient comme élément discriminant. Elle correspond à une vision historique de la réalité modélisée.



les principales extensions proposées dans Merise 2

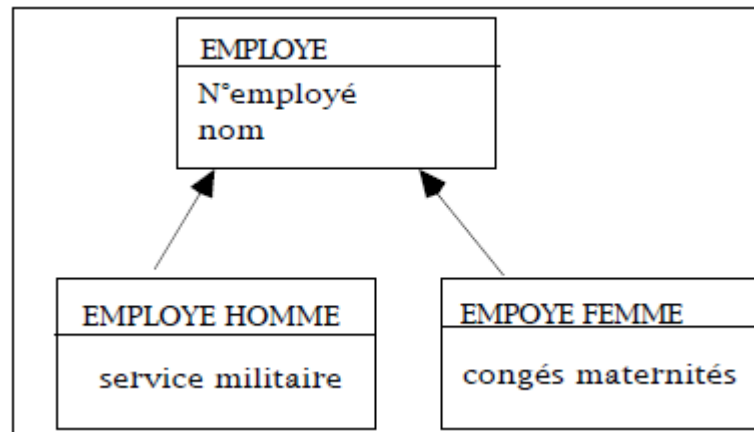
- Sous-type d'objet
- Sous-type de relation
- La structure Agrégation (pseudo entité)
- Contraintes d'intégrité statique

➤ Sous-type d'objet

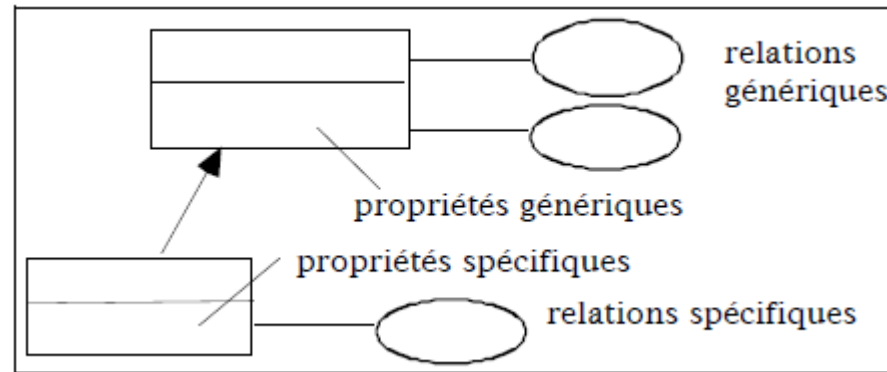
Un sous-type d'objet correspond à un sous-ensemble d'occurrences d'objet dotées de caractéristiques propres (propriétés spécifiques et/ou relations spécifiques).

On pourra aussi employer les termes **d'objet générique** (*sur-type*) et **objet spécialisé** (*sous-type*).

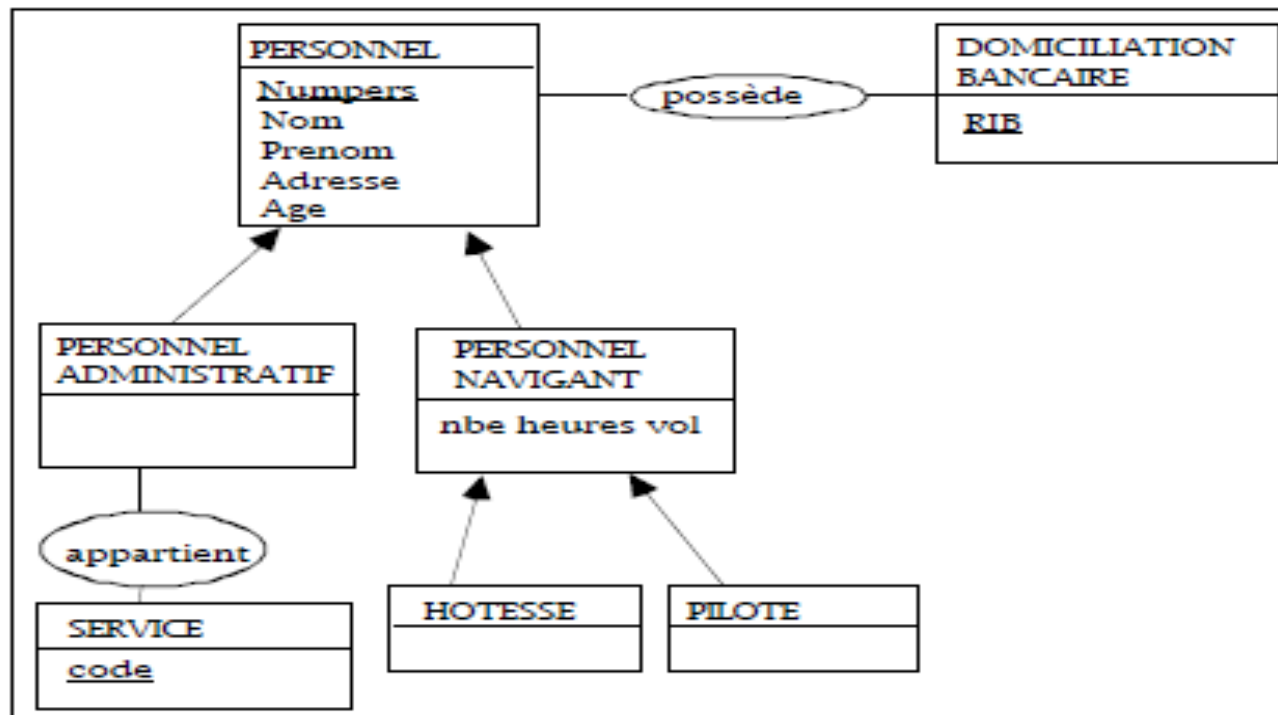
- **L'objet générique** ou sur-type porte les caractéristiques communes aux objets spécialisés.



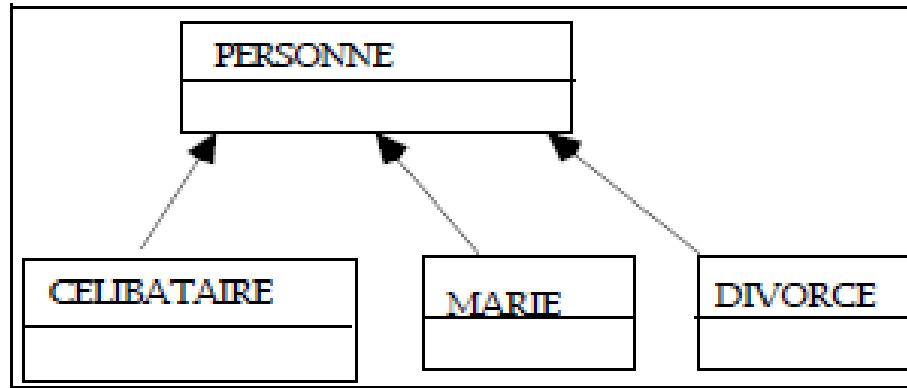
- Chaque sous-type **hérite** des propriétés et des relations du sur-type.



- Le mécanisme de spécialisation permet d'avoir plusieurs niveaux de description d'un même objet : de la description la plus générale à la description la plus spécifique. Il s'agit d'une spécialisation par "catégorie".



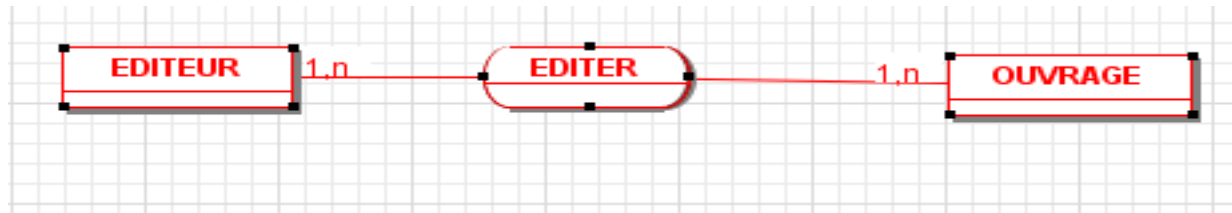
- Cas particulier de spécialisation : *la spécialisation par états*
Un sous-type correspond dans ce cas à un état de l'objet. Il peut y avoir transfert d'une occurrence de l'objet d'un sous-type à un autre sous-type



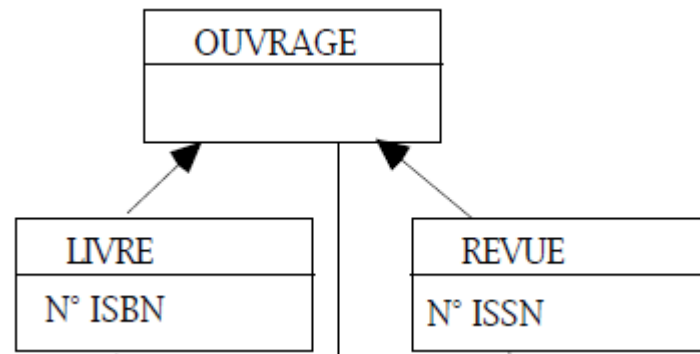
➤ Sous-type de relation

Un sous-type de relation correspond à un sous-ensemble d'occurrences d'une relation dotées de propriétés et/ou de cardinalités spécifiques.

Un éditeur édite un ou plusieurs ouvrages.



Un ouvrage peut être un livre ou une revue

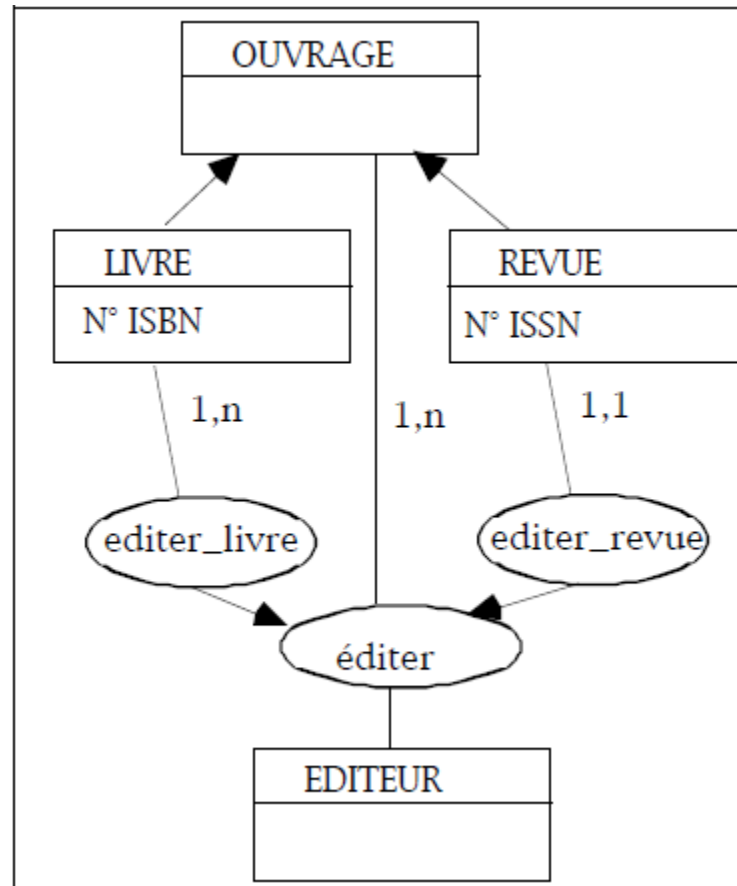


Cependant, un livre peut être édité par plusieurs éditeurs : cardinalité 1,n

Et une revue n'est éditée que par un seul éditeur : cardinalité 1,1

Comment représenter cette réalité ?

→ Spécialisation de la relation : *Sous type de relation* :

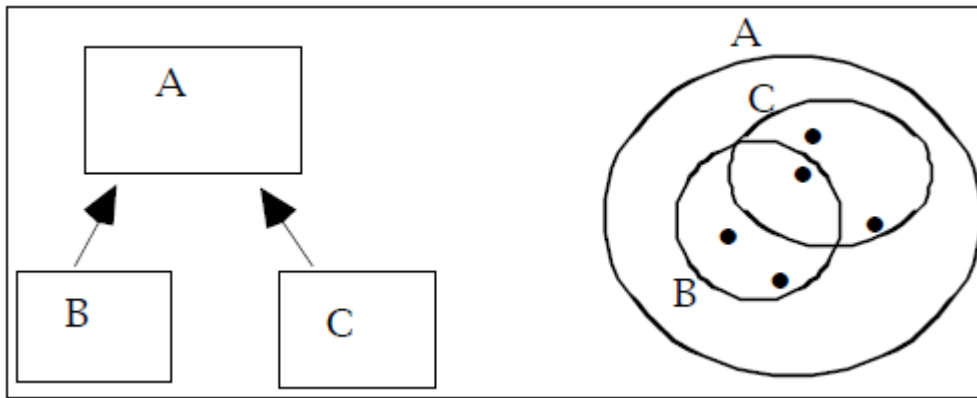


➤ Contraintes d'intégrité statique

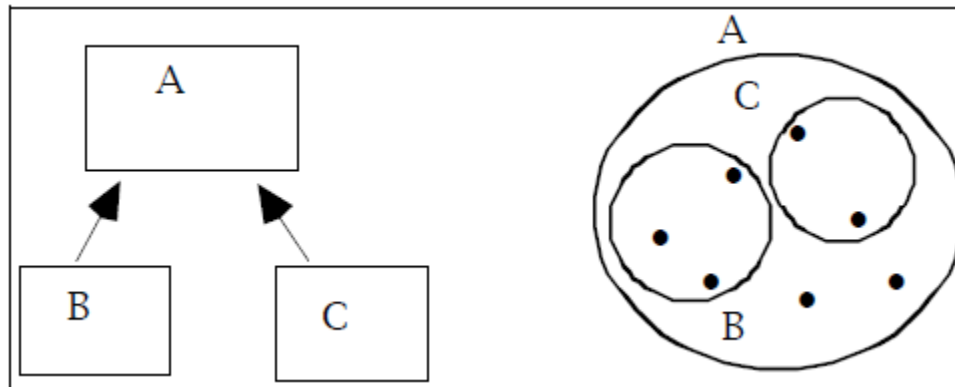
➔ Contraintes de base : couverture et disjonction

il s'agit de contraintes ensemblistes

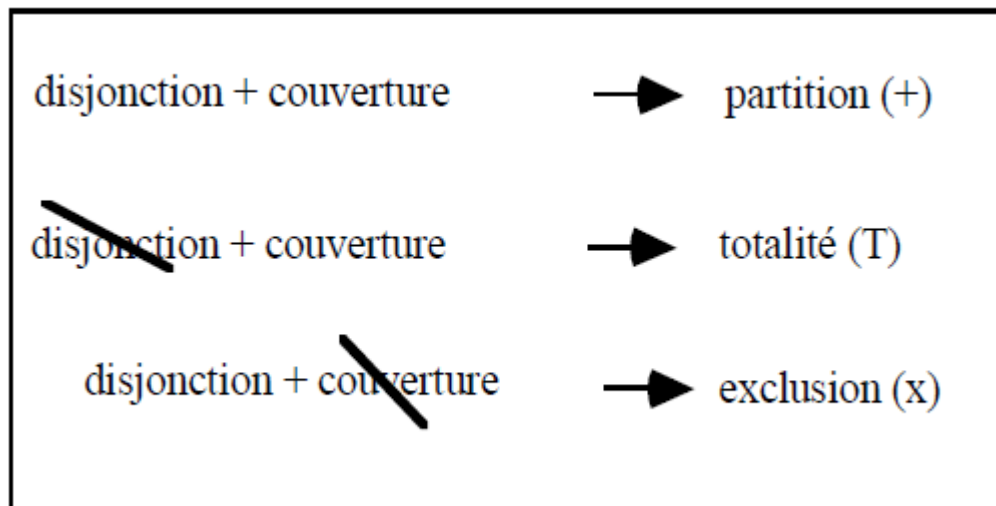
- **contrainte de couverture** : toute occurrence de l'objet générique doit appartenir à au moins un des sous-types.



- **contrainte de disjonction** : toute occurrence de l'objet générique doit appartenir à un seul sous type (les sous-types sont mutuellement exclusifs).

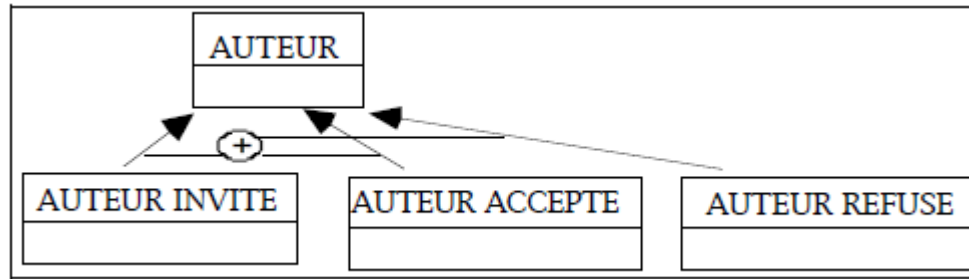


Contraintes d'intégrité entre les sous-types d'objets



Contrainte de partition :

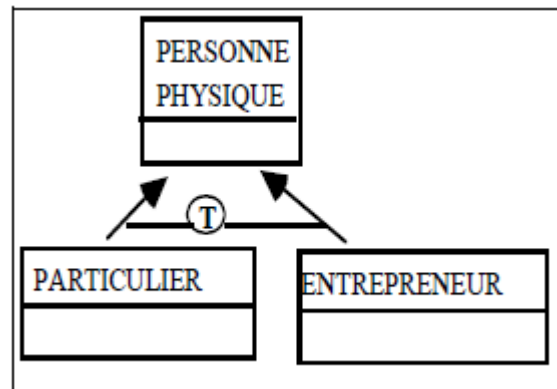
Un auteur est soit invité, soit accepté, soit refusé.



partition = disjonction + couverture

Contrainte de totalité :

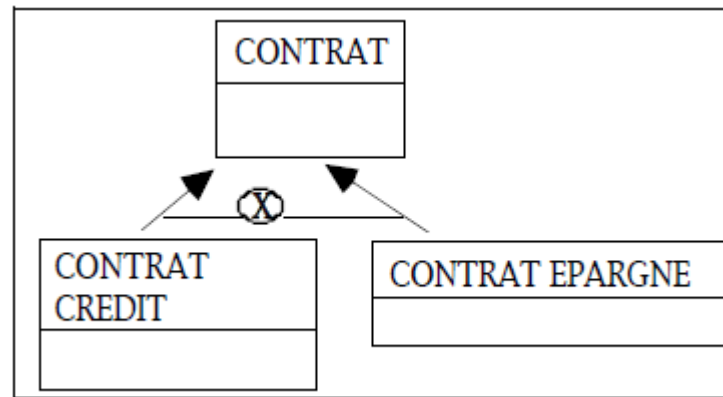
Une personne physique peut être à la fois Particulier et Entrepreneur individuel, elle est l'un ou l'autre.



totalité = couverture et non disjonction

Contrainte d'exclusion :

Un contrat ne peut être à la fois un contrat de crédit et un contrat d'épargne, il existe d'autres types de contrats.

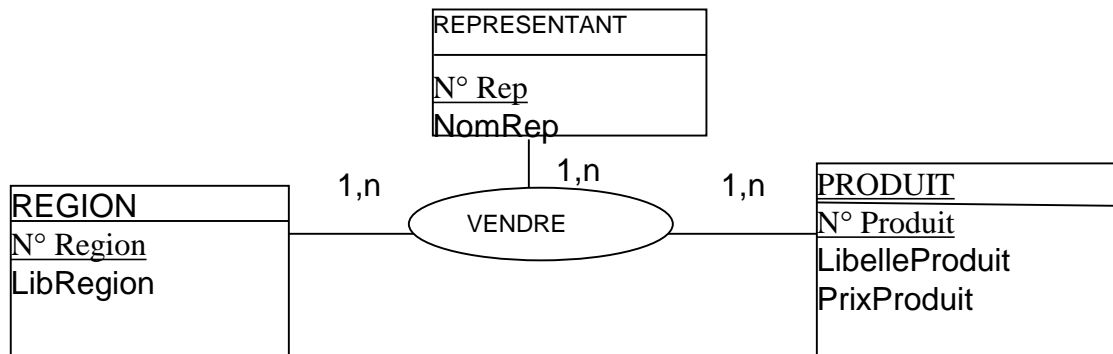


exclusion = disjonction et non couverture

➤ La structure Agrégation (pseudo entité)

La structure d'Agrégation permet de traduire les situations où une relation ne peut être représentée correctement que si on associe une entité à un groupe d'entités

Exemple : Dans une entreprise, les représentants vendent des produits dans différentes régions. *Mais un produit pour **une région donnée** n'est vendu que **par un seul représentant**.*



N°REP	N°Region	N°Produit
1	5	1
1	3	2
2	5	2
2	3	3
3	3	1
3	5	3

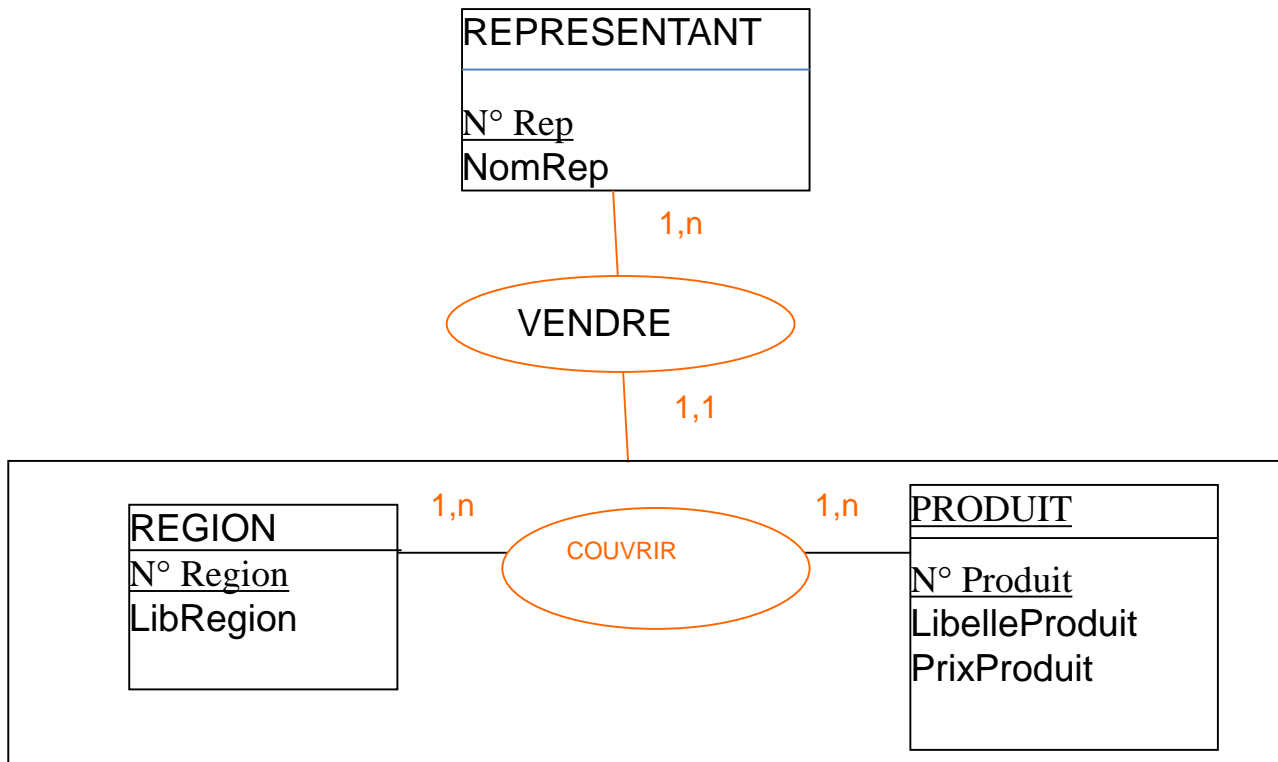
1 - D'après les règles de gestion de l'énoncé, est-il possible que le représentant 3 vende le produit 2 dans la région 5 ?

NON car un produit pour une région donnée ne peut être vendu que par un seul représentant, or le couple région/Produit (5,2) est assuré par le représentant 2

2- D'après le MCD, le représentant 3 vende le produit 2 dans la région 5 ?

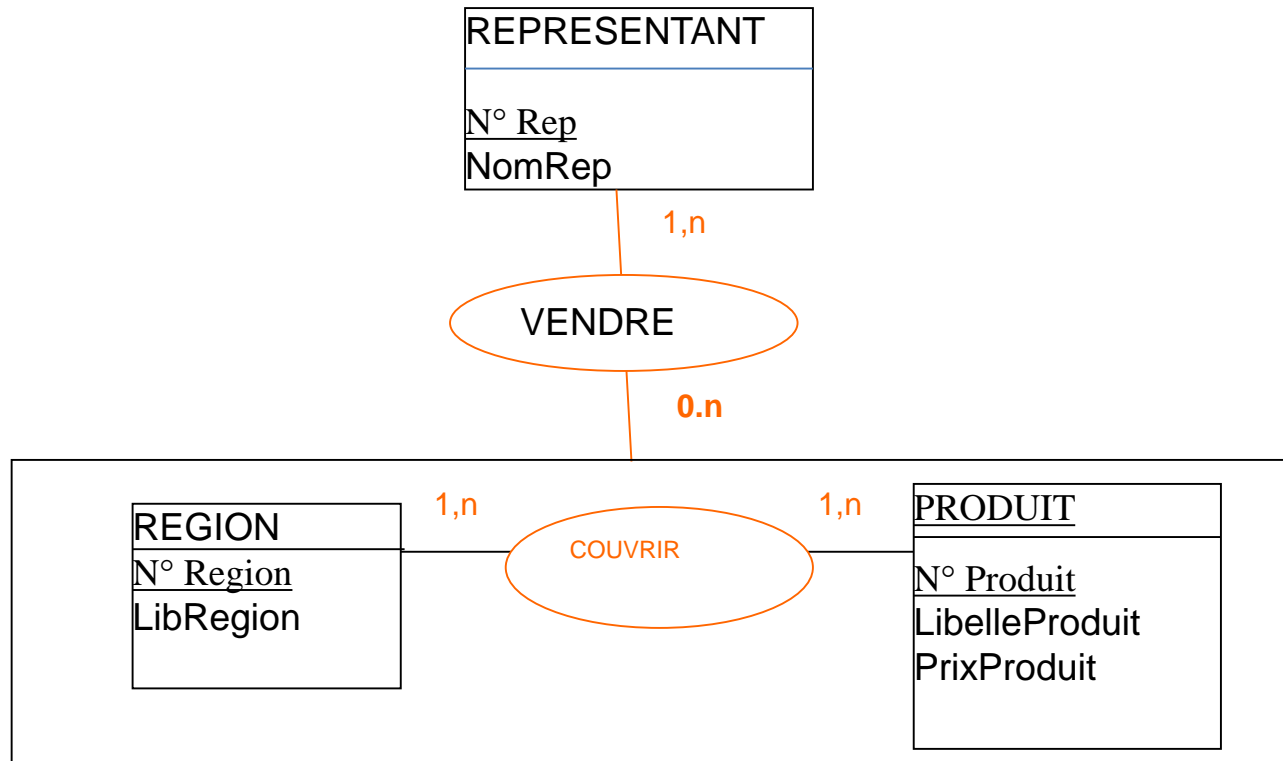
Rien ne s'y oppose conceptuellement parlant étant donné car l'identifiant de l'association vendre (N°Rep, N°Region, N° produit) est égal à 3 5 2, ce qui est différent de 2 5 2

Donc, le MCD actuel ne traduit pas correctement la réalité, il faut recourir à la *structure « agrégation »* qui permet d'associer une entité à un couple d'entités:



L'association
Couvrir est
agrégée et
constitue une
pseudo-entité

Si par contre, on considère cette fois ci qu'un produit pour une région donnée, s'il est vendu, peut être vendu par plusieurs représentants



Contraintes d'intégrité sur les relations

Les contraintes d'intégrité statiques peuvent aussi porter sur des relations ou des pattes de relations :

- * *contrainte de partition (+)*

- * *contrainte d'exclusion (x)*

- * *contrainte de totalité (T)*

- * *contrainte d'inclusion (I)*

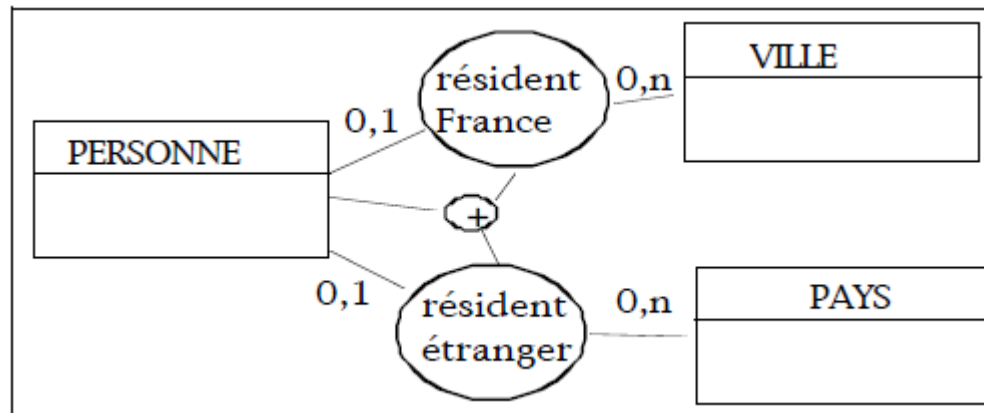
- * *contrainte d'égalité (=)*

- * *contrainte d'unicité (1)*

Contrainte de partition (couverture et disjonction)

Une contrainte de partition permet d'exprimer que toutes les occurrences d'un objet (dit objet pivot) impliqué dans deux (ou plus) relations sont présentes dans une et une seule d'entre elles (ou exclusif).

Toute personne est soit résidente en France soit résidente à l'étranger, mais ne peut être les deux.

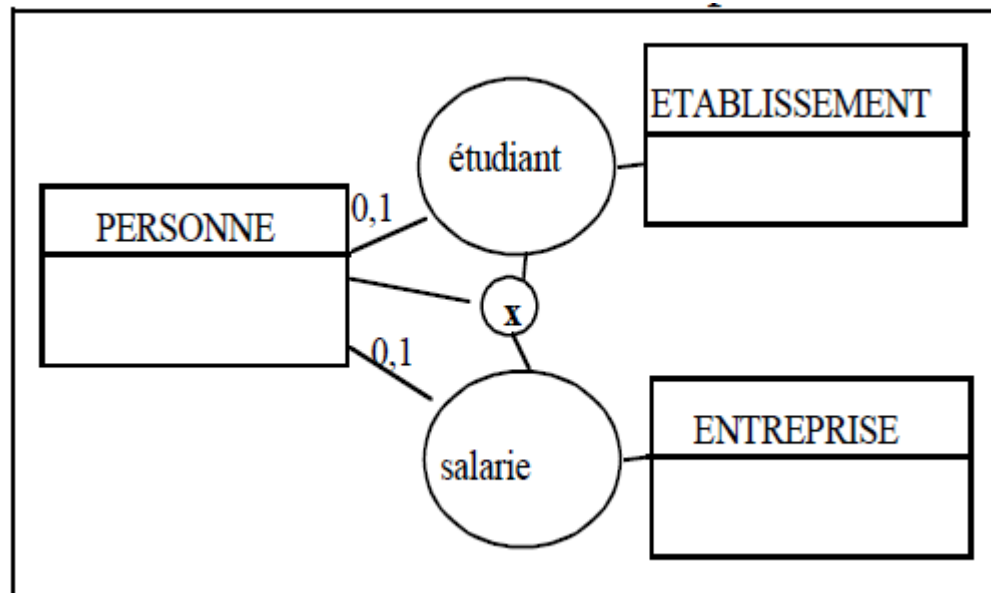


partition

Contrainte d'exclusion (non couverture et disjonction)

Une contrainte d'exclusion interdit qu'une occurrence d'un objet (dit objet pivot) impliqué dans deux (ou plus) relations soit présente dans deux d'entre elles.

On ne peut être salarié et étudiant, certaines personnes ne sont ni l'un ni l'autre

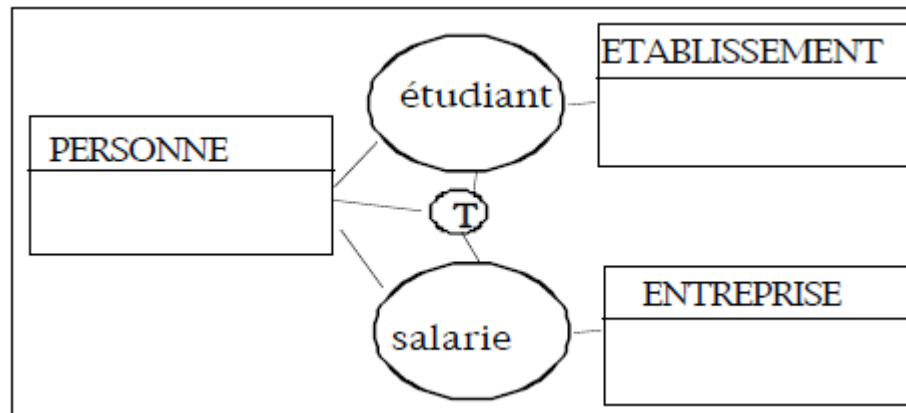


exclusion

Contrainte de totalité (couverture et non disjonction)

Une contrainte de totalité permet d'exprimer que toutes les occurrences d'un objet (dit objet pivot) impliqué dans deux (ou plus) relations sont présentes dans au moins l'une d'entre elles (ou inclusif).

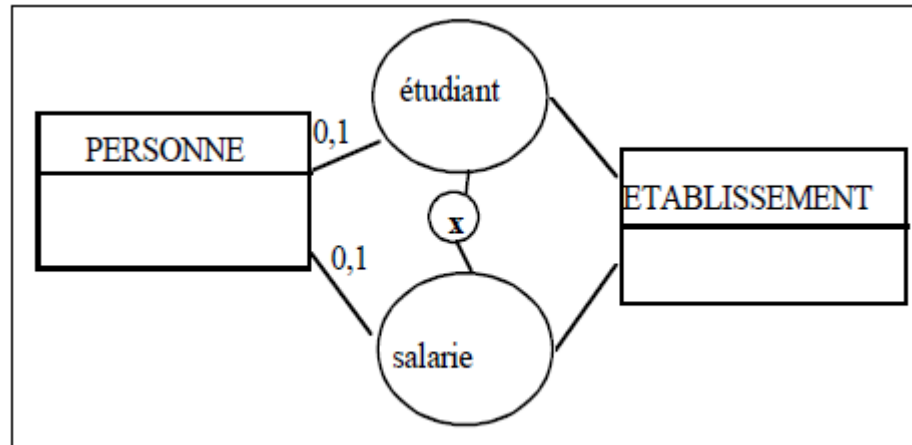
Toute personne est étudiant dans un établissement, ou salarié dans une entreprise ou les deux à la fois.



totalité

Remarque : Pour les contraintes de partition, d'exclusion et de totalité, si aucun objet pivot n'est mentionné la contrainte porte sur la collection de la relation.

On ne peut être salarié et étudiant du même établissement.



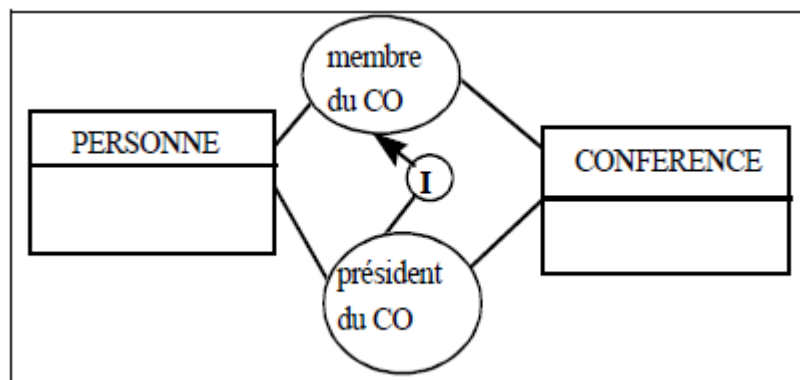
plusieurs objets pivots

Contrainte d'inclusion

cas 1 : pas d'objet pivot explicitement désigné

Une contrainte d'inclusion permet d'exprimer que l'ensemble des occurrences d'une relation est inclus dans l'ensemble des occurrences d'une autre relation.

Le président du comité d'organisation d'une conférence est choisi parmi les membres du comité d'organisation (de cette même conférence).



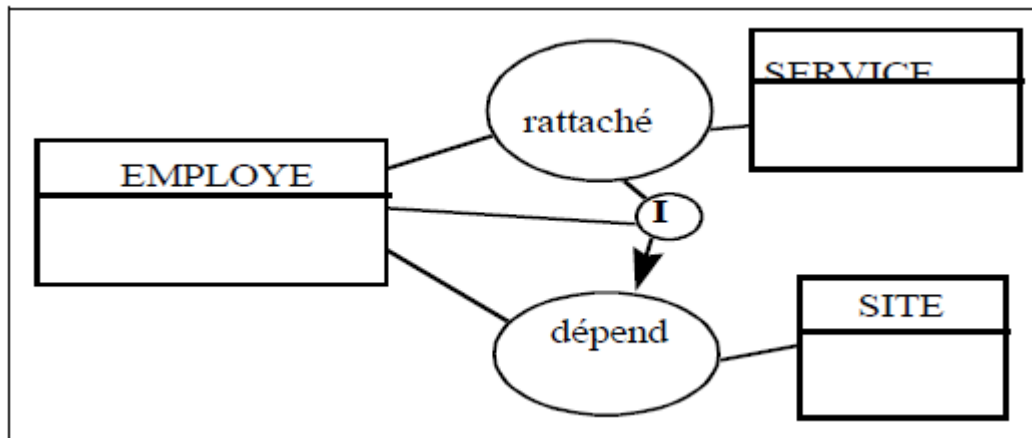
La relation président du CO est appelée "portée" de la contrainte d'inclusion, La relation membre du CO est la "cible" de la contrainte d'inclusion.

inclusion sans objet pivot explicite

cas 2 : un objet pivot

Dans ce cas la contrainte d'inclusion porte sur l'ensemble des occurrences d'un objet (dit objet pivot). L'ensemble des occurrences de l'objet pivot présentes dans une des relations est inclus dans l'ensemble des occurrences de l'objet pivot présentes dans l'autre relation.

Si un employé est rattaché à un service il doit dépendre d'un site de l'entreprise (par contre il peut dépendre d'un site sans être rattaché à un service).



portée de la CI : "rattaché", cible
de la CI : "dépend", objet pivot :
EMPLOYE

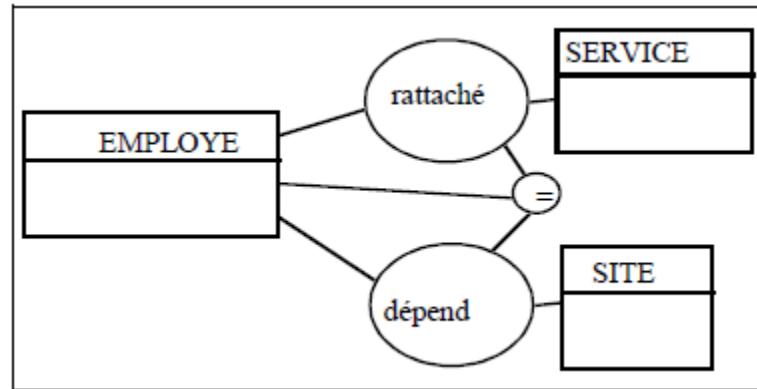
inclusion avec objet pivot

*Remarque : dans le cas 1 le pivot est constitué des deux objets
Personne et Conférence.*

Contrainte d'égalité

Une contrainte d'égalité est une combinaison de deux inclusions symétriques.

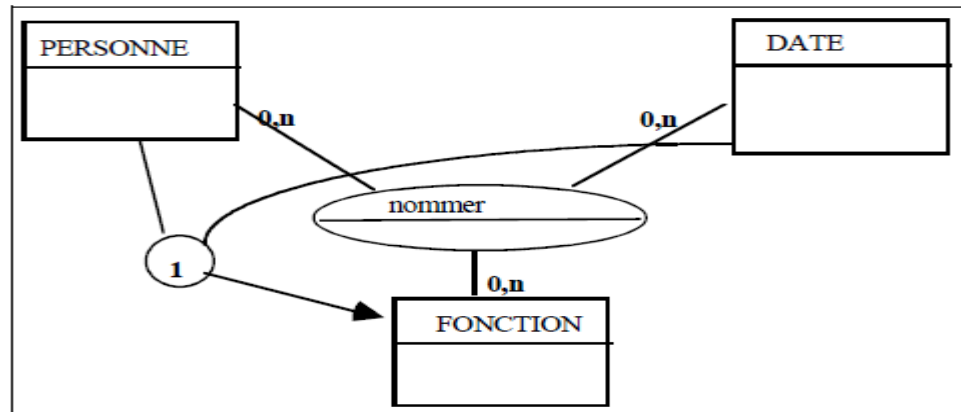
Tout employé rattaché à un service dépend de l'un des sites de l'entreprise. Inversement tout employé dépendant d'un site doit être rattaché à un service.



égalité

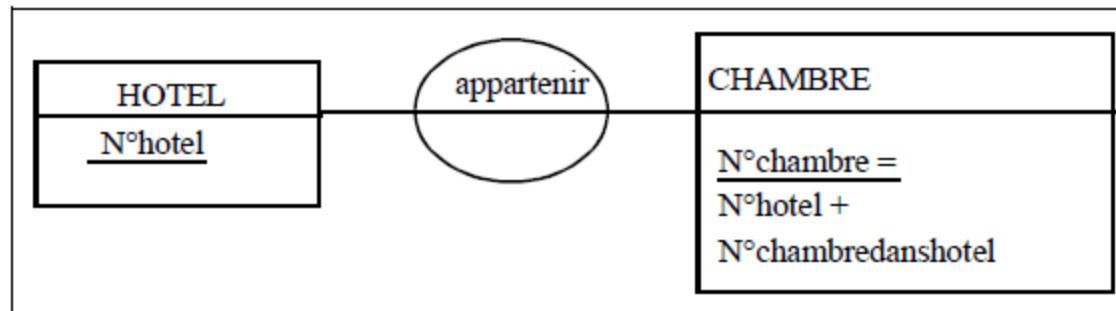
Contrainte d'unicité

Une personne à une date donnée ne peut occuper qu'une et une seule fonction.



Contrainte d'identification

Jusqu'à présent un objet était défini comme ayant une existence et donc une identification propre. Cependant il arrive que certain objet n'ont d'existence que par rapport à un ou plusieurs autres objets. Ses relations avec ces objets peuvent être utilisées pour l'identifier.



identification

RÉPERCUSSION DES NOTIONS DE MERISE 2 SUR LA TRADUCTION DES CONCEPTS DU MCD AU NIVEAU RELATIONNEL

- Traduction des sous-types d'entités
- Traduction des associations spécialisées
- Impacts des contraintes statiques sur le niveau relationnel

➤ Traduction des sous-types d'entités

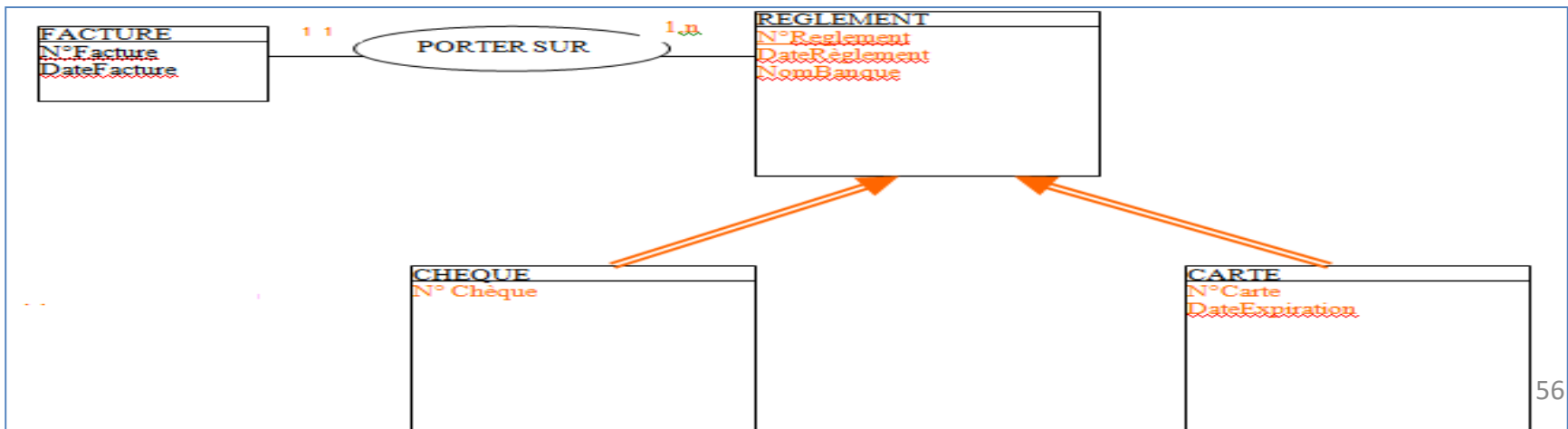
On peut traduire les sous-types d'entités de 3 manières différentes, toutes équivalentes

1) Une seule relation

2) Deux relations: Transformation des entités spécialisées en relation

3) Trois relations : Transformation de l'entité générique et des entités spécialisées en relations

Exemple pour visualiser les différentes solutions :



1) Une seule relation

Les propriétés des entités spécialisées sont regroupées dans l'entité spécialisée. **MAIS** *il faut créer un attribut permettant de différencier l'occurrence*

REGLEMENT(N°Reglement, DateRèglement, NomBanque, **TypeReglement**
N° Chèque, N°Carte, DateExpiration)



Attribut ajouté

Remarque : Cette solution est utilisable lorsque *les entités spécialisées* ne comportent pas beaucoup de propriétés.

2) Deux relations: Transformation des entités spécialisées en relation

Dans cette situation, on ne traduit pas l'entité générique en relation

CHEQUE(N°Reglement, DateRèglement, NomBanque , N°Cheque)

CARTE(N°Reglement, DateRèglement, NomBanque , N°Carte,
DateExpiration)

Remarque : Cette solution permet de faire l'économie d'une relation. En revanche, elle entraîne de la redondance car les propriétés de l'entité générique sont reprises dans chaque relation.

3) Trois relations : Transformation de l'entité générique et des entités spécialisées en relations

Les relations « spécialisées » héritent de la clé primaire de la relation « générique »

REGLEMENT(N°Reglement, DateRèglement, NomBanque)

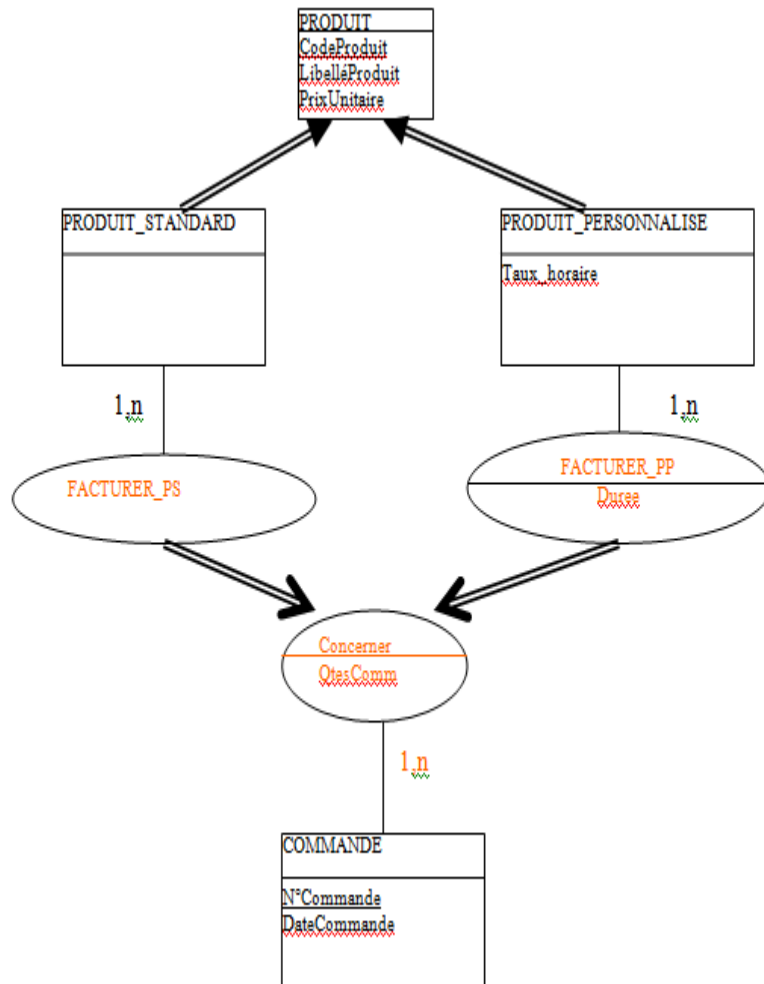
CHEQUE(N°Reglement, N°Cheque)

CARTE(N°Reglement, N°Carte, DateExpiration)

Remarque : Cette solution est à privilégier lorsque chaque entité comporte beaucoup de propriétés.

➤ Traduction des associations spécialisées

Principe général : Les sous-types d'association se traduisent de la même manière que les sous-types d'entités



Dans cet exemple nous traduirons chaque association en relation :

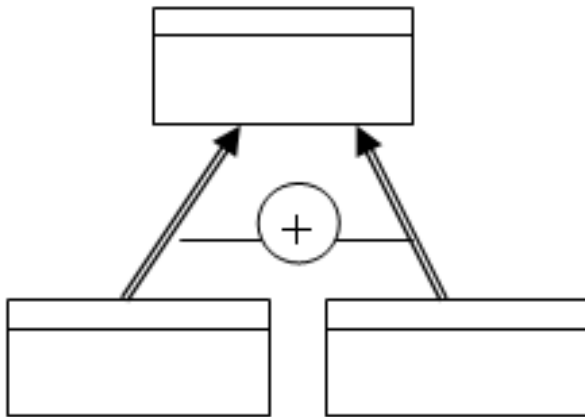
COMMANDE(N°Commande, DateCommande)
PRODUIT(CodeProduit, LibelleProduit, PrixUnitaire)
PRODUIT_STANDARD(CodeProduit)
PRODUIT_PERSONNALISE
(CodeProduit, TauxHoraire)

CONCERNER(#N°Commande, #CodeProduit, QtesCom)
FACTURER_PS(#N°Commande, #CodeProduit)
FACTURER_PP(#N°Commande, #CodeProduit, Duree)

Les sous-types d'association ont la même clé primaire que l'association générique

➤ Impacts des contraintes statiques des sous types sur le niveau relationnel

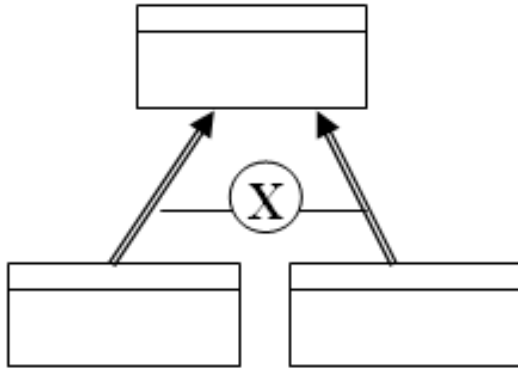
+ Partition :



PARTITION

Dans cette situation , la solution la plus adaptée est celle des 2 relations , surtout si le sur type comporte peu d'attributs

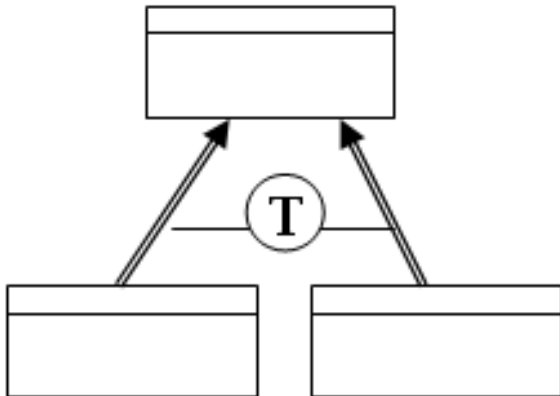
+ Exclusion :



EXCLUSION

Dans cette situation , la solution la plus inadaptée est celle des 2 relations ,
Si les nombres d'attributs des sous types sont importants , la seule solution pertinente est la définition de 3 relations

+ Totalité :



Dans cette situation , la solution la plus adaptée est celle des 2 relations , surtout si le sur type comporte peu d'attributs