

(masquer detail) (afficher detail)

## Merise - Introduction (Données)

### 1) Système d'Information et méthode Merise

### 2) Modélisation conceptuelle des données

### 3) Modélisation logique des données

### 4) Modélisation physique des données

### 5) Bases de données

## 1) Système d'Information (Index)

organisation

ensemble de personnes ayant un objectif commun et mettant en oeuvre des moyens pour les atteindre (entreprise, association, etc.)

Système d'Information

ensemble des moyens humains, matériels et immatériels mis en oeuvre afin de gérer l'information au sein d'une organisation

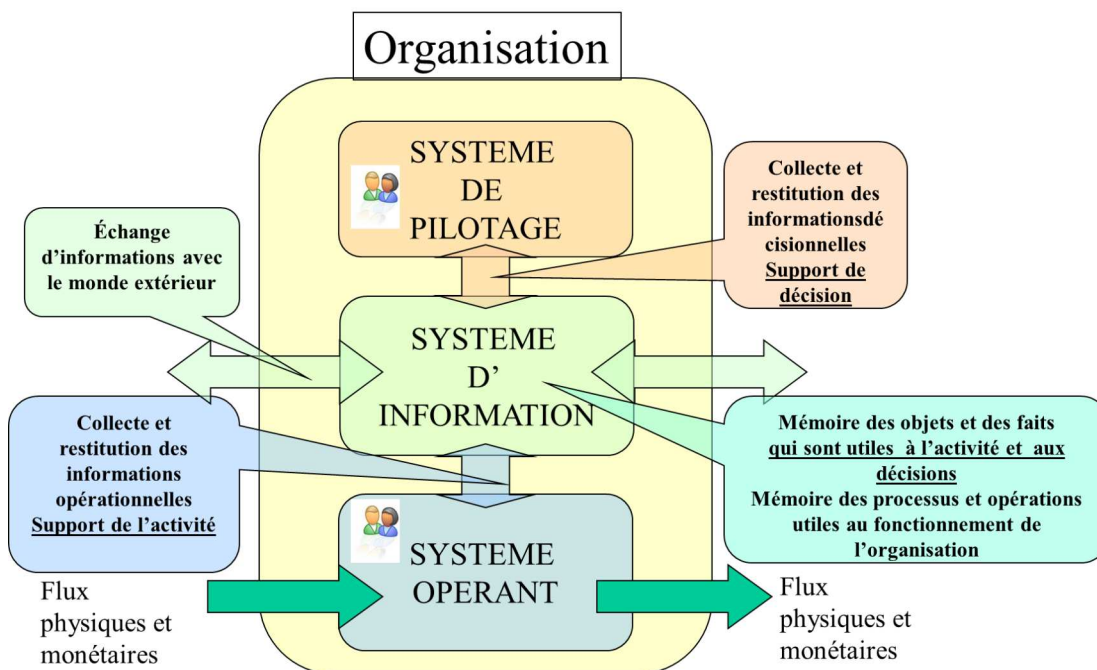
système informatique

ensemble de technologie matérielles et logicielles mises en oeuvre pour garantir le bon fonctionnement du SI

Fonctions du SI relatives aux informations :

- collecter
- mémoriser
- traiter
- diffuser

Le SI représente la mémoire de l'activité de l'organisation.



## Merise : méthode de construction des SI

Merise est une méthode créée dans les années 1970 permettant d'analyser et de concevoir les SI des organisations. Elle décompose l'activité de l'organisation en :

- informations qui décrivent les objets manipulés au cours de l'activité
- traitements qui représentent les opérations effectuées au sein de l'organisation

## Techniques

Merise utilise des techniques (schéma) qui permettent de représenter ces éléments selon 4 niveaux, du plus invariant (conceptuel) au plus sujet à modification (physique) :

	niveau	informations	traitements
Organisation	conceptuel : QUOI ?	MCD	MCT
	organisationnel : QUI ? QUAND ? OU ?	MOD	MOT
Moyens techniques	logique : COMMENT ?	MLD	MLT

physique : COMMENT ?

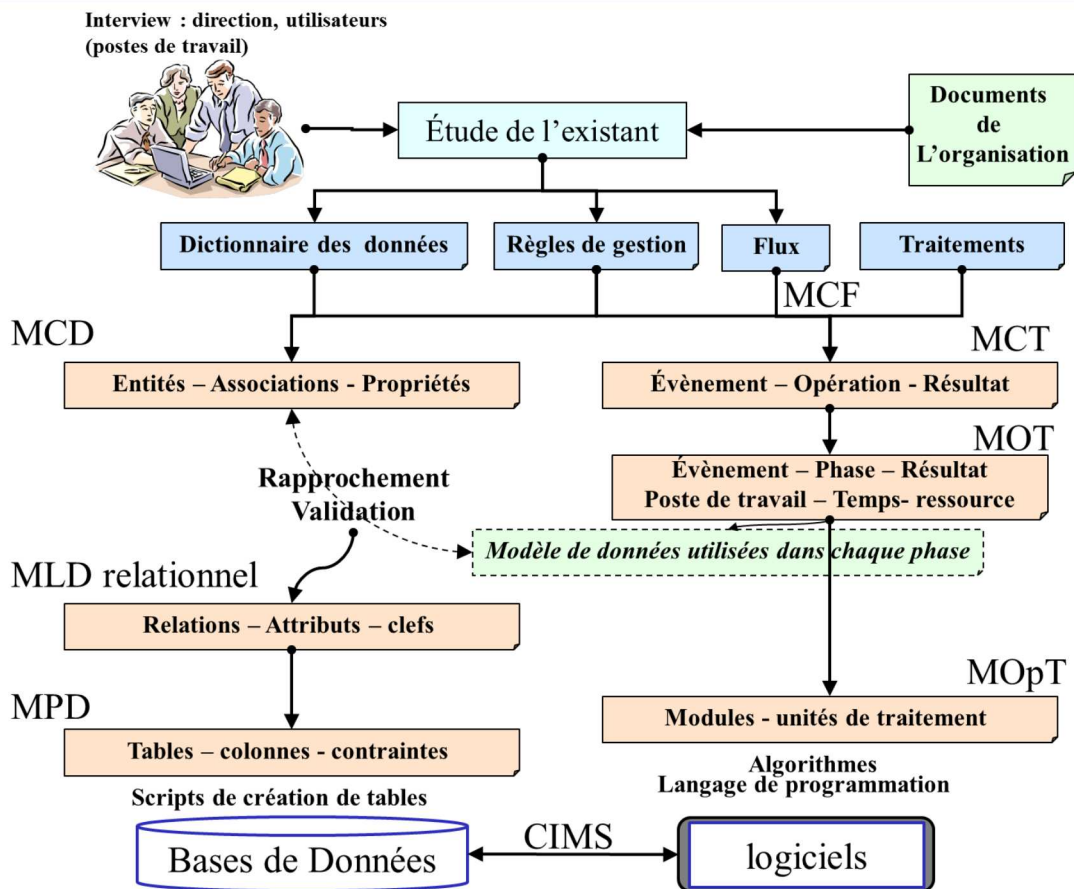
MPD

MOpT

## Démarche générale

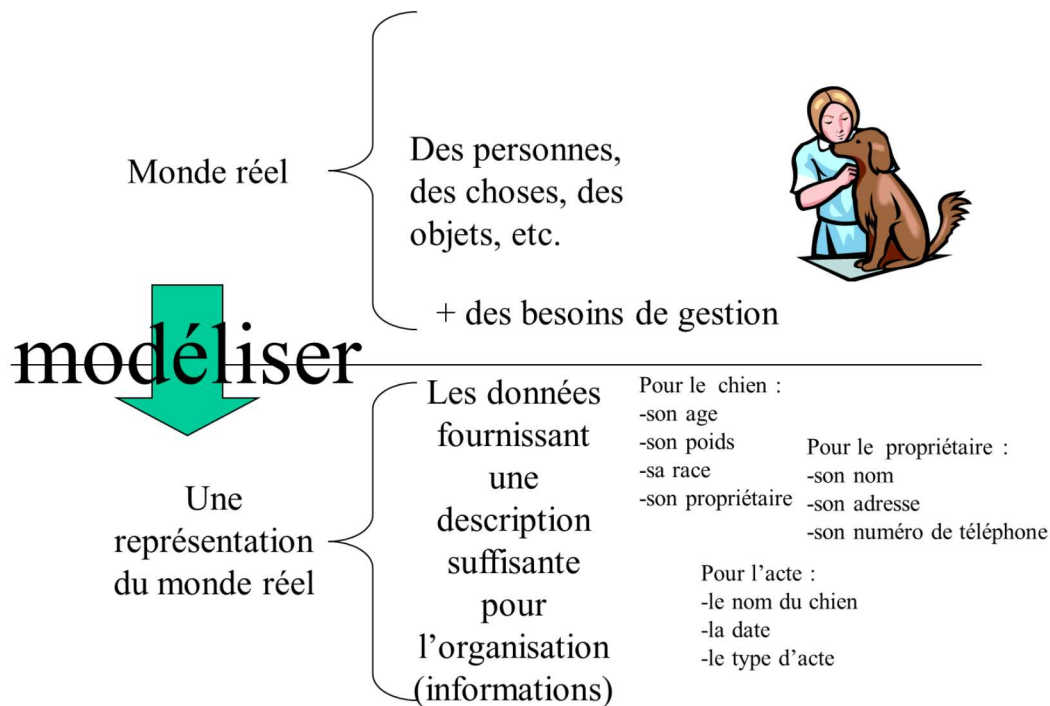
Ces techniques interviennent à certaines étapes de la démarche Merise

-	Etape	Description	Données	Flux et Traitements
Plannification	Schéma directeur	définition des orientations générales	-	-
Conception	Etude préalable	Proposition de solutions d'organisation	MLD - MCD	MOT - MCT - MCF
	Etude détaillée (conception générale, puis conception détaillée par domaine)	spécification utilisateurs complètes du futur système	MLD - MCT	MOT - MCT
Réalisation	Etude technique	spécifications techniques	MLD, MPD	MOpT
	Production logicielle	Ecritures des programmes, création des bases de données, tests	script SQL	Algorithmes et programmes
	Mise en service	installation, formation, chargement des données	-	-
Maintenance		Suivi, corrections, améliorations et évolution	-	-



## Les modèles

Les modèles sont de représentations destinées à représenter la réalité en la simplifiant, de manière à en conserver ce qui est utile. Les représentations sont essentiellement graphiques.



## 2) Modèle Conceptuel de Données (Index)

Objectif : représenter les objets et faits de l'entreprise en terme d'informations manipulées

(+)

- exemple d'objets (permanent) : produit, client, fournisseur, etc.
- exemple de faits (moins permanents) : commande, facture, livraison, etc.

*données*

élément d'information

*dictionnaire de données*

tableau récapitulatif un ensemble de données utiles à l'organisation

*dépendance fonctionnelle*

lien sémantique entre 2 données qui définit qu'une valeur unique de la 1ere détermine une et une seule valeur de la seconde. La 1ere est appelée 'source', la seconde 'but'. On note : *donnee\_source* -> *donnee\_but* (par exemple un numéro d'étudiant détermine de manière précise et unique un seul nom d'étudiant, le nom d'étudiant dépend fonctionnellement du numéro d'étudiant)

### Les étapes

1. recueil des informations : dictionnaire de données et règles de gestion (+)
  - a. étudier l'énoncé : mettre en évidence les données brutes (éléments porteurs d'une valeur : texte, quantité, etc.)
  - b. récupérer les données élémentaires (dans une liste de prénoms, le prénom est la donnée élémentaire, non répétée)
  - c. alimenter le dictionnaire de données et préciser la nature de chaque donnée
  - d. éliminer les synonymes (noms différents mais même signification)
  - e. renommer les polysèmes (nom identique mais significations différentes)
  - f. éliminer les données non pertinentes (pas utiles pour la gestion de l'organisation)
  - g. éliminer les données calculées (à condition d'avoir les données élémentaires permettant de les calculer)
  - h. classer les données alphabétiquement et leur attribuer un numéro d'ordre (nécessite une feuille dans un classeur ou un tableau dans un document texte...)
2. définition des liens entre les données : liste des dépendances fonctionnelles et graphe des dépendances fonctionnelles (ou matrices des dépendances fonctionnelles) (+)
  - a. lister les dépendances fonctionnelles directes (s'assurer que les valeurs des sources des DF aient des valeurs stables, et non pas sujettes à modification : sinon définir des sources stables supplémentaires comme idX où X correspond à l'entité) (+) Par exemple : un numéro de sécurité sociale peut être amené à changer, tout comme le SIREN d'une entreprise, un nom, etc.. Ils peuvent néanmoins faire l'objet d'une identification comme 'identifiant alternatif'
  - b. éliminer les dépendances fonctionnelles transitives (si "P1 -> P2,P3" et "P2 -> P3", alors on peut éliminer P3 de la 1ère)
  - c. s'interroger sur les données non utilisées et décider de leur avenir : soit des paramètres à conserver, soit des données inutiles à éliminer (s'en assurer !)
  - d. représenter le Graphe des Dépendances Fonctionnelles
3. représentation du modèle Conceptuel de Donnée (+)
  - a. encadrer chaque DF source avec ses DF buts (non sources de DF) : on obtient un regroupement des caractéristiques d'une entité
  - b. nommer les entités
  - c. choisir et souligner en trait plein les identifiants (source stables de DF dans chaque entité)
  - d. souligner en trait discontinu les identifiants alternatifs ou candidats (source non stables de DF dans chaque entité)
  - e. les sources de DF vers d'autres entités deviennent des associations qu'on nomme à l'aide d'un verbe
  - f. utiliser les règles de gestion pour définir les cardinalités
  - g. compléter le schéma avec des rôles sur certaines pattes des associations afin d'améliorer la lisibilité
  - h. ce schéma peut être repris de manière normalisée (cf. éléments graphiques du modèle conceptuel)

### Entités

**entité**

objet ayant une existence et un intérêt pour l'organisation

**propriété**

caractéristique décrivant une entité

**identifiant**

une propriété particulière permettant d'identifier une occurrence d'une entité

**identifiant alternatif**

une propriété particulière permettant aussi d'identifier une occurrence d'une entité, mais qui n'est pas choisi comme clef primaire

**occurrence d'entité**

un exemplaire particulier d'entité avec ses propres valeurs de propriétés

**occurrence de propriété**

un exemplaire particulier de propriété avec sa propre valeur

## Associations

---

**association**

relation entre plusieurs entités

**identifiant d'association**

jamais décrit explicitement, il est constitué des identifiants des entités participant à l'association et sa valeur doit être unique

**association binaire**

association entre 2 entités

**association ternaire**

association entre 3 entités (à vérifier)

**association n-aires**

association entre plus de 3 entités : bien s'en assurer...

**occurrence d'association**

un exemplaire particulier d'association, identifié de manière unique par les valeurs des identifiants des entités qui participent à l'association

**propriété**

caractéristique qui participe à la description d'une entité (ou d'une association)

**cardinalités**

quantification du nombre d'occurrences d'une entité pouvant participer à une association (nombre minimal et maximal)

**rôle**

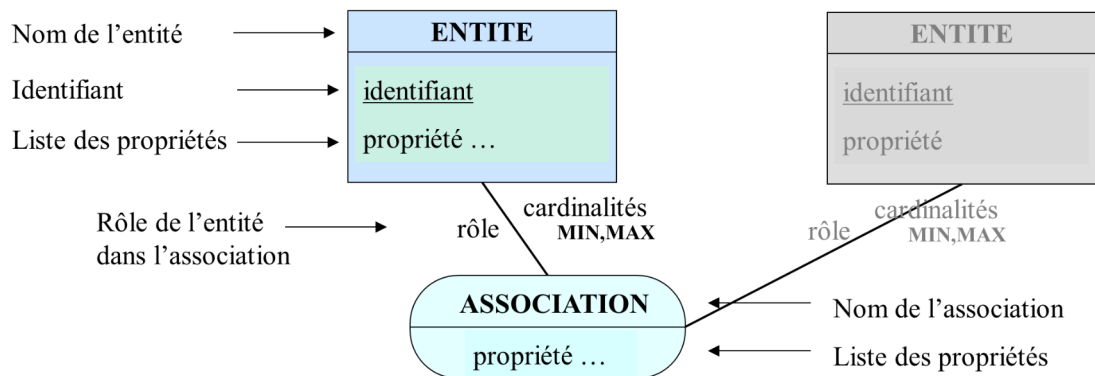
sens donné à une branche (appelée parfois "patte") entre une entité et une association (une association comporte toujours au moins 2 branches)

**CIF**

Contrainte d'Intégrité Fonctionnelle (association hiérarchique, association fonctionnelle) : permet de définir une contrainte d'unicité dans une association, entre une ou plusieurs occurrence d'entités et une occurrence d'une autre entité participant à cette association. Les associations hiérarchiques de cardinalités 1,1-1,N portent par défaut une CIF.

**CIM**

Contrainte d'Intégrité Multiple (association non fonctionnelle) : définit une association avec des cardinalités maximales à N sur chacune de ses branches



## Cas particuliers (+)

---

### • Gestion du temps et entités temporelles

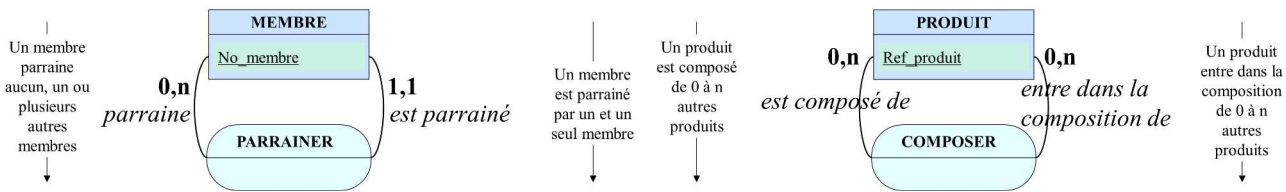
**(+)**

La gestion du temps peut présenter plusieurs formes :

- en tant que propriété élémentaire si elle est simplement cible d'une dépendance fonctionnelle directe (pour un véhicule, date de la prochaine révision)
- en tant qu'entité si elle est source d'une dépendance fonctionnelle directe (date -> heureLeverSoleil, heureCoucherSoleil)
- en tant qu'entité si elle participe en tant que source à une dépendance fonctionnelle (association non fonctionnelle : numCde, refProd, dateLiv -> quantité) ; on a à faire à une association ternaire à vérifier (cf. plus bas)

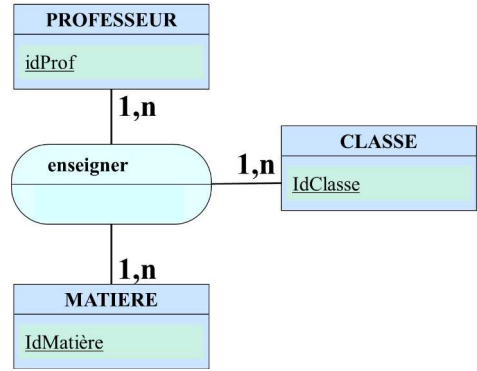
### • Association réflexive

**(+)** Une association réflexive est une association qui relie une entité avec elle-même. L'utilisation de la notion de 'rôle' permet de préciser le sens de chacune des branches (pattes) de l'association.

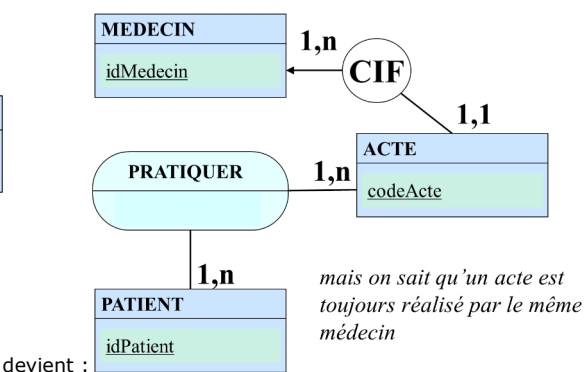
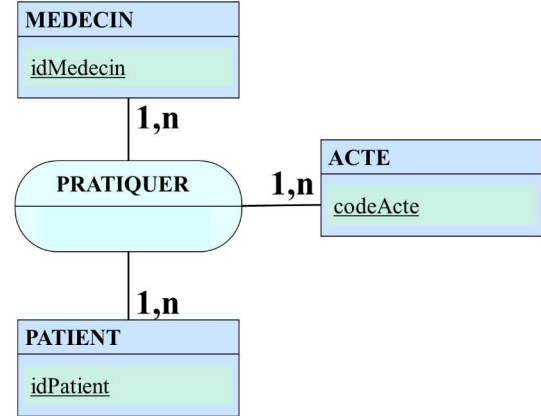


**• Association ternaire et n-aires**

(+) Une association ternaire relie 3 entités (au delà, on parle de 'n-aires') : il est indispensable de s'assurer de sa validité en vérifiant qu'aucune dépendance fonctionnelle entre les entités (identifiant) n'existe, c'est à dire que idEnt1,idEnt2 -X-> idEnt3, que idEnt1,idEnt3 -X-> idEnt2, et que idEnt2,idEnt3 -X-> idEnt1 (avec '-X->' signifiant 'ne détermine pas')  
 Pour un professeur et une matière, on peut avoir plusieurs classes, Pour une classe et une matière, on peut avoir plusieurs professeurs (c'est le cas ici, d'enseignant pour CM, TD ou TP), Pour une classe et un professeur, on peut avoir plusieurs matières (c'est le cas ici, où un enseignant intervient en mathématiques et en physique)

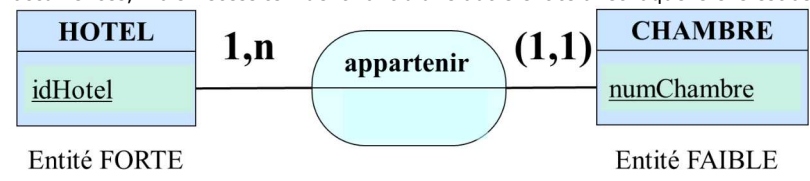


Pour un medecin et un type d'acte, on peut avoir plusieurs patients, Pour un medecin et un patient, on peut avcoir plusieurs type d'actes, Pour un patient et un type d'acte, on peut avoir plusieurs medecin...mais...on vient d'apprendre qu'en fait, un type d'acte n'est réalisé que par un et un seul medecin !



**• Entité faible**

(+) Une entité faible, ou dépendante, possède un identifiant qui n'est pas suffisant pour identifier de manière unique chacune des occurrences, mais nécessite l'identifiant d'une autre entité avec laquelle elle est associée. L'identifiant est dit 'relatif',



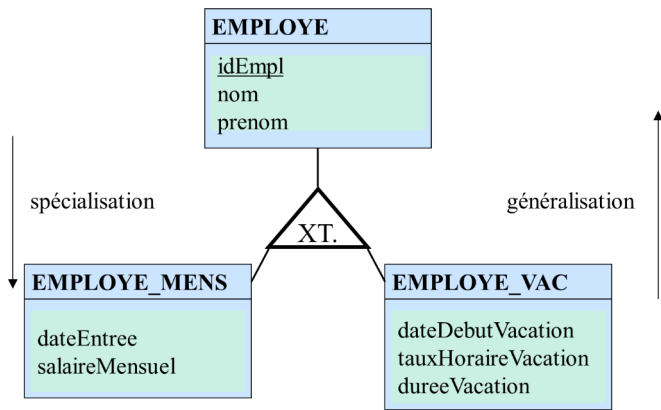
**• Héritage et spécialisation**

(+)  
 Si toutes les occurrences d'une entité-type ne possèdent pas toujours les mêmes propriétés ou ne sont pas liées aux mêmes associations, l'analyse peut montrer qu'on a affaire à

- o d'une part, une entité générique qui regroupe les propriétés communes à toutes les occurrences, et qui est reliée aux associations communes,
- o et d'autre part à des entités spécialisées qui comporteront leurs propres propriétés et seront reliées à leurs propres associations. Elles héritent également des propriétés de l'entité générique : on est en effet dans une forme de relation hiérarchique (sans association)

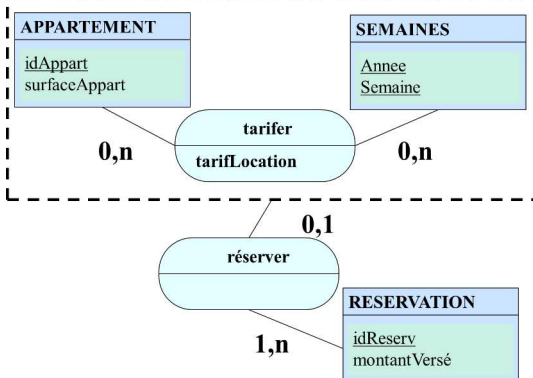
Pour qualifier la contrainte associée à cette spécialisation :

- o **X** : pour eXclusion, une occurrence de l'entité générique ne peut être spécialisée que d'une seule forme ou n'être pas spécialisée
- o **T** : pour Totalité, une occurrence de l'entité générique est spécialisée d'une au moins des formes proposées
- o **XT** (ou +) : pour eXclusion et Totalité, une occurrence de l'entité générique est spécialisée d'une seule des formes proposées
- o sinon pas de contrainte: des occurrences peuvent être spécialisées d'une seule ou plusieurs formes ou n'être pas spécialisées



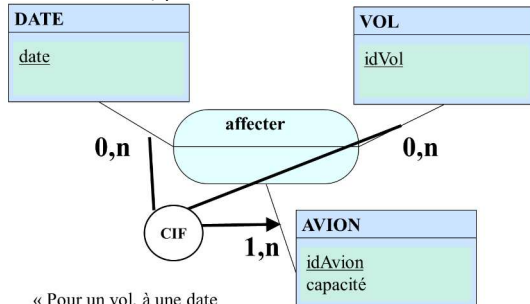
### • Agrégats ou pseudo-entité

(+) Un agrégat est un regroupement d'entités et d'associations qui peut être associé à une autre entité (exemple de réservations de chambre - ou de tout autre objet - pour une période donnée)



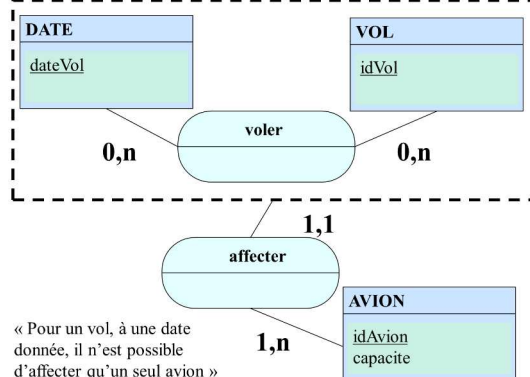
### • Contrainte d'Intégrité Fonctionnelle sur associations ternaires

(+) Dans le cas d'une association ternaire mettant en oeuvre les entités A, B et C, et si on a  $idA, idB \rightarrow idC$ , signifie que, dans le cadre de l'association, pour une occurrence de A et une occurrence de B, on a une et une seule occurrence de C.



« Pour un vol, à une date donnée, il n'est possible d'affecter qu'un seul avion »

On a donc ici une fausse ternaire. Un agrégat apporte une réponse au modèle :

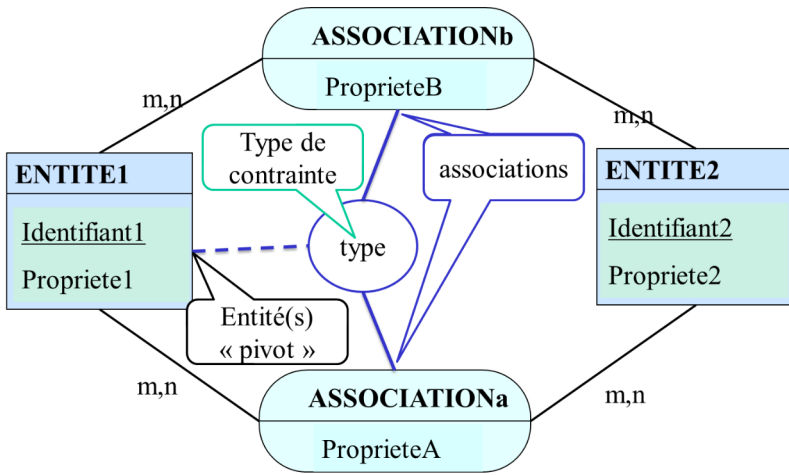


« Pour un vol, à une date donnée, il n'est possible d'affecter qu'un seul avion »

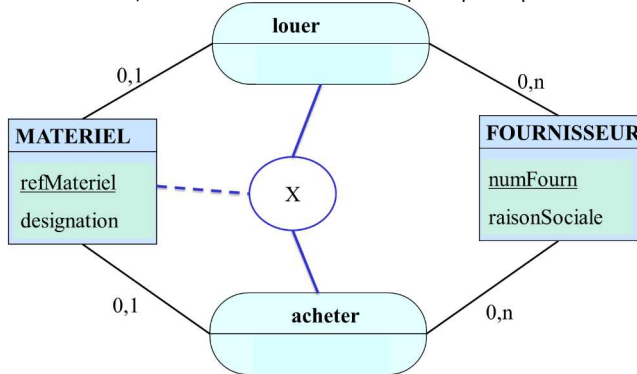
### • contraintes ensemblistes inter-associations

(+) Permettent d'apporter des précisions relatives aux occurrences d'entités Pour qualifier la contrainte associée à cette spécialisation :



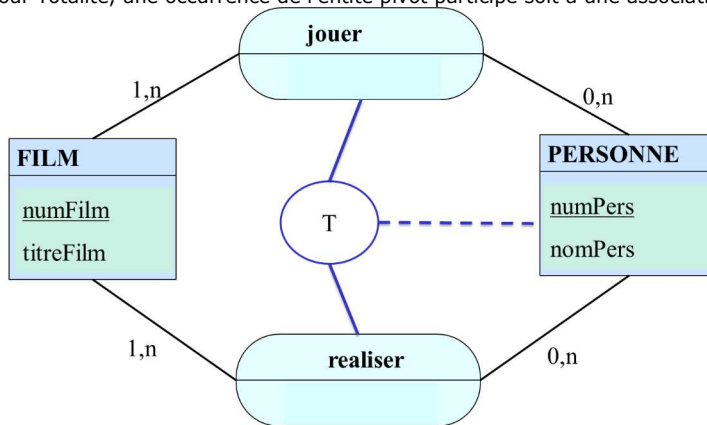


o **X** : pour eXclusion, une occurrence de l'entité pivot participe soit à une association soit à l'autre, soit à aucune des 2



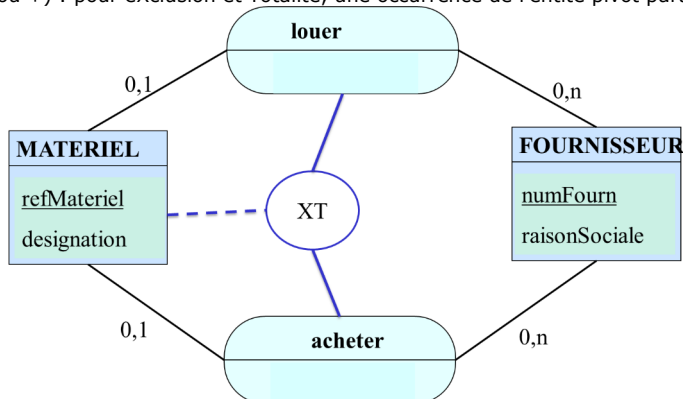
toute occurrence de MATERIEL peut participer à louer ou acheter, ou à aucune des 2  
 = un matériel peut être loué ou acheté (ou avoir été donné, cas non géré)

o **T** : pour Totalité, une occurrence de l'entité pivot participe soit à une association soit à l'autre ou au 2



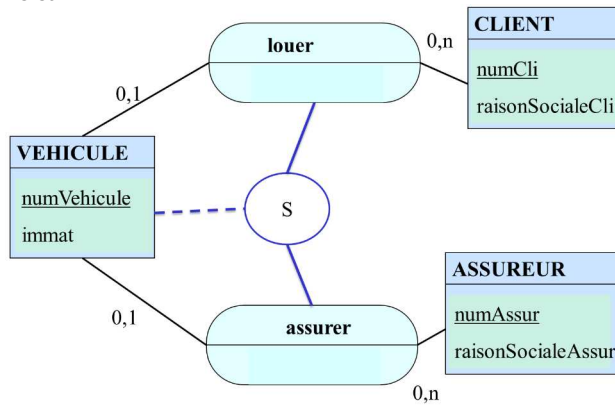
toute occurrence de PERSONNE participe à au moins jouer ou realiser  
 = une personne est soit acteur ou réalisateur (peut-être les 2)

o **XT** (ou +) : pour eXclusion et Totalité, une occurrence de l'entité pivot participe soit à une association soit à l'autre, pas au 2



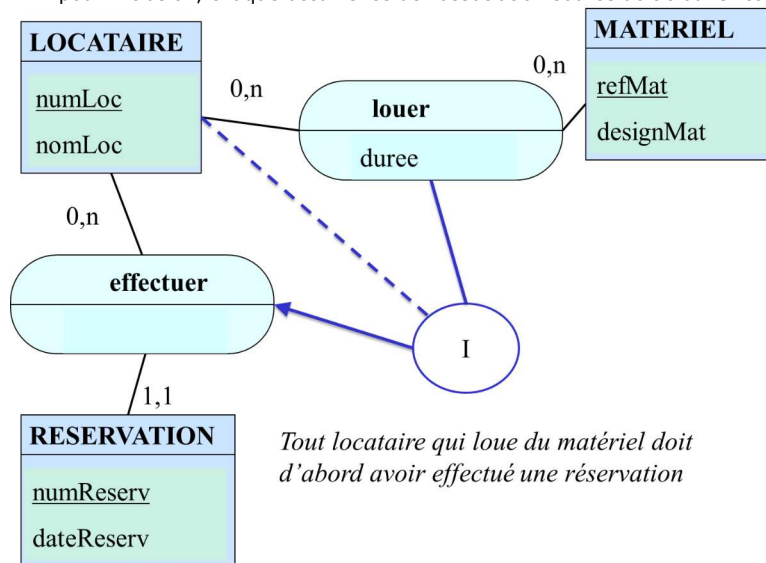
toute occurrence de MATERIEL participant à louer ne peut participer à acheter  
 = un matériel est soit loué soit acheté (pas les 2)

- **S** (ou =) : pour simultanéité, chaque occurrence de l'association source doit trouver correspondance dans l'association cible et vice-versa



*toute occurrence de VEHICULE participant à louer doit également participer à assurer  
= un véhicule loué est aussi assuré*

- **I** : pour Inclusion, chaque occurrence de l'association source doit trouver correspondance dans l'association cible



*Tout locataire qui loue du matériel doit d'abord avoir effectué une réservation*

- sinon pas de contrainte

### 3) Modèle Logique de Données relationnel (Index)

Objectif : choisir un mode d'organisation des données

#### Les étapes

1. application des règles de transformation du MCD en MLDr
2. vérifier l'application des formes normales

#### Le MLD relationnel

**Table** (clef primaire, attribut, ..., clef candidate, #clef étrangère, ...)

où :

- 'Table' est le nom de la table (ou relation)
- la liste des attributs de la table est entre parenthèses
- l'attribut clef primaire est souligné
- l'attribut clef candidate est souligné en pointillés
- les attributs clefs étrangères sont préfixés (ou parfois suffixés) par # (ou bien on indique explicitement l'attribut clef primaire auquel ils font référence)

#### Les règles

Des règles de passage permettent la transformation mécanique du MCD au MLDr. Elles sont à appliquer strictement afin de ne perdre aucune information :

1. entité : toute entité devient une table, son identifiant devient clef primaire de la table, les propriétés deviennent les attributs de la table, les identifiants alternatifs deviennent clefs candidates
2. associations cardinales 1,1 - 1,N : la table côté 1,1 reçoit l'identifiant côté 1,N comme attribut clef étrangère



3. associations cardi **1,N - 1,N** : une nouvelle table est créée, sa clef primaire est constituée des identifiant des tables participantes (on a donc une clef primaire unique, mais composée de plusieurs attributs), ces attributs sont clefs étrangères vers les tables d'origine, les propriétés sont récupérées
4. associations cardi (**1,1**) - **1,N** : la table côté (1,1) provient d'une entité faible et la clef étrangère reçue de la règle 1,1-1,N participe également à la clef primaire.
5. Autres règles (+)
  - o associations cardi **0,1 - 1,N** : 2 solutions
    - soit : règle du 1,1 - 1,N
    - soit : une nouvelle table est créée, sa clef primaire est constituée par l'identifiant côté 0,1, l'attribut correspondant est clef étrangère, et elle a comme attribut l'identifiant côté 1,N (les propriétés, si certaines sont définies, sont conservées)
  - o entités filles (spécialisation) : la table récupère l'identifiant de l'entité mère comme clef primaire; cet attribut est également clef étrangère vers la table d'origine
  - o pseudo-entités : traitement classique par les règles précédentes, la pseudo-entité doit être considérée comme un entité à part entière ayant comme identifiant, celui de l'association centrale (id. des entités qui participent)

## L'algèbre relationnelle

### 4) Modèle Physique de Données (Index)

Objectif : choisir un type de bases de données

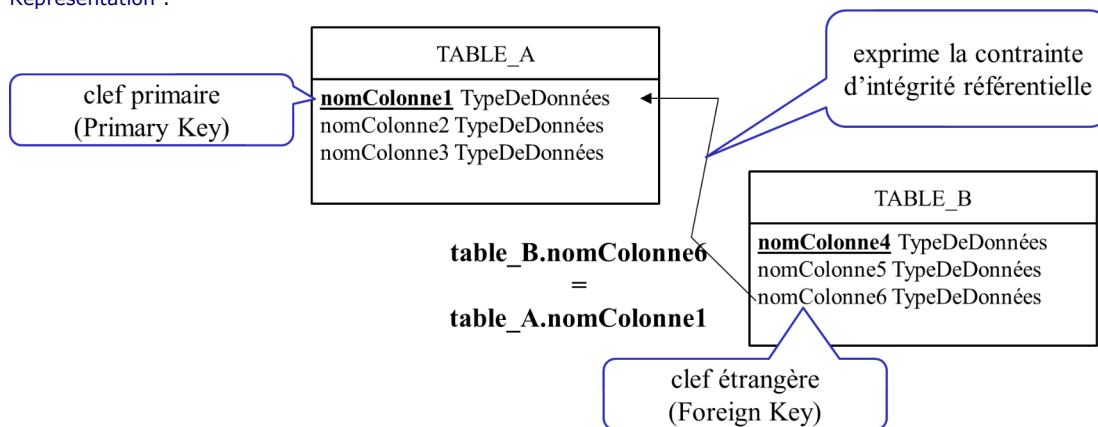
1. utilisation personnelle : Microsoft Access, OpenOffice Base
2. utilisation d'un groupe de travail : MySQL, PostgreSQL, Oracle, IBM DB2, Microsoft SQL Server

Le modèle physique reprend les tables et met en évidence les contraintes d'intégrité d'identité (clef primaires) et référentielles (clefs étrangères vers clefs primaires) ainsi que la nature des colonnes des tables:

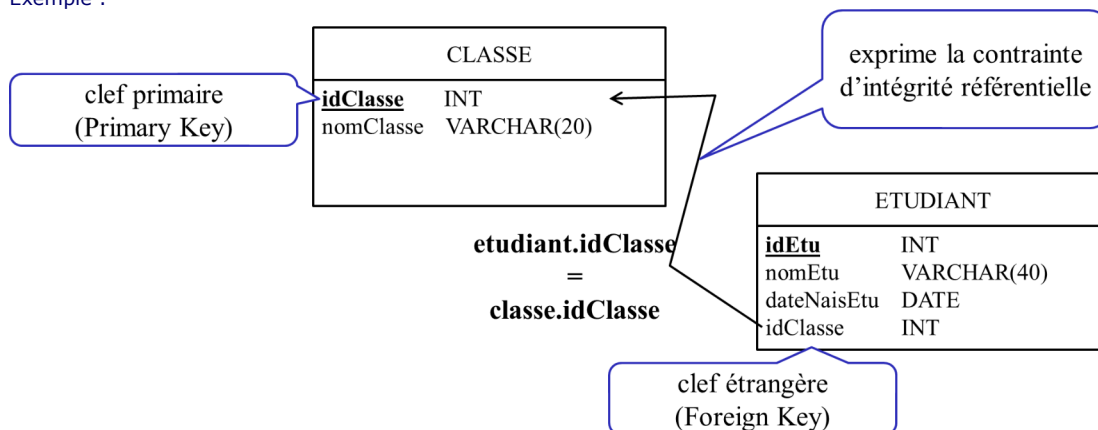
1. tables et colonnes avec leurs natures
2. contraintes d'intégrité (d'identité : clef primaire, référentielle : clef étrangère vers clef primaire)

Les index peuvent également être indiqués (en conservant, par exemple, le souligné pointillés)

Représentation :



Exemple :



### 5) Bases de données (Index)

Objectif : implanter la base de données

1. utilisation d'interface graphique et du langage QBE : Microsoft Access, OpenOffice Base
2. utilisation du langage SQL : MySQL, PostgreSQL, Oracle, IBM DB2, Microsoft SQL Server

Opérations à réaliser :

1. créer les tables : colonnes et leurs natures, contrainte d'intégrité d'identité
2. définir les contraintes d'intégrités référentielles
3. mettre en oeuvre les autres contraintes définies dans le MCD : utilisation des déclencheurs (triggers)

4. insérer des données à partir de fichiers existants (les données sources ont dû déjà être préparées afin d'être intégrées dans les tables de manière cohérente)
5. vérifications, contrôles croisés
6. Autres opérations importantes : configuration du SGBD et optimisation/sécurité (répartition des fichiers sur différents disques, etc.), mise en place et test des sauvegardes

Développement d'applications :

1. définir des formulaires pour Créer, Interroger, Mettre à jour et Supprimer
2. définir des requêtes pour essentiellement Interroger
3. définir des rapports pour présenter les résultats