

---

# **Le modèle Conceptuel et Logique de Données (MCD & LMD) Merise**

**Présenté par  
G. Mopolo-Moké  
prof. PAST UNSA  
2018 / 2019**

# Plan Général

---

## □ Plan

- Les modèles de MERISE : les 3 niveaux d'abstractions
- Les modèles de MERISE
- MERISE : transformations des différents modèles de DONNEES
- M.C.D. MERISE
- M.C.D. : le formalisme entité-association
- M.C.D. : conception
- M.C.D. : construction de la liste des informations du domaine
- Les propriétés : 1<sup>e</sup> concept du formalisme « entité-association »
- M.C.D. : L'entité
- M.C.D. : L'entité & identifiant
- M.C.D. : relation (ou association)
- M.C.D. : exemple d'association
- M.C.D. : aide-mémoire
- M.C.D. : cardinalités d'une entité dans une association

# Plan Général

---

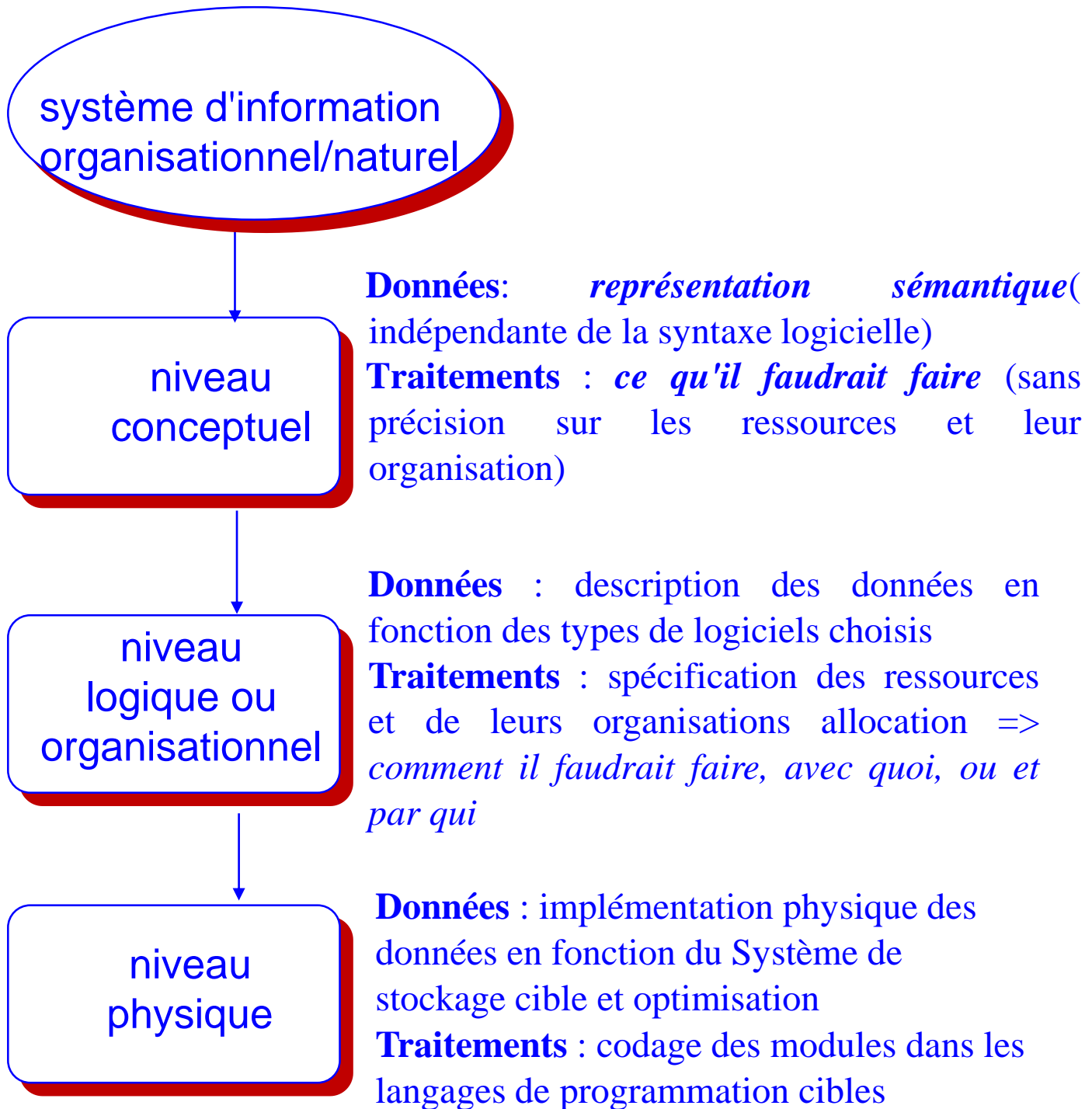
## □ Plan

- M.C.D. :valeurs typiques des cardinalités
- M.L.D. : les objectifs
- M.L.D. : les modèles logiques
- Transformation du formalisme entité-relation en formalisme relationnel
- Objectifs des S.G.B.D.

# Les modèles de MERISE :

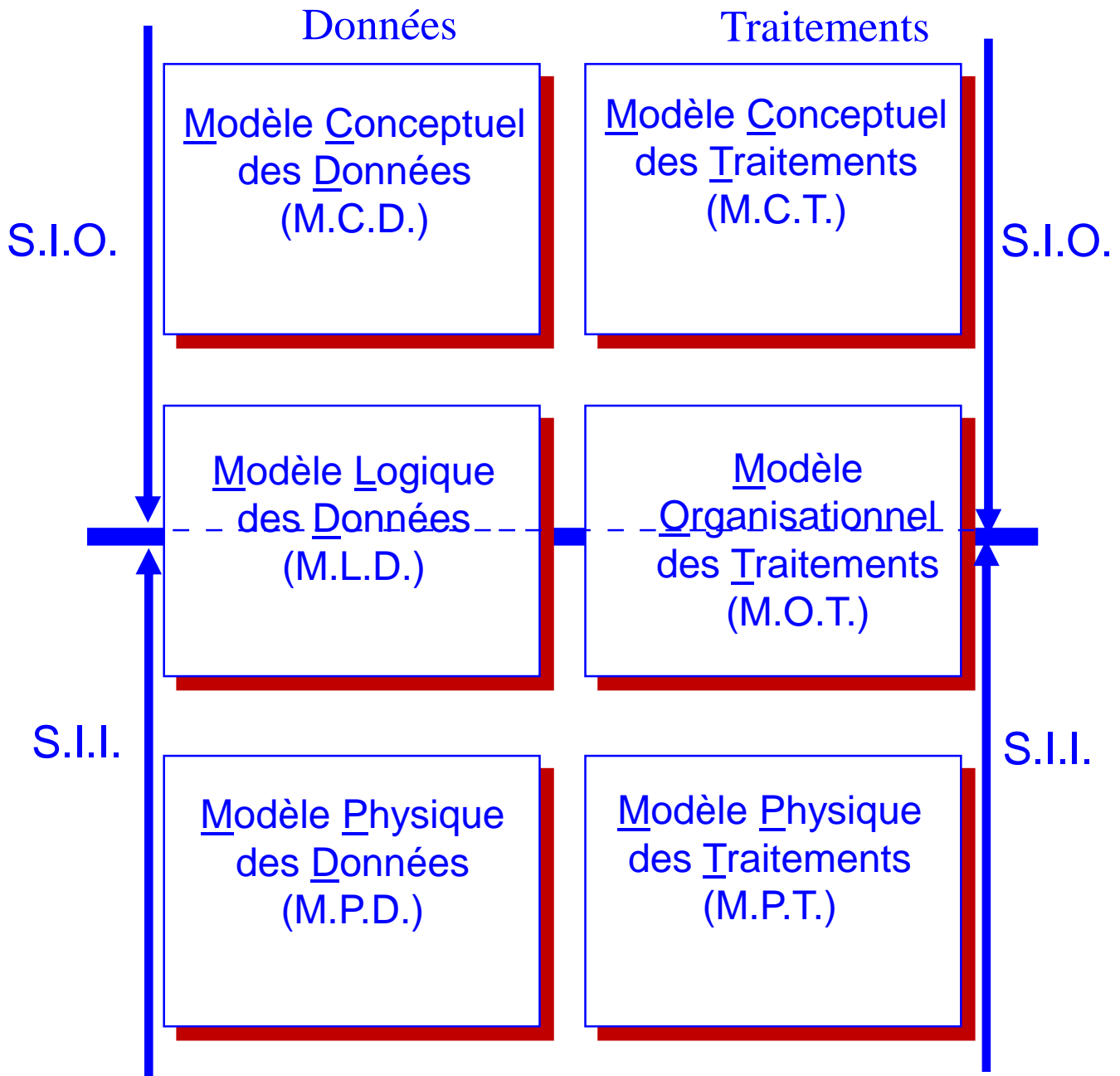
## les 3 niveaux d'abstractions

---



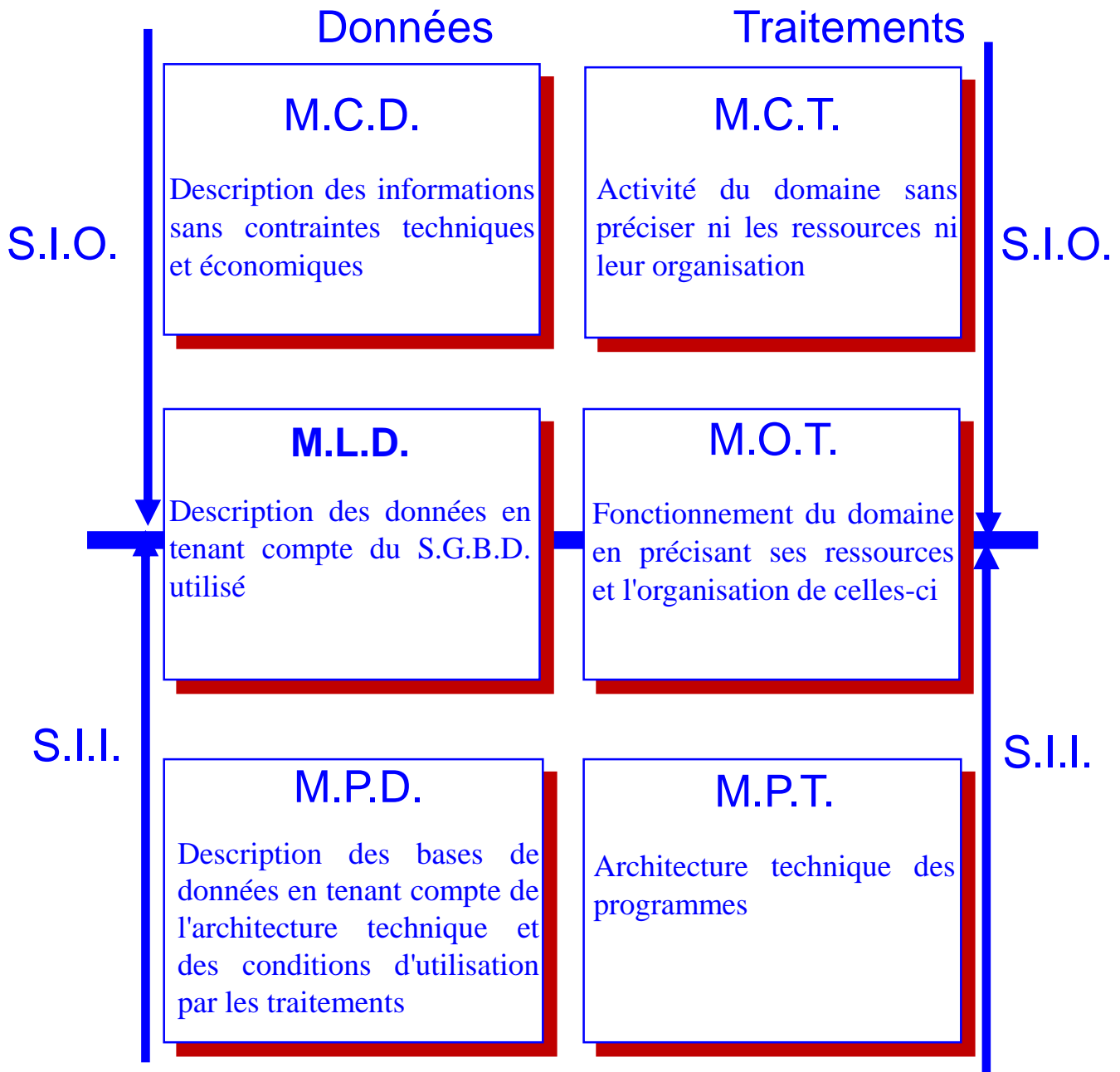
# Les modèles de MERISE

---



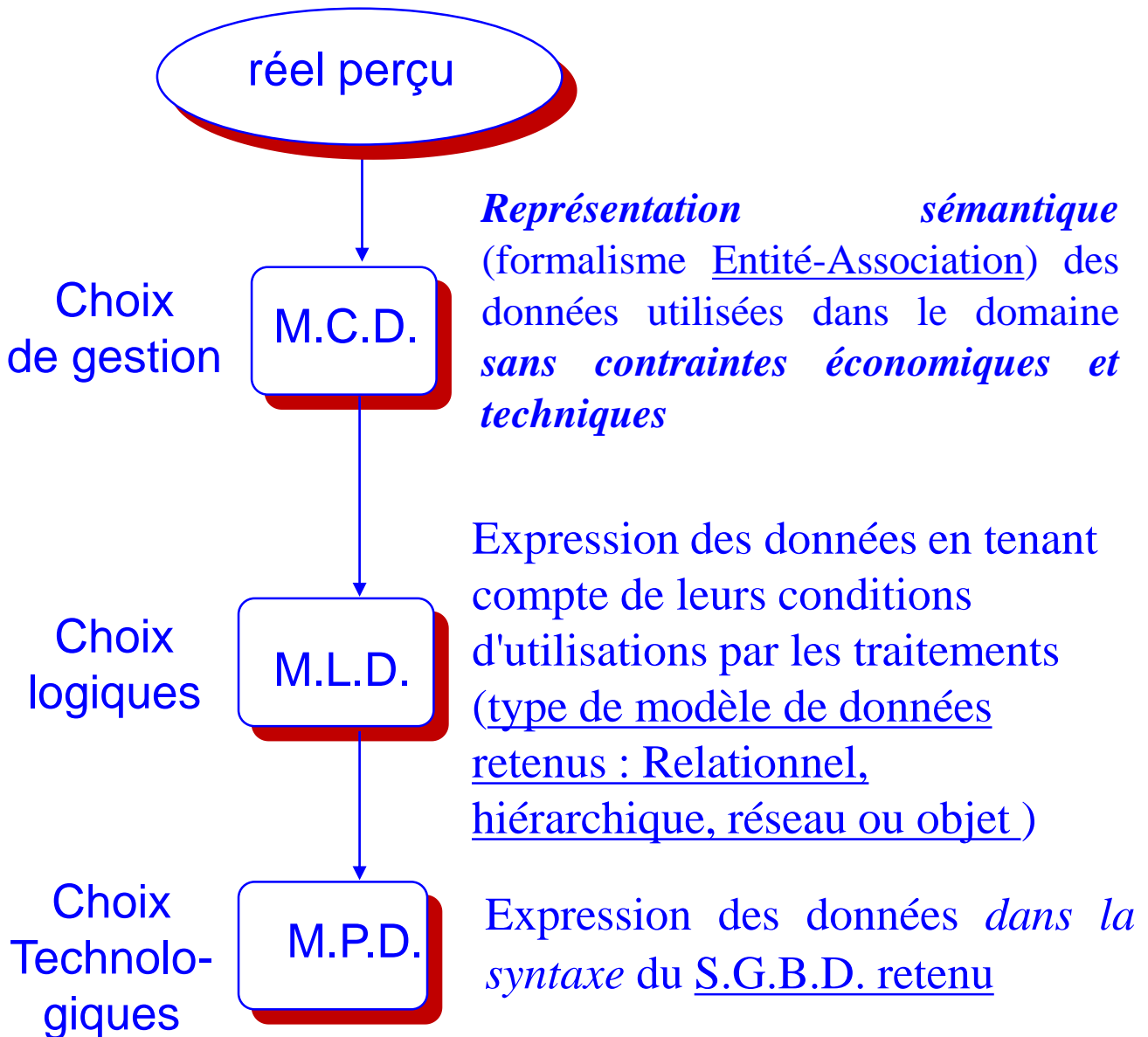
# Les modèles de MERISE

---



# MERISE : transformations des différents modèles de DONNEES

---



- La transformation d'un niveau vers un autre peut se faire Manuellement ou via les logiciels (Power AMC, Architecte, ...) qui implémentent ces modèles.
- La transformation de bas en haut s'appelle le reverse Engineering

# M.C.D. MERISE

---

□ **M.C.D : Modèle Conceptuel de Données Merise**

□ **Représente l'ensemble des données mémorisables du domaine et leur organisation sémantique**

□ **Sans y intégrer des considérations économiques et techniques (conditions d'utilisation, optimisation, ...)**

□ **Le formalisme de description des données et le formalisme Entité / Association**



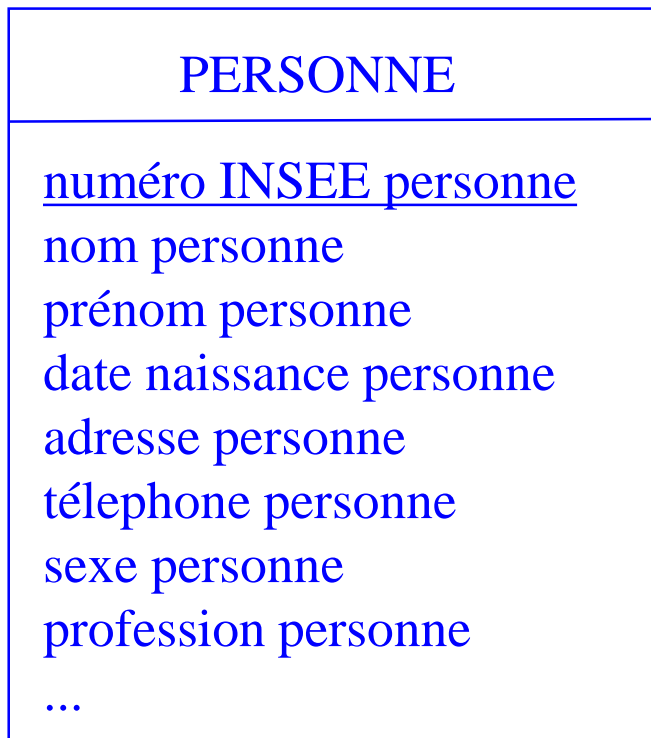
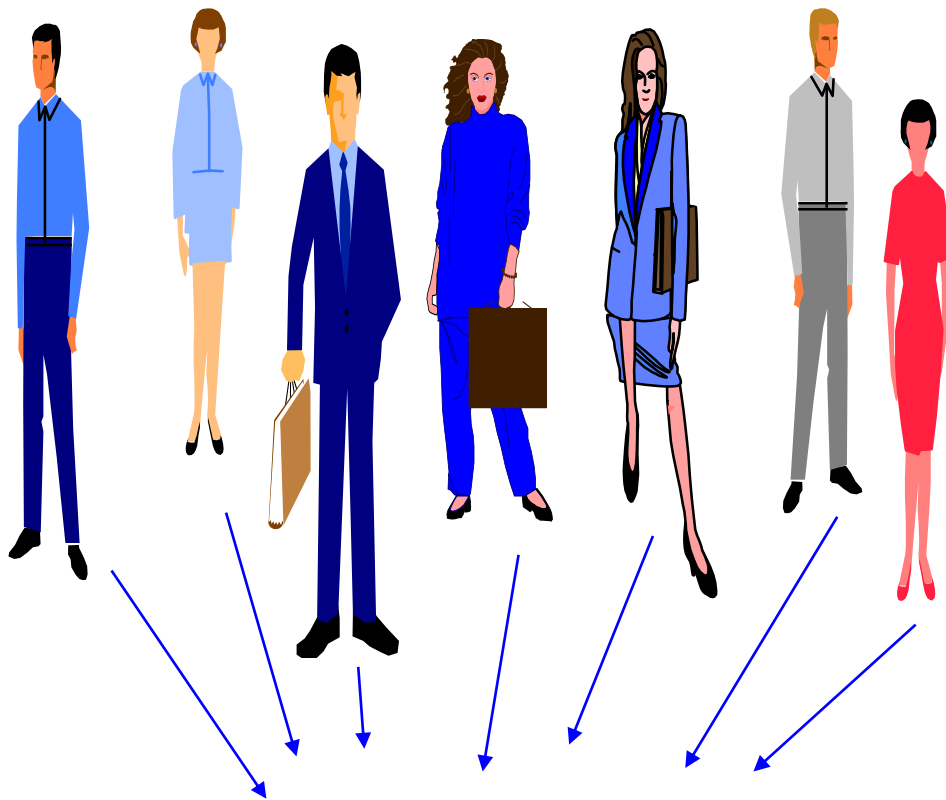
# M.C.D. : le formalisme entité-association

---

□ Les données et leur organisation sémantique sont représentées avec le formalisme «*Entité-Association*» ou «*Entité-Relation* » à partir de 4 concepts :

- L'entité ou *entité type* ou *individu type*
- L'association ou *association type* ou *relation type*
- Les propriétés ou *propriétés type* (dont les **identifiants**)
- Les cardinalités caractérisant les associations

# M.C.D. : le formalisme entité-association



**Entité**  
**Personne** ou  
Individu type  
(ou entité type)  
Personne

# M.C.D. : conception

---

□ La conception d'un MCD est faite à partir :

- Des entretiens avec les utilisateurs et les gestionnaires
- De l'étude des documents utilisés dans le domaine

□ Vous devez pour cela :

- Créer la liste des informations utilisées par l'ensemble des acteurs du domaine
- Déterminer l'origine (document, fichier, ...) et les utilisations des informations
- Identifier les Entités : *concepts*, les *classes*, les *objets* (naturels ou artificiels) et les relations les associant

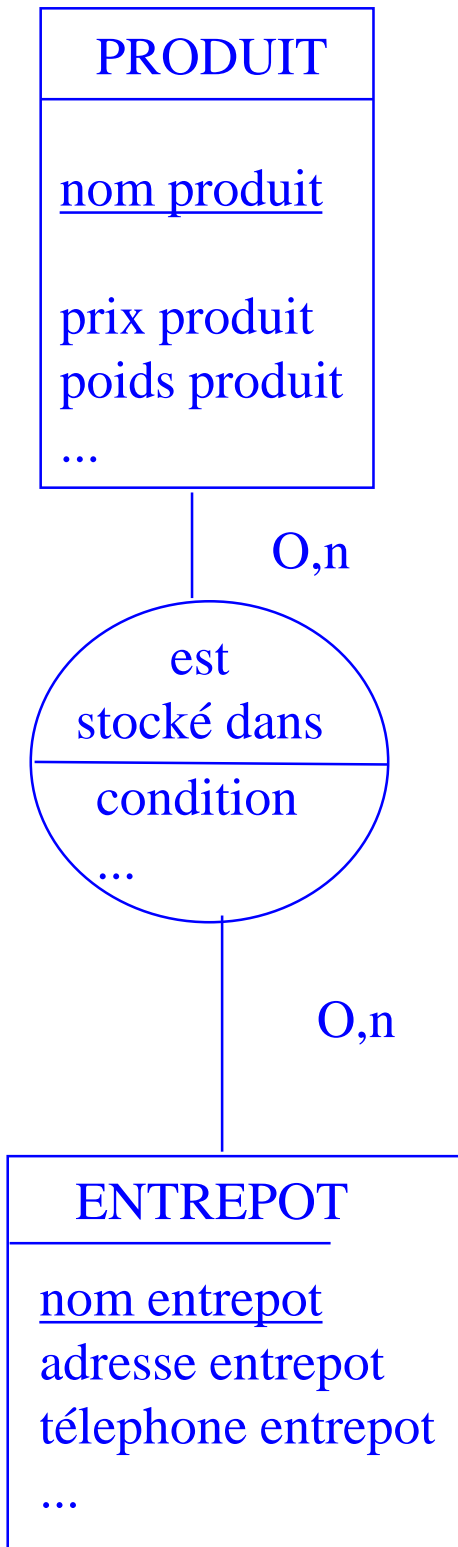
# M.C.D. : construction de la liste des informations du domaine

---

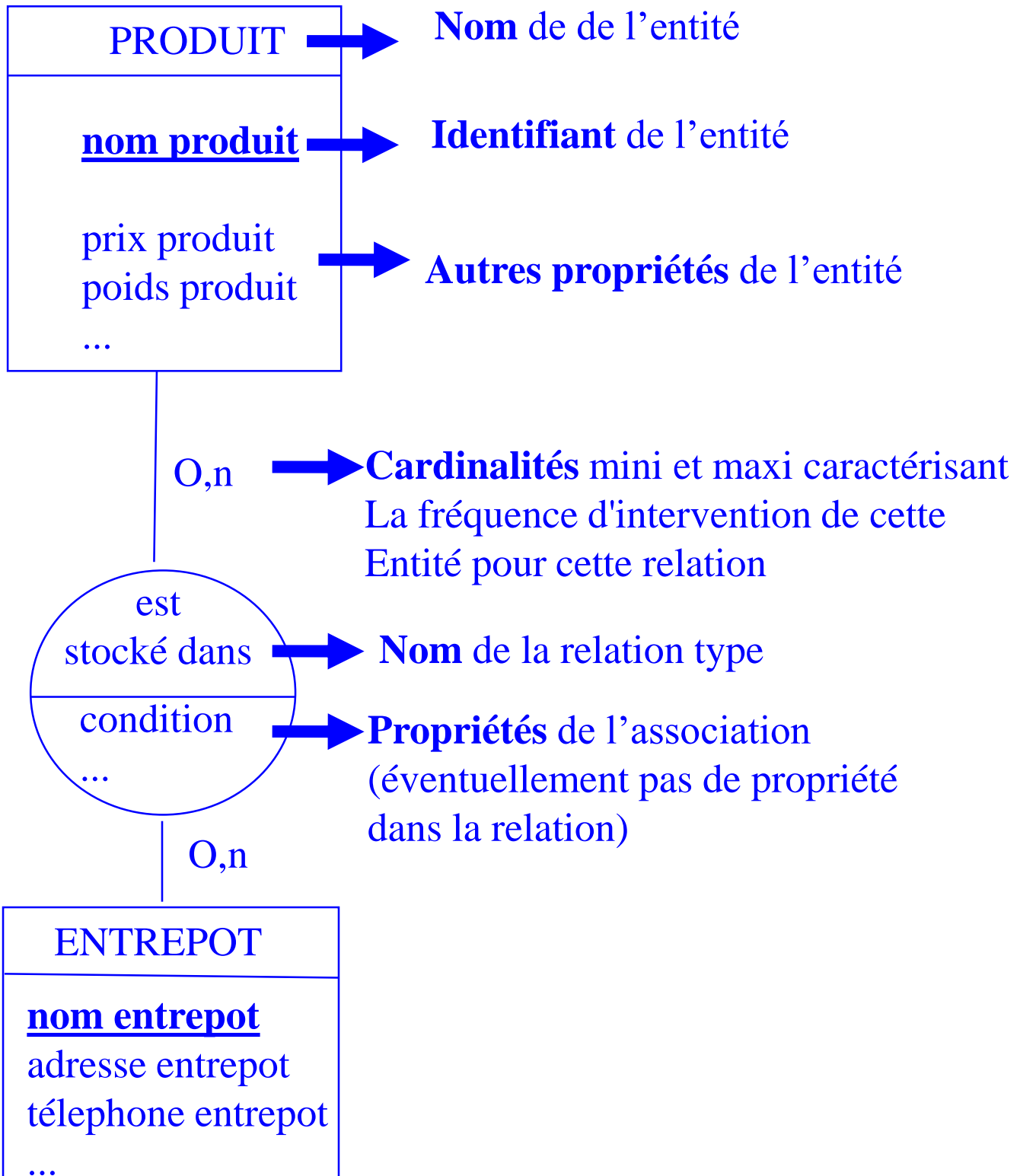
- Vérifier pour chaque informations qu'elle n'est pas *déjà répertoriée* dans la liste (sous une autre appellation *synonyme*)
  
- Vérifier qu'une même appellation ne désigne pas des informations différentes (*homonymie*)
  
- *Confronter* la liste obtenue avec les acteurs du domaine pour obtenir le dictionnaire des données (liste épurée des propriétés)

# M.C.D. : construction de la liste des informations du domaine

---



# M.C.D. : construction de la liste des informations du domaine



# Les propriétés : 1<sup>e</sup> concept du formalisme "entité-association"

---

- **Modélise une information élémentaire** utilisée dans le domaine
- **Plus petit élément d'information** ayant une signification
- **Représente le type de l'information et non pas la valeur de l'information**
- **Obligatoirement rattachée à une et une seule entité ou (exclusivement) *une et une seule association*** qu'elle décrit
- **Une propriété doit être unique** par son nom et par son sens (pas de synonyme et d'homonyme)

## M.C.D. : L'entité

---

- Une entité représente un *ensemble* (une classe) *d'objets* (concrets ou abstraits) *de même nature* dont les éléments sont appelés *occurrences* d'entités
  
- On représente *les entités* et non leurs *occurrences*



## M.C.D. : L'entité

---

- Une entité est identifiée dans un interview, dans un document, etc.
- Toutes les occurrences d'une entité peuvent être *complètement décrites avec les seules propriétés de l'individu type*
- Pour toutes les occurrences d'une entité, il ne peut avoir à un instant donnée *qu'au plus une valeur pour chacune des ses propriétés*

# M.C.D. : Entité et identifiant

---

- Une entité possède obligatoirement une *propriété permettant d'identifier chacune de ses occurrences*. Elle est appelée *identifiant de l'entité* (à une valeur de l'identifiant correspond une occurrence de l'entité et vice versa)
- *L'identifiant doit être stable* : sa valeur ne peut changer de la création de l'entité jusqu'à sa destruction
- **L'identifiant peut être :**
  - Une propriété naturelle (ex. le nom d'un pays pour l'individu pays)
  - Une propriété artificielle inventée par le concepteur du S.I. pour identifier un individu (ex. numéro client pour identifier les occurrences de l'individu CLIENT)
  - Une propriété composée mais minimale (ex. nom + prénom + date, heure, lieu de naissance)

# M.C.D. : Entité et identifiant

---

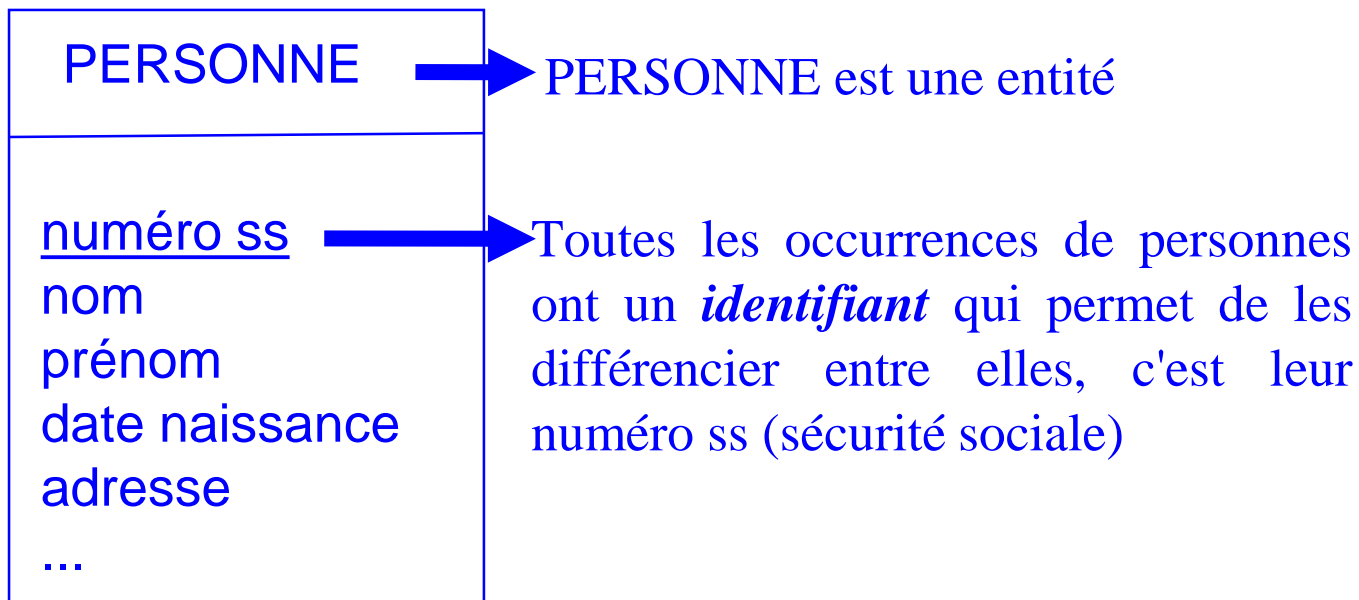


# M.C.D. : Entité et identifiant

---

## Exemple d'entité avec son identifiant

### ENTITE PERSONNE



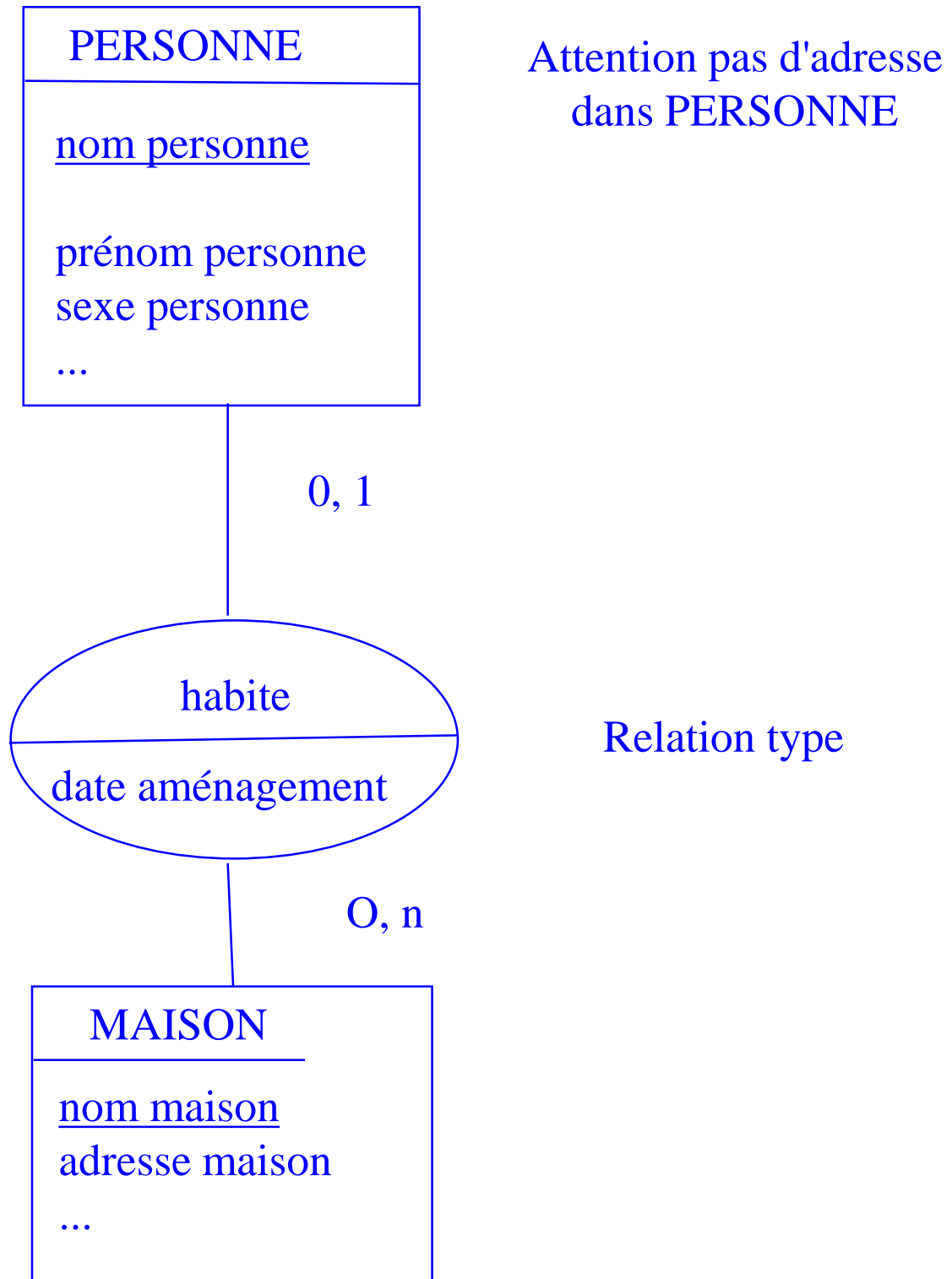
## **M.C.D. : Entité et relation (ou association)**

---

**□ Information(s) caractérisant un lien sémantique entre au moins 2 occurrences d'entités**

# M.C.D. : relation (ou association)

---



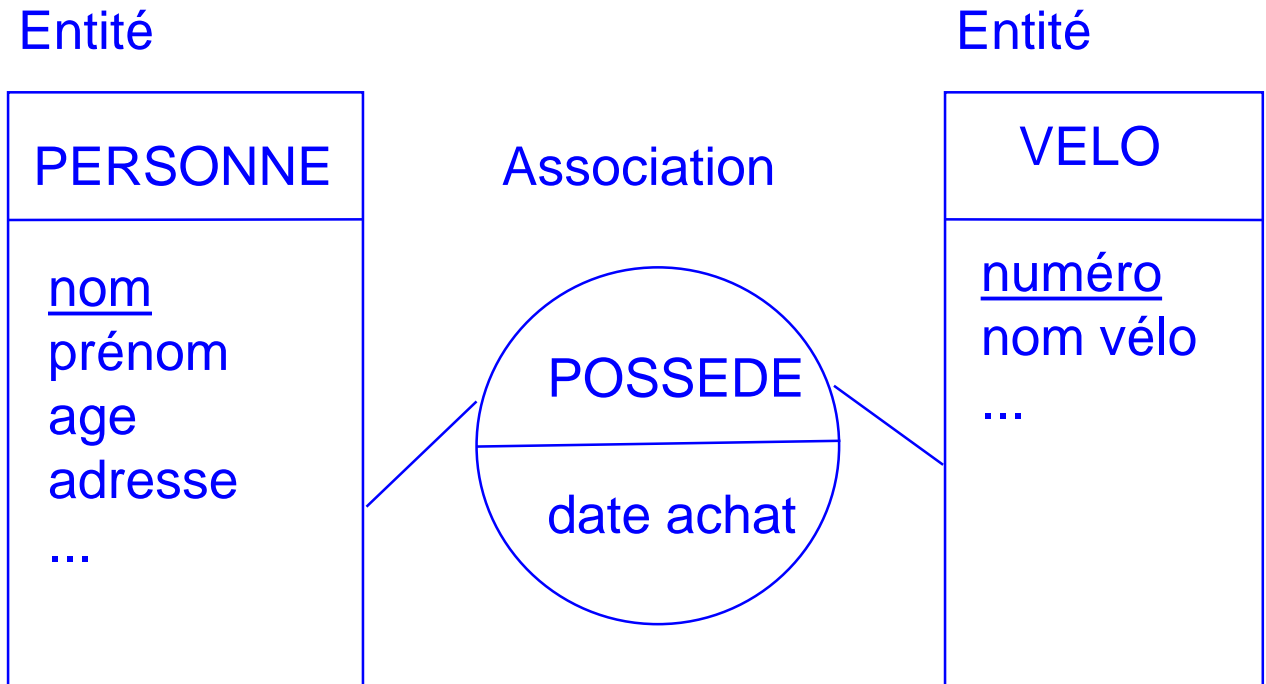
## M.C.D. : la relation ou association

---

- La *collection* d'une relation est l'ensemble des individus participant à la relation
- La *dimension* d'une relation type est le nombre d'individus participant à la relation
- La *dimension* d'une relation est *stable* pour toutes ses occurrences
- Une occurrence de relation ne possède *pas d'identifiant propre*, elle est identifiable par les identifiants des occurrences des individus de sa collection
- Pour toutes occurrences d'une relation il ne peut avoir à un instant donnée qu'*au plus une valeur pour chacune des ses propriétés*

# M.C.D. : exemple d'association

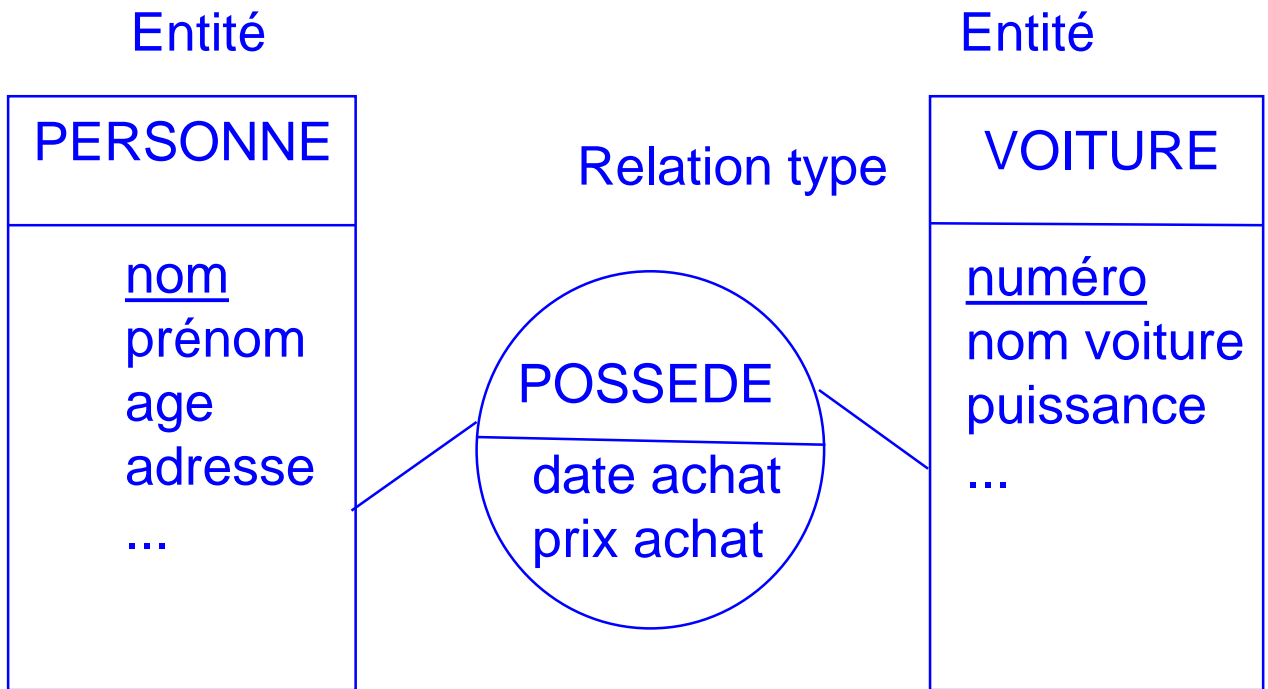
---





# M.C.D. : exemple

---



*PERSONNE* et *VOITURE* sont des *entités*

*POSSEDE* est une *relation ou Association*

*nom, prénom, age, adresse, date achat, prix achat, numéro, nom voiture, puissance* sont des *types de propriété* caractérisant un individu type ou une relation type

*nom* et *numéro* (soulignés) sont respectivement les *identifiants* de PERSONNE et VOITURE

## M.C.D. : aide-mémoire n° 1

---

- Une occurrence d'une relation ne peut exister que reliée à chacun des individus de sa collection => *la dimension d'une relation est sable*
- Une propriété d'une relation n'a de *sens que par rapport à la totalité des individus* de sa collection
- Une association peut ne pas avoir de propriété
- Si une propriété caractérise seulement un sous-ensemble des individus de la collection d'une relation => modification de la modélisation
- La dimension d'une relation n'est pas limitée
- Plusieurs associations peuvent partager la même collection
- Une même entité peut intervenir plusieurs fois dans une même association (réflexivité)

# M.C.D. : aide-mémoire n° 2

---

## □ *Attention :*

- La présence de *redondances* ou de (*trop*) *nombreuses valeurs nulles* dans les occurrences d'entités ou de relation provient souvent d'une *mauvaise modélisation des données*

# M.C.D. : cardinalités d'une entité dans une association

---

- *Traduisent le nombre d'occurrences de la l'association liées à une occurrence de l'entité*
- *S'analyse par rapport à une occurrence quelconque d'une entité participant à une relation*
- S'exprime par 2 valeurs pour chacune des entités participant à une relation : *cardinalité minimum* et *cardinalité maximum*
- Les cardinalités se notent sur la "patte" de la **relation concernée vers l'entité concerné**

# M.C.D. :valeurs typiques des cardinalités

---

□ *Cardinalité mini = 0, certaines occurrences* de l'entité participent à la relation

- Dans ce cas, les occurrences d'une relation possèdent toujours le même nombre d'individu
- Mais toutes les occurrences d'une entité concernée **ne participent pas obligatoirement** à cette relation

□ *Cardinalité mini = 1, toutes les occurrences* d'une entité participe à la relation concernée **au moins 1 fois**

## M.C.D. :

### valeurs typiques des cardinalités

---

□ *Cardinalité maxi = 1*, quand une occurrence d'une entité participe à la relation concernée, *elle n'y participe au plus qu'une fois*

□ *Cardinalité maxi = n*, quand une occurrence de l'entité participe à la relation concernée, *elle peut y participer plusieurs fois* (attention : au niveau conceptuel on ne cherche pas à chiffrer cette multiplicité tout au moins pas dans la méthode MERISE)

**M.C.D. :**

**valeurs typiques des cardinalités**

---

4 cardinalités typiques :

**0,1**

**1,1**

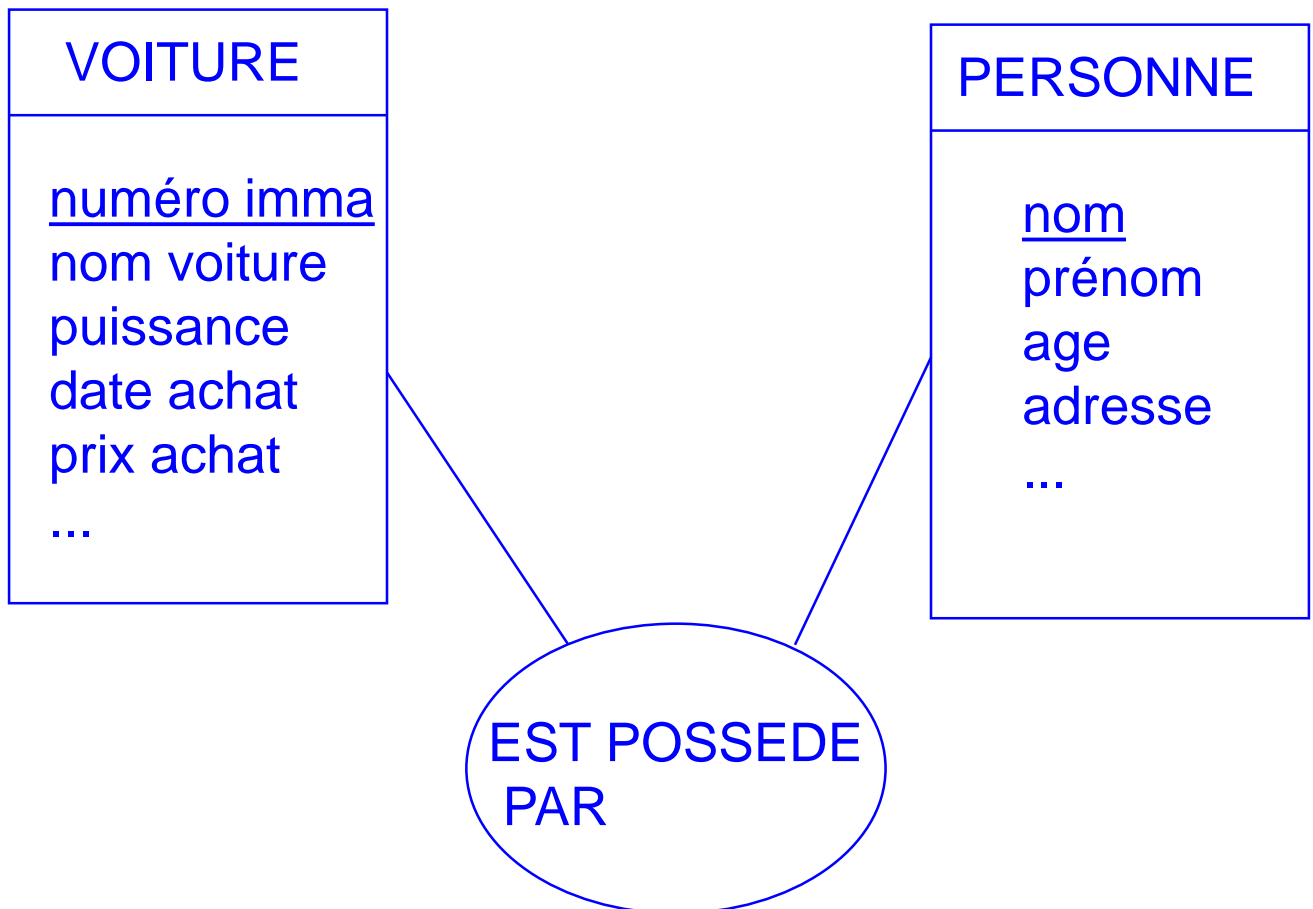
**0,n**

**1,n**

# M.C.D. :

## valeurs typiques des cardinalités

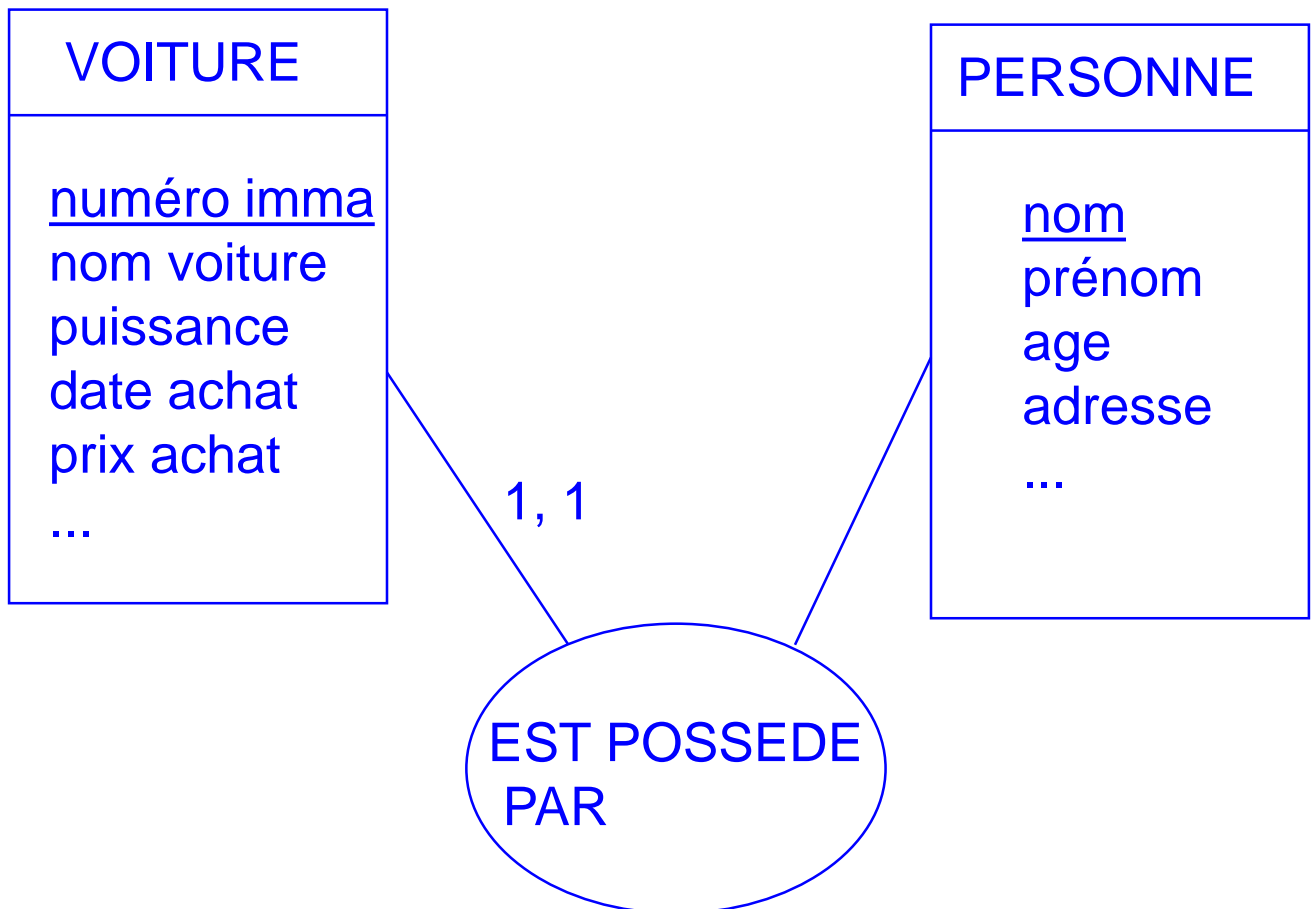
---





# M.C.D. : valeurs typiques des cardinalités

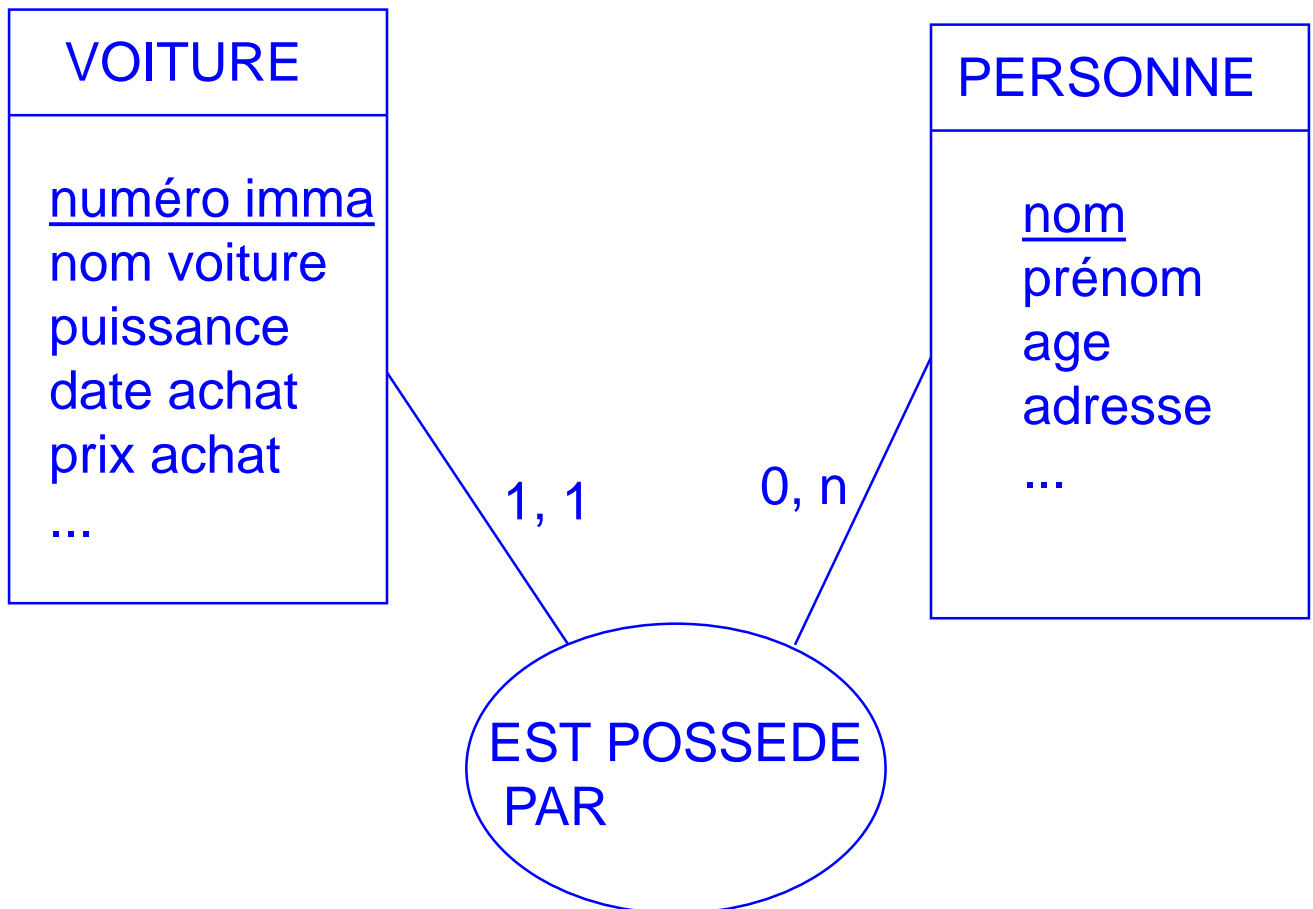
---



# M.C.D. :

## valeurs typiques des cardinalités

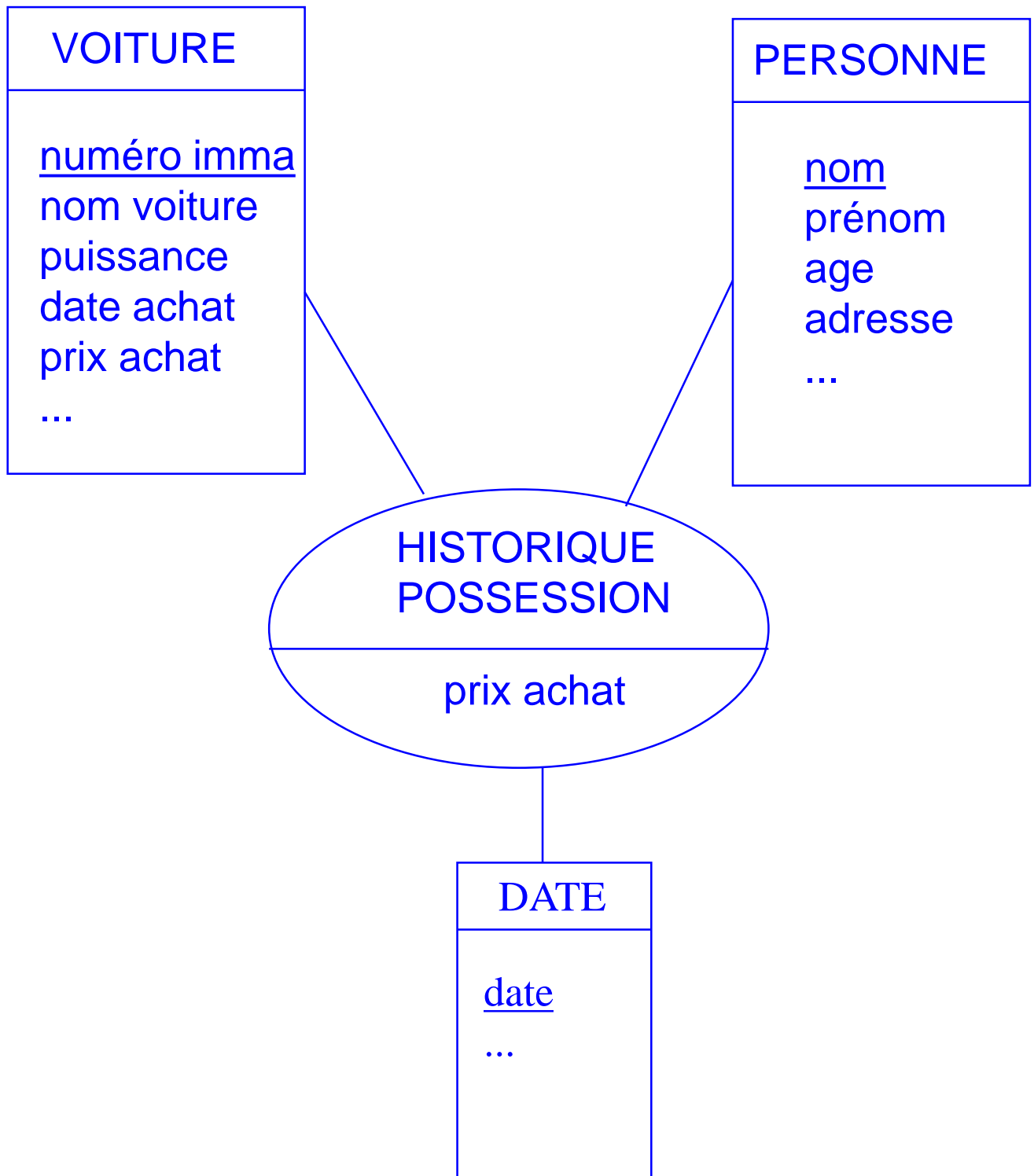
---



# M.C.D. :

## valeurs typiques des cardinalités

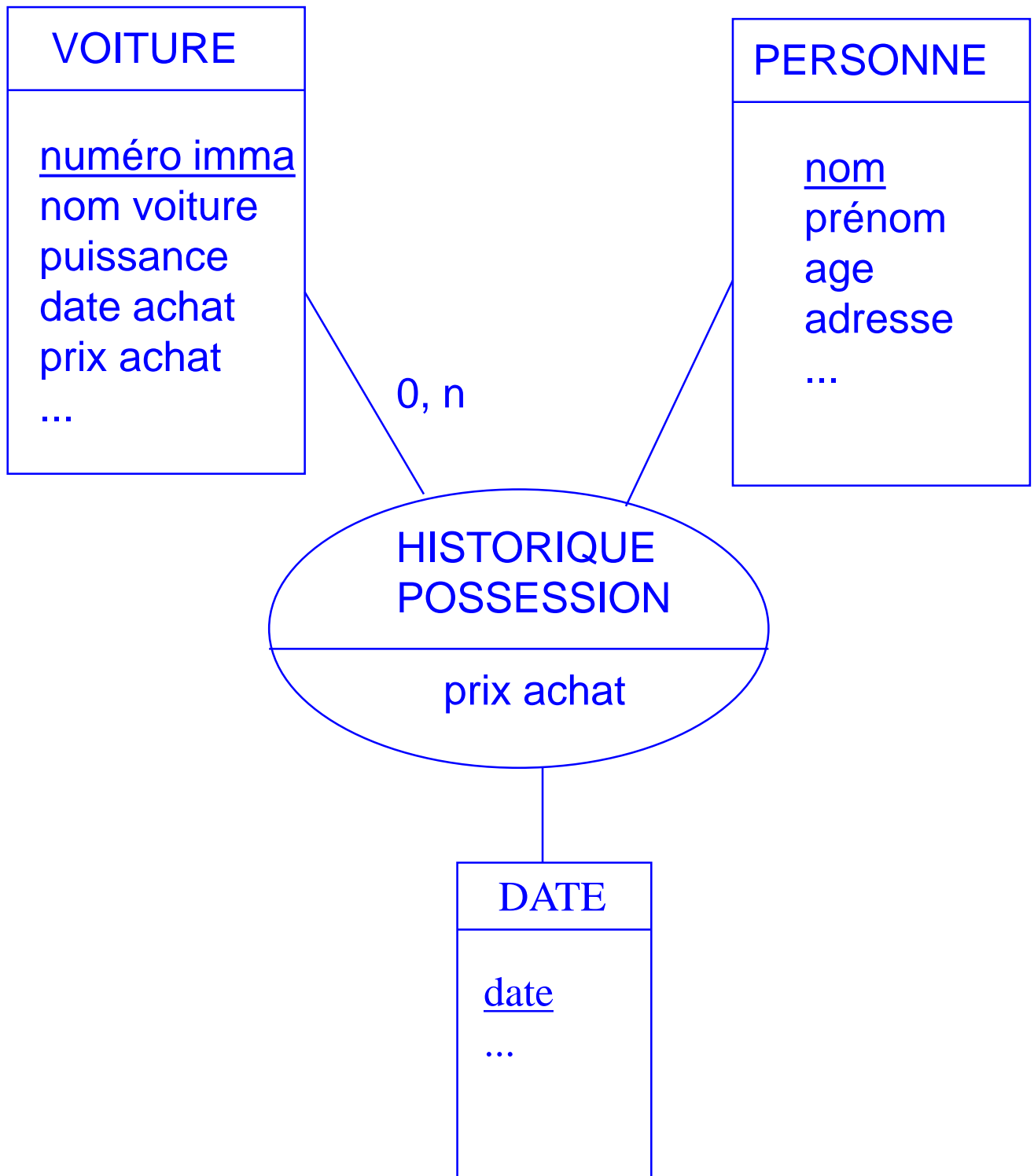
---



# M.C.D. :

## valeurs typiques des cardinalités

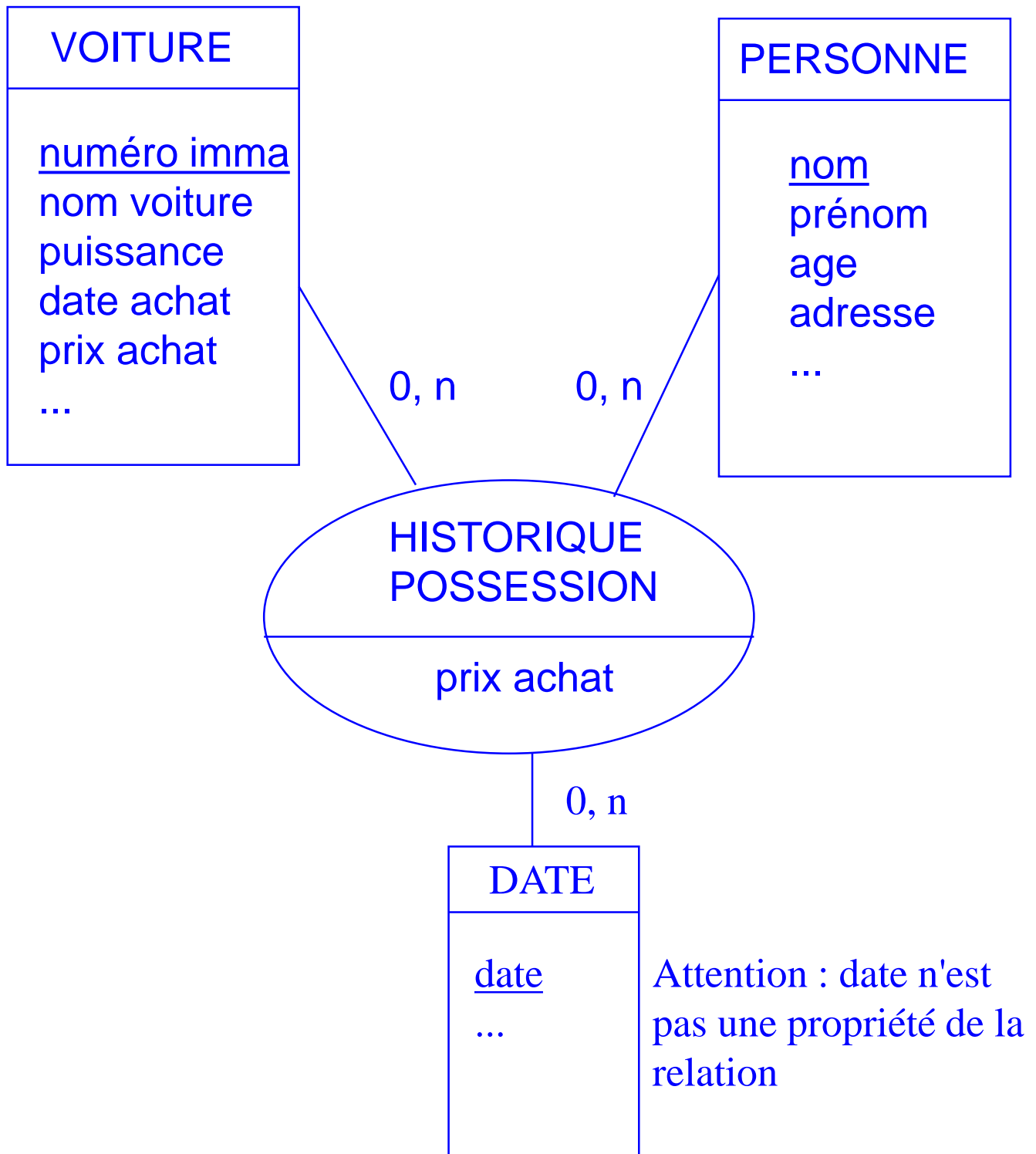
---



# M.C.D. :

## valeurs typiques des cardinalités

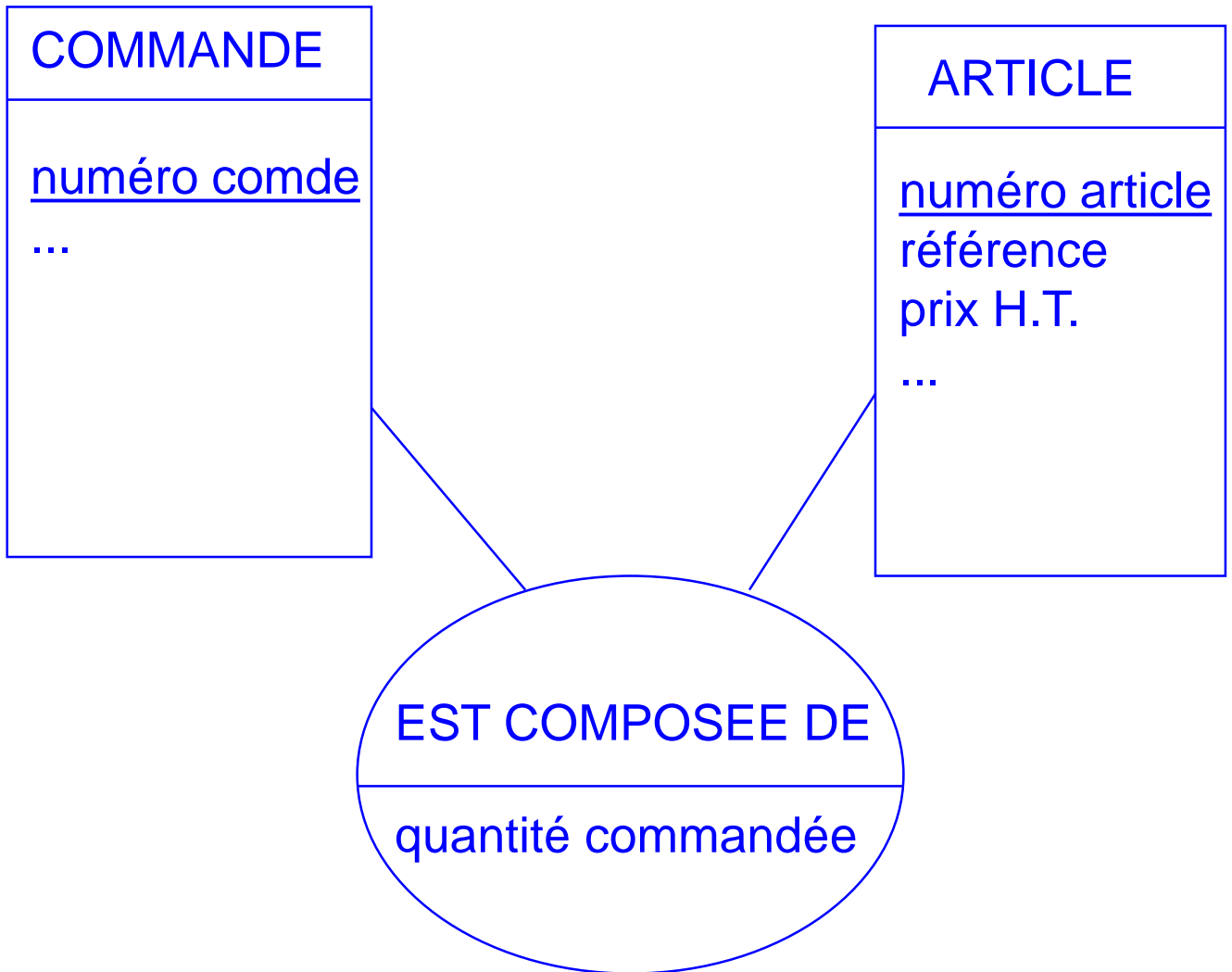
---



# M.C.D. :

## valeurs typiques des cardinalités

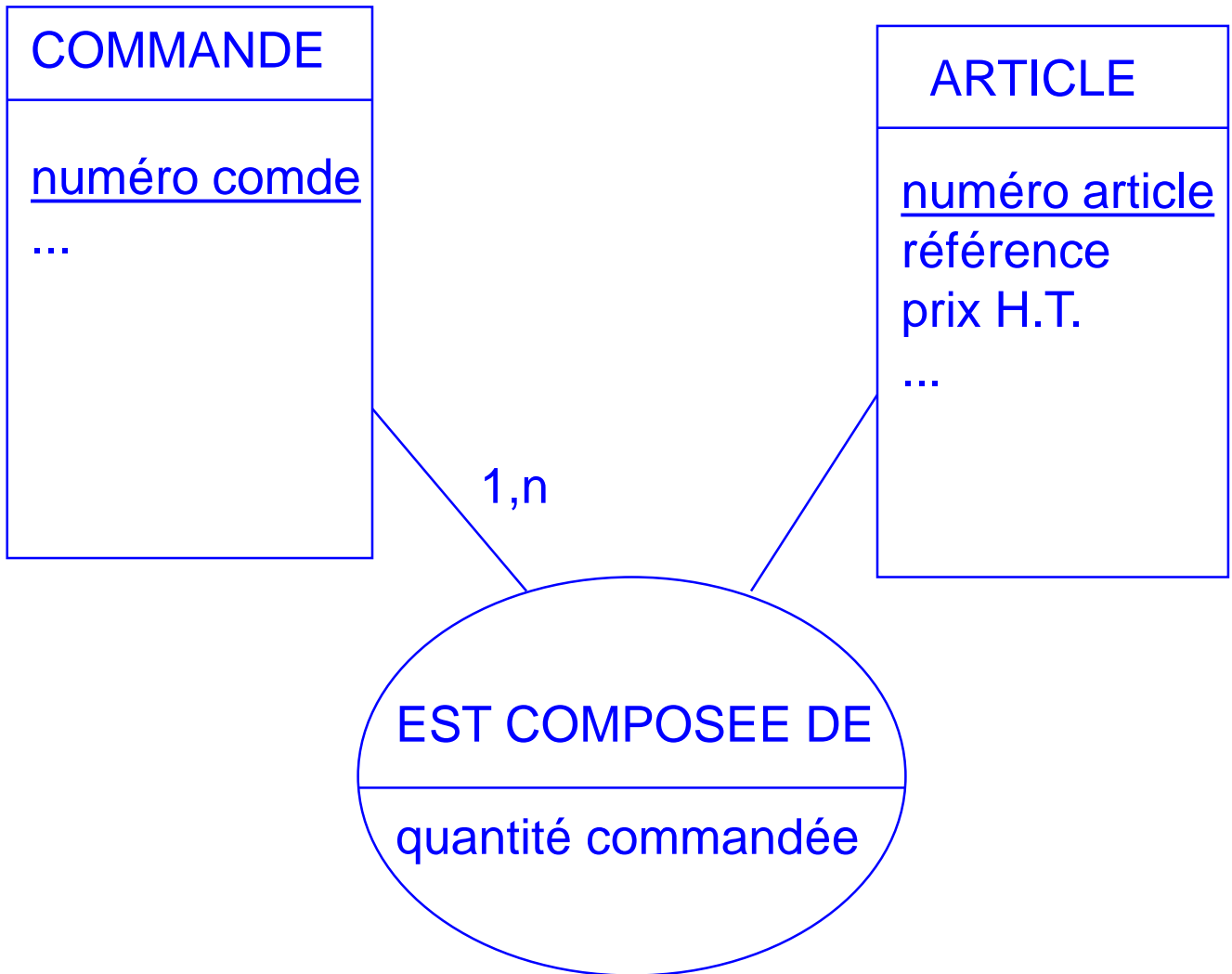
---



# M.C.D. :

## valeurs typiques des cardinalités

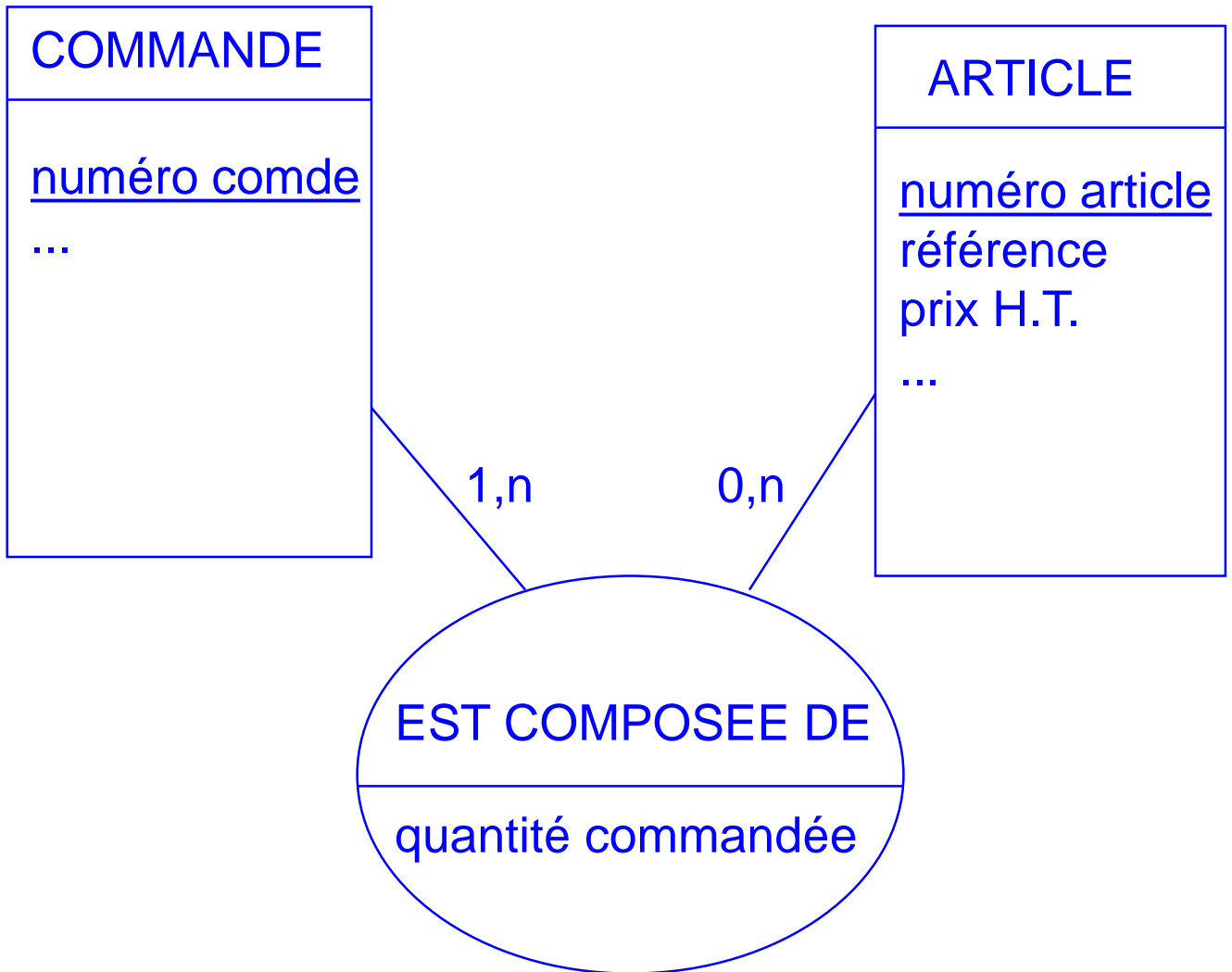
---



# M.C.D. :

## valeurs typiques des cardinalités

---





# M.L.D. : les objectifs

---

□ *Intégrer les choix organisationnels* (données informatisées ou manuelles, domaine de valeurs des propriétés, répartition organisationnelles, ...)

□ Exprimer les données (à partir du M.C.D.) dans un *formalisme logique* tenant compte de leurs conditions de mémorisation (adaptées au modèle de donné du S.G.B.D. envisagé)

- *Traduire avec un algorithme les entités, les relations et les cardinalités du M.C.D. dans un formalisme logique retenu (Codasyl, Relationnel, ...)*

□ Remarque :

- C'est dans le M.P.D. que le M.L.D. sera traduit directement dans la syntaxe du S.G.B.D. retenu

# M.L.D. : les modèles logiques

---

□ Plusieurs modèles sont disponibles pour la représentation du modèle logique de données, par exemple :

- *Le modèle relationnel* (défini par E.F. Codd en 1970)
- *Le modèle navigationnel*
- *Le modèle hiérarchique*
- *Le modèle objet voir objet relationnel*

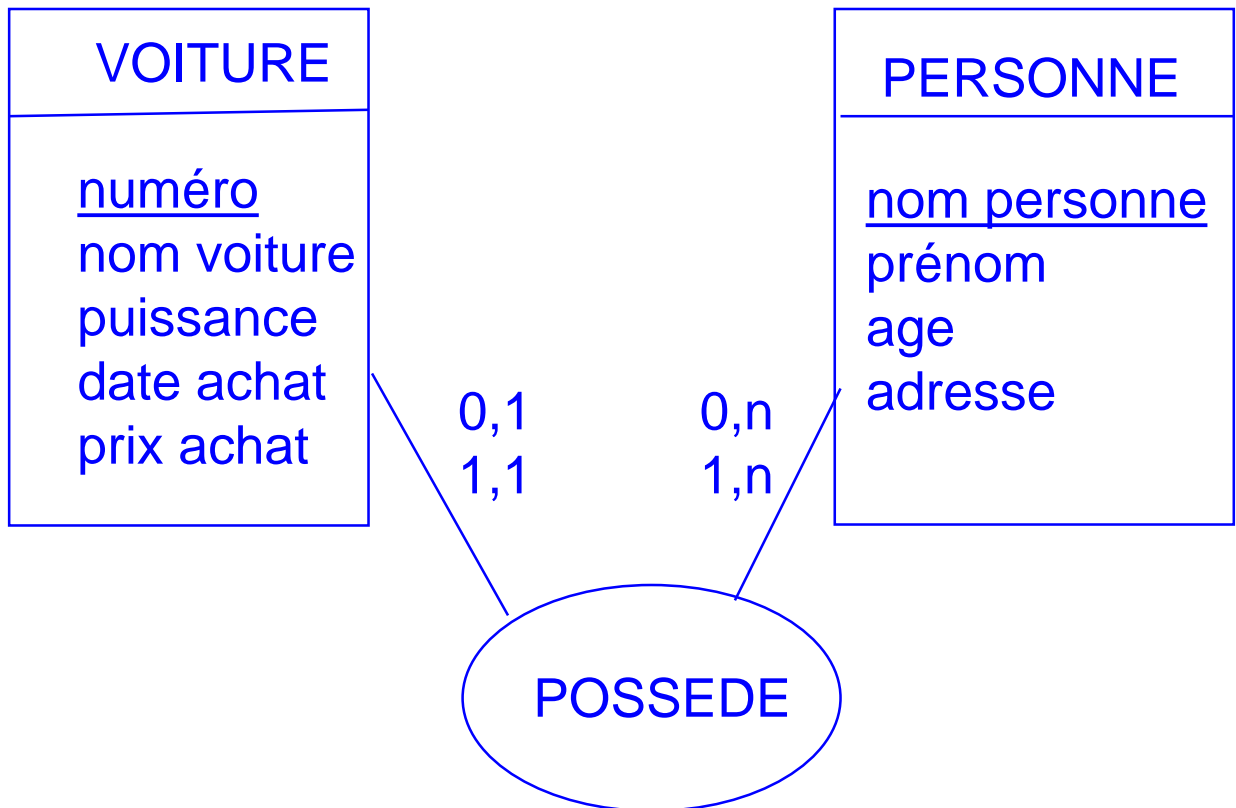
□ A ces modèles sont associés les :

- *S.G.B.D. relationnels* (maintenant les plus utilisés)
- *S.G.B.D. navigationnels*
- *S.G.B.D. hiérarchique*
- *SGBD objet voir objet relationnel*

# Transformation du formalisme entité-relation en formalisme relationnel

---

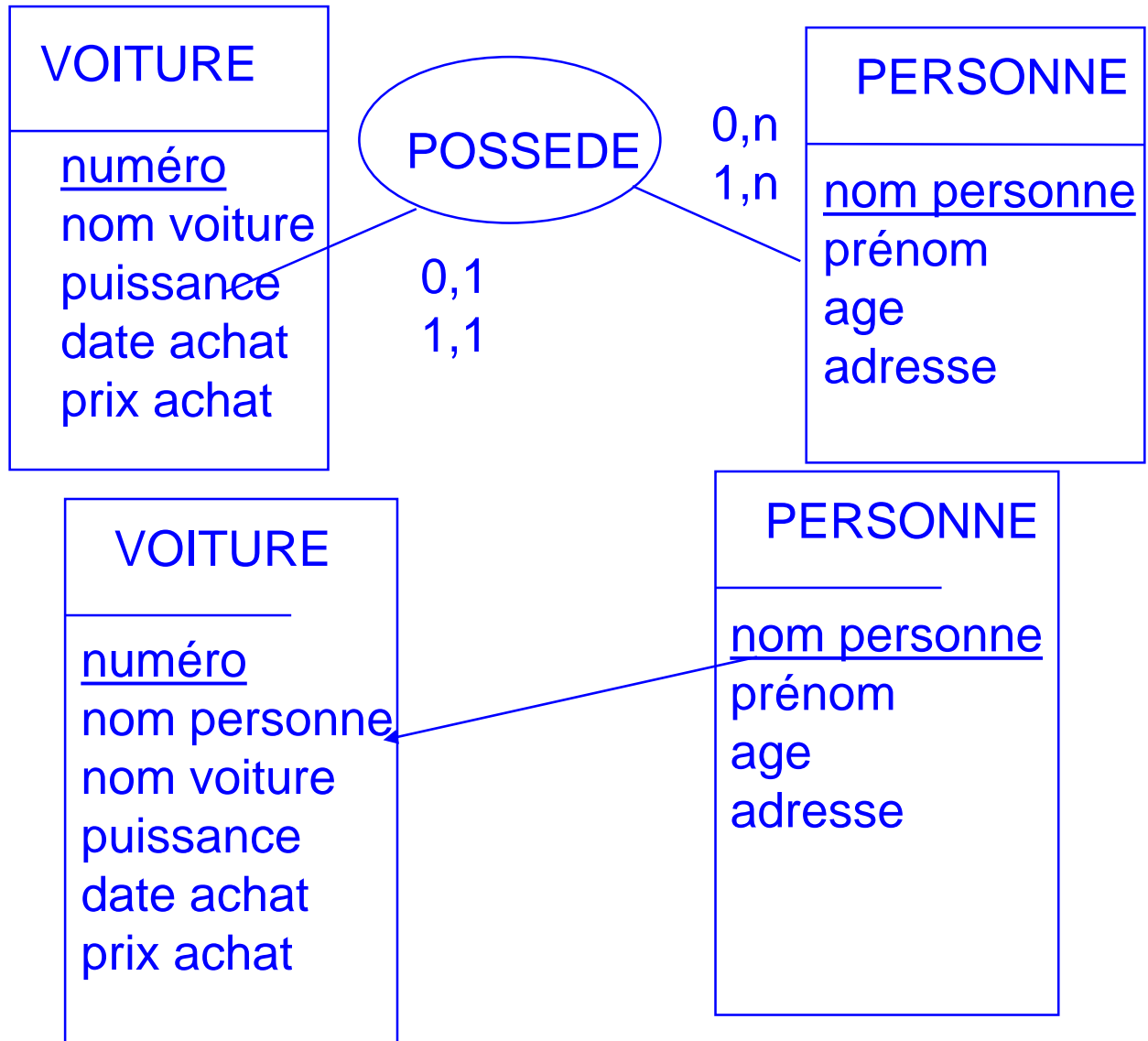
Relation binaire du type  $(0,n)-(1,1)$ ,  $(1,n)-(1,1)$ ,  $(0,n)-(0,1)$  ou  $(1,n)-(0,1)$



□ On *duplique la clé de la table issue de l'individu  $(0,n)$  ou  $(1,n)$  dans la table issue de l'individu à cardinalité  $(0,1)$  ou  $(1,1)$* , les éventuelles propriétés de la relation devenant aussi des attributs de cette table.

□ Pour chaque attribut, on précise son domaine de variation

# Transformation du formalisme entité-relation en formalisme relationnel



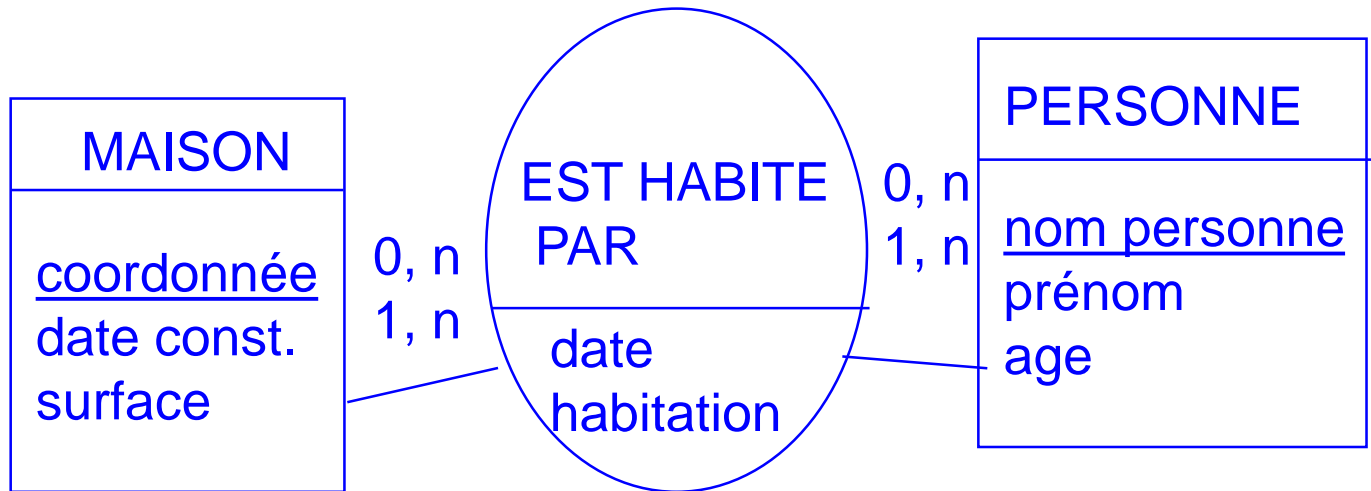
VOITURE (numéro : entier ; **nom personne** : car (20)  
; nom voiture : car(20) ; puissance : [2, 40] ; date  
achat : date ; prix achat : réel)

PERSONNE (nom personne : car (20) ; prénom : car  
(20) ; age : [18, 150] ; adresse : car (40) )

# Transformation du formalisme entité-relation en formalisme relationnel

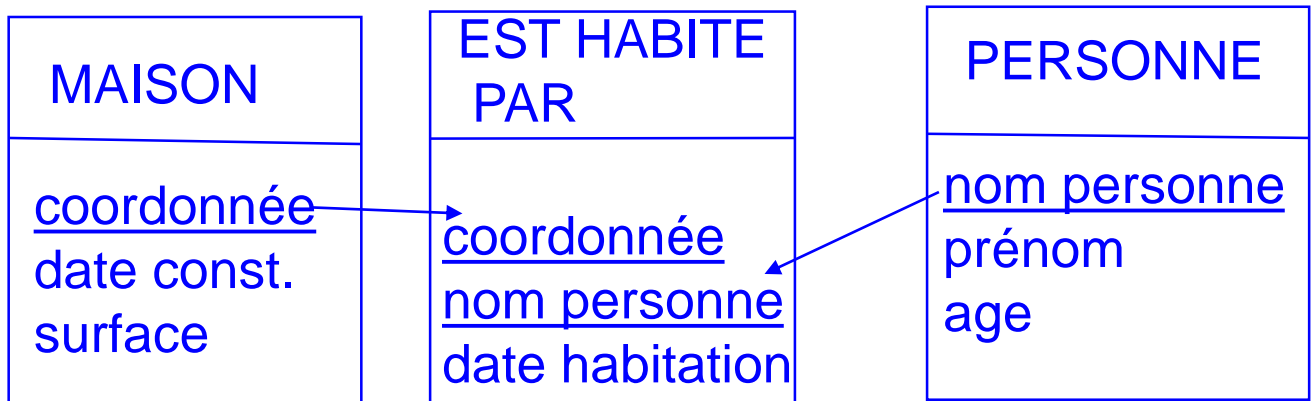
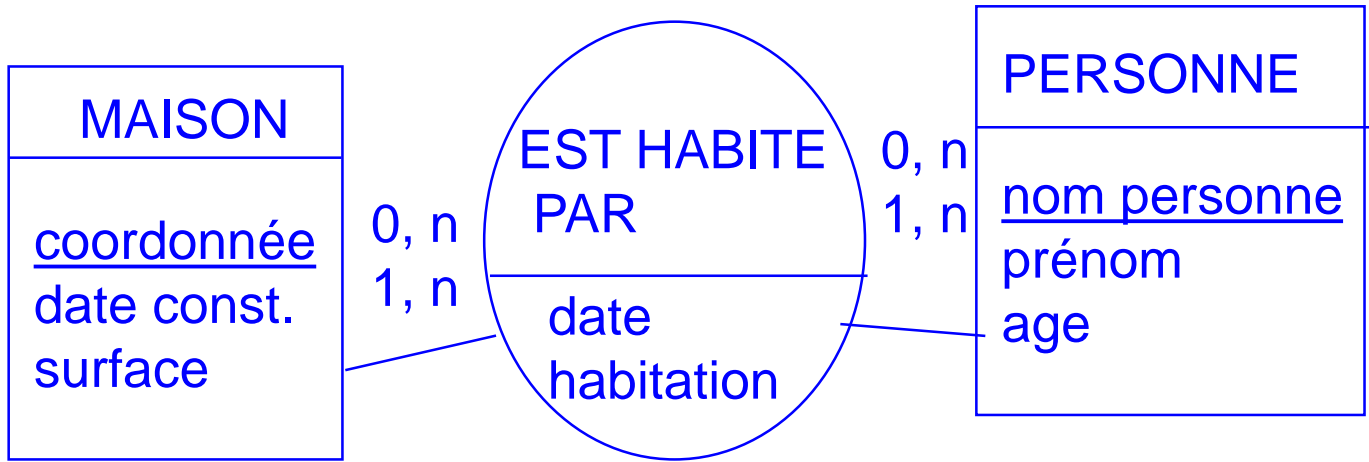
---

Relation binaire du type  $(0,n)-(0,n)$ ,  $(1,n)-(1,n)$  ou  $(0,n)-(1,n)$



- A partir de la relation, on *crée une table ayant une clé composée des clés des tables issues des 2 individus*,
- Les éventuelles propriétés de la relation devenant aussi des attributs de cette table.

# Transformation du formalisme entité-relation en formalisme relationnel



**EST HABITE PAR** (coordonnée : entier ; nom personne : car (20) ; date habitation : date)

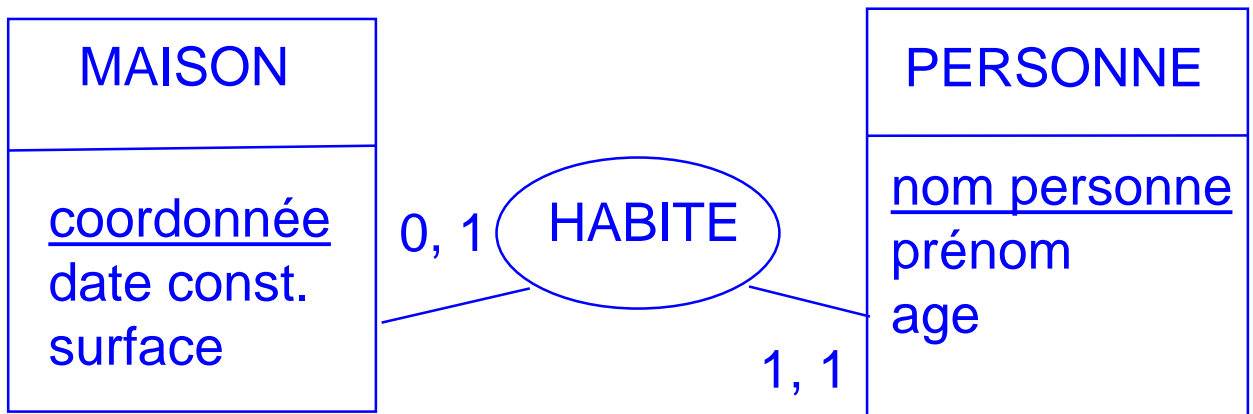
**PERSONNE** (nom personne : car (20) ; prénom : car (20) ; age : [18, 150])

**MAISON** (coordonnée : entier ; date const. : date ; surface : réel)

# Transformation du formalisme entité-relation en formalisme relationnel

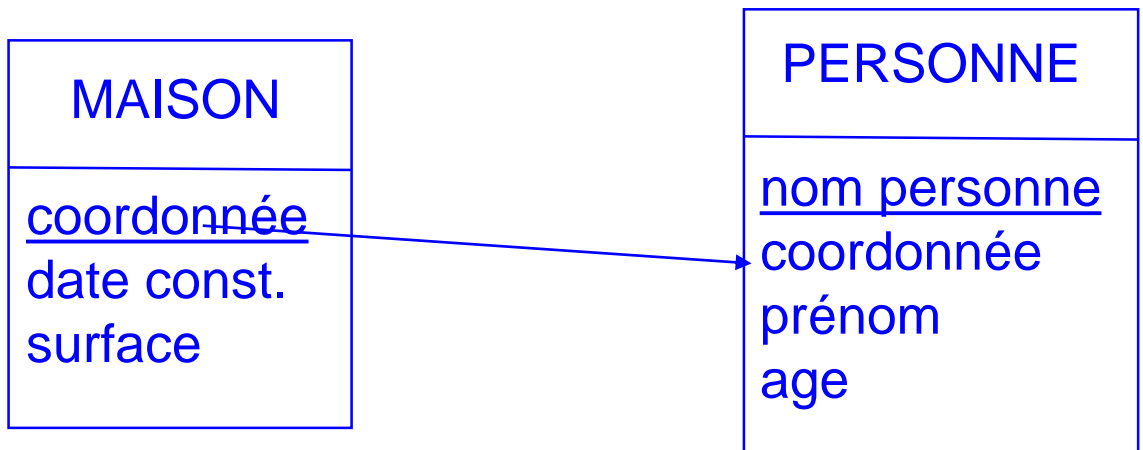
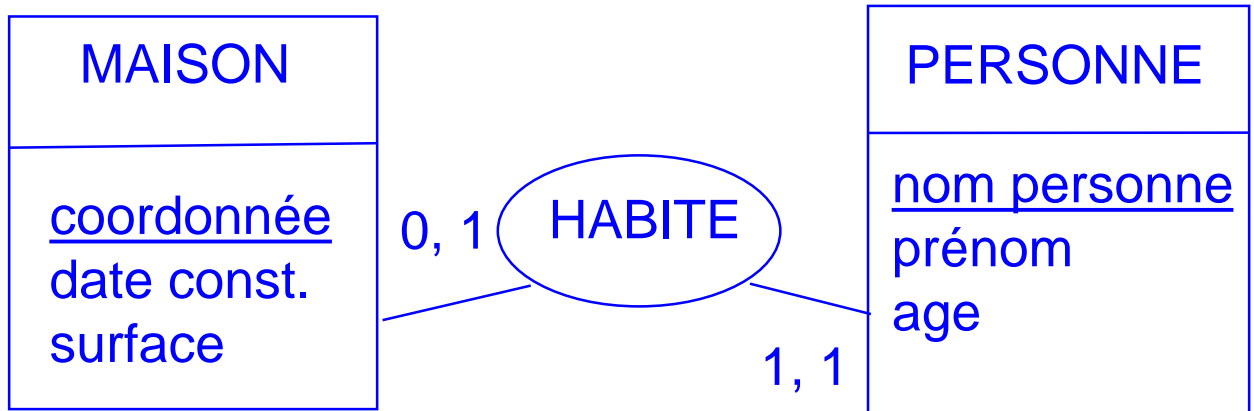
---

Relation binaire du type (O,1)-(1,1)



- On *duplique la clé de la table issue de l'entité à cardinalité (0,1)* dans la table issue de l'entité à cardinalité (1,1),
- Les éventuelles propriétés de la relation devenant aussi des attributs de cette table issue de l'entité à cardinalité (1,1)

# Transformation du formalisme entité-relation en formalisme relationnel



PERSONNE (nom personne : car (20) ; **coordonnée** : entier ; prénom : car (20) ; age : [18, 150])

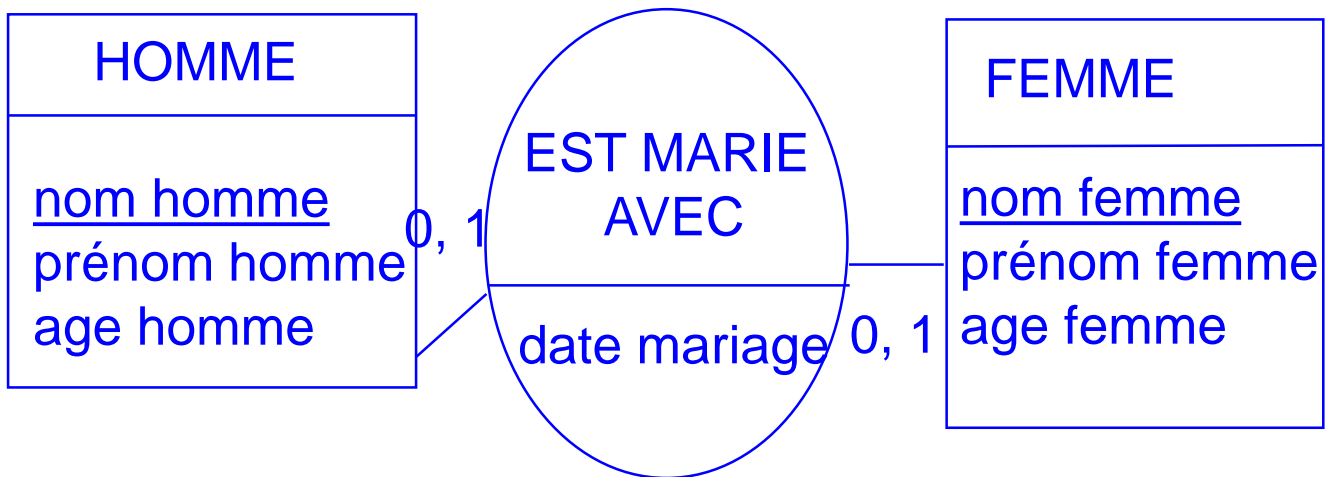
MAISON (coordonnée : entier ; date const. : date ; surface : réel)



# Transformation du formalisme entité-relation en formalisme relationnel

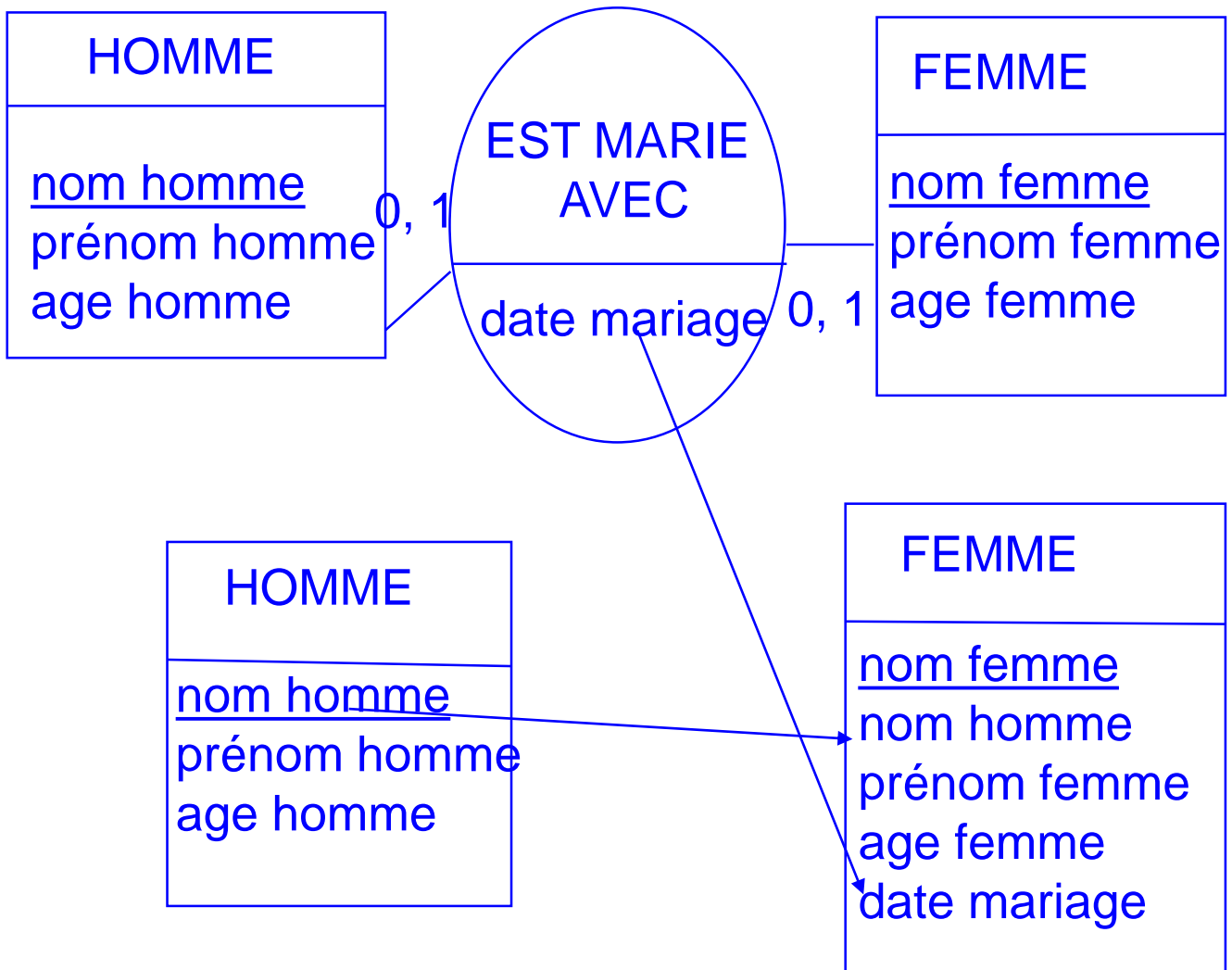
---

Relation binaire du type (0,1)-(0,1)



Les cardinalités étant identiques des 2 côtés, on **duplique au choix la clé d'une table issue d'un l'individu dans la table issue de l'autre individu**, les éventuelles propriétés de la relation devenant aussi des attributs de cette table

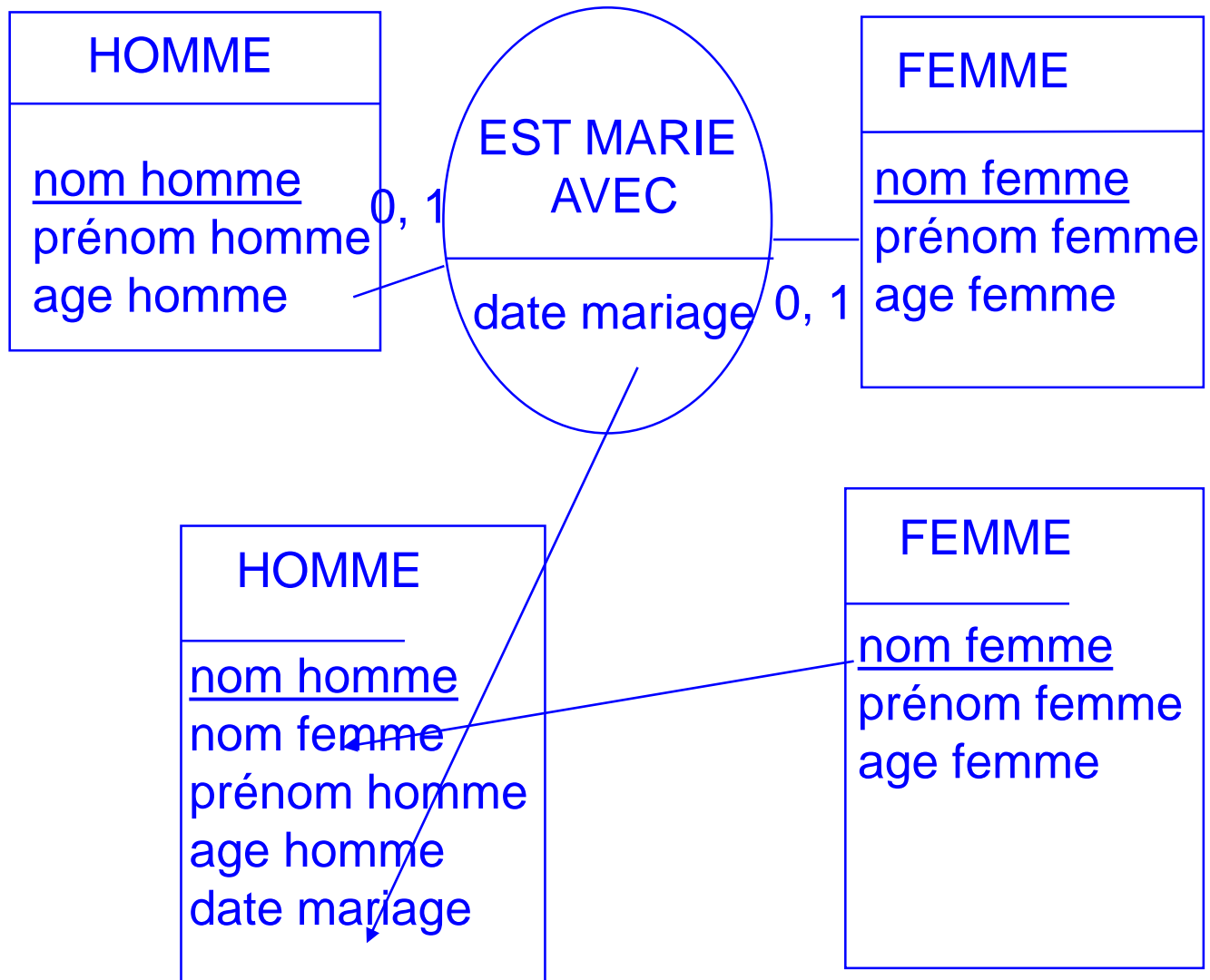
# Transformation du formalisme entité-relation en formalisme relationnel



**HOMME** (nom homme : car (20) ; prénom homme : car (20) ; age : [18, 150])

**FEMME** (nom femme : car (20) ; **nom homme** : car (20) ; prénom femme : car (20) ; age : [18, 150] ; **date mariage** : date)

# Transformation du formalisme entité-relation en formalisme relationnel



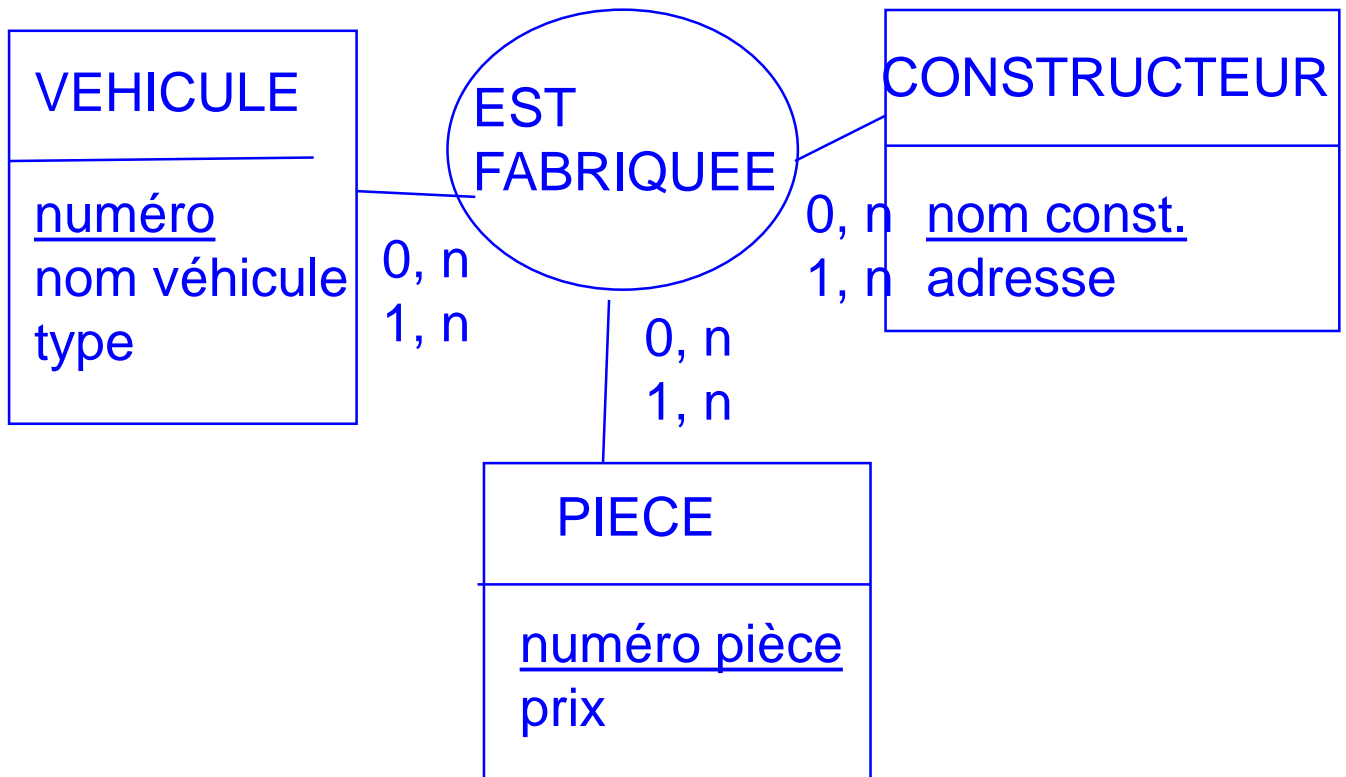
**HOMME** (nom homme : car (20) ; **nom femme** : car (20) ; prénom homme : car (20) ; age : [18, 150] ; **date mariage**)

**FEMME** (nom femme : car (20) ; prénom femme : car (20) ; age : [18, 150])

# Transformation du formalisme entité-relation en formalisme relationnel

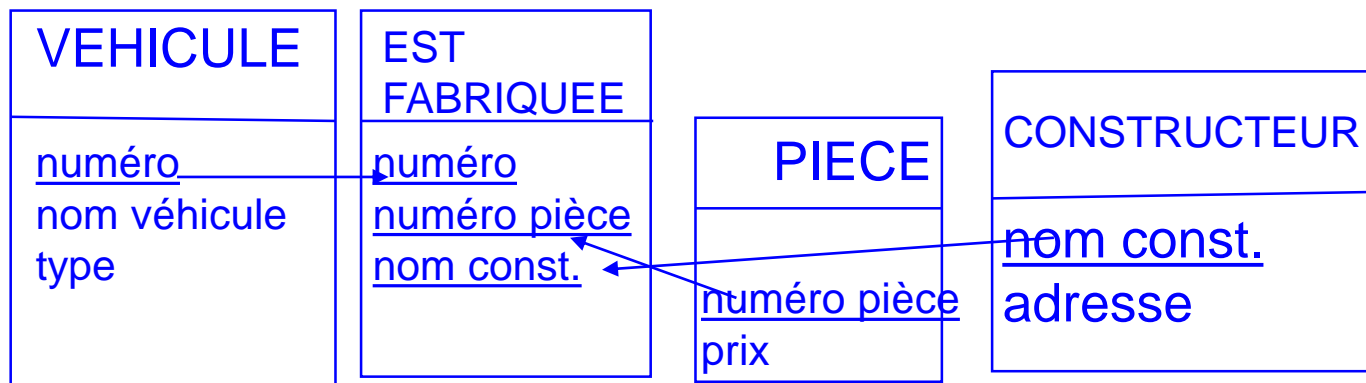
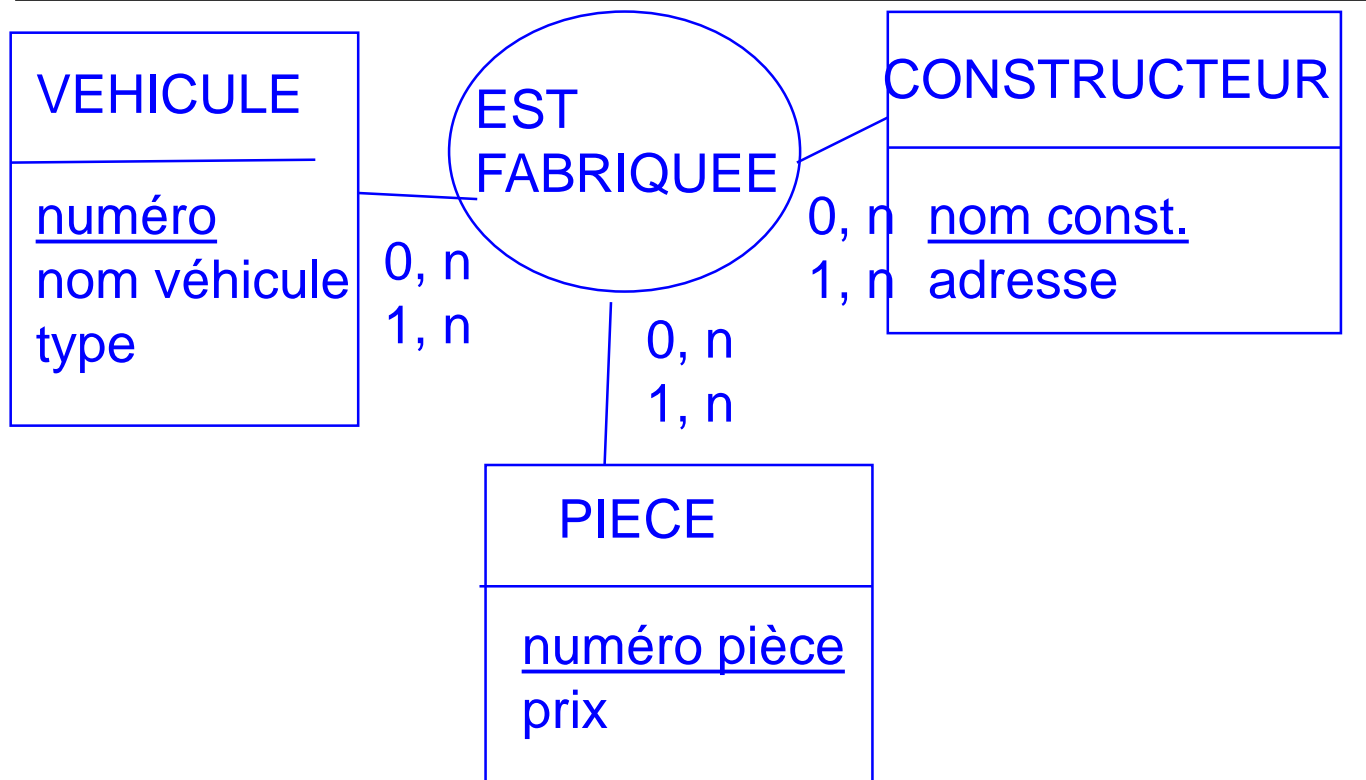
---

Relation ternaire ou supérieure



A partir de la relation, on **crée une table avec une clé composée des identifiants de tous les individus** concernés, les éventuelles propriétés de cette relation devenant des attributs de la nouvelle table.

# Transformation du formalisme entité-relation en formalisme relationnel



**EST FABRIQUEE** (numéro : entier ; numéro pièce : entier ; nom const. : car (20) )

**VEHICULE** ( numéro : entier ; nom véhicule : car (20) ; type : car (10) )

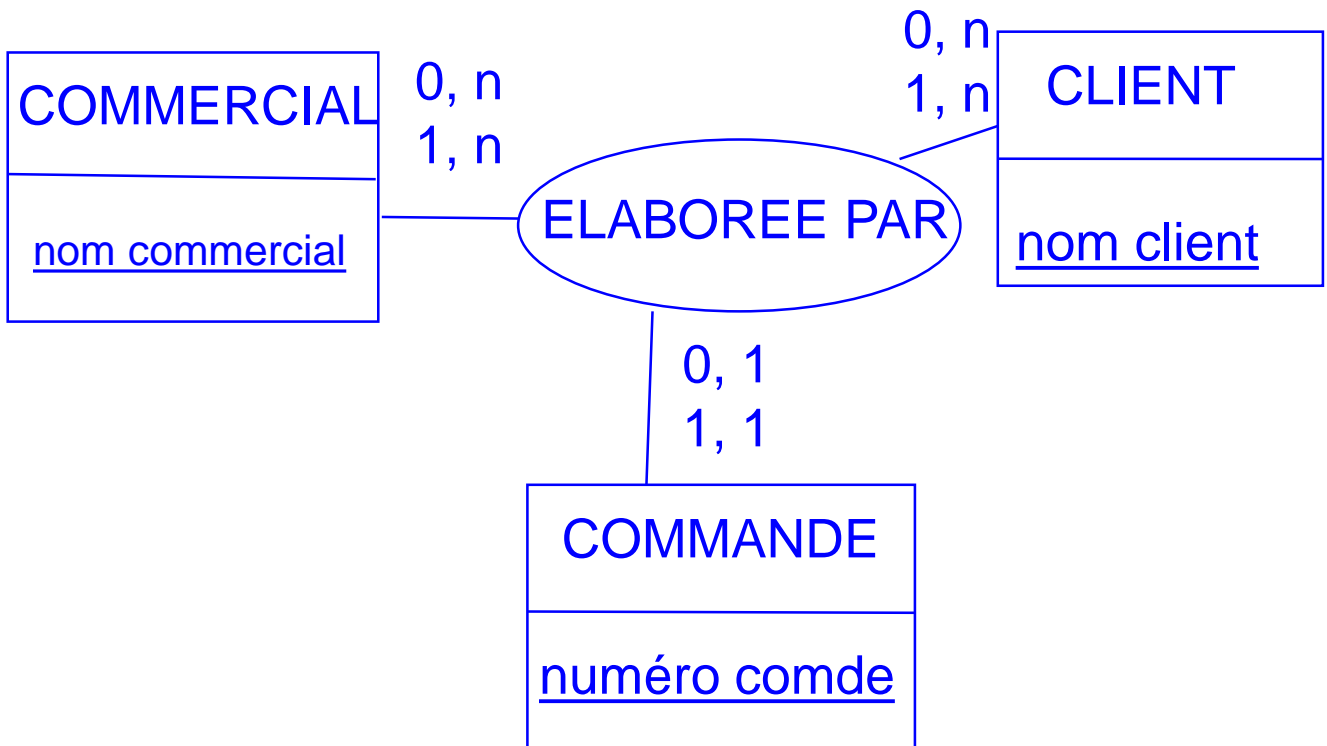
**PIECE** (numéro pièce : entier ; prix : réel)

**CONSTRUCTEUR** (nom const. : car (20) ; adresse : car (40) )

# Transformation du formalisme entité-relation en formalisme relationnel

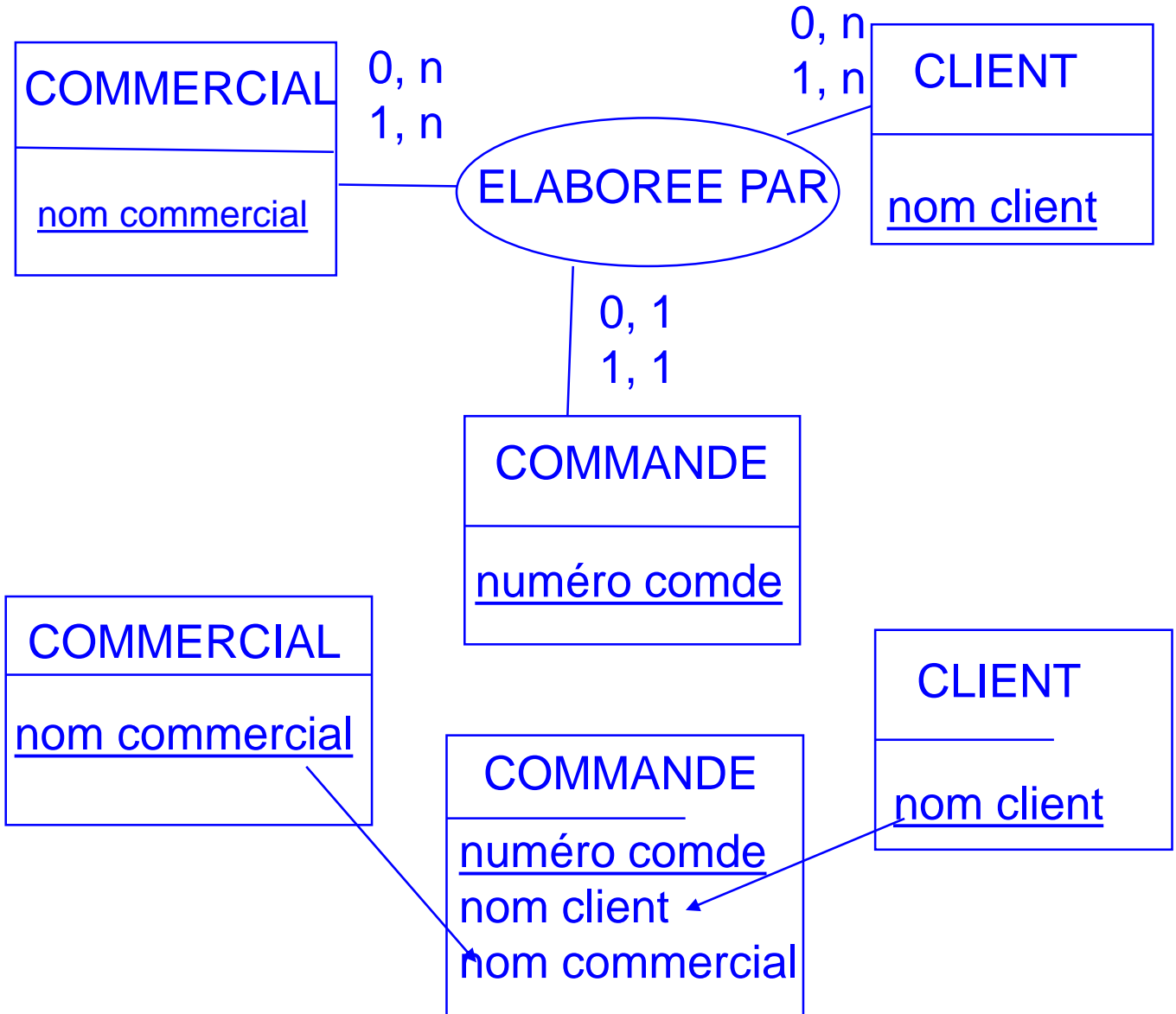
---

## Relation ternaire



On **duplique les clés des tables issues des individus à cardinalité  $(0, n)$  ou  $(1, n)$**  dans la table issue de l'individu à cardinalité  $(0,1)$  ou  $(1,1)$ , les éventuelles propriétés de relation "ELABOREE PAR" devenant aussi des attributs de cette table

# Transformation du formalisme entité-relation en formalisme relationnel



COMMERCIAL (nom commercial : car (20) )

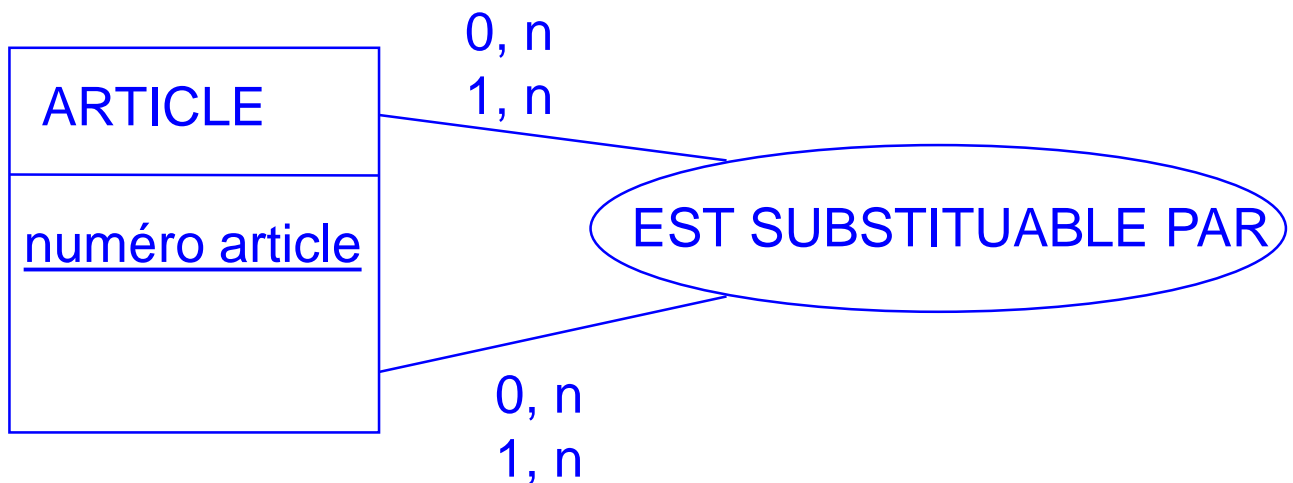
CLIENT (nom client : car (20) )

COMMANDE (numéro comde : entier ; **nom client** : car (20) ; **nom commercial** : car (20) )

# Transformation du formalisme entité-relation en formalisme relationnel

---

Relation réflexive  $(0, n)-(0, n)$ ,  $(1, n)-(1, n)$  ou  $(0, n)-(1, n)$

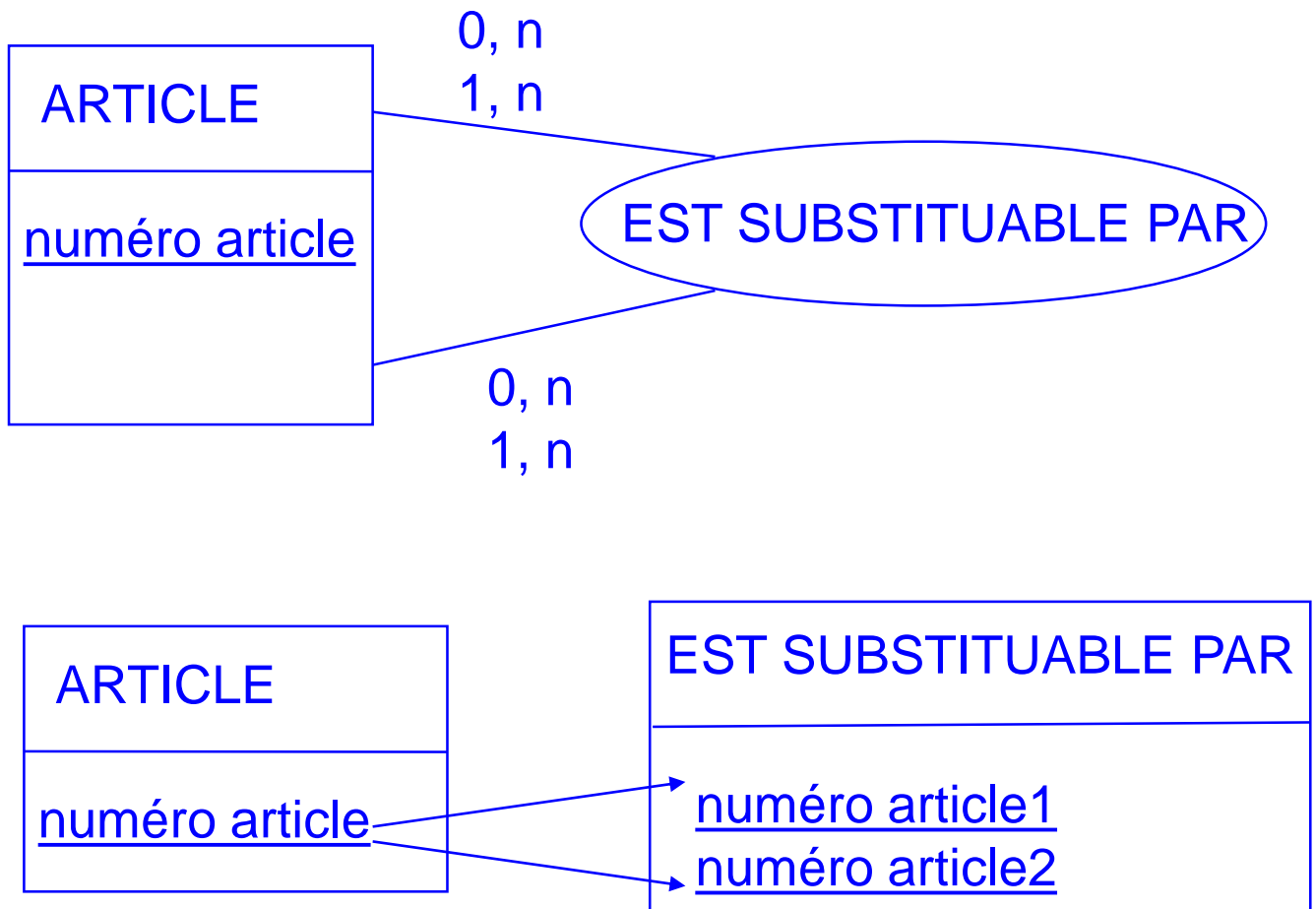


A partir de la relation, on crée une table ayant comme clé ***une clé composée de 2 fois l'identifiant du seul individu*** participant à cette relation, les éventuelles propriétés de celle-ci devenant des attributs de la nouvelle table.



# Transformation du formalisme entité-relation en formalisme relationnel

---



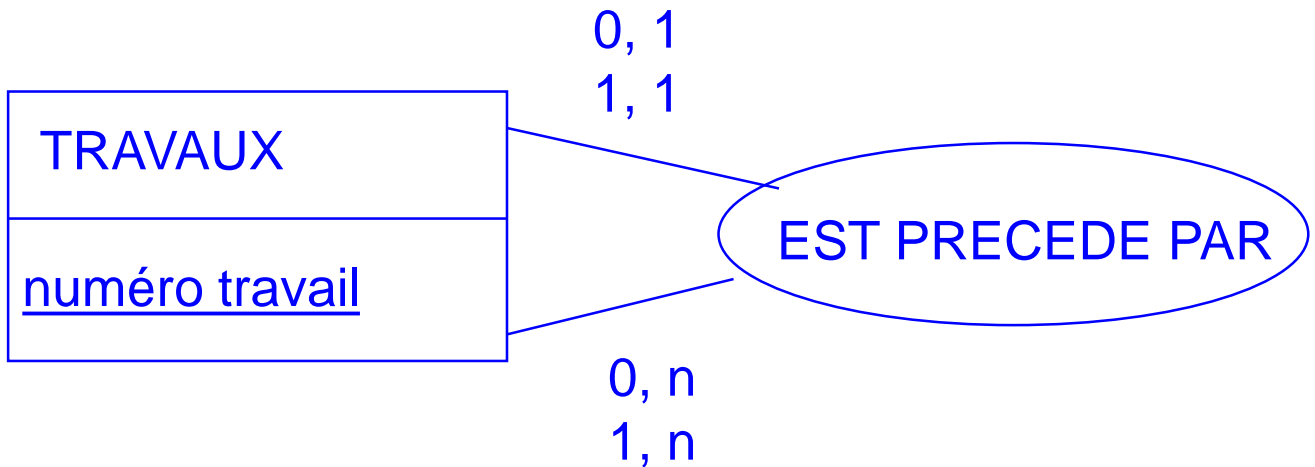
ARTICLE (numéro article : entier)

EST SUBSTITUABLE PAR (  
numéro article1 : entier ; numéro article2 : entier)

# Transformation du formalisme entité-relation en formalisme relationnel

---

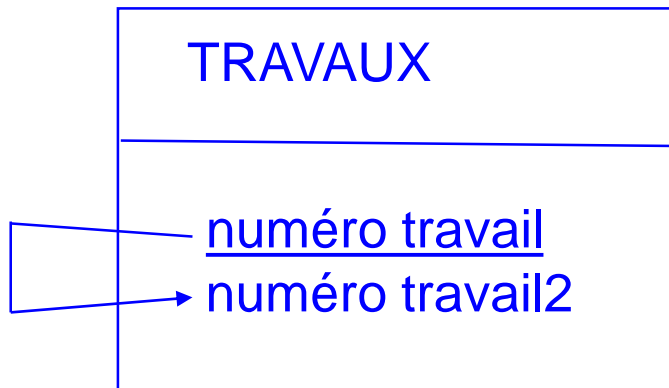
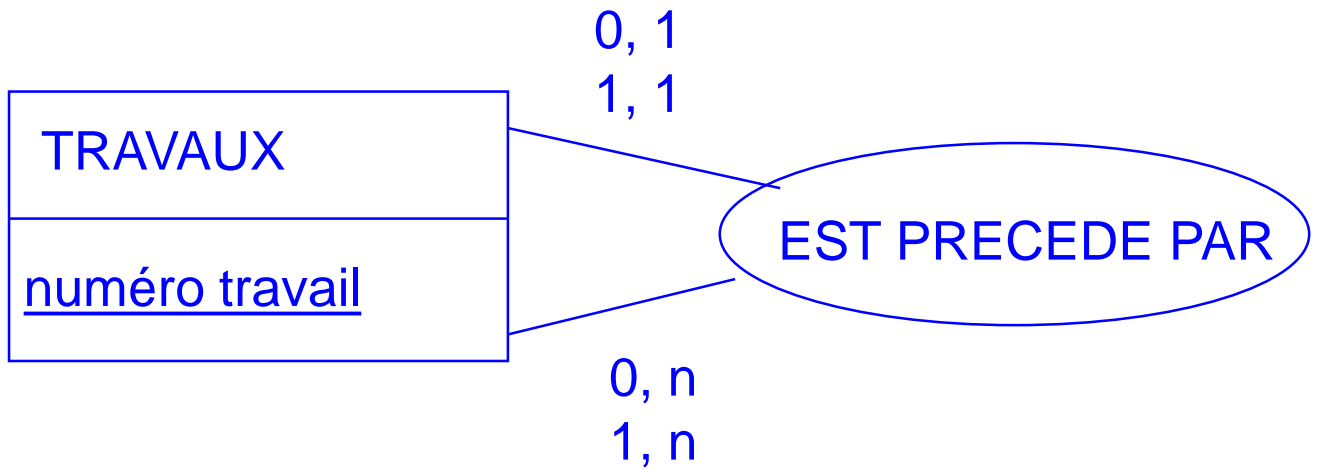
Relation réflexive  $(0, 1)-(0, n)$ ,  $(1, 1)-(0, n)$ ,  $(1, 1)-(0, n)$   
ou  $(1, 1)-(1, n)$



On **duplique la clé de la table issue de l'individu à cardinalité  $(0,n)$  ou  $(1,n)$**  dans la table issue de l'individu à cardinalité  $(0,1)$  ou  $(1,1)$ , les éventuelles propriétés de la relation devenant aussi des attributs de cette table.

# Transformation du formalisme entité-relation en formalisme relationnel

---



**TRAVAUX** (numéro travail : entier ;  
**numéro travail2** : entier)

# Objectifs des S.G.B.D.

---

- *Indépendance physique*
- *Indépendance logique* (données-traitements)
- *Indépendance externe*
- *Manipulation par des non-informaticiens :  
ergonomie, convivialité*
- *Efficacité des accès*
- *Administration centralisée* des données
- *Non redondance*
- *Cohérence*
- *Partageabilité*(gestion des accès concurrents)
- *Sécurité* (pannes , accès mal intentionnés)