

Base de données I– 420-325

Modèle conceptuel des données (MCD)

Enseignant: Naji Bricha
(Naji.Bricha@climoilou.qc.ca)

Plan

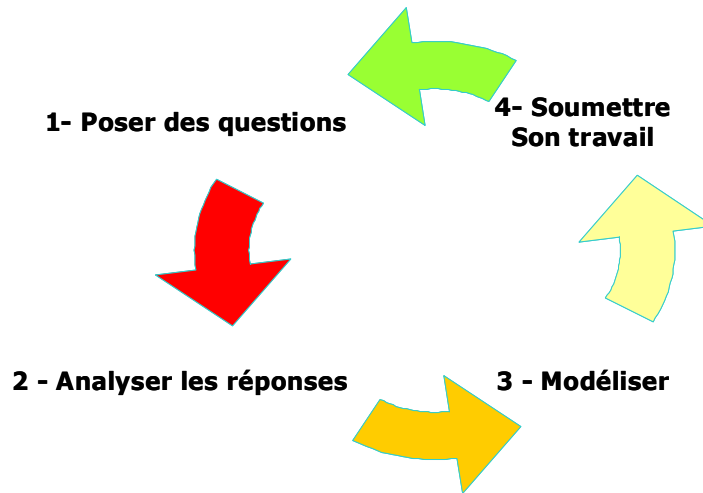
1. Le dictionnaire des données
2. Le modèle conceptuel des données
 - a) Éléments composant un MCD.
 - b) Vérification et normalisation du modèle E/A.
 - c) Contraintes d'intégrité du modèle E/A ou extensions du modèle E/A.
 - d) Association réflexive.

Objectif du MCD

Décrire les données du SI, indépendamment de tout choix d'implantation physique.

1. Le dictionnaire des données

- **Inventaire** des données du domaine étudié.



Nombreuses caractéristiques :

- identificateur (mnémonique),
- description (« sens » précis),
- type (numérique, alphanumérique, ...),
- taille,
- mode d'obtention :
 - donnée mémorisée,
 - donnée calculée,
 - donnée "paramètre" : donnée utile à un traitement et non mémorisée (ex : date d'édition),
- règle de calcul (pour les données calculées),
- contraintes d'intégrité : intervalle de valeurs, liste de valeurs...
- origine (document, système, service)

Descriptif **très simplifié** utilisé dans les exercices où toutes ces caractéristiques ne sont pas toujours disponibles :

documents

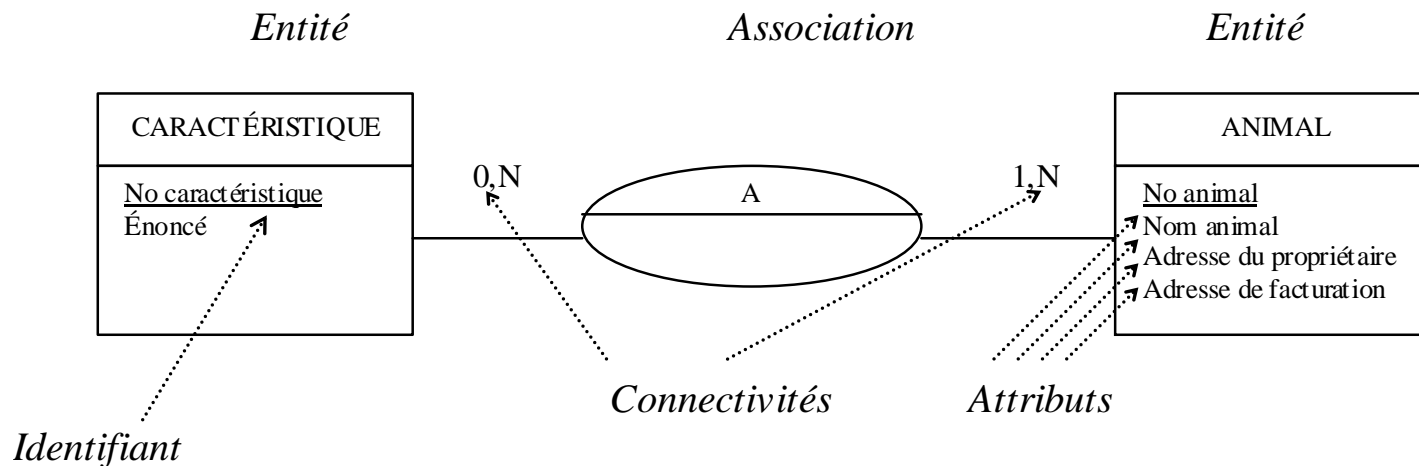
Rubrique	Description	Type	Mode	D1	D2	D3	D4
identificateur	libellé	entier réel date chaîne booléen	mémorisée calculée paramètre		x		x
				x			
				x	x	x	x

2. Le modèle conceptuel des données : le modèle entité/association

- a) Éléments composant un MCD.
- b) Vérification et normalisation du modèle E/A.
- c) Contraintes d'intégrité du modèle E/A ou extensions du modèle E/A.
- d) Association réflexive

a) Éléments composant un MCD

1. Attribut (ou propriété)
2. Entité
3. Identifiant
4. Association (ou relation)
5. Connectivité (ou cardinalité)



Attributs

- Les attributs sont les propriétés qui définissent une entité ou une association
- Exemples:
 - Nom d'un employé,
 - Date de naissance d'un étudiant.

Entité

Tout objet concret ou abstrait ayant une existence propre et conforme aux besoins de gestion de l'organisation.

Exemple:

ETUDIANT
<u>code permanent ETUDIANT</u>
nom ETUDIANT
prenom ETUDIANT

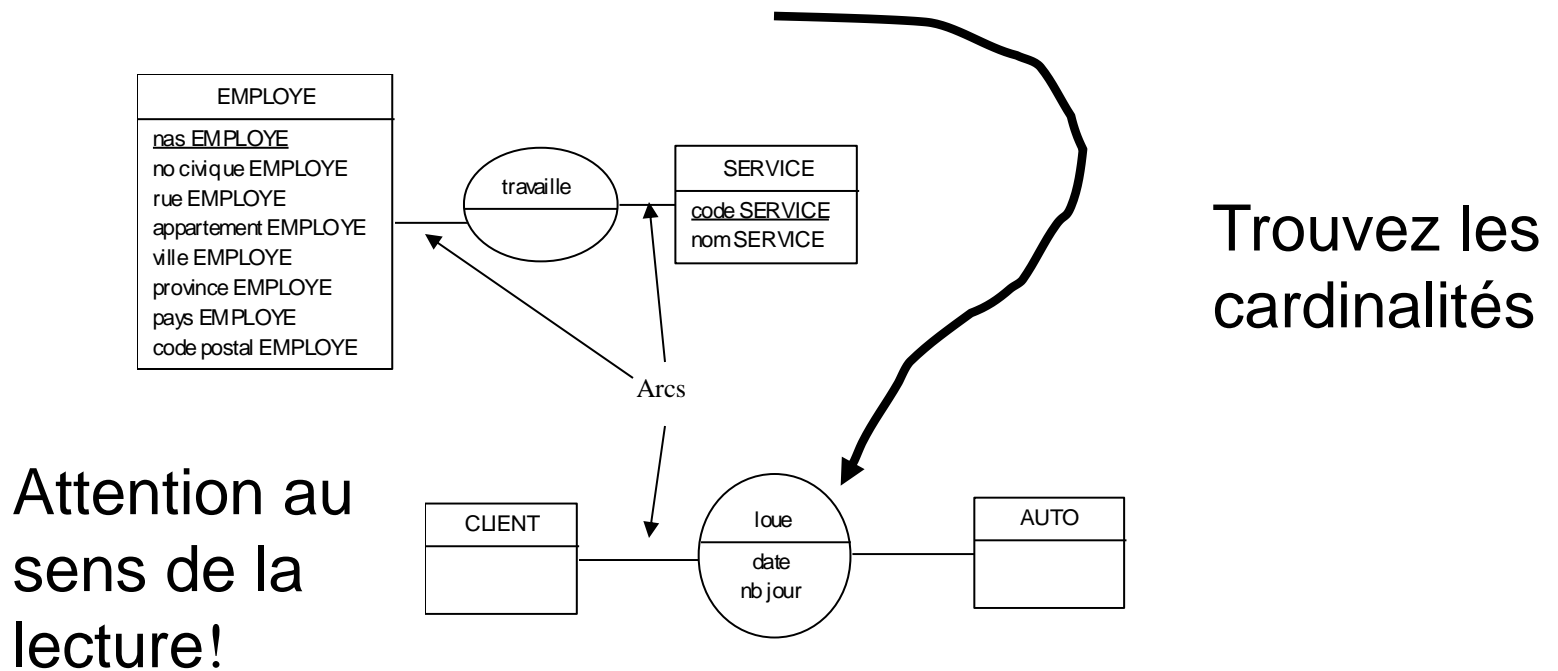
Normes d'appellation: en majuscules et au singulier

Identifiant

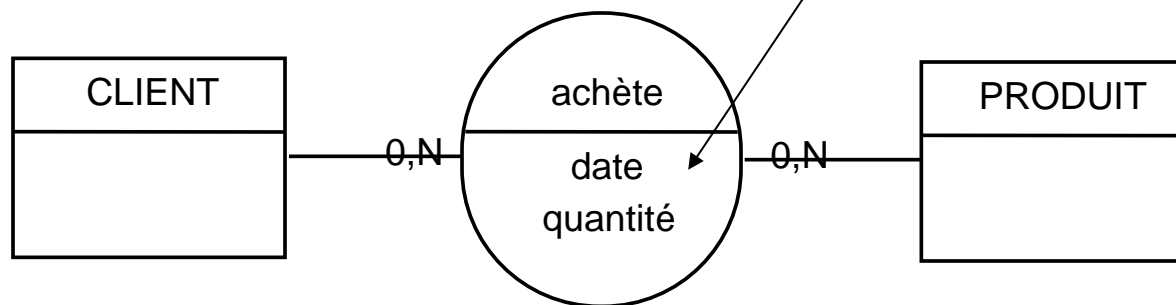
- Attribut(s) de l'entité permettant d'identifier de façon unique chacune des occurrences de l'entité.
- Exemples:
 - Numéro d'assurance sociale du travailleur (NAS) : 222-444-555
 - Numéro de l'employé: 0398
 - Code permanent de l'élève: BALA11090906
 - Numéro du cours: 420-402
 - Nom et prénom du professeur (Martin Blanchet) n'est pas un bon choix d'identifiant puisque l'on peut avoir des doublons: deux professeurs ayant le même nom et prénom.
- L'identifiant doit être présent (obligatoire) et unique (une seule valeur pour l'ensemble des occurrences) afin d'éviter les doublons (deux occurrences ayant le même identifiant).

Association (ou relation)

- Les associations permettent de faire des liens entre les entités. Elles sont régulièrement perçues comme un événement, une **transaction**

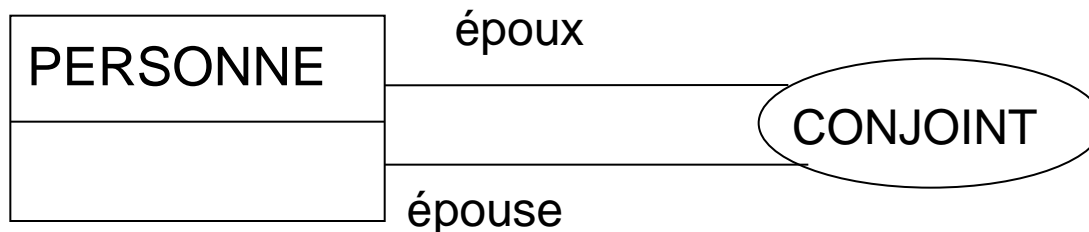


- Normes d'appellation:
Un verbe en minuscules et à l'indicatif présent
(forme active (ex. commande) ou passive (est commandé))
- Les arcs relient l'association aux entités
- Les associations **peuvent** avoir des attributs
(surtout dans les cas de type N:M)



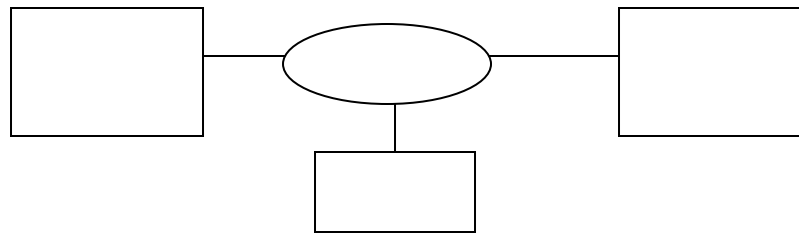
- On peut avoir une classe d'association sur une seule classe d'entités (on parle d'association « réflexive »). On ajoute souvent dans ce cas des noms de **rôles** pour distinguer les deux occurrences.

Ex : CONJOINT(PERSONNE, PERSONNE)



- On peut avoir une classe d'association définie sur n classes d'entités (association n-aire ou d'arité n ou de dimension n ou à « n pattes »).

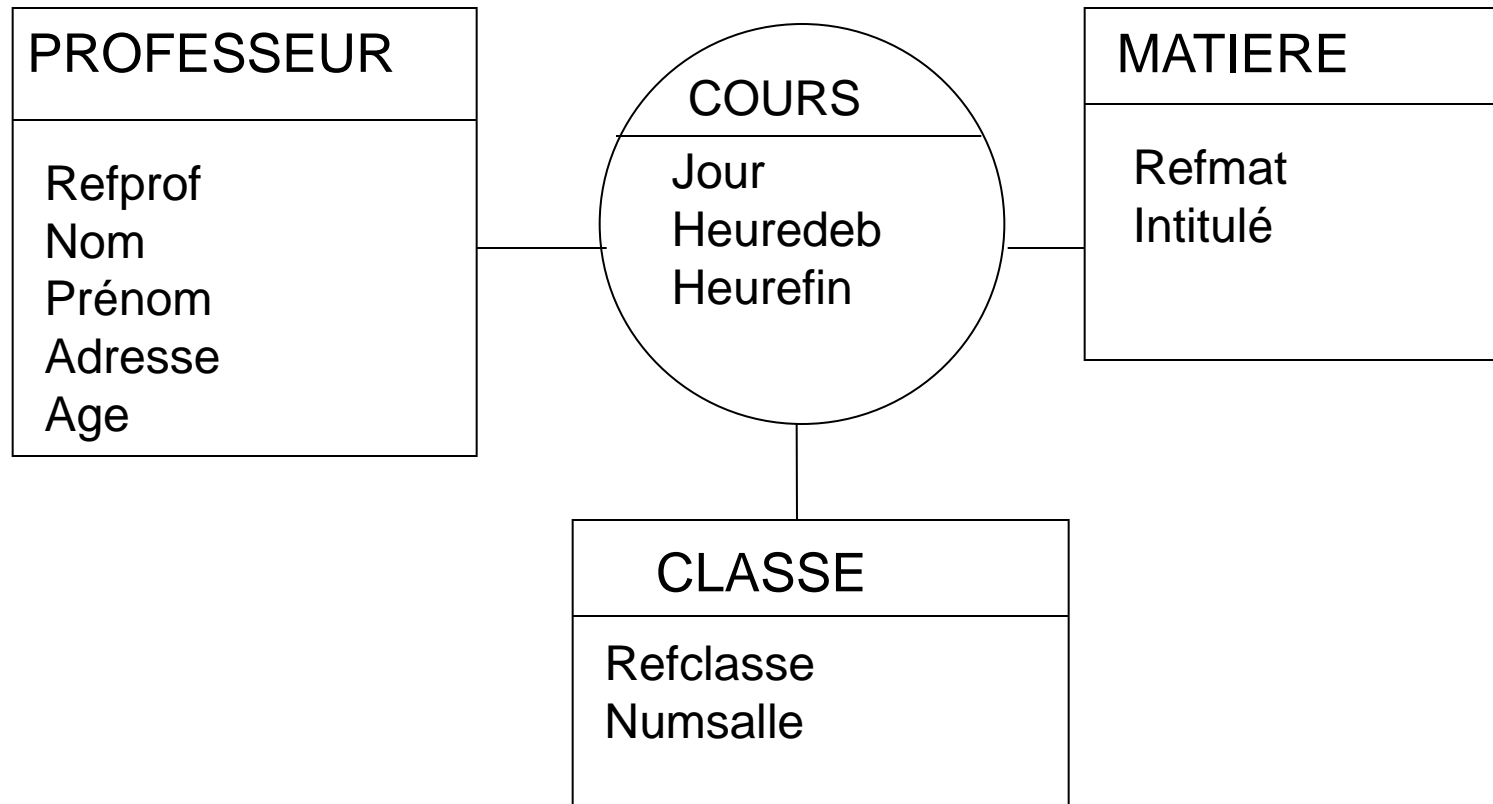
Ex: COURS(MATIERE, CLASSE, PROF)



Attention : les arités élevées sont rares. Elle dénotent souvent des faiblesses dans l'analyse.

arité 2 : 80%
arité 3 : <20%
arité > 3 : ε

Exemple:



Cardinalités

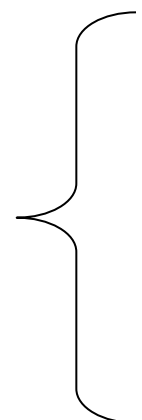
indiquent pour chaque classe d'entités de la classe d'association, les nombres mini et maxi d'occurrences de l'association pouvant exister pour une occurrence de l'entité.

La cardinalité minimum est 0 ou 1.

La cardinalité maximum est 1 ou n.

Une cardinalité minimum à 0 signifie qu'il est possible d'observer (un jour) une occurrence d'entité sans occurrence d'association.

Donc 4 combinaisons possibles :



0,1	au plus 1
1,1	1 et 1 seul
1,n	au moins 1
0,n	un nombre quelconque

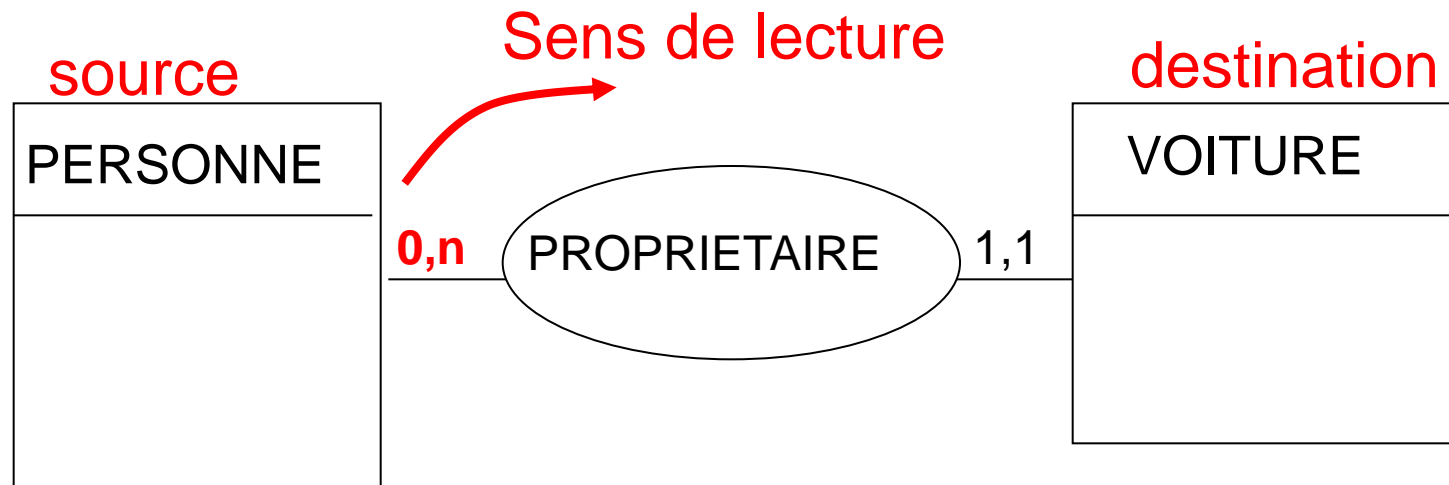
Ex: PROPRIETAIRE(PERSONNE [0,n], VOITURE [1,1])

Une personne a 0 à n voitures; une voiture a 1 et 1 seul propriétaire.

CONDUIT(PERSONNE [0,n], VOITURE [1,n])

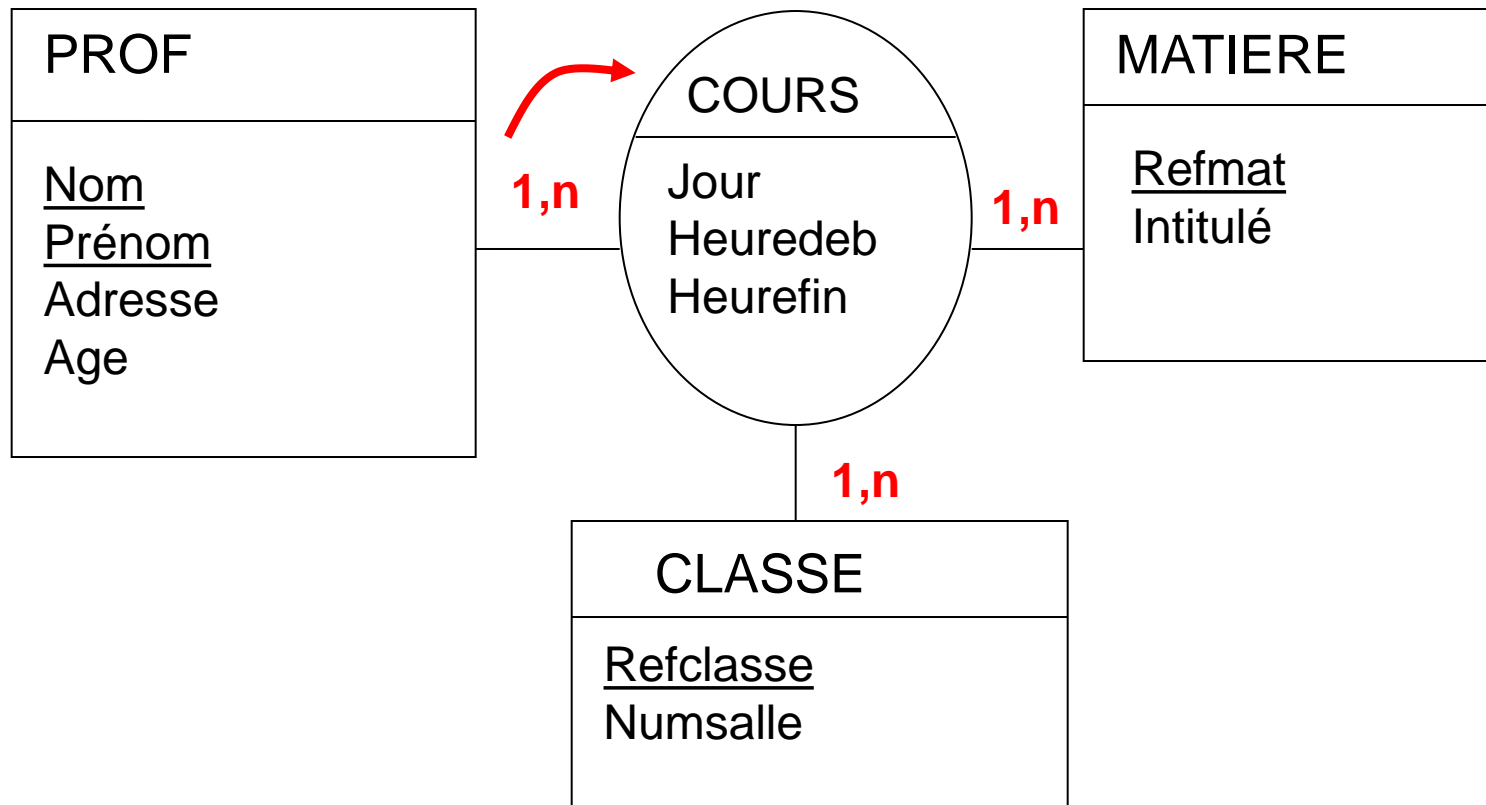
Une personne conduit 0 à n voitures; une voiture est conduite par 1 à n personnes.

Représentation graphique :



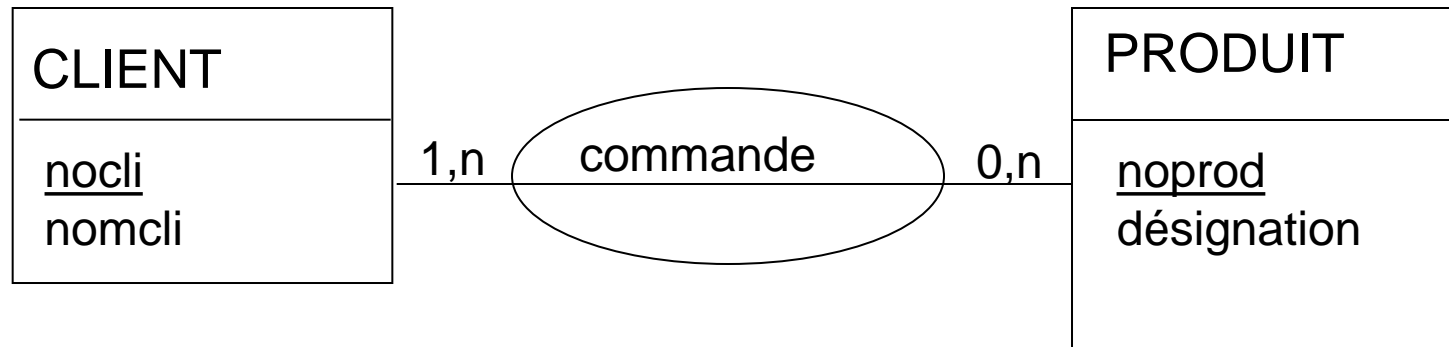
COURS(MATIERE [1,n], CLASSE [1,n], PROF[1,n])

Un prof. a 1 à n cours dans la semaine, une matière a 1 à n cours dans la semaine, une classe a 1 à n cours dans la semaine.



Difficultés : choix entre entité et association ?

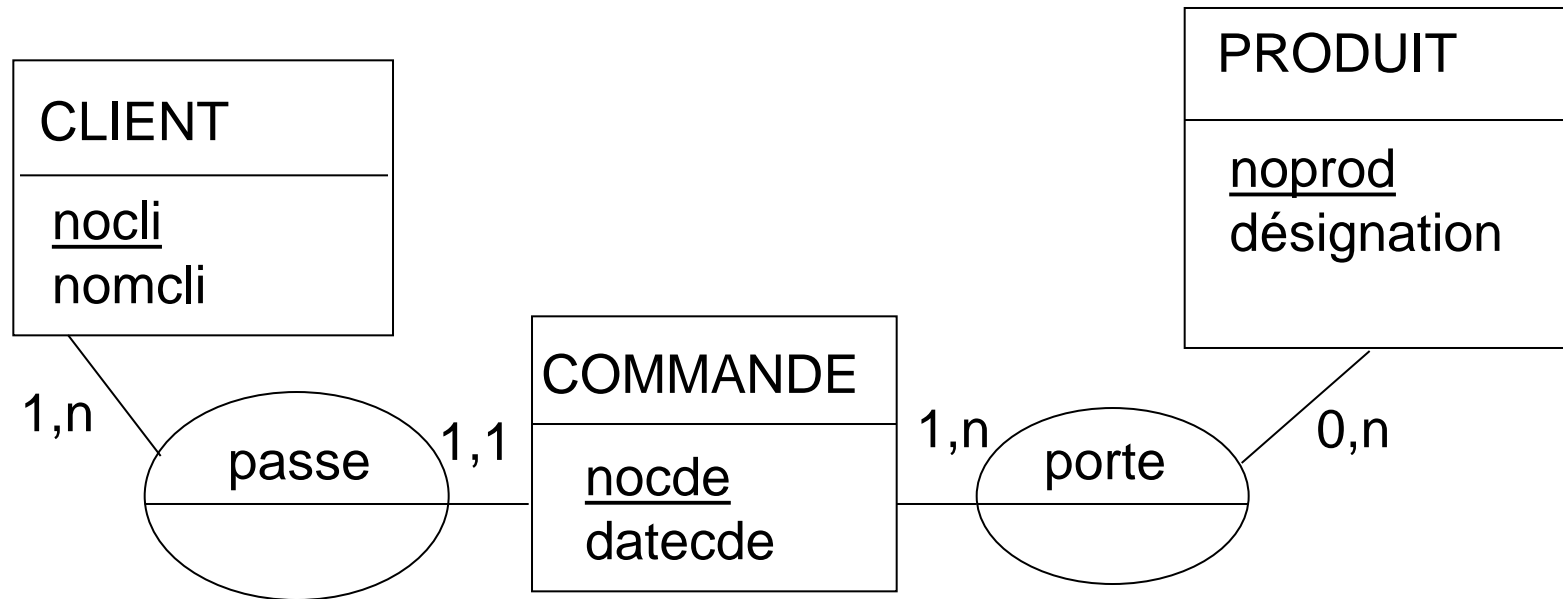
1) Solution avec association



Dans cette première solution la commande n'est pas une entité gérée pour elle même. Elle existe tant que le client et le produit existent.

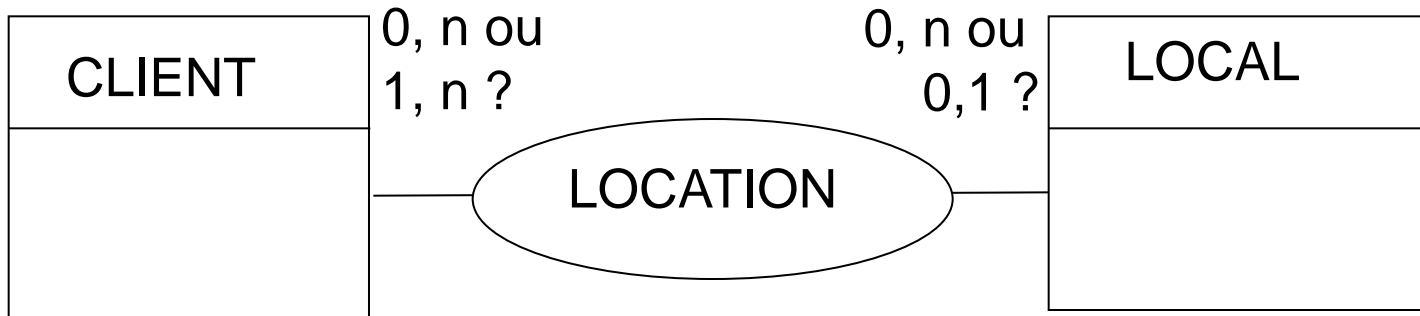
Ce peut être le SI du domaine 'fabrication' : on a juste besoin de savoir que les produits sont destinés à des clients.

2) Solution avec entité



Dans cette seconde solution, les commandes sont identifiées (identifiant nocde) et décrites : on les gère en tant que telles. Elles peuvent être conservées même si le produit ou le client n'existent plus.

Difficultés : choix des cardinalités ?



Un client peut il avoir 0 location ? Est-ce encore un client ?

Un local peut il être loué plusieurs fois ? Non si le local n'est pas partageable. Oui si le local est partageable.

Les cardinalités sont élément essentiel pour définir la sémantique (signification) des données, pas une « décoration » accessoire.

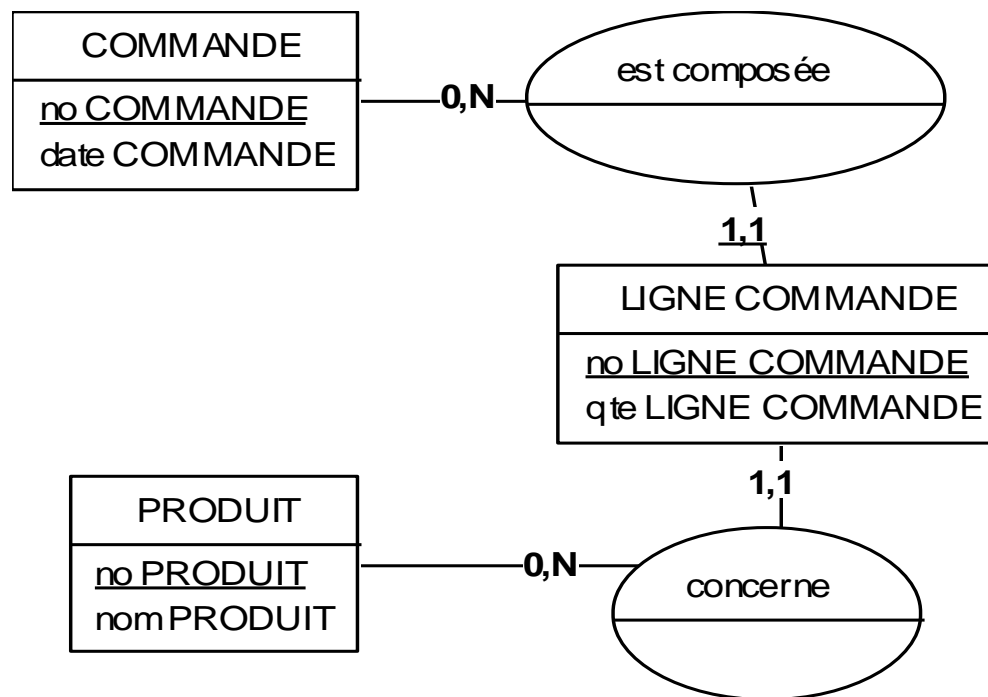
Pour une situation donnée, **il n'existe pas une «solution» unique.**

Le « bon modèle » est celui qui est **accepté par les personnes concernées par le projet.**

b) Dépendance fonctionnelle

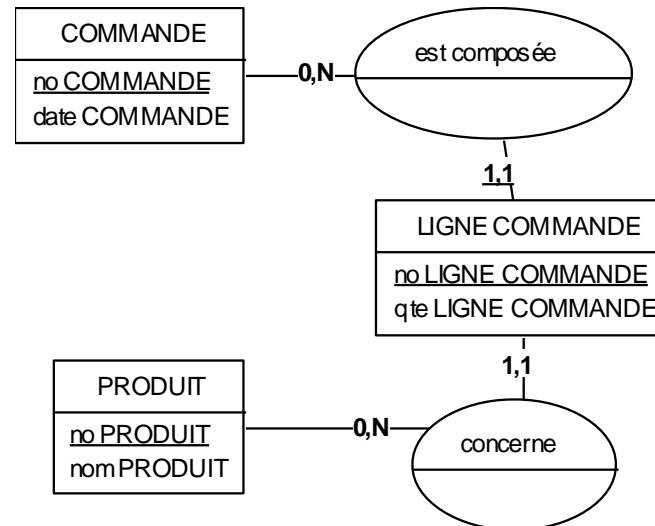
- La **dépendance fonctionnelle**

s'exprime en utilisant les connectivités 1,1 ou 0,1. On parle de dépendance fonctionnelle forte pour 1,1 et faible pour 0,1.



- L'identifiant **no LIGNE COMMANDE** pourrait avoir besoin du **no COMMANDE** pour être unique.

COMMANDE				
				Commande no : 999
Date :	_____			
Numéro du client :	_____			
Nom du client:	_____			
Numéro Produit	Description	Quantité commandée	Prix	Total
1. _____	_____	_____	_____	_____
2. _____	_____	_____	_____	_____
3. _____	_____	_____	_____	_____
4. _____	_____	_____	_____	_____
5. _____	_____	_____	_____	_____
6. _____	_____	_____	_____	_____
TOTAL GLOBAL :				_____



No Commande	No Ligne Com	No Produit
1000	1	22
1000	2	33
1000	3	65
1111	1	23
1111	2	33

L'identifiant implicite de l'entité dépendante est donc composé du numéro de la commande et du numéro de la ligne de commande.

1000-1
1000-2

...
1111-2

c) Vérification et Normalisation

Contrôler la qualité du modèle vis-à-vis :

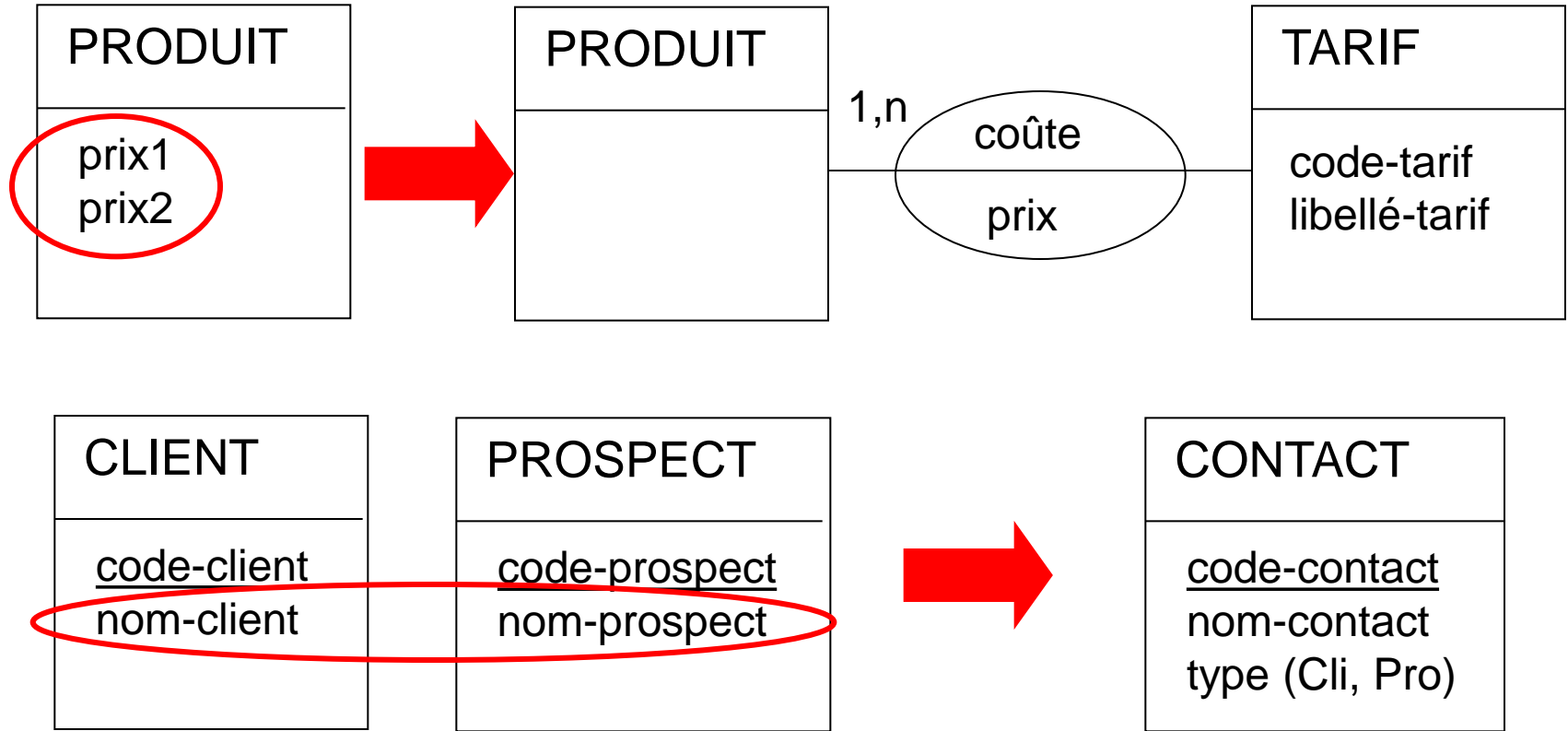
- des fondements du modèle d'une part (règles de vérification),
- de la redondance de données d'autre part (règles de normalisation) .

Permet de détecter certaines incohérences dans la construction des modèles.

1. Règles Générales

- Toute propriété doit apparaître une seule fois dans un modèle.

Il faut éliminer la redondance des propriétés dans la même entité (avec des noms différents) ou dans des entités distinctes :



Pas d'héritage dans le modèle E/A de base !

- Toutes les propriétés identifiées doivent apparaître dans le modèle.

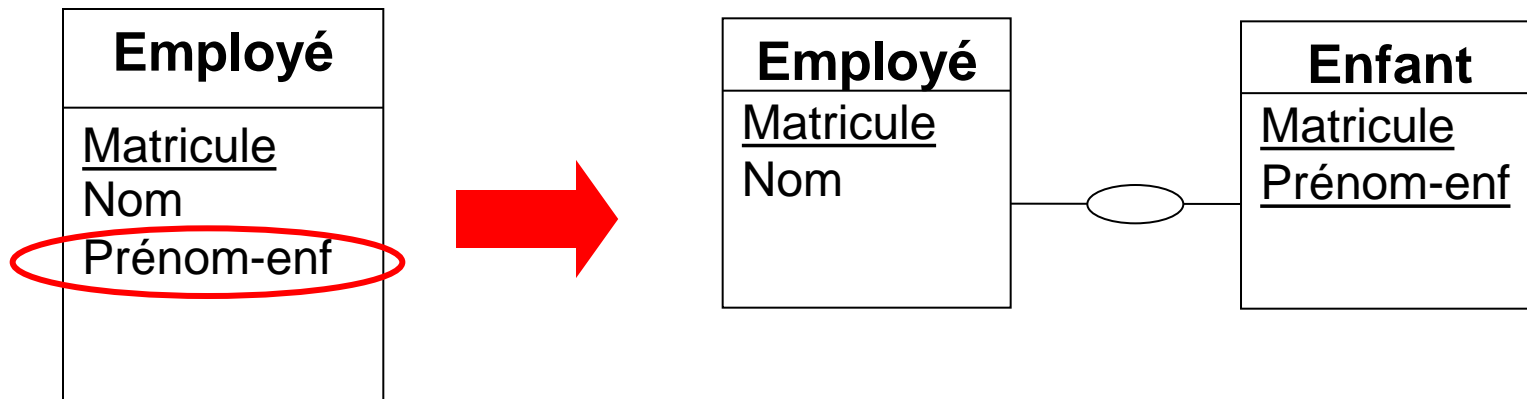
2. Règles sur les entités

2.a Règle de l'identifiant

Toutes les entités ont un identifiant.

2.b Règle de vérification des entités

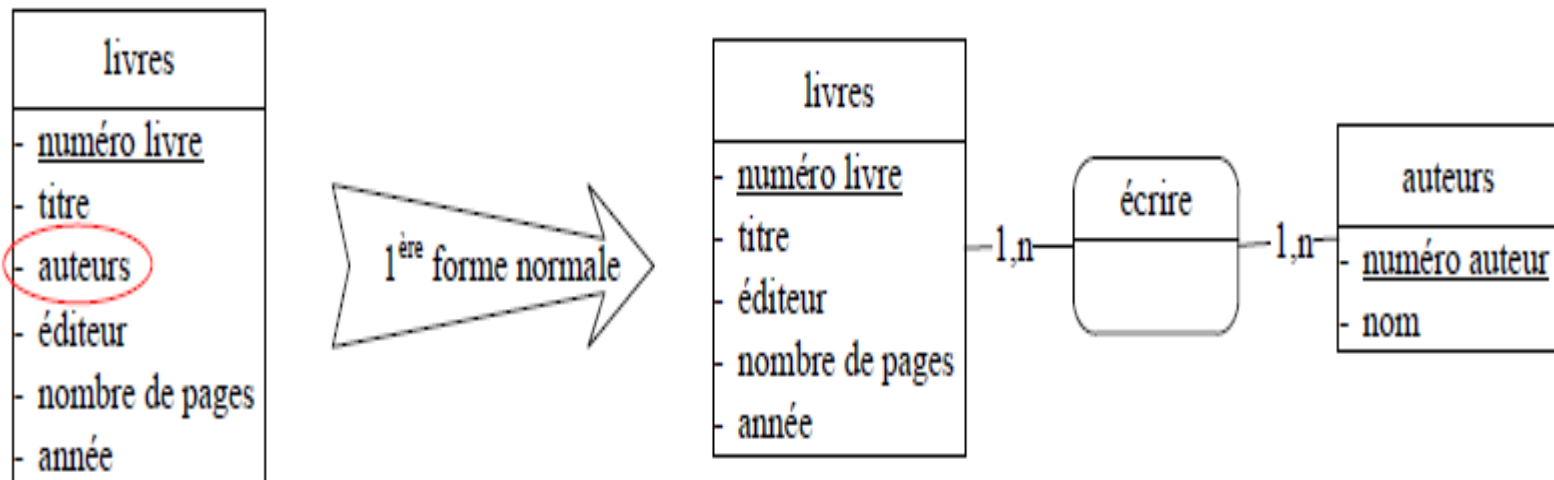
Pour une occurrence d'une entité, chaque propriété ne prend qu'une seule valeur (cf. la 1FN du modèle relationnel); MONO-VALUEE



On décompose l'entité Employé en deux entités : Employé, et Enfant

Règles de normalisation: 1FN

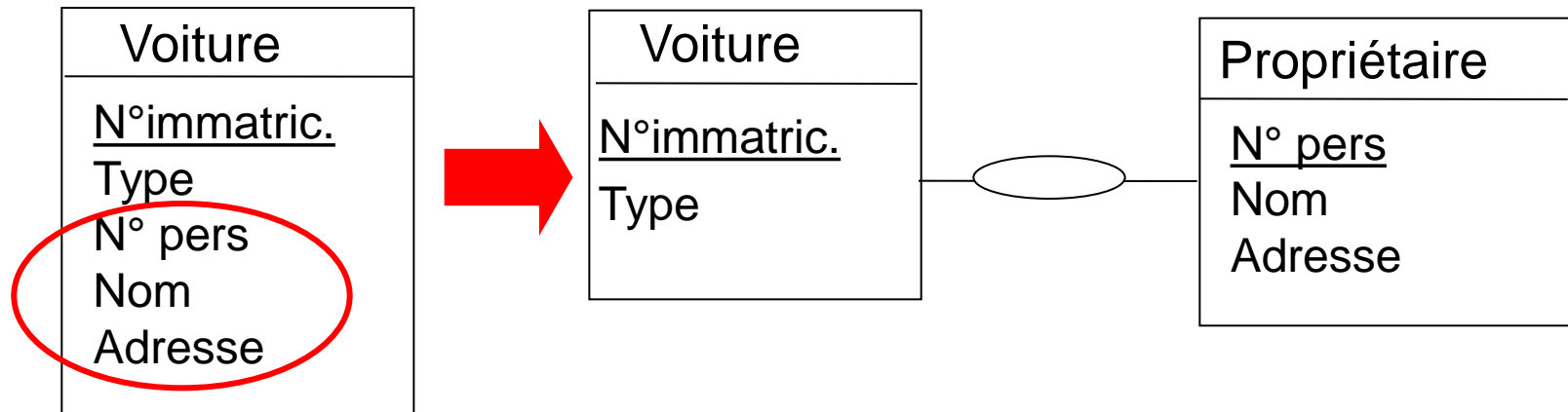
A un instant donné dans une entité, pour un attribut ne peut prendre qu'une valeur et non pas, un ensemble ou une liste de valeurs. Si un attribut prend plusieurs valeurs, alors ces valeurs doivent faire l'objet d'une entité supplémentaire, en association avec la première.



2.c Règles de normalisation des entités

a) Les dépendances fonctionnelles (DF) entre les propriétés d'une entité doivent vérifier la règle suivante : **toutes** les propriétés de l'entité dépendent fonctionnellement de l'identifiant **et uniquement** de l'identifiant.

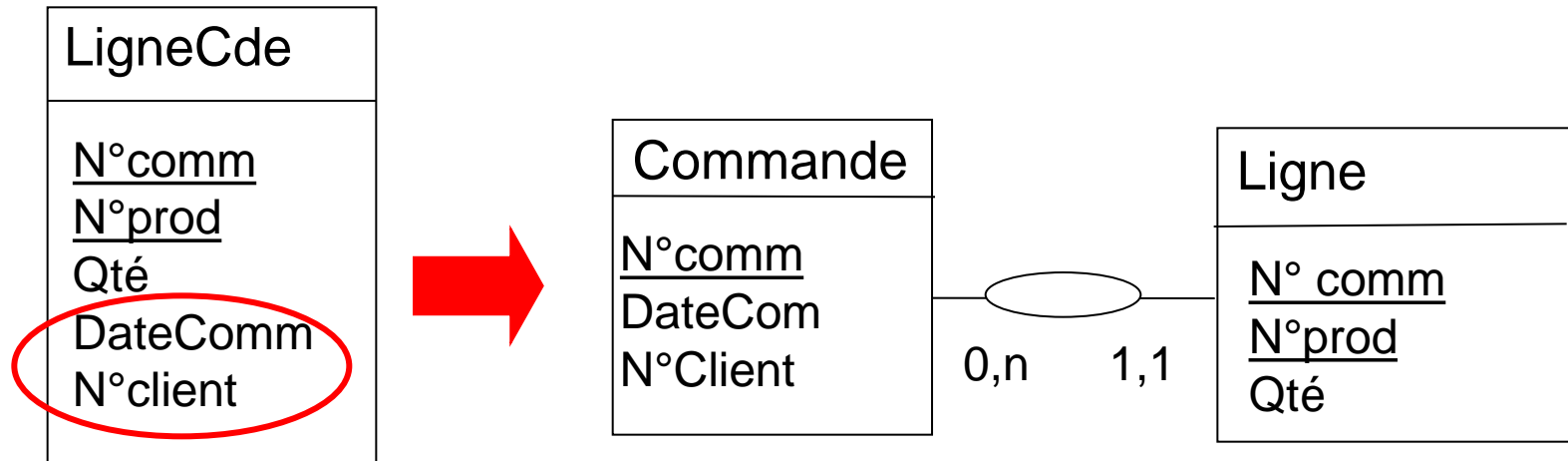
Rappel : \exists une DF $X \rightarrow Y$ si à une valeur de X correspond une et une seule valeur de Y (réciproque pas vraie).



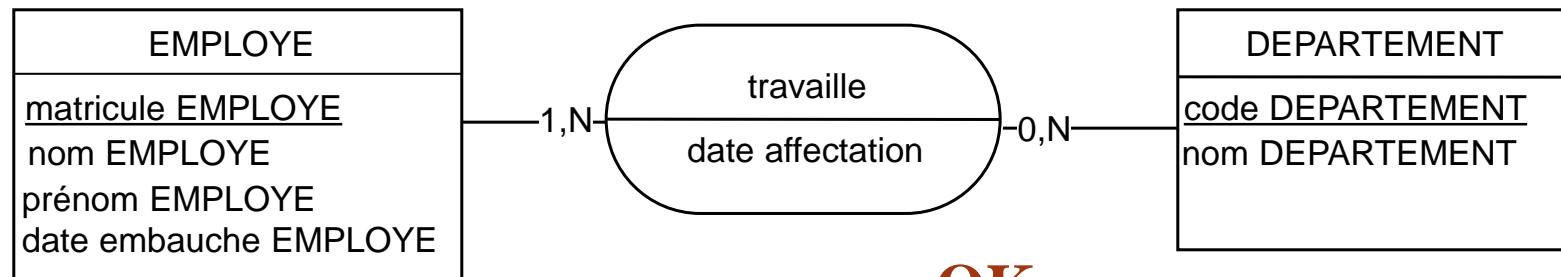
Règles de normalisation: 2FN

La deuxième forme normale (2FN) ne s'applique que sur des entités déjà en première forme normale. Elle stipule que lorsque l'identifiant primaire est composé, tous les attributs doivent fonctionnellement dépendre complètement de l'identifiant primaire et non d'une partie seulement de celui-ci.

b) Une partie de l'identifiant ne peut pas déterminer certaines propriétés.



La DF $n^{\circ}\text{-comm} \rightarrow \text{date-comm}, n^{\circ}\text{-client}$ contredit la règle. On décompose l'entité LigneCde en deux entités.



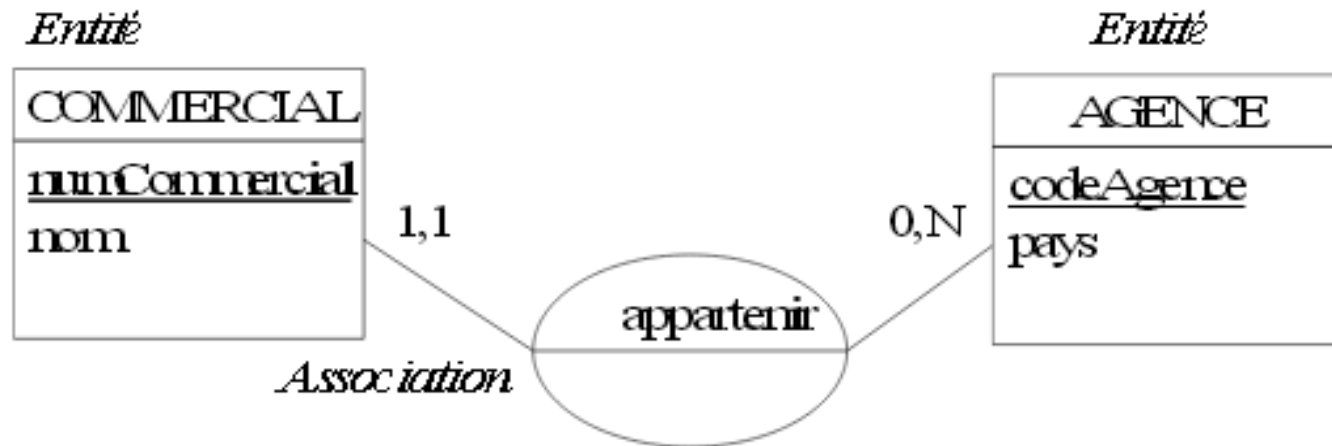
OK

Règles de normalisation: 3FN

La troisième forme normale (3FN) ne s'applique que sur des entités déjà en deuxième forme normale. Elle stipule qu'il ne doit pas y avoir de dépendance transitive à l'intérieur d'une entité. Ceci revient à dire qu'il ne doit pas y avoir de sous-entités dans une entité.

Exemple: COMMERCIAL (numCommercial, nom, codeAgence, pays) ?

La modélisation correcte aurait dû être la suivante:



et ainsi les tables produites :




COMMERCIAL (numCommercial, nom, #codeAgence)

AGENCE (codeAgence, pays)

Exemple

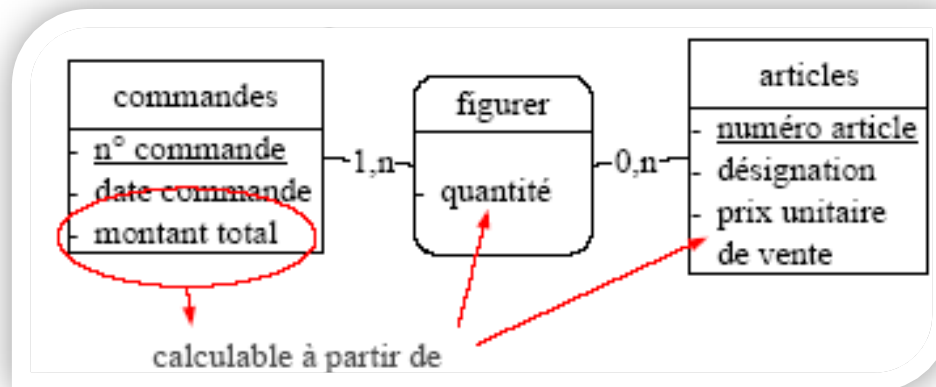
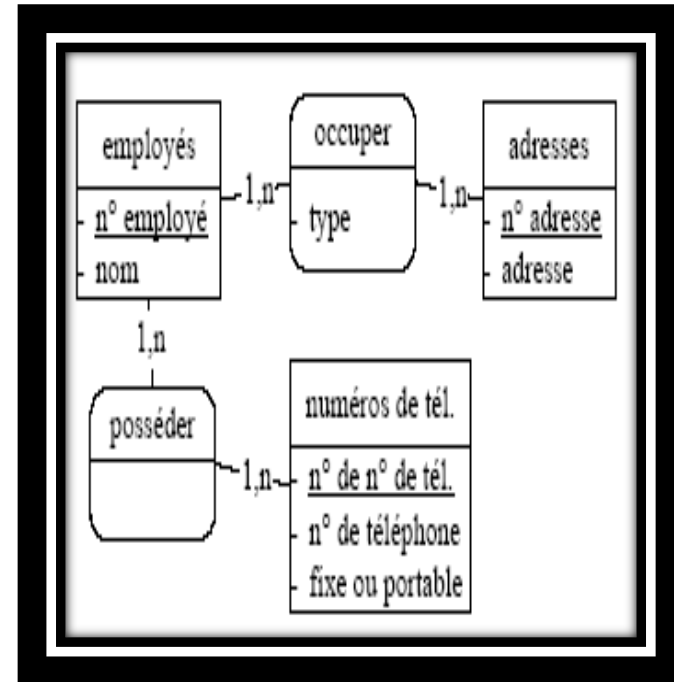
Nom de la donnée	Format	Longueur	Type		Règle de calcul	Règle de gestion	Document
			Elémentaire	Calculée			
NumCli	Numérique		x				Facture
Nom:	Alphabétique	30	x				Facture
Prénom:	Alphabétique	30	x				Facture
Adresse:	Alphanumérique	60	x				Facture
Code postal	Numérique		x				Facture
Ville	Alphabétique	20	x				Facture
Téléphone:	Alphanumérique	14	x				Facture
CodeArticle	Alphanumérique	15	x				Facture
Désignation	Alphabétique	50	x				Facture
Quatité	Numérique		x				Facture
Prix unitaire	Numérique		x				Facture
Date	date		x				Facture
Total ligne	Numérique			x	Prix* Qté		Facture
Total facture	Numérique			x	Somme des Total Ligne		Facture

Exemple

- NumCli  (Nom, Prénom, Adresse, Code Postal, Ville)
- CodeArticle  (Désignation, Prix unitaire)
- (NumCli, CodeArticle, Date)  Quantité

Exemple

employés
- <u>n° employé</u>
- nom
- adresse principale
- adresse secondaire
- n° téléphone domicile
- n° téléphone portable



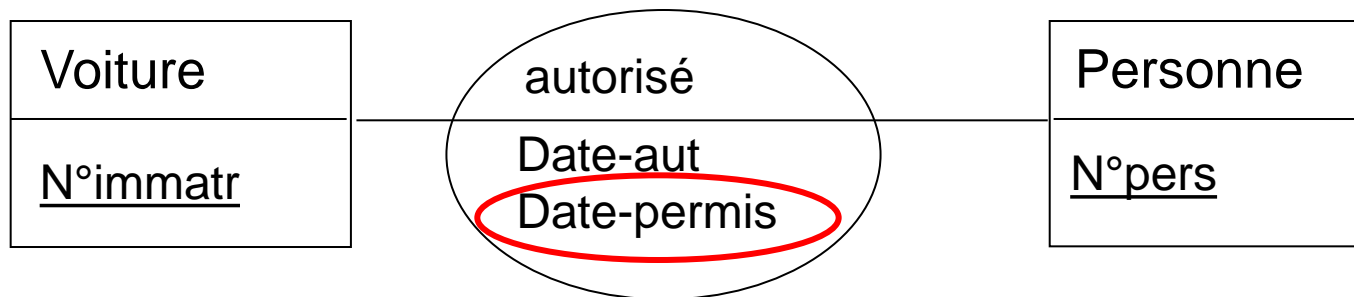
3. Règles sur les associations

3.a Règle de vérification des associations

Pour une occurrence d'association, chaque propriété ne prend qu'une seule valeur.

3.b Règle de normalisation sur les propriétés des associations

Toutes les propriétés de l'association doivent dépendre fonctionnellement de tous les identifiants des entités portant l'association, et uniquement d'eux.

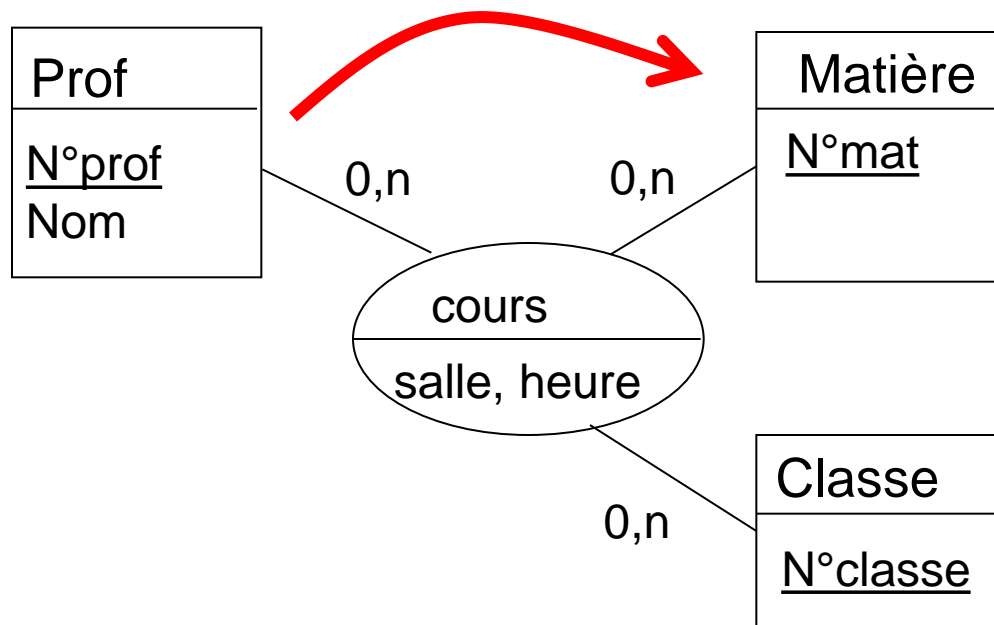


$N^{\circ}\text{-pers} \rightarrow \text{Date-permis}$ pose problème (donc déplacer Date-permis vers Personne)

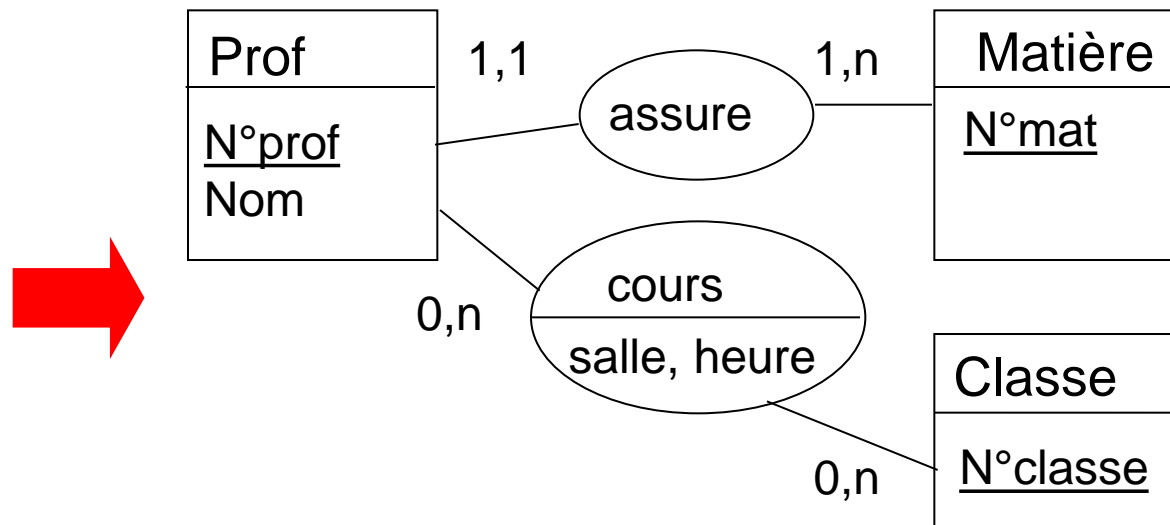
3.c La décomposition des associations n-aires

Il faut garder un minimum d'associations d'arité > 2 .

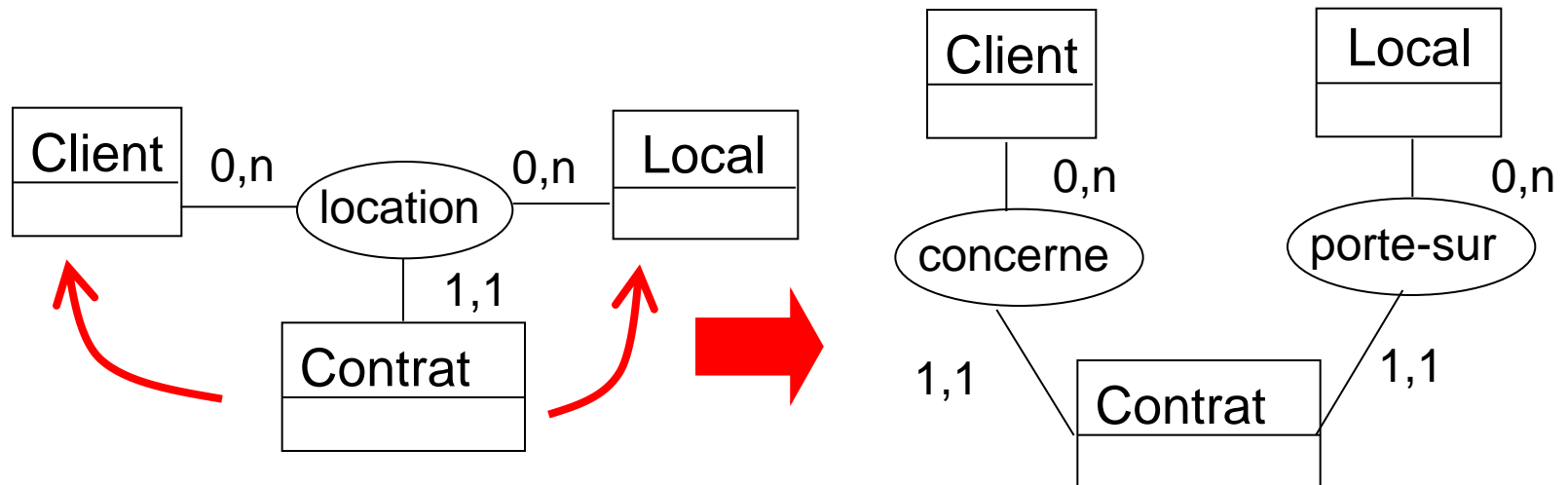
Si on observe une DF entre deux identifiants, on peut décomposer l'association n-aire.



Une éventuelle DF $N^{\circ}prof \rightarrow N^{\circ}mat$ (c.à.d. si un prof enseigne une seule matière) conduit à la décomposition :

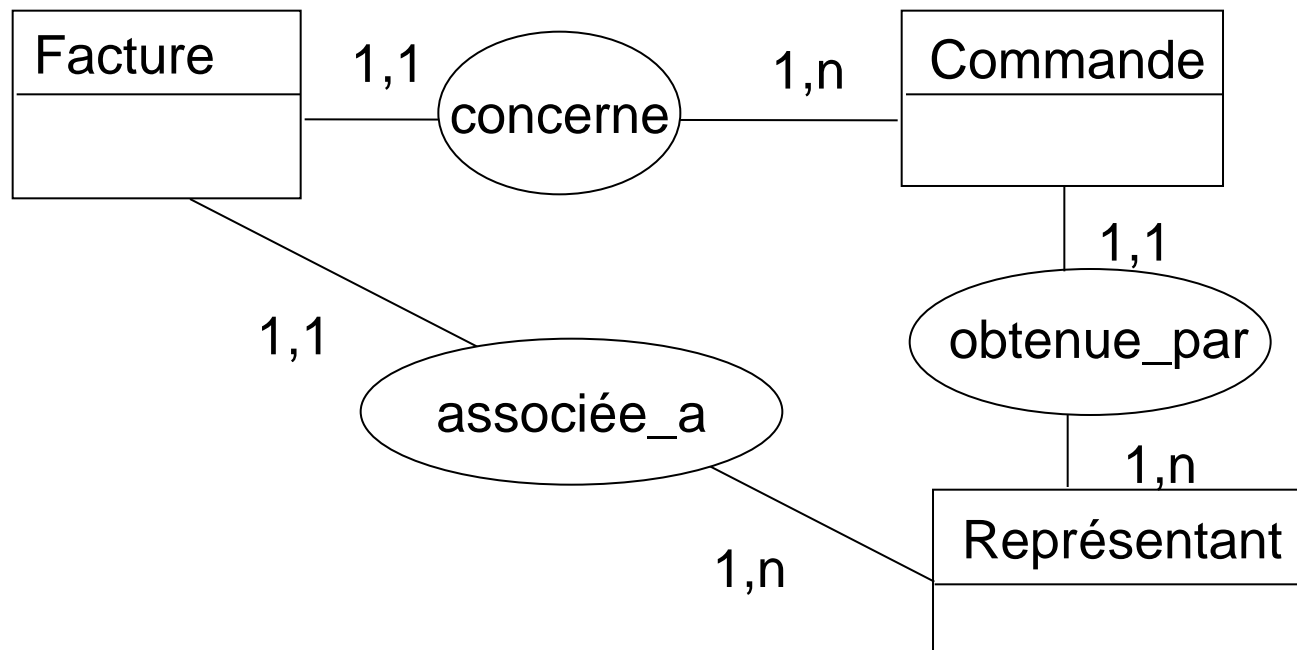


C'est le cas, quand une patte a une cardinalité 1,1.
 Par exemple à 1 contrat est associé un client et un local :



3.d La suppression des associations transitives

Toute association pouvant être obtenue par transitivité de n autres associations peut être supprimée. La transitivité s'évalue en fonction de la **signification** des associations.



On supprime l'association *associée_a*, car elle peut être obtenue par transitivité sur les associations *concerne* et *obtenue_par*

c) Quelques contraintes d'intégrité importantes

Les CI définissent des propriétés qui doivent être vérifiées par les données de la base.

1. Contraintes intégrées au modèle E/A

1.a *Contrainte d'identifiant*

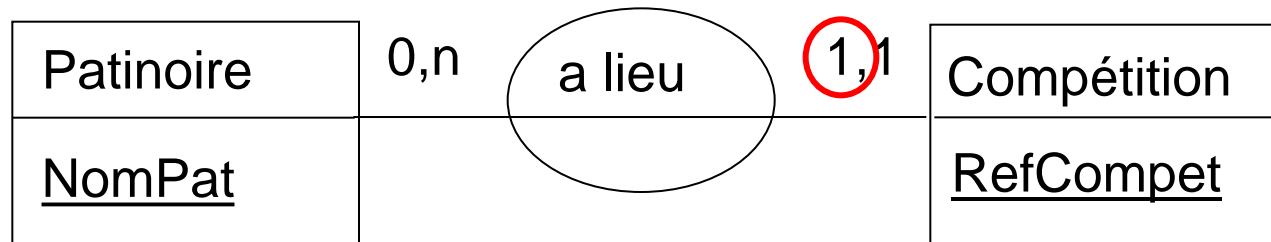
Les valeurs prises par l'identifiant sont uniques (dans le temps) et toujours définies.

Ex : identifiant de l'entité PERSONNE

- Nom + prénom, NAS pas suffisant
- N° téléphone pas stable dans le temps

1.b *Contraintes de cardinalité*

Les cardinalités portées par les entités membres d'association imposent des nombres mini et maxi d'occurrence dans l'association.



Une cardinalité mini de 1 rend l'existence d'une occurrence d'entité dépendante de l'existence d'une occurrence d'une autre entité.

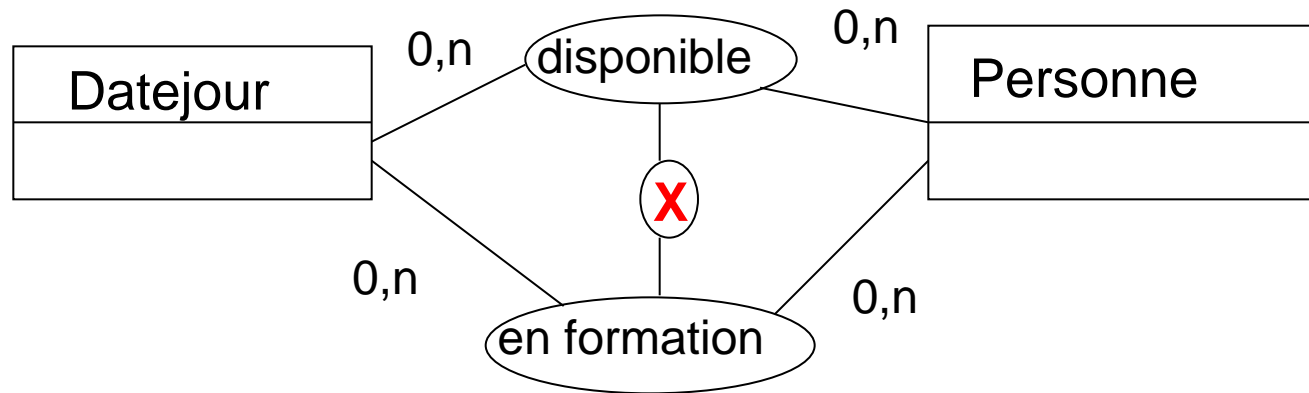
Une compétition de patinage ne peut exister que si la patinoire où elle se déroule existe.

Une patinoire peut exister de manière indépendante de toute compétition.

2. Contraintes extensions du modèle E/A

Exclusion de participation entre associations

Il y a exclusion de participation entre associations si la participation des entités à l'association A1 exclut leur participation à l'association A2.



Une personne à une même date ne peut pas figurer simultanément dans les deux associations: *disponible* et *en formation*.