

Le Modèle Entité Association

HLIN511

Pascal Poncelet

Pascal.Poncelet@umontpellier.fr
<http://www.lirmm.fr/~poncelet>



Organisation du cours

Introduction

Le modèle Entité-Association

Un exemple

Transformation vers le modèle relationnel



Introduction

- Objectif de la conception : représenter la réalité telle qu'elle est perçue par les utilisateurs
 - Contrairement aux modèles logiques qui décrivent la réalité en fonction du modèle du SGBD
-
- Représentation à l'aide de la trilogie de base
 - objets – liens - propriétés
 - Attention portée sur les applications
 - Indépendante des technologies
 - Portabilité
 - Longévité



Introduction

- Orientée utilisateur
 - Compréhensibilité
 - Support du dialogue concepteurs / utilisateurs
- Permet la collaboration et la validation par les utilisateurs
- Facilite les échanges d'informations entre SGBD différents
- La qualité de la conception de la BD est un facteur critique de réussite



Cycle de vie d'une application

Monde
réel



utilisateurs

LMD

Personne
Voiture

Schéma conceptuel



Schéma logique

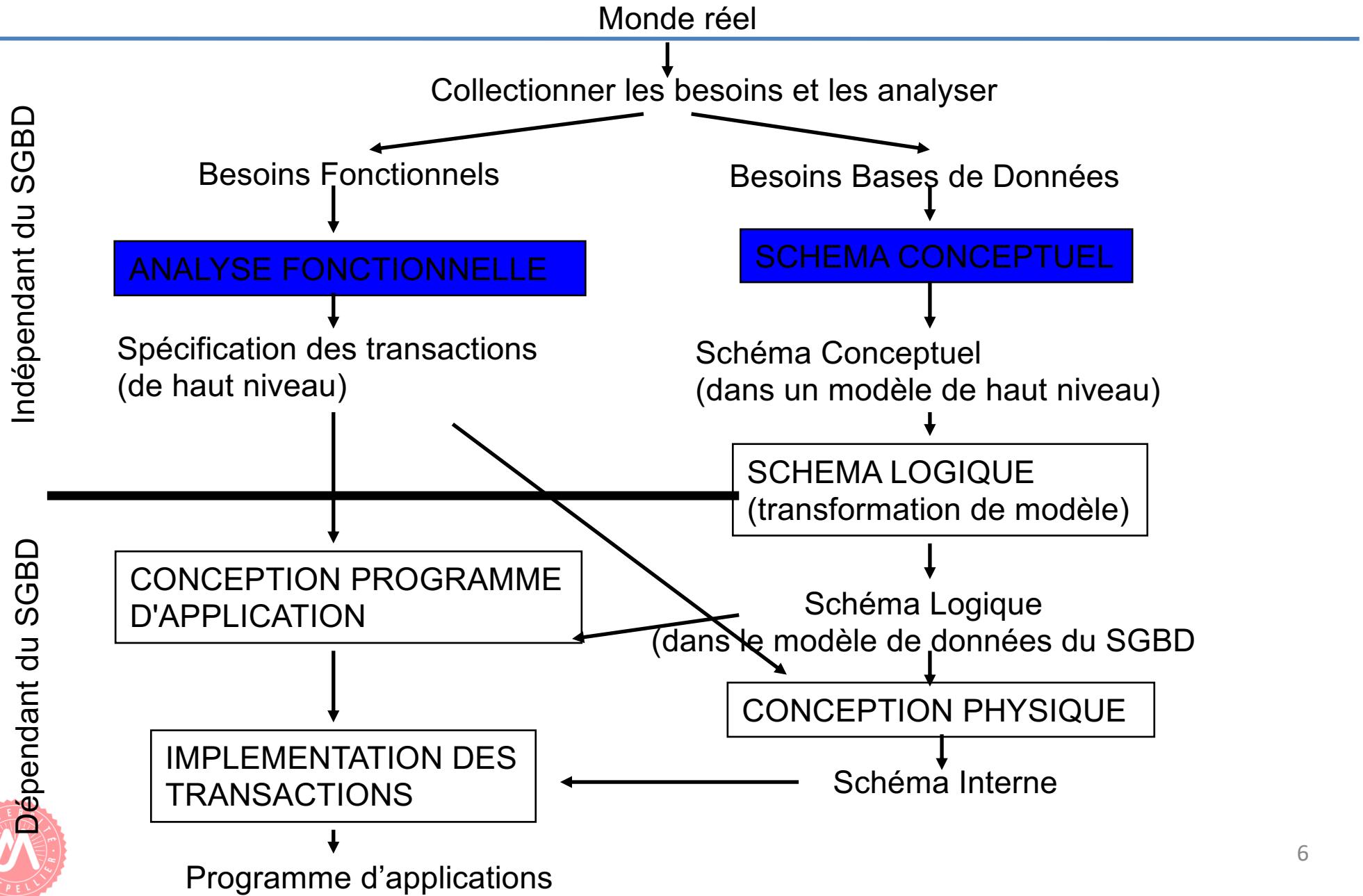
Table Personne
Table Voiture
Table Conduit
...

BD

Personne
Dupont
Durant
Dujardin

Voiture
Peugeot 308
Renault 25
Toyota Kya
...

Cycle de vie d'une application



Introduction

- Quelques exemples de modèles conceptuels
 - Entité Association (ER: Entity-Relationship)
 - UML
 - Autres (OO, OR)
- Attention le modèle relationnel et certains modèles orienté objets sont des modèles logiques (objectif : implémentation)



Organisation du cours

Introduction

Le modèle Entité-Association

Un exemple

Transformation vers le modèle relationnel



Le modèle Entité-Association

- Modèle Entité-Association élaboré par Chen [Chen76]
 - « The Entity Relationship Model - Toward a Unified View of Data », TODS, March 1976
- Modèle pour la conception des bases de données : aspect données
- Modèle sémantique, modèle conceptuel, ...
- Pas d'implantationmais utilisé dans des AGL

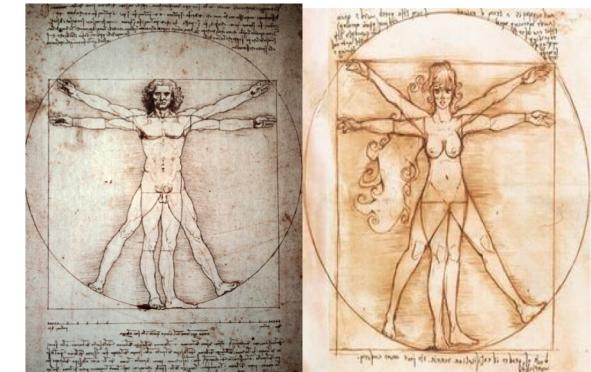


Entité-Association

- **Entité** : représentation d'un objet du monde réel ayant une existence propre
- **Type d'entité (TE) ou Classe d'entité** : représentation d'un ensemble d'entités perçues comme similaires et ayant les mêmes caractéristiques



Dupond Durand Dujardin



Personne



Par abus de langage, Entité = Type d'entité

Entité-Association

- **Association** : représentation d'un lien non orienté entre plusieurs entités (qui jouent un **rôle** déterminé)
- **Type d'association (TA) ou classe d'association** : représentation d'un ensemble d'associations ayant la même sémantique et décrites par les mêmes caractéristiques



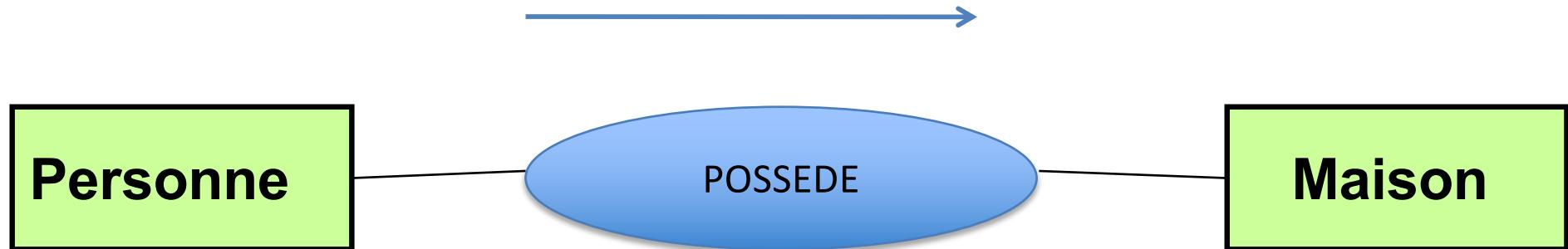
« 1 personne possède 1 maison »

« 1 maison est achetée par une personne »

Par abus de langage, Association = Type d'association

Rôles dans une association

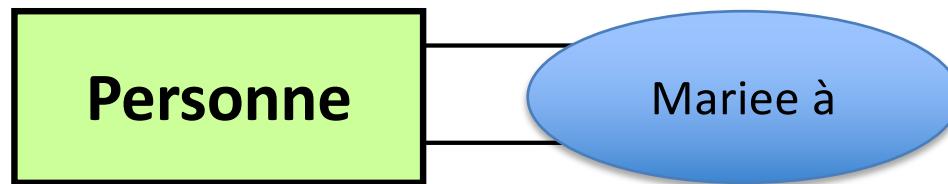
- L'association se reporte dans son nom (un verbe)



- S'il n'y a pas d'ambiguïté il est possible de ne mettre qu'un seul rôle dans l'association mais dans ce cas on privilégie une lecture

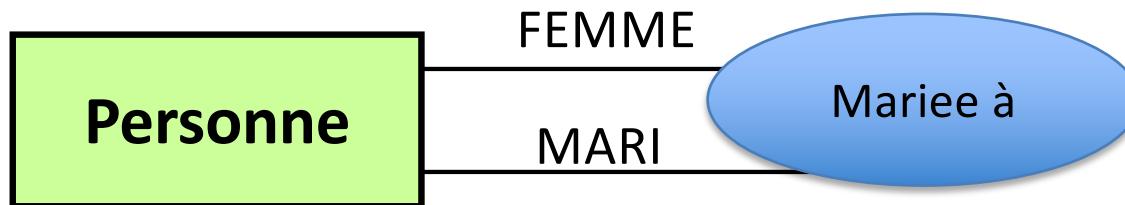
Rôles dans une association

- Les associations peuvent être cycliques



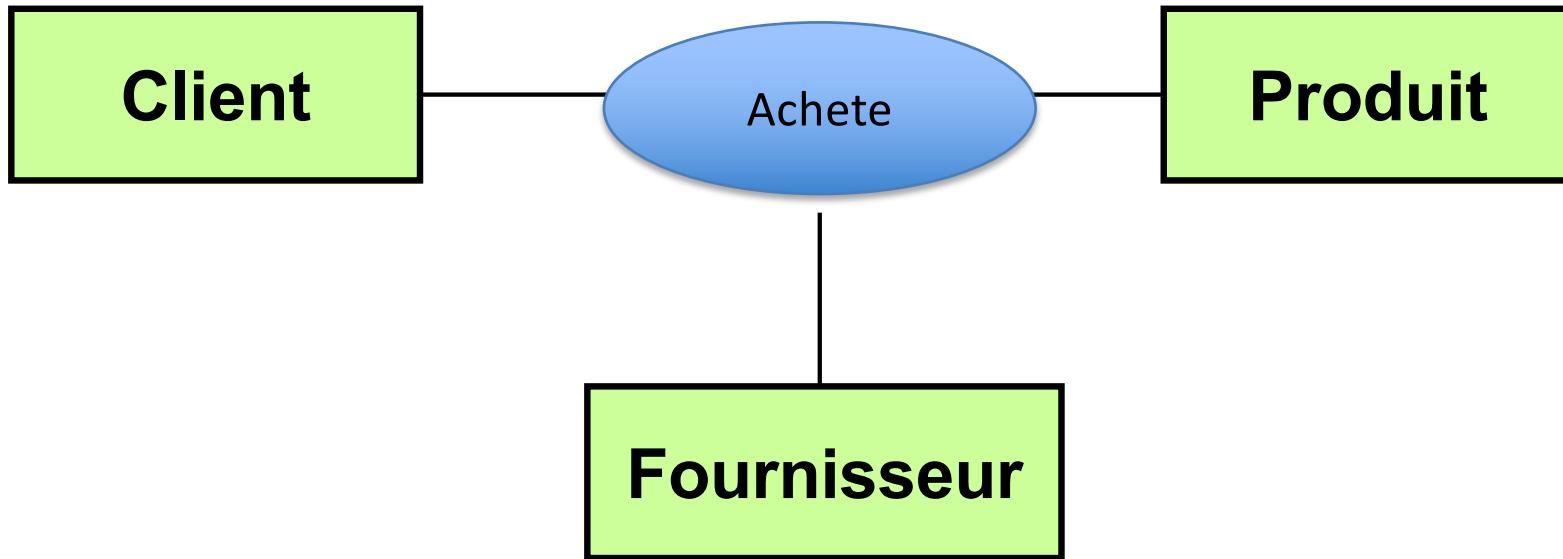
'marié à' = < 1 personne, 1 personne >

- Les cycles posent des problèmes d'ambiguïté : comment savoir dans un couple qui est le mari, qui est la femme ?

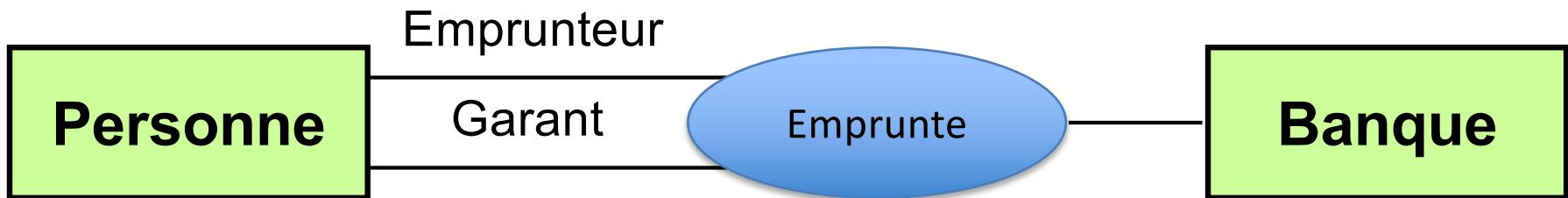


« mariée à » = < 1 personne/FEMME, 1 personne/MARI >

Associations Ternaires



'achète' = < 1 client, 1 produit, 1 fournisseur >



Cardinalités

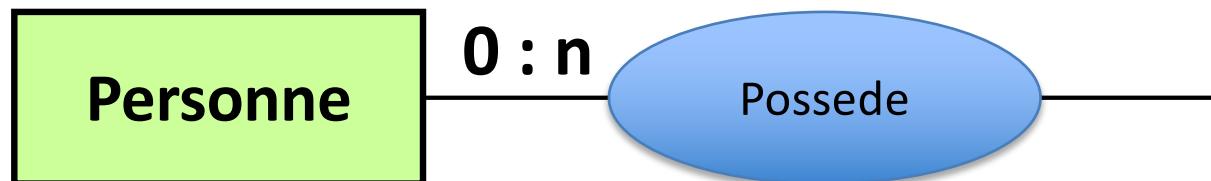


- Combien de voitures (minimum) une personne peut-elle avoir ?
- Combien de voitures (maximum) une personne peut-elle avoir ?



Cardinalités

- Une personne peut ne pas avoir de voiture, en avoir 1, 2, ... n (pas de contraintes)

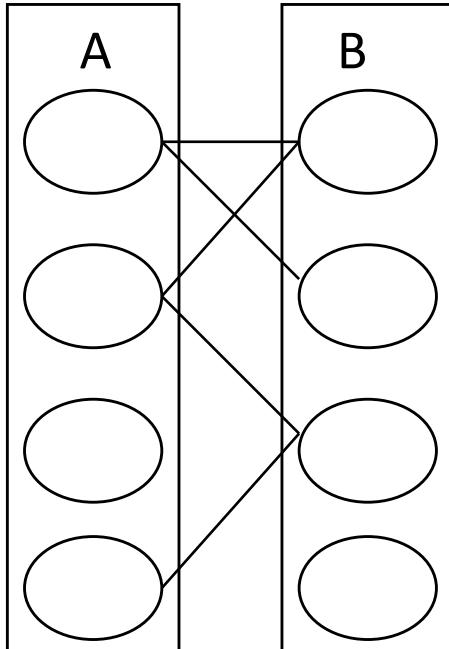


- Une voiture peut avoir un et un seul propriétaire



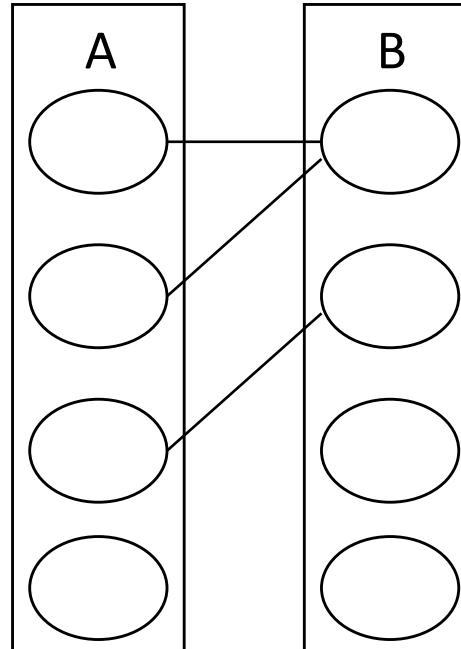
- Cardinalités : contraintes sur les données

Cardinalités

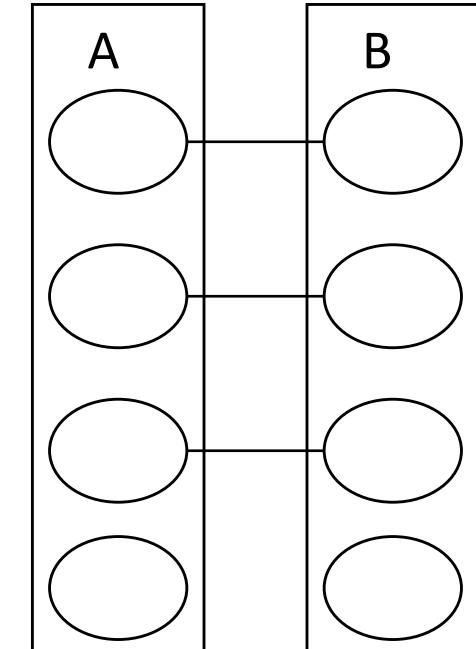


N-M
0,n - 0,n

1 instance de A peut être liée à plusieurs instances de B et réciproquement



N-1
0,n - 0,1

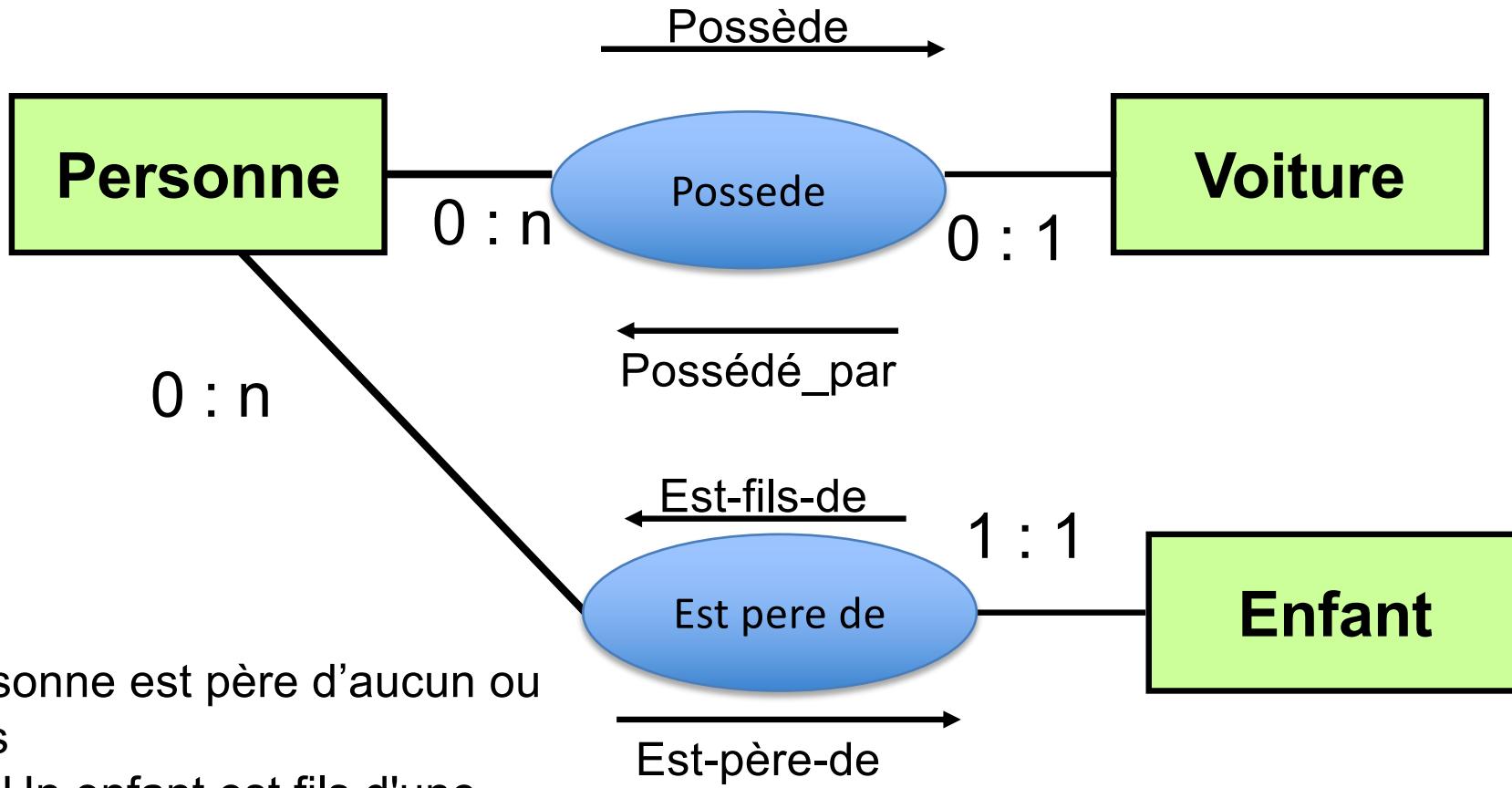


1-1
0,1 - 0,1

1 instance de A ne peut être liée qu'à 1 seule instance de B et réciproquement

Cardinalités

Une personne possède aucune ou plusieurs voitures. Une voiture peut être possédée par au plus une personne

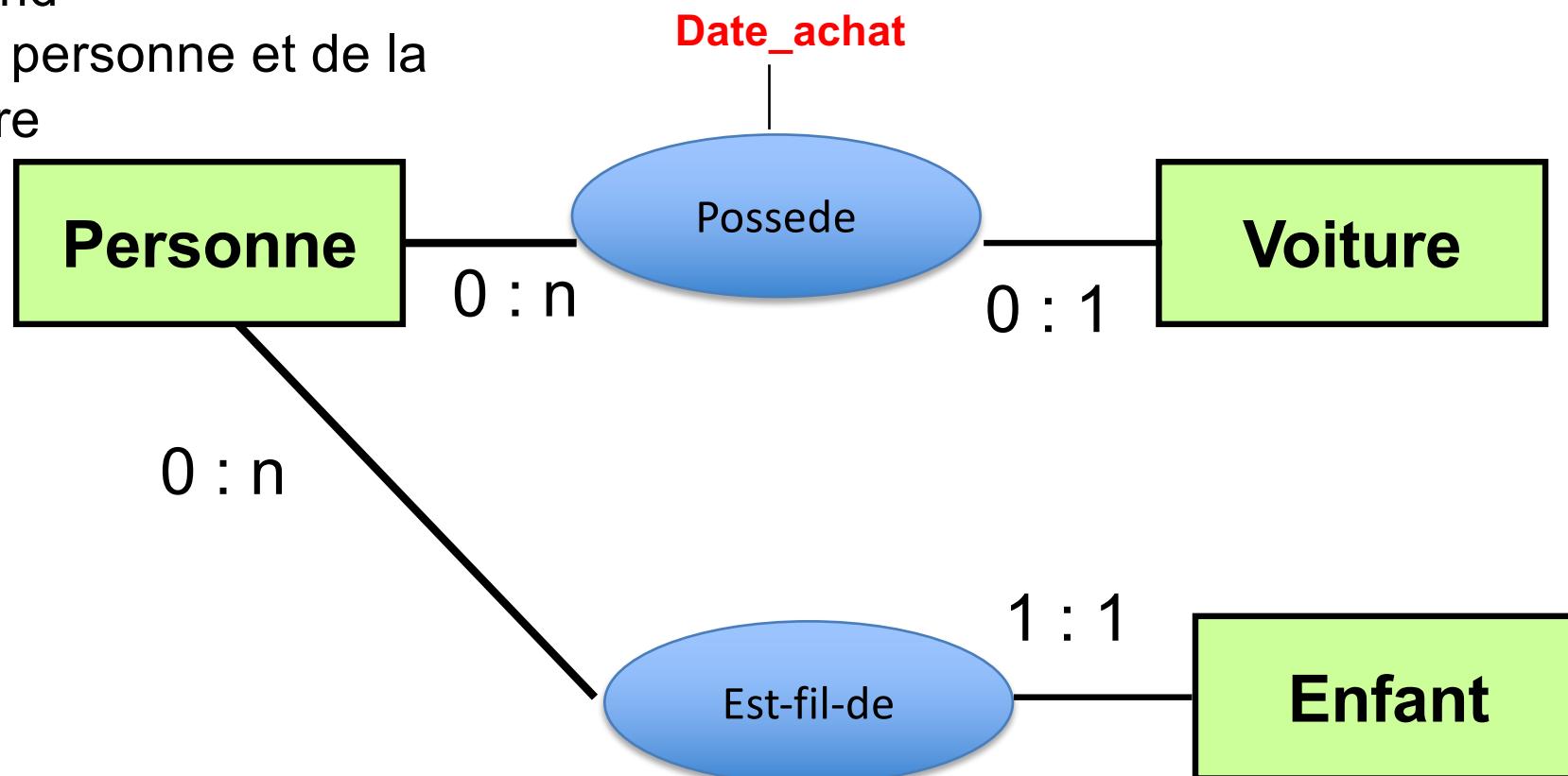


Une personne est père d'aucun ou plusieurs enfants. Un enfant est fils d'une seule personne



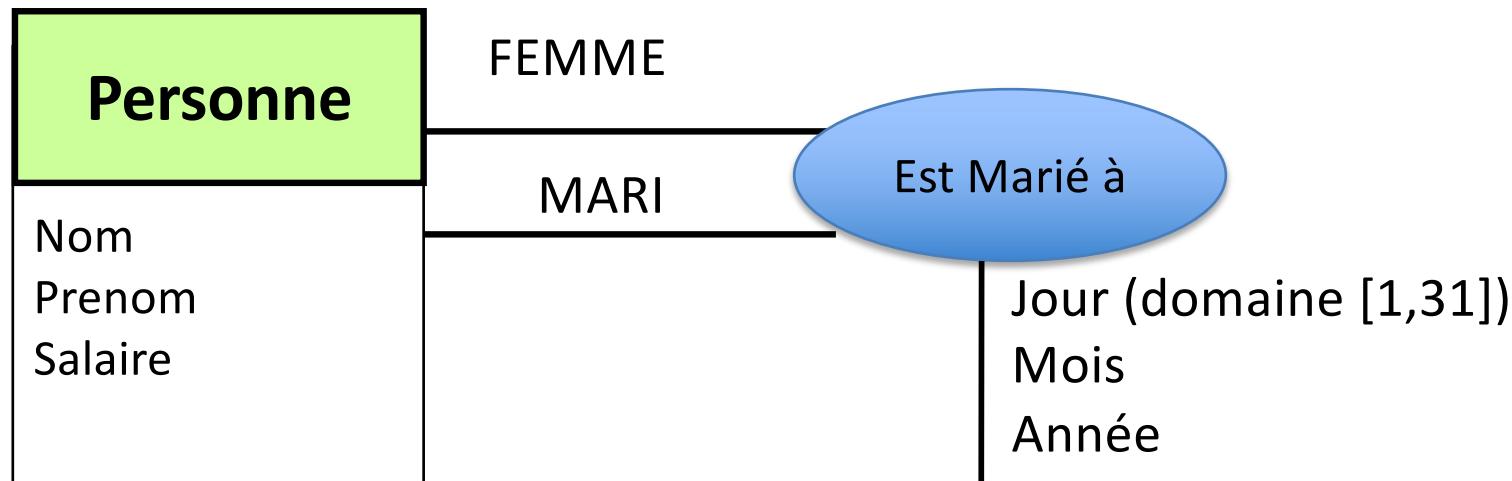
Cardinalités

La **date d'achat** d'une voiture dépend de la personne et de la voiture



Les attributs

- Décrivent les propriétés associées à :
 - un type d'entité
 - un type d'association
 - un autre attribut (**Attention**)



Les attributs

- Un attribut possède :
 - un nom (si possible différent)
 - Un domaine (à mettre dans le dictionnaire de données)

Nom attribut	Sémantique	Type	Contraintes	Calculé	...
Nom	Nom des personnes	STRING	12 CAR	NON	
Jour	Jour dans la date du mariage	INT	{1, 2, ..., 31}	NON	



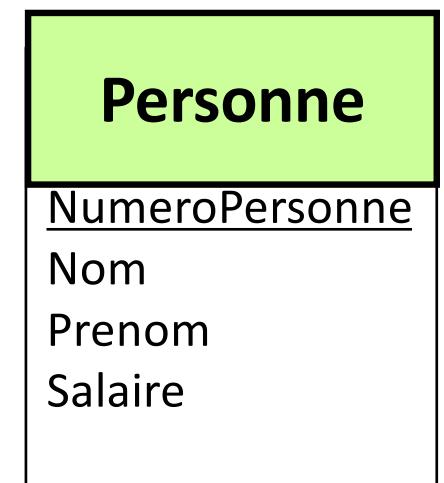
Attributs simples ou complexes

- simple (atomique) : non décomposable
 - Le domaine de valeurs est constitué de valeurs atomiques
 - Ex. : jour - domaine de valeurs : {1, 2,, 31}
 - Domaines prédéfinis standard, intervalles, énumérés
- Complexe : décomposé en d'autres attributs
 - Exemples: date (jour, mois, année), adresse (rue, ville, code postal)
- Un attribut complexe ne porte pas de valeur propre (pas de domaine directement associé)
- Attributs complexes à éviter si vous désirez aller vers du relationnel car plus compliqué. Un attribut complexe peut alors être vu comme un type d'association



Identifiant

- Pour chaque type d'entité il faut un identifiant qui permet de repérer une entité de manière unique et sans ambiguïté. Il peut être composé de plusieurs attributs
 - Numéro de personne
- Le faire apparaître de manière évidente dans le schéma



Nom attribut	Sémantique	Type	Contraintes	Calculé	Identifiants
Numero Personne	Numero de Personne	INT	Numero > 1 et Numero < 1000	NON	Identifiant



Quelques contraintes

- Monovalué (1 seule valeur)
 - Date de naissance
- Multivalué (plusieurs valeurs) (**Attention**)
 - Numéros de téléphones, prénoms
- Obligatoire
 - Nom (**NOT NULL**)
- Facultatif
 - Téléphone



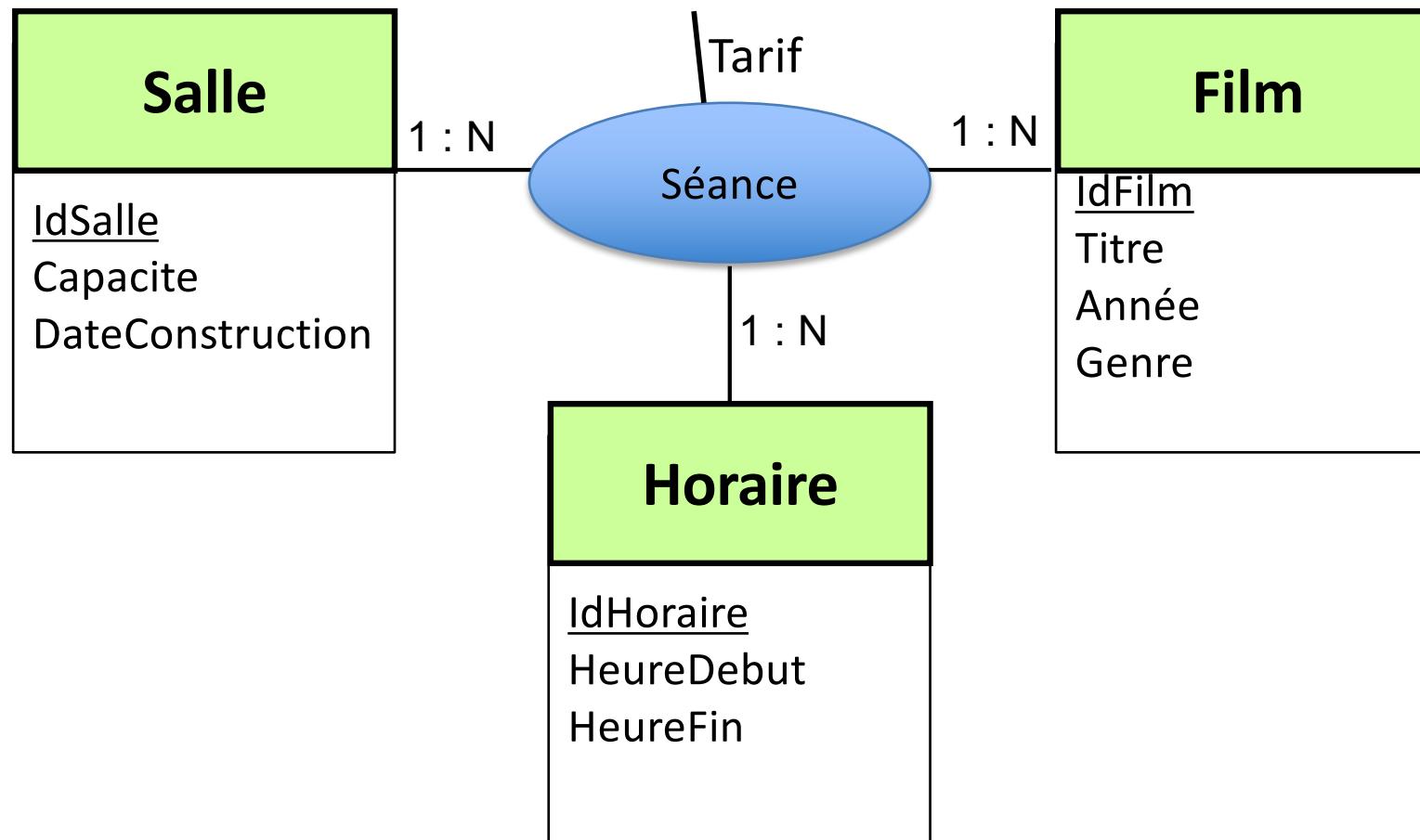
Retour sur les relations ternaires

- En principe il n'y a pas de limitations ...
cependant se limiter aux relations ternaires
- Exemple : Projection de certains films dans des salles à certains horaires



Retour sur les relations ternaires

- Projection de certains films dans des salles à certains horaires – Association ternaire

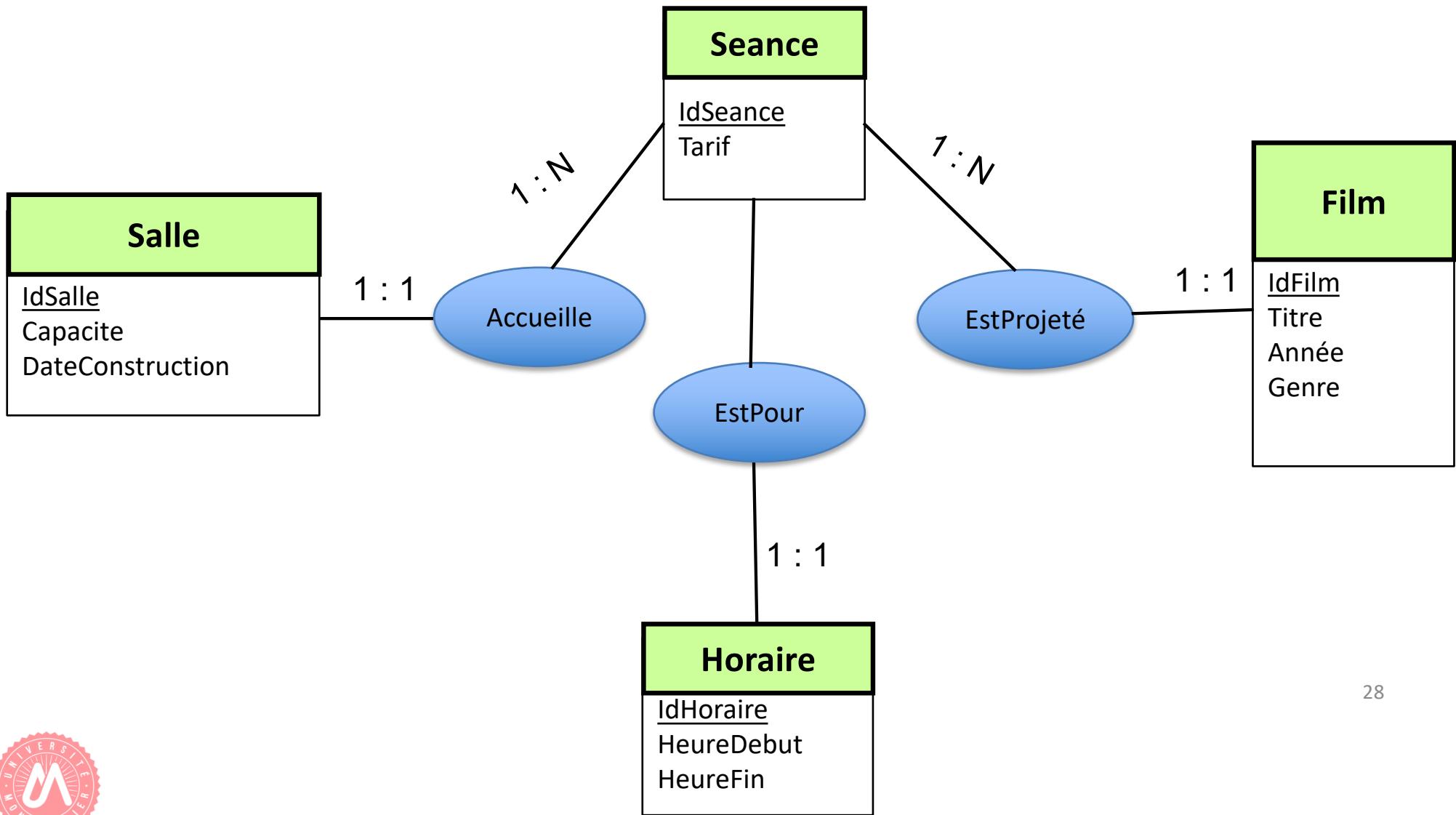


Retour sur les relations ternaires

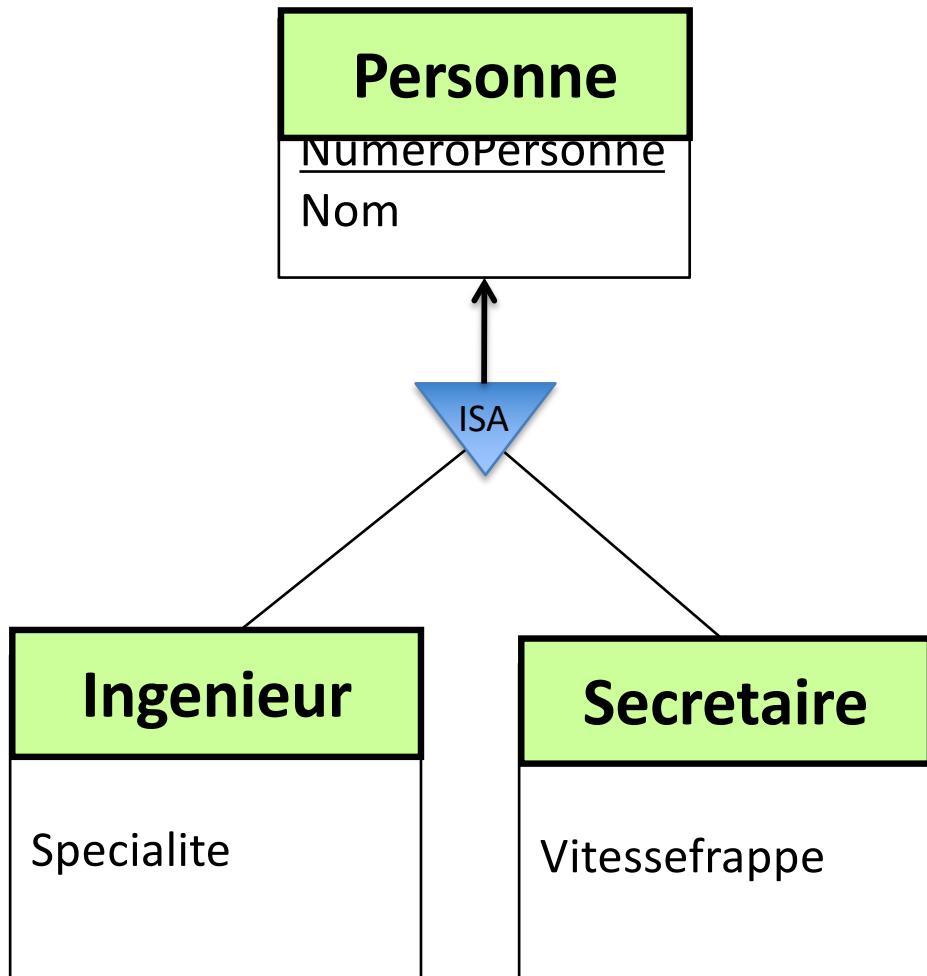
- Généralement de type 1:N
- Les associations de degré supérieur à deux difficiles à manipuler et à interpréter
- Il est toujours possible de remplacer cette association par un type d'entité
 - Création d'un type d'entité qui remplace l'association
 - Les nouvelles associations sont de type 1:N



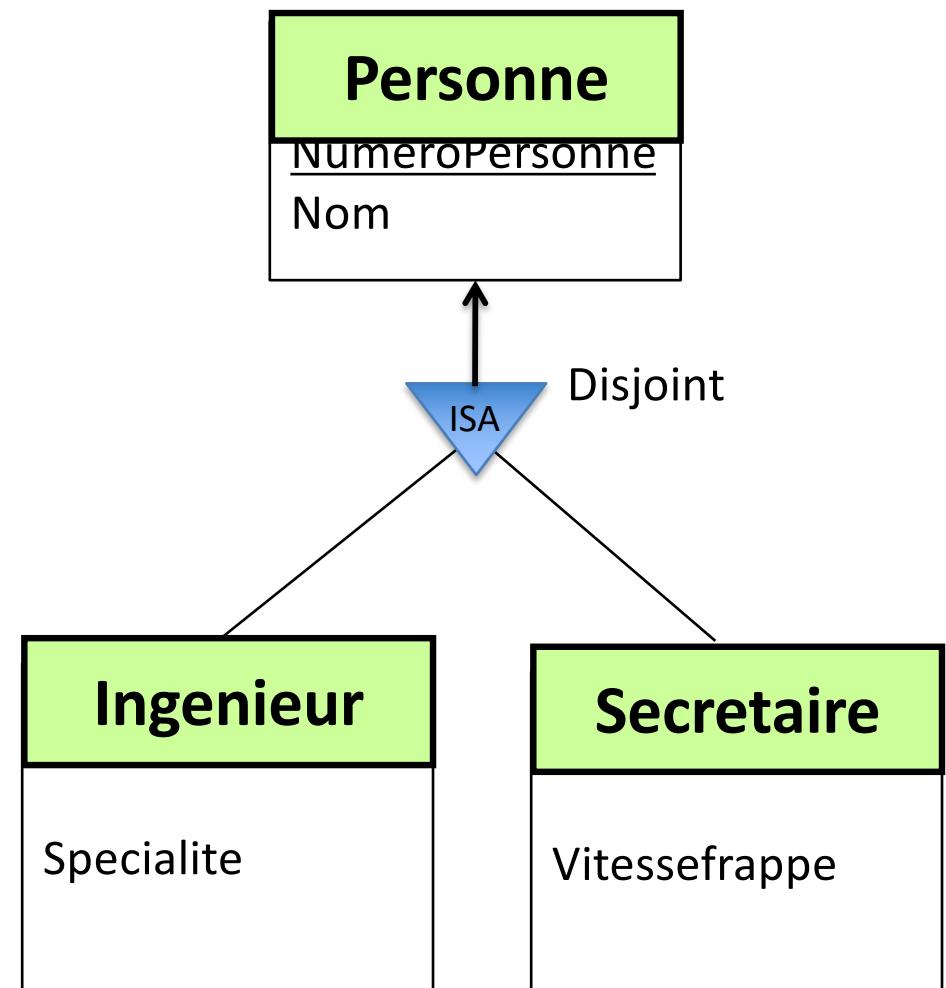
Retour sur les relations ternaires



Héritage



« lien IS-A »



Exclusion mutuelle



A propos des notations

- Il existe pour le modèle conceptuel différentes notations mais elles expriment les mêmes choses
- Exemple UML



A propos des notations

- Type d'Entité = Classe.
- Les associations ne sont pas représentées mais on peut les mettre sur les liens pour faciliter la lecture
- Les cardinalités ...



A propos des notations

- Associations 1 – 1

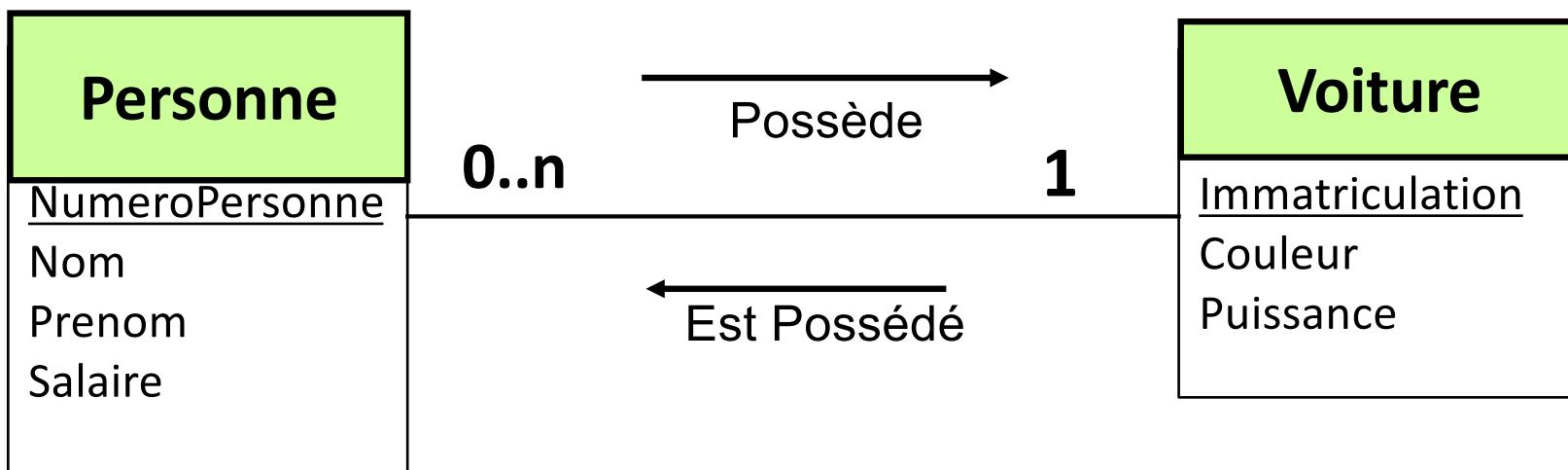
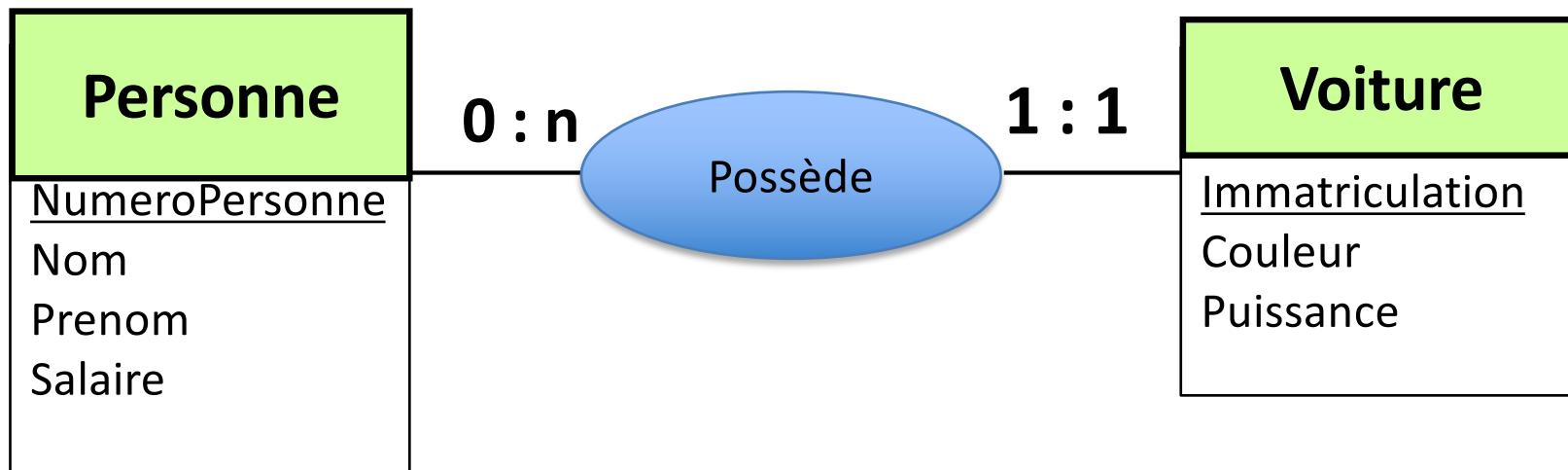
Cardinalités	Multiplicités UML
0,1 – 0,1	0..1 – 0..1
0,1 – 1, 1	0..1 – 1
1,1 – 1,1	1 – 1

- Associations 1 - N

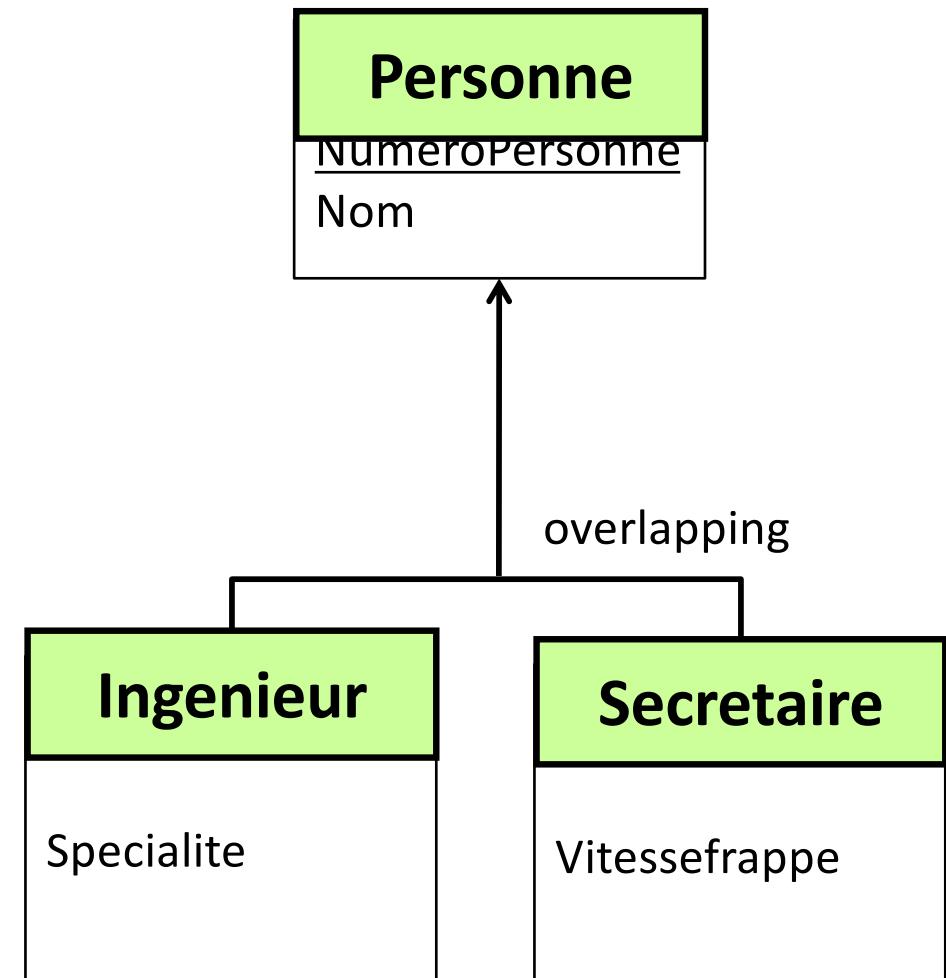
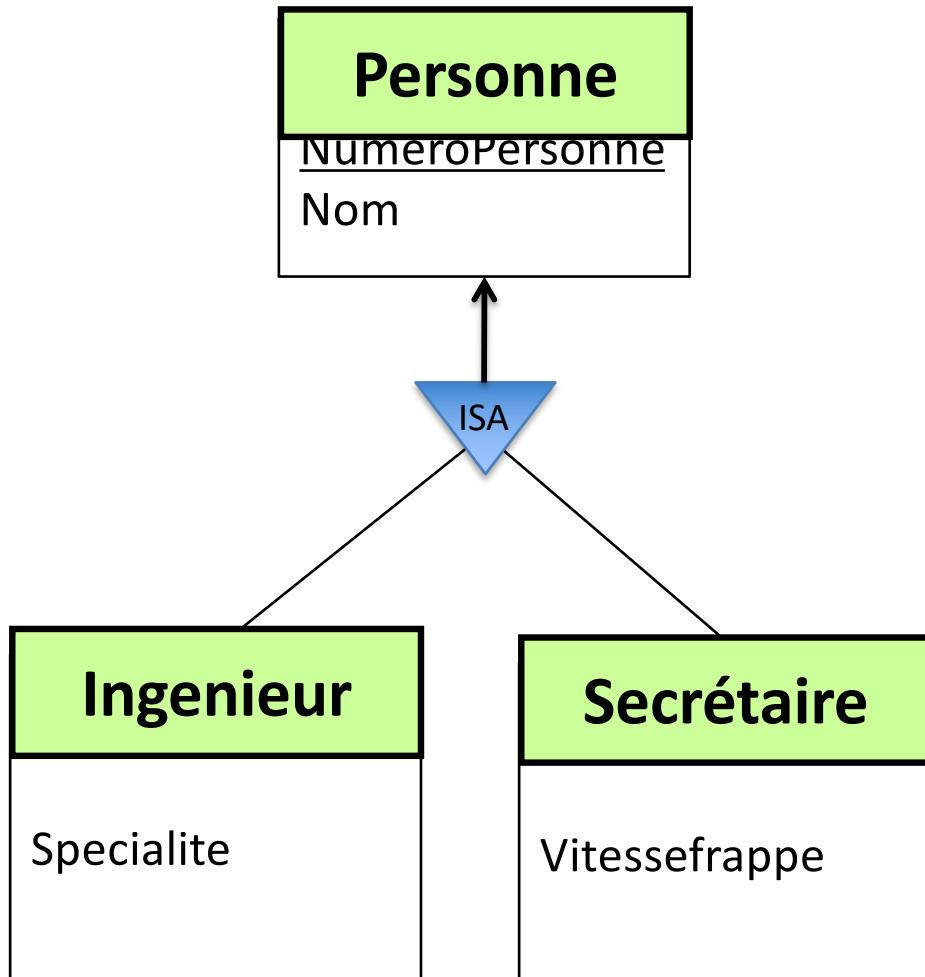
Cardinalités EA	Multiplicités UML
0,1 – 0,N	0..1 – *
0,1 – 1, N	0..1 – 1..*
1,1 – 0,N	1 – *
1,1 – 1,N	1 – 1..*



A propos des notations

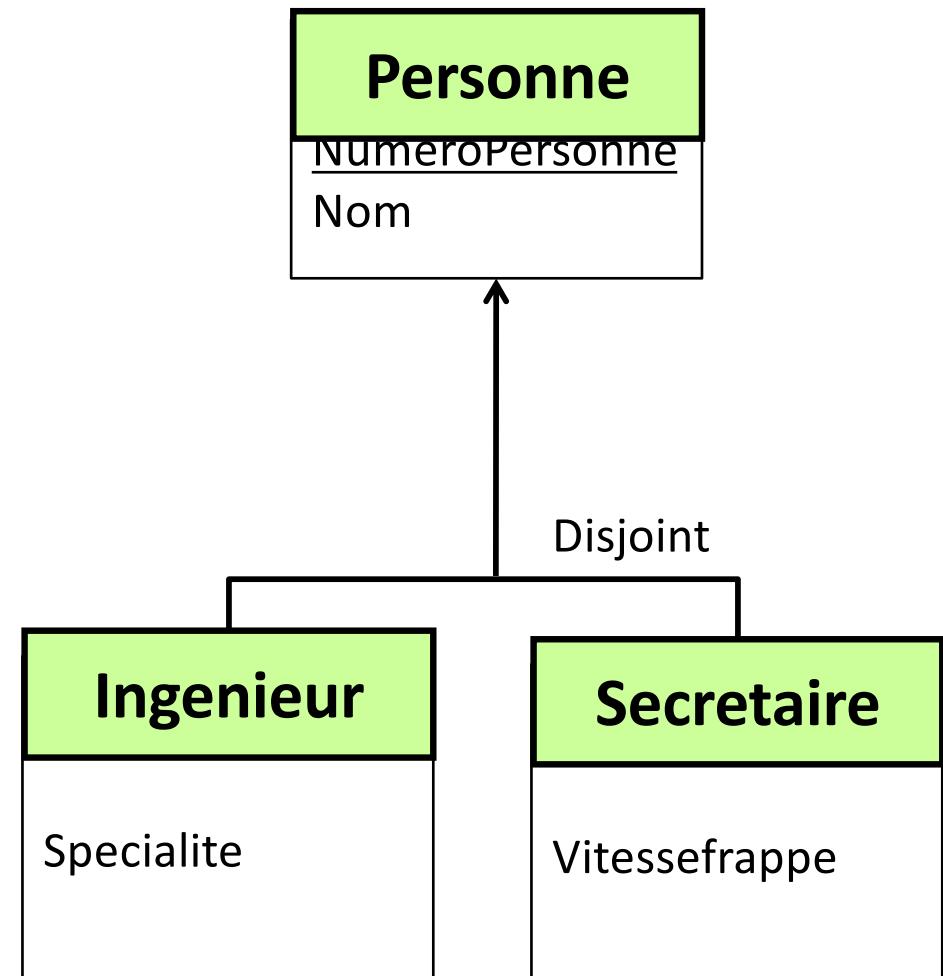
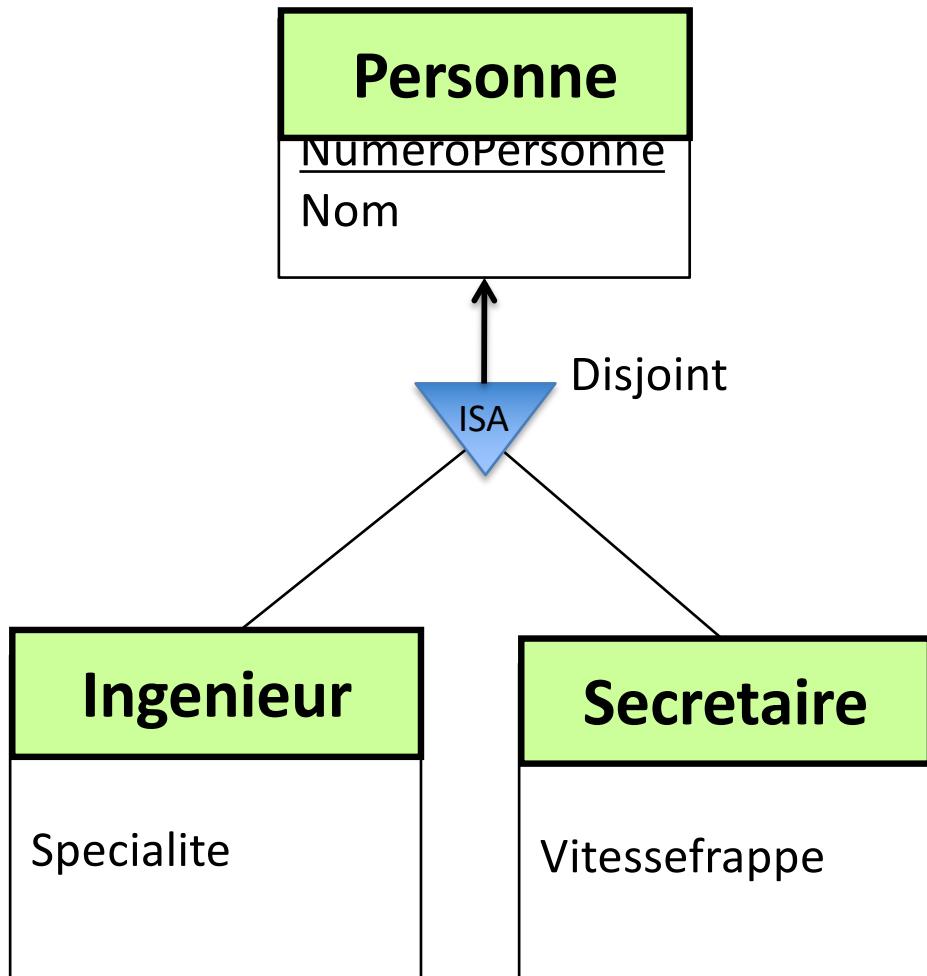


A propos des notations



En UML

A propos des notations



En UML



Caractéristiques d'un « bon schéma »

- Un type d'attribut :
 - est caractérisé par un nom et un domaine
 - est rattaché de façon exclusive à un type d'entité ou d'association
 - les noms des types d'attributs, d'entité et d'associations sont distincts
 - éviter les valeurs d'attributs complexes
 - Éviter les accents dans les noms
- Un type d'entité :
 - a au moins un identifiant
 - a si possible au moins un type d'attributs autre que la clé



Caractéristiques d'un « bon schéma »

- Le dictionnaire de données est important :
- Il doit contenir les différents attributs des types d'attributs , les contraintes, les identifiants, etc
- Rappel : c'est à partir du schéma et du dictionnaire de données que tout peut être automatiquement généré



Caractéristiques d'un « bon schéma »

- Quel est l'usage de la modélisation ?
- Il est important de savoir répondre à cette question !
- Votre modélisation doit être capable de répondre aux différentes requêtes qui vont être posées par le suite
- On ne modélise que pour répondre à un besoin. Ce besoin s'exprime par des requêtes !

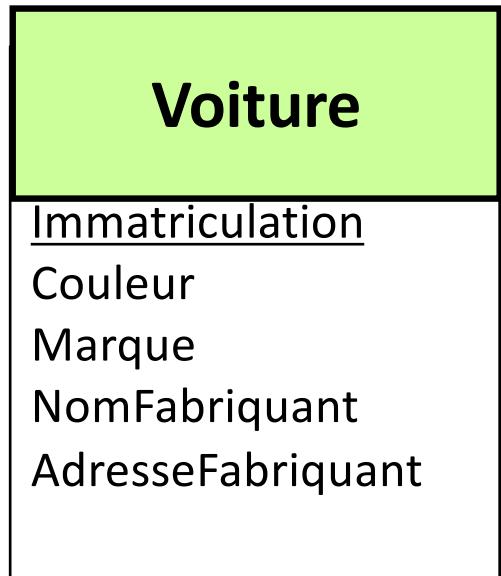


Caractéristiques d'un « bon schéma »

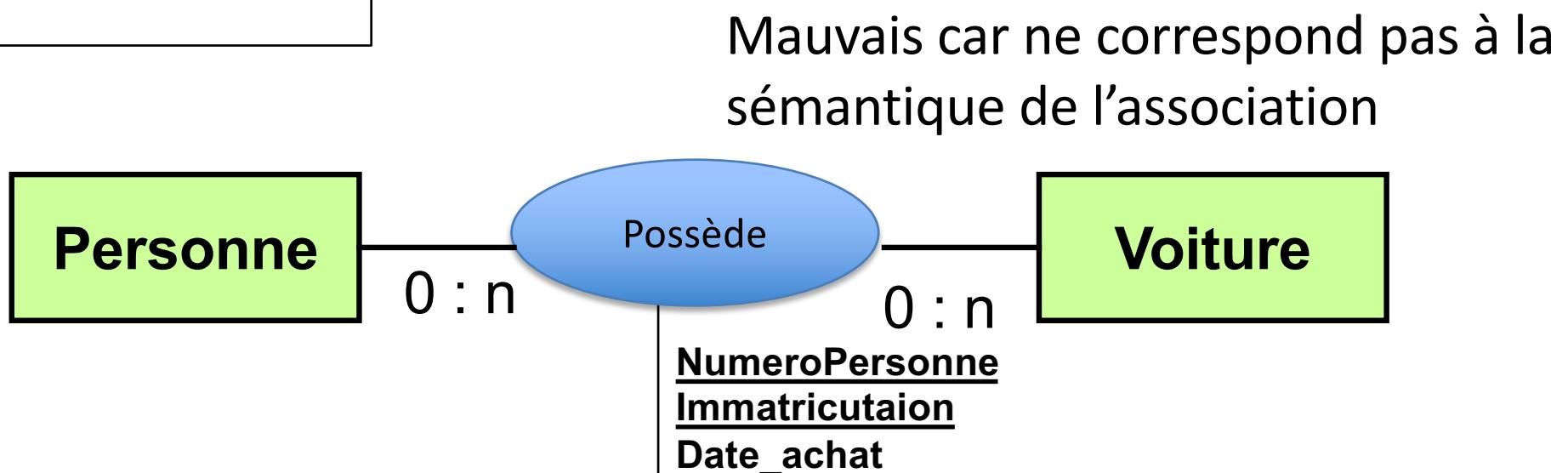
- Le client a peut être une vague idée de ce qu'il veut
- Nécessité de concevoir une base de données qui représente uniquement ses idées
- Respect des besoins : ne pas présager d'hypothèses futures
 - « un cours est assuré par un seul professeur » ... dans le futur **peut être que** « ce cours va être assuré par plusieurs professeurs »
- Faire quelque chose de simple



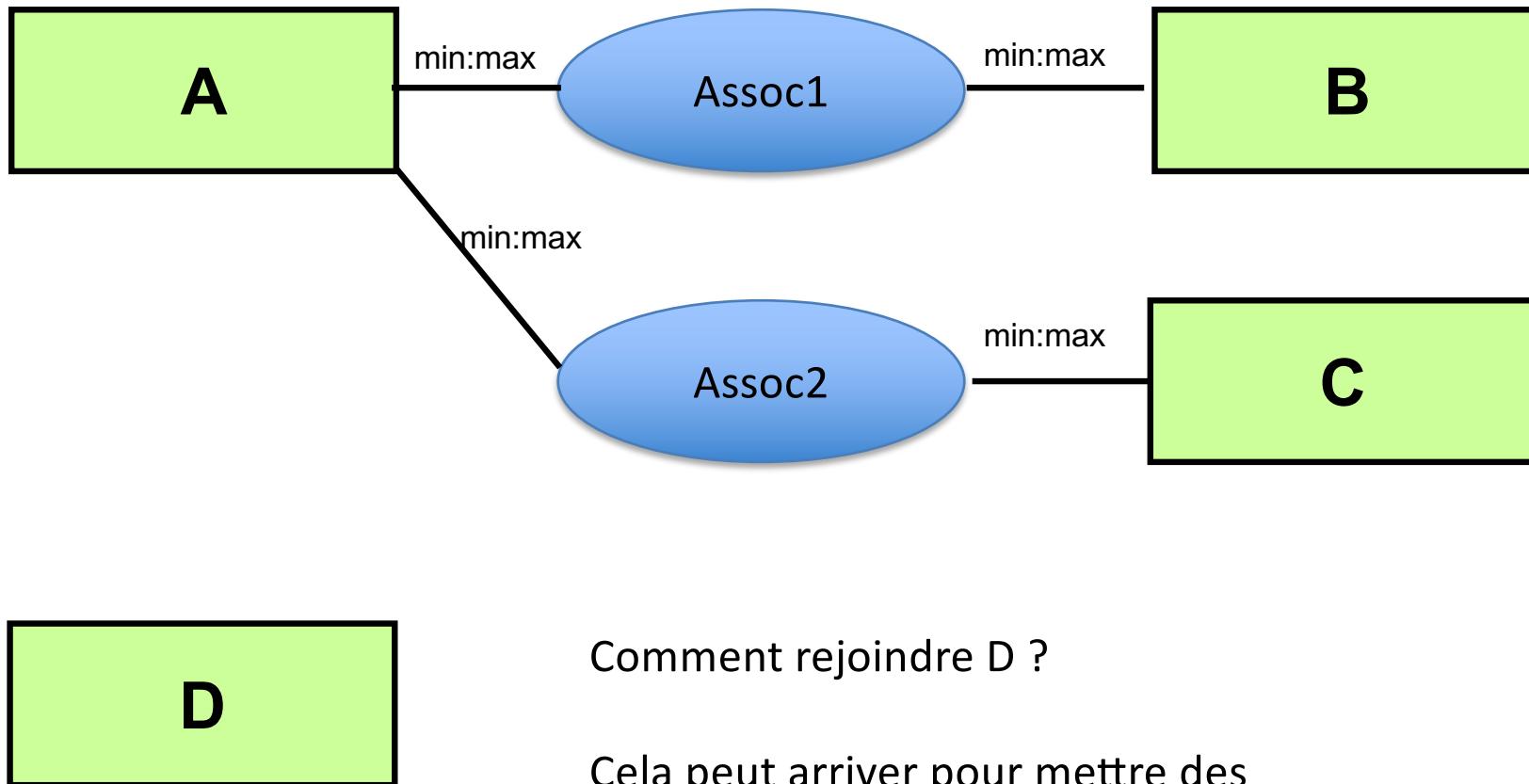
Des exemples de mauvais schémas



Mauvais car redondance du nom et de l'adresse de chaque fabriquant pour chaque voiture



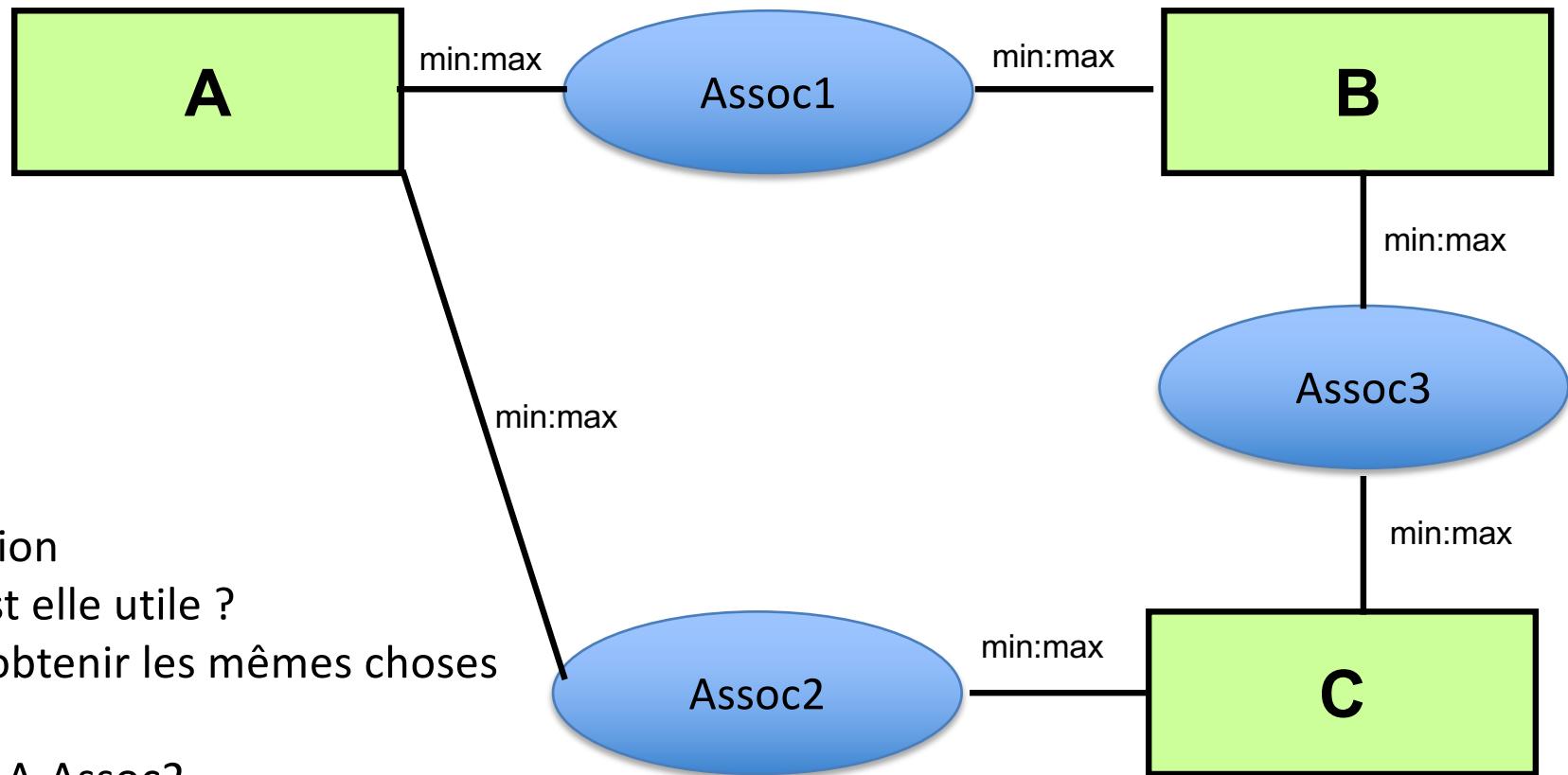
Des exemples de mauvais schémas



Comment rejoindre D ?

Cela peut arriver pour mettre des relations spécifiques du type statistiques sur des relations mais est-ce ce que vous souhaitez modéliser ?

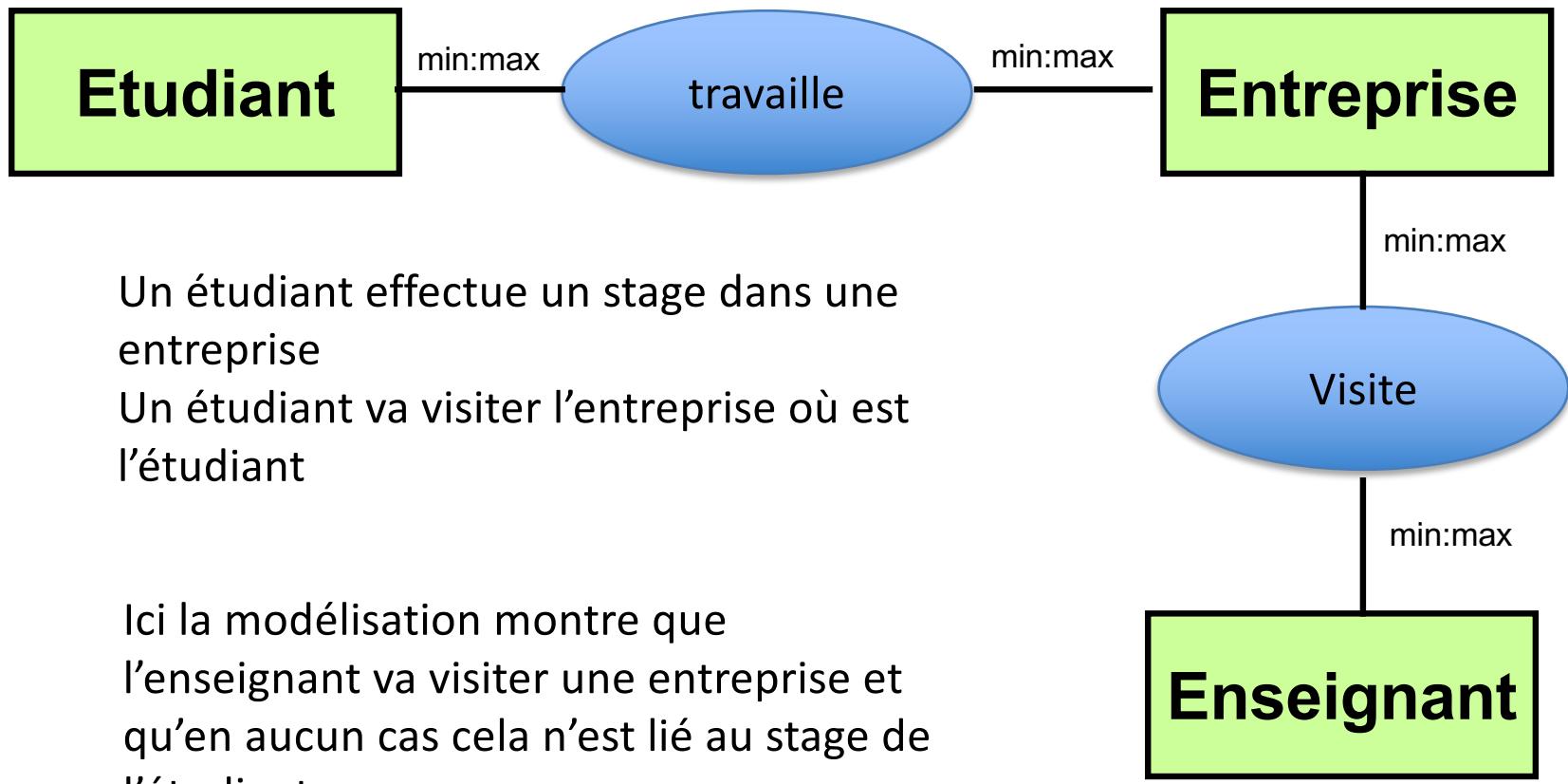
Des exemples de mauvais schémas



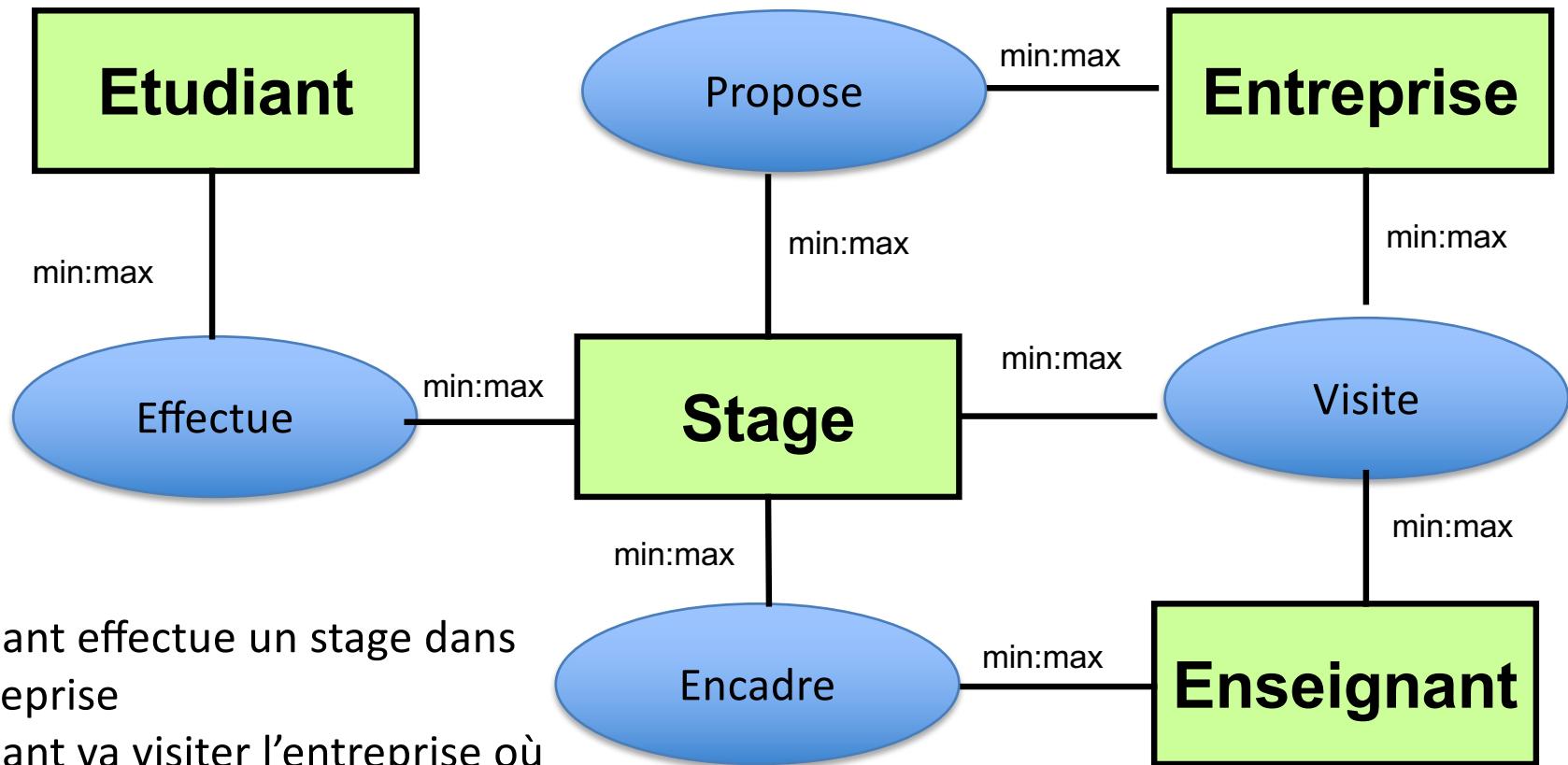
L'association
Assoc3 est elle utile ?
On peut obtenir les mêmes choses
par
B-Assoc1-A-Assoc2

Cela dépend des cas

Des exemples de mauvais schémas



Des exemples de mauvais schémas

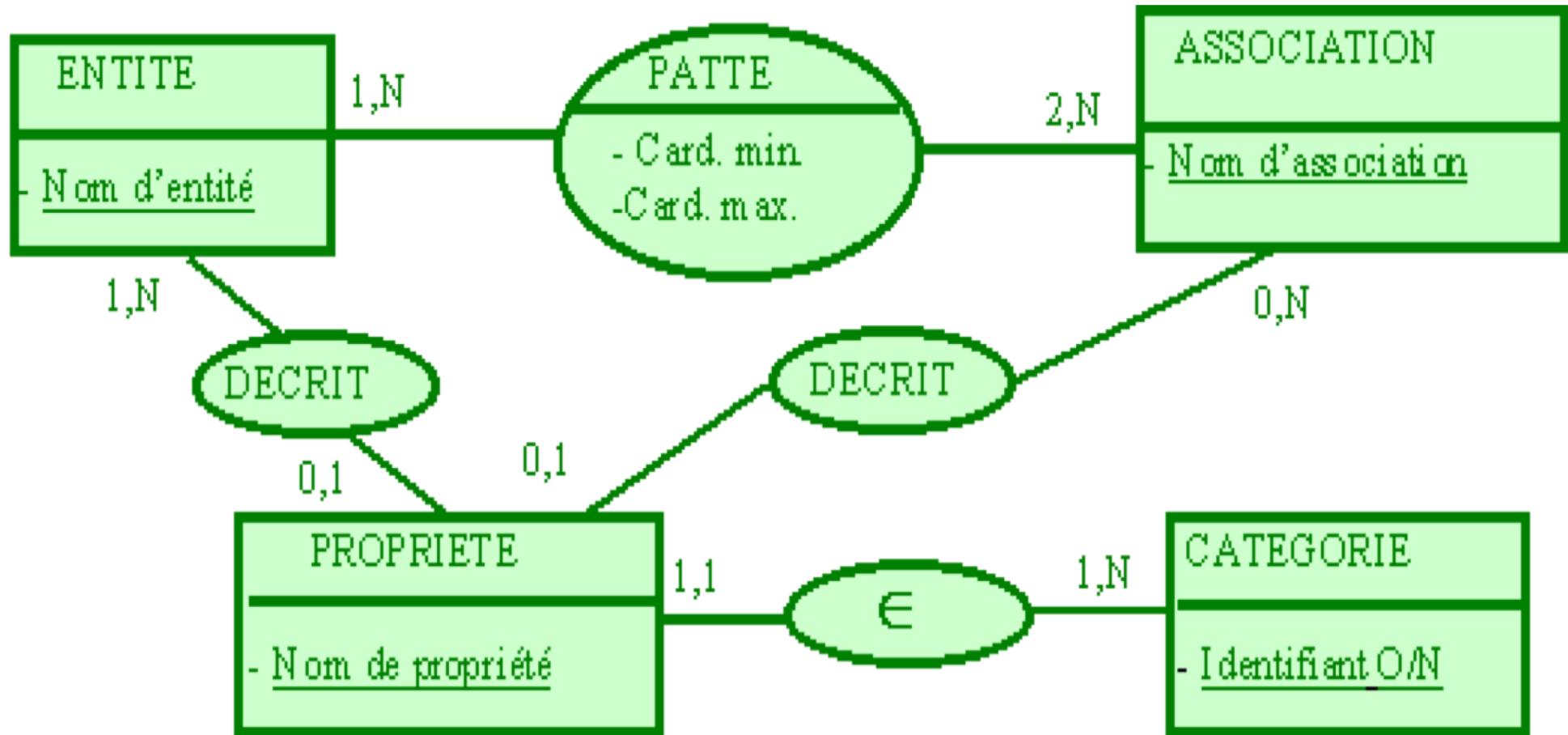


Un étudiant effectue un stage dans une entreprise

Un étudiant va visiter l'entreprise où est l'étudiant

La visite de l'enseignant est maintenant liée au stage
Il y a deux chemins similaires mais avec des sémantiques différentes

E/A - Le métamodèle



Organisation du cours

Introduction

Le modèle Entité-Association

Un exemple

Transformation vers le modèle relationnel



Un exemple

- Un client qui s'inscrit à la bibliothèque verse une caution. Suivant le montant de cette caution il aura le droit d'effectuer en même temps 10 emprunts au maximum.
- Les emprunts durent au maximum 15 jours
- Un livre est caractérisé par son numéro dans la bibliothèque (identifiant), son titre, son éditeur et son (ses) auteur(s).
- Les auteurs sont caractérisés par leurs nom, prénom et date de naissance et chaque livre est écrit à un moment donné.
- On veut pouvoir obtenir, pour chaque client les emprunts qu'il a effectués (nombre, numéro et titre du livre, date de l'emprunt).
- Toutes les semaines, on édite la liste des emprunteurs en retard : nom et adresse du client, date de l'emprunt, numéro(s) et titre du (des) livre(s) concerné(s).



On veut enfin pouvoir connaître pour chaque livre sa date d'achat et son état.

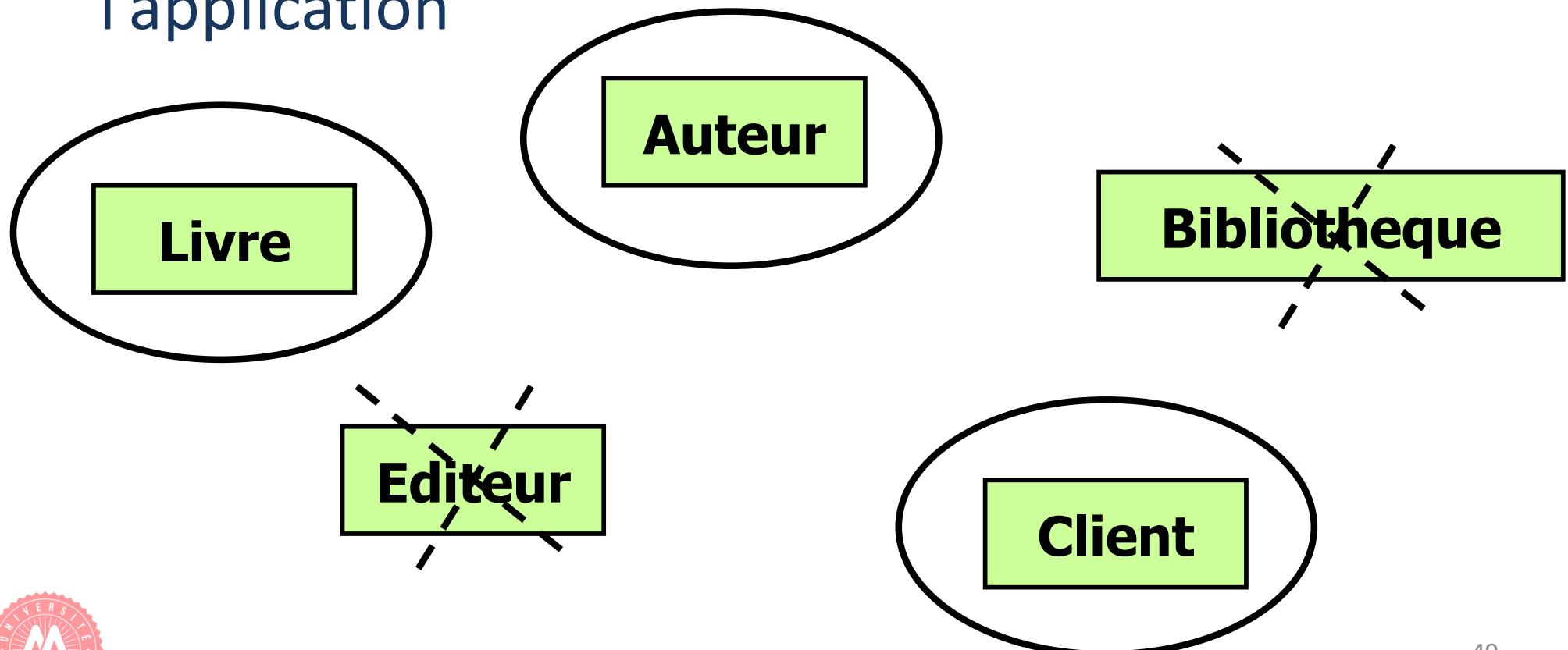
Objets ayant une existence propre

-
- Un **client** qui s'inscrit à la **bibliothèque** verse une caution. Suivant le montant de cette caution il aura le droit d'effectuer en même temps 10 emprunts au maximum.
 - Les emprunts durent au maximum 15 jours
 - Un **livre** est caractérisé par son numéro dans la bibliothèque (identifiant), son titre, son **éditeur** et son (ses) **auteur(s)**.
 - Les auteurs sont caractérisés par leurs nom, prénom et date de naissance et chaque livre est écrit à un moment donné.
 - On veut pouvoir obtenir, pour chaque client les emprunts qu'il a effectués (nombre, numéro et titre du livre, date de l'emprunt).
 - Toutes les semaines, on édite la liste des emprunteurs en retard : nom et adresse du client, date de l'emprunt, numéro(s) et titre du (des) livre(s) concerné(s).
 - On veut enfin pouvoir connaître pour chaque livre sa date d'achat et son état.



Les types d'entités

- Objets ayant une existence propre et ayant un intérêt pour au moins un traitement de l'application

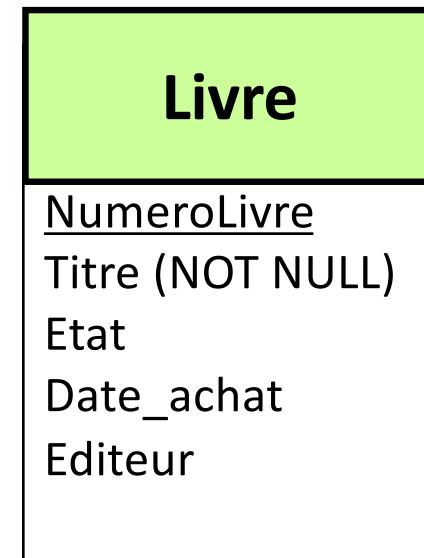


Les attributs du TE Livre

- Un client qui s'inscrit à la bibliothèque verse une caution. Suivant le montant de cette caution il aura le droit d'effectuer en même temps 10 emprunts au maximum.
- Les emprunts durent au maximum 15 jours
- Un **LIVRE** est caractérisé par son numéro dans la bibliothèque (identifiant), son titre, son éditeur et son (ses) auteur(s).
- Les auteurs sont caractérisés par leurs nom, prénom et date de naissance et chaque livre est écrit à un moment donné.
- On veut pouvoir obtenir, pour chaque client les emprunts qu'il a effectués (nombre, numéro et titre du livre, date de l'emprunt).
- Toutes les semaines, on édite la liste des emprunteurs en retard : nom et adresse du client, date de l'emprunt, numéro(s) et titre du (des) livre(s) concerné(s).
- On veut enfin pouvoir connaître pour chaque livre sa **date d'achat** et son **état**.



Les attributs du TE Livre



Les attributs du TE Auteur

- ...
- Les AUTEURS sont caractérisés par leurs *nom*, *prénom* et *date de naissance* et chaque livre est écrit à un moment donné.
- On veut pouvoir obtenir, pour chaque client les emprunts qu'il a effectués (nombre, numéro et titre du livre, date de l'emprunt).
-



Les attributs du TE Auteur

Auteur
<u>NumeroAuteur</u>
Nom (NOT NULL)
Prenom
DateNaissance

Intuitivement il semble raisonnable de considérer qu'un auteur a au moins un nom.

Il peut exister des auteurs qui n'ont pas de prénoms (par exemple nom auteur collectif)

Il peut exister des auteurs pour lesquels la date de naissance n'est pas très précise



Les attributs du TE Client

- Un **CLIENT** qui s'inscrit à la bibliothèque verse une *caution*. Suivant le montant de cette caution il aura le droit d'effectuer en même temps 10 emprunts au maximum.
- Les emprunts durent au maximum 15 jours
- On veut pouvoir obtenir, pour chaque client les emprunts qu'il a effectués (nombre, numéro et titre du livre, date de l'emprunt).
- Toutes les semaines, on édite la liste des emprunteurs en retard : *nom* et *adresse du client*, date de l'emprunt, numéro(s) et titre du (des) livre(s) concerné(s).



Les attributs du TE Client

Client
<u>NuméroClient</u>
NomClient (NOT NULL)
Caution
AdresseNum
AdresseRue
AdresseVille

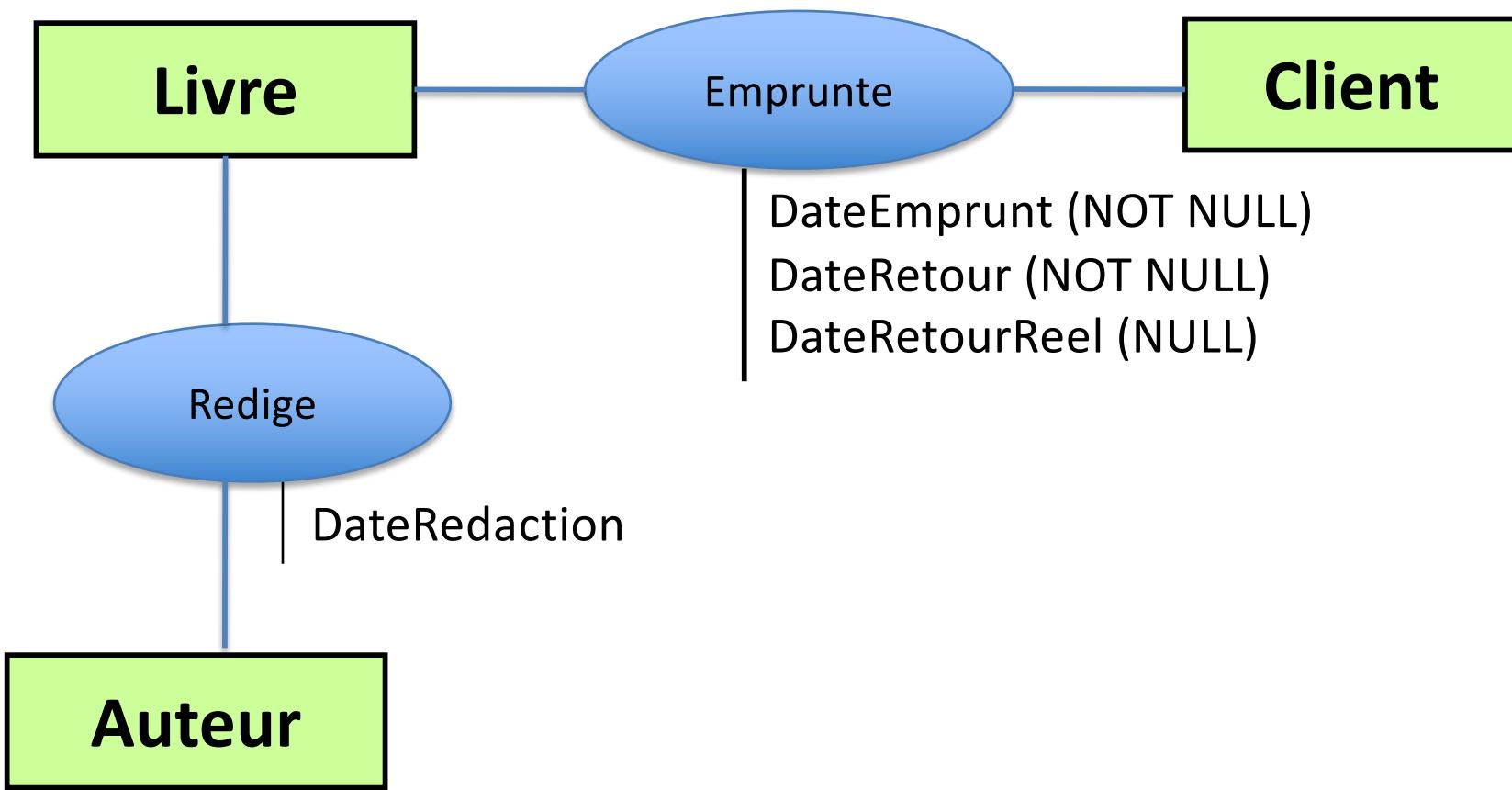
Intuitivement on pourrait ajouter le numéro de téléphone.

Par contre la bibliothèque n'en a pas parlé. Il s'agit d'une question qu'il faudrait poser à la bibliothèque pour lever cette ambiguïté



Les types d'association

- représentation d'un lien non orienté entre plusieurs entités (qui jouent un rôle déterminé)



Le dictionnaire de données

- Il est créé en parallèle
- Il est très important

Nom attribut	Sémantique	Type	Contraintes	Calculé	...
NumeroClient	Numero des clients des personnes	INT	1<NumeroClient <1000	NON	IDENTIFIANT
NomClient	Nom des clients	STRING	25 CAR MAX	NON	NOT NULL
DateEmprunt	Date où l'emprunt a été fait par un client pour un livre	DATE	JJ-MM-AAAA	NON	NOT NULL,



Contraintes

- Attention, ne pas oublier les contraintes, les reporter dans le dictionnaire de données pour certaines d'entre elles. Autrement les lister elles serviront par la suite
- Pour chaque occurrence d'Emprunt si DateRetour existe, alors elle doit être supérieure de 15 jours à DateEmprunt.
- Pour chaque occurrence de Livre, DateAchat doit être inférieure à DateEmprunt pour toutes les occurrences d'Emprunt qui lui sont liées.



Des vérifications

- Suis je capable de répondre à chaque requête en naviguant dans le schéma :
- On veut pouvoir obtenir, pour chaque client les emprunts qu'il a effectués (nombre, numéro et titre du livre, date de l'emprunt)
 - En regardant pour chaque client et en allant vers l'association emprunt on a bien les emprunts avec les livres associés ainsi que les dates d'emprunts. Pour avoir le nombre il faudra simplement les compter
- Toutes les semaines, on édite la liste des emprunteurs en retard : *nom et adresse du client, date de l'emprunt, numéro(s) et titre du (des) livre(s) concerné(s)*.
 - En passant par emprunt on pourra connaître les numéros des emprunteurs donc leur nom et adresse, les dates des emprunts et comme emprunt est relié à livre on pourra connaître les titres
- D'autres requêtes évidentes :
 - Connaître les livres en bon état -> via livre
 - Connaître tous les clients de la base -> via Client
 - Connaître tous les livres qui ne sont pas empruntés -> via livre et emprunt
 - Y a t'il des redondances dans les données stockées ? Non il n'y a pas de répétitions.
 - Il pourrait y avoir Editeur mais là on ne s'intéresse pas à l'adresse de l'éditeur donc on ne reporte que le nom ce qui n'est pas une répétition



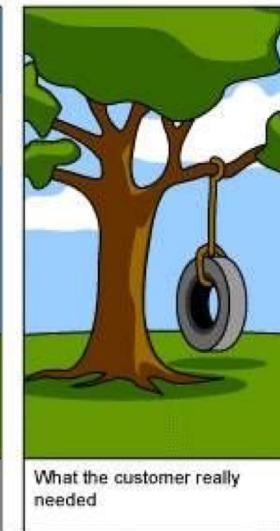
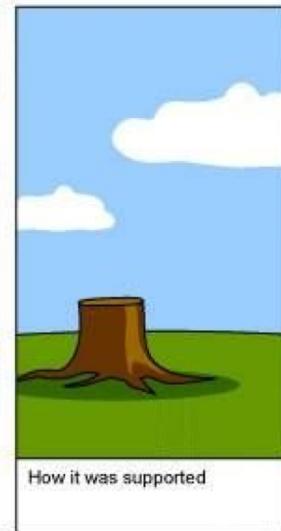
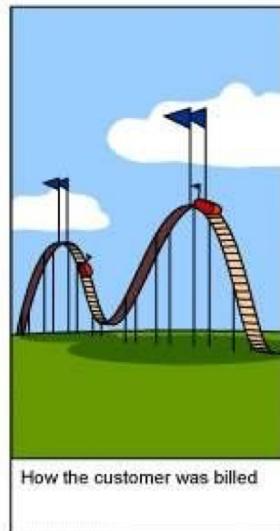
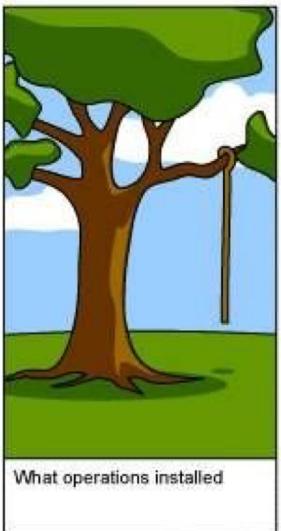
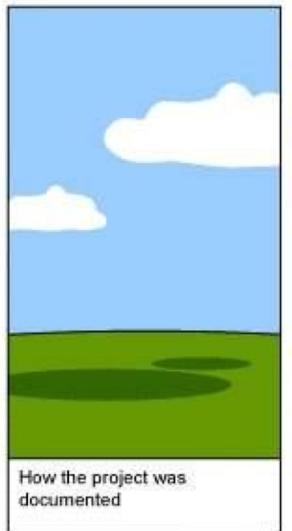
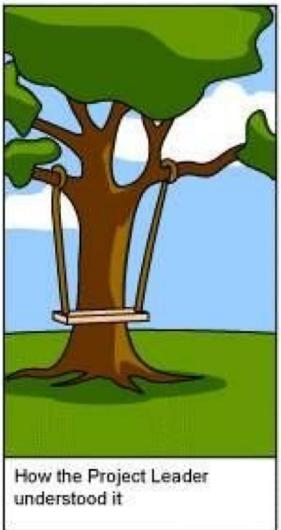
Il est important avant de passer à l'étape suivante de faire ces vérifications

Conclusion

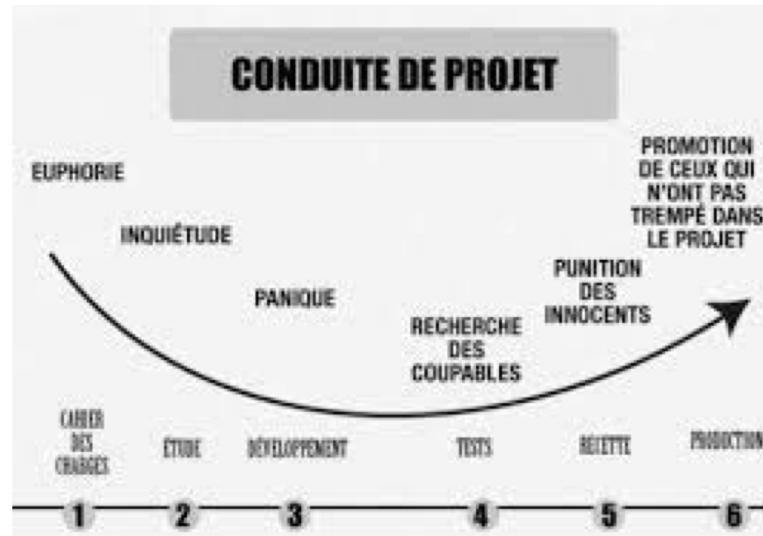
- Il s'agit de l'étape la plus importante d'un projet
- Après vous ne maîtrisez plus rien !!
- Toute modification sur les données en relationnel sera très couteuses et engendrera des incohérences
 - Nécessité de faire du rétro-engineering
 - Nécessité d'utiliser des algorithmes compliqués dont les résultats sont approximatifs
- Il s'agit de la mémoire de votre SI



La réalité



Des réalités



Organisation du cours

Introduction

Le modèle Entité-Association

Un exemple

Transformation vers le modèle relationnel



Transformation E/A - Relationnel

- Pour chaque type d'entité
 - Type d'entité => Relation
 - Les attributs du type d'entité deviennent les attributs de la relation
 - La clé de la relation est la clé du type d'entité

Client
<u>Numeroclient</u>
NomClient (NOT NULL)
Caution
AdresseNum
AdresseRue
AdresseVille

Relation Client (Numeroclient, NomClient, AdresseNum, AdresseRue, AdresseVille)

Importance du dictionnaire de données :
Le type des attributs provient du dictionnaire des données ainsi que les contraintes (exemple NomClient NOT NULL)



Transformation E/A - Relationnel

- Pour les Types d'Associations
 - cas 1 : cardinalité 1,1 sur l'une des pattes

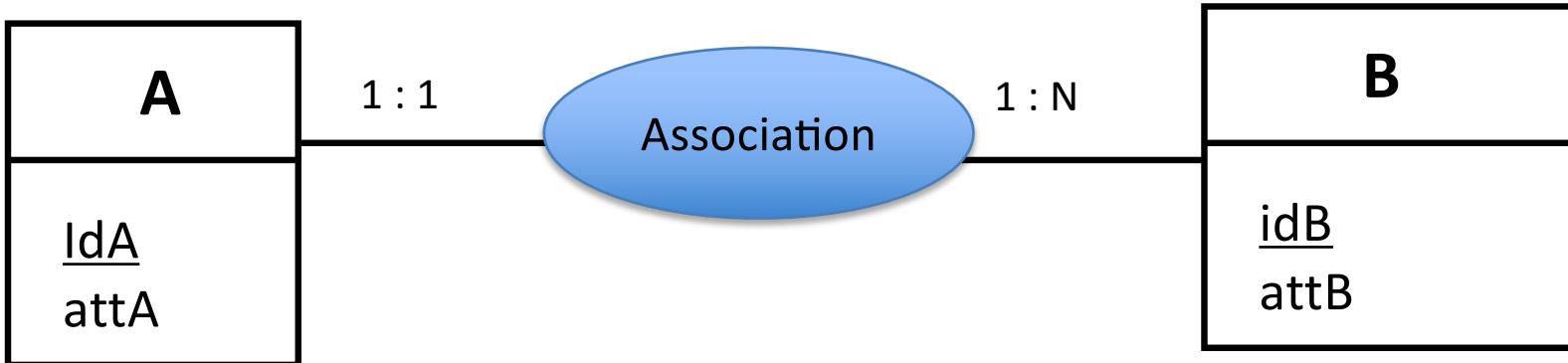
compléter la relation concrétisant l'entité avec la patte 1,1 en y ajoutant une propriété qui référence l'identifiant de l'autre entité
 - cas 2 : toutes les cardinalités maximale = n

créer une relation pour l'association, dont la clé se compose des clés des entités liées, avec éventuellement les propriétés portées par l'association
 - cas 3 : cardinalité 0,1 sur l'une des pattes

choisir entre une traduction selon le cas 1 ou le cas 2 en veillant toutefois : dans le cas 1, si la relation a des propriétés, ne pas oublier de les adjoindre à la relation correspondant à l'entité avec la patte 0,1 et dans le cas 2 de simplifier l'identifiant
- Des cas particuliers ...



Cas 1 - cardinalité 1 : 1 sur une patte



Relation A

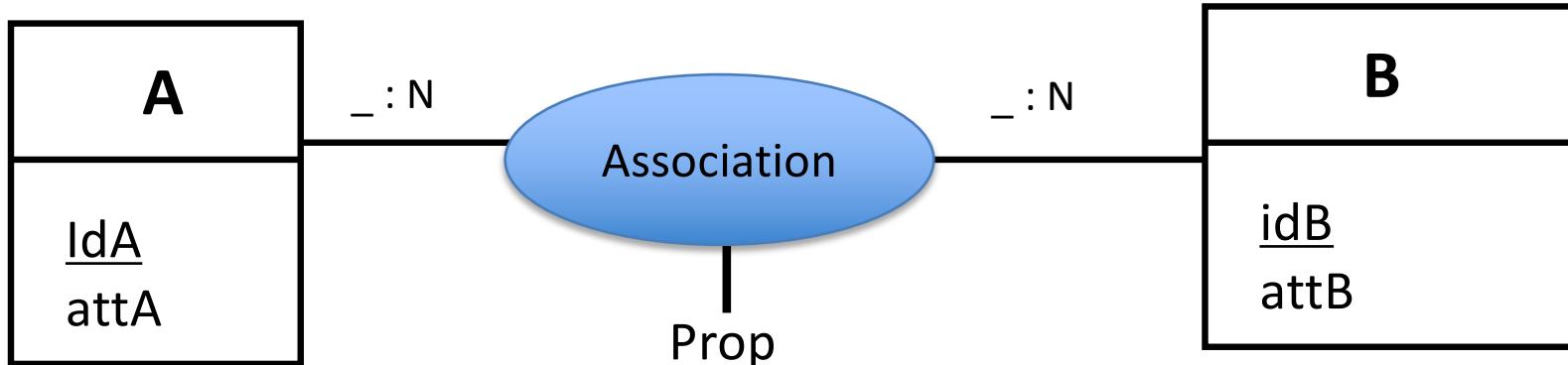
<u>IdA</u>	<u>IdB</u>	attA	
------------	------------	------	--

idB est aussi clé étrangère

Relation B

<u>IdB</u>	attB		
------------	------	--	--

Cas 2 - cardinalité _ : N



Relation A

<u>IdA</u>	attA		
------------	------	--	--

Relation B

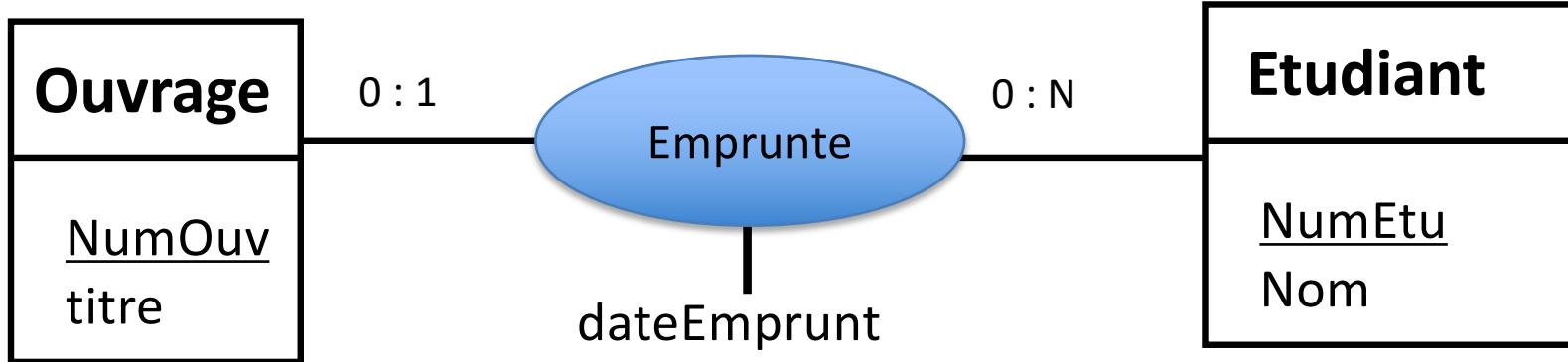
<u>idB</u>	attB		
------------	------	--	--

Relation Assoc

<u>IdA</u>	<u>idB</u>	Prop	
------------	------------	------	--

idA et idB sont aussi clés étrangères

Cas 3 - Cardinalité 0 : 1 sur une patte



Relation Ouvrage

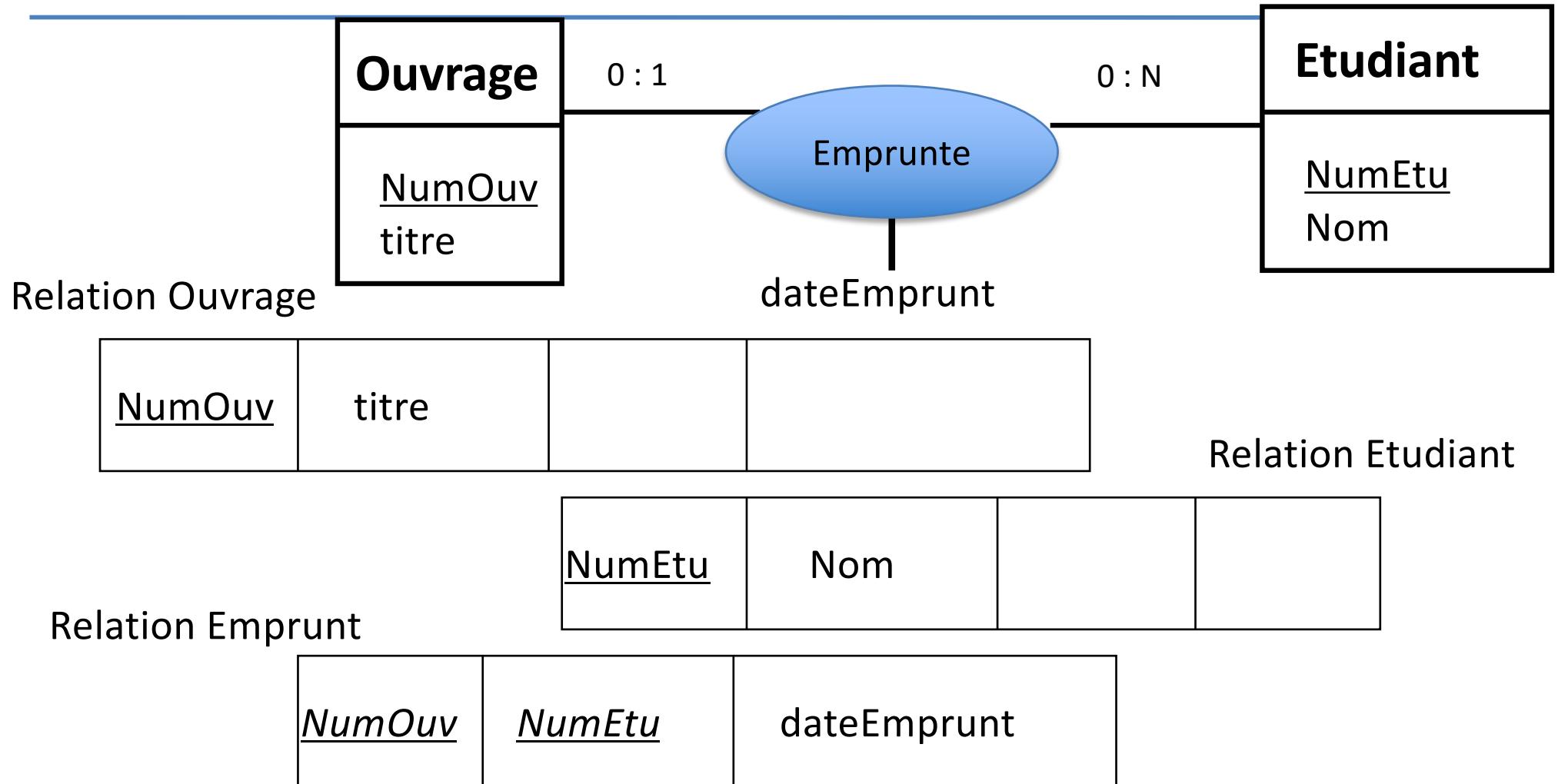
<u>NumOuv</u>	<i>NumEtu</i>	<i>titre</i>	<i>dateEmprunt</i>

Relation Etudiant

<u>NumEtu</u>	<i>Nom</i>		

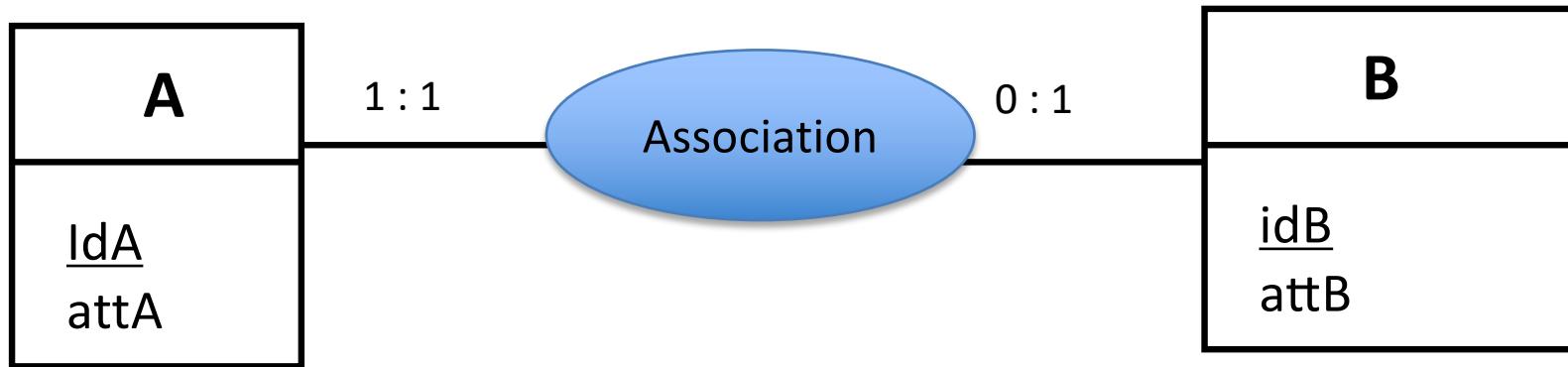
les ouvrages non empruntés auront une valeur non-renseignée (NULL) – Problème potentiel dans les requêtes

Cas 3 - Cardinalité 0 : 1 sur une patte



Tous les ouvrages sont répertoriés dans la table Ouvrage,
et seuls les ouvrages empruntés figurent dans la table Emprunt

Cas 1-1 et 0-1



Relation A

<u>IdA</u>	attA	<u><i>IdrefB</i></u>	
------------	------	----------------------	--

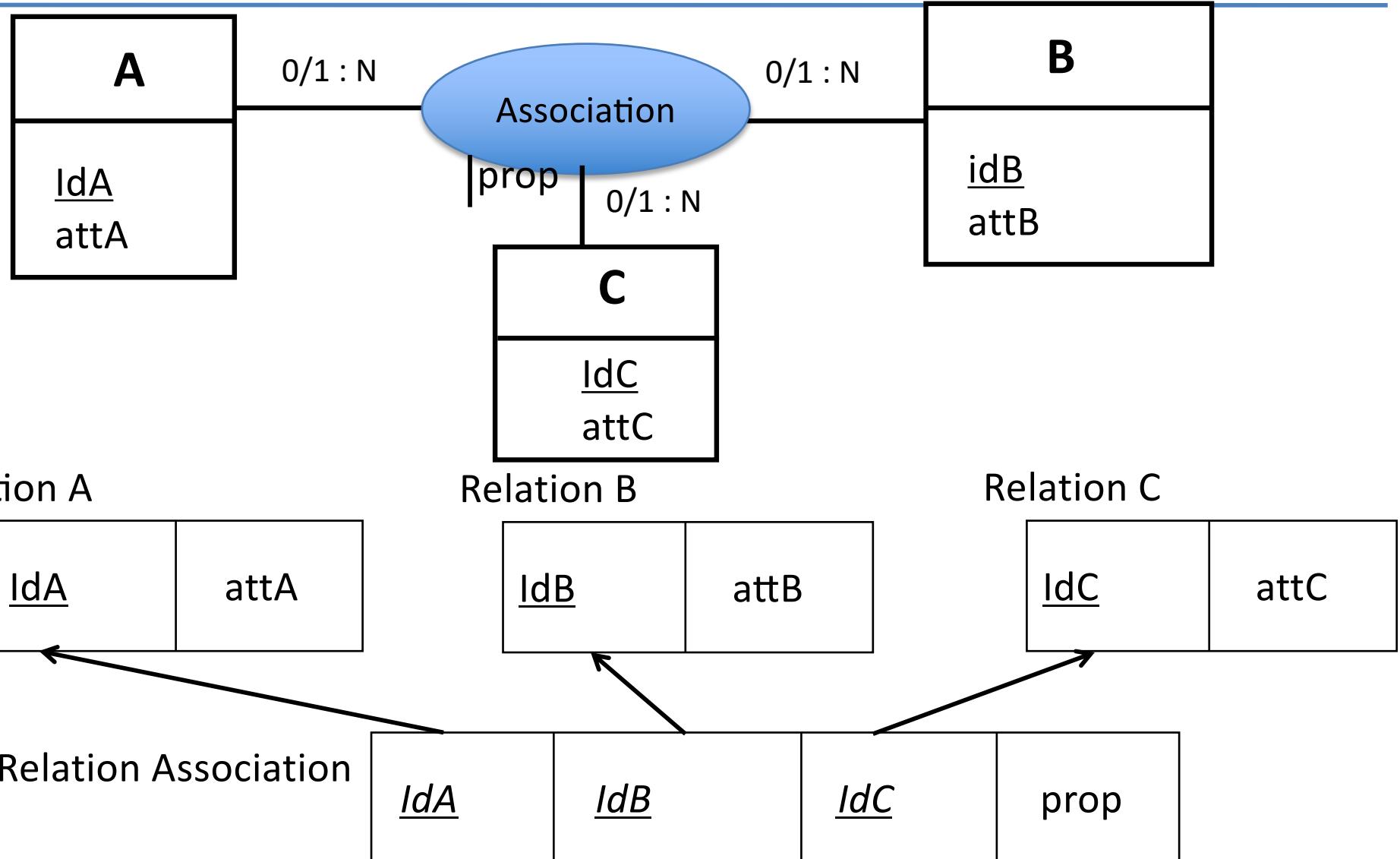
idrefB clé étrangère

Il ne peut y avoir qu'un
B pour un A
Un B n'a pas forcément
de A

Relation B

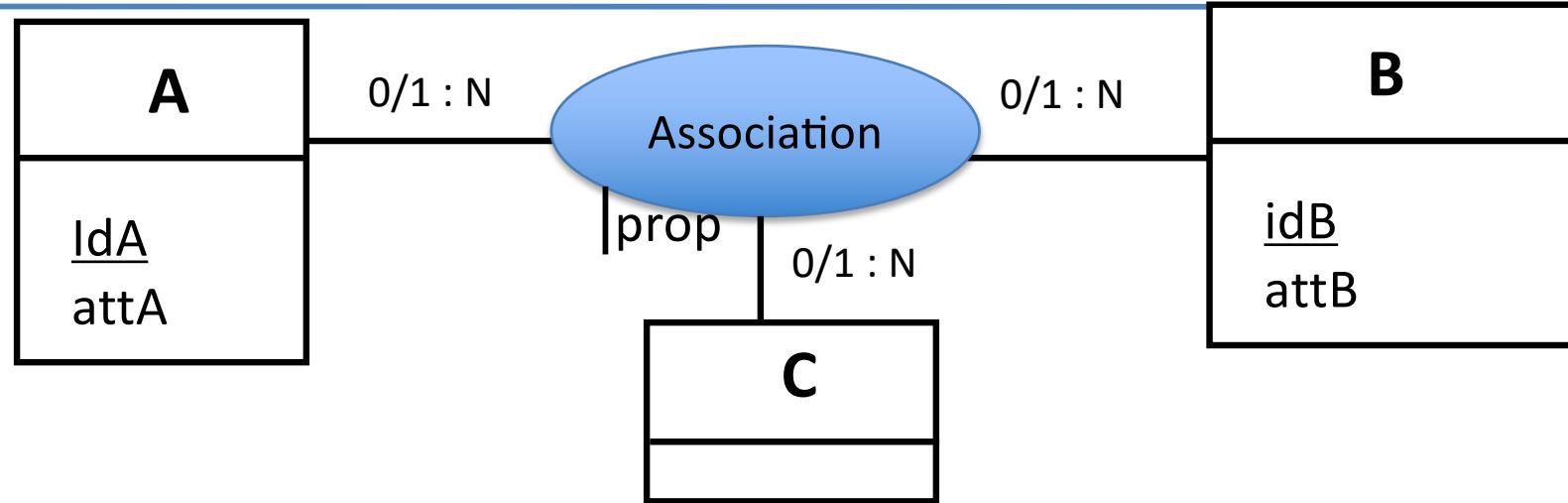
<u>IdB</u>	attB		
------------	------	--	--

Cas n-aires (:n, :n, :n)



La clé est idA, idB, idC. idA, idB et idC sont aussi clés étrangères

Cas n-aires (:n, :n, :n)



QUID DE C ?

Relation A

<u>IdA</u>	attA
------------	------

Relation B

<u>IdB</u>	attB
------------	------

Relation C

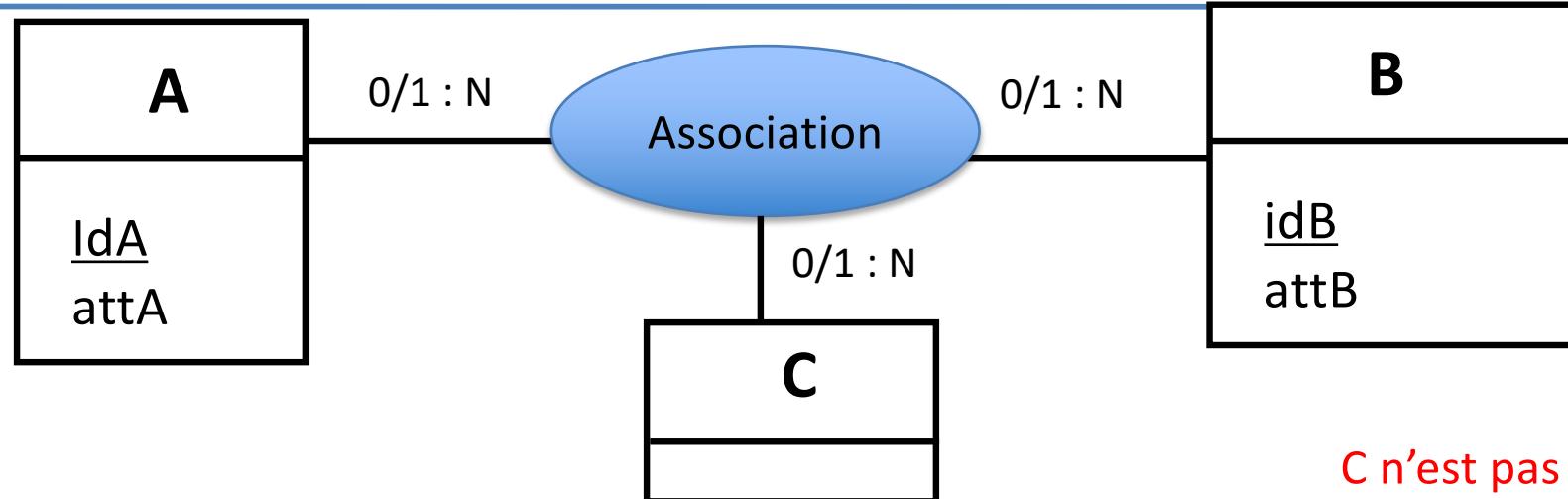
<u>IdC</u>	
------------	--

Relation Association

<u>IdA</u>	<u>IdB</u>	<u>IdC</u>	prop
------------	------------	------------	------

La clé est idA, idB, idC. idA, idB et idC sont aussi clés étrangères

Cas n-aires (:n, :n, :n)



Relation A

<u>IdA</u>	attA
------------	------

Relation B

<u>idB</u>	attB
------------	------

Relation C

<u>IdC</u>	
------------	--

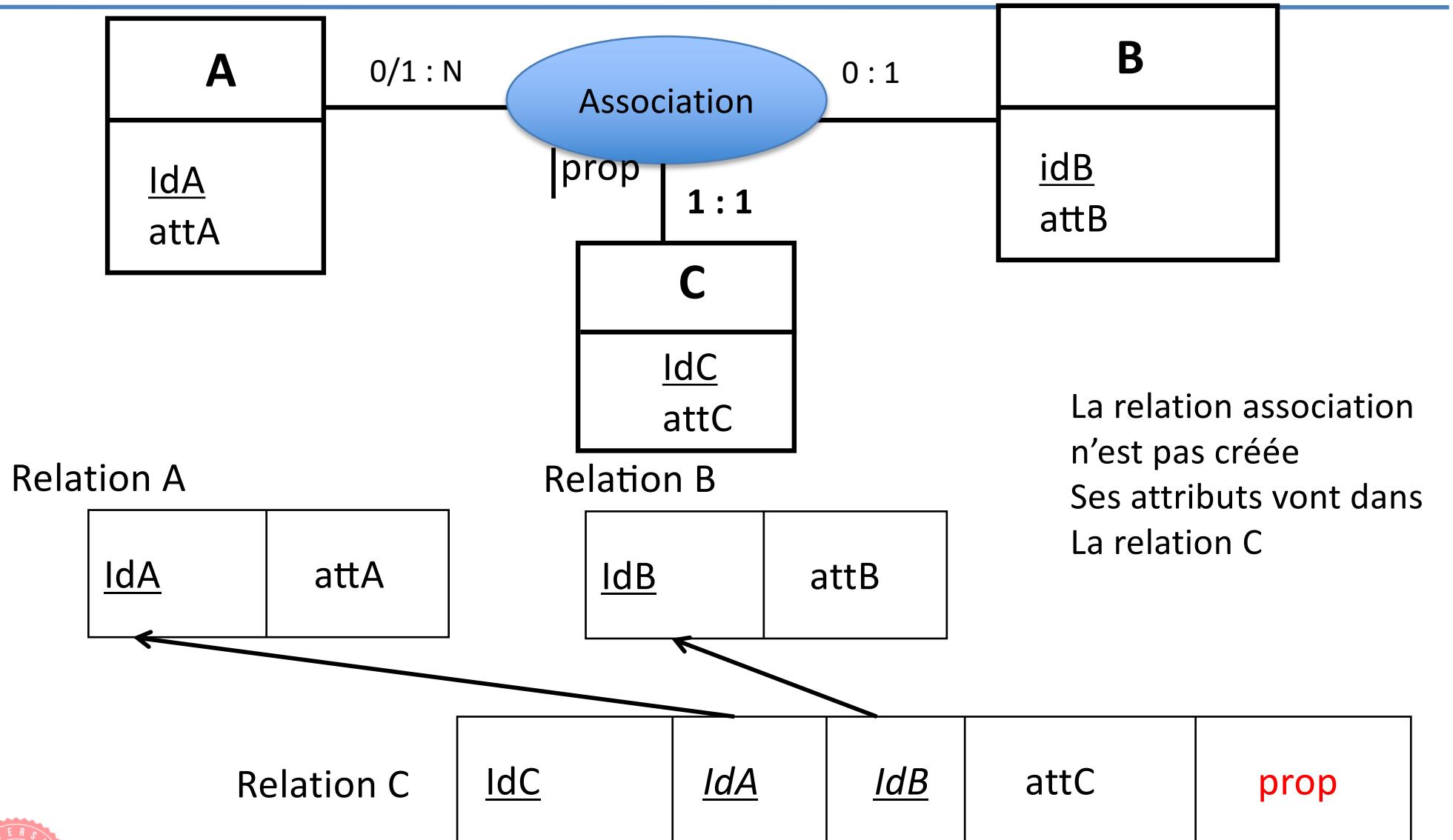
Relation Association

<u>IdA</u>	<u>idB</u>	<u>IdC</u>	prop
------------	------------	------------	------

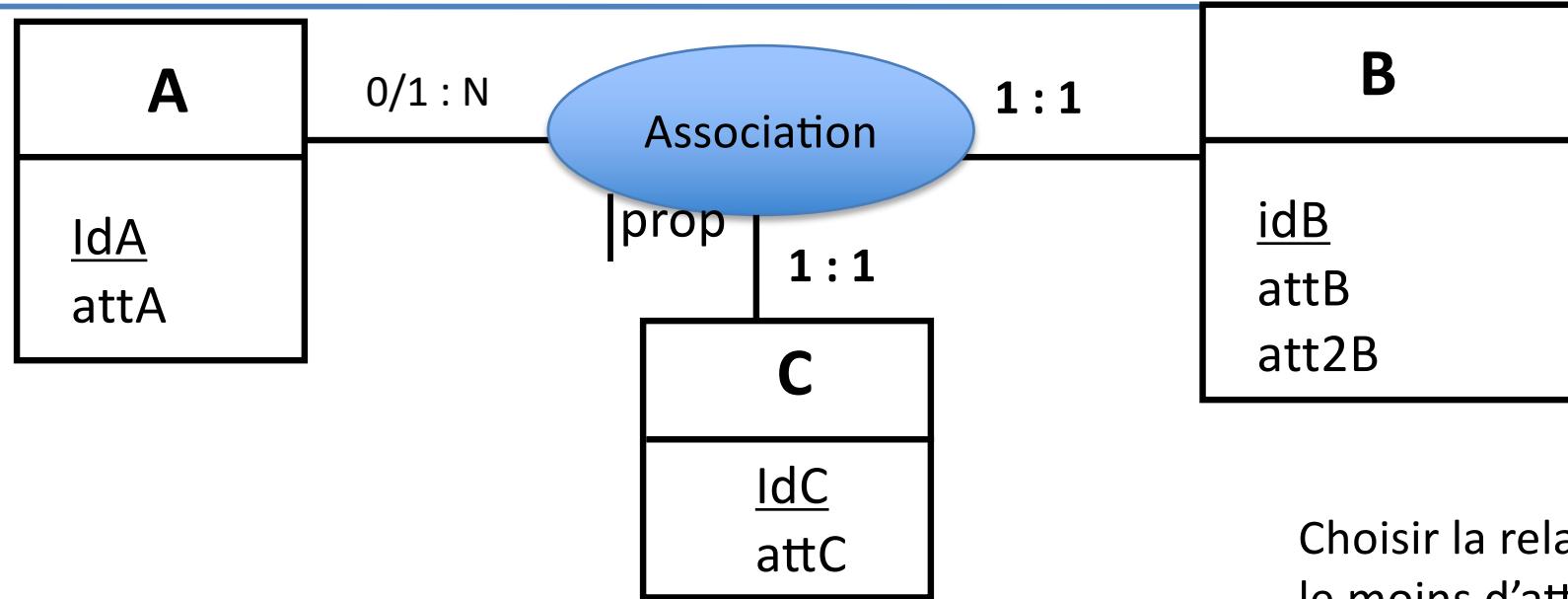
C n'est pas important
(exemple DATE)
Ne pas créer la relation

La clé est idA, idB, idC. idA, idB sont clés étrangères. IdC n'est plus clé étrangère

Cas n-aires (:n, 0:1, 1:1)



Cas n-aires (:n, 1:1, 1:1)



Relation A

<u>IdA</u>	attA
------------	------

Relation B

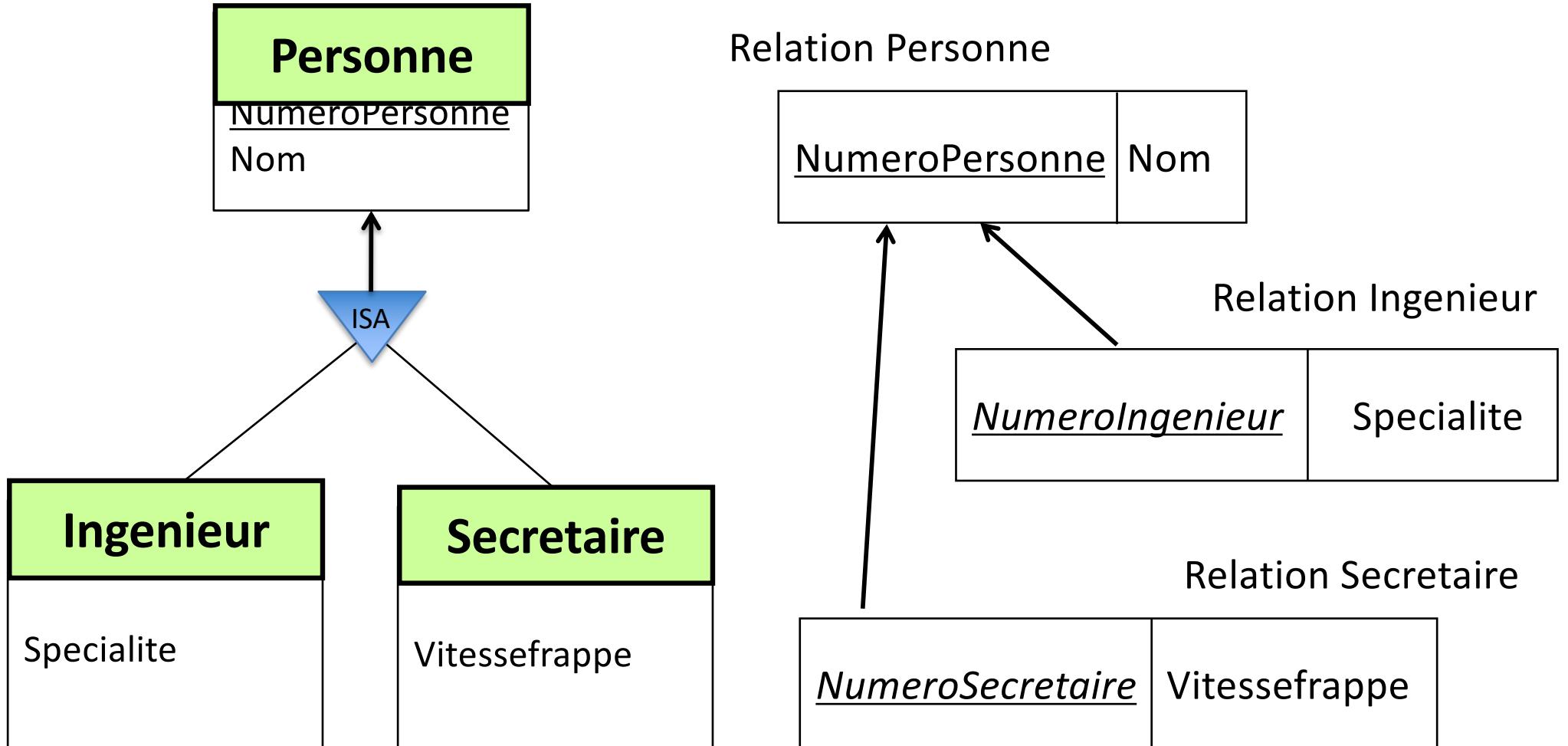
<u>IdB</u>	<u>IdA</u>	attB	att2B	prop	attC
------------	------------	------	-------	------	------

Choisir la relation ayant le moins d'attribut et la supprimer. Ici relation C



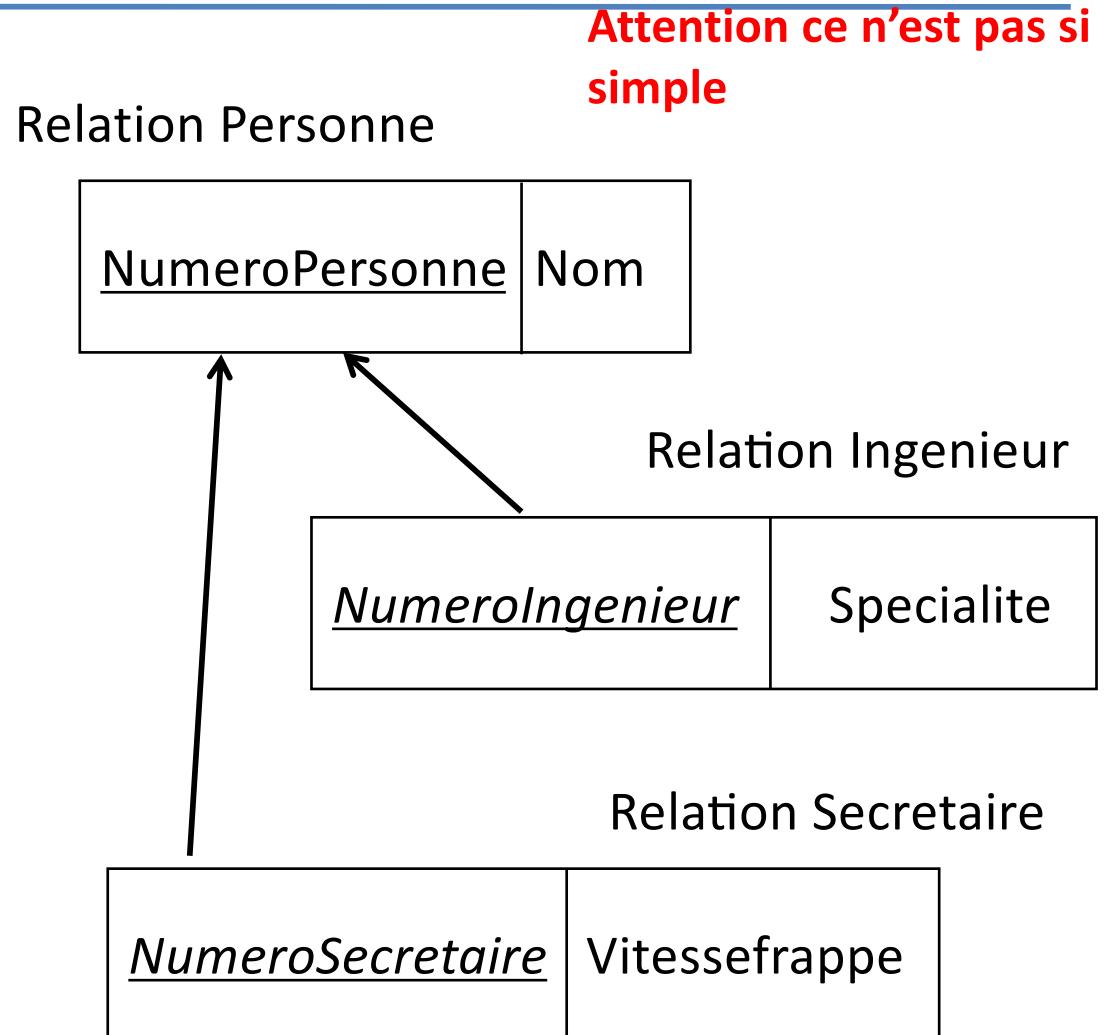
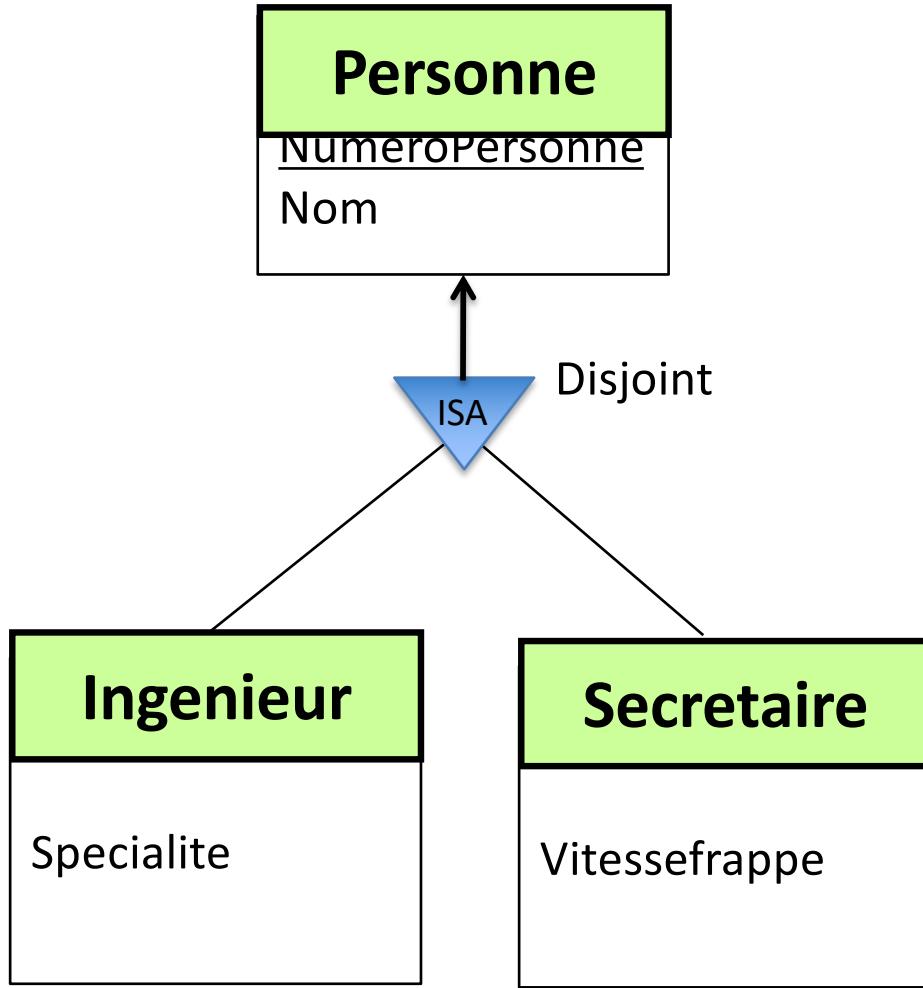
La clé est idC. idA est clé étrangère. attC est recopié ainsi que prop

Les héritages



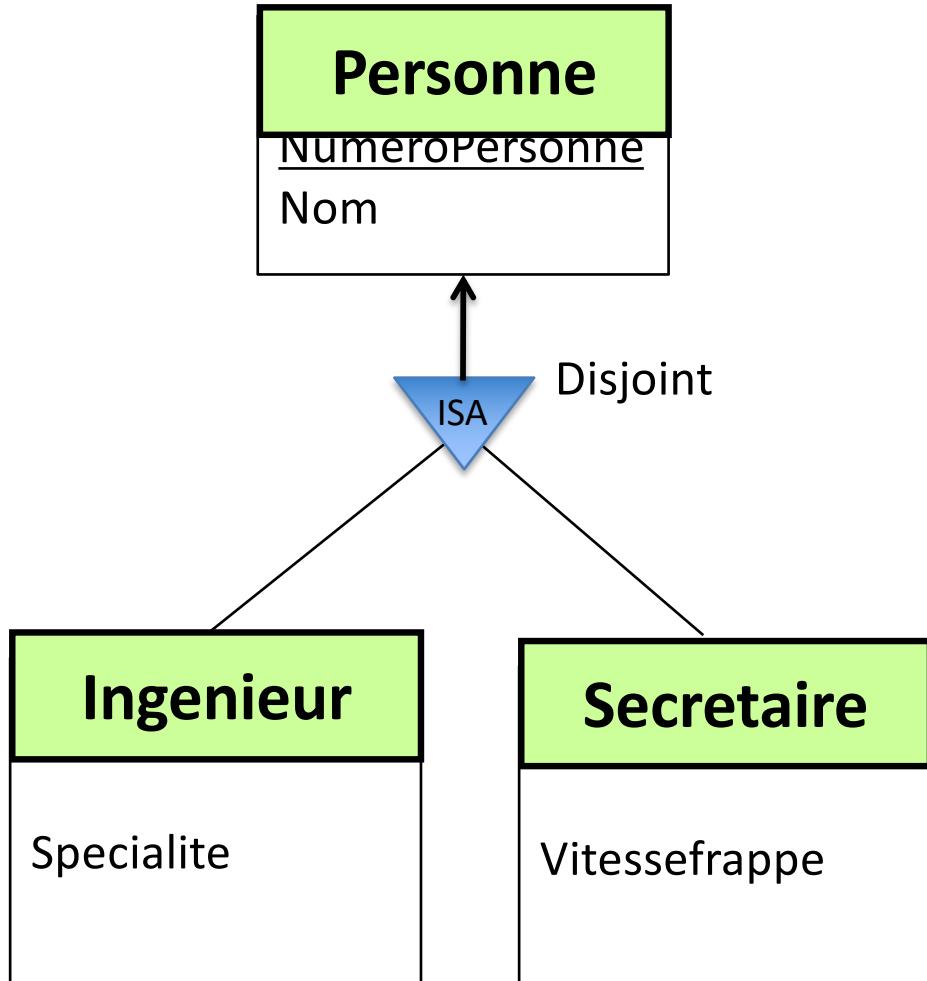
NumerolIngenieur et NumeroSecretaire sont des clés étrangères référençant NumeroPersonne. Il est conseillé de renommer l'attribut

Les héritages



Ajouter une contrainte à prendre en compte. On ne peut pas insérer un Numéro d'Ingénieur s'il est déjà dans Secrétaire et inversement.
Nécessité de passer par des vues, des triggers, etc

Les héritages



Suppression de personne

Relation Ingenieur

<u>NumerolIngenieur</u>	Nom	Specialite
-------------------------	-----	------------

Relation Secretaire

<u>NumeroSecretaire</u>	Nom	Vitessefrappe
-------------------------	-----	---------------

Ajouter une contrainte à prendre en compte. Les numéros d'ingénieurs ou de secrétaires sont pairs/impairs ?
Dans tous les cas il faut prendre cette contrainte en compte (notion de vues, de triggers, etc)

Au final

Client

NumeroClient
NomClient (NOT NULL)
Caution
AdresseNum
AdresseRue
AdresseVille

```
/*
Creation de la relation client avec les contraintes
*/
CREATE TABLE CLIENT (
    NUMEROCLIENT INT,
    NOMCLIENT VARCHAR(15) NOT NULL,
    CAUTION NUMERIC(3,0),
    ADRESSENUM NUMERIC(3,0),
    ADRESSERUE VARCHAR(25),
    ADRESSEVILLE VARCHAR(15),
    CONSTRAINT PK_Client (NUMEROCLIENT)
);

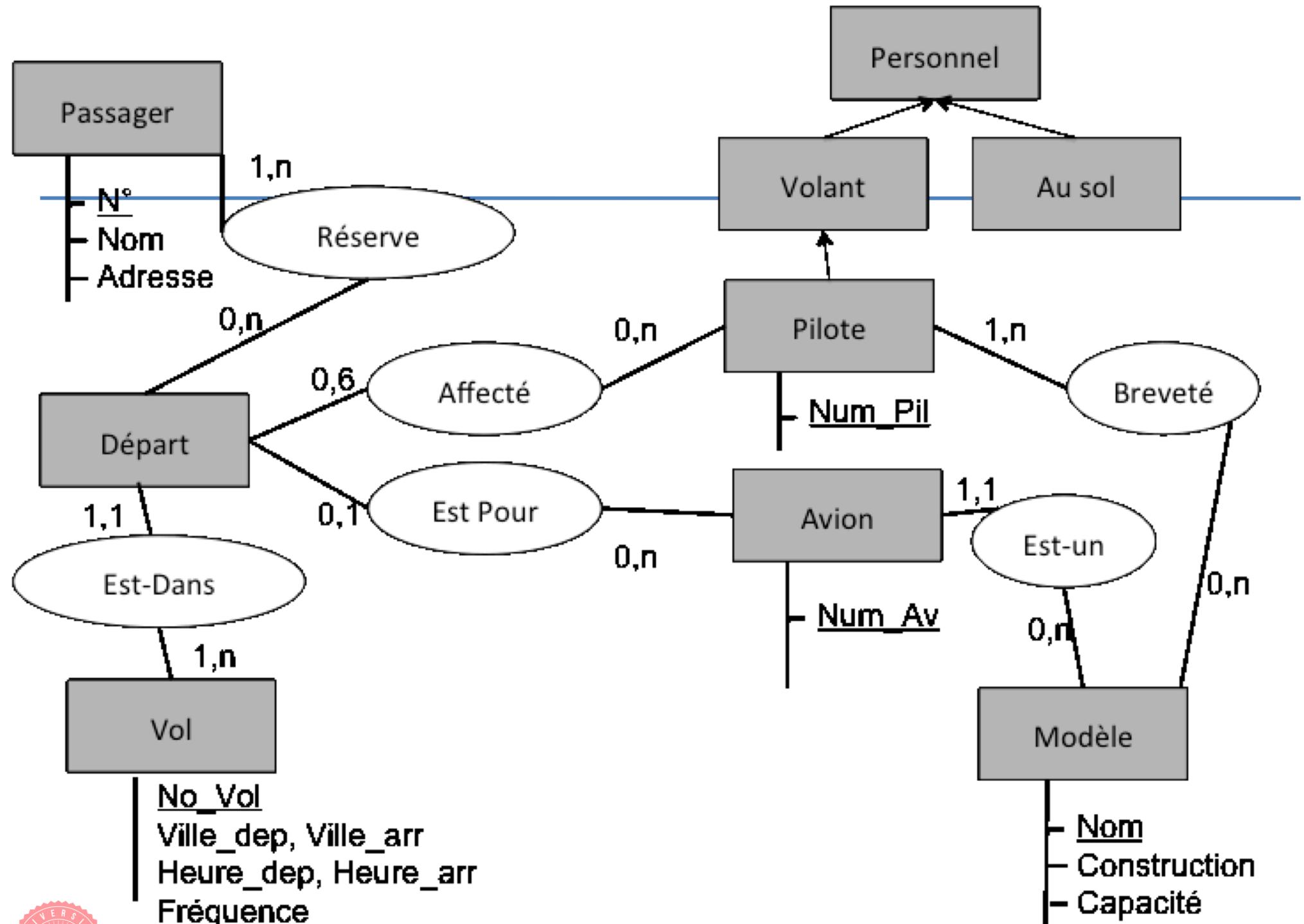
/*
Creation de l'index sur la cle primaire pour optimiser les jointures
*/
CREATE INDEX ICLIENT ON CLIENT(NUMEROCLIENT);
```



Pilote-Avion-Vol plus compliqué

- La compagnie veut conserver les coordonnées des passagers, y compris s'ils se sont désistés, ou après le départ
- Un passager peut avoir plusieurs réservations
- Un vol est une liaison entre 2 lieux. Il peut être régulier, tous les jours à telle heures, ou occasionnel
- En conséquence, un vol peut avoir plusieurs départs (un départ est un exemplaire d'un et un seul vol)
- La compagnie dispose d'un ensemble de personnels, dont certains sont des « volants » (pilotes ou personnels de bord) qui sont affectés au départ de certains vols, et d'autres sont à terre (entretien, accueil etc.)
- Il n'y a jamais plus de 6 « volants » affectés à un départ. Un départ peut n'avoir encore aucun personnel affecté
- La compagnie a décidé de ne considérer comme pilote que ceux qui sont brevetés pour au moins un des modèles d'avions qu'elle possède ou prévoit à terme. Elle prend en compte des modèles d'avion même si elle n'a pas encore de pilote breveté pour ce modèle





-
- Des questions ?

