

Bases de Données Relationnelles

Introduction Générale
Les modèles conceptuels et relationnels
L2 MIASH

Rafael Angarita
Maitre de Conférences
rangarit@parisnanterre.fr

Objectifs du module

- S'introduire aux Base de données Relationnelles
- Concevoir une BD par le modèle E/A
- La gestion des BD par MySQL
- Manipuler les données par les requêtes SQL
 - Création/modification/Interrogation d'une base de données par les requêtes SQL sous Mysql

Contexte général

- Élaborer le système d'information d'un organisme.
- « Un système d'information est l'ensemble des ressources (matériels, logiciels, données, procédures, humains, ...) **structurés** pour **acquérir, traiter et mémoriser** l'information (sous forme de données, textes, sons, images, ...) dans et entre les organisations. »

Robert Reix (1934-2006), systèmes d'information et management Des organisations, édition 1995.

Contexte général

Le SI représente l'ensemble des ressources (humaines, matérielles, logicielles) organisées pour :

- **Collecter l'information** : Enregistrer une information
(support papier, informatique...) avant son traitement
- **Mémoriser l'information (stockage)** : Conserver, archiver (utilisation ultérieure ou obligation légale)
- **Traiter l'information** : effectuer des opérations (calcul, tri, classement, résumé, ...)
- **Diffuser** : transmettre à la bonne personne (éditer, imprimer, afficher, ... une info après traitement)



Base de données

- Une Base de données est un **gros ensemble d'informations** structurées **mémorisées** sur un **support permanent**.
- Applications
 - Gestion (stocks, salaires, ...)
 - Réservations (avions, trains, ...)
 - Librairies et commerce électroniques
- Requêtes
 - Liste des produits qui ont été commandés par un client déterminé ;
 - A quel client la commande numéro X appartient-elle ?
 - Quelle est la date de la dernière commande du client s'appelant Y ?

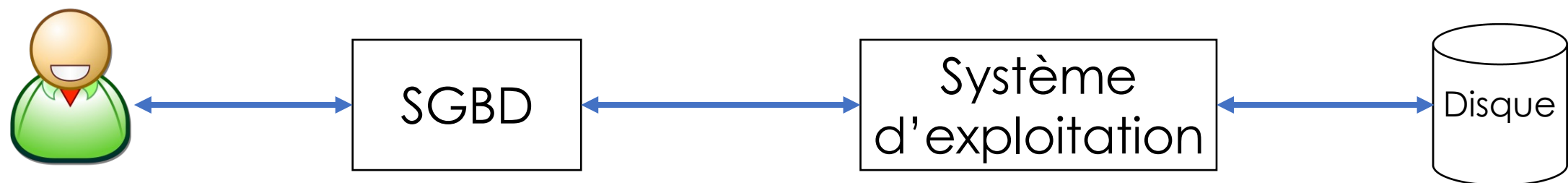
Stocker dans des fichiers ?

- Inconvénients
 - **Lourdeur d'accès** aux données
 - **Manque de sécurité**
 - **Partage** et gestion **des droits d'accès difficiles**
 - Pas de contrôle de **concurrency**
 - Capacité de **stockage très limitée**
 - Taille maximale des feuilles de calcul **Excel**
1 048 576 lignes et 16 384 colonnes
 - **Redondance des données** → mises à jour difficiles



SGBD: Système de Gestion de Base de Données

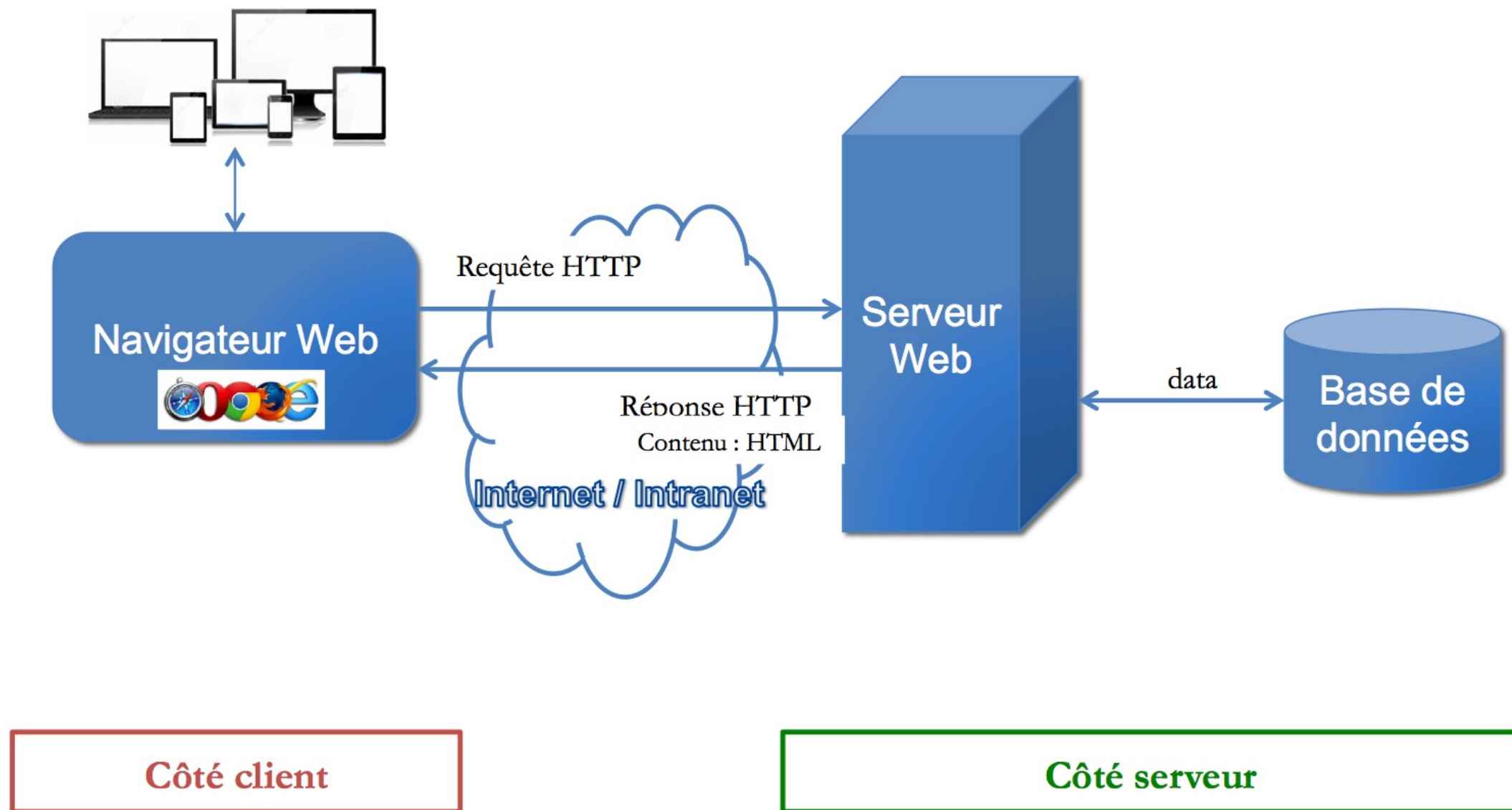
- Un Système de Gestion de Base de Données (SGBD) est un ensemble de logiciels permettant la recherche, la mise à jour et la sauvegarde de données stockées.
- En anglais: DBMS: Data Base Management System
- Exemples de SGBD: Access, Oracle, PostgreSQL, MySQL, ...



Systeme de Gestion de Base de Données assure

- La description des données
- leur recherche et mise à jour
- la sûreté
 - vérifier les droits d'accès des utilisateurs;
 - limiter les accès non autorisés ;
 - crypter les informations sensibles
- la sécurité
 - sauvegarde et restauration des données;
 - limiter les erreurs de saisie, de manipulation.
- l'intégrité : définir des règles qui maintiennent l'intégrité de la base de données (contraintes d'intégrité)
- la concurrence d'accès : détecter et traiter les cas où il y a conflit d'accès entre plusieurs utilisateurs.

Base de données et un site web



Donc on veut créer une base
de données (relationnelle)

On commence par quoi ?

Donc on veut créer une base
de données (relationnelle)

On commence par quoi ?

On a besoin d'un modèle !

Le modèle Entité/Association : avantages

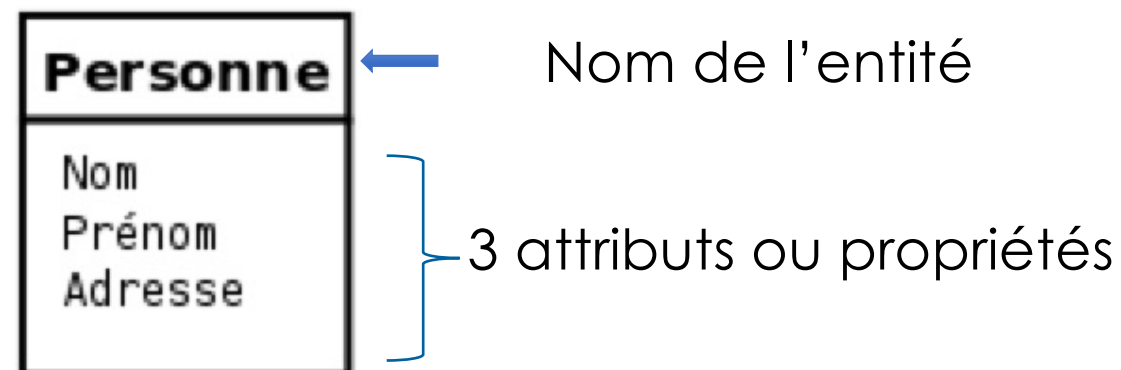
- Proposé en 1976 par Chen
- Le modèle Entité/Association (E/A) est utilisé à peu près universellement pour la conception de bases de données relationnelles
- Il est utilisé pour élaborer le **schéma conceptuel** des données
- Il repose sur une **représentation graphique** qui facilite considérablement sa compréhension par les acteurs de l'entreprise, non spécialistes en informatique
- Il intègre des concepts simples à appréhender : **Entité – Association**

Le modèle Entité/Association : concepts

- trois concepts de base
 - l'objet ou entité
 - L'attribut (appelé également propriété)
 - l'association
- L'**objet** est une entité ayant une existence propre.
- L'**association** est un lien ou relation entre objets sans existence propre.
- La **propriété** est la plus petite information décrivant un objet ou une association.

L'entité et ses attributs

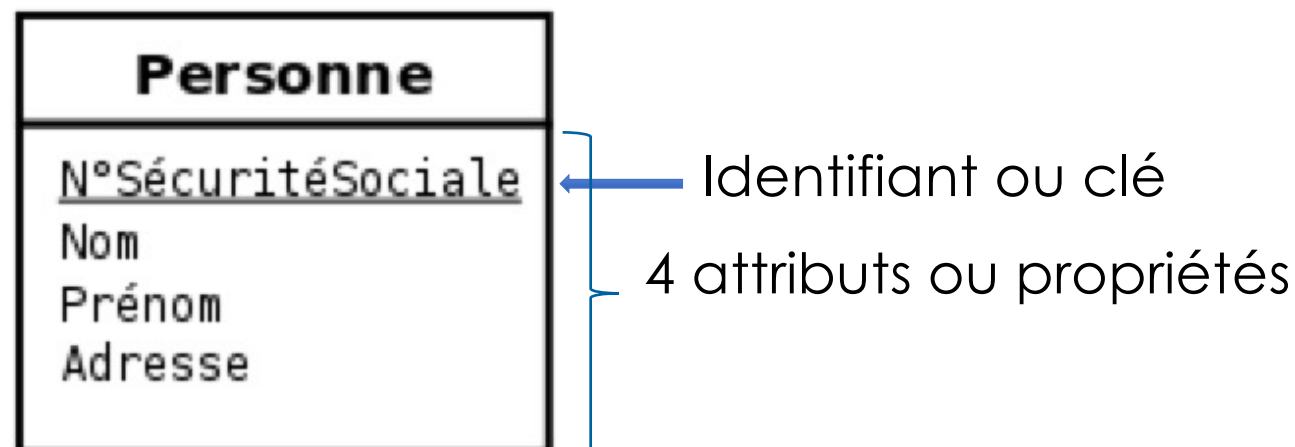
- Un attribut est une donnée élémentaire, brute
 - Ne pas confondre avec valeur de l'attribut! (exp: Nom et Dupont)



Exemple d'entité
(Dupont, Jean, 21 rue d'Assas 75006)

Identifiant ou clé

- **identifiant, clé** d'une entité est constitué par un de ses attributs qui doit avoir une valeur **unique** pour chaque entité
- Un identifiant peut être constitué de plusieurs attributs (clé composée)
 - Exemples de clés : numéro d'immatriculation d'une voiture, code barre d'un produit, numéro de sécurité sociale, entier auto-incrémenté, etc.



La clé doit être soulignée

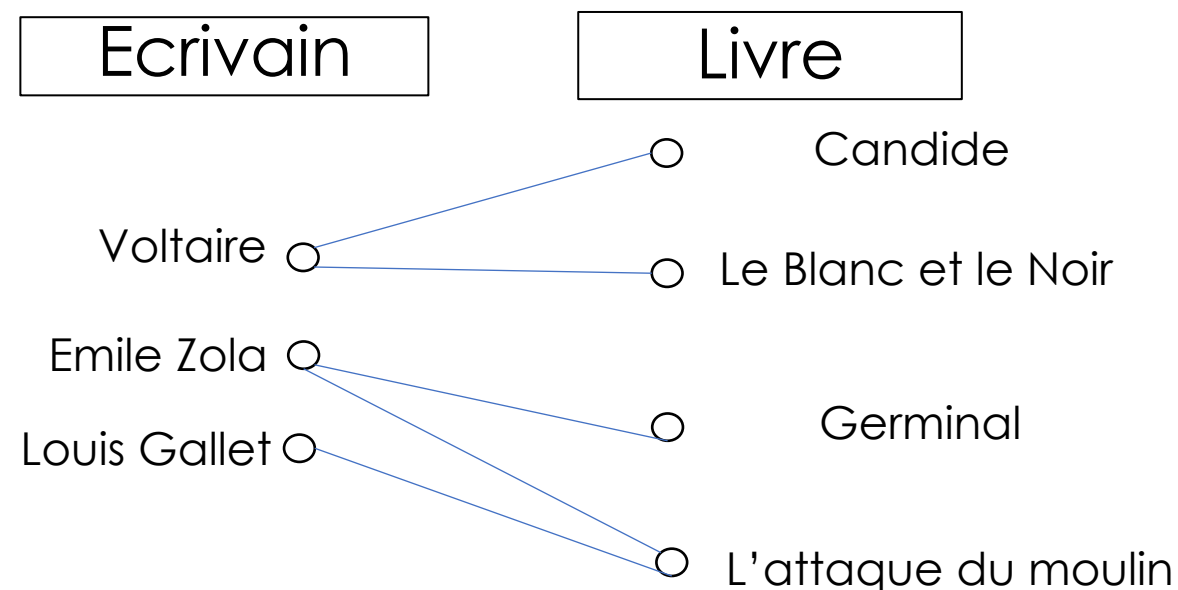
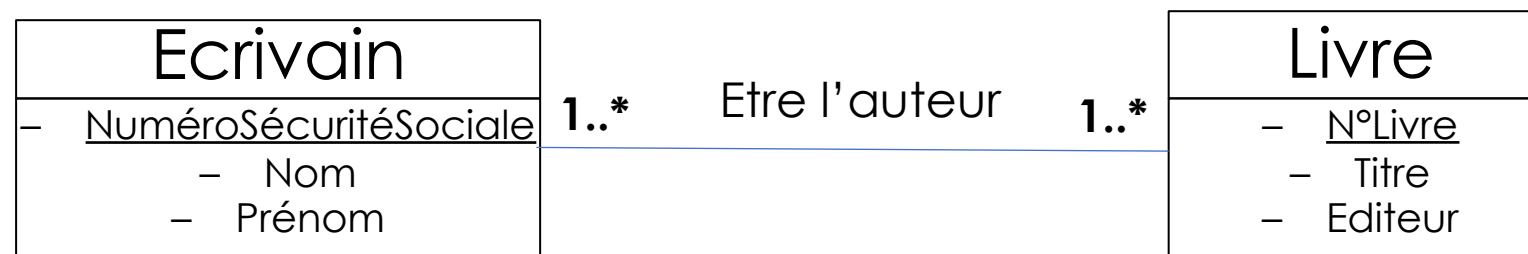
Association ou relation

- Une association (ou une relation) est un **lien** entre plusieurs entités :
 - Un livre est emprunté par une personne → association: Emprunt ou emprunter
 - Une voiture est possédée par une personne → association: possession ou posséder
- Une association peut avoir des attributs (propriétés)
- Elle peut relier plusieurs entités ensemble
- La plupart des associations sont de degré 2 (liaison entre 2 entités)
- Une association possède une **cardinalité** (*multiplicity*)
 - Nombre d'instance d'une entité pour chaque instance de l'autre

Association et cardinalités

- Notations de cardinalité

- UML: min..max 0..1 1..1 or 1 0..* 1..* ou *



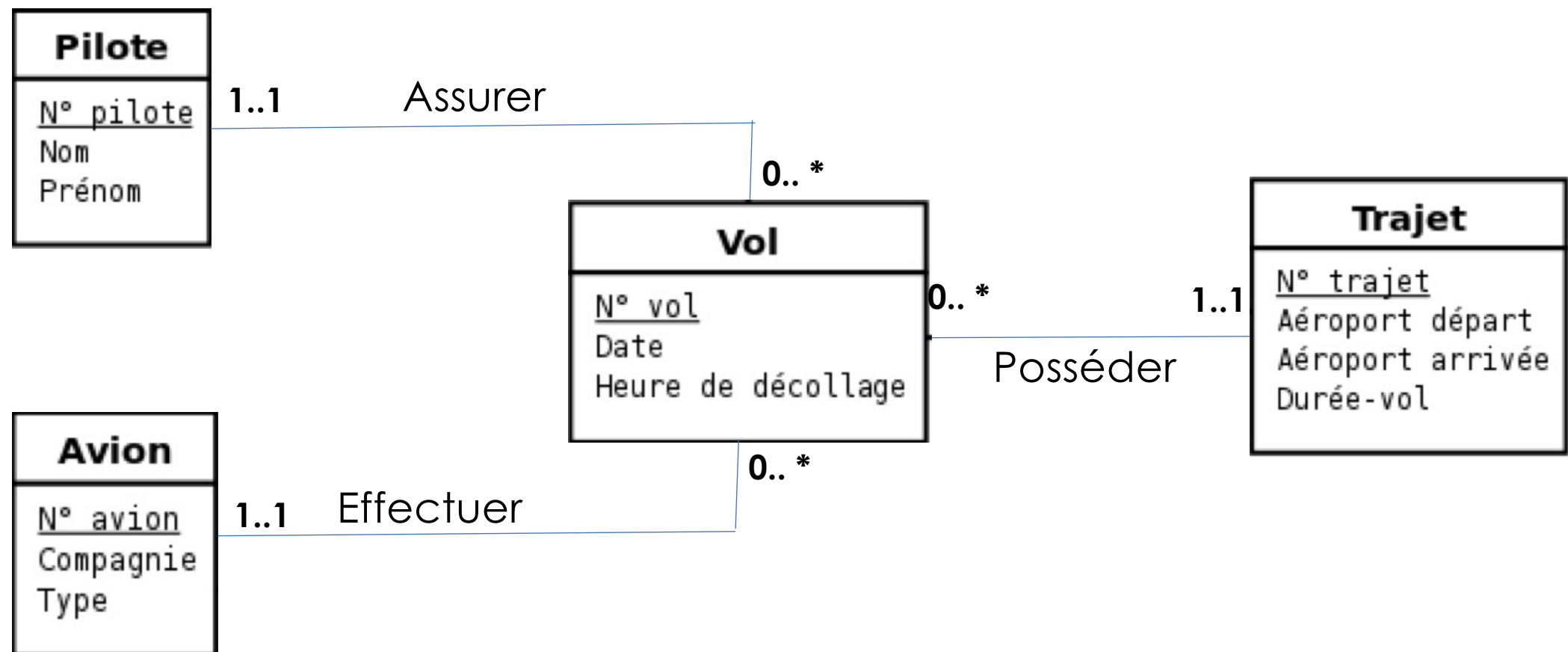
Exemple d'association

- Représentation diagramme de classe UML : *Unified Modeling Language*

Une personne A possède plusieurs voitures.
Une voiture est possédée par une seule personne



Modèle conceptuel / Exemple



Des besoins vers un modèle conceptuel : Exemple

Nous souhaitons créer un site Web autour des séries télévisées.

Une série se définit avant tout par un titre, un ou plusieurs genres et une chaîne de diffusion.

Une chaîne diffuse plusieurs séries, et une série n'est diffusée que sur une seule chaîne.

Une série peut appartenir à plusieurs genres.

Une série est composée d'un ensemble d'épisodes, qui sont classés par saison.

Chaque épisode est donc décrit par un titre, une date de première diffusion, un numéro de saison et le numéro de l'épisode dans cette saison.

Un acteur est décrit simplement par son nom et son prénom.

Un acteur peut jouer dans plusieurs épisodes, on précisera son rôle dans l'épisode en donnant le nom et le prénom du personnage joué.

Des besoins vers un modèle conceptuel : Exemple

1- Identification des entités

Nous souhaitons créer un site Web autour des séries télévisées.

Une **série** se définit avant tout par un titre, un ou plusieurs **genres** et une **chaîne de diffusion**.

Une **chaîne** diffuse plusieurs **séries**, et une **série** n'est diffusée que sur une seule **chaîne**.

Une **série** peut appartenir à plusieurs **genres**.

Une **série** est composée d'un ensemble d'**épisodes**, qui sont classés par saison.

Chaque **épisode** est donc décrit par un titre, une date de première diffusion, un numéro de saison et le numéro de l'épisode dans cette saison.

Un **acteur** est décrit simplement par son nom et son prénom.

Un **acteur** peut jouer dans plusieurs **épisodes**, on précisera son rôle dans l'épisode en donnant le nom et le prénom du personnage joué

Des besoins vers un modèle conceptuel : Exemple

2- Identification des associations

Nous souhaitons créer un site Web autour des séries télévisées.

Une **série** se définit avant tout par un titre, **un ou plusieurs genres** et **une chaîne de diffusion**.

Une **chaîne diffuse plusieurs séries**, et une **série n'est diffusée que sur une seule chaîne**.

Une **série peut appartenir à plusieurs genres**.

Une **série** est **composée d'un ensemble** d'**épisodes**, qui sont classés par saison.

Chaque épisode est donc décrit par un titre, une date de première diffusion, un numéro de saison et le numéro de l'épisode dans cette saison.

Un **acteur** est décrit simplement par son nom et son prénom.

Un **acteur peut jouer dans plusieurs épisodes**, on précisera son rôle dans l'épisode en donnant le nom et le prénom du personnage joué

Des besoins vers un modèle conceptuel : Exemple

3- Identification des attributs (propriétés)

Nous souhaitons créer un site Web autour des séries télévisées.

Une **série** se définit avant tout par un **titre**, **un ou plusieurs genres** et **une chaîne de diffusion**.

Une **chaîne diffuse plusieurs séries**, et une **série n'est diffusée que sur une seule chaîne**.

Une **série peut appartenir à plusieurs genres**.

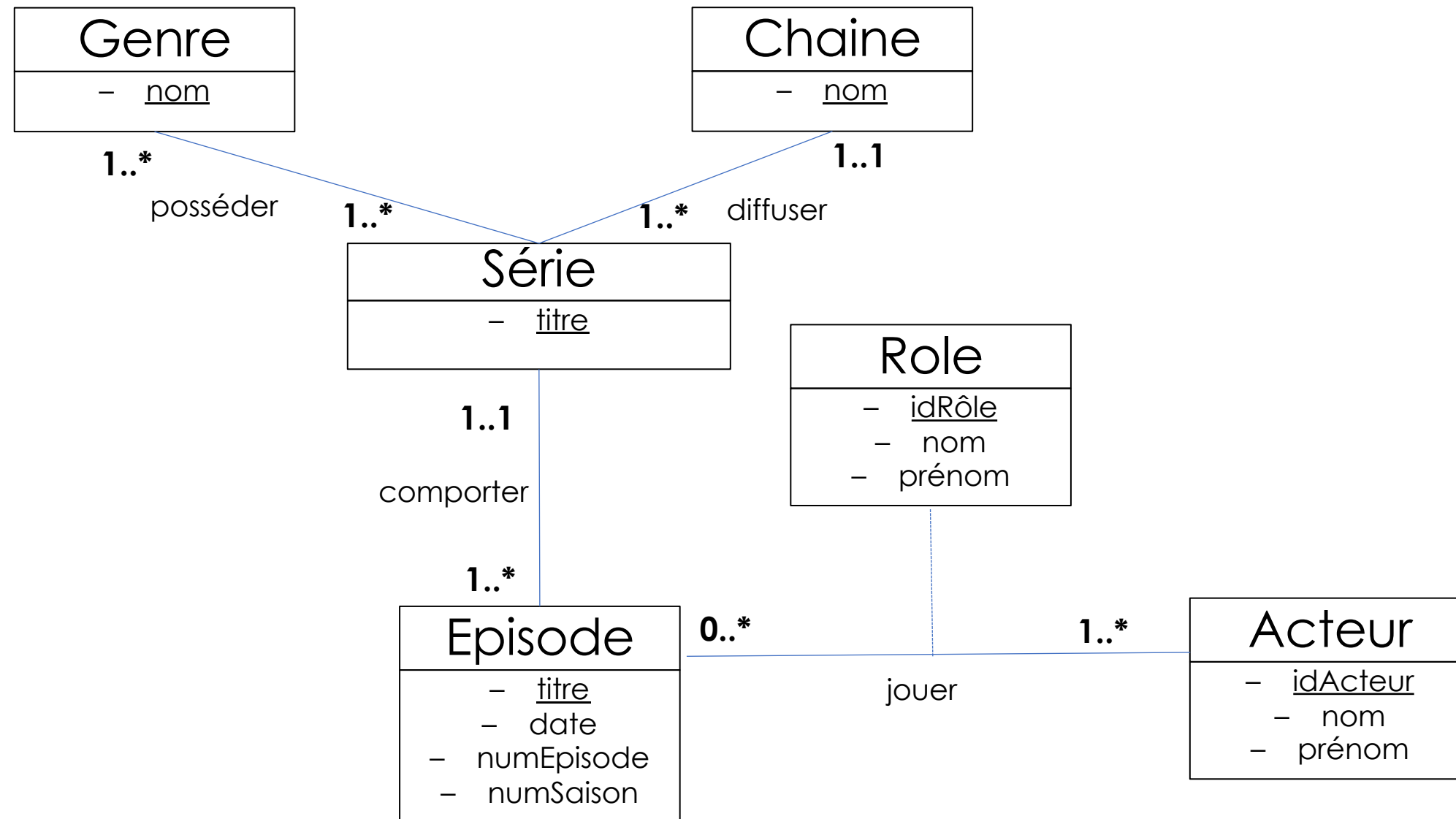
Une **série** est **composée d'un ensemble d'épisodes**, qui sont classés par saison.

Chaque épisode est donc décrit par un, **titre**, une **date de première diffusion**, un **numéro de saison** et le **numéro de l'épisode** dans cette saison.

Un **acteur** est décrit simplement par son **nom** et son **prénom**.

Un **acteur peut jouer dans plusieurs épisodes**, on précisera son **rôle** dans l'épisode en donnant le **nom** et le **prénom du personnage** joué

Des besoins vers un modèle conceptuel : Solution



Modèle conceptuel : bonnes pratiques

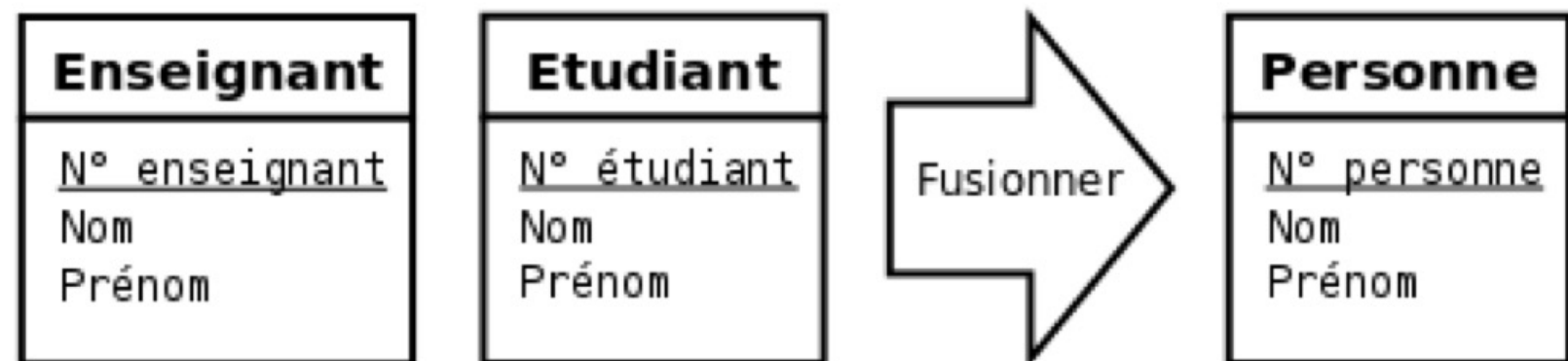
- Bien comprendre le problème à résoudre
- Essayer de conserver le modèle simple
- Bien choisir les noms
- Ne pas cacher les associations sous forme d'attributs
 - Utiliser les associations

Conception: démarche à suivre

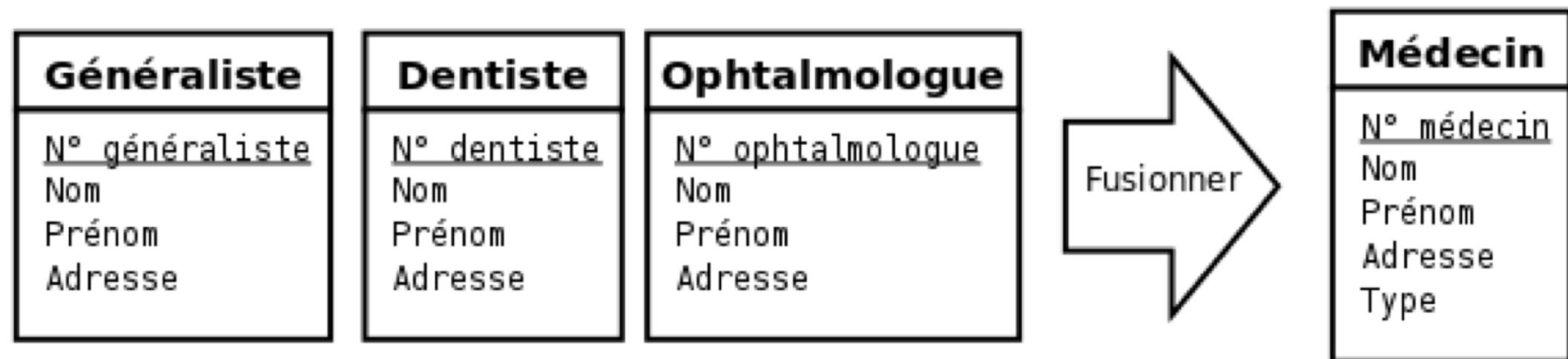
- Repérer les types d'entités
- Pour chaque entité
 - Lister les attributs
 - Définir l'identifiant
- Lister les associations (liaison entre les entités)
- Pour chaque association
 - Lister les attributs
 - Fixer les cardinalités
- Vérifier le modèle construit
 - Supprimer les transitivités (redondance)
 - Minimiser les traits qui se croisent
 - Est ce que le modèle répond à tous les besoins?

Erreurs à éviter: répétition d'attributs

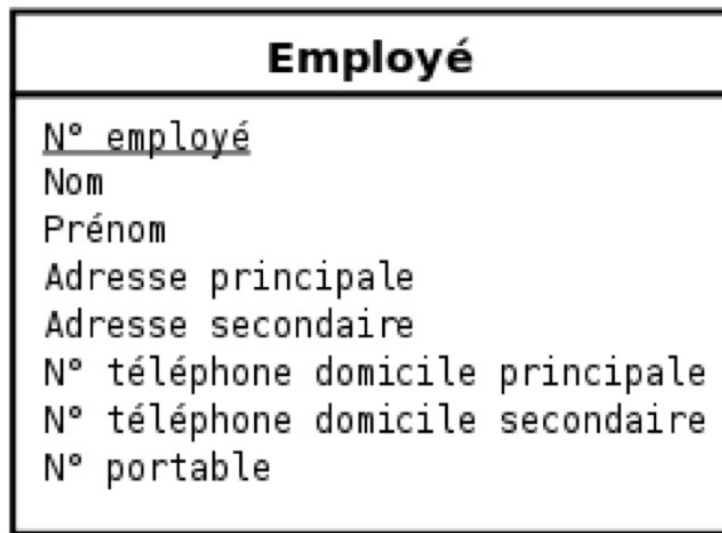
- Utiliser la généralisation pour éviter la répétition d'attributs dans les types d'entités



- Utiliser un attribut "type" pour supprimer des types d'entités inutiles

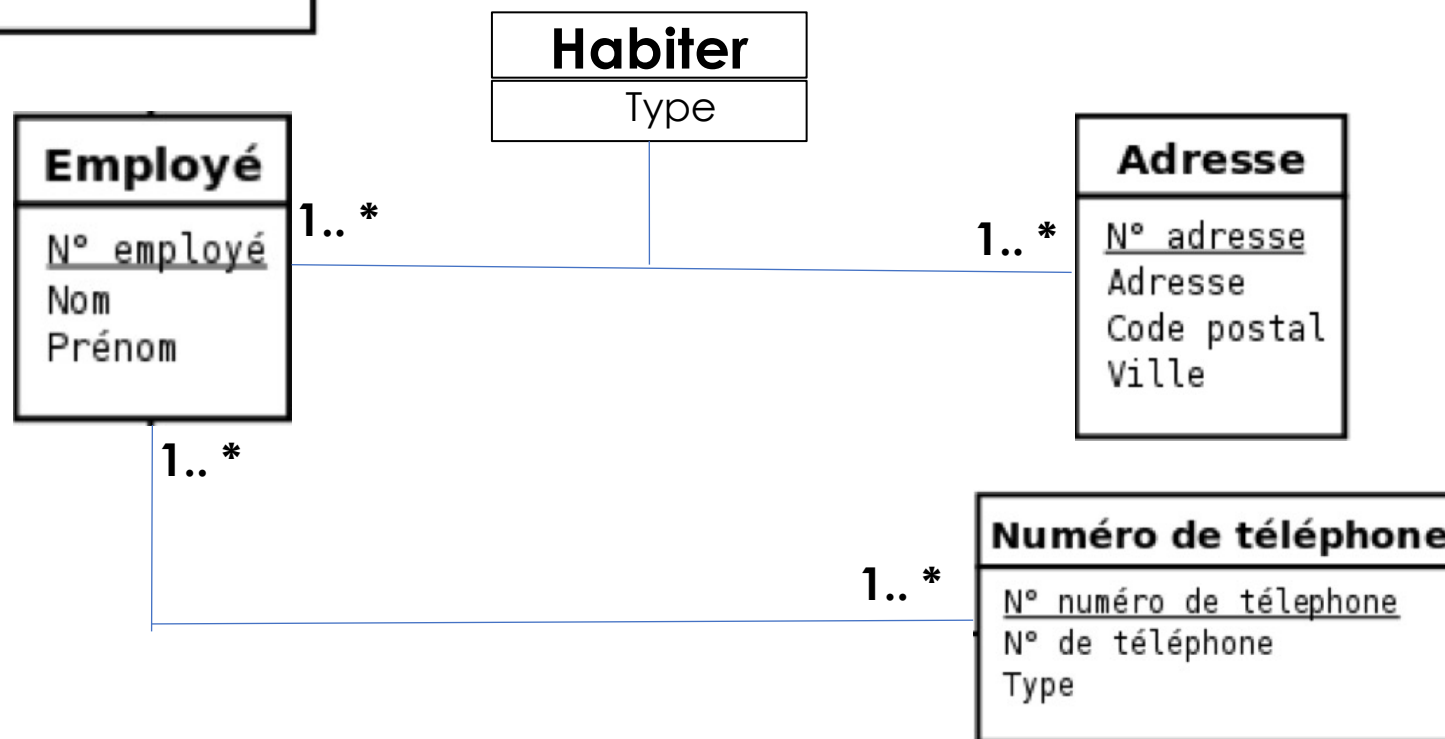


Erreurs à éviter: attributs complexes



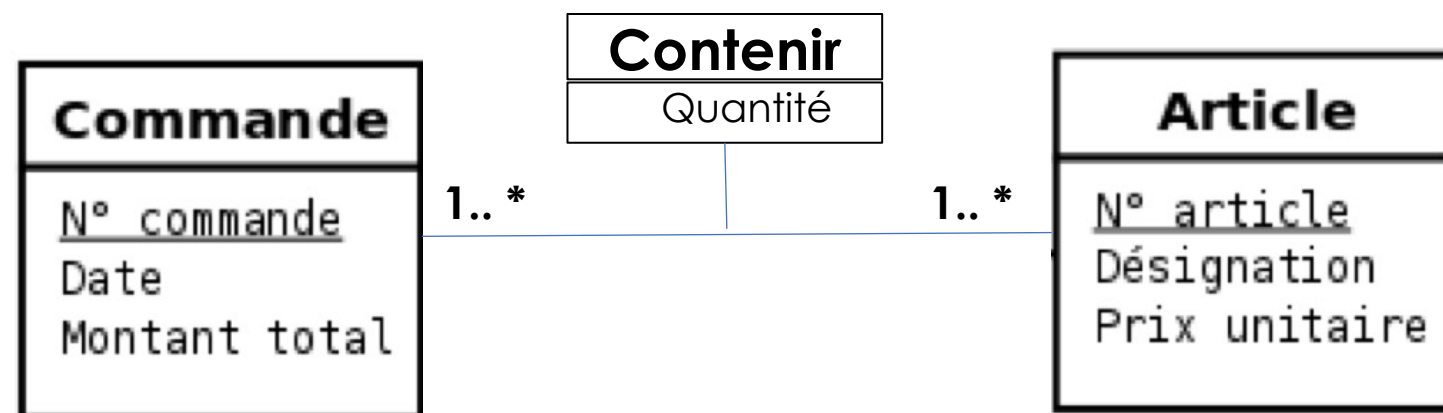
■ Limites du modèle

- Comment faire si un employé possède deux adresses secondaires ou plusieurs numéros de portable ?
- La ville où habite l'employé est noyée dans un unique attribut Adresse.



Erreurs à éviter: attributs dérivés d'autres attributs

- Il ne faut jamais ajouter un attribut dérivé d'autres attributs
- Risque d'incohérence entre les valeurs des attributs de base et celles des attributs dérivés



- Il faut supprimer l'attribut *Montant total* du type entité *Commande* car on peut le calculer à partir des attributs

Exercices à faire à la maison

Exercice 1: Dessiner un modèle conceptuel (avec UML) permettant de représenter les données décrites

- On souhaite gérer les vols reservables dans une agence en modélisant les informations relatives aux vols afin de les stocker dans une base de données. Pour cela, on dispose des renseignements suivants
 - Une compagnie aérienne possède un nom et propose différents vols
 - Un vol a un numéro, un aéroport de départ, un aéroport d'arrivée, une heure de départ et une durée
 - Un aéroport a un nom et une superficie
 - Un vol peut comporter des escales dans un ou plusieurs aéroport(s)
 - Une escale a une heure d'arrivée et une heure de départ
 - Chaque aéroport dessert une ou plusieurs villes
 - Une ville a un nom et un nombre d'habitants.

Exercice 2 : Dessiner un modèle conceptuel (avec UML) permettant de représenter les données décrites

- On souhaite mettre en place une base de données pour le suivi des films projetés dans les salles de cinéma du département
- Une salle de cinéma ne projette qu'un seul film à une heure donnée. Un même film peut être projeté simultanément dans plusieurs salles.
- Une salle de cinéma ne projette chaque film qu'une seule fois par jour et toujours à la même heure
- Chaque spectateur émet une appréciation qui peut être résumée par bien, quelconque, nul. On ne s'intéresse pas à l'information sur la salle dans laquelle il a regardé ce film.
- On dispose pour chaque salle des données suivantes : nom, adresse et liste des films projetés avec l'heure de leur projection dans la salle. Les informations stockées sont celles de la semaine en cours. Chaque spectateur est identifié par un numéro. On connaît d'autre part son nom, son prénom, son adresse, sa date de naissance et sa catégorie professionnelle. Pour chaque film, on souhaite stocker son visa d'exploitation, son titre, le nom du réalisateur et son année de sortie. Enfin, on enregistre, pour chaque spectateur interrogé, la liste des films visionnés et son impression sur chacun des films.

Exercice 3 : Dessiner un modèle conceptuel (avec UML) permettant de représenter les données décrites

- Un médecin identifié par un numéro de sécurité sociale nSS et ayant un nom, un prénom, une adresse etc... prescrit une ordonnance à un patient. Un patient est identifié par un nSS et possède un nom, un prénom, une adresse etc...
- Cette ordonnance est datée et peut demander des tests identifiés par un libellé. Un test effectué est identifié par un numéro et a un libellé, et un résultat. Le test se déroule pour un malade donné à une date donnée.

Sources bibliographiques

- <http://www.ksi.mff.cuni.cz/~svoboda/courses/2015-1-A7B36DBS/>

Modèle relationnel

Modèle relationnel

- Conçu par Edgar Frank Codd (IBM) en 1970
- Modèle relationnel
 - modèle conceptuel (ou E/A) → Schéma de base de données relationnel
 - Permet de représenter les ensembles d'entités et d'associations par des tables
 - Utilisable dans tout SGBD relationnel
- 3 concepts fondamentaux
 - La table ou la relation
 - L'attribut
 - Le domaine: les valeurs possibles pour un attribut particulier

Exemple de table

Nom de la table

Attribut clé

Attribut

Valeur de l'attribut

Personne

N° Sécu	Nom	Prénom
354338532195874	Durand	Caroline
345353545435811	Dubois	Jacques
173354684513546	Dupont	Lisa
973564213535435	Dubois	Rose-Marie

schéma de relation Personne(N° sécu, Nom, Prénom)

Ligne ou Tuple

Table ou relation: les règles

- Une **table** ou **relation** est un ensemble de tuples représenté sous la forme tabulaire et ayant les propriétés suivantes
 - Chaque table porte un **nom unique**.
 - À l'intérieur de la table, le nom de chaque attribut est unique et désigne une colonne avec des propriétés spécifiques.
 - Une table peut contenir un nombre quelconque d'attributs, l'ordre des colonnes dans la table est indifférent.
 - Une table peut contenir un nombre quelconque de tuples, l'ordre des tuples dans la table est indifférent.
 - L'un des attributs identifie de façon unique chaque tuple dans la table et sera la **clé primaire**.
- Une clé primaire doit être
 - Unique
 - Minimale

Passage du modèle conceptuel au modèle relationnel

Les règles du passage

- Chaque **type entité** donne naissance à **une relation**.

Personne
- <u>N° Sécu</u>
- Nom
- Prénom



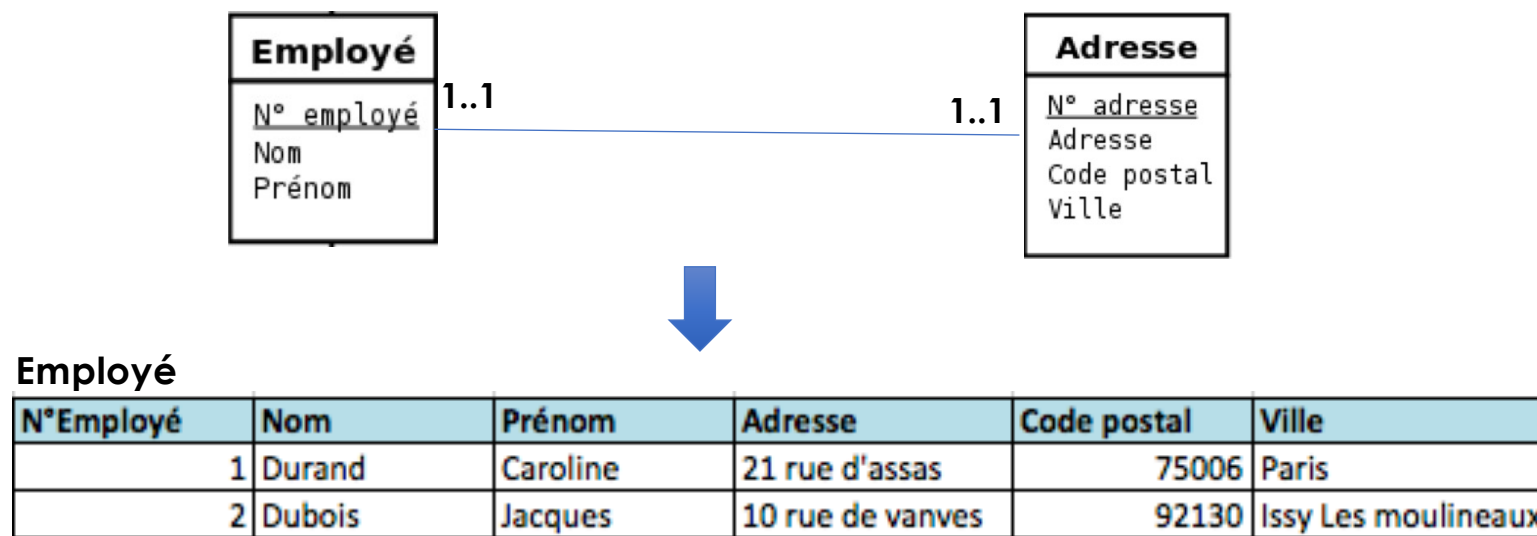
Personne

N° Sécu	Nom	Prénom
354338532195874	Durand	Caroline
345353545435811	Dubois	Jacques
173354684513546	Dupont	Lisa
973564213535435	Dubois	Rose-Marie

Personne (N° sécu, Nom, Prénom)

- La traduction d'**une association** dépend de **ses cardinalités**.

Association du type 1 vers 1

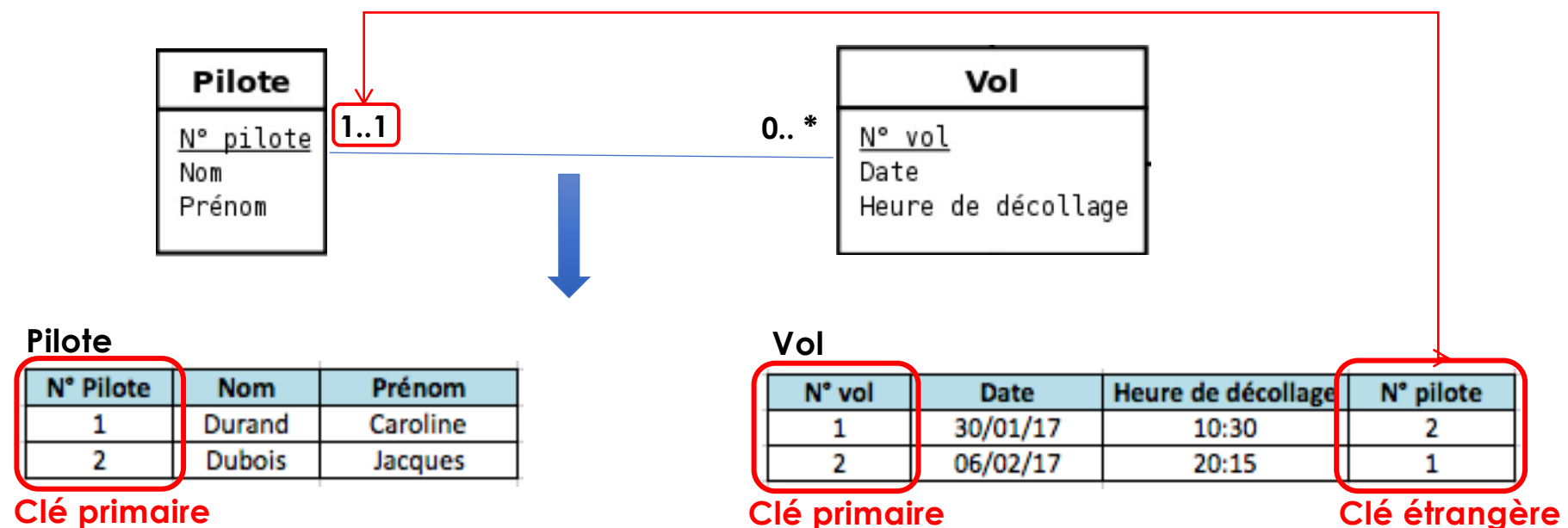


Employé (N° Employé, Nom, Prénom, Adresse, CodePostal, Ville)

- Une seule table, fusionner tous les attributs, garder une seule clé primaire.

Association du type un vers plusieurs

- Associations: **1..1 – 0..*** or **1..1 – 1..***



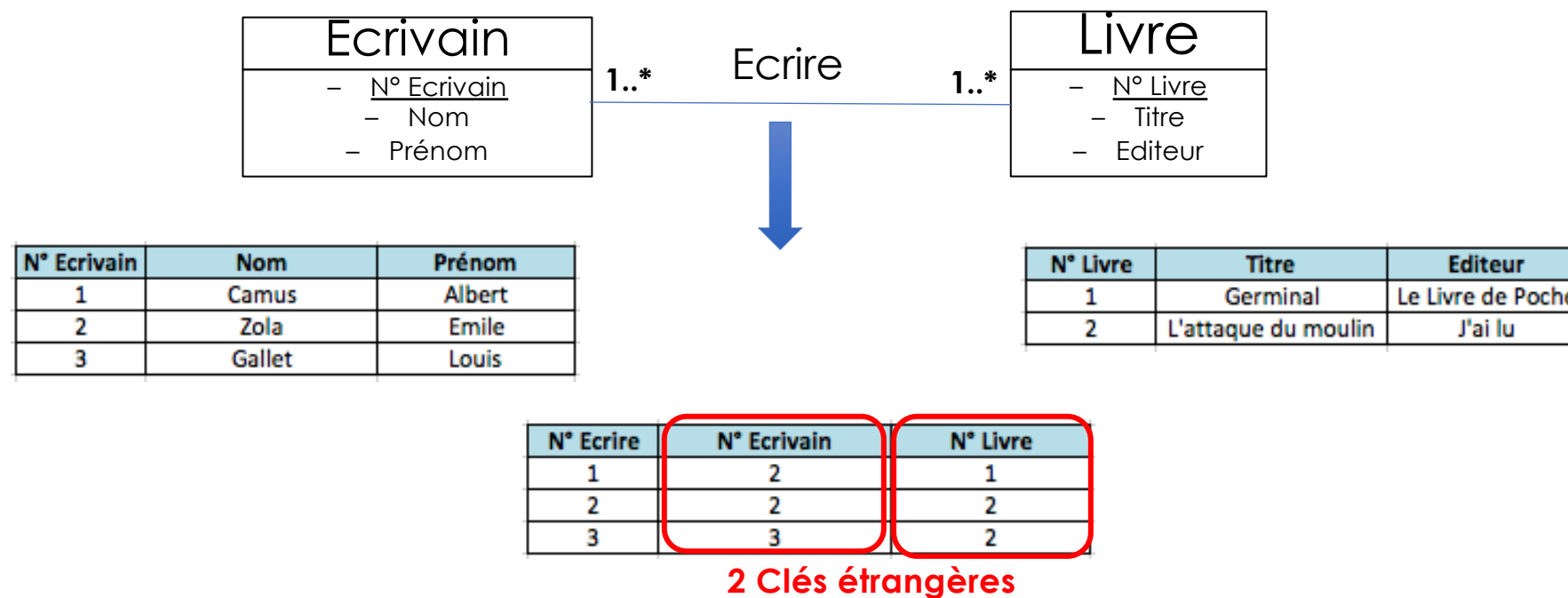
Pilote (N° Pilote, Nom, Prénom)

Vol (N° Vol, Date, heure de décollage, N° Pilote)

- 2 tables, ajouter une clé étrangère à la table de cardinalité « plusieurs ».

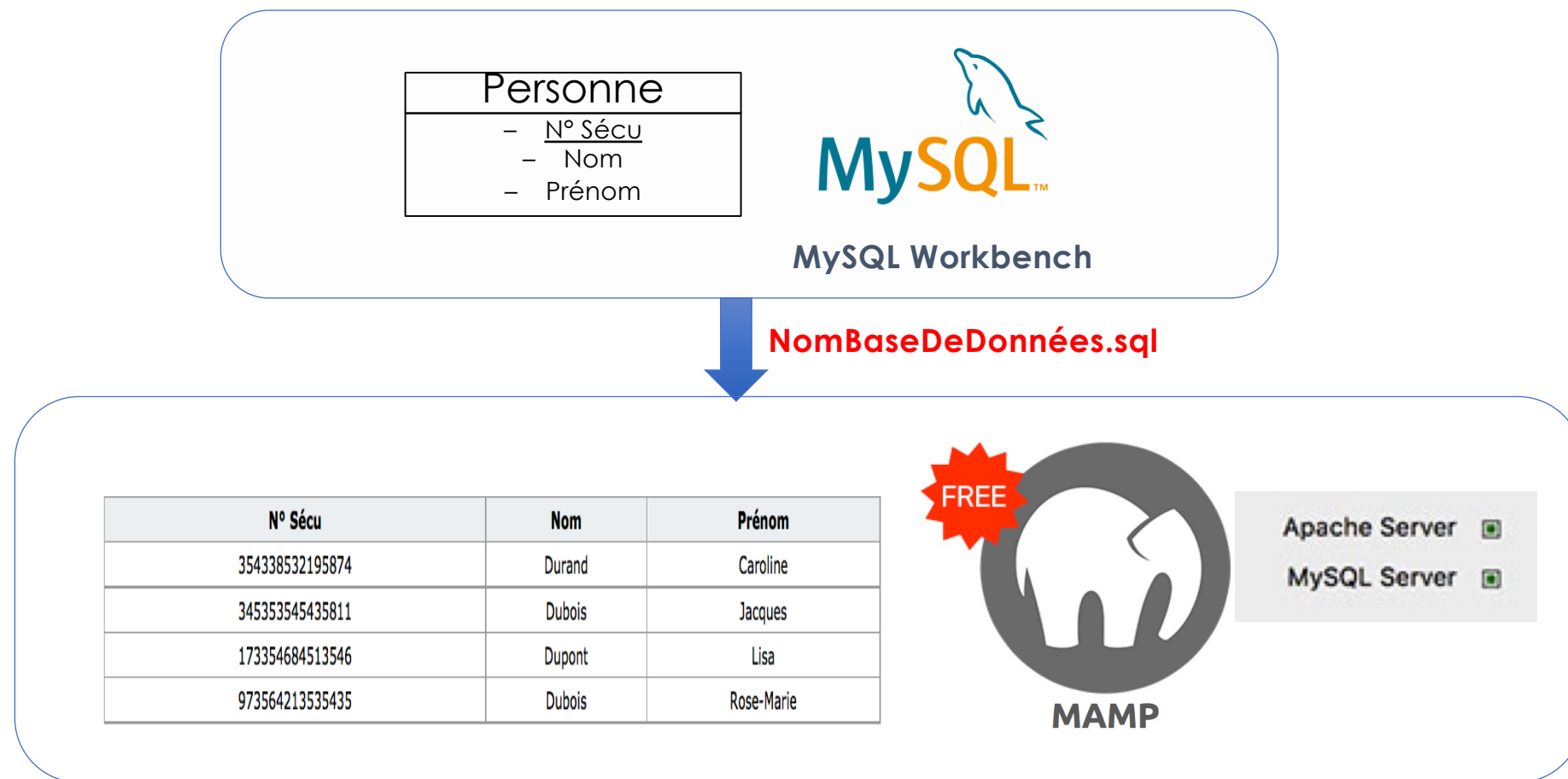
Association du type plusieurs vers plusieurs

- Associations: **0..* – 0..* or 1..* – 1..***



- Transformer l'association « Ecrire » en une table avec 2 clés étrangères et une clé primaire
→ 3 tables en tout

Logiciels



Passage de MySQLWorkbench → phpMyAdmin

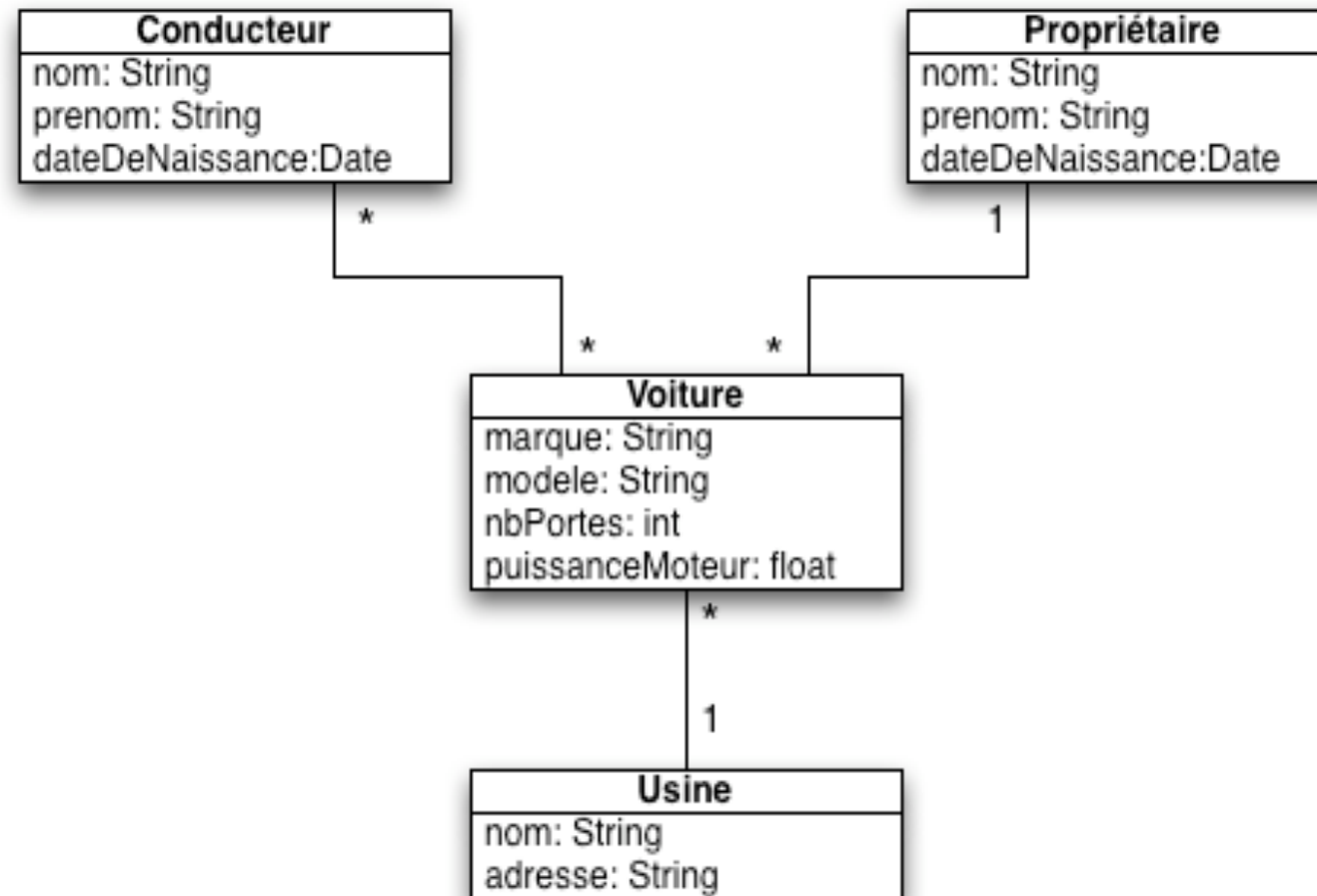
- **MySQLWorkbench**

- 1: Créer les types d'entités et les associations sous MySQLWorkbench
- 2: File → export → export as single page pdf → **fichier.pdf**
- 3: File → export → forward engineer SQL CREATE Script → **fichier.sql**

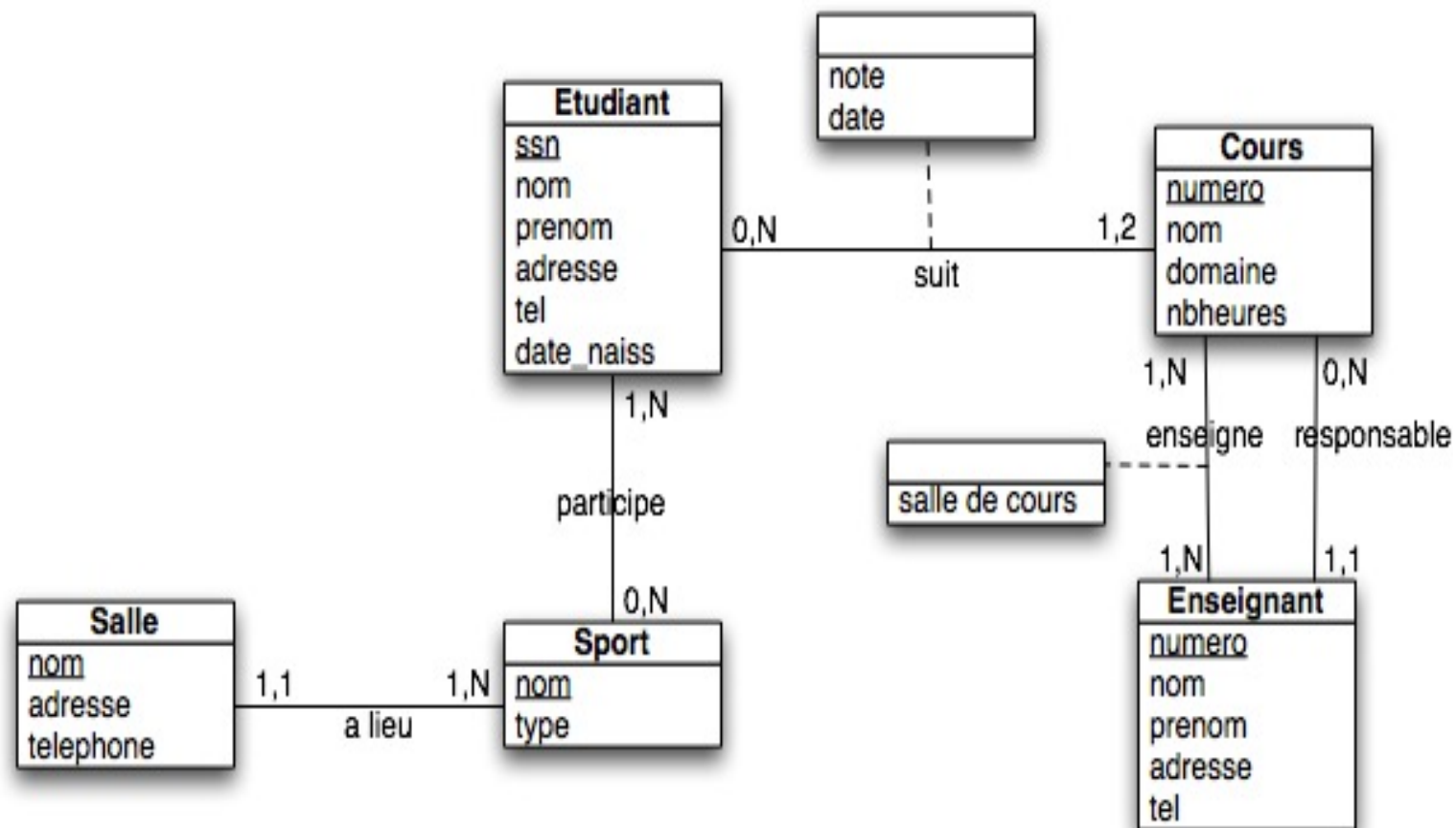
- **Mamp, phpMyAdmin**

- 4: Importer fichier.sql permet de créer automatiquement le modèle relationnel
- Le passage de Mamp vers MySQLWorkbench est aussi possible, mais moins utile
 - MySQLWorkbench → File → export → reverse engineer SQL CREATE Script → **fichier.sql**

Exercice 1 : faire le modèle relationnel



Exercice 2 : faire le modèle relationnel



Exercice 3 : Faire le modèle relationnel

