

Bases de données relationnelles

-

Modélisation entités-associations
et
Modèle relationnel

2023-2024

NEGRE Elsa

Programme

- CM/TD (5 séances) :
 - Modélisation et Modèle relationnel
 - SQL
- TP (1 séance) :
 - SQL (J. ZHONG)
- Evaluation : 100% Examen

Introduction

- **Base de données :**

- collection d'informations ou de données qui existent sur une **longue période de temps** [UW97] et qui décrivent les activités d'une ou plusieurs organisations [RG00]
- ensemble de données **modélisant les objets d'une partie du monde réel** et servant de support à une application informatique [Gar99]
- *Exemple 1 :*
 - *Organisation: une bibliothèque*
 - *Données: les livres, les emprunts, les emprunteurs, ...*
- *Exemple 2 :*
 - *Organisation: une Université*
 - *Données: les étudiants, les enseignants, les cours, ...*

Introduction

- **SGBD** : Systèmes de Gestion de Bases de Données (*DataBase Management Systems - DBMS*)
 - **ensemble de logiciels systèmes** permettant aux utilisateurs d'insérer, de modifier, et de rechercher efficacement des données spécifiques dans **une grande masse d'informations** (pouvant atteindre plusieurs milliards d'octets) **partagée par de multiples utilisateurs** [Gar99]
 - *Exemples : MySQL, PostgreSQL (utilisé en TP), Oracle, Microsoft SQLServer, ...*
 - *<https://db-engines.com/en/ranking/relational+dbms>*

SGBD

- Principaux composants :
 - Système de gestion de fichiers
 - Gestionnaire de requêtes
 - Gestionnaire de transactions
- Principales fonctionnalités :
 - Contrôle de la redondance d'information
 - Partage des données (car plusieurs utilisateurs en même temps)
 - Gestion des autorisations d'accès
 - Vérifications des contraintes d'intégrité
 - Sécurité et reprise sur panne

Exemple de transaction

Virement d'une somme X d'un compte A vers un compte B :

1. Vérifier que $SoldeA \geq X$ (Lecture)
2. $SoldeA = SoldeA - X$ (Ecriture)
3. $SoldeB = SoldeB + X$ (Ecriture)

Atomicité: les 3 opérations seront effectuées ou aucune

Cohérence: la base est cohérente au début de la transaction et à la fin de son exécution

Isolation: les mises à jour de la transaction ne sont visibles qu'à la fin de son exécution

Durabilité: une fois validées, les mises à jour doivent être pérennes même en cas de panne.

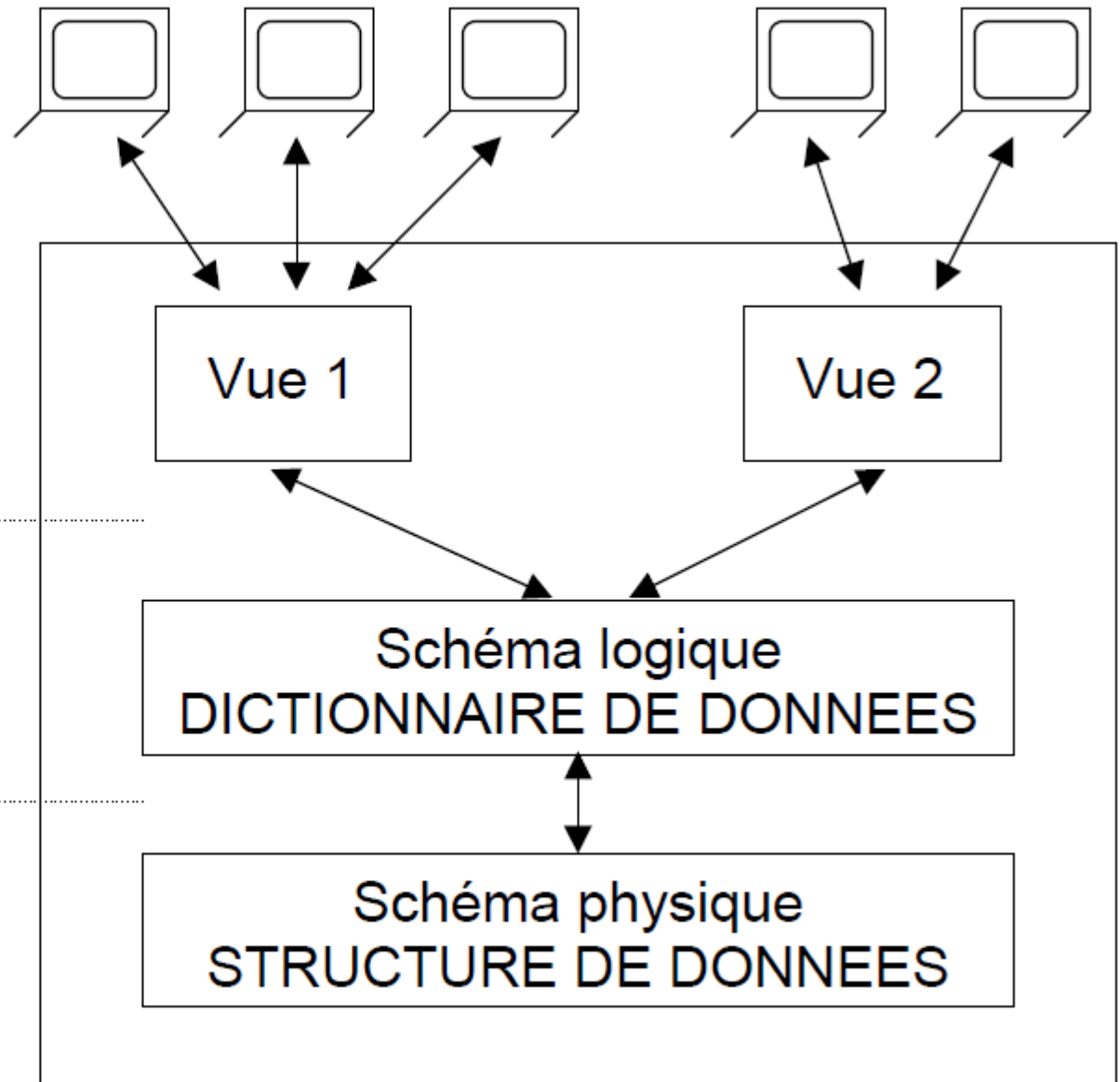
Abstraction des données (3 niveaux)

- **Niveau interne ou physique :**
 - plus bas niveau
 - indique **comment** (avec quelles structures de données) sont stockées physiquement les données
- **Niveau logique ou conceptuel :**
 - décrit par un **schéma conceptuel** *ou logique*
 - indique quelles sont les données stockées et quelles sont leurs relations **indépendamment de l'implantation physique**
- **Niveau externe ou vue :**
 - propre à chaque utilisateur
 - décrit par un ou plusieurs **schémas externes**

EXTERNE

CONCEPTUEL

INTERNE



Petit historique

- **1960** : systèmes de gestion de fichiers
- **1970** : début des SGBD réseaux et hiérarchiques proches des systèmes de gestion de fichiers \Rightarrow pas d'interrogation sans savoir où est l'information recherchée ("navigation") et sans écrire de programmes
- **1970** : papier fondateur de CODD sur la théorie des relations
fondement de la théorie des bases de données relationnelles
INGRES à Berkeley - langage QUEL
System R IBM à San Jose - langages SEQUEL et QBE
- **1980** : **Apparition des SGBD relationnels sur le marché** (Oracle, Ingres, Informix, Sybase, DB2 ...)
- **1990** : **début des SBGD orientés objet** (Gemstone, O₂, Orion, Objectstore, Versant, Matisse...).
- **Aujourd'hui** : **relationnel-objet, semi-structuré, multimédia ...**

BIBLIOGRAPHIE

Ouvrages de référence utilisés pour le cours :

- T. Connolly, C. Begg et A. Strachan, *Database Systems A Practical Approach to Design, Implementation and Management*, 1998, ISBN: 0-201-34287-1,
- G. Gardarin, *Bases de Données - objet/relationnel*, Eyrolles, 1999, ISBN: 2-212-09060-9, disponible à la BU 005.74 GAR
- R. Ramakrishnan et J. Gehrke, *Database Management Systems*, Second Edition; McGraw-Hill, 2000, ISBN: 0-07-232206-3, disponible à la BU 055.7 RAM
- A. Silberschatz, H.F. Korth et S. Sudarshan, *Database System Concepts*, McGraw-Hill, 1996, ISBN: 0-07-114810-8, disponible à la BU 005.7 DAT
- J.D. Ullman et J. Widom, *A first Course in Database Systems*, Prentice Hall, 1997, ISBN: 0-13-887647-9, disponible à la BU 005.7 ULL
- F. Brouad, R. Bruchez, C. Soutou, *SQL*, Syntex Informatique, Pearson, 2012, ISBN: 978-2-7440-7630-5, disponible à la BU 005.72 SQL

BIBLIOGRAPHIE

Autres ouvrages de référence, disponibles à la BU :

- C.J. Date, *An Introduction to Database Systems*, Addison Wesley
- C.J. Date, *A Guide to SQL Standard*, Addison Wesley
- R.A. El Masri et S.B. Navathe, *Fundamentals of Database Systems*, Prentice Hall

Ouvrages pédagogiques contenant des exercices corrigés :

- Philip J. Pratt, *Initiation à SQL - Cours et Exercices corrigés*, Eyrolles, 2001 – BU : 005.72 SQL
- Christian Soutou, *De UML à SQL - Conception de bases de données*, Eyrolles, 2002 – BU : 005.72 SOU
- F. Brouard, C. Soutou , *SQL (Synthèse de cours et exercices corrigés)*. Pearson Education 2005 – BU : 005.72 SQL
- Christian Soutou, *SQL Pour Oracle (avec exercices corrigés)*, Eyrolles, 2005 – BU 005.72 SOU
- Nicolas Larousse, *Création de bases de données*, Coll. Synthex, Pearson Education, 2006

Modélisation

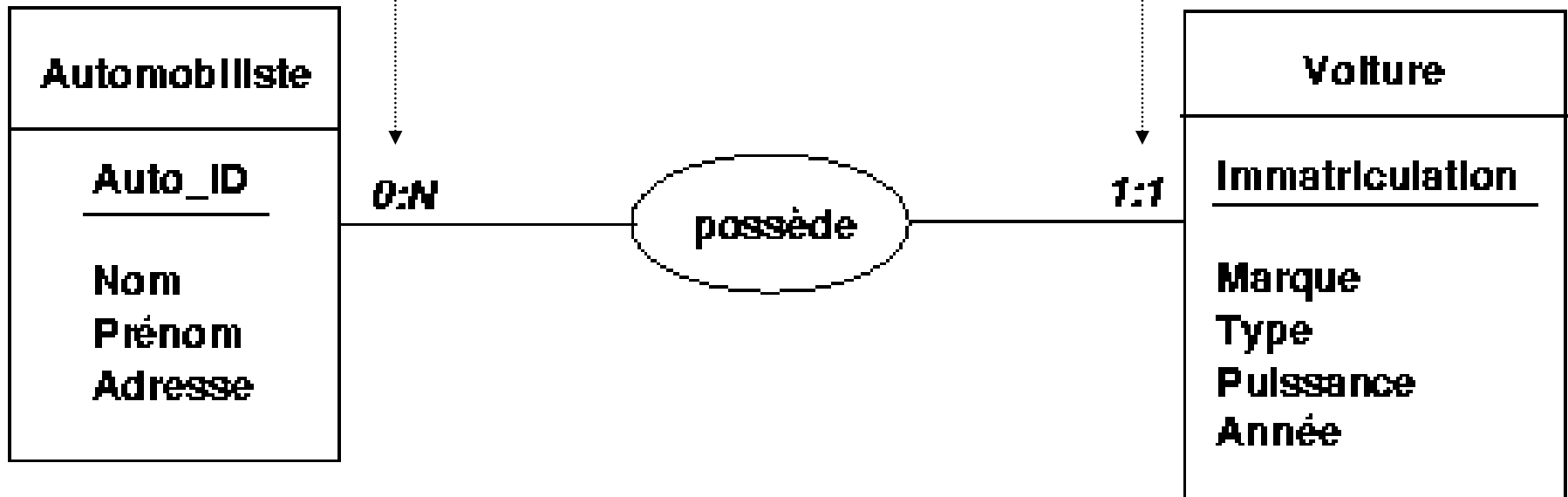
Méthodologie à suivre pour modéliser un problème

- Déterminer les **entités/classes** et **attributs** :
 - entité/instance de classe = objet décrit par de l'information
 - objet caractérisé uniquement par un identifiant = attribut
 - attribut multi-valué ou avec une association 1:N = entité ou instance
 - attacher les attributs aux ensemble d'entités/classes qu'ils décrivent le plus directement
 - éviter au maximum les identificateurs composites
- Identifier les **généralisations-spécialisations/héritage**
- Définir les **associations**
 - éliminer les associations redondantes
 - éviter les associations n-aires
 - calculer les **cardinalités** de chaque association

Modélisation Entité/Association (Format Merise)

*Un automobiliste
possède entre zéro et N
voitures*

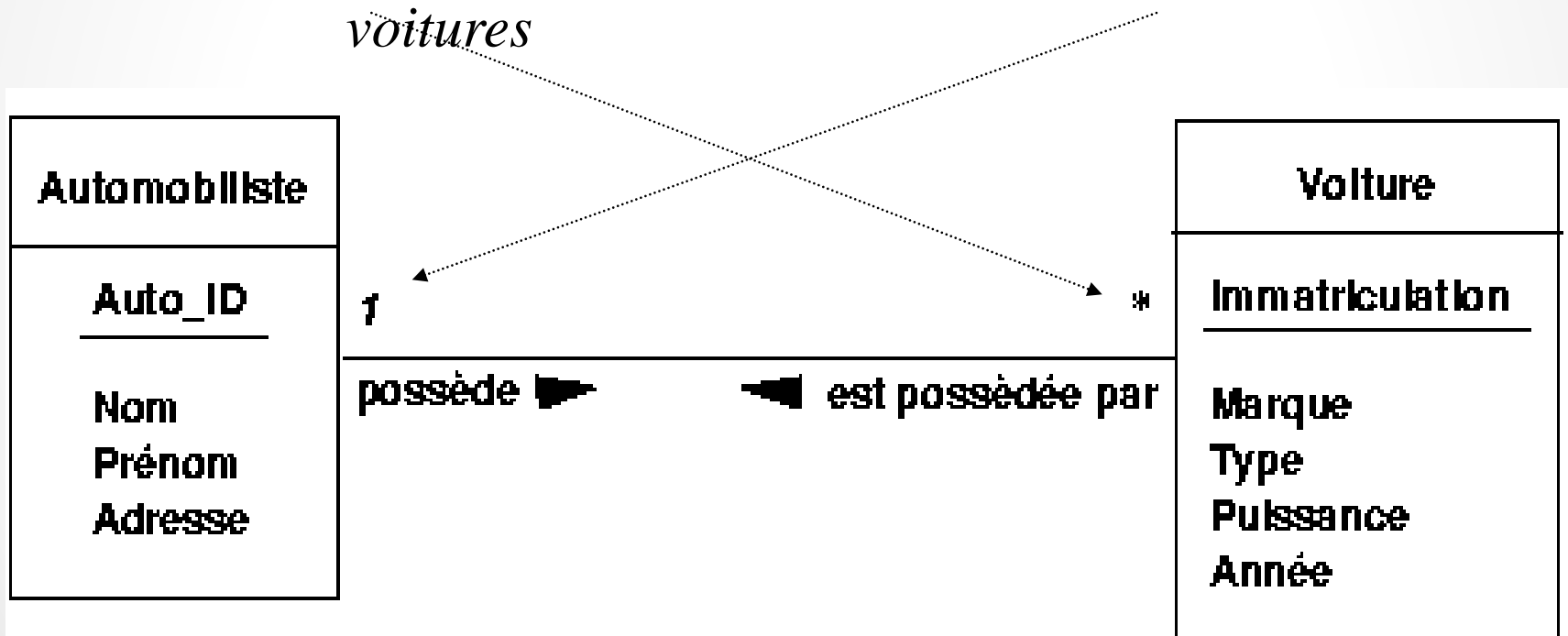
*Une voiture a un et un
seul propriétaire*



Modélisation UML

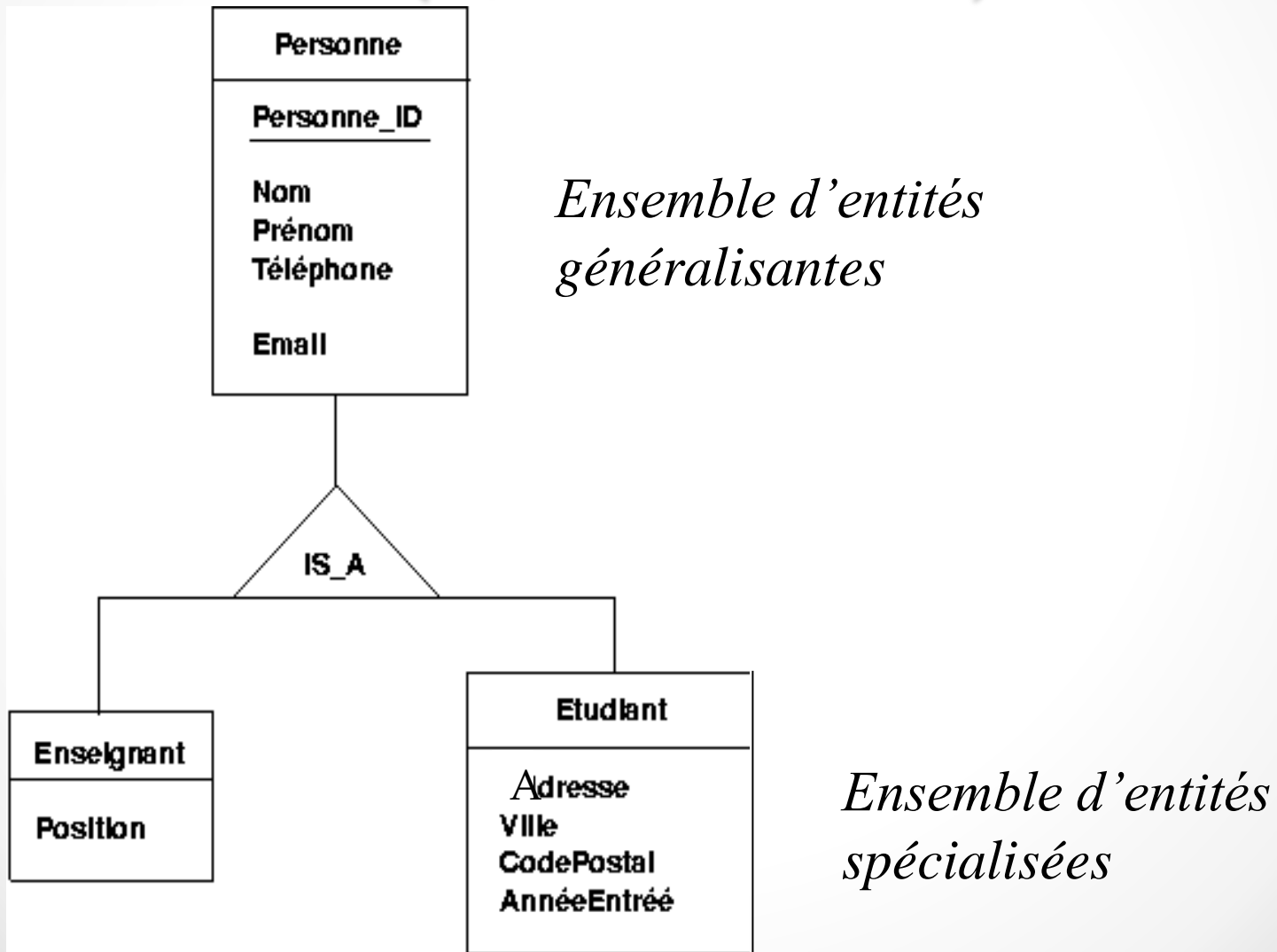
*Un automobiliste
possède entre zéro et N
voitures*

*Une voiture a un et un
seul propriétaire*

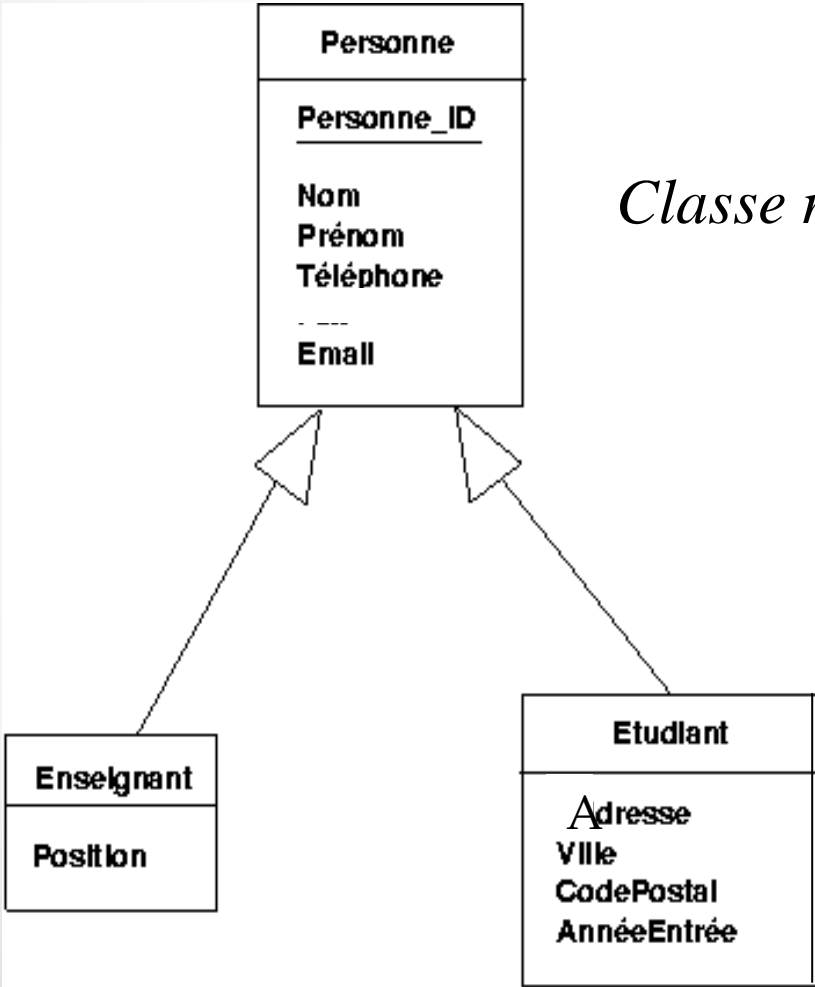


Attention : petite liberté prise avec UML, les attributs soulignés ici ne correspondent pas à des attributs dérivés mais aux identificateurs (pour ne pas oublier lors du passage au relationnel!!)

Généralisation/Spécialisation (E/A - Merise)



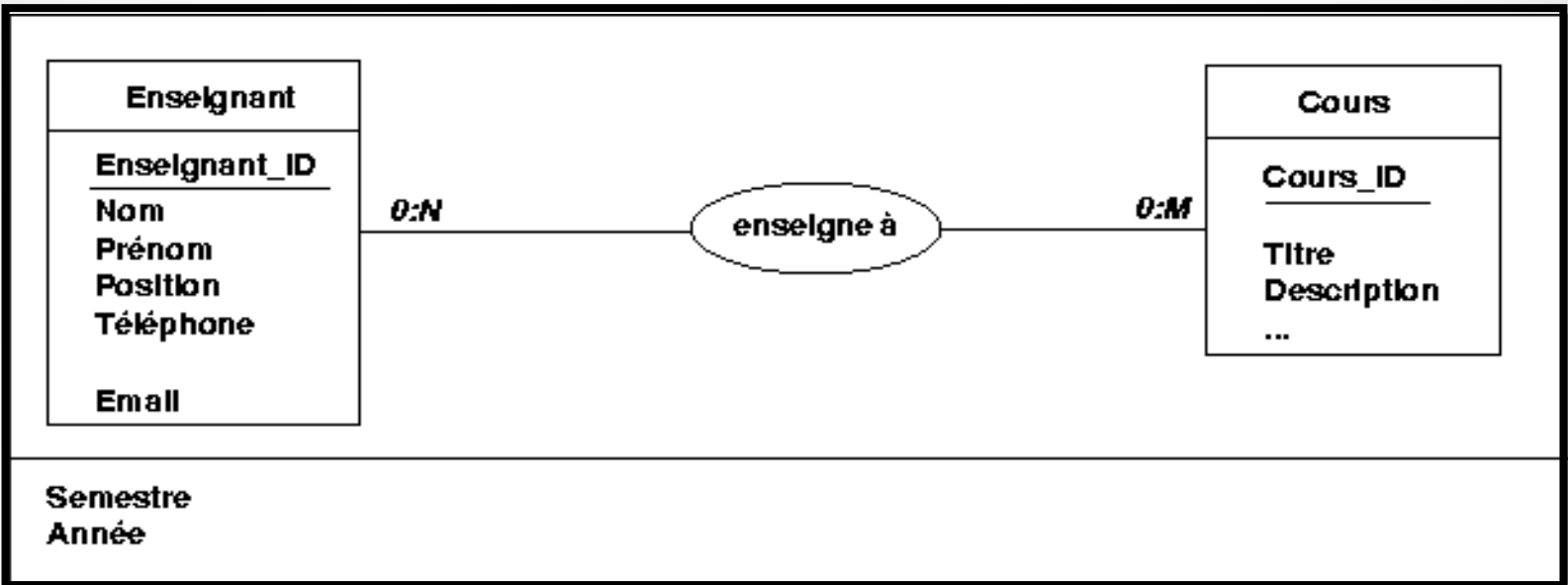
Héritage (UML)



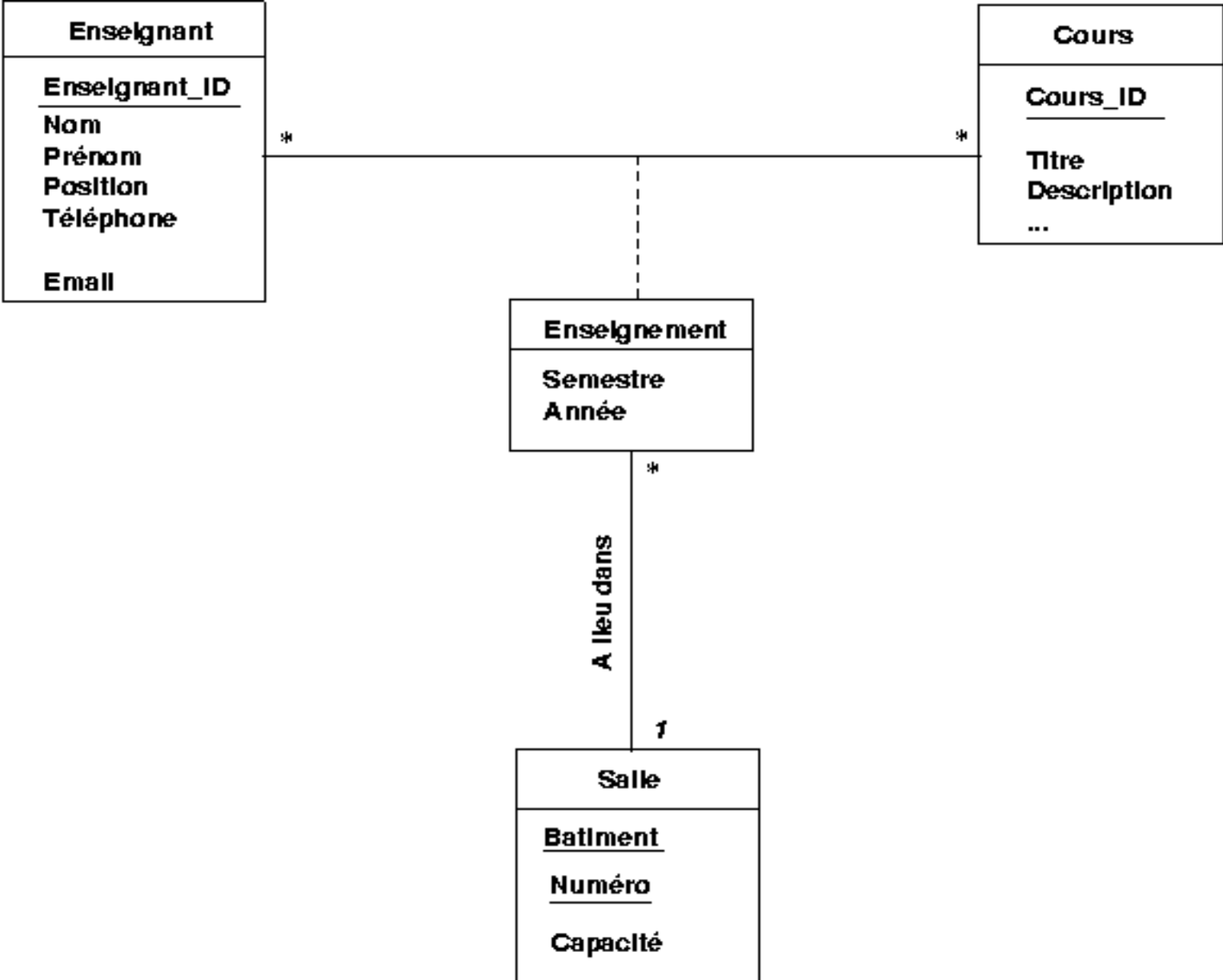
Classe mère / Sur-classe

Classes dérivées ou filles / sous-classes

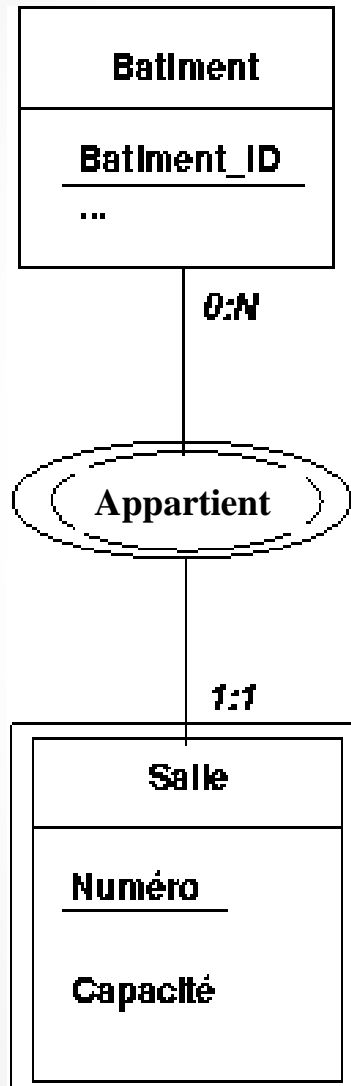
Agrégat (E/A - Merise)



Classe-Association (UML)



Entité Faible (E/A - Merise)

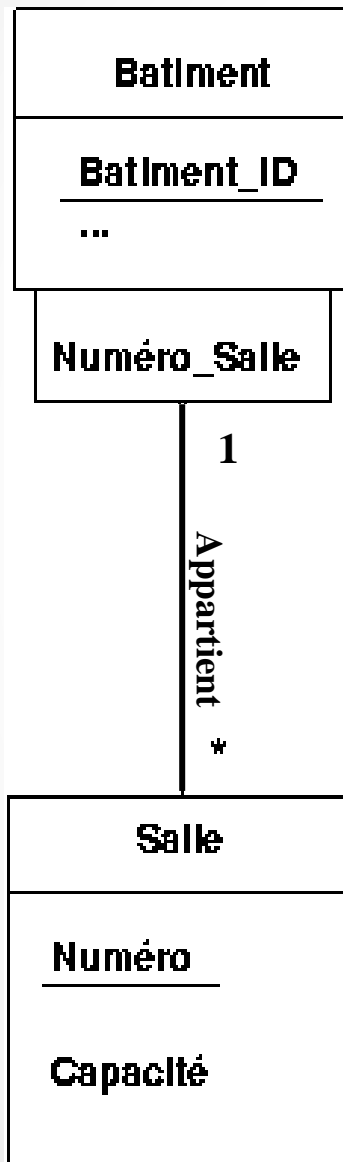


Chaque salle a un numéro unique dans un bâtiment donné

Ex. Salle 1 du bâtiment A et Salle 1 du bâtiment C

Pour distinguer une salle d'une autre, il faut connaître le bâtiment auquel elle est rattachée

Association qualifiée (UML)

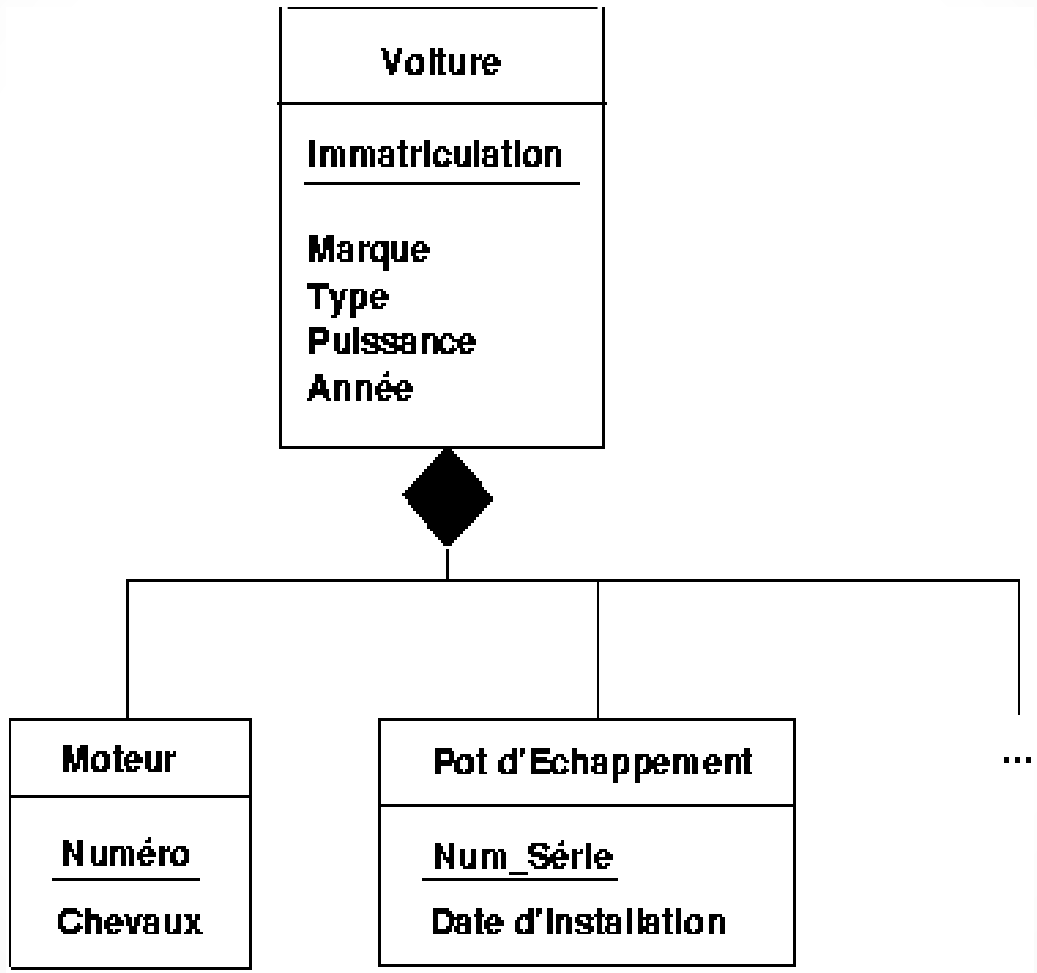


Chaque salle a un numéro unique dans un bâtiment donné

Ex. Salle 1 du bâtiment A et Salle 1 du bâtiment C

Pour distinguer une salle d'une autre, il faut connaître le bâtiment auquel elle est rattachée

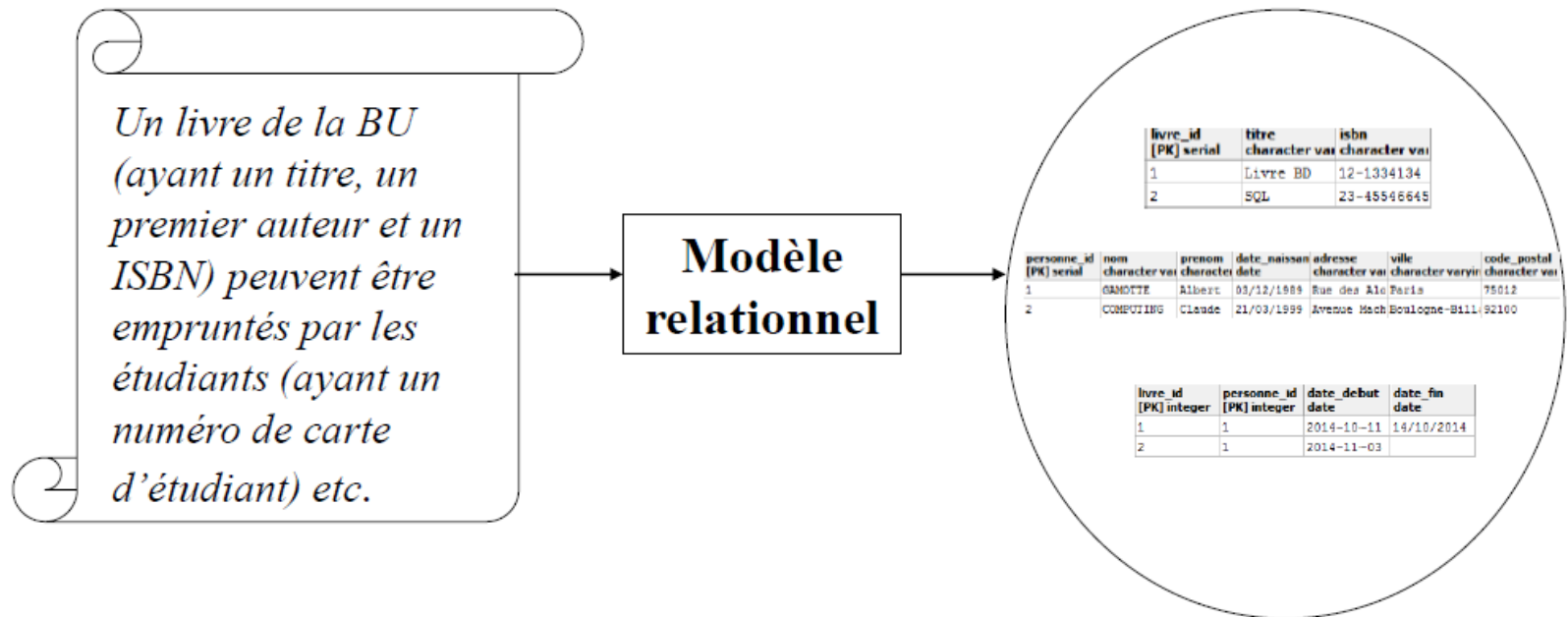
Composition (UML)



Modèle relationnel

Modèle relationnel

- **Données** : Ce que l'on stocke
- **Modèle relationnel** : Modèle permettant d'organiser les données en une représentation schématique qui autorisera son exploitation par le SGBD relationnel



Relations

Relations

- Collection de **nuplets** (*tuples* en anglais) décrivant des données de même structure
- Tableau à deux dimensions composé d'**attributs** (ou champs - en colonnes) et de **nuplets** (ou enregistrements - en ligne)

Noms des 8 attributs

| enseignant_id [PK] serial | departement_id integer | nom character varying(25) | prenom character varyin | grade character vai | telephone character vai | fax character var | email character varying(100) |
|------------------------------|---------------------------|------------------------------|----------------------------|------------------------|----------------------------|----------------------|---------------------------------|
| 1 | 1 | MANOUVRIER | Maude | MCF | | | manouvrier@lamsade.dauphine.fr |
| 2 | 1 | NEGRE | Elsa | MCF | | | negre@lamsade.dauphine.fr |
| 3 | 1 | BELHADJJAME | Khalid | MCF | | | Khalid.Belhajjame@dauphine.fr |

3 nuplets



Dans une relation :

- Pas de doublon
- Pas deux attributs de même nom

Modèle relationnel - vocabulaire

- **Domaine** : ensemble de valeurs caractérisé par un nom
- **Relation** : sous-ensemble du produit cartésien d'une liste de domaines caractérisé par **un nom unique**
 - représentée sous forme de **table à deux dimensions**
 - colonne = un domaine du produit cartésien
 - **un même domaine peut apparaître plusieurs fois**
 - ensemble de **nuplets sans doublon**
- **Attribut** : une colonne dans une relation
 - caractérisé par un nom et dont **les valeurs appartiennent à un domaine**
 - **les valeurs sont atomiques**
- **Nuplet** : une ligne d'une relation
 - correspondant à un enregistrement, c-à-d **une entité/instance de classe**
 - **les nuplets d'une relation sont tous différents**

Instances et schéma

- **Instances de base de données :**
 - données de la base à un instant donné
 - les nuplets (les valeurs) contenus dans la base à un instant donné
 - manipulées par un **langage de manipulation de données (DML - *Data Manipulation Language*)**
- **Schéma de base de données :**
 - description de la structure des données
 - ensemble de **schémas de relation**
 - modélisation logique de la base de données à l'aide du modèle relationnel
 - ensemble de définitions exprimées en **langage de description de données (DDL - *Data Definition Language*)**
- **Schéma de relation :**
 - liste d'attributs et leurs domaines

Clé

Clé

Attribut (ou ensemble d'attributs) permettant d'**identifier de manière unique** les nuplets de la relation

Exemples :

- *L'attribut ISBN pour une relation Livre*
- *L'attribut NuméroImmatriculation pour une relation Voiture*
- *L'attribut NuméroCarte pour une relation Emprunteur*

Par défaut : Création d'un attribut numérique s'incrémentant automatiquement

Clé artificielle
(*surrogate key*)

| departement_id [PK] serial | nom_departement character varying(25) |
|-------------------------------|--|
| 1 | MIDO |
| 2 | LSO |
| 3 | MSO |



Une clé est unique (pas deux fois la même valeur) et a forcément une valeur (pas de valeur *null*)

Intégrité structurelle

- **Unicité des clés**

- **ensemble minimal d'attributs** dont la connaissance des valeurs permet d'identifier un nuplet unique de la relation considérée

- **R a pour clé K si : $\forall t_1, t_2$ nuplets d'une instance de R**

$$t_1.K \neq t_2.K$$

- **Clé minimale** : si K est une clé minimale alors $\nexists K' \subset K$ tel que K' est une clé

- **Clé primaire** : une clé parmi toutes les clés minimales

- **Contraintes de référence**

- **contrainte référentielle** : contrainte d'intégrité portant sur une relation R qui consiste à imposer que la valeur d'un groupe d'attributs apparaissent comme valeur de clé dans une autre relation

- **clé étrangère** : un groupe d'attributs qui doit apparaître comme clé dans une autre relation, i.e. qui fait référence à la clé primaire d'une autre relation.

Clé /Clé minimale /Surclé

Accident (Auto_ID, Nom, Prénom, Adresse, Immatriculation, Marque, Type, Puissance, Année)

Clés primaires possibles : Auto_ID ou Immatriculation

Surclé : (Auto_ID, Immatriculation) + d'autres attributs

Voiture (Immatriculation, Marque, Puissance, Type, Année, #Auto_ID)

Clé primaire : Immatriculation

Surclé : (Immatriculation, Marque, Puissance, Type, Année, Auto_ID)

Location (# Auto_ID, # Immatriculation, Date)

Clé primaire : (Auto_ID, Immatriculation, Date)

NULL

- **Valeur nulle**

- valeur conventionnelle introduite dans une relation pour représenter une information inconnue ou inapplicable
- tout attribut peut prendre une valeur nulle **excepté les attributs de la clé primaire** (contrainte d'entité)
- Toute clé étrangère peut prendre une valeur nulle

Contraintes

Contraintes d'intégrité :

toutes règles implicites ou explicites que doivent suivre les données [Gar99]

- **Contraintes d'entité**: toute entité doit posséder un identificateur
- **Contraintes de domaine** : les valeurs de certains attributs doivent être prises dans un ensemble donné
- **Contraintes d'unicité** : une valeur d'attribut ne peut pas être affectée deux fois à deux entités différentes
- **Contraintes générales** : règle permettant de conserver la cohérence de la base de manière générale

Exemples de contraintes

- **Contraintes de domaine :**

"La fonction d'un enseignant à l'Université prend sa valeur dans l'ensemble {vacataire, moniteur, ATER, MCF, Prof., PRAG, PAST}."

- **Contraintes d'unicité :**

"Un département, identifié par son numéro, a un nom unique (il n'y a pas deux départements de même nom)."

- **Contraintes générales :**

"Un même examen ne peut pas avoir lieu dans deux salles différentes à la même date et à la même heure.«

"La date de début d'emprunt doit être antérieure à la date de fin d'emprunt."

Passage au modèle relationnel

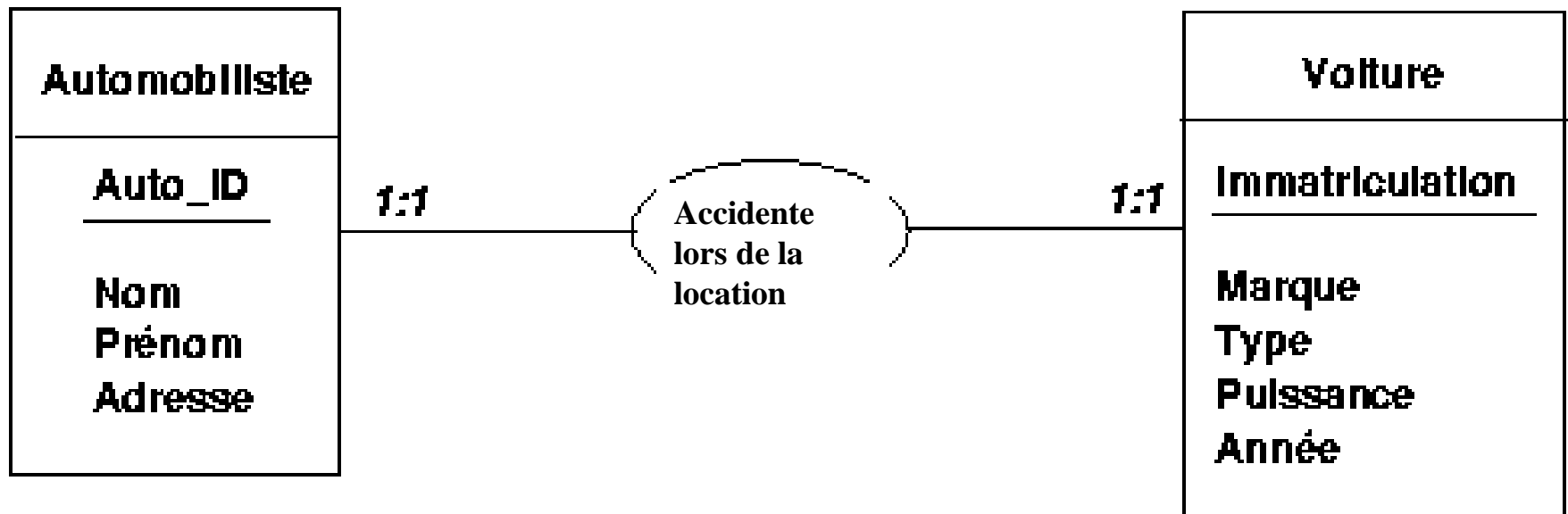
(à partir d'un modèle E/A)

Passage au relationnel

Transformation des ensembles d'entités :

- **chaque ensemble d'entités/classes $E \Rightarrow$**
 - une relation R dont le schéma est celui de l'ensemble d'entités/classe
 - l'identificateur de E devient la clé de R
- **chaque ensemble d'entités faibles/association qualifiée $E \Rightarrow$**
une relation R qui comprend tous les attributs de E +
l'identificateur de l'ensemble d'entités fortes/classe associé(e)
- **généralisation-spécialisation/héritage \Rightarrow**
 - l'ensemble d'entités généralisante/classe mère $E \Rightarrow$ une relation R
 - chaque ensemble d'entités E_i spécialisé/classe fille
 \Rightarrow une relation R_i dans laquelle l'identifiant est de même domaine que
l'identifiant de E

Transformation des ensembles d'associations E/A

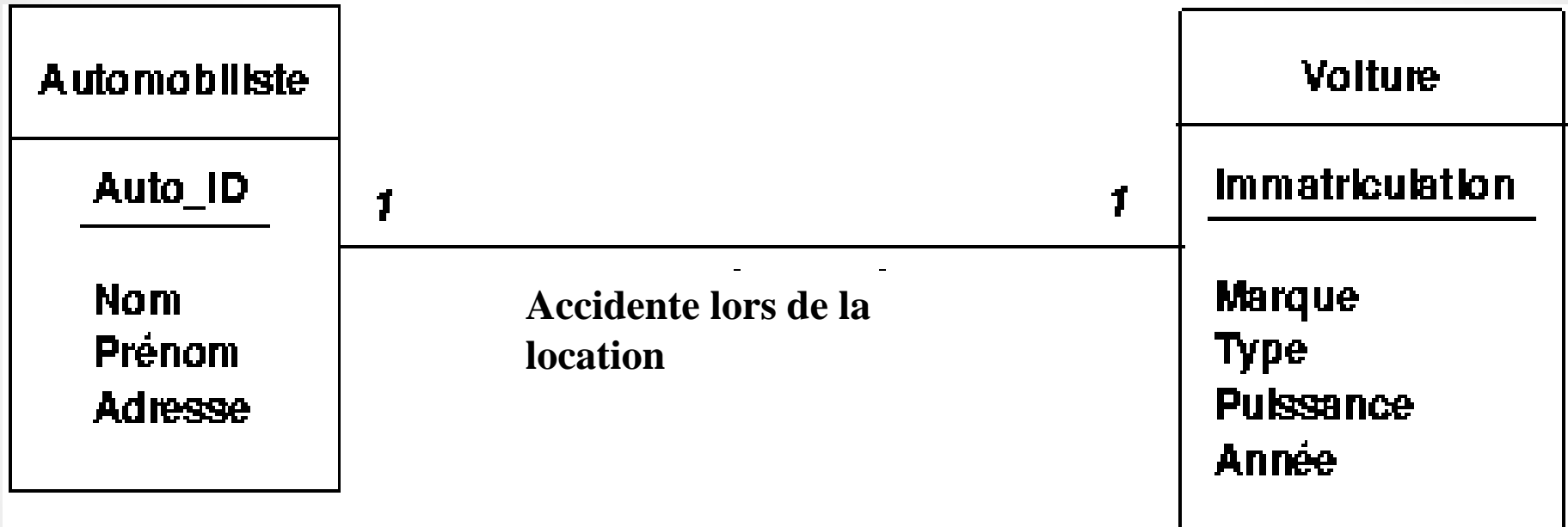


Automobiliste (Auto_ID, Nom, Prénom, Adresse)

Voiture (Immatriculation, Marque, Type, Puissance, Année)

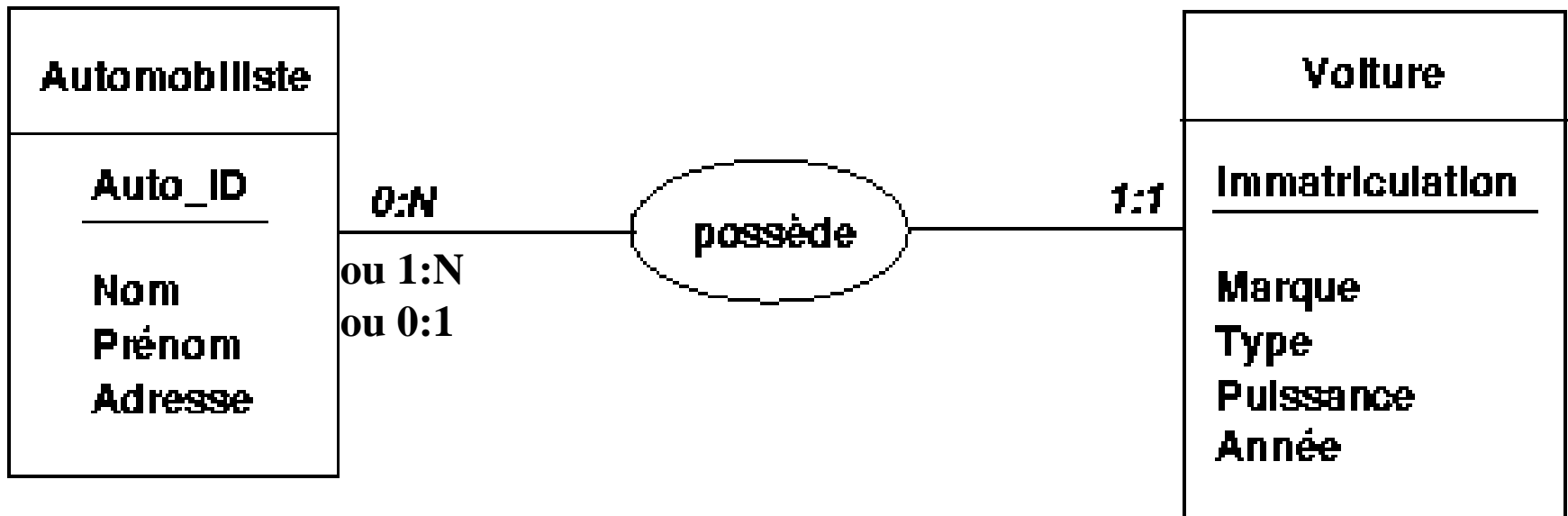
Comment faire le lien ?

Transformation des ensembles d'associations UML



Accidente (Auto_ID, Nom, Prénom, Adresse, Immatriculation, Marque, Type, Puissance, Année)

Transformation des ensembles d'associations E/A

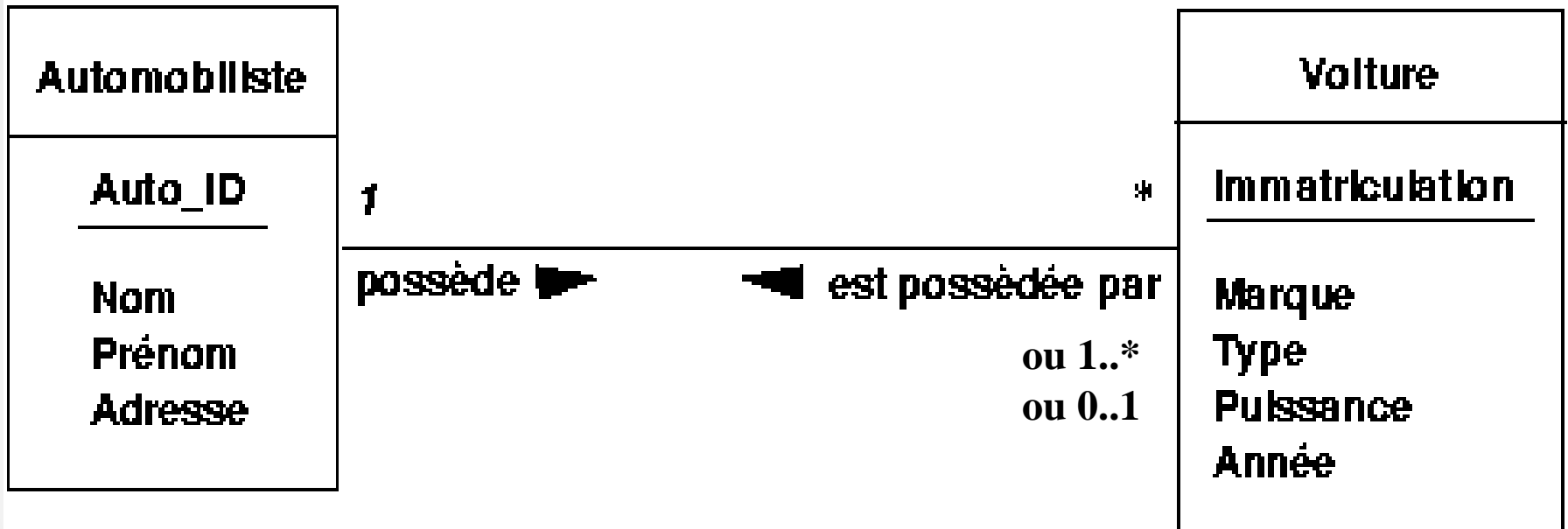


Automobiliste (Auto_ID, Nom, Prénom, Adresse)

Voiture (Immatriculation, Marque, Puissance, Type, Année,
#Auto_ID)

NB : #Auto_ID fait référence à Auto_ID de Automobiliste

Transformation des ensembles d'associations UML

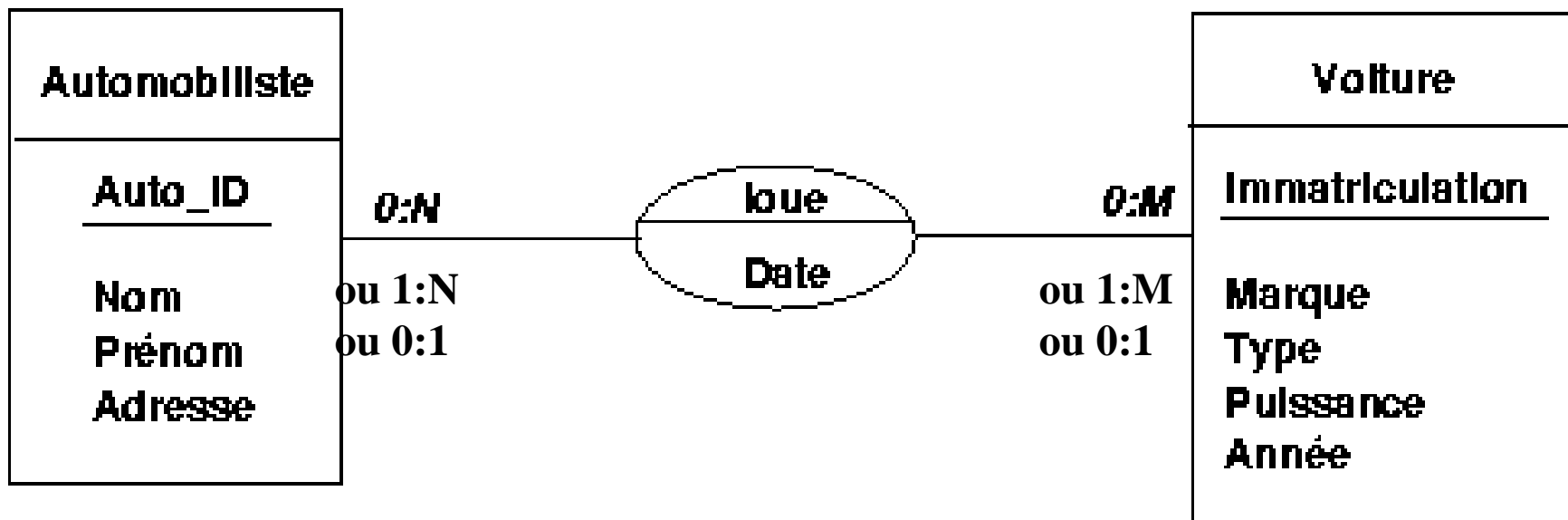


Automobiliste (Auto_ID, Nom, Prénom, Adresse)

Voiture (Immatriculation, Marque, Puissance, Type, Année,
 #Auto_ID)

NB : #Auto_ID fait référence à Auto_ID de Automobiliste

Transformation des ensembles d'associations E/A



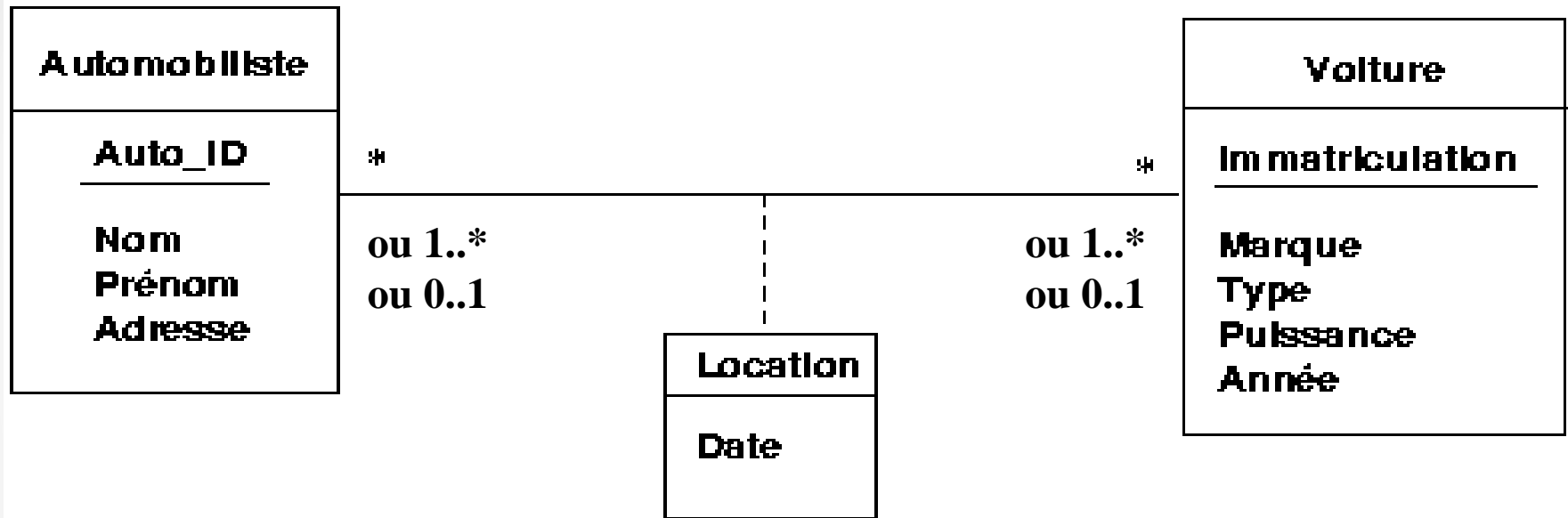
Automobiliste (Auto_ID, Nom, Prénom, Adresse)

Voiture (Immatriculation, Marque, Puissance, Type, Année)

Location (#Auto_ID, #Immatriculation, Date) ou

Location (Loc_ID, #Auto_ID, #Immatriculation, Date)

Transformation des ensembles d'associations UML



Automobiliste (Auto_ID, Nom, Prénom, Adresse)

Voiture (Immatriculation, Marque, Puissance, Type, Année)

Location (#Auto_ID, #Immatriculation, Date) ou

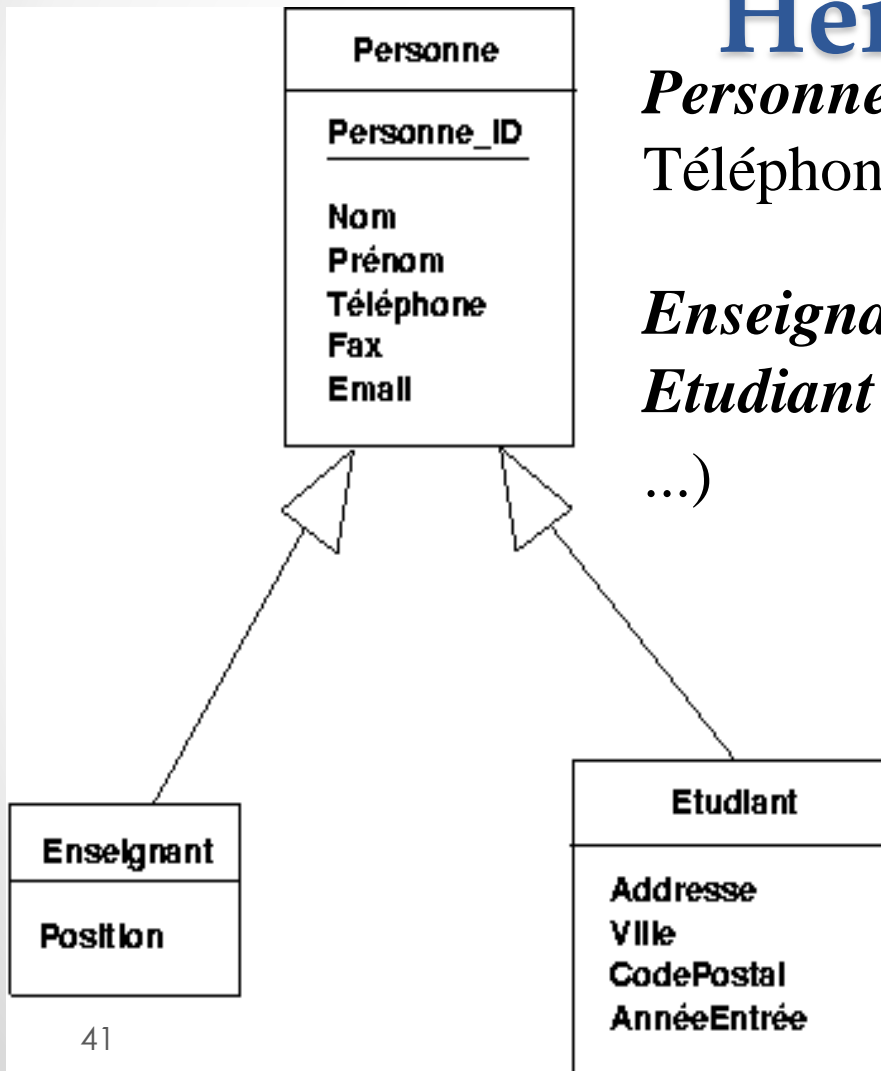
Location (Loc_ID, #Auto_ID, #Immatriculation, Date)

Transformation des concepts Généralisation-Spécialisation / Héritage

Personne (Personne_ID, Nom, Prénom, Téléphone ...)

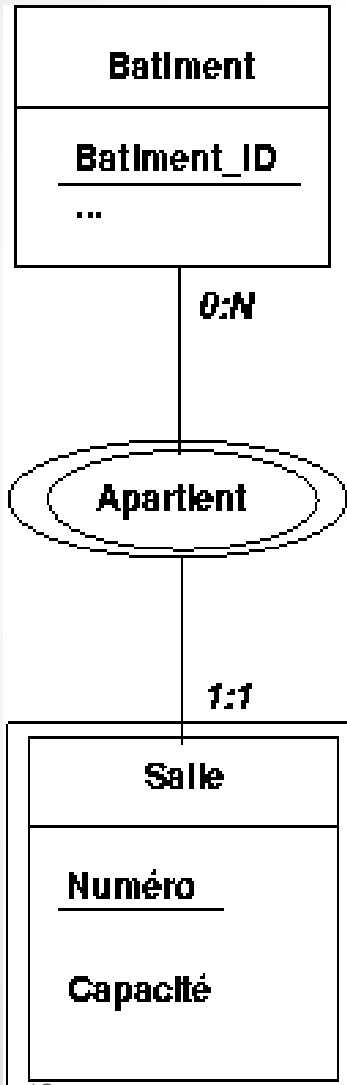
Enseignant (#Personne_ID, Position)

Etudiant (#Personne_ID, Adresse, Ville ...)



NB : #*Personne_ID* dans *Enseignant* et *Etudiant* font référence à *Personne_ID* dans *Personne*

Transformation des entités faibles E/A



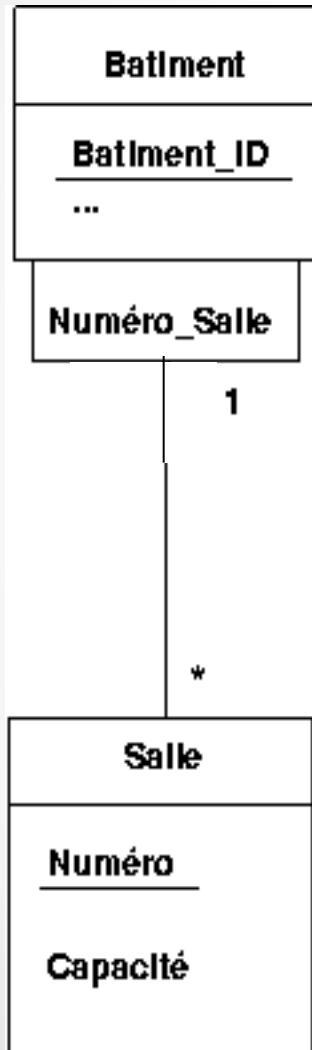
Bâtiment (Bâtiment_ID, ...)

Salle (Numéro, #Bâtiment_ID, Capacité)

NB : Une salle est identifiée par le couple
(Numéro, #Bâtiment_ID)

#Bâtiment_ID fait référence à
Bâtiment_ID de Bâtiment

Transformation des associations qualifiées UML



Bâtiment (Bâtiment_ID, ...)

Salle (Numéro, #Bâtiment_ID, Capacité)

NB : Une salle est identifiée par le couple
(Numéro, #Bâtiment_ID) ;

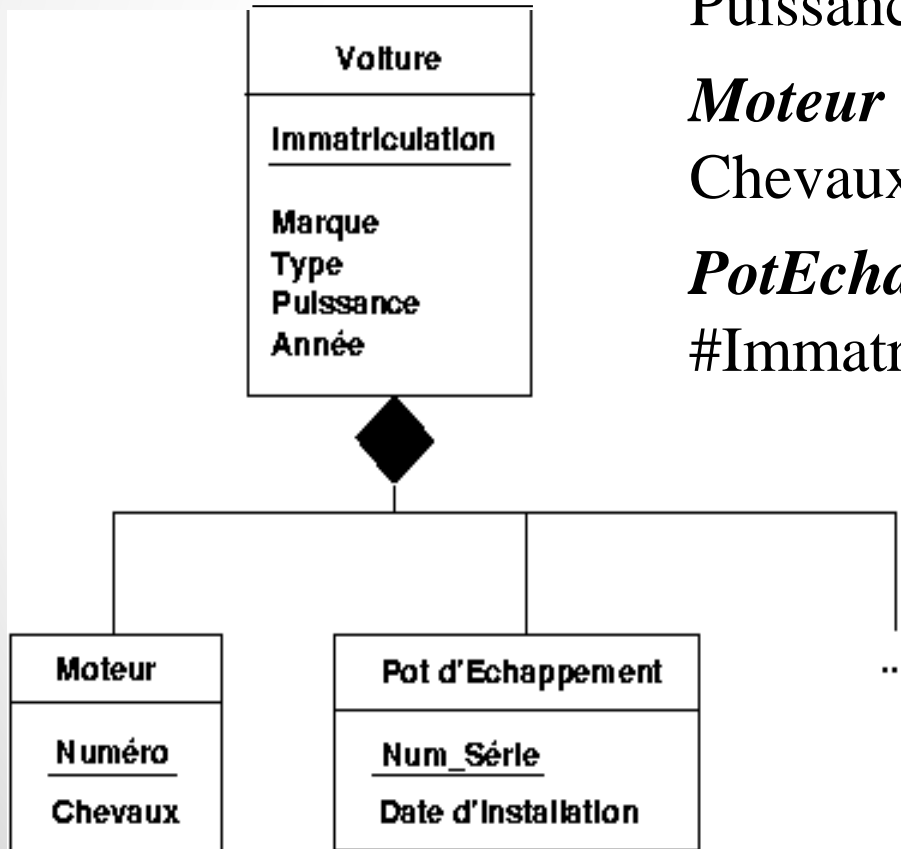
#Bâtiment_ID fait référence à
Bâtiment_ID de Bâtiment

Transformation de la composition UML

Voiture (Immatriculation, Marque, Puissance, Type, Année)

Moteur (Numéro, #Immatriculation, Chevaux)

PotEchappement (Num_Série, #Immatriculation, DateInstallation)



Création d'une base de données (1/2)

Étape N° 1 : Concevoir la base de données

= Réfléchir à ce que va contenir la base de données et comment structurer les données

= Modélisation de la base de données

⇒ **Modèle conceptuel de données**

(Modèle Entité/Association ou UML - hors programme)

Démarche :

- Établir la liste des données devant être stockées dans la base
- Définir la structure des données

Création d'une base de données (2/2)

Étape N° 2 : Définir le modèle relationnel

= le **schéma** des relations de la base de données

Démarche :

- Pour chaque relation :
 - Définir les différents attributs
 - **Définir la clé primaire**
- Pour chaque attribut de chaque relation
 - **Définir le type et le domaine**
 - Préciser les propriétés (taille, format, etc.)
- Quand il y a plusieurs relations : **définir les clés étrangères**

Quelques règles

- Bien réfléchir aux schémas des relations et vérifier qu'ils sont corrects avant d'y insérer des données
- Utiliser des noms de relations et d'attributs compréhensibles (penser aux utilisateurs!!)
- Choisir le type de données adéquate pour chaque attribut
- Ne pas créer d'attribut de trop grande taille
- Ne pas créer d'attribut ayant des valeurs trop variables (ex. Age)
- Spécifier toutes les contraintes de domaines (en particulier quand l'ensemble de valeurs est limité), d'unicité etc.
- Préférer les clés primaires de type entier et en particulier des clés artificielles