

Bases de données relationnelles

Chapitre 01 : Introduction

1- Base de données:

- collection d'informations ou de données qui existent sur une **longue période de temps** et qui décrivent les activités d'une ou plusieurs organisations.
- Ensemble de données **modélisant les objets d'une partie du monde réel** et servant de support à une application informatique.

2- SGBD:

Systèmes de Gestion de Bases de Données (*Data Base 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.**

a- Principaux composants d'un SGBD :

- Système de gestion de fichiers
- Gestionnaire de requêtes
- Gestionnaire de transactions

b- Principales fonctionnalités :

- Contrôle de la redondance d'information
- Partage des données
- Gestion des autorisations d'accès
- Vérifications des contraintes d'intégrité
- Sécurité et reprise sur panne

3- Abstraction des données

- **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**

- 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**

4- Instances et schéma

- **Instances de base de données:**

- données de 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 définitions exprimées en **langage de description de données (DDL-Data Definition Language)**

Petit historique

- **1960**: systèmes de gestion de fichiers
- **1970** : début des SGBD réseaux et hiérarchiques proches des systèmes de gestion de fichiers 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 SGBD orientés objet** (Gemstone, O2, Orion, Objectstore, Versant, Matisse...).
- **Aujourd'hui**: relationnel-objet, semi-structuré, multimédia ...

Chapitre 02 : Modélisation

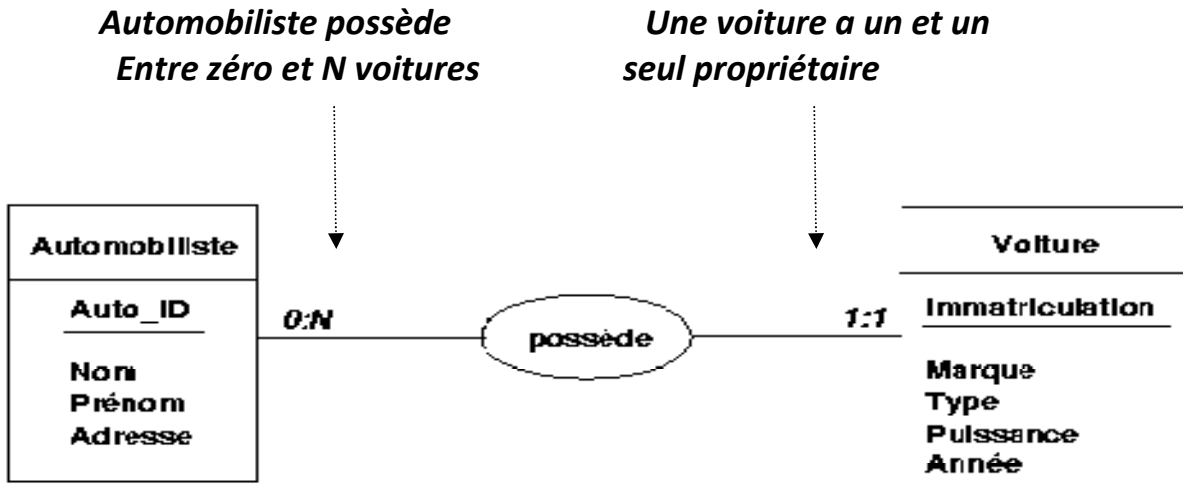
1- Méthodologie à suivre pour modéliser un problème

- 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 ensembles 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

2- Modélisation E/A Vs UML

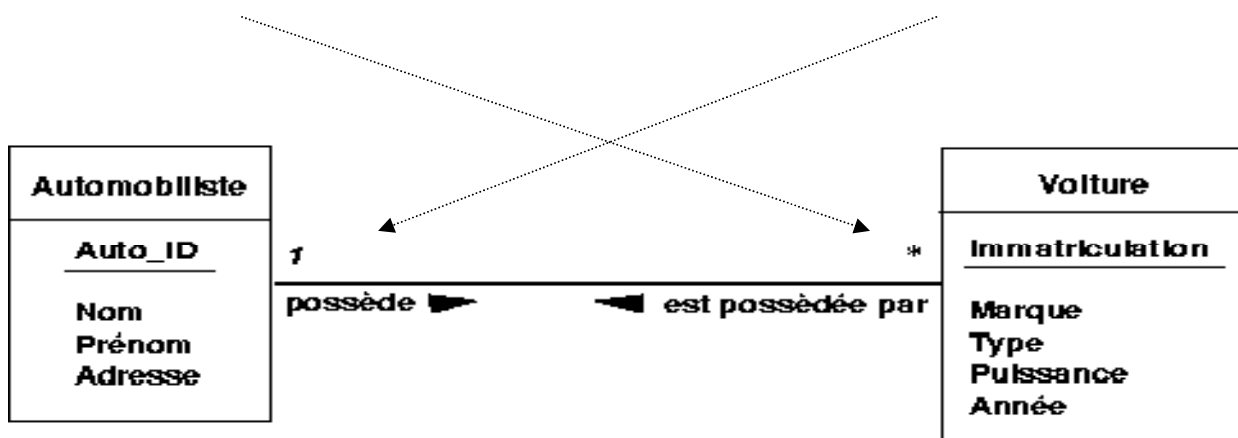
Modélisation Entité/Association (Format Merise)



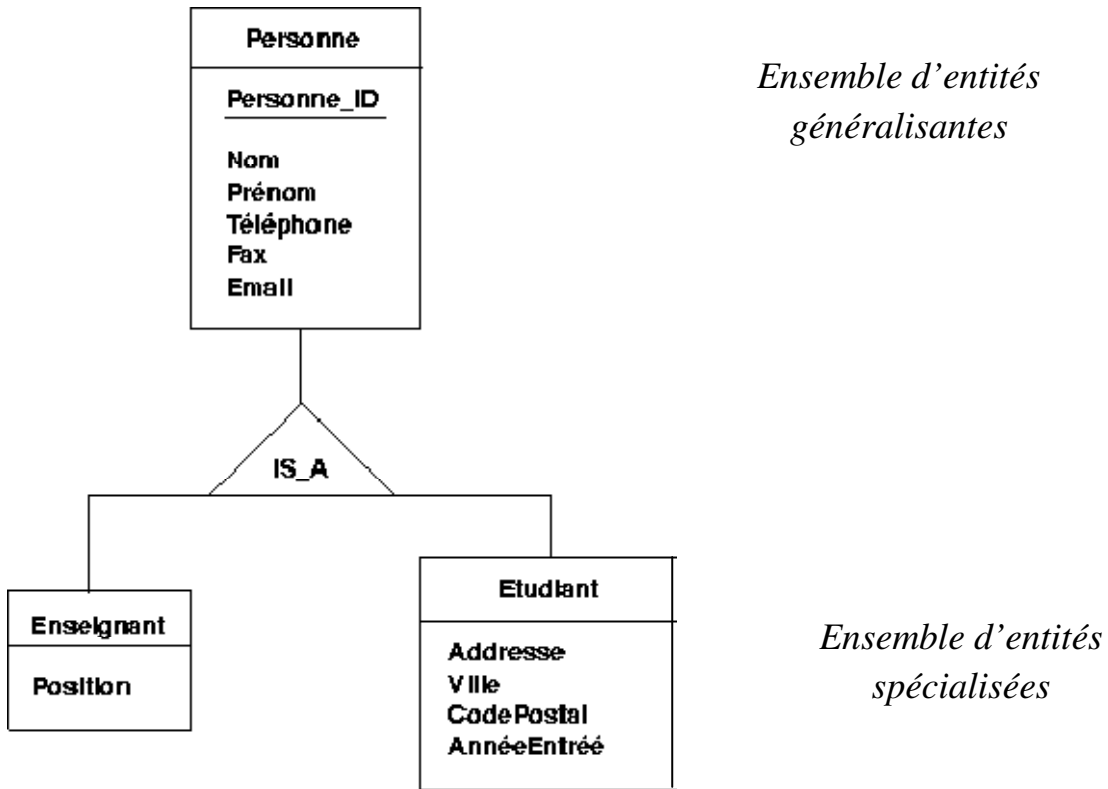
Modélisation UML

*Un automobiliste possède
entre zéro et N voitures*

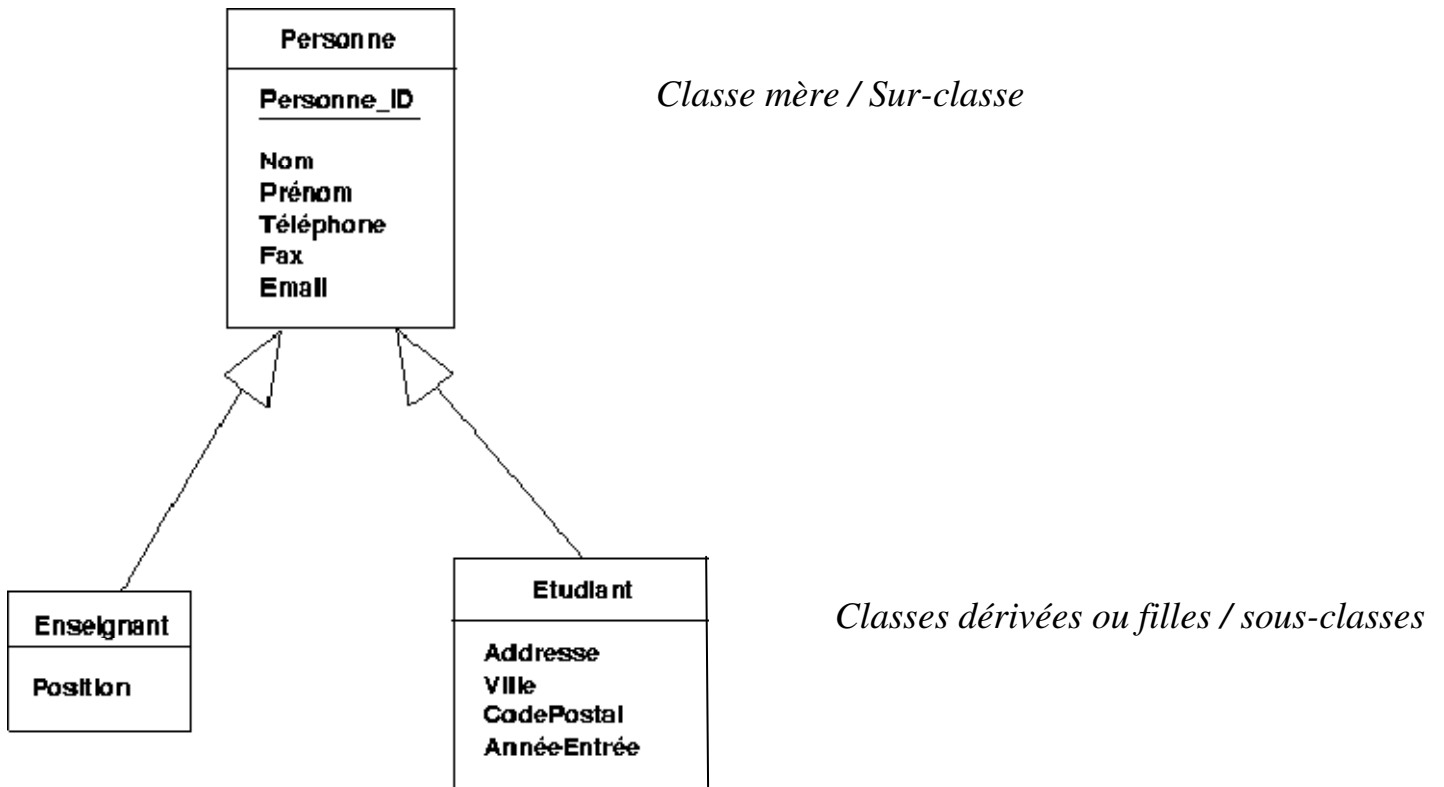
*Une voiture a un et
un seul propriétaire*



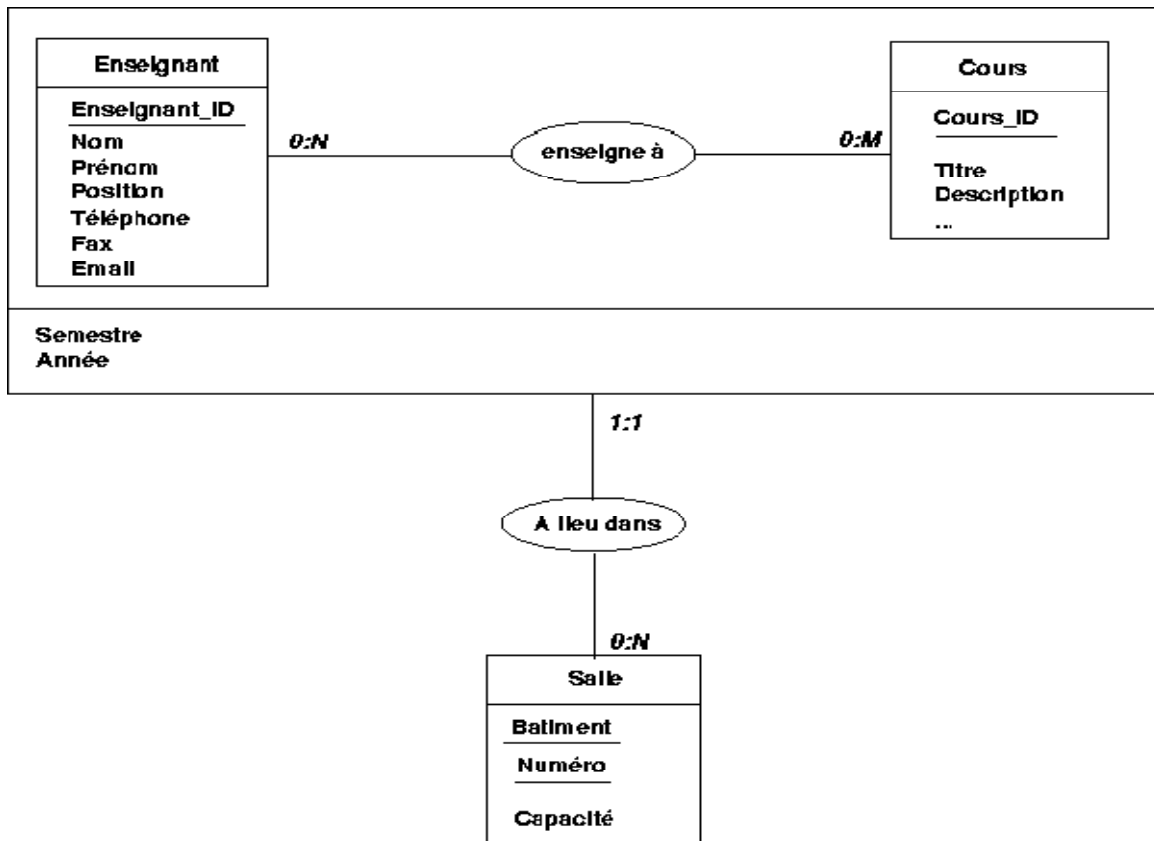
Généralisation/Spécialisation (E/A -Merise)



Héritage (UML)

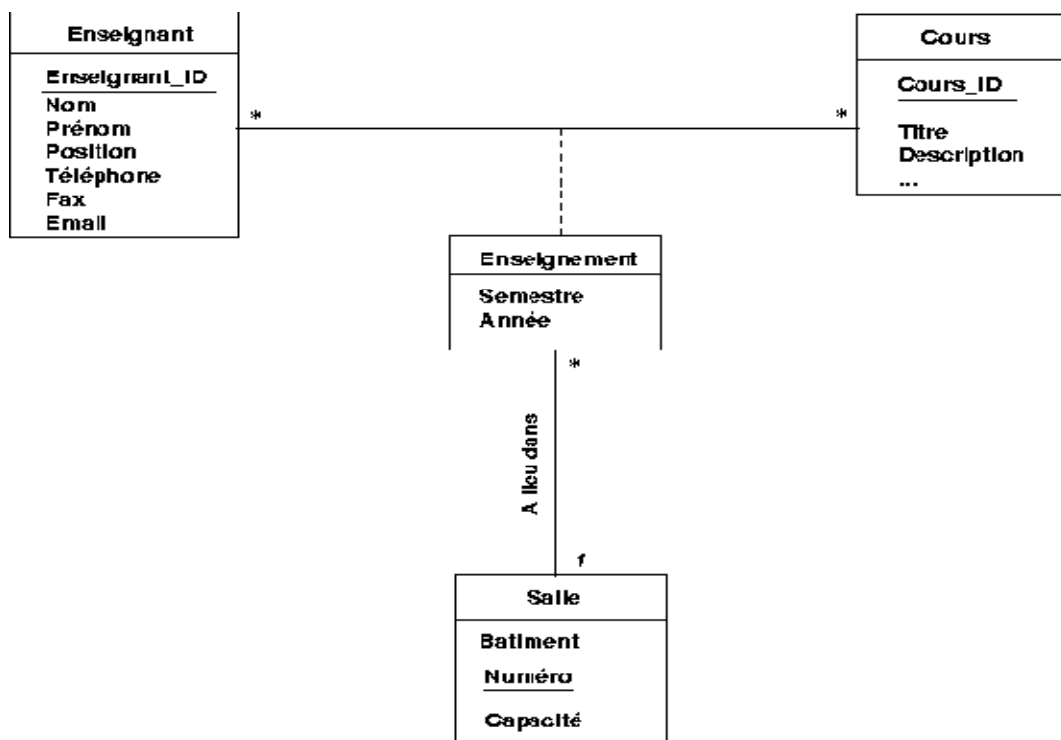


Agrégat (E/A -Merise)



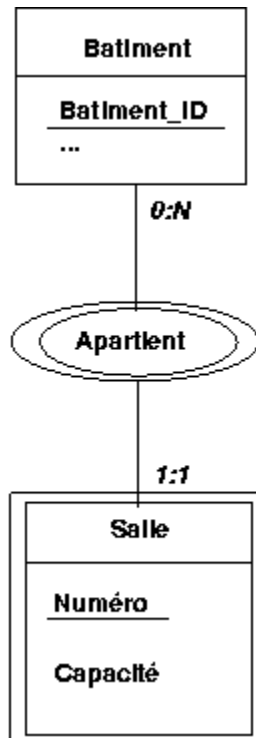
On peut nommer ou non l'agrégat

Classe-Association (UML)



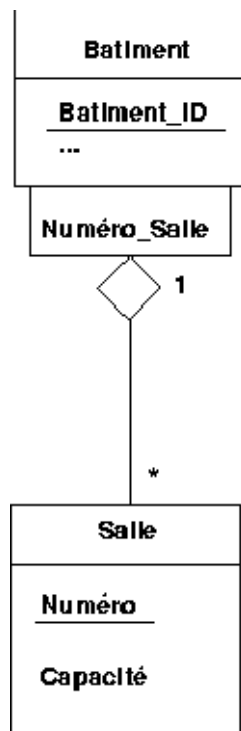
On peut nommer ou non la classe-association

Entité Faible (E/A -Merise)



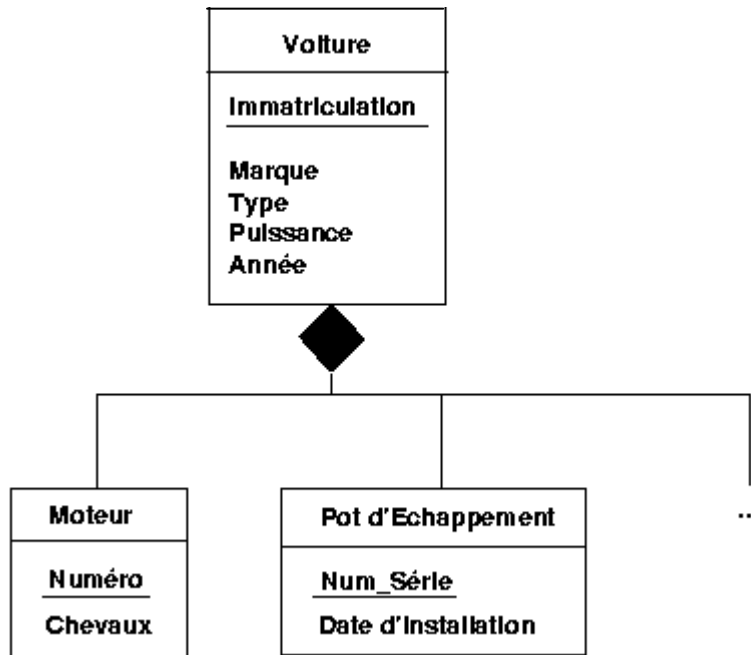
*Chaque salle a un numéro unique dans un bâtiment donné
Ex. Salle 1 du bâtiment A et Salle 1 du bâtiment C
Pour distinguer une salle d'une autre, il faut connaître le bâtiment auquel elle est rattachée*

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)



3- Contraintes

Contraintes d'intégrité:

Toutes règles implicites ou explicites que doivent suivre les données

–**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, Prof..... }."

–**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."

4- Dépendances fonctionnelles

Un attribut (ou un groupe d'attributs) Y **dépend fonctionnellement** d'un attribut(ou groupe d'attributs) X si:

étant donné une valeur de X , il lui correspond une valeur unique de Y (\forall l'instant considéré)

$X \longrightarrow Y$: Y **dépend fonctionnellement de X** ou X **détermine Y**

Déclaration des dépendances **au niveau du schéma conceptuel**

Exemple de dépendances fonctionnelles

Voiture
<u>Immatriculation</u>
Marque
Type
Puissance
Année

identificateur

Tous les autres attributs

Immatriculation → Marque, Type, Puissance, Année

~~Marque, Type, Puissance, Année → Immatriculation~~

Type → Marque *Ex. Le type "Twingo" sera toujours associé, dans la base de données, à la marque "Renault".*

Enseignant
<u>Enseignant_ID</u>
Nom
Prénom
Position
Téléphone
Fax
Email

EnseignantID → Nom, Prénom, Position ...

Nom, Prénom, Position, ... → Enseignant_ID

Si un numéro de téléphone est associé à un seul enseignant :

Telephone → Enseignant_ID

Chapitre 03 : Modèle relationnel

1-

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

Exemple de relation

Nom d'attribut



NSS	Nom	Prénom	Fonction
273 ...	Manouvrier	Maude	MCF
...			
...			

La relation Enseignant

Nuplets ou tuples

2- Instances et schéma

• **Instances de base de données:** les nuplets (les valeurs) contenus dans la base à un instant donné.

• **Schéma de base de données:**

– ensemble de **schémas de relation**.

– modélisation logique de la base de données à l'aide du modèle relationnel.

• **Schéma de relation:** liste d'attributs et leurs domaines.

3- Passage au relationnel

Transformation des ensembles d'entités:

• **chaque ensemble d'entités/classes E** \implies

– 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** \implies

Une relation R qui comprend tous les attributs de E + l'identificateur de l'ensemble d'entités fortes/classe associé(e)

• **généralisation-spécialisation/héritage** \implies

– l'ensemble d'entités généralisante/classe mère E une relation $R \implies$

– chaque ensemble d'entités E_i spécialisé / classe fille

\implies Une relation R_i dans la quelle l'identifiant est de même domaine que l'identifiant de E

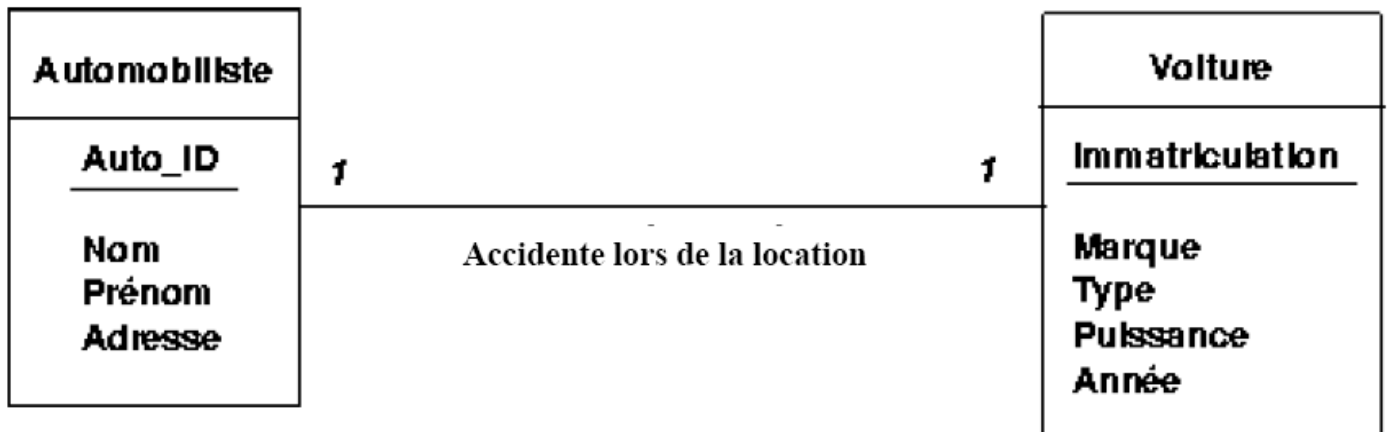
Transformation des ensembles d'associations E/A



Automobiliste (Auto_ID, Nom, Prénom, Adresse)

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

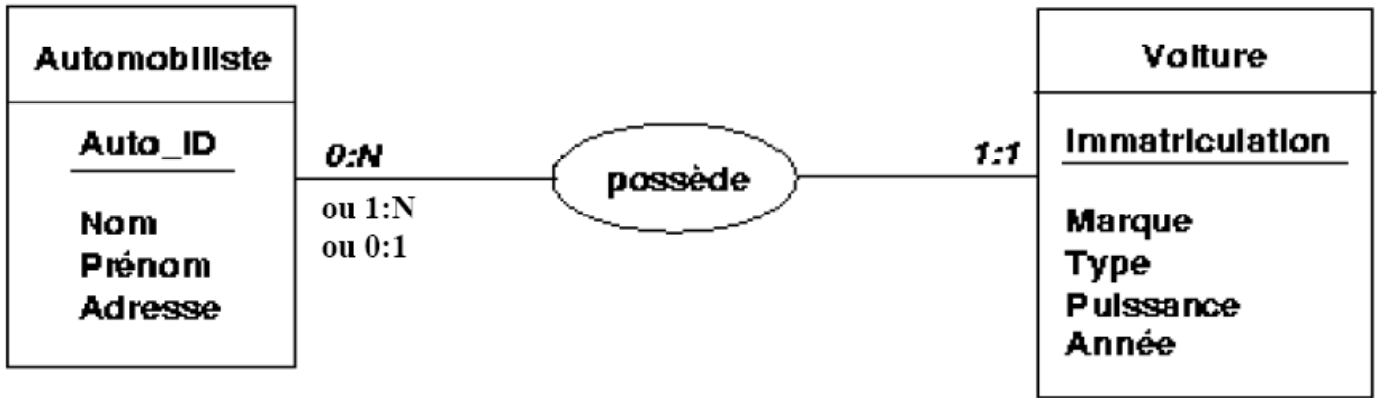
Transformation des ensembles d'associations UML



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

On peut choisir l'un ou l'autre comme clé primaire

Transformation des ensembles d'associations E/A

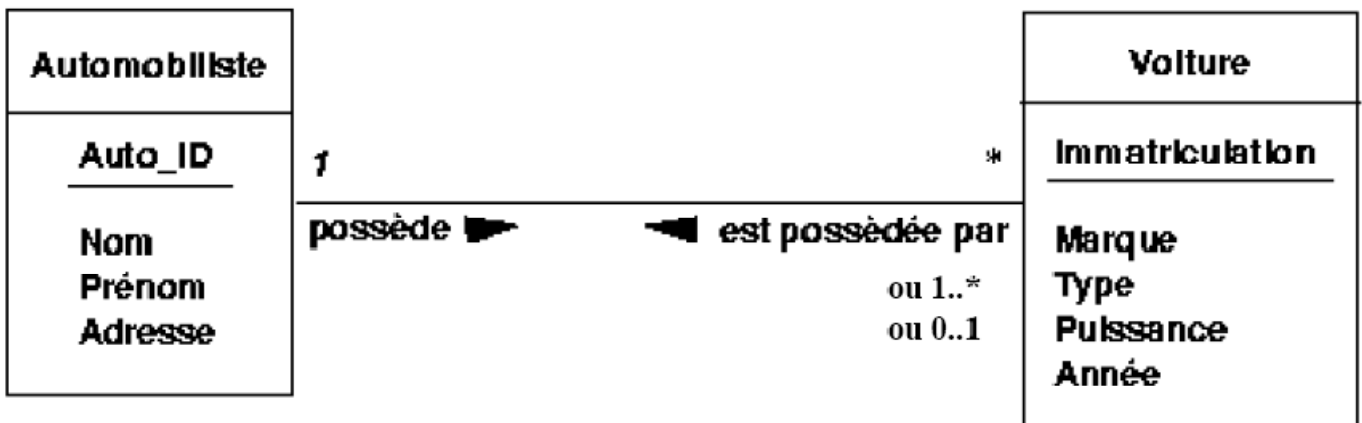


Automobiliste (Auto_ID, Nom, Prénom, Adresse)

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

NB : #Auto_ID fait référence à Auto_ID de Automobiliste

Transformation des ensembles d'associations UML

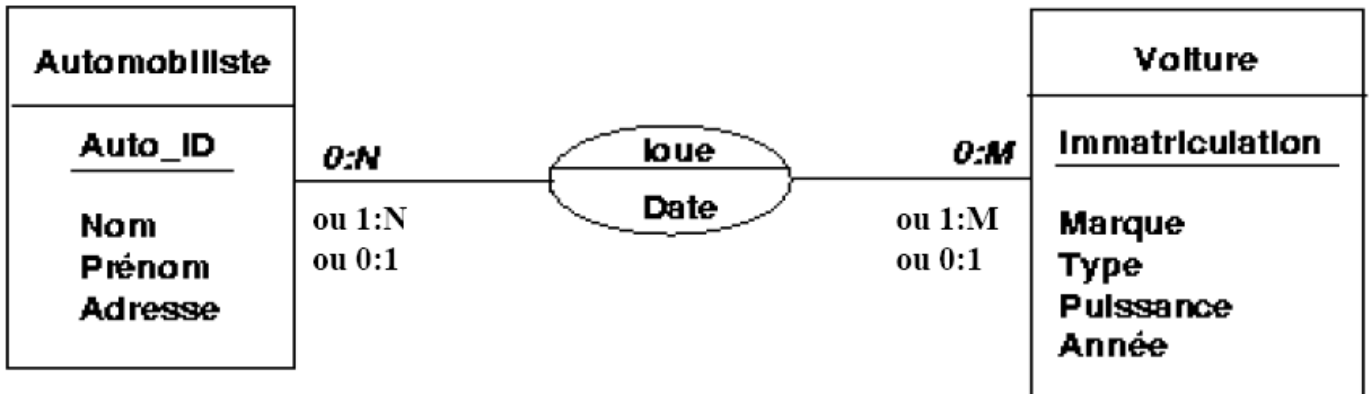


Automobiliste (Auto_ID, Nom, Prénom, Adresse)

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



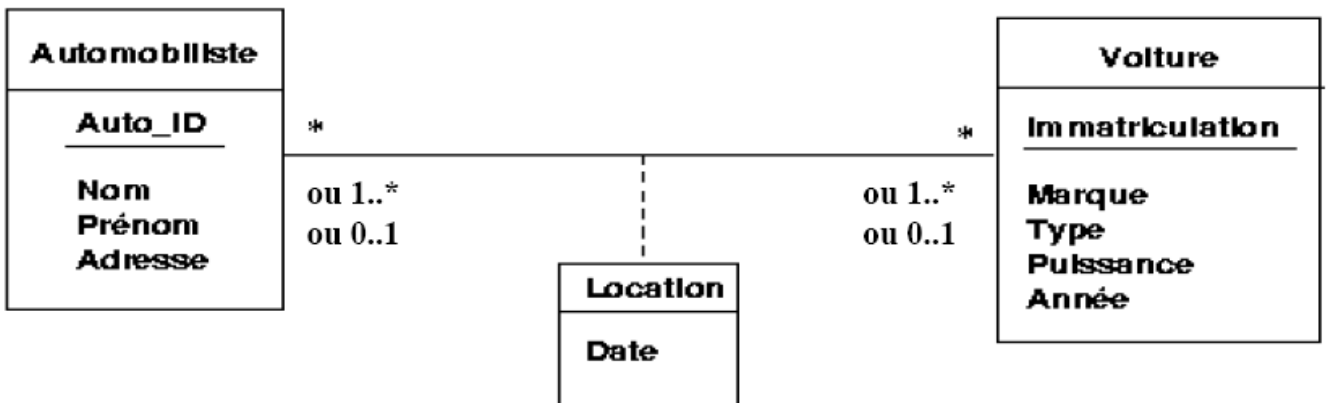
Automobiliste (Auto_ID, Nom, Prénom, Adresse)

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



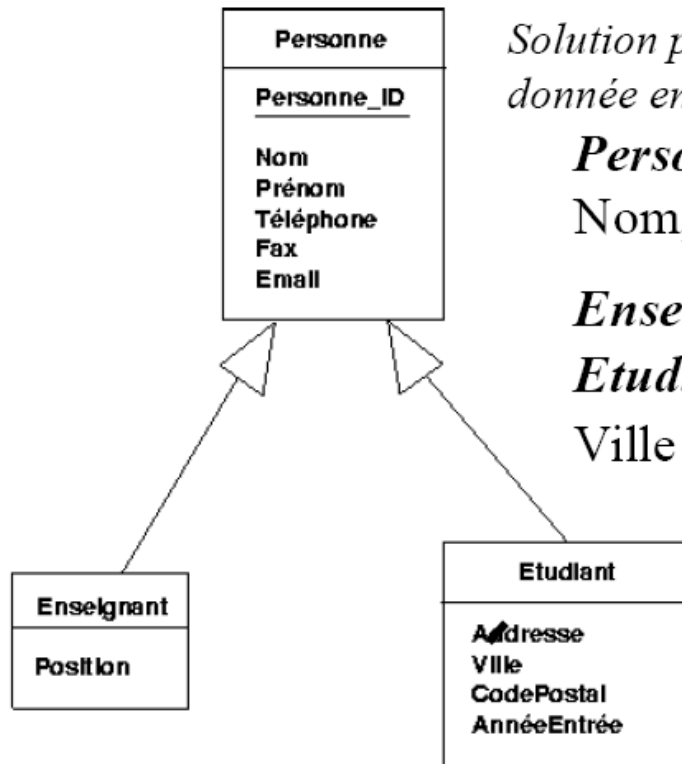
Automobiliste (Auto_ID, Nom, Prénom, Adresse)

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



Solution possible (une autre sera donnée en cours) :

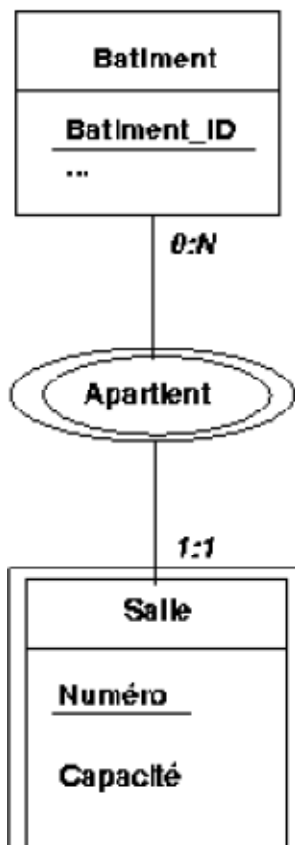
***Personne** (Personne_ID,
Nom, Prénom, Téléphone ...)*

***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



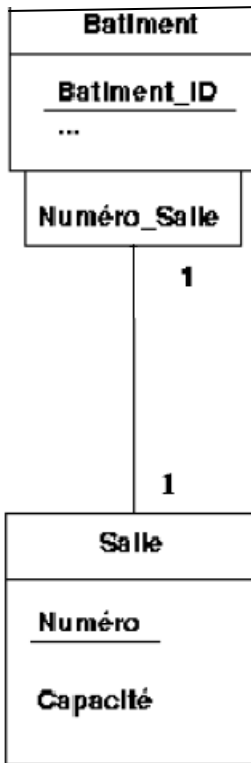
***Bâtiment** (Bâtiment_ID, ...)*

***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



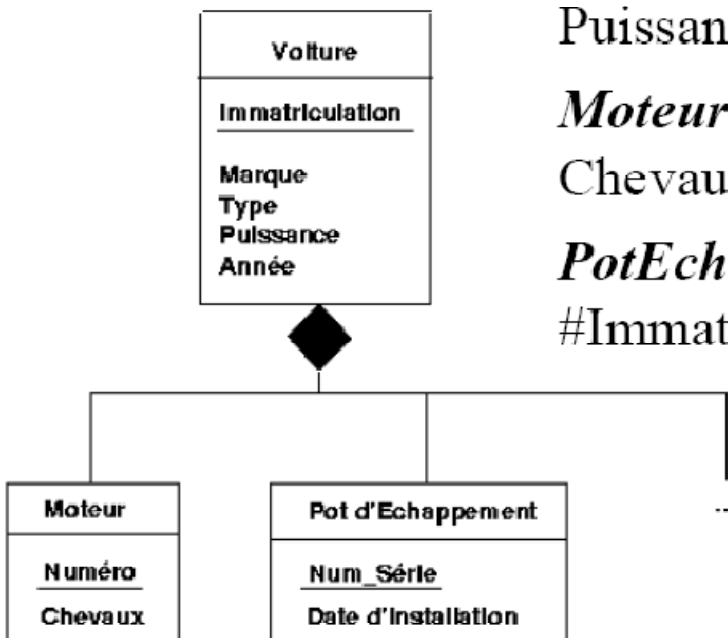
Bâtiment (Bâtiment_ID, ...)

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)