

# BASES DE DONNÉES AVANCÉES

# Contenu du module

2

- ❑ **Conception d'une base de données**
  - ❑ Introduction
  - ❑ Modèle entité-association
- ❑ **Implémentation d'une base de données**
  - ❑ Passage du MEA au MR
  - ❑ Présentation générale d'un SGBD
- ❑ **Manipulation des données**
  - ❑ Langage SQL et SQL avancé
  - ❑ SGBD par la pratique : PostgreSQL

# Plateforme UMTICE

3

- Cours : L3 SPI – Bases de données avancées
  - <http://umtice.univ-lemans.fr/enrol/self/edit.php?courseid=374>
  - Supports de cours
  - Supports de TD... quelques corrections
  - Supports de TP
  - Plateforme de dépôt du TP noté
- Clé d'inscription : L3BDA2017

# Évaluations

4

- Modalités de contrôle des connaissances du module
  - ▣ CC1 (coeff.0,5)
    - 1h écrit (le 10/02)
    - Questions de cours modélisation/implémentation + modélisation Entité-Association
  - ▣ CC2 (coeff. 1,5)
    - 2h écrit (le 24/04)
    - De la modélisation Entité-Association à l'implémentation en base de données relationnelle avec requêtes SQL
  - ▣ TP (coeff. 1)
    - 2h30 sur machine
    - Requêtes SQL – pg/plsql

5

# Introduction

# Une donnée

6

## □ Définition

- ▣ Une donnée est la *représentation codée d'une information quelconque du monde réel*

## □ Information du monde réel

- ▣ Un fait sur un « objet »
  - « Cette personne s'appelle Tony »
- ▣ Une relation entre « objets »
  - « Tony joue au basket ball »

# Base de Données

7

- Définition générale

*Un ensemble organisé d'informations avec un objectif commun.*

→ Ex : annuaire

- Base de données informatisée

- ▣ *Un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur pour consultation ou mise à jour.*

*« Ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique » (Gardarin 2001)*

# Enjeux des bases de données

8

- La gestion des données
  - ▣ De plus en plus de données (web!)
  - ▣ Gestion des données en mémoire secondaire
  - ▣ Bases de données distribuées
  - ▣ Accès rapide, sûr et efficace
  - ▣ Accès multi-utilisateurs
  - ▣ ...
- Besoin d'une application performante!



# Système de Gestion de Bases de Données

9

- Ensemble de procédures
  - ▣ Accès à la base de données
  - ▣ Gestion (ajout/modification) des données
  - ▣ Recherche par critères de données
  
- Hébergement de plusieurs bases de données
  - ▣ Thématiques différentes
  
- Architecture client/serveur (souvent)
  - ▣ Accès multiples, concurrentiels
  - ▣ Cohérence des données

# Objectifs d'un SGBD

10

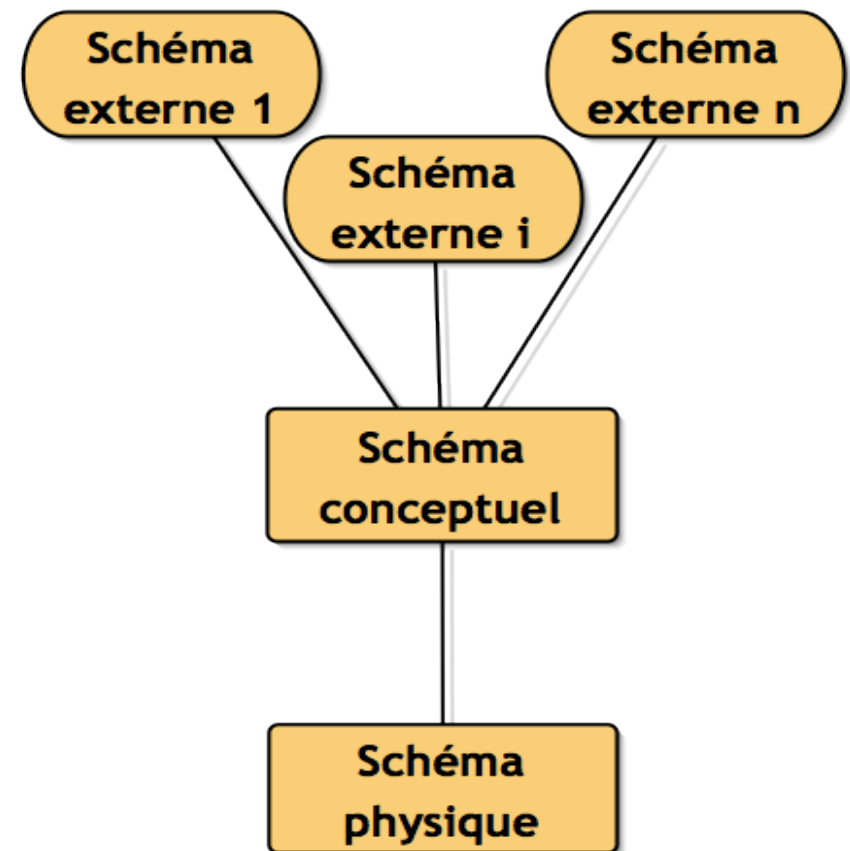
→ Masquer la représentation physique des données et assurer la protection et la cohérence des données dans un environnement multi-utilisateurs.

- ▣ Indépendance physique/logique
- ▣ Accès optimisé aux données
- ▣ Intégration des données
- ▣ Non redondance des données
- ▣ **Cohérence** des données
- ▣ Partage des données
- ▣ Sécurité des données
- ▣ Résistance aux pannes

# Architecture ANSI/SPARC

11

- Modélisation standard abstraite proposée en 1975 par la commission SPARC de l'ANSI
- Objectif :
  - ▣ séparation entre
    - Vision de l'utilisateur
    - Représentation physique



\* *American National Standards Institute*

\*\* *Standards Planning And Requirements Committee*

# Modèle 3-couches ANSI/SPARC

12

- Niveau physique (interne)
  - Organisation physique des fichiers et des méthodes d'accès (gestion des fichiers, des index...)
  - Fortement dépendant du SGBD
- Niveau conceptuel
  - Implémentation du *schéma conceptuel des données*
  - Indépendante du niveau physique
- Niveau externe
  - Vision de tout ou partie de la BDD par un utilisateur, indépendamment des autres (*vues*)
  - Indépendant de la complexité du schéma conceptuel de la BDD

# Conséquences et avantages

13

- Vues personnalisées indépendantes
  - Chacun sa vue! modification indépendante entre vues
  - Modification du schéma conceptuel => transparent pour l'utilisateur.
- Non-préoccupation du stockage physique
  - Travail sur des données, peu importe où, comment, celles-ci sont stockées.
  - Modification sur le stockage par l'administrateur sans modification des vues.

# Principaux SGBD en 2010

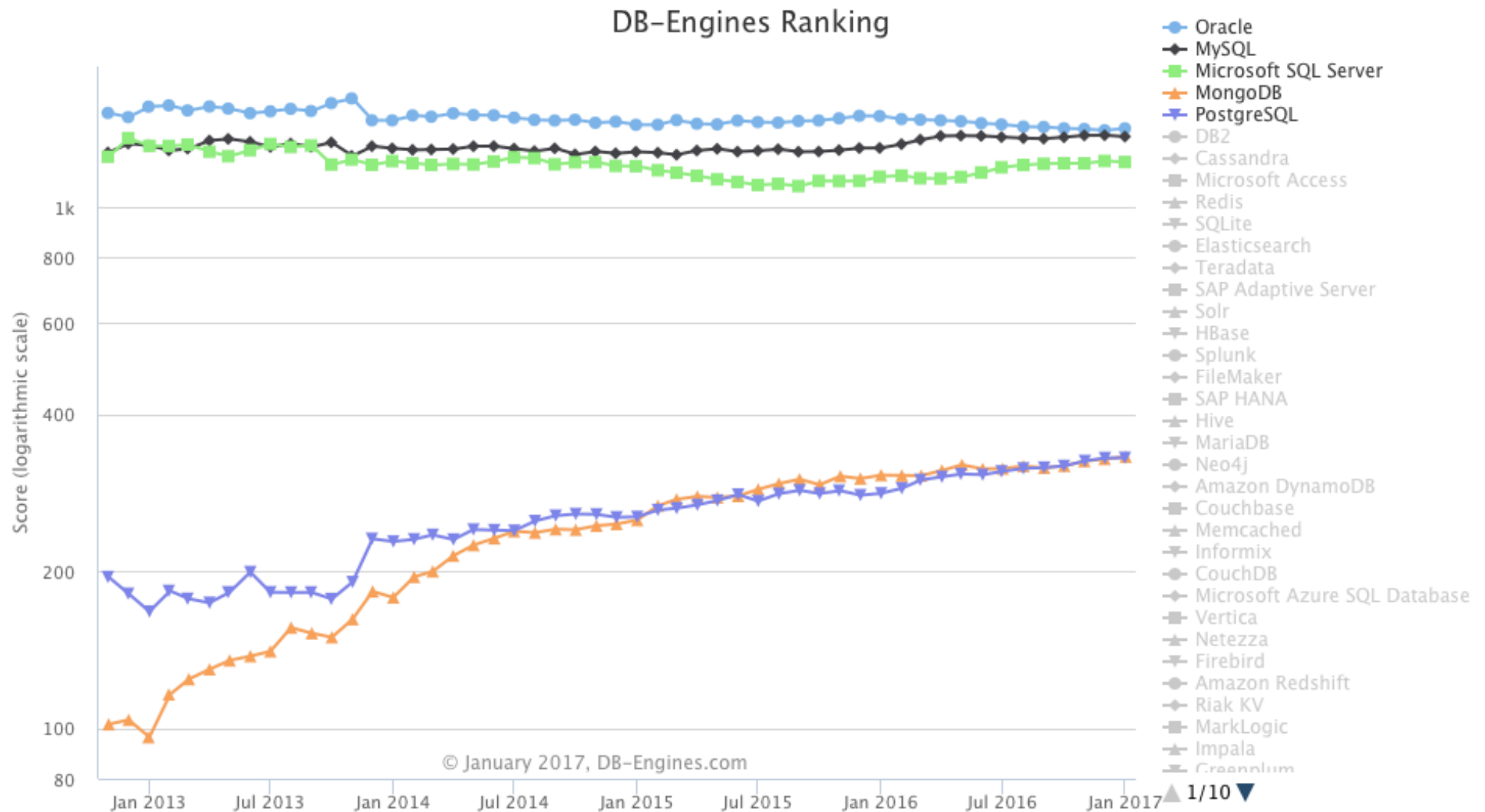
14

- Logiciels commerciaux
  - ▣ Oracle (45%\*)
  - ▣ IBM DB2 (21%\*)
  - ▣ Microsoft SQL Server (19%\*)
  - ▣ Sybase (3,5%\*)
- Logiciels libres
  - ▣ MySQL
  - ▣ PostgreSQL

\* : % en parts de marché pour 2010

# Evolution : open-source et no-sql

15



*16*

# Conception d'une base de données



# Concevoir une BD

17

- Description des données du monde réel pour stockage, mises à jour et accessibilité par ordinateur
- Formalisation par le schéma conceptuel
- Efficacité si au minimum:
  - ▣ Pas de perte d'information
  - ▣ Non redondance des données
  - ▣ Intégrité des données

# Composants du schéma conceptuel

18

- ▣ **Objet**

- Objet distinct, concret ou abstrait du monde réel

- ▣ **Entité**

- Ensemble d'objets possédant la même sémantique et des propriétés communes (abstraction de l'objet)

- ▣ **Propriétés** (caractéristiques, attributs)

- Caractéristiques décrivant un objet

- ▣ **Liens** (associations)

- Modélisation des contraintes existant sur les propriétés et les entités

- ▣ **Contraintes d'intégrité**

- Entités et liens y sont soumis pour garantir la vraisemblance avec le monde réel

# Modèles conceptuels

19

## □ Historiques

### ▣ Années 60 : Modèle hiérarchique

- Structure arborescente (lien 1:n uniquement)
- Besoin de connaître le chemin d'une donnée
- Très lié à l'organisation des fichiers sur l'ordinateur

### ▣ Fin années 60 : Modèle réseau

- Extension du modèle hiérarchique (possibilité n:m)
- Toujours trop lié à la structure physique

### ▣ 1970 : Modèle relationnel

# Modèle relationnel

20

- Introduit par E.F. Codd en 1970
  - ▣ Mathématicien au centre de recherche IBM San-José
  - ▣ *A Relational Model of Data for Large Shared Data Banks*
- Pourquoi?
  - ▣ Non satisfait des modèles
  - ▣ Utilisation d'une branche spécifique des mathématiques
- Objectifs
  - ▣ Accès performants à de grandes quantités de données
  - ▣ Réponse au problème de redondance ou d'intégrité des données
  - ▣ Séparation entre structure logique et mise en oeuvre physique

# Modèle relationnel

21

- À la base de la grande majorité des SGBD actuels
- Fin années 90 : Extension avec le modèle relationnel-objet
- Représentation de l'information selon
  - ▣ Une seule et unique structure de données : la relation
  - ▣ Des contraintes : spécification des règles que doit respecter la base de données
    - Notamment des types
  - ▣ Un langage non procédural
    - manipulation, interrogation et mise à jour des données

# Domaine de valeurs

22

## □ Définition

- Ensemble d'instances d'un type *élémentaire*
- Défini en intention :
  - Ex : entiers, réels, chaînes de caractères, ...
- Défini en extension :
  - Ensemble fini de valeurs

## □ En pratique

- Ensemble de définition des valeurs d'un attribut
- Exemples sur les pilotes:
  - Date de naissance : type *date*
  - Ville de rattachement: {"Marseille", "Toulouse", "Lyon", ...}

# Relation

23

- Appelée TABLE lorsque implémentée
- Définition
  - ▣ Sous-ensemble fini du produit cartésien d'une liste de domaines
- Exemple :
  - ▣ soient  $D1 = \{\text{Cathy, Paul, Xavier, Géraldine}\}$  et  $D2 = \{\text{Le Mans, Marseille, Nantes}\}$

*Relation*  
*“Est Né à”*

<i>“Est Né à”</i>	<i>D1</i>	<i>D2</i>
	<i>Cathy</i>	<i>Le Mans</i>
	<i>Paul</i>	<i>Le Mans</i>
	<i>Xavier</i>	<i>Le Mans</i>
	<i>Géraldine</i>	<i>Nantes</i>

# Attributs

24

- Colonne d'une relation caractérisée par un nom
  - ▣ Nom porteur de sens vis à vis de son contenu
  - ▣ Toujours associé à un domaine de valeurs
  - ▣ Référence pour effectuer des opérations
- ▣ Exemple :

*Relation*  
*"Est Né à"*

<i>"Est Né à"</i>	<b><i>Prénom</i></b>	<b><i>Ville natale</i></b>
	<i>Cathy</i>	<i>Le Mans</i>
	<i>Paul</i>	<i>Le Mans</i>
	<i>Xavier</i>	<i>Le Mans</i>
	<i>Géraldine</i>	<i>Nantes</i>



# N-uplets ou tuples

25

- Définition
  - ▣ Liste des valeurs des attributs (!domaine)
- ▣ Aussi appelé ENREGISTREMENT, correspond aux lignes d'une table
- Règles
  - ▣ Ordre des tuples dans la relation sans importance
  - ▣ Aucun doublon possible

□ Exemple :

*Relation*  
*"Est Né à"*

<i>"Est Né à"</i>	<b>Prénom</b>	<b>Ville natale</b>
	<i>Cathy</i>	<i>Le Mans</i>
	<i>Paul</i>	<i>Le Mans</i>
	<i>Xavier</i>	<i>Le Mans</i>
	<i>Géraldine</i>	<i>Nantes</i>

# Schéma de relation

26

## □ Définitions

- ▣ Nom de la relation suivi de la liste des attributs et de la définition de leurs domaines

$$R(A1:D1, A2:D2, \dots, An:Dn)$$

- ▣ *Degré d'une relation : nombre d'attributs*

## □ Exemple :

- `EST_NÉ(Prénom:CHARVAR, VilleNatale:CHARVAR)`

# Représentations de relation

27

- Représentation en INTENTION de la relation
  - ▣ Un *schéma de relation* représente les propriétés communes et invariantes des tuples qu'elle est susceptible de contenir.
  
- Représentation en EXTENSION de la relation
  - ▣ Une *table* propose une vue des tuples que la relation contient à un instant donné.

# Bien concevoir une Base de données

28

- Rappel en relationnel : 1 seule structure
  - ▣ Entités et liens sont représentés par des relations
- Conception d'une BDR
  - ▣ Représenter au mieux le monde réel en définissant un ensemble de schémas de relations
    - Comment?
    - Plusieurs possibilités...

# Conception de schémas relationnels

29

## □ Exemple : Voyages par avion

### ▣ Choix 1: 1 relation Voyage

VOYAGE (NUMPIL, NOMPIL, ADRPIL, NUMAV, NOMAV, CAPAV,  
LOCAV, NUMVOL, V-D, V-A, H-D, H-A)

### ▣ Choix 2 : 3 relations

AVION (NUMAV, NOMAV, CAPAV, LOCAV)

PILOTE (NUMPIL, NOMPIL, ADRPIL)

VOL (NUMVOL, PILOTE, AVION, V-D, V-A, H-D, H-A)

# Conception de schémas relationnels

30

## □ Choix 1 (projection sur 10 attr. par manque de place ...)

NumVo l	NomPi l	AdrPil	NomAv	CapA v	LocAv	V-D	V-A	H-D	H-A
V001	Jean	Arles	B747	500	Paris	Paris	Nice	8:00	9:00
V023	Jean	arles	A300	300	Antibes	Nice	Pau	13:00	13:45
V045	Arthur	Nantes	B747	500	Nice	Nice	Paris	11:00	12:00

- Représentation des pilotes en vacances? des avions en révision?
- Problème de redondance? Risque de perte ? Risque d'incohérence?

# Conception de schémas relationnels

31

## □ Choix 2

NumPil	NomPil	AdrPil
Pil1	Jean	Arles
Pil2	Jean	Arles
Pil3	Arthur	Nantes

NumAv	NomAv	CapAv	LocAv
Av1	B747	500	Paris
Av1	A300	300	Antibes
Av2	B747	500	Nice

NumVol	NumPil	NumAv	V-D	V-A	H-D	H-A
V001	Pil1	Av1	Paris	Nice	8:00	9:00
V023	Pil1	Av1	Nice	Pau	13:00	13:45
V045	Pil2	Av2	Nice	Paris	11:00	12:00

- Comment être certain que tous les critères sont remplis?

# Normalisation

32

- Relation universelle
  - ▣ Relation composée de tous les attributs des entités et liens à modéliser par la BD
- Approche par décomposition
  - ▣ Objectif : Décomposer la relation universelle **en sous relations** n'ayant pas les anomalies signalées précédemment
    - Application de l'algèbre relationnel
    - Étude des dépendances fonctionnelles et multi-valuées
    - Formes normales ...



# Modèle entité-association

33

- Proposé en 1976 par Chen
- Représentation graphique permettant de modéliser le monde réel
  - ▣ Selon les concepts d'entité et d'association
  - ▣ Indépendamment du logiciel utilisé pour l'implémentation
  - ▣ Description graphique intelligible
- Synonyme :
  - ▣ entité-relation (mal traduit de l'anglais : entity-relationship)

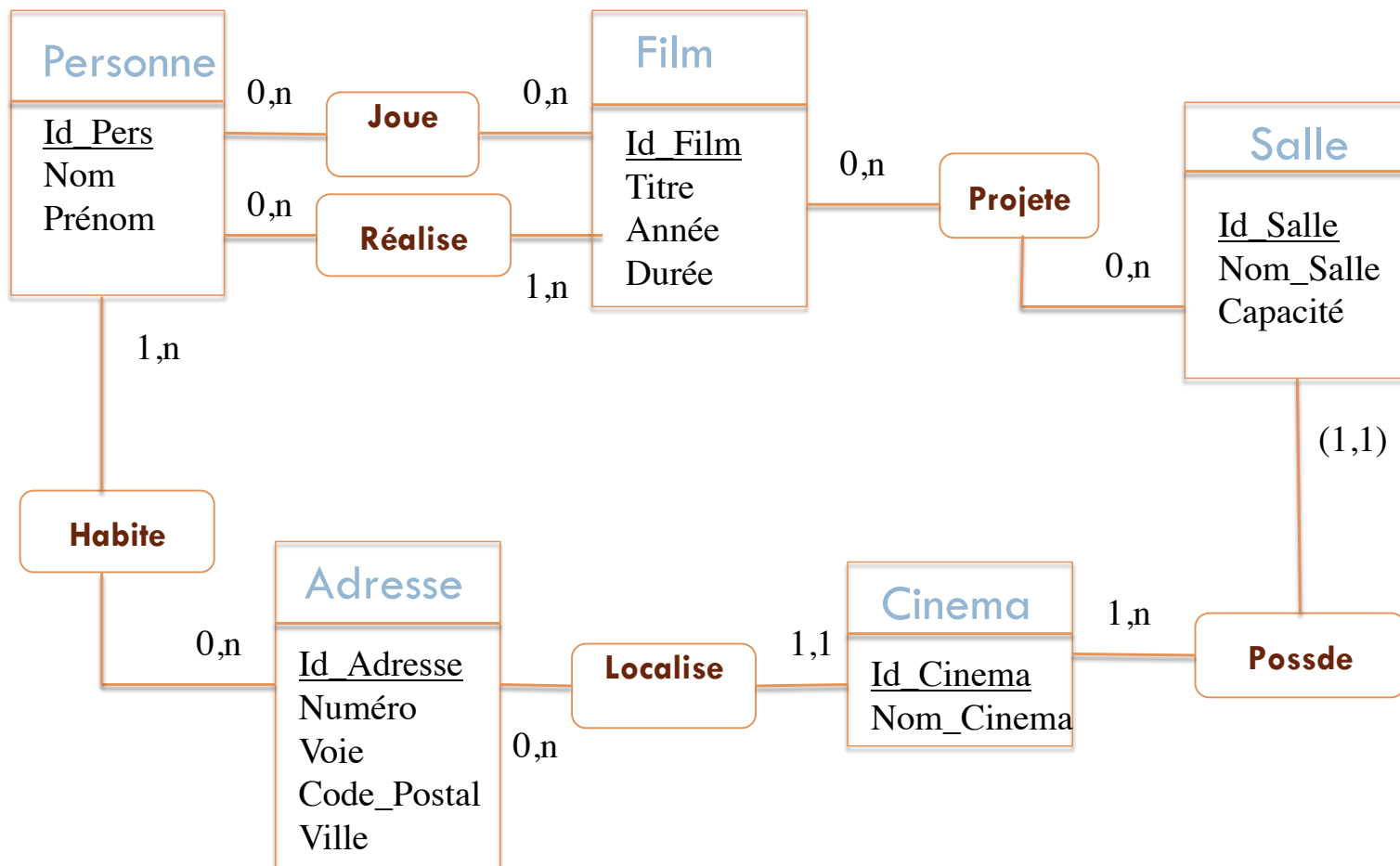
# Mise en oeuvre

34

- Préoccupations définies selon une démarche en 3 niveaux :
  - ▣ Niveau conceptuel
    - Description du monde réel selon les entités et les associations
  - ▣ Niveau logique
    - Choix du modèle conceptuel et mapping MEA/MC (MR dans notre cas)
  - ▣ Niveau physique
    - Choix du SGBD

# À quoi ça ressemble?

35



# Type-entité et entité

36

- Entité (ou objet) :
  - ▣ Objet concret ou abstrait du monde réel
    - Ex : mon stylo rouge, le vol Paris-Lyon
  - ▣ Correspond à un tuple dans une BDR
- Type-entité :
  - ▣ Regroupement **nommé** cohérent d'entités
    - Ex : stylo, voiture, vol
- Attention à l'abus Type-entité /Entité ...

# Exemples de type-entité

37

## □ Exemple

### ▣ Type-entité Article

- Dans une entreprise, les articles vendus peuvent être regroupés dans un même type-entité car les informations d'un article à l'autre ne changent pas (désignation, prix,...)

### ▣ Type-entité Article-Client ?

- Dans une entreprise on ne regroupera pas des articles et des clients au sein d'un même type-entité car leurs informations ne sont pas homogènes : un article n'a pas d'adresse, un client n'a pas de prix

# Les attributs du type-entité

38

- Attribut
  - ▣ Caractéristique/propriété de l'entité
  - ▣ Associé à un type-entité
- Donnée élémentaire
  - ▣ Pas d'attribut calculé
- Unicité de l'attribut dans le modèle
  - ▣ **Pas de clé étrangère**
  - ▣ Deux attributs ne peuvent représenter la même caractéristique

# Les valeurs

39

- Valeur
  - ▣ Chaque attribut est associé à un domaine de valeurs
  - ▣ Une valeur compatible pour chaque attribut de chaque entité
- Pas de valeurs possibles car non-sens?
  - ▣ Revoir sa modélisation !

# Identifiant (clé) du type-entité

40

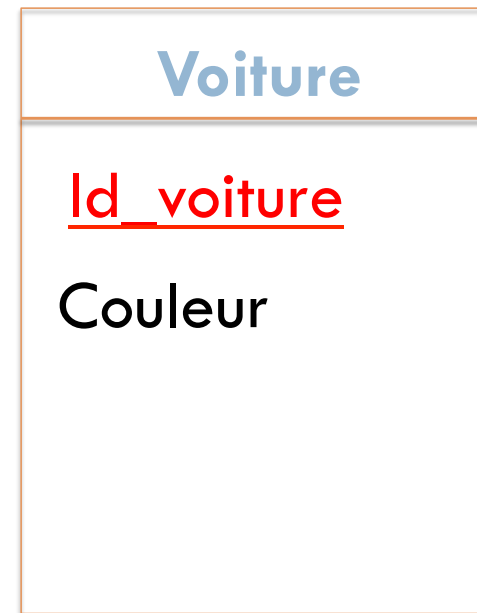
- Ensemble minimal d'attributs du type-entité permettant d'identifier de manière unique chaque entité
  - ▣ Notion de clé primaire en MR
  - ▣ Impossibilité pour 2 entités d'un même type-entité d'avoir la même valeur d'identifiant
- Chaque Type-entité possède obligatoirement un identifiant, éventuellement composé de plusieurs attributs



# Formalisme du type-entité

41

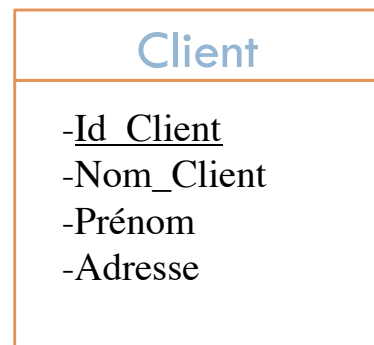
- Un rectangle
- Le nom encadré en haut
- L'identifiant souligné
- Les attributs
  
- Exemple sur l'entité
  - ▣ Nom : voiture
  - ▣ Identifiant: id\_voiture
  - ▣ Autre attribut : couleur



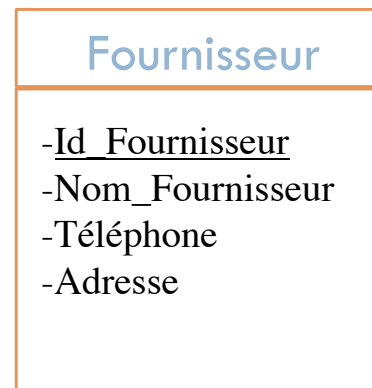
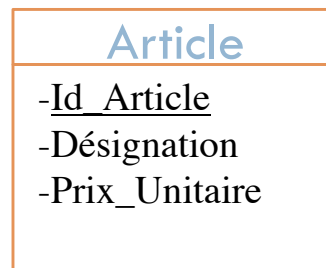
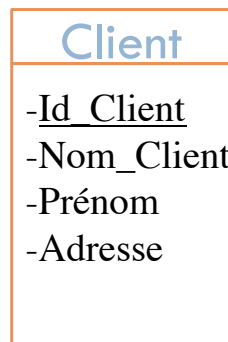
# Mauvaise conception d'un type-entité

42

- Exemple 1 : 2 types-entités au sein d'une seule



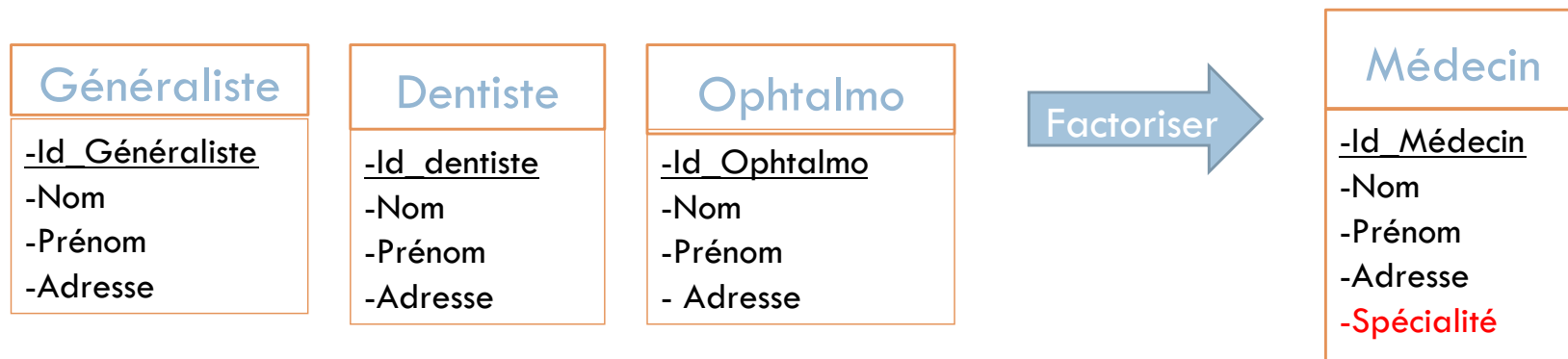
**NON**



# Mauvaise conception d'un type-entité

43

## □ Exemple 2 : Factorisation de 2 types-entités



# Type-association et association

44

## □ Association

- Lien sémantique existant entre plusieurs entités

  - Lie généralement 2 ou plus entités

  - Ex : François *commande* un iphone

## □ Type-association

- Ensemble ***nommé*** d'associations de mêmes caractéristiques

- Description d'un lien entre plusieurs types-entités

# Type-association et association

45

- Participant
  - ▣ Type-entité impliqué dans le type-association
- Collection
  - ▣ Ensemble des participants du type-association
- Arité
  - ▣ Nombre de participants du type-association
- Attribut
  - ▣ Possibilité d'avoir des attributs uniquement si dépendance avec tous les participants
    - Ex : François a commandé son iphone le 5 février 2012

# Cardinalité du type-association

46

## □ Cardinalité

- ▣ Nombre de fois au minimum et au maximum où l'entité du type-entité intervient dans l'association du type-association, avec généralement :
  - $\text{Minimum} \in \{0,1\}$
  - $\text{Maximum} \in \{1,n\}$
- ▣ Indication obligatoire !!!
- ▣ Sens de lecture
  - !! Différent de l'UML ...

# Cardinalités admises

47

- **0,1** : Une entité participe une fois au plus à l'association. Elle existe même si elle n'est pas impliquée dans la relation.
- **0,n** : Une entité participe ou non sans limitation de nombre à l'association
- **1,1** : Une entité participe obligatoirement une fois à l'association. Elle n'existe pas sinon.
- **1,n** : Une entité n'existe que si elle participe au moins une fois à l'association.

# Formalisme du type-association

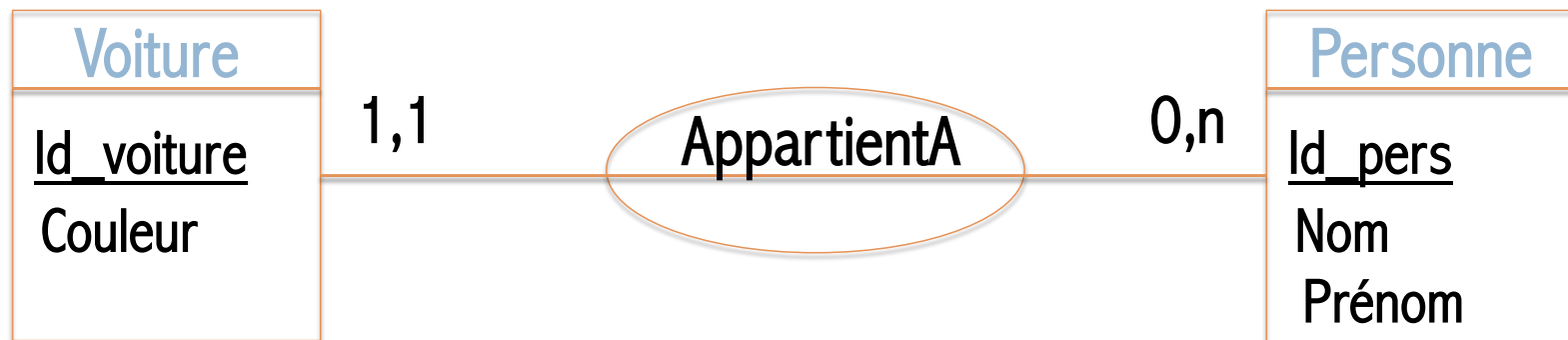
48

## □ Formalisme

- Un ovale
- Un nom
- Des pates de liaison
- Des cardinalités

## □ Exemple

Un propriétaire possède de 0 à n voitures. Une voiture n'a qu'un seul et unique propriétaire.





# Classement des types-associations

49

## □ **Type-association hiérarchique**

- $1,1 - 0|1,n$
- Arité: 2
- Aucun attribut dans le type-association
- Le type-entité  $1,1$  est le type-entité inférieur
- Le type-entité  $0|1,n$  est le type-entité supérieur

# Classement des types-associations

50

- **Type-association non-hiérarchique (maillée)**

- $0|1,n - 0|1,n$
- Arité: 2 à n
- Possibilité d'attributs dans le type-association

- **Type-association semi-hiérarchique**

- $0,1 - 0|1,n$
- Arité: 2
- Possibilité d'attributs dans le type-association

# Classement des types-associations

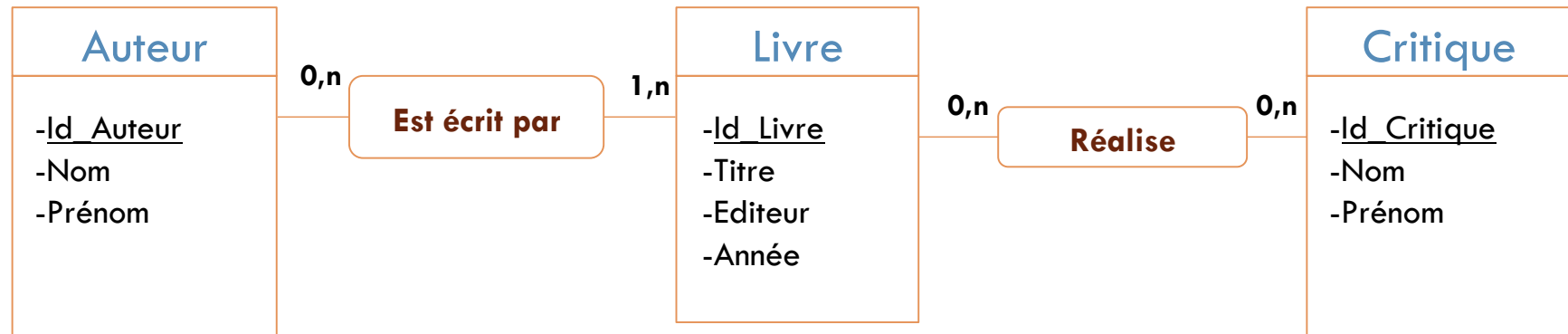
51

## □ Tableau récapitulatif

Cardinalités	Arité	Type	Attributs?
1,1 – 0,1	2	Hiérarchique	non
1,1 – 0   1,n	2	Hiérarchique	non
0,1 – 0   1,n	2	Semi-hiérarchique	oui/non
0   1,n – 0   1,n	2 à n	Non-hiérarchique	oui/non
1,1 – 1,1	??	??	??

# Type-association plurielle

52



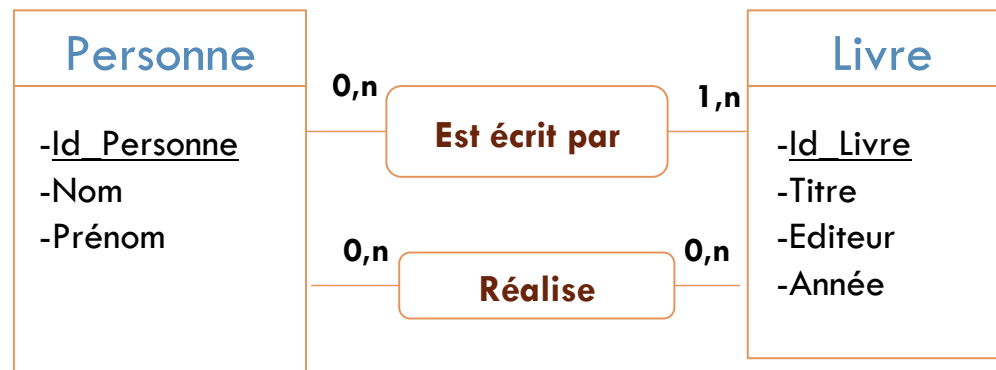
- ▣ Le type-association ***Est écrit par*** modélise que des personnes écrivent des livres
- ▣ Le type-association ***Réalise*** modélise que des personnes critiquent des livres

!! Auteur et Critique modélisent une personne, peut-être la même?? → Factorisation

# Type-association plurielle

53

- Deux mêmes types-entités peuvent être impliqués dans plusieurs types-associations



- ▣ Une même personne peut être à la fois auteur et critique
- ▣ L'association détermine le rôle de l'entité

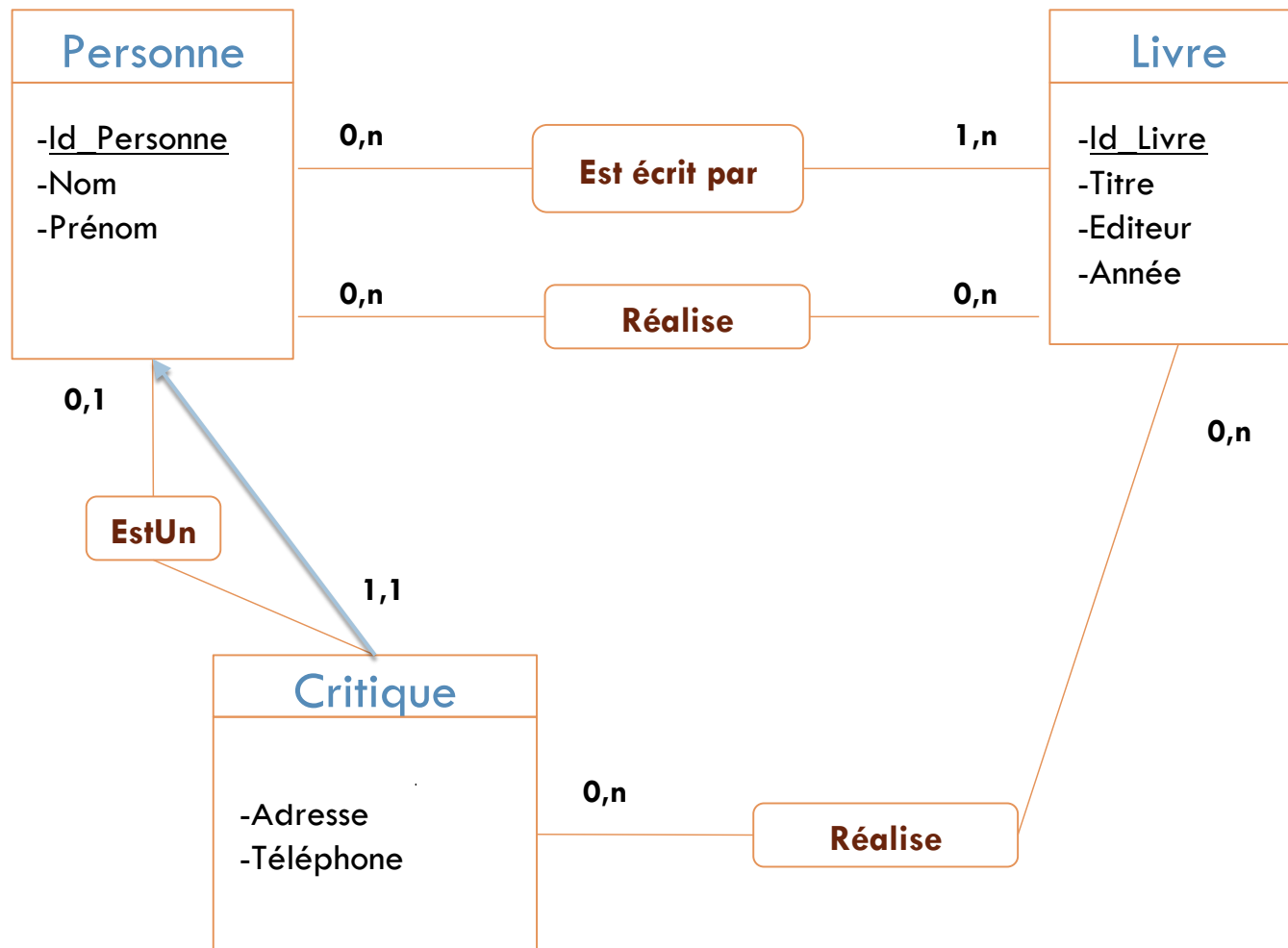
# Spécialisation

54

- Type-association « EstUn »
  - ▣ Type-entité générique
  - ▣ Type-entité spécialisé
  - ▣ Type association de cardinalité 0,1 – 1,1
    - Chaque entité générique est associée à au plus une entité spécifique
    - Chaque entité spécifique est associée à exactement une entité générique
  - ▣ Conséquence : L'entité spécialisée n'a pas besoin d'identifiant propre...

# Spécialisation

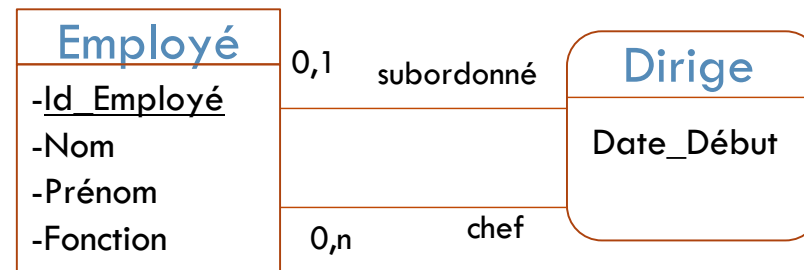
55



# Type-association réflexif

56

- Un type-association peut être lié plusieurs fois au même type-entité



- ▣ Tout employé est dirigé par un autre employé (sauf le directeur général)
- ▣ Un employé peut diriger plusieurs autres employés



# Type-association réflexif

57

- Quelques remarques :
  - ▣ Spécification des rôles sur les pattes
  - ▣ Un type-association réflexif n'entraîne pas que ces instances soient réflexives
  - ▣ Un type association peut être symétrique
    - Ex : être frère
    - Dans ce cas, pas de spécification du rôle

# Identifiant absolu ou relatif

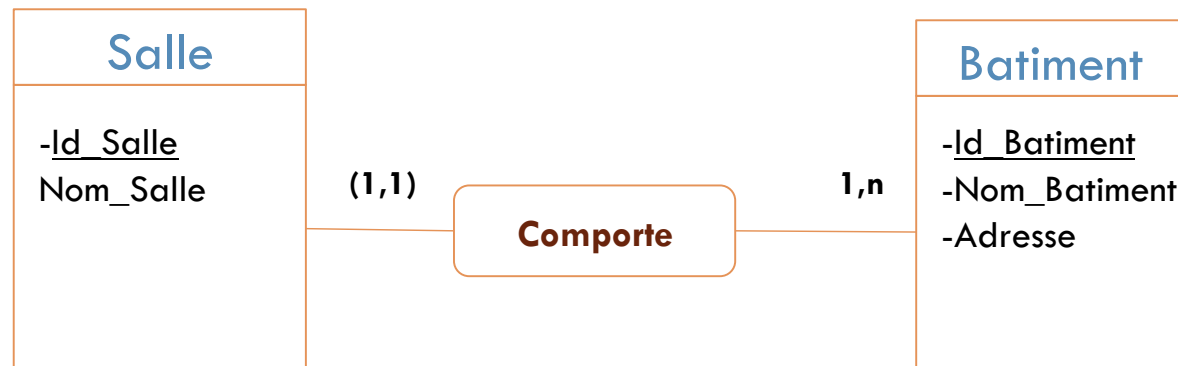
58

- Identifiant absolu
  - ▣ Composé uniquement d'attributs propres
- Identifiant relatif
  - ▣ Composé au minimum d'un attribut qui ne lui est pas propre
  - ▣ Pas de représentation particulière
- Lien identifiant
  - ▣ Lien particulier avec une patte 1,1 du côté de l'agrégé
  - ▣ Symbolisé par des parenthèses

# Identifiant absolu ou relatif

59

## □ Exemple



- La dépendance forte entre l'existence du bâtiment et de la salle est modélisée
- Deux salles de deux bâtiments distincts pourront avoir le même identifiant Id\_Salle

# Décomposition d'un type-association n-aire

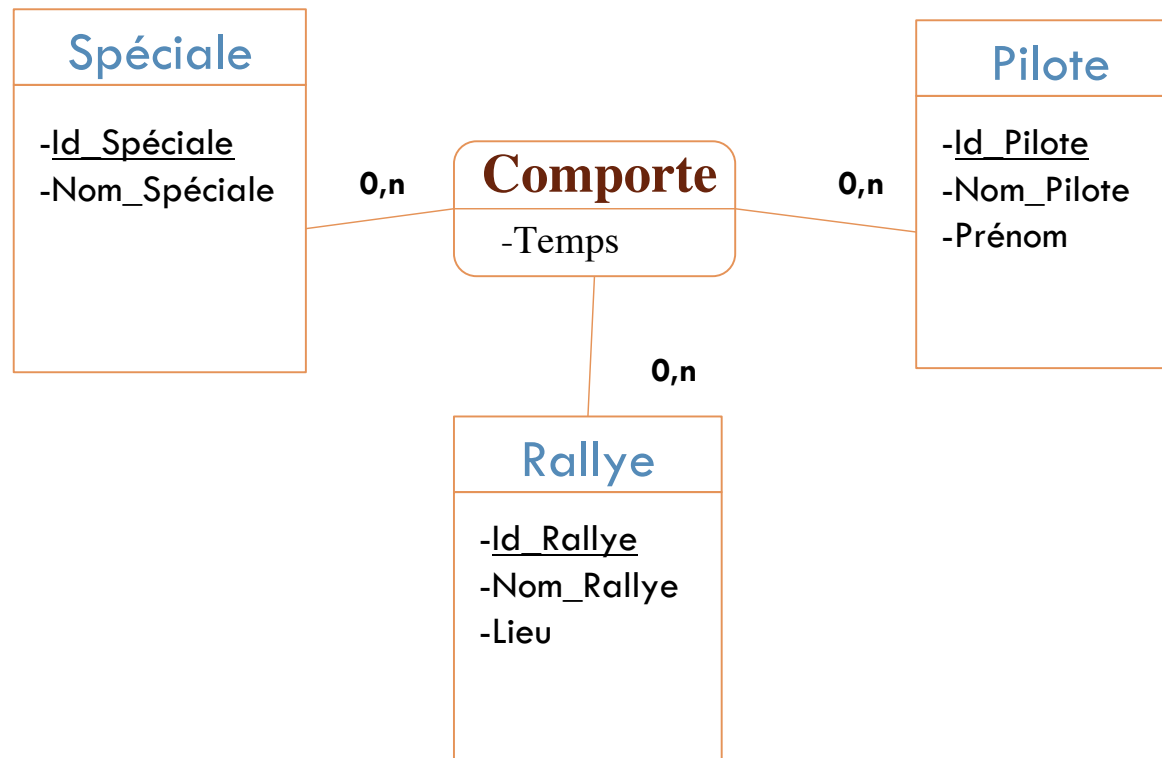
60

- Type-association n-aire : plus de 2 pattes
- À vérifier obligatoirement car souvent non valide
- Conseil : décomposition
  - ▣ Créer un type-entité à la place du type-association
    - Identifiant explicite non obligatoire
  - ▣ Créer des types-associations binaires à la place de chaque patte
  - ▣ Cardinalités (1,1) du côté de l'ex-type-association

# Décomposition d'un type-association n-aire

61

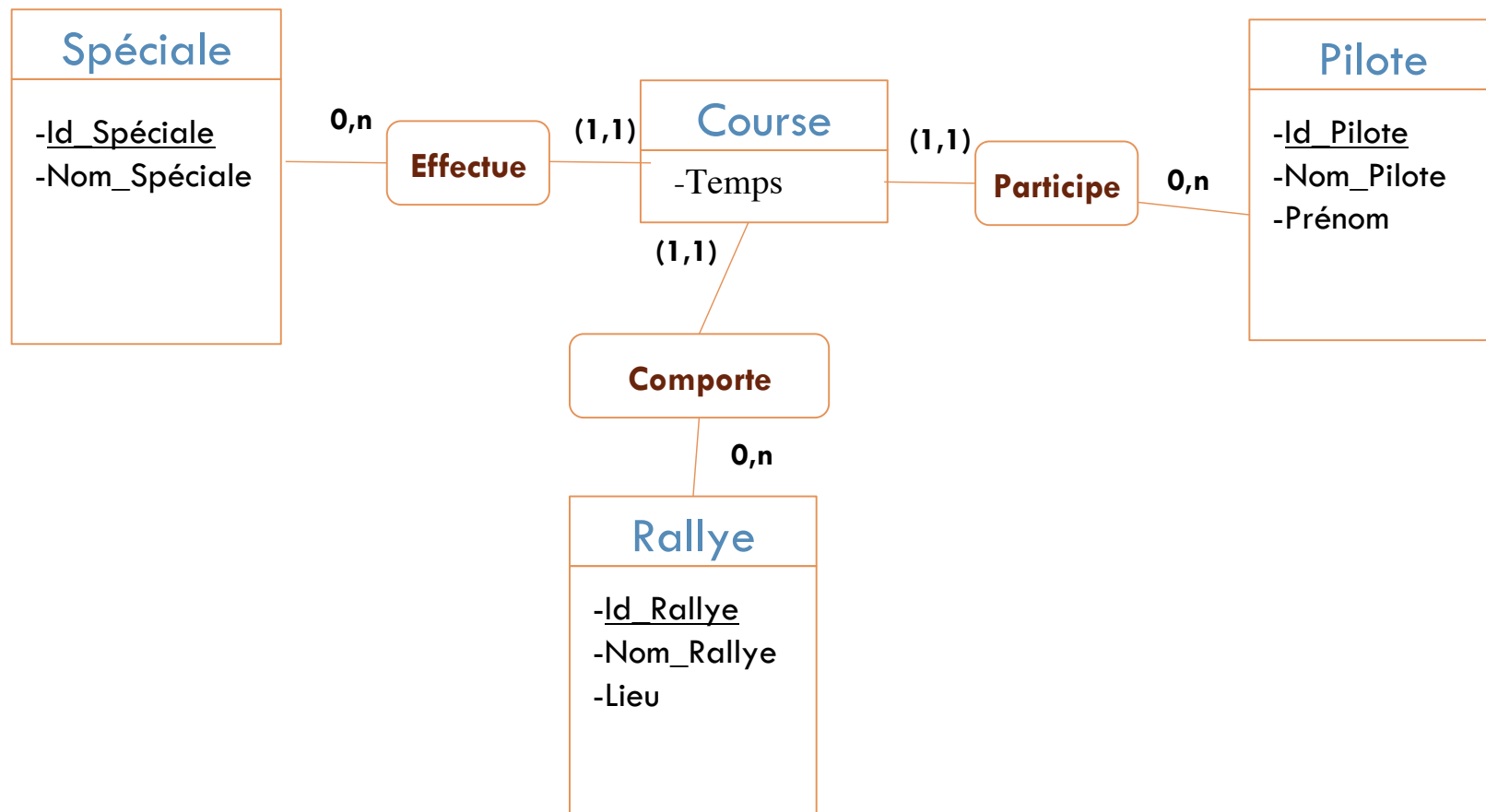
## □ Exemple



# Décomposition d'un type-association n-aire

62

## □ Exemple



# Vérification du MEA

63

- Ai-je bien modélisé le problème?
  - ▣ Règles syntaxiques
  - ▣ Règles sémantiques
  - ▣ Formes normales

# Vérifications syntaxiques

64

- ▣ Unicité de tout nom : type-entité, type-association ou d'attribut
- ▣ Normalisation des attributs multiples et/ou composites
- ▣ Clé primaire :
  - ▣ Une clé primaire pour chaque type-entité
  - ▣ Un seul attribut pour clé primaire dans un type-entité
  - ▣ Aucune clé étrangère
- ▣ Type-association :
  - ▣ Aucun attribut dans un type-association hiérarchique
  - ▣ Aucun type-association hiérarchique avec arité  $> 2$



# Vérifications sémantiques

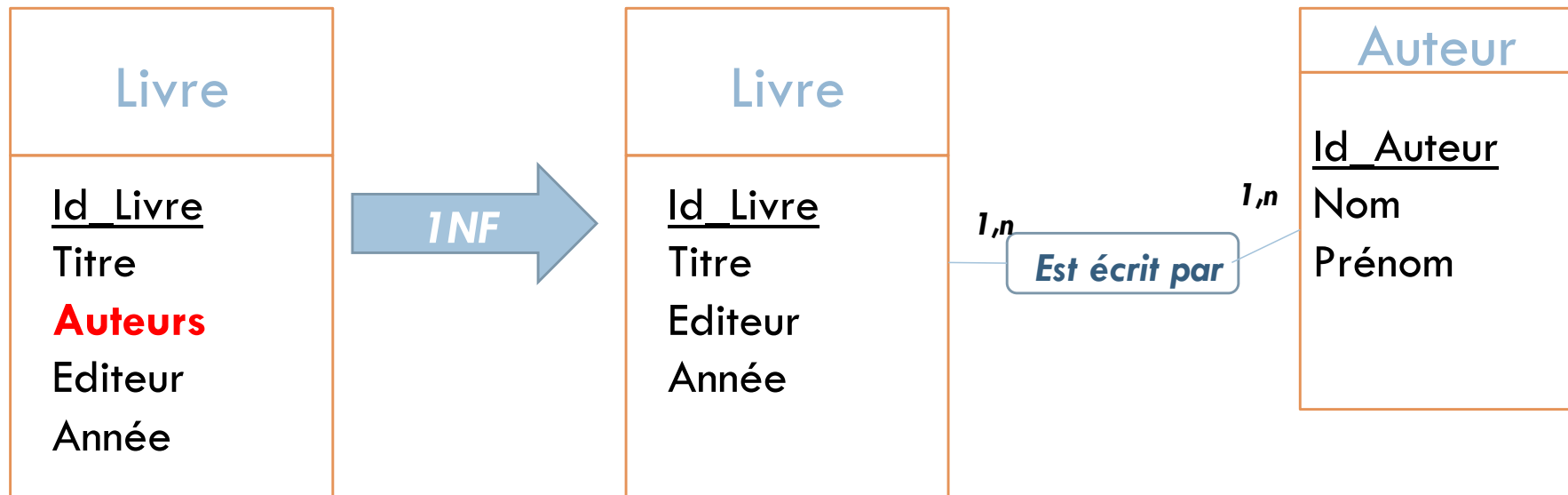
65

- Clé primaire:
  - ▣ Dépendance fonctionnelle de chaque attribut avec la clé primaire de son type-entité
  - ▣ Type de la clé primaire non chaîne de caractères
- Attribut
  - ▣ Apparition unique de chaque attribut
    - Deux attributs ne peuvent désigner la même réalité
  - ▣ Aucun attribut calculé à partir d'autres attributs (sauf si nécessaire à la compréhension)
- Type-entité :
  - ▣ Factoriser si nécessaire/possible

# Vérification des formes normales

66

- 1 NF : Tout attribut est atomique, non décomposable



# Vérification des formes normales

67

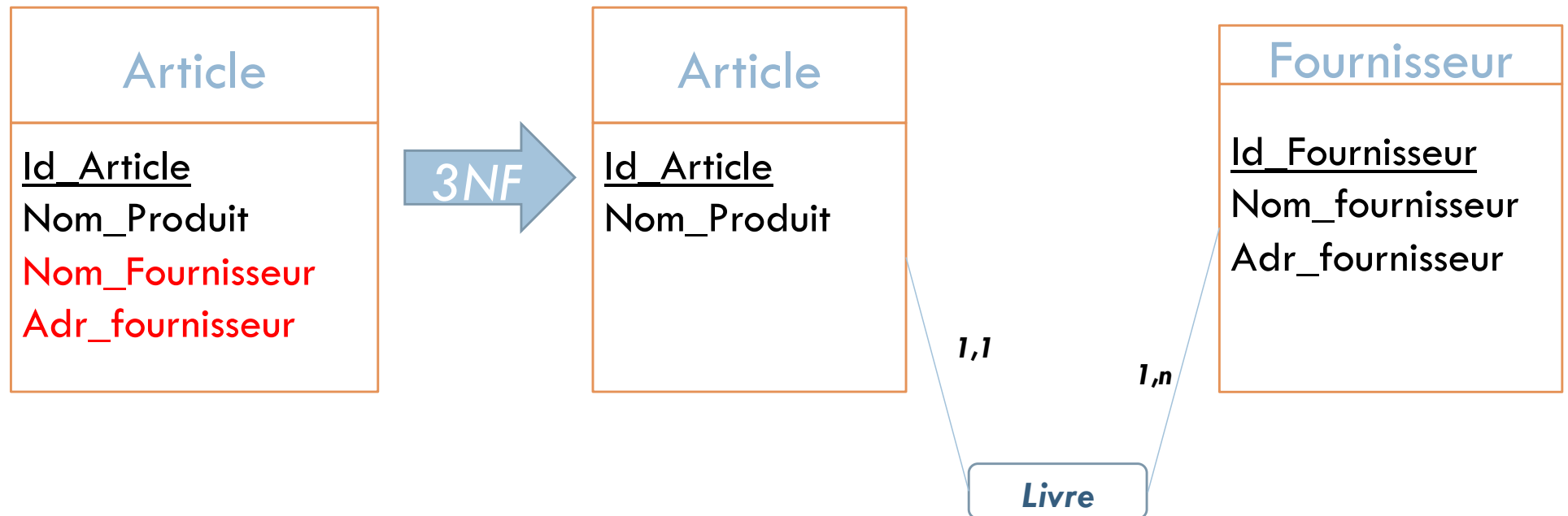
- 2 NF : 1 NF + Tout attribut non clé ne doit pas dépendre que d'une partie de la clé



# Vérification des formes normales

68

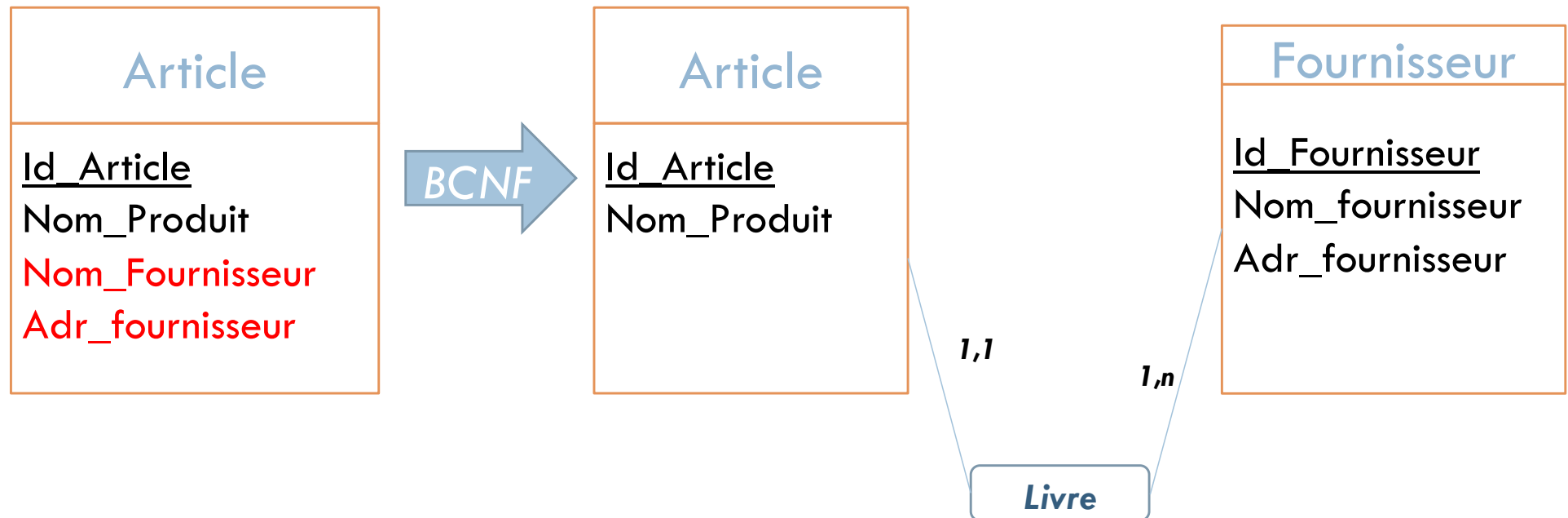
- 3 NF : 2NF + Tout attribut non clé ne doit pas dépendre d'un attribut non clé



# Vérification des formes normales

69

- BCNF : Toute dépendance est de la forme clé primaire détermine attribut



# Etapes de conception d'un MEA

70

- Inventaire de l'ensemble des données à considérer
- Analyse de l'information
  - ▣ Créations des attributs
  - ▣ Redondance?
- Types-entités?
  - ▣ Clé primaire?
- Types-associations?
  - ▣ Cardinalités?
- Vérification du modèle

# Du MEA au MR

71

- Cette étape a obligatoirement lieu après la normalisation
- À partir d'une modélisation en entité-association :
  - ▣ Définition de l'ensemble des relations à implémenter
    - Application de 4 règles
    - Obtention du schéma de relations
  - ▣ Représentation logique indépendante du stockage physique

# Du MEA au MR

72

## □ Règle 1 – Type-entité:

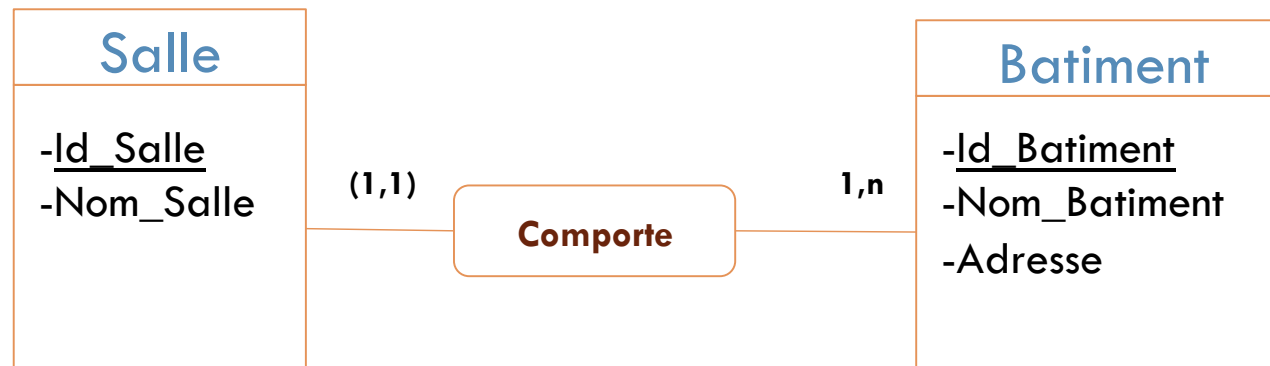
- ▣ Tout type-entité devient un schéma de relation
- ▣ Tout attribut du type-entité devient attribut du schéma de relation
- ▣ L'identifiant du type-entité devient clé primaire du schéma de relation
  - Attention aux types-entités spécifiques ou agrégés



# Du MEA au MR

73

## □ Règle 1 – Exemple



- Batiment(Id\_Batiment, Nom\_Batiment, Adresse)
- Salle(Id\_Salle, #Id\_Batiment, Nom\_Salle)

# Du MEA au MR

74

## □ Règle 2 – Type-association non hiérarchique:

- ▣ Le type-association devient un schéma de relation
- ▣ Tout attribut du type-association devient attribut du schéma de relation
- ▣ Les clés primaires des types-entités de la collection deviennent clés étrangères dans le schéma de relation
- ▣ La détermination de la clé primaire n'est pas automatique
  - Souvent : concaténation des clés primaires des types-entités participant

# Du MEA au MR

75

## □ Règle 2 – Exemple



□ Ecriture(#Id\_Livre,#Id\_Auteur)

# Du MEA au MR

76

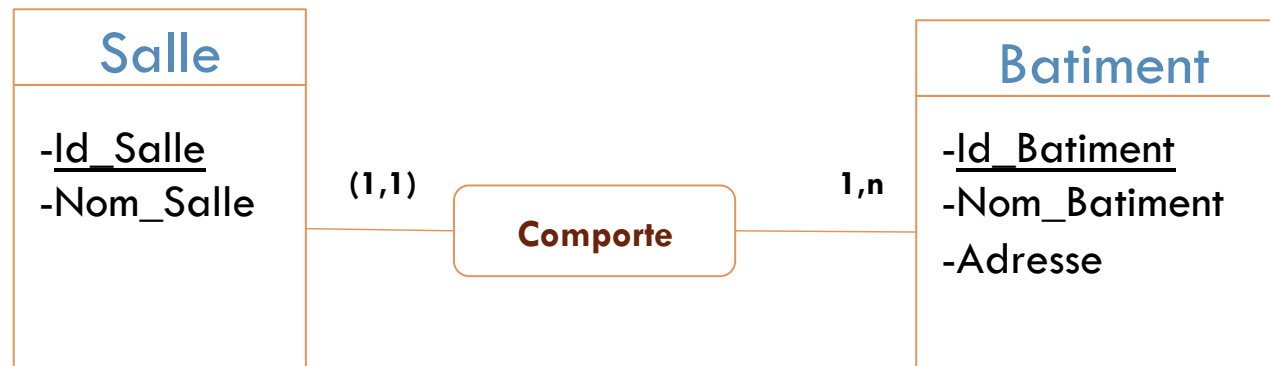
## □ Règle 3 – Type-association hiérarchique

- Clé primaire du type-entité "0 | 1,n" devient clé étrangère du type-entité "1,1"
  - Contrainte : not null
- Cas d'un type-association réflexif :
  - ajout d'un nouvel attribut dans le type-entité
- Cas d'un type-association « EtreUn » :
  - La clé étrangère doit être rajouté dans la clé primaire

# Du MEA au MR

77

## □ Règle 3 – Exemple :



□ Salle(Id\_Salle, #Id\_Batiment<sup>∅</sup>, NomSalle)

# Du MEA au MR

78

- Règle 4 – Type-association semi-hiérarchique
  - ▣ Si attributs : cas non-hiérarchique
  - ▣ Si pas attributs : cas hiérarchique avec contrainte *not null* non obligatoire