

# Conception de Bases de données

**Dr. Rania OTHMAN**

[rania.othman@intervenants.efrei.net](mailto:rania.othman@intervenants.efrei.net)



**02**

## **Conception et modélisation de base de données**



# Modèle relationnel

# Vers un schéma relationnel

Un modèle de données définit un mode de représentation de l'information selon 3 composants :

1. Structures de données
2. Ensemble de contraintes permettant de spécifier les règles que doit respecter la base de données
3. Un ensemble d'opérations permettant de manipuler les données, en interrogation et en mise à jour

⇒ **1 et 2. définissent le langage de définition des données (LDD ou DDL) : Description du schéma d'une BD**

⇒ **3. représente le langage de manipulation des données (LMD ou DML) : SQL**

# MR : Relation, clé primaire et attributs

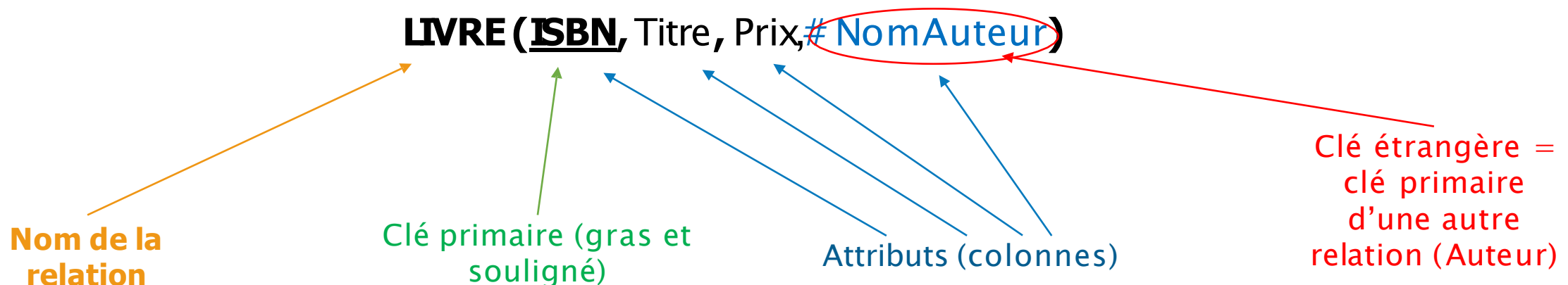
Les données sont stockées dans des **relations/tables** (représentées par des tableaux à deux dimensions).

Exemple : « Les informations sur les livres dans une bibliothèque »

ISBN ⚡	Auteur ⚡	Titre ⚡	Prix ⚡
9782738108395	Samuel P. Huntington	Le choc des civilisations	25
2-213-02191-0	Nicolas Grimal	Histoire de l'Égypte ancienne	42
2-89448-084-9	Jacques Lacoursière	Histoire populaire du Québec	15

→ Une relation est un ensemble de **t-uple**

→ Un t-uple est défini par **un ou plusieurs attributs**



# Clé primaire (identifiant de la relation)

**Clé primaire (identifiant)** : *sous-ensemble des attributs tel qu'il ne peut exister à aucun moment plus d'une ligne possédant les mêmes valeurs de ces attributs*

## Quelques propriétés :

1. identifiant *minimal*
2. tout ensemble d'attributs dont une partie stricte est un identifiant est aussi un identifiant (mais *non minimal*)
3. plusieurs identifiants minimaux peuvent coexister dans une relation
4. un attribut peut appartenir à plusieurs identifiants
5. il est possible de calculer automatiquement les identifiants d'une relation

# Production du schéma relationnel à partir du modèle E/A

**Objectif** : Produire, à partir du schéma conceptuel (modèle E/A), un schéma de base de données relationnel en LDD (langage de description des données).

## **Comment ?**

→ traduction du schéma conceptuel sous forme de tables, de colonnes, d'identifiants et de clés étrangères

# 1<sup>ère</sup> Règle de transformation

## De Entité vers Relation (Table)

- ✓ Toute entité du modèle conceptuel devient une relation, et donc une table dans la BD.
- ✓ Chaque propriétés de l'entité devient un attribut de cette relation, et donc une colonne de la table correspondante
- ✓ L'identifiant de l'entité devient la Clé Primaire de la relation, et donc la clé primaire de la table correspondante

MCD

CLIENT	
<u>ID_CLIENT</u>	INT
NOM_CLIENT	VARCHAR(50)
NUM_TEL	BIGINT

MLD

**CLIENT = (ID\_CLIENT INT, NOM\_CLIENT VARCHAR(50), NUM\_TEL BIGINT);**

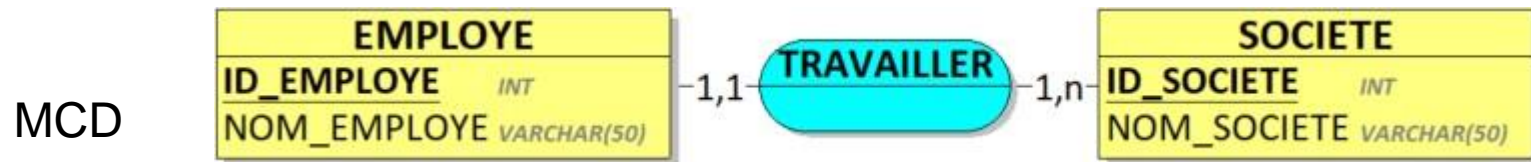


## 2<sup>ème</sup> règle de transformation

### Relation binaire aux cardinalités (X,1)-(X,N) avec X=0 ou X=1

La clé primaire de la table à la cardinalité (X,N) devient une **Clé étrangère** dans la table à la cardinalité (X,1)

**Exemple :** « Un employé a une et une seule société. Une société a 1 ou N employés »



MLD

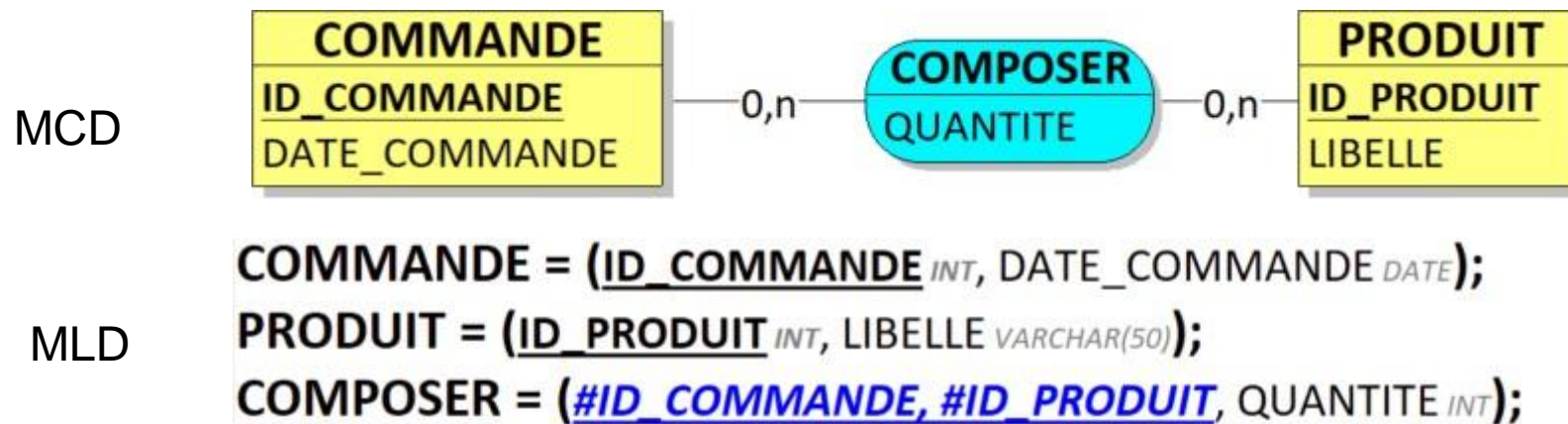
```
SOCIETE = (ID_SOCIETE INT, NOM_SOCIETE VARCHAR(50));
EMPLOYE = (ID_EMPLOYE INT, NOM_EMPLOYE VARCHAR(50), #ID_SOCIETE);
```

# 3ème règle de transformation

## Relation binaire aux cardinalités (X,N)-(X,N), avec X=0 ou X=1

- ⇒ Création d'une table supplémentaire ayant comme Clé Primaire une clé (Concaténation des deux clés primaires des deux tables en relations)
- ⇒ Si la relation est porteuse de donnée, celles-ci deviennent des attributs pour la nouvelle table.

**Exemple :** « Une commande est composée de 1 ou N produits distincts en certaine quantité. Un produit est présent dans 0 ou N commandes en certaine quantité. »



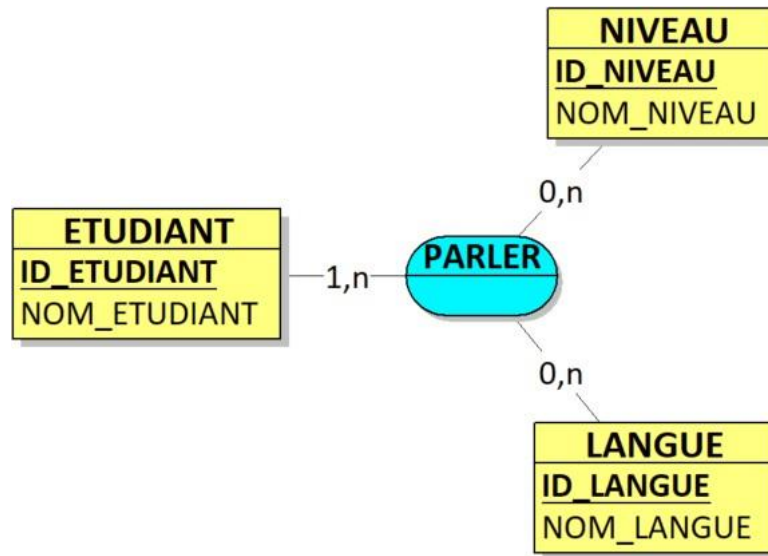
## 4ème règle de transformation

### Relation n-aire (quelles que soient les cardinalités)

Création d'une table supplémentaire ayant comme clé primaire la concaténation des identifiants des entités participants à la relation.

**Exemple :** « Un étudiant parle une ou plusieurs langues avec un certain niveau. Chaque langue est parlée par 0 ou n étudiants avec un niveau. Pour chaque niveau, il y a 0 ou plusieurs étudiants qui parlent une langue. »

MCD



MLD

```
ETUDIANT = (ID_ETUDIANT INT, NOM_ETUDIANT VARCHAR(50));
NIVEAU = (ID_NIVEAU INT, NOM_NIVEAU VARCHAR(50));
LANGUE = (ID_LANGUE VARCHAR(50), NOM_LANGUE VARCHAR(50));
PARLER = (#ID_ETUDIANT, #ID_NIVEAU, #ID_LANGUE);
```

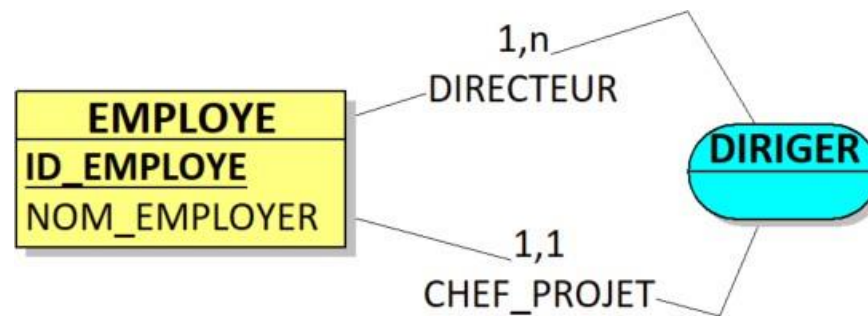
# 5ème règle de transformation

## Association Réflexive

**Premier cas** : cardinalité  $(X,1) - (X,N)$ , avec  $X=0$  ou  $X=1$

La clé primaire de l'entité se dédouble et devient une clé étrangère dans la relation ou une nouvelle table. Comme si l'entité se dédoublait et était reliée par une relation binaire  $(X,1)-(X,N)$

MCD



MLD

**EMPLOYE** = (ID\_EMPLOYE INT, NOM\_EMPLOYER VARCHAR(50), #ID\_EMPLOYE\_DIRECTEUR);

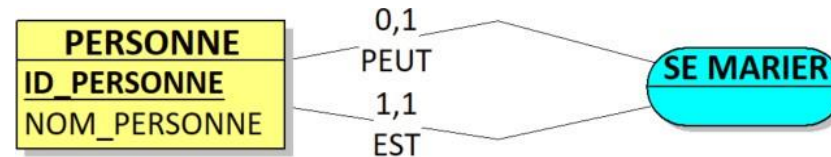
# 5ème règle de transformation

## Association Réflexive

Deuxième cas : cardinalité (0,1) – (1,1)

La clé primaire de la table à la cardinalité (0,1) devient une clé étrangère dans la table à la cardinalité (1,1)

MCD



MLD

```
PERSONNE = (ID_PERSONNE INT, NOM_PERSONNE VARCHAR(50), #ID_PERSONNE_PEUT);
```

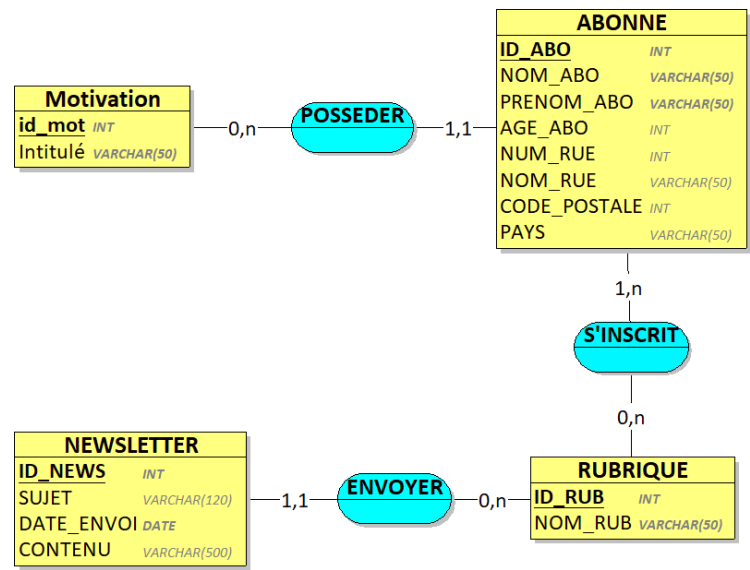
# Le modèle relationnel en bref !



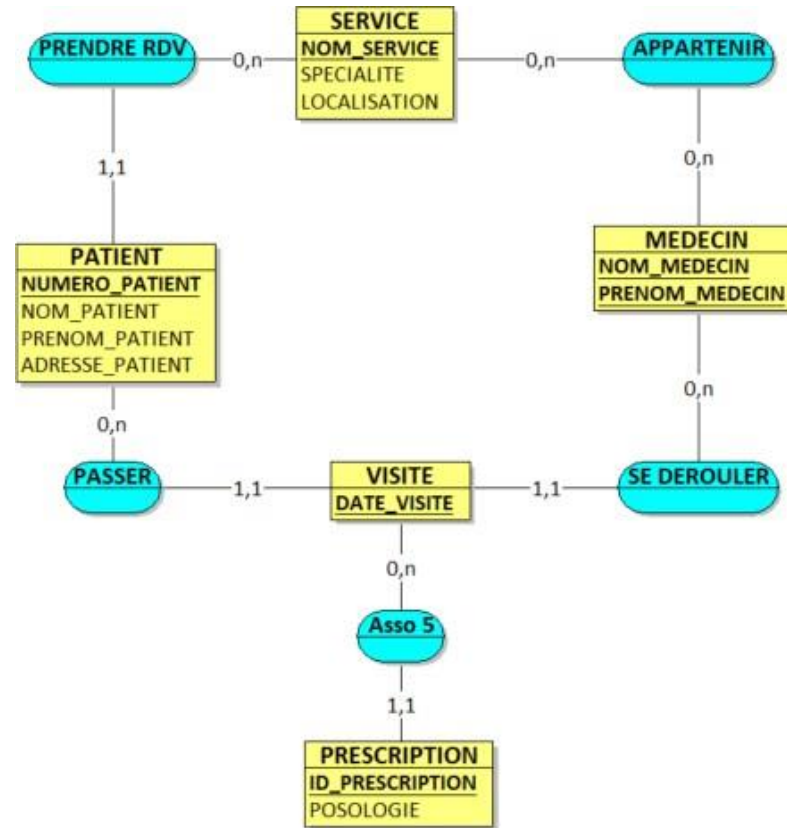
## Avantages

- Modèle théorique sous la forme d'une algèbre
- Modèle abstrait : pas de détails techniques
- Structure de données simple (voire simpliste) : tabulaire
- Compréhensible par l'utilisateur final
- Nombre minimal de concepts
- Interprétation sans ambiguïté (interprétation mathématique)
- Pourrait même faire l'objet d'une implémentation !
- Grand succès scientifique mais indifférence des milieux industriels !

# Application : Newsletter

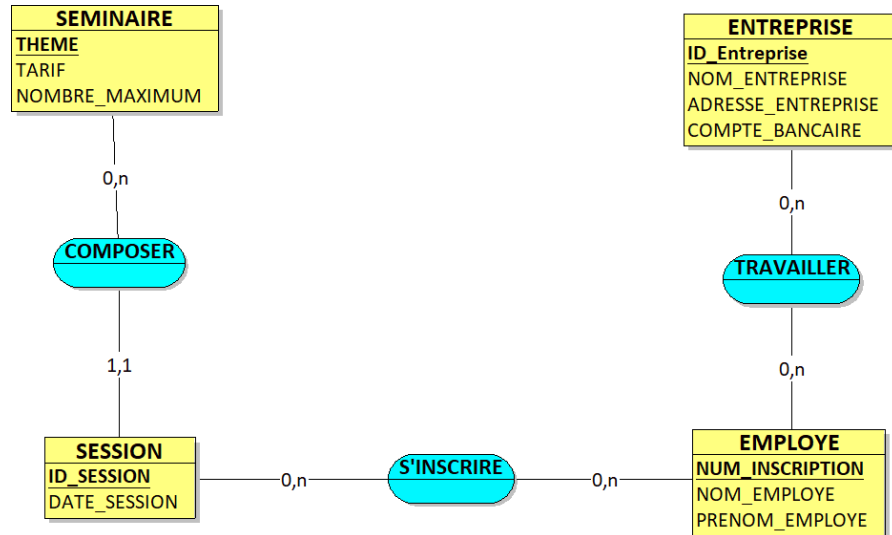


# Application : Gestion d'un hôpital





# Application : Société de formation



# Application : Gestion commerciale

Dans le cadre d'une application de gestion commerciale, les données sont gérées dans une Base de Données dont le schéma relationnel est le suivant :

**CLIENT** ( mat\_client, nom\_cl, ad\_cl )

**COMMANDE** ( num\_com, date\_com, #mat\_client)

**LIGNE\_COM** ( # num\_com, #num\_prod, qte\_com )

**PRODUIT** ( num\_prod, nom\_produit, prix\_unit, qte\_stock )

Déduire le MCD correspondant au modèle relationnel donné.

# Normalisation

# Normalisation

**Problème :** Mélanger dans une même relation des informations relatives à plusieurs entités entraîne, en général des redondances d'information qui provoquent les anomalies suivantes :

- anomalies d'insertion
- anomalies de mise à jour
- anomalies de suppression

**Principe :** La théorie de la normalisation est une théorie destinée à concevoir un bon schéma d'une base de données sans redondance d'information et sans risques d'anomalie de mise à jour. Elle a été introduite dès l'origine dans le modèle relationnel.

# Normalisation : formes normales

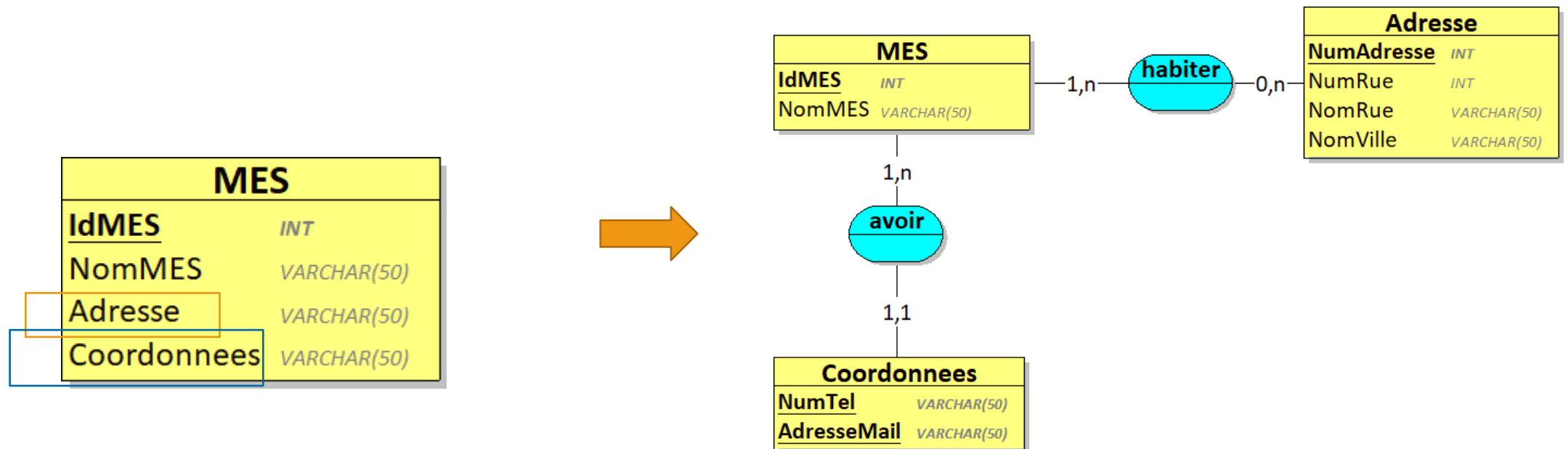
Les formes normales permettent d'éviter la redondance et les sources d'anomalies

Il existe plusieurs formes normales :

- ✓ 1FN : 1<sup>ère</sup> forme normale
- ✓ 2FN : 2<sup>ème</sup> forme normale
- ✓ 3FN : 3<sup>ème</sup> forme normale
- ✓ FNBC : Forme normale de Boyce-Codd
- ✓ 4FN : 4<sup>ème</sup> forme normale
- ✓ 5FN : 5<sup>ème</sup> forme normale
- ✓ ...

# Normalisation : 1FN

- ❑ 1<sup>ère</sup> forme normale (1FN) : Une entité ou une association est en première forme normale si tous ses attributs sont *élémentaires*, c'est-à-dire non décomposables
- ❑ Un élément composite doit être décomposés en :
  - ❑ Attributs élémentaires (ex. coordonnées)
  - ❑ Ajoutant une nouvelle entité (ex. adresse)



# Normalisation : 2FN

- 2<sup>ème</sup> forme normale (2FN) : une entité ou une association est en deuxième forme normale si, et seulement si, elle est en 1<sup>ère</sup> forme normale et si tout attribut n'appartenant pas à l'identifiant ne dépend pas d'une partie de l'identifiant

➔ Autrement dit, l'identifiant peut être composé de plusieurs attributs mais les autres attributs de l'entité doivent être dépendant de l'identifiant en entier (et non pas une partie de cet identifiant)

$$\forall a \in \textit{identifiant} : \textit{partie identifiant} \rightarrow a$$

Seance	
<u>NumSalle</u>	INT
<u>IdCinema</u>	INT
NomCinema	VARCHAR(50)
Date_seance	DATE
Heure_debut	TIME



Seance	
<u>NumSalle</u>	INT
Date_seance	DATE
Heure_debut	TIME

1,1

appartenir

1,n

Cinema	
<u>IdCinema</u>	INT
NomCinema	VARCHAR(50)

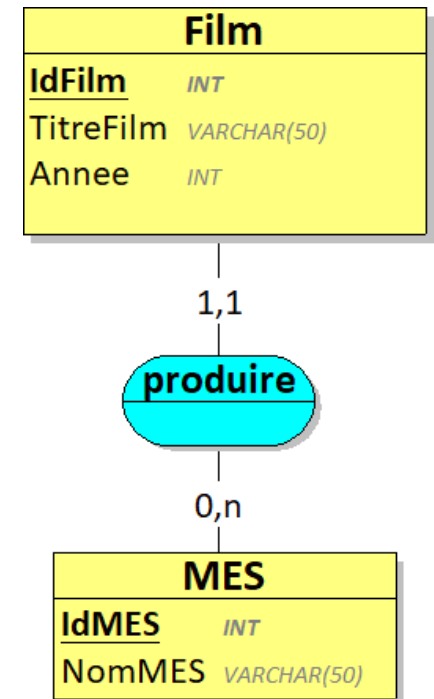
IdCinema → NomCinema

# Normalisation : 3FN

- 3<sup>ème</sup> forme normale (3FN) : une entité ou une association est en troisième forme normale si, et seulement si, elle est en 2<sup>ème</sup> forme normale et si tout attribut n'appartenant pas à l'identifiant ne dépend pas d'un autre attribut
- Garantir que seulement l'identifiant détermine tous les attributs d'une relation

$$\forall a, b \notin \textit{identifiant} : a \rightarrow b$$

Film	
<u>IdFilm</u>	INT
TitreFilm	VARCHAR(50)
Annee	INT
IdMES	INT
NomMES	VARCHAR(50)



IdMES → NomMES



# Normalisation : 4FN (FNBC)

- Forme normale de Boyce-Codd (FNBC) : les attributs d'un identifiant composé ne doivent pas dépendre d'un autre attribut de l'entité
- Une entité ou une association est en forme normale de Boyce-Codd si, et seulement si, elle est en troisième forme normale et si aucun attribut faisant partie de l'identifiant (exemple : CP) ne dépend pas d'un autre attribut ne faisant pas partie de l'identifiant (exemple : NomVille)

$\forall a \in \text{identifiant} : a \rightarrow \text{partie identifiant}$

Adresse	
<u>NumAdresse</u>	INT
<u>CP</u>	INT
NumRue	INT
NomRue	VARCHAR(50)
NomVille	VARCHAR(50)

NomVille → CP



Adresse	
<u>NumAdresse</u>	INT
<u>CP</u>	INT
NumRue	INT
NomRue	VARCHAR(50)

0,n

setrouver

0,n

Localisation	
<u>CP</u>	INT
NomVille	VARCHAR(50)