

# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales

### Application des règles

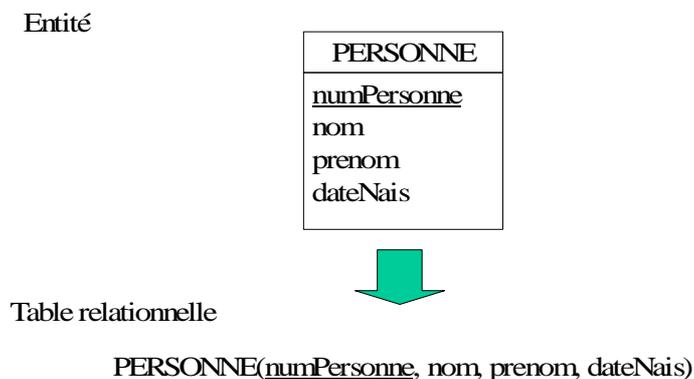
L'application des règles est "mécanique" : aucune information présente dans le MCD ne doit être perdue sous peine de produire un MLD (donc une base de données) qui ne correspondrait pas totalement aux besoins du "client".

Certaines informations ne seront pas reprises ici : les cardinalités correspondant à des valeurs littérales ou les contraintes appliquées lors de l'héritage ou appliquées aux associations. Par exemple, un étudiant peut s'inscrire à 0 à 3 formations; pour nous 3 sera pris comme 'n'. Or cette information est une règle de gestion qui devra être implantée dans la base de données : les informaticiens pourront définir des déclencheurs qui vérifieront, à chaque inscription d'un étudiant, que le nombre de fois où il est inscrit ne dépasse pas 3.

### Règle 1 : chaque entité du MCD se transforme en table du MLD

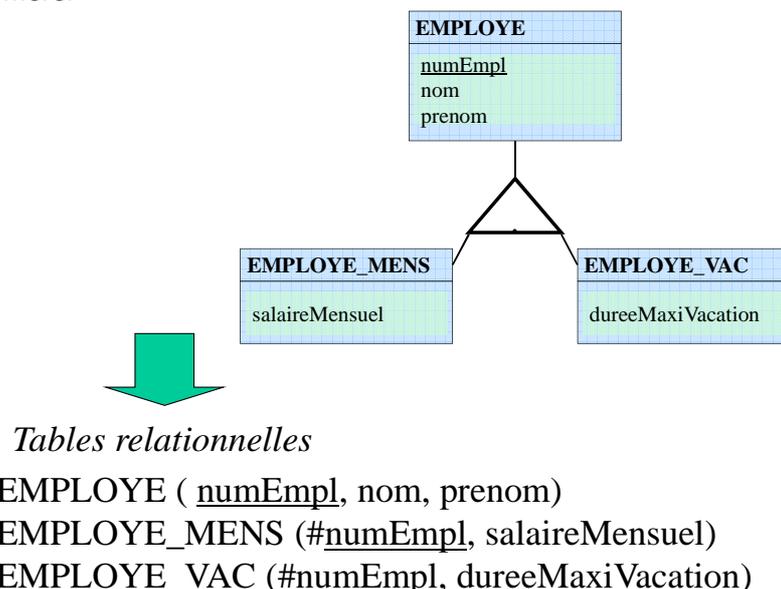
- Chaque entité se transforme en table relationnelle
- Chaque propriété se transforme en colonne de la table
- L'identifiant de l'entité devient la clef de la table

Exemple :



### Cas particulier des entités spécialisées (héritage)

Dans le cas de la spécialisation, plusieurs solutions sont possibles, mais nous retiendrons la plus cohérente : les tables relationnelles issues des entités spécialisées récupèrent l'identifiant de l'entité mère comme clef primaire et clef étrangère vers l'entité mère.

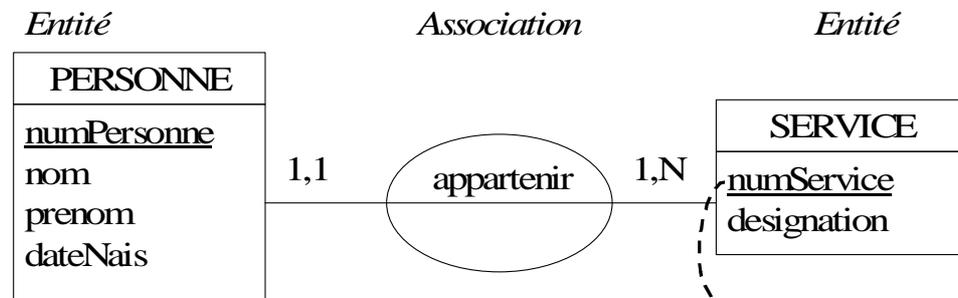


# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales

### Règle 2 : association binaires 1,1 - 1,N ou 1,1 - 0,N (0,1-1,N ou 0,1-0,N)

- Chaque association binaire non porteuse de données ayant les cardinalités se traduit par une redondance de l'identifiant de l'entité côté 1,N dans la table issue de l'entité côté 1,1
- Si l'association est porteuse de propriétés (elle ne devrait pas l'être), celles-ci se retrouvent comme colonnes dans la table issue de l'entité côté 1,1



Tables relationnelles

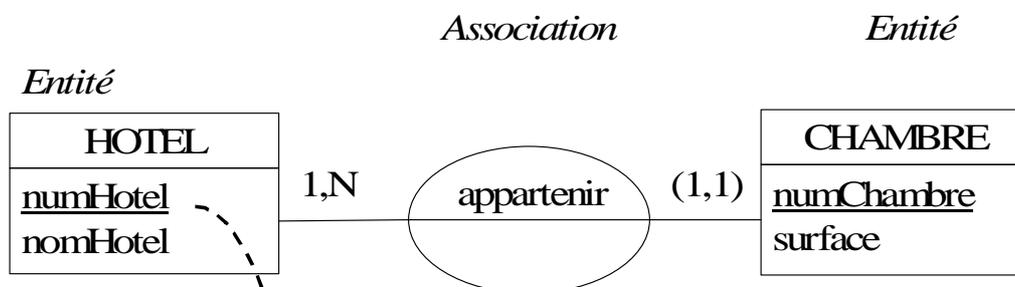
PERSONNE(numPersonne, nom, prenom, dateNais, #numService)

SERVICE(numService, designation)

Dans la table PERSONNE, :numService est « clef étrangère », et fait référence<sup>1</sup> à numService « clef primaire » dans la table SERVICE.

### Cas particulier de l'entité faible, avec un identifiant relatif

Dans le cas d'une entité faible, la clef étrangère qui a été récupérée est intégrée à la clef primaire.



Tables relationnelles

HOTEL(numHotel, nomHotel)

CHAMBRE(#numHotel, numChambre, surface)

Dans la table CHAMBRE, numHotel fait partie de la clef primaire composée de numHotel et numChambre, et est « clef étrangère » faisant référence à numHotel, « clef primaire » de la table HOTEL

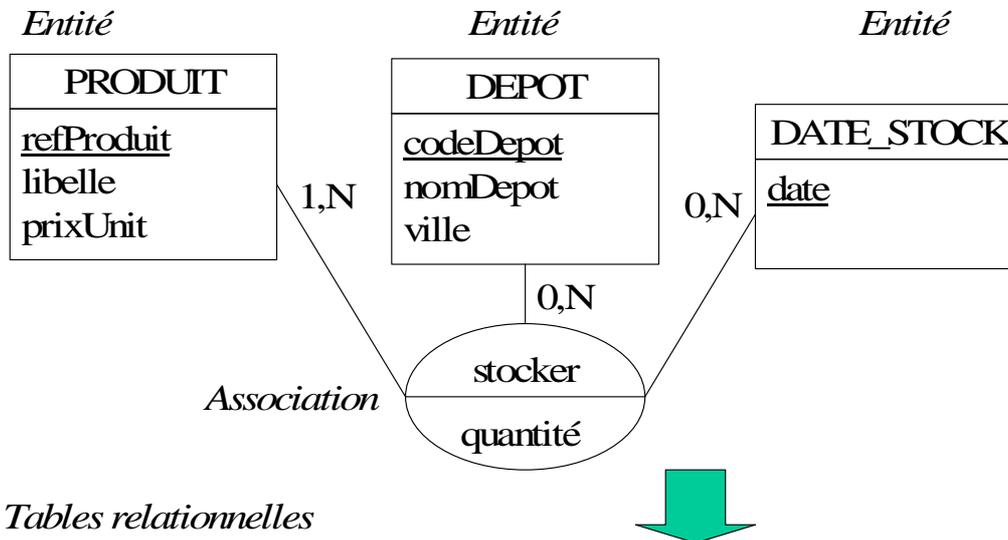
<sup>1</sup> on parle du maintien de l'intégrité "référentielle"

# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales

### Règle 3 : association n-aires : 1,N - 1,N ou 1,N - 0,N (0,N-1,N ou 0,N-0,N)

- Une association N-aire se transforme en une nouvelle table ayant comme clef primaire les identifiants des entités participant à l'association
- Si l'association possède des propriétés, elles deviennent des attributs de la table.



*Tables relationnelles*

PRODUIT(refProduit, libelle, prixUnit)

DEPOT(codeDepot, nomDepot, ville)

DATE\_STOCK(date)

STOCKER(#refProduit, #codeDepot, #date, quantité)

Dans la table STOCKER, :

- refProduit est « clef étrangère » et fait référence à refProduit « clef primaire » de la table PRODUIT
- codeDepot est « clef étrangère » et fait référence à codeDepot « clef primaire » de la table DEPOT
- date est « clef étrangère » et fait référence à date « clef primaire » de la table DATE\_STOCK

RAPPEL : une association n-aire devra toujours être vérifiée

# Analyse SI – FICHE 07 suite Modèle Logique de Données

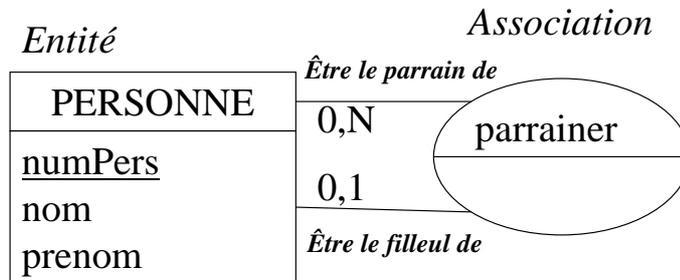
## Règles de passage MCD → MLD, Formes normales

### Cas particuliers

Ils sont généralement traités par les règles 1 à 3, mais doivent faire l'objet de légères adaptations.

#### Association réflexive

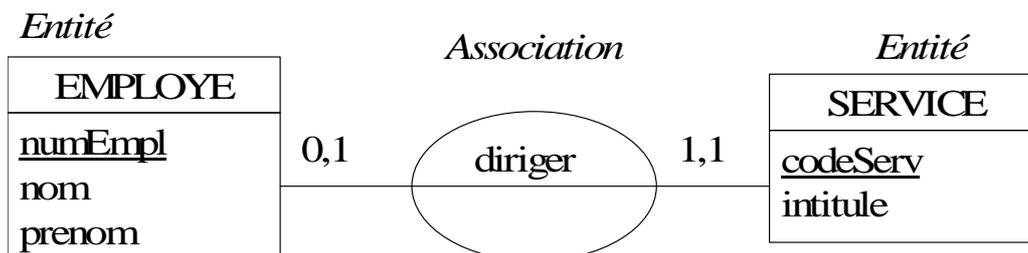
Une association réflexive se traduit selon l'application des règles 2 ou 3, mais nécessite la modification des noms de colonnes afin d'éviter les noms identiques (de plus, cela aide à la compréhension des liens).



PERSONNE (numPers, nom, prenom, #numPers\_parrain)

#### Association aux cardinalités : 0,1 - 1,1

Dans le cas association ayant des cardinalités 1,1 - 0,1, on introduit une redondance de l'identifiant côté 0,1 comme clé étrangère dans la table côté 1,1



*Tables relationnelles*

EMPLOYE (numEmpl, nom, prenom)

SERVICE (codeServ, intitule, #numEmpl\_dirigeant)

# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales

### Modèle relationnel et formes normales

La structure d'une table relationnelle doit obéir à un certain nombre de règles afin d'éviter :

- La redondance d'informations (répétition d'une même donnée)
- Les risques d'incohérence lors de modifications (insertion de lignes, modifications de colonnes et suppression de lignes)

les formes normales (fn) constituent un ensemble de règles auxquelles doivent obéir les tables relationnelles afin de construire un schéma de relations exempt d'anomalies. Elles ont été conçues pour palier à des problèmes de modélisation conceptuelle; mais l'application des règles de construction du MCD permet de garantir les formes normales jusqu'à la 3ème incluse.

### Validité initiale : dépendances fonctionnelles

les valeurs des différents attributs d'une relation doivent être en dépendance fonctionnelle de la clé primaire.

Il existe une dépendance fonctionnelle entre un ensemble d'attributs X et un ensemble d'attributs Y, que l'on note  $X \rightarrow Y$ , si connaissant une occurrence de X, on ne peut lui associer qu'une seule occurrence de Y (Y est en dépendance fonctionnelle de X, X est source de la DF, Y est cible de la DF).

- DF élémentaire : Une dépendance fonctionnelle  $X \rightarrow Y$  est élémentaire si tous les éléments qui composent X sont utiles à la détermination de Y.
- DF directe ! : • DF directe : la dépendance entre X et Y ne peut pas être obtenue par transitivité

### 1NF – élémentarité des attributs

une table est dite en première forme normale si :

tous ses attributs ont des valeurs simples (atomiques, non multiples, non composées, non multi-valués)

Exemples :

#### *Table ECRIRE*

##### **ECRIRE**

<u>numLivre</u>	Auteurs
1	Alain Collongues, Jean Hugues, Bernard Laroche
2	Irène Charon
3	Christophe Dabancourt
4	Philippe Chaleat, Danier Charnay, Jean-rené Rouet

La table ECRIRE n'est pas en première forme normale : l'attribut Auteurs contient 1 à plusieurs valeurs .

Autres exemples de relations qui ne sont pas en 1FN :

- Personne (no\_secu, nom, prenom, prenom\_enfants, age\_enfants)
- Etudiant (no\_etudiant, nom, prenom, diplomes)
- Salarie (matr, nom, prenom, adresse) : l'adresse peut être décomposée

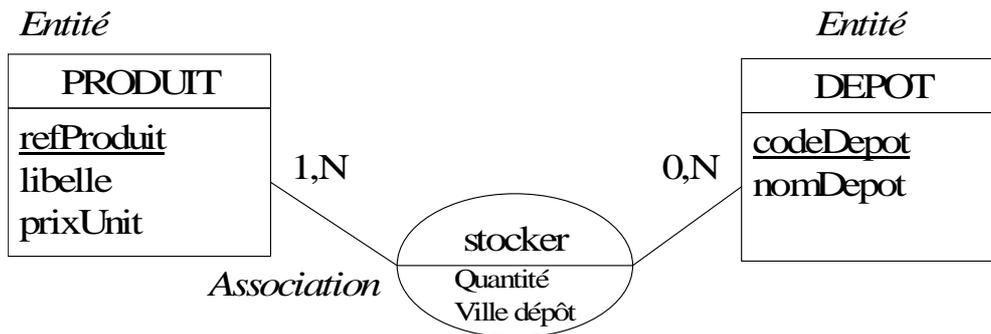
### 2NF – dépendance pleine de la clef

une table est en deuxième forme normale si :

- elle est en première forme normale
- tout attribut non-clef est en dépendance fonctionnelle de toute la clef

# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales



L'application des règles de transformation donne la table STOCKER :

STOCKER (#refProduit, #codeDepot, quantite, ville)

Les lignes de la tables STOCKER s'avèrent être les suivantes :

**Table STOCKER**

<u>#refProduit</u>	<u>#codeDepot</u>	quantite	ville
100	A	12	ARRAS
100	B	48	LILLE
350	A	50	ARRAS
420	A	76	ARRAS

La table STOCKER n'est pas en deuxième forme normale : en effet l'attribut « ville » n'est pas en dépendance fonctionnelle de toute la clef, mais seulement de « codeDepot ».

Exemple 2 :

Soit le schéma de relations :

CATALOGUE (ref\_prod, num\_fourn, libelle\_prod, nom\_fourn, adr\_fourn, prix\_achat)

Avec les dépendances :

- Num\_fourn → nom\_fourn, adr\_fourn
- Num\_prod → libelle\_prod
- ref\_prod + num\_fourn → prix\_achat

Cette relation n'est pas en FN2, la decomposition nous donne :

FOURNISSEUR (num\_fourn, nom\_fourn, adr\_fourn)

CATALOGUE (ref\_prod, libelle\_prod,)

FOURNIR (ref\_prod, num\_fourn, prix\_achat)

•

### 3NF – pas de dépendances transitives

une table est en troisième forme normale si :

- elle est en deuxième forme normale
- tout attribut non-clef n'est en dépendance fonctionnelle directe que de la clef et pas d'un attribut qui ne fait pas partie de la clef.

# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales

*Entité*

COMMERCIAL
<u>numCommercial</u>
nom
codeAgence
pays

L'application des règles de transformation à produit la table relationnelle suivante :

COMMERCIAL (numCommercial, nom, codeAgence, pays)

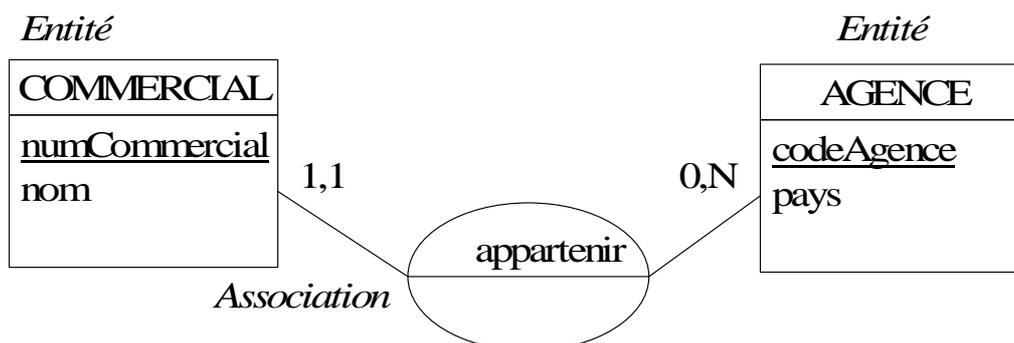
En termes de lignes, nous pouvons avoir :

**COMMERCIAL**

<u>numCommercial</u>	nom	codeAgence	pays
1	Dupont	Paris	FR
2	Durant	Londres	GB
3	Lebon	Copenhague	DK
4	Martin	Stockholm	S

La table COMMERCIAL n'est pas en troisième forme normale : en effet l'attribut « pays » est en dépendance fonctionnelle de « codeAgence ».

La modélisation correcte aurait dû être la suivante :



et ainsi les tables produites :

COMMERCIAL (numCommercial, nom, #codeAgence)

AGENCE (codeAgence, pays)

Exemple 2 :

PROFESSEUR (num\_prof, num\_etablissement)

ETUDIANT( num\_etu, num\_etablissement, num\_prof)

Avec les dépendances fonctionnelles :

# Analyse SI – FICHE 07 suite Modèle Logique de Données

## Règles de passage MCD → MLD, Formes normales

- Num\_prof → num\_etablissement
- Num\_etud → num\_etablissement, num\_prof

Cette relation n'est pas en FN3, le schéma de relation en FN3 est :

PROFESSEUR (num\_prof, num\_etablissement)  
 ETUDIANT( num\_etu, num\_etablissement, num\_prof)

### BCNF (Boyce-Codd Normal Form)- tout attribut non-clef n'est pas source d'une DF

une table est en forme normale de boyce-code si :

- elle est en troisieme forme normale
- tout attribut non-clef n'est pas source d'une dependance fonctionnelle

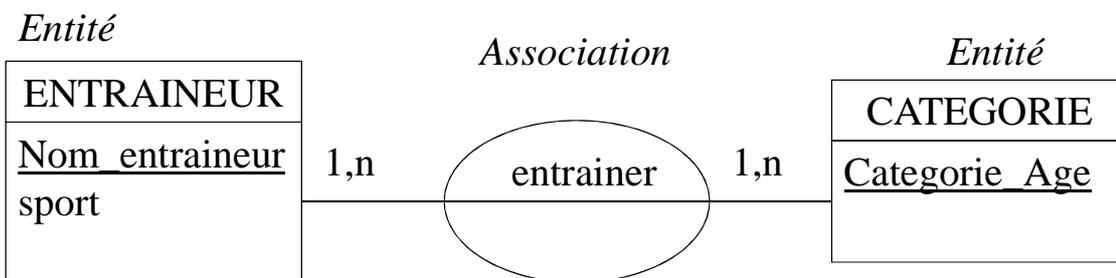
EQUIPE	Sport	categorie	Nom_entraineur
	football	junior	M. Boule
	Football	Cadet	M. Bill
	Basket	Minime	M. Paul
	Football	Minime	M. Boule
	Football	Benjamin	M. Boule

EQUIPE
<u>Sport</u>
<u>Categorie Age</u>
Nom_entraineur

M. Boule est plusieurs fois repris comme entraineur de football. Chaque entraineur ne s'occupe que d'un seul sport. IL existe une dépendance fonctionnelle entre le nom d'entraineur et une partie de la clef, le sport.

Le résultat de la modélisation aurait dû être :



ENTRAINEUR (nom\_entraineur, sport)

CATEGORIE(categorie, age)

ENTRAINEUR (#nom\_entraineur, #categorie, age)

### Retro-conception (reverse engineering)

La retro-conception consiste ici à produire le MCD correspondant à un MLD (ou une base de données).

Elle est utilisée pour faire évoluer le Système d'Information : à partir d'un MCD produit à partir d'un MLD, il sera possible de détecter des problèmes de modélisation et faire évoluer l'organisation des données et produire une nouvelle base de donnée.