

GL

# Ch 3 – Le Modèle Logique de Données relationnel

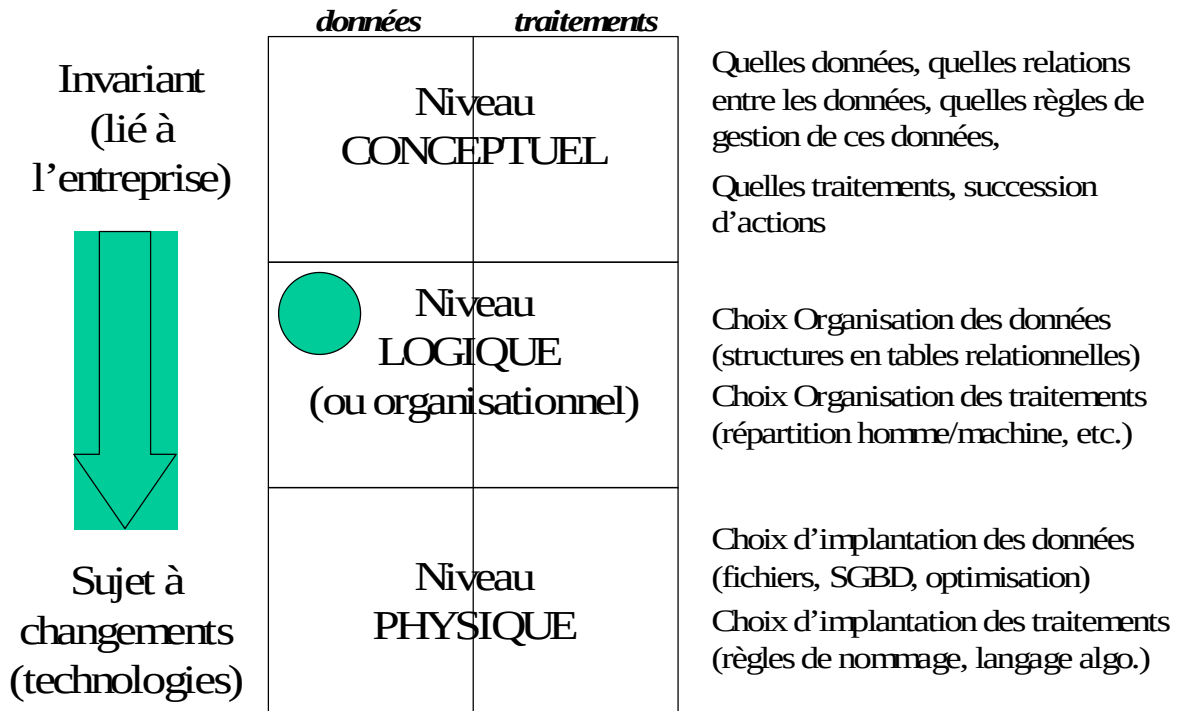
## MLD R

|             |                                                                       |           |
|-------------|-----------------------------------------------------------------------|-----------|
| <b>I.</b>   | <b>INTRODUCTION.....</b>                                              | <b>2</b>  |
| A.          | POSITIONNEMENT DU MODELE RELATIONNEL DANS L'ANALYSE DES DONNEES ..... | 2         |
| B.          | ORIGINE DU MODELE RELATIONNEL.....                                    | 2         |
| <b>II.</b>  | <b>ELEMENTS DU MODELE RELATIONNEL.....</b>                            | <b>3</b>  |
| A.          | ELEMENTS DE BASE DU MODELE RELATIONNEL : TABLE, LIGNE, COLONNE.....   | 3         |
| B.          | NOTION DE CLEF .....                                                  | 4         |
| C.          | REPRESENTATION.....                                                   | 5         |
| <b>III.</b> | <b>REGLES DE PASSAGE DU MCD AU MLDR.....</b>                          | <b>5</b>  |
| A.          | REGLE 1 : CHAQUE ENTITE SE TRANSFORME EN TABLE.....                   | 5         |
| B.          | REGLE 2 : ASSOCIATION BINAIRES 1,1 - 1,N OU 1,1 - 0,N .....           | 6         |
| C.          | REGLE 3 : ASSOCIATIONS N-AIRES .....                                  | 7         |
| D.          | REGLE 4 : ENTITE FAIBLE .....                                         | 7         |
| E.          | CAS PARTICULIERS .....                                                | 8         |
| 1.          | Association 0,1-1,N.....                                              | 8         |
| 2.          | Association réflexive.....                                            | 9         |
| 3.          | Association 0,1 – 1,1.....                                            | 9         |
| 4.          | Entités spécialisées .....                                            | 10        |
| 5.          | Association avec contraintes sur la relations .....                   | 11        |
| F.          | PRECISIONS .....                                                      | 11        |
| <b>IV.</b>  | <b>DECOMPOSITION DE SCHEMA DE RELATION.....</b>                       | <b>11</b> |
| <b>V.</b>   | <b>MODELE RELATIONNEL ET FORMES NORMALES .....</b>                    | <b>12</b> |
| A.          | VALIDITE INITIALE : DEPENDANCES FONCTIONNELLES.....                   | 12        |
| 1.          | DF élémentaire .....                                                  | 12        |
| 2.          | DF directe.....                                                       | 12        |
| B.          | 1NF – ELEMENTARITE DES ATTRIBUTS .....                                | 12        |
| C.          | 2NF – DEPENDANCE PLEINE DE LA CLEF.....                               | 13        |
| D.          | 3NF – PAS DE DEPENDANCES TRANSITIVES .....                            | 14        |
| E.          | BCNF – ATTRIBUT NON-CLEF NON SOURCE DF.....                           | 16        |

## I. Introduction

### A. Positionnement du Modèle Relationnel<sup>1</sup> dans l'analyse des données

EXPRESSION D'UN BESOIN : abstraction



PRODUIT LOGICIEL : concrétisation

Figure 1 : découpage en niveaux de la démarche d'analyse

### B. Origine du Modèle Relationnel

« Dr. Edgar E Codd joined IBM in 1949 as a programming mathematician for the Selective Sequence Electronic Calculator. During the 1950s he participated in developing several important IBM products. Beginning in 1968, Dr. Codd turned his attention to the management of large commercial databases and developed the relational model as a foundation. Since the mid-1970s, Dr. Codd has been working persistently to encourage vendors to develop relational DBMSs and to educate users, DBMS vendors, and standards committees regarding the services such a DBMS should supply and why users need all these services. »...

Extrait de E.F.Codd – The relational Model for Database Management - Version 2

<sup>1</sup> Le modèle logique de données relationnel n'est pas le seul modèle d'organisation des données : d'autres modèles l'ont précédé : modèle hiérarchique, modèle réseau (Codasyl). Mais la majorité des bases de données actuelles s'appuient sur le modèle relationnel.

EF Codd est l'auteur de la théorie du modèle relationnel, dans les années 1970.

Le modèle relationnel est basé sur les concepts mathématiques de la théorie des ensembles et s'appuie sur les opérateurs de l'algèbre relationnelle pour effectuer toutes sortes de requêtes sur un schéma de relations (projection, sélection, produit cartésien, jointures, union, intersection, différence).

Le modèle relationnel présente une structure très simple mais s'appuie sur des bases théoriques solides, ce qui en a fait une référence dans le domaine des systèmes de gestion de bases de données.

Le modèle relationnel est à l'origine

- des bases de données relationnelles
- du langage SQL qui permet l'interrogation et la manipulation des données d'une BD relationnelle.

## II. Eléments du Modèle Relationnel

---

Le **MODELE RELATIONNEL de DONNEES, MRD** (Modèle Logique de Données Relationnel, MLD<sub>r</sub> pour Merise) est une représentation de l'**ORGANISATION DES DONNEES** tenant compte d'une technologie, ici la technologie relationnelle ;

Les données y sont organisées en

- **RELATIONS** (ou **TABLES RELATIONNELLES** ou simplement tables), celles-ci étant décrites en terme de d'**ATTRIBUTS de la relation** (ou **COLONNES**) et
- **TUPLES** ou **n-uplets** (ou **LIGNES**)

L'établissement du modèle logique fournit un résultat **INVARIANT (élément stable)** face

- Aux **logiciels** qui seront développés
- Au **choix d'implantation physique des données** (un système de gestion de bases de données précis)

### A. Eléments de base du modèle relationnel : TABLE, LIGNE, COLONNE

L'élément de base est la **TABLE RELATIONNELLE** (ou **RELATION**). Une table représente les différentes occurrences d'une « relation-type » (cf Modèle relationnel – T Codd).

Une **LIGNE** d'une table est une occurrence de la « relation-type ».

Une **COLONNE** de la table contient l'ensemble des valeurs prises par un attribut de la relation pour l'ensemble des occurrences de cette relation.

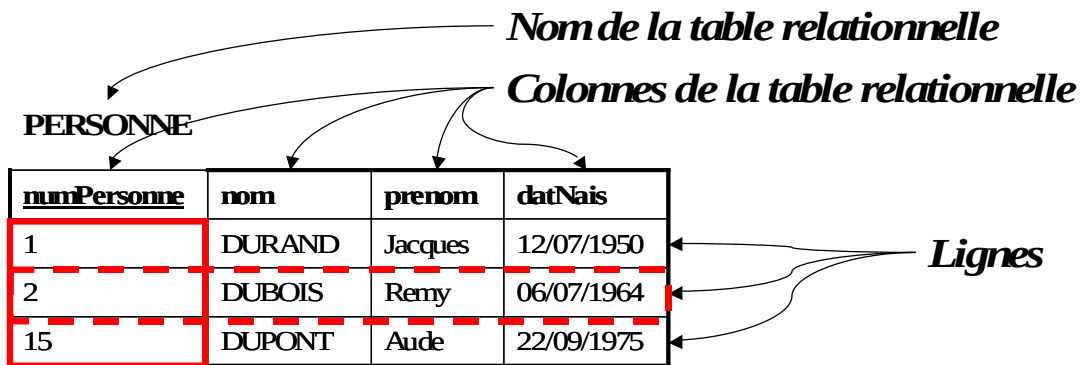


Figure 2 : représentation des données d'une table

## B. Notion de clef

La notion de clef permet de maintenir l'intégrité des valeurs dans les relations.

Plusieurs types des clefs sont définis :

- **CLEF PRIMAIRE** (*anglais : Primary Key*): c'est la **clef qui assure l'unicité d'un tuple** d'une relation ; tous les autres attributs d'une relation sont en dépendance unique de la clef primaire
  - o la clef peut être simple ou composée ;
  - o on la représente généralement soulignée « numEtudiant »
- **CLEF ETRANGERE** (*anglais : Foreign Key*): un attribut clef étrangère dans une relation est clef primaire dans une autre table ;
  - o elle **matérialise un lien logique** entre ces 2 relation ;
  - o on la représente généralement préfixée d'un « # » : « #classe »
- **CLE CANDIDATE** : c'est une colonne qui n'est pas clef primaire, mais qui aurait pu l'être et dont on souhaite, parfois, garantir l'unicité des valeurs (un code INSEE, une adresse mail);
  - o On la représente généralement soulignée en tiret : « numINSEE » ;

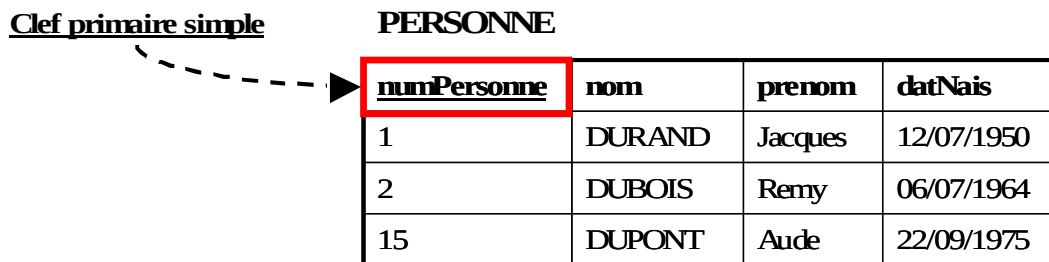


Figure 3 : clef primaire simple, composée d'une seule attributs

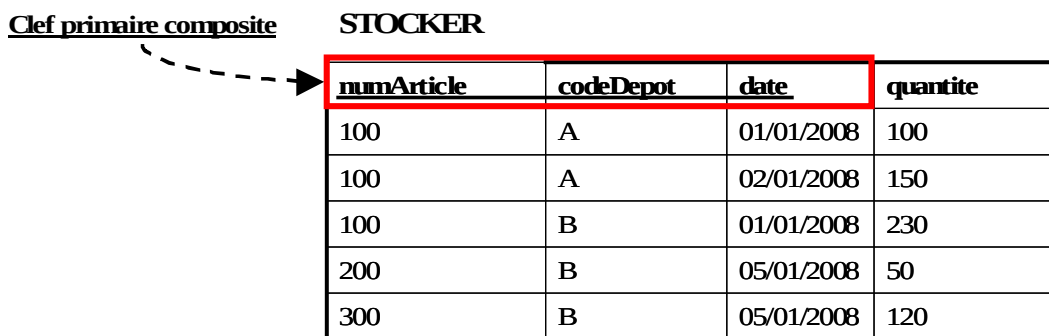


Figure 4 : clef primaire composite, composée de plusieurs attributs

## C. Représentation

La représentation du MRD indique le nom de la table suivi des colonnes entre parenthèses, la clef est placée en premier et est soulignée.

ETUDIANT (numEtudiant, nom, prenom, #classe, numINSEE)

La table relationnelle ETUDIANT comporte 5 attributs ou colonnes :

- numEtudiant : clef primaire,
- nom et prenom : attributs de la relation,
- #classe : clef étrangère (fait référence à la clef primaire d'une autre table),
- numInsee : clef candidate.

## III. Règles de passage du MCD au MLDR

---

La phase de Modélisation Conceptuelle des Données étant réalisée (modèle MCD), il va s'agir de déterminer un choix d'organisation de ces données en fonction de technologies existantes : actuellement le Modèle Relationnel prédomine, mais il n'y a pas si longtemps, le modèle CODASYL a eu son heure de gloire et rien ne dit que dans quelques années le modèle OBJET ne deviendra pas la référence.

La transformation du Modèle Conceptuel de Données, vers le Modèle Logique de Données Relationnel résulte de l'application mécanique de 4 règles principales.

### A. Règle 1 : chaque entité se transforme en table

- |                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Chaque entité se transforme en table relationnelle</li><li>• Chaque propriété se transforme en colonne de la table</li><li>• L'identifiant de l'entité devient la clef de la table</li></ul> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Entité

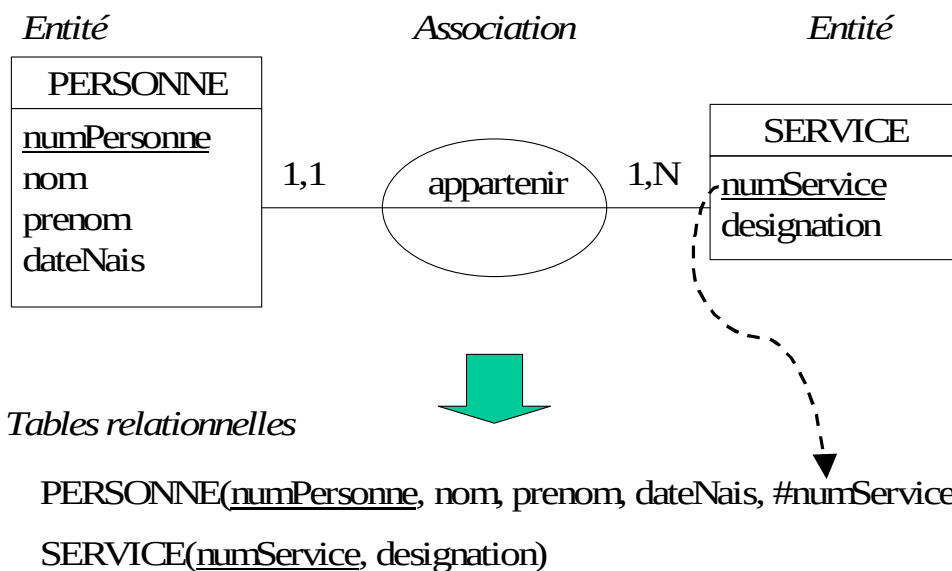


## Table relationnelle

PERSONNE(numPersonne, nom, prenom, dateNais)

## B. Règle 2 : association binaires 1,1 - 1,N ou 1,1 - 0,N

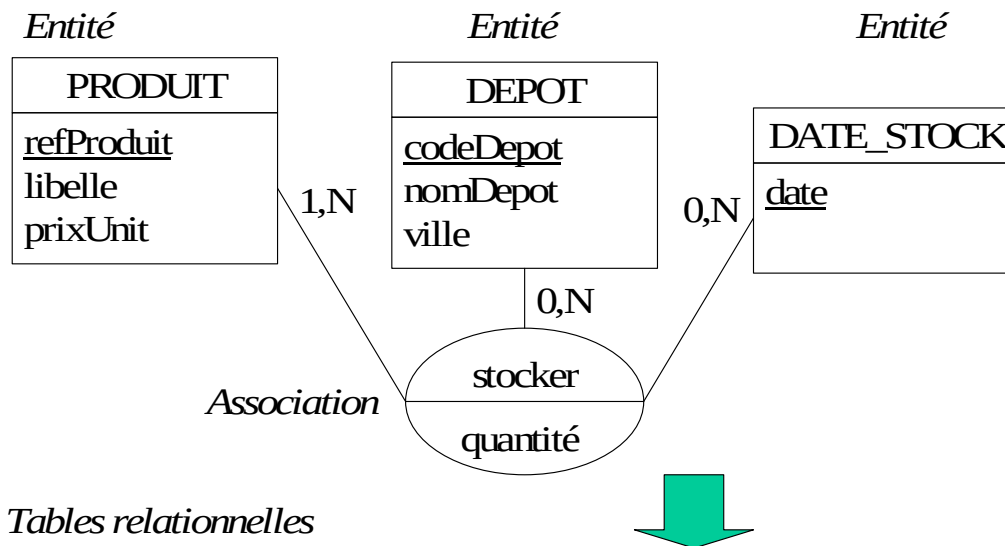
- Chaque association binaire non porteuse de données ayant les cardinalités (1,1-1,N ou 1,1 - 0,N) se traduit par une redondance de l'identifiant de l'entité ayant la cardinalité 1,N ou 0,N dans la table issue de l'entité ayant la cardinalité 1,1
- Si l'association est porteuse de propriétés (elle ne devrait pas l'être), celles-ci se retrouvent comme colonnes dans la table issue de l'entité participant avec la cardinalité 1,1



**Dans la table PERSONNE, :numService est « clef étrangère », et fait référence à numService « clef primaire » dans la table SERVICE.**

### C. Règle 3 : associations N-aires

- Une association N-aire, porteuse ou non de propriétés, ayant les cardinalités 0,N ou 1,N, se transforme en une table ayant comme clef composite les colonnes issues des identifiants des entités participant à l'association



Tables relationnelles

PRODUIT(refProduit, libelle, prixUnit)

DEPOT(codeDepot, nomDepot, ville)

DATE\_STOCK(date)

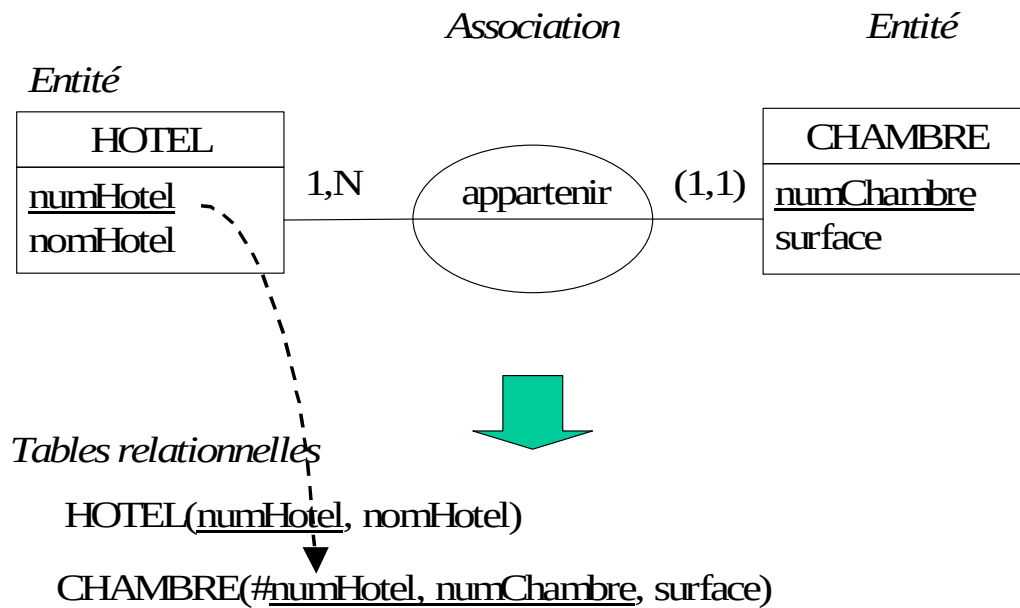
STOCKER(#refProduit, #codeDepot, #date, quantité)

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

### D. Règle 4 : entité faible

- Dans le cas d'une entité faible, la clef primaire de la table relationnelle est constituée par l'identifiant de l'entité forte et par l'identifiant de l'entité faible



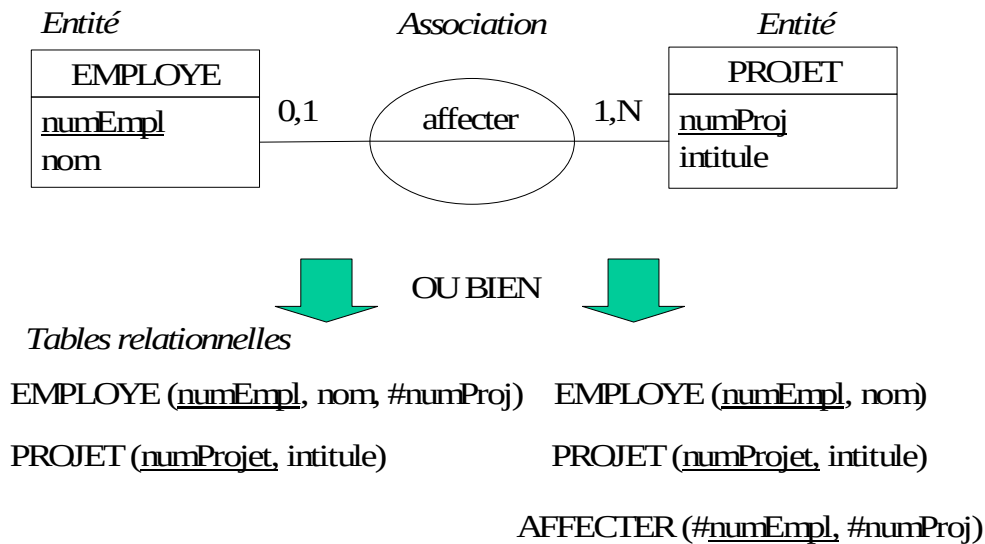
Dans la table CHAMBRE, numHotel fait partie de la clef primaire composée et est « clef étrangère » faisant référence à numHotel « clef primaire » de la table HOTEL

## E. Cas particuliers

### 1. Association 0,1-1,N

- Ces entités se décomposent selon la règle numéro 2 ou bien :
- Création d'une table intermédiaire réservée aux seules occurrences de l'entité pour lesquelles existe une occurrence de l'association



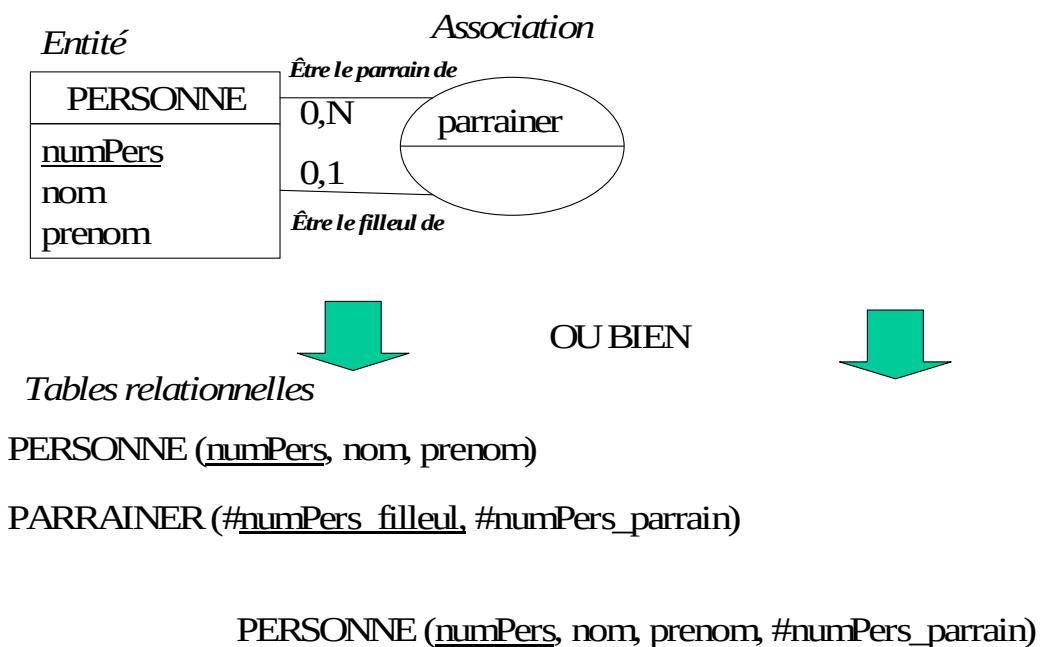


Cette décomposition spécifique évite d’avoir des attributs non renseignés (à la valeur NULL) dans le cas d’une entité ne possédant pas d’occurrence d’association.

Dans le cas de la transformation classique (à gauche), un employé peut n’être affecté à aucun projet et avoir sa valeur d’attribut « #numproj » vide.

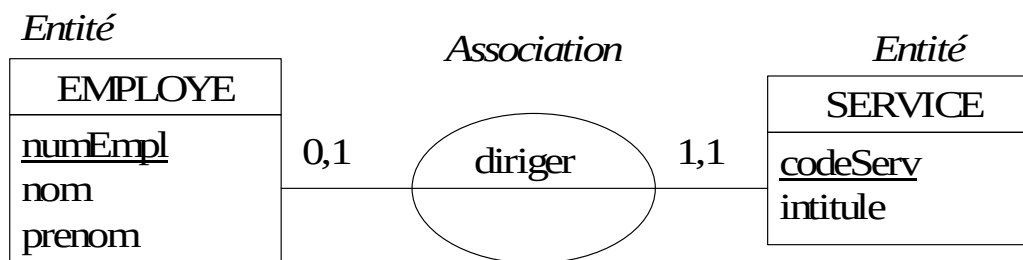
## 2. Association réflexive

- 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).



## 3. Association 0,1 – 1,1

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



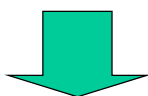
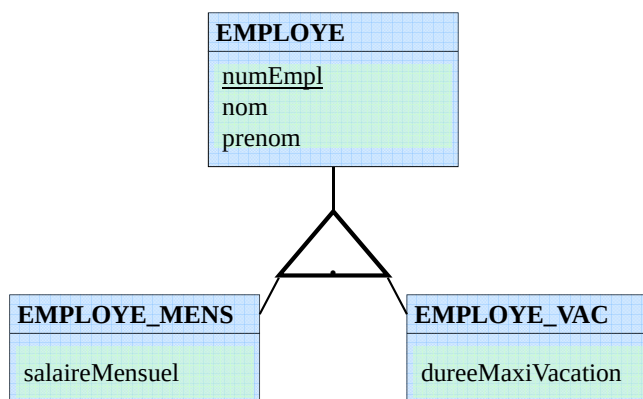
#### Tables relationnelles

EMPLOYEE (numEmpl, nom, prenom)

SERVICE (codeServ, intitule, #numEmpl\_dirigeant)

#### 4. Entités spécialisées

- Dans le cas de la spécialisation, 2 solutions sont possibles : les relations issues des entités spécialisées récupèrent l'identifiant et les propriétés de l'entité mère, ou bien elles deviennent des tables indépendantes et récupèrent l'identifiant de l'entité mère comme clef primaire.

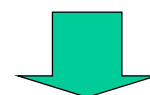


#### Tables relationnelles

EMPLOYEE ( numEmpl, nom, prenom)

EMPLOYEE\_MENS (#numEmpl, salaireMensuel)

EMPLOYEE\_VAC (#numEmpl, dureeMaxiVacation)



#### Tables relationnelles

EMPLOYEE\_MENS (numEmpl, nom, prenom ,salaireMensuel)

EMPLOYEE\_VAC (numEmpl, nom, prenom, dureeMaxiVacation)

## Analyse et modélisation

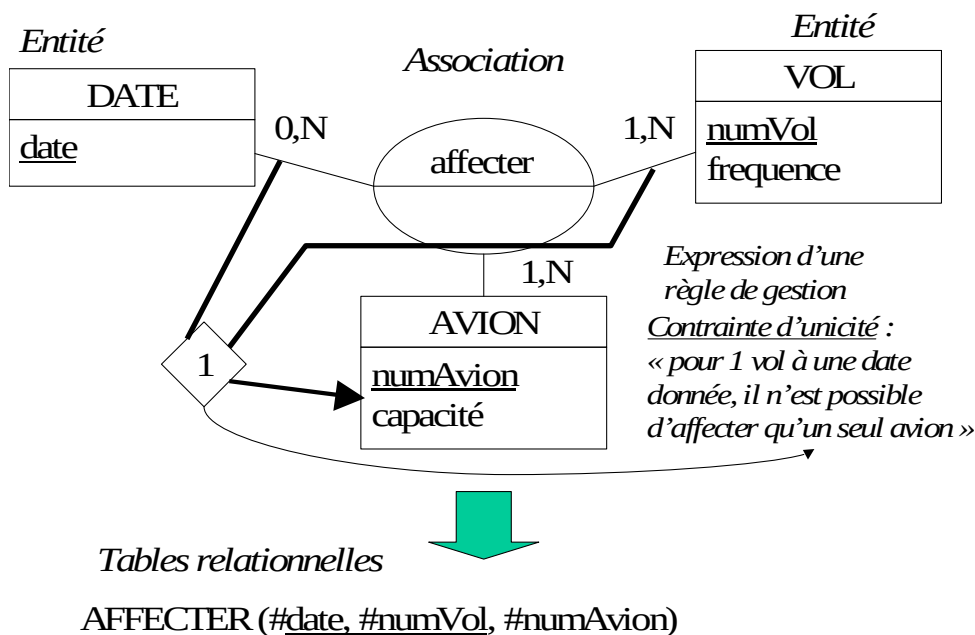
Une 3<sup>ème</sup> solution peut être envisagée : on ne conserve que l'entité générique, celle-ci récupérant les propriétés des entités spécialisées :

**EMPLOYE (numEmpl, nom, prenom, salaireMensuel, dureeMaxiVacation)**

Cette solution comporte un inconvénient majeur : les attributs « salaireMensuel » et « dureeMaxiVacation » auront une valeur nulle alternativement dans l'un ou l'autre cas de spécialisation.

### 5. Association avec contraintes sur la relations

- Si une contrainte d'unicité a été définie sur une association, on sort l'identifiant de l'entité concernée par la contrainte.



## F. Précisions

Certains ajustements sont parfois nécessaires pour obtenir un modèle relationnel plus précis. Ainsi, il est parfois judicieux de modifier légèrement des noms d'attributs afin de préciser leur sens, par exemple, dans les associations réflexives dans lesquelles un nom d'attributs sera en lien avec un rôle (= une des pattes de l'association)

## IV. Décomposition de schéma de relation

La décomposition d'un schéma de relation intervient dans le cas où la phase de modélisation conceptuelle n'a pas été réalisée ou lors d'une validation du modèle relationnel. Il est alors indispensable de décomposer les relations de telle manière à minimiser la redondance d'informations et les anomalies de mises à jour des tuples qui s'en suivent.

**Théorème de Heath :** toute relation  $R(A,B,C)$  est décomposable sans perte d'information en 2 relations  $R_1(A,B)$  et  $R_2(A,C)$  s'il existe une dépendance fonctionnelle  $A \rightarrow B$ .

Exemple :

- Relation Fournisseur (numFour, nom, adr, tel, refProd, prix)

Décomposable en :

- Relation Fournisseur (numFour, nom, adr, tel)
- Relation FournisseurProduit (numFour, refProd, prix)

## V. 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)

[http://perso.univ-lr.fr/pboursie/BD\\_Niv1/Ch8/page8-1.htm](http://perso.univ-lr.fr/pboursie/BD_Niv1/Ch8/page8-1.htm)

Les **FORMES NORMALES (FN)** constituent un ensemble de **REGLES AUXQUELLES DOIVENT OBEIR LES TABLES RELATIONNELLES AFIN DE CONSTRUIRE UN SCHEMA DE RELATIONS EXEMPT D'ANOMALIES.**

La version anglo-saxonne : Normal Form (NF)

Si le modèle relationnel est issu d'une transformation d'un MCD, les dépendances fonctionnelles ayant été vérifiées, le modèle généré doit intégrer les formes normales (jusqu'à la 3<sup>ème</sup> ou la 4<sup>ème</sup>).

Si les tables produites par cette transformation ne sont pas au moins en 3<sup>ème</sup> F.N, cela révèle **une anomalie de conception.**

### A. Validité initiale : dépendances fonctionnelles

**LES VALEURS DES DIFFERENTS ATTRIBUTS D'UNE RELATION DOIVENT ETRE EN DEPENDANCE FONCTIONNELLE DE LA CLE 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).

#### 1. DF élémentaire

Une dépendance fonctionnelle  $X \rightarrow Y$  est élémentaire si tous les éléments qui composent X sont utiles à la détermination de Y.

#### 2. DF directe

- DF directe : la dépendance entre X et Y ne peut pas être obtenue par transitivité

### B. 1NF – élémentarité des attributs

Une **TABLE** est dite en **PREMIERE FORME NORMALE** si :

- **TOUS SES ATTRIBUTS ONT DES VALEURS SIMPLES** (atomiques, non multiples, non composées, non multi-valués)

Les attributs contiennent des données élémentaires.

**Table ECRIRE****ECRIRE**

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

Figure 5 : lignes de la table ECRIRE

La table ECRIRE n'est pas en première forme normale : l'attribut Auteurs contient 1 à plusieurs valeurs<sup>2</sup>.

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**)
  - o L'adresse peut être décomposée

Remarque : il existe des extensions du modèle relationnel mettant en œuvre des attributs à valeur complexe (listes de valeurs) qui ne sont donc plus en 1FN (NF2, Non First Normal Form), sous forme d'imbrication d'attributs. Des opérateurs de l'algèbre relationnelle doivent être mis en œuvre pour en tenir compte.

**C. 2NF – dépendance pleine de la clef**

Une **TABLE** est en **DEUXIEME FORME NORMALE** si :

- ELLE EST EN **PREMIERE FORME NORMALE**
- **TOUT ATTRIBUT NON-CLEF EST EN DEPENDANCE FONCTIONNELLE DE TOUTE LA CLEF**

Exemple 1 :

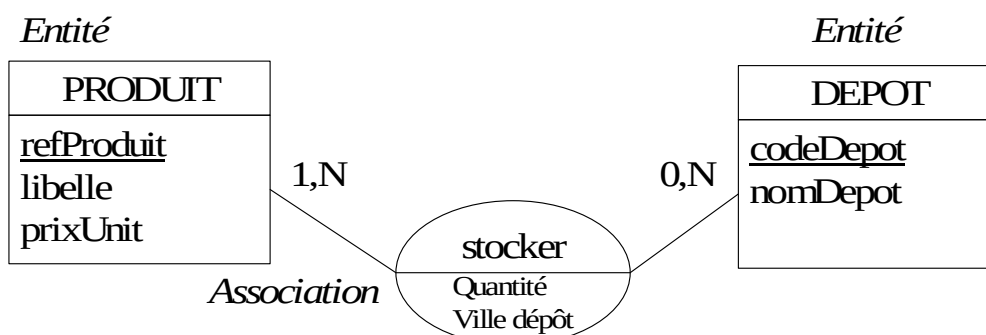


Figure 6 : Modèle Entité-Association Propriété (issu de la modélisation conceptuelle des données)

<sup>2</sup> Par contre, si chaque nom d'auteur n'est pas intéressant d'un point de vue de la recherche d'informations, on peut considérer que l'attribut auteurs est justifié.

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**  
**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 |

Figure 7 : lignes de la table STOCKER

**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)

## D. 3NF – pas de dépendances transitives

Une **TABLE** est en **TROISIEME FORME NORMALE** si :

- ELLE EST EN **DEUXIEME FORME NORMALE**
- **TOUT ATTRIBUT NON-CLEF N'EST EN DEPENDANCE FONCTIONNELLE DIRECTE QUE DE LA CLEF** et pas d'un attribut qui ne fait pas partie de la clef.

Pas de dépendances transitives

*Entité*

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

Exemple 1 :

Figure 8 : Figure 6 : l'entité COMMERCIAL résultant de la modélisation conceptuelle

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

COMMERCIAL (numCommercial, nom, codeAgence, pays)

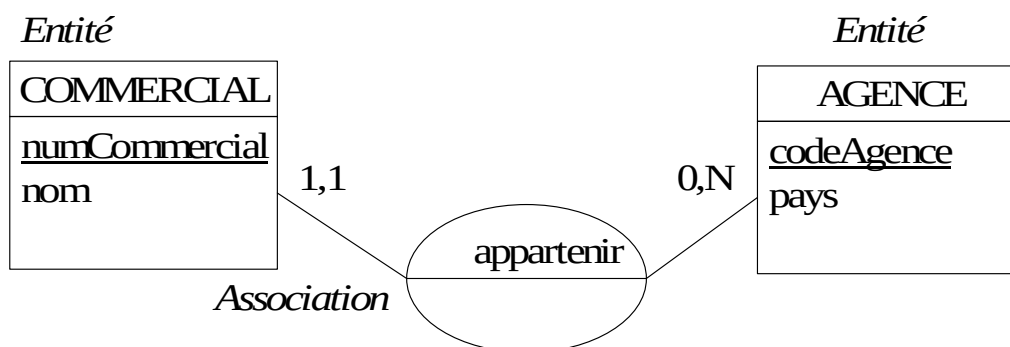
En termes de lignes, nous pouvons avoir :

**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)

## Analyse et modélisation

Avec les dépendances fonctionnelles :

- 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)

## E. BCNF – attribut non-clef non source DF

Une **TABLE** est en **FORME NORMALE DE BOYCE-CODE** si :

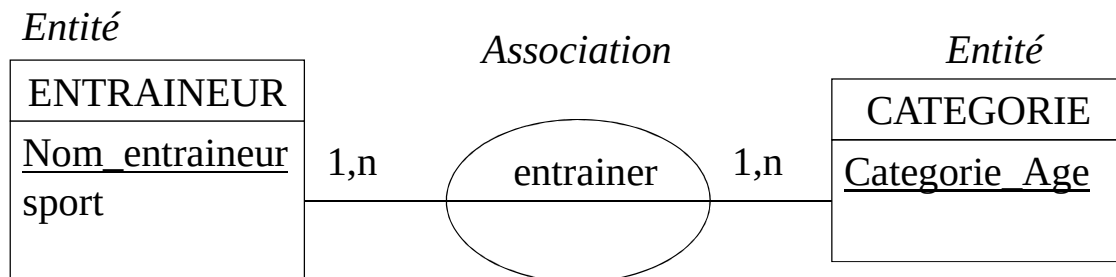
- ELLE EST EN **TROISIEME FORME NORMALE**
- **TOUT ATTRIBUT NON-CLEF N'EST PAS SOURCE D'UNE DEPENDANCE FONCTIONNELLE**

| 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</u> <u>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)

ENTRAINER (#nom\_entraineur, #categorie age)