

# **CONCEPTION D'UN SYSTÈME D'INFORMATION AVEC MERISE**

# SOMMAIRE

## Qu'est-ce que MERISE

### Base de données relationnelle : Définition

### Base de données relationnelle : Exemple

### La démarche MERISE selon trois cycles

### Les trois niveaux du cycle d'abstraction

#### Le niveau conceptuel

#### Les niveaux logique et organisationnel

#### Le niveau physique

### Le dictionnaire de données

### Les règles de gestion

### Le modèle conceptuel de données (MCD)

#### L'entité

#### Les propriétés

#### La propriété identifiant

#### Les occurrences

#### Les associations

#### Association porteuse de données

## Les différents types d'associations

### Association un à un

### Association un à plusieurs

### Association plusieurs à plusieurs

## Les cardinalités

### Les différents types de cardinalités

### Quelques exemples de cardinalités

## La représentation du temps dans un MCD

### Exemples de représentations synchronique et diachronique

### Représentation synchronique dans la base de données

### Représentation diachronique dans la base de données : 1<sup>er</sup> exemple

### Représentation diachronique dans la base de données : 2<sup>e</sup> exemple

## Les différentes étapes de l'élaboration d'un MCD

# SOMMAIRE

## Le modèle logique de données (MLD)

Un domaine

Un attribut

Une relation

Une cardinalité

Les contraintes d'intégrité

La clé primaire

Les clés étrangères

Schéma d'une relation

Schéma d'une base de données relationnelle

Les dépendances fonctionnelles : Définition

Les dépendances fonctionnelles : 1<sup>er</sup> exemple

Les dépendances fonctionnelles : 2<sup>e</sup> exemple

Les axiomes d'Armstrong

Autres propriétés

La pseudo-transitivité

L'union

La décomposition

## Les dépendances fonctionnelles élémentaires

Les dépendances fonctionnelles directes

Le graphe des dépendances fonctionnelles

Les formes normales

La 1<sup>ère</sup> forme normale

La 2<sup>e</sup> forme normale

La 3<sup>e</sup> forme normale

La 3<sup>e</sup> forme normale de Boyce-Codd

Passage du MCD au MLD

Règle N°1

Règle N°2

Règle N°3

Règle N°4

Règle N°5

Placement des attributs d'association

# SOMMAIRE

Le modèle physique de données

Passage du MLD au MPD

Etape de création d'une base de données

Passage du MPD au MLD ou Rétroconception : Définition

Règle N°1

Règle N°2

Règle N°3

Règle N°4

Règle N°5

MERISE 2

Lien relatif dans un MCD

L'héritage dans MERISE

Webographie

## QU'EST-CE QUE MERISE

**MERISE : Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprises**  
Méthode de modélisation pour la conception de systèmes d'information née dans les années 1970 et qui se base sur la séparation entre l'analyse des données et l'analyse des traitements.

### Pour les données :

- Identifier et regrouper les données de même nature.
- Identifier les relations entre ces données.

### Pour les traitements:

- Identifier les fonctionnalités qui vont permettre le traitement de ces données.
- Identifier les découpages et enchainements de ces fonctionnalités.

Cette méthode s'inscrit selon une démarche progressive comprenant trois cycles principaux :

- Le cycle de vie
- Le cycle de décision
- Le cycle d'abstraction

## BASE DE DONNEES RELATIONNELLE : Définition

Une base de données relationnelle est une entité structurée, permettant le stockage, en grande quantité, d'informations décomposées et organisées sous forme de tables reliées entre elles.

Une base de données correctement structurée permet :

- D'éviter la redondance d'informations.
- D'assurer l'intégrité des données ( c'est-à-dire la cohérence, la fiabilité et la pertinence des données).
- D'améliorer la taille de la base de données.
- De rendre les traitements performants sur la base de données (notamment le temps d'exécution des requêtes qui doit être le plus court possible).

## BASE DE DONNEES RELATIONNELLE : Exemple

N° livre	Titre du livre	Nom de l'auteur	Prénom de l'auteur
1	L'exposition coloniale	Orsena	Erik
2	Les chevaliers du subjonctif	Orsena	Erik
3	Le Horlat	Maupassant	Guy (de)
4	Les chouans	Balzac	Honoré (de)
5	Le père Goriot	Balzac	Honoré (de)
6	Eugénie Grandet	Balzac	Honoré (de)
7	Manon des sources	Pagnol	Marcel
8	Le tour du monde en 80 jours	Vernes	Jules
9	20 000 lieues sous les mers	Vernes	Jules
10	L'eau des collines	Pagnol	Marcel

Ci-dessus, nous avons représenté une table de bases de données, volontairement, mal construite. En effet, vous observez qu'il y a des redondances d'informations pour les auteurs, ce qui augmente inutilement le volume de la base de données

N° du livre	Titre du livre
1	L'exposition coloniale
2	Les chevaliers du subjonctif
3	Le Horlat
4	Les chouans
5	Le père Goriot
6	Eugénie Grandet
7	Manon des sources
8	Le tour du monde en 80 jours
9	20 000 lieues sous les mers
10	L'eau des collines

Il est préférable de scinder les enregistrements dans deux tables différentes afin d'éviter les redondances et améliorer la taille et les performances de traitements de la base de données.

N° auteur	Nom de l'auteur	Prénom de l'auteur
1	Orsena	Erik
2	Balzac	Honoré (de)
3	Maupassant	Guy (de)
4	Pagnol	Marcel
5	Vernes	Jules

## LA DEMARCHE MERISE SELON TROIS CYCLES

La méthode MERISE permet une démarche progressive, menée en même temps, dans la conception d'un système d'information selon trois cycles principaux :

### -Le cycle de vie

Grâce à un ensemble de documents précis, il est possible de maîtriser la chronologie des opérations à réaliser en terme d'objectifs fonctionnels, administratifs et financiers.

### -Le cycle de décision

Des étapes de réalisation sont mises en place afin de comprendre, vérifier et valider le bon déroulement de l'implémentation du système d'information.

### Le cycle d'abstraction

C'est au cours de ce cycle qu'une séparation est effectuée entre les données et les traitements. Ce cycle comprend trois niveaux :

- Le niveau conceptuel (Quoi? Combien? Pourquoi?)
- Le niveau logique et organisationnel (Qui? Quand? Où?)
- Le niveau physique (Comment?)



# LES TROIS NIVEAUX DU CYCLE D'ABSTRACTION

## LE NIVEAU CONCEPTUEL

Le niveau conceptuel s'attache à comprendre l'activité de l'entreprise, d'un point de vue fonctionnel, comprendre la réalité du domaine à modéliser, quels en sont les différents métiers? Quelles sont les principales tâches effectuées?

Cela permet de savoir :

- Quelles sont les données que l'on va gérer?, comment sont-elles organisées?
- Quels sont les traitements à réaliser?
- Les règles de gestion à prendre en compte

Pour ce niveau, les principaux modèles utilisés sont :

- Le modèle conceptuel de données (MCD) pour les données.
- Le modèle conceptuel des traitements (MCT) pour les traitements.

## LE NIVEAU LOGIQUE ET LE NIVEAU ORGANISATIONNEL

Le niveau logique (les données) s'attache à définir la forme que doit prendre la solution logicielle pour être adaptée à l'utilisateur et lui permettre de gérer ses données correctement. (Ex : Formulaire web, traitement de texte, tableur, logiciel de dessin,...). Il précise la vision informatique de la solution(comment) et décrit les écrans, enchainements et rapports.

Le niveau organisationnel (les traitements) s'attache à comprendre :

- Quels sont les acteurs dans l'entreprise et quels sont leur tâches? → "Qui fait quoi"
- A quel moment sont réalisées les tâches? → "Quand"
- Dans quels lieux sont effectués les traitements? → "Où"

Pour ces niveaux, les principaux modèles utilisés sont :

- Le modèle logique de données (MLD) pour les données.
- Le modèle logique des traitements(MLT) pour les traitements.

## LE NIVEAU PHYSIQUE

**Le niveau physique traduit les choix techniques et/ou matériels mis en œuvre pour répondre aux besoins de la solution logicielle décidée au niveau logique.**

**(ex: un serveur avec un disque dur de 5TO, un ordinateur équipé d'un lecteur de carte , une connexion adsl pour internet, un SGBD pour une base de données, etc....).**

**Pour ce niveau, les principaux modèles sont**

- Le modèle physique de données (MPD) pour les données.**
- Le modèle physique des traitements (MPT) pour les traitements.**

## LE DICTIONNAIRE DES DONNEES

### DEFINITION

Le dictionnaire des données est un document qui regroupe toutes les données du système d'information à modéliser.

Ces données seront représentées dans le modèle conceptuel de données et seront stockées dans la future base de données.

Il se représente sous la forme d'un tableau où chaque colonne contient :

- ➡ Un code mnémonique. (Ex : NoClt)
- ➡ La description des données. (Ex : Ce champ représente le N° du client)
- ➡ Le type des données. ( Ex: A pour alphanumérique)
- ➡ Un commentaire facultatif pour chaque donnée.
- ➡ Les contraintes ou règles de calcul. (Ex: non nul pour indiquer qu'une valeur est obligatoire)

## LES REGLES DE GESTION

### DEFINITION

Les règles de gestion (ou règles métier) sont définies lors de la phase d'analyse. Elles permettent d'indiquer la manière dont seront gérées les données dans la future solution logicielle, afin de répondre, de façon optimale, aux objectifs fonctionnels définis par le client.

### Exemples

- La liste des clients devra être affichée dans un tableau.
- La liste des fournisseurs devra être sauvegardée dans un fichier Word.
- Un auteur peut écrire plusieurs livres.
- Un étudiant ne peut emprunter que trois livres maximum à la bibliothèque
- Un message d'erreur s'affiche lors d'une mauvaise saisie de la date de naissance
- etc....

# LE MODELE CONCEPTUEL DE DONNES (MCD)

## LE MODELE CONCEPTUEL DE DONNES (MCD)

### DEFINITION

Le modèle conceptuel de données (aussi appelé modèle entité-association) est une représentation statique, graphique et structurée du système d'information d'une entreprise.

Il a pour but de schématiser de façon claire et cohérente, l'organisation des données que manipule une entreprise et les relations entre celles-ci.

Sa représentation se compose :

- ➡ D'entités
- ➡ De propriétés
- ➡ D'associations
- ➡ De cardinalités



## L'ENTITE

### DEFINITION

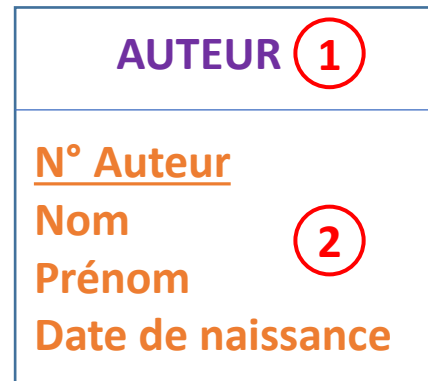
Une entité (ou classe d'entité) est une représentation abstraite d'un objet matériel ou immatériel du monde réel.

Elle se schématise par un rectangle en deux parties :

La première partie contient le nom de l'entité.

La seconde partie contient les propriétés (ou attributs) de l'entité.

Exemple:



① Nom de l'entité

② Propriétés de l'entité

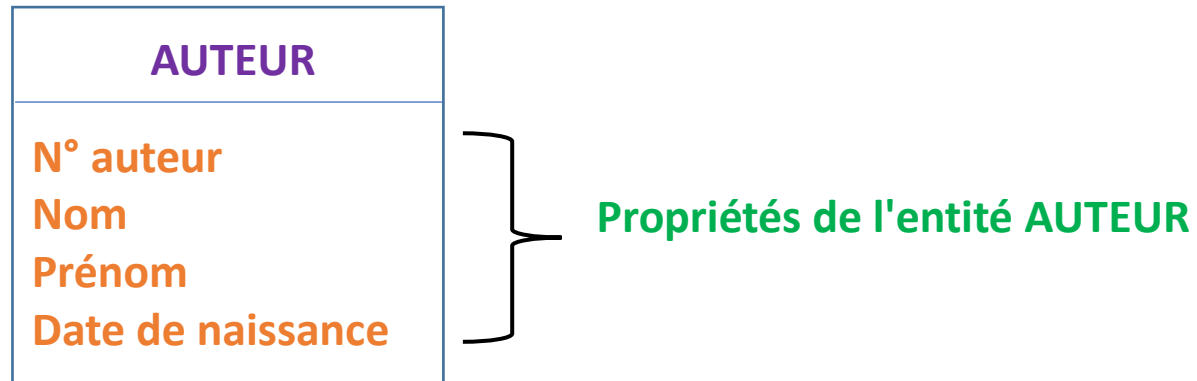
# LES PROPRIETES

## DEFINITION

Une propriété (ou attribut) est une donnée élémentaire (ou atomique) qui:

- Est un élément distinctif d'une entité et qui participe à l'identification d'un objet du monde réel.
- Ne doit avoir qu'une valeur unique (monovalué)
- Peut enrichir une association

Exemple :



## LA PROPRIETE IDENTIFIANT

### DEFINITION

Une entité comprend, en général, plusieurs propriétés dont une, que l'on qualifie d'identifiant (ou clé d'entité).

Une propriété identifiant :

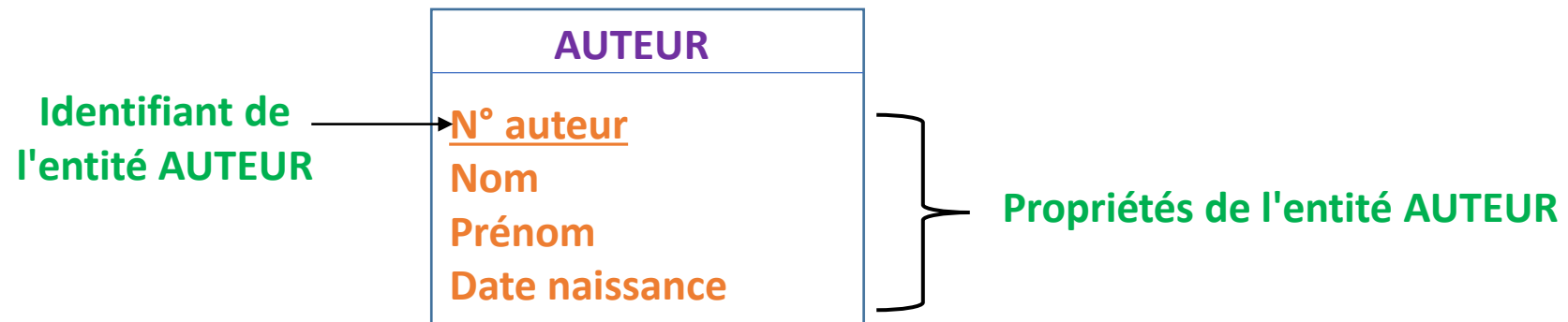
- Se place en tête de liste des propriétés de l'entité.

- Doit obligatoirement avoir une valeur et celle-ci doit être unique.

- Est, généralement, soulignée.

Il est à noter qu'une entité peut avoir plusieurs propriétés qui jouent le rôle d'identifiant.

Exemple :



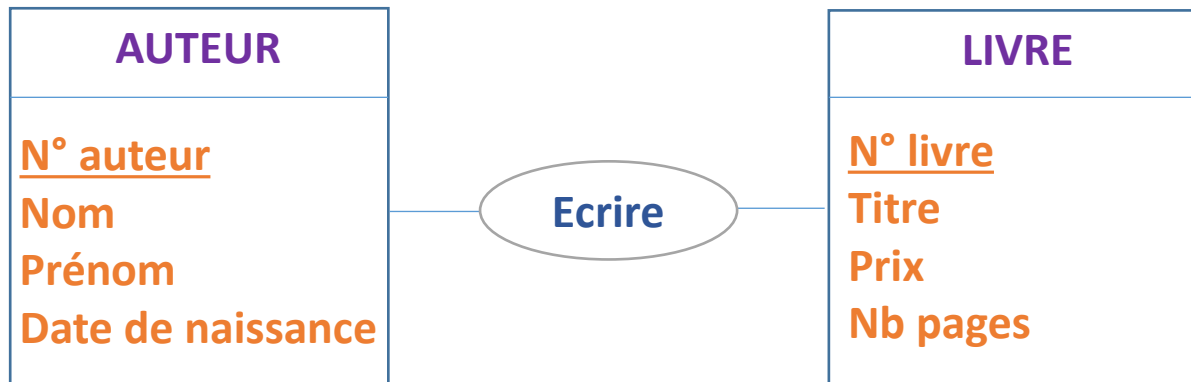
# LES OCCURRENCES

## DEFINITION

Une occurrence représente soit :

- La valeur d'une propriété (d'entité ou d'association).
- L'ensemble des valeurs des propriétés d'une entité.
- La valeur prise par le lien d'une association.

## Exemple



- {1,"Giono","Jean",30/03/1895"} est une occurrence de l'entité Auteur.

- {9,"Le hussard sur le toit", 5,95€,50} est une occurrence de l'entité Livre.

- "Giono" est une occurrence de la propriété 'nom' de l'entité Auteur.

- "Le hussard sur le toit" est une occurrence de la propriété 'titre' de l'entité Livre.

- {"Giono", "Jean", 30/03/1895 a écrit "le Hussard sur le toit", 5,95€, 50} est une occurrence de l'association ECRIRE

# LES ASSOCIATIONS

## DEFINITION

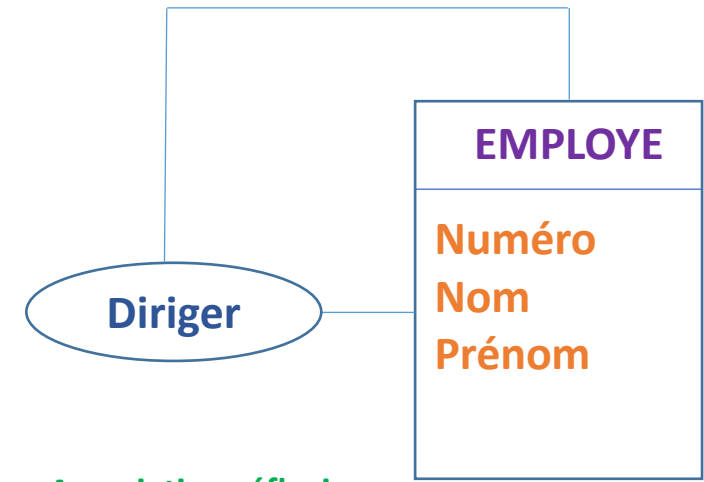
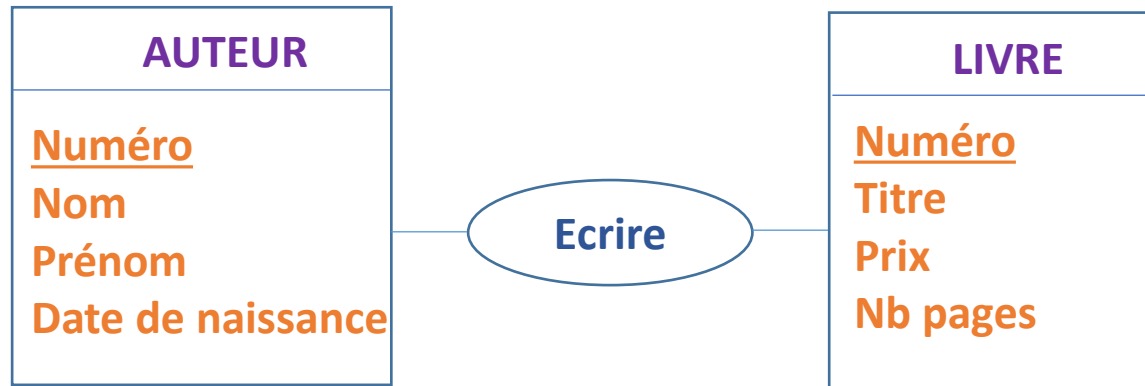
Une association (ou classe d'association) décrit le lien (ou relation) qui existe entre deux ou plusieurs entités.

Une association est dite binaire lorsqu'elle lie deux entités, ternaire lorsqu'elle lie trois entités et n-aires lorsqu'elle relie n entités.

Une association qui associe les occurrences d'une même entité est dite réflexive.

Le nom de l'association doit être un verbe et elle se représente par une ellipse et des traits reliant chaque entités.

Exemples :



## ASSOCIATION PORTEUSE DE DONNEES

### DEFINITION

Une propriété appartient à une association lorsqu'elle dépend de toutes les entités liées dans cette association.

Pour chaque ensemble formé par les occurrences des entités liées à l'association, il n'existe qu'une seule et unique valeur pour cette propriété.

On dit alors que l'association est porteuse de données.

Une association peut contenir plusieurs propriétés.

Les propriétés d'association se situent dans la partie inférieure de celle-ci, sous son nom.

Exemple :



SI L'ON CONNAIT

- le numéro du client

ET

- le numéro du livre

ALORS la date de commande sera connue de façon certaine.

## LES DIFFERENTS TYPES D'ASSOCIATIONS

### DEFINITION

On appelle type d'association le couple déterminé par le nombre d'occurrences des entités considérées de chaque côté d'une association.

On distingue trois types d'associations qui sont fonction des cardinalités maximales situées de part et d'autre d'une association :



Association un à un [1,1]



Association un à plusieurs [1,N]



Association plusieurs à plusieurs [n,m]

Les types d'association concernent uniquement les associations binaires.

## ASSOCIATION UN A UN

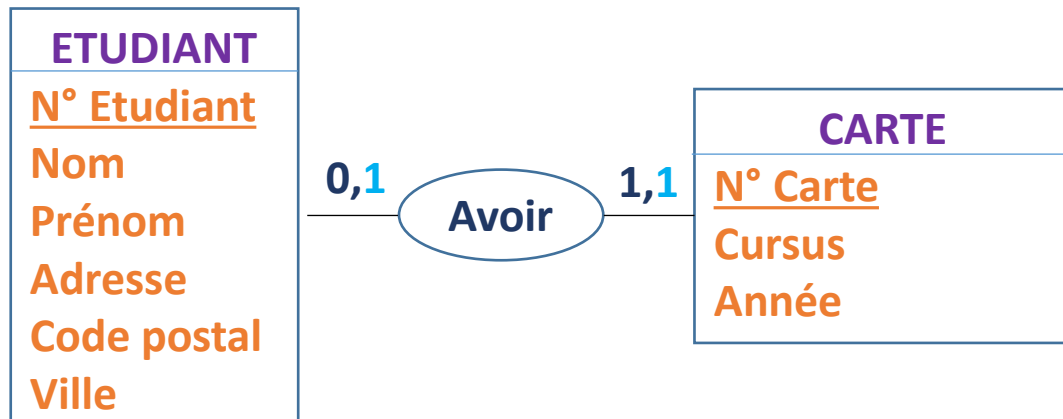
### DEFINITION

Une association est de type **un à un** :

Si pour une occurrence d'une entité A peut correspondre, par l'association, au plus, une et une seule occurrence de l'entité B.

Et, inversement, pour une occurrence de l'entité B, ne peut correspondre, par l'association, au plus, qu'une et une seule occurrence de l'entité A.

### Exemple



L'association est de type un à un car les cardinalités maximales sont à 1.

- Supposons une occurrence de l'entité Etudiant:  
{101250, DUPONT, Nicolas, 15 rue des près, 75010, PARIS}

- Supposons une occurrence de l'entité Carte  
{25025, Sorbonne, Lettres, 2014}

A l'occurrence de l'entité Etudiant ne correspond qu'une et une seule occurrence de l'entité Carte. L'inverse est vraie aussi.



# ASSOCIATION UN A PLUSIEURS

## DEFINITION

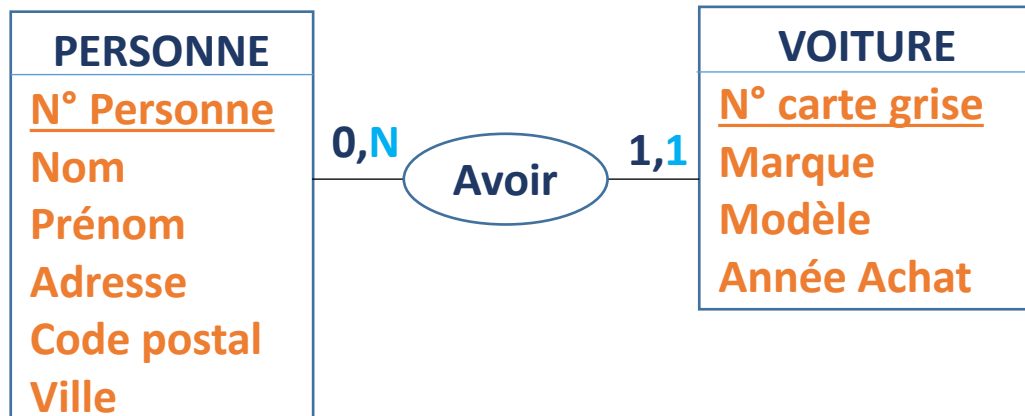
Une association est de type **un à plusieurs** :

Si pour une occurrence d'une entité A peut correspondre, par l'association, plusieurs occurrences de l'entité B.

Et pour une occurrence de l'entité B, peut correspondre, par l'association, au plus, qu'une et une seule occurrence de l'entité A.

L'association est de type un à plusieurs car les cardinalités maximales sont à 1 et N.

## Exemple



- Supposons une occurrence de l'entité PERSONNE:  
{1455, DURANT Franck, 50 rue des acacias, 75015, PARIS}
- Supposons deux occurrences de l'entité VOITURE  
{104A2545, MERCEDES, CLS500, 2014}  
{104B4590, BMW, 525, 2010}

A l'occurrence de l'entité PERSONNE correspond plusieurs occurrences de l'entité VOITURE. A une occurrence de l'entité VOITURE ne correspond qu'une et une seule occurrence de l'entité PERSONNE.

# ASSOCIATION PLUSIEURS A PLUSIEURS

## DEFINITION

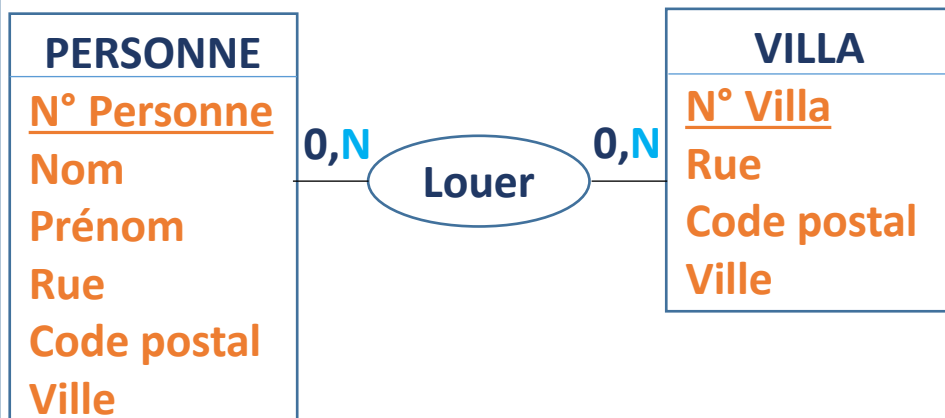
Une association est de type **plusieurs à plusieurs** :

Si pour une occurrence d'une entité A peut correspondre, par l'association, plusieurs occurrences de l'entité B.

Et pour une occurrence de l'entité B, peut correspondre, par l'association, plusieurs occurrences de l'entité A.

L'association est de type un à plusieurs car les cardinalités maximales sont à N et N.

## Exemple



- Supposons une occurrence de l'entité **PERSONNE**:

{14055, DURANT, Franck, 50 rue des acacias, 75015, PARIS}

{25055, DUPONT, Nicolas, 25 avenue des marronniers, 75005 PARIS}

- Supposons trois occurrences de l'entité **VILLA**

{14055, 15 rue des peupliers, 34000 MONTPELLIER}

{14055, 45 avenue des chênes, 34000 MONTPELLIER}

{25055, 15 rue des peupliers, 34000 MONTPELLIER}

→ Franck Durant a loué 2 villas à Montpellier.

L'appartement au 15 rue des peupliers a été loué par Franck Durant et Nicolas Dupont.

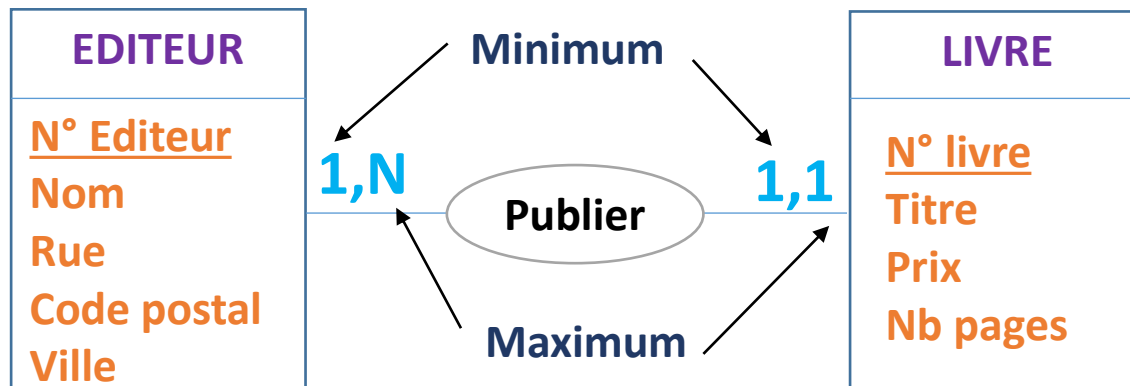
# LES CARDINALITES

## DEFINITION

La cardinalité d'un couple entité-association est un couple d'entiers,  $(x, y)$ , qui exprime le nombre minimum  $(x)$  et maximum  $(y)$  d'occurrences de l'association qui existe pour chaque occurrence de l'entité.

Les cardinalités sont une traduction des règles de gestion.

## Exemple



### Cardinalité du couple EDITEUR-Publier : 1,N

Un éditeur peut publier au minimum un livre.  
Un éditeur peut publier au maximum N livres.

### Cardinalité du couple LIVRE-Publier : 1,1

Un livre est publié au minimum par un éditeur.  
Un livre est publié au maximum par un éditeur.  
→ En effet, en général, un livre n'est publié que par un et un seul éditeur.

# LES DIFFERENTS TYPES DE CARDINALITES

## Cardinalités minimum

Valeurs	Explications
<b>0</b>	Une occurrence de l'entité peut exister mais ne pas participer à l'association
<b>1</b>	Une occurrence de l'entité participe obligatoirement au moins une fois à une occurrence d'association

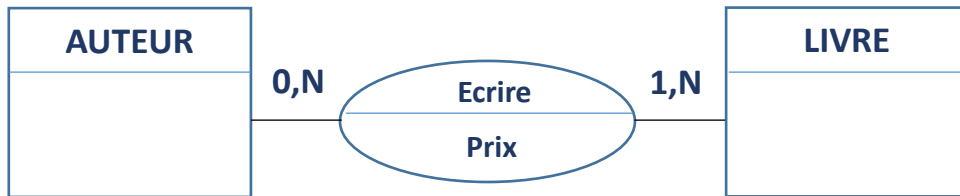
## Cardinalités maximum

Valeurs	Explications
<b>1</b>	Une occurrence de l'entité participe une fois au maximum à l'occurrence de l'association
<b>N</b>	Une occurrence de l'entité peut participer plusieurs fois à l'occurrence de l'association

## Valeurs possibles dans un MCD

Valeurs	Explications
<b>0,1</b>	Une occurrence de l'entité peut ne pas participer ou participer 1 fois au maximum à l'occurrence d'association
<b>1,1</b>	Une occurrence de l'entité participe une et une seule fois à l'occurrence d'association
<b>0,N</b>	Une occurrence peut ne pas participer ou participer plusieurs fois à l'occurrence d'association
<b>1,N</b>	Une occurrence participe au moins 1 fois à l'association ou participer plusieurs fois à l'association

## QUELQUES EXEMPLES DE CARDINALITES



### Cardinalité 0,N :

Un auteur peut ne pas écrire de livre au minimum ou N livres au maximum

### Cardinalité 1,N

Un livre peut être écrit par un auteur au minimum ou plusieurs

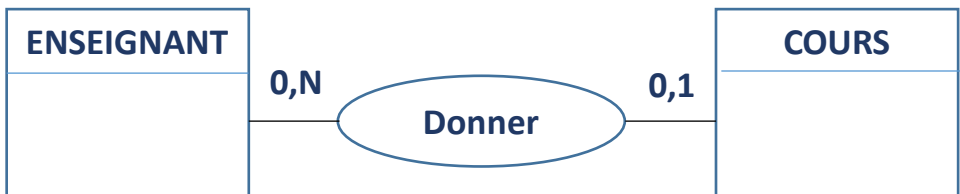


### Cardinalité 0,n

Une personne peut avoir aucune ou plusieurs voitures

### Cardinalité 1,1

Une voiture a un seul et unique propriétaire

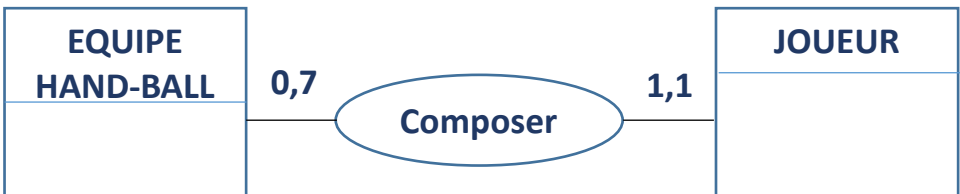


### Cardinalité 0,N

Un enseignant peut donner aucun cours ou plusieurs cours

### Cardinalité 0,1

Un cours peut être dispensé par aucun ou par un enseignant



### CAS PARTICULIER : Les cardinalités bornées

### Cardinalité 0,7

Une équipe de hand-ball peut avoir aucun joueur ou 7 joueurs au maximum

### Cardinalité 1,1

Un joueur fait partie d'une et une seule équipe

# LA REPRESENTATION DU TEMPS DANS UN MODELE CONCEPTUEL DES DONNES

Il existe deux points de vue concernant la représentation du temps dans un modèle conceptuel de données :

## ➡ La représentation synchronique

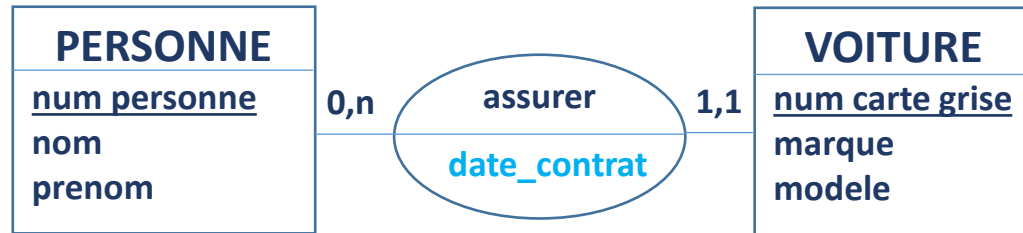
La notion de temps n'intervient pas comme élément distinctif pour rechercher une information. Cette représentation permet une vision instantanée de la réalité modélisée.

## ➡ La représentation diachronique

La notion de temps doit être prise en compte par l'intermédiaire d'éléments temporels (entités temporelles ou entités représentant des événements datés) pour rechercher une même information à différents moments dans le temps. Cette représentation permet une vision chronologique, historique de la réalité modélisée. Cette notion d'historique permet de conserver les valeurs antérieures prises par une propriété pour une occurrence d'une entité ou d'une relation.

# EXEMPLES DE REPRESENTATION SYNCHRONIQUE ET DIACHRONIQUE

## EXEMPLE SYNCHRONIQUE

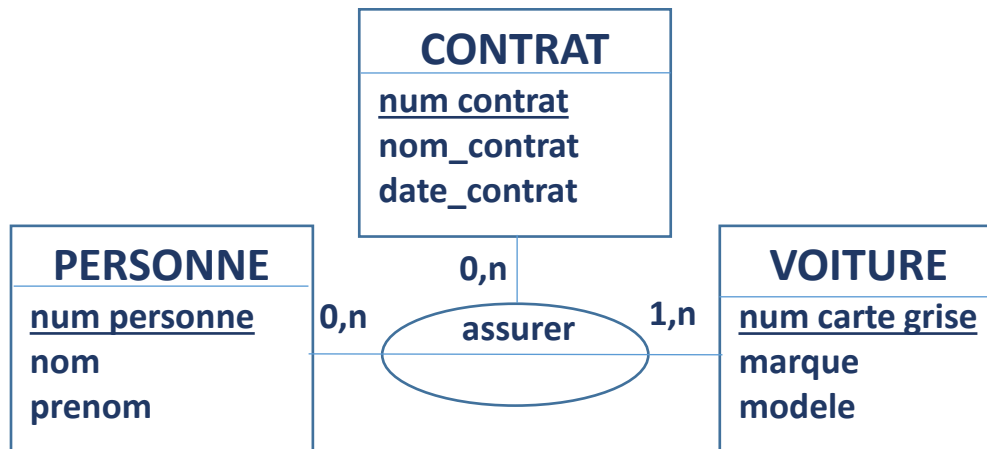


Cette modélisation nous permet de connaître la date à laquelle une personne a assuré sa voiture.

En effet, en connaissant le N° de la personne et le N° de carte grise, on peut connaître la date à laquelle la personne a assuré sa voiture .

Mais cette modélisation ne nous permet pas de connaître l'historique de ses contrats pour la voiture. En effet, si la personne souhaite modifier les caractéristiques de son contrat pour passer d'un contrat "Tous risques" à un contrat au tiers, nous ne pouvons pas garder les informations sur le contrat "Tous risques".

## EXEMPLE DIACHRONIQUE



Cette modélisation nous permet de prendre en compte la vision historique des contrats d'assurance.

En créant une entité temporelle, nous pouvons enregistrer les informations sur chaque contrat.

# REPRESENTATION SYNCHRONIQUE DANS LA BASE DE DONNEES

## EXEMPLE SYNCHRONIQUE



Dans l'exemple ci-contre, nous supposons que monsieur Benjamin Dubois assure sa voiture :

- 1) Le 14 janvier 2014 avec un contrat "Tout Risques".
- 2) Le 19 janvier 2015 avec un contrat "Bris de glace".
- 3) Le 09 janvier 2017 avec un contrat "Tiers".

Avec la modélisation ci-contre deux solutions s'offrent à nous pour enregistrer ces contrats :

N° Personne	Nom	Prenom	N° Carte grise	Marque	Modele	Date du contrat
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	14/01/2014
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	19/01/2015
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	09/01/2017

1

N° Personne	Nom	Prenom	N° Carte grise	Marque	Modele	Date du contrat
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	14/01/2014
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	19/01/2015
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	09/01/2017

N° Personne	Nom	Prenom	N° Carte grise	Marque	Modele	Date du contrat
<del>1</del>	<del>DUBOIS</del>	<del>Benjamin</del>	<del>1AB50</del>	<del>MERCEDES</del>	<del>CLS 500</del>	<del>14/01/2014</del>
<del>1</del>	<del>DUBOIS</del>	<del>Benjamin</del>	<del>1AB50</del>	<del>MERCEDES</del>	<del>CLS 500</del>	<del>19/01/2015</del>
1	DUBOIS	Benjamin	1AB50	MERCEDES	CLS 500	09/01/2017

2

1

Soit nous enregistrons les trois types de contrats dans la base de données.

Dans ce cas, les dates de contrats sont enregistrées dans la table VOITURE, ce qui n'a pas de sens. En outre, il y a redondance d'information puisque le numéro, le nom et le prénom de Mr Dubois seront enregistrés trois fois dans la table PERSONNE et le numéro de carte grise, la marque et le modèle de la voiture seront, eux aussi, enregistrés trois fois dans la table VOITURE.

2

Soit nous effaçons les contrats précédents de la base de données pour ne garder que le plus récent.

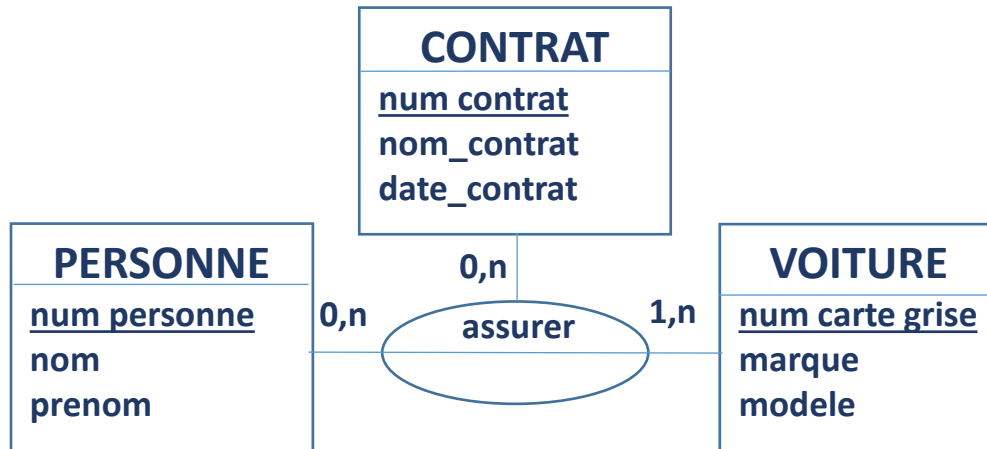
Dans ce cas, nous perdons l'historique des différents contrats de Mr Dubois et cela nous oblige à effectuer des requêtes SQL inutiles.

LES DEUX SOLUTIONS POSSIBLES AVEC CETTE MODELISATION NE SONT PAS OPTIMALES ET NE REPONDENT PAS A NOTRE BESOIN.



# REPRESENTATION DIACHRONIQUE DANS LA BASE DE DONNEES – 1<sup>er</sup> EXEMPLE

## EXEMPLE DIACHRONIQUE



Avec la modélisation diachronique :

- ① Les types de contrats sont enregistrés dans la table CONTRAT.
- ② Nous n'avons plus la redondance des informations dans les tables PERSONNE et VOITURE.
- ③ En outre, cette modélisation nous permet de ne plus indiquer la date de contrat dans l'entité Voiture.

①

N° Contrat	Nom	Date du contrat
1505550	Tous Risques	14/01/2015
1555040	Bris de glace	19/01/2015
2555550	Tiers	09/01/2017

②

N° Personne	Nom	Prenom
1	DUBOIS	Benjamin

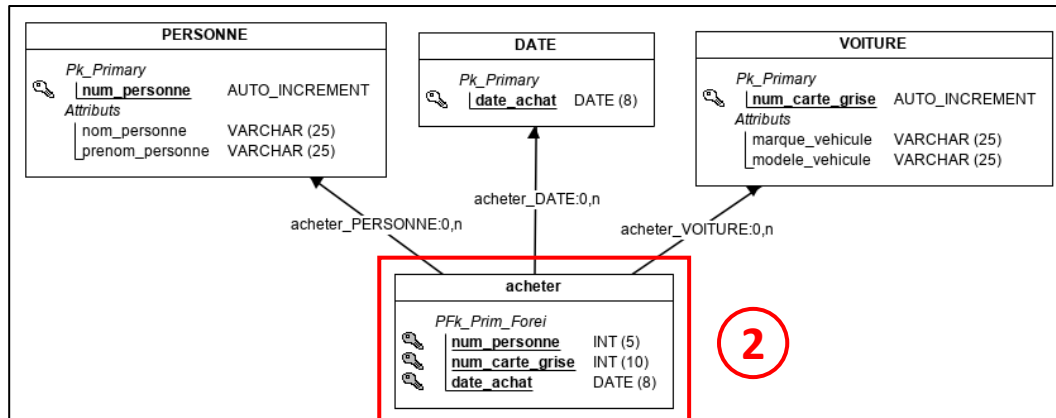
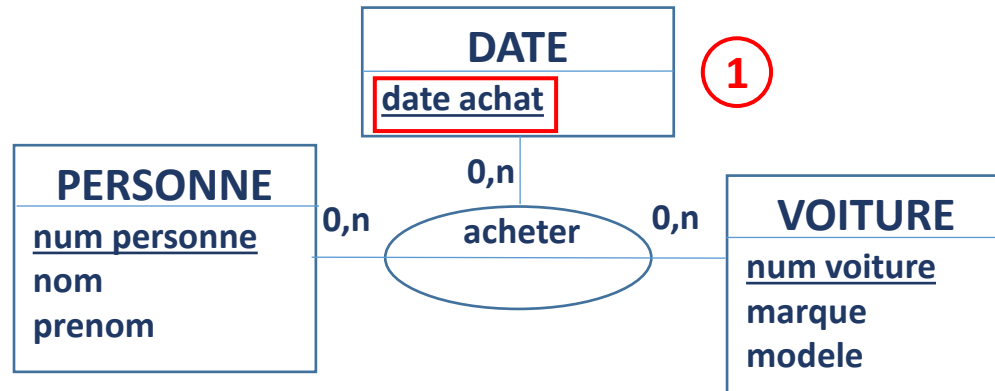
②

N° Carte grise	Marque	Modele
1AB50	MERCEDES	CLS 500

③

# REPRESENTATION DIACHRONIQUE DANS LA BASE DE DONNEES – 2<sup>e</sup> EXEMPLE

## EXEMPLE DIACHRONIQUE



Avec la modélisation diachronique ci-contre, nous avons créé une entité DATE pour pouvoir modéliser l'historisation de la date d'achat de la voiture.

Dans l'exemple précédent, il était normal de transformer l'entité CONTRAT en table puisque nous voulions enregistrer les différents noms de contrats de Mr Dubois.

- 1 Dans cet exemple, l'entité DATE ne contient qu'un attribut de type date et n'a pas vocation à être transformée en table dans la base de données. Si vous offrez la possibilité de choisir des dates depuis l'année 1900, cela signifierait qu'il faudrait enregistrer toutes les dates depuis 1900 jusqu'à aujourd'hui (....et rajouter un enregistrement chaque jour....).

LORSQUE QUE L'ON SOUHAITE MODÉLISER UNE HISTORISATION AVEC UNE ENTITÉ DATE QUI NE CONTIENT QUE DES ATTRIBUTS DE TYPE DATE, CETTE ENTITÉ N'EST VALABLE QUE POUR LA REPRÉSENTATION DANS LE MCD MAIS NE DOIT PAS ÊTRE TRANSFORMÉE EN TABLE DANS LA BASE DE DONNÉES.

- 2 Nous verrons lors du passage du MCD vers le MLD que cette entité va, en fait, disparaître au profit d'une table d'association car les cardinalités maximales de l'association sont de type N. Cette table d'association représentera l'association acheter et aura, comme attributs, les clés primaires des entités PERSONNE, VOITURE et DATE. Nous pourrions, ainsi, effectuer des requêtes dans la base de données pour associer les personnes aux voitures avec les dates d'achat.

## LES DIFFERENTES ETAPES POUR L'ELABORATION D'UN MCD

- 1** ETABLIR LE DICTIONNAIRE DES DONNEES
- 2** EPURER LE DICTIONNAIRE DE DONNEES ( sans synonymes ni polysèmes)
- 3** DETERMINER LES ENTITES DU MODELE
- 4** DETERMINER LES PROPRIETES DES ENTITES
- 5** DETERMINER LES ASSOCIATIONS ENTRE ENTITES
- 6** DETERMINER LES CARDINALITES ENTRE ENTITES ET ASSOCIATIONS

# LE MODELE LOGIQUE DE DONNES (MLD)

## LE MODELE LOGIQUE DE DONNEES (MLD)

### DEFINITION

Le modèle logique de données (ou modèle relationnel) a été conçu par Edgar Franck Codd en 1970 et se fonde sur la notion de relation, d'algèbre relationnel et de dépendances fonctionnelles.

C'est une représentation du modèle conceptuel de données, traduite sous une forme compréhensible par le système de gestion de bases de données (SGBD) où sera implémentée la future base de données.

# UN DOMAINE

## DEFINITION

Un domaine est un ensemble fini ou infini de valeurs.

## Exemples :

Le domaine des booléens :

$D1=\{0,1\}$

Le domaine des lettres de l'alphabet :

$D2=\{A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z\}$

Le domaine des entiers :

$D3=\{0,1,2,3,4,5,.....,1000,....., 500000,.....\}$

# UN ATTRIBUT

## DEFINITION

Dans un modèle relationnel, un attribut représente le nom d'une colonne d'une table et prend ses valeurs dans un domaine.

## Exemple :

Dans l'exemple ci-dessous ,l'attribut NOM est une colonne.  
La valeur de l'attribut NOM est "DUPONT" pour le client N°1

N° CLIENT	NOM	PRENOM	VILLE
1	DUPONT	Paul	PARIS
2	DURANT	Nicolas	NICE
3	DUBOIS	Benjamin	BORDEAUX
4	DUHAMEL	Fabien	BIARRITZ

## UNE RELATION

### DEFINITION

Une relation est un ensemble d'enregistrements relatif à un sujet et stockée dans une table. Chaque enregistrement(chaque ligne) est appelé un tuple.

Exemple :

CLIENT(#code\_clt, nom, rue, code postal, ville)

r={ (DUPONT,Paul, 10 rue des peupliers, 75010, PARIS),  
 (DURANT, Nicolas, 15 rue des marronniers, 75005, PARIS),  
 (DUBOIS, Benjamin, 25 rue des près, 75005, PARIS),  
 (DUHAMEL, Fabien, 9 rue des acacias, 750010 PARIS) }

CODE_CLT	NOM	RUE	CODE POSTAL	VILLE
1	DUPONT	10 rue des peupliers	75010	PARIS
2	DURANT	15 rues des marronniers	75005	PARIS
3	DUBOIS	25 rues des près	75007	PARIS
4	DUHAMEL	9 rue des acacias	750010	PARIS



## UNE CARDINALITE

### DEFINITION

Une cardinalité est le nombre de tuples contenu dans la relation (dans la table).

Exemple :

La cardinalité pour la relation ci-dessous est 4

NOM	PRENOM	RUE	CODE POSTAL	VILLE
DUPONT	Paul	10 rue des peupliers	75010	PARIS
DURANT	Nicolas	15 rues des marronniers	75005	PARIS
DUBOIS	Benjamin	25 rues des près	75007	PARIS
DUHAMEL	Fabien	9 rue des acacias	750010	PARIS

## LES CONTRAINTES D'INTEGRITE

Une contrainte d'intégrité est une règle qui permet d'interdire tous les cas d'incohérence pouvant être générés lors des ajouts ou des mises à jour des données pour avoir des données collant toujours à la logique de la réalité.

Pour la conception d'une base de données, il y a deux types de contraintes d'intégrité :

### Les contraintes d'intégrité référentielle

Une contrainte d'intégrité référentielle est une règle qui permet de garantir qu'une information d'une table B référencée dans une table A, existe bien dans la table B. On utilisera pour cela la notion de clé primaire et clé étrangère.

### Les contraintes d'intégrité fonctionnelles (CIF)

Une contrainte d'intégrité fonctionnelle(CIF) se définit par le fait que l'une des entités de l'association est complètement déterminée par la connaissance d'une ou plusieurs entités participant à cette même association.

On utilisera pour cela la notion de dépendance fonctionnelle.

## LA CLE PRIMAIRE

### DEFINITION

La clé primaire(ou clé principale) d'une relation est l'attribut ou l'ensemble d'attributs qui permet de désigner, de façon certaine, un tuple (un enregistrement d'une table).

### Exemple

Ci-dessous le n° de client 1 permet de connaitre de façon certaine le tuple {1, DUPONT, Paul, PARIS)

N° CLIENT	NOM	PRENOM	VILLE
1	DUPONT	Paul	PARIS
2	DURANT	Nicolas	NICE
3	DUBOIS	Benjamin	BORDEAUX
4	DUHAMEL	Fabien	BIARRITZ

## LES CLES ETRANGERES

### DEFINITION

Une clé étrangère est un attribut présent dans une table A qui fait référence à la clé d'une table B et qui permet d'identifier de façon unique un tuple de cette table B.

### EXEMPLE

TABLE A

N° CLIENT	NOM	PRENOM	N° VILLE
1	DUPONT	Paul	1
2	DURANT	Nicolas	2
3	DUBOIS	Benjamin	3
4	DUHAMEL	Fabien	4

TABLE B

N° VILLE	LIBELLE VILLE
1	PARIS
2	NICE
3	BORDEAUX
4	BIARRITZ

## SCHEMA D'UNE RELATION

### DEFINITION

Un schéma de relation est la liste des noms des attributs d'une relation (d'une table) ainsi que le domaine de chacun d'entre eux.

On note le schéma de relation R :

$R = [\text{nomAttribut1:Domaine1}, \text{nomAttribut2:Domaine2}, \text{nomAttribut3:Domaine3}, \dots]$

Par convention, on sous-entend le domaine et on note :

$R = [\text{nomAttribut1}, \text{nomAttribut2}, \text{nomAttribut3}]$

Exemple :

- Soit une relation client avec les attributs nom, prénom et date de naissance
- Soit le domaine D1 défini comme l'ensemble des chaînes de caractères de longueur maximale = 20
- Soit le domaine D2 défini comme l'ensemble des dates depuis le 1<sup>er</sup> janvier 1900

Le schéma de relation se note :

$\text{client} = [\text{nom}, \text{prenom}, \text{datenaissance}]$

## SCHEMA D'UNE BASE DE DONNEES RELATIONNELLE

### DEFINITION

Un schéma de base de données relationnelle est l'ensemble des relations qui composent la base de données.

### Exemple

- Soit une base de données COMMANDES
- Soit D1 : domaine des chaines de caractères de longueur maximale = 20.
- Soit D2 : domaine des entiers positifs

Clients=[ref\_clt, nom, prenom, rue, code postal, ville]

Fournisseurs = [ref\_frns, nom, rue, code postal, ville]

Produits=[ref\_pdt, desc\_pdt]

Afin de réaliser un schéma de base de données relationnelles cohérent, il est nécessaire d'effectuer une démarche de normalisation du modèle logique de données grâce aux dépendances fonctionnelles et aux formes normales.

## LES DEPENDANCES FONCTIONNELLES (DF) : Définition

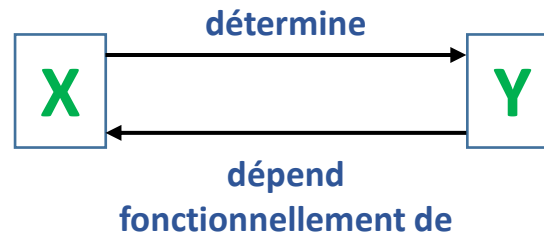
### DEFINITION

Un attribut Y "dépend fonctionnellement" d'un attribut X si et seulement si pour une valeur de l'attribut X, il existe une et une seule valeur pour l'attribut Y.

On dit alors que X détermine Y où X est la source et Y est le but.

On le note  $X \rightarrow Y$ .

Autrement dit, si l'on connaît X, on peut en déduire une seule valeur de Y.



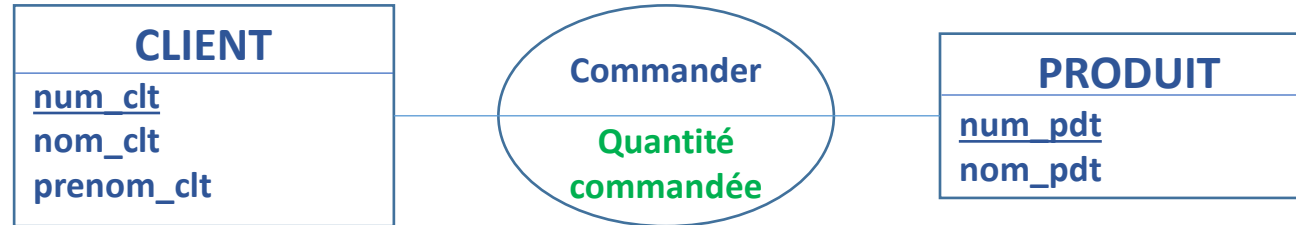
## LES DEPENDANCES FONCTIONNELLES (DF) : 1<sup>er</sup> exemple

CLIENT
<u>num_clt</u>
nom_clt
prenom_clt
date_naiss_clt
num_tel_clt

- Est-ce que num\_clt est en dépendance fonctionnelle avec nom\_clt?  
OUI, pour un numéro de client, il n'y a qu'un nom de client
- Est-ce que num\_clt est en dépendance fonctionnelle avec prenom\_clt?  
OUI, pour un numéro de client, il n'y a qu'un prénom
- Est-ce que num\_clt est en dépendance fonctionnelle avec date\_naiss\_clt?  
OUI, pour un numéro de client, il n'y a qu'une seule date de naissance.
- Est-ce que num\_clt est en dépendance fonctionnelle avec num\_tel\_clt?  
NON, le client pourrait avoir un numéro de téléphone fixe et un numéro de téléphone portable. Il faudrait donc ajouter une propriété pour différencier le numéro de téléphone fixe et le numéro de téléphone portable.



## LES DEPENDANCES FONCTIONNELLES (DF) : 2<sup>e</sup> exemple



### Pourquoi mettre Quantité commandée comme attribut d'association?

Si quantité commandée est indiquée dans l'entité client :

- 1) Dépendance fonctionnelle :  $\text{num\_clt} \rightarrow \text{quantité commandée}$ ? Non puisque un client peut commander plusieurs produits donc plusieurs quantités.
- 2) Plus simplement, cela signifierait que quelque soit le produit commandé, le client ne pourrait commander qu'une quantité donnée.

Si quantité commandée est indiquée dans l'entité produit :

- 1) Dépendance fonctionnelle :  $\text{num\_pdt} \rightarrow \text{quantité commandée}$ ? Non puisque un produit peut être commandé par plusieurs clients donc plusieurs quantités pour un produit donné.
- 2) Plus simplement, cela signifierait que quelque soit le client, le produit ne pourrait être commandé qu'avec une quantité fixe, le client n'aurait pas la possibilité de choisir la quantité pour le produit en question.

LA SOLUTION : Mettre quantité commandée en attribut d'association reliant l'entité CLIENT et l'entité PRODUIT :

- 1)  $\text{num\_clt}, \text{num\_pdt} \rightarrow \text{Quantité commandée}$ ? OUI puisque pour un client donné et un produit donné, nous connaissons forcément la quantité commandée.
- 2) Plus simplement, si nous connaissons le client et si nous connaissons le produit, nous connaissons la quantité du produit que le client a commandé.

## LES AXIOMES D'ARMSTRONG

A partir d'un ensemble F de dépendances fonctionnelles entre les attributs d'un schéma de relation R, il est possible de déduire d'autres dépendances fonctionnelles à partir des trois propriétés suivantes :

Soit X et Y deux attributs d'un schéma de relation R

- La réflexivité : Si X contient Y alors  $X \rightarrow Y$  (Tout ensemble d'attributs détermine un de ses sous ensemble)
- La transitivité : Si  $X \rightarrow Y$  et  $Y \rightarrow Z$  alors  $X \rightarrow Z$
- L'augmentation : Pour tout attribut Z inclus dans le schéma de relation R  
Si  $X \rightarrow Y$  est vérifiée dans R alors  $XZ \rightarrow YZ$  ( on écrit aussi  $X \rightarrow Y \mid = XZ \rightarrow YZ$ )

## AUTRES PROPRIETES : LA PSEUDO-TRANSITIVITE

### DEFINITION

Si l'on a deux dépendances fonctionnelles:

$$X \rightarrow Y \text{ et } WY \rightarrow Z$$

Alors on peut écrire :

$$XW \rightarrow Z$$

### EXEMPLE

- Soit **X**, l'attribut qui représente la **référence d'une maison**
- Soit **Y**, l'attribut qui représente la **désignation de la maison**
- Soit **W**, l'attribut qui représente la **superficie de la maison**
- Soit **Z**, l'attribut qui représente le **prix de vente de la maison**

Si l'on a la **référence X** → la **désignation Y**,

ET si l'on a la **superficie et la désignation WY** → le **prix de vente Z**

ALORS on a **référence et la superficie XW** → le **prix de vente Z**

## AUTRES PROPRIETES : L'UNION

### DEFINITION

Si l'on a deux dépendances fonctionnelles:

$$X \rightarrow Y \text{ et } X \rightarrow Z$$

Alors on peut écrire :

$$X \rightarrow YZ$$

### EXEMPLE

- Soit **X**, l'attribut qui représente la **référence d'une maison**
- Soit **Y**, l'attribut qui représente la **désignation de la maison**
- Soit **Z**, l'attribut qui représente le **prix de vente de la maison**

SI l'on a la **référence X** → la **désignation Y**,

ET si l'on la **référence X** → le **prix de vente Z**

ALORS on a **référence X** → **désignation Y, prix de vente Z**

## AUTRES PROPRIETES : LA DECOMPOSITION

## DEFINITION

Si l'on a :

$$X \rightarrow YZ$$

Alors :

$$X \rightarrow Y$$

Et  $X \rightarrow Z$

## EXEMPLE

- Soit **X**, l'attribut qui représente la **référence d'une maison**
- Soit **Y**, l'attribut qui représente la **désignation de la maison**
- Soit **Z**, l'attribut qui représente le **prix de vente de la maison**

SI l'on a la **référence X**  $\rightarrow$  la **désignation Y**, le **prix de vente Z**

ALORS on a la **référence X**  $\rightarrow$  la **désignation Y**

ET on a **référence X**  $\rightarrow$  le **prix de vente Z**

## LES DEPENDANCES FONCTIONNELLES ELEMENTAIRES

### DEFINITION

Une dépendance fonctionnelle est dite élémentaire lorsque la partie gauche de la dépendance (la source) ne contient pas d'attribut superflus pour connaître la partie droite (le but).

- Soit un attribut **Ref livre**
- Soit un attribut **Num commande**
- Soit un attribut **Titre livre**
- Soit un attribut **Quantité commandée**

**Ref livre** → **Titre livre**

{**Num commande**, **Ref livre**} → **Quantité commandée**

{**Num commande**, **Ref livre**} → **Titre livre**

- La première dépendance est élémentaire car la référence du livre permet de connaître le titre du livre.
- La seconde dépendance est élémentaire car le N° de commande seul ou la référence seule ne permettent pas de connaître la quantité commandée.
- La troisième dépendance n'est pas élémentaire car le N° de commande n'est pas obligatoire pour connaître le titre du livre, seule la référence du livre suffit.

## LES DEPENDANCES FONCTIONNELLES DIRECTES

### DEFINITION

On dit qu'une dépendance fonctionnelle  $X \rightarrow Y$  est directe s'il n'existe aucun attribut  $Z$  tel que l'on puisse avoir  $X \rightarrow Z$  et  $Z \rightarrow Y$ .

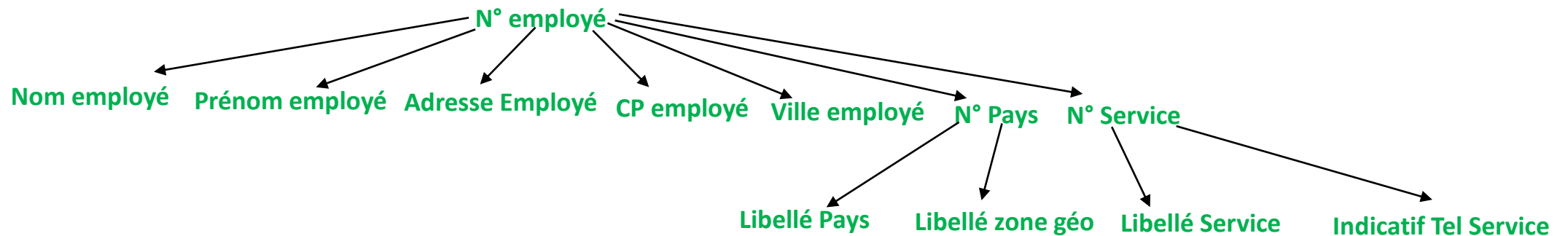
Si c'est le cas, cela signifie que la dépendance entre  $X$  et  $Y$  a été obtenue par transitivité. Les DF déduites par transitivité doivent être éliminées car elles sont redondantes.

## LE GRAPHE DES DEPENDANCES FONCTIONNELLES

### DEFINITION

Le graphe des dépendances fonctionnelles est une représentation graphique des dépendances fonctionnelles directes entre les données répertoriées dans le dictionnaire des données élaboré lors de la phase conceptuelle (modèle conceptuel de données).

Exemple :





## LES FORMES NORMALES

### DEFINITION

Les formes normales peuvent se définir comme des règles à appliquer dans un modèle logique de données afin de vérifier l'atomicité des données d'une part et l'absence de redondances d'autre part.

Il existe huit formes normales mais seules les trois premières sont les plus utilisées.

Le respect de ces formes normales par le modèle logique de données doit se faire dans l'ordre et permettra dans la future base de données :

- ➡ D'éviter la redondance des données.
- ➡ L'intégrité des données.
- ➡ De limiter les incohérences et la perte des données.
- ➡ D'améliorer la taille de la base et les performances des traitements.
- ➡ De faciliter la mise à jour des données.

## LA 1<sup>ère</sup> FORME NORMALE

### DEFINITION

Une relation (table) est en 1<sup>ère</sup> forme normale (1FN) si :

- Elle possède une clé primaire identifiant de façon unique et stable chaque ligne
- Tous ses attributs sont constants dans le temps et sont monovalués.
- Tous ses attributs sont atomiques (non décomposables).

Exemple :

EDITEUR
<u>N° Editeur</u>
Nom
<b>Titre</b>
Rue
Code postal
Ville

Passage en 1<sup>ère</sup> forme normale



EDITEUR
<u>N° Editeur</u>
Nom
Rue
Code postal
Ville

LIVRE
<u>N° Livre</u>
N° Editeur
<b>Titre</b>
Nb Pages
Date de parution

EDITEUR n'est pas en 1<sup>ère</sup> forme normale car l'attribut Titre contient tout les titres des livres édités chez l'éditeur.  
L'attribut n'est donc pas monovalué.

Nous avons créer une entité LIVRE pour chaque livre que l'éditeur publie.  
L'attribut Titre Livre n'a, ainsi, qu'une seule valeur.

## LA 2<sup>e</sup> FORME NORMALE

### DEFINITION

Une relation (table) est en 2<sup>e</sup> forme normale (2FN) si :

- Elle est en 1<sup>ère</sup> forme normale.
- Tous les attributs non identifiants sont en dépendance fonctionnelle avec l'identifiant en entier et non d'une partie de l'identifiant.

Exemple :

INTERVENTION
<u>N° Client</u>
<u>N° Technicien</u>
<u>Nom technicien</u>
Description
Date

Passage à la 2<sup>e</sup> forme normale



TECHNICIEN
N°Technicien
Nom
Prenom
Specialite

INTERVENTION
N° Client
N° Technicien
Description
Date

Le nom du technicien dépend fonctionnellement du N° technicien et non du couple N°Client/ N°Technicien qui forme l'identifiant.

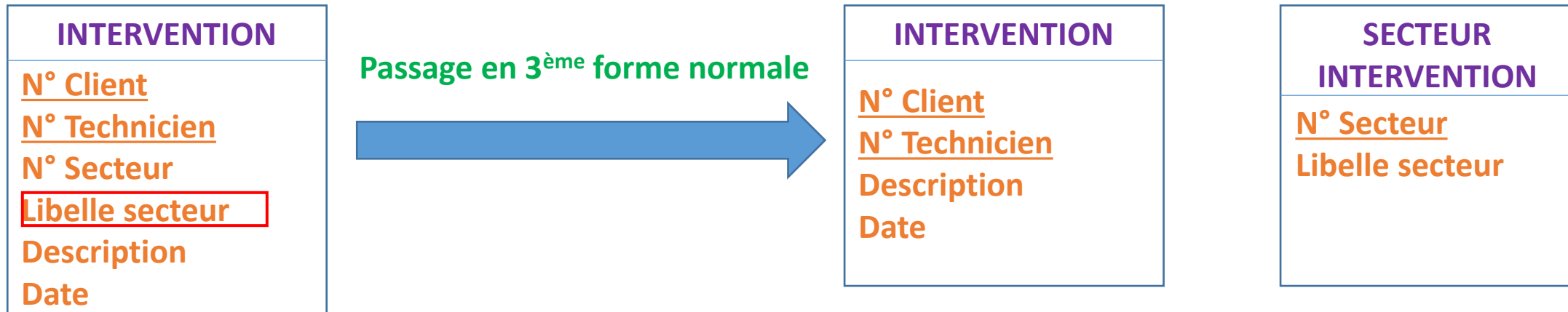
LA 3<sup>e</sup> FORME NORMALE

## DEFINITION

Une relation est en 3<sup>e</sup> forme normale (3FN) si :

- Elle est en 2<sup>e</sup> forme normale.
- Chaque attribut de la relation(table) est en dépendance fonctionnelle avec l'identifiant et pas avec un autre attribut non identifiant.

Exemple :



INTERVENTION n'est pas en 3<sup>ème</sup> forme normale car l'attribut 'Libellé secteur' dépend fonctionnellement de 'N° Secteur' qui n'est pas un attribut identifiant.

## LA 3<sup>e</sup> FORME NORMALE DE BOYCE-CODD

### DEFINITION

Une relation (table) est en 3<sup>e</sup> forme normale de Boyce-Codd (FNBC) si :

- Elle est en 3<sup>e</sup> forme normale
- Un des attributs clé formant la clé composée ne dépend pas fonctionnellement d'un attribut non clé → Tout attribut non clé ne doit pas déterminer une partie de la clé.

### Exemple :

PERSONNE
<u>N° Sécurité sociale</u>
<u>N° Pays</u>
Nom
Région

Passage en FNBC



PERSONNE
<u>N° Sécurité sociale</u>
N° Région
Nom
Rue
Code postal
Ville

REGION
<u>N° Région</u>
Nom Région
<u>N° Pays</u>
Superficie

PERSONNE n'est pas en 3<sup>e</sup> forme normale de Boyce\_Codd car l'attribut clé N° Pays dépend fonctionnellement de l'attribut non clé Région.

Nous avons décomposé la table PERSONNE et créé une seconde table nommée REGION dans laquelle nous indiquons le nom de la région et le N° du pays auquel elle est rattachée.  
(Le N° Pays implique qu'il existe déjà une table pays mais nous ne l'avons pas représentée dans cet exemple.)

# Passage du MCD au MLD

**5 règles doivent être appliquées pour effectuer le passage du MCD au MLD**

## REGLE N°1

Toute entité devient une table dans laquelle les propriétés de l'entité deviennent les colonnes de la table.

L'identifiant de l'entité devient la clé primaire de la table.

## EXEMPLE :

## ENTITE

CLIENT
<u>N° client</u>
Nom
Prenom
Rue
Code postal
Ville

L'entité CLIENT devient  
la table CLIENT



## TABLE

CLIENT
<u>num_clt</u>
nom_clt
prenom_clt
rue_clt
cp_clt
ville_clt

①

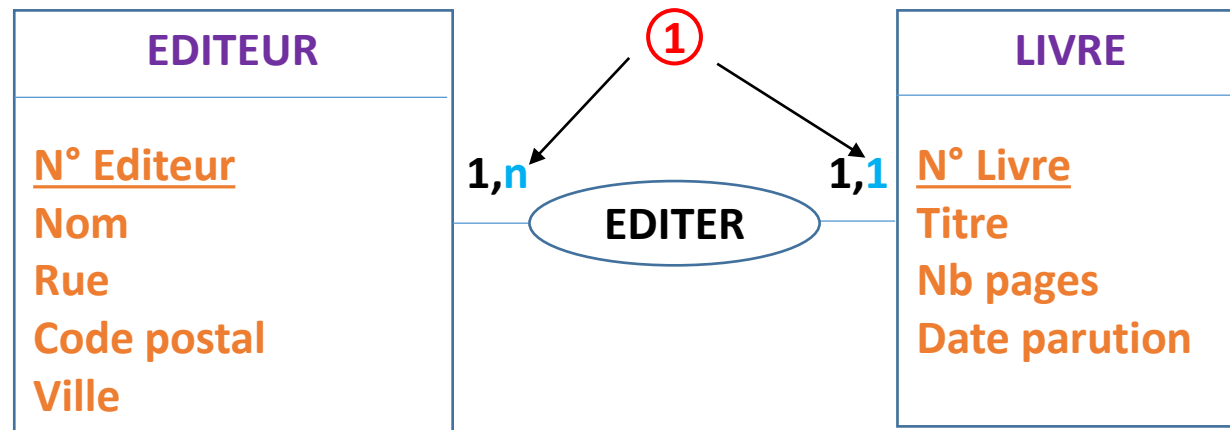
Il est recommandé de  
nommer les colonnes des  
tables selon les règles de  
nommage pour les bases  
de données.

## REGLE N°2

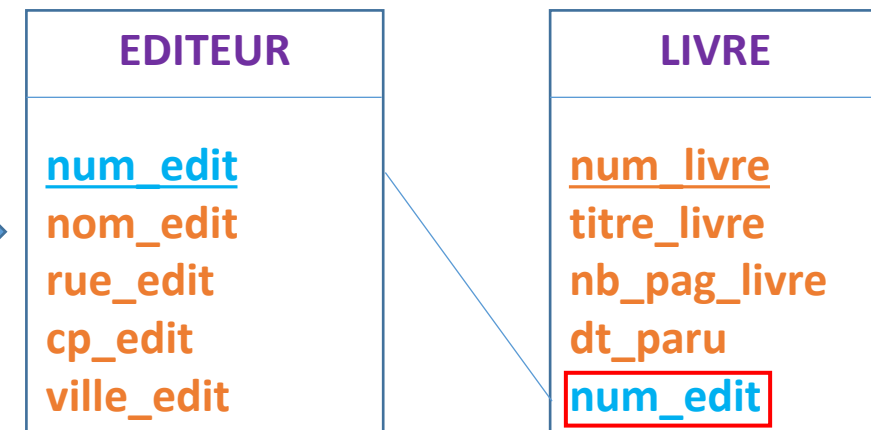
Une association binaire de type 1:n disparaît pour être remplacée par une clé étrangère dans la table, côté 0,1 ou 1,1, qui référence la clé primaire de l'autre table.  
(La valeur de la clé étrangère doit être non vide si la cardinalité est égale à 1,1.)

## EXEMPLE :

## ENTITES



## TABLES



① RAPPEL : L'association est de type 1:n car les cardinalités maximales sont 1 et n ( Le sens de lecture n'a pas d'importance).



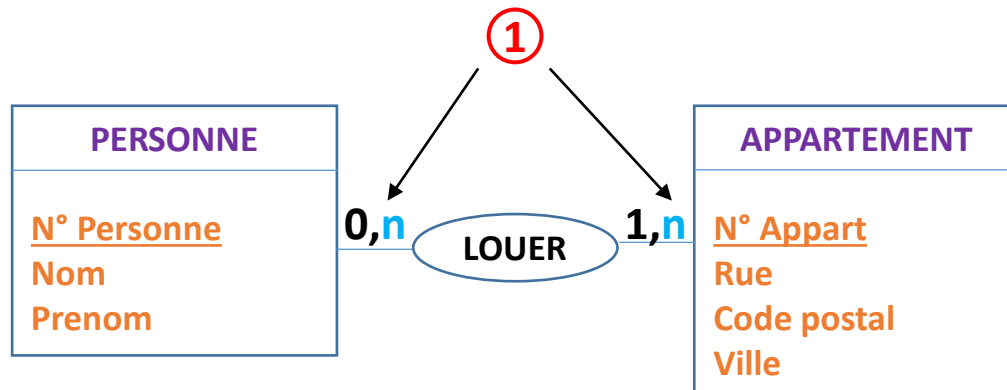
## REGLE N°3

Une association binaire de type n:m devient une table supplémentaire (table d'association) dont la clé primaire est composée de deux clés étrangères qui sont les clés primaires des deux tables liées à l'association.

Ces deux clés étrangères sont les colonnes de cette nouvelle table.

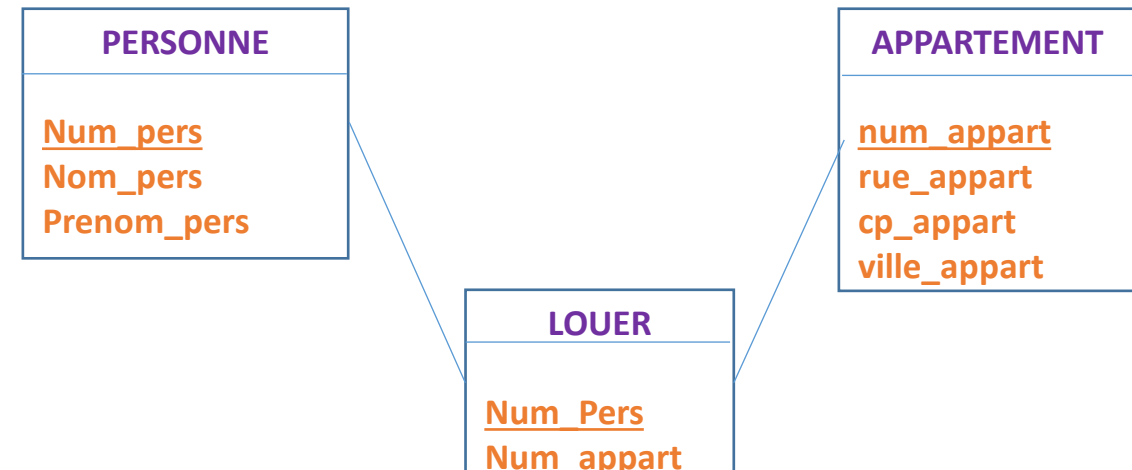
## EXEMPLE

## ENTITES



① RAPPEL : L'association est de type n:m car les cardinalités maximales sont n et n.

## TABLES



## REGLE N°4

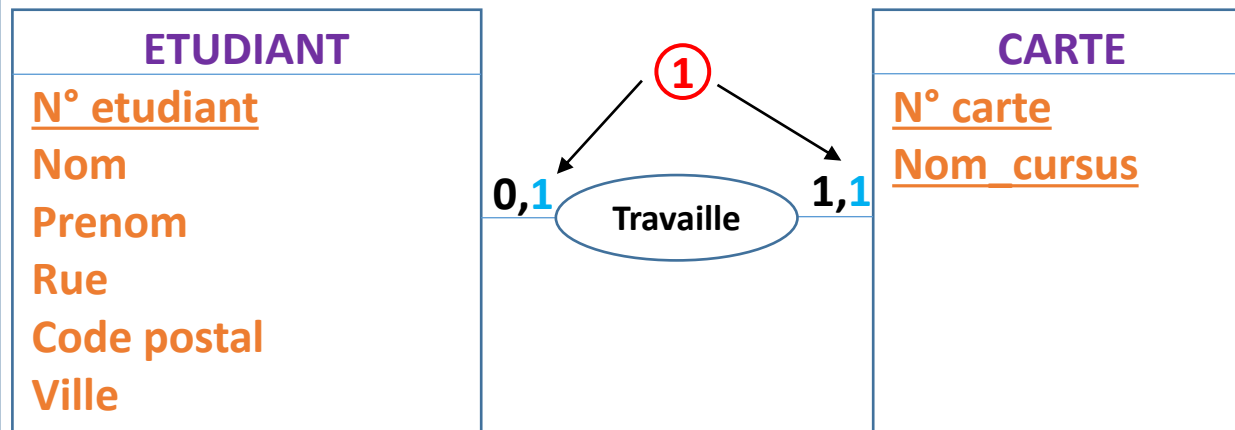
Pour une association binaire de type 1:1 :

Si la relation est 0,1 – 1,1 (ou 1,1-0,1) alors on applique le même principe que la règle n°2

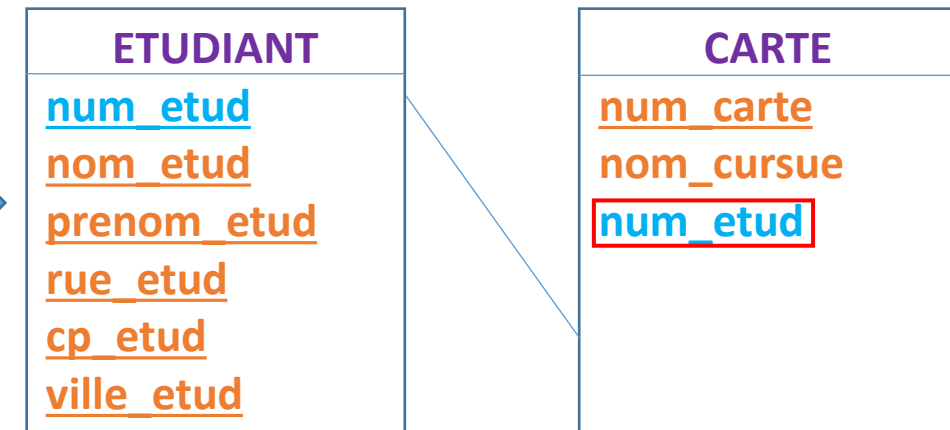
Si la relation est 0,1-0,1 , la clé étrangère va dans l'une ou l'autre des deux entités.

## EXEMPLE :

## ENTITES



## TABLES



① RAPPEL : L'association est de type 1:1 car les cardinalités maximales sont 1 et 1.

L'attribut num\_etud peut glisser comme clé étrangère dans l'entité Carte (comme ci-dessus).

OU

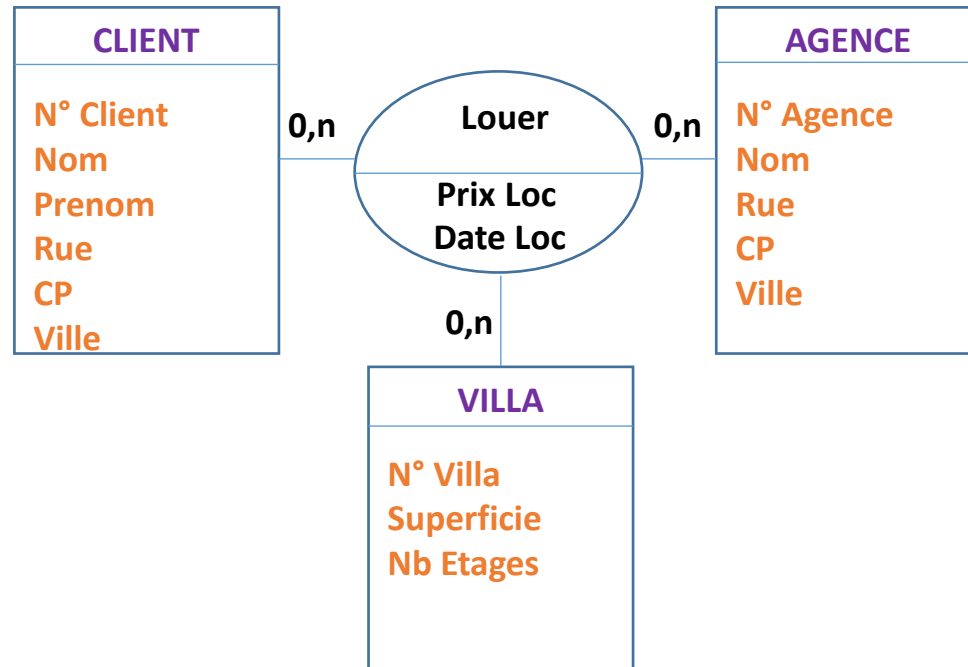
L'attribut num\_carte peut glisser comme clé étrangère dans l'entité Etudiant.

## REGLE N°5

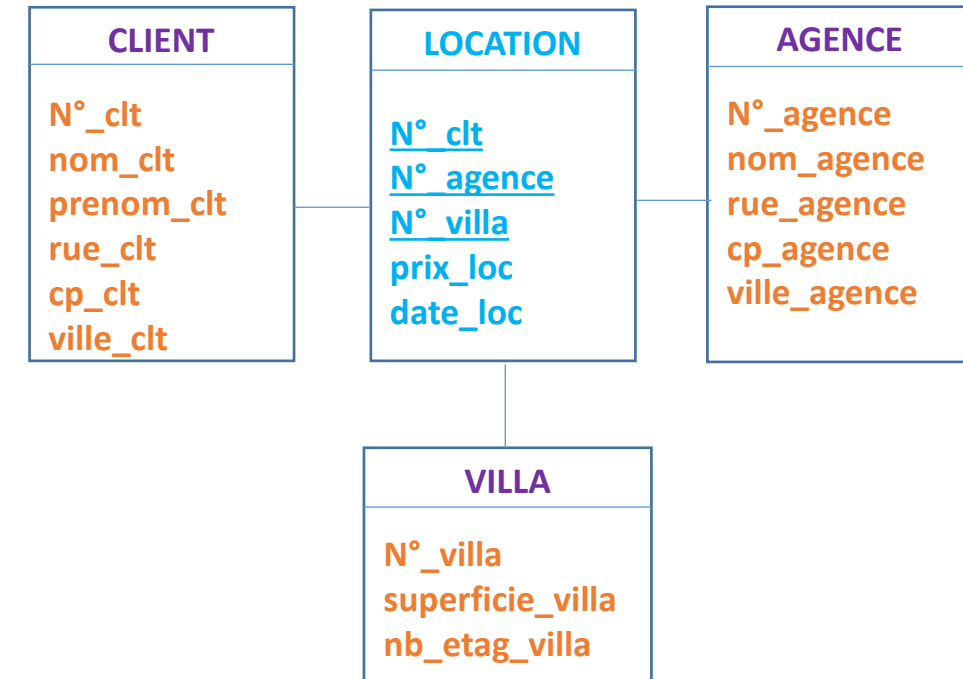
Une association non binaire est traduite par l'ajout d'une table supplémentaire dont la clé primaire est composée d'autant de clés étrangères qu'il y a d'entités liées à l'association.  
Les attributs de l'association deviennent les colonnes de cette nouvelle table.

## EXEMPLE

## ENTITES



## TABLES



## PLACEMENT DES ATTRIBUTS D'ASSOCIATION

Lors du passage du modèle conceptuel de données vers le modèle logique de données, les attributs d'association se placent dans les tables du modèle logique de données selon le type de l'association.

➡ **Si l'association binaire est de type n:m**

Les attributs de l'association deviennent les attributs de la table d'association.

➡ **Si l'association binaire est de type 1:n**

Les attributs de l'association se placent dans la table qui représente l'entité où la cardinalité est égale à 1,1

➡ **Si l'association binaire est de type 1:1**

Les attributs de l'association se placent dans l'une ou l'autre des tables qui représentent les deux entités liées par l'association.

# MODELE PHYSIQUE DE DONNEES (MPD)

## Passage du MLD au MPD

### DEFINITION

**Le modèle physique de données (MPD) est l'implémentation du modèle logique de données (MLD) par un logiciel.**

**Au cours de cette implémentation, nous précisons notamment la manière dont sont stockées les données en définissant le type et la taille de chaque donnée.**

**Cela se traduit par la création d'une base de données dans un SGBD(Oracle, SQL Server, MySQL ou Access par exemple) à l'aide du langage SQL.**

## ETAPES DE CREATION D'UNE BASE DE DONNEES (1)

- 1) ELABORATION DU MODELE CONCEPTUEL DE DONNEES A PARTIR DU CAHIER DES CHARGES.
  - a) Etablir le dictionnaire de données.
  - b) Déterminer les entités du modèle.
  - c) Déterminer les attributs des entités.
  - d) Déterminer les associations entre entités.
  - e) Déterminer les cardinalités entre associations.
  
- 1) TRANSFORMER LE MODELE CONCEPTUEL DE DONNEES EN MODELE LOGIQUE DE DONNEES (MLD) (OU MODELE RELATIONNEL).
  - a) Vérifier les dépendances fonctionnelles entre l'identifiant des entités et les attributs des entités.
  - b) Vérifier les dépendances fonctionnelles pour les attributs d'association.
  
- 2) TRANSFORMER LE MODELE RELATIONNEL DE DONNEES EN MODELE PHYSIQUE DE DONNEES (MPD).
  - a) Créer la base de données et les tables de la base de données avec le langage SQL.
  
- 3) TESTER LA BASE DE DONNEES.
  - a) Créer des enregistrements dans les tables de la base de données.
  - b) Effectuer des requêtes entre les tables.

## ETAPES DE CREATION D'UNE BASE DE DONNEES (2)

**MCD****MLD****MPD**

Entité



Relation



Table

Propriétés



Propriétés



Champs (ou colonnes)


Identifiant



Clé primaire



Clé primaire

CLIENT	
 <b>id_clt</b>	AUTO_INCREMENT
nom_clt	VARCHAR (25)
prenom_clt	VARCHAR (25)



CLIENT( id\_clt, nom\_clt, prenom\_clt)  
id\_clt : clé primaire



```
CREATE TABLE CLIENT(
  id_clt      Int  Auto_increment NOT NULL ,
  nom_clt     Varchar (25) NOT NULL ,
  prenom_clt  Varchar (25) NOT NULL
  ,CONSTRAINT CLIENT_PK PRIMARY KEY (id_clt)
)ENGINE=InnoDB;
```

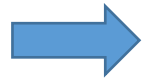
**ATTENTION :**

Syntaxe SQL différente selon le SGBDR choisi



## ETAPES DE CREATION D'UNE BASE DE DONNEES (3)

MCD



MLD



MPD

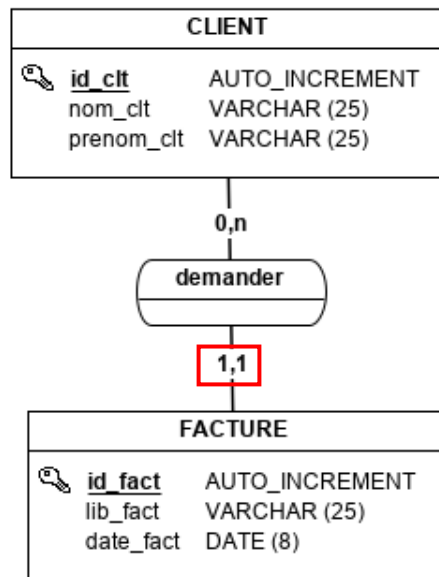
CIF



Clé étrangère



Clé étrangère



CLIENT( id\_clt, nom\_clt, prenom\_clt)  
id\_clt : clé primaire

FACTURE( id\_fact, lib\_fact, date\_fact, id\_clt)  
id\_fact : clé primaire  
id\_clt : clé étrangère en référence à id\_clt de client

```

CREATE TABLE CLIENT(
    id_clt    Int Auto_increment NOT NULL ,
    nom_clt   Varchar (25) NOT NULL ,
    prenom_clt Varchar (25) NOT NULL
    ,CONSTRAINT CLIENT_PK PRIMARY KEY (id_clt)
)ENGINE=InnoDB;
  
```

```

CREATE TABLE FACTURE(
    id_fact    Int Auto_increment NOT NULL ,
    lib_fact   Varchar (25) NOT NULL ,
    date_fact  Date NOT NULL ,
    id_clt     Int NOT NULL
    ,CONSTRAINT FACTURE_PK PRIMARY KEY (id_fact)
    ,CONSTRAINT FACTURE_CLIENT_FK FOREIGN KEY (id_clt)
    REFERENCES CLIENT(id_clt)
)ENGINE=InnoDB;
  
```

ATTENTION :  
Syntaxe SQL différente selon le SGBDR choisi

# **PASSAGE DU MPD AU MLD OU RÉTRO - CONCEPTION**

## Passage du MPD au MLD ou Rétro - conception

### DEFINITION

Dans la majorité des cas, le travail d'un concepteur de bases de données consiste, non pas, à créer une bases de données à partir de zéro mais à effectuer une modernisation de celle-ci. Il est donc nécessaire qu'il s'appuie sur le modèle physique de données existant et suive deux étapes :

- 1) Traduire le modèle physique en modèle relationnel normalisé.
- 2) Suivre les règles de passage du MCD au MLD mais dans le sens inverse.

Ce processus s'appelle la rétro-conception ou le reverse-engineering.  
Il est, pour cela, nécessaire de respecter certaines règles.

## Passage du MLD au MCD

### REGLE N°1

**Chaque table dont la clé primaire ne contient pas de clé étrangère devient une entité dont l'identifiant est la clé primaire de la table et dont les attributs sont les colonnes de la table qui ne sont pas clé étrangère**

## Passage du MLD au MCD

### REGLE N°2

**Chaque table dont la clé primaire est composée exclusivement de clé étrangères qui référencent une seule clé primaire, devient une sous entité ou une sous association (les autres colonnes non clé étrangères de la table deviennent des attributs de cette sous-entité)**

## Passage du MLD au MCD

### REGLE N°3

Chaque table dont la clé primaire est composée exclusivement de clés étrangères qui référencent plusieurs clés primaires, devient une association autour de laquelle toutes les cardinalités maximales valent  $n$ , c'est-à-dire soit une association binaire de type  $n:m$  soit une association ternaire ou plus.  
(Les autres colonnes non clé étrangères de la table deviennent des attributs de l'association).

## Passage du MLD au MCD

### REGLE N°4

**Chaque table dont la clé primaire est composée partiellement de clés étrangères provient soit d'une optimisation qu'il faut défaire soit d'un identifiant relatif d'une entité.**

## Passage du MLD au MCD

### REGLE N°5

**Les colonnes clés étrangères restantes deviennent des associations binaires de type 1:n s'il n'y a pas de contraintes d'unicité ou de type 1:1 s'il n'y a une contrainte d'unicité**



# MERISE 2

## Lien relatif dans un MCD

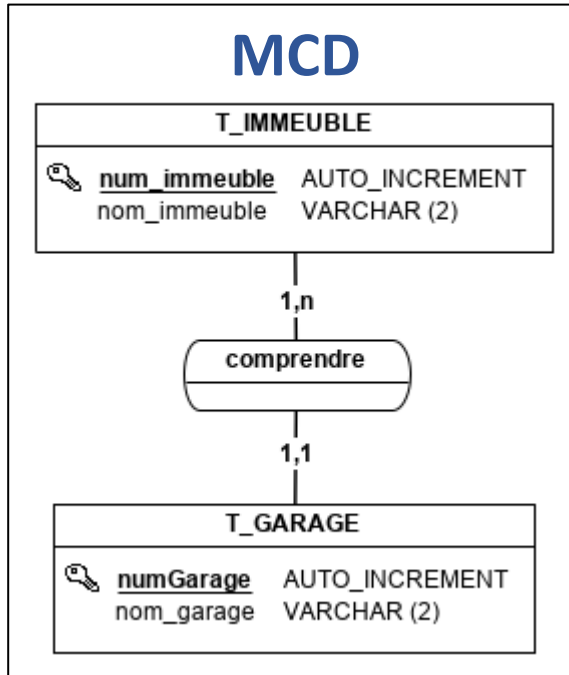
### DEFINITION

Un lien relatif (ou identification relative) entre un entité A et une entité B, permet d'identifier, de façon plus précise, l'entité B par l'intermédiaire d'un attribut supplémentaire provenant de l'entité A.

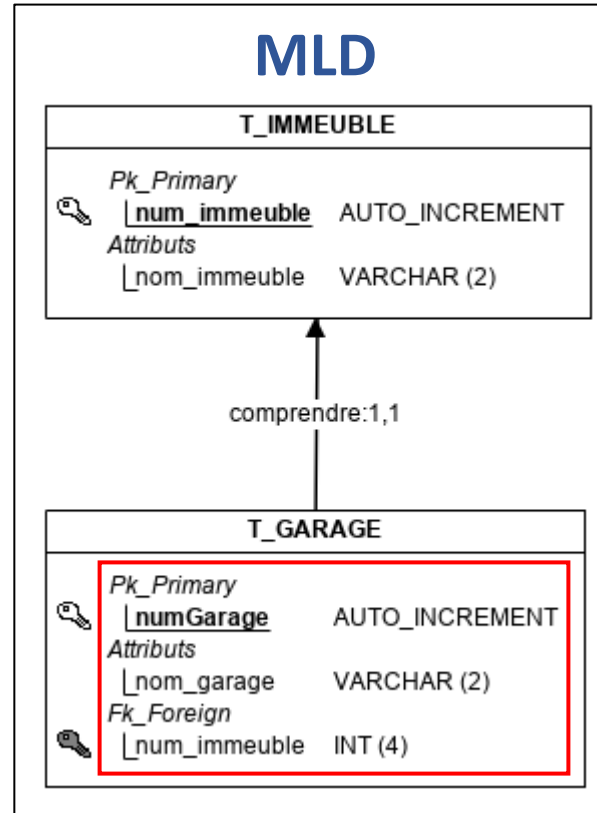
En général, on ajoute l'identifiant de l'entité A à l'identifiant de l'entité B dans l'entité B. La clé primaire de l'entité B est alors formée du couple (identifiantA, identifiantB).

## Lien relatif dans un MCD : Exemple (1)

### MCD



### MLD



Nous souhaitons modéliser la situation selon laquelle des immeubles ont, pour chacun d'entre eux, un ensemble de garage sachant qu'un garage d'un immeuble A peut avoir le même numéro qu'un garage de l'immeuble B.

Avec la modélisation ci-contre, nous ne pouvons pas répondre à la question.

- 1 Si, par exemple, dans la table T\_GARAGE, nous ajoutons un garage N°1 pour l'immeuble B (valeur clé primaire = 1) alors qu'il existe déjà un autre garage N°1 pour l'immeuble A (clé primaire = 1), le SGBDR indique une erreur puisque la contrainte d'unicité n'est pas respectée.

Avec la modélisation du lien relatif, nous allons faire en sorte que chaque garage soit associé à un immeuble de façon plus importante et permettre la réinitialisation des numéros identifiants pour chaque garage.

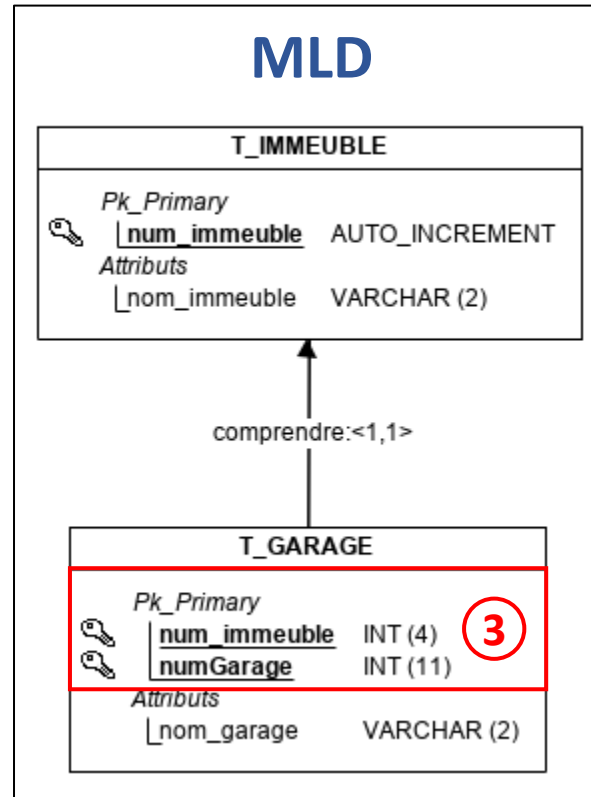
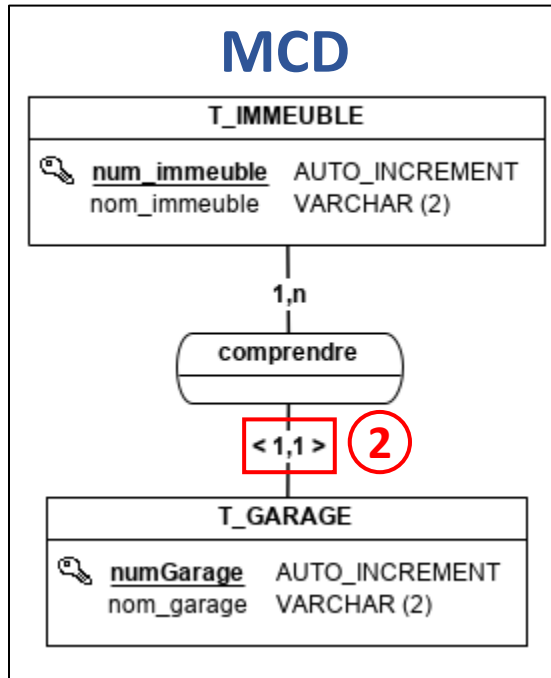
```

1 INSERT INTO `t_garage`(`num_garage`, `nom_garage`, `num_immeuble`)
2 VALUES (1, "G2", 2);
  
```

#1062 - Duplicata du champ '1' pour la clef 'PRIMARY'

1

## Lien relatif dans un MCD : Exemple (2)



- ② Dans le MCD, nous modifions la cardinalité de l'association pour la transformer en lien relatif.
- ③ Dans le MLD, le lien relatif se traduit, maintenant, par le fait que l'identifiant d'un garage est modélisé par le couple (num\_immeuble, num\_garage).
- ④ Ce qui permet de réinitialiser les identifiants des garages dès lors que l'on associe le garage à un immeuble différent.  
Il est, désormais possible de modéliser le fait que plusieurs garages peuvent avoir le même numéro dès lors qu'ils sont situés dans des immeubles différents.

```

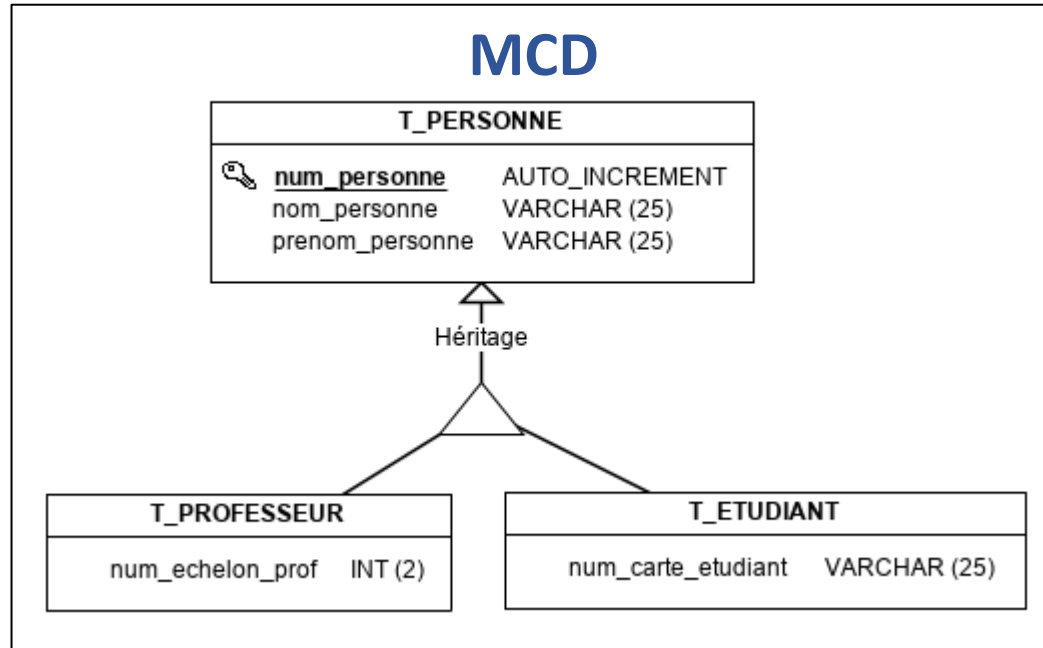
1 INSERT INTO `t_garage`(`num_immeuble`, `numGarage`, `nom_garage`)
2 VALUES (1, 1, "G1"),
3         (1, 2, "G2"),
4         (1, 3, "G3"),
5         (1, 4, "G4"),
6         (2, 1, "G1"),
7         (2, 2, "G2"),
8         (2, 3, "G3"),
9         (2, 4, "G4");
  
```

## L'héritage dans MERISE

### DEFINITION

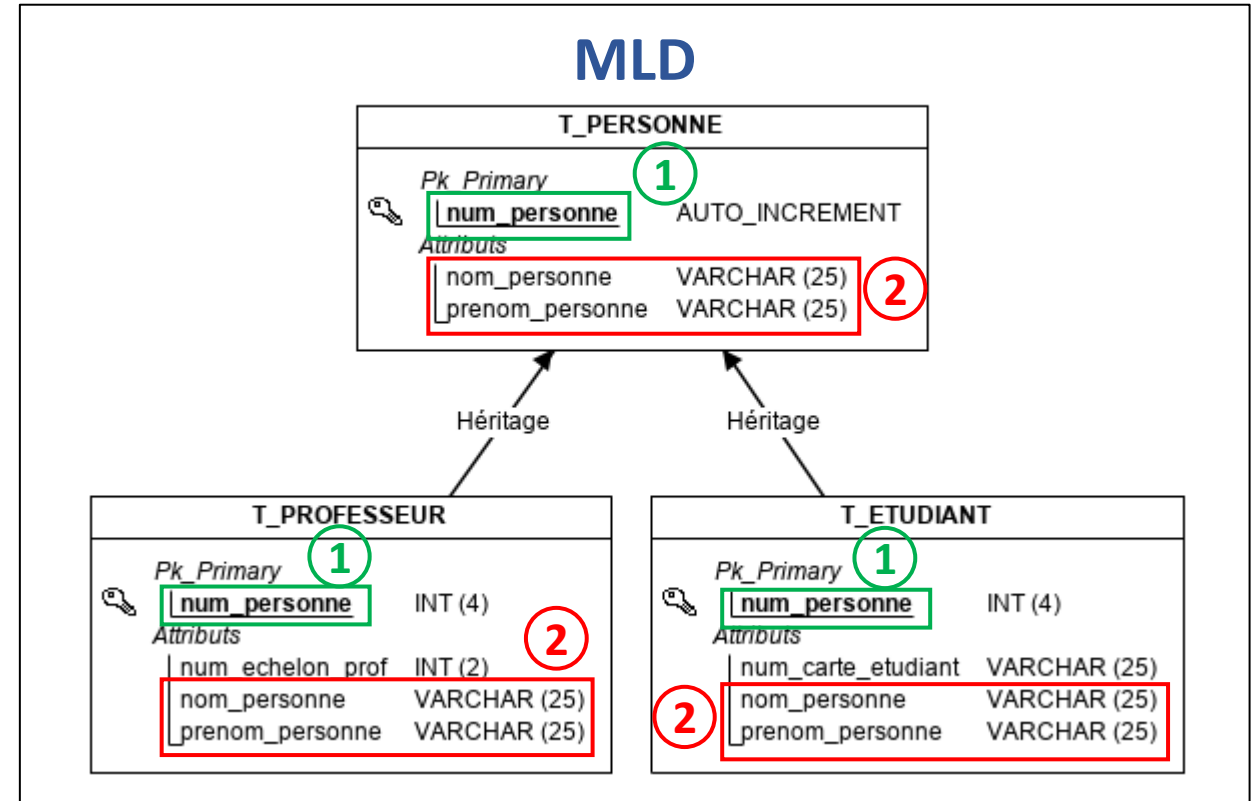
Avec MERISE 2, il est désormais possible de modéliser l'héritage dans un modèle conceptuel de données. Une ou plusieurs entités du modèle peuvent hériter des attributs d'une entité parent.

# L'héritage dans MERISE : Exemple



Les entités T\_PROFESSEUR et T\_ETUDIANT héritent :

- ① De la clé primaire num\_personne de l'entité parent T\_PERSONNE qui représente leur clé primaire respective.
- ② Des attributs nom\_personne et prenom\_personne de l'entité parent T\_PERSONNE.



# WEBOGRAPHIE

## MERISE

### La démarche Merise

- [http://www.compucycles.com/NouveauSite/articles/Merise/Articles\\_07b.htm](http://www.compucycles.com/NouveauSite/articles/Merise/Articles_07b.htm)
- [sigfrance.free.fr/ressources/filebrowser/downloads/SGBD/La methode MERISE.pdf](http://sigfrance.free.fr/ressources/filebrowser/downloads/SGBD/La_methode_MERISE.pdf)

### Le MCD

- [philippe.guezlou.free.fr/mcd/mcd.htm](http://philippe.guezlou.free.fr/mcd/mcd.htm)
- [courspc.pagesperso-orange.fr/formation/telecharg/mcd.pdf](http://courspc.pagesperso-orange.fr/formation/telecharg/mcd.pdf)
- [richard-ostrowski.eu/SILNTI/ressources/Merise-SupportMCD.pdf](http://richard-ostrowski.eu/SILNTI/ressources/Merise-SupportMCD.pdf)
- [cterrier.com/plateformedemo/exercices/sbdr/00-intro-exercice-sbdr.pdf](http://cterrier.com/plateformedemo/exercices/sbdr/00-intro-exercice-sbdr.pdf)

### Règle de passage du MCD au MLD

- [steveostine.free.fr/pdf/Rgles de transformation du MCD au MLD.pdf](http://steveostine.free.fr/pdf/Rgles_de_transformation_du_MCD_au_MLD.pdf)

### Livres à télécharger sur developpez.com

- Parlez-vous Merise?
- MERISE, 60 affaires classées

## Conception d'une base de données relationnelle

- [cyril-gruau.developpez.com/merise](http://cyril-gruau.developpez.com/merise)
- [lneumann.developpez.com/tutoriels/merise/initiation-merise/](http://lneumann.developpez.com/tutoriels/merise/initiation-merise/)
- <http://www.inf.it-sudparis.eu/cours/bd/?idr=15>
- [Laurent-audibert.developpez.com/Cours-BD/](http://Laurent-audibert.developpez.com/Cours-BD/)

## VIDEOS

### Modèle relationnel et MCD

<https://www.youtube.com/watch?v=VFHVNA8xgK0>

### MERISE 2

<https://www.youtube.com/watch?v=smTFM4GCEgc>