Merise

m2iformation.fr

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

Merise

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

1. Introduction

Le système d'information

Le système d'information (SI), peut être défini comme étant l'ensemble des moyens humains, matériels et immatériels mis en œuvre afin de gérer l'information au sein d'une unité, une entreprise par exemple.

Fonction d'un SI

Le SI possède quatre fonctions essentielles :

- **la saisie** ou collecte de l'information
- **la mémorisation** de l'information à l'aide de fichier ou de base de données
- **le traitement** de l'information afin de mieux l'exploiter (consultation, organisation, mise à jour, calculs pour obtenir de nouvelles données...)
- **la diffusion** de l'information

Histoire

- Merise est une méthode d'analyse, de conception et de gestion de projet informatique très utilisée dans les années **1970** et **1980**
- Cette méthode est adaptée pour la gestion des projets internes aux organisations, se limitant à un **domaine précis**
- Merise est une méthode française qui se base sur **le modèle relationnel** de **Edgar Frank Codd**

Caractéristiques

Merise possède un certain nombre de modèles (ou schémas) qui sont répartis sur trois niveaux :

- le niveau conceptuel
- le niveau logique ou organisationnel
- le niveau physique

Étapes de conception

1. Dictionnaire de données
2. Le modèle conceptuel de données
3. Le modèle logique de données
4. Le Langage SQL

Outils de conception

Il existe plusieurs outils pour concevoir à l'aide de la méthode Merise, initialement les schémas étaient réalisés à la main.

- Looping-mcd: outils de modélisation gratuit, très complet
- Mocodo: text to diagram

2. Modélisation d'une base de données au niveau conceptuel

Introduction

- Le modèle conceptuel des données (**MCD**) est une représentation graphique et structurée des informations mémorisées par un SI
- Le MCD est basé sur deux notions principales : **les entités** et **les associations**, d'où sa seconde appellation : le schéma **Entité/Association** (ER Diagram)

Étapes d'élaboration du MCD

L'élaboration du MCD passe par les étapes suivantes :

- la mise en place de règles de gestion
- l'élaboration du dictionnaire des données
- la recherche des dépendances fonctionnelles entre ces données
- l'élaboration du MCD (création des entités puis des associations puis ajout des cardinalités)

Les règles de gestions

- La première étape de la création du modèle consiste à recueillir le besoin
- A partir de ce besoin, il suffit ensuite d'établir des règles de gestion qui s'appliqueront lors de la modélisation
- Prenons l'exemple du SI d'une bibliothèques, les règles pourraient être :
  - pour chaque livre, on doit connaître le titre, l'année de parution, un résumé et le type
  - Un livre peut être rédigé par aucun ou plusieurs auteurs

Le dictionnaire de données

- Le dictionnaire des données est un document qui regroupe toutes les données qui seront conservées dans la base (et qui figureront donc dans le MCD)
- On le représente généralement sous forme de tableau

Attribut	Description	Type	Taille
ClientNom	Nom du client	chaîne	50
ClientPrenom	Prénom du client	chaîne	50

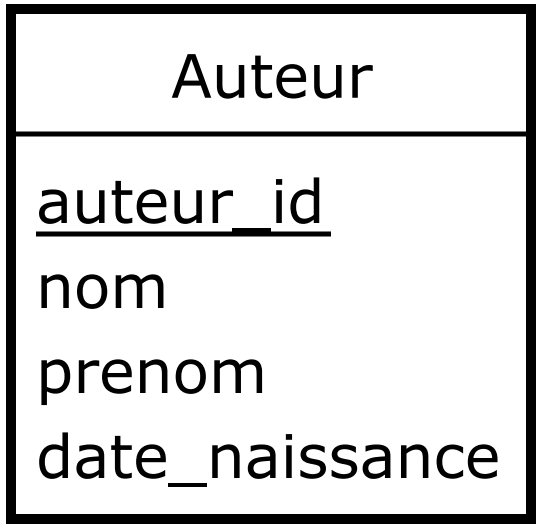
- Les données du dictionnaires ne sont généralement ni calculées ni composées

Les entités

- Chaque entité est unique et est décrite par un ensemble de propriétés encore appelées **attributs** ou **caractéristiques**
- Une des propriétés de l'entité est **l'identifiant**, elle doit posséder une valeur unique vis à vis de l'entité et doit être source des dépendances fonctionnelles avec toutes les autres propriétés de l'entité
- Bien souvent, on utilise une donnée de type entier qui s'incrémente pour chaque occurrence, ou encore un code unique spécifique du contexte

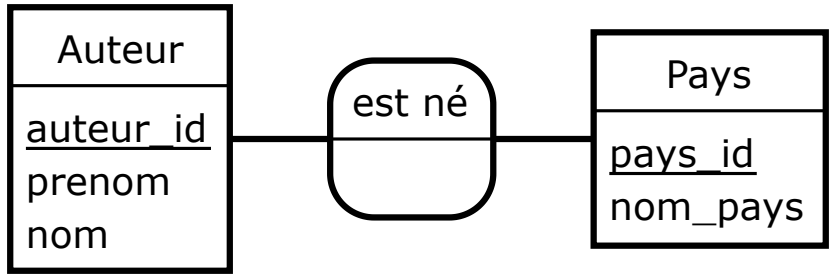
Représentation d'une entité

Les entités sont représentées sous forme de rectangles avec en entête de le nom de l'entité suivi des attributs qui la compose



Les associations

- Une association définit un **lien sémantique** entre **une** ou **plusieurs** entités
- En effet, la définition de liens entre entités permet de traduire une partie des règles de gestion qui n'ont pas été satisfaites par la simple définition des entités
- le nom de l'association est **un verbe** définissant le lien entre les entités



Les cardinalités

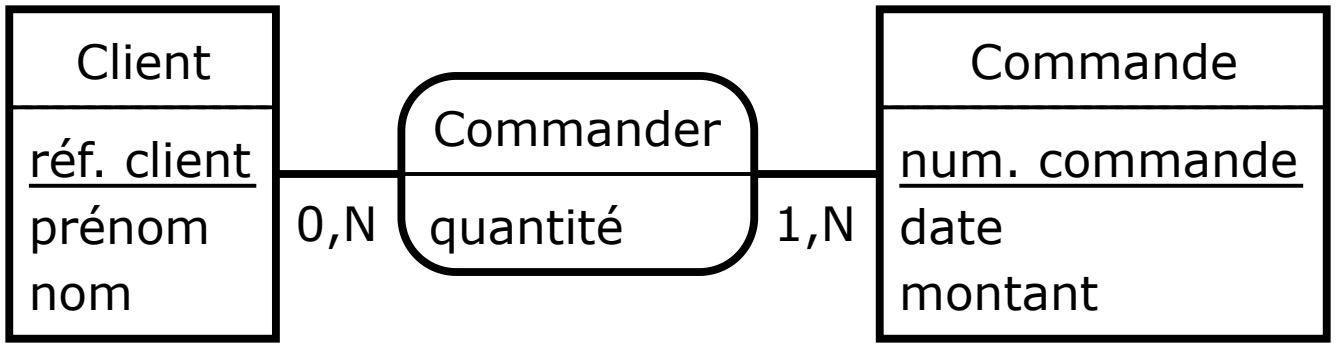
- Les cardinalités expriment le nombre de fois ou l'occurrence d'une entité participe aux occurrences de la relation
- Dans notre exemple on peut se poser les questions suivantes :
  Quel est le minimum de personnes nés dans un pays ?
  Quel est le maximum de personnes nés dans un pays ?

Détail sur les cardinalités

- La cardinalité minimale (0 ou 1) exprime le nombre de fois minimum qu'une occurrence d'une entité participe aux occurrences d'une relation
- La cardinalité maximale (1 ou n) exprime le nombre de fois maximal qu'une occurrence d'une entité participe aux occurrences de la relation
- Si le maximum est connu, il faut inscrire sa valeur. Par exemple, si dans les règles de gestion le client n'a le droit de commander qu'un maximum de 3 articles

Les relations porteuses

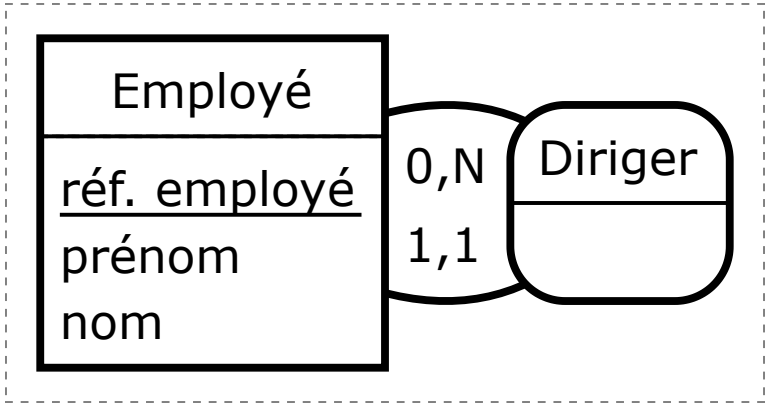
Une relation est dite porteuse lorsqu'elle contient des propriétés.



Nous pouvons interpréter ce schéma de la façon suivante : Le client X a commandé la quantité Y d'articles Z. Si nous désirons connaître la date d'achat, il nous suffit de créer une entité Date à la relation Commander.

Les relations reflexives

- Une relation réflexive est une relation d'une entité sur elle même.
- Par exemple, on désire modéliser le fait qu'un employé peut diriger d'autres employés :



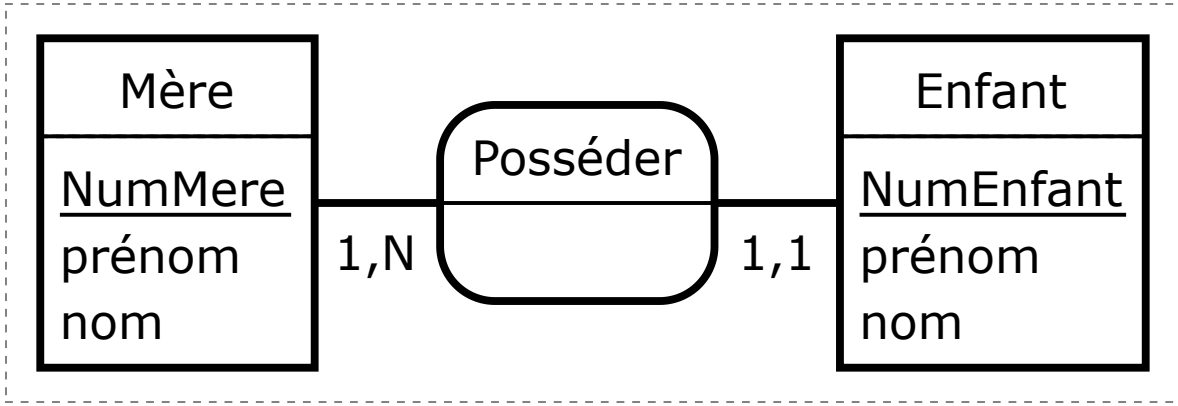
Règles d'usage

- Toute entité doit **comporter un identifiant**
- Toutes les propriétés de l'entité **dépendent fonctionnellement de l'identifiant**
- Le nom d'une propriété ne doit **apparaître qu'une seule fois** dans le modèle conceptuel des données
- Les propriétés résultantes d'un calcul ne doivent **pas apparaître** dans le modèle conceptuel des données

Entité forte et entité faible

Entité forte: Une entité forte est une entité qui, disposant de son identifiant, peut être considérée de façon isolée.

Entité faible: Une entité faible est une entité qui ne peut être considérée qu'en association avec une autre entité.



Dans ce cas l'entité forte est l'entité Mères et l'entité faible est l'entité Enfants.

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

3. Modélisation d'une base de données au niveau logique

Introduction

- Le **M**odèle **L**ogique des **D**onnées (MLD) est la suite normale du processus Merise
- Son but est de nous rapprocher au plus près du modèle physique
- Pour cela, nous partons du Modèle Conceptuel des Données et nous lui enlevons les relations, mais pas n'importe comment, il faut en effet respecter certaines règles

Règle d'écriture du MLD

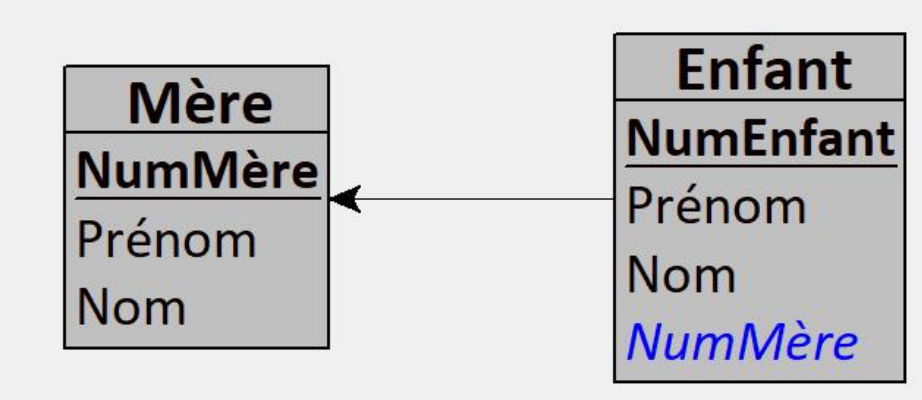
- Chaque **ligne** représente une **table**
- C’est toujours **le nom de la table** qui est écrit **en premier**
- Les champs sont **listés entre parenthèses** et **séparés** par des **virgules**
- Les **clés primaires** sont **soulignées** et placées **au début** de la liste des champs
- Les clés étrangères sont préfixées par un **dièse** `#`

One to Many

- Dans le cadre d'une relation `0,N - 1,1` ou `1,N - 0,1` (un à plusieurs) il est nécessaire de supprimer la relation, cela se réalise de façon tout à fait mécanique
- L'entité ayant la cardinalité de type `1,1` ou `0,1` (**relation faible**) absorbe l'identifiant de **l'entité la plus forte** (`0, N` ou `1, N`)
- Cet identifiant est alors appelé la **clé étrangère**

MLD One to Many

- Enfant (NumEnfant, prénom, nom, #NumMère)
- Mère (NumMère, prénom, prénom)



Représentation des tables

Mère

NumMère	Prénom	Nom
1	Sandra	Bibelot
2	Michelle	Bird

Enfant

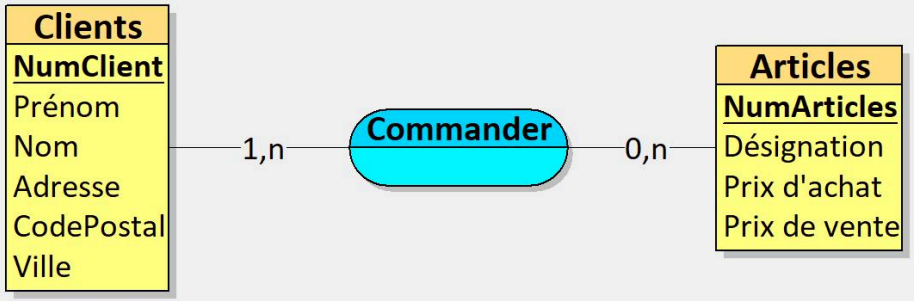
NumEnfant	Prénom	Nom	NumMère
11	Thomas	Herdot	1
14	Sonia	Blade	1

Many to Many

- Dans le cas où la cardinalité maximale est `N` de chaque côté de la relation, celle-ci **se transforme en entité** et **absorbe les identifiants** de chaque entité reliée
- Les identifiants ainsi absorbés forment la nouvelle clé de l'entité
- Cette nouvelle clé est donc formée par la **concaténation des clés étrangères des entités reliées**

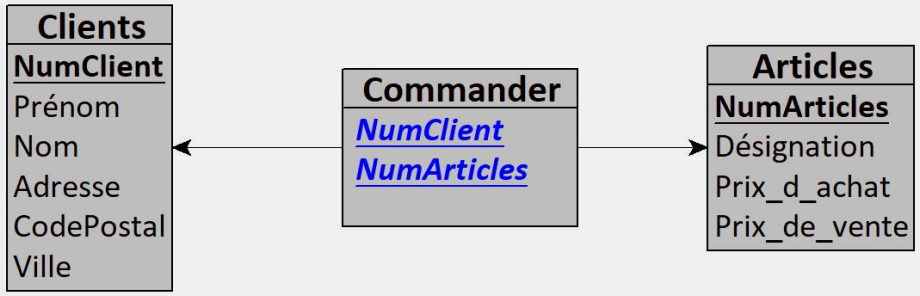
MCD Many to Many

Le modèle conceptuel de données d'une relation many-to-many ressemble généralement à ça :



Modèle logique Many-To-Many

- Clients (NumClient, Prénom, Nom, Adresse, CodePostal, Ville)
- Articles (NumArticle, Désignation, PrixAchat, PrixVente)
- Commander (#NumClient, #NumArticle)



One to One

- Une relation de type `1,1 -- 1,1` est théoriquement une erreur de conception (cela voudrait dire que c'est la même entité)
- Il est possible de concevoir une relation `1,1 --- 0,1` d'un point de vue pratique et lisibilité
- Par exemple, une **personne** peut posséder un **passeport** (0,1), et un passeport doit être possédé par **au moins une personne** (1,1)

Règles pour MCD vers MLD

- L'entité qui possède la cardinalité maximale égale à 1, recevra l'identifiant ou les identifiants des entités ayant les cardinalités maximales les plus fortes
- Les relations ayant toutes leurs entités reliées avec des cardinalités maximales supérieures à 1, se transformeront en entité en absorbant les identifiants des entités jointes
- Toute relation porteuse de propriétés (association) se transformera en entité et absorbera comme clé étrangère les identifiants des entités qui lui sont liées

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

4. Les formes normales

Introduction

Pour être parfaites, les relations doivent respecter certaines règles qu'on appelle **les formes normales**. Cette théorie a été élaborée par E.F. Codd en 1970 pour éviter:

- Les problèmes de mise à jour
- La suppression des redondances d’informations
- La simplification de certaines contraintes d’intégrité

Les 4 premières formes sont les plus importantes

1FN - Première forme normale

Une relation est en première forme normale si :

- Tous les attributs ne contiennent qu'une seule valeur atomique (non divisible)
- Les attributs ne contiennent pas de valeurs répétitives

Exemple 1FN

Clients (NumCli, Nom, Prénom, Adresse, Téléphone)

NumCli	Nom	Prénom	Adresse	Téléphone
1	Baptiste	JeanLuc	25, rue de la forêt 12000 Rodez	0565420000

Cette relation n'est pas en première forme normale, car **Adresse n'est pas atomique**

Correction: Clients (NumCli, Nom, Prénom, Adresse, CodePostal, Ville, Téléphone)

2FN - Deuxième forme normale

Une relation est en deuxième forme normale si :

- Elle est en première forme normale
- Si tous les attributs nonclés ne dépendent pas d'une partie d'un sous-ensemble de la clé primaire

Autrement dit, un attribut non identifiant ne dépend pas d'une partie de l'identifiant mais de tout l'identifiant

Exemple 2FN

Commande (Numcli, CodeArticle, Date, Qté commandée, Désignation)

- Cette relation estelle en première forme normale ? **Oui**.
- Est-elle en deuxième forme normale ? **Non**, car la propriété *Désignation* ne dépend pas intégralement de la clé (Numcli, CodeArticle, Date)

Connaissant {1, Art1, 28/02/2009} pouvons nous connaître de façon sûre et unique « **Bocal d'un kilo de Tripoux** » ? La réponse est évidemment non ! « **Bocal d'un kilo de Tripoux** » ne dépend pas intégralement de la clé {1, Art1, 28/02/2009}

Correction: Commandes(Numcli, CodeArticle, Date, Qté commandée) Articles(CodeArticle, Désignation)

3FN - Troisième forme normale

Une relation est en troisième forme normale si :

- Elle est en deuxième forme normale
- Si toutes les dépendances fonctionnelles par rapport à la clé sont directes (s'il n'y a pas de DF transitives entre les attributs non clé)

Autrement dit, tous les attributs non identifiants doivent dépendre directement de l'identifiant

Exemple 3FN

Commande(NuméroCommande, #CodeClient, Nom client, #RefArticle)

- Est-elle en première forme normale ? Oui
- Est-elle en deuxième forme normale ? Oui
- Est-elle en troisième forme normale ? Non !
- En effet **Nom** client dépend d'une propriété non clé : **CodeClient**

Correction:

Commande(NuméroCommande, #CodeClient, #RefArticle)

Clients(CodeClient, Nom client)

BCNF - Forme normale de Boyce-Codd

Une relation est en forme normale de BOYCE-CODD (BCNF) si et seulement si :

- Elle est en troisième forme normale
- Il n'existe pas de colonne qui soit source d'une dépendance fonctionnelle avec une partie de la clé primaire

En d'autres termes, la BCNF permet d'éliminer les dépendances entre les attributs qui ne font pas partie de la clé vers les parties de la clé. Seules les tables comportant une clé primaire composée de plusieurs colonnes sont concernées.

Exemple BNCF

Considérions la relation: Vaches(Race, Pays, Région), ces dépendances fonctionnelles sont les suivantes:

- Région → Pays
- (Race, Pays) → Région

Cette relation est bien en troisième forme normale car aucun attribut non clé ne dépend d'une partie de la clé ou d'un attribut non clé.
Cependant, on y trouve de nombreuses redondances (des races peuvent venir des même pays et région).

Exemple BNCF

Afin d'éliminer ces redondances, Boyce et Codd ont introduit une forme normale qui porte leur nom :

> “ Une relation est en BCNF si et seulement si les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé détermine un attribut. ”

La relation Vaches pourra être décomposée en deux relations :

- Races (Race, Région)
- Régions (Région, Pays)

4FN - Quatrième forme normale

Une relation est en quatrième forme normale si et seulement si :

- Elle est en 3FN (et non nécessairement la BCNF)
- Les seules dépendances multivaluées élémentaires sont celles dans lesquelles une clé composée détermine un attribut

La quatrième forme est une généralisation de la BCNF.

Cette normalisation conduit parfois à décomposer une relation complexe en deux relations plus simples.

Exemple 4FN

Imaginons la table suivante: EtudiantCoursProf(#CodeEtu, #CodeCours, #CodeProf)

- Dans cet exemple on considère qu'un cours est donné par un seul prof et qu'un prof peut donner plus d'un cours.
- Cette relation contient beaucoup de redondances et le CodeCours implique le NumProf alors qu'elle n'est pas en en 4FN.

Une solution serait: EtudiantCours(#CodeEtu, #CodeCours) CoursProf(#CodeCours, #NumProf)

5FN - Cinquième forme normale

Cette forme normale n'étant quasiment jamais utilisée, en voici juste la définition :

Une association est en cinquième forme normale si et seulement si :

- Elle est en quatrième forme normale
- Elle ne possède pas de dépendance de jointure

La cinquième forme normale est une généralisation de la quatrième forme normale qui nécessite de prendre en compte les dépendances de jointure induites par la connaissance des clés d'une relation.

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

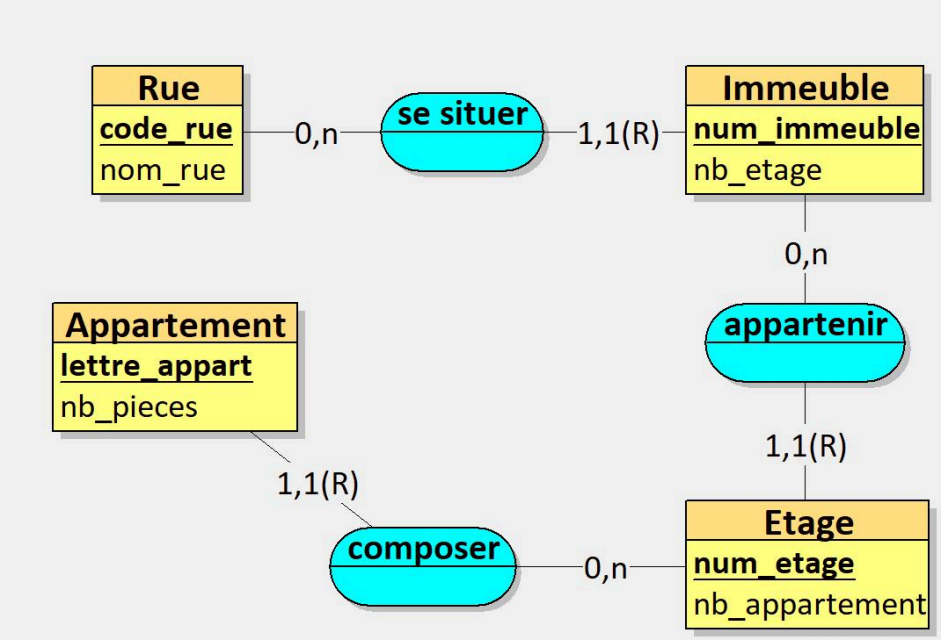
5. Extensions de Merise/2

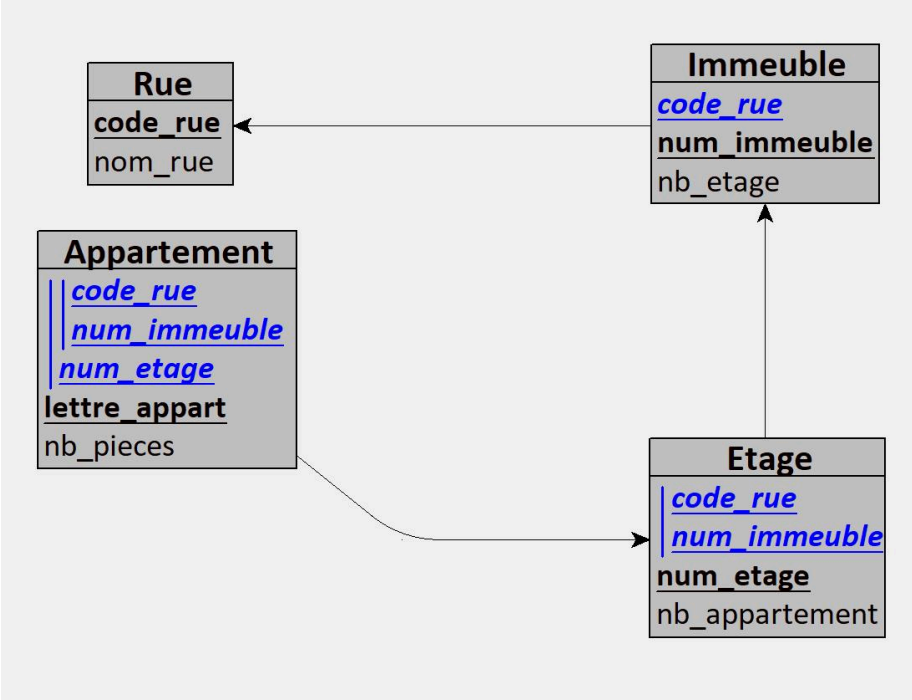
Identifiant relatif

- Certaines entités ont une existence **totalement dépendante** d'autres entités. Dans ce cas nous avons recours à un identifiant relatif.
- L'identifiant relatif est utilisé pour spécifier qu'**une entité est nécessaire pour en identifier une autre**. En d'autres termes, il permet de définir une relation de dépendance entre deux entités.
- L'identification d'un identifiant relatif se traduit par le symbole **(R)** à côté de la cardinalité.
- L'entité faible ne peut exister sans l'entité forte.

Exemple d'identifiant relatif

MCD et MLD visuel montrant le fonctionnement de l'identifiant relatif



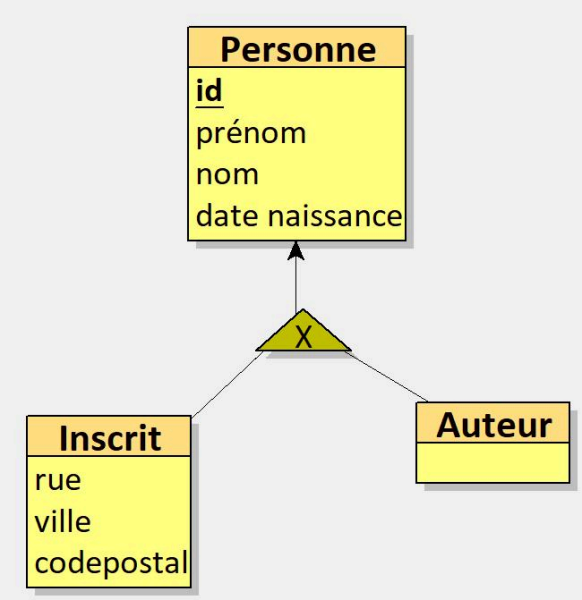


L'héritage et ses limites

- Merise/2 permet aussi de modéliser l'héritage entre les entités
- L'héritage a du sens lorsque plusieurs entités **possèdent des propriétés similaires**
- On parle alors de **généralisation** avec un sur-type (ou entité mère) et de **spécialisation** avec des sous-types (entités filles)
- Il existe plusieurs types d'héritage: **l'héritage par disjonction** (ou exclusion), **l'héritage par couverture** (ou totalité) et enfin **l'héritage par partition** (totalité et exclusion).

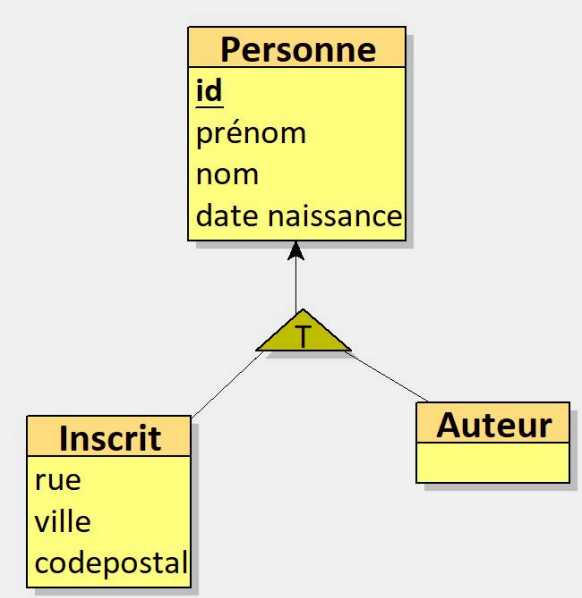
Héritage pas disjonction (exclusion)

- Toutes les occurrences du sur-type ne peuvent se trouver que dans aucun ou un seul sous-type
- Un auteur ne peut pas être également un inscrit et un inscrit ne peut pas être également un auteur



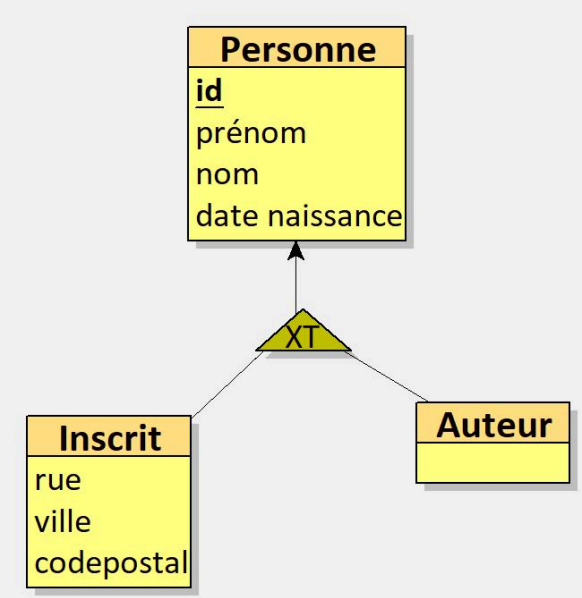
Héritage par couverture (totalité)

- Toutes les occurrences du sur-type se trouvent dans au moins un des sous-types existants
- Dans notre exemple, une personne est forcément un auteur ou un inscrit (ou les deux)



Héritage par partition (totalité et exclusion)

- C'est une combinaison des deux héritages précédents : toutes les occurrences du sur-type se trouvent forcément dans un et un seul des sous-types
- Une personne est soit un auteur, soit un inscrit



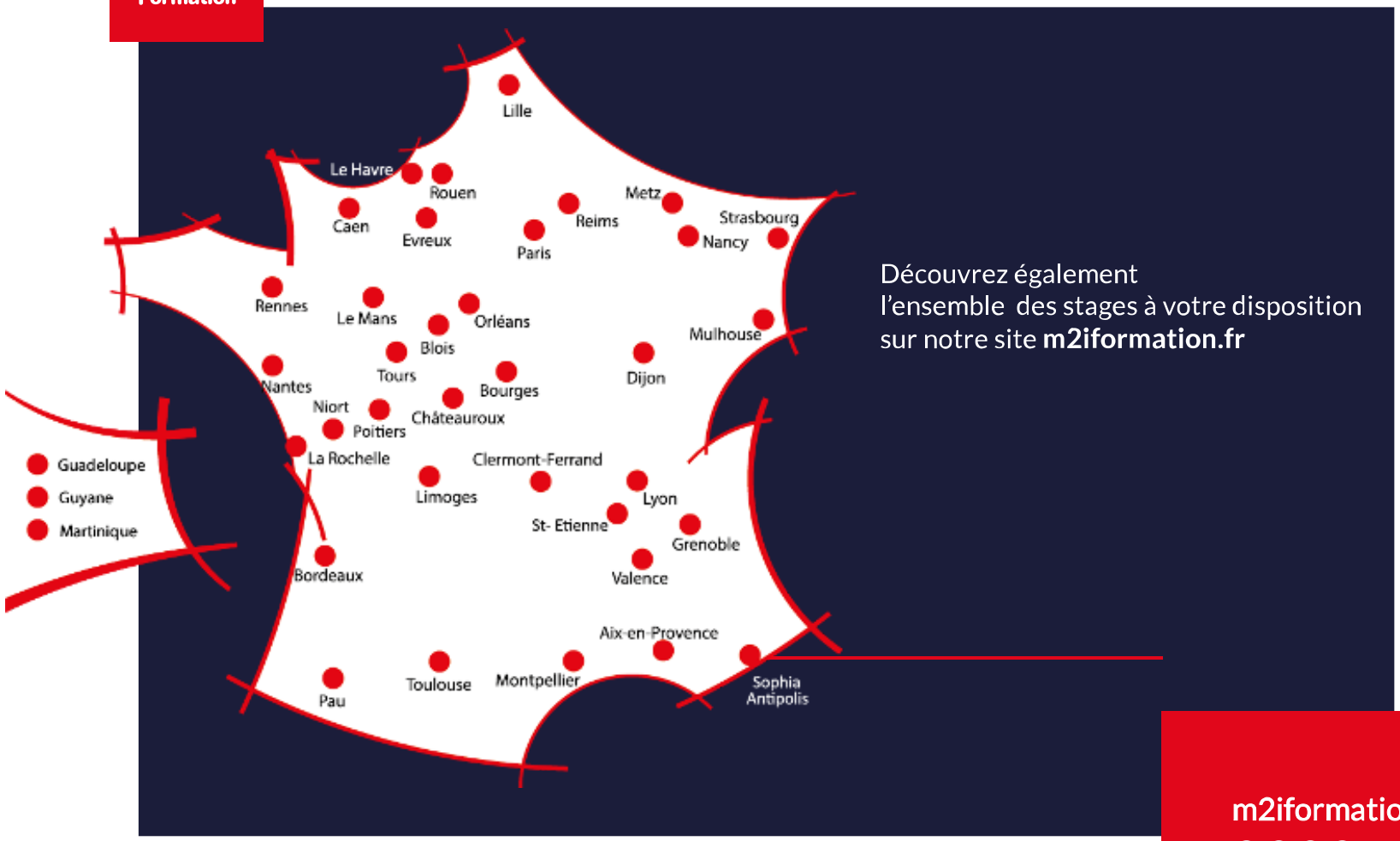
Pour approfondir

1. **Les contraintes entre associations**
   Il existe différentes contraintes qui peuvent exister entre deux ou plusieurs associations. Bien que non implantées au niveau relationnel, ces contraintes qui sont des règles de gestion devront être satisfaites par des traitements supplémentaires.
2. **Les contraintes d'intégrités fonctionnelles** (CIF)
   Dépendances fonctionnelles qui sont directement représentées sur le MCD afin de réduire les identifiants d'associations jugés « trop larges ».

UTOPIOS®
TRANSFORMATION DIGITALE INTELLIGENTE

Merci pour votre attention

Des questions ?



Découvrez également
l'ensemble des stages à votre disposition
sur notre site **m2iformation.fr**

m2iformation.fr