



UNIVERSITÉ
DE REIMS
CHAMPAGNE-ARDENNE

Méthode Merise

Nils Schaefer

nils.schaefer@snicw.fr

Introduction

- ❑ Qu'est-ce qu'une base de données ?
 - Une façon **structurée** de stocker des données
 - Plus pratique que de manipuler des fichiers (CSV, Excel, Word...)
- ❑ Où sont utilisées les bases de données ?
 - Dans tous les systèmes d'information
 - Derrière chaque site Web pour stocker des produits, des clients...
- ❑ Ce cours concerne les bases de données les plus courantes
 - Les bases de données **relationnelles**
- ❑ Comment créer une base de données ?
 - Il faut commencer par **modéliser** la base de données !

Méthode Merise

- ❑ Méthode courante de modélisation de BD
- ❑ Touts les données du système d'information doivent être représentées
- ❑ 3 modèles de données
 - MCD : Modèle Conceptuel des Données
 - MLD : Modèle Logique des Données
 - MPD : Modèle Physique des Données
- ❑ Commençons avec le plus important : le MCD
- ❑ Modèle important même pour les non-informaticiens
- ❑ Comment construire un MCD ?

Entités et attributs

- ❑ Une entité représente un objet physique ou abstrait
- ❑ Une entité contient des attributs
- ❑ Chaque attribut est utilisé pour contenir une donnée

Emplois
Poste
Salaire
Societe

Personnes
Prenom
Nom
DateDeNaissance
VilleDeNaissance
Sexe
Taille
NumeroSecuriteSociale

Maisons
Adresse
CodePostal
Ville
Pays
Superficie
Prix

❑ Le choix des attributs dépend du contexte

❑ Exemple avec une personne

- L'utilisation des propriétés *Nom* et *Prenom* est évidente
- Que pensez-vous de l'utilisation de la propriété *Poids*...
 - dans un contexte médical ?
 - dans un contexte universitaire ?
- Que pensez-vous de l'utilisation de la propriété *DernierDiplome*...
 - dans un contexte médical ?
 - dans un contexte universitaire ?

Attributs

- ❑ Chaque attribut doit contenir une donnée atomique
- ❑ Une donnée atomique correspond à la plus petite donnée que l'on peut avoir à manipuler
- ❑ L'atomicité est relative en fonction de vos besoins
- ❑ Exemple avec l'attribut *NomComplet* pour une personne
 - Est-ce que *NomComplet* avec la valeur Pierre Dupont est atomique ?
 - Non il faudrait utiliser *Nom* et *Prenom*
 - Du coup *Nom* contiendrait Dupont et *Prenom* contiendrait Pierre
 - Quelqu'un pourrait toutefois dire que l'attribut *Nom* seul est atomique par rapport à ses besoins s'il est sûr qu'il n'aura jamais à décomposer les données contenues dans cet attribut
- ❑ Qu'en est-il de l'attribut Adresse ?

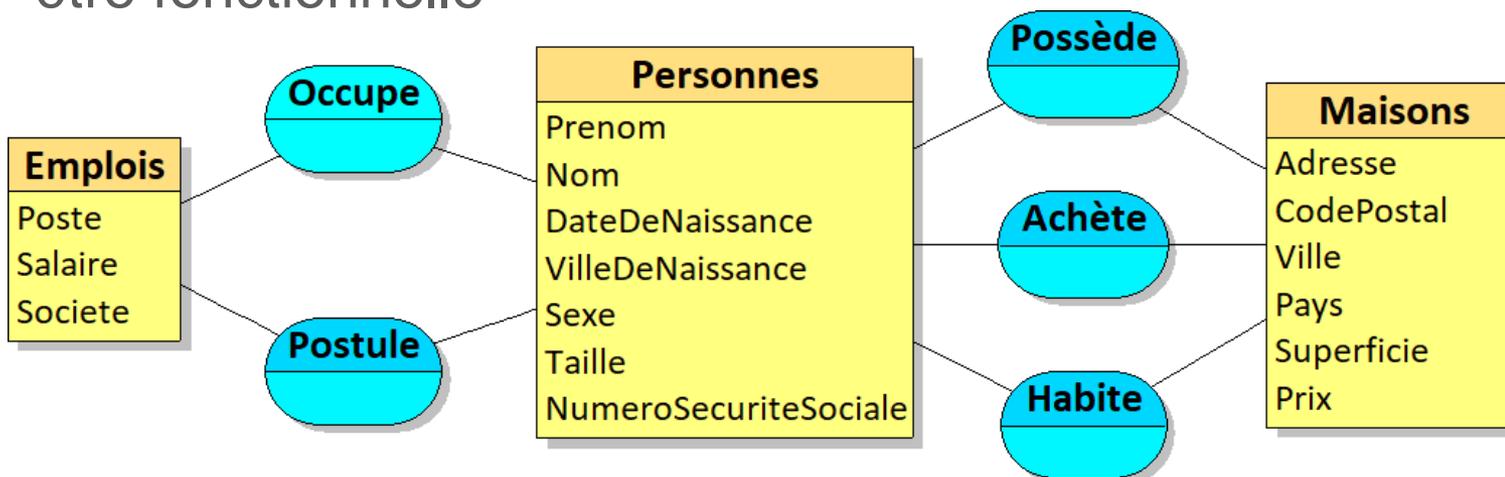
Associations

❑ Une association est un lien logique

- Entre 2 entités (ou plus)

❑ Une association représente des entités qui travaillent ensemble

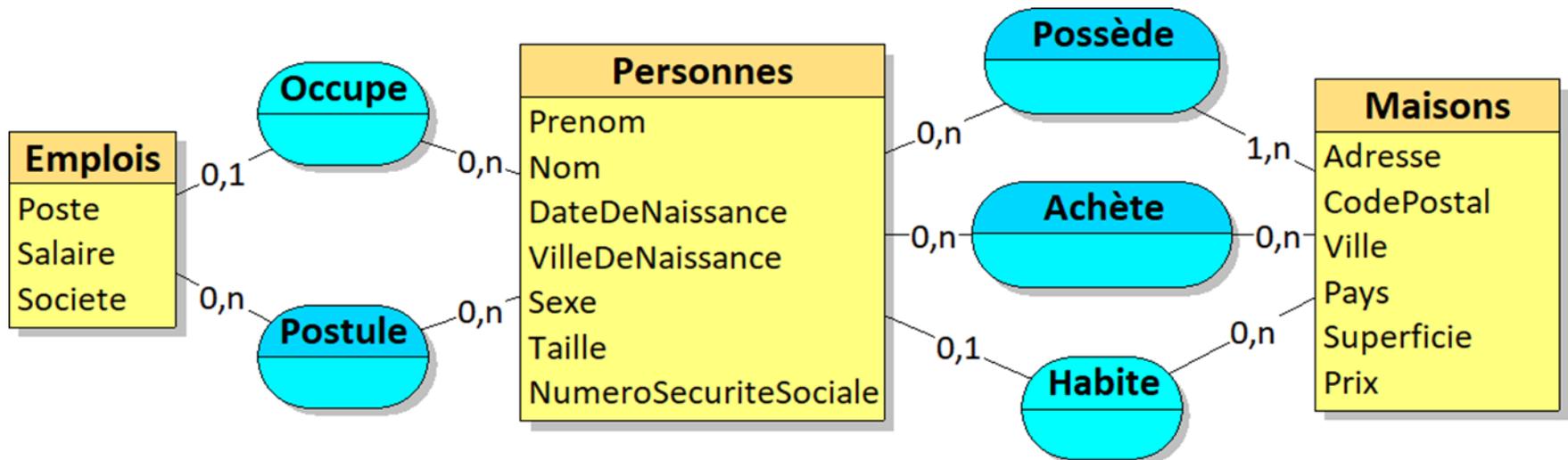
- Une entité a donc besoin d'une ou plusieurs autres pour être fonctionnelle



Associations et cardinalités

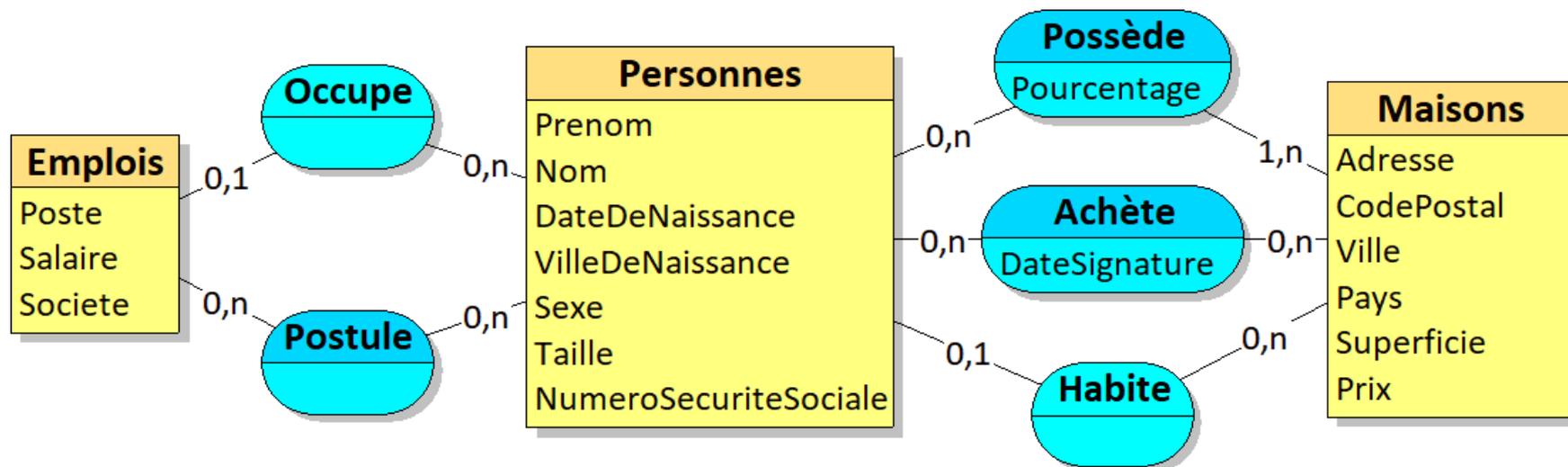
□ Chaque associations possède des cardinalités

- Indique la multiplicité de chaque côté de l'association
- Une cardinalité min et une cardinalité max pour chaque entité liée
- La valeur n (infini) ne peut pas être utilisée pour une cardinalité min



Associations, cardinalités et attributs

- ❑ Chaque association peut avoir des attributs
- ❑ Important quand un attribut ne caractérise aucune des entités liées mais seulement l'association

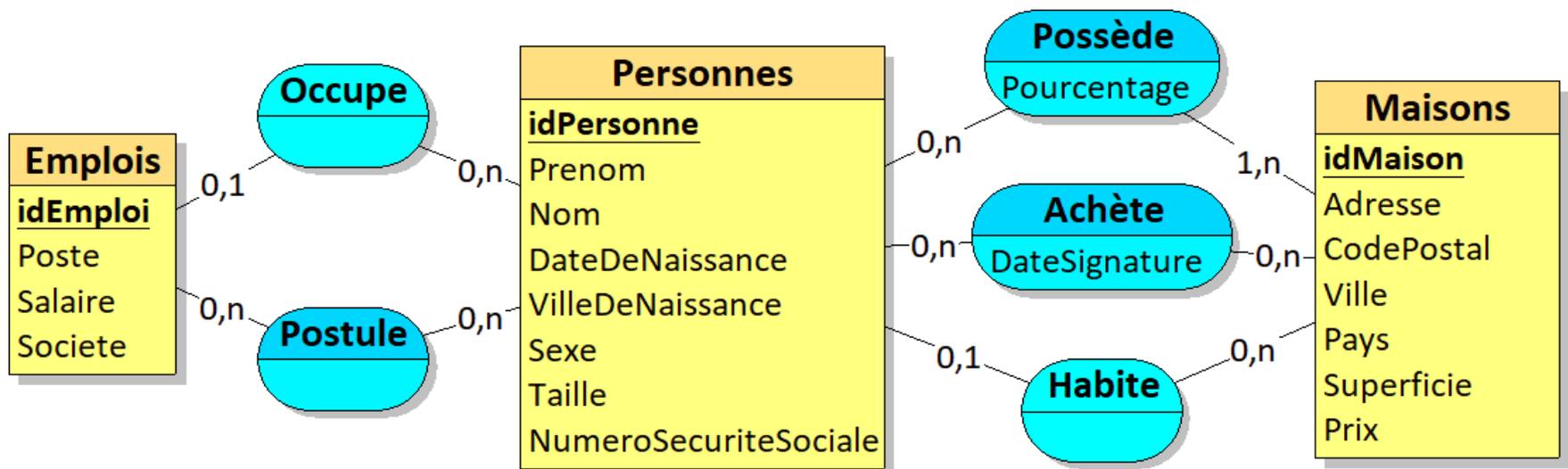


Clés primaires

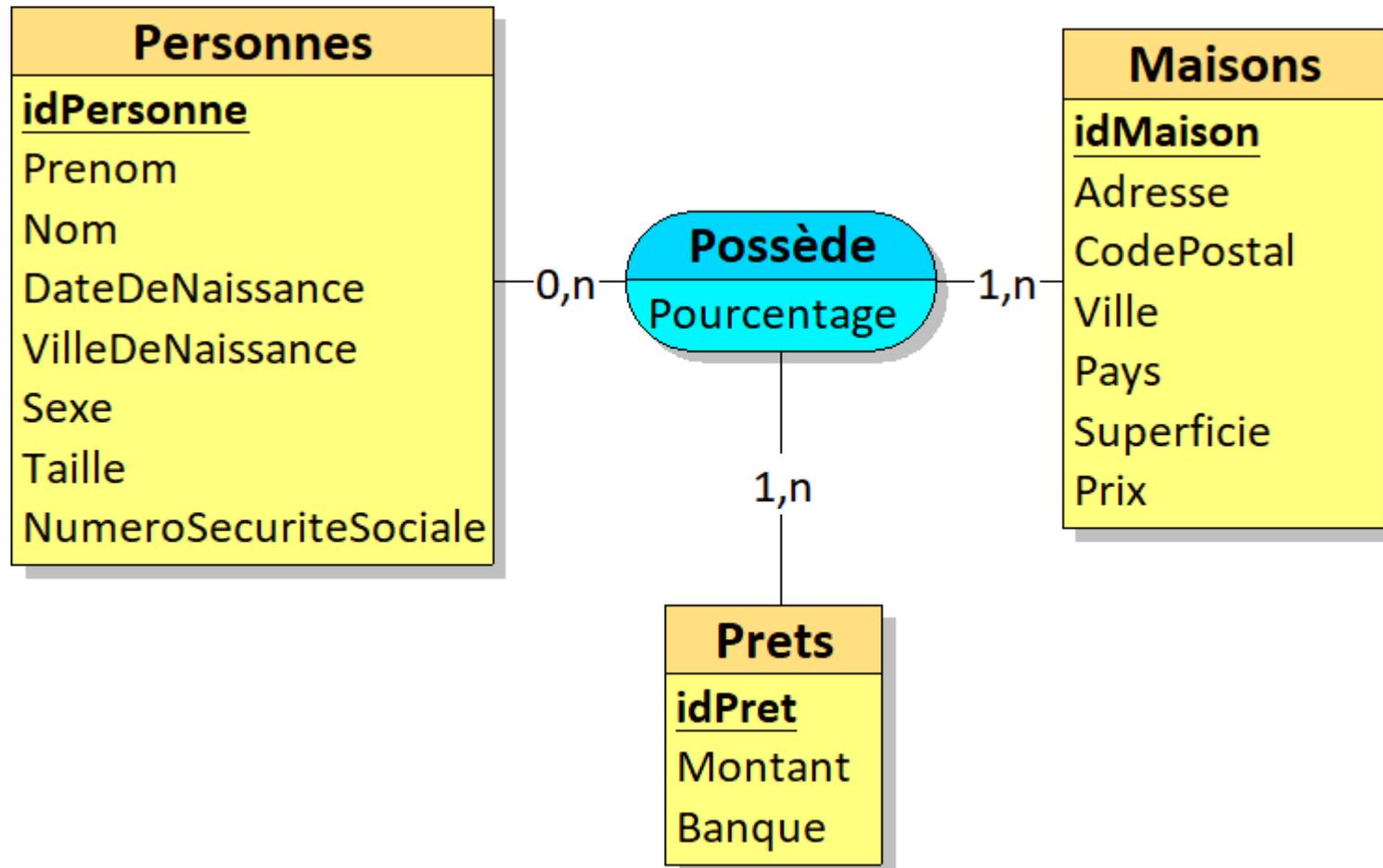
- ❑ Chaque entité a besoin d'une clé primaire
- ❑ Une clé primaire permet d'identifier de manière unique chaque enregistrement de l'entité
- ❑ Une clé primaire est composée d'un ou plusieurs attributs
- ❑ Techniquement il est possible d'utiliser n'importe quel(s) attribut(s) qui convien(nen)t
 - Que pensez-vous du couple *Prenom* et *Nom* pour *Personnes* ?
 - Que pensez-vous de *NumeroSecuriteSociale* pour *Personnes* ?
 - Que pensez-vous du triplet *Adresse*, *CodePostal* et *Ville* pour *Maisons* ?

Clés primaires

- ❑ En pratique il est préférable d'utiliser une clé primaire numérique arbitraire auto-incrémentée
- ❑ Moins d'ambigüité, moins de problèmes...



Association entre 3 entités



Structure et données

- ❑ Ne pas confondre la structure de la base de données avec les données elles-mêmes
- ❑ Attribut ≠ donnée
- ❑ Chaque entité va devenir une table qui contiendra les données

Emplois
<u>idEmploi</u>
Poste
Salaire
Societe

Table Emplois			
<u>idEmploi</u>	Poste	Salaire	Societe
1	Directeur	50 000,00 €	Google
2	Secrétaire	30 000,00 €	Microsoft
3	Comptable	40 000,00 €	Netflix
4	Comptable	45 000,00 €	Tesla
5	Consultant	50 000,00 €	Uber

❑ Windows

- Looping MCD
- Site Web : <https://www.looping-mcd.fr>

❑ MacOS

- Analyse SI
- Site Web : <https://launchpad.net/analysesi>
- Nécessite Java (JRE)
- Site Web : <https://www.java.com/fr/download>

Modèle Logique des données

- ❑ MLD connu aussi sous le nom de Modèle Relationnel
- ❑ Ce modèle dérive du MCD
 - Conversion du MCD au MLD
- ❑ Ce modèle représente la structure physique d'une BD
 - Le MLD montre toutes les **tables** nécessaires à la BD
 - Les entités et les associations (certaines) sont converties en tables
- ❑ Les logiciels peuvent produire automatiquement le MLD à partir du MCD...
- ❑ Mais il est important de comprendre comment tout ça fonctionne !

Règle #1

- Chaque entité devient une table

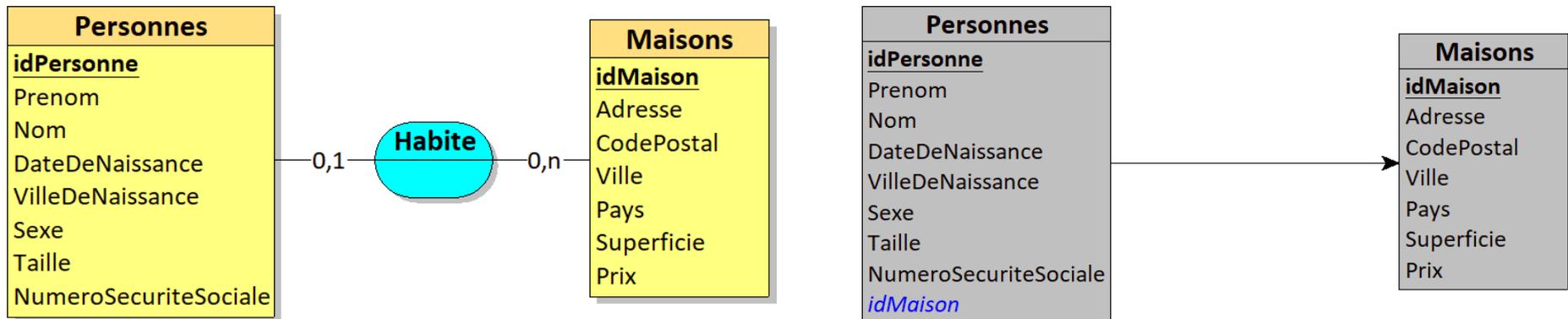
Personnes
<u>idPersonne</u>
Prenom
Nom
DateDeNaissance
VilleDeNaissance
Sexe
Taille
NumeroSecuriteSociale

Personnes
<u>idPersonne</u>
Prenom
Nom
DateDeNaissance
VilleDeNaissance
Sexe
Taille
NumeroSecuriteSociale

Règle #2

❑ Pour une association 1-n, l'association disparaît et du côté du 1 l'entité reçoit une copie de la clé primaire de l'autre table sous forme de clé étrangère

- Une association 1-n veut dire 1 = cardinalité max d'un côté et n = cardinalité max de l'autre côté



Règle #2

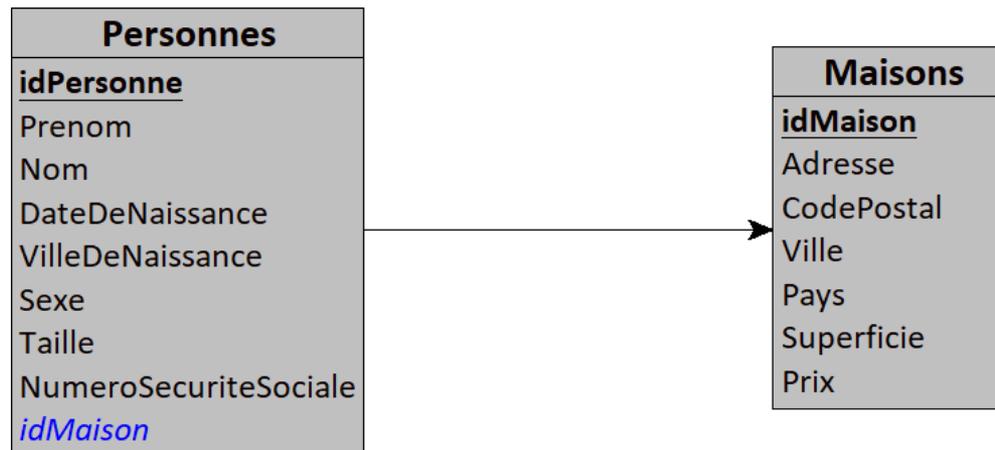
❑ Généralement on utilise un symbole # devant le nom d'une propriété qui est une clé étrangère

❑ Version textuelle du MLD

Personnes(idPersonne,Prenom,Nom,DateDeNaissance,VilleDeNaissance,Sexe,Taille,NumeroSecuriteSociale,#idMaison)

Maisons(idMaison,Adresse,CodePostal,Ville,Pays,Superficie,Prix)

❑ Version visuelle du MLD



Règle #2

❑ Voici un exemple avec des données pour ces 2 tables

Table Personnes				
<u>idPersonne</u>	Prenom	Nom	...	#idMaison
1	John	Doe		3
2	Lisa	Doe		2
3	Pierre	Doe		
4	Amélie	Dupont		3
5	Paul	Dupont		4
6	Sophie	Dupont		

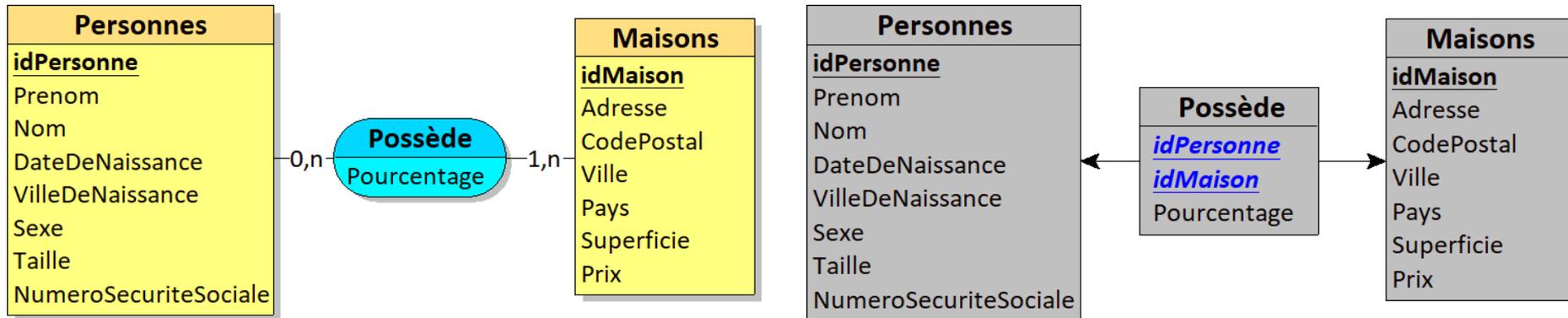
Table Maisons		
<u>idMaison</u>	Adresse	...
1	5 Rue Bleue	
2	15 Rue Principale	
3	10 Place Verte	
4	25 Rue du Pont	

❑ Réalisez ces modifications...

- Sophie Dupont habite avec Paul Dupont
- Pierre Doe habite au 5 Rue Bleue
- Amélie Dupont préfère vivre dans la rue plutôt qu'avec John Doe

Règle #3

- ❑ Pour une association n-n, l'association devient une table et reçoit les clés primaires des 2 entités sous la forme de clés étrangères formant la clé primaire



Règle #3

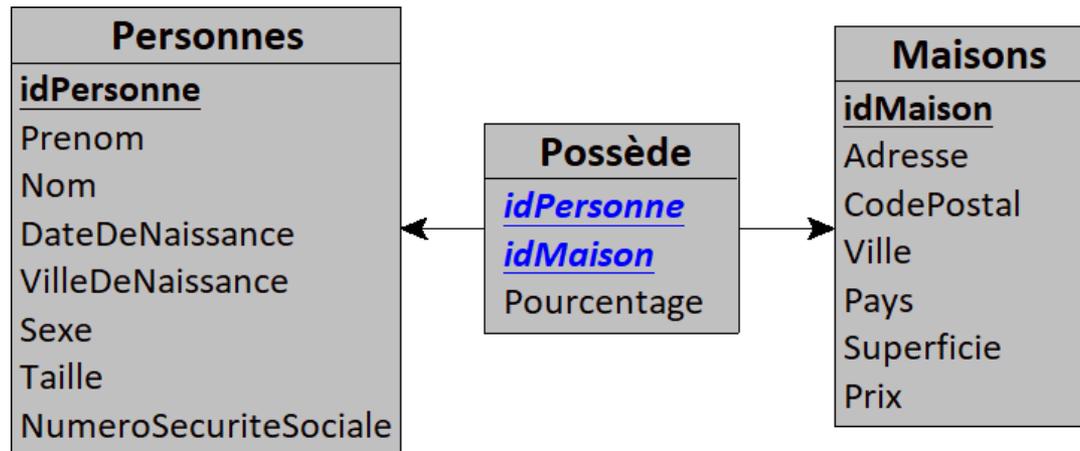
□ Version textuelle du MLD

Personnes(idPersonne,Prenom,Nom,DateDeNaissance, VilleDeNaissance,Sexe,Taille,NumeroSecuriteSociale)

Maisons(idMaison,Adresse,CodePostal,Ville,Pays,Superficie,Prix)

Possède(#idPersonne,#idMaison,Pourcentage)

□ Version visuelle du MLD



Règle #3

❑ Voici un exemple avec des données pour ces 3 tables

Table Personnes			
idPersonne	Prenom	Nom	...
1	John	Doe	
2	Lisa	Doe	
3	Pierre	Doe	
4	Amélie	Dupont	
5	Paul	Dupont	
6	Sophie	Dupont	

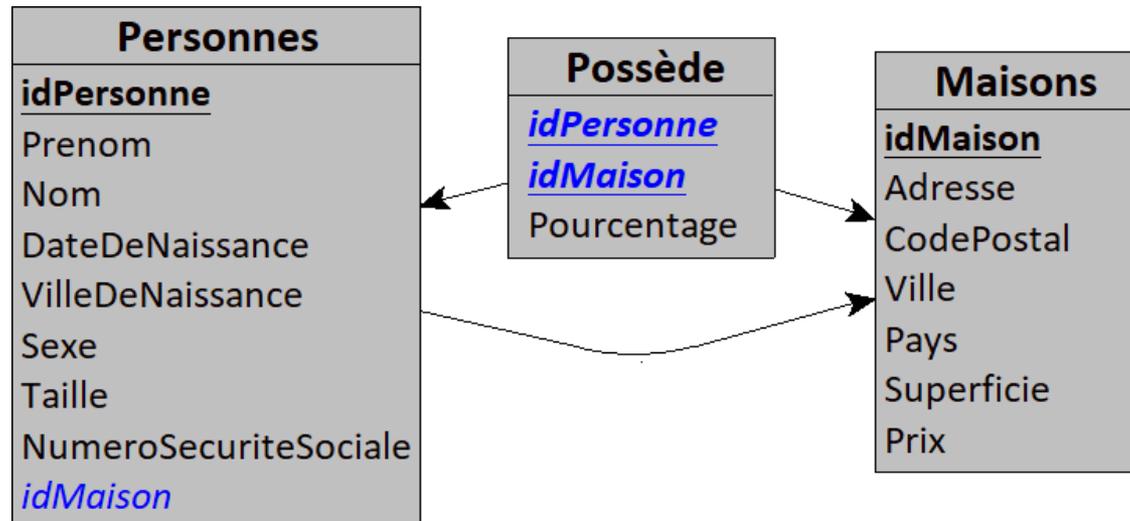
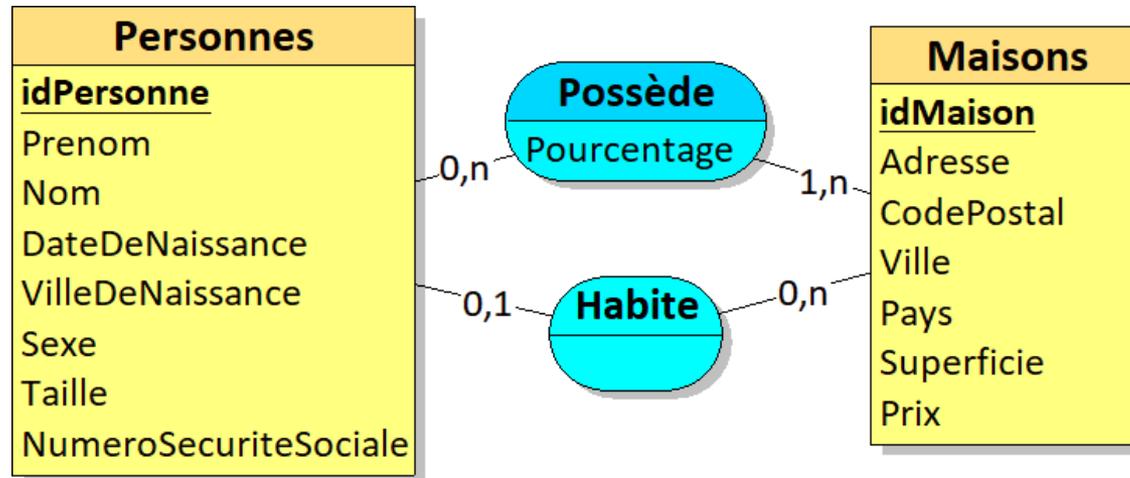
Table Possede		
#idPersonne	#idMaison	Pourcentage
1	3	40%
2	3	60%
3	4	100%
5	2	50%
6	2	50%

Table Maisons		
idMaison	Adresse	...
1	5 Rue Bleue	
2	15 Rue Principale	
3	10 Place Verte	
4	25 Rue du Pont	

❑ Réalisez ces modifications...

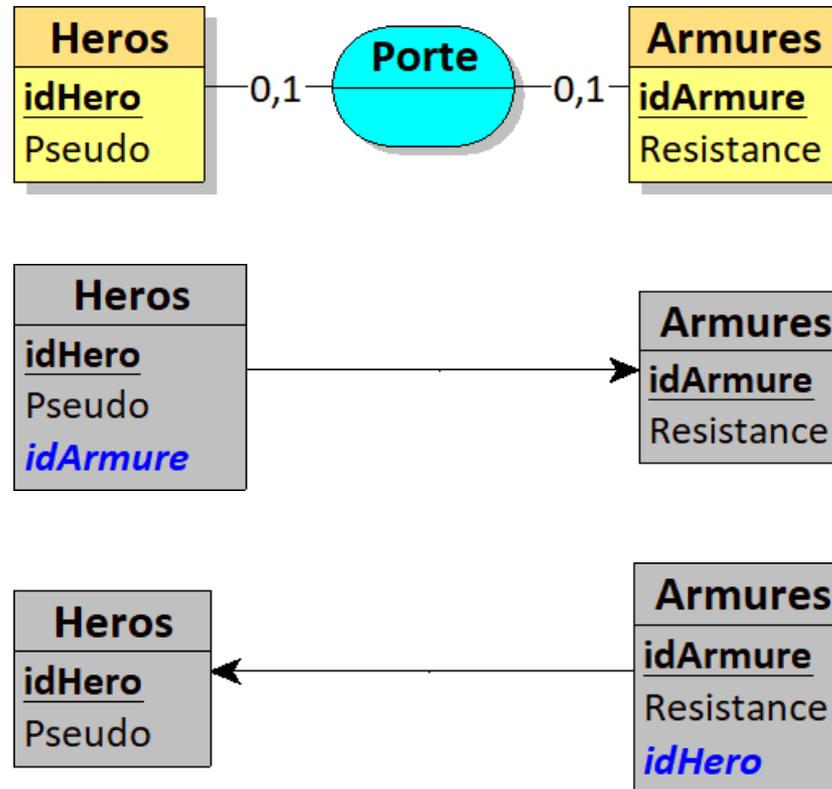
- Pierre Doe vend la moitié de sa maison à Sophie Dupont
- Amélie Dupont achète la maison de Pierre Doe
- John Doe vend 10% de sa maison à Lisa Doe

Exemple avec 2 associations



Règle #4

- Pour une association 1-1, l'association disparaît et il est possible de choisir le côté qui va recevoir une copie de la clé primaire comme clé étrangère



- ❑ Système de Gestion de Base de Données Relationnelles
- ❑ Logiciel qui gère physiquement les bases de données
 - Accès distant possible
- ❑ Gestion structurée des fichiers des BD
 - Comment sont structurés les fichiers ? Peu importe !
- ❑ Gestion des accès aux BD
 - Il faut toujours avoir un utilisateur pour se connecter à un SGBDR
 - Il est possible de définir des droits spécifiques pour les utilisateurs
- ❑ Exemples de SGBDR
 - MySQL, SQL Server, Access, PostgreSQL, Oracle, DB2...

Modèle physique des données

- ❑ MPD
- ❑ Ce modèle dérive du MLD
 - Conversion du MLD au MPD
- ❑ Ce modèle représente la structure physique d'une BD pour un SGBD donné
 - Le MLD montre toutes les **tables** nécessaires à la BD
 - Le MPD donne toutes les informations techniques pour créer les tables sur le SGBDR cible
- ❑ Les logiciels peuvent automatiquement créer le code du MPD à partir du MLD...
- ❑ Mais il faut faire certains choix
 - Donc ça ne peut pas être automatisé à 100%

- ❑ MySQL est un SGBDR très courant
 - En seconde position après Oracle
- ❑ Il fonctionne sur la plupart des systèmes d'exploitation
 - Windows, MacOS and Linux
- ❑ MySQL est une application de type serveur
 - Elle fonctionne en continue et en arrière-plan
 - Par exemple, Access n'est pas une application de type serveur
 - Access est lancé quand on souhaite s'en servir
 - Pas d'accès réseau
- ❑ Nous utiliserons MySQL pour la suite de ce cours

Langage SQL

❑ Structured Query Language

- Langage normalisé

❑ Langage pour parler avec les SGBDR

- Langage de définition de données (DDL : Data Definition Language)
- Langage de manipulation de données (DML : Data Manipulation Language)

❑ Pour créer une BD il faut utiliser SQL en tant que DDL

- Création de la BD elle-même : CREATE DATABASE ou CREATE SCHEMA
- Création des tables : CREATE TABLE

Types de données

□ Chaque propriété est définie par un type de données

- Nombres entiers
 - TINYINT (1 octet) : de -128 à 127
 - SMALLINT (2 octets) : de -32 768 à 32767
 - MEDIUMINT (3 octets) : de -8 388 608 à 8 388 607
 - INT (4 octets) : de -2 147 483 648 à 2 147 483 647
 - BIGINT (8 octets) : de -9 223 372 036 854 775 808 à 9 223 372 036 854 775 807
- Nombres à virgule
 - FLOAT (4 octets)
 - DOUBLE (8 octets)
 - DECIMAL(x,y) : x chiffres max incluant y chiffres max après la virgule
- Pour utiliser uniquement des nombres entiers : UNSIGNED
 - Exemple : UNSIGNED TINYINT (1 octet) : de 0 à 255
 - ...

Types de données

□ Il est possible de stocker du texte

- Textes courts (Prénom, nom, ville, pays...)
 - CHAR(x) : texte avec exactement x caractères
 - VARCHAR(x) : texte avec au maximum x caractères
- Textes longs
 - TINYTEXT : texte avec au maximum 256 caractères
 - TEXT : texte avec au maximum 65 536 caractères
 - MEDIUMTEXT : texte avec au maximum 16 777 216 caractères
 - LONGTEXT : texte avec au maximum 4 294 967 296 caractères
- Attention aux valeurs faussement numériques
 - Un code postal n'est pas un nombre, on utilisera donc CHAR(5) pour un code postal français
 - Un numéro de téléphone n'est pas un nombre, on utilisera donc CHAR(10) pour un numéro de téléphone français
 - Est-ce une valeur faussement numérique ? Demandez-vous s'il est possible de faire des calculs (ayant un sens) avec cette valeur ? Si non ce n'est pas un nombre !

□ Types spécifiques

- DATE : date (yyyy-mm-dd)
 - Attention format anglais par défaut
- DATETIME : date et heure (yyyy-mm-dd hh:mm:ss)
- ...

□ Informations additionnelles

- AUTO_INCREMENT : valeur auto-incrémentée (pour une clé primaire)
- NOT NULL : la propriété doit avoir une valeur
- DEFAULT x : la propriété aura la valeur x par défaut (si non définie)
- ...

Clés primaires et étrangères

□ Clés primaires

- PRIMARY KEY(x) : la propriété x est la clé primaire
- PRIMARY KEY(x,y) : le couple de propriétés (x,y) est la clé primaire

□ Clés étrangères

- Exemple d'une clé étrangère x qui provient d'une propriété y, clé primaire de la table t

CONSTRAINT fk1

FOREIGN KEY(x) REFERENCES t(y)

Code SQL (Exemple 1)

Personnes
<u>idPersonne</u>
Prenom
Nom
DateDeNaissance
VilleDeNaissance
Sexe
Taille
NumeroSecuriteSociale

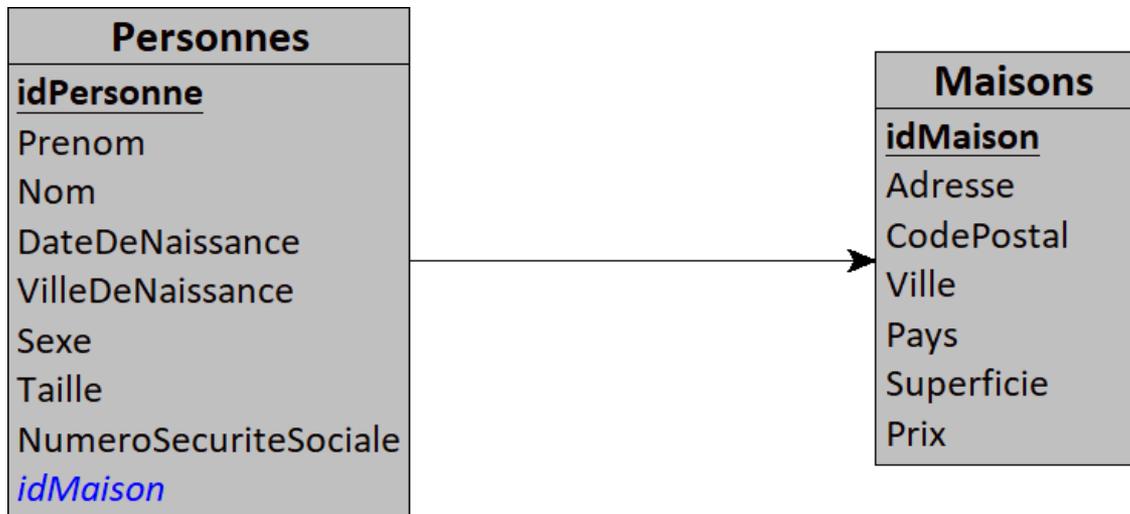
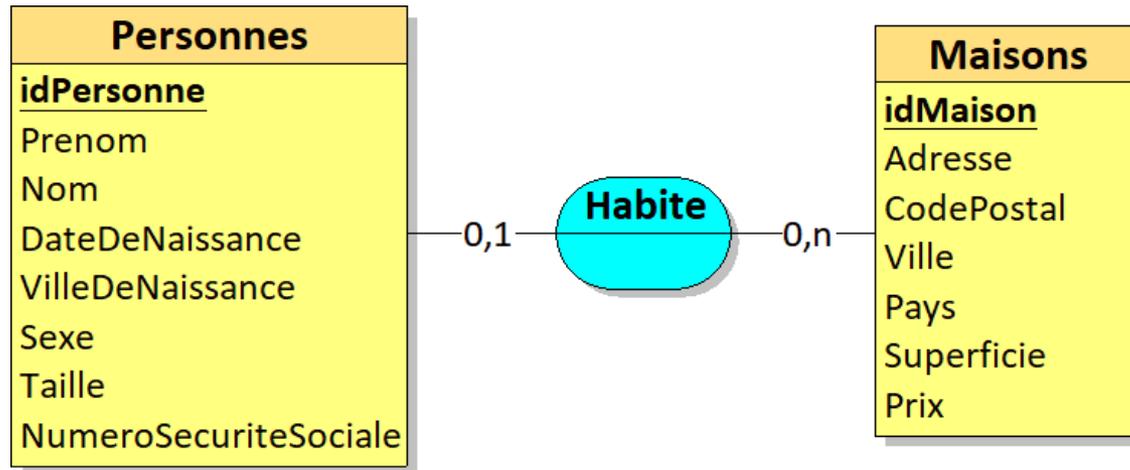
Personnes
<u>idPersonne</u>
Prenom
Nom
DateDeNaissance
VilleDeNaissance
Sexe
Taille
NumeroSecuriteSociale

Code SQL (Exemple 1)

```
CREATE TABLE Personnes  
(  
    idPersonne INT NOT NULL AUTO_INCREMENT,  
    Prenom VARCHAR(100) NOT NULL,  
    Nom VARCHAR(100) NOT NULL,  
    DateDeNaissance DATE NOT NULL,  
    VilleDeNaissance VARCHAR(100),  
    Sexe TINYINT,  
    Taille FLOAT,  
    NumeroSecuriteSociale CHAR(15),  
    PRIMARY KEY (idPersonne)  
)
```

Personnes
<u>idPersonne</u>
Prenom
Nom
DateDeNaissance
VilleDeNaissance
Sexe
Taille
NumeroSecuriteSociale

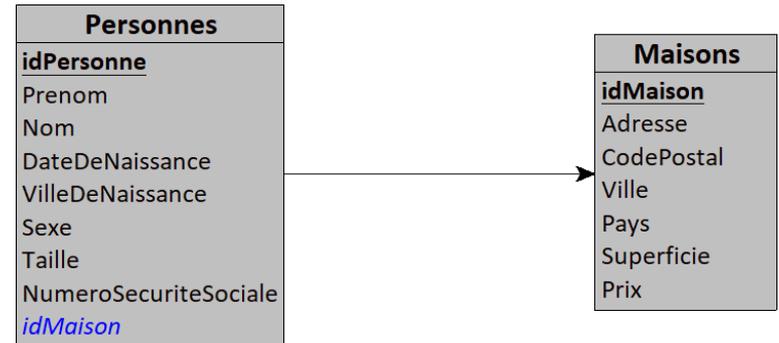
Code SQL (Exemple 2)



Code SQL (Exemple 2)

```
CREATE TABLE Maisons
```

```
(  
    idMaison INT NOT NULL AUTO_INCREMENT,  
    Adresse VARCHAR(200) NOT NULL,  
    CodePostal CHAR(5) NOT NULL,  
    Ville VARCHAR(100) NOT NULL,  
    Pays VARCHAR(100),  
    Superficie FLOAT,  
    Prix FLOAT,  
    PRIMARY KEY (idMaison)  
)
```



Code SQL (Exemple 2)

```
CREATE TABLE Personnes
```

```
(
```

```
    idPersonne INT NOT NULL AUTO_INCREMENT,
```

```
    Prenom VARCHAR(100) NOT NULL,
```

```
    Nom VARCHAR(100) NOT NULL,
```

```
    DateDeNaissance DATE NOT NULL,
```

```
    VilleDeNaissance VARCHAR(100),
```

```
    Sexe TINYINT,
```

```
    Taille FLOAT,
```

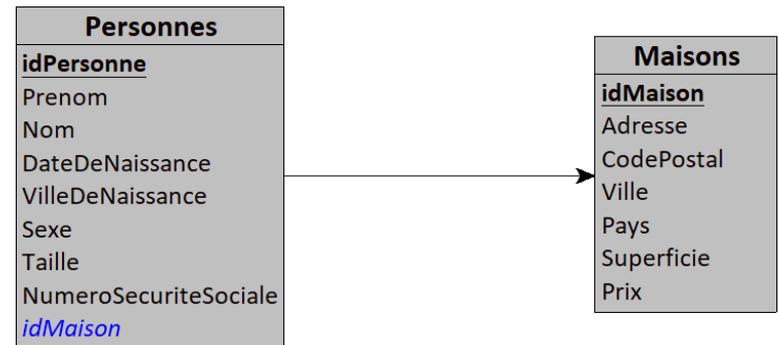
```
    NumeroSecuriteSociale CHAR(15),
```

```
    idMaison INT,
```

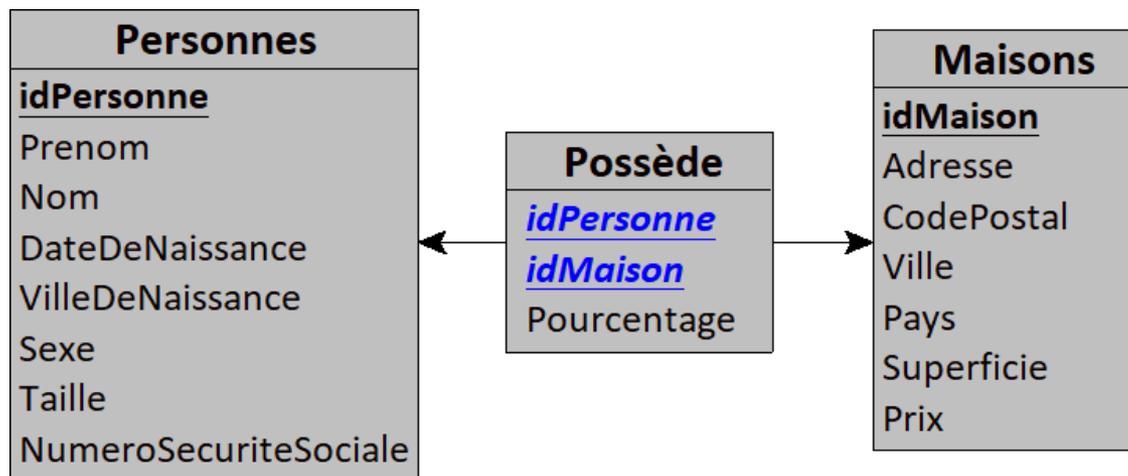
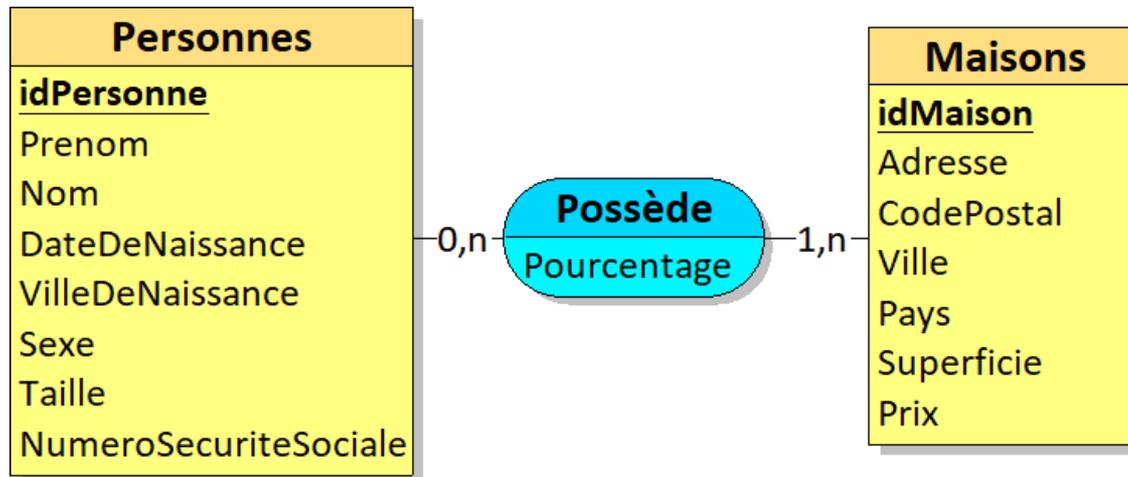
```
    PRIMARY KEY (idPersonne),
```

```
    CONSTRAINT fk1 FOREIGN KEY(idMaison) REFERENCES Maisons(idMaison)
```

```
)
```



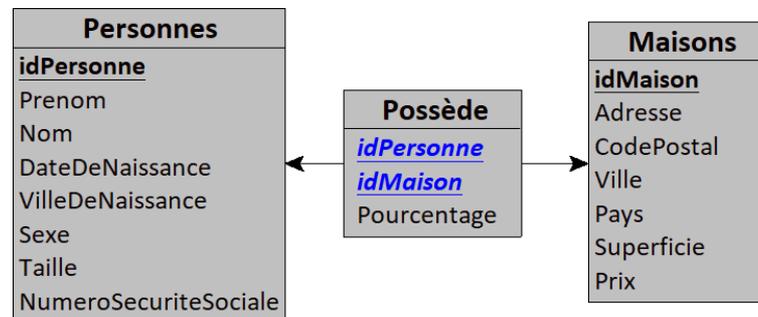
Code SQL (Exemple 3)



Code SQL (Exemple 3)

```
CREATE TABLE Maisons
```

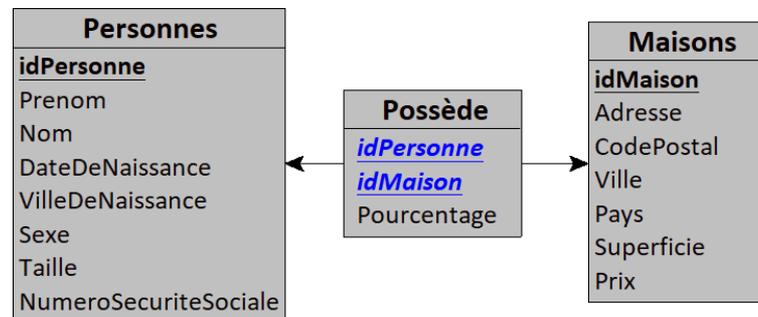
```
(  
    idMaison INT NOT NULL AUTO_INCREMENT,  
    Adresse VARCHAR(200) NOT NULL,  
    CodePostal CHAR(5) NOT NULL,  
    Ville VARCHAR(100) NOT NULL,  
    Pays VARCHAR(100),  
    Superficie FLOAT,  
    Prix FLOAT,  
    PRIMARY KEY (idMaison)  
)
```



Code SQL (Exemple 3)

```
CREATE TABLE Personnes
```

```
(  
    idPersonne INT NOT NULL AUTO_INCREMENT,  
    Prenom VARCHAR(100) NOT NULL,  
    Nom VARCHAR(100) NOT NULL,  
    DateDeNaissance DATE NOT NULL,  
    VilleDeNaissance VARCHAR(100),  
    Sexe TINYINT,  
    Taille FLOAT,  
    NumeroSecuriteSociale CHAR(15),  
    PRIMARY KEY (idPersonne)  
)
```



Code SQL (Exemple 3)

```
CREATE TABLE Possede
```

```
(  
    idPersonne INT NOT NULL,  
    idMaison INT NOT NULL,  
    Pourcentage FLOAT NOT NULL,  
    PRIMARY KEY (idPersonne,idMaison),  
    CONSTRAINT fk1 FOREIGN KEY(idPersonne)  
    REFERENCES Personnes(idPersonne),  
    CONSTRAINT fk2 FOREIGN KEY(idMaison)  
    REFERENCES Maisons(idMaison)  
)
```

