

# La conception des bases de données avec UML

Cinquième partie

# Le modèle conceptuel de données

## Les règles de cohérence

Afin d'obtenir un modèle conceptuel de données bien optimisé, nous avons déjà énoncé un certain nombre de règles.

Ces règles évitent les principaux problèmes mais ne sont pas suffisantes.

Nous allons les reprendre et les compléter:

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 1:**

*Toutes les rubriques du dictionnaire des données doivent être affectées à une classe-entité ou à une classe-association une fois et une seule fois.*

Il faut donc être sûr d'avoir pris en compte toutes les rubriques et seulement une fois.

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 2:**

*Une rubrique affectée à une classe-entité ou à une classe-association ne peut avoir qu'une seule valeur pour chaque instance de la classe.*

Exemple:

Personne ayant plusieurs numéros de téléphone.

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3:**

*Si une rubrique possède la même valeur pour plusieurs entités d'une même classe, il faut envisager que cette rubrique fasse l'objet d'une classe à part.*

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3:**

*Si une rubrique possède la même valeur pour plusieurs entités d'une même classe, il faut envisager que cette rubrique fasse l'objet d'une classe à part.*

Considérons l'exemple caricatural suivant :

# Le modèle conceptuel de données

## Les règles de cohérence

La règle 2 est bien appliquée:

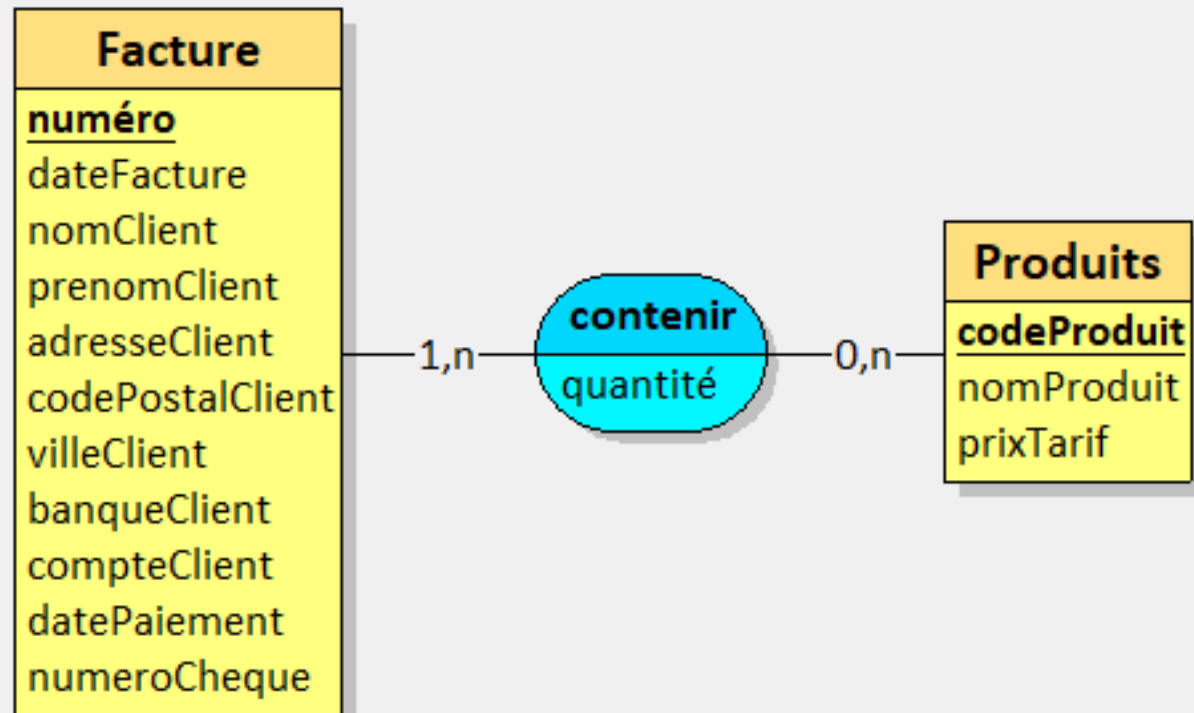
Pour une facture donnée:

Un seul nom de client,

Un seul prénom,

Une seule adresse,

Etc...



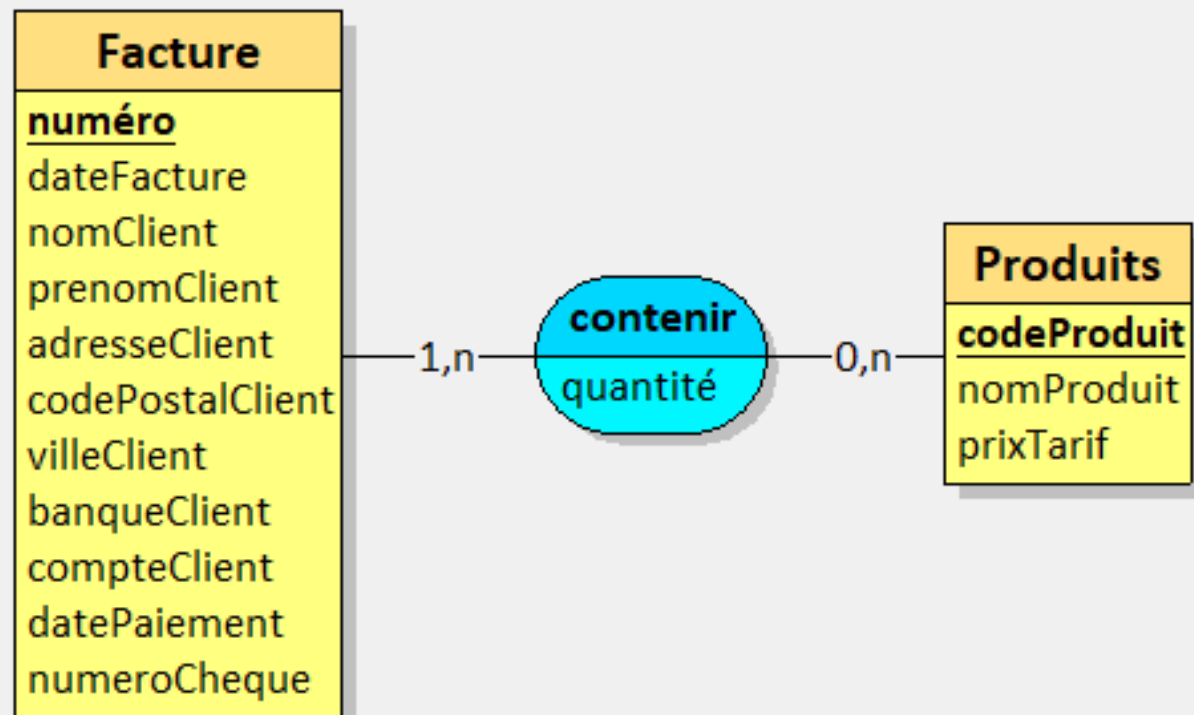
# Le modèle conceptuel de données

## Les règles de cohérence

La règle 2 est bien appliquée:

Toutes les rubriques affectées à la classe-entité client n'ont qu'une seule valeur pour une entité donnée.

Ou est le problème ?





# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3:**

Pour chaque facture destinée au client x, il faudrait donc enregistrer toutes ses coordonnées, avec plusieurs conséquences:

- Il peut y avoir des erreurs et des différences entre les diverses factures (erreurs de saisie).
- Le stockage de la même information va prendre de l'espace sur le support de stockage.
- Si les coordonnées du client changent, il faudra faire beaucoup de modifications.

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3:**

*Si une rubrique possède la même valeur pour plusieurs entités d'une même classe, il faut envisager que cette rubrique fasse l'objet d'une classe à part.*

Si on applique systématiquement cette règle, on va devoir faire les remarques suivantes:

Deux factures différentes peuvent avoir le même nomClient

Donc création d'une classe-entité Noms

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3:**

*Si une rubrique possède la même valeur pour plusieurs entités d'une même classe, il faut envisager que cette rubrique fasse l'objet d'une classe à part.*

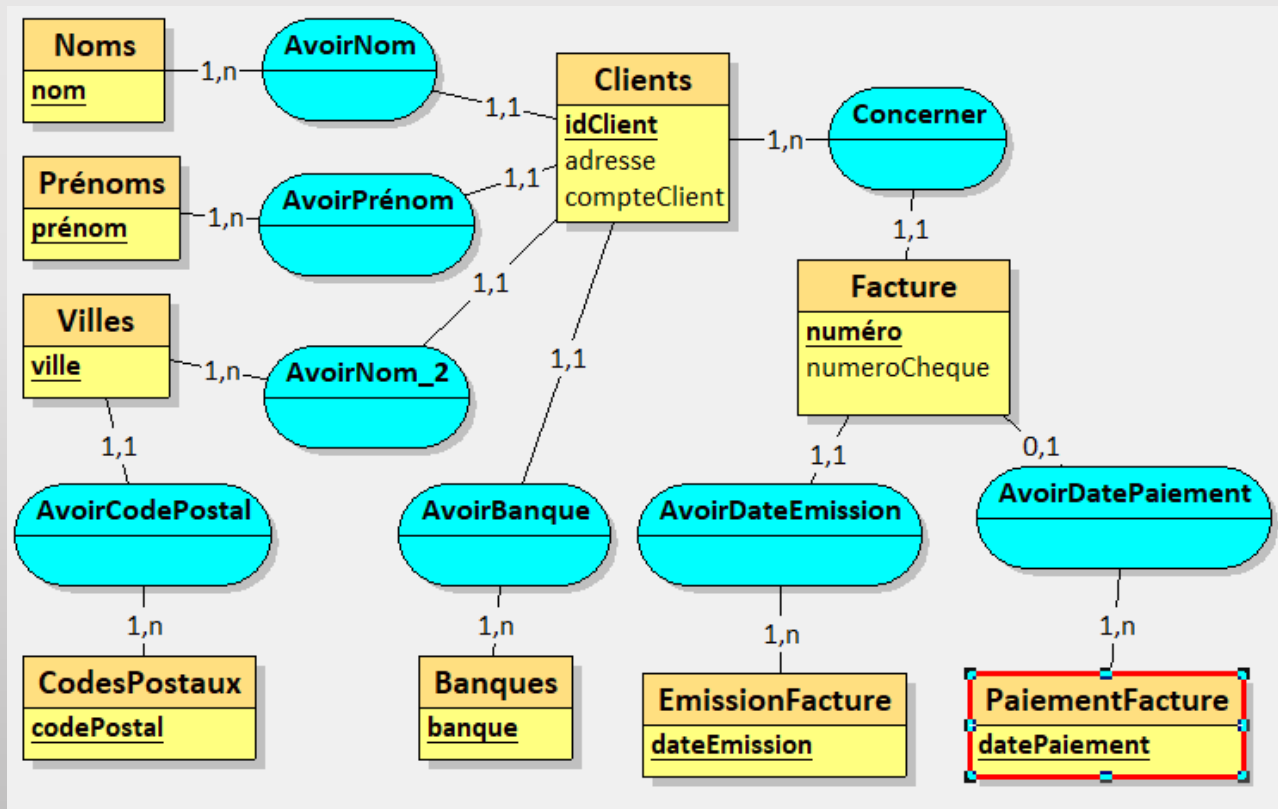
Si on applique systématiquement cette règle, on va devoir faire les remarques suivantes:

Même chose pour prenomClient, ville, codePostal, banque, dateFacture, datePaiement, prixTarif.

On arriverait au modèle suivant:

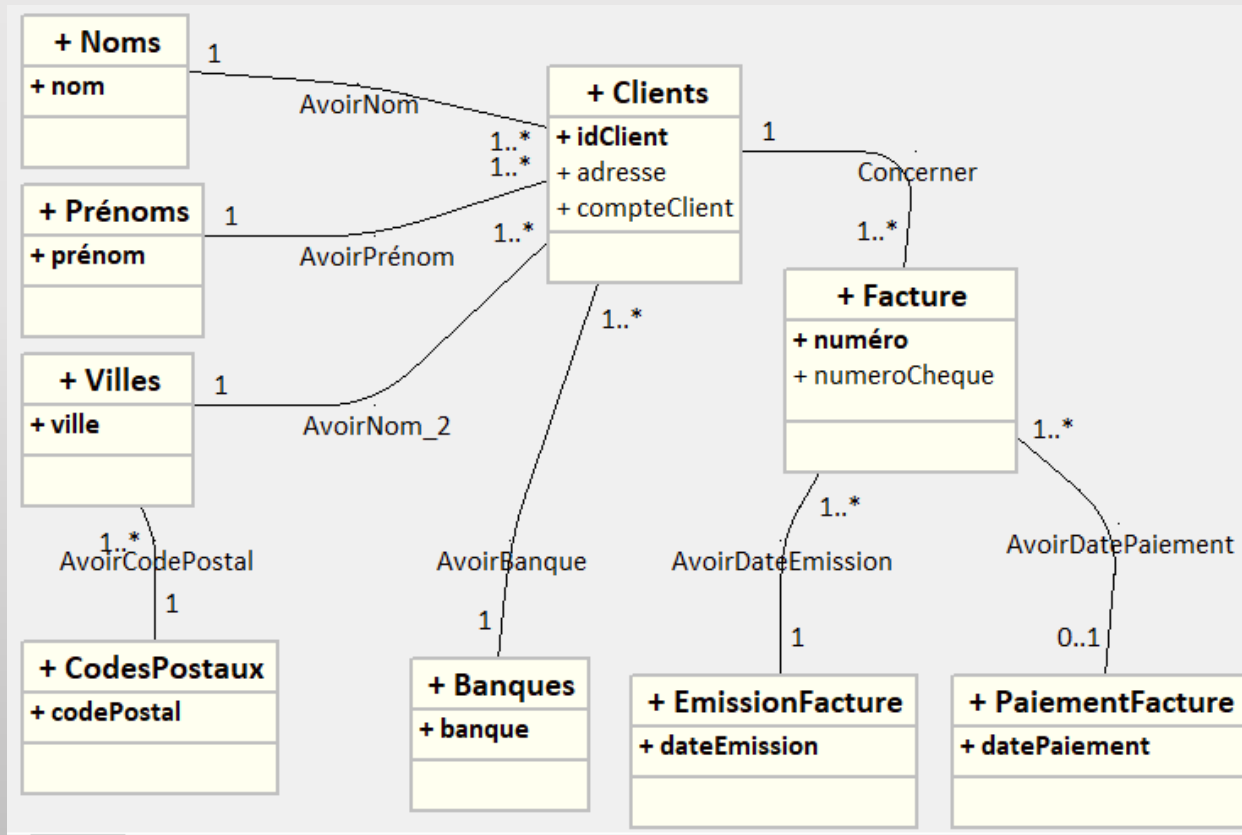
# Le modèle conceptuel de données

## Les règles de cohérence



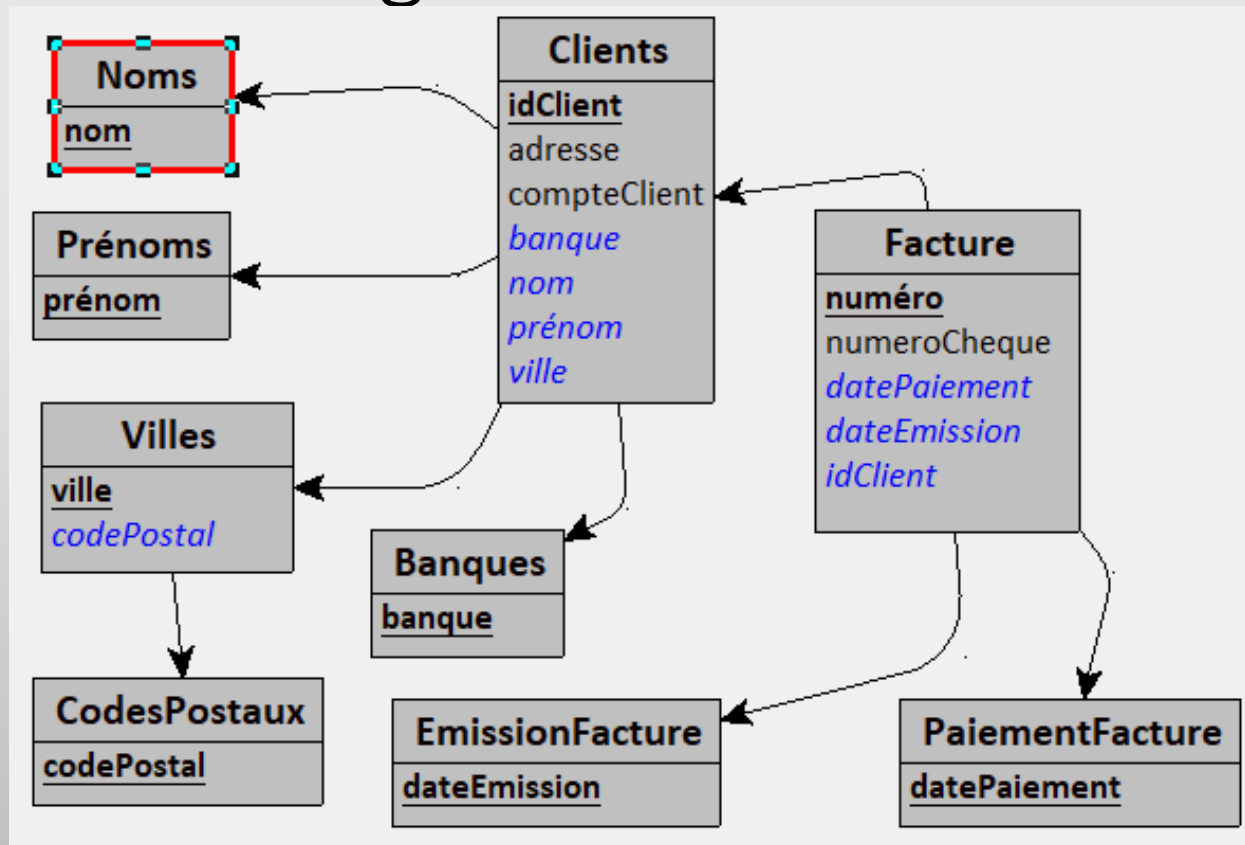
# Le modèle conceptuel de données

## Les règles de cohérence



# Le modèle conceptuel de données

## Les règles de cohérence



# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3:**

*Si une rubrique possède la même valeur pour plusieurs entités d'une même classe, il faut envisager que cette rubrique fasse l'objet d'une classe à part.*

Il faut éviter d'en arriver à cette situation.

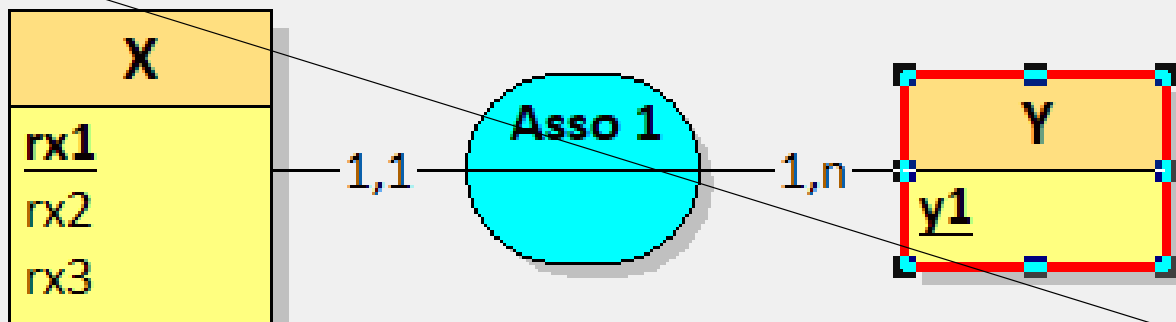
Ce qui pose problème, c'est la multiplicité des classes ne possédant qu'un seul champ associées par des associations 1:n

La recette est simple:

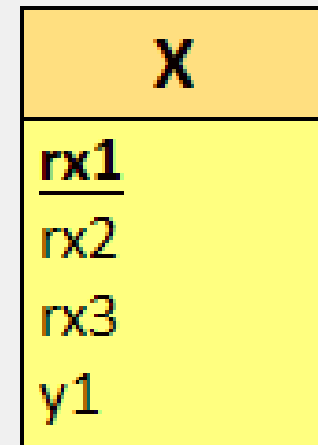
# Le modèle conceptuel de données

## Les règles de cohérence

Remplacer



par:





# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 3 modifiée:**

*Si une rubrique possède la même valeur pour plusieurs entités d'une même classe, il faut envisager que cette rubrique fasse l'objet d'une classe à part.*

*Cette règle est appliquée si, et seulement si, cette nouvelle classe dispose de plusieurs rubriques.*

# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 4:**

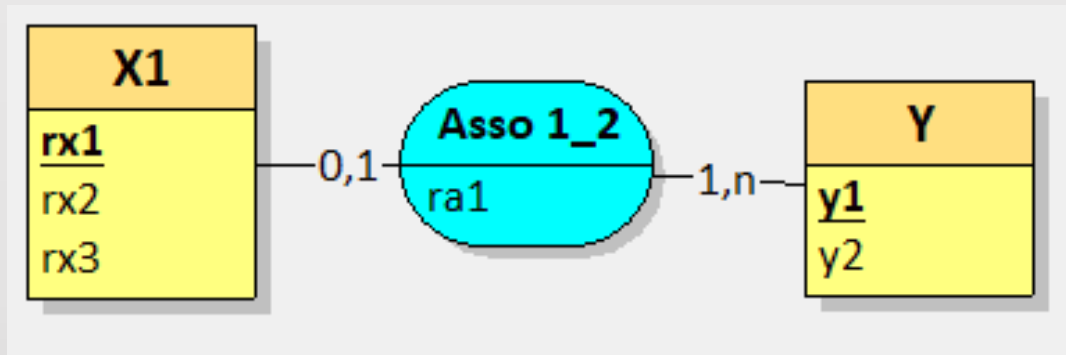
*Seules les associations  $n:m$  peuvent disposer de rubriques.*

*Les rubriques des autres associations doivent être placées dans les classes-entités coté 0,1 ou 1,1.*

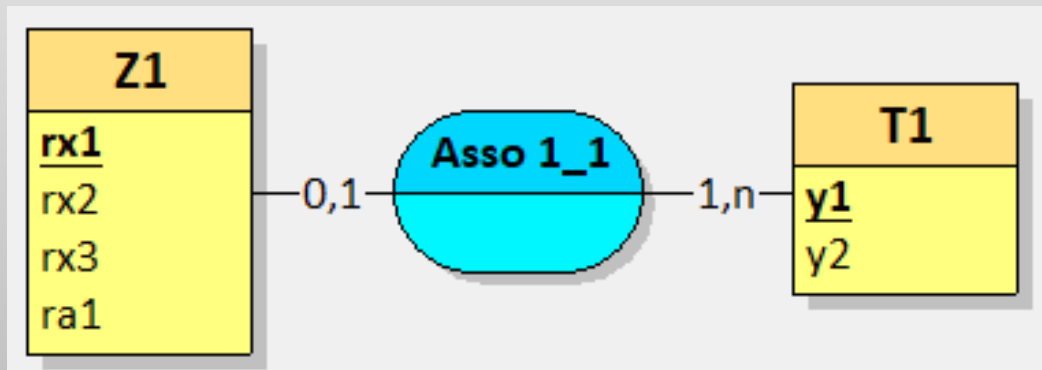
# Le modèle conceptuel de données

## Les règles de cohérence

Remplacer



Par:



# Le modèle conceptuel de données

## Les règles de cohérence

### **Règle n° 5:**

*Il ne doit jamais y avoir deux cardinalités 1,1 dans une classe-association.*

Toutes les rubriques doivent être placées dans la même classe-entité.

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Lors de la définition des entités, il se peut que plusieurs entités différentes possèdent des mêmes rubriques semblables, n'ayant pas le même nom (respect des règles) mais les mêmes buts.

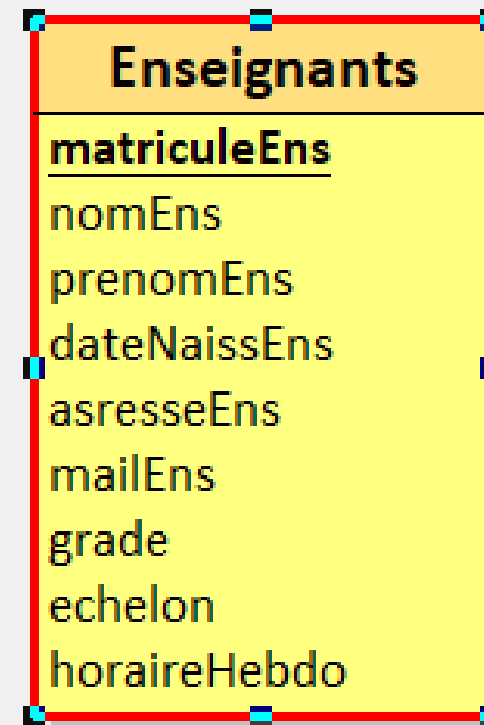
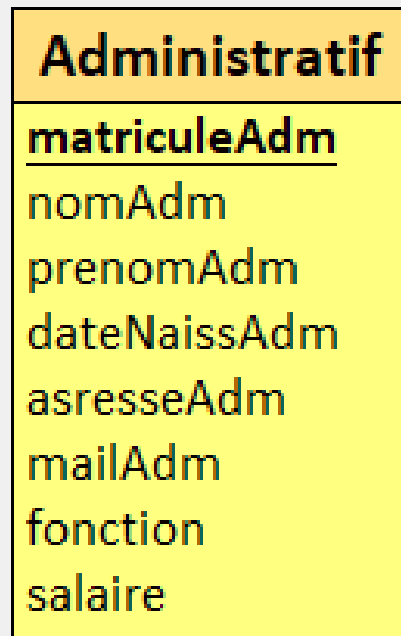
Par exemple, prenons la situation du personnel d'un lycée; Dans cet établissement nous pouvons considérer deux catégories de personnel:

- Les enseignants.
- Les personnels administratifs.

On pourrait concevoir un modèle de ce type :

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation



# Le modèle conceptuel de données Héritage, Spécialisation, Généralisation

Cette approche est satisfaisante si les deux catégories de personnel n'ont pas d'associations communes avec d'autres entités:

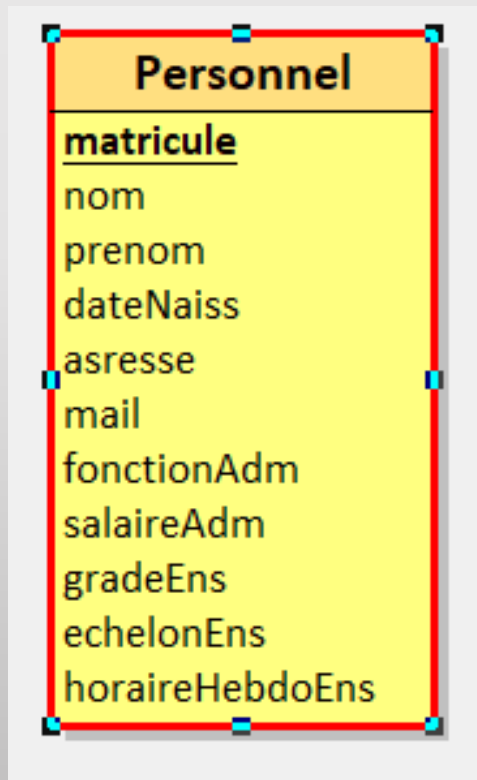
- Gestion des congés.
- Prise en compte des arrêts maladie.

Ces deux problèmes devront être associés à des entités qui, elles mêmes, devront être liées aux entités Administratif et Enseignant alors que leur prise en compte seront les mêmes.

Une seconde approche est la suivante :

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation





# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Dans cette approche, le problème précédent est bien pris en compte, mais ...

Comment va-ton gérer les associations qui ne sont liées qu'à des enseignants ou qu'à du personnel administratif ?

Par exemple, on pourrait associer à un enseignant les classes dans lesquelles il va intervenir (les administratifs ne sont pas concernés).

La solution la plus efficace consiste à utiliser la notion d'*héritage*.

# Le modèle conceptuel de données

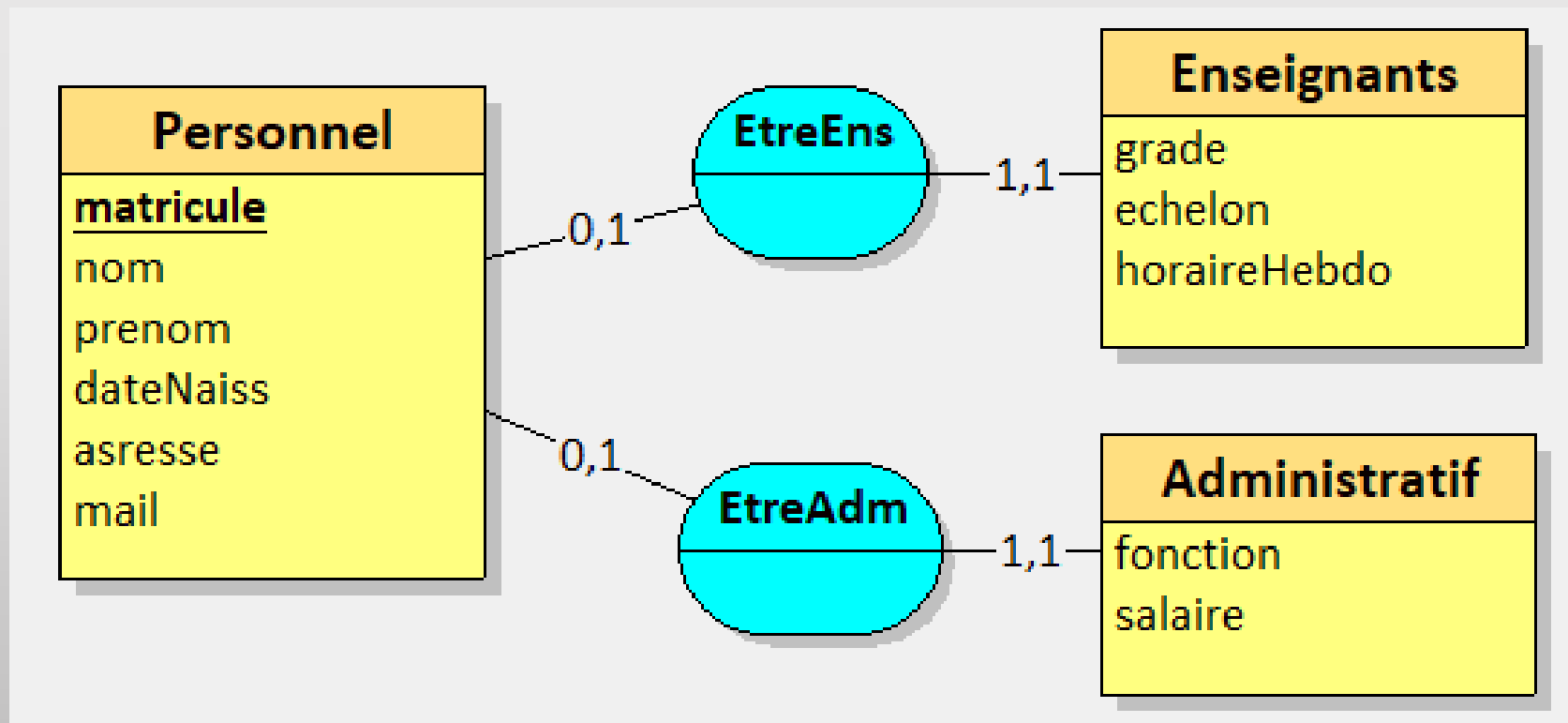
## Héritage, Spécialisation, Généralisation

On va définir deux catégories de classes-entités:

- Une classe-entité contenant toutes les rubriques communes (jouant un même rôle) appelée ***sur-type***.
- Des classes-entités ne contenant que les parties qui sont spécifiques appelées ***sous-type***.

On pourrait modéliser cette approche de la façon suivante:

# Le modèle conceptuel de données Héritage, Spécialisation, Généralisation



# Le modèle conceptuel de données Héritage, Spécialisation, Généralisation

Ce modèle a un gros problème:

Les entités Enseignants et Administratif n'ont pas d'identifiant.

On peut résoudre ce problème de deux façons distinctes:

- On ajoute un identifiant à chaque sous-type : on appelle cela la spécialisation des sous-types.
- On récupère l'identifiant du sur-type dans chaque sous-type: on appelle cela la généralisation des sous-types.

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Pour généraliser les sous-types (ou certains d'entre eux), on pratique de la façon suivante, en modifiant la cardinalité 1,1:

On obtient la boîte de dialogue suivante:

# Le modèle conceptuel de données Héritage, Spécialisation, Généralisation

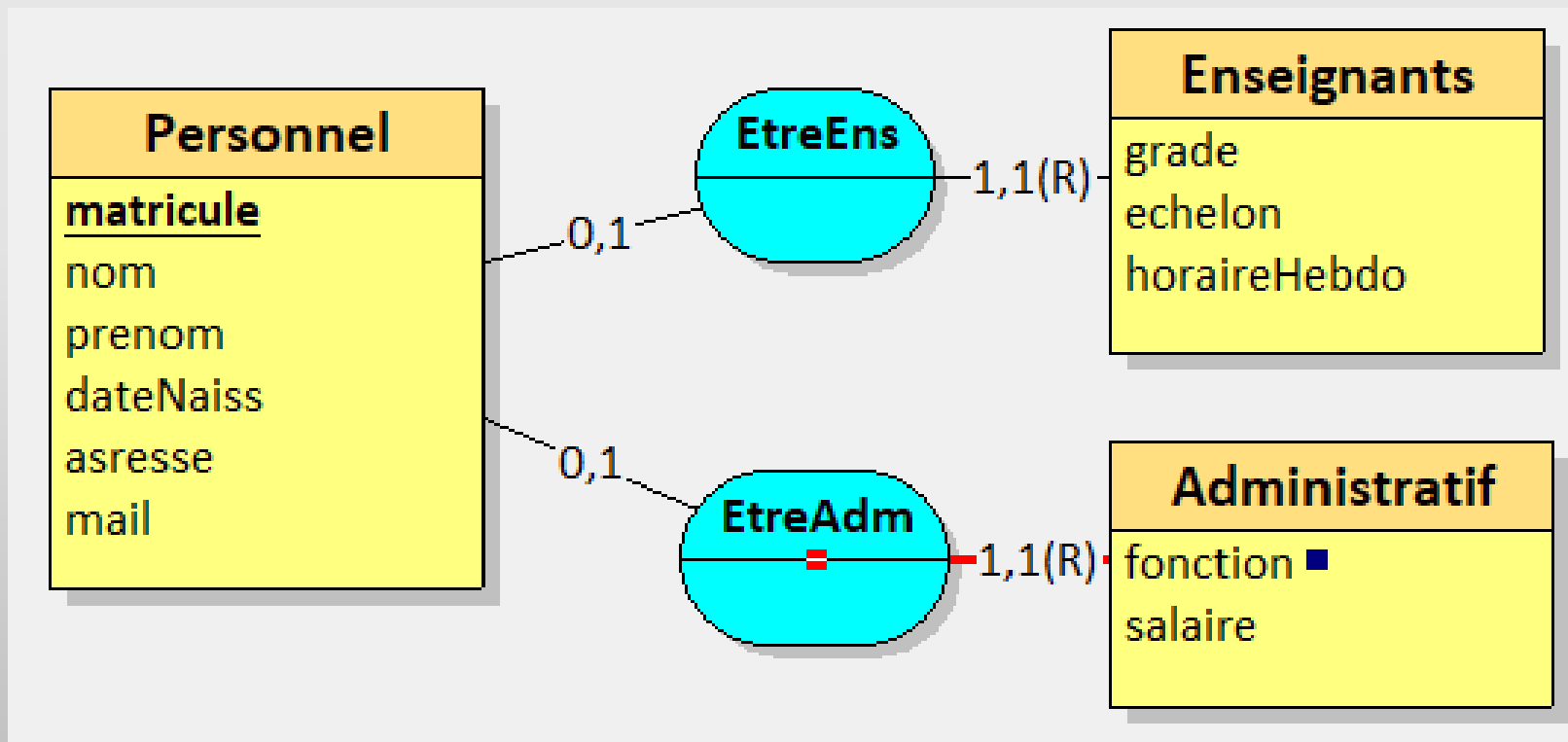
Dans le cas des associations 1:1 qui sont susceptibles de gérer des sous-types, une case à cocher notée « Identifiant relatif » permet de rajouter l'identifiant de l'autre entité à celui de l'entité concernée par le lien.

La mention (R) va apparaître à droite de la cardinalité.

The screenshot shows a dialog box titled 'Cardinalité' with a close button (X) in the top right corner. Inside the dialog, there are several options for defining cardinality:

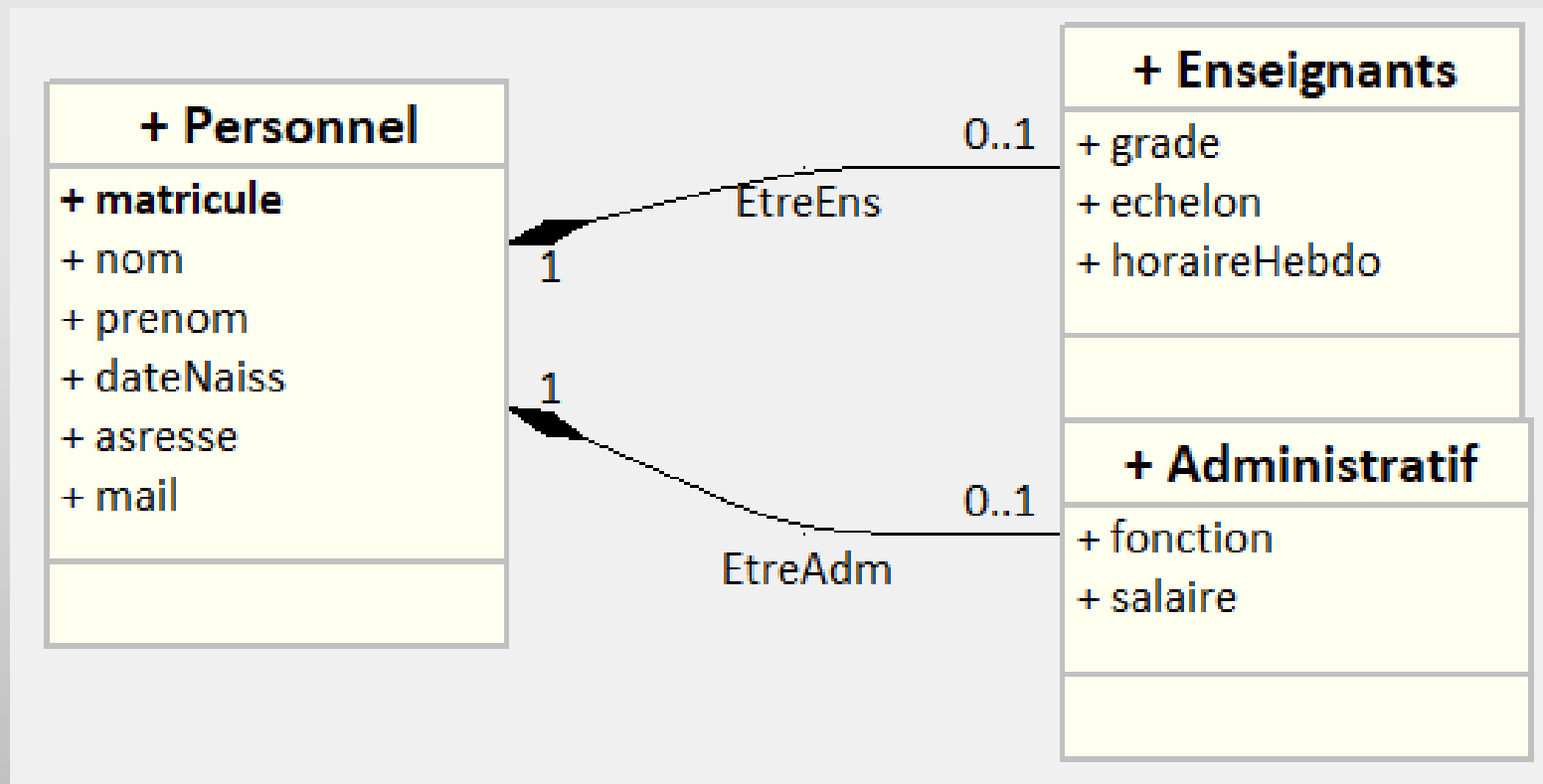
- Radio buttons for '0,1', '1,1' (which is selected), '0,n', and '1,n'.
- A checkbox labeled 'Identifiant relatif' which is currently unchecked.
- Radio buttons for 'Min : ' and 'Max : ' followed by empty input boxes.
- A section labeled 'Rôle' containing a label 'Libellé : ' and an empty text input box.
- At the bottom, there are two buttons: 'OK' and 'Annuler'.

# Le modèle conceptuel de données Héritage, Spécialisation, Généralisation



# Le modèle conceptuel de données

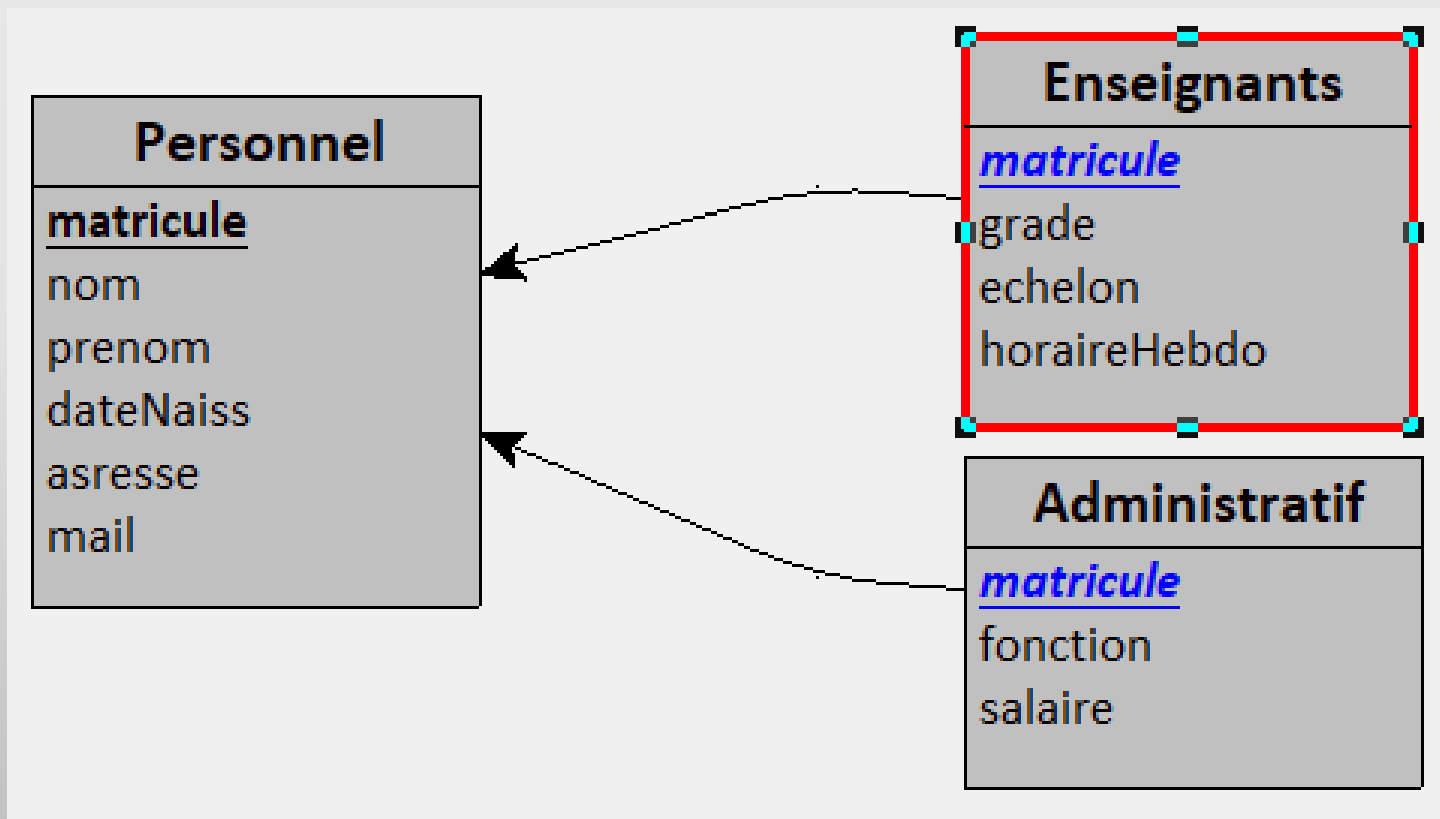
## Héritage, Spécialisation, Généralisation





# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation



# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Ces notions de sur-type et sous-types sont équivalents aux notions de classe de base et de classe héritées de la programmation objet, et c'est pour cela que l'on parlera d'**héritage**.

On dira:

- La classe-entité Enseignants hérite de la classe Personnel.
- La classe-entité Administratif hérite de la classe Personnel.

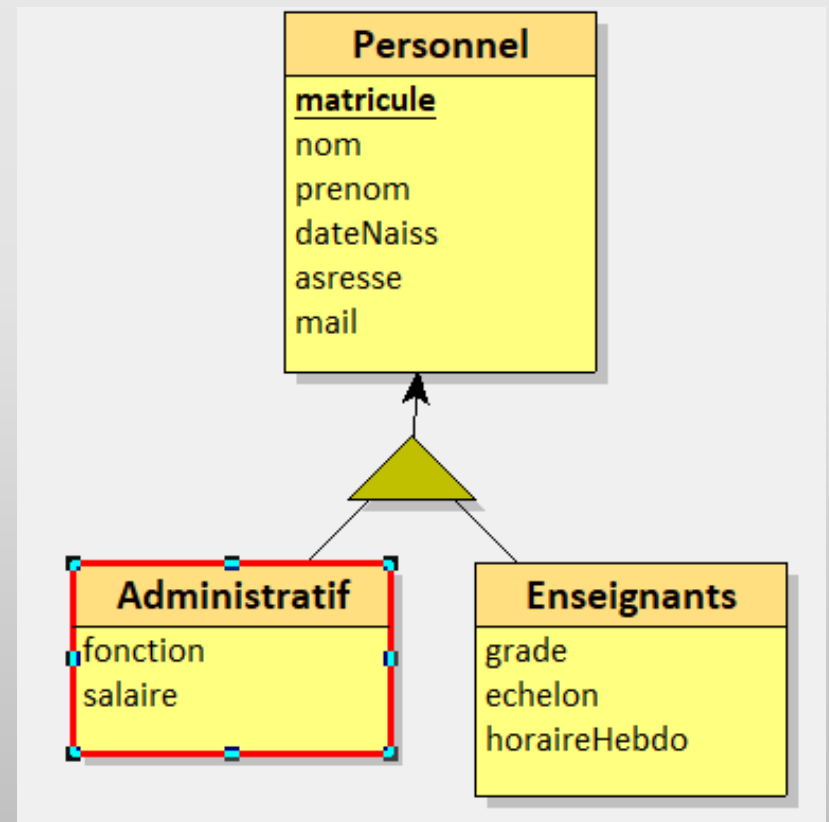
Une classe-entité pourra avoir n'importe quel nombre d'héritiers, mais une classe entité ne peut hériter que d'une seule classe-entité.

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Un formalisme particulier est prévu pour symboliser l'héritage:

En Merise

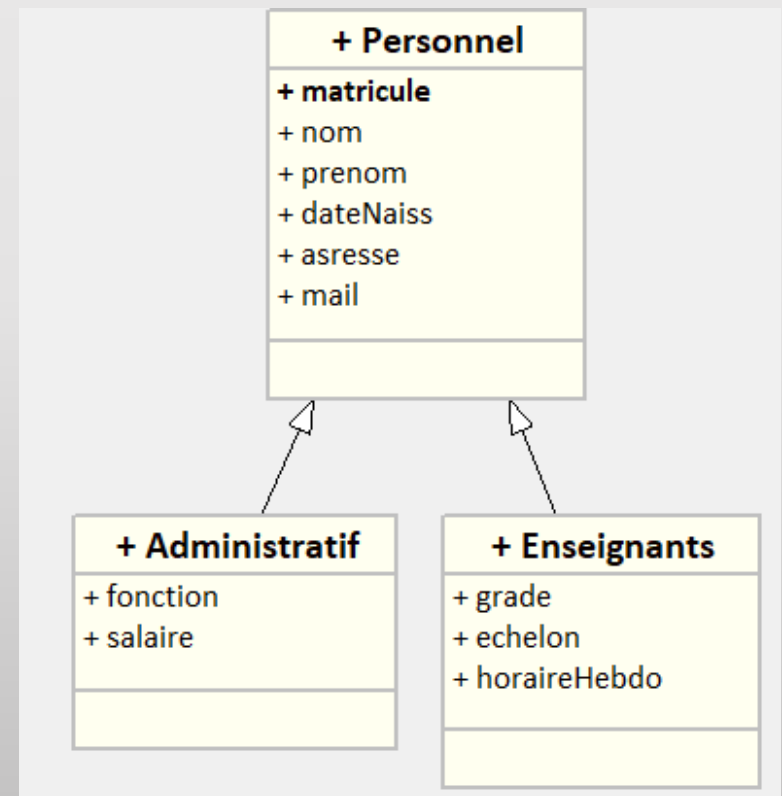


# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Un formalisme particulier est prévu pour symboliser l'héritage:

En U.M.L.



# Le modèle conceptuel de données

## Les contraintes inter-associations

La notion d'héritage va permettre d'intégrer une nouvelle notion au niveau de la modélisation.

Dans notre exemple, est-ce qu'un enseignant peut-être un administratif ?

Plusieurs situations sont envisageables :

# Le modèle conceptuel de données

## Les contraintes inter-associations

Si une entité du sur-type concernée par une entité du sous-type ne peut pas être concerné par une entité d'un autre sous-type, on dira qu'il y a une contrainte d'exclusion, notés X

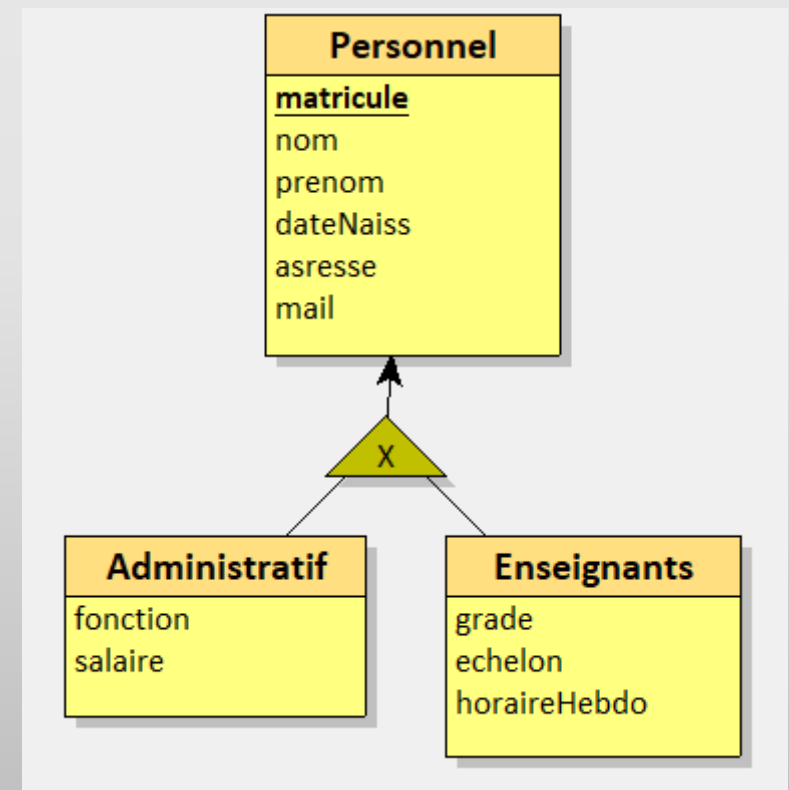
Exemple: un administratif ne peut pas être un enseignant.

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Exclusivité:

Un enseignant ne pourra pas être un administratif.

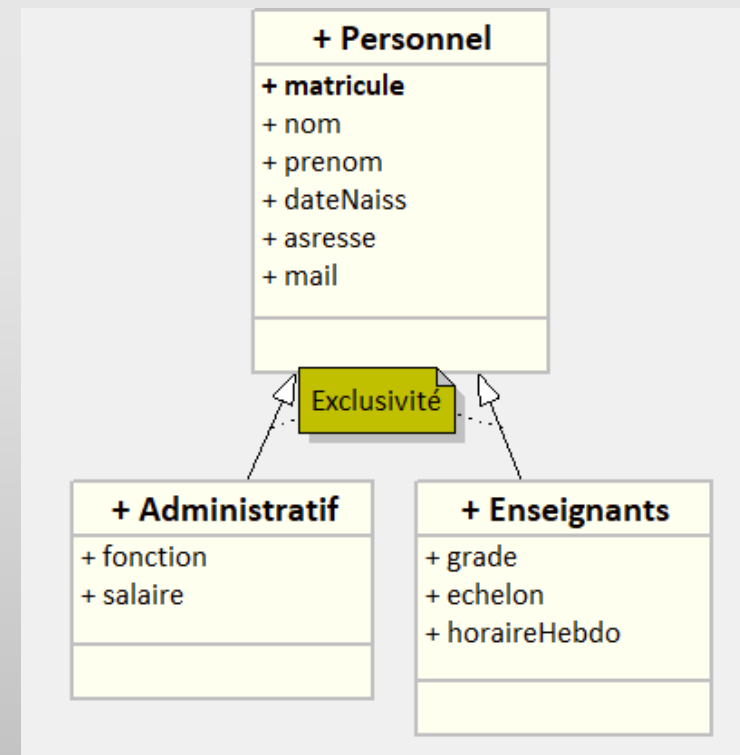


# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Exclusivité:

Un enseignant ne pourra pas être un administratif.





# Le modèle conceptuel de données

## Les contraintes inter-associations

Si toutes les entités du sur-type sont concernées par au moins une entité de l'un des sous-type, on dira qu'il y a une contrainte de totalisation, notée T

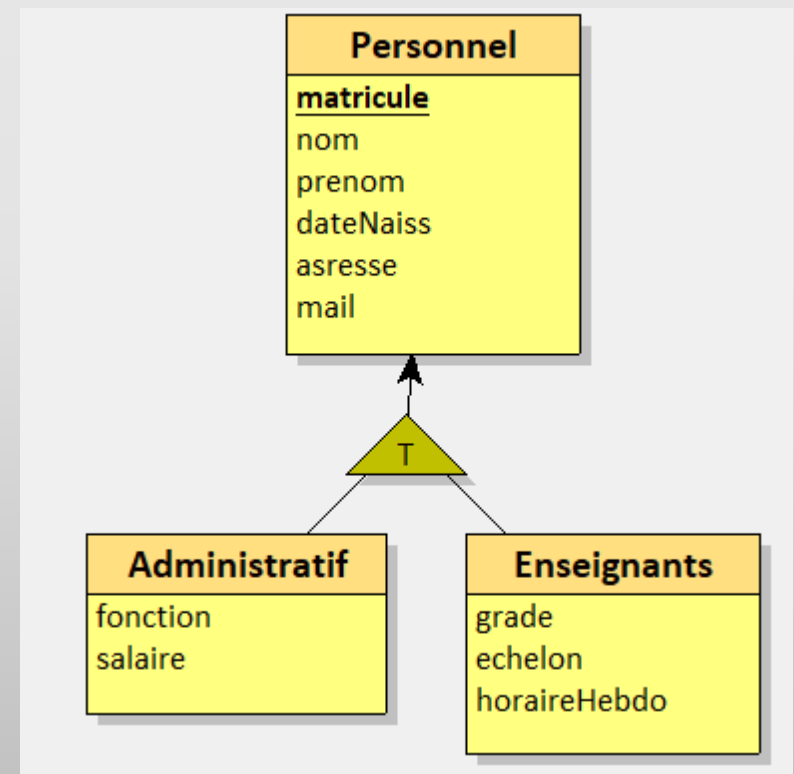
Exemple: un membre du personnel doit être au moins un administratif ou un enseignant (ou les deux).

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Exclusivité:

Un membre du personnel devra être soit un administratif, soit un enseignant, ou les deux à la fois.



# Le modèle conceptuel de données

## Les contraintes inter-associations

Si chaque entités du sur-type ne doit être concernée que par une seule entité de l'un de ses sous-types, on dira qu'il y a une contrainte de partition, notée XT

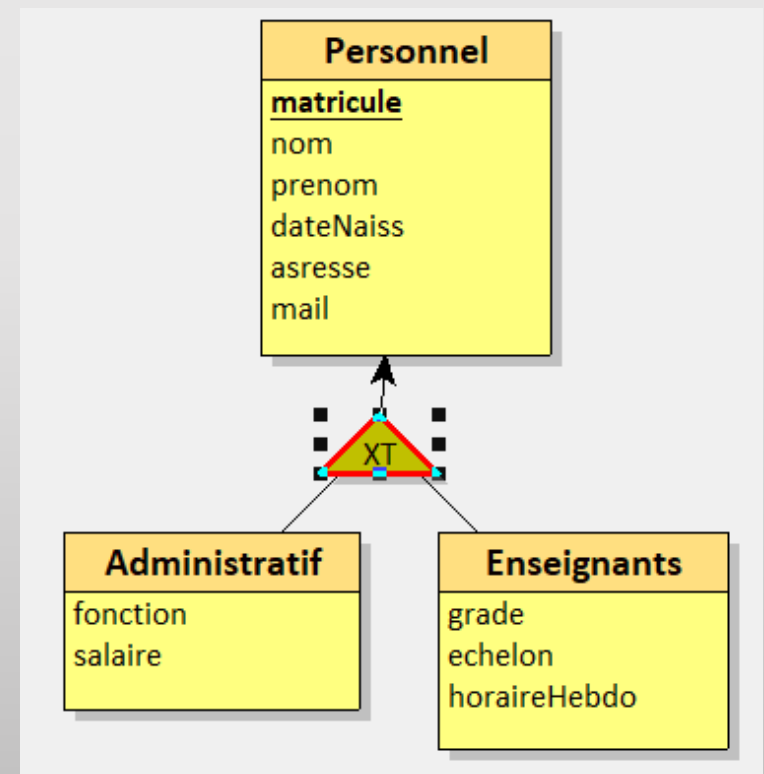
Exemple: un membre du personnel doit être soit un administratif, soit un enseignant (pas les deux).

# Le modèle conceptuel de données

## Héritage, Spécialisation, Généralisation

Exclusivité:

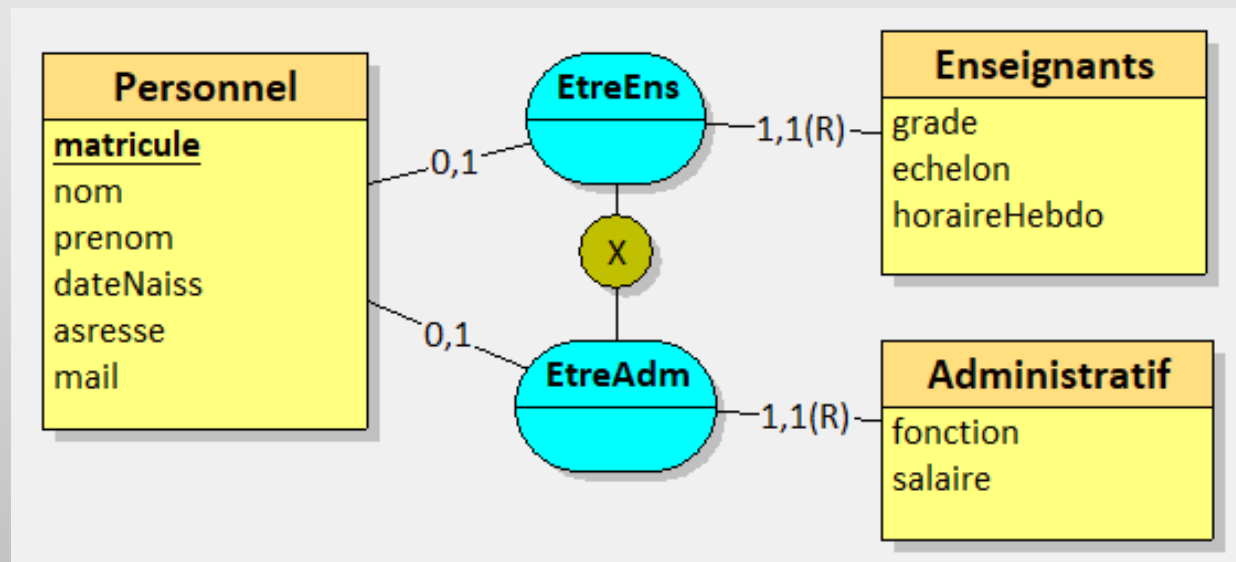
Un membre du personnel devra être soit un administratif, soit un enseignant, mais pas les deux.



# Le modèle conceptuel de données

## Les contraintes inter-associations

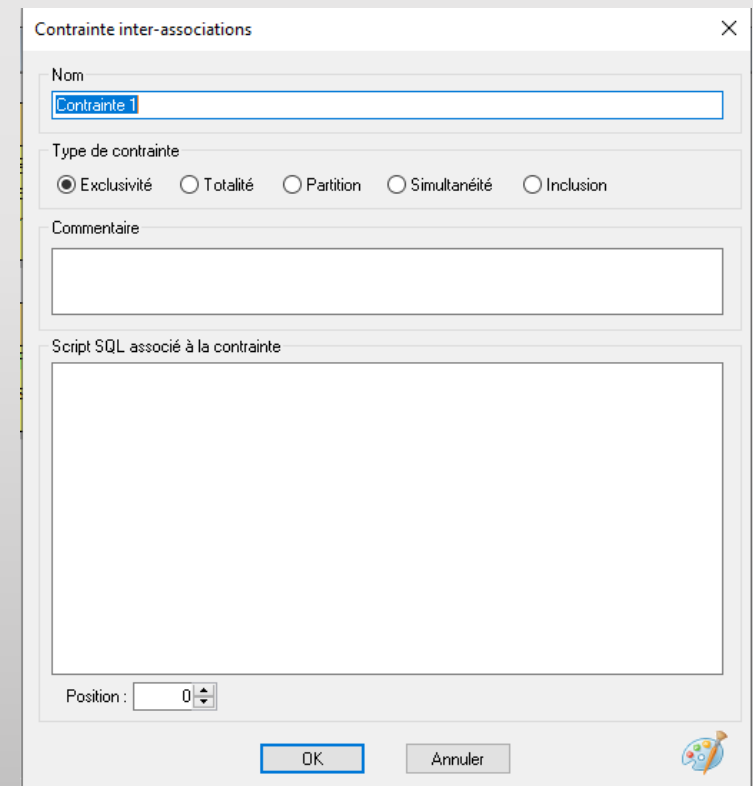
En Merise, dans le formalisme de base, les contraintes inter-associations sont indiquées comme suit:



# Le modèle conceptuel de données

## Les contraintes inter-associations

Le type de contrainte est modifiable  
à l'aide de la boîte de dialogue suivante:



The dialog box is titled "Contrainte inter-associations" and contains the following fields and controls:

- Nom:** A text field containing "Contrainte 1".
- Type de contrainte:** A group of five radio buttons: "Exclusivité" (selected), "Totalité", "Partition", "Simultanéité", and "Inclusion".
- Commentaire:** A large empty text area.
- Script SQL associé à la contrainte:** A large empty text area.
- Position:** A small numeric field with a spinner, currently set to "0".
- Buttons:** "OK" and "Annuler" buttons at the bottom right.
- Icon:** A small icon of a palette and brush in the bottom right corner.