

Bases de Données 1

#3 - Schéma relationnel

Matthieu Nicolas
Polytech S5 - II

Slides réalisées à partir de celles de Claude Godart et Malika Smaïl

Plan

- Modèle relationnel
- Du schéma conceptuel au schéma relationnel

Modèle relationnel

Base de Données 1
#3 - Schéma relationnel

Historique

- Proposé en 1970 par E.F. Codd (*IBM San José*)
- 1ères réalisations (*System-R, Ingres*) vers 1976
- 1ers systèmes commerciaux au début des années 80

Modèle relationnel

- Données organisées sous forme de tables à 2 dimensions, appelées **relations**
- Les lignes de ces tables sont appelées **tuple** ou **n-uplet**
 - Une **ligne** représente un **objet**, une **colonne** un **attribut**
- Données manipulées par des opérateurs de l'algèbre relationnelle
- État cohérent de la base défini par ensemble de contraintes d'intégrités

Avantages du modèle relationnel

- **Simple**, facile à appréhender, même pour un non-spécialiste
- Dispose de solides bases théoriques
 - Manipulations à l'aide des opérateurs de l'algèbre relationnelle a permis de mettre en place un langage standardisé : **SQL**

Attributs et Domaines

- Un **attribut** est un **nom décrivant une information** stockée dans une base
 - L'âge d'une personne, son *nom*, *numéro de sécurité sociale*...
- Le **domaine** d'un attribut est l'**ensemble**, fini ou infini, **de ses valeurs possibles**
 - L'attribut *nom* a pour domaine l'**ensemble des chaînes de caractères**
 - L'attribut *numéro de sécu* a pour domaine l'**ensemble des combinaisons de quinze chiffres**

Relations - 1

- De manière informelle : une **relation** est une table qui associe pour chaque ligne (**tuple**) une valeur pour chaque colonne (**attribut**)

| attribut → | | no | nom | prenom | age |
|--------------------------|--|----|--------|---------|-----|
| tuple ou n-uplet → | | 15 | Dupond | Loïc | 19 |
| | | 12 | Dupond | Michèle | 23 |
| | | 1 | Martin | Michèle | 23 |

↑ domaine de l'attribut *age* : **entier**

Relations - 2

- De manière plus formelle : une **relation** est un sous-ensemble du produit cartésien de n domaines d'attributs ($n > 0$)
- **Rappel** : le produit cartésien de n ensembles est l'ensemble de toutes les combinaisons possibles entre les valeurs des n ensembles
 - *Exemple* : $\{1,2\} \times \{X, Y\} = \{<1,X>, <1,Y>, <2,X>, <2,Y>\}$

Schéma de relation

- Un **schéma de relation** précise le **nom de la relation** ainsi que la liste ordonnée des **attributs avec leurs domaines**

| no | nom | prenom | age |
|----|--------|---------|-----|
| 15 | Dupond | Loïc | 19 |
| 12 | Dupond | Michèle | 23 |
| 1 | Martin | Michèle | 23 |

Etudiant(no: entier, nom: chaine, prenom: chaine, age: entier)

Etudiant(no, nom, prenom, age)

Degré et cardinalité

- Le **degré d'une relation** est son nombre d'attributs
- Une **occurrence**, ou **n-uplet** ou **tuple**, est un élément de l'ensemble figuré par la relation
 - Autrement dit, une **occurrence** est une ligne de la table qui représente une relation
- La **cardinalité d'une relation** est son nombre d'occurrences

Schéma relationnel et BD relationnelle

- Un **schéma relationnel** est constitué par l'**ensemble des schémas de relations**
 - On parle aussi d'**intension** de la base de données
- Une **base de données relationnelle** est constituée par l'**ensemble des tuples des différentes relations** du schéma relationnel
 - On parle aussi d'**extension** de la base de données

Valeur NULL

- Valeur particulière indiquant
 - **Valeur** d'un attribut n'est **pas connue** pour un **tuple**
 - **Attribut** ne s'**applique pas** pour un **tuple**
- Exemples
 - Client dont on **ignore** la date de naissance
 - Personne **ne possédant pas** de téléphone portable
- **NULL appartient au domaine des attributs non obligatoires d'une relation**

Clés candidates - 1

- Une **clé candidate** d'une **relation** est un **ensemble minimal d'attributs** de la relation dont les valeurs **identifient** à coup sûr une **occurrence**.
- Règle : **toute relation a au moins une clé candidate** et peut en avoir plusieurs
 - Il ne peut jamais y avoir deux tuples identiques au sein d'une relation

Clés candidates - 2

- Quel(s) ensemble(s) d'attributs nous permet d'identifier de manière unique chaque tuple de cette relation ?

| no | nom | prenom | age |
|----|--------|---------|-----|
| 15 | Dupond | Loïc | 19 |
| 12 | Dupond | Michèle | 23 |
| 1 | Martin | Michèle | 23 |

- $\{no\}$ est la seule clé candidate de la relation ***Etudiant***

Clés candidates - 3

- Une **relation** peut avoir **plusieurs clés candidates** n'ayant pas forcément le même nombre d'attributs
- Une clé candidate peut être formée d'un **attribut arbitraire** qui n'a d'autre **objectif** que de **servir de clé**

Contraintes sur clé primaire

- La **valeur** de la **clé primaire** d'une relation **doit être unique pour chaque tuple** de la relation
- La **valeur** de la **clé primaire** d'une relation **doit être définie** (*non nulle*) **pour chaque tuple** de la relation

Clé étrangère - 1

- On appelle **clé étrangère** d'une relation tout **ensemble d'attributs** qui constitue une **clé candidate d'une autre relation**
 - Généralement sa clé primaire...
 - ... Mais pas forcément

Clé étrangère - 2

- Considérons le schéma relationnel suivant :
 - ***Buveur*** (*nb*, *nom*, *prenom*)
 - ***Vin*** (*nv*, *cru*, *millesime*, *degre*)
 - ***Abus*** (*nb*, *nv*, *date*, *quantite*)
- ***Abus.nv*** est une clé étrangère, elle référence ***Vin.nv***
- ***Abus.nb*** est une clé étrangère, elle référence ***Buveur.nb***

Contraintes sur clé étrangère - 1

- Les clés étrangères nous **permettent de relier les différentes relations** de notre schéma relationnel...
- ... Mais lorsqu'une relation contient des clés étrangères, **les tuples correspondant à ces clés doivent exister** dans la relation référencée
- On parle alors de **contrainte de référence**

Contraintes sur clé étrangère - 2

- Reprenons le schéma relationnel suivant :
 - **Buveur** (*nb*, *nom*, *prenom*)
 - **Vin** (*nv*, *cru*, *millesime*, *degre*)
 - **Abus** (*nb*, *nv*, *date*, *quantite*)
- On **peut signaler** (*insérer*) que des abus des *buveurs* *qui existent* avec des *vins* *qui existent*
- On ne **peut pas supprimer** un buveur ou un vin pour lequel un abus a été signalé

Du schéma conceptuel au schéma relationnel

Base de Données 1
#3 - Schéma relationnel

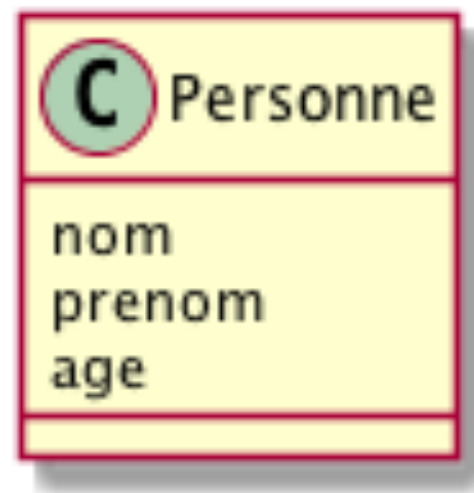
Passage d'un diagramme de classes à un schéma relationnel

- Représente chaque classe d'objet par une table et choisir sa clé primaire
- Représente chaque association par une ou plusieurs tables (en fonction du type d'association et de ses cardinalités) et choisir leur clés primaires

Classe -> Table

- Dans un 1er temps, on fait **correspondre une classe à une table**
- **Le nom de la table** est le **nom de la classe**
- **Les attributs de la table** sont **ceux de la classe**
 - Peut éventuellement ajouter un attribut *identifiant*
- On **laisse tomber les opérations**
 - Seront gérées par les programmes des applications

Exemple Classe -> Table



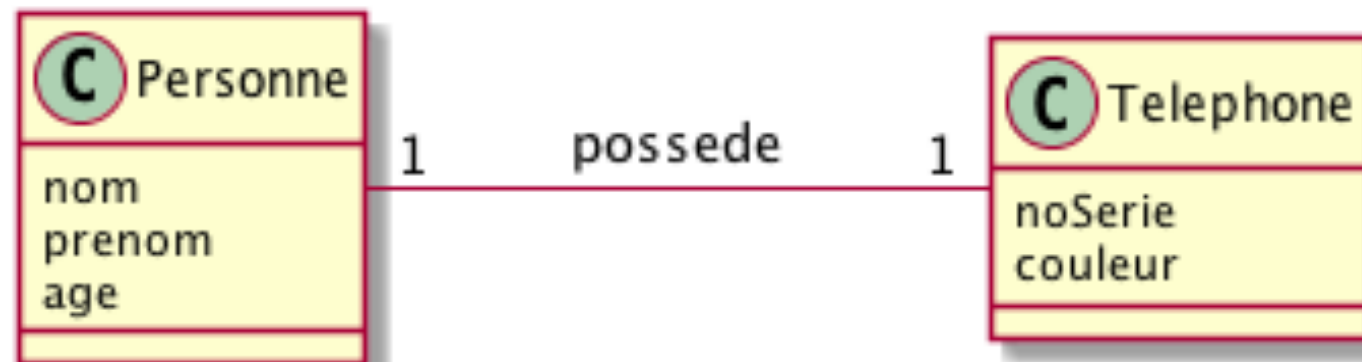
équivalent à

Personne(*idPersonne*, *nom*, *prenom*, *age*)

Associations -> Table

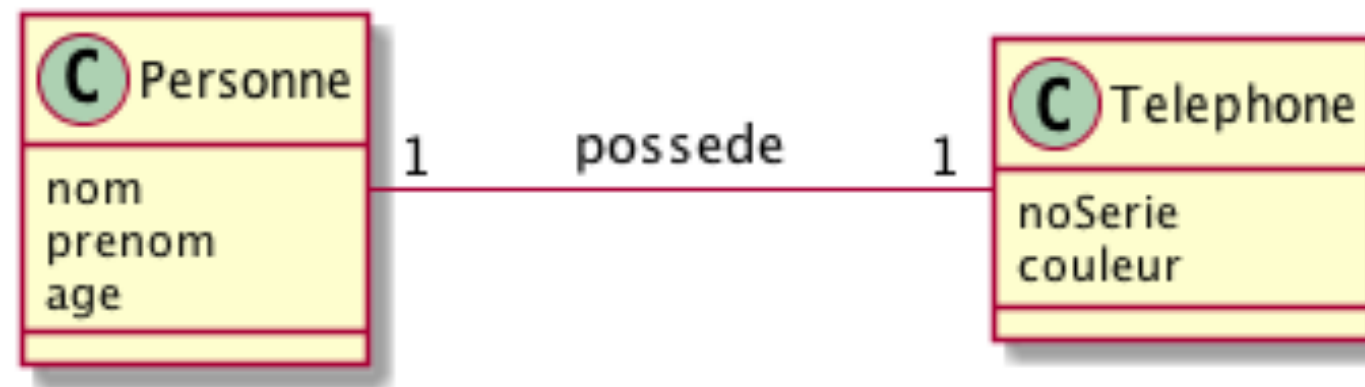
- Là, ça se complique...
- La **méthode de conversion** à utiliser **dépend des cardinalités** de l'association
- Existe **plusieurs approches** possibles pour **certaines cardinalités**

Association 1:1 -> Table



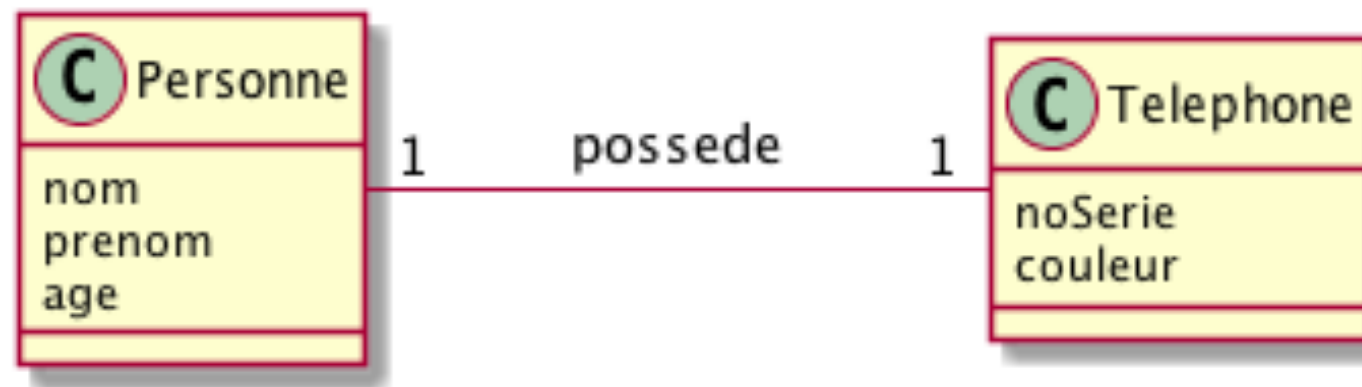
- Trois tables (approche simple)
 - **Personne**(*idPersonne*, *nom*, *prenom*, *age*)
 - **Telephone**(*noSerie*, *couleur*)
 - **Propriete**(*idPersonne*, *noSerie*)
- Quelle clé primaire choisir pour **Propriete** ?

Association 1:1 -> Table



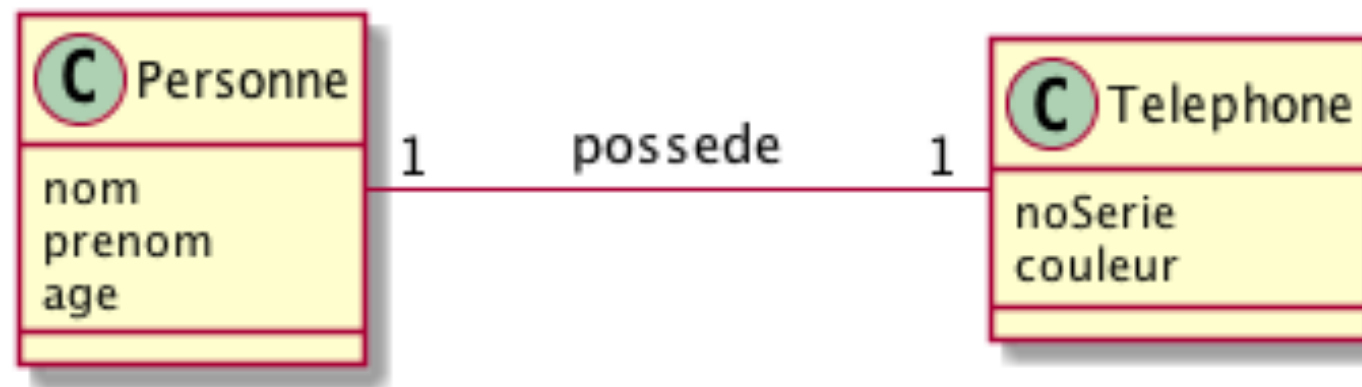
- Trois tables (approche simple)
 - **Personne**(idPersonne, nom, prenom, age)
 - **Telephone**(noSerie, couleur)
 - **Propriete**(idPersonne, noSerie) ou **Propriete**(idPersonne, noSerie)

Association 1:1 -> Table



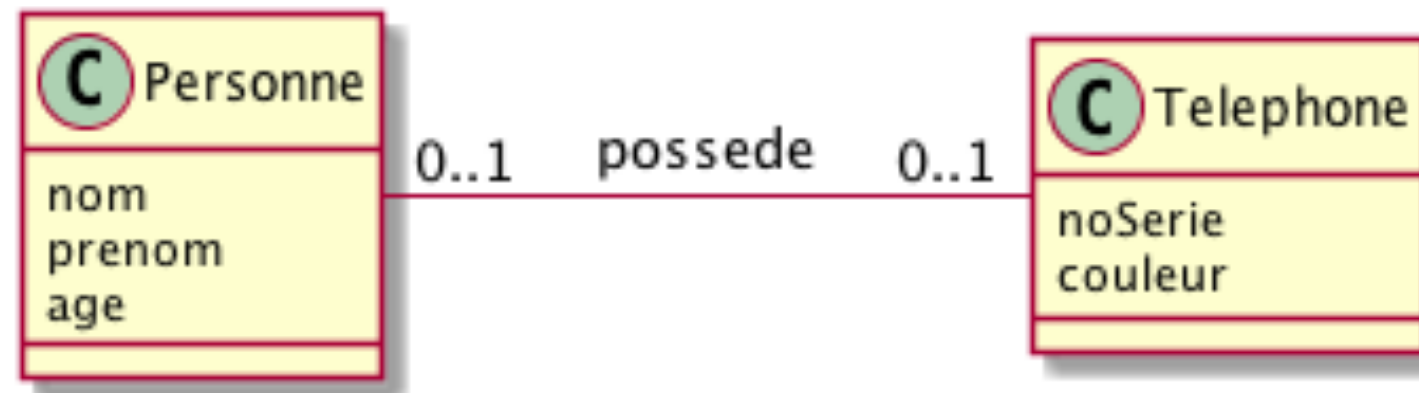
- Une seule table
 - **Personne**(*idPersonne*, *nom*, *prenom*, *age*, *noSerie*, *couleur*)
- Quelle clé primaire choisir pour **Personne** ?

Association 1:1 -> Table



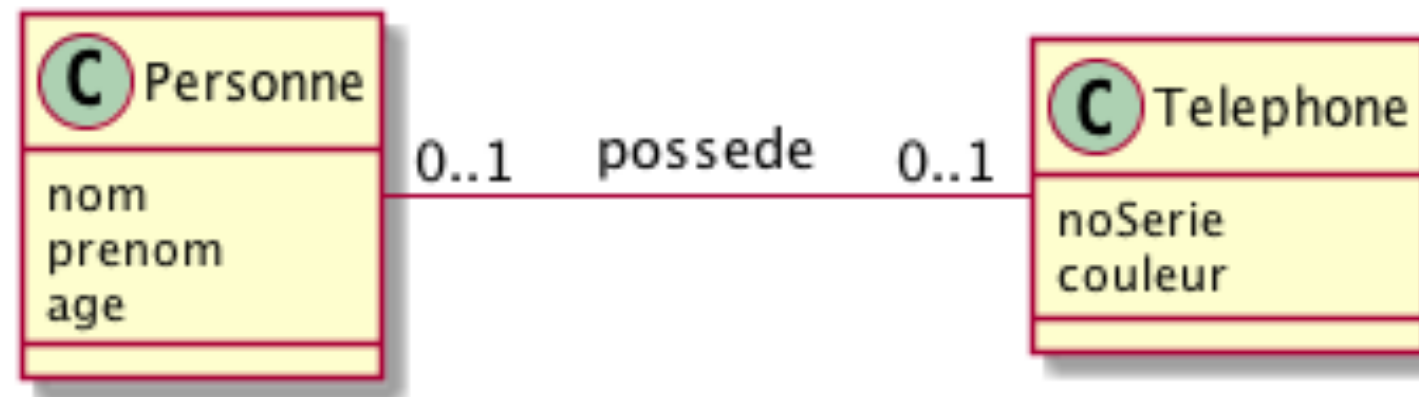
- Une seule table
- **Personne**(*idPersonne*, *nom*, *prenom*, *age*, *noSerie*, *couleur*) ou **Personne**(*idPersonne*, *nom*, *prenom*, *age*, *noSerie*, *couleur*)

Association 0..1:0..1 -> Table



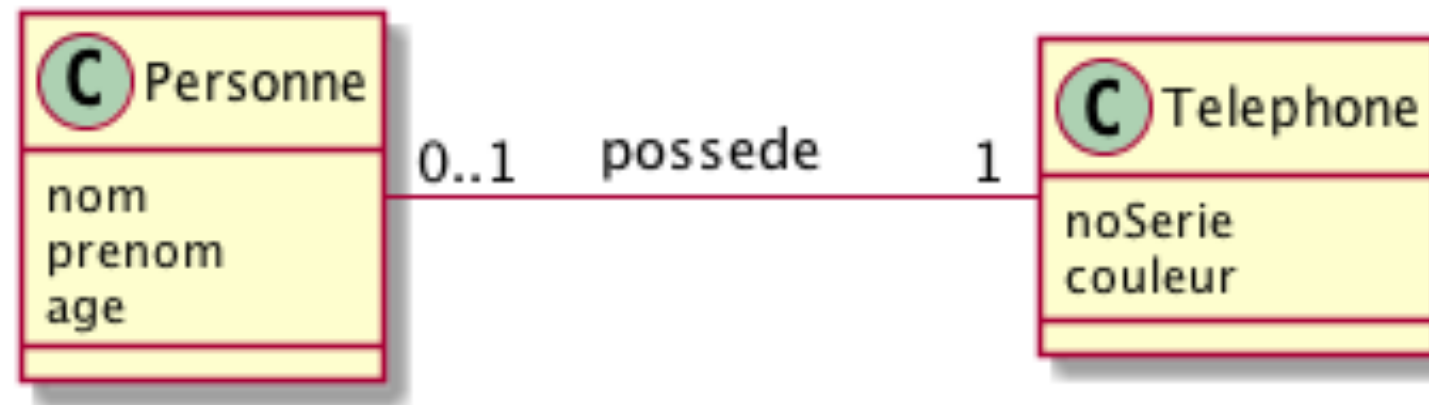
- Trois tables
 - **Personne**(idPersonne, nom, prenom, age)
 - **Telephone**(noSerie, couleur)
 - **Propriete**(idPersonne, noSerie)
- Quelle clé primaire choisir pour **Propriete** ?

Association 0..1:0..1 -> Table



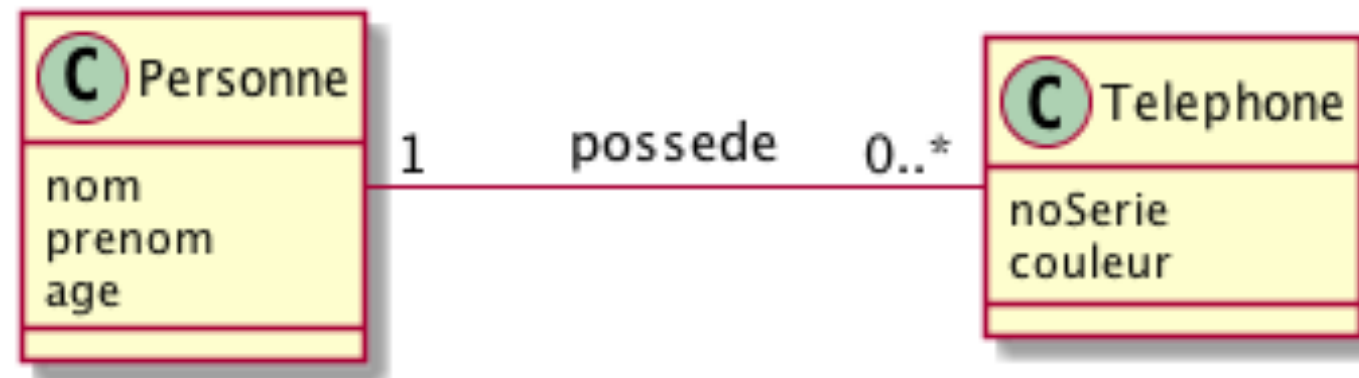
- Trois tables
 - **Personne**(*idPersonne*, *nom*, *prenom*, *age*)
 - **Telephone**(*noSerie*, *couleur*)
 - **Propriete**(*idPersonne*, *noSerie*) ou **Propriete**(*idPersonne*, *noSerie*)

Association 0..1:1 -> Table



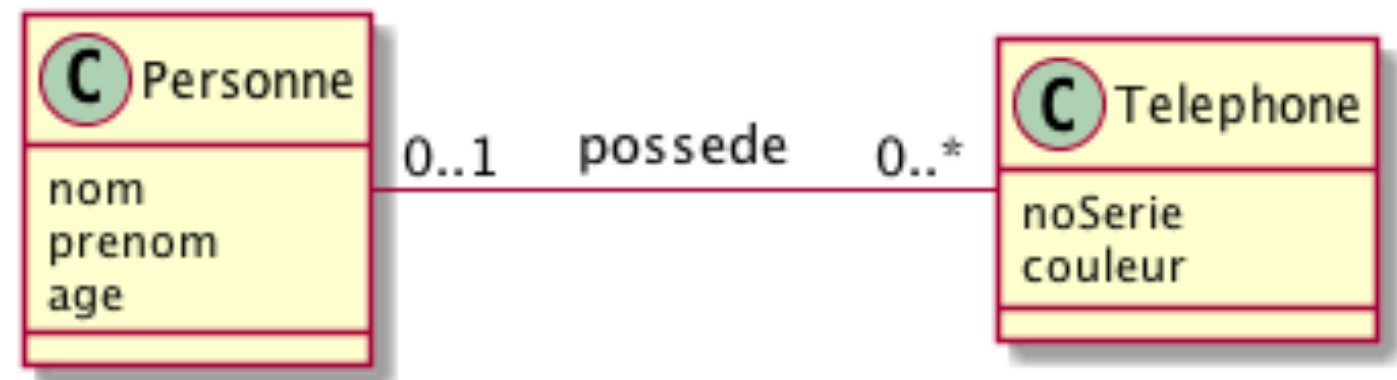
- Deux tables
 - **Personne**(*idPersonne*, *nom*, *prenom*, *age*, *noSerie*)
 - **Telephone**(*noSerie*, *couleur*)
- Pourquoi pas juste 1 table ?
 - Un téléphone peut exister indépendamment

Association 1:0..n -> Table



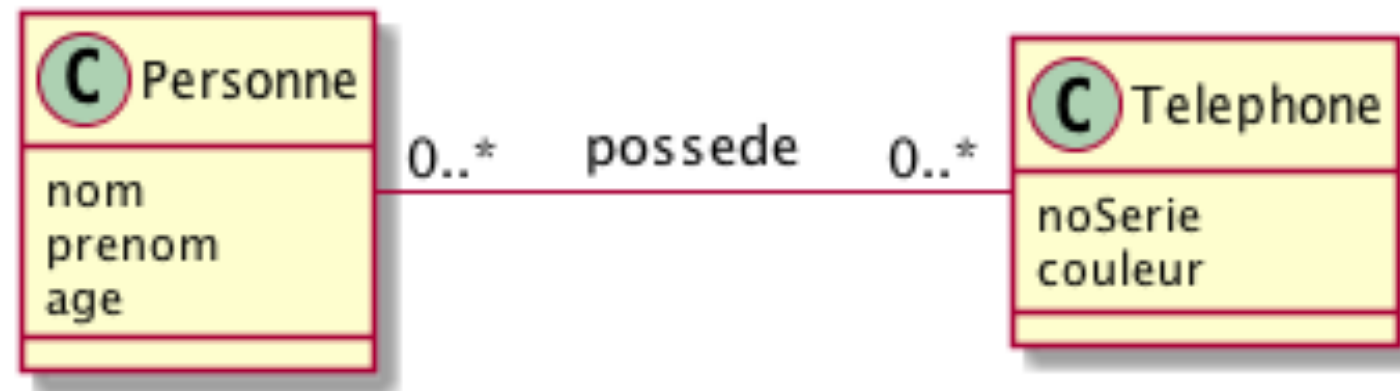
- Deux tables
 - **Personne**(idPersonne, nom, prenom, age)
 - **Telephone**(noSerie, couleur, idPersonne)
- Pourquoi *idPersonne* n'est pas une clé possible de **Telephone** ?
 - Une personne peut avoir plusieurs téléphones

Association 0..1:0..n -> Table



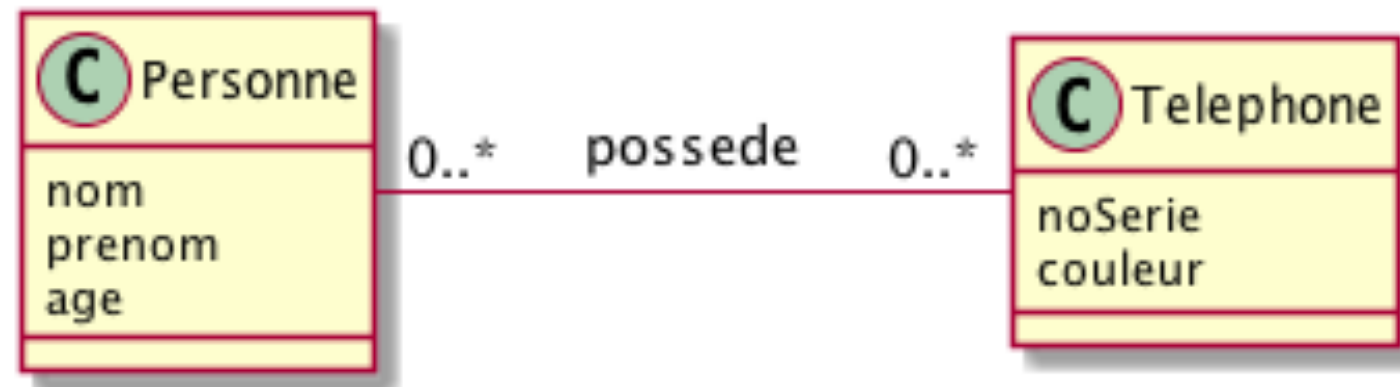
- Trois tables
 - **Personne**(*idPersonne*, *nom*, *prenom*, *age*)
 - **Telephone**(*noSerie*, *couleur*)
 - **Propriete**(*idPersonne*, *noSerie*)
- Pourquoi *idPersonne* n'est pas une clé possible de **Propriete** ?
 - Une personne peut avoir plusieurs téléphones

Association 0..n:0..n -> Table



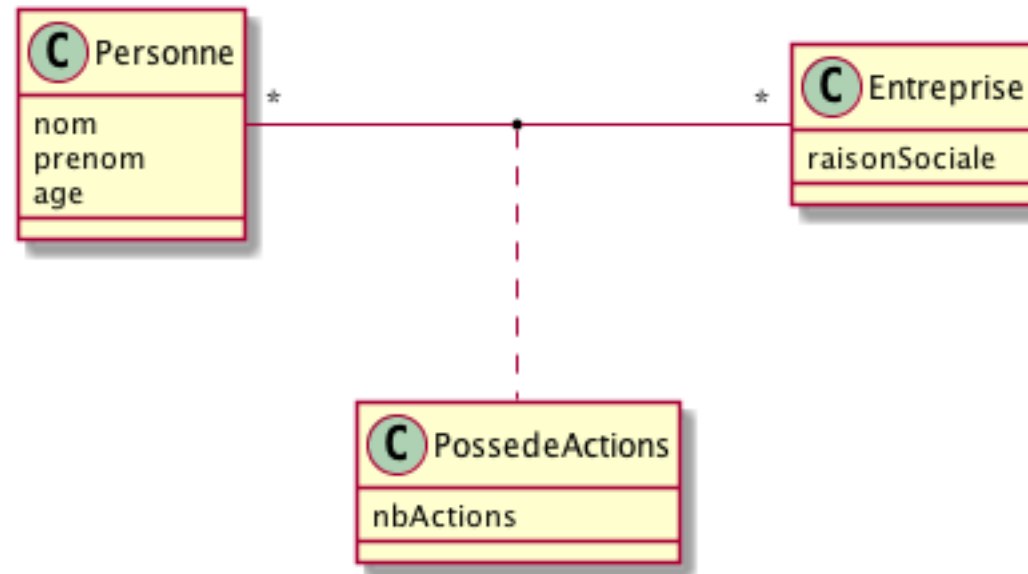
- Trois tables
 - **Personne**(*idPersonne*, *nom*, *prenom*, *age*)
 - **Telephone**(*noSerie*, *couleur*)
 - **Propriete**(*idPersonne*, *noSerie*)
- Quelle clé primaire choisir pour **Propriete** ?

Association 0..n:0..n -> Table



- Trois tables
 - **Personne**(idPersonne, nom, prenom, age)
 - **Telephone**(noSerie, couleur)
 - **Propriete**(idPersonne, noSerie)

Exemple Associations -> Table



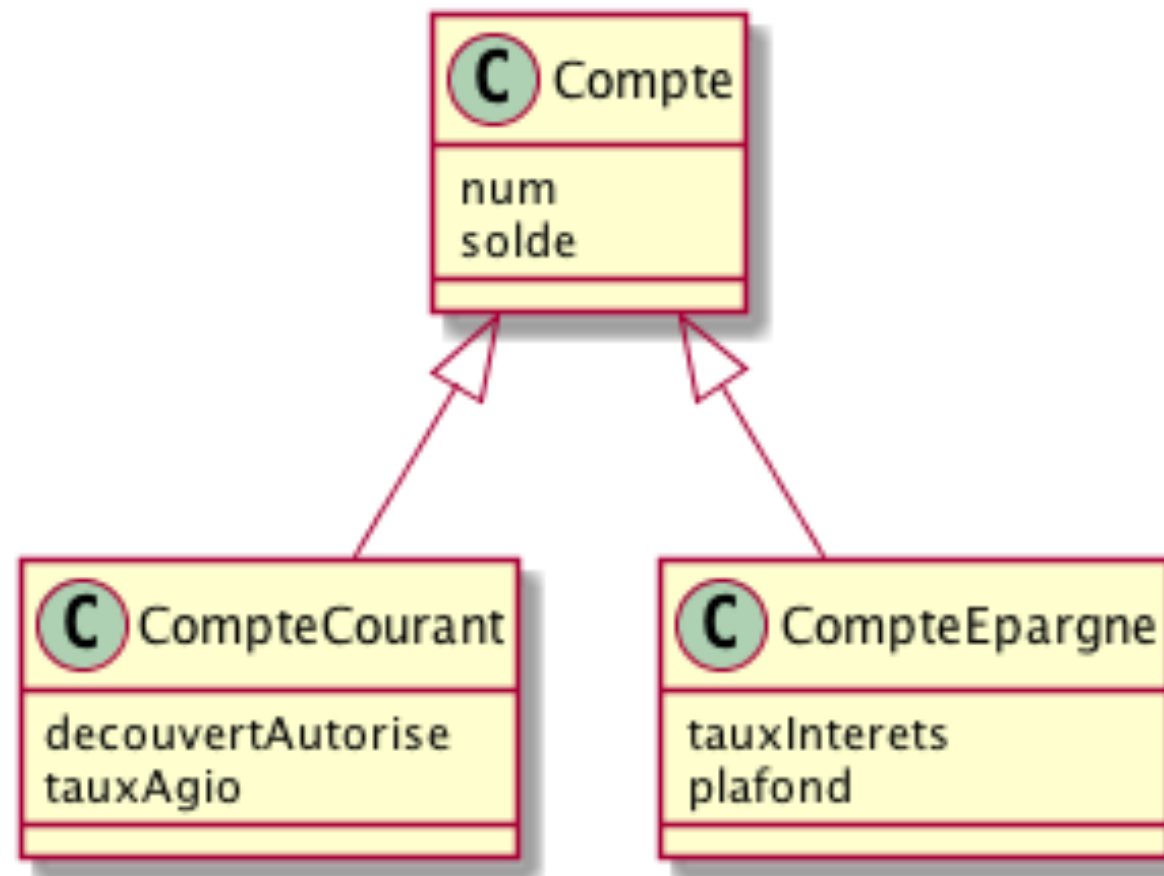
équivalent à

| idPersonne | idEntreprise | nbActions |
|------------|--------------|-----------|
| ... | ... | ... |

PossedeActions(*idPersonne*, *idEntreprise*, nbActions)

Hiérarchie de classes -> Table

- 3 approches possibles
- Considérons l'exemple suivant :



Approche 1

- Une **table** pour la **super-classe** + une **table** par **spécialisation**

| num | solde |
|-----|-------|
| ... | ... |

Compte(num, solde)

| num | decouvertAutorise | tauxAgio |
|-----|-------------------|----------|
| ... | ... | ... |

CompteCourant(num, decouvertAutorise, tauxAgio)

| num | tauxInterets | plafond |
|-----|--------------|---------|
| ... | ... | ... |

CompteEpargne(num, tauxInterets, plafond)

Approche 1 - commentaires

- Solution la plus **générale**
- L'utilisation de la **même clé** dans les **différentes relations** permet de **faire le lien**
- Peut impacter négativement les performances lors de la manipulation des données

Approche 2

- Une **table** par **spécialisation**

| num | solde | decouvertAut | tauxAgio |
|-----|-------|--------------|----------|
| ... | ... | ... | ... |

CompteCourant(num, solde, decouvertAutorise, tauxAgio)

| num | solde | tauxInterets | plafond |
|-----|-------|--------------|---------|
| ... | ... | ... | ... |

CompteEpargne(num, solde, tauxInterets, plafond)

- Bien si les spécialisations sont disjointes
- Pas toujours applicable sinon

Approche 3

- Une **table globale**

| num | solde | decouvert... | tauxAgio | tauxInterets | plafond |
|-----|-------|--------------|----------|--------------|---------|
| ... | ... | ... | ... | ... | ... |

Compte(num, solde, *decouvertAutorise*, *tauxAgio*, *tauxInterets*, *plafond*)

- Simple, pas très élégant, mais efficace
- Attention au nombre de valeurs nulles

Réflexions sur Hiérarchie de classes -> Table

- On n'est **plus capable de distinguer** les relations de spécialisation/généralisation des autres associations
- On observe une **perte de sémantique** au passage au schéma relationnel

Composition -> Table

- Les relations de composition sont **représentées comme les associations**
- On n'est **plus capable de distinguer** les relations de composition des autres associations
- On observe une **perte de sémantique** au passage au schéma relationnel

Résumé

- Modèle relationnel décrit notre base de données sous la forme de **relations**
- Besoin de convertir notre schéma conceptuel en **schéma relationnel** pour utiliser SGBD relationnel
- Les **entités** sont converties en **relations**
- La méthode de **conversion des associations** dépend des **cardinalités**
- **Différentes approches** possibles pour **convertir** les **hiérarchies** de classes

Supports

- Cours et sujet de TD disponibles
 - Sur Arche
 - Sur mon repo Github : <http://cpc.cx/sxH>