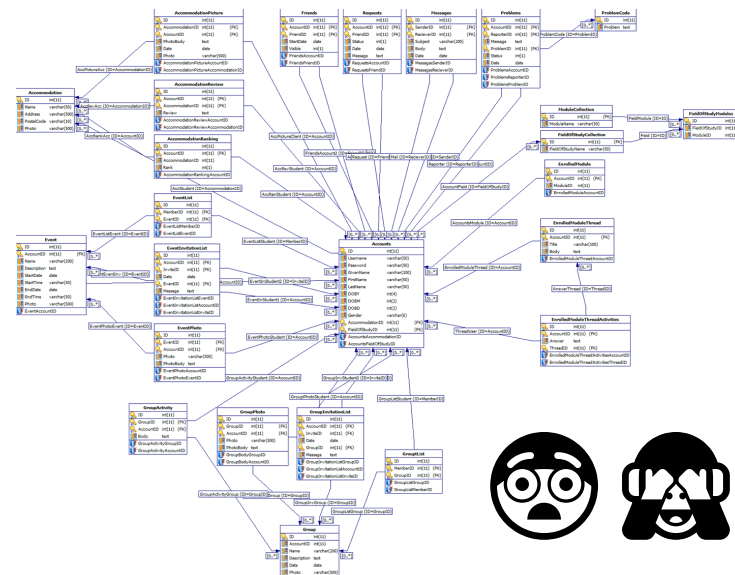






Semaine 1 – Partie 2

Modélisation conceptuelle

Bases de données et programmation Web





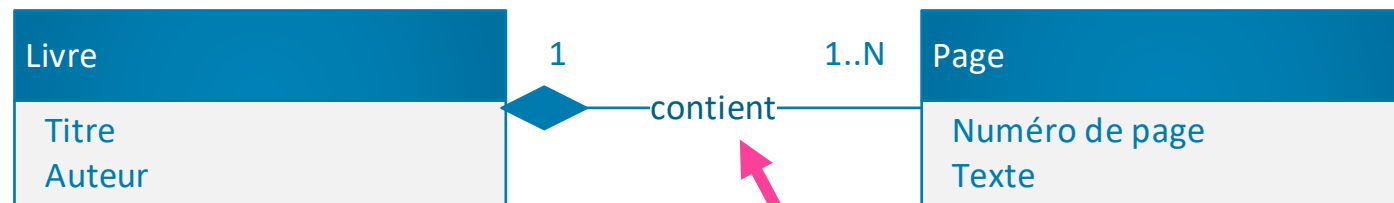
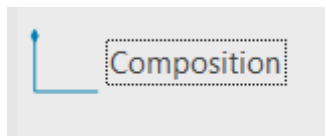
- ❖ Composition et agrégation  
- ❖ Généralisation et spécialisation  
- ❖ Gros exemple



❖ Composition

- ◆ Relation forte qui sert à décrire une entité parent composée d'une ou plusieurs autres entités composites.
 - Représentée par un lien avec un losange noirci à l'extrémité parent.
- ◆ Dans une relation de composition, si l'entité parent est supprimée, les entités composites ne peuvent pas exister et sont supprimées aussi.

Exemple : « Un **livre** est décrit par son **auteur**, son **titre** et son **nombre de pages**. De plus, un livre contient plusieurs **pages**, avec chacune du **texte** et un **numéro de page**. Si on supprime un livre, ses pages n'existent plus. »

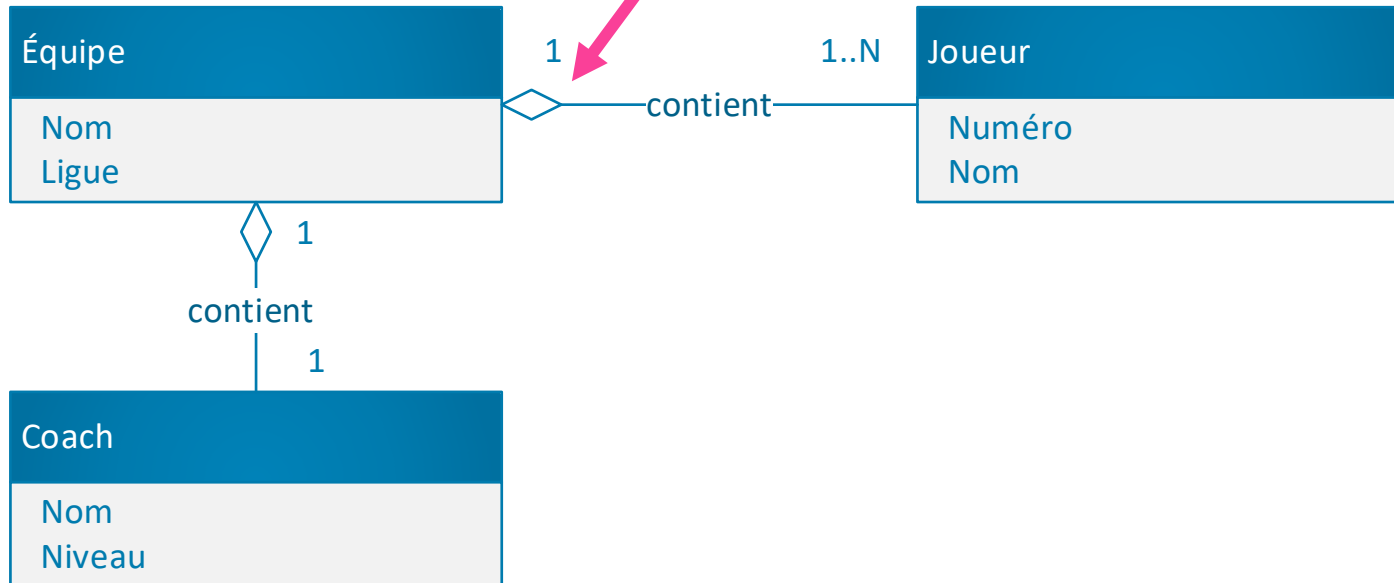
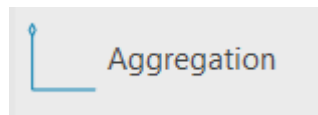


Si possible, on peut mettre les **cardinalités** qui décrit combien de pages font partie de combien livres. Dans ce cas, c'est **1** livre pour plusieurs pages, donc **1..N**.



❖ Agrégation

- ◆ Relation forte similaire à une composition. La différence, c'est qu'une entité enfant (composite) peut continuer d'exister même sans son entité parent. (Composée)
 - Représentée par un lien avec un losange vide à l'extrémité parent.

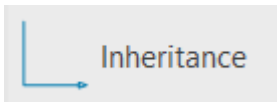


Exemple : « Une **équipe de sport** (Nom, ligue) est composée d'un **coach** (nom) et de plusieurs **joueurs**. (Nom, numéro) Si une équipe de sport est supprimée, on souhaiterait pouvoir assigner le coach et les joueurs à une autre équipe. »



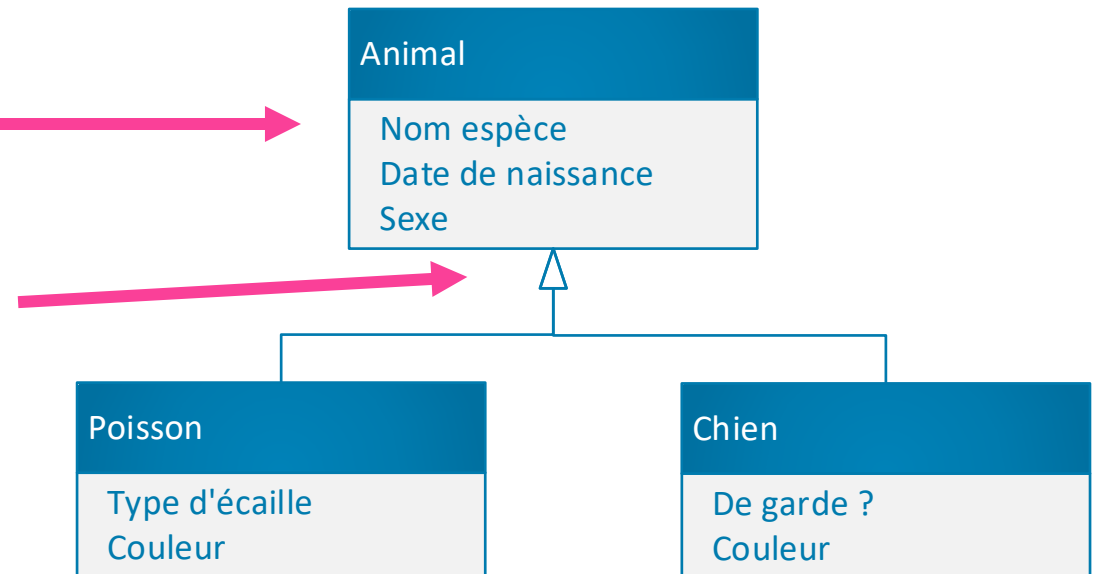
❖ Généralisation 🐾🐕

- ◆ Lorsqu'on remarque que 2+ entités ont plusieurs *propriétés communes*, on peut créer une « super-entité » plus générale pour éviter la redondance. 📦📦



Les **propriétés communes** des **sous-entités** sont attribuées à la **super-entité**.

La généralisation est une relation représentée à l'aide de l'héritage. (Flèche vide)



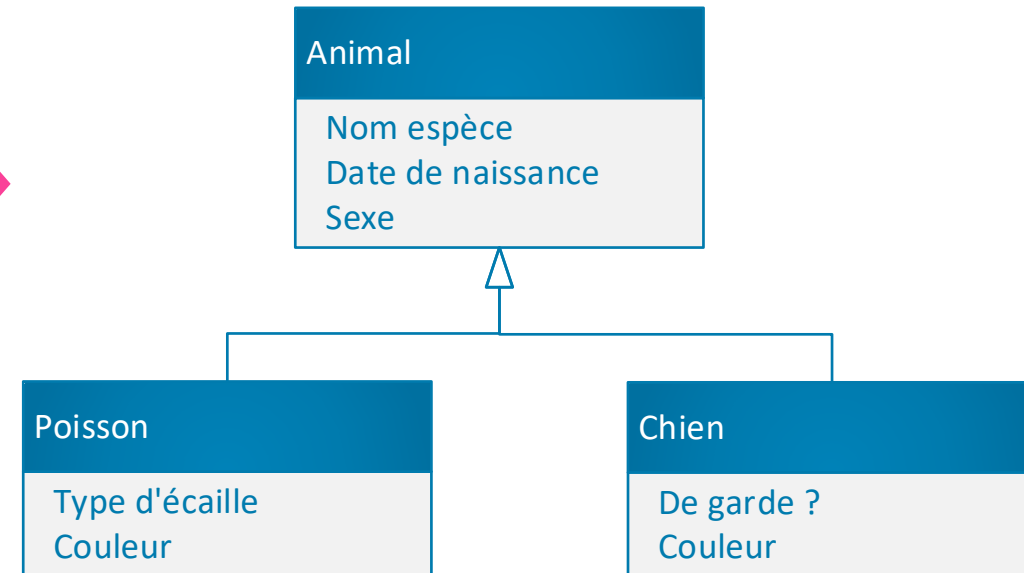
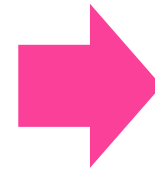
Notez que la généralisation peut très bien être utilisée sur **plusieurs couches**. (Avec modération sinon cela complexifie l'interaction avec les données) Ex. un **chien** est un **mammifère** et un mammifère est un **animal**.



❖ Généralisation

◆ Processus :

- Je remarque que j'ai 2+ entités qui ont quelques attributs communs et qui pourraient appartenir à une famille plus générale. 🧑🧒
- Je rassemble les attributs communs dans une **super-entité** et je la relie aux **sous-entités**.



Un instant ! 😞 Pourquoi **couleur** n'a pas été rangé dans la **super-entité** ? C'est un choix volontaire. Par exemple, on prévoit ajouter d'autres sous-entités animales dans le futur et on anticipe le fait que, pour certains animaux, on ne veut pas noter la couleur. (ex. **Zèbre** 🦓) Ainsi, ce genre de décision peut dépendre des besoins (actuels et futurs) de la base de données.

Nous pourrions avoir une réflexion similaire pour l'attribut **sexe** si on songe à stocker des données sur des **vers de terre** et des **méduses**, qui sont hermaphrodites.



❖ Spécialisation

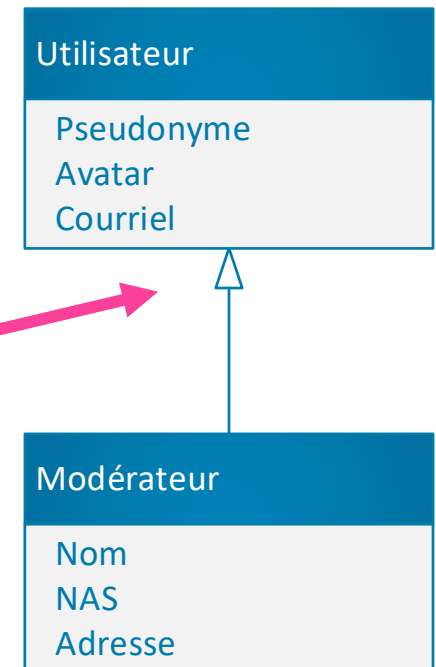
- ◆ Presque exactement l'inverse de la généralisation. Lorsqu'on a une entité et qu'on souhaite créer une nouvelle entité similaire, mais avec certaines propriétés supplémentaires, on **spécialise**. 

Exemple : Sur un forum, on a des **utilisateurs**. (Pseudonyme, avatar, courriel et nombre de messages postés.

Cela dit, on aimerait également avoir des **modérateurs**.
Imaginons que ces modérateurs sont réénumérés : on a alors besoin d'informations comme le nom, le NAS et l'adresse du modérateur puisque c'est un employé.

Étant donné qu'un modérateur partage beaucoup d'attributs avec un utilisateur, on **spécialise** l'utilisateur en créant la nouvelle entité nommée Modérateur.

Représenté avec
l'héritage également.



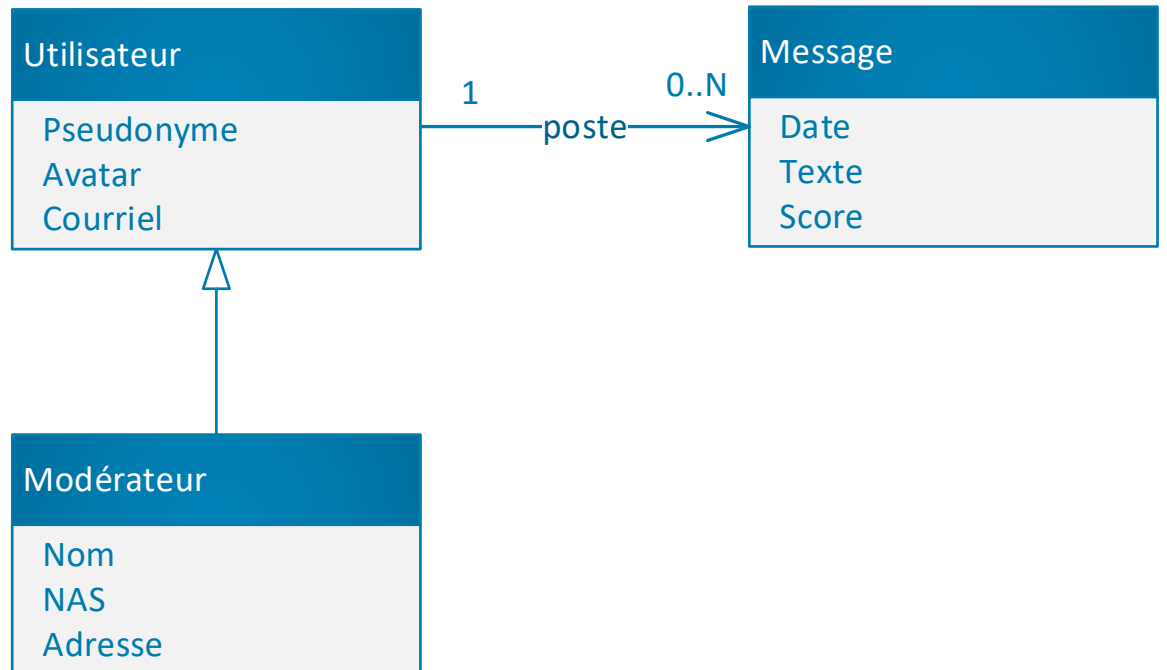


❖ Spécialisation et Généralisation

- ◆ Partage de relations : Puisqu'un modérateur EST un utilisateur, il partage les relations que l'utilisateur possède.

Exemple :

- Un utilisateur peut poster un message. **Message** et **Utilisateur** sont reliés parce qu'on voudra pouvoir les associer dans la base de données. (Ex. Pour récupérer la liste des messages d'un utilisateur)
- Puisqu'un **modérateur** EST un **utilisateur**, il peut lui aussi **poster des messages**. (C'est comme si la relation **<poste>** était à la fois reliée à Utilisateur et à Modérateur)



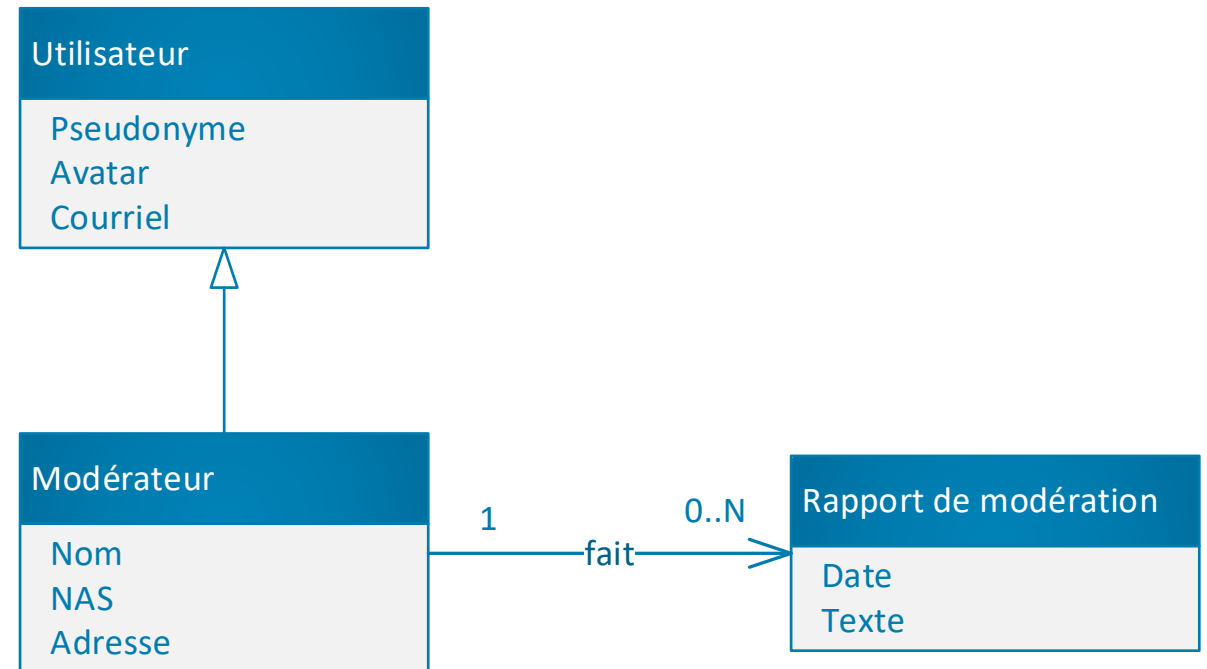


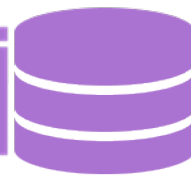
❖ Spécialisation et Généralisation

- ◆ Partage de relations : L'inverse n'est pas vrai. Même si le modérateur peut faire un rapport de modération, l'utilisateur n'a pas accès à cette relation.

Exemple :

- Un **modérateur** peut faire un **rapport de modération**.
- Toutefois, un **utilisateur** ne peut pas.
- La relation **<Fait>** est réservée à la spécialisation **Modérateur**.



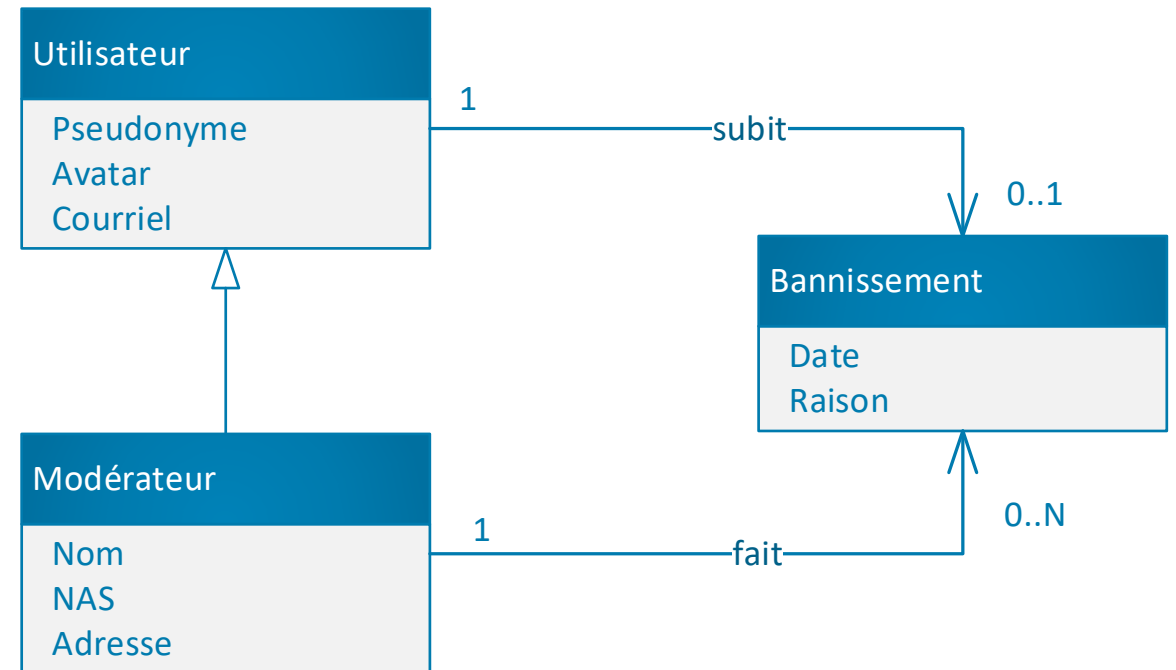


❖ Spécialisation et Généralisation

- ◆ Partage de relations : On peut également créer des relations un peu moins évidentes. La logique restera la même.

Exemple :

- Un **utilisateur** peut **subir** un **bannissement**. Un **modérateur** peut également **subir** un **bannissement**, car c'est également un utilisateur.
- Un **modérateur** peut faire un **bannissement**. Toutefois, un **utilisateur** **ne peut pas bannir**, bien entendu.
- En résumé, modérateur possède la relation **<Subit>** et **<Fait>** avec l'entité bannissement.



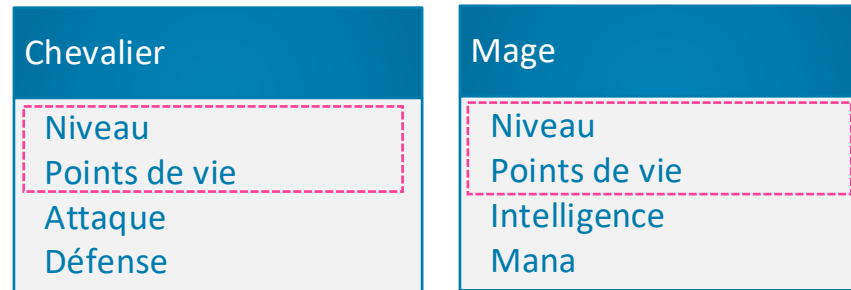


❖ Généralisation ou spécialisation ?

- ◆ Un instant... la généralisation et la spécialisation... c'est la même chose ?

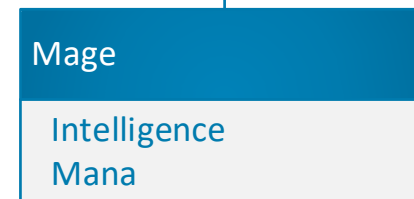
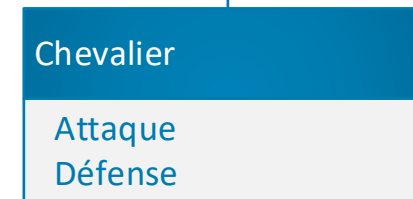
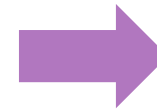
Généralisation

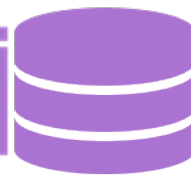
« Oups, j'ai des attributs redondants pour des entités similaires ! Je vais généraliser »



Spécialisation

« J'aimerais créer des types de personnages plus précis, je vais spécialiser en me servant de mon entité Personnage ! »



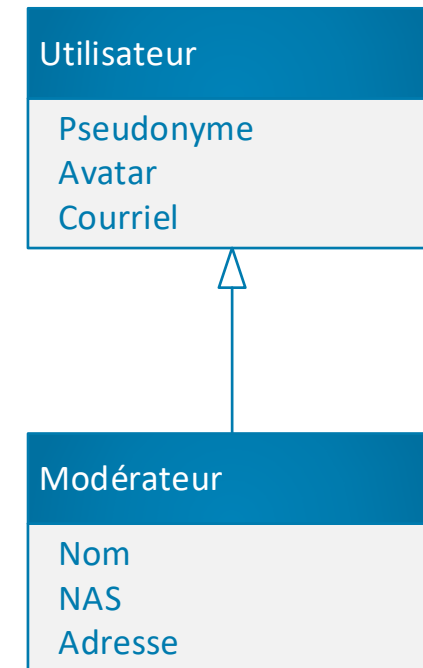


❖ Généralisation ou spécialisation ?

- ◆ Un instant... la généralisation et la spécialisation... c'est la même chose ?
 - Oui ! Le résultat est « le même », c'est seulement le processus qui est différent.
- ◆ Petite exception : c'est forcément de la **spécialisation** s'il n'y a qu'une seule sous-entité.

Spécialisation

- J'ai déjà l'entité **Utilisateur**, qui est auto-suffisante et représente les usagers d'un forum de discussion.
- J'aimerais créer un seul nouveau type d'utilisateur avec des propriétés supplémentaires : je **spécialise** en créant la sous-entité **Modérateur**.
- C'est impossible de prétendre qu'il s'agit d'une **généralisation** si Modérateur est le seul sous-type, parce que ça n'aurait éliminé aucune **redondance** de **généraliser**.





❖ En résumé

◆ Entités

◆ Attributs

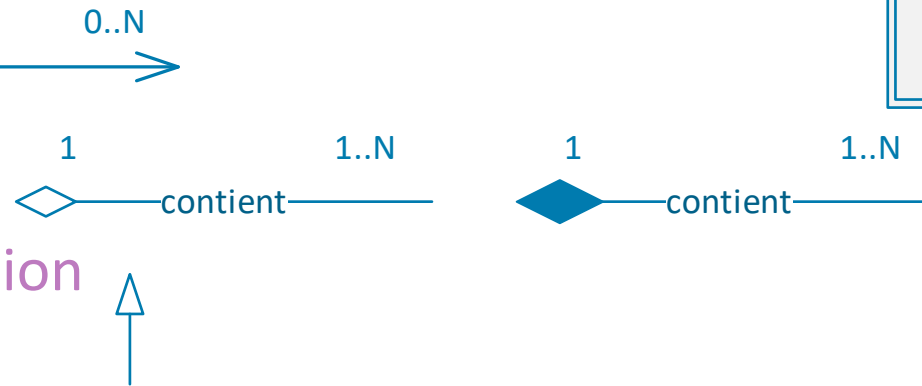
- Atomique, à valeurs multiples, composite, dérivé, optionnel

◆ Relations

- Cardinalité, Degré

◆ Composition et agrégation

◆ Généralisation et spécialisation



Entité

Atomique

Optionnel [0..1]

Multiples [1..N]

\Dérivé

Composé

Atomique

Atomique

Atomique

◆ Et maintenant ?

- Il faut être capable de lire un énoncé de besoins et le transformer en schéma conceptuel à l'aide de tous ces outils de schématisation ! 🧠🧠🧠



❖ Pour illustrer le processus complet de schématisation, voici un gros exemple

« Une boutique d'abat-jours souhaite tenir une base de données pour stocker des informations sur leurs clients et leurs achats. Le but est d'identifier le profil et les préférences d'un client pour ensuite lui envoyer des courriels et des dépliant publicitaires adaptés à ses préférences.

Lorsqu'un compte client est créé, on a absolument besoin des infos suivantes : courriel (unique pour le client), prénom, nom, nom complet, téléphone(s) et date de naissance (jour, mois et année). Si le client accepte, on peut aussi noter son adresse, mais ce n'est pas obligatoire.

Les clients peuvent acheter des abat-jours style rond, de style empire et de style carré. Dans tous les cas, on doit noter la hauteur, la couleur et le prix de l'abat-jour pour cerner les préférences du client. De plus, pour les trois types d'abat-jours, on note des infos supplémentaires : style rond -> rayon, style empire -> rayon supérieur et rayon inférieur et style carré -> taille du côté.

Enfin, les clients peuvent avoir une carte de membre (de type or, diamant ou platine) sur laquelle on note les points bonis accumulés. Si jamais on doit supprimer un compte client, inutile de conserver les données de sa carte de membre. »



Rond



Carré



Empire



Premier paragraphe

« Une boutique d'**abat-jours** souhaite tenir une base de données pour stocker des informations sur leurs **clients** et leurs **achats**. Le but est d'identifier le **profil** et les **préférences** d'un client pour ensuite lui envoyer des **courriels** et des **dépliants publicitaires** adaptés à ses préférences. [...] »

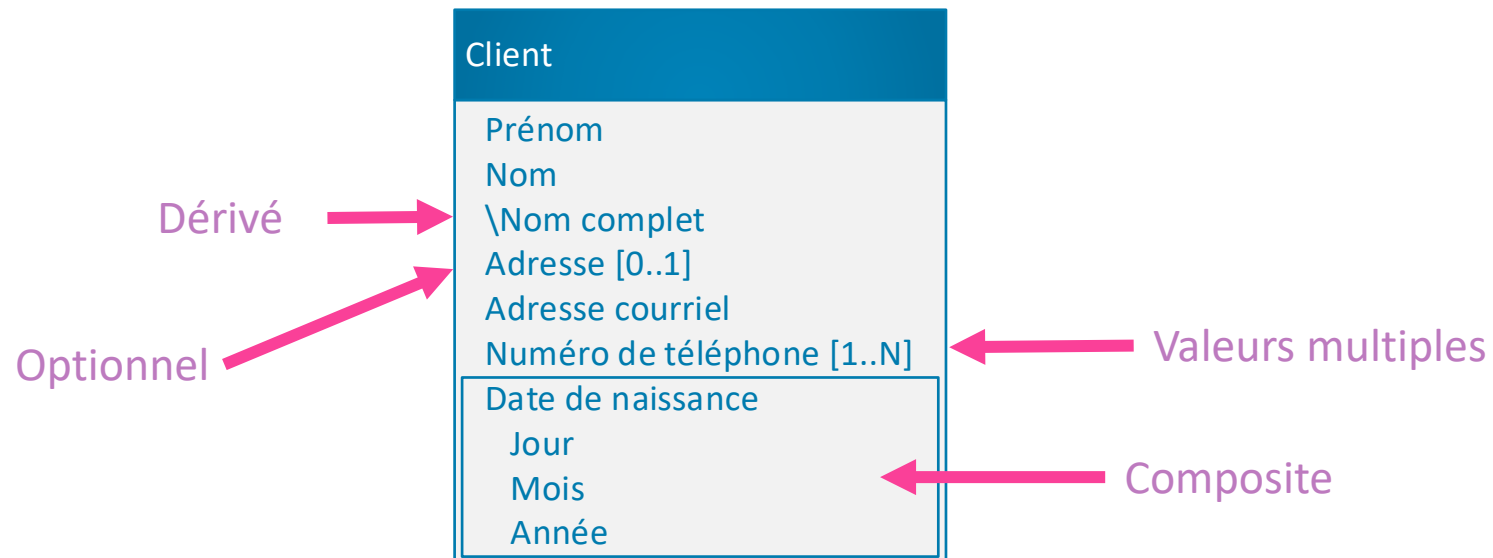
- Pour le moment on voit quelques **entités** potentielles, mais on ne voit pas vraiment d'**attributs**... (On voit des objectifs / fonctionnalités) Si on n'est pas capable d'associer une entité à des attributs, c'est parce qu'on n'en a pas besoin dans la base de données.
- Attendons la suite du texte pour se décider sur les entités.



Deuxième paragraphe

« [...] Lorsqu'un **compte client** est créé, on a absolument besoin des infos suivantes : **courriel** (unique pour le client), **prénom**, **nom**, **nom complet**, **téléphone(s)** et **date de naissance** (jour, **mois** et **année**). Si le client accepte, on peut aussi noter son **adresse**, mais ce n'est pas obligatoire. [...] »

- Ah ! Ici c'est clair que **client** (ou compte client) est une entité à laquelle on peut associer de nombreux attributs.



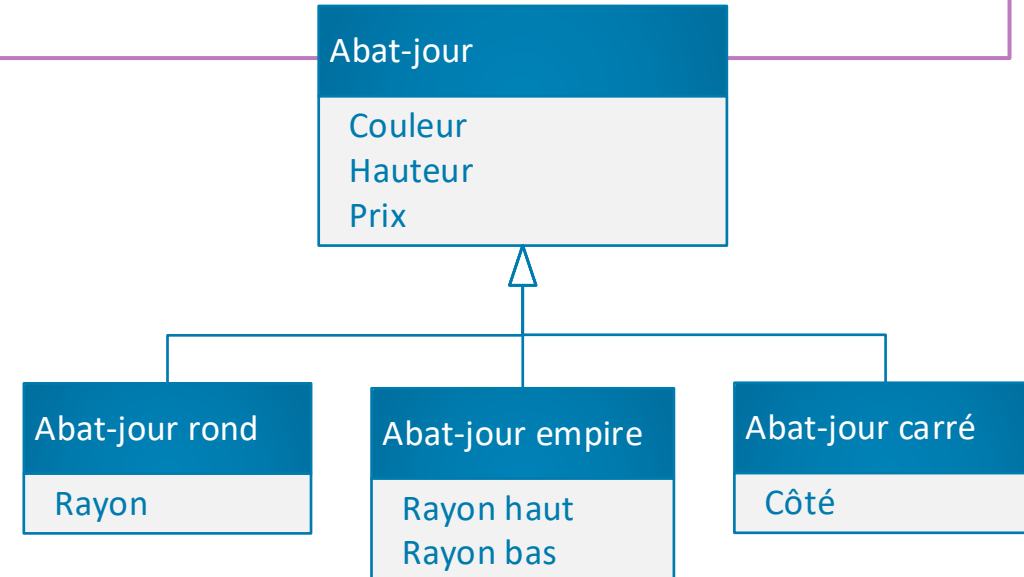


Troisième paragraphe

Les **clients** peuvent acheter des **abat-jours** style **rond**, de style **empire** et de style **carré**. Dans tous les cas, on doit noter la **hauteur**, la **couleur** et le **prix** de l'abat-jour pour cerner les **préférences** du client. De plus, pour les trois types d'abat-jours, on note des infos supplémentaires : style rond -> **rayon**, style empire -> **rayon supérieur** et **rayon inférieur** et style carré -> **taille du côté**.

- Ici on remarque clairement des attributs pour des entités de type **Abat-jour**, dont certains attributs sont communs aux trois styles d'abat-jours. Pour éviter la redondance, la **généralisation / spécialisation** est de mise.

- De mon côté, je vois plutôt **Abat-jour** comme une **généralisation**, car l'entité **Abat-jour** n'est pas auto-suffisante (un abat-jour appartient forcément à un des trois styles, on n'a pas d'abat-jour sans style spécifique) Ultimement, une **spécialisation** n'aurait pas été incorrecte de toute façon.

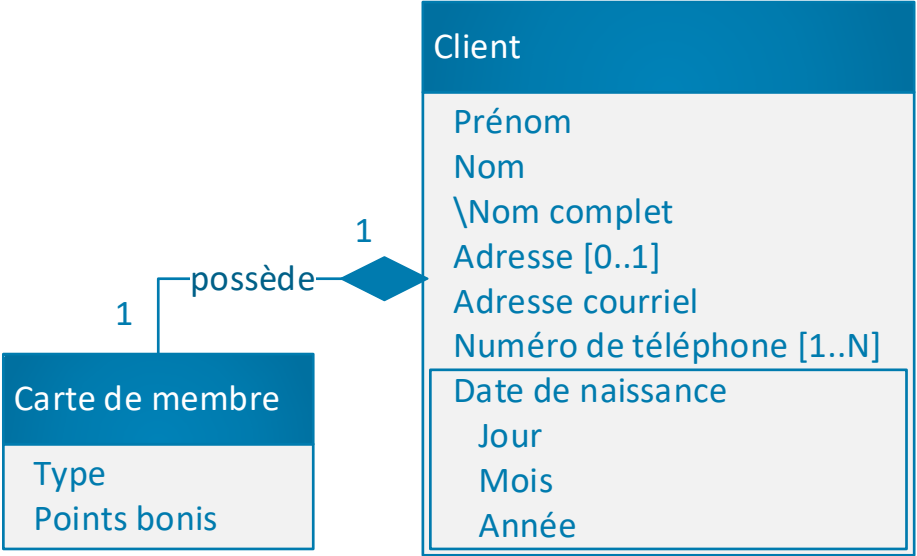




Quatrième paragraphe

Enfin, les **clients** peuvent avoir une **carte de membre** (de **type** or, diamant ou platine) sur laquelle on note les **points bonis** accumulés. Si jamais on doit supprimer un compte client, inutile de conserver les données de sa carte de membre.

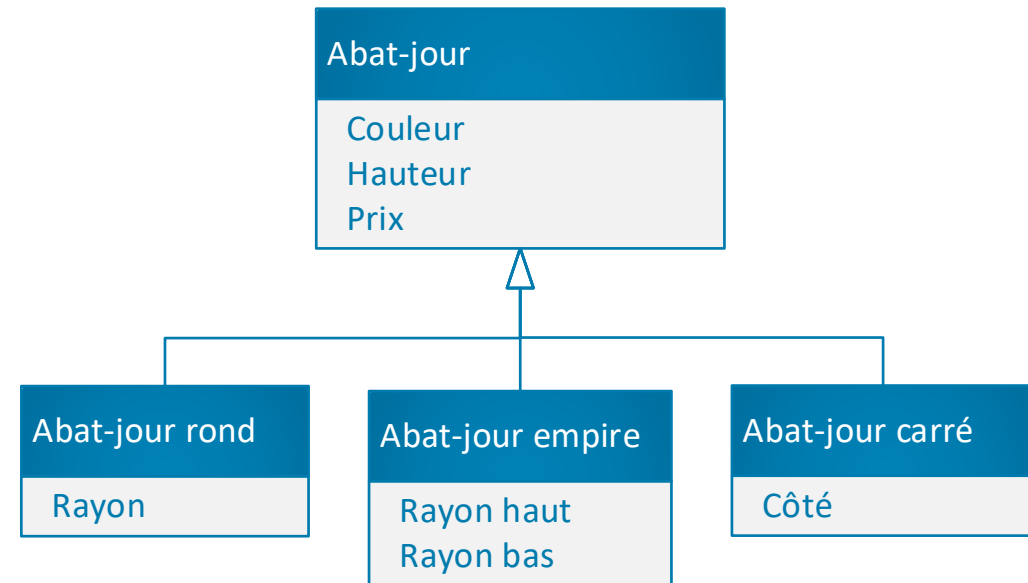
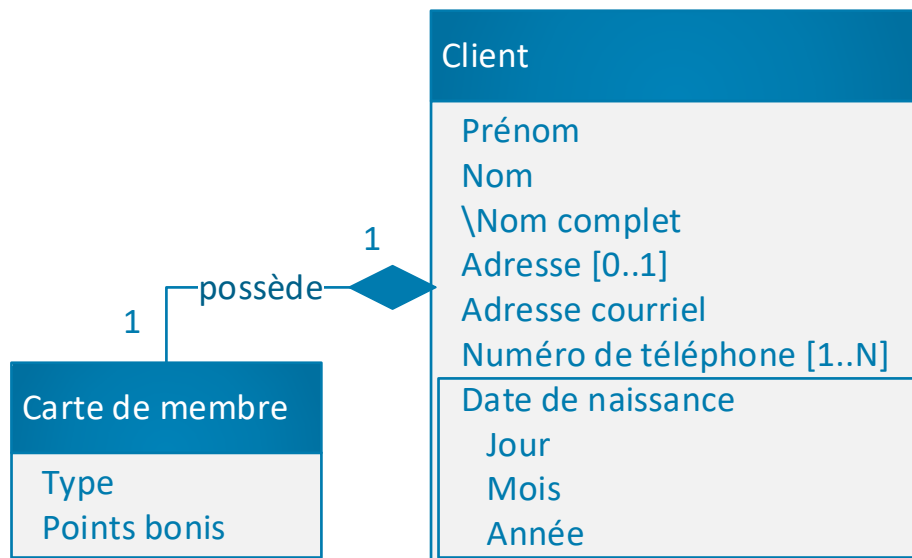
- Ici il est surtout question de l'entité **Carte de membre**. Une carte de membre semble avoir une relation spéciale avec un **client** : elle lui appartient. (mais elle est optionnelle) D'ailleurs, on doit supprimer les données de la carte de membre si le client est supprimé : ça ressemble à une **composition**.
- J'en profite pour mettre la cardinalité **1** ainsi qu'une **ligne pointillée** pour la relation de **composition** pour garder à l'esprit qu'une **carte de membre** est **optionnelle** et qu'un **client** ne peut en avoir **qu'une seule**. (Si je n'avais pas mis ces 2 indices, ça n'aurait pas été dramatique)





Prendre un peu de recul et relire les besoins si nécessaire

- Je remarque que mon schéma est « *détaché* ». C'est généralement suspect : il manque peut-être des **relations**.
- En relisant le texte, je me souviens que le but est d'identifier le profil d'acheteur des clients en notant les abat-jours qu'ils achètent. Cela signifie qu'**un lien doit exister** entre **client** et **abat-jour**.





Prendre un peu de recul et relire les besoins si nécessaire

- Cela semble plus complet maintenant ! Les fonctionnalités / besoins énoncés semblent mieux adressés.
- Il ne faut pas hésiter à relire l'énoncé des besoins et si on a l'impression qu'un aspect n'est pas représenté dans notre schéma, y faire des changements.

