

# Bases de Données 1

Matthieu Nicolas

Polytech S5 - EMME

Slides réalisées à partir de celles d'Abir Ismaili-Alaoui, Claude Godart et  
Malika Smaïl

# Se présenter

- [matthieu.nicolas@univ-lorraine.fr](mailto:matthieu.nicolas@univ-lorraine.fr)
- Doctorant, équipe COAST, LORIA
- Algorithmes pour l'édition collaborative pair-à-pair

# Organisation

- **Volume horaire** : 18h
- **CMs** : 2 séances de 2h
  - Responsable du module et CMs : Matthieu Nicolas
- **TDs** : 6 séances de 2h
  - Groupe 1, 2 et 3 : ???
  - Groupe 4 : Matthieu Nicolas

# Arche

- Supports de cours et sujets de TD à disposition
- Infos du Module
  - *2021 - Polytech EMME : Base de Données 1*
  - Identifiant : 50014
  - Clef d'inscription : 2021-EMME-BD1
- Pour ceux n'ayant pas encore accès à Arche
  - Repo Github dispo : <http://cpc.cx/vNe>

# Évaluation

- **2 notes** pour l'évaluation
  - 1 TD noté de 2h (probablement le 13/10/2021)
  - 1 DS de 2h (12/11/2021)

# Objectifs du module

- Découvrir les notions de **BD** et **SGBD**
- Apprendre à **concevoir** une BD
- Apprendre à **utiliser** une BD

# Syllabus

- *Conception d'une BD*
  - Schéma conceptuel
  - Schéma relationnel
- *Utilisation d'une BD*
  - Langage de requêtes SQL

# Mini-bibliographie

- Designing Data-Intensive Applications - M. Kleppmann, O'Reilly, 2017
- *Fundamentals of database systems* - R. Elmasri et S. B. Navathe, The Benjamin/Cummings Pub., 2000
- *Bases de données et systèmes d'information* - N. Boudjlida, Dunod, 1999
- *Bases de données : objet et relationnel* - G. Gardarin, Eyrolles, 1999



# Bases de Données 1

## #0 - Intro

Matthieu Nicolas

Polytech S5 - EMME

Slides réalisées à partir de celles d'Abir Ismaili-Alaoui, Claude Godart et  
Malika Smaïl

# Plan

- **Problématiques et besoins** liés au stockage de données
- Notions de **Base de Données (BD)** et de **Système de Gestion de BD (SGBD)**
- **Fonctionnalités** d'un SGBD

# Problématiques et besoins liés au stockage de données

Base de Données 1  
#0 - Intro

# Stockage de données

- Chaque application dispose de ses données
- Besoin de stocker ces données
  - **Pour** une **utilisation ultérieure**
  - **Pour** les **partager** avec les autres composants de l'application ou d'autres applications
- Historiquement, utilise des **fichiers**

# Limites du Système de Gestion de Fichiers - 1

- Mauvaises performances pour un gros volume de données
  - **Lecture et écriture lente** sur disque dur par rapport à en mémoire vive
- Plusieurs façons de manipuler les fichiers
  - **Séquentielle et Directe**
  - Mais chacune a ses avantages et inconvénients...

# Limites du Système de Gestion de Fichiers - 2

- Lecture/écriture **séquentielle**
  - Parcourt le fichier **ligne après ligne**
  - Solution **simple**
  - Mais **peut pas “sauter”** à la partie qui nous intéresse
- Lecture/écriture **directe**
  - Permet d'**accéder directement** à la donnée
  - Mais besoin de **connaître son emplacement sur le disque**

# Limites du Système de Gestion de Fichiers - 3

- **Chaque application** dispose de ses **données**
  - **Redondance** des données
- **Chaque application** dispose de ses **programmes**
  - Complexité du **maintien** de la **cohérence**

# Problématiques redondance de données

- Répétition **non-nécessaire** de données
- Entraîne plusieurs problématiques
  - **Coût** en volume et en argent **dupliqué**
  - Risque d'**incohérence** entre les différentes copies
- On souhaite donc **limiter** la redondance des données



# Systèmes de Gestion de Bases de Données

- Besoin d'un outil, d'un standard pour
  - **Stocker les données** de manière **pérenne...**
  - Et **sûre...**
  - Tout en étant **performant...**
  - Et “**simple**” d'utilisation...
- Les Systèmes de Gestion de Bases de Données (SGBD) ont été proposés pour répondre à ce besoin

# Notions de Base de Données (BD) et de Système de Gestion de BD (SGBD)

Base de Données 1  
#0 - Intro

# Définition : Base de Données (BD)

- **Collection importante de données**
  - Cohérentes
  - Structurées indépendamment d'une application
  - De redondance minimale
  - Accessibles par plusieurs utilisateurs à la fois
- Apparaît au début des années 60
- Une BD est faite pour enregistrer des faits, des opérations au sein d'un organisme (administration, banque, université, hôpital...)

# Exemples d'utilisation de BDs

- **Twitter** enregistre tweets, likes, retweets et réponses dans sa BD
- **Steam** conserve librairies, succès, et messages dans sa BD
- **Netflix** stocke historiques de visionnage et likes dans sa BD

# Définition : Système de Gestion de BD (SGBD)

- Un ensemble de **programmes** permettant à des utilisateurs de **créer** et **d'utiliser** des BDs

# Objectifs d'un SGBD - 1

- Garantir indépendance données-programme
  - **Indépendance physique** : un changement de l'organisation physique des données n'entraîne pas de changement dans les programmes
  - **Indépendant logique** : un changement de l'organisation logique des données (ex. nouvelle rubrique) n'entraîne pas de changement dans les programmes non concernés

# Objectifs d'un SGBD - 2

- **Manipulation aisée des données** : par des utilisateurs non informaticiens (interrogation et mise à jour)
- **Administration aisée des données** : un SGBD doit fournir des outils pour décrire les données et autoriser leur évolution (tâche de l'administrateur)

# Objectifs d'un SGBD - 3

- **Partage des données** : utilisation simultanée par différentes applications
- **Efficacité des accès aux données** : garantie d'un bon *débit* (nombre de transactions exécutées par seconde) et d'un bon *temps de réponse* (temps d'attente moyen par transaction)



# Fonctionnalités d'un SGBD

Base de Données 1

#0 - Intro

# Fonctionnalités d'un SGBD

- **Définition des données**
- **Manipulation des données**
- **Intégrité des données**
- Partage des données
- Confidentialité des données
- Sécurité de fonctionnement
- Persistance des données et performances

# Définition de données

- Un **Langage de Définition de Données** (LDD) permet de décrire :
  - des **objets** (*personnes, véhicules*)
  - des **attributs** d'objets (nom, *n° d'immat*)
  - des **liens** entre objets (*personne possède véhicule*)
  - des **contraintes** sur objets, attributs et liens (*un véhicule n'a qu'un seul propriétaire*)
- **Schéma** = description d'une BD à l'aide du LDD

# Manipulation de données

- Un **Langage de Manipulation de Données** (LMD) permet de *créer, rechercher, supprimer et modifier des données*
- Offre plusieurs types d'interfaces
- **Interfaces orientées utilisateur final** : langages de requêtes déclaratifs comme SQL avec mise en oeuvre graphique, interface de type formulaire...
- **Interfaces orientées développeurs d'applications** : interface avec des langages de programmation classiques (PHP, Java...)

# Intégrité des données - 1

- Un schéma BD se compose d'une **description des données**, de leurs **relations**, ainsi que d'un ensemble de **contraintes d'intégrité** (CI)
- Une CI est une **propriété de l'application à modéliser**
- Le SGBD doit assurer que les données stockées dans une BD vérifient ces CI

# Intégrité des données - 2

- 2 types de contraintes :
  - Contraintes **structurelles** :
    - *une personne a bien un prénom*
    - *le nom d'utilisateur est unique*
    - *un employé a un et un seul chef*
  - Contraintes **dynamiques** :
    - *un salaire ne peut pas diminuer*

# Partage de données

- Plusieurs applications peuvent accéder à la même information en même temps
- Mécanismes de gestion de conflits d'accès
  - **autorisation** d'accès multiples en consultation (lecture de la BD)
  - **verrouillage** en cas d'accès en modification (écriture dans la BD)

# Confidentialité des données

- Cache certaines informations à certaines applications
- Mise en oeuvre :
  - comptes utilisateurs, **mots de passe**, privilèges d'accès
  - utilisation de **sous-schémas**



# Sécurité de fonctionnement

## - 1

- Mécanisme de **transaction**
  - Séquence d'opérations faisant passer une BD **d'un état cohérent à un nouvel état cohérent**
  - Une transaction est **exécutée** complètement **avec succès**, ou est **annulée**
- **Exemple** : Virement d'un compte bancaire à un autre
  - Si erreur, ne doit pas me retrouver avec juste un compte crédité ou juste un compte débité
  - Soit le virement a été effectué, soit il a été annulé

# Sécurité de fonctionnement

## - 2

- Remettre rapidement une BD dans un état opérationnel après un incident hardware ou software
- **Journalisation** des opérations réalisées sur la BD  
(*dans le journal ou log*)
- **Ré-exécution** automatique en cas d'incident

# Persistance des données et performances

- **Stockage** des données **sur disques** pour la durée de vie
- Mais **disques durs lents**
- Pour garantir un bon **débit** et **temps de réponse**
  - Gestion de **cache** pour un accès efficace (*index, hachage*)
  - Techniques d'**optimisation de requêtes** pour minimiser les transferts

# Résumé

- Besoins de stocker une **large collection de données**
  - Système de **fichiers non-adapté**
  - Utilisation d'un **SGBD recommandé**
- Un SGBD nous permet de :
  - **Définir le schéma**
  - **D'interroger et de manipuler** efficacement
  - Tout en garantissant sa **sécurité et robustesse**

# Types de SGBD

- Il existe différents types de SGBD : *relationnels, orienté documents, orientés graphes, orienté séries chronologiques...*
- Il n'y a **pas de type “ultime”** de SGBD...
- ... ils sont justes conçus pour des **applications différentes**
- Le **choix** du type de SGBD **doit dépendre** :
  - Schéma des données
  - Utilisation des données

# SGBD relationnels - 1

- Se focalise sur les SGBD relationnels dans le cadre de ce module
- Modèle proposé par E.F. Codd en 1970 : *A Relational Model of Data for Large Shared Databanks*
- Le type le plus utilisé encore de nos jours

# SGBD relationnels - 2

- Fondé sur la notion de **relation** : *existe un lien entre les valeurs des attributs d'un objet*
- À partir d'un numéro de sécu d'une personne, peut retrouver nom, prénom, date de naissance...
- À partir d'un titre de film, peut retrouver réalisateur-rice, date de sortie, durée...

# SGBD relationnels - 3

- Existe de nombreux SGBD relationnels
  - Oracle
  - MySQL
  - PostgreSQL
  - MariaDB
  - ...



# Bases de Données 1

## #1 - Conception d'une BD

Matthieu Nicolas

Polytech S5 - EMME

Slides réalisées à partir de celles d'Abir Ismaili-Alaoui, Claude Godart et  
Malika Smaïl

# Plan

- Processus de conception
- Diagramme de classes
- Réalisation du schéma conceptuel

# Processus de conception

Base de Données 1  
#1 - Conception d'une BD

# Conception d'une BD

- *Une description vague de l'application au début...*
- *... un système complet et fonctionnel à l'arrivée*
- Notre rôle :
  - **Comprendre et modéliser** la réalité
  - **Organiser** les informations obtenues
  - **Prendre en compte et ordonner** les besoins
  - **Créer** un schéma de BD adapté

# Exemple de vente par correspondance

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les clients peuvent effectuer des commandes
- Les commandes sont caractérisées par une date, le produit concerné (*pour simplifier*) et la quantité demandée
- Les produits sont décrits par un numéro, un libellé et un prix unitaire
- Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- Les fournisseurs sont décrits par un numéro et leur raison sociale

# Niveaux d'abstraction

- Utilise 3 niveaux d'abstraction de représentation des données pour mettre en place notre BD
  - Niveau **conceptuel**
  - Niveau **logique**
  - Niveau **physique**

# Niveau conceptuel

- Démarre d'une description abstraite et globale du monde réel
  - Aspect statique (*données à stocker*)
  - Aspect dynamique (*traitements sur les données à effectuer*)
- Aboutit au **schéma conceptuel**
- Le schéma conceptuel décrit la structure de la base de données **indépendamment de son implémentation; il doit être compréhensible par des non techniciens**

# Niveau logique

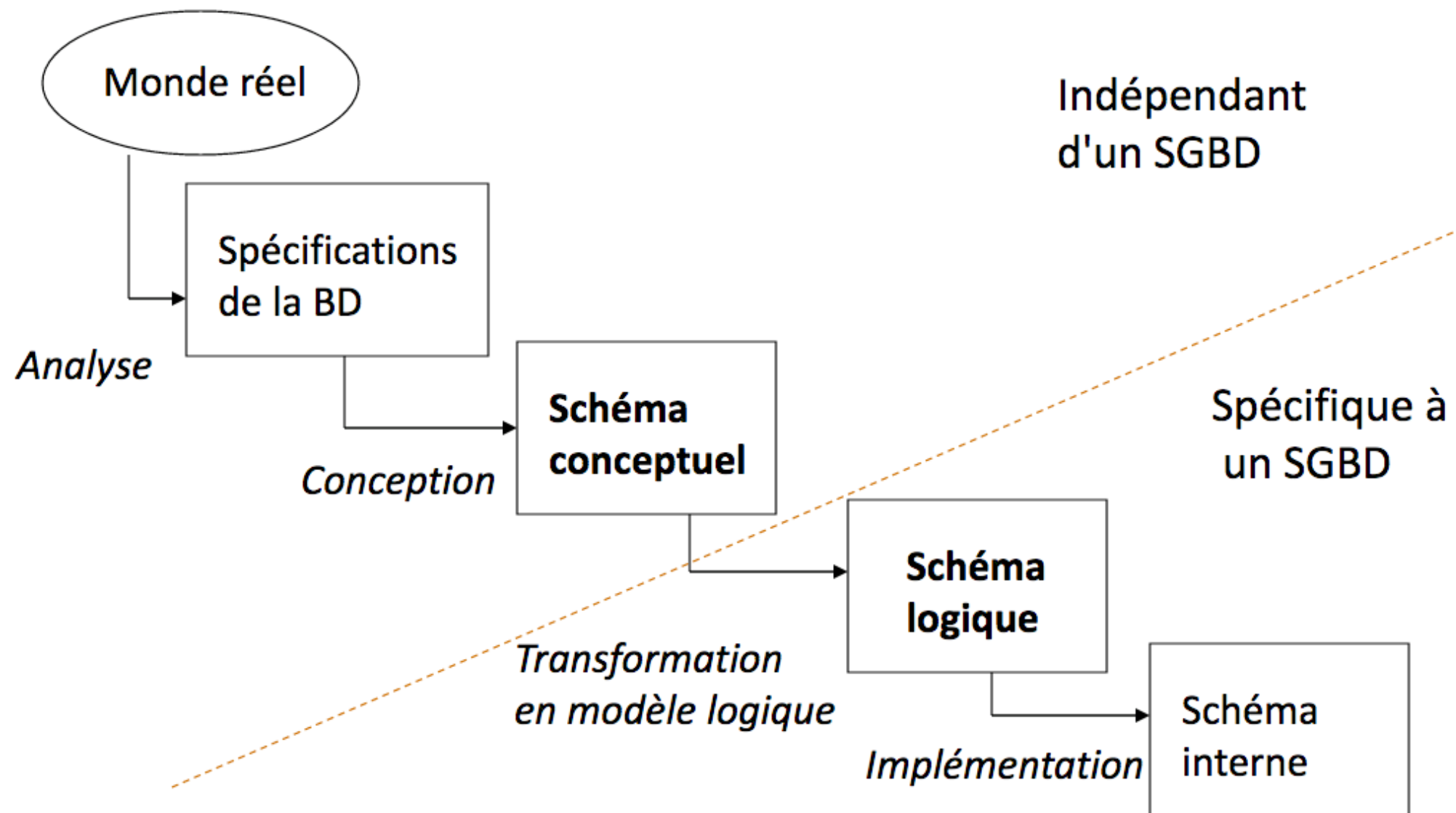
- Spécifie comment le schéma conceptuel se traduit avec les **structures de données du SGBD**
  - Les structures et types utilisés dépendent du type de SGBD (*relations, documents, graphes*)
  - Permet d'évaluer les coûts de mise en place
  - La **normalisation** de la BD se fait à cette étape
- Aboutit au **schéma logique**
- Généralement, les **développeurs** d'applications et **administrateurs** de la DB travaillent à ce niveau



# Niveau physique

- Spécifie comment le schéma logique est implémenté dans un **environnement technique spécifique**
  - Prise en compte de contraintes liées au matériel et logiciel
  - Choix de l'organisation physique des données, des structures des indexes...
- Aboutit au **schéma physique**
- Généralement, géré par les administrateurs de la BD

# Processus de conception



# Quelques mots sur la conception de BD - 1

- Processus de conception : **difficile** à formaliser
  - Nécessite **pratique**, **expérience** et *flair*
  - Pas de solution parfaite...
  - ... uniquement des choix

# Quelques mots sur la conception de BD - 2

- Choix portent sur deux aspects interdépendants du SI :
  - Partie **statique** : objets structurés et leurs relations (*données*)
  - Partie **dynamique** : activité du système (*traitements opérant sur les données*)
- Doit proposer un schéma offrant le bon équilibre entre ces deux aspects
- Le schéma peut évoluer au fil du temps si besoin

# Choix du modèle pour schéma conceptuel

- Existe plusieurs modèles pour représenter le schéma conceptuel d'une BD
  - **Diagrammes de classes** *Unified Modeling Language (UML)*
  - Modèle *Entité-Association (EA)*
  - *Object-Role Modeling (ORM)*
- Chacun dispose de son **formalisme** et de sa **représentation visuelle**

# Diagramme de classes

Base de Données 1  
#1 - Conception d'une BD

# Unified Modeling Language (UML) - 1

- Langage de modélisation **graphique** et **standardisé**
- Permet de représenter de manière graphique une application
- Outil pour **raisonner sur la conception orientée objet** notamment
- Indépendant des langages orientés objets

# Unified Modeling Language (UML) - 2

- Propose 14 types de diagrammes
  - **Diagrammes de comportement** : *diagramme des cas d'utilisation*
  - **Diagrammes de structure** : *diagramme de classes*
  - **Diagrammes d'interaction** : *diagramme de séquence*



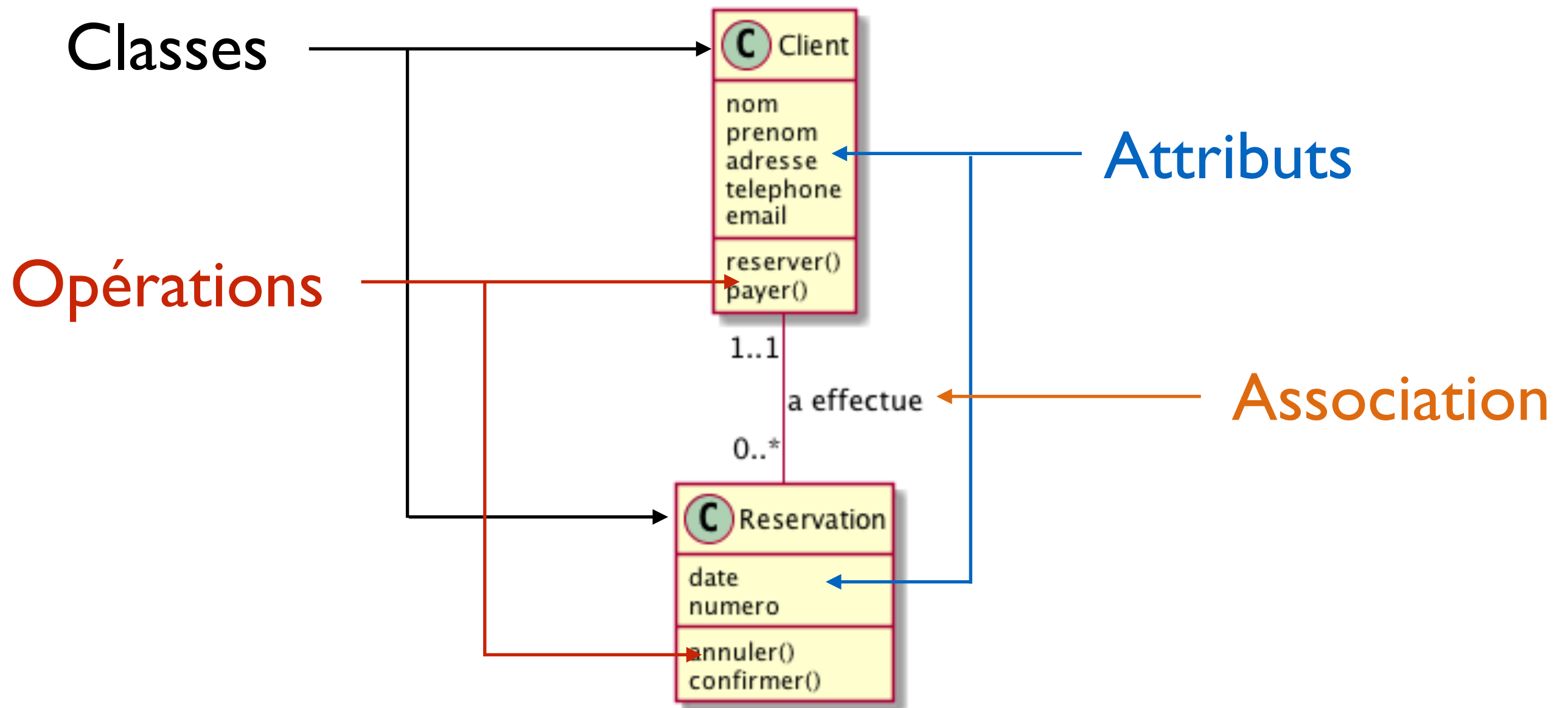
# Diagramme de classes

- Représente la **structure** d'une application en décrivant
  - ses **classes** (*entités*)
  - leurs **associations** (*relations*)

# Notion de classe

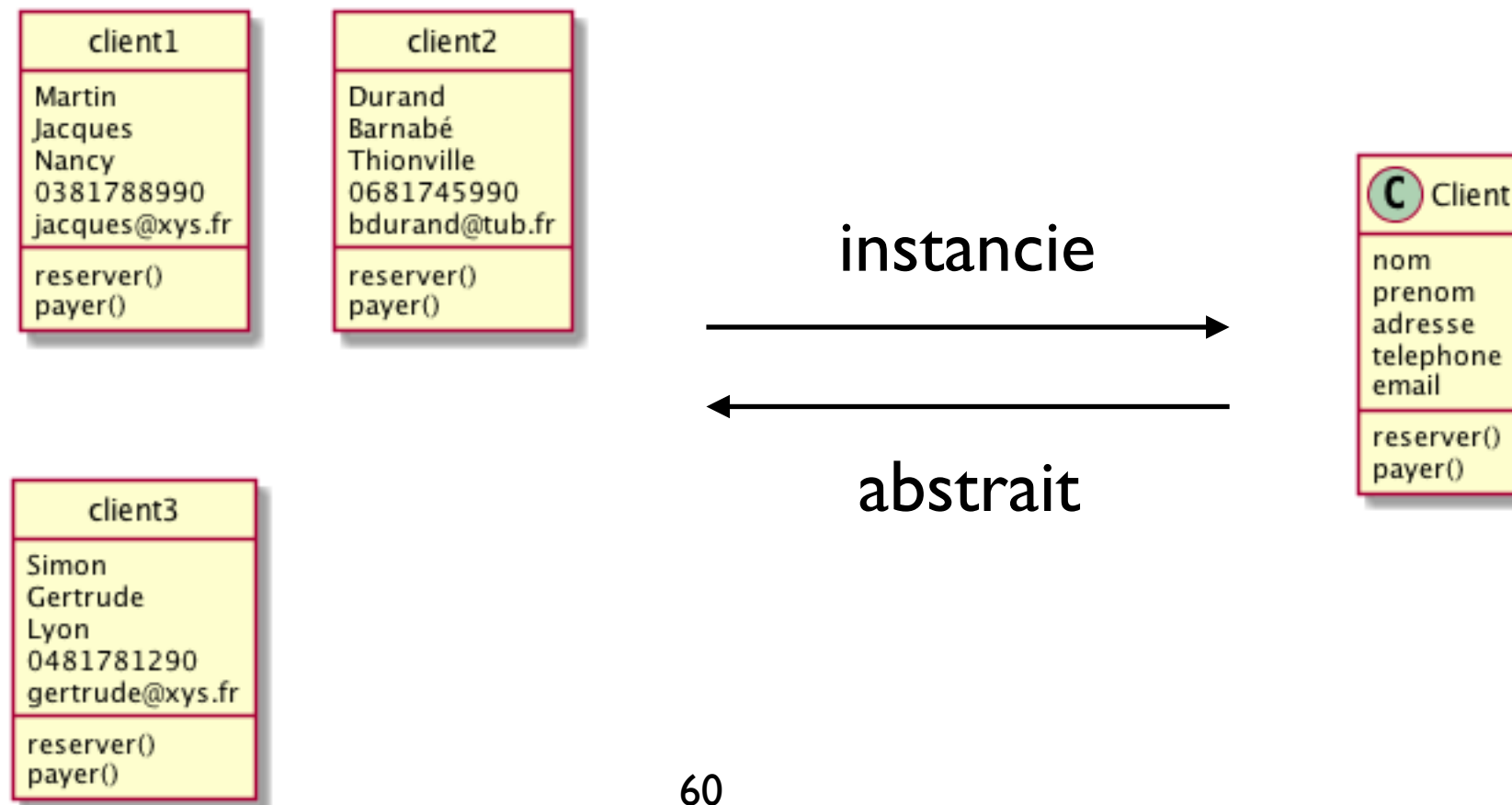
- Ensemble d'objets qui ont les mêmes propriétés
- Les propriétés d'un objet sont
  - ses **attributs** : valeurs qui caractérisent son état
  - ses **opérations** : qui caractérisent son savoir-faire, ses responsabilités, les services qu'il peut rendre
    - Nous intéresse pas dans ce module
  - ses **associations** (liens) avec les autres objets avec lesquels il interagit

# Exemple diagramme de classes



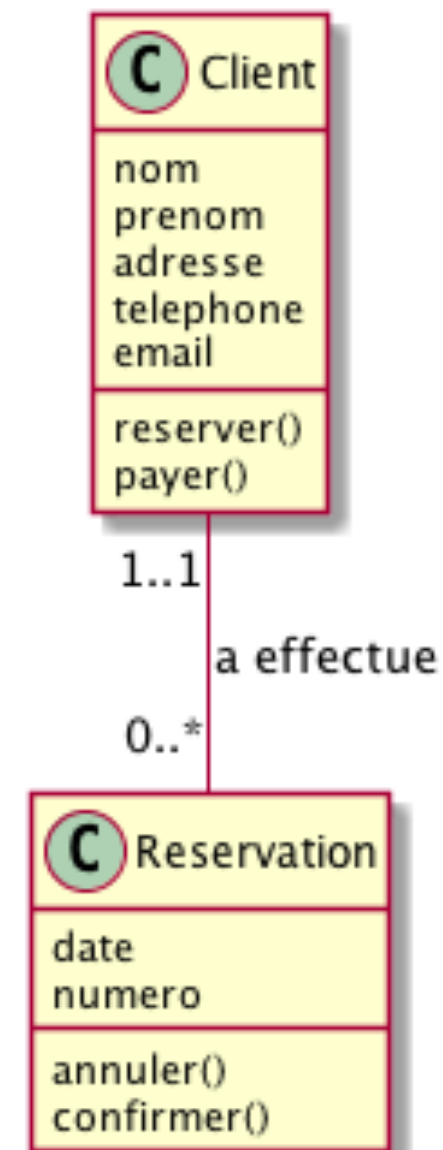
# Distinction objets/classes

- **Objet** : entité identifiable, définie par son nom et ses propriétés
- **Classe** : regroupement d'objets de même nature (même propriétés)



# Associations entre classes

- Les associations représentent les liens, les interactions entre objets
- Les associations sont caractérisées par leur **nom** et leurs **cardinalités**



# Cardinalités d'une association - 1

- Nombre d'objets de la classe B associés à la classe A



Exactement 1 (par défaut)



0 ou plusieurs



Au moins 1



Optionalité (0 ou 1)

# Cardinalités d'une association - 2

- Nombre d'objets de la classe B associés à la classe A



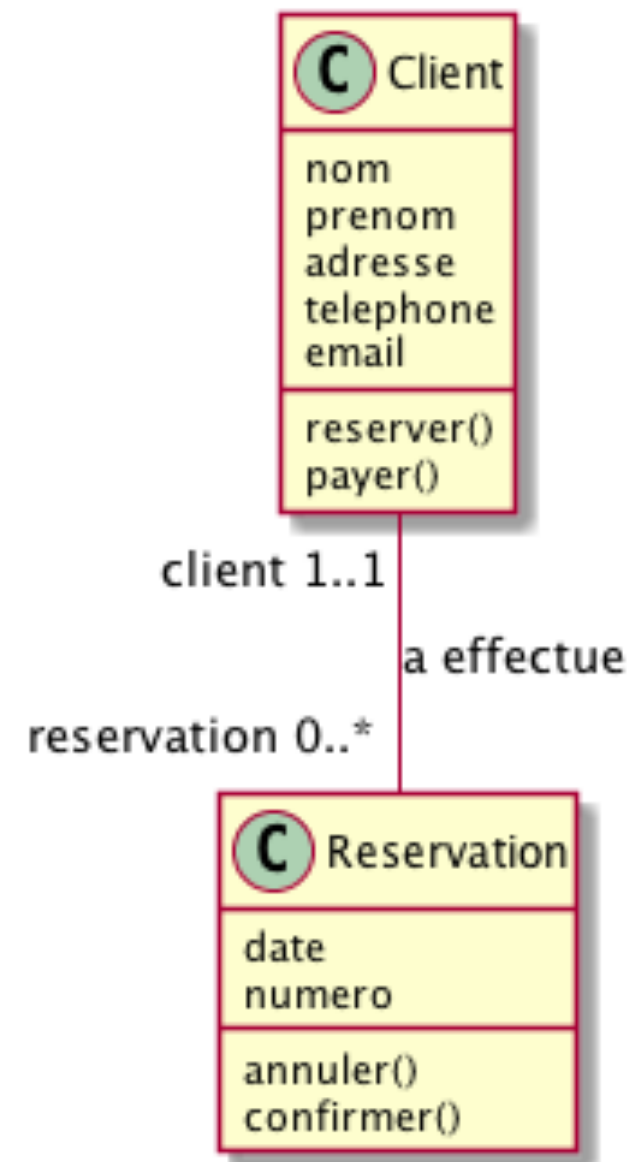
Exactement n



Entre n et m

# Liens entre classes/rôles

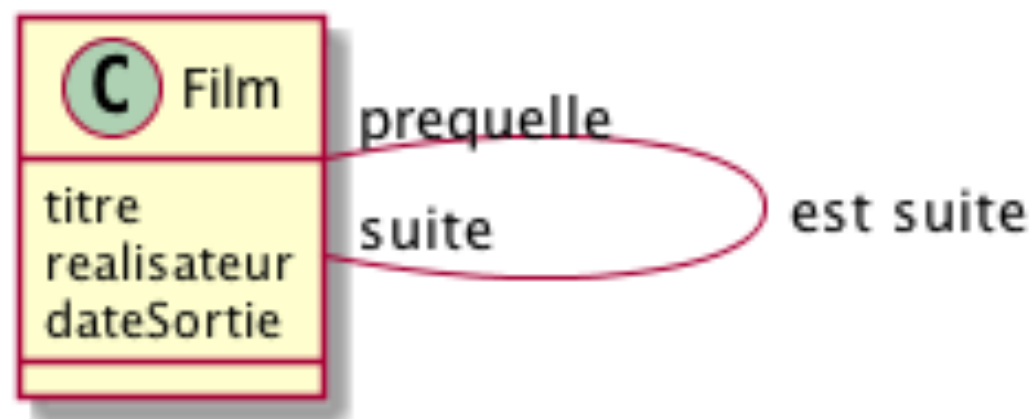
- Associations représentent les liens entre classes
- Une association se lit dans les 2 sens
- Chaque classe joue un **rôle** dans une association
- Parfois **superflu** de le préciser





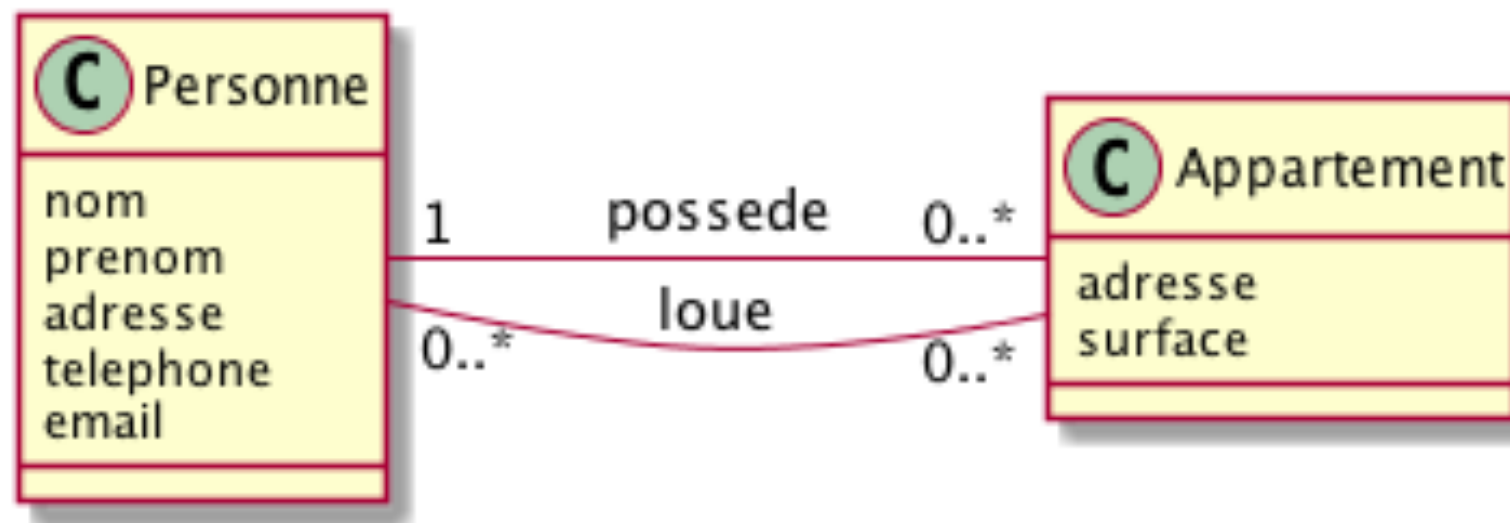
# Association réflexive

- Possible d'avoir une **association** d'une **classe** **vers elle-même**
- Les **rôles** deviennent **nécessaires** dans ce cas



# Associations multiples entre classes

- On peut avoir **plusieurs** associations entre 2 classes



# Faisons un 1er point - 1

- Nous avons vu le résultat demandé
  - *Diagramme de classes* décrivant le **schéma conceptuel** de la BD
- Sa structure doit permettre de
  - Stocker les données demandées
  - D'effectuer les traitements demandés

# Faisons un 1er point - 2

- Il nous reste à voir **comment l'obtenir**
  - **Qu'est-ce** qu'on a **en entrée** ?
  - **Comment** on **procède** ?

# Qu'est-ce qu'on a en entrée ?

- Peut avoir une **spécification du système d'informations** à modéliser
  - **Description**  $\pm$  formelle **de la structure** du système, des données utilisées, de leurs relations
- Plus généralement, **description des fonctionnalités** du système

# Exemple de vente par correspondance

- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les clients peuvent effectuer des commandes
- Les commandes sont caractérisées par une date, le produit concerné (*pour simplifier*) et la quantité demandée
- Les produits sont décrits par un numéro, un libellé et un prix unitaire
- Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- Les fournisseurs sont décrits par un numéro et leur raison sociale

# Comment procéder ?

- Identifier les **entités** qui composent le système
- Identifier les **associations** entre entités et leurs cardinalités
- Identifier les **attributs** de chaque entité

# Exemple de vente par correspondance - entités

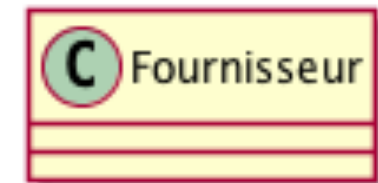
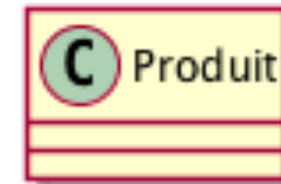
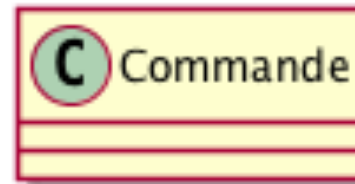
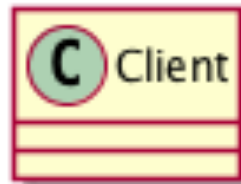
- Les clients sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les clients peuvent effectuer des commandes
- Les commandes sont caractérisées par un numéro, une date, le produit concerné (*pour simplifier*) et la quantité demandée
- Les produits sont décrits par un numéro, un libellé et un prix unitaire
- Chaque produit est fourni par un fournisseur unique
- Un fournisseur peut fournir plusieurs produits
- Les fournisseurs sont décrits par un numéro et leur raison sociale



# Exemple de vente par correspondance - entités

- Les **clients** sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les **clients** peuvent effectuer des **commandes**
- Les **commandes** sont caractérisées par un numéro, une date, le **produit** concerné (*pour simplifier*) et la quantité demandée
- Les **produits** sont décrits par un numéro, un libellé et un prix unitaire
- Chaque **produit** est fourni par un **fournisseur** unique
- Un **fournisseur** peut fournir plusieurs **produits**
- Les **fournisseurs** sont décrits par un numéro et leur raison sociale

# Schéma conceptuel - entités



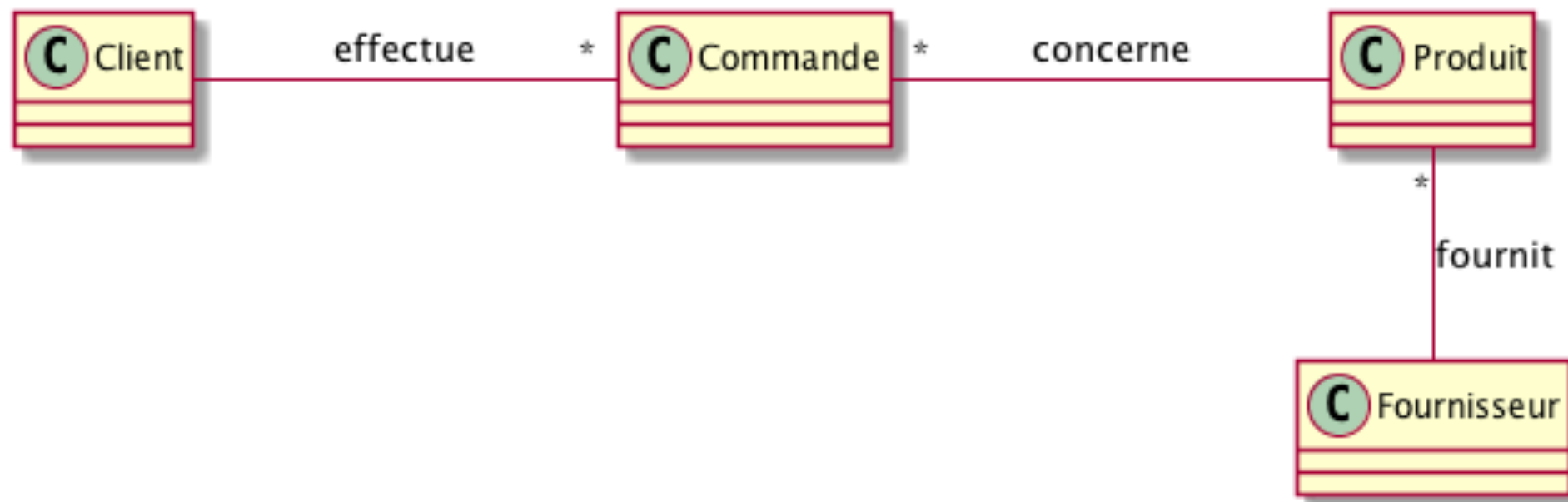
# Exemple de vente par correspondance - associations

- Les **clients** sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les **clients** peuvent effectuer des **commandes**
- Les **commandes** sont caractérisées par un numéro, une date, le **produit** concerné (*pour simplifier*) et la quantité demandée
- Les **produits** sont décrits par un numéro, un libellé et un prix unitaire
- Chaque **produit** est fourni par un **fournisseur** unique
- Un **fournisseur** peut fournir plusieurs **produits**
- Les **fournisseurs** sont décrits par un numéro et leur raison sociale

# Exemple de vente par correspondance - associations

- Les **clients** sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les **clients** **peuvent effectuer des commandes**
- Les **commandes** sont caractérisées par un numéro, une date, **le produit concerné** (*pour simplifier*) et la quantité demandée
- Les **produits** sont décrits par un numéro, un libellé et un prix unitaire
- Chaque **produit est fourni** par **un fournisseur unique**
- Un **fournisseur peut fournir plusieurs produits**
- Les **fournisseurs** sont décrits par un numéro et leur raison sociale

# Schéma conceptuel - associations



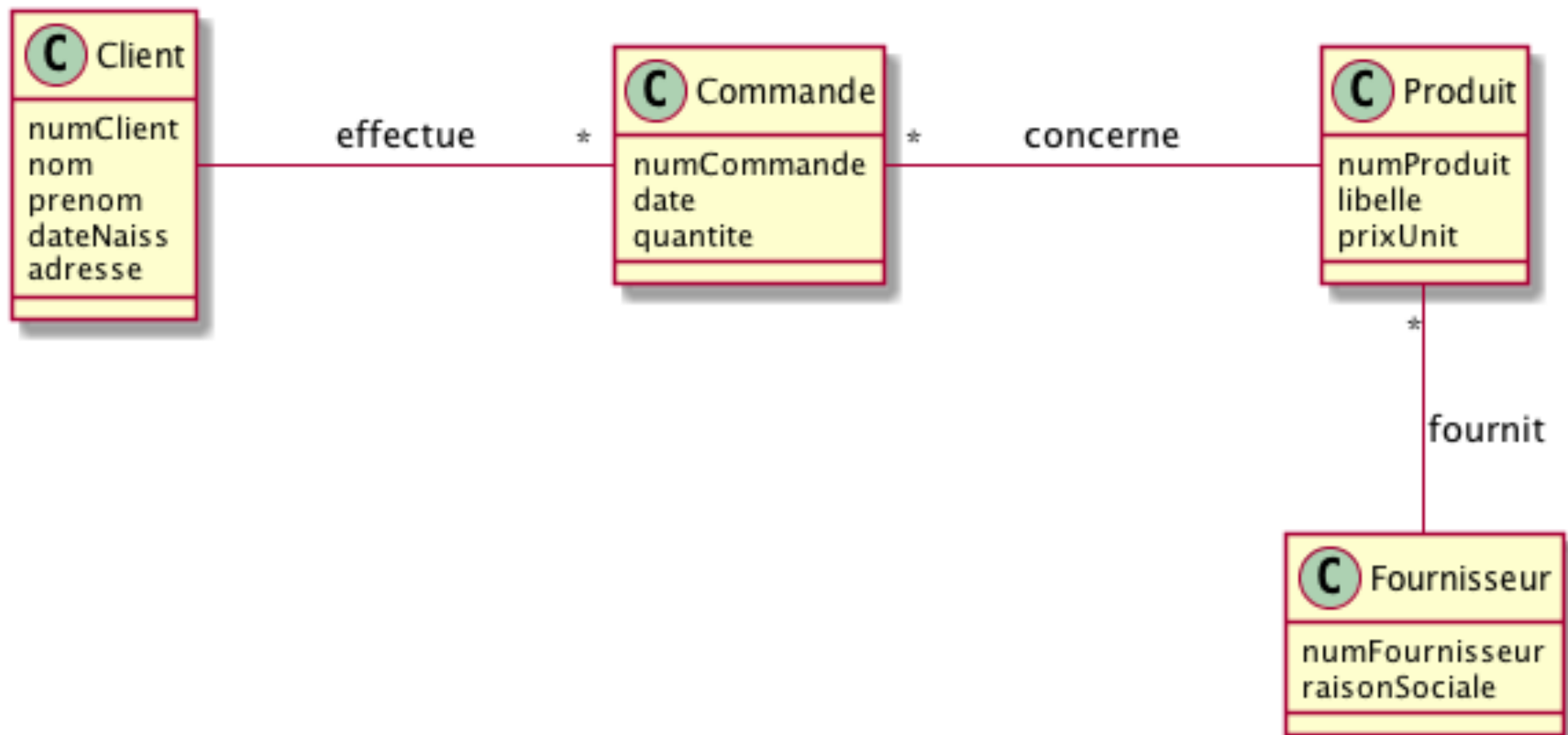
# Exemple de vente par correspondance - attributs

- Les **clients** sont caractérisés par un numéro de client, nom, prénom, date de naissance, adresse
- Les **clients** **peuvent effectuer des commandes**
- Les **commandes** sont caractérisées par un numéro, une date, **le produit concerné** (*pour simplifier*) et la quantité demandée
- Les **produits** sont décrits par un numéro, un libellé et un prix unitaire
- Chaque **produit** **est fourni** par **un fournisseur unique**
- Un **fournisseur** **peut fournir plusieurs produits**
- Les **fournisseurs** sont décrits par un numéro et leur raison sociale

# Exemple de vente par correspondance - attributs

- Les **clients** sont caractérisés par un **numéro de client**, **nom**, **prénom**, **date de naissance**, **adresse**
- Les **clients** **peuvent effectuer des** commandes
- Les **commandes** sont caractérisées par un numéro, une **date**, **le produit concerné** (*pour simplifier*) et la **quantité** demandée
- Les **produits** sont décrits par un **numéro**, un **libellé** et un **prix unitaire**
- Chaque **produit** **est fourni** par **un fournisseur unique**
- Un **fournisseur** **peut fournir plusieurs** produits
- Les **fournisseurs** sont décrits par un **numéro** et leur **raison sociale**

# Schéma conceptuel - attributs





# Second exemple - gestion de cours

- L'école souhaite gérer les cours qu'elle propose
- L'école embauche plusieurs enseignant-es
- Chaque enseignant-e est identifié-e par son nom, prénom, adresse mail et date effective de prise de poste
- Chaque enseignant-e donne un seul cours
- Les étudiant-es, eux, suivent plusieurs cours
- Un-e étudiant-e est identifié-e par son nom, prénom, mail, ainsi que son année d'entrée à l'école
- Pour chaque cours, on note son intitulé, sa description et le public visé

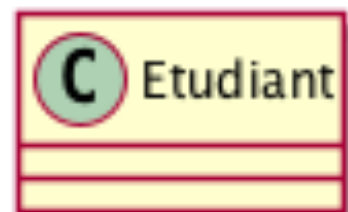
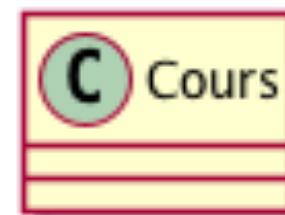
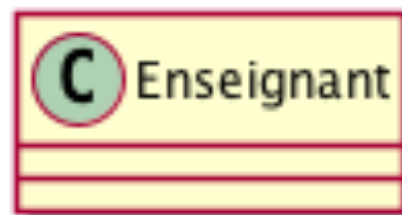
# Second exemple - gestion de cours - entités

- L'école souhaite gérer les cours qu'elle propose
- L'école embauche plusieurs enseignant-es
- Chaque enseignant-e est identifié-e par son nom, prénom, adresse mail et date effective de prise de poste
- Chaque enseignant-e donne un seul cours
- Les étudiant-es, eux, suivent plusieurs cours
- Un-e étudiant-e est identifié-e par son nom, prénom, mail, ainsi que son année d'entrée à l'école
- Pour chaque cours, on note son intitulé, sa description et le public visé

# Second exemple - gestion de cours - entités

- L'école souhaite gérer les **cours** qu'elle propose
- L'école embauche plusieurs **enseignant-es**
- Chaque **enseignant-e** est identifié-e par son nom, prénom, adresse mail et date effective de prise de poste
- Chaque **enseignant-e** donne un seul **cours**
- Les **étudiant-es**, eux, suivent plusieurs **cours**
- Un-e **étudiant-e** est identifié-e par son nom, prénom, mail, ainsi que son année d'entrée à l'école
- Pour chaque **cours**, on note son intitulé, sa description et le public visé

# Second exemple - gestion de cours - entités



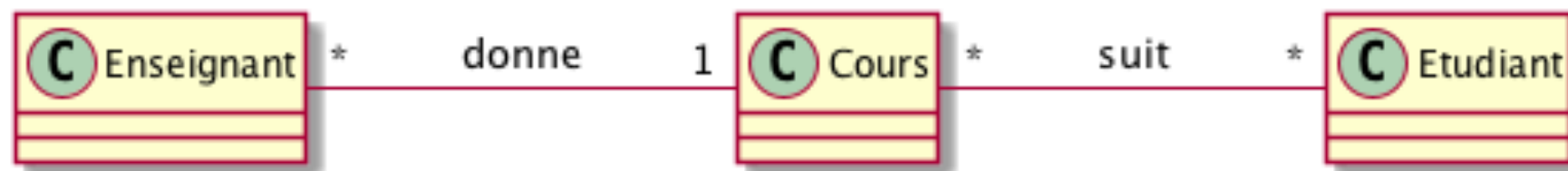
# Second exemple - gestion de cours - associations

- L'école souhaite gérer les **cours** qu'elle propose
- L'école embauche plusieurs **enseignant-es**
- Chaque **enseignant-e** est identifié-e par son nom, prénom, adresse mail et date effective de prise de poste
- Chaque **enseignant-e** donne un seul **cours**
- Les **étudiant-es**, eux, suivent plusieurs **cours**
- Un-e **étudiant-e** est identifié-e par son nom, prénom, mail, ainsi que son année d'entrée à l'école
- Pour chaque **cours**, on note son intitulé, sa description et le public visé

# Second exemple - gestion de cours - associations

- L'école souhaite gérer les **cours** qu'elle propose
- L'école embauche plusieurs **enseignant-es**
- Chaque **enseignant-e** est identifié-e par son nom, prénom, adresse mail et date effective de prise de poste
- Chaque **enseignant-e** **donne un** seul **cours**
- Les **étudiant-es**, eux, **suivent plusieurs** **cours**
- Un-e **étudiant-e** est identifié-e par son nom, prénom, mail, ainsi que son année d'entrée à l'école
- Pour chaque **cours**, on note son intitulé, sa description et le public visé

# Second exemple - gestion de cours - associations



# Second exemple - gestion de cours - attributs

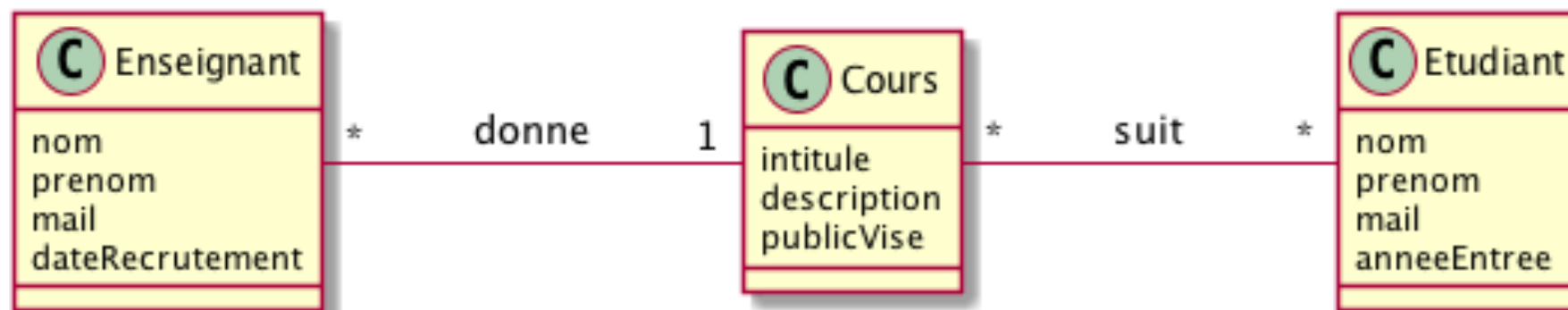
- L'école souhaite gérer les **cours** qu'elle propose
- L'école embauche plusieurs **enseignant-es**
- Chaque **enseignant-e** est identifié-e par son nom, prénom, adresse mail et date effective de prise de poste
- Chaque **enseignant-e** **donne un** seul **cours**
- Les **étudiant-es**, eux, **suivent plusieurs** **cours**
- Un-e **étudiant-e** est identifié-e par son nom, prénom, mail, ainsi que son année d'entrée à l'école
- Pour chaque **cours**, on note son intitulé, sa description et le public visé



# Second exemple - gestion de cours - attributs

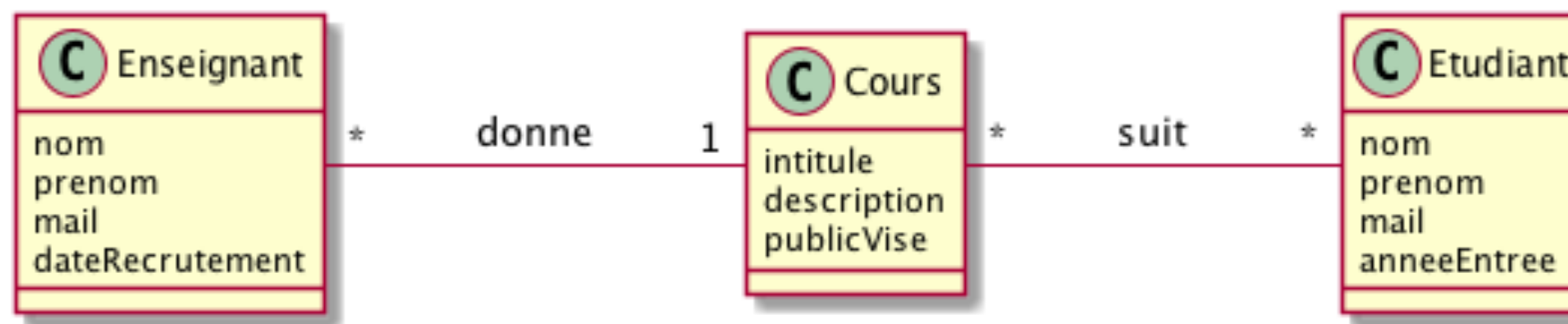
- L'école souhaite gérer les **cours** qu'elle propose
- L'école embauche plusieurs **enseignant-es**
- Chaque **enseignant-e** est identifié-e par son **nom, prénom, adresse mail et date effective de prise de poste**
- Chaque **enseignant-e** **donne un** seul **cours**
- Les **étudiant-es**, eux, **suivent plusieurs** **cours**
- Un-e **étudiant-e** est identifié-e par son **nom, prénom, mail**, ainsi que son **année d'entrée** à l'école
- Pour chaque **cours**, on note son **intitulé, sa description et le public visé**

# Second exemple - gestion de cours - attributs



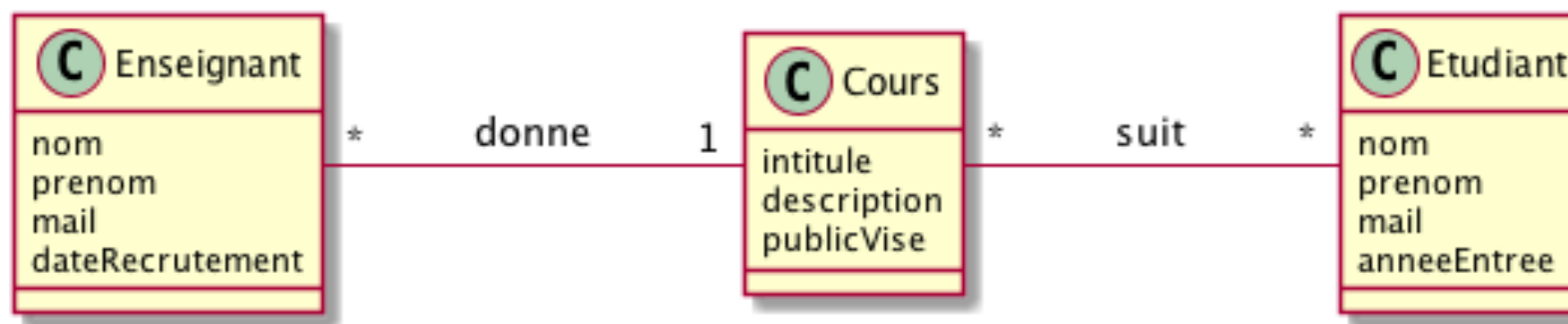
# Second exemple - gestion de cours - suite

- L'école veut maintenant conserver la note obtenue par chaque étudiant-e dans chaque cours suivi (calcul moyenne par cours/par étudiant-e...)
- Comment ajouter cette information au schéma conceptuel ?



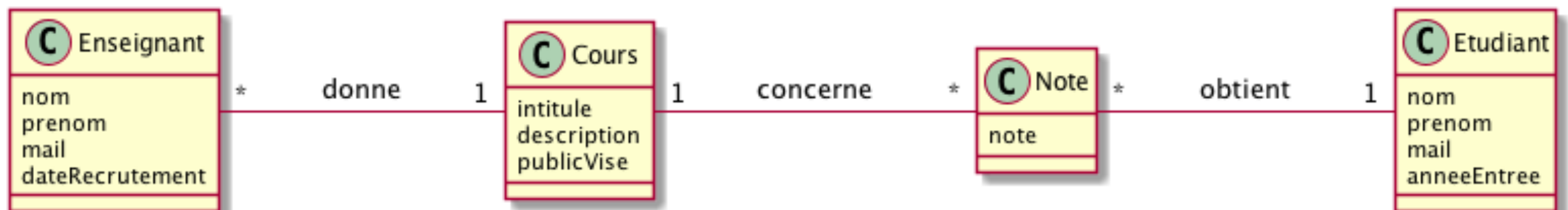
# Second exemple - gestion de cours - suite bis

- La note dépend du couple (**Etudiant**, **Cours**), et non pas de seulement **Etudiant** ou seulement **Cours**
- Il s'agit d'un attribut de l'association **suit**
- On parle alors de **classe d'association**



# Second exemple - gestion de cours - suite ter

- Pour la représenter, fait évoluer l'association suivre en une classe **Note**
- Et ré-associe cette classe avec **Etudiant** et **Cours**



# Résumé

- Concevoir une BD se fait en plusieurs étapes
- La 1ère étape consiste à **réaliser un schéma conceptuel**
  - Plusieurs outils disponibles pour le représenter
  - Utilisons **diagrammes de classes** dans ce module
- Décrit les **entités**, leurs **attributs** et les **associations** entre entités

# Arche

- Supports de cours et sujets de TD à disposition
- Infos du Module
  - *2021 - Polytech EMME : Base de Données 1*
  - Identifiant : 50014
  - Clef d'inscription : 2021-EMME-BD1
- Pour ceux n'ayant pas encore accès à Arche
  - Repo Github dispo : <http://cpc.cx/vNe>