

Généralités

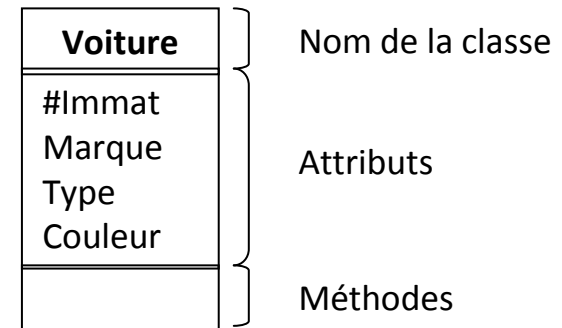
- *Unified Modeling Language* : standard élaboré à l'initiative de l'*Object Management Group*
- Spécifiquement destiné à la modélisation objet
- Langage indépendant de la plateforme utilisée, ainsi que des langages de plus bas niveau orientés objet, tels C++ et Java
- Utilise une notation graphique
- UML a été adapté pour la modélisation de BDD, via un profil spécifique : *UML profile for data modeling*
- UML permet donc de représenter un modèle conceptuel utilisable pour créer des BDD relationnelles
- Il est donc possible d'utiliser des outils du marché pour la description et l'automatisation du processus de passage au SGBD, jusqu'à la génération automatique de code SQL
- UML est fortement répandu, ce qui permet donc d'utiliser le même formalisme et les mêmes outils que les concepteurs d'applications logicielles

Représentation du modèle conceptuel (1)

- UML dispose de 9 types de diagrammes répartis en 2 catégories :
 - diagrammes de structure : partie statique du modèle
 - diagrammes de comportement : partie dynamique, *i.e.* traitements
- Pour les BDD, on utilise les **diagrammes de classes**, qui font partie des diagrammes de comportement

Classes

- Les entités sont Les entités sont interprétées comme des objets et sont représentées comme des **classes**
- En UML, une classe est définie par :
 - un nom
 - des attributs
 - des méthodes
- En UML, une classe est représentée graphiquement par un rectangle séparé en 3 zones



Représentation du modèle conceptuel (2)

Association

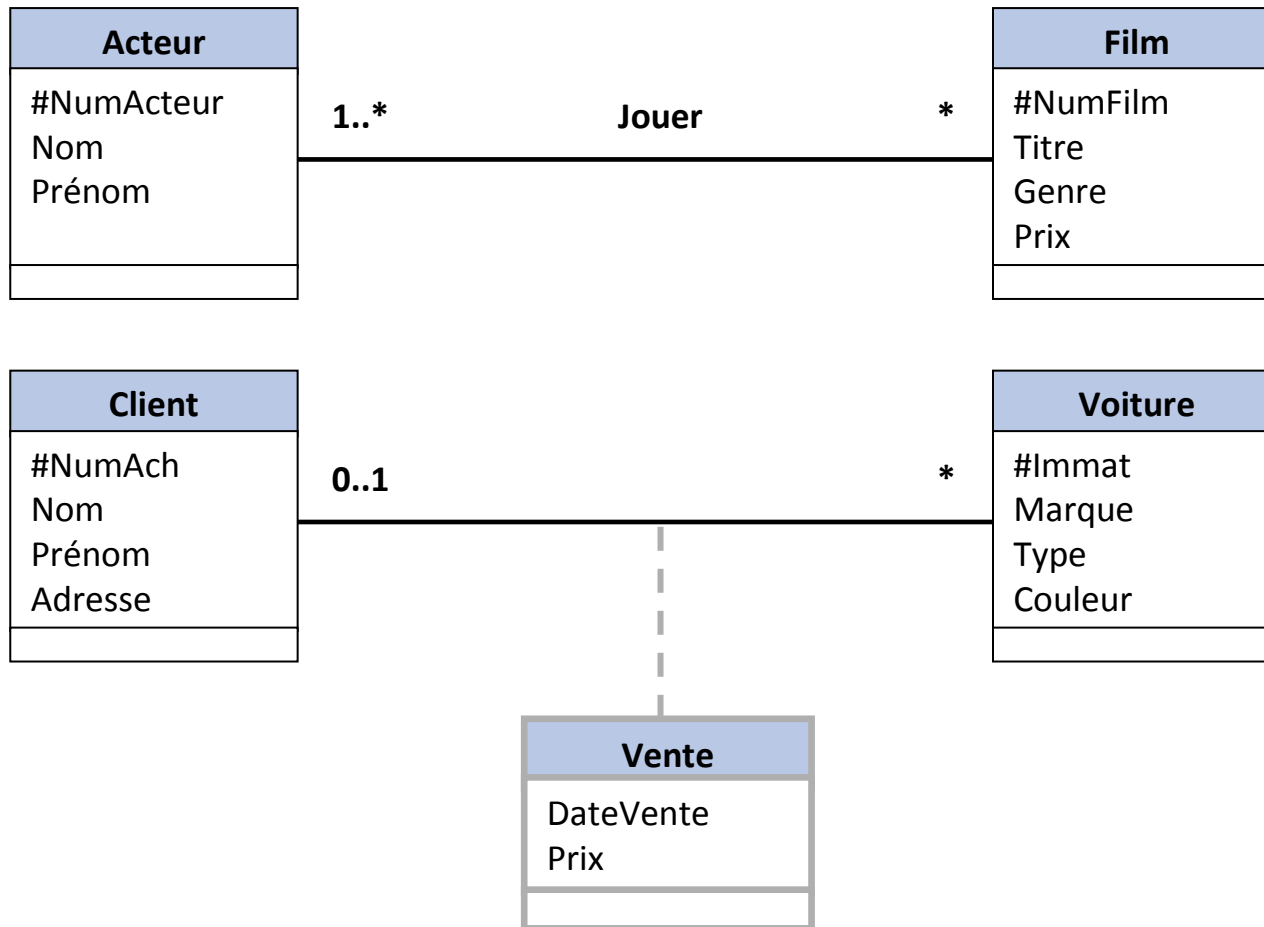
- Les entités sont reliées entre elles par des associations
- Les associations sont caractérisées par
 - leur **dimension**, *i.e.* le nombre de classes liées
 - leur **nom**
 - leur **multiplicité**, qui caractérise le nombre mini et maxi d'objets liés
- L'expression de la multiplicité est quasiment la même que la cardinalité dans le modèle entité-association

Minimum, Maximum	Optionnel	Obligatoire
Unicité	0..1	1..1 ou 1
Multiplicité	0..* ou *	1..*

- Les associations sont matérialisées par un trait entre les classes, le nom de l'association étant inscrit au dessus de trait
- Si l'association possède un attribut, on crée une classe d'association portant le nom de la classe, et on y assigne les attributs de la classe. Elle est reliée par un trait pointillé au trait d'association

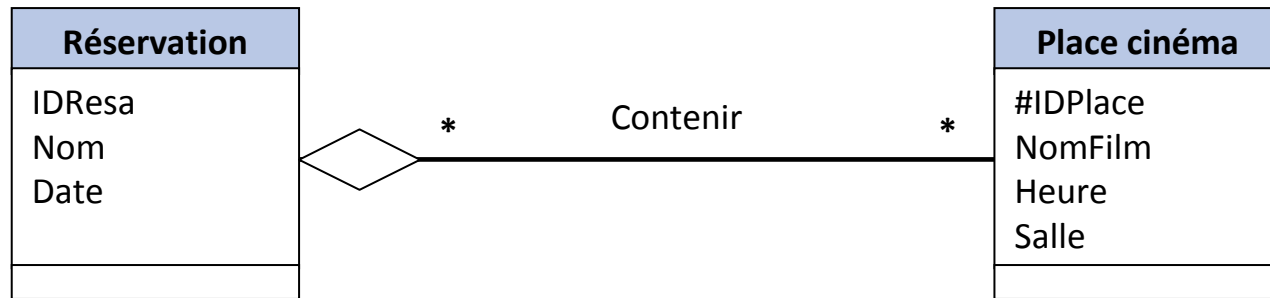
Représentation du modèle conceptuel (3)

Exemples



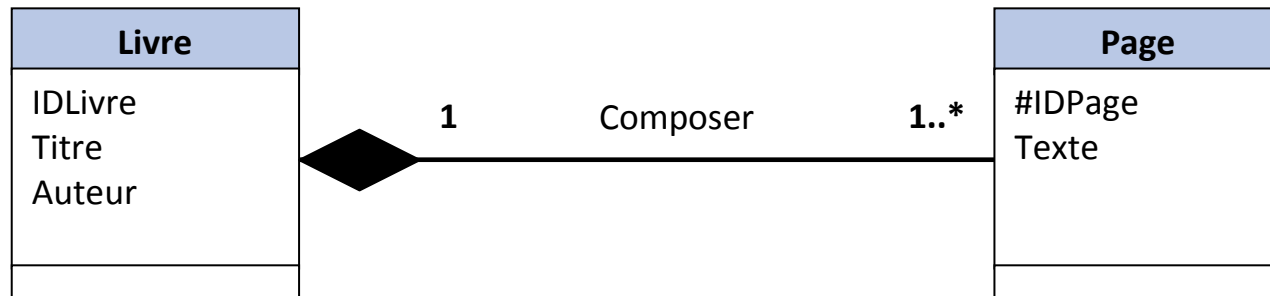
Agrégation

- Une relation d'agrégation indique un principe de subordination entre l'agrégat et les classes agrégées
- Elle indique une possession : l'agrégat peut contenir plusieurs objets d'un type
- En UML, l'agrégation est représentée par un ligne entre 2 classes, terminée par un losange vide du côté de l'agrégat
- **Exemple**



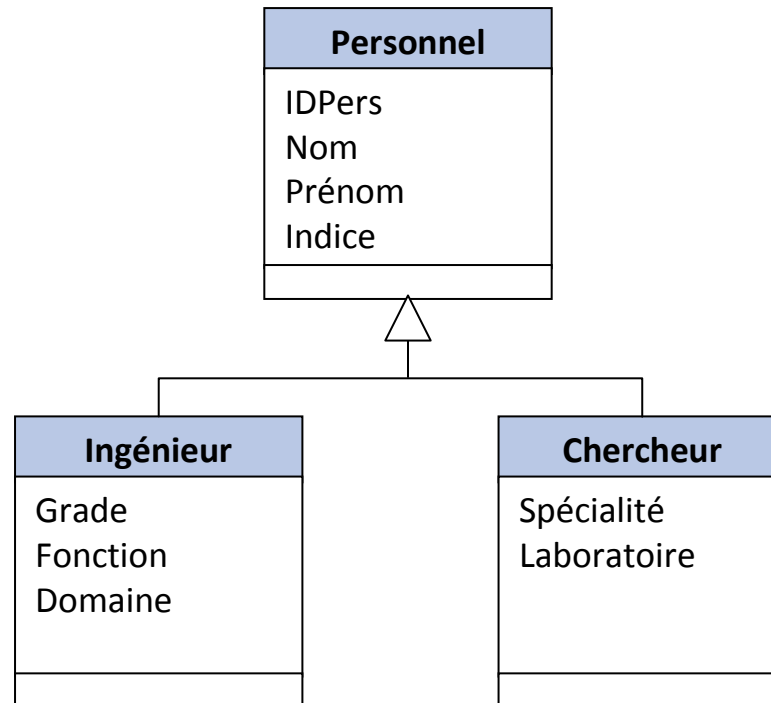
Composition

- Composition = agrégation à laquelle on impose des contraintes :
 - un objet fait partie d'un seul composite
- A un même moment, une instance de composant ne peut être lié qu'à un seul agrégat
- Les cycles de vie des composants et de l'agrégat sont liés : si l'agrégat est détruit, ses composants le sont aussi
- Les compositions sont représentées comme l'agrégation, mais avec un losange plein
- **Exemple**



Bonus : généralisation

- Relation entre un élément général et un élément plus spécifique
- Le terme généralisation signifie « **est un** » ou « **est une sorte de** »
- Elle est représentée au moyen d'une flèche (triangle) qui pointe de la classe la plus spécialisée vers la classe la plus générale
- **Exemple**



Exercice 3 : vente d'ordinateurs (1)

Enoncé général

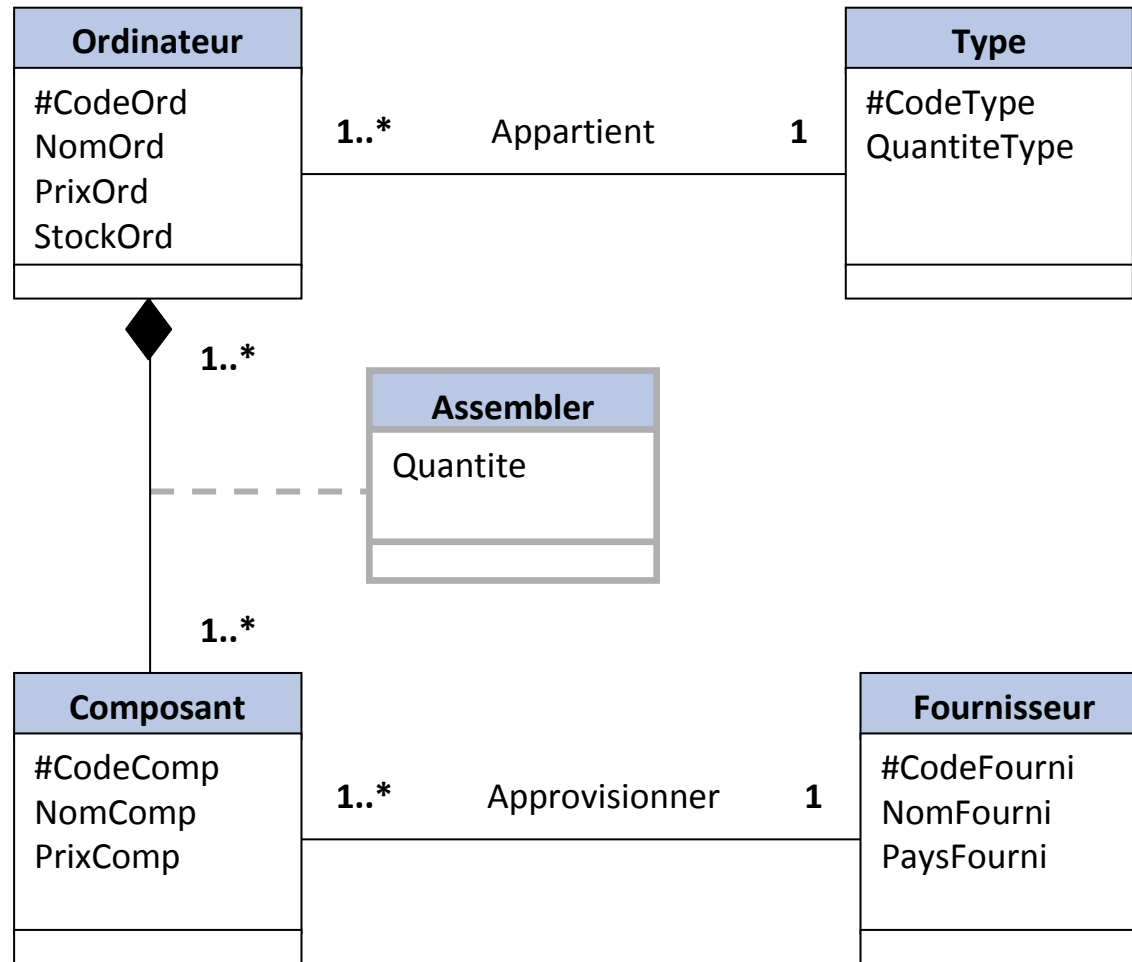
Une société de distribution de matériel informatique vend des ordinateurs assemblés avec des composants standard

Précisions

- Les ordinateurs sont définis par un code, un nom, un prix de vente et une quantité en stock
- Les ordinateurs appartiennent à un type. Pour chaque type, on connaît la quantité
- Chaque composant est caractérisé par un code, un nom, un prix de vente
- On s'approvisionne en composants auprès de fournisseurs définis par un code, un nom et un pays
- Un même composant ne peut être obtenu qu'auprès d'un fournisseur.
- Il peut y avoir plusieurs composants identique dans un ordinateur

→ Construire le diagramme de classes de la BDD

Exercice 3 : vente d'ordinateurs (2)



Du modèle entité-association à UML

- Une classe d'entité est représenté par une **classe**
- Une association est représentée par une **association**
- Une cardinalité est représentée par une **multiplicité**

Intérêts d'UML

- Utilisation d'outils logiciels
- Assistance du processus de passage au modèle relationnel
- Standard très utilisé
- Utilisé dans d'autres domaines, tel le génie logiciel

Avantages et inconvénients

Avantages

- Modèle simple, il n'y a que 3 concepts : entités, associations et attributs
- Approprié à une représentation graphique intuitive
- Permet de modéliser rapidement des structures peu complexes

Inconvénients

- Plusieurs solutions pour un même problème
- Pauvreté :
 - difficile d'exprimer les contraintes d'intégrité
 - difficile de représenter des structures complexes

Définition d'un schéma relationnel (1)

Un peu de vocabulaire

- Dans le modèle relationnel, une base de données est vue comme un ensemble de relations :
$$BD = \{\text{Relations}\} \quad \text{Relation} = \{\text{SR}, \text{ER}\}$$
- Le schéma SR de la relation R se note $R[A_1, A_2, \dots, A_n]$ où R est le **nom** de la relation et A_i représente un **attribut**
- Le nombre d'attributs n est le **degré** de la relation
- A chaque attribut est associé l'ensemble des valeurs plausibles, appelé **domaine de valeurs** : liste de valeurs, intervalle, format
- L'extension ER de la relation est un ensemble de **tuples** (ou **n-uplets**) :
$$ER = \{t_1, t_2, \dots, t_n\} \text{ avec } t_i = \{ \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle \} \text{ et } t_i[A_j] \text{ est la valeur de l'attribut } A_j \text{ dans le tuple } t_i$$
- En représentation tabulaire :

relation \rightarrow table
attribut \rightarrow colonne
tuple \rightarrow ligne

Définition d'un schéma relationnel (2)

Exemple

Schéma relationnel

Bibliothèque			
Cote	Titre	Auteur	Editeur
1	Création de bases de données	N. Larrousse	Pearson
2	Automatique : Cours et exercices corrigés	Y. Granjon	Dunod
3	TD d' informatique	Y. Granjon	Dunod

attribut (pointing to Cote, Titre, Auteur, Editeur)

tuple (pointing to the row with Cote=2)

valeur (pointing to the cell with 'TD d' informatique')

Définition d'un schéma relationnel (3)

Dépendance fonctionnelle

- Etant donnée $R[...., A, ..., B, ...]$, on dit que l'attribut B dépend fonctionnellement de l'attribut A (que l'on note $A \rightarrow B$) si et seulement si pour tout j avec $t_j[A] = a$ on a toujours $t_j[B] = b$
- Les dépendances entre les champs représentent les liens entre les différents éléments du monde réel

Age	Nom	Ville	Etablissement
21	Dupont	Paris	Sorbonne
22	Pierre	Nancy	Henri Poincaré
27	Dupont	Paris	Jussieu
22	Durand	Nancy	INPL
25	Paul	Strasbourg	Robert Schuman

- Les dépendances sont déterminées pendant l'analyse et non déduites des relations

Définition d'un schéma relationnel (4)

Clé de relation

- On appelle clé de relation un champ, ou un ensemble de champs, qui permet d'identifier un enregistrement de manière unique
- Etant donnée $R[X, Y]$, X est clé de R si et seulement si $\forall i, j, ti[X] \neq tj[X]$ dans R
- La clé ne peut être déduite de la relation : on ne peut préjuger du contenu futur des enregistrements

Appareil	Couleur
Réfrigérateur	Blanche
Robot	Rouge
Lave-vaisselle	Inox
Cuisinière	Noire
Congélateur	Blanche

- Pour caractériser les attributs de la clé :
 - dépendance fonctionnelle avec les attributs non-clé
 - liste minimale d'attributs préservant la dépendance fonctionnelle

Passage au modèle relationnel (1)

Règles de base

- Chaque classe d'entité devient une relation composée des champs de l'entité
- L'identifiant de la classe d'identité devient **clé primaire** de la relation

Ordinateur
#CodeOrd
NomOrd
PrixOrd
StockOrd



Ordinateur(CodeOrd, NomOrd, PrixOrd, StockOrd)



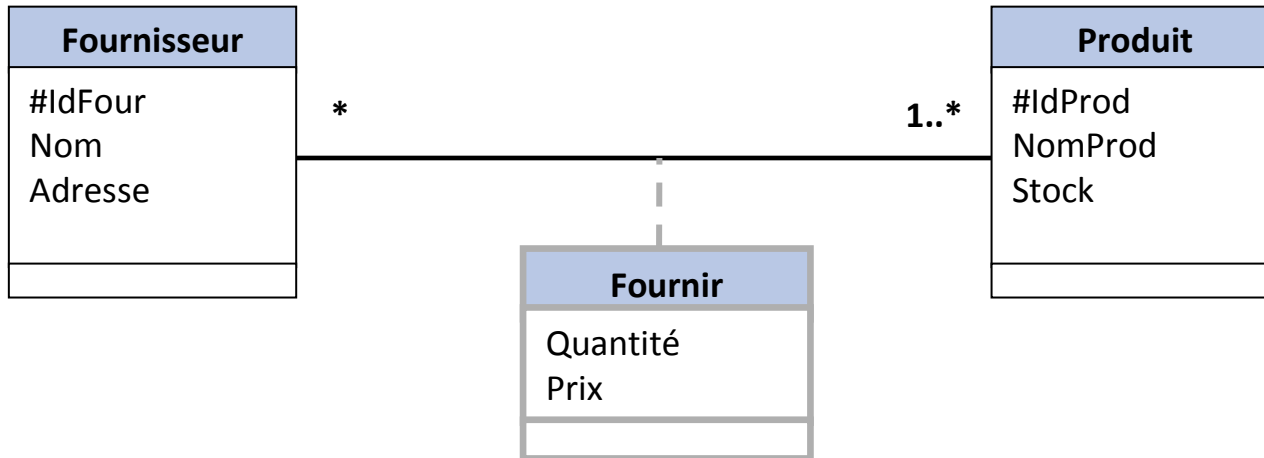
Ordinateur

<u>CodeOrd</u>	NomOrd	PrixOrd	StockOrd
DT4300	Dell Precision 4300	749	17
DL4200	Dell Latitude 4200	900	3

Passage au modèle relationnel (2)

Règles de base

- Une association devient une relation composée des 2 clés des entités associées
- L'ensemble des clés constitue celle de la relation
- Si l'association possède des attributs, ils deviennent attributs de la relation



Fournisseur(IdFour, Nom, Adresse)

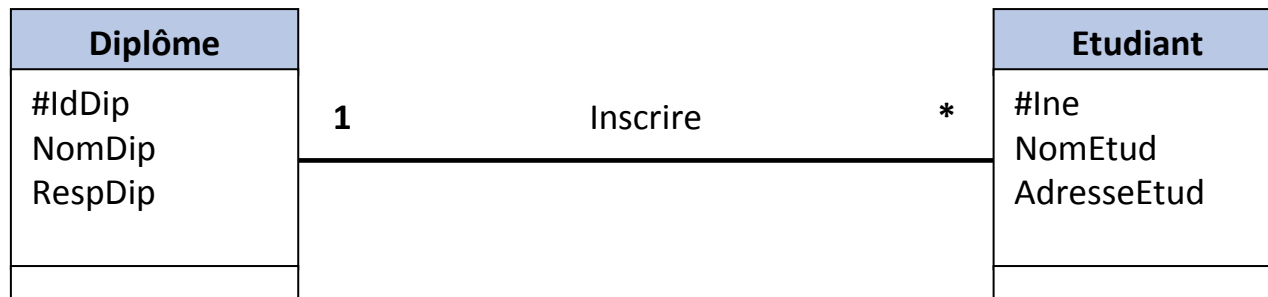
Produit(IdProd, NomProd, Stock)

Fournir(IdProd, IdFour, Quantité, Prix)

Passage au modèle relationnel (3)

Règle spécifique : association 1-N (mère-fille)

- La classe d'association disparaît
- La clé de la relation mère *glisse* dans la relation fille en tant que **clé étrangère**



Si règle précédente : **Diplôme**(IdDip, NomDip, RespDip)

Etudiant(Ine, NomEtud, AdresseEtud)

Inscrire(IdDip, Ine)

Mais la clé de la relation Inscrire n'est pas minimale : *Ine* détermine *IdDip*

→ Donc on choisit **Ine** comme clé

Mais *Ine* est déjà la clé de la relation *Etudiant*

→ On peut alors regrouper les relations *Etudiant* et *Inscrire* :

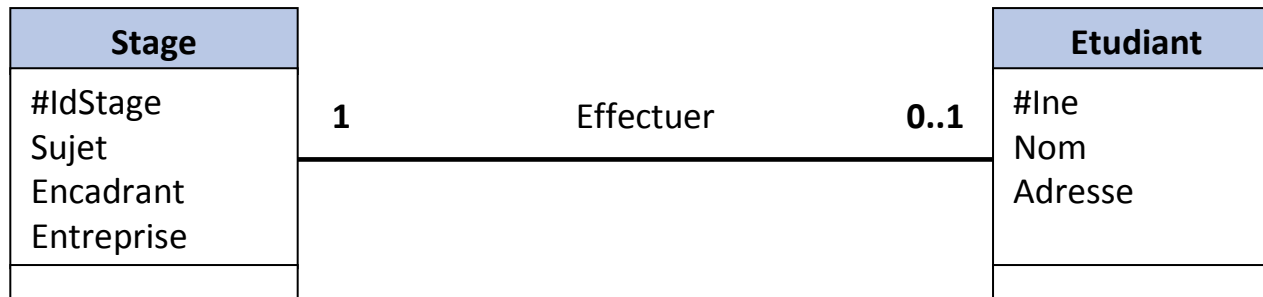
Diplôme(IdDip, NomDip, RespDip)

Etudiant(Ine, NomEtud, AdresseEtud, #IdDip)

Passage au modèle relationnel (4)

Règle spécifique : association 1-1 (*symétrique*)

- **0..1 — 0..1** : on crée une relation pour l'association
- **0..1 — 1** : idem association *mère-fille*



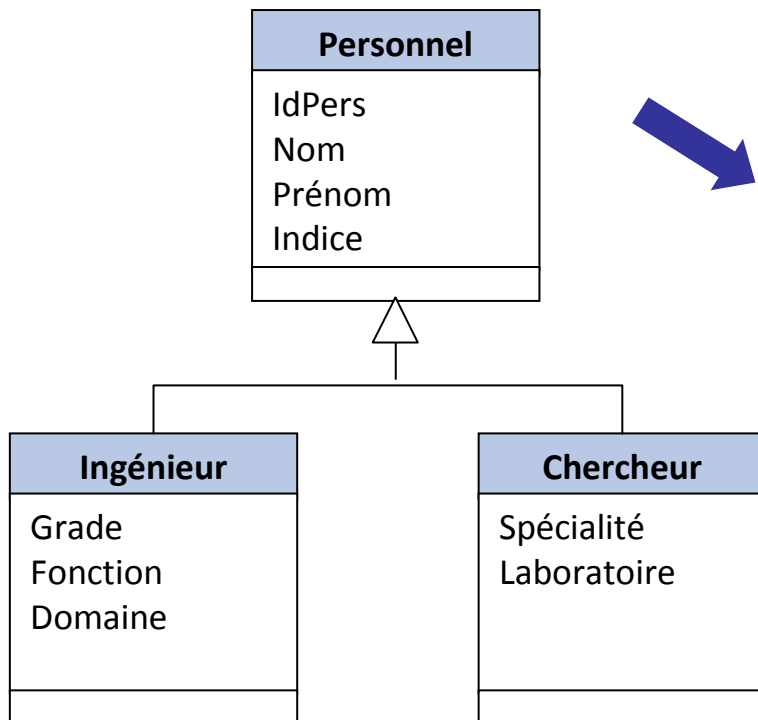
Stage(IdStage, Sujet, Encadrant, Entreprise)

Etudiant(Ine, Nom, Adresse, #IdStage)

Passage au modèle relationnel (5)

Règle spécifique : héritage (bonus)

- Chaque classe devient une relation
- La "super-classe" contient un attribut de spécialisation
- chaque relation de l'ensemble possède la même clé primaire



Personnel(IdPers, Nom, Prénom, Indice, Type)

Ingénieur(IdPers, Grade, Fonction, Domaine)

Chercheur(IdPers, Spécialité, Laboratoire)

Exercice 4 : location de DVD (1)

Enoncé général

Une société loue des DVD à des particuliers

Précisions

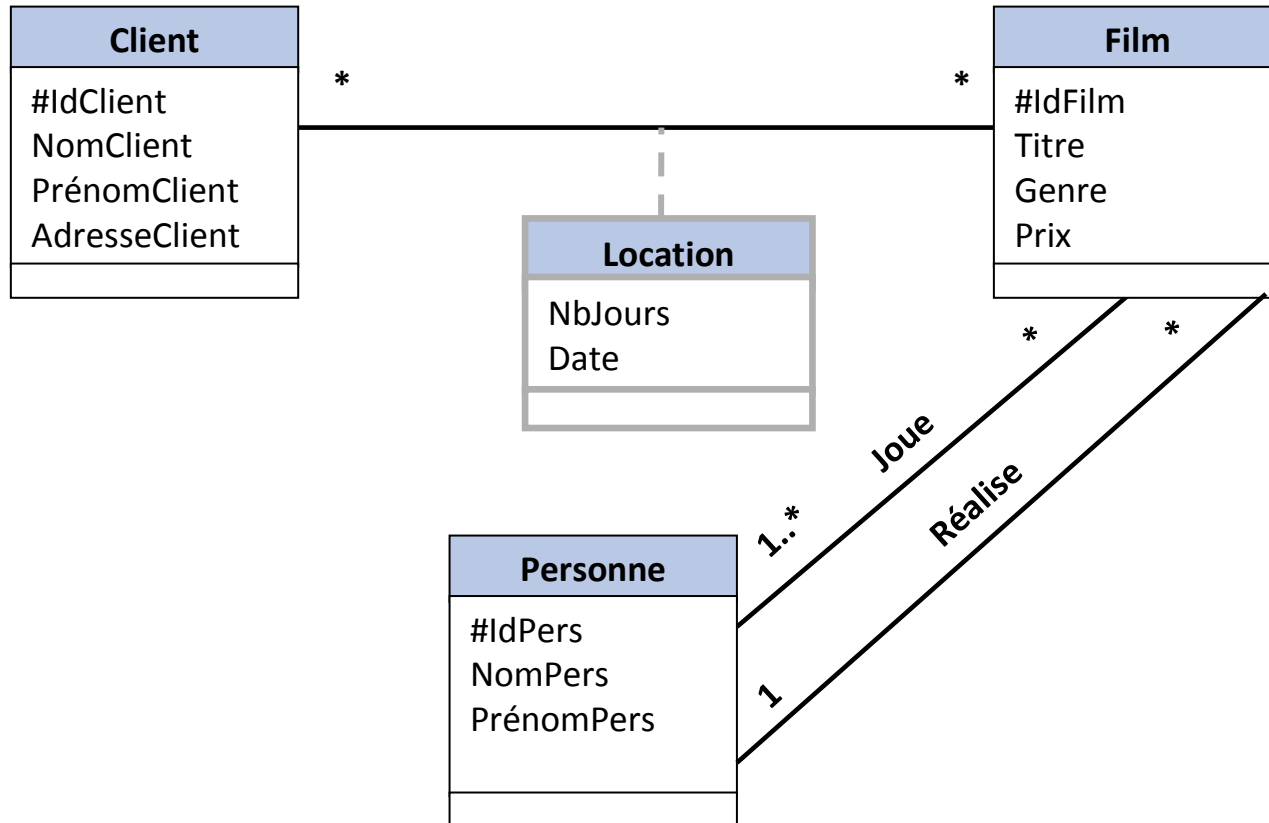
- Les clients sont définis par un numéro, un nom, un prénom et une adresse
- Les DVD sont définis par un numéro, un titre, un genre et un prix
- Les films (DVD) sont réalisés ou joués par des personnes
- Les personnes sont caractérisées par un identifiant, un nom et un prénom
- Une location est définie par un nombre de jours et une date

→ Construire le diagramme de classes de la BDD

→ Passer au modèle relationnel

Exercice 4 : location de DVD (2)

Diagramme de classes



Exercice 4 : location de DVD (3)

Passage au relationnel (1)

Client(IdClient, NomClient, PrénomClient, AdresseClient)

Film(IdFilm, Titre, Genre, Prix)

Personne(IdPers, NomPers, PrénomPers)

Location(IdClient, IdFilm, NbJours, Date)

Joue(IdFilm, IdPers)

Réalise(IdFilm, IdPers)

Exercice 4 : location de DVD (4)

Passage au relationnel (2)

Client(IdClient, NomClient, PrénomClient, AdresseClient)

Film(IdFilm, Titre, Genre, Prix, #IdPers)

Personne(IdPers, NomPers, PrénomPers)

Location(IdClient, IdFilm, NbJours, Date)

Joue(IdFilm, IdPers)

~~**Réalise**(IdFilm, IdPers)~~

Généralités

- Ensemble d'opérateurs algébriques qui manipulent les relations d'une base de données
- Le résultat de l'application d'un opérateur sur 1 ou 2 relations restitue toujours une relation
- Il existe 9 opérateurs :
 - **classiques** : union, intersection, différence, produit cartésien
 - **spécifiques** : projection, sélection, jointure, division
 - + l'agrégation

Opérations ensemblistes (1)

Union

- L'opération d'union consiste à mettre en commun les enregistrements de chaque relation
- Les enregistrements identiques ne sont pas dupliqués
- L'union est représentée par le caractère \cup
- Les relations doivent être compatibles :
 - même nombre d'attributs
 - les couples d'attributs doivent être définis sur le même domaine

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
DL4200	Dell Latitude 4200	900	3

T1

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
SV2540	Sony Vaio 2540	1529	2

T2

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

T1 \cup T2

Opérations ensemblistes (2)

Intersection

- L'opération d'union consiste à désigner les enregistrements qui sont communs à deux relations
- L'intersection est représentée par le caractère \cap
- Les relations doivent être compatibles :
 - même nombre d'attributs
 - les couples d'attributs doivent être définis sur le même domaine

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

T1

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
ITP42	IBM Thinkpad T42	1849	9
SV2540	Sony Vaio 2540	1529	2

T2

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
SV2540	Sony Vaio 2540	1529	2

T1 \cap T2

Opérations ensemblistes (3)

Différence

- L'opération de différence consiste à désigner les enregistrements qui appartiennent à une relation sans appartenir à une autre
- La différence est représentée par le caractère \setminus
- L'opération de différence n'est pas symétrique
- Les relations doivent être compatibles

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

T1

<u>Code</u>	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
ITP42	IBM Thinkpad T42	1849	9
SV2540	Sony Vaio 2540	1529	2

T2

<u>Code</u>	Nom	Prix	Stock
DL4200	Dell Latitude 4200	900	3

T1 \setminus T2

Opérations ensemblistes (4)

Produit cartésien

- Le produit cartésien permet la combinaison des enregistrements de deux relations sans tenir compte du contenu des données
- Les relations ne sont donc pas nécessairement compatibles
- Le produit cartésien est représenté par le caractère **X**
- Association de chaque n-uplet d'une relation à chaque n-uplet de l'autre relation

Appareil	Couleur
Ordinateur	Noir
Ecran	Blanc

T1

Marque	Type
Volkswagen	Polo
Peugeot	206

T2

Appareil	Couleur	Marque	Type
Ordinateur	Noir	Volkswagen	Polo
Ecran	Blanc	Volkswagen	Polo
Ordinateur	Noir	Peugeot	206
Ecran	Blanc	Peugeot	206

T1 X T2

Opérations relationnelles (1)

Projection

- La projection consiste à extraire certains **champs** d'une relation
- Le degré de la relation obtenue est donc inférieur au degré initial
- On conserve uniquement les **colonnes** sur lesquelles la projection est faite
- Notation $R = \Pi_X R1$ (tuples de R1 restreints aux attributs X)
- Les duplicatas n'apparaissent pas dans le résultats

Code	Nom	Prix	Stock
DT4300	Dell Precision 4300	900	3
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

T1

Code	Nom
DT4300	Dell Precision 4300
DL4200	Dell Latitude 4200
SV2540	Sony Vaio 2540

$\Pi_{\text{CODE, NOM}} T1$

Prix	Stock
900	3
1529	2

$\Pi_{\text{PRIX, STOCK}} T1$

Opérations relationnelles (2)

Sélection (1)

- La sélection consiste à extraire certains **enregistrements** d'une relation
- On utilise des critères pour caractériser les enregistrements sélectionnés
- On conserve uniquement les **lignes** sur lesquelles la sélection est faite
- Notation $R = \sigma_p R1$
- R comprend les mêmes attributs que R1 et les lignes vérifiant le prédicat P
- **Prédicat simple**
 - attribut **op** valeur
 - valeur **op** attribut
 - attribut **op** attribut

} **op** (opérateur de comparaison) : = != > < >= <=
- **Prédicat composé** : combinaison de prédicats simples interconnectés avec les opérateurs logiques **or**, **not** et **and**
- Le prédicat est évalué pour chaque ligne de la relation

Opérations relationnelles (3)

Sélection (2)

Code	Nom	Prix	Stock
DT4300	Dell Precision 4300	749	17
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

T1

Code	Nom	Prix	Stock
DL4200	Dell Latitude 4200	900	3
SV2540	Sony Vaio 2540	1529	2

 $\sigma_P T1$ avec $P = \text{stock} < 5$