

# BDW - Conception des BD - modélisation conceptuelle

Fabien Duchateau

*fabien.duchateau [at] univ-lyon1.fr*

Université Claude Bernard Lyon 1

2023 - 2024



<https://perso.liris.cnrs.fr/fabien.duchateau/BDW/>

# Positionnement dans BDW

## Modélisation

Schéma entité/  
association

Niveau conceptuel

Modèle  
relationnel

Niveau logique

SQL (DDL)

Niveau physique

## SGBD

Concepts

Optimisation

Base de  
données

...

Base de  
données

## Manipulation

Algèbre  
relationnelle

Combinaison  
d'opérateurs

Calculs  
relationnels

{projetés | formule}

SQL (DML)

SELECT ...  
FROM ...

## Prog. web

HTML

CSS

PHP

```
<html>
...
<link ... css>
...
<?php
?>
...
</html>
```

Ces diapositives utilisent **le genre féminin** (e.g., chercheuse, développeuses) plutôt que **l'écriture inclusive** (moins accessible, moins concise, et pas totalement inclusive)

# Motivation

Imaginons que l'on gère les candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

---

Jeu de données francisé inspiré du cours [Databases de Stanford](#)

# Motivation

Imaginons que l'on gère les candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- ▶ Bob candidate une nouvelle fois ?

# Motivation

Imaginons que l'on gère les candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- ▶ Bob candidate une nouvelle fois ?
- ▶ l'effectif de l'université *INSA* change ?

# Motivation

Imaginons que l'on gère les candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- ▶ Bob candidate une nouvelle fois ?
- ▶ l'effectif de l'université *INSA* change ?
- ▶ l'on veut ajouter une université qui n'a pas encore de candidatures ?

---

Jeu de données francisé inspiré du cours [Databases de Stanford](#)

# Motivation

Imaginons que l'on gère les candidatures d'élèves dans des universités, avec un outil type tableur :

idE	nomE	moyenneLycée	effectifLycée	nomU	département	décision	ville	effectif
123	Ana	19.5	1000	INSA	informatique	O	Lyon	36000
123	Ana	19.5	1000	UCB	electronique	N	Lyon	15000
234	Bob	18	1500	INSA	biologie	N	Lyon	36000
...	...	...	...	...	...	...	...	...

Que se passe t-il si...

- ▶ Bob candidate une nouvelle fois ?
- ▶ l'effectif de l'université *INSA* change ?
- ▶ l'on veut ajouter une université qui n'a pas encore de candidatures ?

⇒ **Mauvaise modélisation !**

---

Jeu de données francisé inspiré du cours [Databases de Stanford](#)

# Objectifs de la modélisation

Conception de base de données = modélisation et implémentation des spécifications pour la partie données

Pourquoi modéliser ?

- ▶ Structurer et organiser les données d'un domaine pour décrire un système d'informations
- ▶ Optimiser les performances
- ▶ Garantir la cohérence et la non-redondance (**ACID**)
- ▶ Difficulté à modéliser un problème directement sous une forme exploitable par un SGBD

---

<http://laurent-audibert.developpez.com/Cours-BD/>

<http://ineumann.developpez.com/tutoriels/merise/initiation-merise>



# Étapes de la modélisation (Merise)

**Niveau conceptuel** = modèle avec une forte abstraction  
(indépendant des SGBD), qui organise les données  
issues des spécifications ⇒ **diagramme**  
**Entité/Association**

**Niveau logique** = modèle qui se rapproche du type de SGBD choisi  
(e.g., les SGBD relationnels) ⇒ **modèle relationnel**

**Niveau physique** = modèle de création des données pour un SGBD  
particulier (e.g., MariaDB, PostgreSQL) ⇒ **SQL**  
**comme LDD**

# Plan

Diagramme Entité/Association de base

Modèle Entité/Association étendu

Méthodologie

# Modèles conceptuels

Pour concevoir une base de données, divers modèles conceptuels :

- ▶ Diagramme de Hasse (peu utilisé) : exploitation de dépendances entre les données
- ▶ Méthode Unified Modeling Language (UML) : diagramme de classes (décrivant les responsabilités, le comportement et le type d'un ensemble d'objets)
- ▶ Diagramme Entité/Association ("*entity-relationship diagram*")
  - ▶ Exploitation d'hypothèses descriptives du contexte
  - ▶ Différents formalismes (Merise, Chen, etc.)

---

[http://fr.wikipedia.org/wiki/Mod%C3%A8le\\_entit%C3%A9-association](http://fr.wikipedia.org/wiki/Mod%C3%A8le_entit%C3%A9-association)  
[http://fr.wikipedia.org/wiki/Base\\_de\\_donn%C3%A9es\\_relationnelle](http://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle)

# Diagramme Entité/Association

Caractéristiques du diagramme E/A :

- ▶ Modélisation graphique des entités, associations et propriétés
- ▶ Détection facilitée des erreurs de conception
- ▶ Traduction automatique dans un modèle logique (Relationnel)



---

Chen, P. *The entity-relationship model—toward a unified view of data.*  
ACM Transactions on Database Systems (1976)

# Entités

Type d'entité :

- ▶ Objet concret (e.g., un vélo) ou abstrait (e.g., une responsabilité), qui possède une "**identité**" et des **propriétés**
- ▶ Souvent dérivé à partir du dictionnaire de données et des dépendances fonctionnelles (DF)

Exemples de type d'entité et exemples d'instance :

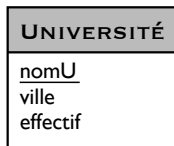
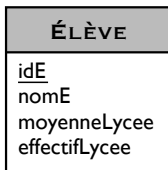
- ▶ Une université (*UCBL, INSA, ...*)
- ▶ Une élève (*Ana, Bob, Chloé, ...*)

Par abus de langage : type d'entité → entité

# Entités - représentation

Représentation graphique de deux entités :

- ▶ ÉLÈVE avec 4 propriétés
- ▶ UNIVERSITÉ avec 3 propriétés



# Associations

Type d'association :

- ▶ Relation entre deux ou plusieurs entités
- ▶ Souvent dérivés à partir du dictionnaire de données et des dépendances fonctionnelles (DF)

Exemples de type d'association et exemples d'instance :

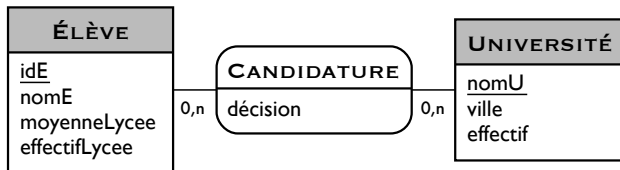
- ▶ “candidature”, entre une entité *élève* et une entité *université*  
→ *Ana* candidate à l'*INSA*
- ▶ “enseigne” entre les entités *enseignant*, *UE* et *promotion*  
→ *Fabien* enseigne *BDW* aux *étudiant.e.s de L2*

Par abus de langage : type d'association → association

# Associations - représentation

Représentation graphique d'une association entre deux entités :

- ▶ CANDIDATURE possède 1 propriété





# Propriétés

## Propriété :

- ▶ Représente une donnée pour une entité ou une association
- ▶ Valeur dans un domaine de valeur d'un type simple (entier, booléen, chaîne de caractères, date, ...)

## Exemples de propriétés :

- ▶ "*nomE*" de l'entité *élève*, de type chaîne de caractères, désigne le nom de l'élève
- ▶ "*décision*" dans l'association *candidature*, de type caractère, représente la décision sur la candidature

## Identifiants d'entité

Un **identifiant** d'entité est une propriété qui permet de repérer une instance **de manière unique et sans ambiguïté** parmi toutes les instances de cette entité. Exemples d'identifiants :

- ▶ Numéro INSEE (numéro de sécurité sociale) pour une personne (en France)
- ▶ Plaque d'immatriculation pour les voitures

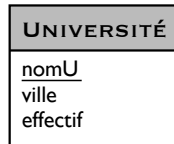
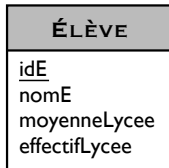
Un identifiant peut être constitué d'une ou plusieurs propriétés

La ou les propriétés identifiant d'une entité sont soulignés

# Identifiants d'entité - représentation

Représentation graphique d'identifiants :

- ▶ ÉLÈVE avec 4 propriétés, et dont *idE* est l'identifiant
- ▶ UNIVERSITÉ avec 3 propriétés, et dont *nomU* est l'identifiant



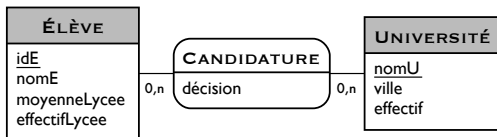
# Identifiants d'association

Identifiant d'une association :

- ▶ Non implicite
- ▶ S'obtient en combinant les identifiants des entités porteuses (i.e., liées à cette association)

Exemple d'identifiant d'association :

- ▶ L'identifiant de l'association `CANDIDATURE` se compose de nomU et idE



# Cardinalités

Une cardinalité exprime le nombre d'associations dans lesquelles une instance d'entité peut apparaitre

- ▶ Définition de restrictions (minimum et maximum)

Exemples de restrictions :

- ▶ Une élève peut candidater dans plusieurs universités
- ▶ Une université ne peut accueillir plus de 1000 étudiantes

## Cardinalités (2)

Cardinalité sous la forme  $c_{min}, c_{max}$  ou  $c_{min} : c_{max}$  avec :

- ▶  $c_{min}$  est le nombre minimal d'associations possibles
- ▶  $c_{max}$  est le nombre maximal d'associations possibles
- ▶  $c_{max}$  peut prendre une valeur infinie (notée  $N$  ou  $M$ )

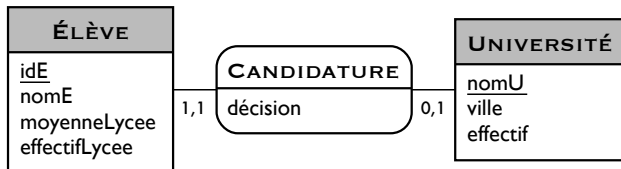
Cardinalités les plus fréquentes :

- ▶ 0, 1 (zéro à un)
- ▶ 1, 1 (un à un)
- ▶ 0,  $N$  (zéro à plusieurs)
- ▶ 1,  $N$  (un à plusieurs)

# Cardinalités - exemples (1)

Exemple de cardinalités 0,1 et 1,1 :

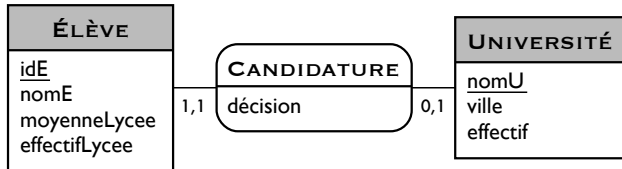
- ▶ Une élève peut candidater dans une et une seule université
- ▶ Une université peut recevoir zéro ou une candidature



# Cardinalités - exemples (1)

Exemple de cardinalités 0,1 et 1,1 :

- ▶ Une élève peut candidater dans une et une seule université
  - ▶ Une université peut recevoir zéro ou une candidature
- ⇒ **Exemple évidemment irréal !**

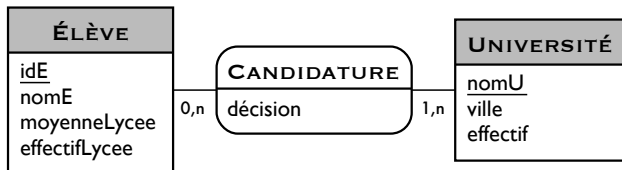




## Cardinalités - exemples (2)

Exemple de cardinalités 0,  $N$  et 1,  $N$  :

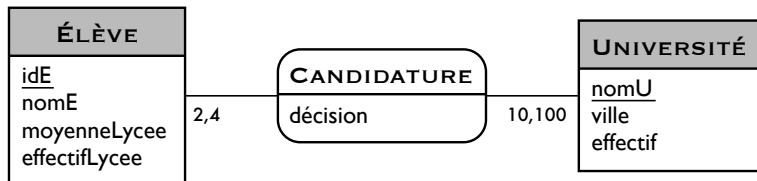
- ▶ Une élève peut candidater dans zéro ou plusieurs universités
- ▶ Une université reçoit une candidature au minimum



## Cardinalités - exemples (3)

Exemple de cardinalités "moins fréquentes" :

- ▶ Une élève candidate dans deux universités au minimum et jusqu'à quatre universités au maximum
- ▶ Une université reçoit entre dix et cent candidatures



# Associations multiples

Jusqu'à présent, uniquement des associations binaires

Possibilité d'associer plusieurs entités :

- ▶ Associations ternaires (3 entités)
- ▶ Association n-aires ( $n$  entités,  $n > 3$ )

---

R. Camps, *Transforming N-ary Relationships to Database Schemas : An Old and Forgotten Problem*, RR (2002)

# Associations multiples

Jusqu'à présent, uniquement des associations binaires

Possibilité d'associer plusieurs entités :

- ▶ Associations ternaires (3 entités)
- ▶ Association n-aires ( $n$  entités,  $n > 3$ )

Il est possible d'éviter les associations n-aires

⇒ Une association n-aire peut souvent être décomposée en une combinaison d'association binaires

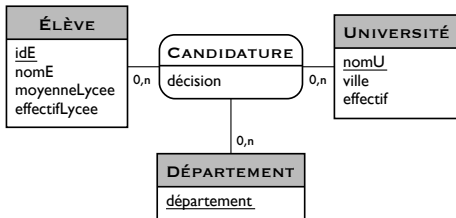
---

R. Camps, *Transforming N-ary Relationships to Database Schemas : An Old and Forgotten Problem*, RR (2002)

# Associations multiples - représentation

Représentation graphique d'une association ternaire :

- ▶ Une élève peut candidater dans plusieurs départements de différentes universités
- ▶ Une université reçoit 0 ou plusieurs candidatures d'élève dans différents départements
- ▶ Idem pour département (pour un département donné, on peut trouver plusieurs paires élève / université)



# En résumé

Le diagramme Entité/Association :

- ▶ Modèle conceptuel (niveau d'abstraction élevé)
- ▶ Concepts principaux = entité, association, identifiant, propriété, cardinalité
- ▶ Formalisme Merise :
  - ▶ Les propriétés des entités et associations sont regroupées
  - ▶ Lecture des cardinalités en suivant le sens  
"entité<sub>1</sub> → association → cardinalité côté entité<sub>1</sub> → entité<sub>2</sub>"



# Plan

Diagramme Entité/Association de base

Modèle Entité/Association étendu

Méthodologie

# Extensions au modèle de base

Limitations du modèle de base par rapport à la réalité :

- ▶ Identifiant d'une entité basé sur une autre entité
- ▶ Relations hiérarchiques entre entités
- ▶ Association vers une autre association

Quelques extensions pour répondre à ces besoins :

- ▶ Entité faible
- ▶ Association réflexive
- ▶ Spécialisation, généralisation
- ▶ Agrégation

---

[https://en.wikipedia.org/wiki/Enhanced\\_entity%E2%80%93relationship\\_model](https://en.wikipedia.org/wiki/Enhanced_entity%E2%80%93relationship_model)

J. Smith and D. Smith. *Database abstractions : aggregation and generalization*.  
ACM Transactions on Database Systems (1977)



## Entité faible - motivation

Exemple : modélisons une salle de cours, qui se trouve dans un bâtiment. Chaque bâtiment possède un nom unique et une adresse. Une salle a un numéro et une capacité.



*Une solution possible. Vous semble-t-elle correcte ?*

## Entité faible - motivation

Exemple : modélisons une salle de cours, qui se trouve dans un bâtiment. Chaque bâtiment possède un nom unique et une adresse. Une salle a un numéro et une capacité.



*Une solution possible. Vous semble-t-elle correcte ?*

L'identification d'une salle est relative au bâtiment car une salle peut avoir le même numéro dans deux bâtiments différents (e.g., salle 1 à *Ariane* et salle 1 à *Grignard*)

# Entité faible - définition

Définition d'une entité faible :

- ▶ L'identifiant d'une entité ne permet pas de l'identifier de manière unique
- ▶ L'existence de ses instances dépend de l'existence d'une instance d'une autre entité

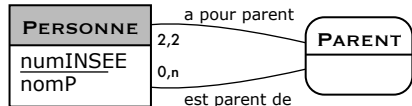
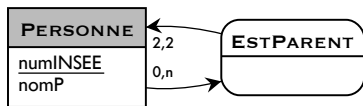


*Représentation graphique d'une entité faible*

# Association réflexive

Une association réflexive est une association dont les deux branches sont reliées à la même entité

- ▶ Les cardinalités de chaque branche sont en général différentes
- ▶ Il peut être nécessaire de préciser le rôle de chaque branche pour faciliter la compréhension du diagramme

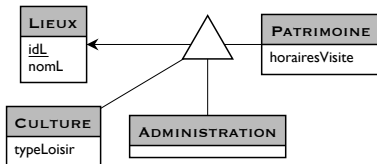


*Représentations graphiques d'une association réflexive*

# Spécialisation, généralisation

Définition de la spécialisation et généralisation ("is a" en anglais) :

- ▶ Une entité A est une spécialisation d'une entité B si chaque instance de A est une instance de B
- ▶ L'entité B est un sur-type (ou "classe mère", généralisée)
- ▶ L'entité A est un sous-type (ou "classe fille", spécialisée)

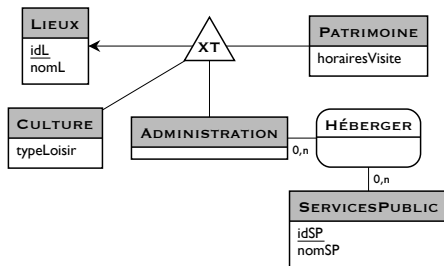


*Représentation graphique d'une spécialisation / généralisation par un triangle et une flèche vers le sur-type*

# Spécialisation, généralisation - intérêts

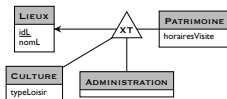
Intérêts de la spécialisation et généralisation :

- ▶ Héritage des propriétés : le sous-type hérite des propriétés de l'entité sur-type
- ▶ Propriétés uniquement pour un sous-type ou sur-type
- ▶ Association uniquement pour un sous-type ou sur-type

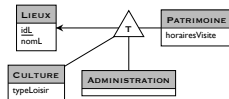


# Spécialisation, généralisation - cardinalités

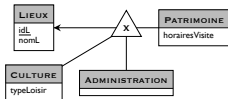
- ▶ **X** (exclusive), une instance est au maximum d'un sous-type
- ▶ **T** (totale), toute instance a au moins un sous-type
- ▶ **XT** (exclusive et totale)
- ▶ Ni exclusive, ni totale



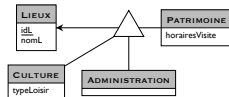
*Un lieu est forcément culture, administration ou patrimoine*



*Un lieu est forcément culture, administration et/ou patrimoine*



*Un lieu peut être d'un type parmi culture, administration ou patrimoine*

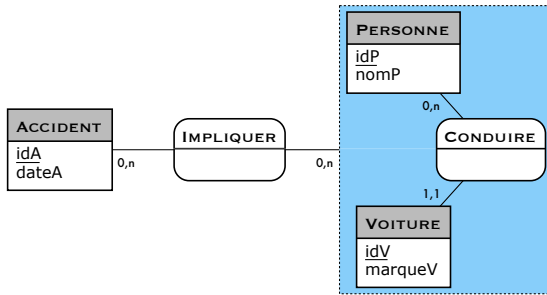


*Un lieu peut être culture, administration et/ou patrimoine*

# Agrégation

L'agrégation est une abstraction par laquelle un groupe d'une ou plusieurs associations et leurs entités porteuses sont considérés comme une nouvelle entité

- ▶ Cette nouvelle entité peut participer à une autre association
- ▶ Cela permet de décrire des entités complexes



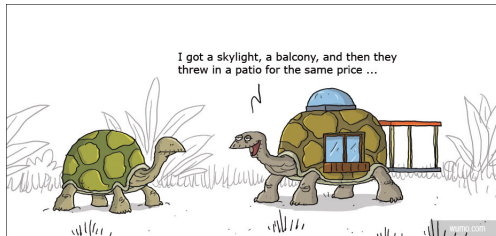
*Représentation graphique d'une agrégation*



# En résumé

Plusieurs extensions du diagramme E/A :

- ▶ Entité faible, rôles d'une association réflexive, spécialisation / généralisation et agrégation
- ▶ Peuvent complexifier la compréhension du modèle et sa transformation ultérieure



<http://wumo.com/>

# Plan

Diagramme Entité/Association de base

Modèle Entité/Association étendu

Méthodologie

# Comment construire un diagramme E/A

Des spécifications sont établies en échangeant avec les utilisatrices ou clientes, notamment concernant les données à stocker en base

Quelques réflexions pour la modélisation :

- ▶ Ce concept doit-il être représenté comme une entité ou une propriété (e.g., une adresse) ?
- ▶ Ce concept doit-il être représenté comme une entité ou une association ?
- ▶ Cette cardinalité est-elle correcte ?
- ▶ Faut-il utiliser les extensions (e.g., agrégation) ?

# Entité ou propriété ?

Un concept peut être représenté comme une propriété ou une entité à part entière (e.g., l'adresse d'une personne)

Trois cas où l'on privilégiera l'utilisation d'une entité :

- ▶ Différentes instances du concept sont reliées à une même instance (e.g., une personne possède plusieurs adresses)
- ▶ Modélisation de la structure du concept (e.g., l'adresse est décomposée en numéro, rue, code postal, etc.)
- ▶ Besoin d'un troisième identifiant pour une association (e.g., une personne emménage à une adresse, besoin d'avoir la date dans une entité à part et pas dans l'association pour les cas où la personne revient à la même adresse)

# Entité ou association ?

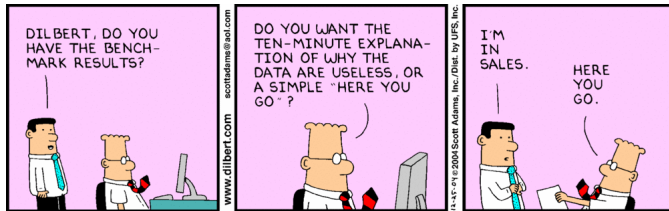
Un concept peut être propriété d'une association ou nécessiter la création d'une autre entité

Exemple : des employées, dont certaines dirigent un ou plusieurs départements. Un budget est alloué à chaque dirigeante :

- ▶ Soit le budget est spécifique à chaque département, alors le budget sera une propriété de l'association entre les entités employée et département
- ▶ Soit le budget est global pour la dirigeante, alors il faudra créer une entité dirigeante avec une propriété budget

# Choix correct d'une cardinalité ?

- ▶ Ne pas interpréter (e.g., référentiel personnel pour les cardinalités)
- ▶ Utiliser le bon sens pour mettre les cardinalités qui ne sont pas spécifiées dans les spécifications, et mettre des cardinalités ouvertes en cas d'incertitude (e.g., 0,  $N$ )
- ▶ Ne pas ajouter d'informations dans le diagramme



# Association ternaire ou agrégation ?

Une association ternaire et une agrégation ont une utilisation proche, et le choix est souvent dicté par des contraintes

Utilisation de l'agrégation :

- ▶ Si une propriété ne concerne qu'une seule des deux associations (de l'agrégation)
- ▶ Si l'on a une contrainte de cardinalité 0, 1 ou 1, 1

# Méthodologie de conception d'un diagramme E/A

Une méthodologie pour modéliser des spécifications textuelles en diagramme E/A :

1. Lire l'ensemble des spécifications
2. Établir la liste des entités
3. Déterminer les propriétés de chaque entité en choisissant un identifiant
4. Établir les relations entre les différentes entités
5. Déterminer les propriétés de chaque relation et définir les cardinalités
6. Vérifier la cohérence et la pertinence du schéma obtenu



# Exemple de modélisation conceptuelle

## Étape 1 : lecture des spécifications

Une série, identifiée par son nom, est découpée en plusieurs saisons. Chaque saison possède un identifiant et une date de lancement. Les saisons contiennent un certain nombre d'épisodes. Chaque épisode, identifié par un numéro qui redémarre à 1 à chaque saison, possède un titre et inclut au moins deux acteurs. Une actrice a un numéro INSEE unique, un nom et un prénom, et son salaire dépend de l'épisode.

# Exemple de modélisation conceptuelle

## Étape 2 : détection des entités

Une **série**, identifiée par son nom, est découpée en plusieurs saisons. Chaque **saison** possède un identifiant et une date de lancement. Les saisons contiennent un certain nombre d'épisodes. Chaque **épisode**, identifié par un numéro qui redémarre à 1 à chaque saison, possède un titre et inclut au moins deux acteurs. Une **actrice** a un numéro INSEE unique, un nom et un prénom, et son salaire dépend de l'épisode.

# Exemple de modélisation conceptuelle

## Étape 2 : détection des entités



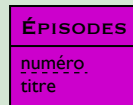
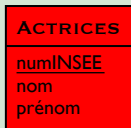
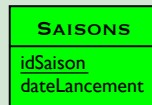
# Exemple de modélisation conceptuelle

## Étape 3 : détection des propriétés et identifiants

Une **série**, identifiée par son **nom**, est découpée en plusieurs saisons. Chaque **saison** possède un **identifiant** et une **date de lancement**. Les saisons contiennent un certain nombre d'épisodes. Chaque **épisode**, identifié par un **numéro** qui redémarre à 1 à chaque saison, possède un **titre** et inclut au moins deux acteurs. Une **actrice** a un **numéro INSEE** unique, un **nom** et un **prénom**, et son salaire dépend de l'épisode.

# Exemple de modélisation conceptuelle

## Étape 3 : détection des propriétés et identifiants



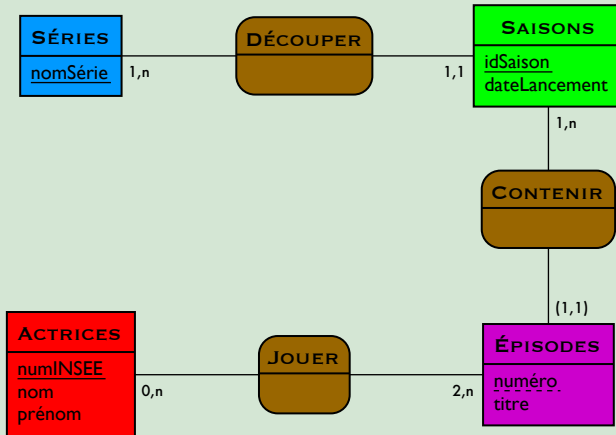
# Exemple de modélisation conceptuelle

## Étape 4 : détection des associations et cardinalités

Une **série**, identifiée par son **nom**, est **découpée** en plusieurs saisons. Chaque **saison** possède un **identifiant** et une **date de lancement**. Les saisons **contiennent** un certain nombre d'épisodes. Chaque **épisode**, identifié par un **numéro qui redémarre à 1 à chaque saison**, possède un **titre** et **inclut** au moins deux acteurs. Une **actrice** a un **numéro INSEE** unique, un **nom** et un **prénom**, et son salaire dépend de l'épisode.

# Exemple de modélisation conceptuelle

## Étape 4 : détection des associations et cardinalités



# Exemple de modélisation conceptuelle

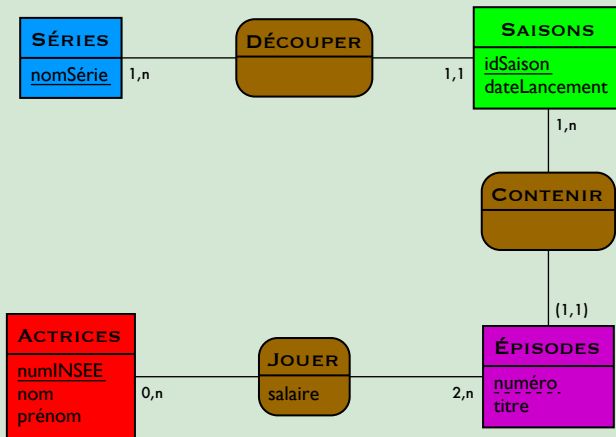
## Étape 5 : détection des propriétés d'associations

Une **série**, identifiée par son **nom**, est **découpée** en plusieurs saisons. Chaque **saison** possède un **identifiant** et une **date de lancement**. Les saisons **contiennent** un certain nombre d'épisodes. Chaque **épisode**, identifié par un **numéro qui redémarre à 1 à chaque saison**, possède un **titre** et **inclut** au moins deux acteurs. Une **actrice** a un **numéro INSEE** unique, un **nom** et un **prénom**, et son **salaire** dépend de l'épisode.



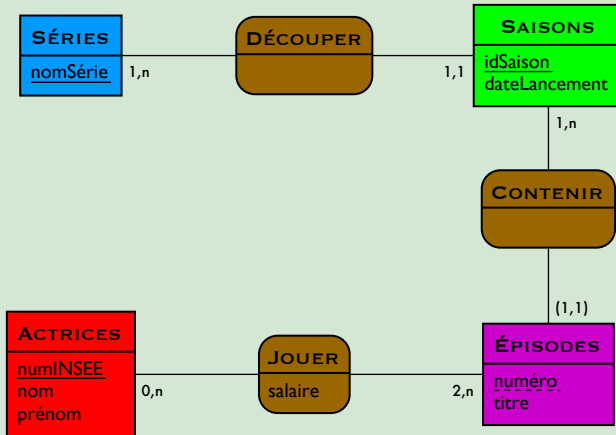
# Exemple de modélisation conceptuelle

## Étape 5 : détection des propriétés d'associations



# Exemple de modélisation conceptuelle

Cette modélisation respecte t-elle la réalité ?



# Outils de modélisation (1)

Différentes catégories d'outil :

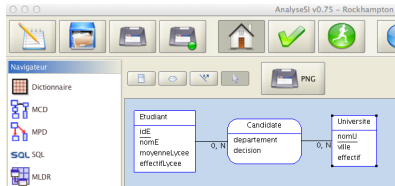
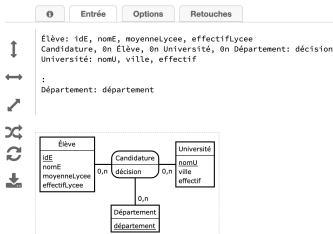
- ▶ Des outils graphiques (pour dessiner, e.g., [Dia](#) ou [draw.io](#))
- ▶ Des outils de modélisation (assistance pour dessiner un diagramme E/A, puis générer les modèles suivants)

Attention :

- ▶ Vérifier ce qui est généré par les outils !
- ▶ De nombreux outils de modélisation proposent directement de modéliser en Relationnel (niveau logique) et pas en Entité/Association (niveau conceptuel)

## Outils de modélisation (2)

- ▶ **Mocodo** : <http://www.mocodo.net/>
- ▶ **Analyse SI** : <https://launchpad.net/analysesi>
- ▶ **Looping** (Windows) : <http://www.looping-mcd.fr/>
- ▶ JMerise (limité) : <http://www.jfreesoft.com/JMerise/>
- ▶ Open Modelsphere : <http://www.modelsphere.org/>
- ▶ MySQL Workbench : <http://mysqlworkbench.org/>



# En résumé

- ▶ Une méthodologie basée sur l'exploitation des spécifications
- ▶ Pour un ensemble de spécifications, pas forcément un seul diagramme correct !
- ▶ Le diagramme E/A doit être validé par les utilisatrices
- ▶ En BDA, une méthodologie plus formelle basée sur les dépendances fonctionnelles



[https://fr.wikipedia.org/wiki/Calvin\\_et\\_Hobbes](https://fr.wikipedia.org/wiki/Calvin_et_Hobbes)