

# INFORMATIQUE Développement d'applications

## BLOC 1

### UE03 Conception d'applications

#### Analyse

#### Chapitre 4.2 : MCD Patterns et anti-patterns

Vincent Reip

Octobre 2025

# Objectif

- Au terme de ce chapitre, l'étudiant sera capable de :
  - Expliquer ce que sont les patterns de modélisation de données.
  - Identifier les situations où l'utilisation d'un pattern est pertinente.
  - Reconnaître les principaux patterns de modélisation et les anti-patterns.
  - Appliquer ces patterns à partir d'un discours client pour produire un modèle conceptuel de données (MCD) correct et cohérent.
  - Éviter les erreurs classiques de modélisation (anti-patterns).

# Patrons (patterns) de modélisation de données ?

- Lors de la conception d'un modèle conceptuel de données (MCD), l'analyste est régulièrement confronté à des problèmes récurrents :
  - Comment modéliser les états d'une entité
  - Comment modéliser l'adhésion à une/des catégories ?
  - Comment conserver l'historique des prix d'un produit ?
  - Comment gérer une hiérarchie entre employés d'une même entreprise ?
  - ...
- Ces situations apparaissent dans de nombreux contextes différents, mais leur nature reste similaire.
- Pour les résoudre de manière robuste et cohérente, on peut s'appuyer sur des **patterns de modélisation de données**.

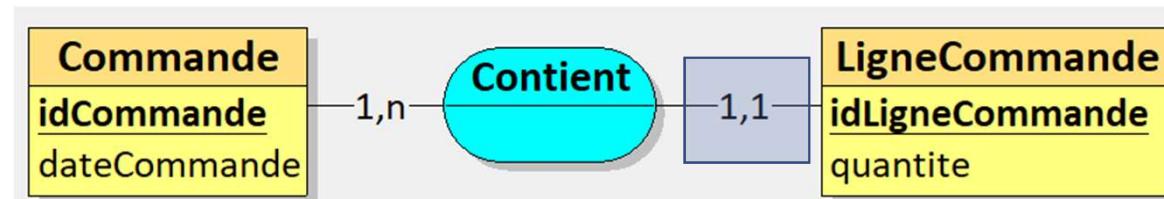
# Patterns / anti-patterns

- Un *pattern de modélisation* est une **solution récurrente et validée** à un problème typique rencontré dans la modélisation des données.
- Il s'agit donc d'un **schéma de conception réutilisable en l'adaptant à la situation métier**.
- Inversement, il existe aussi des **anti-patterns** (cfr. musée des erreurs): ce sont des solutions pouvant être perçues comme correctes mais qui posent des problèmes graves :
  - Ne sont pas conformes aux besoins fonctionnels
  - Provoquent de la redondance
  - Sortent du cadre conceptuel.
- Les connaître et les identifier permet d'éviter les erreurs classiques.

# Entité faible (entité dépendante)

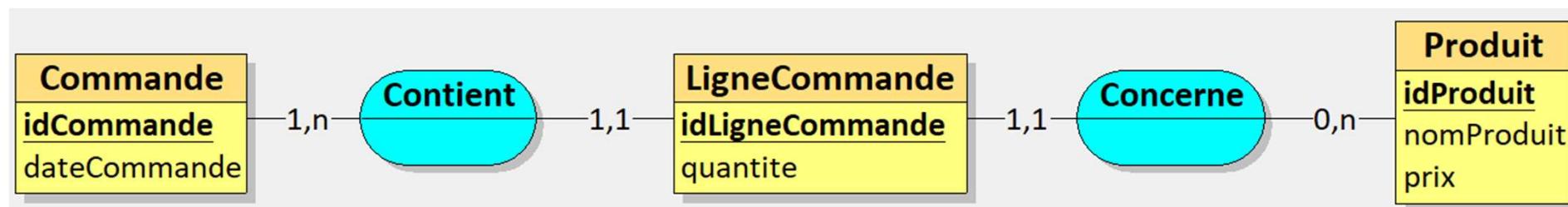
- Certaines entités représentent des objets ou des faits qui n'ont de sens que parce qu'ils sont rattachés à une autre entité.
  - Une occurrence de cette entité ne peut pas exister de manière autonome : elle doit obligatoirement être liée à une occurrence d'une autre entité, qu'on appelle parfois l'**« entité mère »**.
- **Discours client :**
  - « *On peut commander plusieurs produits dans une seule commande.* »
    - Une commande est donc composée de plusieurs lignes de commande.
    - Chaque ligne de commande doit être obligatoirement liée à une commande
    - Il faut pouvoir spécifier le produit concerné par la ligne de commande ainsi que la quantité
  - « *La facture contient les détails des services facturés.* »
    - Une ligne de facture n'a de sens que rattachée à une facture.
  - « *Une série télévisée comporte plusieurs épisodes.* »
    - Un épisode n'existe que par rapport à une (et une seule) série.

# Entité faible (entité dépendante)



Ce sont les cardinalités (1,1) qui assurent qu'une ligne de commande sera toujours liée à une facture

Une occurrence de l'entité 'LigneCommande' ne peut exister sans être liée à une occurrence de l'entité 'Commande'.



Il faut pouvoir spécifier le produit concerné par la ligne de commande (ainsi que la quantité).

# Catégorisation simple

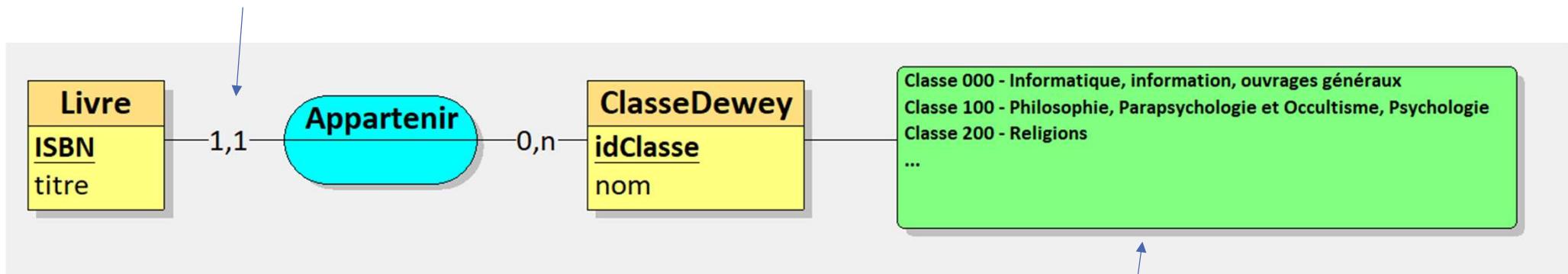
- Une entité doit être rattachée à une autre pour en catégoriser les occurrences (catégorie, modèle, classe, espèce...)
- **Discours client :**
  - « *La Classification Décimale de Dewey (CDD) est un système de classification universellement utilisé par les bibliothèques pour organiser leurs collections par sujet. Elle divise le savoir en dix classes principales telles que les arts, l'informatique, la philosophie, les sciences sociales, la littérature... Chaque ouvrage sera classé dans une seul classe.* »
  - « *Chaque voiture appartient à une catégorie (coupé, berline, SUV...).* »
    - On considère qu'une voiture ne peut être en même temps une berline et un SUV
  - « *Les logements sont catégorisés en 'Appartement', 'Maison' ou 'Château'.* »



# Catégorisation simple



Les cardinalités assurent que chaque livre est catégorisé dans une et une seule classe de la classification de Dewey



**Bonne pratique :** donner des exemples d'occurrences pour améliorer la compréhension du modèle.

# Catégorisation multiple

- Une entité doit être rattachée à une autre pour en catégoriser les occurrences (catégorie, modèle, classe, espèce...)
- **Discours client :**
  - « *Un film peut appartenir à plusieurs genres.* »



## 1. Mickey 17 (2025)

2 h 17 min. Sortie : 5 mars 2025 (France). **Science-fiction, Action, Comédie**

Film de [Bong Joon-Ho](#)



## 2. The Brutalist (2024)

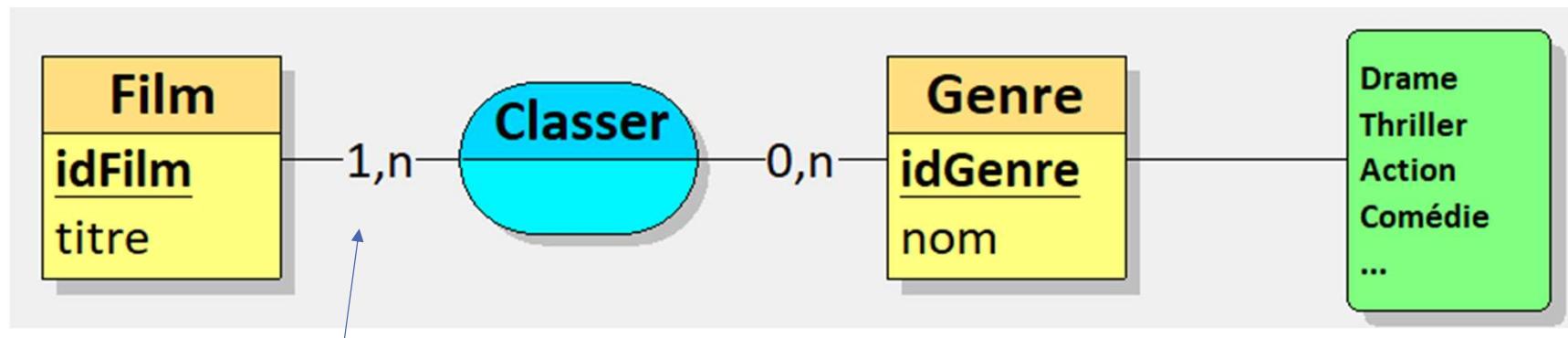
3 h 35 min. Sortie : 12 février 2025 (France). **Drame, Romance**

Film de [Brady Corbet](#)

- « *La recette des lasagnes sera classée dans les ‘Plats principaux’ mais aussi dans ‘Cuisine italienne’ et dans ‘Recettes au four’.* »



# Catégorisation multiple



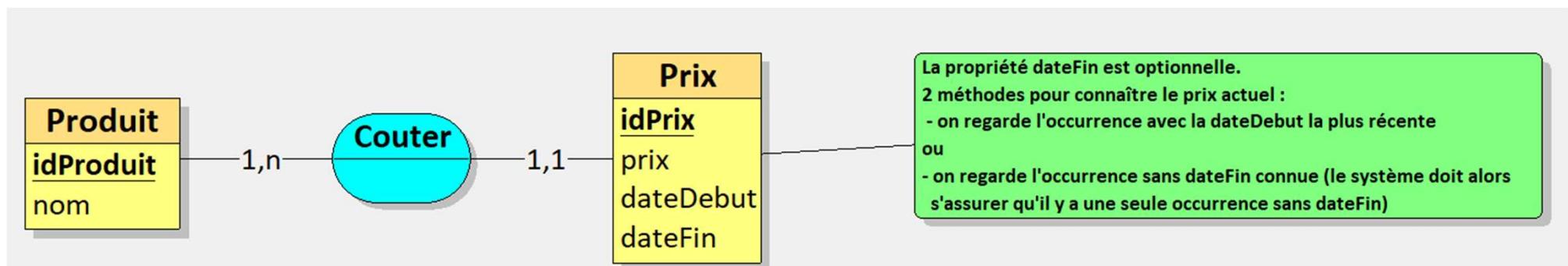
Les cardinalités permettent à un film d'être classé dans plusieurs genres (et chaque film doit être classé dans au moins un genre).

# Historisation d'attributs

- Les données d'une entité évoluent au fil du temps mais il faut conserver les versions successives.
- **Discours client :**
  - « *Le prix d'un produit peut augmenter ou diminuer au fil du temps mais je dois pouvoir connaître le prix qui était pratiqué à n'importe quelle date dans le passé. »*
  - « *Le salaire de nos employés est révisé annuellement et il augmente en fonction des performances... »*

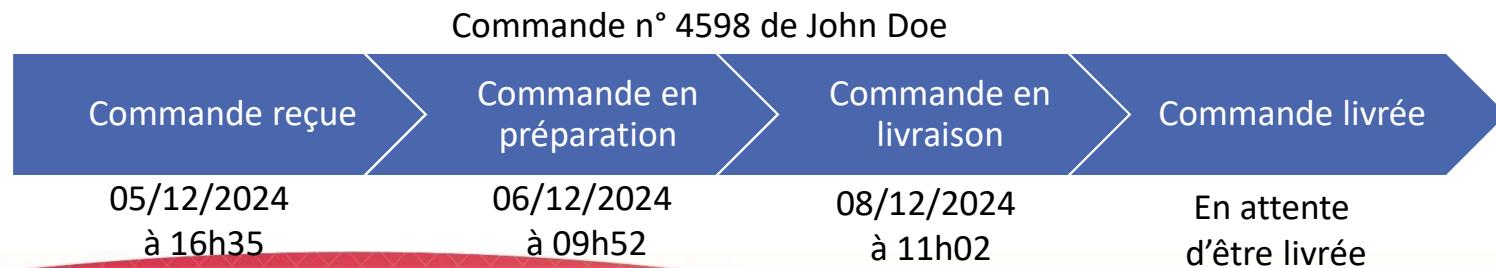


# Historisation d'attributs

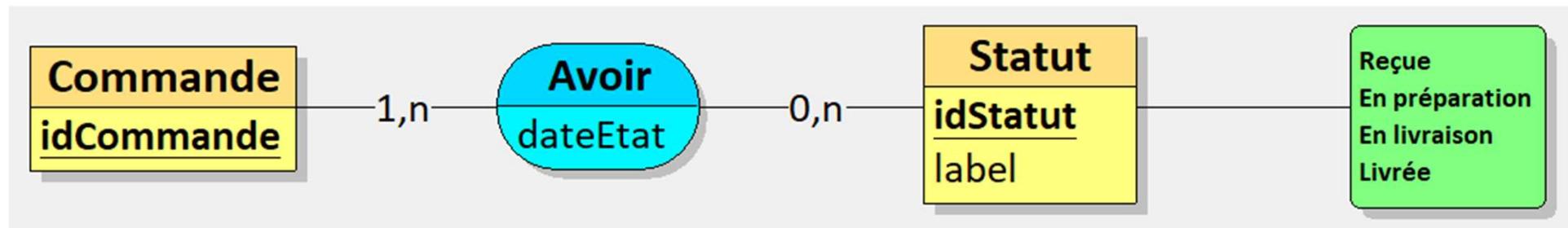


# Historisation par référence

- Les caractéristiques d'une entité évoluent au fil du temps mais il faut conserver les versions successives (en évitant la redondance).
- **Discours client :**
  - « *La fonction d'un employé peut changer au cours de sa carrière mais le département Ressources humaines doit pouvoir imprimer la fiche-carrière d'un employé.* »
  - « *L'évolution temporelle du statut des commandes doit pouvoir être consultée aisément.* »



# Historisation par référence



Comment déterminer l'état actuel d'une commande ?

# Le cas de l'adresse...

- Les données d'une adresse postale doivent être structurées pour permettre un traitement et une validation adéquate.
- Cas particulier des systèmes spécialisés :
  - Le système connaît toutes les rues / localités et peut faire des suggestions et/ou valider les données d'une adresse
- **Discours client :**
  - « *Je veux pouvoir obtenir la liste de mes clients habitant en province de Liège.* »
  - « *Le rapport des ventes indiquera le chiffre d'affaire par localité et par pays.* »

# Le cas de l'adresse...

**Address Form**

First Name  Last Name

Residence Address

City  State / Province  ZIP / Postal code

**Submit**

**Vos informations**

Prénom*	Marion
Nom*	Jomphe
E-mail*	demo@qualifio.com
Nom de rue*	
Numéro*	
Numéro de boîte	
Code postal*	83200
Localité*	Toulon
Pays*	France

**Envoyer**

**SAISIR LES INFORMATIONS DU VOYAGE**

Veuillez donner toutes les réponses en anglais.  
Les champs obligatoires sont indiqués par un astérisque rouge \*.

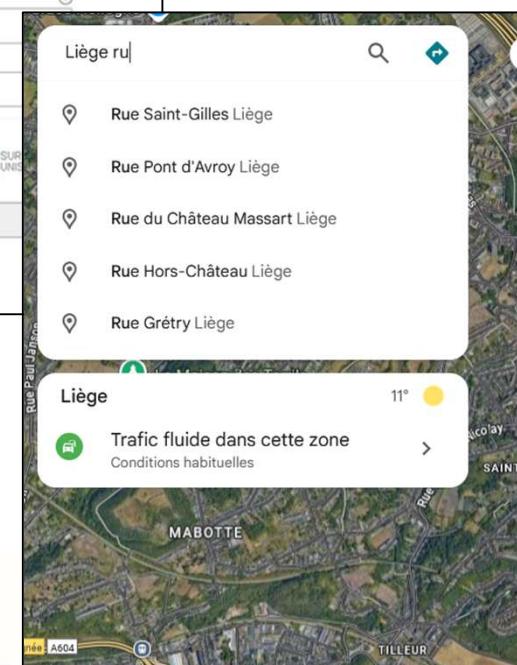
Votre voyage aux États-Unis est-il un transit vers un autre pays ?  Non

**RENSEIGNEMENTS SUR LE POINT DE CONTACT AUX ÉTATS-UNIS**

Nom *	Hotel CNEWYORK	Accueil Adresse Ligne 1 *	72nd Street
Accueil Adresse Ligne 2	Numero d'appartement	Ville *	New York
État *	Code pays + numéro *	MÊME CHOIX QUE LES RENSEIGNEMENTS CI-DESSUS SUR POINT DE CONTACT AUX ÉTATS-UNIS	
NEW YORK	UNITED STATES (USA) (+1)	<input checked="" type="checkbox"/>	

**ADRESSE AUX ÉTATS-UNIS**

Accueil Adresse Ligne 1	Accueil Adresse Ligne 2	Numéro d'appartement
72nd Street		
Ville	État	
New York	NEW YORK	





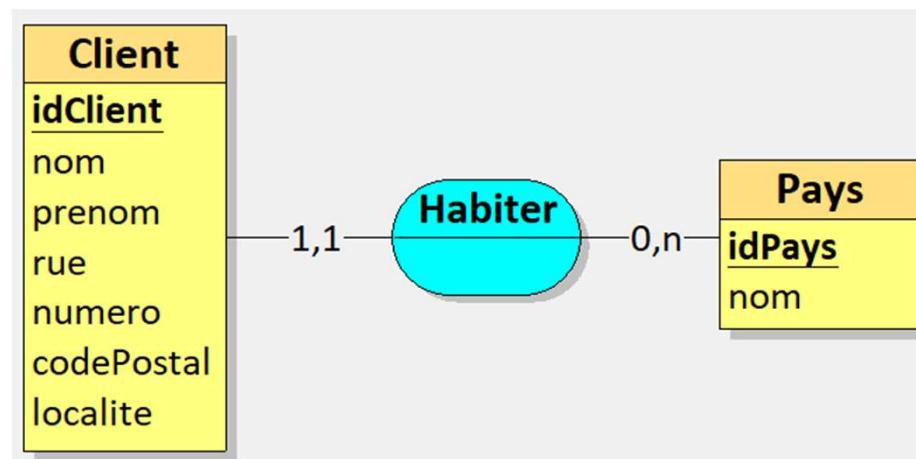
# Le cas de l'adresse...

Client
<u>idClient</u>
nom
prenom
rue
numero
codePostal
localite
pays

## Modélisation basique :

convient pour une majorité de situations proposées dans le cadre de ce cours.

Le système ne connaît que les pays, localités, rues... des clients encodés.

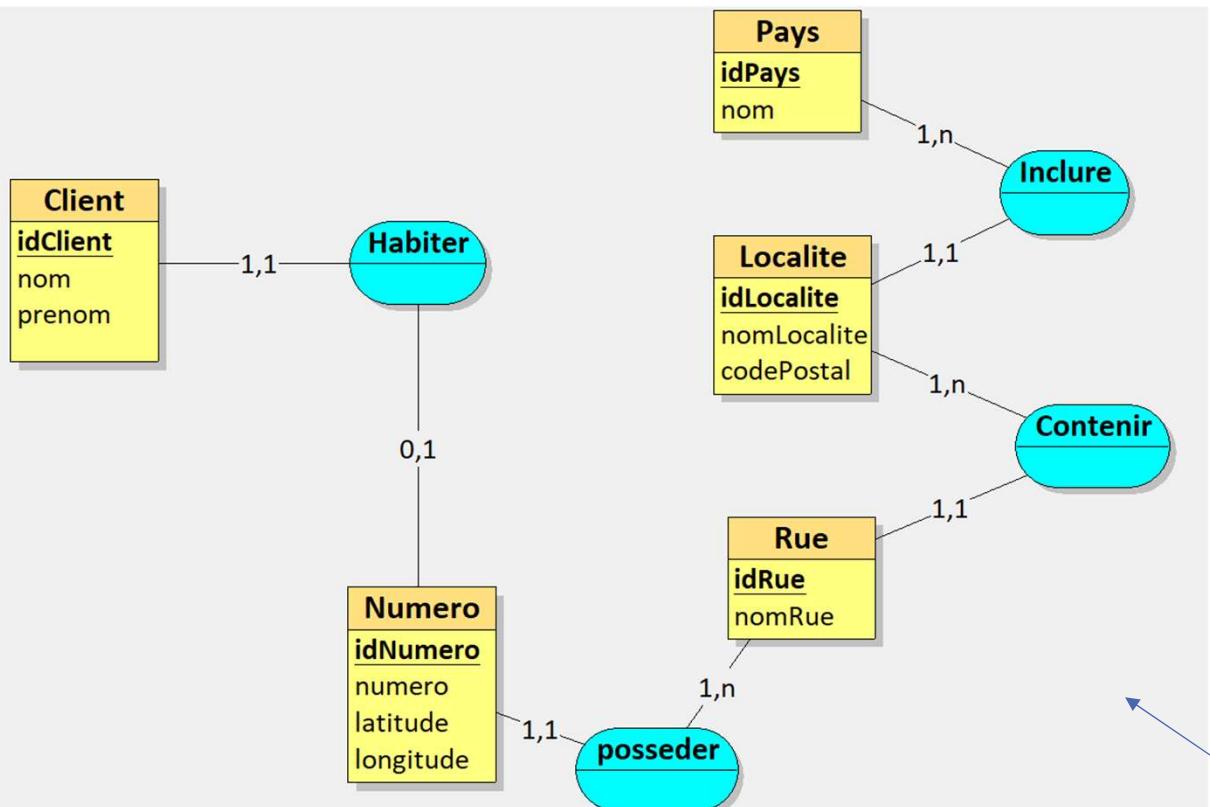


## Modélisation avancée :

la liste des pays du monde étant très stable, cette modélisation nécessite peu d'adaptation des données (l'entité Pays contient une occurrence par pays du monde). Cela permet de proposer une liste déroulante de pays au niveau de l'UI.

Cette pratique peut être étendue par exemple aux provinces, districts, états...

# Le cas de l'adresse...



**Modélisation pour système spécialisé :**  
 Le système a connaissance de toutes les adresses officielles possibles du pays, continent, monde.... (suivant la cible).  
 Possibilité de faire des suggestions ou de valider une adresse mais travail important de mise à jour des données.  
 Que se passe-t-il si une rue n'est pas connue par le système ? (nouvelle rue , changement de nom...)

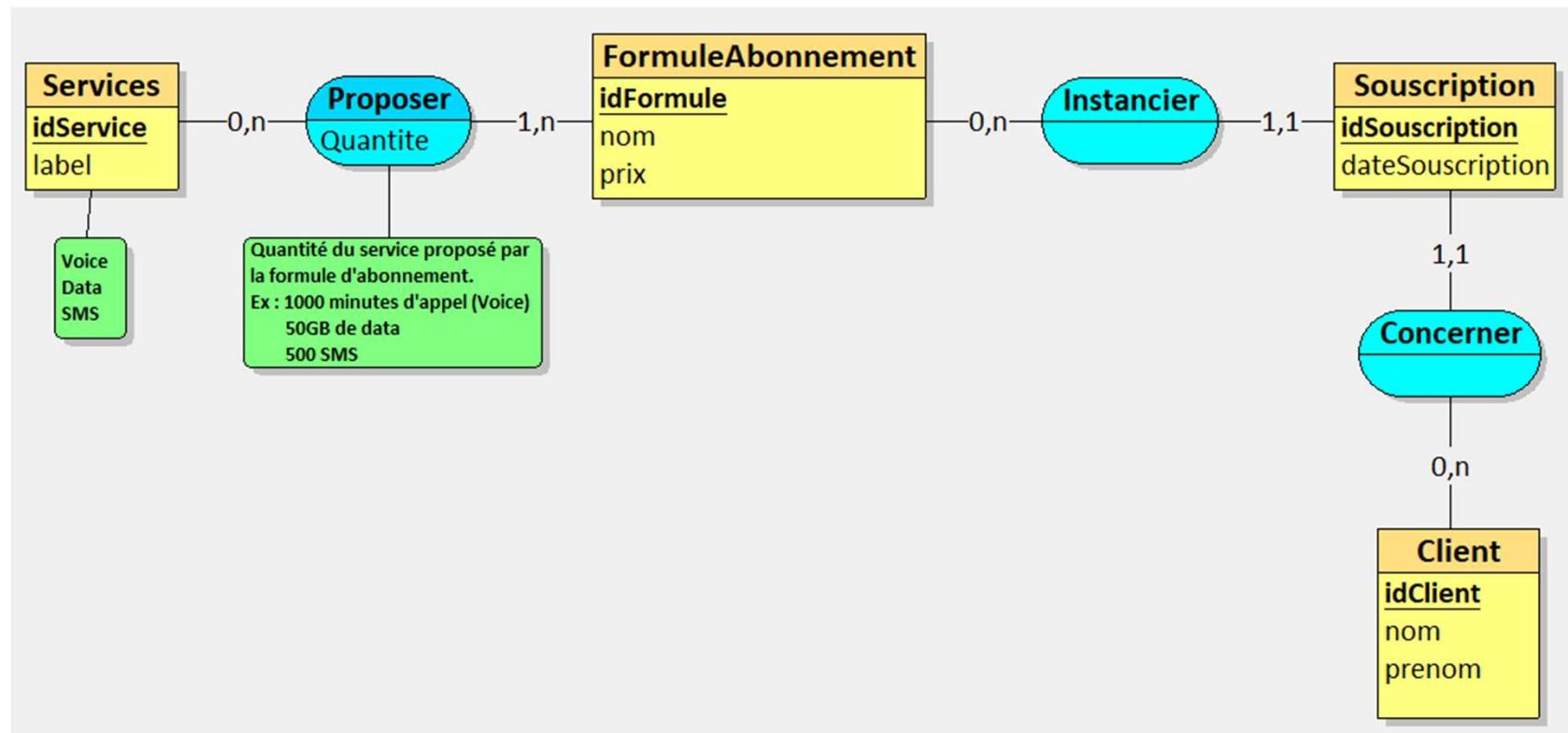
Exemple de modélisation simplifiée  
 Un pays inclut des localité  
 Un localité contient des rues  
 Chaque rue possède des numéros de bâtiment.

# Type / Instance

- Certaines entités représentent un type abstrait qui peut être concrétisé par une occurrence d'un autre entité (instance).
- **Discours client :**
  - « *Nous avons plusieurs exemplaires du même livre et pour chaque exemplaire, on doit connaître sa date d'achat.* »
  - « *Notre société propose des abonnements incluant des services différents (Voice, data, SMS...). Le client souscrit à l'abonnement qui lui convient le mieux suivant son profil d'utilisation.* »



# Pattern Type/Instance

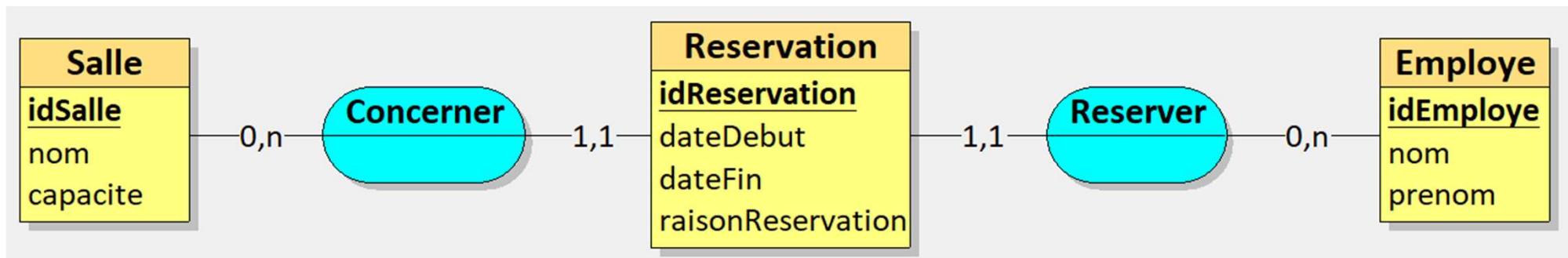


# Horaires, durées et planning

- Le système doit gérer des plages de temps (horaire de cours, agenda de rendez-vous, disponibilités). Il faut donc représenter des intervalles de temps (début, fin) ou (début, durée).
- **Discours client :**
  - « *Je dois pouvoir consulter mon agenda et voir quels sont mes rendez-vous de la journée.* »
  - « *Les salles peuvent être réservées pour des créneaux d'un heure.* »
  - « *Un médecin peut recevoir des patients, mais il ne faut pas que deux rendez-vous se chevauchent.* »
  - « *L'avocat reçoit ses clients sur rendez-vous le lundi de 13h à 16h et le mardi de 10h à 12h et de 14h à 17h.* »



# Horaires et planning



Comment s'assurer qu'il n'y a pas plusieurs réservations pour la même période ?

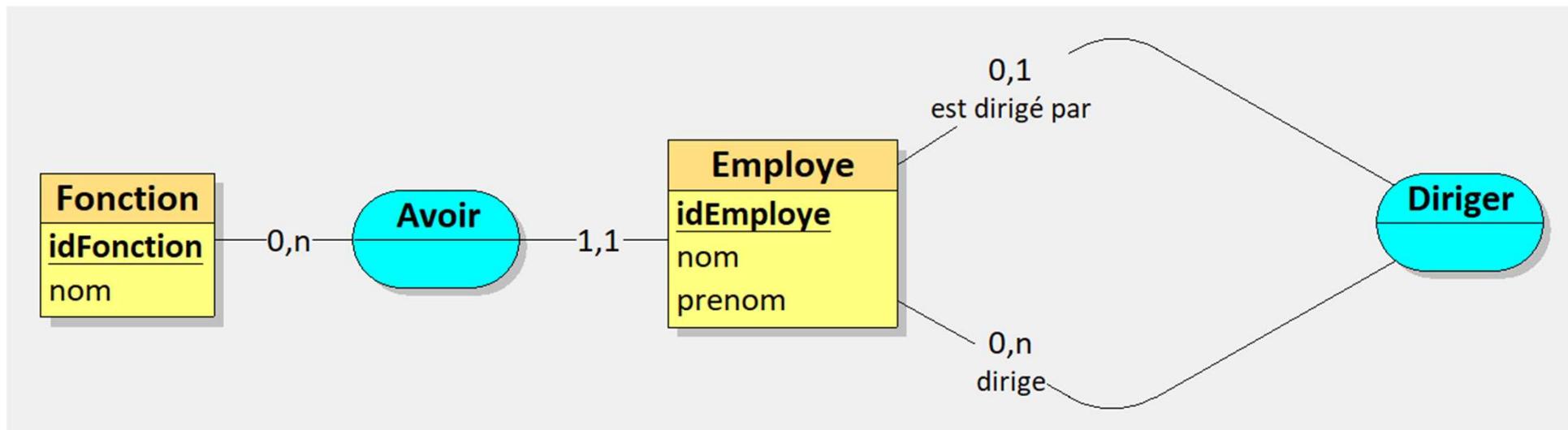
Comment faire si je veux pouvoir spécifier un créneau temporel sans avoir obligatoirement une fin ? (ex : CDI)

# Hiérarchisation

- Le système doit pouvoir gérer un lien hiérarchique qui existe entre plusieurs occurrences.
- **Discours client :**
  - « *Chaque employé a un manager, sauf le directeur.* »
  - « *Les catégories d'un site e-commerce sont organisées en sous-catégories.* »
  - « *Les dossiers d'un système peuvent contenir d'autres dossiers.* »



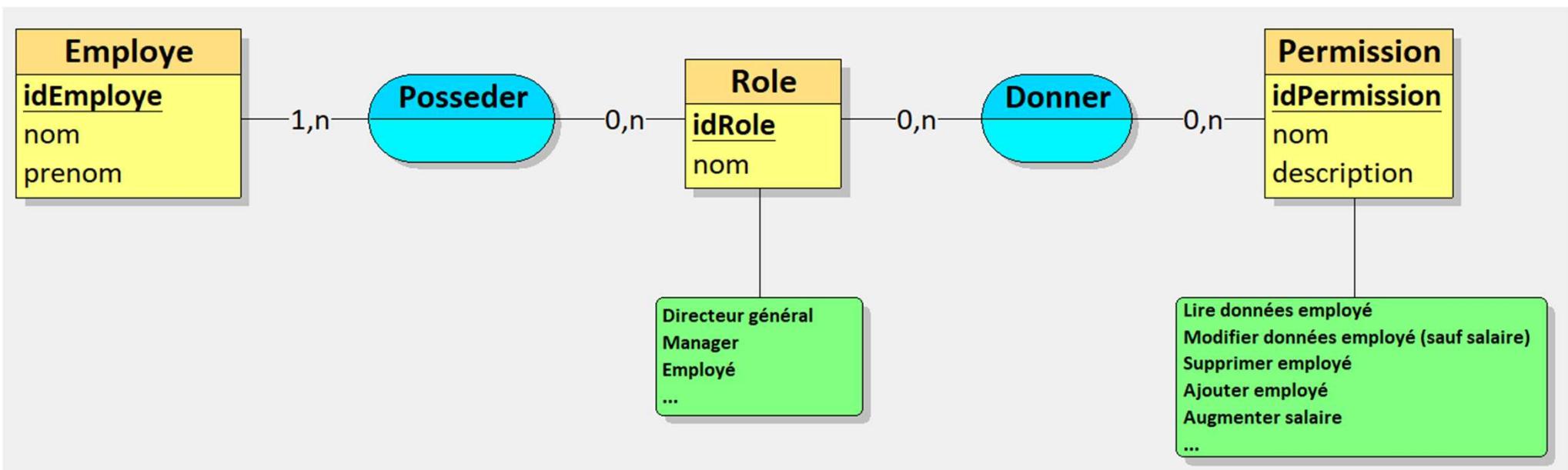
# Hiérarchisation



# RBAC (Role Based Access Control)

- Un système doit pouvoir gérer de manière flexible les droits d'accès des utilisateurs en fonction de leurs rôles. Chaque utilisateur peut avoir un ou plusieurs rôles et chaque rôle donne accès à un ou plusieurs droits (ou permissions).
- **Discours client :**
  - « *Dans un premier temps, un employé doit pouvoir consulter (en lecture) la liste des employés de son département alors que le manager du département doit pouvoir modifier les informations concernant les employés ainsi qu'ajouter et supprimer un employé. Il faut que je puisse facilement modifier cette configuration.* »

# RBAC



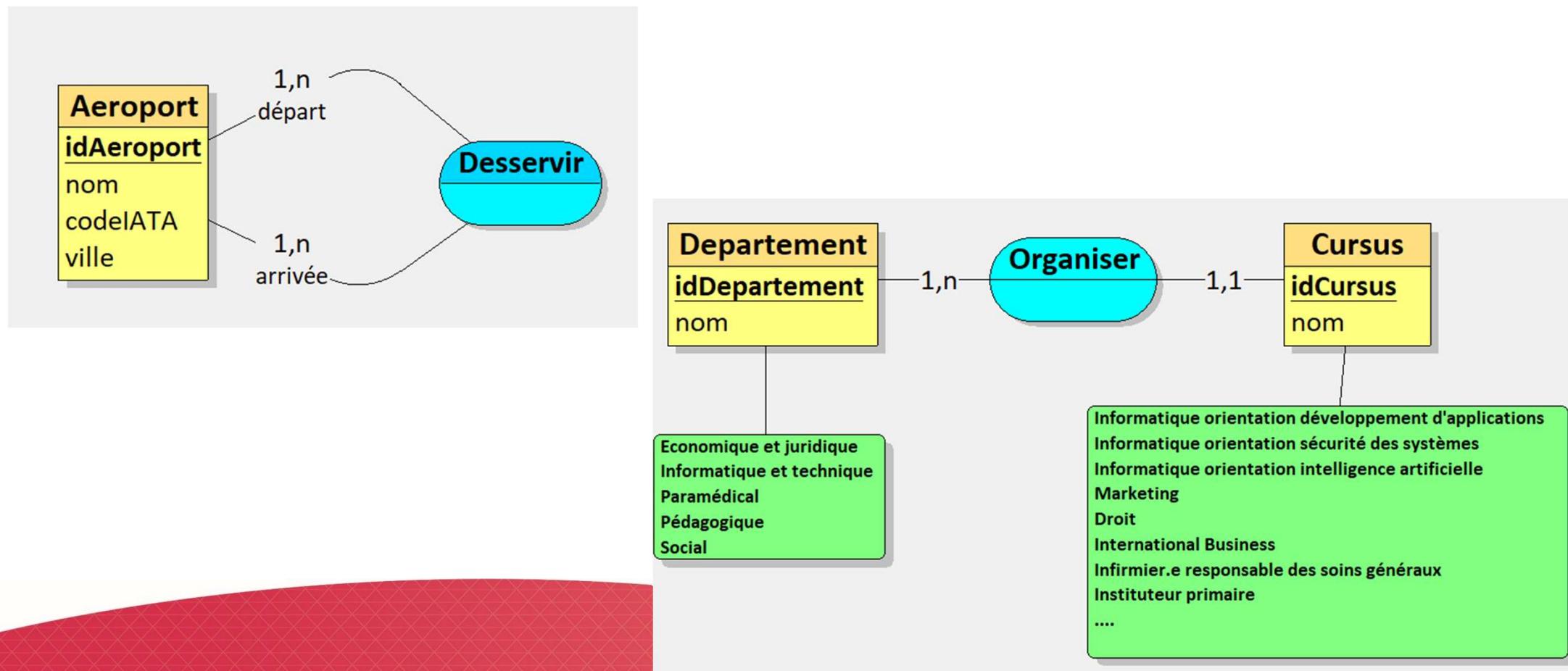
# Données dépendantes

- Un système doit connaître les combinaisons cohérentes / autorisées de données.
- **Discours client**
  - « Les destinations possibles à partir d'un aéroport sont limitées. Il faudra qu'en fonction de l'aéroport de départ, le système ne propose que les destinations possibles ».
  - « Chaque département de la haute-école propose des cursus. »

Province	Ville
Hainaut	ATH LA LOUVIERE MONS MOUSCRON THUIN



# Données dépendantes



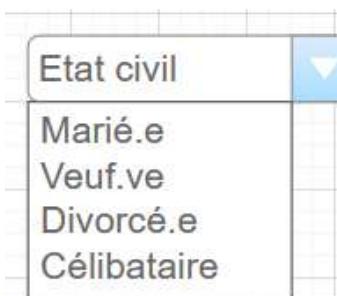
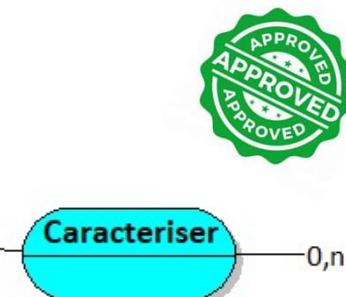
# Les anti-patterns (a.k.a. le musée des erreurs)

- A éviter !
- Inventaire des erreurs fréquemment reproduites par les étudiants
- Fortement pénalisées lors de l'évaluation !



# Le musée des erreurs : Propriété vs. Entité

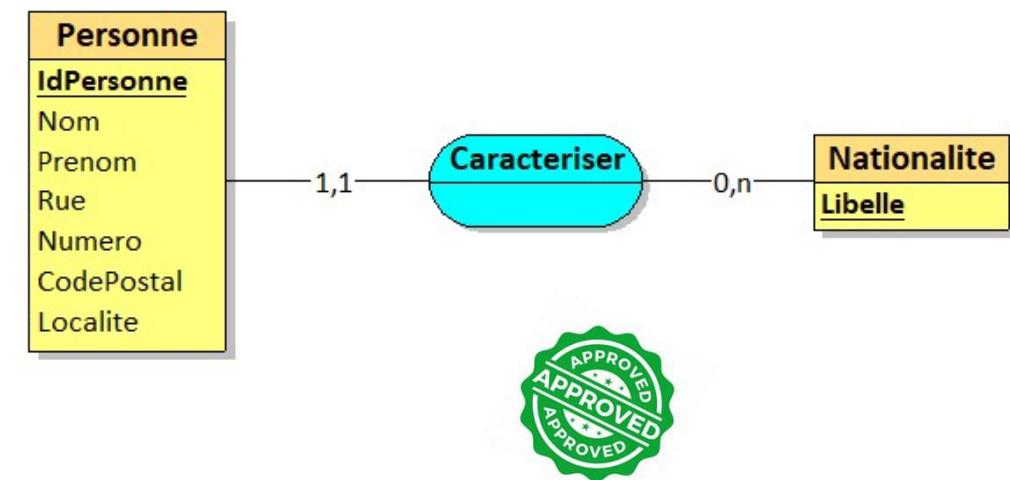
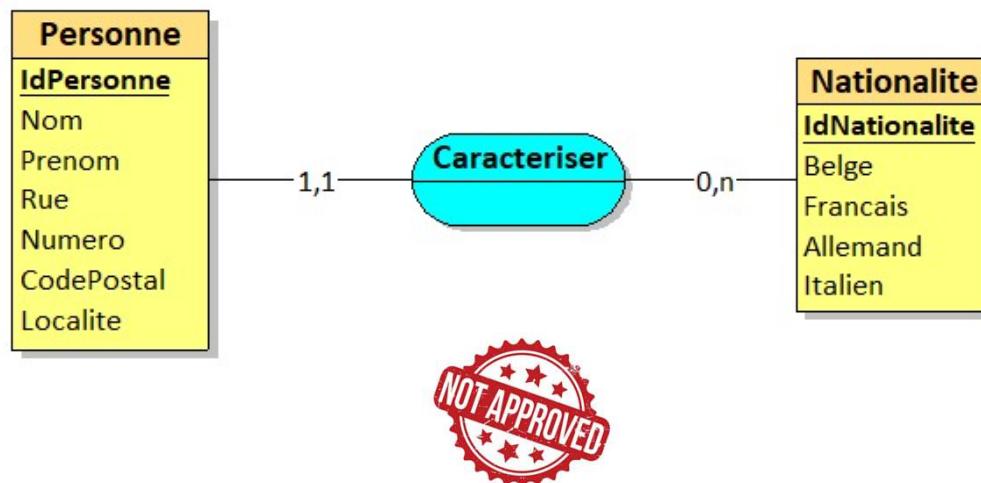
- Quand une information doit-elle être modélisée sous forme d'une **propriété ou d'une entité** ?
  - cohérence : si le système d'information doit connaître toutes les valeurs possibles d'une propriété (on peut établir une liste des valeurs)
  - si on veut « piloter l'UI à partir de la BD » (proposer des listes déroulantes)



# Le musée des erreurs : Propriétés ≠ Valeurs

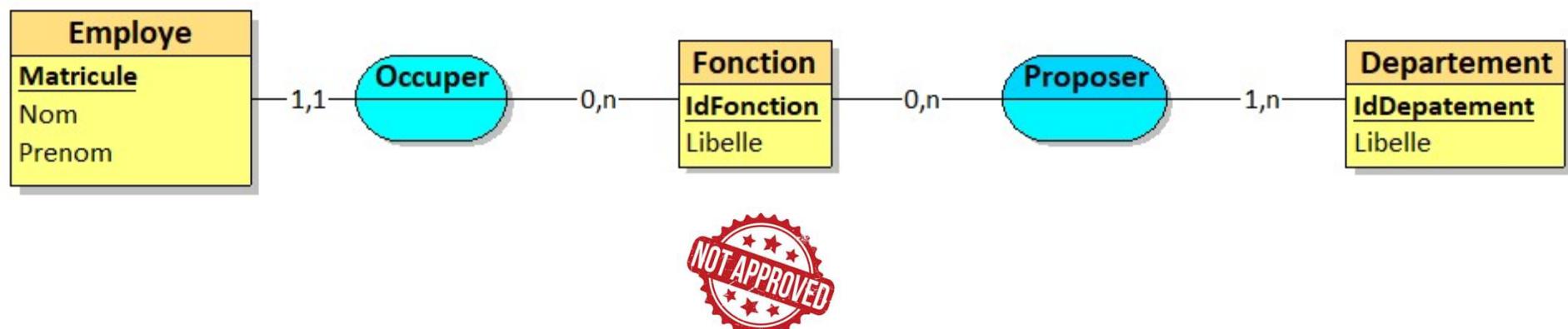


- Les propriétés d'une entité ne représentent pas les valeurs que pourront prendre ces propriétés pour chaque occurrence



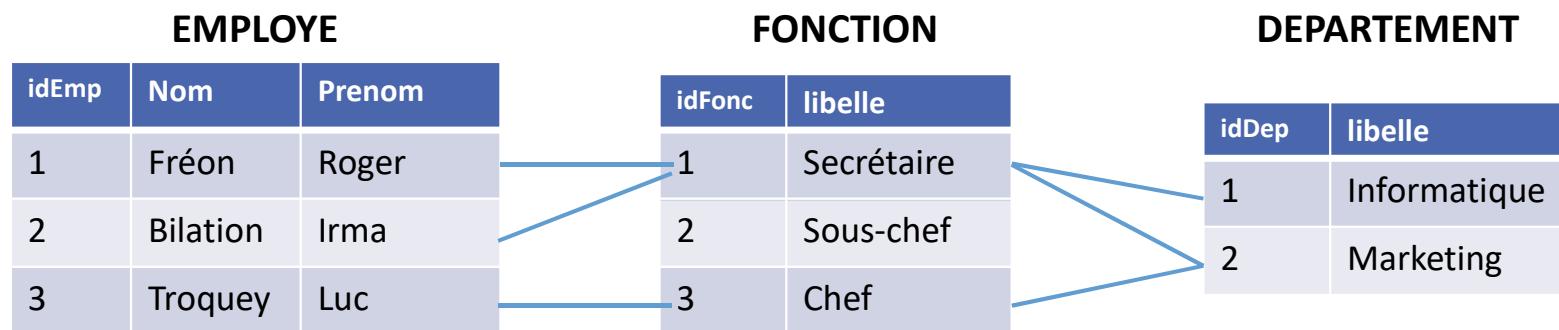
# Le musée des erreurs : le chemin perdu

- Certaines « chaînes » de liens entre occurrences peuvent être perdues à cause de cardinalités mal-choisies



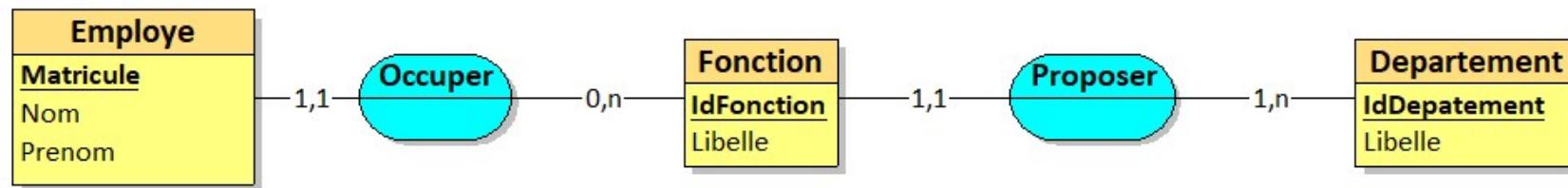
Quel est le problème ?

# Le musée des erreurs : le chemin perdu



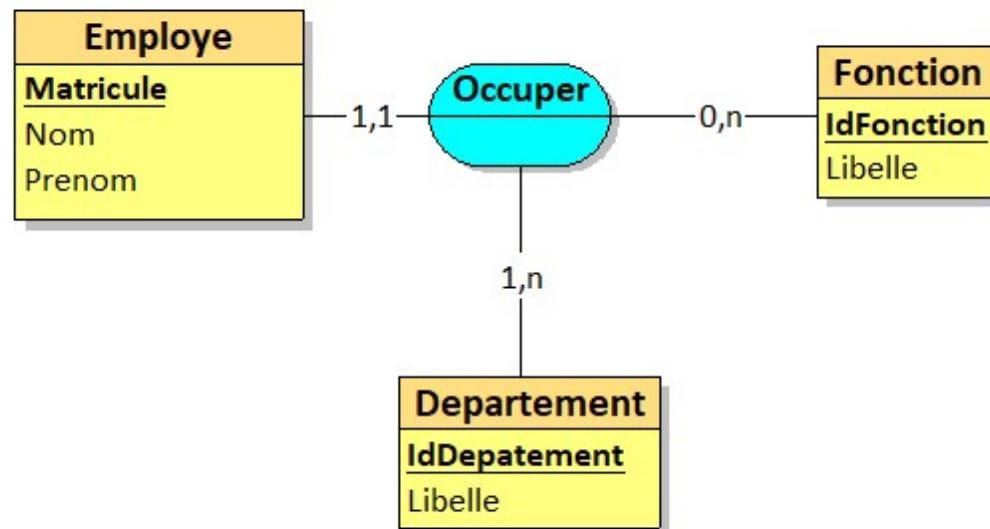
Roger et Irma sont « Secrétaire »  
mais on ne sait pas pour quel  
département.

# Le musée des erreurs : le chemin perdu



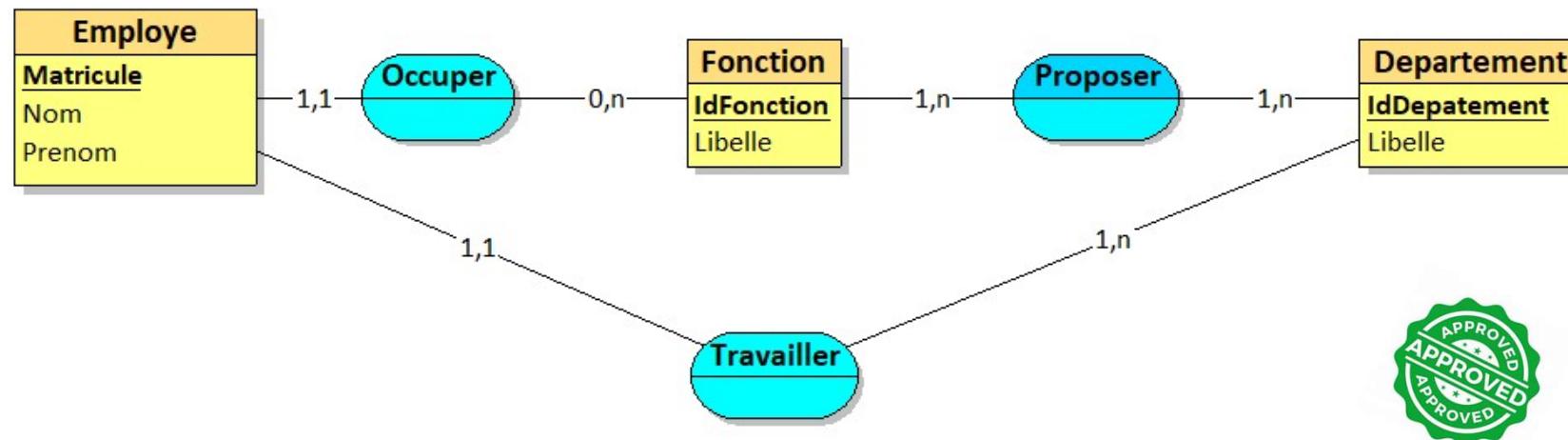
**Faiblesse :** une fonction est liée à un et un seul département → répétition des mêmes fonctions par département (il y a probablement un.e secrétaire dans tous les départements).

# Le musée des erreurs : le chemin perdu

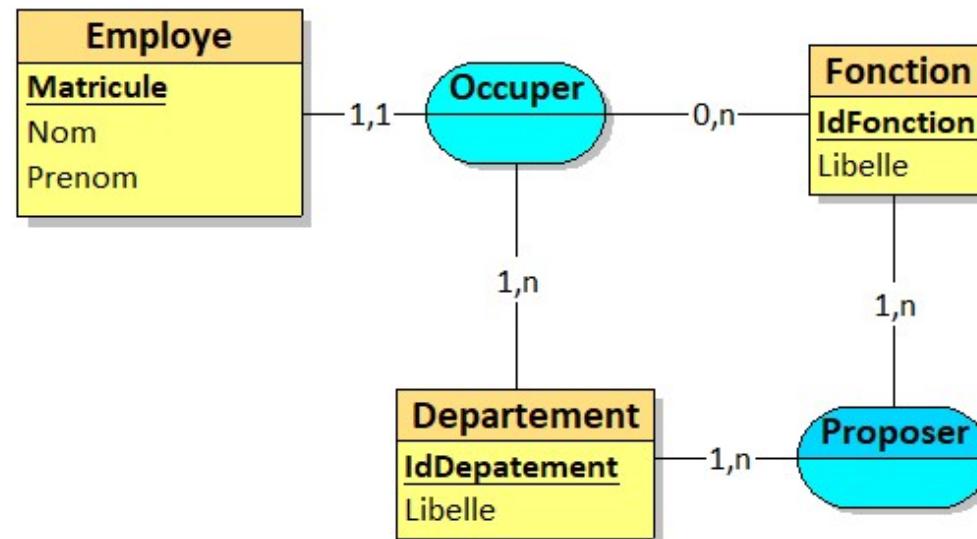


**Faiblesse :** on perd les données de référence qui listent les fonctions possibles par département (il n'y a pas de développeur au département comptabilité).

# Le musée des erreurs : le chemin perdu



# Le musée des erreurs : le chemin perdu

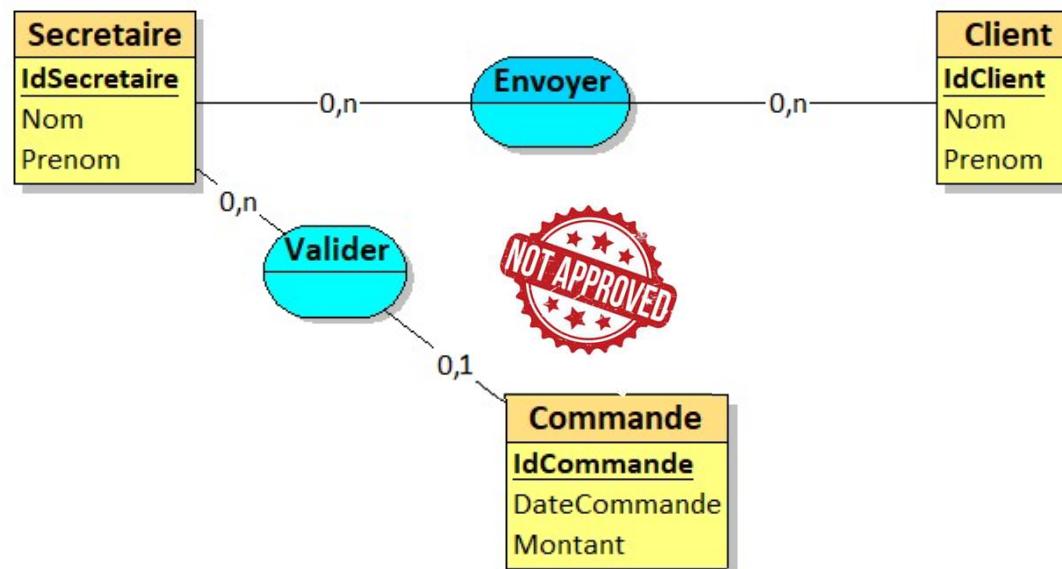


# Le musée des erreurs : Statique vs. dynamique

- Le MCD est un **modèle statique** : on cherche à modéliser les **données** qui seront nécessaires à la réalisation des tâches supportées par le S.I.
- On ne veut généralement pas modéliser « qui a fait quoi » dans un MCD
  - Sauf si pour des raisons d'audit, on doit pouvoir dire qui a effectué une tâche dans le système (mais cette situation sera généralement modélisée dans des tables d'audit spécialisées)
- Les diagrammes d'activité permettront de modéliser ces tâches : comment elles sont réalisées, par qui et quand.

# Le musée des erreurs : Statique vs. dynamique

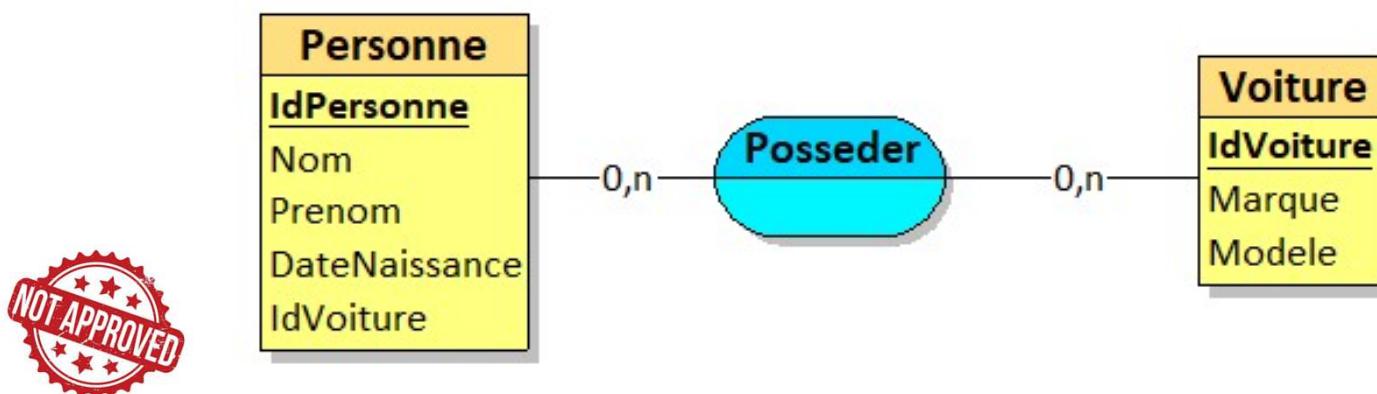
« Lorsqu'il a validé la commande, le secrétaire envoie un email de confirmation au client »



Sauf si on doit absolument tracer les actions effectuées.

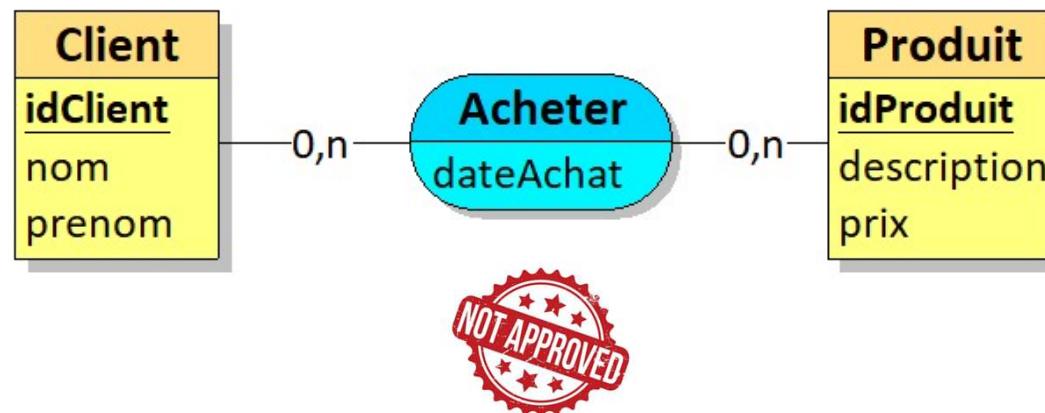
# Le musée des erreurs : Conceptuel vs. Logique

- Un Modèle Conceptuel de Données ne contient pas de « clé primaire », « attributs » ni de « clé étrangère » (voir cours Base de données)



# Le musée des erreurs : unicité des associations d'occurrences

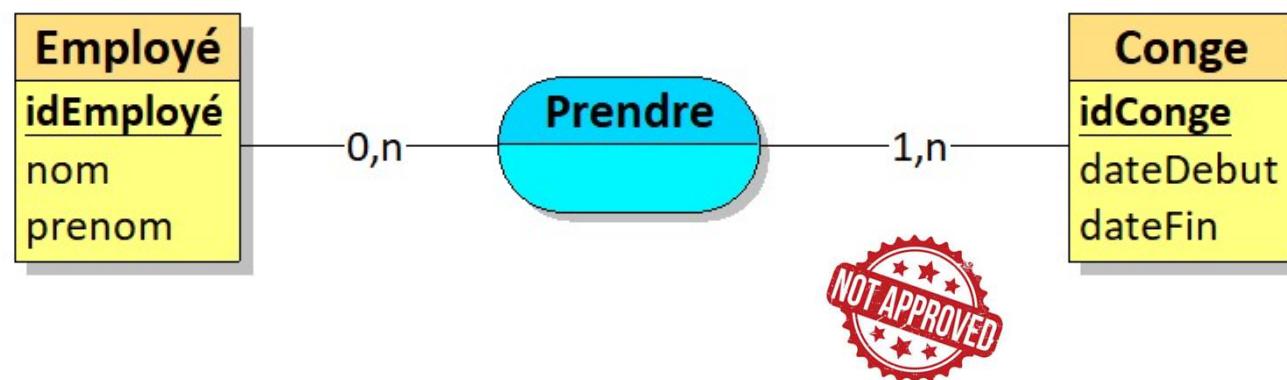
- Les associations ont un identifiant : il s'agit de la combinaison des identifiants des entités qu'elles relient. Comme l'identifiant se doit de respecter la contrainte d'unicité, cela peut entraîner des restrictions qui ne sont pas en phase avec les règles métier.



# e Ecole Mosane

## Le musée des erreurs : partage contrignant

- L'association d'une même occurrence d'une entité avec plusieurs occurrences d'une autre entité peut entraîner une complexité de gestion si l'association est susceptible d'évoluer.



## Suppression ou modification d'un congé pour un employé

## Ajout d'un congé pour un employé