

Modélisation des données

BD-BioInfo

Bases de données

Cristina Sirangelo

IRIF, Université Paris Diderot

cristina@irif.fr

Introduction à la modélisation

Modéliser plusieurs concepts

Rappel : la table Films

Films

| <i>titre</i> | <i>annee</i> | <i>realisateur</i> |
|---------------------|---------------------|---------------------------|
| Alien | 1979 | Scott |
| Vertigo | 1958 | Hitchcock |
| Psychose | 1960 | Hitchcock |
| Kagemusha | 1980 | Kurosawa |
| Volte-face | 1997 | Woo |
| Pulp Fiction | 1995 | Tarantino |
| Titanic | 1997 | Cameron |
| Sacrifice | 1986 | Tarkovski |

Et si on voulait représenter plusieurs informations sur les réalisateurs (nom prénom, date de naissance, ...) ?

Modéliser plusieurs concepts

Pourquoi pas tout mettre dans la même table?

Films

| <i>titre</i> | <i>annee</i> | <i>realisateur</i> | <i>prenom</i> | <i>naissance</i> |
|--------------|--------------|--------------------|---------------|------------------|
| Alien | 1979 | Scott | Ridley | 1943 |
| Vertigo | 1958 | Hitchcock | Alfred | 1899 |
| Psychose | 1960 | Hitchcock | Alfred | 1899 |
| Kagemusha | 1980 | Kurosawa | Akira | 1910 |
| Volte-face | 1997 | Woo | John | 1946 |
| Pulp Fiction | 1995 | Tarantino | Quentin | |
| Titanic | 1997 | Cameron | James | 1954 |
| Sacrifice | 1986 | Tarkovski | Andrei | 1932 |

Problèmes avec la table simple

Redondance

Les informations sur les réalisateurs sont dupliquées pour chaque film qu'ils ont réalisé.

Anomalies d'insertion

Possibilité d'insérer des données incohérentes

exemple : le même réalisateur avec deux dates de naissance différentes

Anomalies de mise à jour

Si on a besoin de rectifier une erreur sur l'année de naissance, il faut penser à le faire pour tous les films. Sinon, la table contient des informations incohérentes...

Anomalies de suppression

La suppression d'un film de la table, entraîne la suppression des informations associées sur le réalisateur.

Si le réalisateur n'était présent que pour un seul film, la suppression de ce film entraîne la disparition de toutes les informations relatives au réalisateur.

Solutions

Utiliser plusieurs tables pour représenter les films et les réalisateurs indépendamment les uns des autres

insertions, mises-à-jour et suppressions indépendantes.

Identifier les films (et les réalisateurs) pour s'assurer qu'aucun doublon ne figure dans nos tables

Lier les films et les réalisateurs sans introduire de redondance d'information

Solution : ébauche

Films: 2 films ne peuvent avoir le même titre (supposons-le)

Réalisateurs : 2 réalisateurs peuvent avoir le même nom; on les distingue grâce à un identificateur (*id*)

Films

| <i>titre</i> | <i>annee</i> |
|--------------|--------------|
| Alien | 1979 |
| Vertigo | 1958 |
| Psychose | 1960 |
| Kagemusha | 1980 |
| Volte-face | 1997 |
| Pulp Fiction | 1995 |
| Titanic | 1997 |
| Sacrifice | 1986 |

Realisateurs

| <i>id</i> | <i>nom</i> | <i>prenom</i> | <i>naissanc</i> |
|-----------|------------|---------------|-----------------|
| 1 | Scott | Ridley | 1943 |
| 2 | Hitchcock | Alfred | 1899 |
| 3 | Kurosawa | Akira | 1910 |
| 4 | Woo | John | 1946 |
| 5 | Tarantino | Quentin | |
| 6 | Cameron | James | 1954 |
| 7 | Tarkovski | Andrei | 1932 |

Solution : ébauche

Il n'y a plus de redondance dans la base de données :

Les informations sur chaque réalisateur n'apparaissent qu'une fois

Plus d'incohérence en cas de mise à jour

Mais l'information :

“Qui est le réalisateur du film ?” a disparu

Solution

Ajout d'un attribut dans la table film : “realisateur”

avec une contrainte de clef étrangère $\text{Films}[\text{realisateur}] \subseteq \text{Realisateurs}[\text{id}]$

Toujours pas de redondance

Films

| <i>titre</i> | <i>annee</i> | <i>realisateu</i> |
|--------------|--------------|-------------------|
| Alien | 1979 | 1 |
| Vertigo | 1958 | 2 |
| Psychose | 1960 | 2 |
| Kagemusha | 1980 | 3 |
| Volte-face | 1997 | 4 |
| Pulp Fiction | 1995 | 5 |
| Titanic | 1997 | 6 |
| Sacrifice | 1986 | 7 |

Realisateurs

| <i>id</i> | <i>nom</i> | <i>prenom</i> | <i>naissanc</i> |
|-----------|------------|---------------|-----------------|
| 1 | Scott | Ridley | 1943 |
| 2 | Hitchcock | Alfred | 1899 |
| 3 | Kurosawa | Akira | 1910 |
| 4 | Woo | John | 1946 |
| 5 | Tarantino | Quentin | |
| 6 | Cameron | James | 1954 |
| 7 | Tarkovski | Andrei | 1932 |

Solution

Insertion

Les informations concernant un même réalisateur sont présentés une seule fois dans la base : pas possible des stocker des information incohérentes (e.g. deux dates de naissance différentes)

ou bien il s'agit d'un réalisateur different !

Mise à jour

Plus de redondance, donc une mise à jour ne risque pas d'introduire d'incohérence

Suppression

La suppression d'un film n'affecte pas le réalisateur

Modélisation

Schéma final avec ses contraintes (une clef primaire choisie pour chaque table, et les clefs étrangères)

Films

| | | |
|--------------|-------|-------------|
| titre | annee | realisateur |
|--------------|-------|-------------|

Realisateurs

| | | | |
|-----------|-----|--------|-----------|
| id | nom | prenom | naissance |
|-----------|-----|--------|-----------|

Aussi dénoté :

Films (titre, année, réalisateur)

Realisateurs (id, nom, prénom, naissance) **Films[realisateur] \subseteq Realisateurs[id]**

Remarque :

la modélisation ne concerne que le schéma de la base de données
pas les données (lignes)

Un schéma de base de données plus complexe

- On veut représenter:
 - ▶ Des films,
 - ▶ Les réalisateurs et les acteurs qui jouent,
 - ▶ Les pays où ces films ont été réalisés,
 - ▶ Des utilisateurs du site des films
- Permettre aux utilisateurs de noter les films

Solution : ébauche

- Les informations concernant les artistes et les réalisateurs seront vraisemblablement les mêmes (nom, prénom, année de naissance)
 - ▶ on les représente avec une unique table Artistes

| Artistes | | | |
|----------|-----|--------|-----------|
| id | nom | prenom | naissance |

- Avec les films on représente l'id de l'artiste qui en est le réalisateur

| Films | | |
|-------|-------|----------------|
| titre | annee | id_realisateur |

- Comment représenter les acteurs qui jouent dans un film sans redondance?
 - ▶ Pourquoi la solution adoptée pour le réalisateur (ajouter son id dans la table Films) n'est pas valable?

Solution : ébauche

- Un film a plusieurs acteurs, un attribut id_acteur peut représenter une seule valeur!
- Solution : une nouvelle table qui fait le “lien” entre films et acteurs
 - ▶ Seuls les identifiants dans cette table, pour éviter la redondance

Cast

| | |
|------------|-----------|
| titre_film | id_acteur |
|------------|-----------|

Films

| | | |
|-------|-------|----------------|
| titre | annee | id_realisateur |
|-------|-------|----------------|

Artistes

| | | | |
|----|-----|--------|-----------|
| id | nom | prenom | naissance |
|----|-----|--------|-----------|

- Et si on voulait représenter également les rôles des acteurs dans les film ?

Solution : ébauche

- Le rôle n'est pas associé au film ou à l'acteur, mais à la participation de l'acteur dans un film.
 - ▶ attribut de la table Cast

Cast

| | | |
|------------|----------|------|
| titre_film | id_acteu | rôle |
|------------|----------|------|

Films

| | | |
|-------|-------|----------------|
| titre | annee | id_realisateur |
|-------|-------|----------------|

Artistes

| | | | |
|----|-----|--------|---------|
| id | nom | prenom | naissan |
|----|-----|--------|---------|

Solution : ébauche

- Représentation des pays :

| Pays | | |
|------|-----|--------|
| code | nom | langue |

- Et des utilisateurs : chaque utilisateur a un pseudo, nom, prénom, mdp, mais également un pays
 - ▶ comment représenter le pays?

Solution : ébauche

- Représentation des pays :

| Pays | | |
|------|-----|--------|
| code | nom | langue |

- Et des utilisateurs : chaque utilisateur a un pseudo, nom, prénom, mdp, mais également un pays
 - ▶ comment représenter le pays?

| Utilisateurs | | | | | |
|--------------|-------|-----|--------|-----|-----------|
| pseudo | email | nom | prenom | mdp | code_pays |

- Comment représenter les notes que les utilisateurs donnent aux films?

Solution : ébauche

- Un utilisateur peut noter plusieurs films et un film peut être noté par plusieurs utilisateurs
 - ▶ \Rightarrow la note ne peut pas être un attribut du film, ni de l'utilisateur
- Solution : une nouvelle table qui fait le “lien” entre films et utilisateurs
 - ▶ Seuls les identifiants dans cette table, pour éviter la redondance
 - ▶ la note est un attribut additionnel de cette table

Notation

| | | |
|------------|--------|------|
| titre_film | pseudo | note |
|------------|--------|------|

Solution : schema complet

Films

| | | |
|-------|-------|----------------|
| titre | annee | id_realisateur |
|-------|-------|----------------|

Artistes

| | | | |
|----|-----|--------|-----------|
| id | nom | prenom | naissance |
|----|-----|--------|-----------|

Cast

| | |
|------------|-----------|
| titre_film | id_acteur |
|------------|-----------|

Pays

| | | |
|------|-----|--------|
| code | nom | langue |
|------|-----|--------|

Utilisateurs

| | | | | | |
|--------|-------|-----|--------|-----|-----------|
| pseudo | email | nom | prenom | mdp | code_pays |
|--------|-------|-----|--------|-----|-----------|

Notation

| | | |
|------------|--------|------|
| titre_film | pseudo | note |
|------------|--------|------|

Schéma relationnel : spécification des contraintes

- Une **clé primaire** est choisie pour chaque relation

Film (titre, année, id_realisateur)

Artiste (id, nom, prénom, naissance)

Utilisateur (pseudo, e-mail, nom, prénom, mdp, code_pays)

Pays (code, nom, langue)

Cast (titre_film, id_acteur)

Notation (titre_film, pseudo, note)

- Remarques :
 - clef primaire pour Cast : un film a plusieurs acteurs et un acteur peut jouer dans plusieurs films
 - clef primaire pour Notation : un film peut être noté par plusieurs utilisateurs et un utilisateurs peut noter plusieurs film, en revanche un utilisateur donne une seule note à un film

Schéma relationnel : spécification des contraintes

- Les attributs qui font références à d'autres tables génèrent des contraintes de **clé étrangère**

Film (titre, année, id_realisateur)

Film[id_realisateur] \subseteq Artiste[id]

Artiste (id, nom, prénom, naissance)

Utilisateur (pseudo, e-mail, nom, prénom, mdp, code_pays)

Utilisateur[code_pays] \subseteq Pays[code]

Pays (code, nom, langue)

Cast (titre_film, id_acteur)

Cast[titre_film] \subseteq Film[titre] Cast[id_acteur] \subseteq Artiste[id]

Notation (titre_film, pseudo, note)

Notation[titre_film] \subseteq Film[titre] Notation[pseudo] \subseteq Utilisateur[pseudo]

- D'autres contraintes (attributs requis, clef candidates,...) sont exprimés en phase d'implémentation

Implémentation du schéma relationnel

Le schéma relationnel est ensuite implémenté dans le SGBD avec une suite de commandes CREATE TABLE, après avoir choisi le type de chaque attribut

De plus :

- ▶ les **clefs candidates** génèrent des contraintes **UNIQUE**
- ▶ les **attributs requis** sont identifiés et génèrent des contraintes **NOT NULL**
- ▶ des **contraintes plus complexes** sont implémentées avec des conditions de **CHECK** et/ou des **assertions**

Implémentation du schéma relationnel

```
CREATE TABLE Artiste (  
    id INTEGER PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL,  
    prenom VARCHAR(50) NOT NULL,  
    naissance INTEGER,  
    UNIQUE (nom, prenom, naissance)  
);
```

Implémentation du schéma relationnel

```
CREATE TABLE Pays (  
    code CHAR(2) PRIMARY KEY,  
    nom VARCHAR(50) NOT NULL,  
    langue VARCHAR(30)  
);
```


Implémentation du schéma relationnel

```
CREATE TABLE Utilisateur (  
    pseudo VARCHAR(50) PRIMARY KEY,  
    email VARCHAR(50) NOT NULL UNIQUE,  
    nom VARCHAR(20) NOT NULL,  
    prenom VARCHAR(20),  
    mdp VARCHAR(60) NOT NULL,  
    naissance INTEGER,  
    code_pays CHAR(2) NOT NULL,  
    FOREIGN KEY (code_pays) REFERENCES Pays(code)  
);
```

Implémentation du schéma relationnel

```
CREATE TABLE Films (  
    titre VARCHAR(50) PRIMARY KEY,  
    annee INTEGER CHECK (annee BETWEEN 1930 AND 2016) NOT NULL,  
    genre VARCHAR(30)  
        CHECK (genre IN ('Documentaire', 'Action', 'Drame'));  
    id_realisateur INTEGER,  
    FOREIGN KEY (id_realisateur) REFERENCES Artiste (id)  
);
```

Implémentation du schéma relationnel

```
CREATE TABLE Notation (  
    titre_film VARCHAR(50),  
    pseudo VARCHAR(50),  
    note INTEGER NOT NULL,  
    PRIMARY KEY (titre_film, pseudo),  
    FOREIGN KEY (titre_film) REFERENCES Films(titre),  
    FOREIGN KEY (pseudo) REFERENCES Utilisateurs(pseudo),  
);
```

Similaire pour la table `Cast`

Modélisation

- Comment arriver à produire un schéma correct ? (tables, attributs, contraintes)

Film (titre, année, id_realisateur)

Film[id_realisateur] \subseteq Artiste[id]

Artiste (id, nom, prénom, naissance)

Utilisateur (pseudo, e-mail, nom, prénom, mdp, code_pays)

Utilisateur[code_pays] \subseteq Pays[code]

Pays (code, nom, langue)

Cast (titre_film, id_acteur)

Cast[titre_film] \subseteq Film[titre] Cast[id_acteur] \subseteq Artiste[id]

Notation (titre_film, pseudo, note)

Notation[titre_film] \subseteq Film[titre] Notation[pseudo] \subseteq Utilisateur[pseudo]

- Modélisation : Passage d'une spécification informelle des données à un schéma de BD qui les représente correctement
 - ▶ une procédure qui donne plus de garantie que le "bon sens"

Modélisation conceptuelle

Modélisation conceptuelle

Pour simplifier le processus de modélisation en général on ne cherche pas à trouver les bonnes tables directement (comme fait dans l'exemple précédent)

On s'appuie sur des **modèles** dits “**Entités / Associations**” (ou E/R)

- modèle E/R Introduit par Chen :

- ▶ *Peter P.-S. Chen. The Entity-Relationship Model, Toward a Unified View of Data, ACM Transactions on Database Systems (TODS) 1:(1), 1976*

- La plupart des modèles conceptuels utilisés encore aujourd'hui dans l'industrie s'inspirent du modèle E/R

E/R est à la base de méthodes de conception comme OMT (UML)

- plus haut-niveau que le modèle relationnel

Il existe ensuite des règles qui nous permettent de traduire un diagramme E/R en un schéma relationnel, et définir donc les tables de la base

Modèle E/R (*Entity/Relationship* en anglais)

- Un formalisme pour la modélisation conceptuelle des données:
 - ▶ quelles données
 - ▶ comment elles sont reliées
 - ▶ quelle contraintes elle satisfont
- Plusieurs variantes dans la notation : on fixera une notation, beaucoup de notations alternatives existent !

Modélisation des données avec E/R

- Les données peuvent être modélisées comme :
 - ▶ un ensemble d'*entités*,
 - ▶ des *associations* entre ces entités

Entité

- **Entité** : une classe d'objets avec des propriétés communes
 - ▶ Exemples : les étudiants de Paris 7, les employés d'une entreprise, les spectacles dans Paris,...
- Les entités ont des **attributs** (i.e. leurs propriétés)
 - ▶ Exemple: les étudiants de Paris 7 ont un *nom*, un *numero étudiant*, une *adresse*
 - ▶ Les spectacles ont une *description*, un *genre*, un *lieu*, des *horaires*

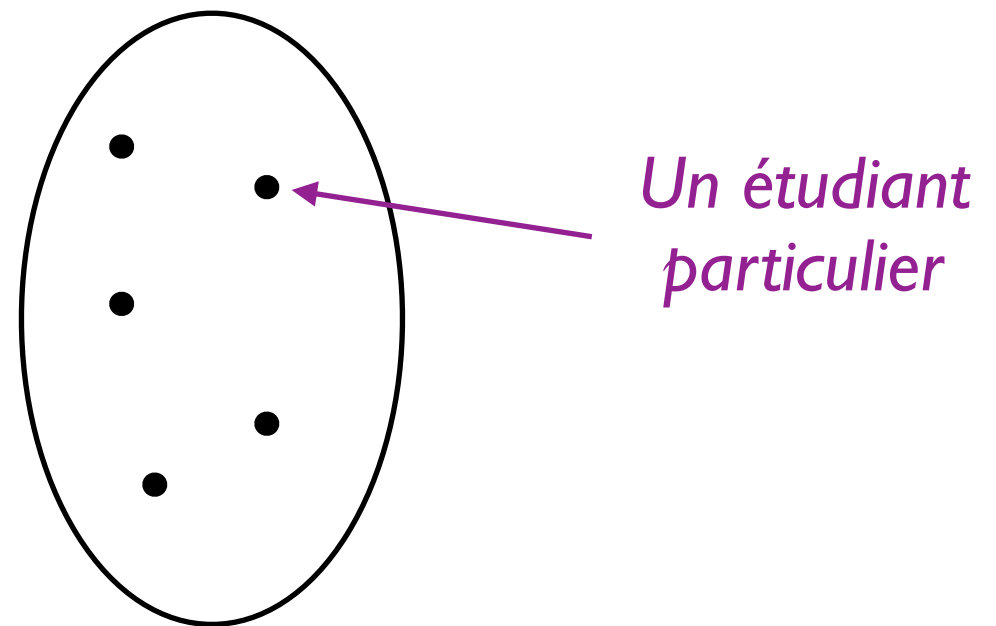
Entités et attributs

Représentation graphique :

| Étudiant |
|-----------------------------------|
| numero étudiant nom adresse |

Instances :

- Une **entité** dénote un ensemble d'objets, appelés **instances** de l'entité



instances de **Étudiant**

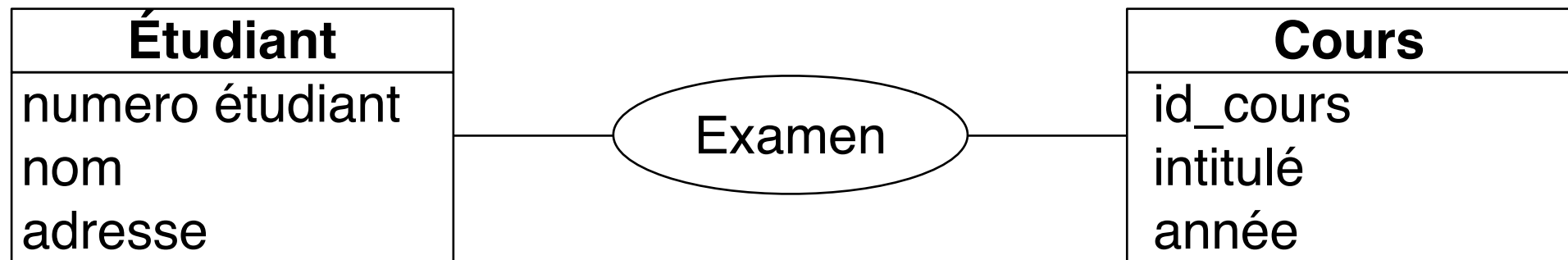
- Un **attribut** décrit une propriété des instances d'une entité
 - Ex. chaque étudiant a un **nom**

Associations

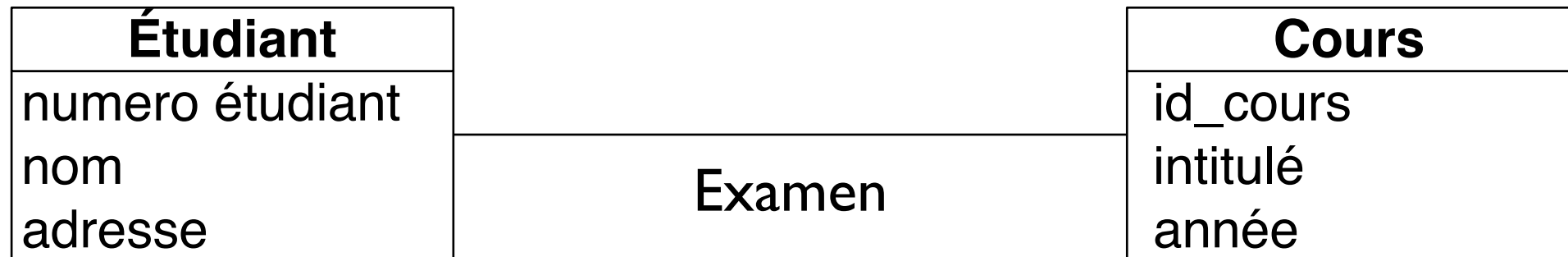
- **Association** : représente un lien entre deux ou plusieurs entités
- Exemples d'associations :
 - ▶ l'association **Examen** relie l'entité **Etudiant** à l'entité **Cours**
 - ▶ l'association **Affectation** relie l'entité **Employé** à l'entité **Département**

Associations

Représentation
graphique :



alternative :



Le nom de l'association peut être absent

Associations

Repr.
graphique :

| Étudiant |
|-----------------|
| numero étudiant |
| nom |
| adresse |

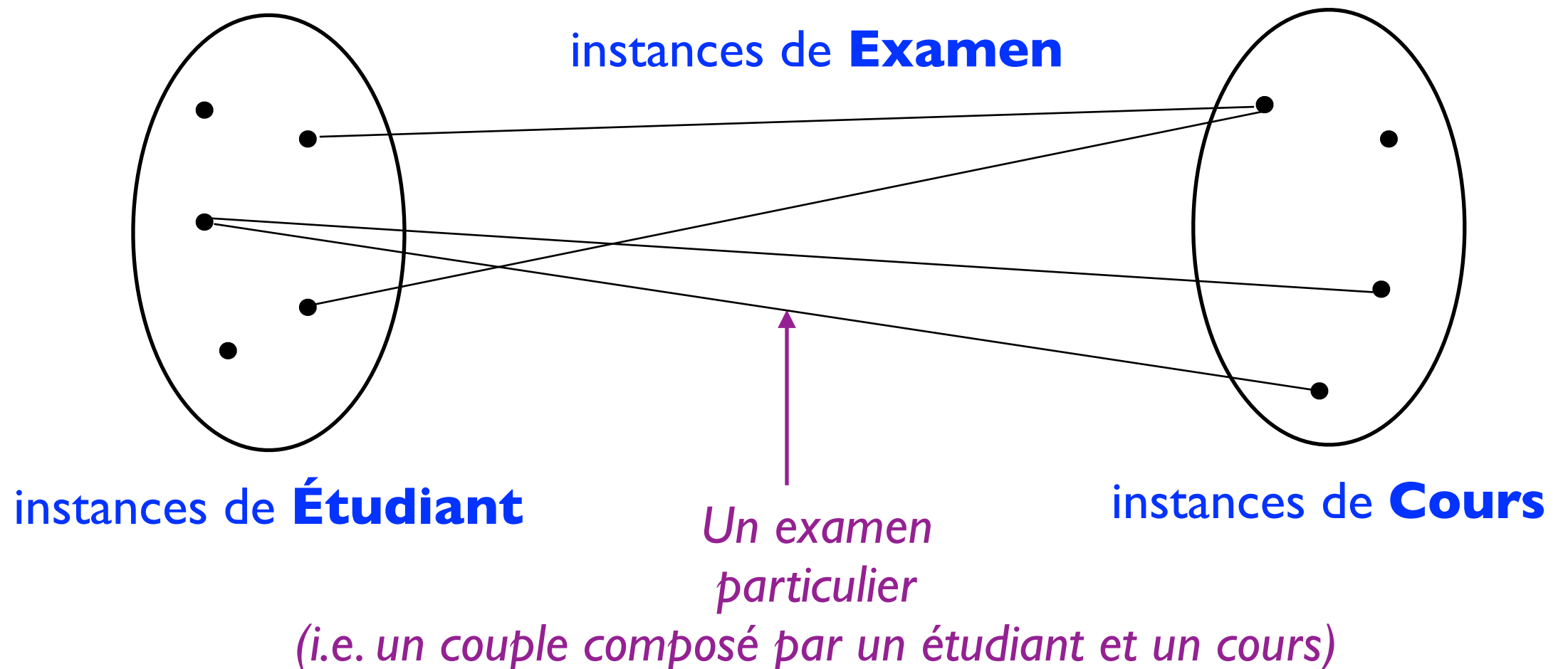
Examen

| Cours |
|----------|
| id_cours |
| intitulé |
| année |

Instances :

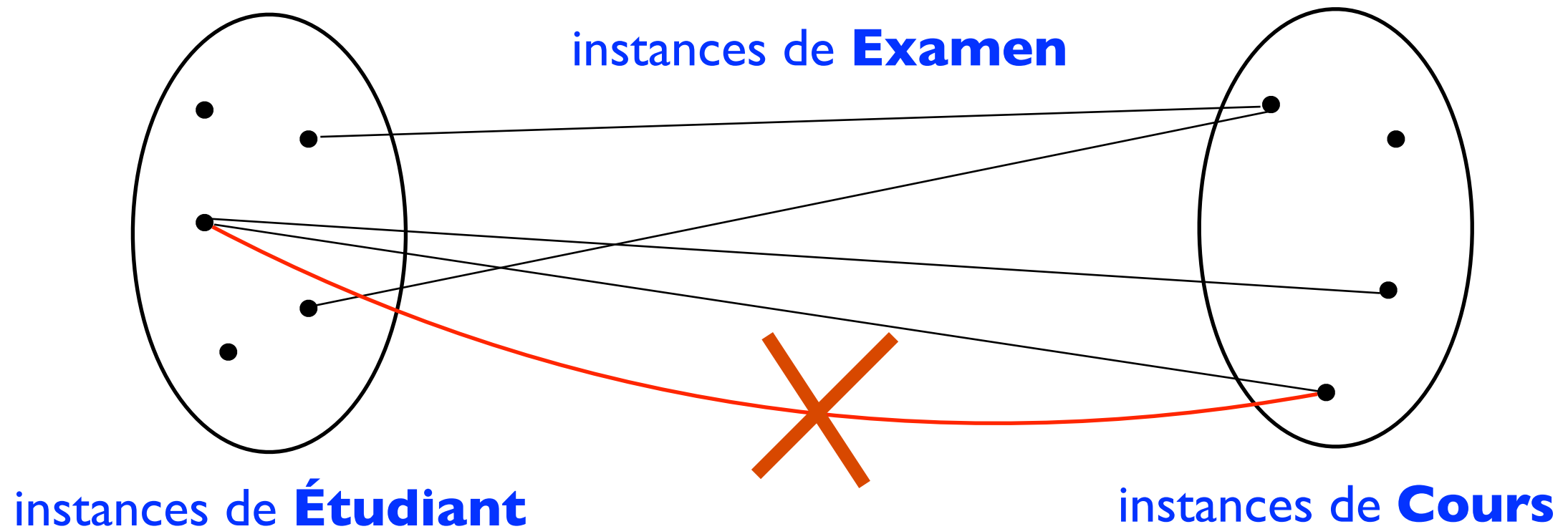
Un ensemble de couples.

Chaque couple est composé par un étudiant et un cours



Associations

- **Remarques sur les instances d'une association**



instances(**Examen**) est un **ensemble** de couples : pas de doublons

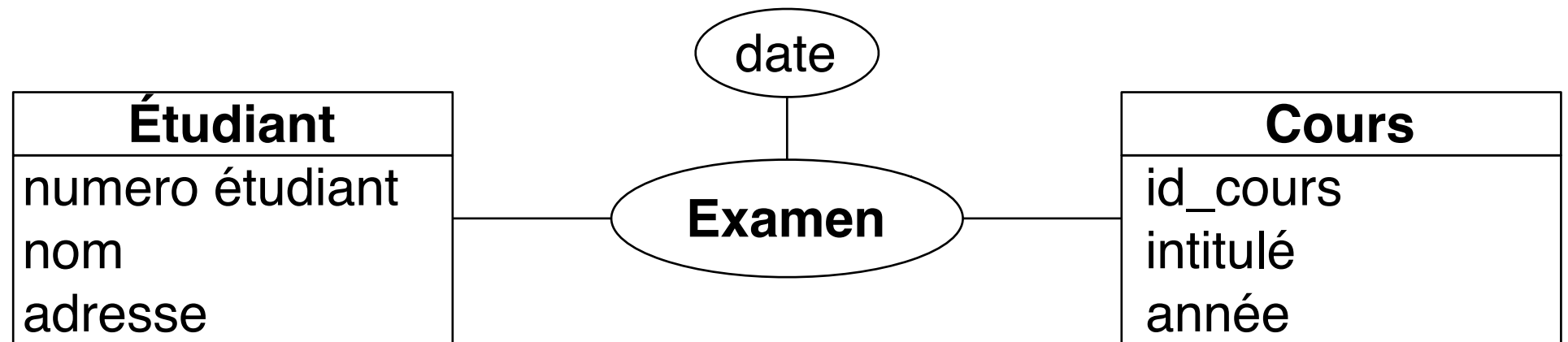
⇒ modéliser Examen comme association introduit automatiquement la contrainte suivante : **un étudiant ne peut pas passer l'examen d'un même cours plusieurs fois**

Si ce n'est pas le cas dans la réalité que nous modélisons (e.g. examens non réussis, à répéter...) il faut changer la modélisation

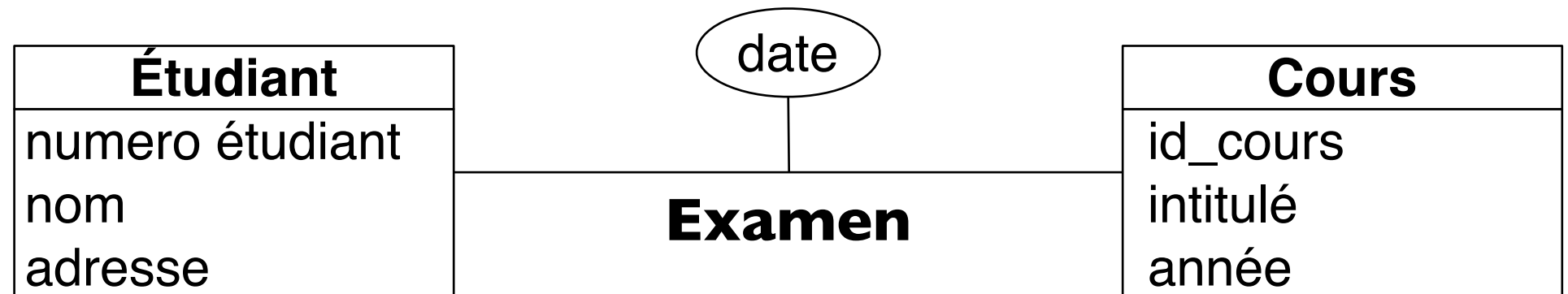
Attributs d'associations

- **Attribut d'une association** : décrit une propriété des instances de l'association
 - ▶ Exemple : chaque examen a une date

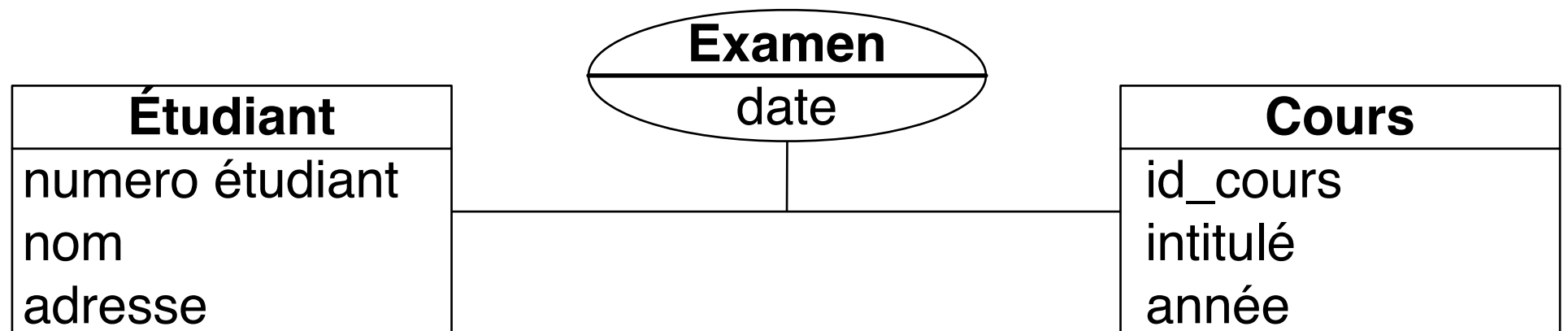
Représentation
graphique :



alternative :



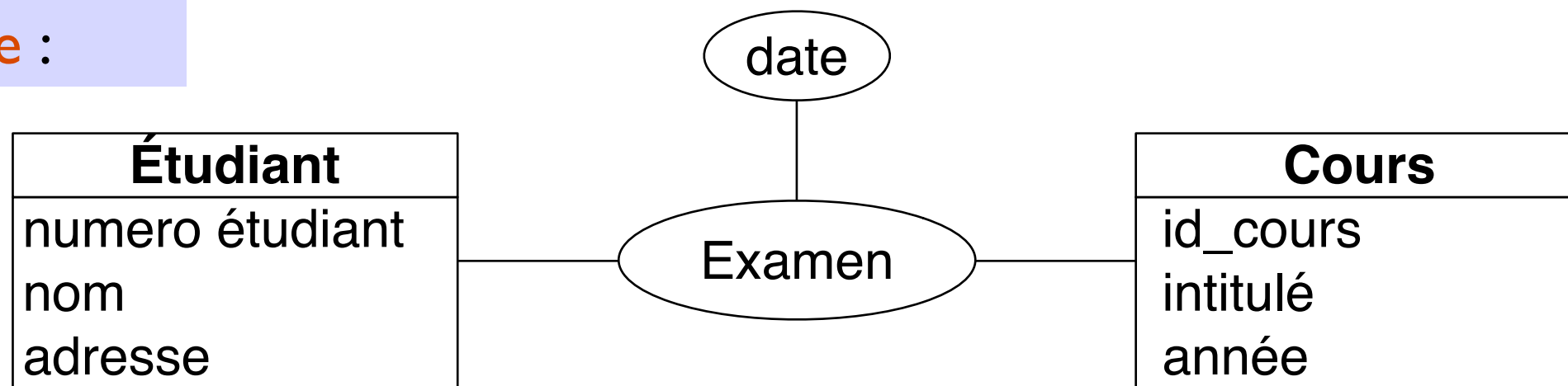
alternative :



Attributs d'associations

- **Attribut d'une association** : décrit une propriété des instances de l'association
 - ▶ Exemple : chaque examen a une date

Représentation
graphique :



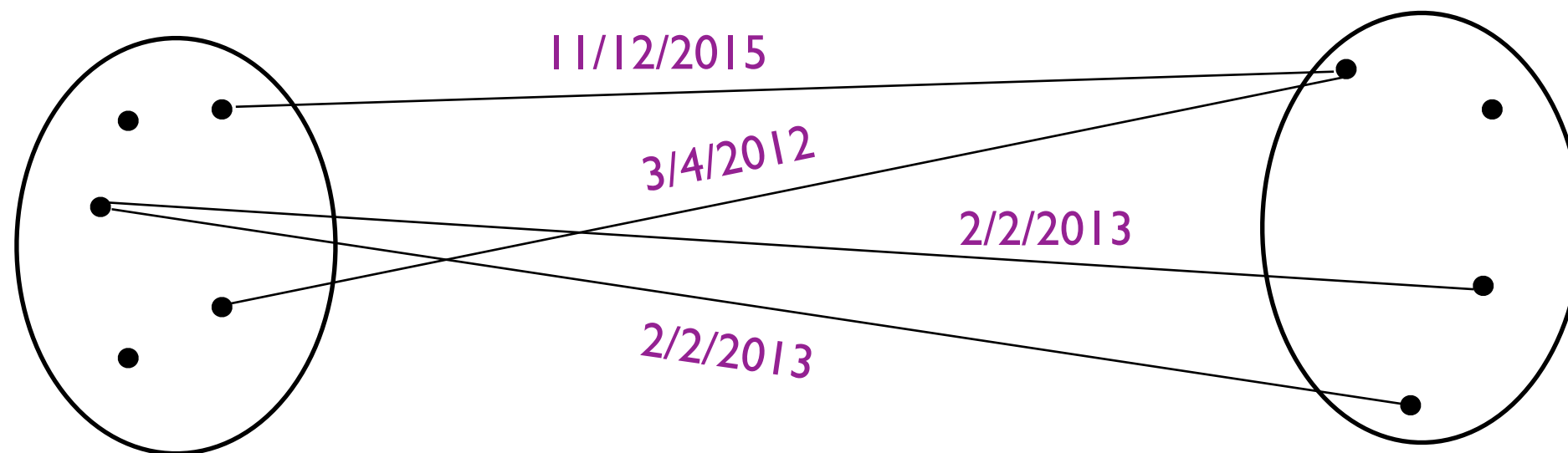
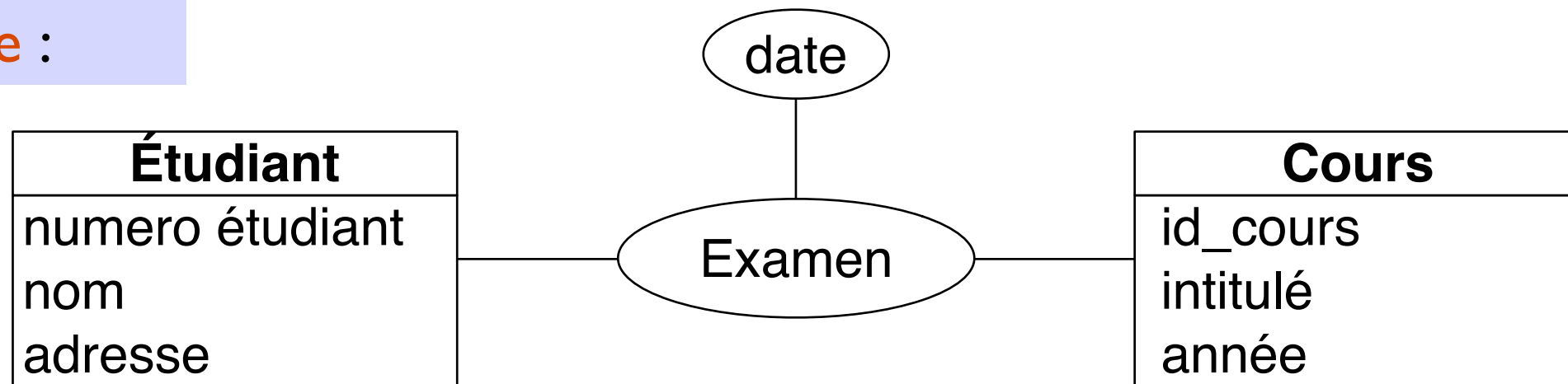
Remarque :

La date est une propriété des couples <étudiant, cours> qui représentent un examen

Attributs d'associations

- **Attribut d'une association** : décrit une propriété des instances de l'association
 - ▶ Exemple : chaque examen a une date

Représentation
graphique :

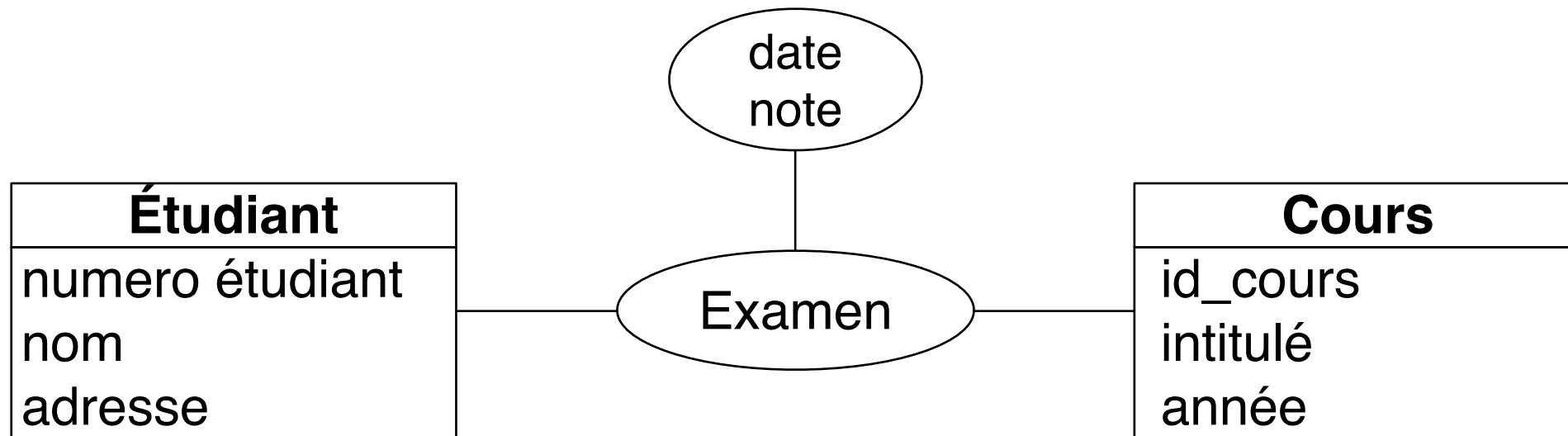


instances de **Étudiant**

instances de **Cours**

Attributs d'associations

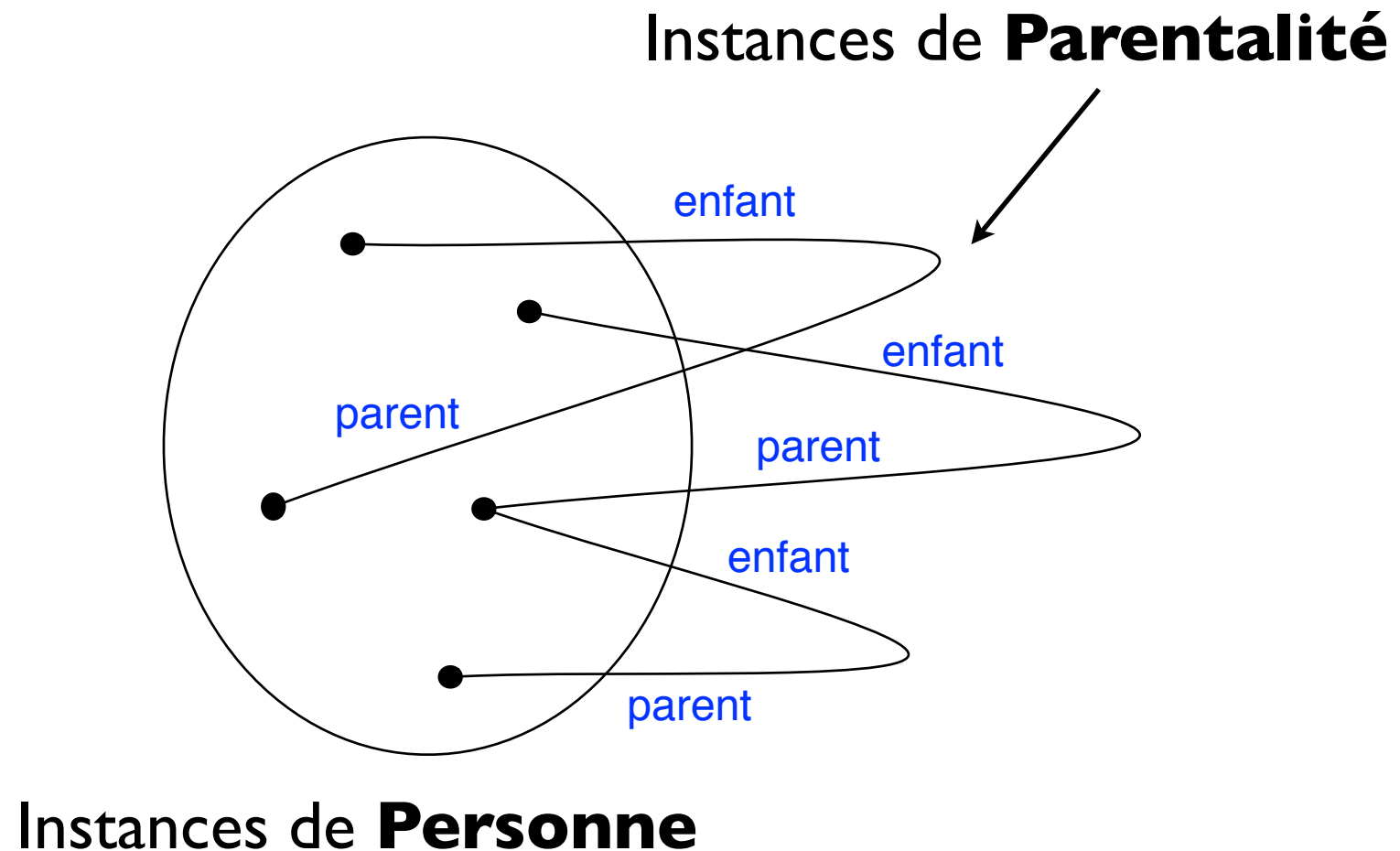
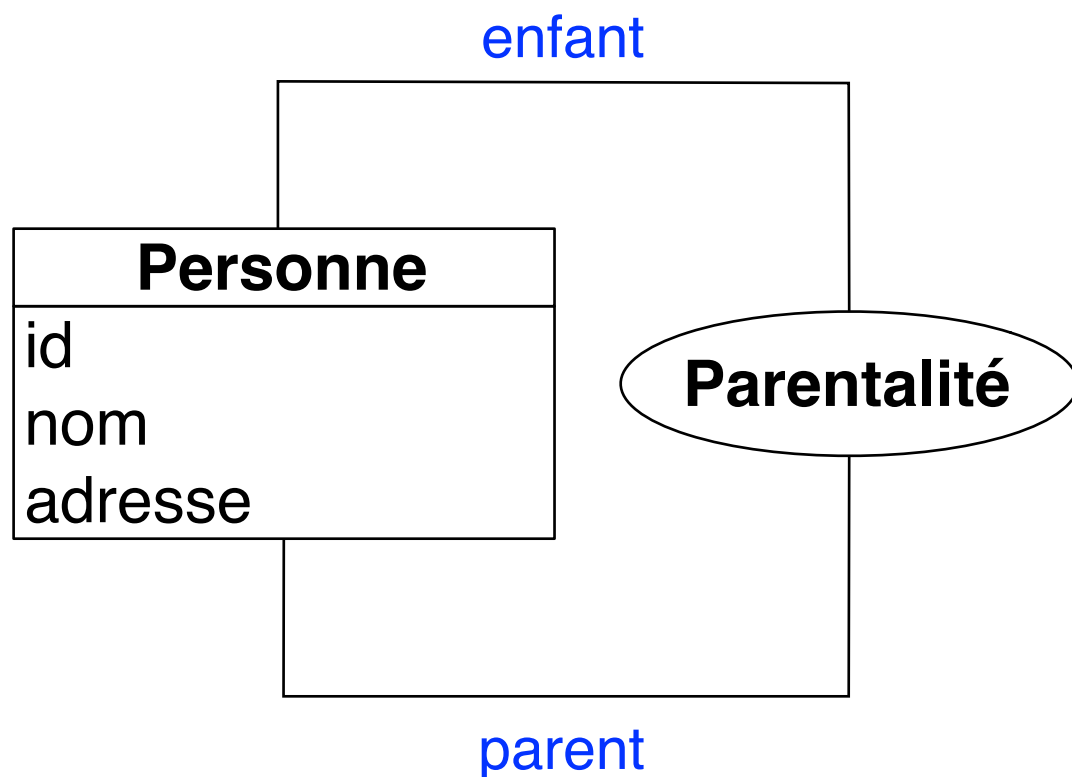
- Plusieurs attributs sont possibles :



chacun représente une propriété différente

Associations réflexives

- Une association peut relier une entité à elle-même, dans ce cas une étiquette (appelé **rôle**) doit distinguer les deux cotés de l'association :



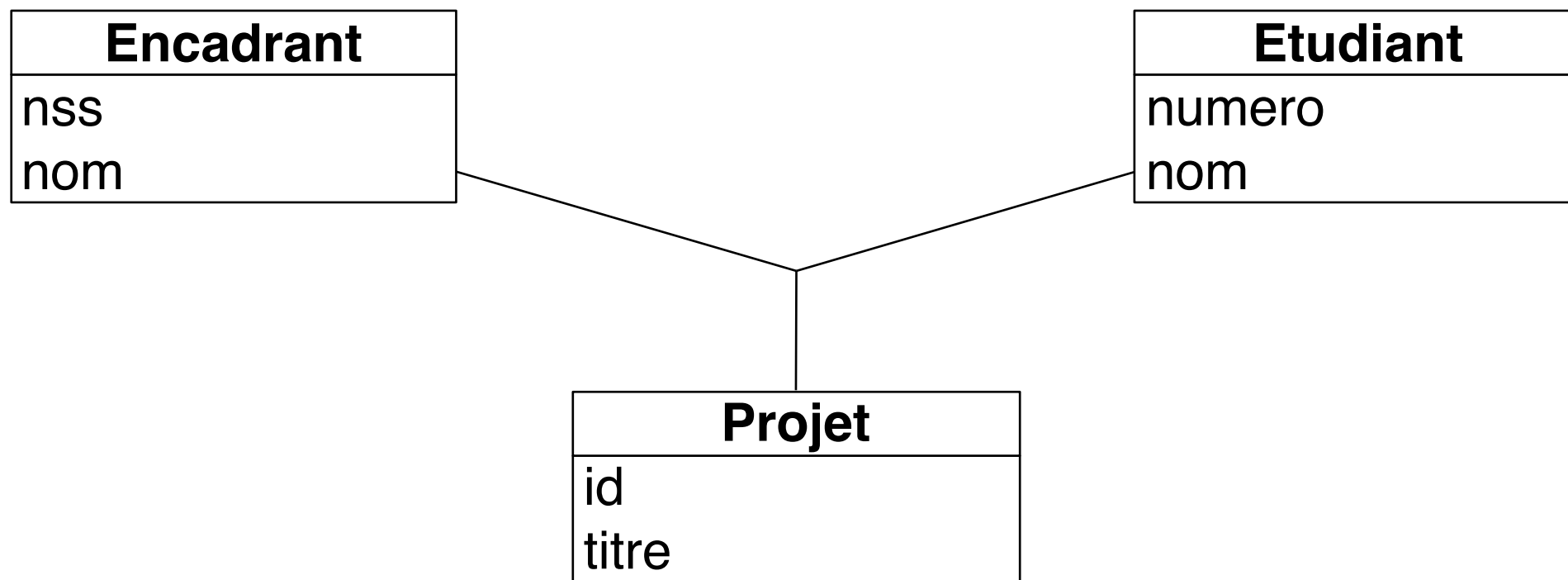
- Une instances de Parentalité est un couple de personnes étiquetées par les rôles. Dans un chaque couple : une personne a le rôle “parent” et l’autre le rôle “enfant”

Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Encadrement**

(Les étudiants sont encadrés par des encadrants sur des projets)

Représentation
graphique :

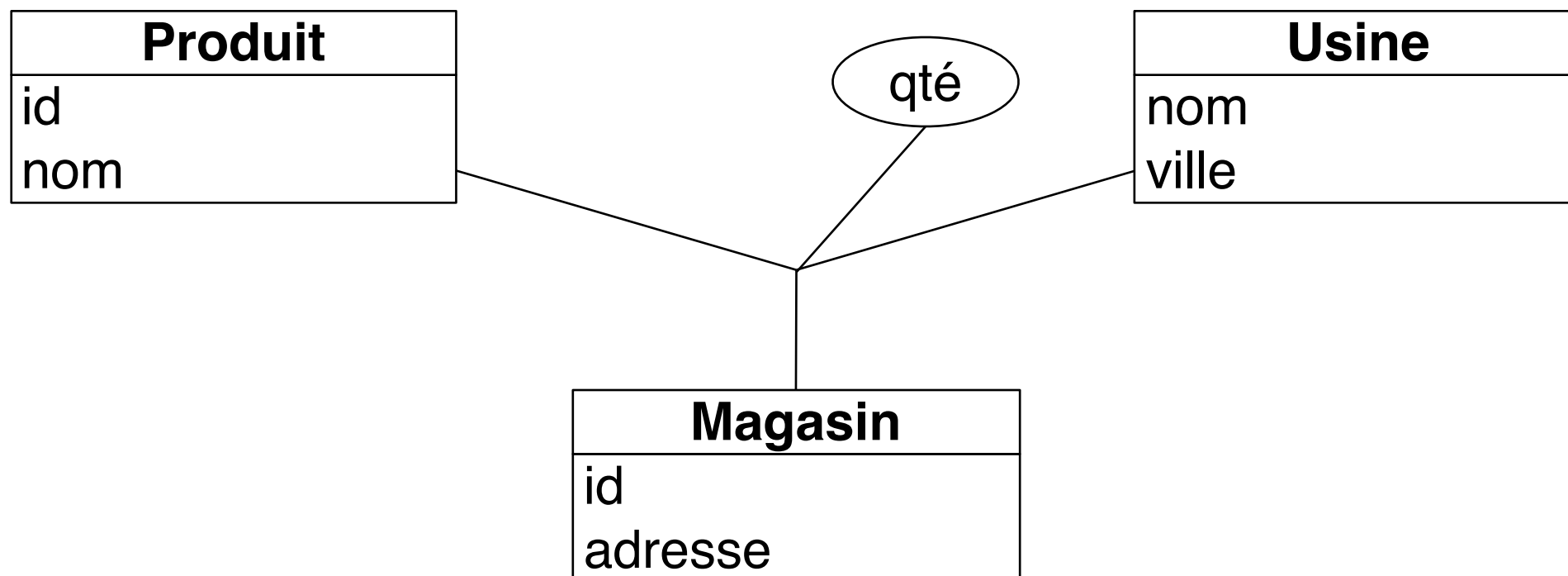


Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Disponibilité**

(Dans chaque magasin il y a une certaine quantité disponible de produits fabriqués dans des usines)

Représentation
graphique :

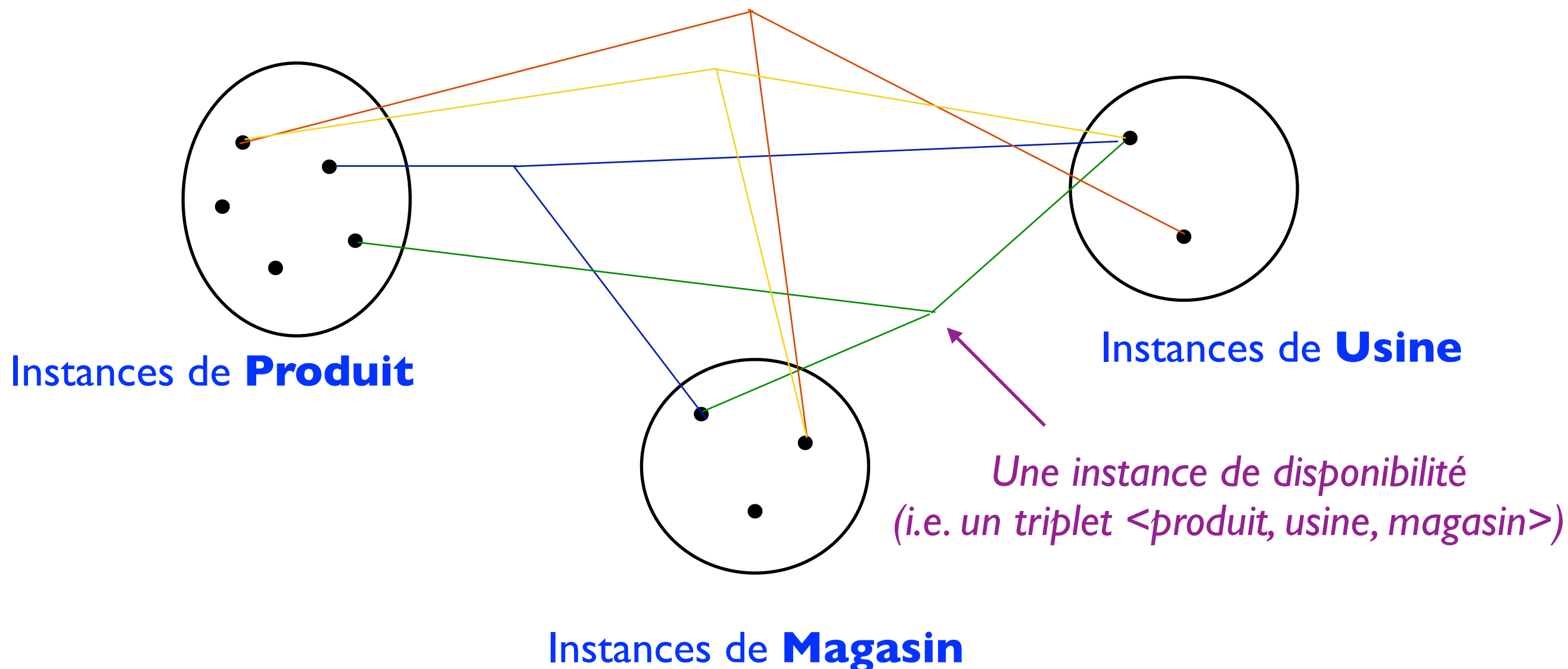


Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Disponibilité**

Instances :

Instances de **Disponibilité**

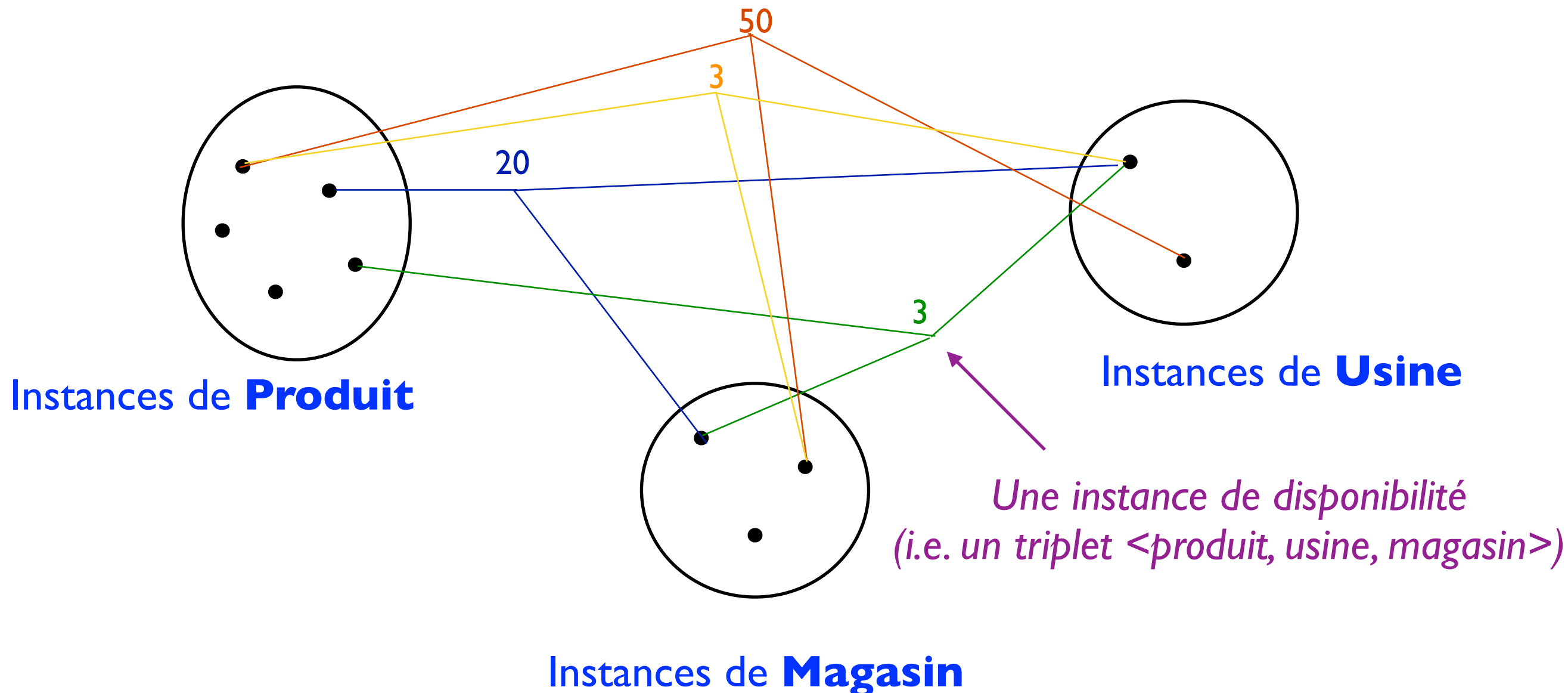


Associations n-aires

- Une association peut relier plus que deux entités. Ex. : l'association **Disponibilité**

Instances : **qté** : une propriété des triplets <produit, usine, magasin>

Instances de **Disponibilité**



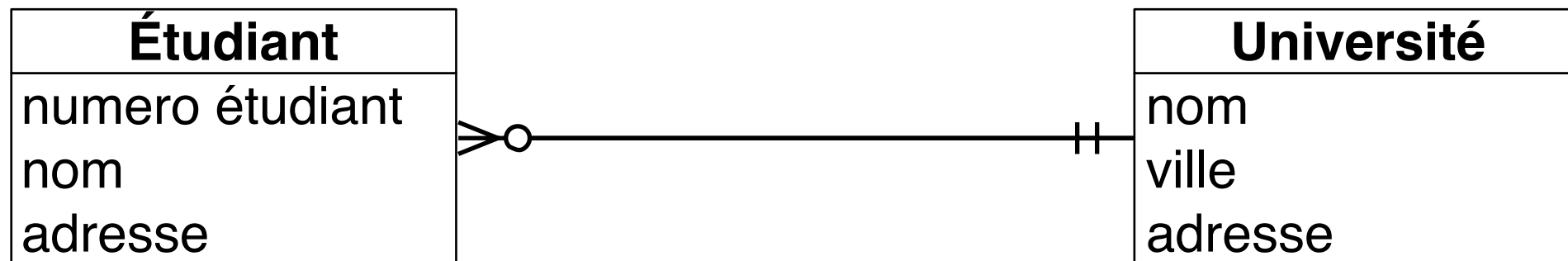
Contraintes dans le modele E/R

- Le modele E/R peut exprimer certaines contraintes sur les données :
 - ▶ **Contraintes** : conditions additionnelles pour que les instances d'un diagramme E/R soient valides
- Deux types de contraintes exprimables par le modèle E/R:
 - ▶ cardinalité
 - ▶ identification
- D'autres contraintes plus complexes doivent être ajoutées au diagramme E/R en tant que **contraintes externes**

Contraintes de cardinalité

- Expriment le nombre d'instances des entités qui peuvent être associées via une association

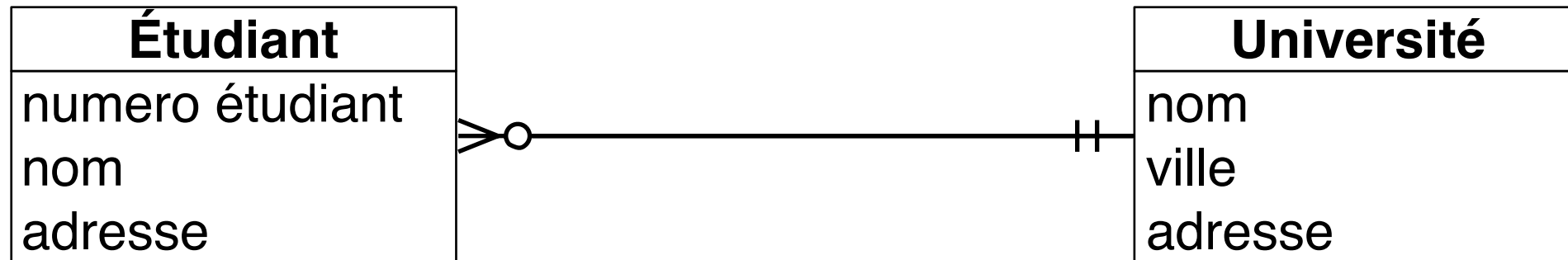
Exemple : Contraintes de cardinalité pour l'association **Inscription**



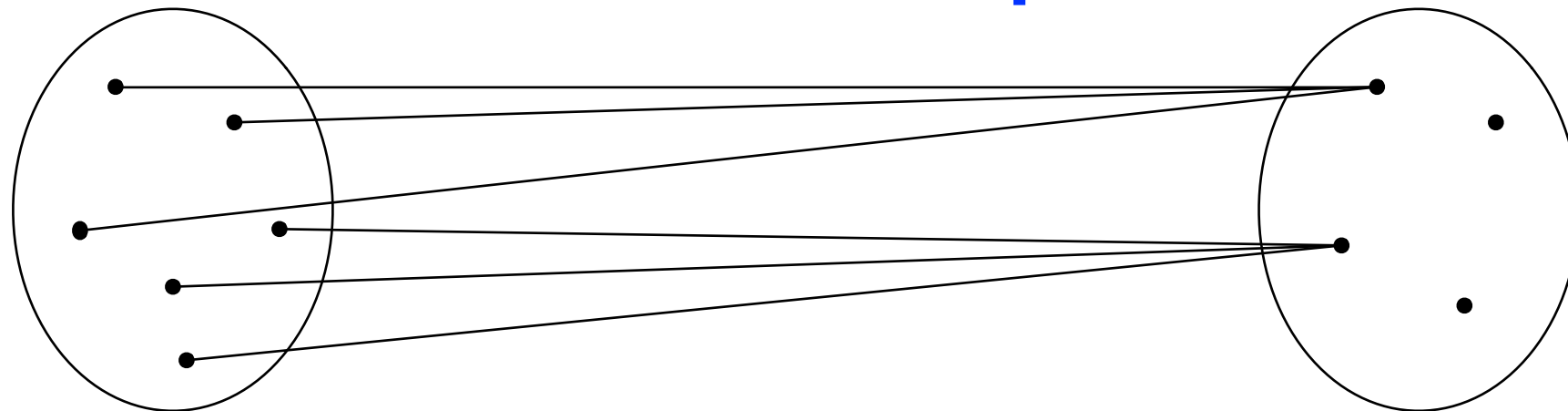
(Notation Crow's Foot)

- un même étudiant peut être inscrit à **une et une seule** université (min : 1, max : 1)
- une université peut avoir **entre 0 et plusieurs** inscrits (min : 0, pas de max)

Contraintes de cardinalité



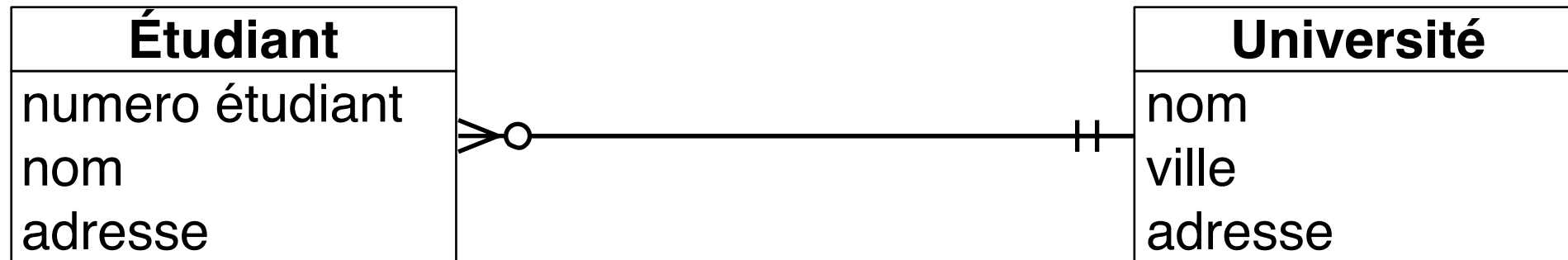
Instances de **Inscription**



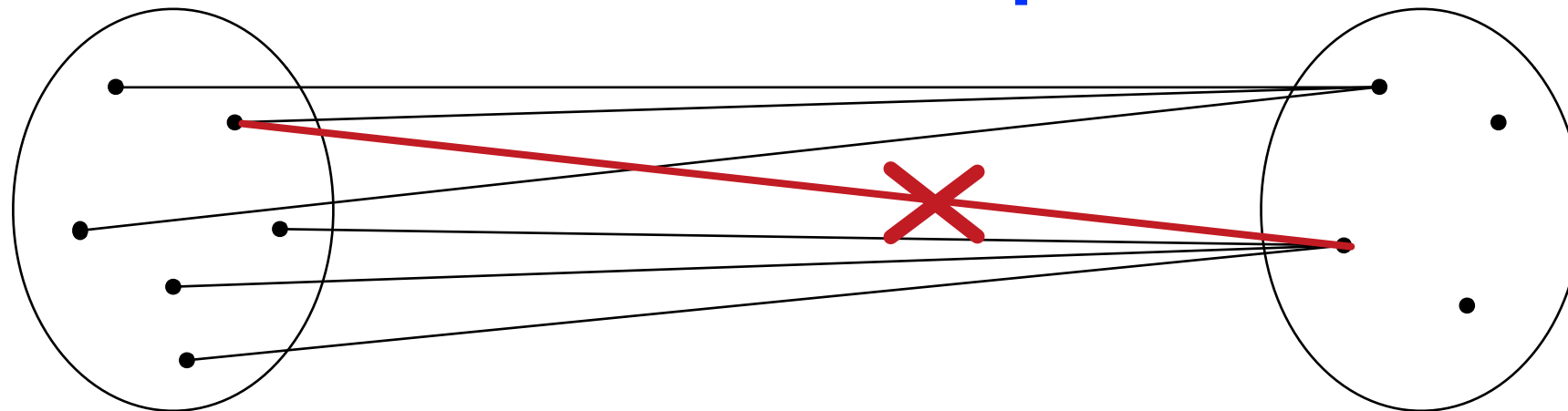
Instances de **Étudiant**

Instances de **Université**

Contraintes de cardinalité



Instances de **Inscription**



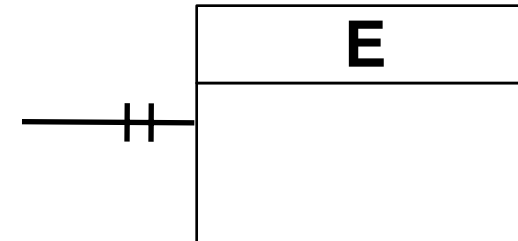
Instances de **Étudiant**

Instances de **Université**

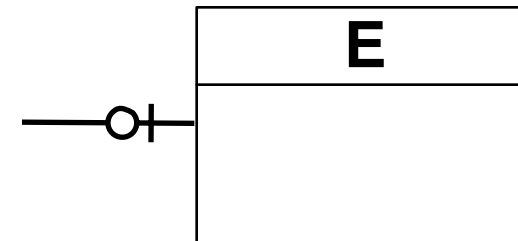
Contraintes de cardinalité

- Cardinalités possibles :

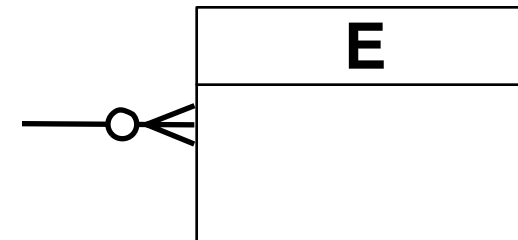
min: 1, max : 1



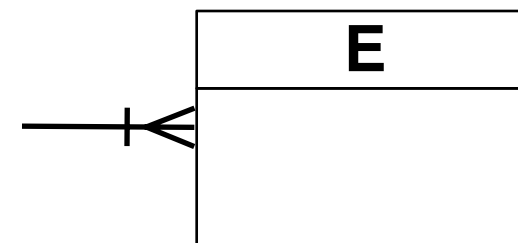
min: 0, max : 1



min: 0, pas de max

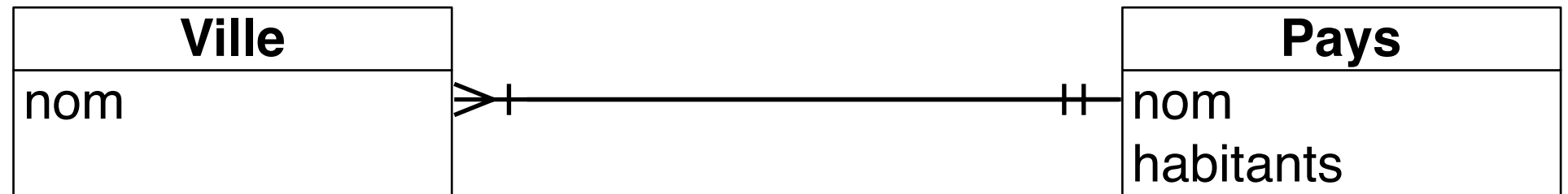


min: 1, pas de max

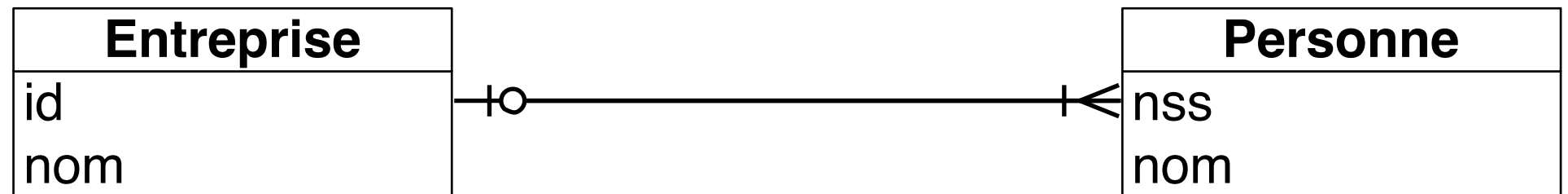


Contraintes de cardinalité : exemples

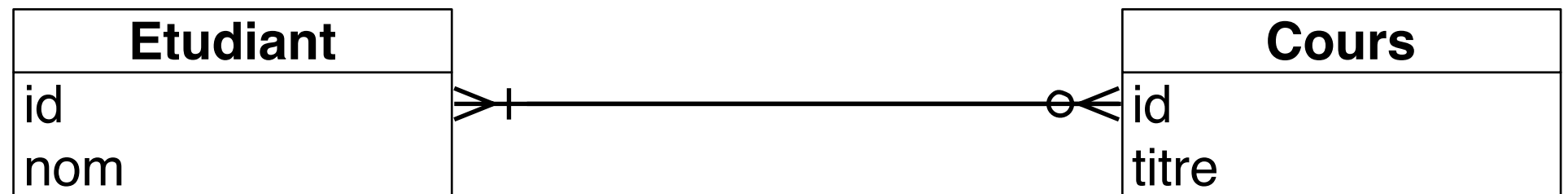
- Localisation :



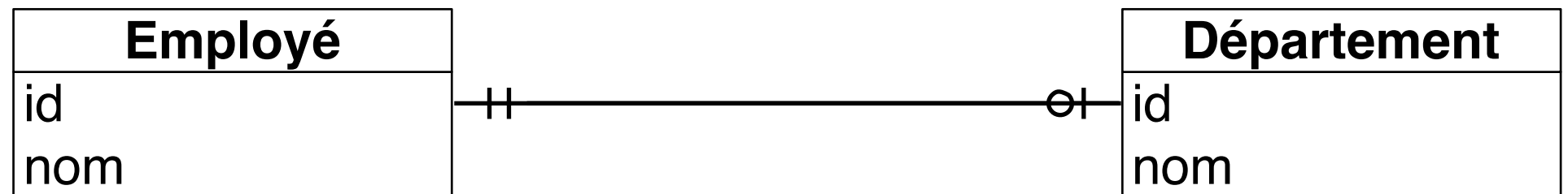
- Emploie :



- Inscription :



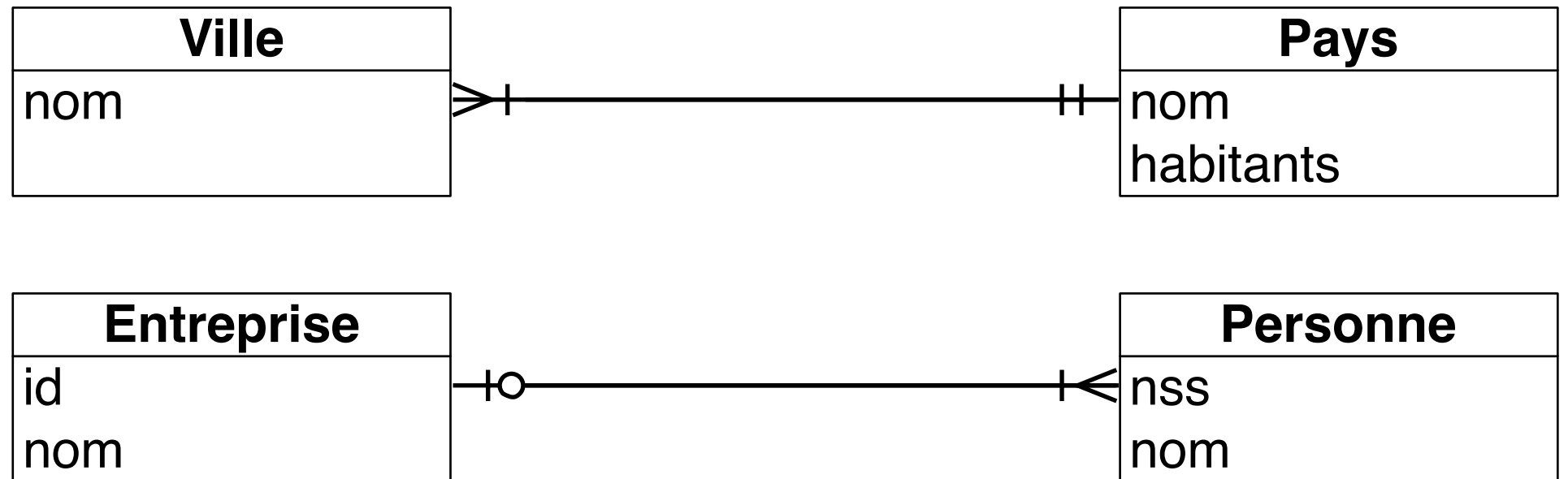
- Directeur :



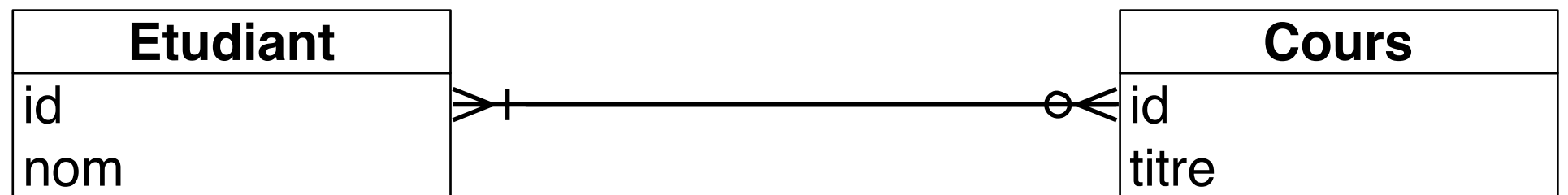
Multiplicité des associations

- En faisant référence uniquement au **cardinalités maximales** de chaque coté, on distingue entre associations :

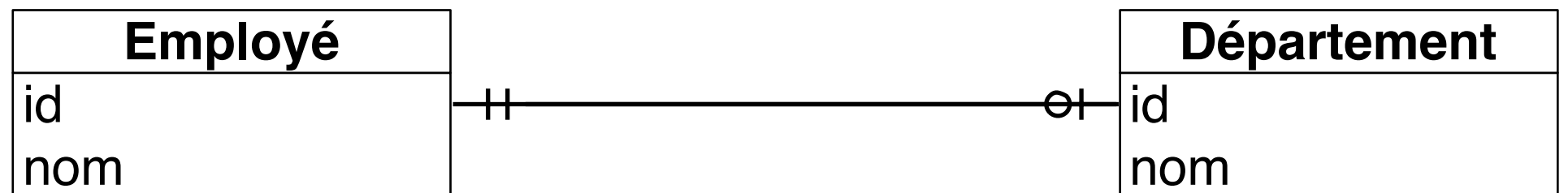
un à plusieurs



plusieurs
à plusieurs

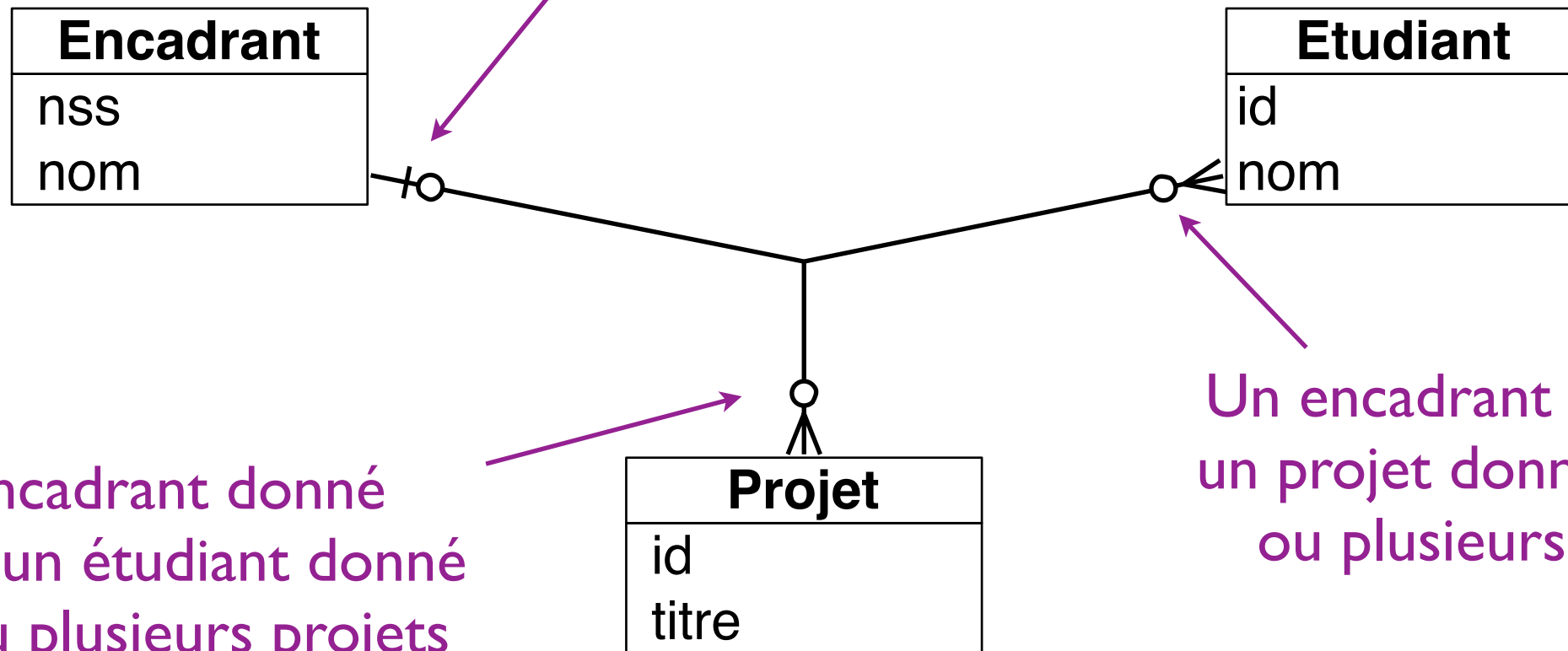


un à un



Contraintes de cardinalité dans les associations n-aires

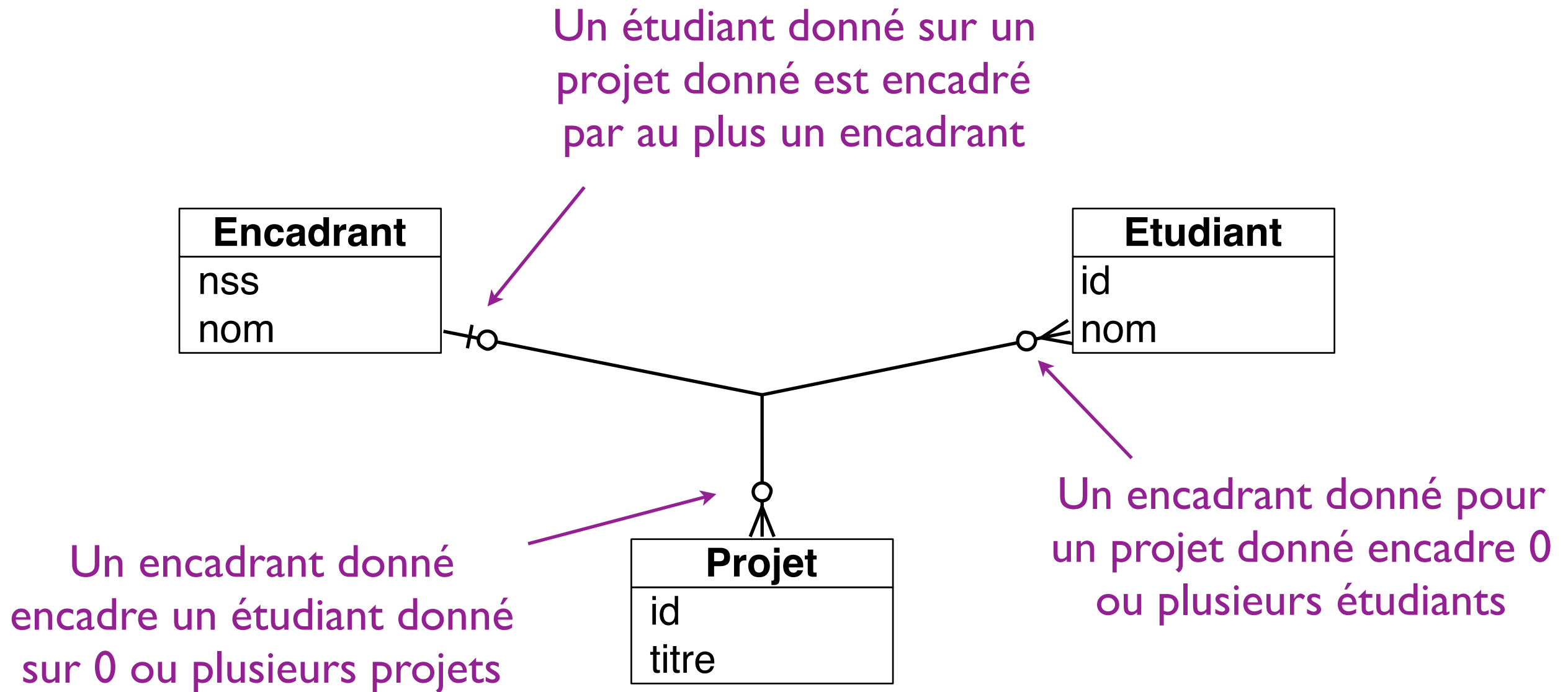
Un étudiant donné sur un projet donné est encadré par au plus un encadrant



Un encadrant donné encadre un étudiant donné sur 0 ou plusieurs projets

Un encadrant donné pour un projet donné encadre 0 ou plusieurs étudiants

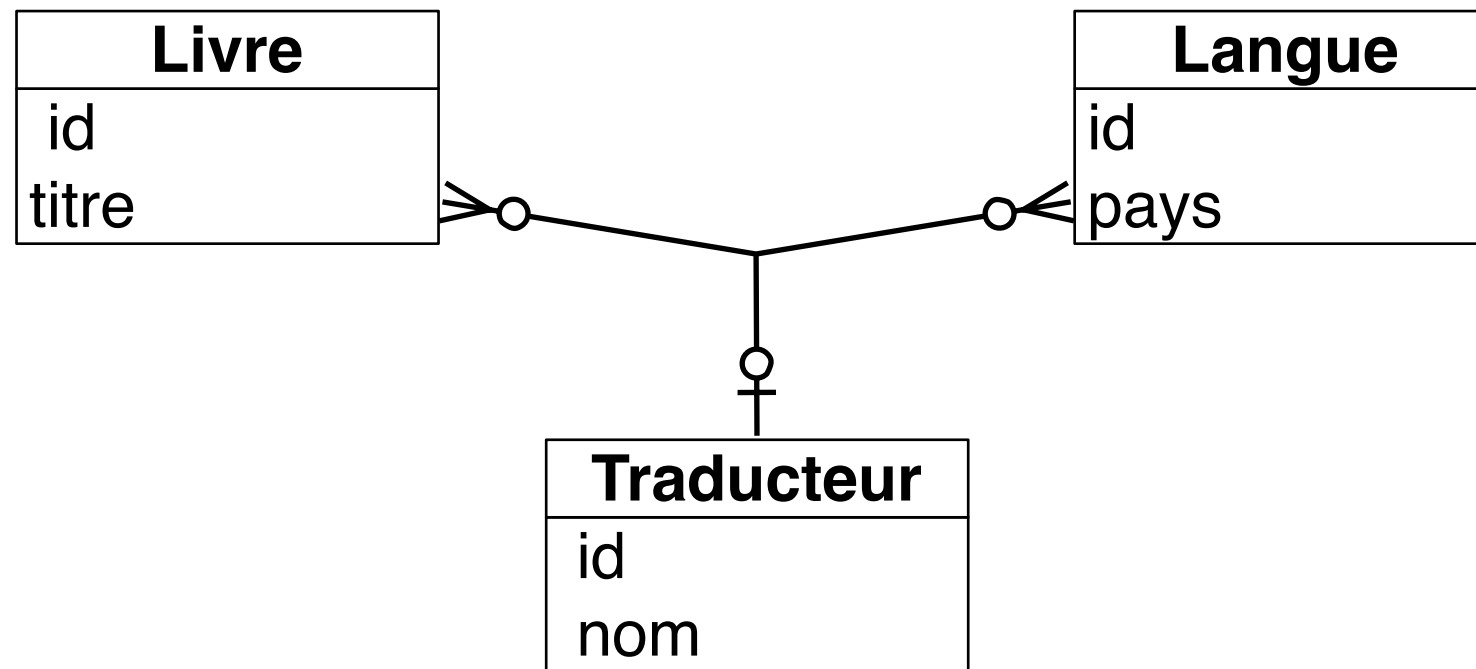
Contraintes de cardinalité dans les associations n-aires



- **Remarque** : les cardinalités minimales sont moins intéressantes sur les associations ternaires : elles sont en général 0
 - ▶ E.g : si chaque couple <étudiant, projet> était associé à min 1 encadrant, cela voudrait dire que tous les étudiants sont affectés à tous les projets!

Contraintes de cardinalité dans les associations n-aires

D'autres exemples : l'association **Traduction**



À 1 livre et 1 langue correspond 0 ou 1 traducteur.

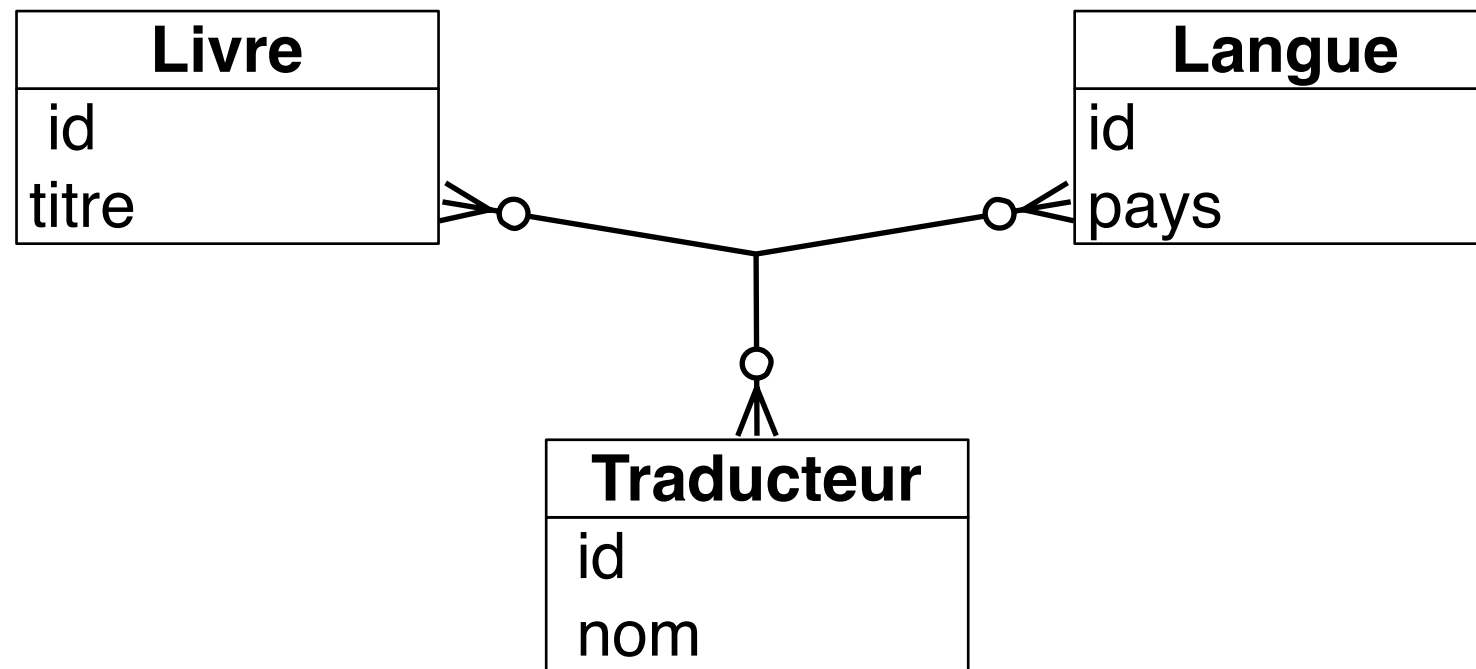
À 1 langue et 1 traducteur correspond 0 ou plusieurs livres.

À 1 livre et 1 traducteur correspond 0 ou plusieurs langues.

Contraintes de cardinalité dans les associations n-aires

D'autres exemples : l'association **Traduction**

Si en revanche un livre peut être traduit dans la même langue par plusieurs traducteurs :



Rappel : Contraintes dans le modèle E/R

- Le modèle E/R peut exprimer certaines contraintes sur les données :
 - ▶ **Contraintes** : conditions additionnelles pour que les instances d'un diagramme E/R soient valides
- Deux types de contraintes :
 - ▶ cardinalité
 - ▶ **identification (clefs)**

Clefs

- Une **superclef** d'une entité est un ensemble de un ou plusieurs attributs tel que: il n'existe pas deux instances de l'entité avec la même valeur de tous ces attributs
 - ▶ Exemples :
 - **nss** est une superclef pour l'entité Personne
 - **(titre, réalisateur)** est une superclef pour l'entité Film
 - **(nss, nom)** est une superclef pour l'entité Personne
 - **(ville, rue, numéro)** est une superclef pour l'entité Bâtiment
- Une **clef** (ou **clef candidate**) d'une entité est une superclef minimale
 - **nss** est une clef pour l'entité Personne
 - **(titre, réalisateur)** est une clef pour l'entité Film
 - **(ville, rue, numéro)** est une clef pour l'entité Bâtiment
- Plusieurs clefs candidates peuvent exister pour une entité, mais pour chaque entité une seule clef est choisie comme **clef primaire**.

Clefs primaires

- Dans un diagramme E/R une clef primaire est spécifiée pour chaque entité

| Bâtiment |
|--------------------|
| <u>numero</u> |
| <u>rue</u> |
| <u>ville</u> |
| nombre-étages |
| année-construction |

Contrainte imposée sur les instances :

Il n'existe pas deux bâtiments différents b_1 et b_2 dans les instances de Bâtiment tels que : $\text{numero}(b_1) = \text{numero}(b_2)$ **et** $\text{rue}(b_1) = \text{rue}(b_2)$ **et** $\text{ville}(b_1) = \text{ville}(b_2)$

- **Remarque** : il peut exister deux bâtiments avec le même numéro et le même nom de rue (dans des villes différentes), mais pas avec le même numéro, la même rue et la même ville.

Clefs primaires

- Un autre exemple de clef primaire :

| Film |
|--------------------|
| <u>titre</u> |
| <u>réalisateur</u> |
| année |

Contrainte imposée sur les instances :

Il n'existe pas deux films différents f_1 et f_2 dans les instances de Film tels que :
 $\text{titre}(f_1) = \text{titre}(f_2)$ **et** $\text{réalisateur}(f_1) = \text{réalisateur}(f_2)$

Clefs primaires

- Un dernier exemple :

| Personne |
|----------------|
| <u>nss</u> |
| nom |
| prenom |
| date-naissance |

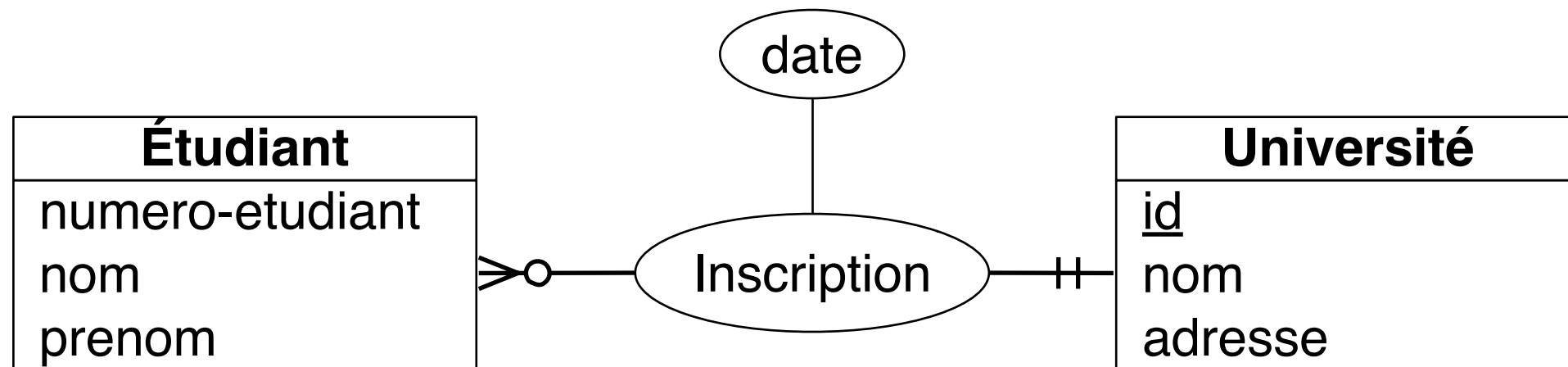
Contrainte imposée sur les instances :

Il n'existe pas deux personnes différents $p1$ et $p2$ dans les instances de Personne tels que: $nss(p1) = nss(p2)$

Entités faibles

- Pour certaines entités, aucun ensemble d'attributs ne forme une clef

Exemple :

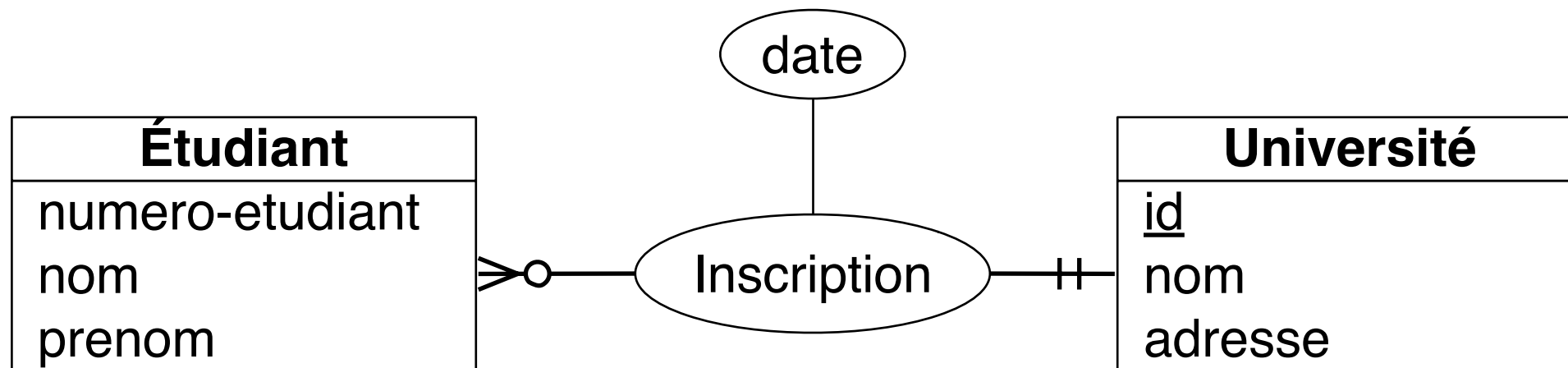


Quel ensemble d'attributs est une clef pour l'entité Étudiant ?

Entités faibles

- Pour certaines entités, aucun ensemble d'attributs ne forme une clef

Exemple :



Un étudiant est identifié par son numéro étudiant au sein de son université.

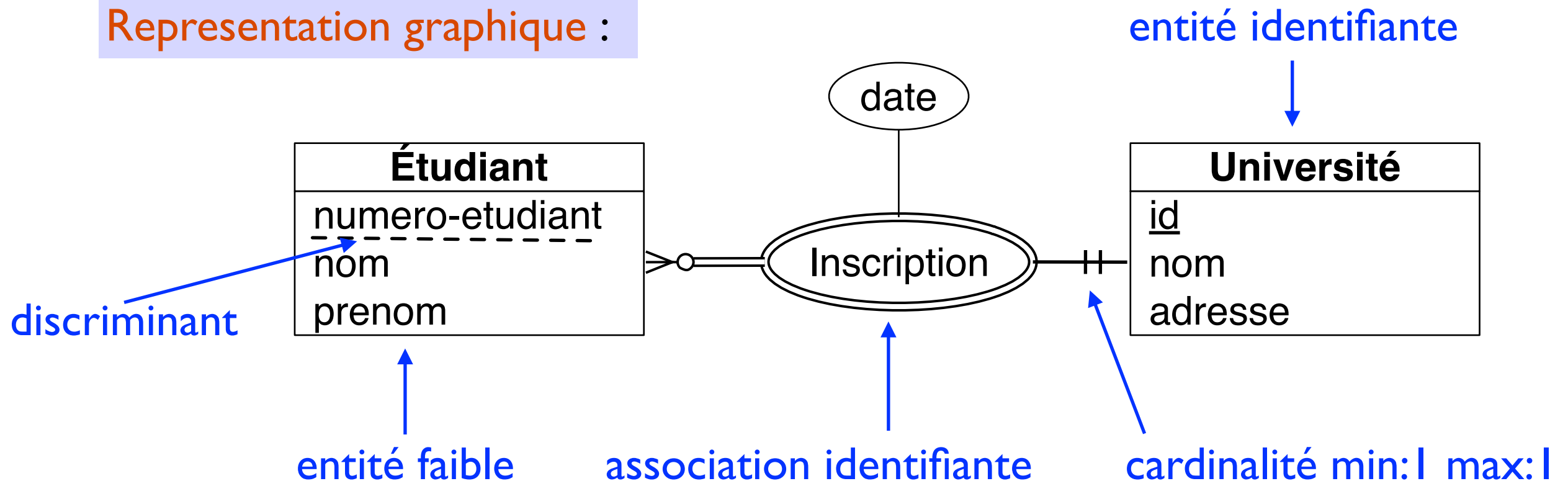
En d'autres termes une clef pour Étudiant est donnée par le couple

- numéro étudiant
- université d'inscription

Entités faibles

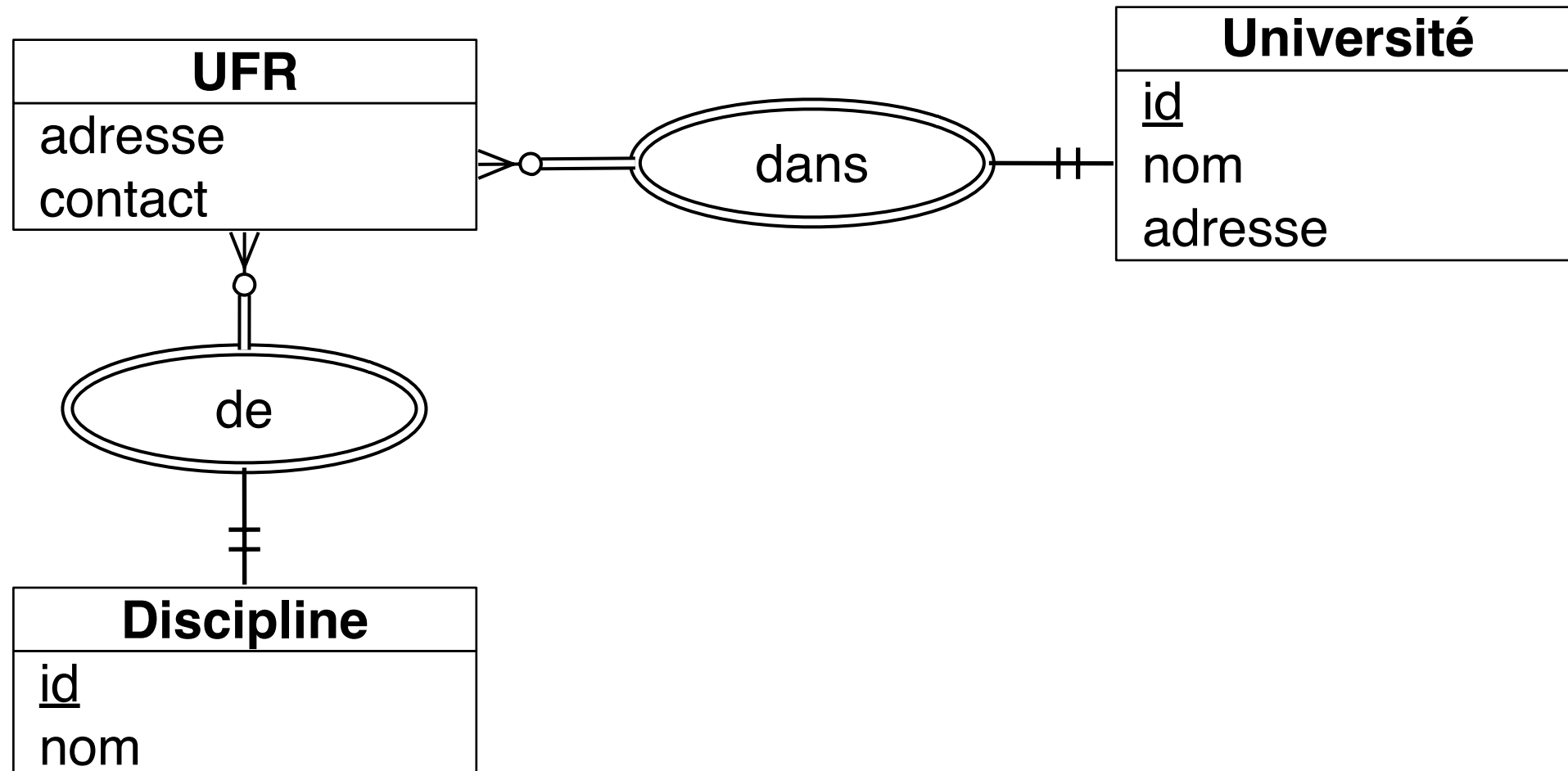
- Une entité faible est une entité identifiée par
 - ▶ un ensemble d'attributs internes appelés *discriminant* et
 - ▶ un ensemble d'entités appelés *entités identifiantes*
- Chaque entité identifiante doit être reliée à l'entité faible par une *association binaire* appelée *association identifiante*
- l'entité faible doit participer à l'association identifiante avec une *cardinalité min:1 max:1*

Représentation graphique :



Entités faibles

- Un exemple avec plusieurs entités identifiantes :



Remarque : dans cet exemple il n'y a pas d'attributs discriminants :
la clef de l'entité faible est totalement externe : Université et Discipline

Aspect pas abordé du modèle E/R

- **Spécialisation** (aussi appelé **héritage** ou **généralisation**)
 - ▶ permet de spécifier des contraintes d'inclusion entre les instances de plusieurs entités

Un exemple complet de modélisation E/R

Un exemple complet de modélisation E/R

Représenter les informations suivantes concernant les vols d'une compagnie aérienne.

La compagnie propose des vols, chacun avec un numéro qui l'identifie (par exemple AF326), une date, un horaire de départ (8:00) et un horaire d'arrivée (12:00), un aéroport de départ (e.g. Paris CDG) et un aéroport d'arrivée (e.g. New York JFK). Le même numéro de vol peut être programmé à plusieurs dates, cependant à chaque date il fait le même trajet et est prévu aux mêmes horaires.

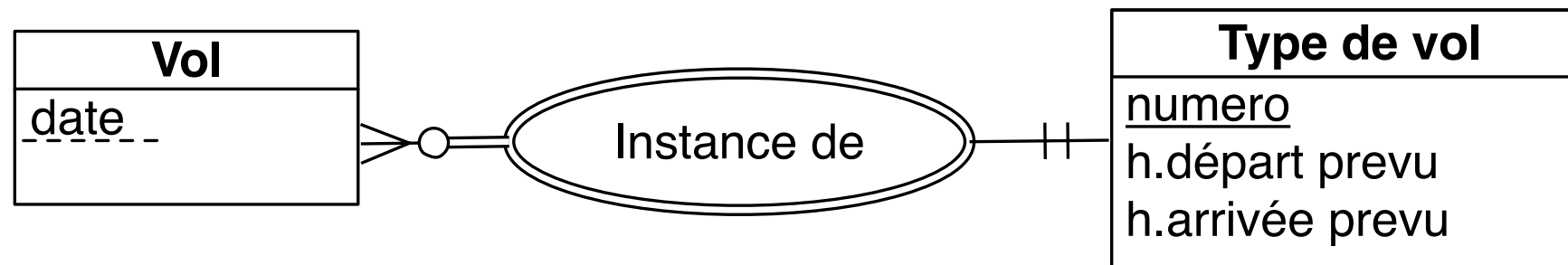
Certains vols peuvent avoir un ou plusieurs arrêts intermédiaires.

Pour les vols qui ont déjà eu lieu on représente également les horaires effectifs de départ et d'arrivée (il peut y avoir des retards/anticipations par rapport aux horaires prévus, e.g. pour le vol ci-dessus 8:05 et 12:07). Pour les vols futurs on représente le nombre de places disponibles.

Un exemple complet de modélisation E/R

Remarque. Deux concepts distincts de vol :

- **type de vol** : avec un numéro, des horaires prévus, des aéroports de départ, d'arrivée et possiblement d'arrêt;
- **vol effectif** : une instance d'un certain type de vol, qui a lieu à une date, avec des horaires effectifs (s'il a déjà eu lieu) ou des places disponibles (s'il n'a pas encore eu lieu).

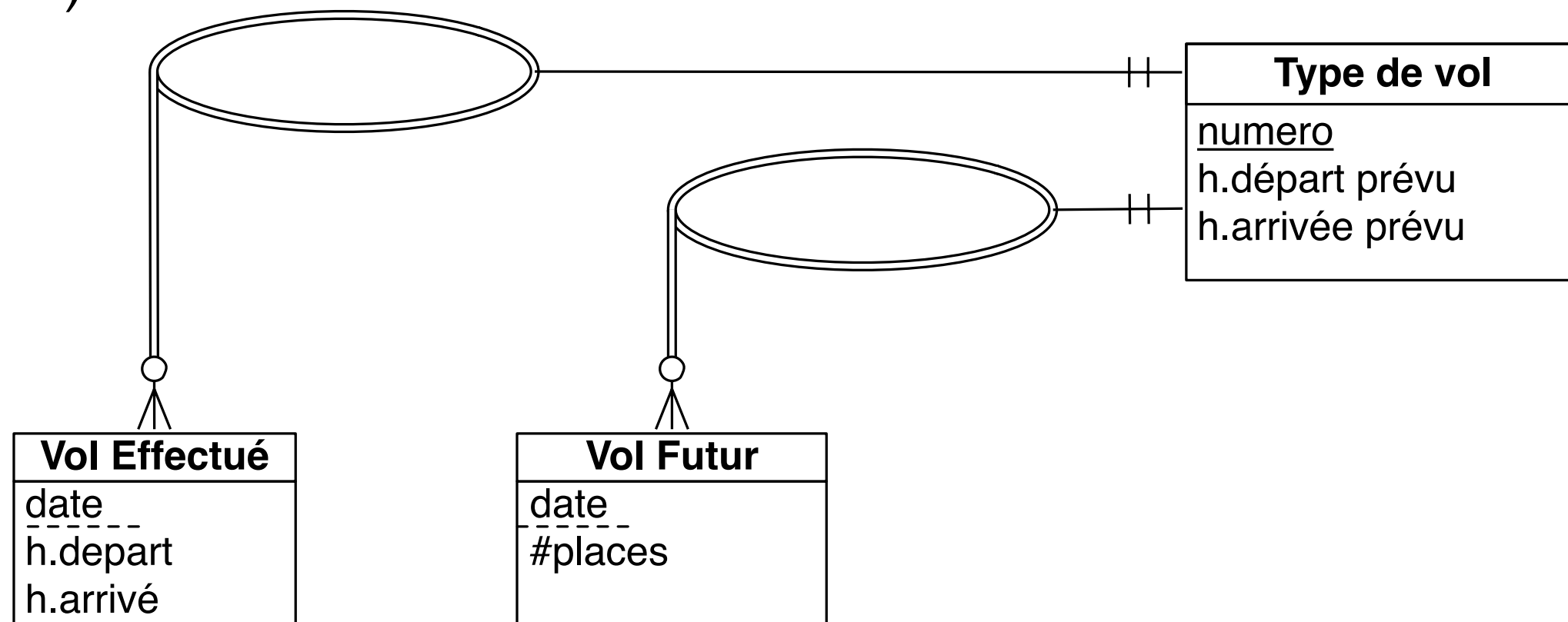


mais il faut distinguer entre vol futur et vol effectué (propriétés différentes)

Un exemple complet de modélisation E/R

Remarque. Deux concepts distincts de vol :

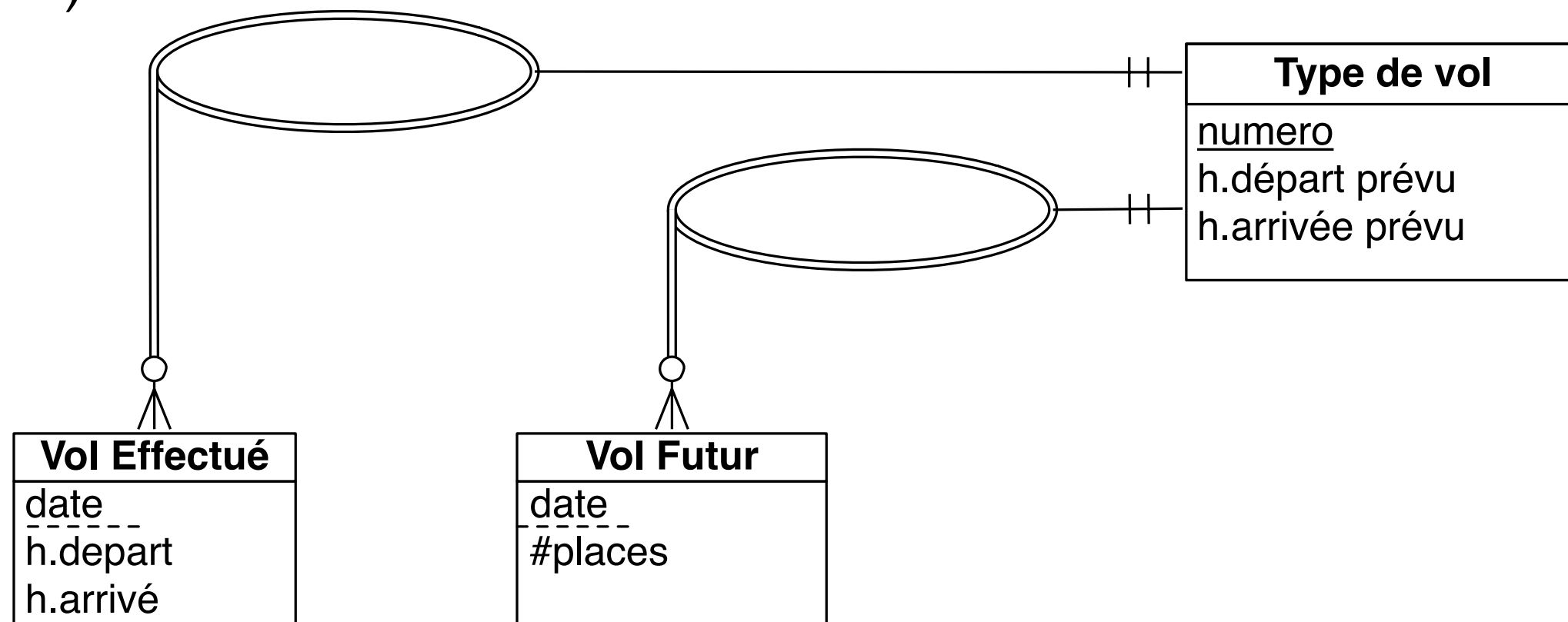
- **type de vol** : avec un numéro, des horaires prévus, des aéroports de départ, d'arrivée et possiblement d'arrêt;
- **vol effectif** : une instance d'un certain type de vol, qui a lieu à une date, avec des horaires effectifs (s'il a déjà eu lieu) ou des places disponibles (s'il n'a pas encore eu lieu).



Un exemple complet de modélisation E/R

Remarque. Deux concepts distincts de vol :

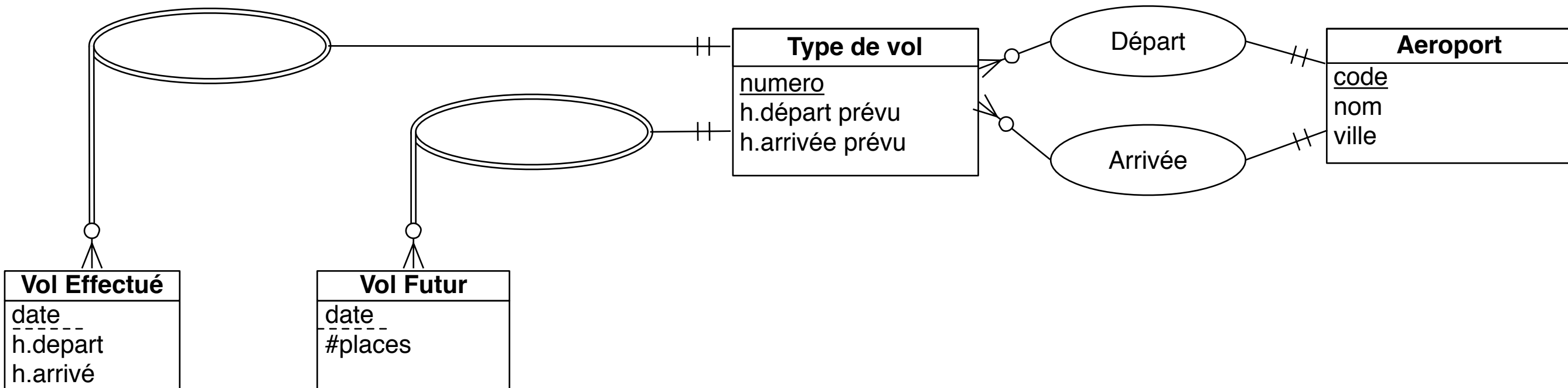
- **type de vol** : avec un numéro, des horaires prévus, des aéroports de départ, d'arrivée et possiblement d'arrêt;
- **vol effectif** : une instance d'un certain type de vol, qui a lieu à une date, avec des horaires effectifs (s'il a déjà eu lieu) ou des places disponibles (s'il n'a pas encore eu lieu).



+ **contrainte externe** : la date de chaque vol effectué est \leq à la date de chaque vol futur

Un exemple complet de modélisation E/R

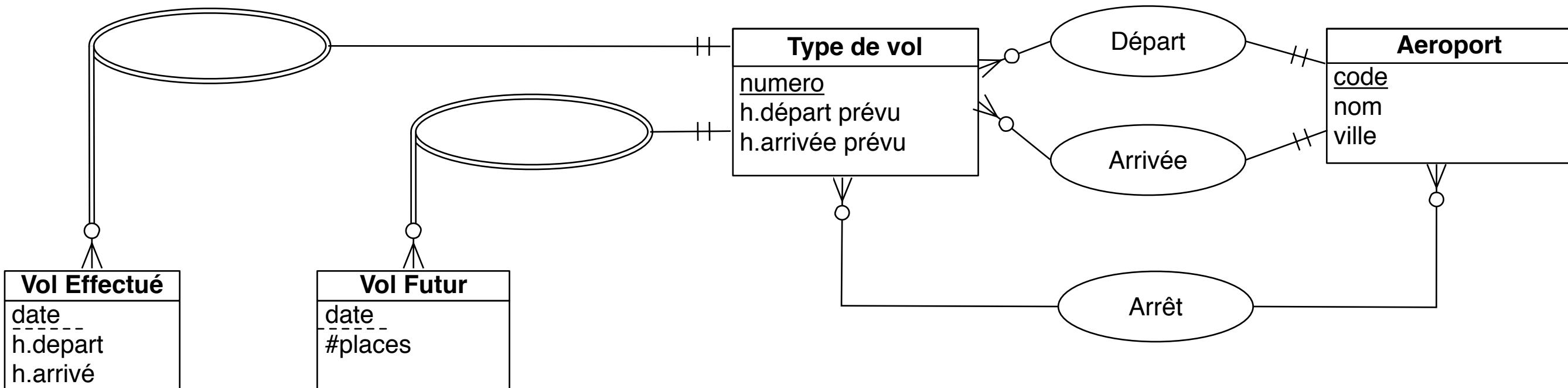
- Aéroport de départ et d'arrivée sont des propriétés du type de vol :



+ **contrainte externe** : la date de chaque vol effectué est \leq à la date de chaque vol futur

Un exemple complet de modélisation E/R

- Certains vols peuvent avoir des arrêts



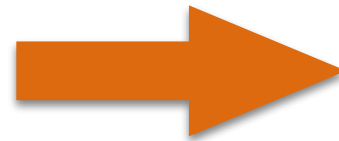
+ **contrainte externe** : la date de chaque vol effectué est \leq à la date de chaque vol futur

Traduction : diagramme E/R → schéma relationnel

- Règle générale
 - ▶ chaque entité est traduite en
 - ▶ un schéma de relation
 - ▶ des contraintes de clef primaire / clef secondaire
 - ▶ chaque association est traduite en
 - un schéma de relation
 - des contraintes de clef primaire / clef secondaire
 - des contraintes de clefs étrangère
 - ▶ d'autres contraintes du schéma relationnel (inclusion, NOT NULL, ..) peuvent être déduites des contraintes de cardinalité du diagramme E/R
 - ▶ les contraintes externes plus complexes du diagramme E/R sont traduites par des assertions ou conditions de CHECK

Entités

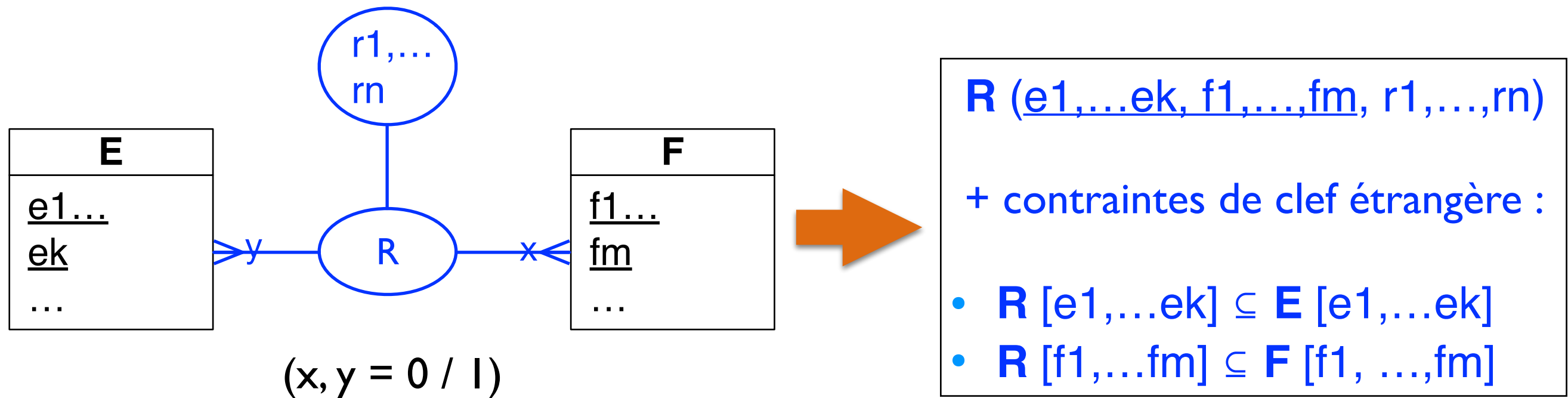
| E |
|---------------------------|
| <u>c1...</u> <u>ck</u> |
| a1... an |



E (c1, ...ck, a1,...,an)

- Cette traduction **peut être en suite modifiée** au fur et à mesure que les associations auxquels E participe sont traduites, cf. plus loin
 - ▶ cela est vrai **en particulier si E est une entité faible** et c1..ck son discriminant

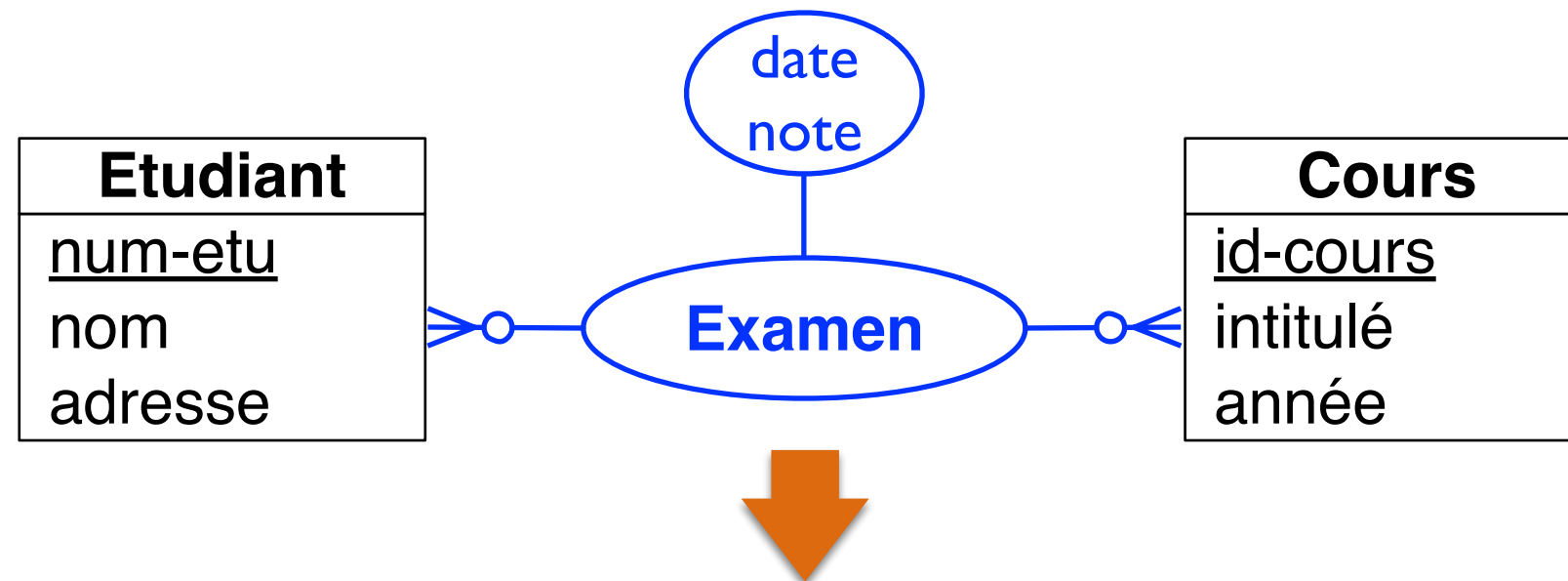
Associations plusieurs à plusieurs



- Remarque. Celle ci-dessus est une traduction de R uniquement (E et F doivent être traduites indépendamment)

Associations plusieurs à plusieurs - exemples

- **Exemple :** association plusieurs à plusieurs



Examen (num-et, id-cours, date, note)

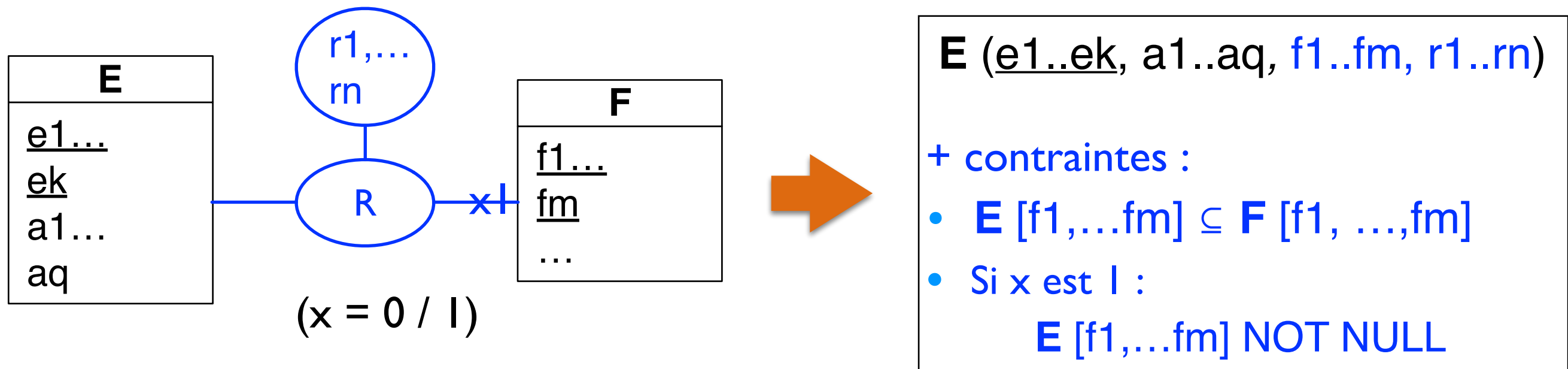
+ contraintes :

Examen [num-et] \subseteq **Etudiant** [num-et],

Examen [id-cours] \subseteq **Cours** [id-cours]

- **Remarque sur la clef de la relation produite.** La relation **Examen** représente les instances de l'association **Examen**, i.e. des couples <étudiant, cours>
 - ▶ un tel couple est identifié par <clef de l'étudiant, clef du cours>
 - ▶ cet identifiant est minimal pour **Examen** parce que un étudiant peut être associé à plusieurs cours et un cours à plusieurs étudiants

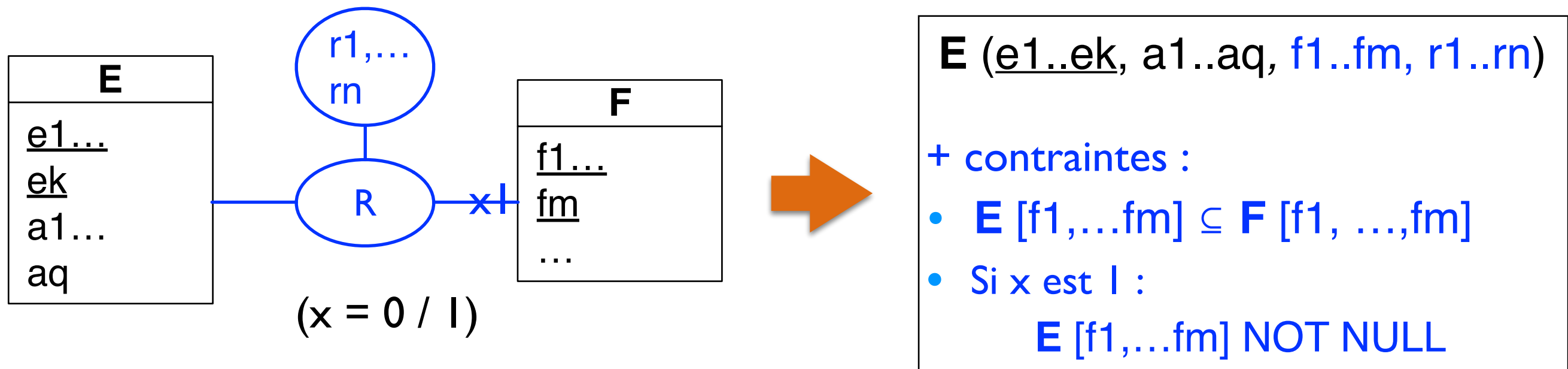
Associations un à plusieurs ou un à un



Remarques :

- ▶ on modifie la traduction de E en ajoutant à ses attributs l'identifiant de F , et les attributs de R
- ▶ cela est correct puisque chaque instance de E est associée par R à au plus une instance de F .
- ▶ Dans le cas de cardinalité $0/1$ cela pourrait introduire des NULL dans les valeurs de $f1 \dots fm$ (instances de E qui ne participent pas à R)

Associations un à plusieurs ou un à un

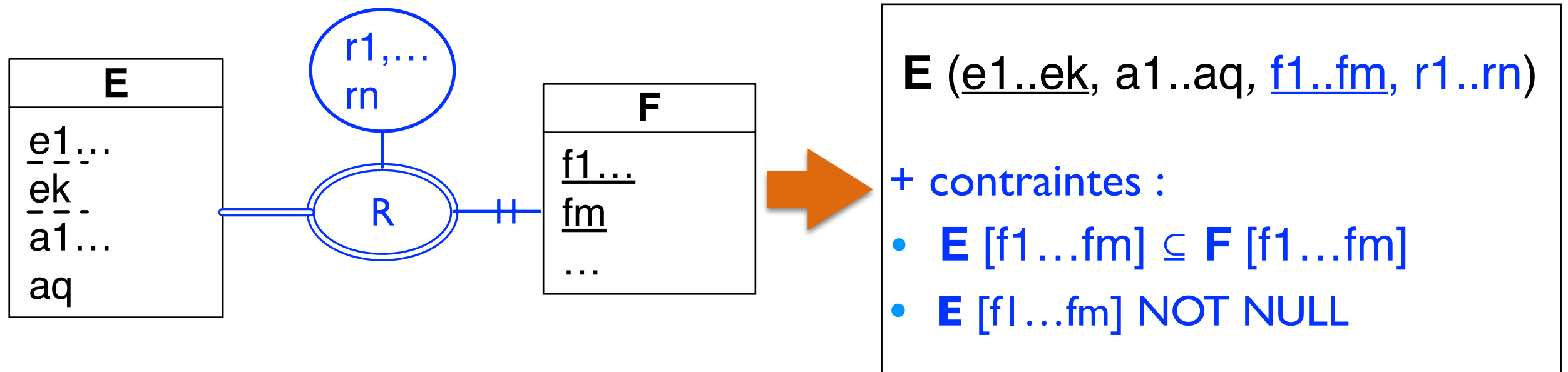


Encore des remarques :

- ▶ la traduction de R est la partie en bleu (E et F doivent être traduites indépendamment)
- ▶ l'effet de la traduction de plusieurs associations de ce type peut se cumuler en E
- ▶ si la cardinalité max est 1 des deux cotés, on modifie E ou F , **pas les deux**

Associations un à plusieurs ou un à un

- De plus : si R est identifiante pour l'entité avec participation max 1 (E)

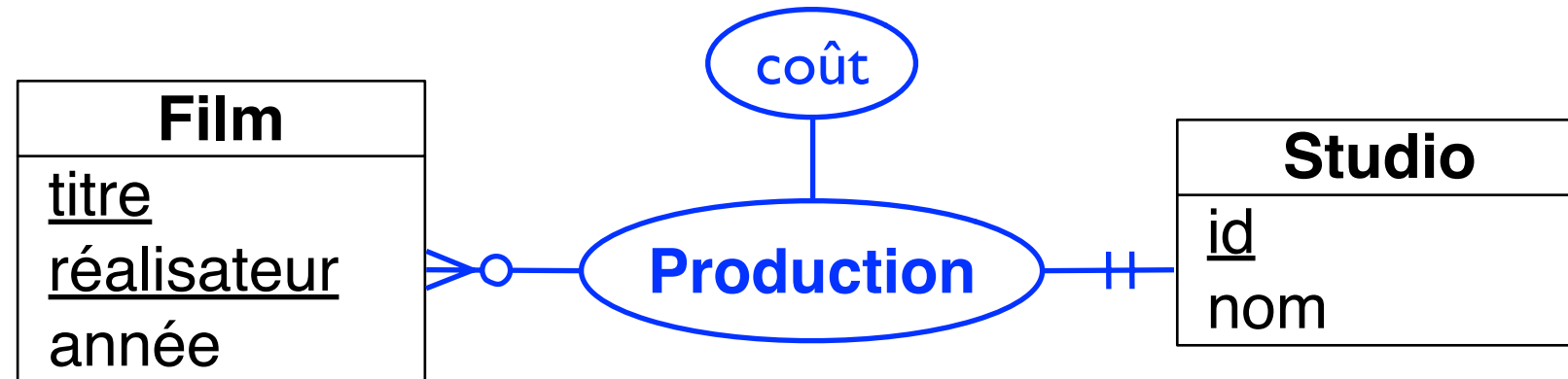


Remarques :

- ▶ même traduction que dans le cas précédent, mais la clef de E est étendue avec celle de F
- ▶ l'effet de la traduction de plusieurs associations de ce type (et du premier type) peut se cumuler en E
- ▶ si la cardinalité est 1 1 des deux cotés, R doit être traduite dans E (dont R est identifiante), pas dans F

Associations un à plusieurs ou un à un

- **Exemple**



Film (titre, réalisateur, année, id-studio, coût-production)

+ contrainte de clef étrangère :

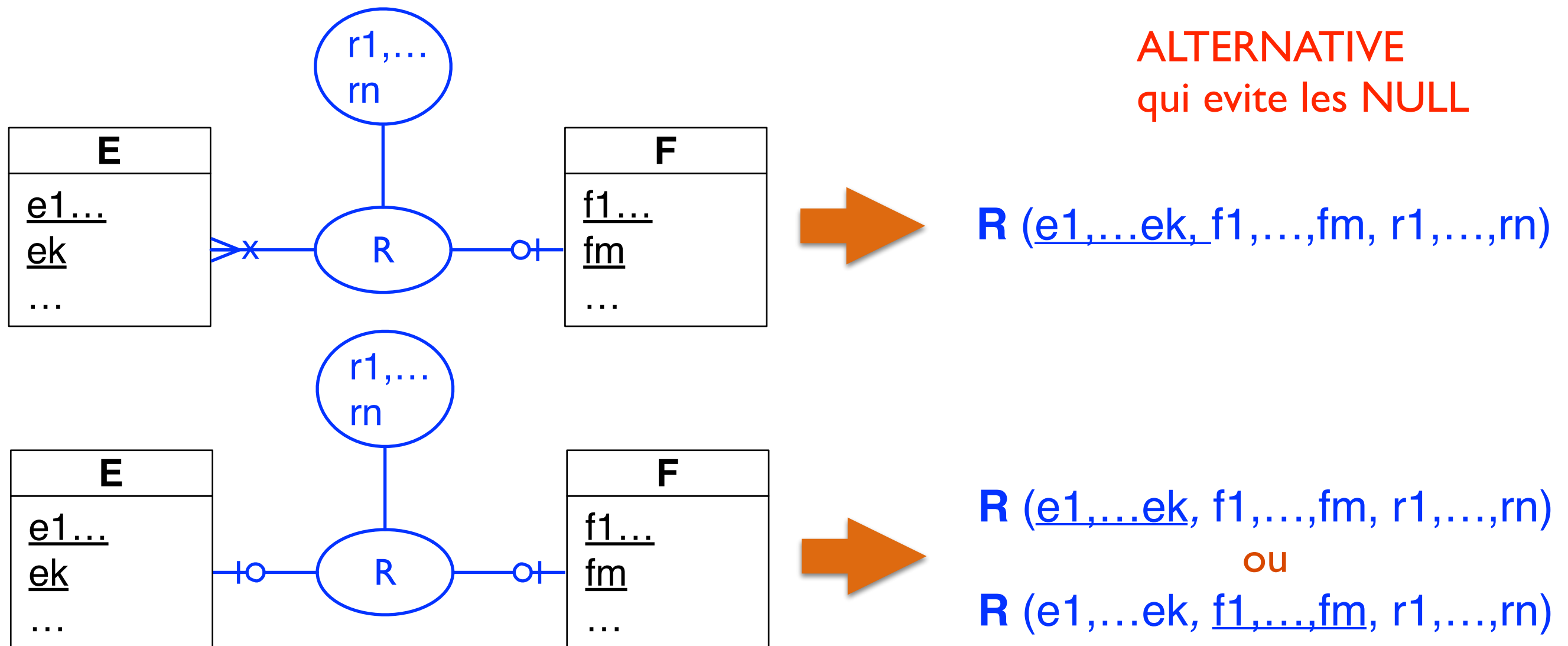
Film [id-studio] \subseteq **Studio** [id]

Film [id-studio] NOT NULL

- **Remarque.** On peut renommer les concepts (entités, associations, attributs) en passant au schéma relationnel

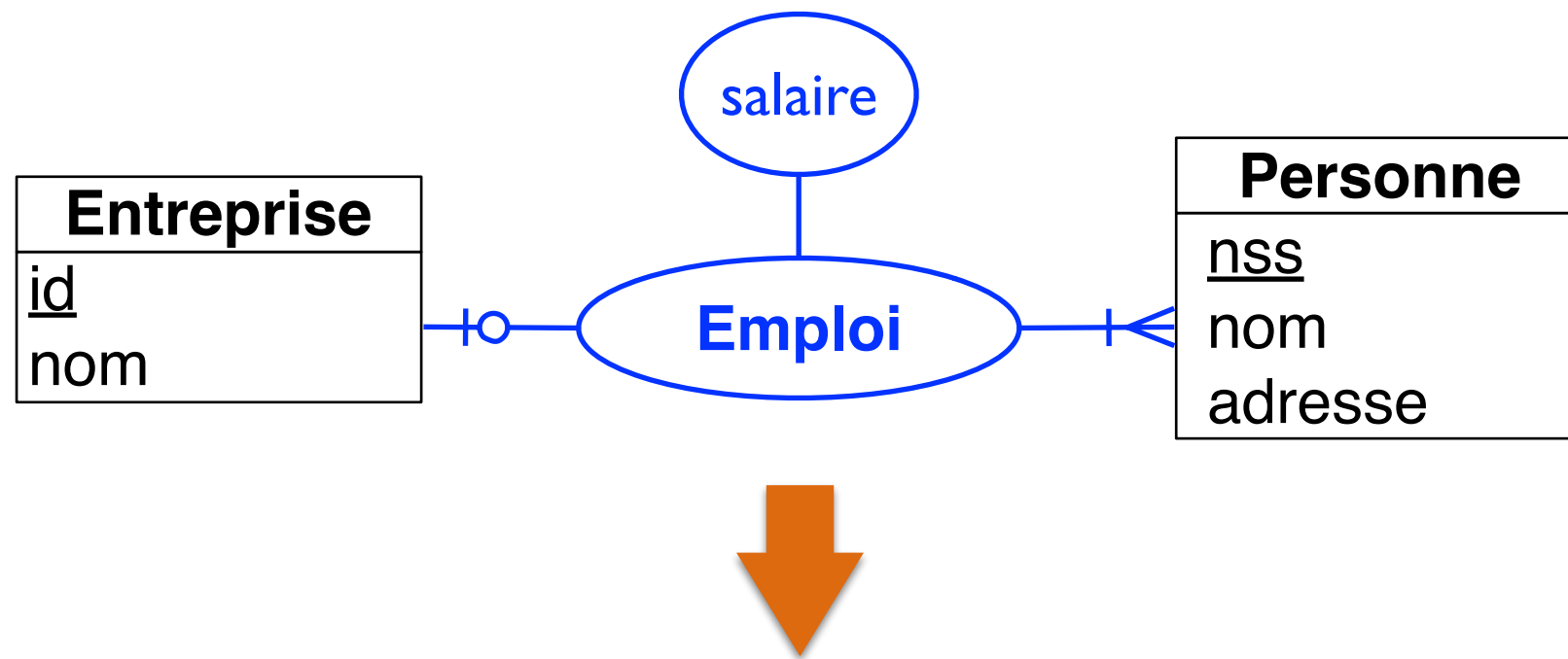
Associations un à plusieurs ou un à un

- En cas de cardinalité 01, cette traduction entraine des NULL
- Si on souhaite l'éviter, une alternative est de traduire l'association comme une association plusieurs à plusieurs
- Mais la clef est plus simple



Associations un à plusieurs ou un à un

- Exemple de traduction alternative : Association un à plusieurs



Traduction qui entraine des NULL

Personne (nss, nom, adresse,
id-entreprise, salaire)

+ contraintes :

Personne[id-entreprise] \subseteq Entreprise[id]

Traduction alternative qui évite les NULL

Emploi (nss, id-entreprise, salaire)

+ contraintes :

Emploi[nss] \subseteq Personne[nss]

Emploi[id-entreprise] \subseteq Entreprise[id]

Associations un à plusieurs ou un à un

- Exemple de traduction alternative : Association un à un



Traduction qui entraine des NULL (1)

Chambre-Simple (num, etage, id-client)

+ contraintes :

$\text{Chambre-Simple}[\text{id-client}] \subseteq \text{Client}[\text{id}]$

Traduction qui évite les NULL (1)

Occupation (num-chambre, id-client)

+ contraintes :

$\text{Occupation}[\text{num-chambre}] \subseteq \text{Chambre}[\text{num}]$

$\text{Occupation}[\text{id-client}] \subseteq \text{Client}[\text{id}]$

Associations un à plusieurs ou un à un

- Exemple de traduction alternative : Association un à un



Traduction qui entraine des NULL (2)

Client (id, nom, adresse, num-chambre)

+ contraintes :

$\text{Client}[\text{num-chambre}] \subseteq$
 $\text{Chambre-Simple}[\text{num}]$

Traduction qui évite les NULL (2)

Occupation (num-chambre, id-client)

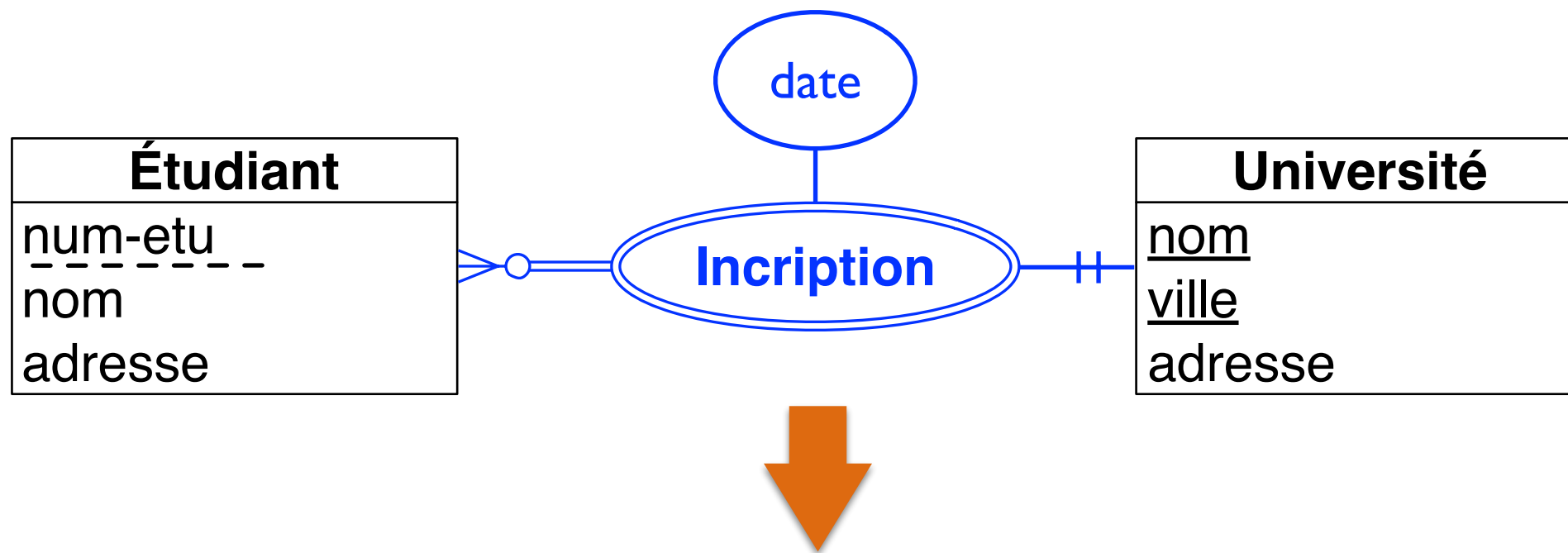
+ contraintes :

$\text{Occupation}[\text{num-chambre}] \subseteq \text{Chambre}[\text{num}]$

$\text{Occupation}[\text{id-client}] \subseteq \text{Client}[\text{id}]$

Associations un à plusieurs ou un à un

- Exemple d'association identifiante



Étudiant (num-etu, nom-univ, ville-univ, nom, adresse, date-inscription)

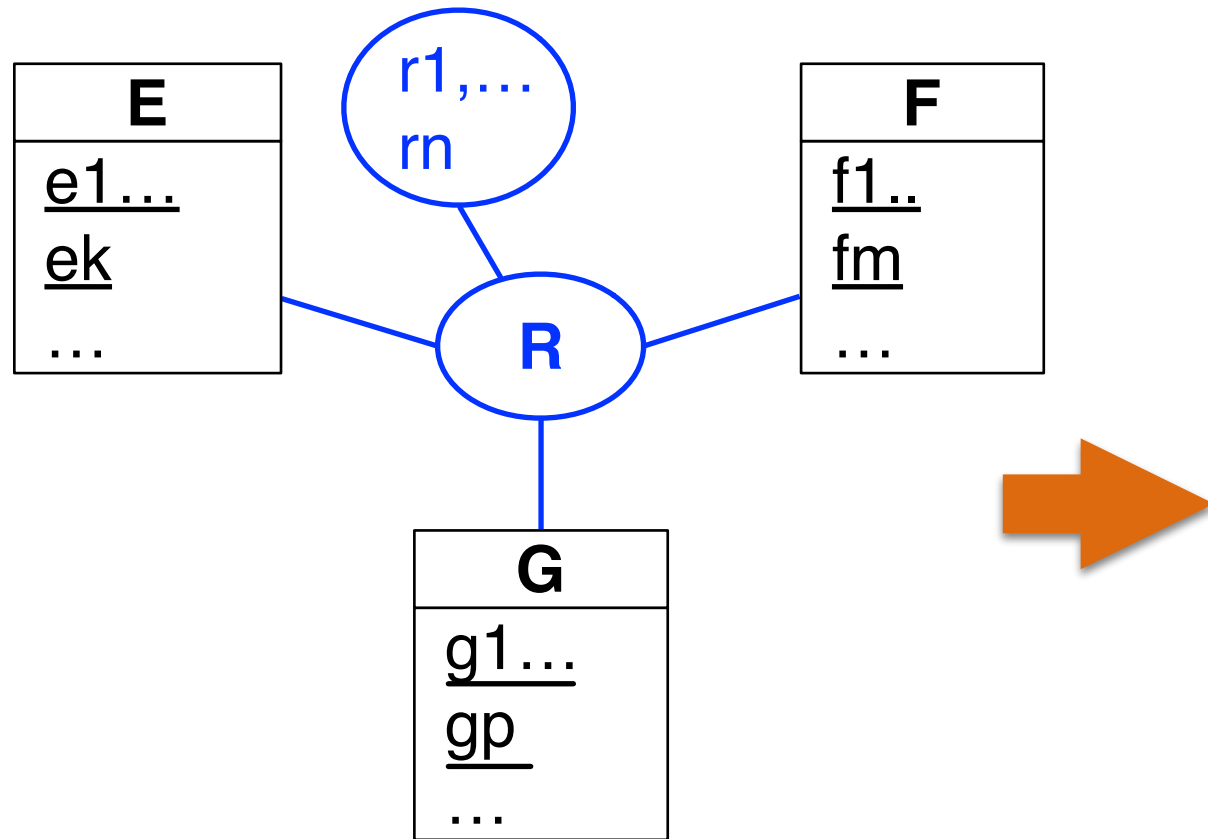
+ Contrainte de clef étrangère :

Étudiant [nom-univ, ville-univ] \subseteq Université[nom, ville]

+ Contrainte :

Étudiant [nom-univ, ville-univ] NOT NULL

Traduction des associations n-aires



$R(e_1, \dots, e_k, f_1, \dots, f_m, g_1, \dots, g_p, r_1, \dots, r_n)$

+ contraintes :

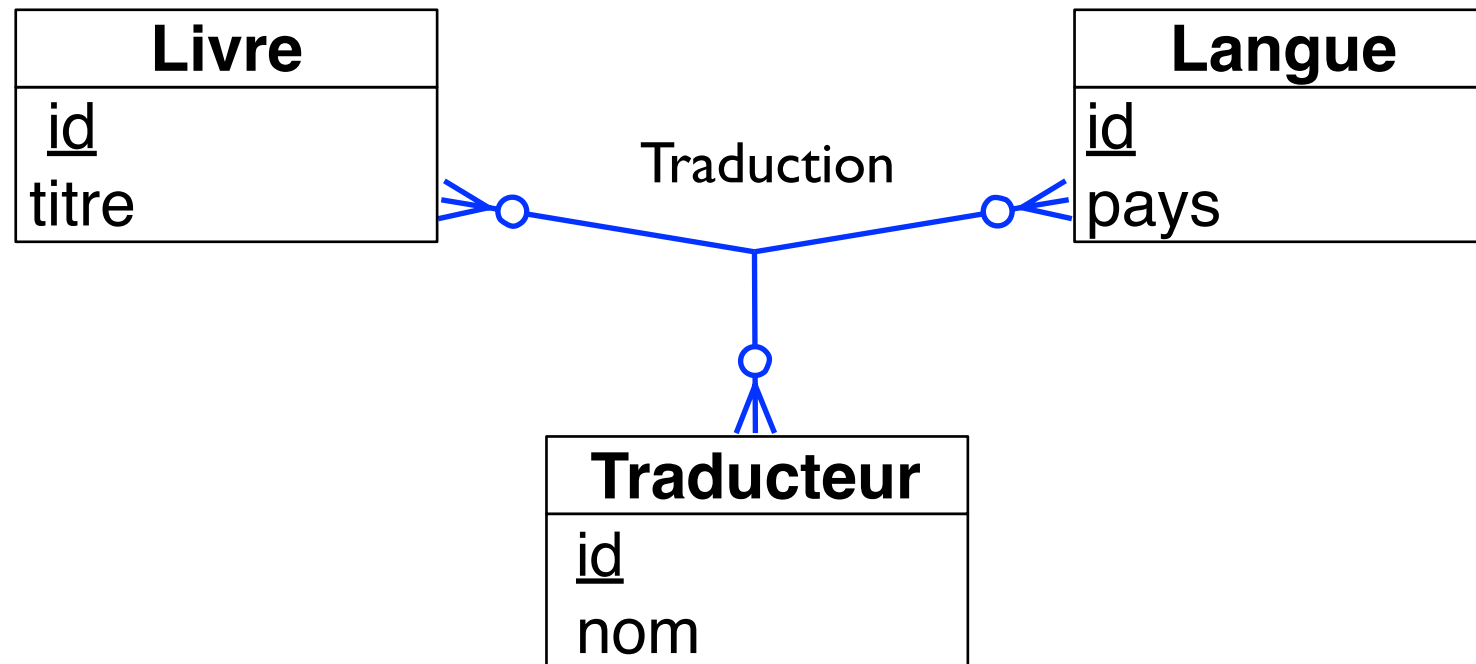
- $R[e_1, \dots, e_k] \subseteq E[e_1, \dots, e_k]$
- $R[f_1, \dots, f_m] \subseteq F[f_1, \dots, f_m]$
- $R[g_1, \dots, g_p] \subseteq G[g_1, \dots, g_p]$

• Clef de R :

- ▶ $(e_1, \dots, e_k, f_1, \dots, f_m, g_1, \dots, g_p)$ est une superclef de R , mais pas nécessairement minimale
- ▶ superclef minimale : dépend des contraintes de cardinalité mais aussi de possibles contraintes externes; à voir le cas par cas.
- ▶ Si la cardinalité maximale du côté de G est 1 alors $(e_1, \dots, e_k, f_1, \dots, f_m, g_1, \dots, g_p)$ n'est pas minimale puisque $(e_1, \dots, e_k, f_1, \dots, f_m)$ est aussi une superclef
 - mais $(e_1, \dots, e_k, f_1, \dots, f_m)$ pourrait aussi ne pas être minimale

Traduction des associations n-aires

- Exemple I



Traduction (id-livre, id-langue, id-traducteur)

+ contrantes :

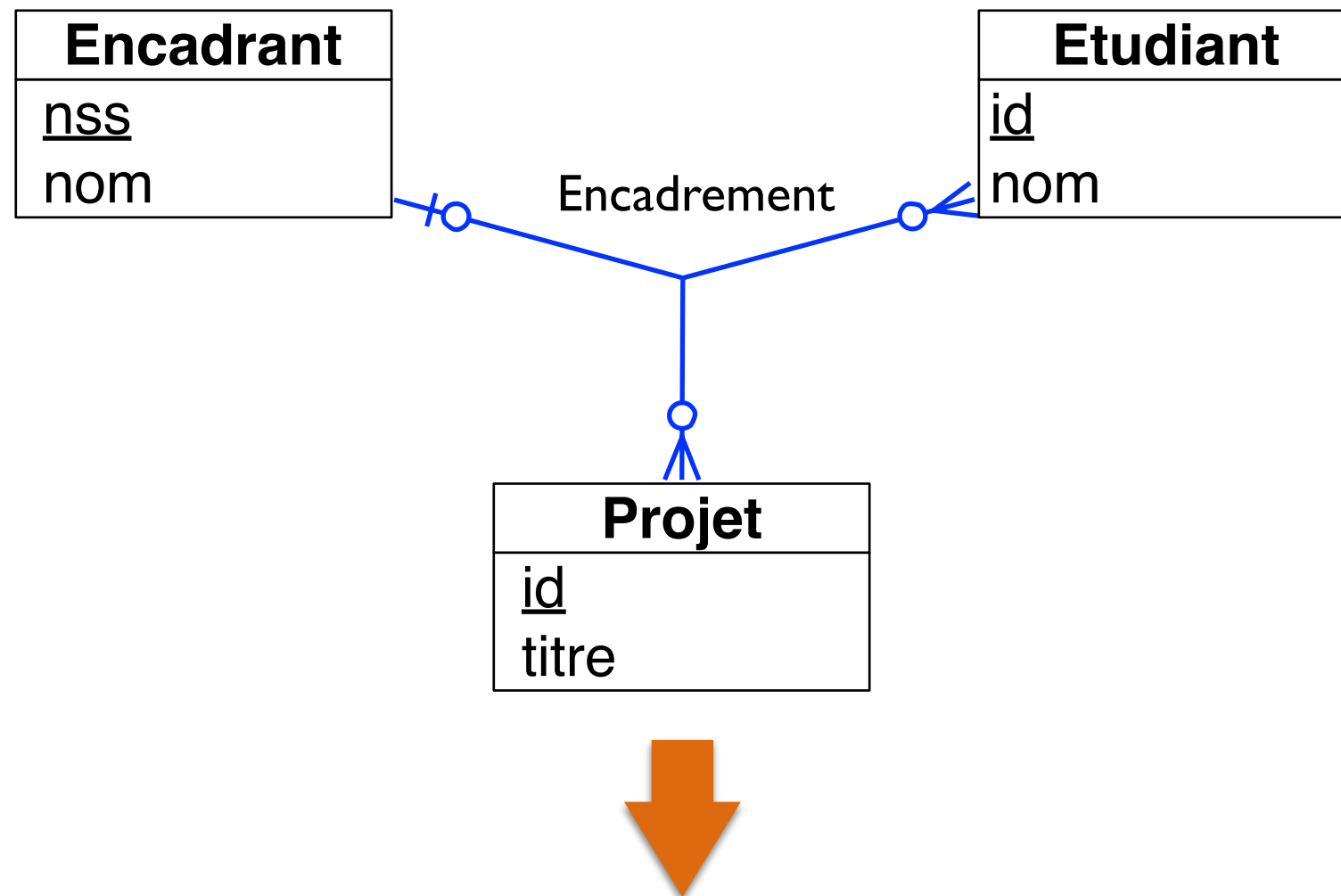
$\text{Traduction}[\text{id-livre}] \subseteq \text{Livre}[\text{id}]$

$\text{Traduction}[\text{id-langue}] \subseteq \text{Langue}[\text{id}]$

$\text{Traduction}[\text{id-traducteur}] \subseteq \text{Traducteur}[\text{id}]$

Traduction des associations n-aires

- Exemple 2



Encadrement (id-etu, id-projet, nss-enc)

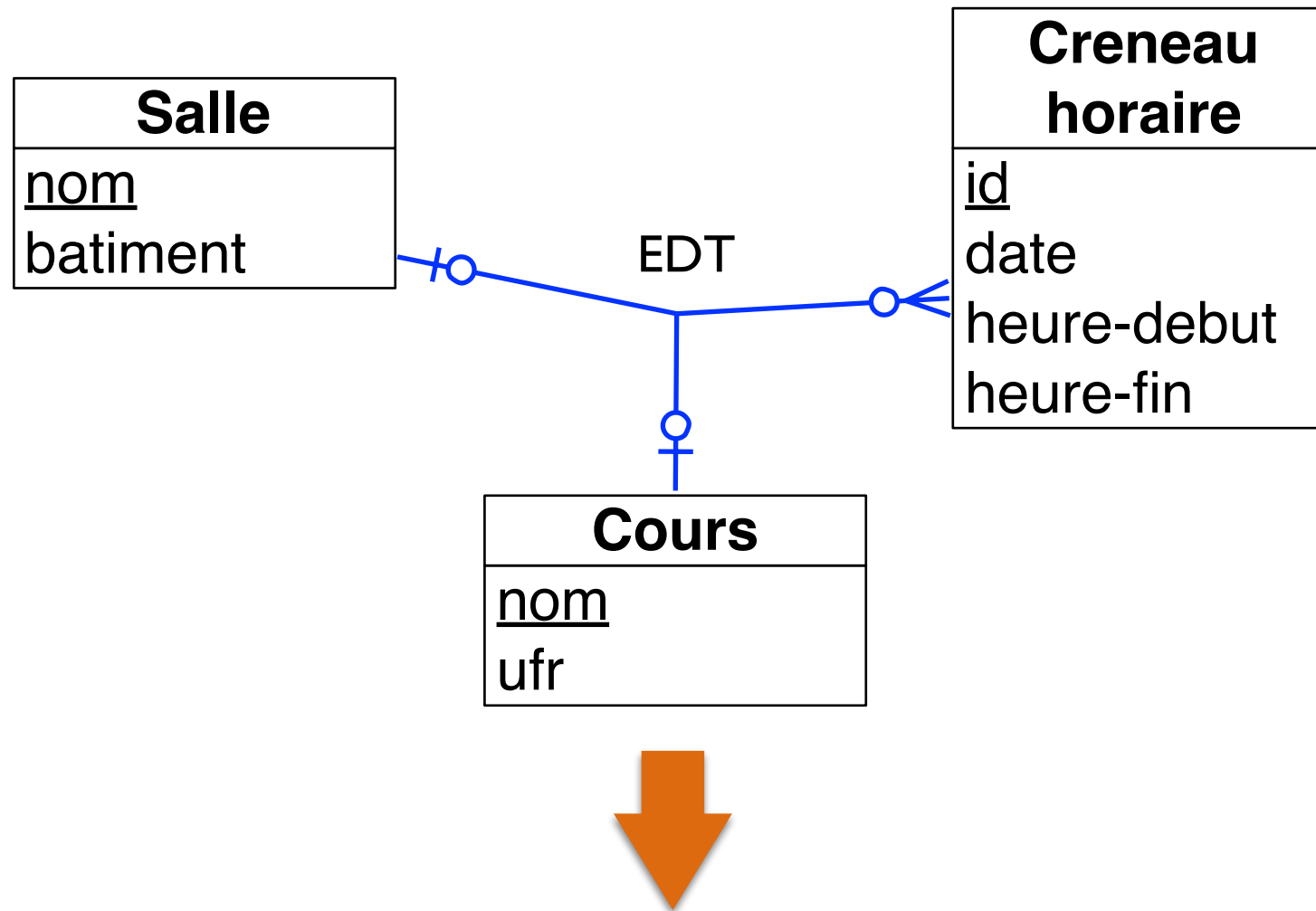
Encadrement [id-etu] \subseteq Etudiant[id]

Encadrement [id-projet] \subseteq Projet[id]

Encadrement [nss-enc] \subseteq Encadrant [id]

Traduction des associations n-aires

- Exemple 3



EDT (nom-salle, id-creneau, nom-cours)

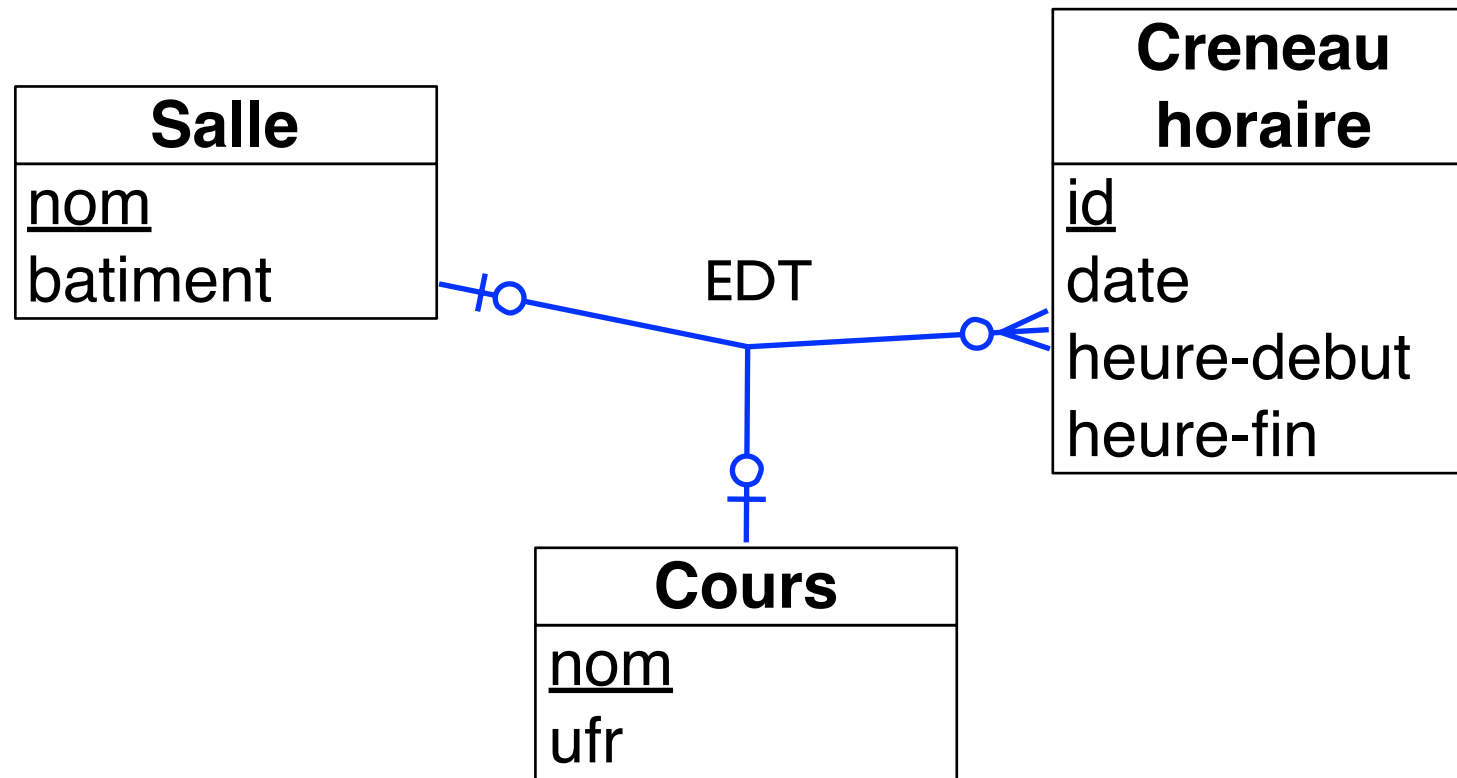
EDT [nom-salle] \subseteq Salle[nom]

EDT [id-creneau] \subseteq Creneau-horaire [id]

EDT [nom-cours] \subseteq Cours [nom]

Traduction des associations n-aires

- Exemple 3



Alternative

EDT (nom-salle, id-creneau, nom-cours)

EDT [nom-salle] \subseteq Salle[nom]

EDT [id-creneau] \subseteq Creneau-horaire [id]

EDT [nom-cours] \subseteq Cours [nom]

Traduction des associations n-aires

- Exemple 4

| Vendeur |
|-----------|
| <u>id</u> |
| nom |
| ville |

| Voiture |
|---------------|
| <u>plaque</u> |
| modele |
| prix |

Achat



Contraintes externes

chaque voiture a au plus un vendeur et au plus un client acheteur

(Achat de voitures neuves)



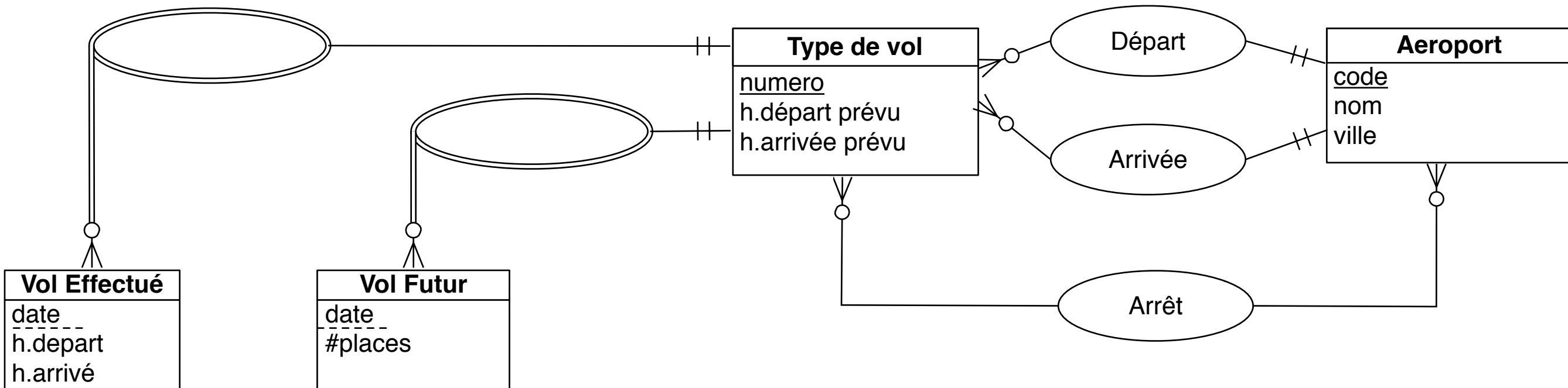
Achat (plaque-voiture, id-vendeur, nss-client)

Achat [plaque-voiture] \subseteq Voiture [plaque]

Achat [nss-client] \subseteq Client [nss]

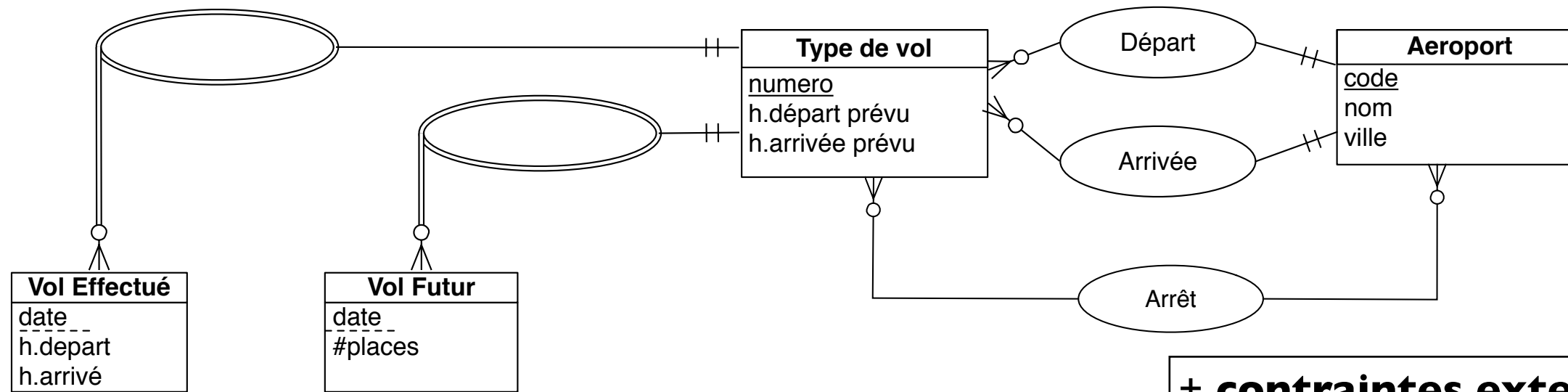
Achat [id-vendeur] \subseteq Vendeur [id]

Un exemple complet de traduction



+ **contrainte externe** : la date de chaque vol effectué est \leq à la date de chaque vol futur

Un exemple complet de traduction



+ **contraintes externes** :

* la date de chaque vol effectué est
≤ à la date de chaque vol futur

Diagramme E/R

Schéma relationnel

Aeroport (code, nom, ville)

Arret (num_vol, code_aero)

Vol_Effectue (num_vol, date, h_dep, h_arr)

Vol_Futur (num_vol, date, #places)

Type_Vol (numero, aero_dep, aero_arr,
hp_dep, hp_arr)

Contraintes de clefs étrangère :

- R [num_vol] ⊆ Type_Vol [numero]
(pour R = Vol_Effectué, Vol_Futur, Arret)
- Arret [code_aero] ⊆ Aeroport [code]
- Type_Vol [aero_dep] ⊆ Aeroport [code]
- Type_Vol [aero_arr] ⊆ Aeroport [code]

Contraintes d'unicité : Aeroport (nom, ville)

Autres contraintes : Check ou assertions

Outils gratuits pour dessiner des diagrammes E/R

- [Dia](#)
- [Calligra Flow](#)
- Beaucoup d'autres outils permettent également la transformation automatique ER → Relationnel (MySQL Workbench, ER2SQL, ...)
 - ▶ Attention : ils ont leur propre syntaxe et sémantique pour ER
 - ▶ on obtient souvent un meilleur modèle quand on traduit “à la main”