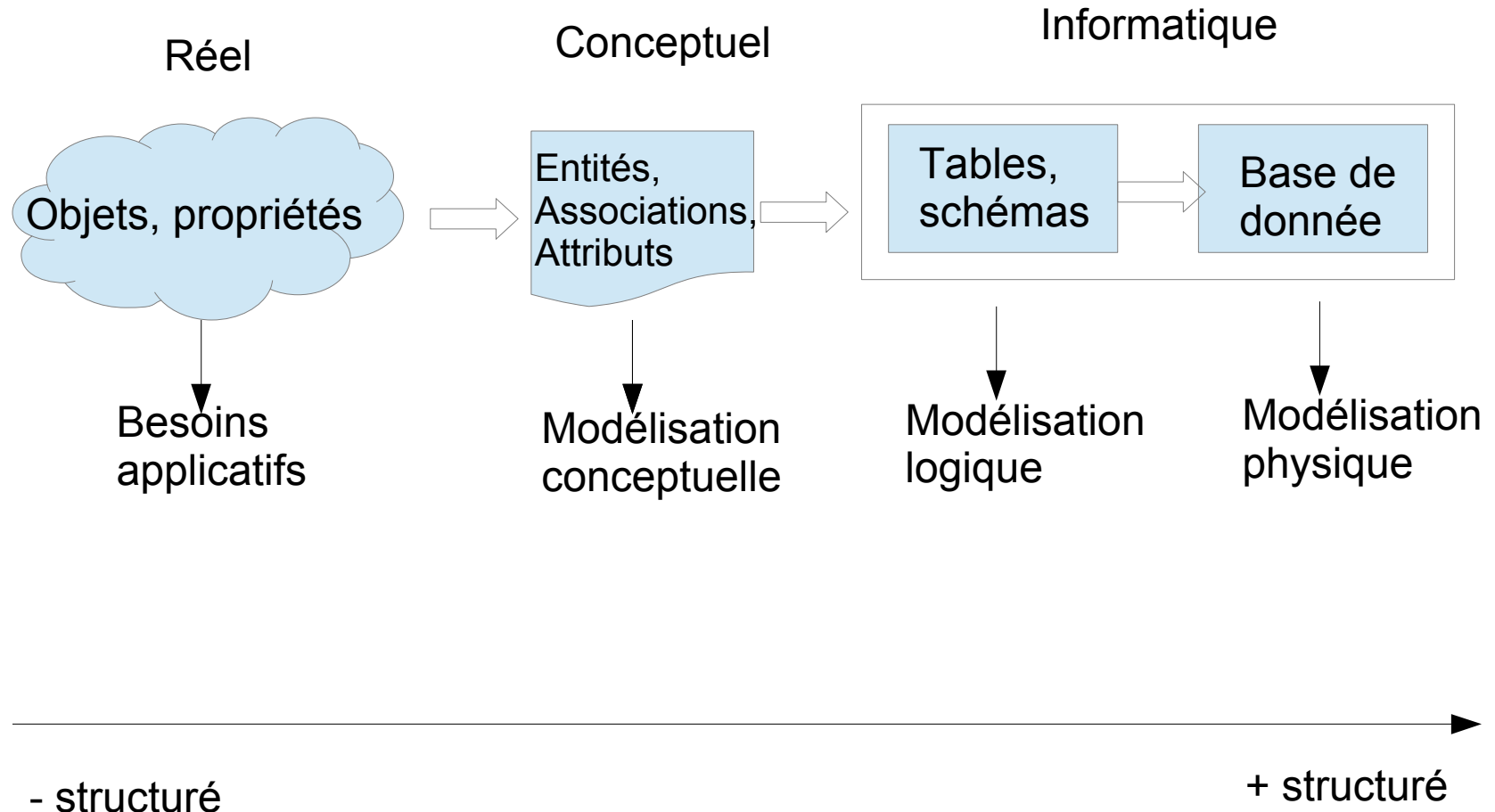


Introduction aux Bases de données

Cours 2 : Modèle relationnel- Passage E/A vers le modèle relationnel

LU2IN009

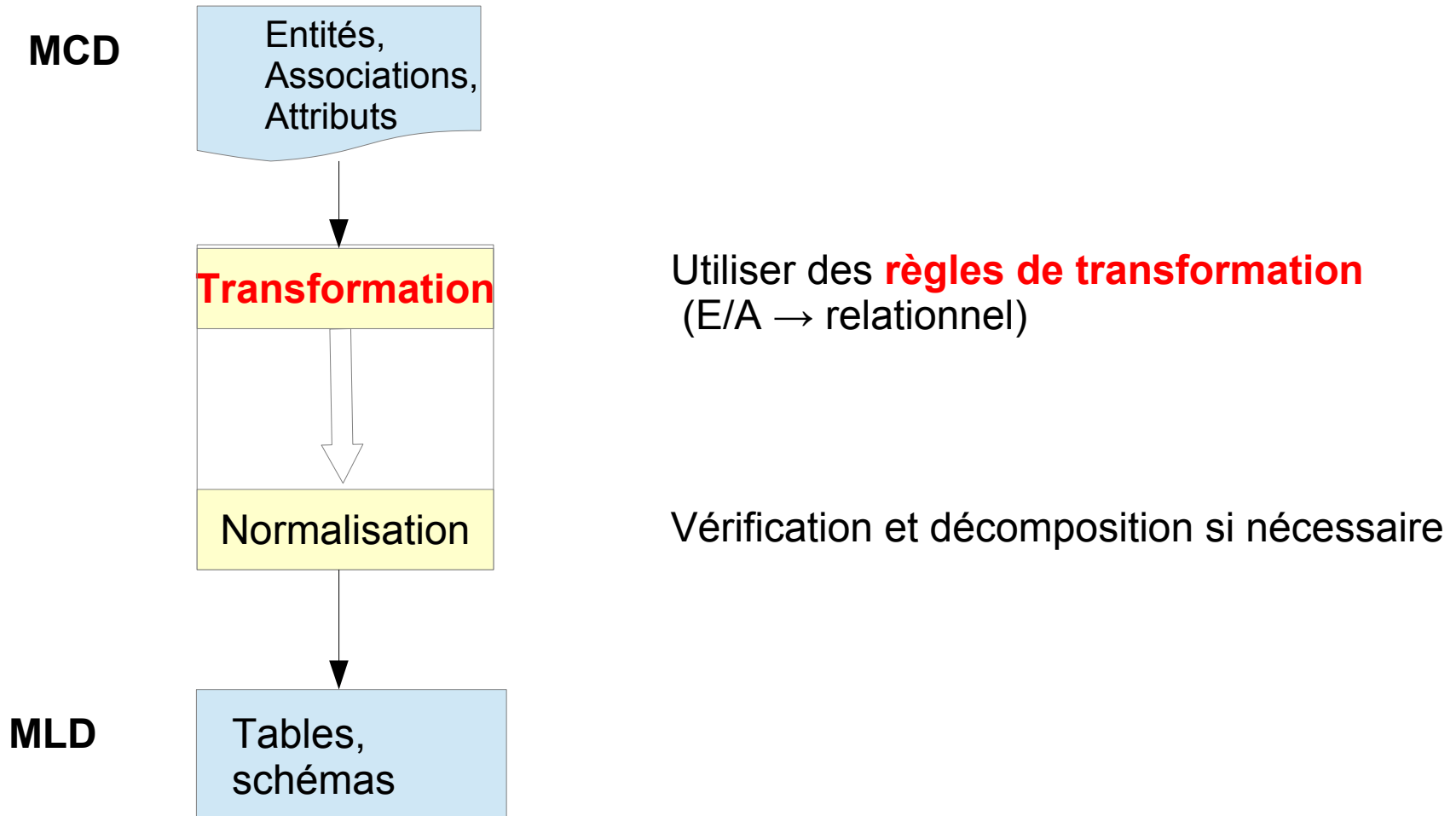
Rappel: construction d'une BD



Rappel : construction d'une BD

- **Modèle conceptuel des données (MCD):** description de l'application dans un langage de haut niveau (Entité-Association) qui ne tient pas compte du SGBD
- **Modèle logique des données (MLD):** description des données dans un formalisme compatible avec un SGBD (schemas, tables, colonnes, clés primaires et étrangères)
- **Modèle physique des données (MPD):** implémentation du modèle logique dans le SGBD (affiner le MLD en un schéma pour un SGBD spécifique), utilisation de SQL (create TABLE..), types des attributs, index, dénormalisation

Transformation MCD \rightarrow MLD



Le modèle relationnel

Basé sur la définition et la manipulation de **relations**:

- Données: organisées dans des relations (**perçues par l'utilisateur comme tables**)
- *Table (relation)* = **ensemble** de **n-uplets** avec mêmes attributs, représentée sous la forme d'un tableau à deux dimensions:
 - Chaque *colonne* correspond à un *attribut* A_i
 - Chaque *ligne* (tuple, n-uplet) est une séquence de n valeurs atomiques (v_1, \dots, v_n) où chaque v_i est la valeur (nombre, chaîne de caractères, date, ...) d'un attribut A_i ou **NULL** (absence de valeur).

Relation et attribut

La table (relation) **Etudiants** = ensemble de lignes (tuples ou n-uplets)

*{1753, Smith, Joe, 11 CP NYC}, {2410, Hersh, Dan, 22 Rd NJ},
{0148, Clay, Maissa, NULL}*

Les attributs ou colonnes



<u>matricule</u>	nom	prénom	adresse
1753	Smith	Joe	11 CP NYC
2410	Hersh	Dan	22 Rd NJ
0148	Clay	Maissa	NULL

Les lignes,
n-uplets ou tuples



Valeur de l'attribut adresse
non spécifiée (pas connue)



Attributs

Attribut : un nom qui décrit une propriété

- *Exemple* : les propriétés matricule, nom, prénom, adresse d'un étudiant

Domaine d'un attribut :

- l'ensemble des *valeurs atomiques* de l'attribut
- *Exemple* : **matricule** $\in \{'1753', '2410', '0148'\}$, **adresse** est une chaîne de 20 caractères

Valeur NULL : l'absence temporaire de valeur (inconnu) ou l'inapplicabilité d'une valeur pour un attribut dans un tuple

Les attributs ou colonnes



<u>matricule</u>	nom	prénom	adresse
1753	Smith	Joe	11 CP NYC
2410	Hersh	Dan	22 Rd NJ
0148	Clay	Maissa	NULL

Les lignes,
n-uplets ou tuples



Sous-ensemble du domaine de valeurs de prénom

Clés

- Plus petit sous-ensemble d'attributs qui identifient chaque ligne de *manière unique*.
→ il n'existe pas deux lignes avec les mêmes valeurs pour l'ensemble de ces attributs
- Exemple: l'attribut matricule pour la relation Étudiant

Chaque ligne a une valeur de matricule différente

<u>matricule</u>	nom	prénom	dateNaiss	adresse
1753	Smith	Joe	1992-01-12	11 CP NYC
9832	Smith	Dan	1989-04-03	22 Rd NJ
4755	Smith	Joe	1994-11-29	7 HW NJ
6842	Roy	Ian	1992-05-18	NULL

- Est-ce que les ensembles suivants peuvent être des clés?
 - $\{nom, prénom\}$
 - $\{nom, prénom, dateNaiss\}$

Clé primaire et clé candidate

- Chaque relation doit posséder *au moins une clé*
- Une relation a *au moins une clé candidate* (*chacun des attributs est renseigné, pas de valeurs NULL*)
 - on choisit *une seule* comme *clé primaire*
- Exemple: matricule est clé primaire, {*nom, prénom, dateNaiss*} est clé candidate

Chaque ligne a une valeur de matricule et de l'ensemble {*nom, prénom, dateNaiss*} différente

<u>matricule</u>	nom	prénom	dateNaiss	adresse
1753	Smith	Joe	1992-01-12	11 CP NYC
9832	Smith	Dan	1989-04-03	22 Rd NJ
4755	Smith	Joe	1994-11-29	7 HW NJ
6842	Roy	Ian	1992-05-18	NULL

- Notation: la **clé primaire** est soulignée, les **clés candidates** sont mentionnées en langage naturel

Clé primaire et clé candidate

■ *Contraintes de l'application:*

- Chaque module doit avoir un code différent
- Il n'existe pas deux modules avec le même intitulé pour un niveau donné
- Un enseignant ne peut pas être responsable de plus d'un module par niveau

La table **Modules**

code	intitulé	niveau	responsable
2I009	Bases de Données	L2	Smith
MI005	Bases de Données	M1	Roy
3I004	Programmation	L3	Smith

■ Quelles sont les clés candidates?

■ Quelle est la clé primaire?

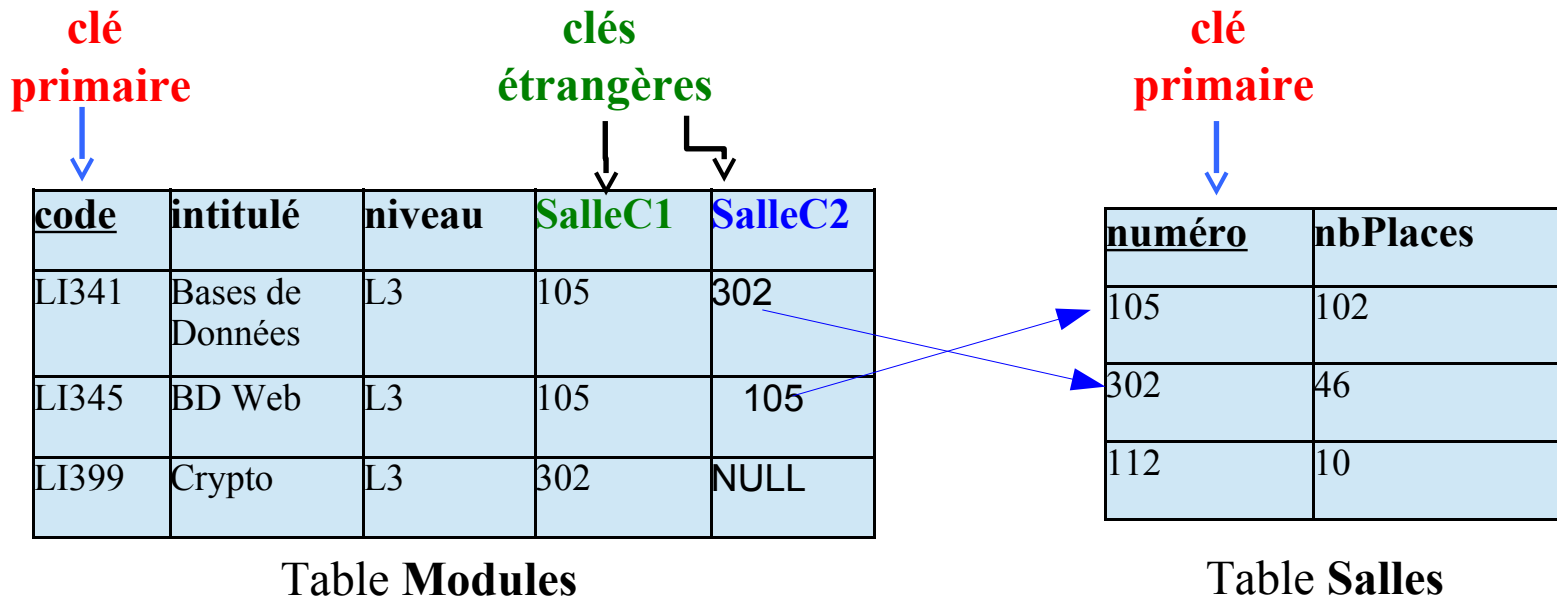
■ Les tuples suivants peuvent-ils exister dans la table **Modules**?

- {'2I009', 'BD Web', 'L3', 'Roy'}
- {'MI006', 'Crypto', 'M1', 'Roy'}
- {'3I009', 'Bases de Données', 'L2', 'Roy'}

Clés étrangères

Clé étrangère:

- *sous-ensemble d'attributs* dont les valeurs proviennent des clés candidates de la même table ou d'autre table
- mécanisme de référencement des n-uplets

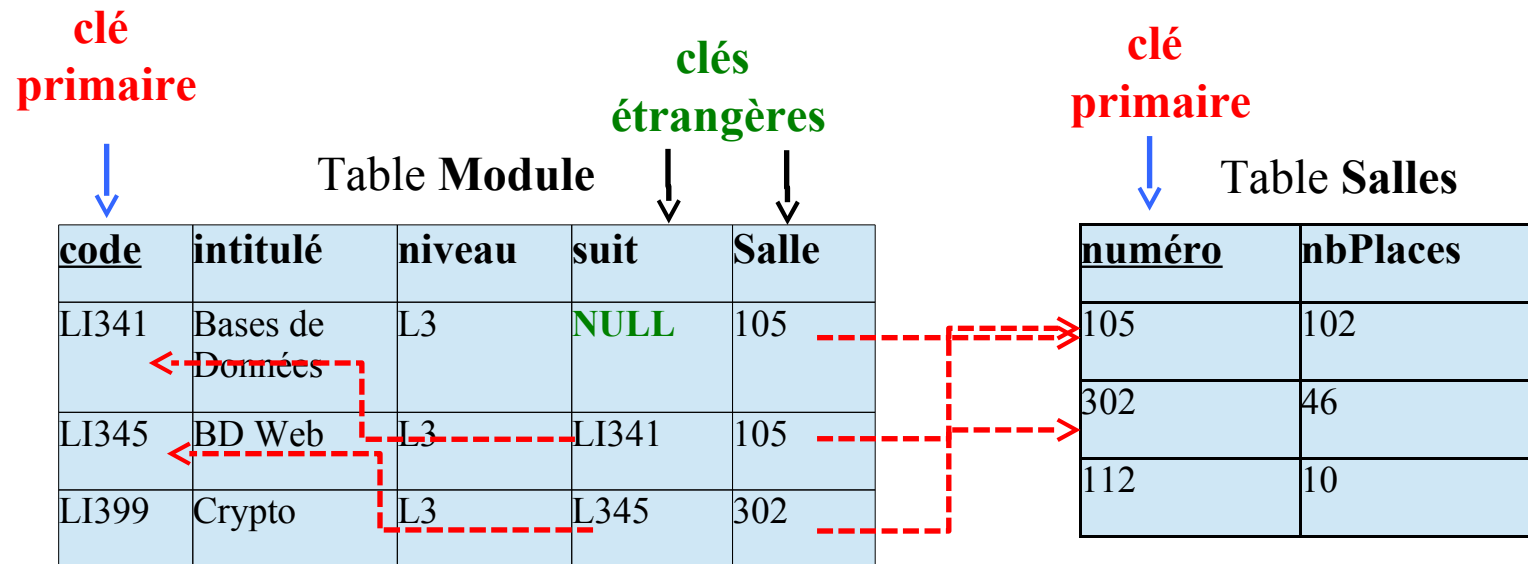


- Le nom d'une clé étrangère n'est pas nécessairement le même que celui de la clé référencée

Clés étrangères : autre exemple

■ La table **Module** contient *deux clés étrangères*:

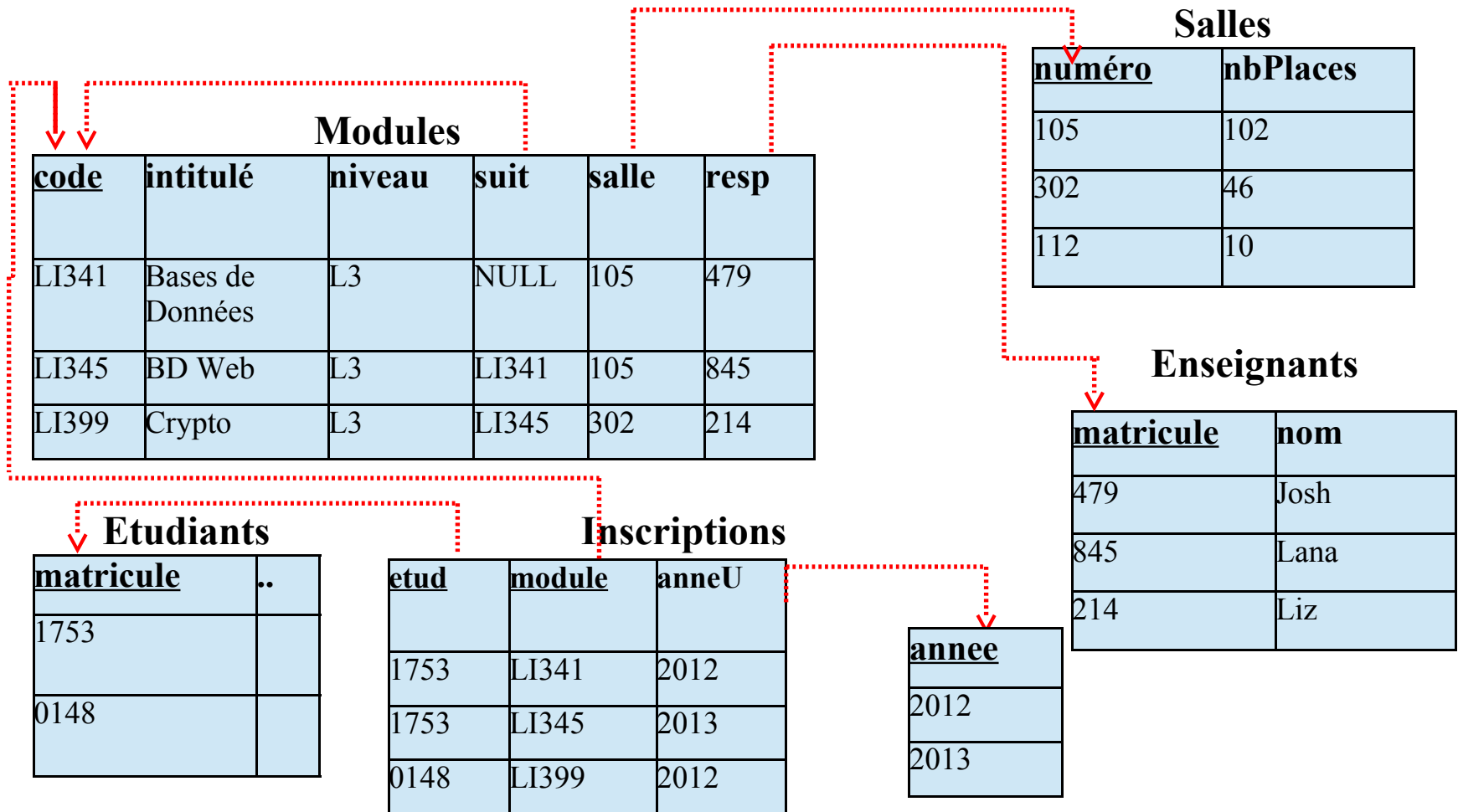
- **salle** : fait référence à l'attribut numéro de la table **Salle**
- **suit**: fait référence à l'attribut code de la table **Module**



■ Valeurs permises pour une clé étrangère:

- Valeurs *déjà existantes* des clés candidates ou *NULL* (inconnu)

Clés étrangères : exemple complet



■ Des clés étrangères peuvent composer la clé candidate/primaire d'une relation:

■ Exemple: la clé {**etud**, **module**} de la table **Inscriptions**

Schéma relationnel : aperçu

Schéma d'une relation:

- Nom de la relation + Liste de ses attributs avec leur domaines (nombre, chaîne de caractères, date...) + Clés des tables et contraintes d'intégrité (cf. partie III)

Instance d'une relation:

- Ensemble des n-uplets de la table

Schéma d'une base de données:

- Ensemble des schémas des relations qui la composent

1753	Smith	Joe	11 CP NYC				
2410	Hersh	Dan	22 Rd NJ				
0148	Clay	Maissa	7 HW NJ				

<u>matricule</u>	nom	prénom	adresse
numérique	caractères	caractères	caractères

Clé primaire

Instance Etudiants={3 n-uplets}

Schéma Etudiants

Notations

- **Schéma d'une BD** = ensemble des schémas de relation
 $S = \{R1, R2, ..., Rn\}$ où Ri est un schéma de relation

- **Schéma de relation** = ensemble des attributs
avec leurs domaines respectifs et les contraintes
 $R(\underline{A1}:D1, A2:D2, ..., Am:Dm)$, $A1$ est clé primaire: relation d'arité m

- **Exemple:**

 $\text{Etudiants}(\underline{\text{matricule}} : \text{Number}, \text{nom} : \text{Varchar}, \text{prenom} : \text{Varchar}, \text{adresse} : \text{Varchar})$

 $\text{Modules}(\underline{\text{code}} : \text{Number}, \text{intitule} : \text{Varchar}, \text{niveau} : \text{Varchar}, \text{salle} : \text{Number})$

 $\text{Salles}(\underline{\text{numero}} : \text{Number}, \text{capacite} : \text{Number})$

Schéma de la Base de Données: {Etudiants, Modules, Salle}

- ★ Number =numérique, Varchar =chaîne de caractères de longueur variable

Simplification des notations

■ *Convention de notation:*

- Clé primaire : soulignement
- Clés étrangères : astérisque et désignation de la table référencée
- On omet les domaines des attributs

■ *Exemple:*

Etudiants(matricule, nom, prenom, adresse, collaborateur*)

—► collaborateur fait référence à (la clé primaire de) Etudiants

Modules(code, intitule, niveau, salle*)

—► salle fait référence à Salles

Salles(numero, capacite, précédente *, suivante*)

—► précédente et suivante font chacune référence à Salles

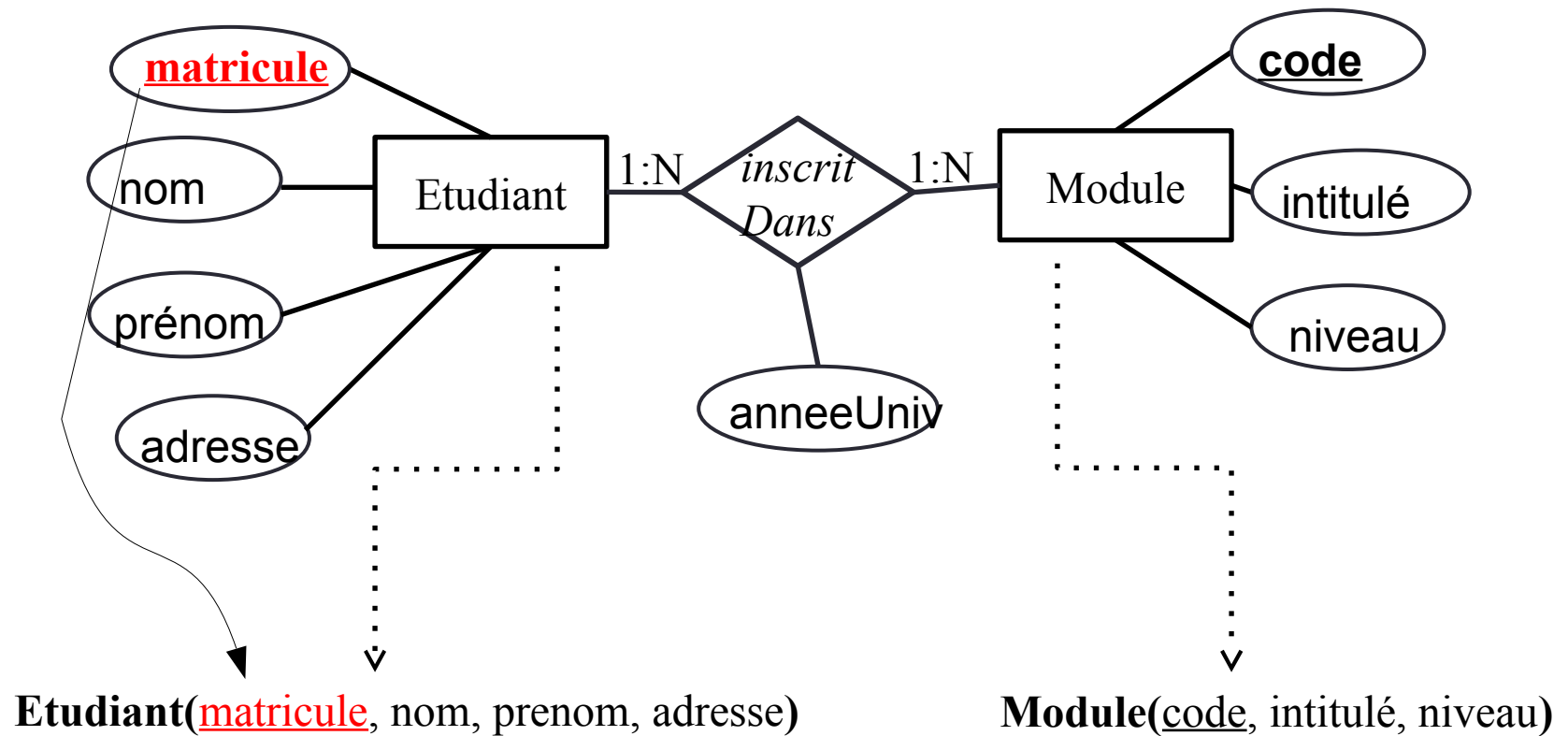
Avantages du modèle relationnel

- Proche de la réalité et simple
 - La plupart des entités du monde réel partagent les mêmes attributs
 - Familiarité des utilisateurs avec les tableaux
- Repose sur des fondements solides
 - Théorie des Ensembles
 - Logique du Premier Ordre
- Doté de langages de requêtes puissant
 - Algèbre relationnelle, Calcul des Prédicats
 - SQL (Structured Query Language)

Traduction E/A – modèle relationnel

- Règles de transformation des Entités:
 - Une entité *devient* une relation
 - Les attributs d'une entité *deviennent* les attributs de la relation
 - Tout ensemble d'attributs identifiant une entité *devient* la clé primaire de la relation
 - Tout ensemble d'attributs susceptibles de jouer le rôle d'identifiant d'entité *devient* clé candidate de la relation

Exemple

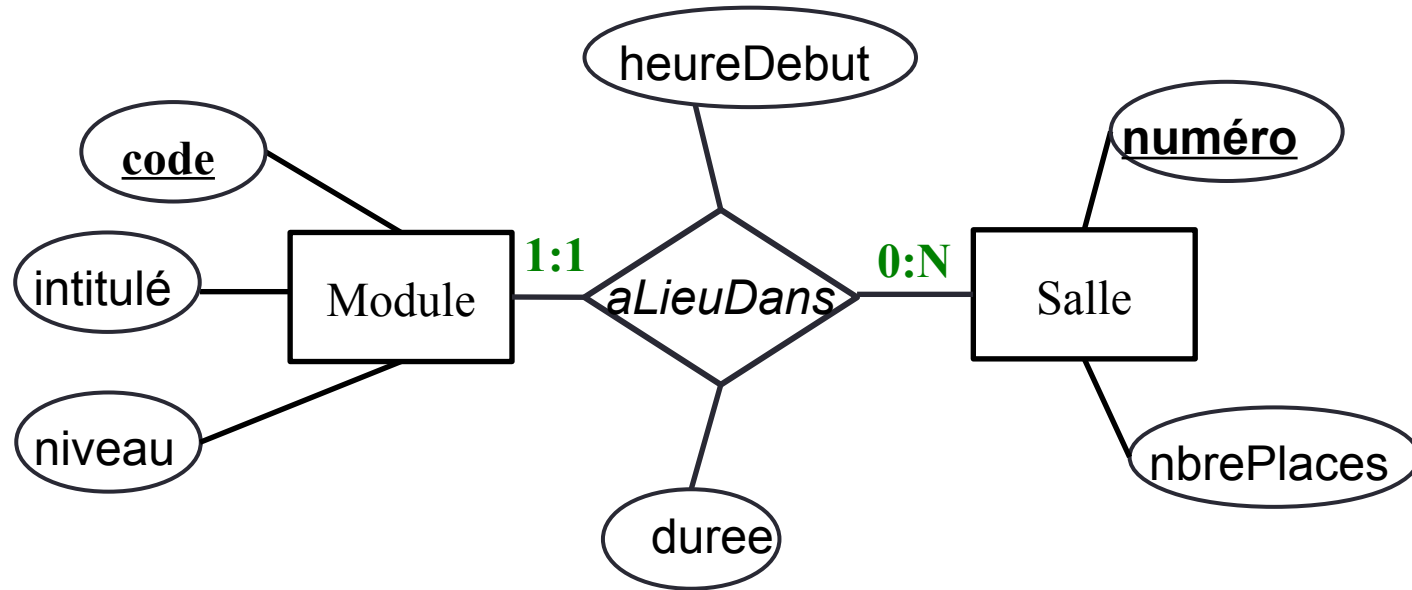


{nom, prenom} est second identifiant de l'entité
Étudiant → {nom, prenom} est clé candidate
dans la relation **Etudiant**

Règles de transformation d'une association

- Traitement différent en fonction des cardinalités:
- **Cas 1** : Association n-aire avec cardinalités (x,y) et au moins une cardinalité $(1,1)$, où (x, y) peut être: $(0,n)$, $(1,n)$, $(0,1)$
 - La table obtenue pour l'entité correspondante à la cardinalité $(1,1)$ contiendra aussi les attributs de l'association → si plusieurs cardinalités $(1,1)$, modifier seulement la table correspondante à l'une de ces entités
- **Cas 2** : Association n-aire avec cardinalités (x, n) ($x: \{0,1\}$)
 - Créer une table ayant comme attributs tous les attributs de l'association et comme identifiants les identifiants de l'association
- **Cas 3** : Associations n-aire avec cardinalités (x, y) et $(0,1)$, où $(x, y) \neq (1,1)$, (x, y) peut être $\{(0, n), (1, n), (0,1)\}$. Transformations possibles:
 - Similaire à celle pour le Cas 1 → problème: valeurs NULL possibles
 - Similaire à celle pour le Cas 2 → préférable car élimination des valeurs NULL

Cas1: Association n-aire (x, y) et (1,1)



Rappel : attributs de l'association {code, numéro, heureDebut, durée}

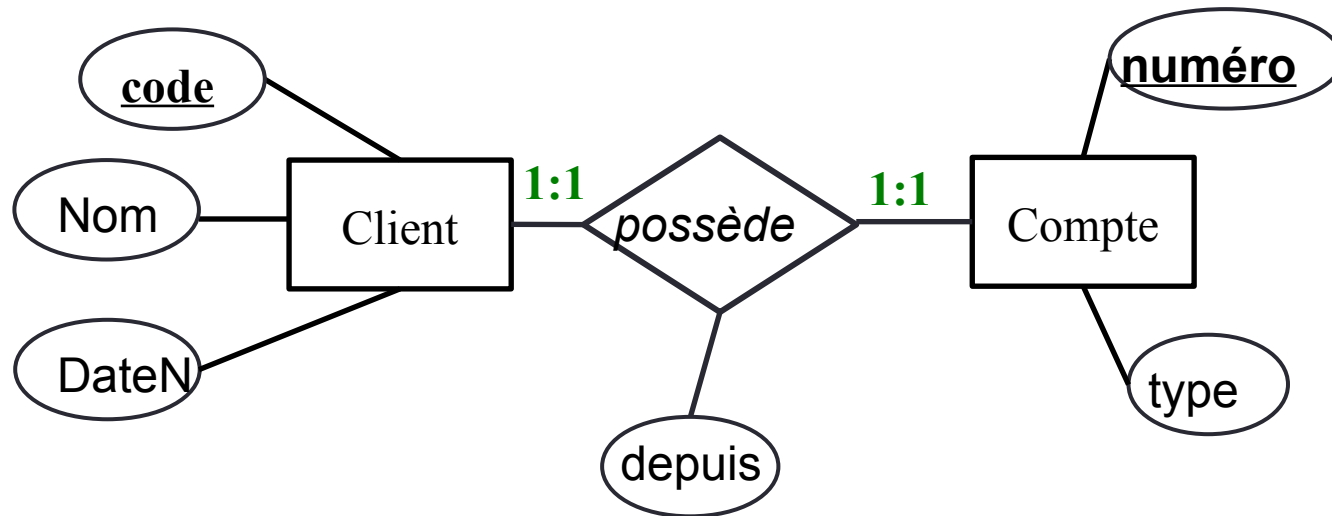
Module(code, intitulé, niveau, numéroSalle*, heureDebut, duree)

→ NuméroSalle référence numéro de la table Salle

Salle(numéro, nbrePlaces)

Cas1: Association n-aire (x, y) et (1,1)

- Si plusieurs cardinalités 1:1, modifier la table de seulement une des relations correspondantes



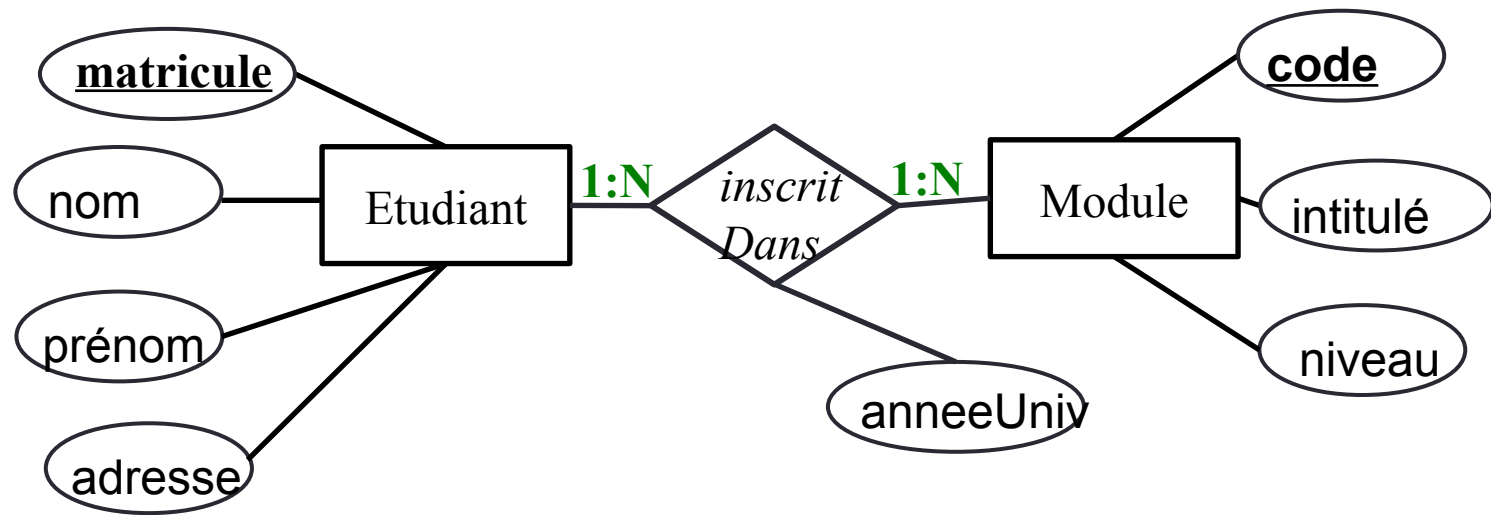
Compte(numéro, type)

Client(code, Nom, DateN, NumeroCompte*, depuis)

→ NumeroCompte référence numéro de la table Compte

- *La table Compte peut être enlevée si l'entité Compte n'est associée à aucune autre entité, tous ses attributs seront stockés dans la relation Client*

Cas 2 : Association n-aire (x, n)

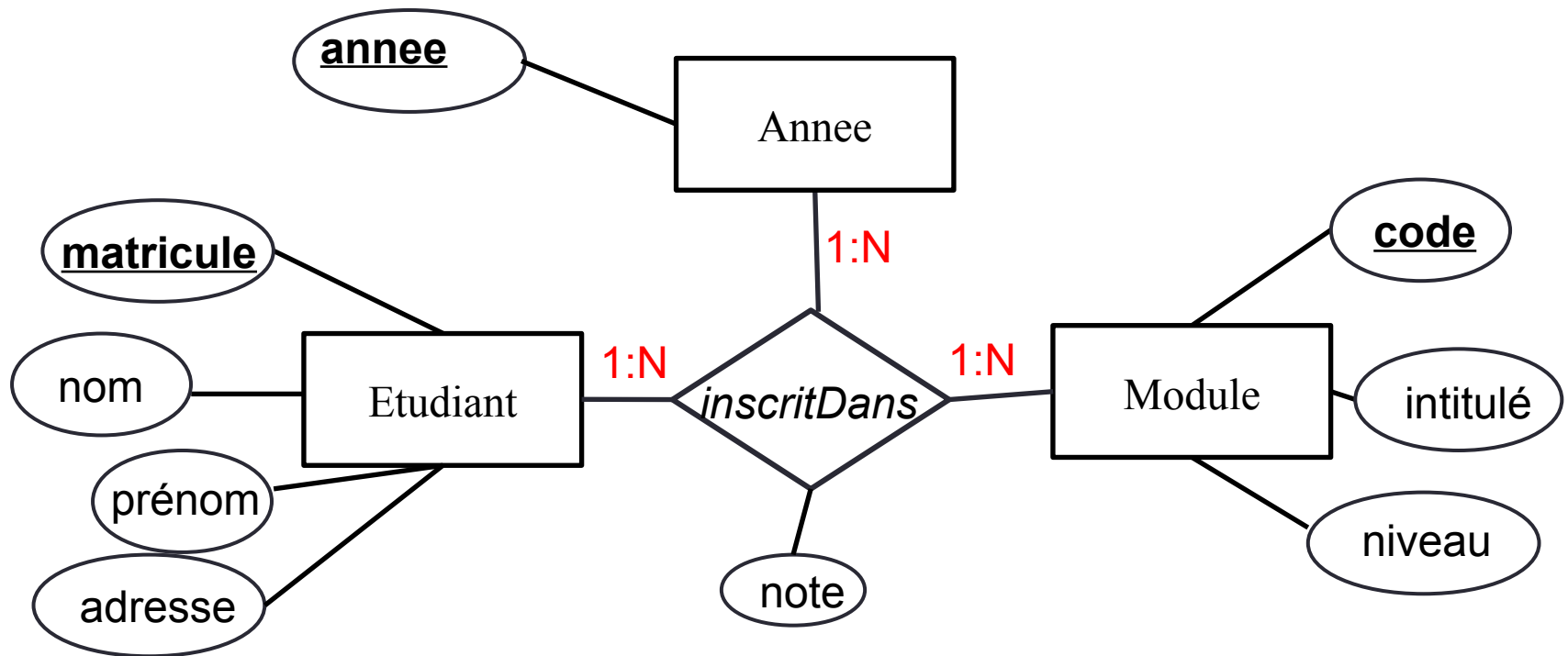


Etudiant(matricule, nom, prenom, adresse)

Module(code, intitulé, niveau)

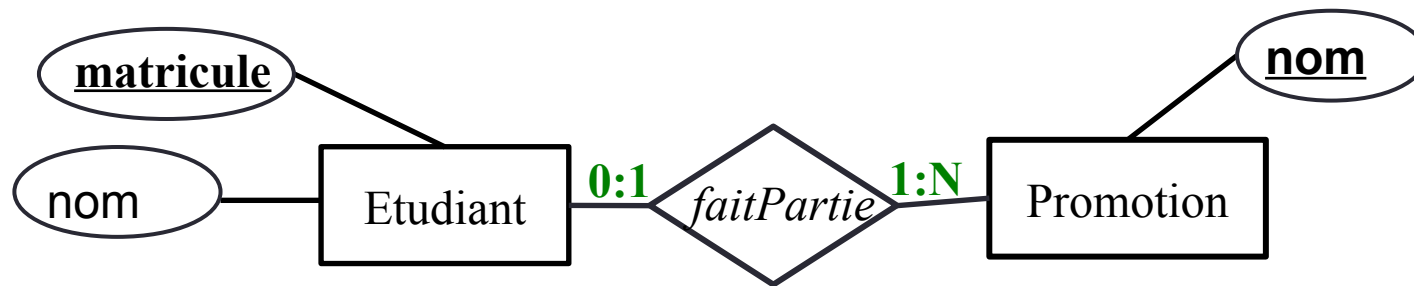
Inscriptions(matricule*, code*, anneeUniv)

Exercice : traduction d'une association ternaire (x,n)



Cas 3 : Association n-aire (x, y) et (0,1)

■ Première possibilité : même transformation que pour le Cas 1



Etudiant(matricule, ... , promo*)

Promotion(nom)

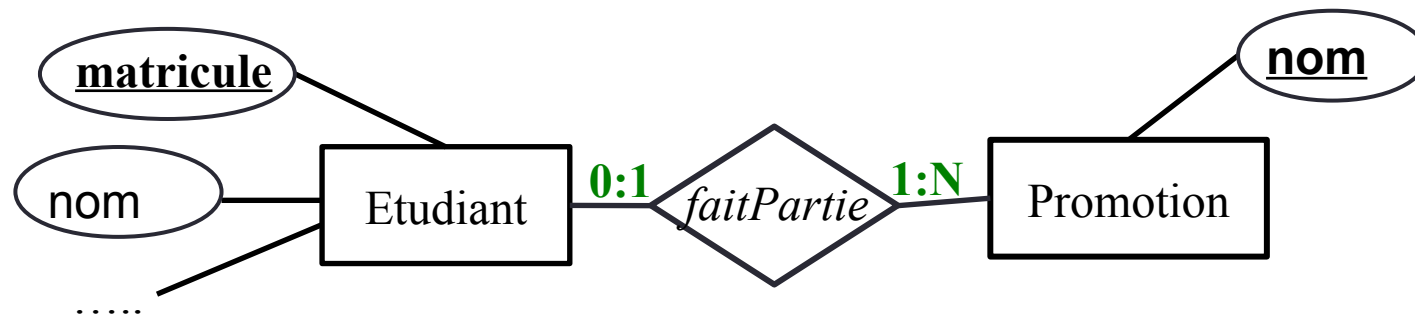
→ promo clé étrangère fait référence à nom de Promotion

<u>matricule</u>	nom	promo
1753	Smith		M.Curie
2410	Hersh		NULL

<u>nom</u>
M. Curie
J. Fourier

■ *Problème* : possibilité d'avoir des valeurs NULL

Cas 3 : Association n-aire (x, y) et (0,1)



Etudiant(matricule, ... , promo*)

Promotion(nom)

Etud-Prom0(matEtu*, promo*)

<u>matricule</u>	nom
1753	Smith	
2410	Hersh	

<u>matEtu</u>	promo
1753	M. Curie

<u>nom</u>
M. Curie
J. Fourier

- *Avantage* : stocker uniquement les paires (étudiant, promo) qui existent
→ (évite les valeurs NULL)

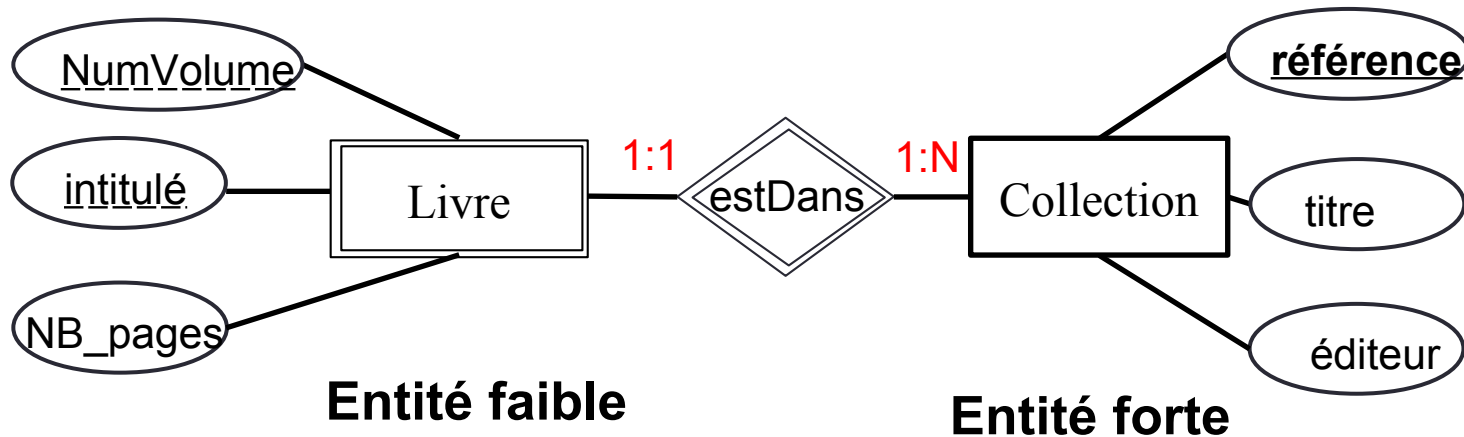
Règle de transformation des entités faibles

■ Transformation des entités faibles:

- Créer une relation pour l'entité faible
- Clé de la relation correspondante à une entité faible = concaténation de l'identifiant de l'entité faible et celui de l'entité dont elle dépend
→ les attributs constituant l'identifiant de l'entité forte constituent une clé étrangère (qui fait aussi partie de la clé primaire)

→ Transformation similaire à celle pour le Cas 1, en incluant en plus la clé étrangère dans la clé primaire

Transformation des entités faibles



Collection (référence, titre, éditeur)

Livre (référence*, numVolume, intitulé, NB_pages)

→ référence est une clé étrangère qui fait référence à la clé primaire de Collection

Clé primaire Livre: {référence, numVolume, intitulé}

Règle de transformation de la spécialisation

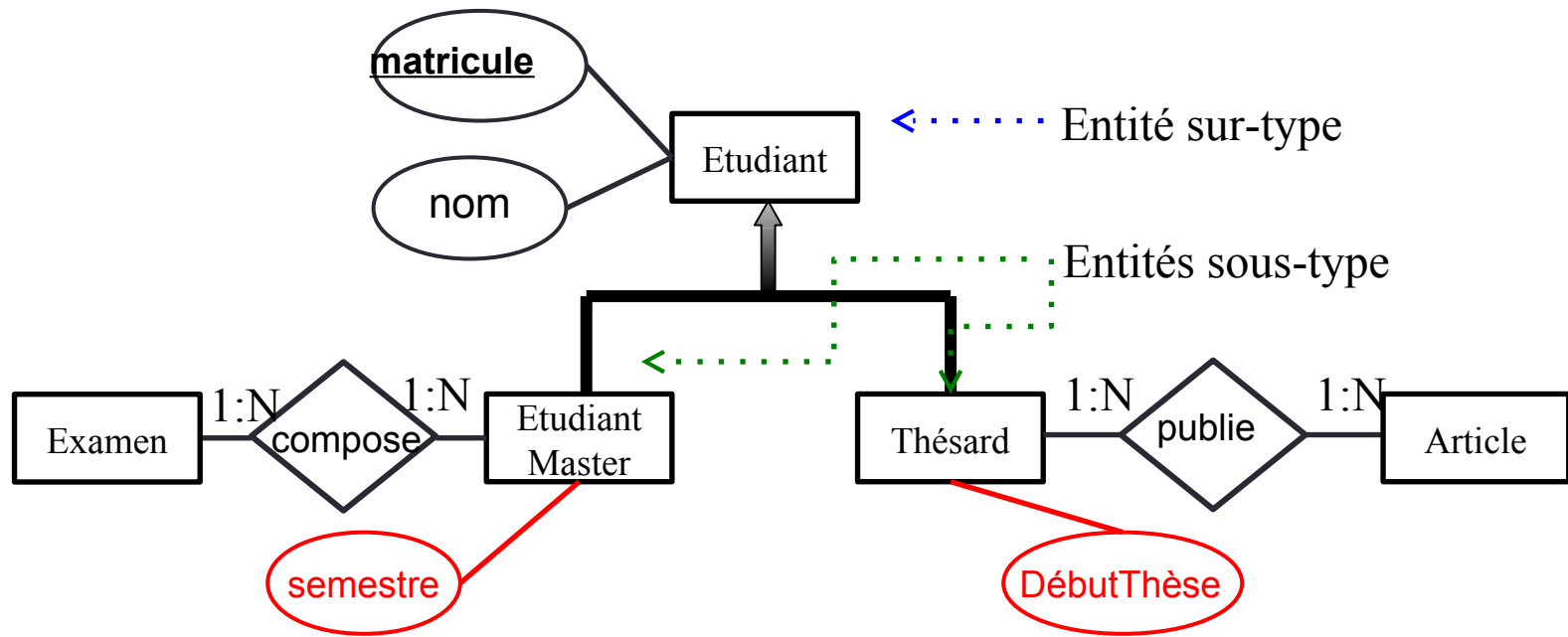
■ Transformation de la spécialisation:

- Créer une relation (table) pour chaque entité sous-type, les attributs de la relation sont ceux de l'entité sous-type
- *Clé de la relation*: l'identifiant de l'entité sur-type

■ **Cas particulier**, si l'entité sur-type est abstraite (il n'existe pas d'instance de cette entité dans l'application):

- supprimer la table correspondante à l'entité sur-type
- rajouter tous ses attributs dans toutes les tables correspondantes aux entités sous-type

Exemple



Etudiant(matricule, nom)

EtudiantMaster(matricule*, semestre)

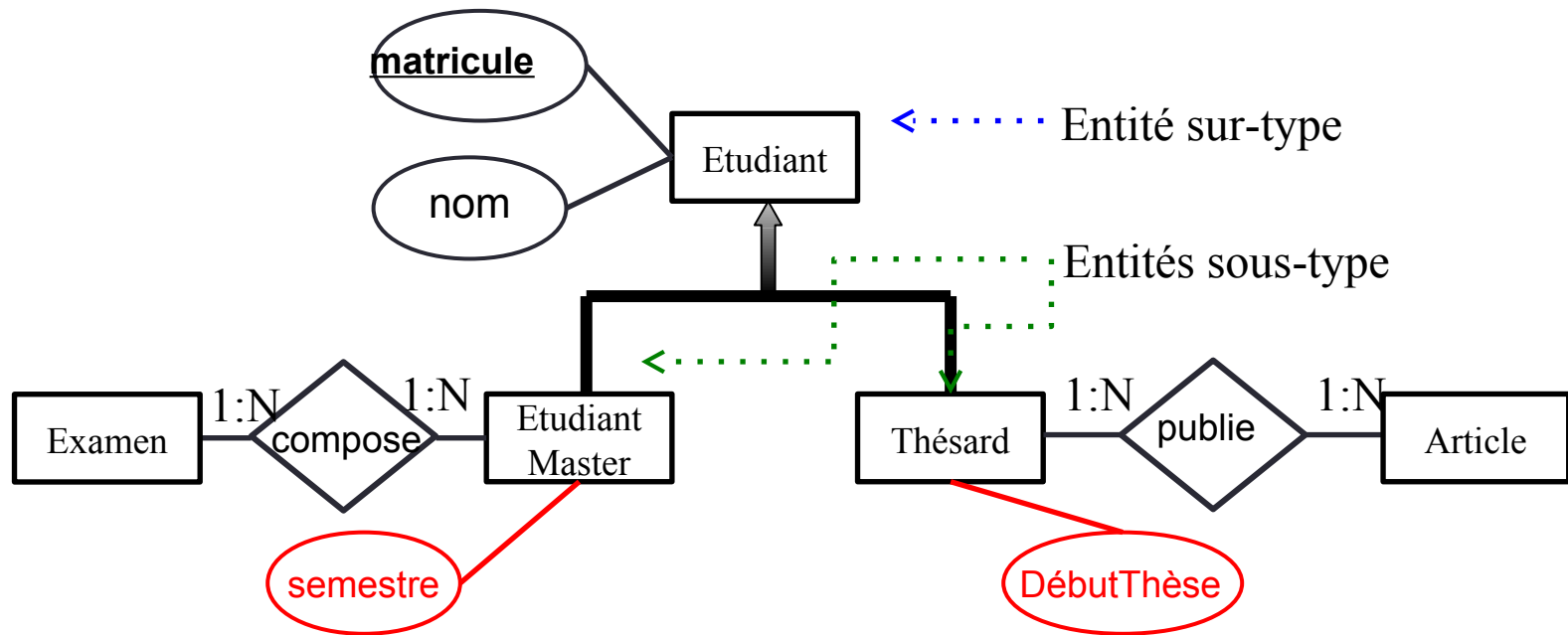
Thesard(matricule*, debutThese)

Dans **EtudiantMaster** et **Thesard**

matricule est clé primaire et étrangère
(référence matricule de la table **Etudiant**)
en même temps!

Exemple

- Il n'existe pas d'étudiants autres que les étudiants inscrits en master ou en thèse : → on supprime la table **Etudiant**



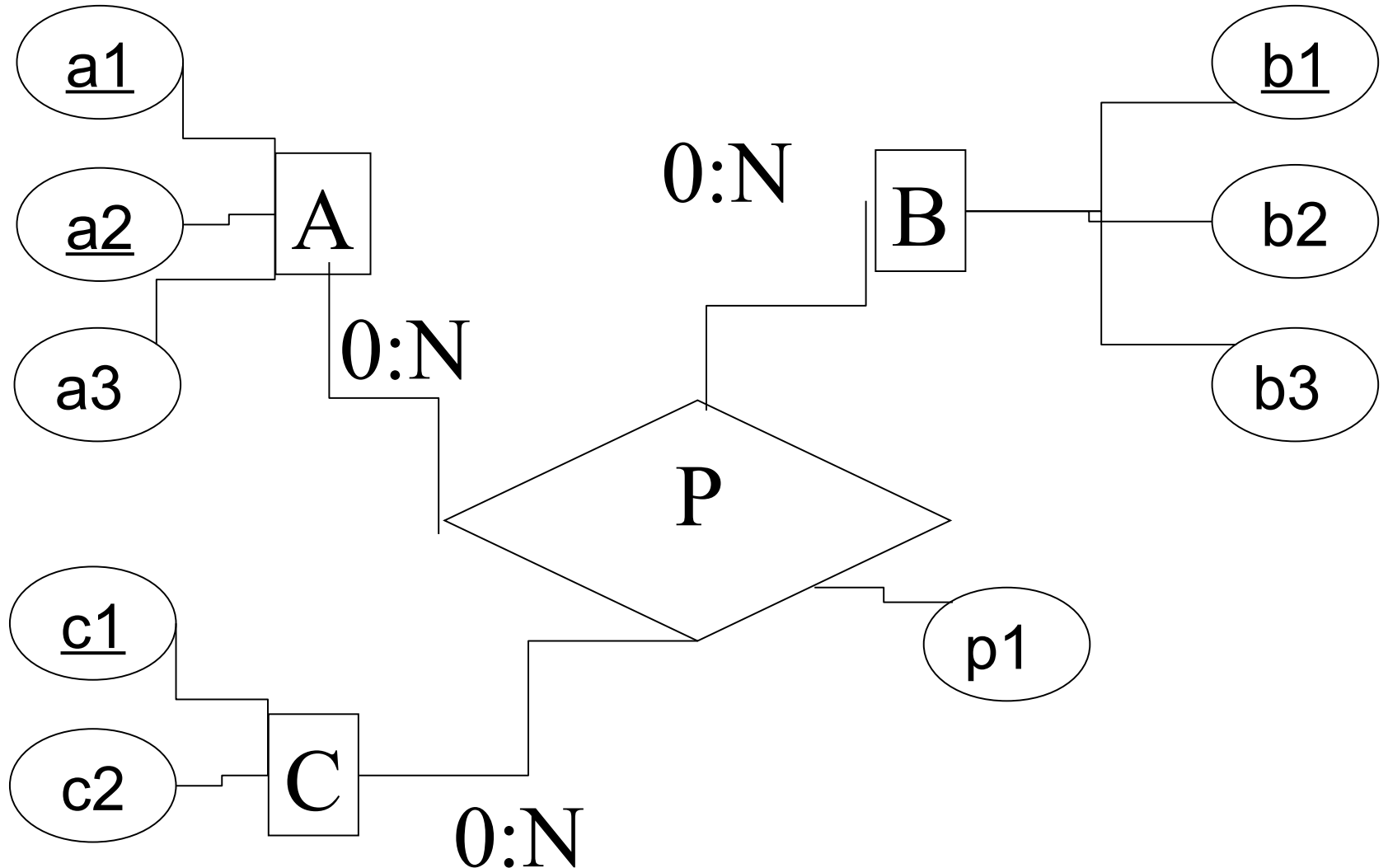
~~**Etudiant**~~(matricule, nom)

EtudiantMaster(matricule, nom, semestre)

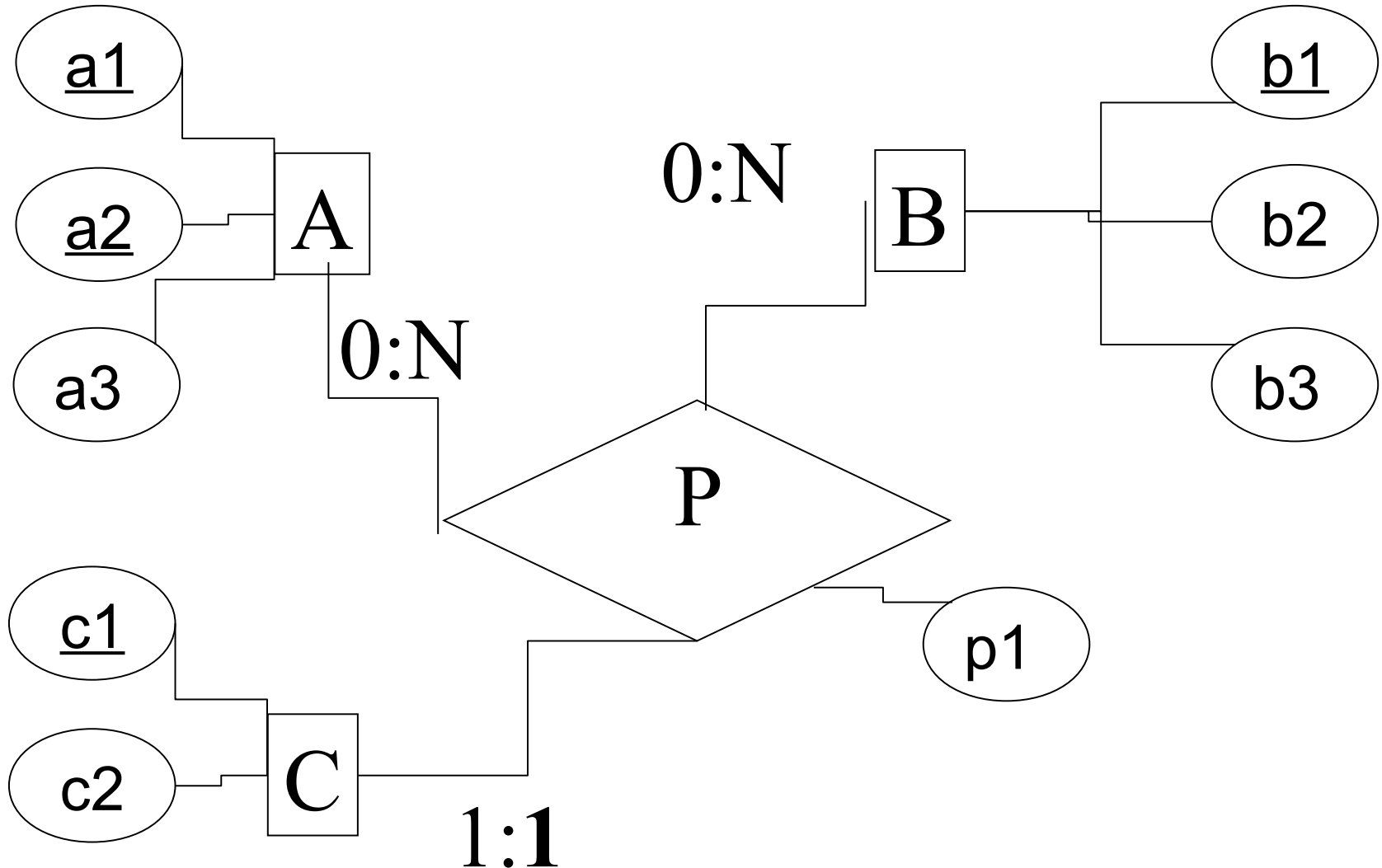
Thésard(matricule, nom, debutThese)

Dans **EtudiantMaster** et **Thésard**
matricule est la clé primaire, elle n'est pas clé étrangère.

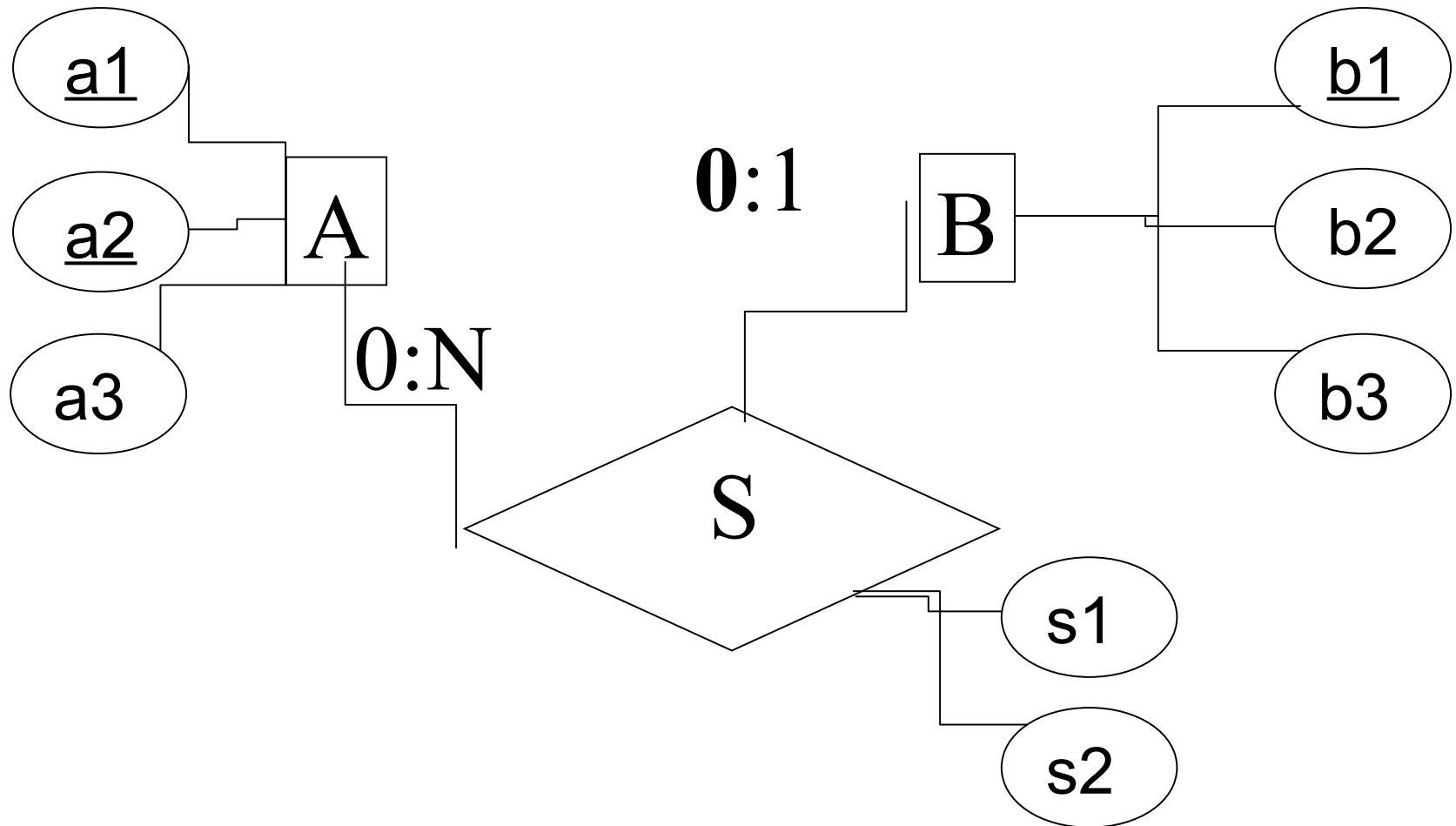
Exercices de traduction : 1



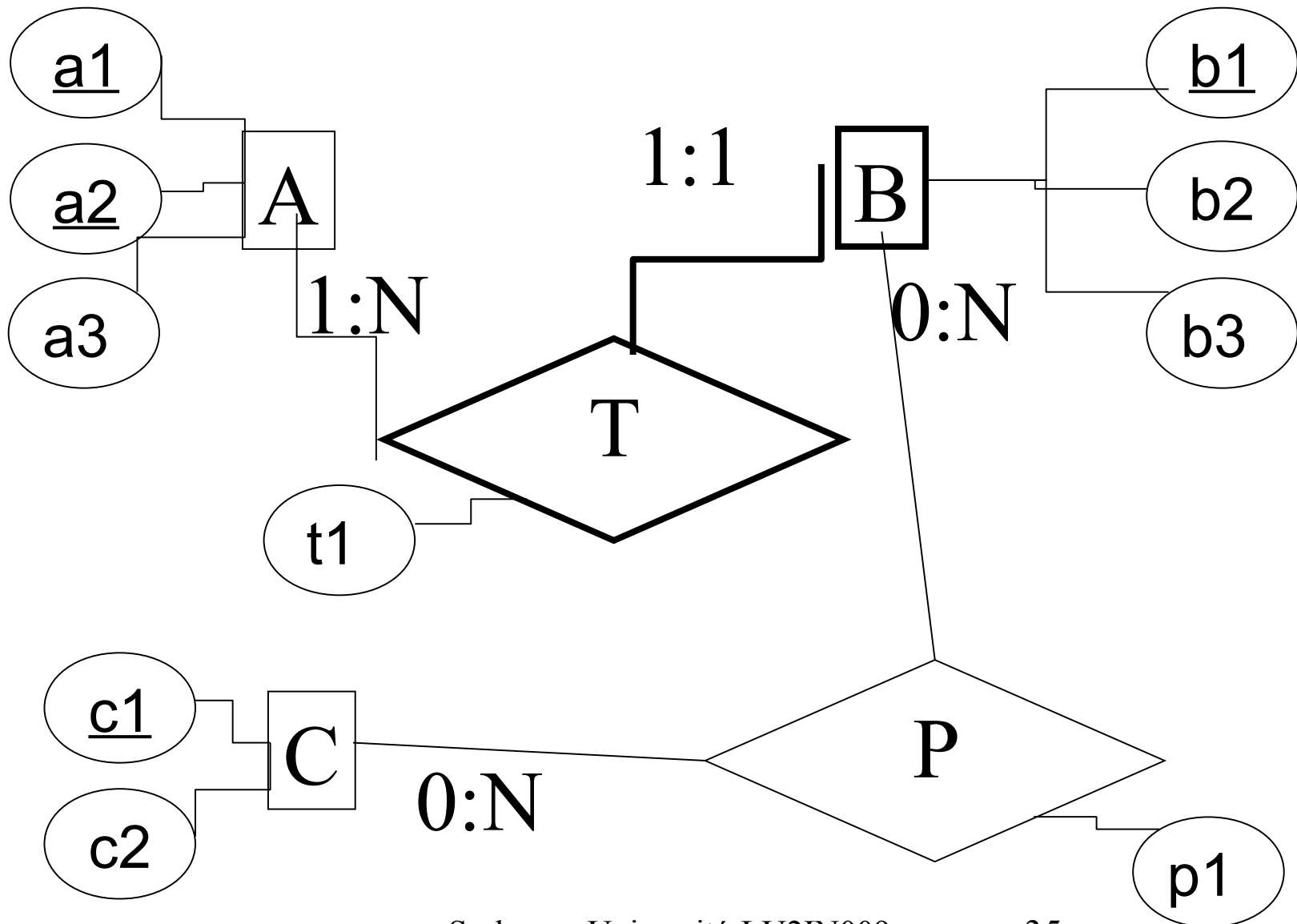
Exercices de traduction : 2



Exercices de traduction : 3



Exercices de traduction : 4



Exercice de rétro-ingénierie : 1

Soit le schéma relationnel suivant. En déduire un schéma /EA correspondant.

- ARTICLE (Aid)
- CATEGORIE (Cid)
- REFERENCE (Aid*, Cid*)
- PERSONNE (Pid, NomP, Prenom, Email)
- ArticleLangue(Aid, Langue, Titre, Contenu)
- CatégorieLangue(Cid, Langue, NomCat)
- ECRIT (Pid*, Aid*, Langue, Titre)

Exercice de rétro-ingénierie : 2

Soit le schéma relationnel suivant. En déduire un schéma EA correspondant.

SPORTIF (SID, Nom, DateNaiss, Manager)

MANAGER (MID, Tarif, Experience)

CLUB (CID, DateCreation, Budget, Division)

CONTRAT (SID, CID, SAISON-DEBUT, NBSais, Salaire, Augmentation)

RENCONTRE (LOCAL, VISITEUR, SAISON, Vainqueur)

Corrigé

- Traduction EA \rightarrow relationnel
 - $A(\underline{a1}, \underline{a2} \ a3)$ commun aux 4 exercices
 - Ex1 : immédiat
 - Ex2 : $B(\underline{b1}, b2, b3) \ P(\underline{a1^*}, \underline{a2^*}, \underline{b1^*}, p1) \ C(\underline{c1}, c2, a1^*, a2^*, b1^*, p1)$
 - ou bien A et B idem, $C(\underline{c1}, c2)$ rajouter à $P \ c1^*$
 - Ex3 : $B(\underline{b1}, b2, b3) \ S(a1^*, a2^*, \underline{b1^*}, s1, s2)$
 - Ex4 : $B(\underline{b1}, \underline{a1^*}, \underline{a2^*}, b2, b3, t1), \ C(\underline{c1}, c2), \ P(\underline{b1^*}, \underline{a1^*}, \underline{a2^*}, \underline{c1^*}, p1)$