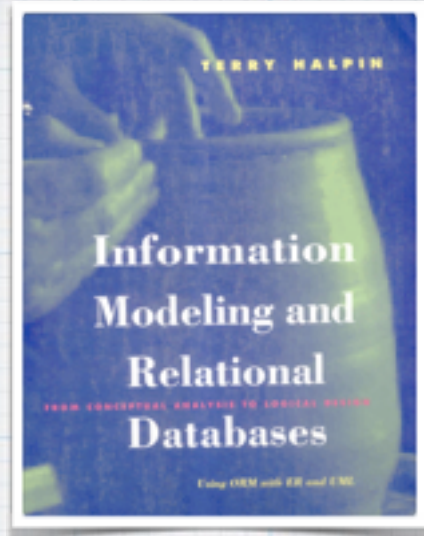


LSINF1 225

Modélisation de données
ORM

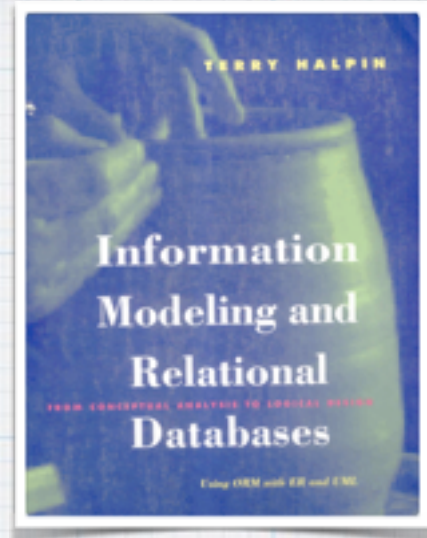


Livre de référence



- ❑ **Information Modeling and Relational Databases, Terry Halpin, 2001, Morgan Kaufmann**
- ❑ **(Chapitre 1 : Introduction)**
- ❑ **Chapitre 2 : Information Levels and Frameworks**
- ❑ **Chapitre 3 : Conceptual Modeling: First Steps**
- ❑ **Chapitre 4 : Uniqueness Constraints**
- ❑ **Chapitre 10 : Relational Mapping**

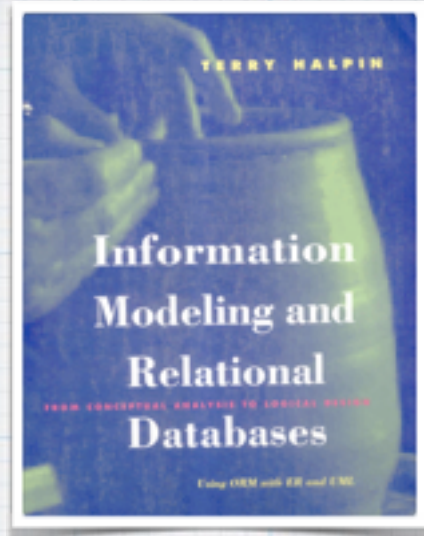
Livre de référence



- ❑ **Information Modeling and Relational Databases, Terry Halpin, 2001, Morgan Kaufmann**
- ❑ **Chapitre 1 : Introduction**
 - ❑ **1.1 Modélisation de données (pp. 2-6)**
 - ❑ **1.2 Approches de modélisation (pp. 6-18)**
(seulement les paragraphes liés à l'approche ORM)
 - ❑ **1.5 Résumé (p. 24)**

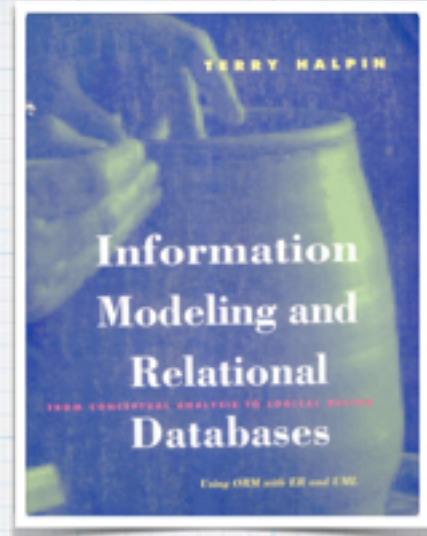


Livre de référence



- ❑ **Information Modeling and Relational Databases, Terry Halpin, 2001, Morgan Kaufmann**
- ❑ **Chapitre 2 : Information Levels and Frameworks**
 - ❑ **2.2 The Conceptual Level (p. 31)**
 - ❑ **2.3 From External to Conceptual to Relational (pp. 40–45)**
 - ❑ **2.5 Summary (pp. 53)**

Livre de référence



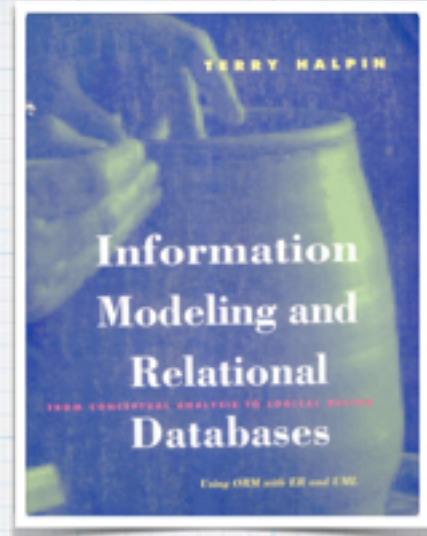
- ❑ **Information Modeling and Relational Databases, Terry Halpin, 2001, Morgan Kaufmann**

- ❑ **Chapitre 3 : Conceptual Modeling: First Steps**



- ❑ **3.2 ORM's Conceptual Schema Design Procedure (pp. 58–60)**
 - ❑ **3.3 CSDP Step 1 : Verbalize information (pp. 60–76)**
 - ❑ **3.4 CSDP Step 2 : Draw fact types and check population (pp. 78–90)**
 - ❑ **3.6 Summary**

Livre de référence



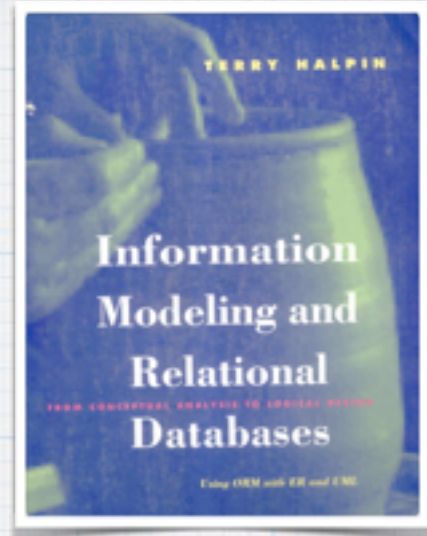
- ❑ **Information Modeling and Relational Databases, Terry Halpin, 2001, Morgan Kaufmann**

- ❑ **Chapitre 4 : Uniqueness Constraints**



- ❑ **4.1 CSDP Step 4 (pp. 110–111)**
- ❑ **4.2 Uniqueness Constraints on Unaries and Binaries (pp. 111–121)**
- ❑ **4.3 Uniqueness Constraints and Longer Fact Types (pp. 122–128)**
- ❑ **4.7 Summary (pp. 158–161)**

Livre de référence



- ❑ **Information Modeling and Relational Databases, Terry Halpin, 2001, Morgan Kaufmann**

- ❑ **Chapitre 10 : Relational Mapping**

- ❑ **10.1 Implementing a Conceptual Schema (pp. 404-405)**
- ❑ **10.2 Relational Schemas (pp. 405-412)**
- ❑ **10.3 Relational Mapping Procedure (pp. 412-441)**
- ❑ **10.5 Summary (pp. 455-456)**



Introduction



- Quelques définitions préliminaires
- Le modèle relationnel
- ORM : Object-Role Modeling

Quelques définitions préliminaires



- ❑ **Base de données**
- ❑ **Système de gestion de base de données**
- ❑ **Univers de discours**
- ❑ **ORM ("Object-Role Modeling")**

Bases de données



- ❑ Une base de données (BD) est un système informatique de stockage d'information
- ❑ contient un ensemble de données
- ❑ peut être interprétée comme un ensemble de faits liés
- ❑ Il existe différents types de bases de données :
 - ❑ relationnelles, hiérarchiques, orienté objets, etc.
 - ❑ les bases de données les plus utilisés s'appuient sur le modèle de données relationnel et utilisent le langage de requêtes SQL

Systeme de gestion de bases de données



- Une systeme de gestion de bases de données (SGBD) est un système informatique permettant de gérer une BD
- maintenir la base de données
- répondre à des requêtes sur les données
- Exemples:
 - Microsoft Access, DB2, Oracle, SQL Server, MySQL, SQLite



ORM



- **ORM* = Object-Role Modeling**
- **Une approche de modélisation de l'information**
- **Pour modéliser l'univers de discours d'une application**
- **Notation intuitive et naturelle pour concevoir la structure conceptuelle ("le schéma conceptuel") d'une base de données (relationnelle)**
- **Permettant de formuler facilement des requêtes sur cette base de données**

* à ne pas confondre avec "Object-Relational Mapping"

Introduction



- Quelques définitions préliminaires
- Le modèle relationnel
- ORM : Object-Role Modeling

Le modèle de données relationnel

- fournit une base mathématique claire pour définir la sémantique des langages de bases de données
- permet d'exprimer mathématiquement et de façon déclarative (non opérationnelle) l'ensemble des requêtes (questions) que l'on peut adresser à une base de données
- fournit un outil mathématique pour optimiser les requêtes de bases de données
- est basé sur la notion mathématique des relations n -aires

Exemple d'une relation

- Supposons que chaque client d'une entreprise possède :
 - un ID, une SORTE ("interne" ou "externe"), un NOM, un PRENOM et une ADRESSE
- Alors, on peut représenter un client particulier par un multiplet ("tuple") comme :

(6789, externe, Durant, Alfred, "32, avenue des alliés, Tournai")

spécifiant l'ID, la SORTE, le NOM, le PRENOM et l'ADRESSE (respectivement) du client ;

et on appellera (ID, SORTE, NOM, PRENOM, ADRESSE) le schéma de la relation

Exemple d'une relation

- La relation **CLIENT** est alors un ensemble de tels multiplets :

{ (459, externe, Horner, Yvette, "7, impasse des capucines, Wavre"),
(3124, interne, Dupont, Jules, "3, rue des combattants, La Hulpe"),
(6347, interne, Durant, Alfred, "17, allée des saules, Plancenoit"),
(6789, externe, Durant, Alfred, "32, avenue des alliés, Tournai") }

- Notation positionnelle

- Pour extraire un des composants d'un multiplet il faut connaître exactement sa position dans le multiplet

- p.e. **PRENOM** -> 4

Le modèle relationnel

- Dans le modèle relationnel, on n'utilise pas la notation positionnelle.
- A la place, on donne à chaque composant un nom qui l'identifie.

{ (ID, 6789), (SORTE, externe), (NOM, Durant), (PRENOM, Alfred),
(ADRESSE, "32, avenue des alliés, Tournai") }

Le modèle relationnel

□ Une relation n-aire est un ensemble de tels multiplets

□ p.e. la relation n-aire **CLIENT** est l'ensemble

{ (ID, 459), (SORTE, externe), (NOM, Horner), (PRENOM, Yvette), (ADRESSE, "7, impasse des capucines, Wavre") },

...

{ (ID, 6789), (SORTE, externe), (NOM, Durant), (PRENOM, Alfred), (ADRESSE, "32, avenue des alliés, Tournai") }

}

□ On représente les relations (finies) sous forme de tables

Exemple d'une "table"

- La relation **CLIENT** est représentée par la table

{ (ID, 459),	(SORTE, externe),	(NOM, Horner),	(PRENOM, Yvette),	(ADRESSE, "7, impasse des capucines, Wavre") }
...				
{ (ID, 6789),	(SORTE, externe),	(NOM, Durant),	(PRENOM, Alfred),	(ADRESSE, "32, avenue des alliés, Tournai") }

- On représente les relations (finies) sous forme de tables

Exemple d'une "table"

- La relation **CLIENT** est représentée par la table

CLIENT(ID, SORTE, NOM, PRENOM, ADRESSE)

Signature de la relation n-aire

ID	SORTE	NOM	PRENOM	ADRESSE
459	externe	Horner	Yvette	"7, impasse des capucines, Wavre"
3124	interne	Dupont	Jules	"3, rue des combattants, La Hulpe"
6347	interne	Durant	Alfred	"17, allée des saules, Plancenoit"
6789	externe	Durant	Alfred	"32, avenue des alliés, Tournai"

Base de données relationnelles

□ Définition [Base de données]

- Une base de données est un ensemble de relations n -aires
- (donc, un ensemble de tables comme celle du transparent précédent)

□ Définition [Schéma de base de données]

- Un schéma de base de données est constitué de l'ensemble des signatures des relations de la base de données

Exemple d'une base de données

CLIENT

ID	SORTE	NOM	PRENOM	ADRESSE
459	externe	Horner	Yvette	"7, impasse des capucines, Wavre"
3124	interne	Dupont	Jules	"3, rue des combattants, La Hulpe"
6347	interne	Durant	Alfred	"17, allée des saules, Plancenoit"
6789	externe	Durant	Alfred	"32, avenue des alliés, Tournai"

PRODUIT

NUM	NOM	PRIX_UNITAIRE
8	bic	12
45	carnet	27
123	règle	18
978	marqueur	13

CLIENT_EXTERNE

ID	NUM_CARTE_CREDIT
459	3452 8765 6775 5321
6789	5643 5568 9978 1200

COMMANDE

NUM	ID_CLI	TOTAL
11345	459	45
11467	3124	485
12523	459	732

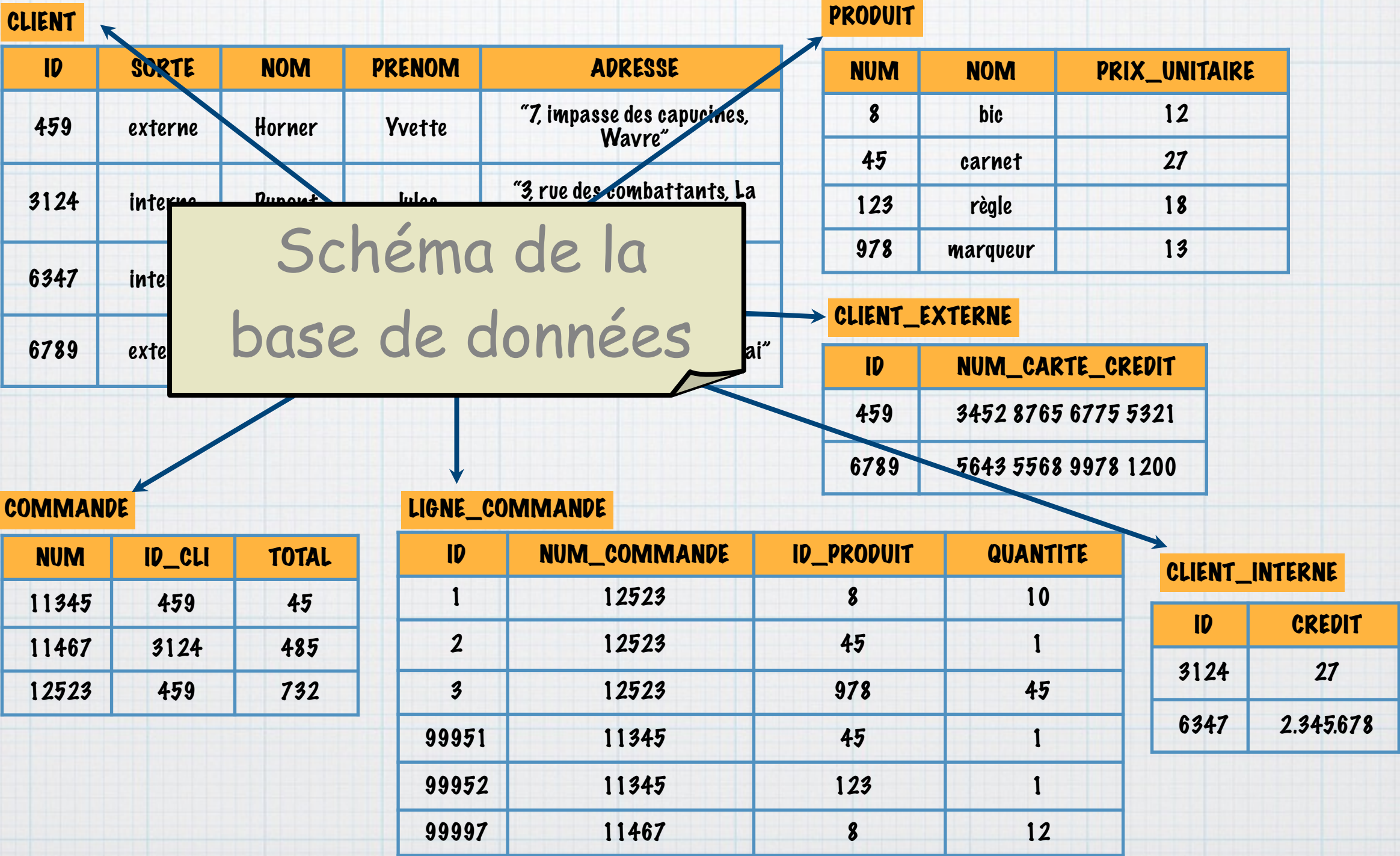
LIGNE_COMMANDE

ID	NUM_COMMANDE	ID_PRODUIT	QUANTITE
1	12523	8	10
2	12523	45	1
3	12523	978	45
99951	11345	45	1
99952	11345	123	1
99997	11467	8	12

CLIENT_INTERNE

ID	CREDIT
3124	27
6347	2.345.678

Exemple d'un schéma d'une base de données



Exemple d'un schéma d'une base de données

CLIENT(ID, SORT, NOM, PRENOM, ADRESSE)

CLIENT_INTERNE(ID, CREDIT)

CLIENT_EXTERNE(ID, NUM_CARTE_CREDIT)

COMMANDE(NUM, ID_CLI, TOTAL)

PRODUIT(ID, NOM, PRIX_UNITAIRE)

LIGNE_COMMANDE(ID, NUM_COMMANDE, ID_PRODUIT, QUANTITE)

Introduction



Comment concevoir le schéma d'une base de données relationnelle ?

- ❑ Quelques définitions préliminaires
- ❑ Le modèle relationnel
- ❑ **ORM : Object-Role Modeling**

Comment structurer une base de données ?

CLIENT

ID	SORTE	NOM	PRENOM	ADRESSE
459	externe	Horner	Yvette	"7, impasse des capucines, Wavre"
3124	interne	Dupont	Jules	"2
6347	interne	Durand		
6789				

PRODUIT

NUM	NOM	PRIX_UNITAIRE
8	bic	12
45	carnet	27
123	règle	18
978	marqueur	13

CLIENT_EXTERNE

ID	NUM_CARTE_CREDIT
459	3452 8765 6775 5321
6789	5643 5568 9978 1200

COMMANDE

NUM	ID_CLIENT	ID_PRODUIT	QUANTITE
11345	459	8	10
11467	3124	45	1
12523	459	978	45
99951	11345	45	1
99952	11345	123	1
99997	11467	8	12

CLIENT_INTERNE

ID	CREDIT
3124	27
6347	2.345.678

Conception d'un schéma
d'une base de données
Approche: ORM

Comment structurer une base de données ?

- ❑ **Comment concevoir un schéma d'une base de données (relationnelle) ?**
 - ❑ **Sans (trop de) redondance**
 - ❑ **Conforme au monde réel**
 - ❑ **Intuitif et facile à comprendre**
- ❑ **... un boulot non-trivial ...**

Modélisation de données

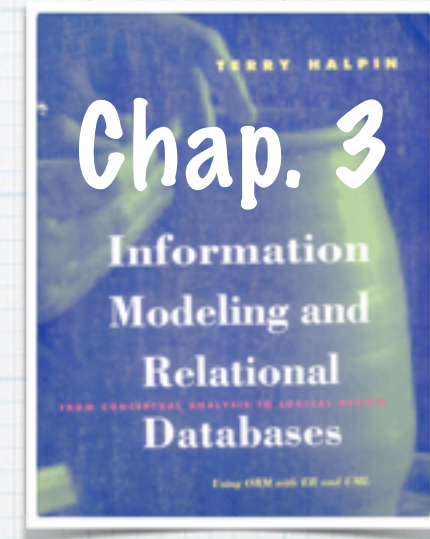
- **Chaque BD modélise un domaine d'application (ou VoD)**
 - **Essentiellement, une BD est une collection de données**
- **Le schéma conceptuel décrit la structure du domaine de l'application :**
 - **quels types de données ?**
 - **quel rôles jouent ces données ?**
 - **quels règles et contraintes s'appliquent entre ces données ?**
- **Mais comment concevoir ce schéma conceptuel ?**

Modélisation de données

- ❑ ORM est une approche “fact-oriented” de modélisation de l'information
- ❑ Une BD peut être vue comme un ensemble de faits (les données) et de règles (décrivant les liens entre les faits). Exemple :
 - ❑ domaine d'application : une bibliothèque
 - ❑ données : description des livres (titres, auteurs, numéro ISBN, ...)
 - ❑ fait : le livre “Information Modeling and Relational Databases” a comme auteur “T. Halpin”
 - ❑ règle : chaque livre a un numéro ISBN unique

Comment structurer une base de données ?

- ❑ Approche: “Object-Role Modeling”
 - ❑ Langage de modélisation = ORM
 - ❑ Processus = “Conceptual Schema Design Procedure” (CSDP)
 - ❑ A partir du schéma obtenu, un schéma d’une base de données peut être généré automatiquement



Conception d'un schéma de base de données en ORM

- **Sept étapes (processus CSDP*)**
 - **Identifier les "types de faits" [Halpin, Chap. 3]**
 1. Trouver les faits élémentaires à partir des exemples de données
 2. Dessiner les types de faits et ajouter une population au diagramme
 3. Combiner des types d'entités et noter des dérivations arithmétiques
 - **Ajouter des contraintes aux types de faits [Halpin, Chap. 4-7]**
 4. Ajouter les contraintes d'unicité et vérifier les arités des types de faits
 - ... Ajouter d'autres types de contraintes

Etape 1 : Transformez des exemples en faits élémentaires

- L'approche ORM propose de commencer par des exemples concrets de données traitées par le système
- par ex. des rapports produits ou des formulaires de saisie
- et de transformer ces exemples de données en faits élémentaires

N° DE FACTURE 0203045893621

N° de contrat: 09867523

Date: 18 juin 2001

Adresse client:

UCL
DEPARTEMENT INGI
PL STE BARBE, 2
1348 LOUVAIN-LA-NEUVE

Adresse de facturation:

UCL
DEPARTEMENT INGI
PL STE BARBE, 2
1348 LOUVAIN-LA-NEUVE

RESUME	Période: 9 mai 2001 - 18 juin 2001	T.V.A.	MONTANT en EUR (hors T.V.A.)
Communications		21%	6,9762

Tarif T.V.A.	21%	Total (EUR)
Montant de base	6,98	6,98
Montant T.V.A.	1,47	1,47
Montant total de cette facture:		8,44
A payer avant le 29 juin 2001		8,44

Résumé par numéro de téléphone

N° TELEPHONE	COMMUNICATIONS	AUTRES CHARGES	TOTAL EN EUR
0475/529811	2,5550	0,0000	2,5550
0475/529812	4,4212	0,0000	4,4212
Total en EUR	6,9762	0,0000	6,9762

N° de téléphone: 0475/529811 JULES CESAR

Communications nationales

N°	DATE	HEURE	DESTINATION	ZONE	DUREE	PRIX EN EUR
1	19 avr	19:14	86239823	Durbuy	1:00	0,1223
2	27 avr	11:09	475151230	Ecouter mail	1:00	0,0000
3	02 mai	13:00	475151230	Ecouter mail	1:08	0,0000
4	03 mai	14:42	475151230	Ecouter mail	1:00	0,0000
5	08 mai	18:28	498734523	Mobistar	1:00	0,3073
6	08 mai	18:57	498734523	Mobistar	1:00	0,3073
7	08 mai	18:57	86239823	Durbuy	1:00	0,4011
8	13 mai	17:25	478125677	Proximus	1:00	0,1230
9	15 mai	17:41	498734523	Mobistar	1:00	0,3073
10	15 mai	18:21	498734523	Mobistar	1:00	0,3073
11	15 mai	18:24	476202312	Proximus	1:38	0,2001
12	16 mai	09:56	476202312	Proximus	1:00	0,1230
13	18 mai	18:35	86239823	Durbuy	1:25	0,1731
14	18 mai	18:42	86239823	Durbuy	1:30	0,1832
15	18 mai	22:59	475151230	Ecouter mail	7:16	0,0000

Minutes utilisées: 22:57

Total des communications nationales: 2,5550

Les entités

- **Les faits élémentaires expriment que des objets jouent certains rôles**
- **Nous distinguons deux types d'objets :**
 - **valeurs et entités**
- **Les valeurs:**
 - **des chaînes de caractères comme "Proximus"**
 - **des chiffres comme 289**
- **Les entités comme Un Client avec nom "UCL"**

Les entités

- Les entités sont décrit par un(e):
 - Type d'entité (p.e., Client, Facture, Montant, ...)
 - Mode de référence (p.e., nom, numéro, Euro)
 - Valeur (p.e., "UCL", 0203045893621, 275)
- Exemples:
 - Un Client avec nom "UCL"
 - Une Facture avec numéro 0203045893621
 - Un Montant de 275 Euro

Deux formats :

$\langle \text{Entité} \rangle = \langle \text{Type} \rangle \langle \text{Mode} \rangle \langle \text{Valeur} \rangle$

OU

$\langle \text{Entité} \rangle = \langle \text{Type} \rangle \langle \text{Valeur} \rangle \langle \text{Mode} \rangle$

Les entités (notation abrégée)

- Notation abrégée :
 - **<Type> (<Mode>) <Valeur>**
- Exemples:
 - Client (nom) "UCL"
 - Facture (numéro) 0203045893621
 - Montant (Euro) 275

Les faits élémentaires

- **Un fait élémentaire est une assertion simple (et atomique*) sur l'univers de discours**
- **Les faits élémentaires expriment des relations entre les objets dans l'univers de discours**
- **Différents types de faits élémentaires existent**
 - **relation unaire : exprime qu'un objet a une certaine propriété**
 - **relation binaire : décrit une relation entre deux objets**
 - **relation n-aire : décrit une relation entre plusieurs objets**

***ne peut être divisé**

Les faits élémentaires

□ Exemple d'une relation unaire

- La Destination avec numéro 0475151230 est gratuit

□ Exemples de relations binaires

- L'Opérateur dénommé "Proximus" a un Préfixe 0478
- La Communication numéro 11 a une Durée de "1:38" minutes
- Le Client avec nom "UCL" a une Adresse de facturation
"DEPARTEMENT INGI, PL STE BARBE 2, 1348 LOUVAIN-LA-NEUVE"
- La Facture numéro 0203045893621 coûte un Montant de 289 Euro

Les faits élémentaires (notation abrégée)

□ Exemple d'une relation unaire

- Destination (numéro) 0475151230 **est gratuit**

□ Exemples de relations binaires

- Opérateur (nom) "Proximus" **a** Préfixe (numéro) 0478
- Communication (numéro) 11 **a** Durée (minutes) "1:38"
- Client (nom) "UCL" **a** Adresse de facturation (string) "DEPARTEMENT INGI, PL STE BARBE 2, 1348 LOUVAIN-LA-NEUVE"
- Facture (numéro) 0203045893621 **coûte** Montant (Euro) 289

Les faits élémentaires

- L'adjectif "élémentaire" signifie que les faits ne peuvent pas être divisés en parties qui sont encore plus petites mais qui expriment la même information que l'original.
- Les faits élémentaires n'utilisent pas de connecteurs logiques (non, et, ou, si) ni de quantificateurs logiques (pour chaque, il existe)

Les faits élémentaires

□ Exemple d'un relation ternaire:

□ Etudiant (NOMA) 662455 a obtenu Cote (chiffre) 12 pour Cours (code) "LSINF1125"

□ est un fait élémentaire; ne peut pas être divisé en faits encore plus élémentaires, par ex:

□ Etudiant (NOMA) 662455 a obtenu Cote (chiffre) 12

□ Etudiant (NOMA) 662455 a suivi Cours (code) "LSINF1125"

□ Cote (chiffre) 12 a été obtenue pour Cours (code) "LSINF1125"

□ ne peuvent pas être combinés pour arriver au fait élémentaire original

Les faits élémentaires

- L'adjectif "élémentaire" signifie que les faits ne peuvent pas être divisés en parties qui sont encore plus petites mais expriment la même information que l'original.
- Les faits élémentaires n'utilisent pas de connecteurs logiques (non, et, ou, si) ni de quantificateurs logiques (pour chaque, il existe)

Contre-exemples de faits élémentaires

- ❑ Les phrases suivantes ne sont pas des faits élémentaires:
 - ❑ Le Client dénommé "UCL" paie pour un Appelant avec nom "Jules Cesar" ET Le Client dénommé "UCL" a une Adresse de facturation "DEPARTEMENT INGI, PL STE BARBE 2, 1348 LOUVAIN-LA-NEUVE"
 - ❑ Le Client dénommé "UCL" n'a PAS de contrat avec l'Opérateur avec nom "Mobistar"
 - ❑ SI la Personne avec nom "Kim Mens" a un Numéro de téléphone, ALORS il est Appelant
 - ❑ CHAQUE personne ayant un Numéro de téléphone est un Appelant

Etape 2 : Dessiner les types de faits et ajouter une population

- Transformez les faits élémentaires en un schéma conceptuel
- Vérifiez l'exactitude du schéma en ajoutant une "population"

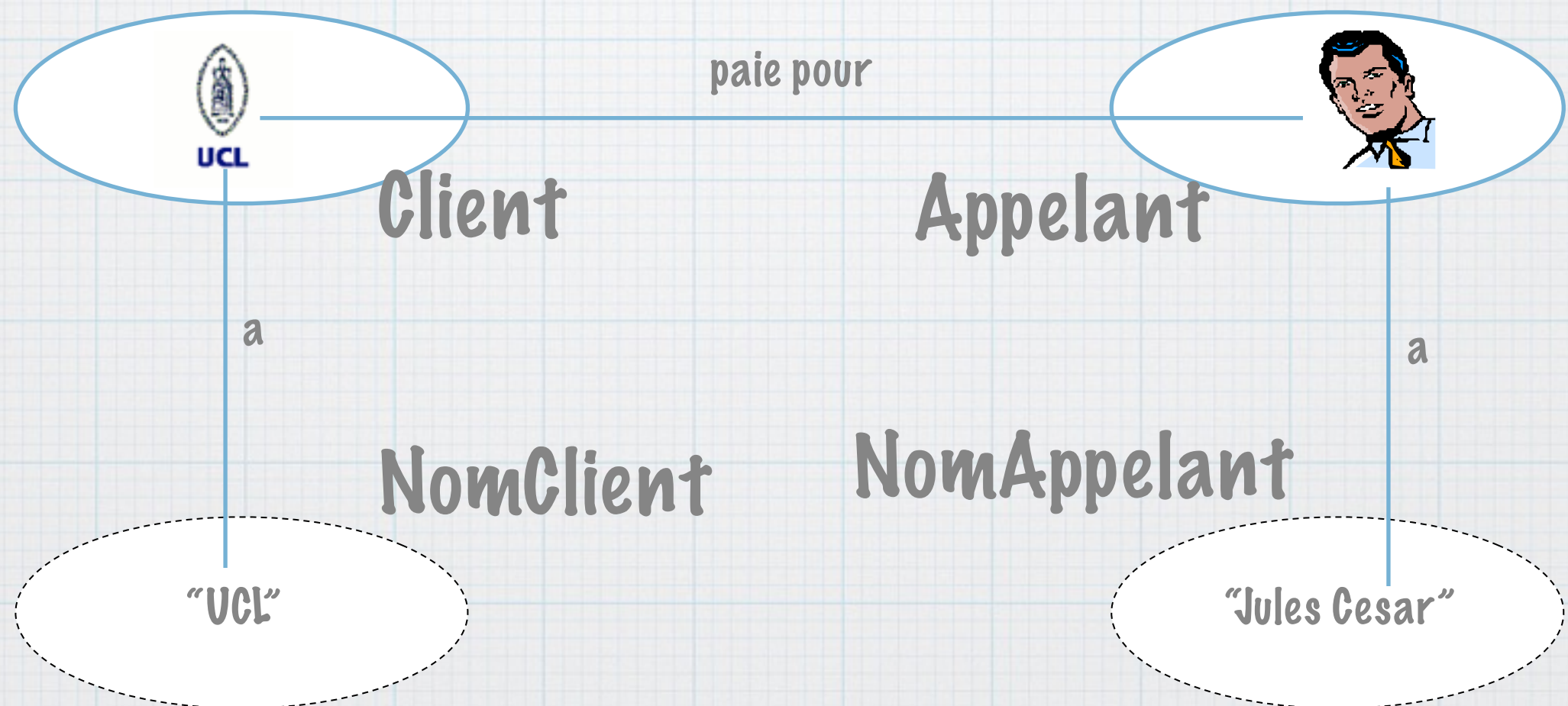
Identifier les "types de faits"

- ❑ Le schéma conceptuel montre tous les types de faits
 - ❑ Sont les faits primitifs qui peuvent être stockés dans la BD
 - ❑ Quels types d'entités existent? (Client, Facture, ...)
 - ❑ Quels types de valeurs existent? (NomClient, NuméroFacture, ...)
 - ❑ Comment les entités sont référencées par une valeur dans la base de données? (une Facture est référencée par un NuméroFacture, un Etudiant est référencée par un NOMA, ...)
 - ❑ Quels relations existent entre les entités? (est né en, vit en, ...)



Exemple

- Le Client avec nom "UCL" paie pour un Appelant avec nom "Jules Cesar"



Les éléments d'un schéma conceptuel



Client

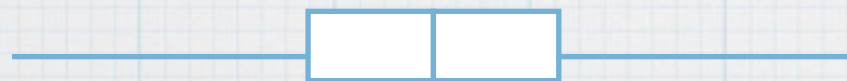
Type d'entité



NomClient

Type de valeur

paie pour / correspond à



Relation n-aire

paie pour



Rôle joué par un objet
(entité ou valeur)

Types de référence

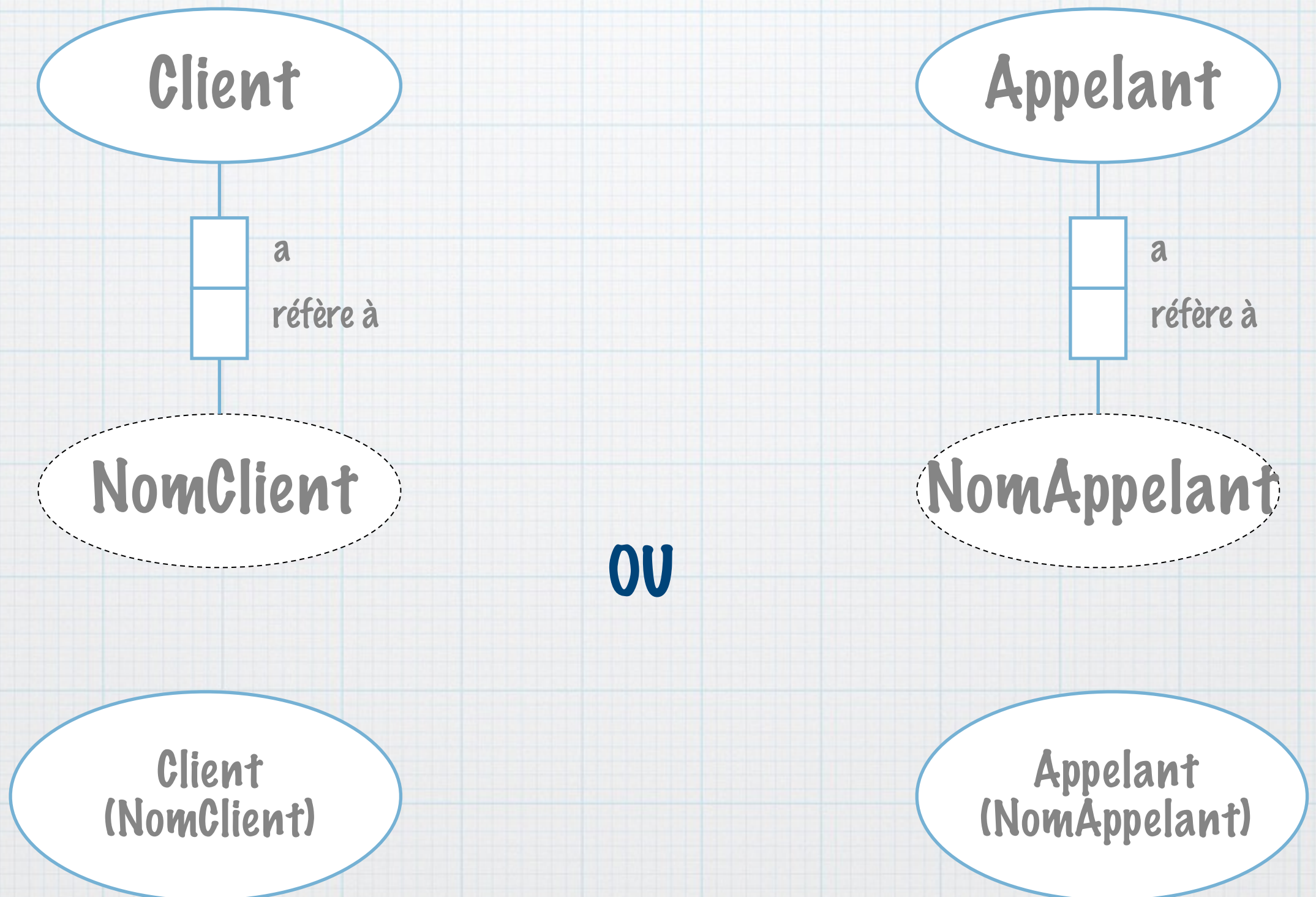
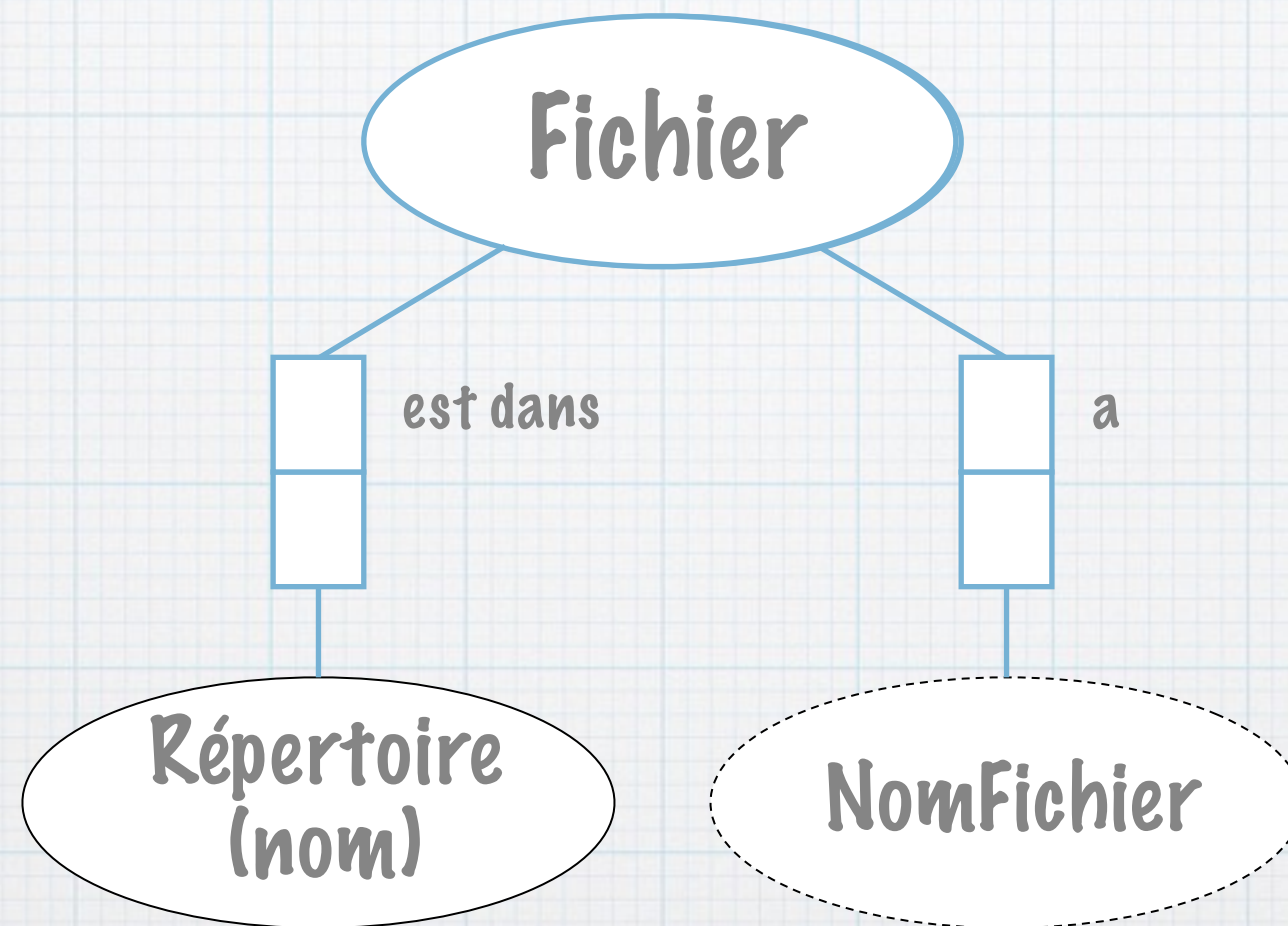
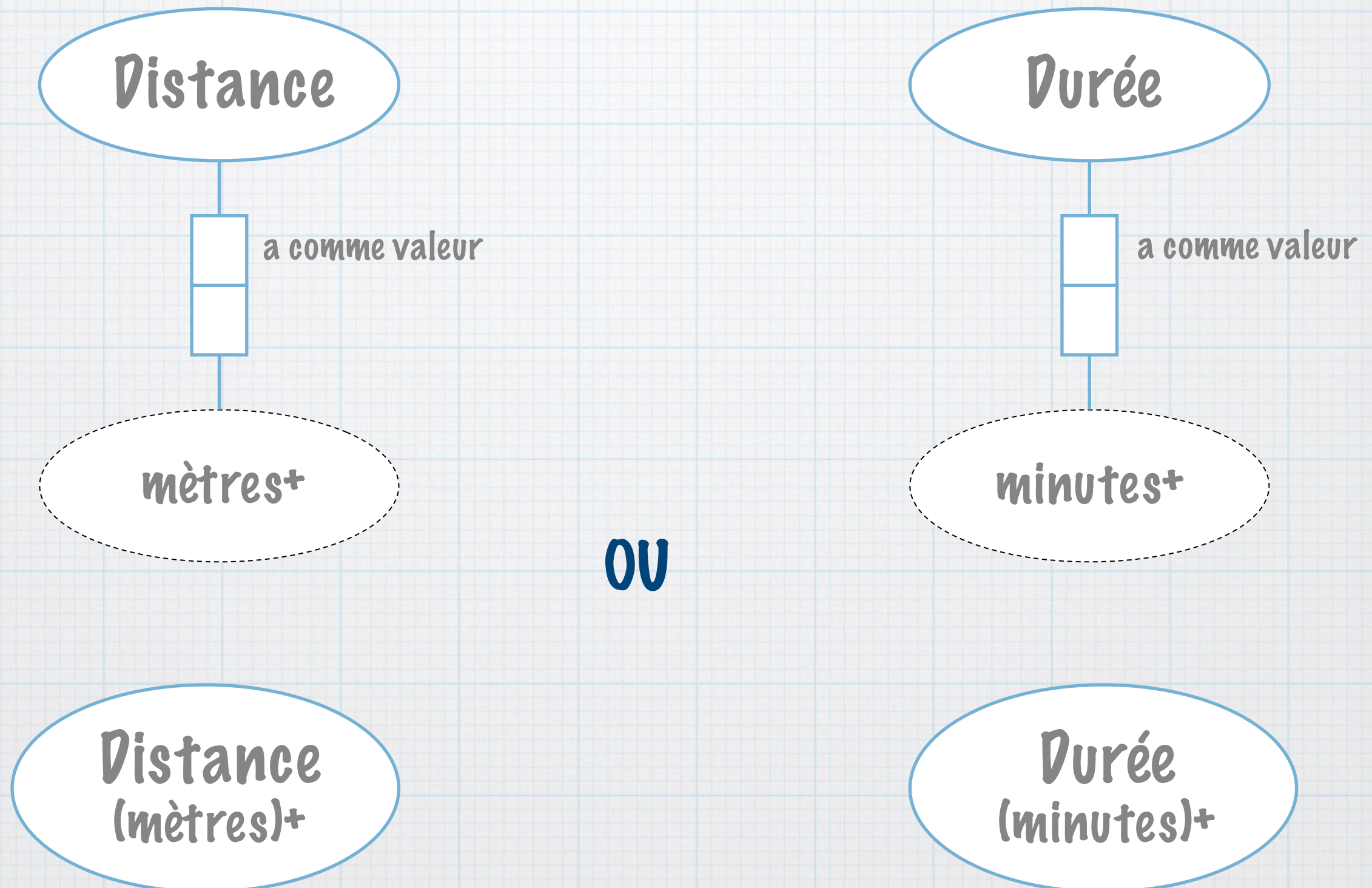


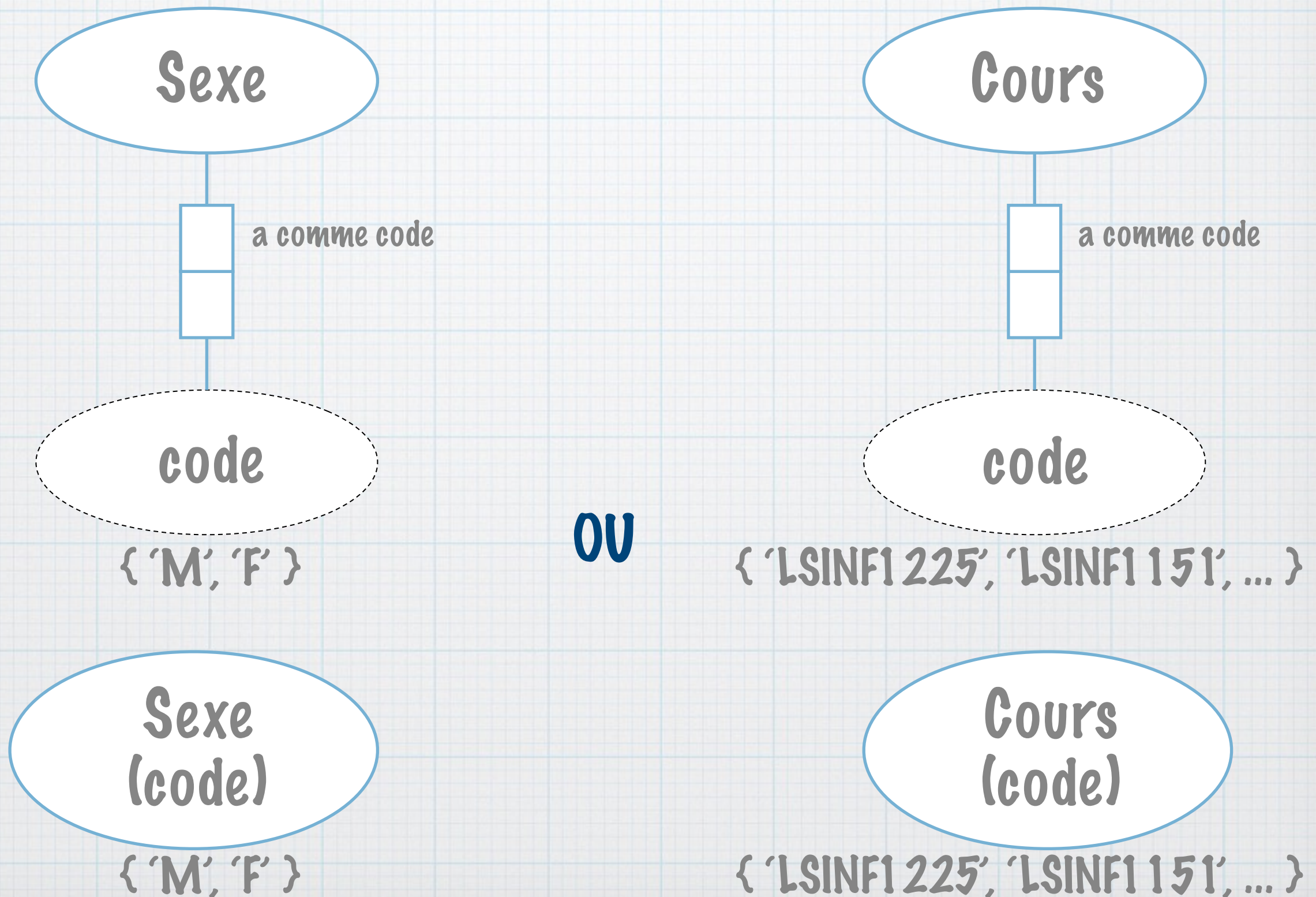
Schéma de référence composé



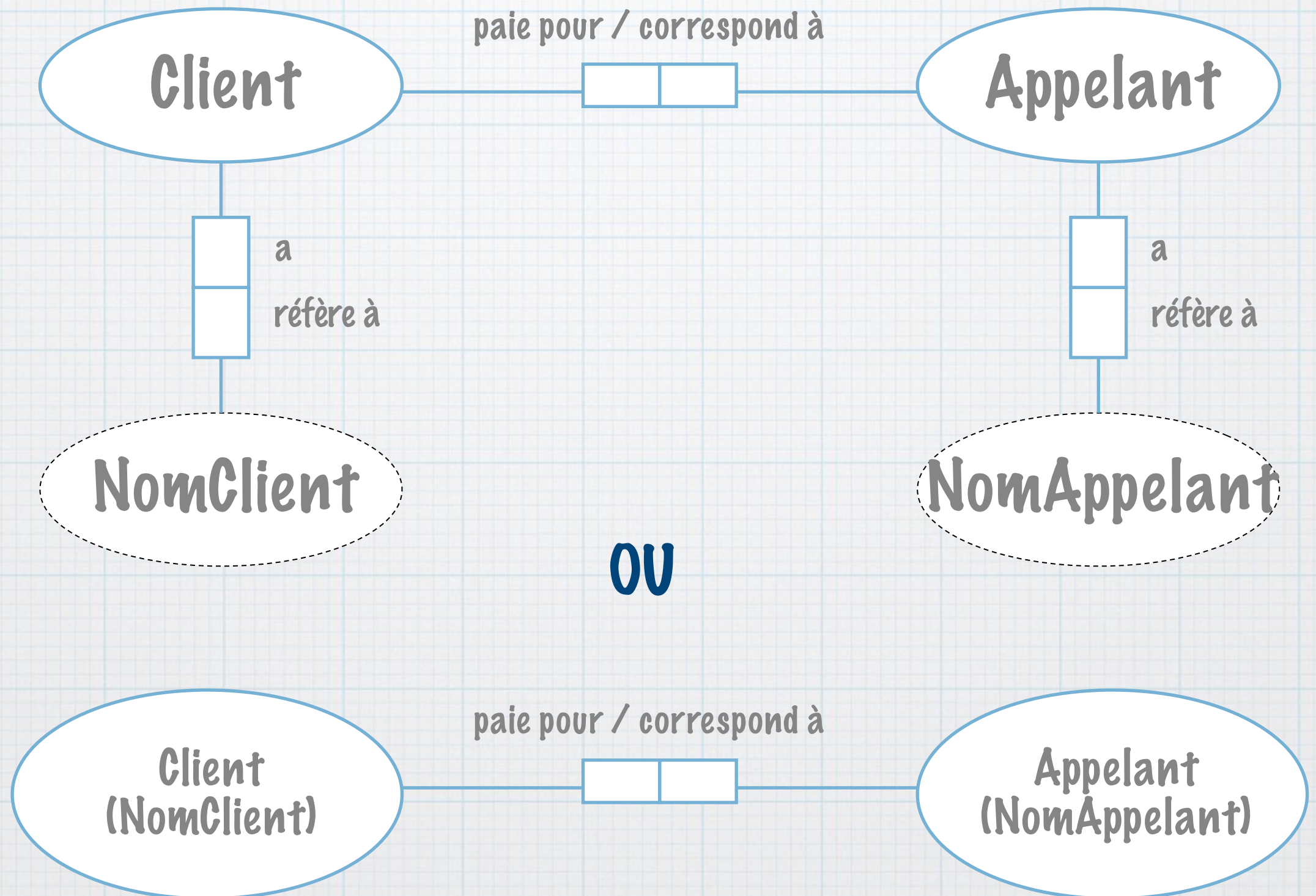
Types de valeurs numériques



Types d'énumération



Exemple d'un schéma conceptuel



Exemple d'un schéma conceptuel

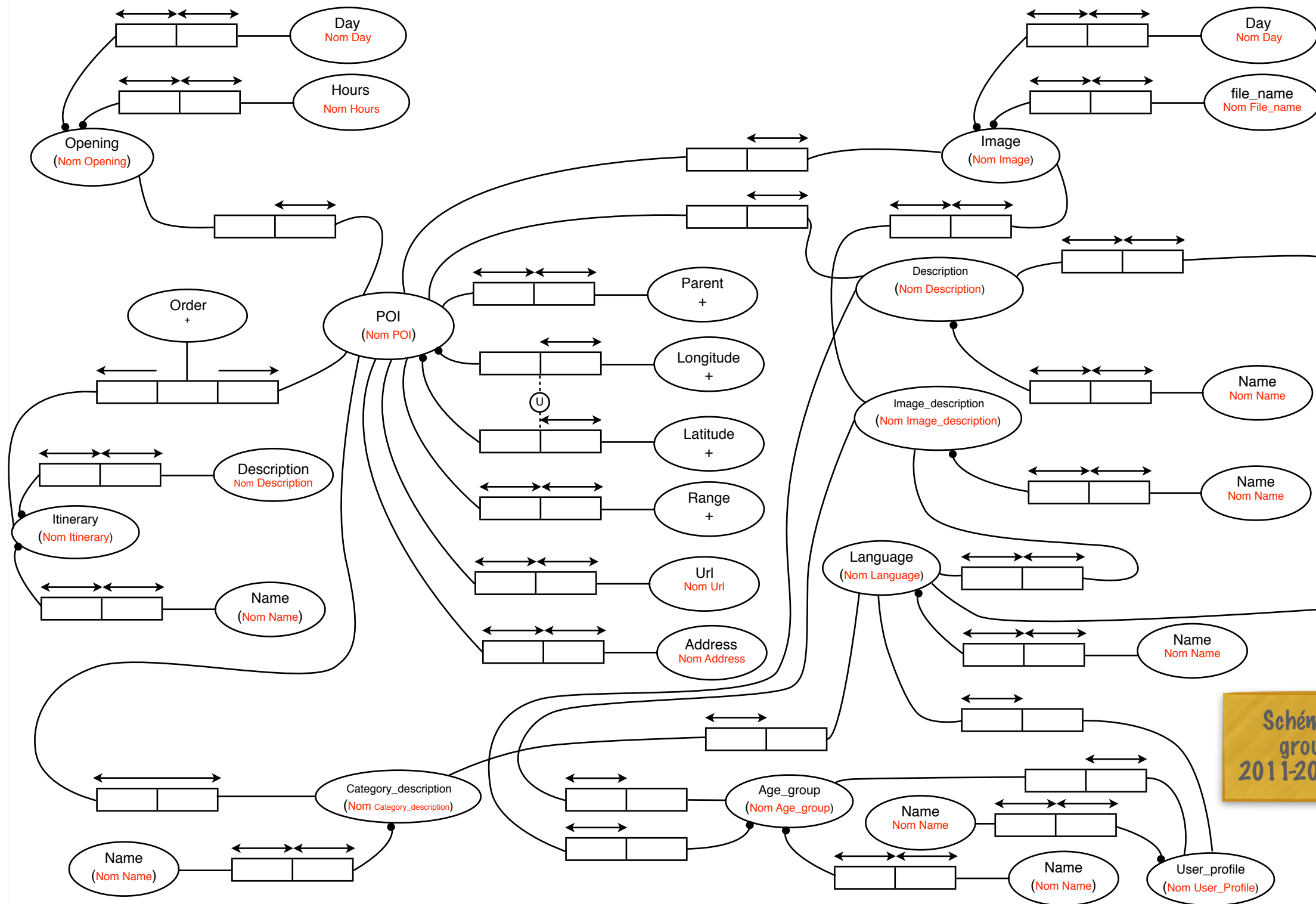
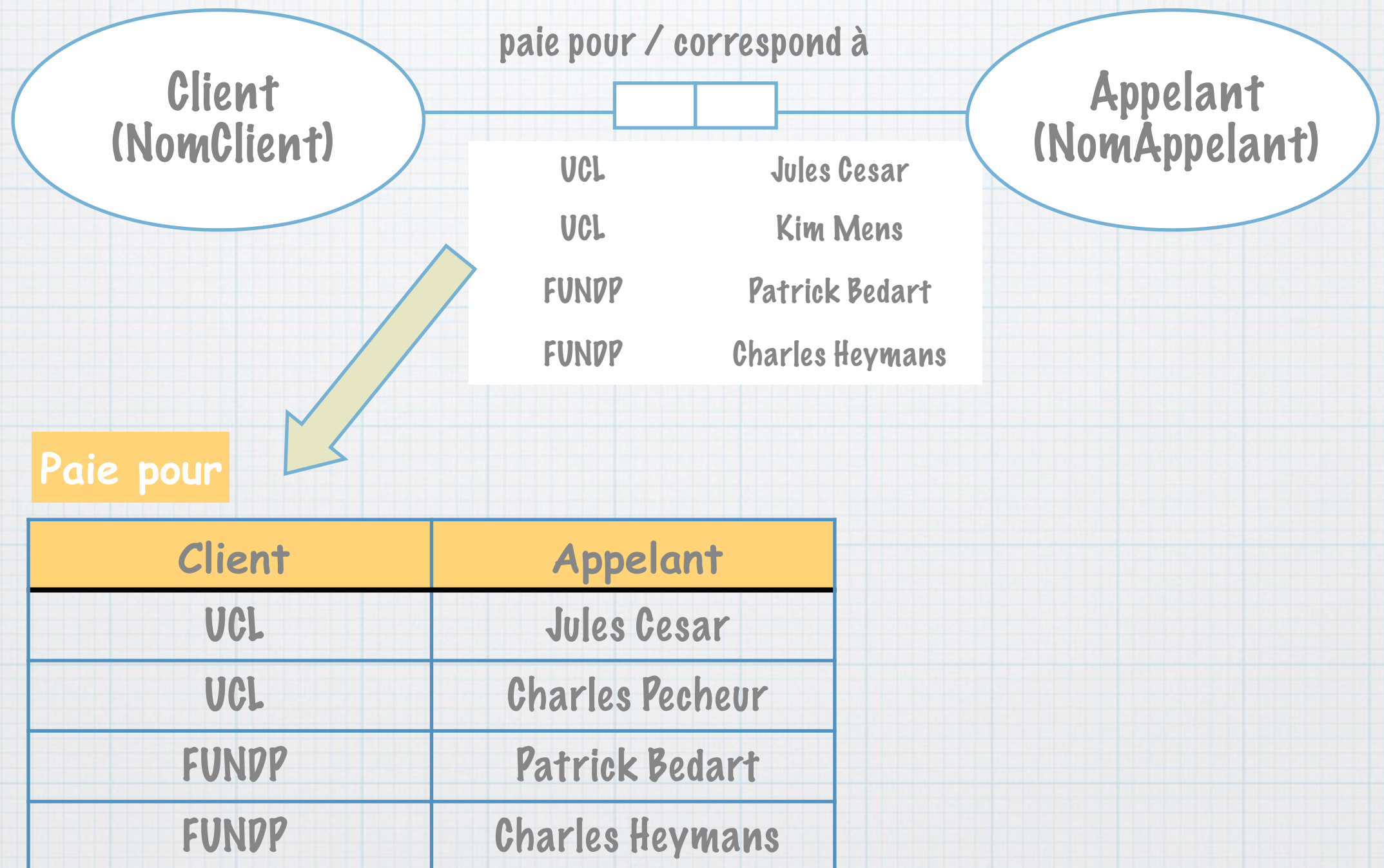


Schéma ORM créé par un groupe d'étudiants en 2011-2012, reproduit tel quel.

Vérifiez l'exactitude du schéma en ajoutant une "population"

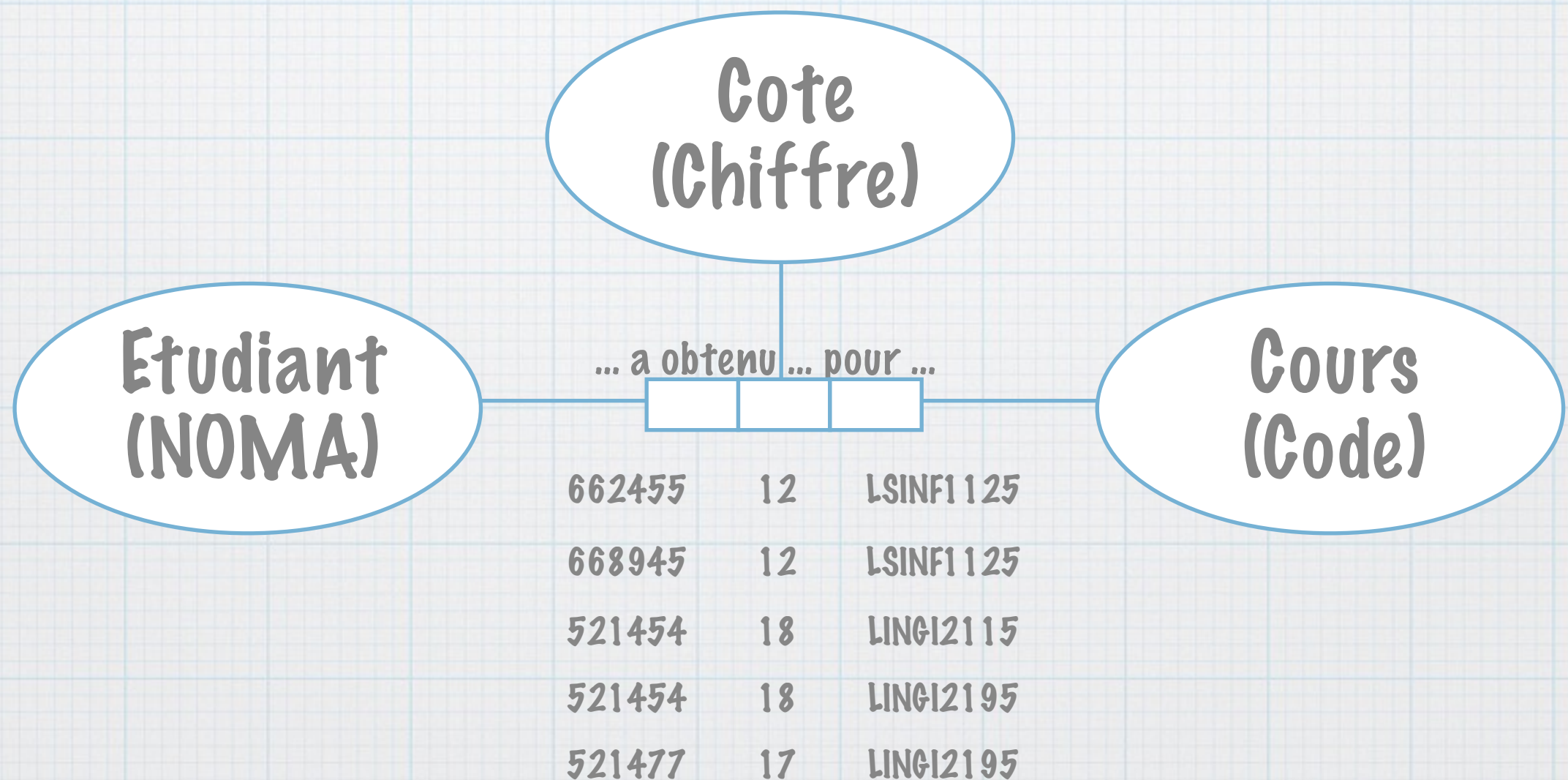


Pourquoi ajouter une population?

- ❑ Pour chaque type de fait, ajoutez quelques données originales
- ❑ Afin de détecter des diagrammes insensés
- ❑ Ainsi que pour clarifier certains contraintes d'unicité (et autres)
- ❑ On ajoute une "table de faits" pour chaque type de fait et on rempli les valeurs dans les colonnes correspondantes de cette table.

Exemple d'une relation ternaire

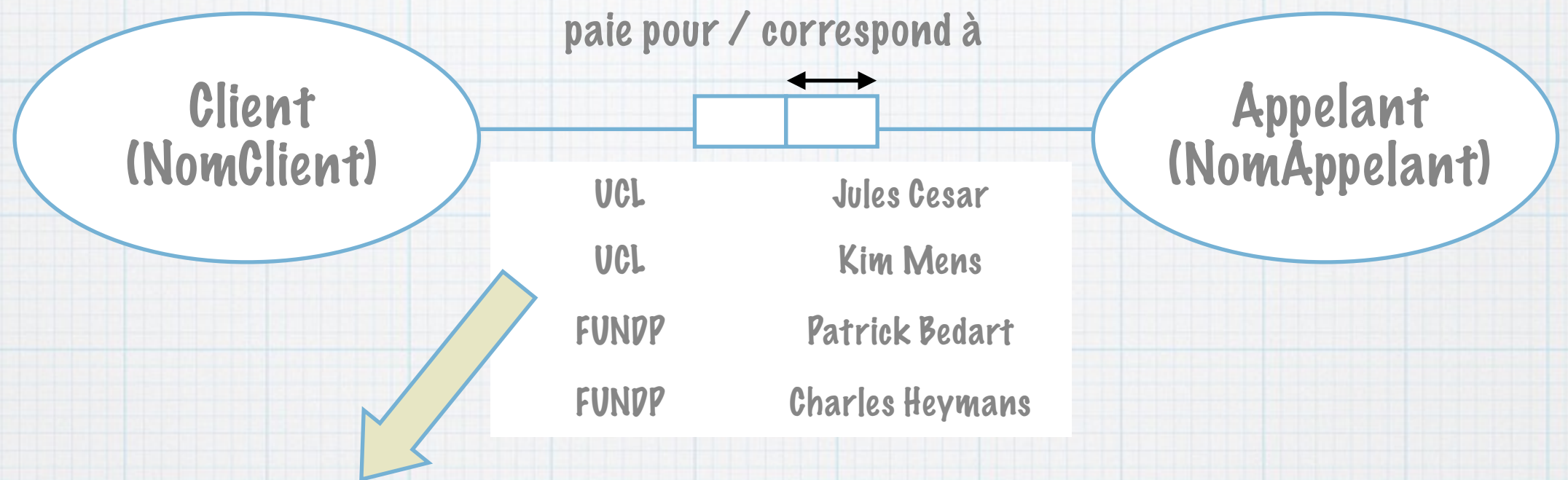
- L'Etudiant avec NOMA 662455 a obtenu une Cote de 12 pour le Cours avec code "LSINF1125"



Etape 4: Ajoutez des contraintes d'unicité

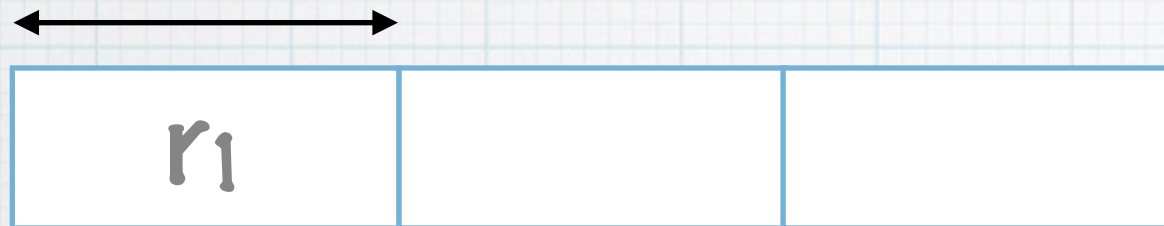
- **Ajoutez des contraintes d'unicité aux relations**
 - **Quelles "colonnes" peuvent avoir des valeurs doublons?**
- **Vérifiez si les contraintes n'impliquent pas une arité inférieure pour la relation**

Exemple

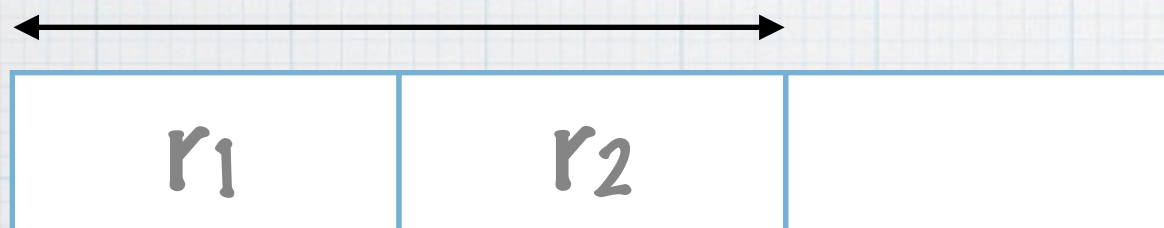


- On peut "déduire" de la population pour cette relation que le rôle du Client n'est pas unique pour cette relation (un client peut avoir plusieurs appelants)
- Mais peut être le rôle d'Appelant est unique pour cette relation ?
 - Un appelant peut-il correspondre à plusieurs clients?
 - Non, car sinon l'opérateur ne sait pas à quel client envoyer la facture.

Les contraintes d'unicité



- Une flèche au dessus d'un rôle signifie que
 - la valeur pour ce rôle doit être unique pour cette relation

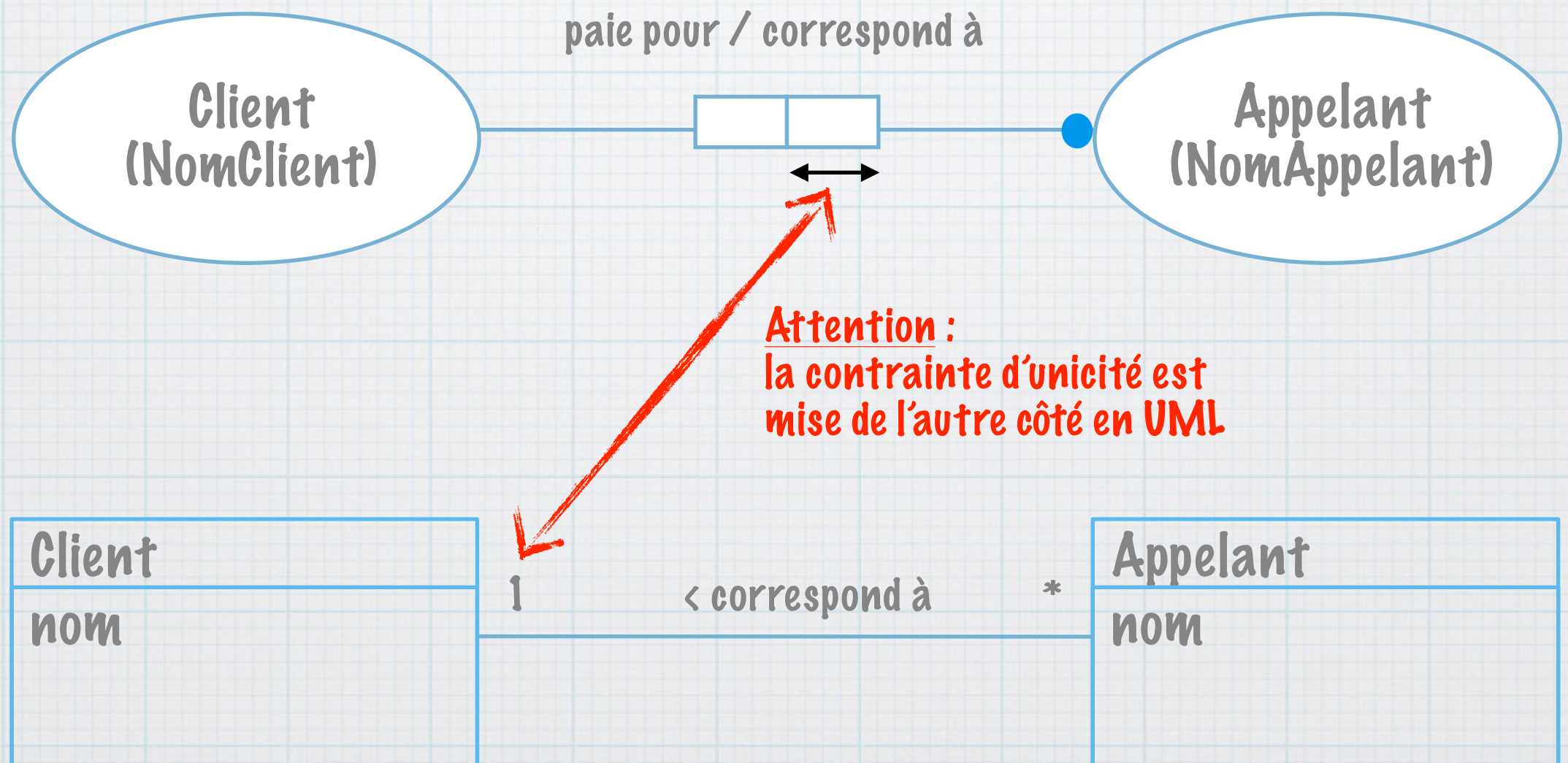


- Une flèche au dessus de plusieurs rôles signifie que
 - l'ensemble des valeurs pour ces rôles doit être unique pour cette relation

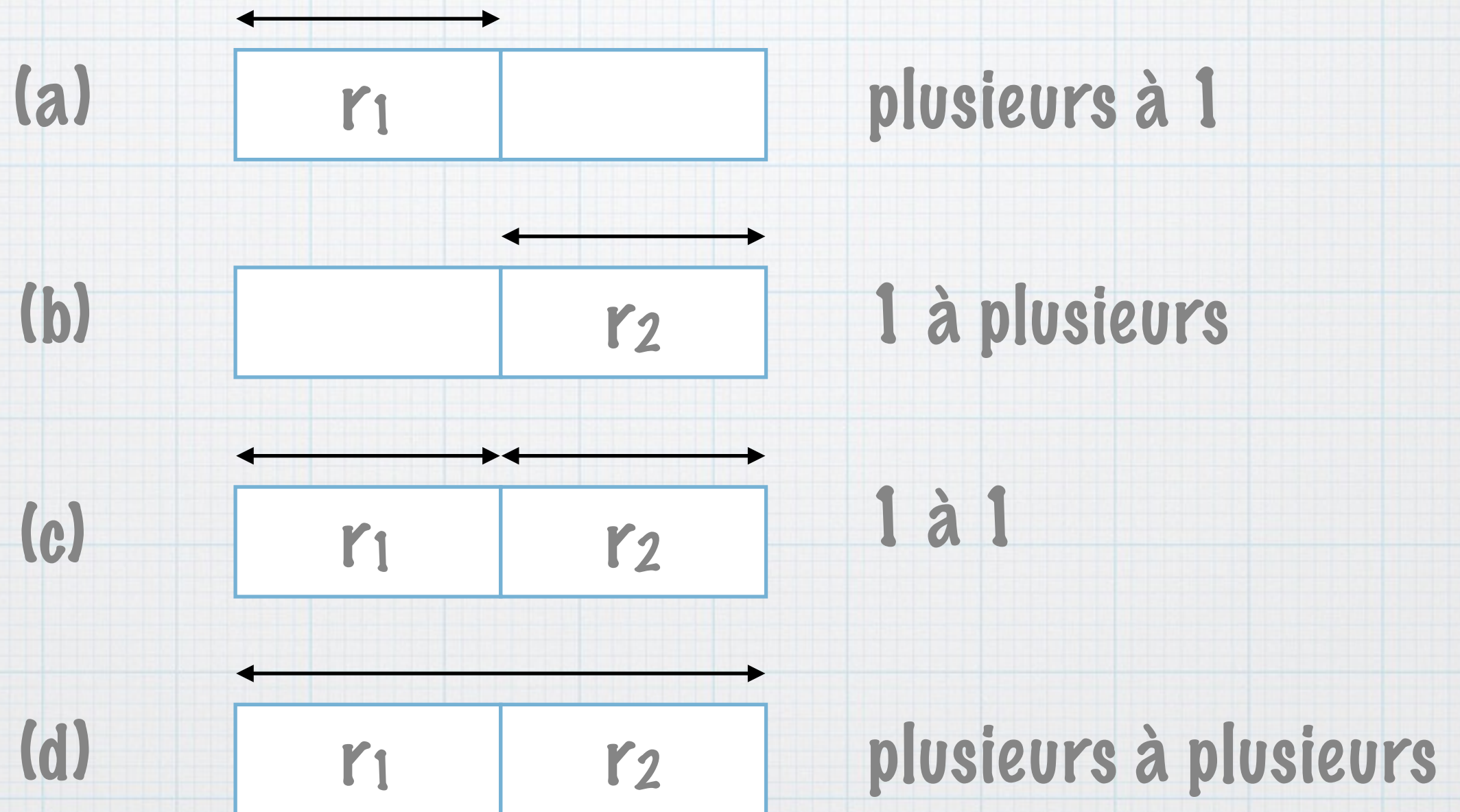


ORM vs. UML

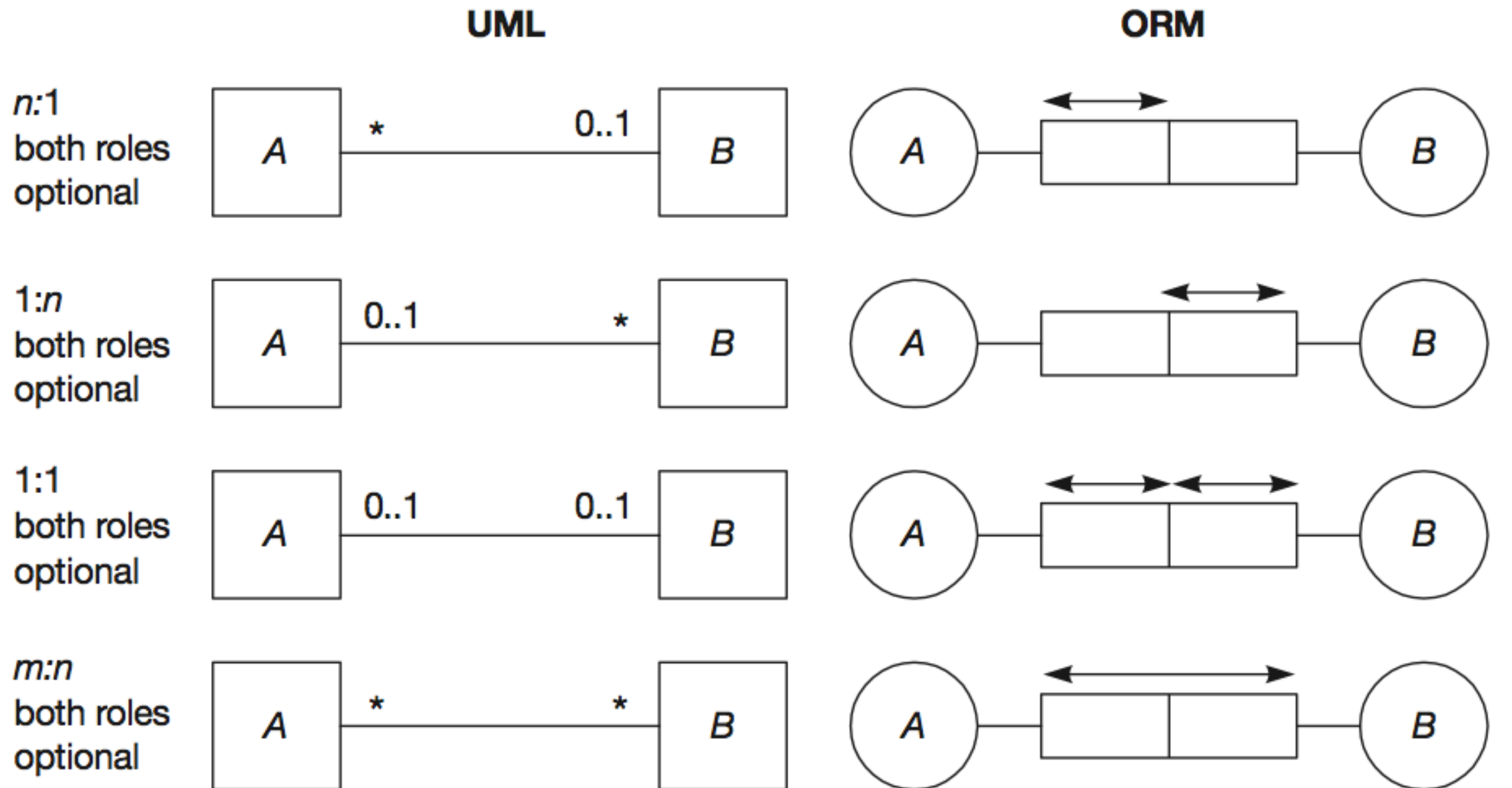
- Comment exprimer: “pour un même client, je peux avoir plusieurs appelants, mais pour chaque appelant il n’y a qu’un seul client?”



Les 4 contraintes d'unicité pour une relation binaire



ORM vs. UML



Relation binaire cas général

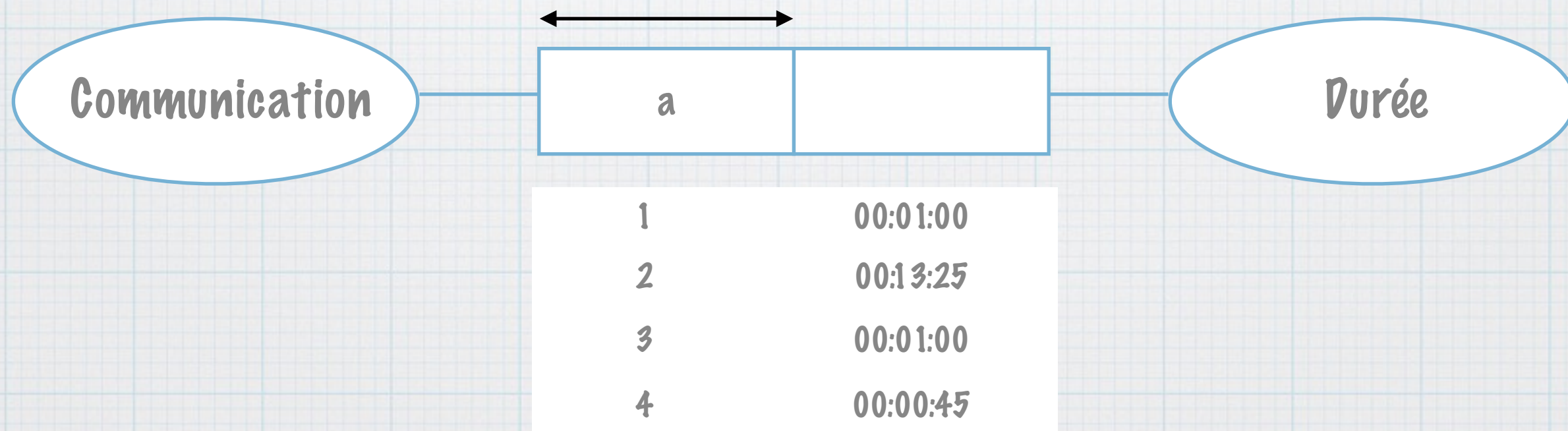
- Une relation binaire est un ensemble de paires (a,b)
 - ◆ Impossible d'avoir deux fois la même paire (a,b)

a1	b1
a1	b2
a2	b1
a2	b2
a1	b1
a3	b3

Relation binaire plusieurs à 1

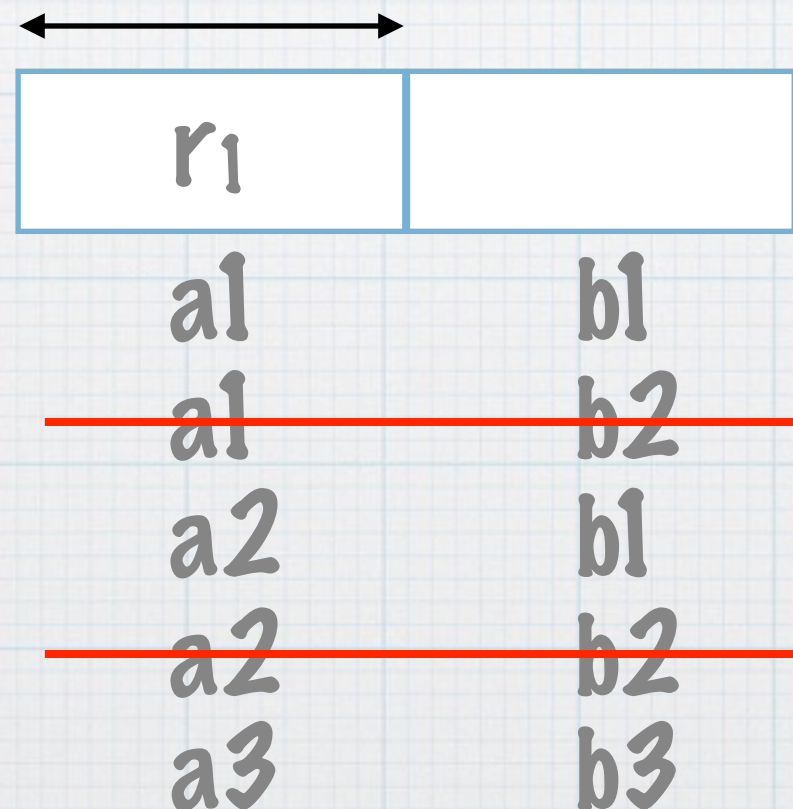
■ Exemple :

- ◆ Plusieurs Communications différentes peuvent avoir la même Durée
- ◆ Mais chaque Communication (appel téléphonique) n'a qu'une seule Durée



Relation binaire plusieurs à 1

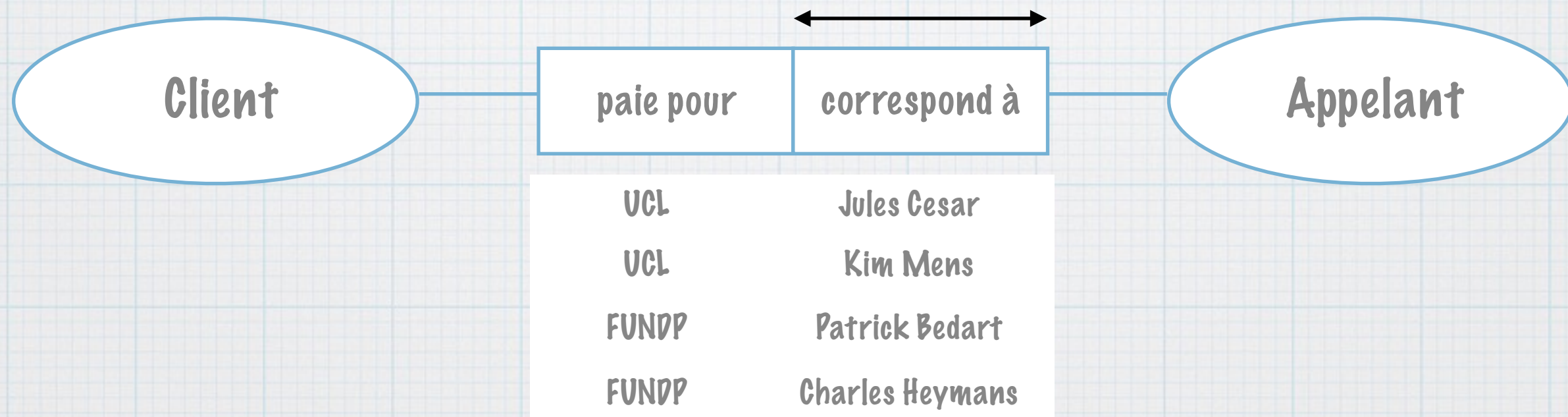
- Pas de doublons permis pour le rôle r_1
 - ◆ Plusieurs a_i 's différents peuvent correspondre à un même b_i
 - ◆ Mais à chaque a_i correspond au plus un b_i



Relation binaire 1 à plusieurs

■ Exemple :

- ◆ Chaque Client peut avoir plusieurs appelants
- ◆ Mais chaque Appelant correspond à un seul Client

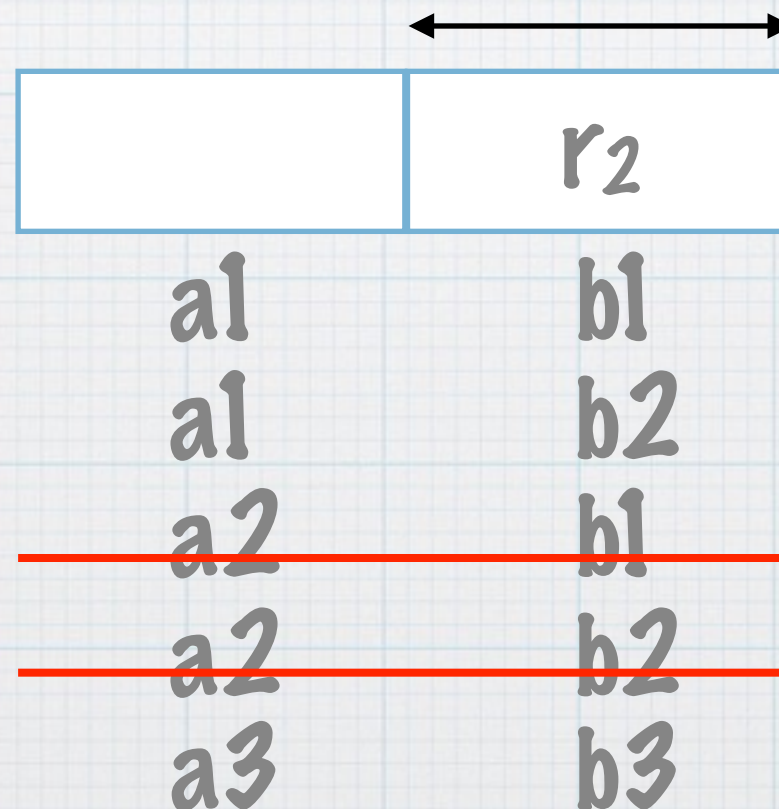


Relation binaire 1 à plusieurs

■ Pas de doublons permis pour le rôle r_2

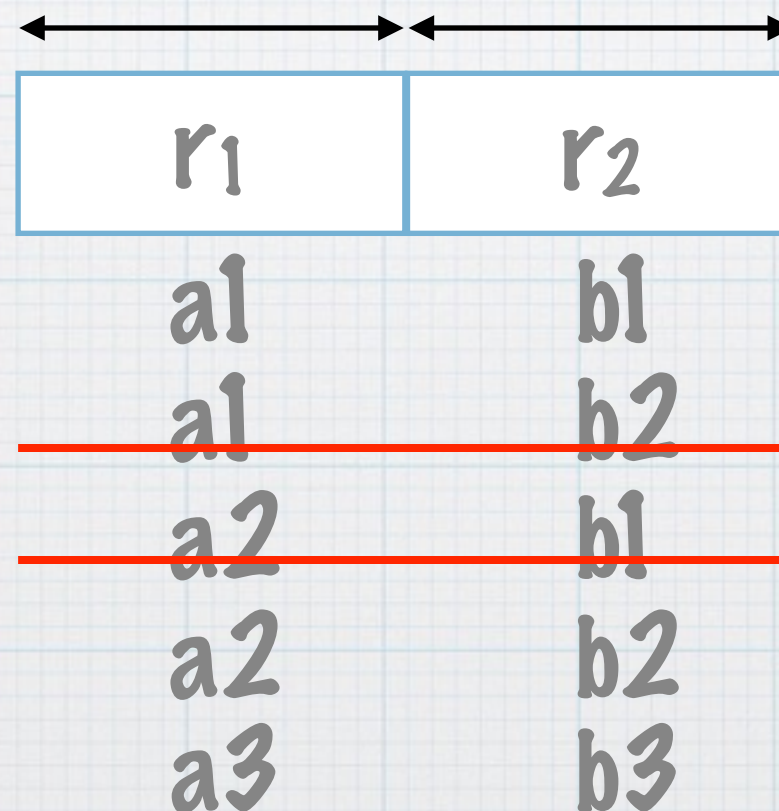
◆ Un a_i peut correspondre à plusieurs b_i 's

◆ Mais à chaque b_i correspond au plus un a_i



Relation binaire 1 à 1

- Pas de doublons permis, ni pour le rôle r_1 , ni pour le rôle r_2
 - ◆ Chaque a_i correspond à exactement un b_i et vice versa



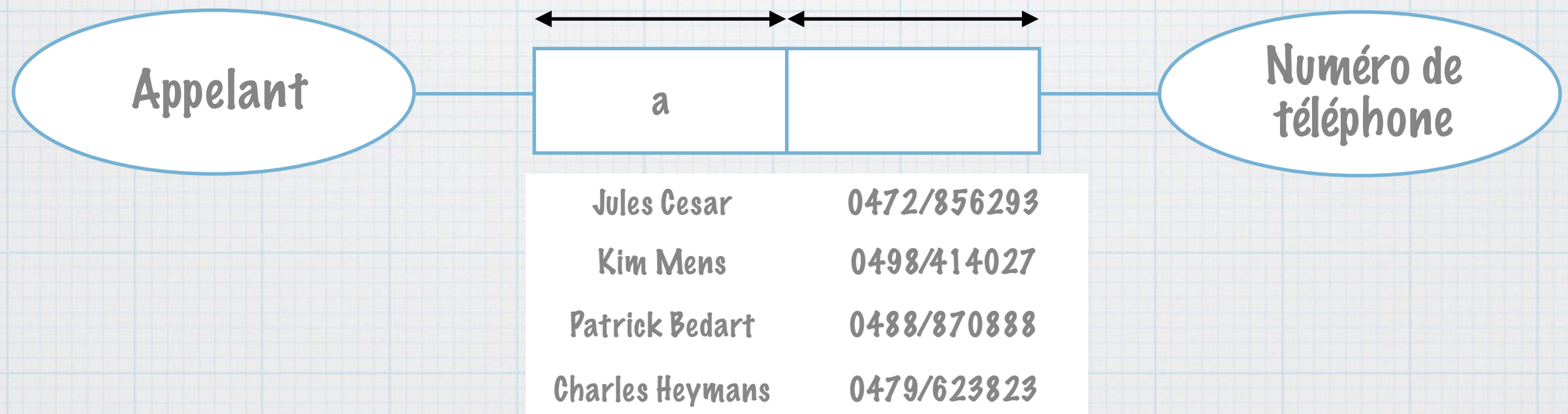
r_1	r_2
a_1	b_1
a_1	b_2
a_2	b_1
a_2	b_2
a_3	b_3

Relation binaire

1 à 1

■ Exemple :

- ◆ Chaque numéro de téléphone correspond à un appelant unique (un même numéro de téléphone ne peut pas être partagé par plusieurs appelants)
- ◆ Chaque appelant a un numéro de téléphone unique*

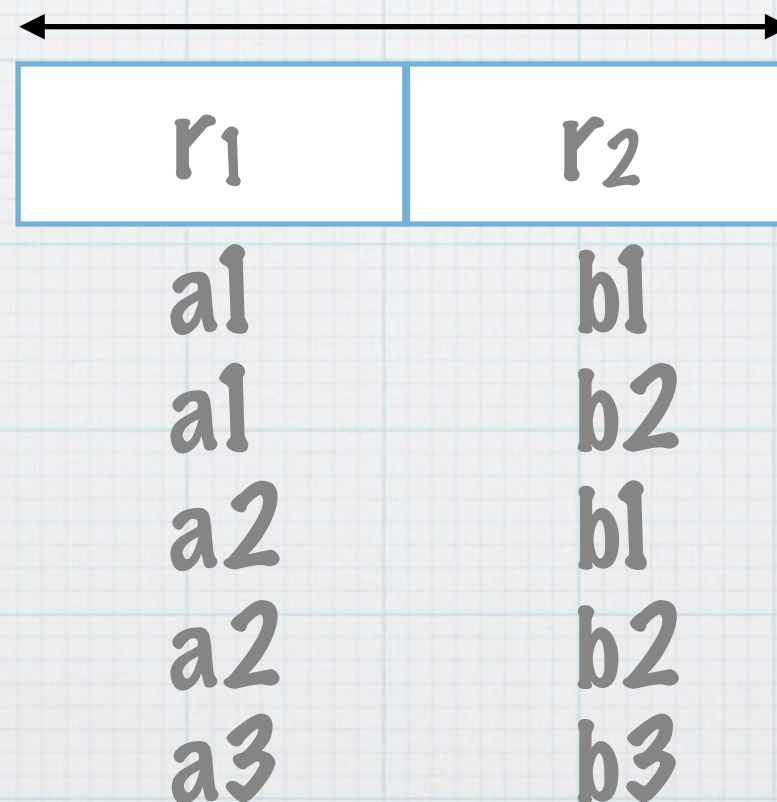


* avec cette contrainte, si on veut permettre d'associer plusieurs numéros de téléphone à une même personne, il faut créer plusieurs appelants pour cette personne

Relation binaire plusieurs à plusieurs

■ Pas de contraintes

- ◆ sauf qu'il est impossible d'avoir deux fois la même paire (a_i, b_i)
- ◆ Chaque a_i peut correspondre à plusieurs b_i 's et vice versa



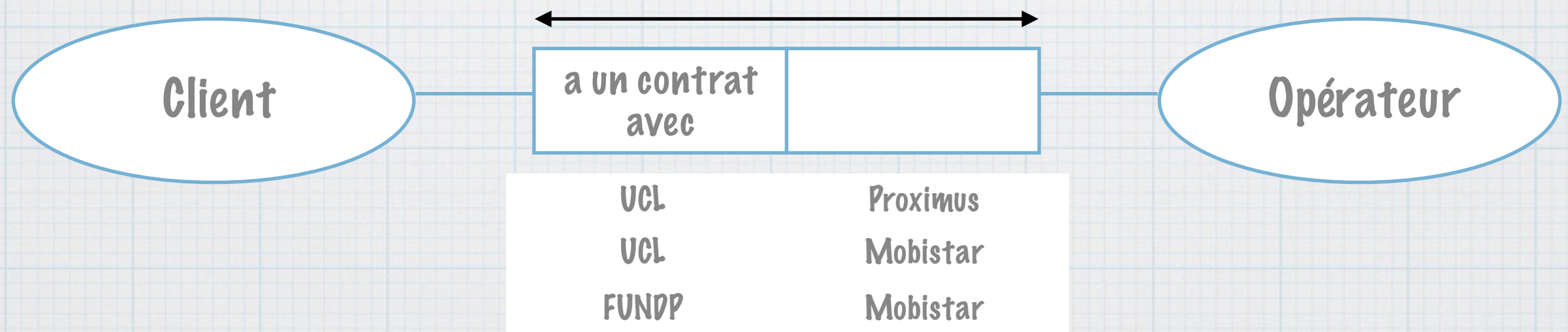
Même si conceptuellement il n'y a pas de différence entre une flèche sur tous les rôles ou pas de flèche du tout, il est mieux d'expliciter la ligne.

(Aucune ligne peut signifier qu'on n'a pas encore réfléchi à la contrainte.)

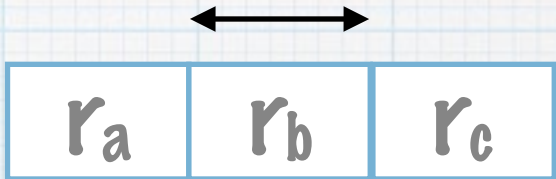
Relation binaire plusieurs à plusieurs

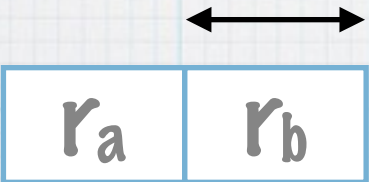
■ Exemple :

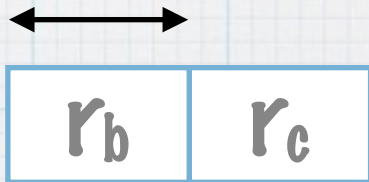
- ◆ Un Client (UCL) peut avoir un contrat avec plusieurs Opérateurs (Proximus, Mobistar)
- ◆ Un Opérateur (Proximus) peut avoir plusieurs clients (UCL, FUNDP)



Contraintes d'unicité pour une relation ternaire

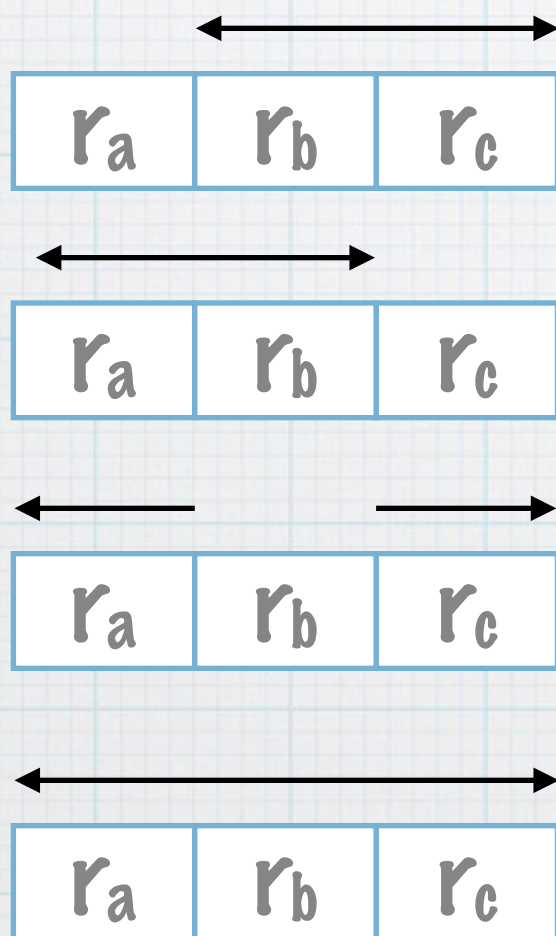
- ❑ Vérifiez l'arité : une relation ternaire ne peut jamais avoir une contrainte d'unicité "simple"
- ❑ Suppose qu'on a, par exemple : 
- ❑ Alors chaque b est unique pour cette relation, et on peut couper la relation en deux relations binaires:



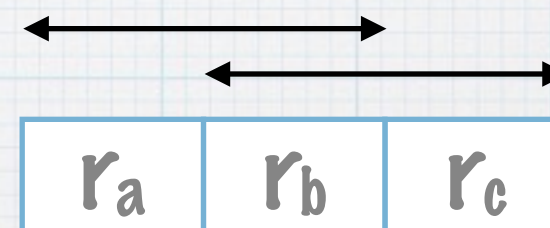

- ❑ Car à partir de ces 2 relations on peut reconstruire la relation ternaire en regroupant les a et c avec le même b
- ❑ Un fait élémentaire ternaire ne peut donc jamais avoir une contrainte d'unicité "simple"

Contraintes d'unicité pour une relation ternaire

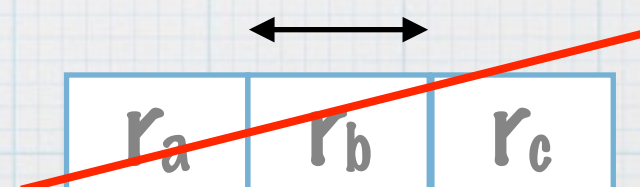
- Contraintes d'unicité possibles pour une relation ternaire



Ou une combinaison, comme



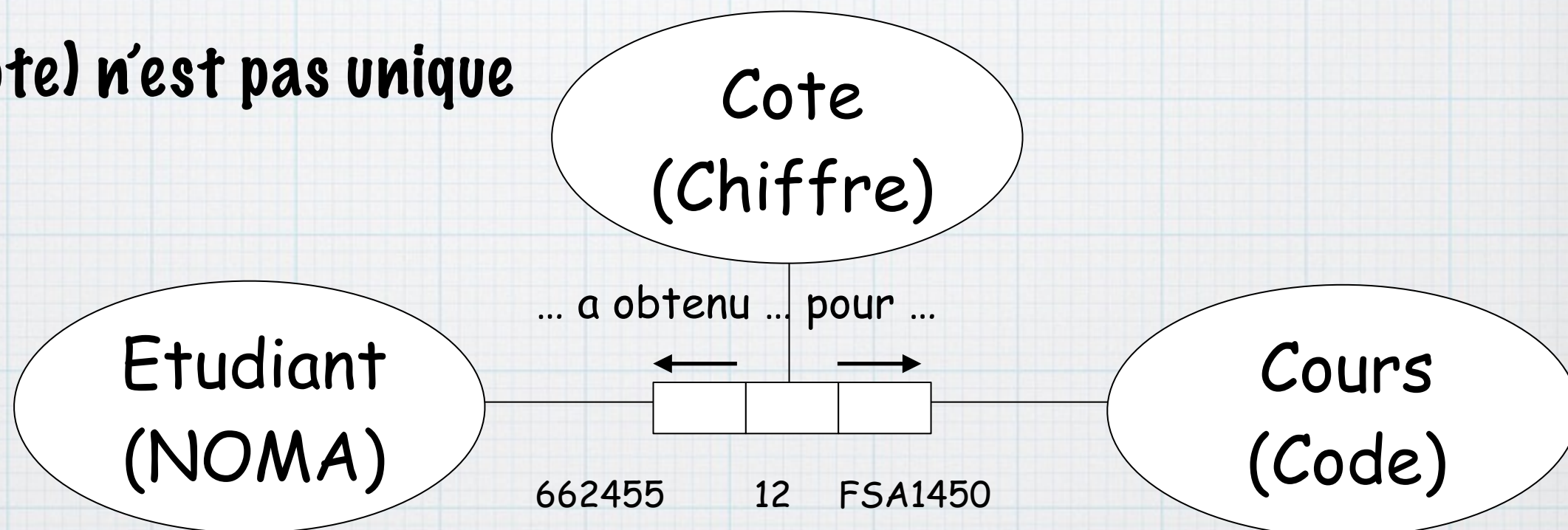
Mais jamais



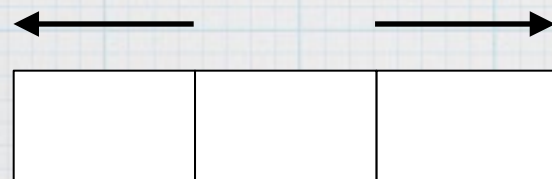
Exemple d'une relation ternaire

- Chaque pair (Etudiant, Cours) est unique
- (Etudiant, Cote) n'est pas unique
- (Cours, Note) n'est pas unique

Hypothèse : on n'encode que la dernière cote de l'étudiant



Donc:



662455	12	FSA1450
668945	12	FSA1450
521454	18	INGI2115
521454	18	INGI2195
521477	17	INGI2195

Contraintes d'unicité pour une relation n -aire

- **Vérifiez l'arité:**
 - Si la relation correspond à un fait élémentaire, la contrainte d'unicité porte sur au moins $n-1$ rôles.
 - Car, si la contrainte d'unicité porte sur moins de $n-1$ rôles, alors la relation ne correspond pas à un fait élémentaire et peut être coupée en plusieurs relations plus élémentaires

Ajoutez d'autres contraintes, si nécessaire

- ❑ **Rôles obligatoires**
- ❑ **Contrainte d'unicité externe**
- ❑ **Contrainte de sous-ensemble**

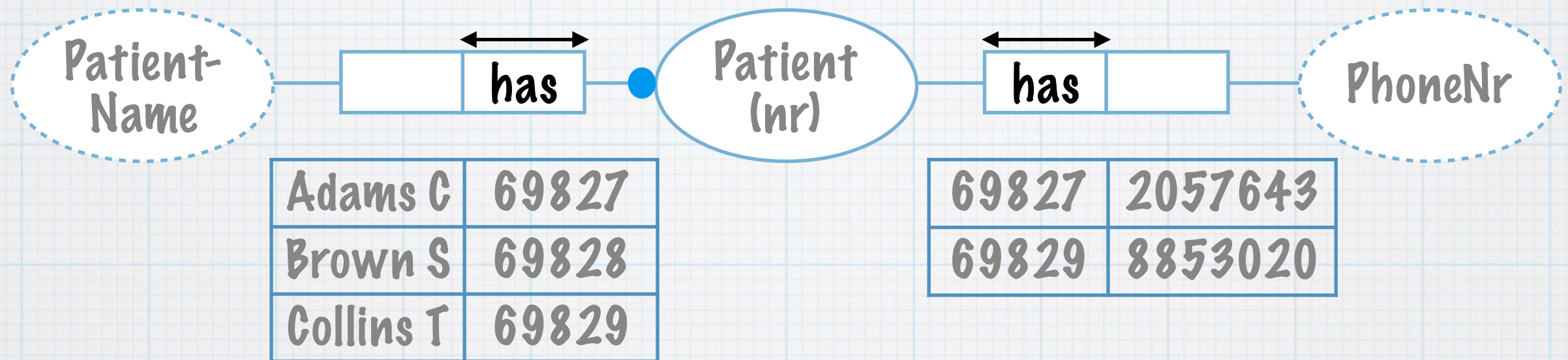
Rôles obligatoires

- Voici quelques données sur les patients d'un hôpital :

PatientNr	Patient	Phone
69827	Adams C	2057643
69828	Brown S	?
69829	Collins T	8853020

- Contraintes :
 - Un patient ne peut pas avoir plusieurs noms ou numéros de téléphone
 - Chaque patient doit avoir un nom
 - Chaque patient peut (ne doit pas) avoir un numéro de téléphone

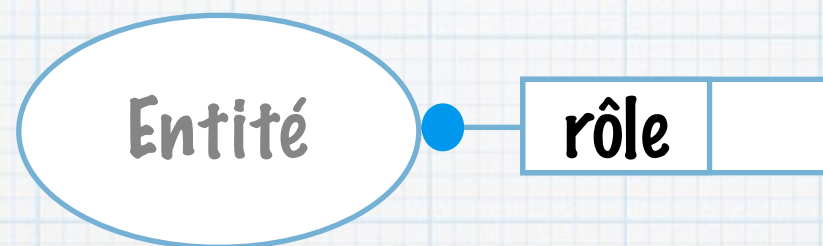
Rôles obligatoires



❑ Contraintes :

- ❑ Un patient ne peut pas avoir plusieurs noms ou numéros de téléphone
- ❑ Chaque patient doit avoir un nom
- ❑ Chaque patient peut (ne doit pas) avoir un numéro de téléphone

Rôles obligatoires

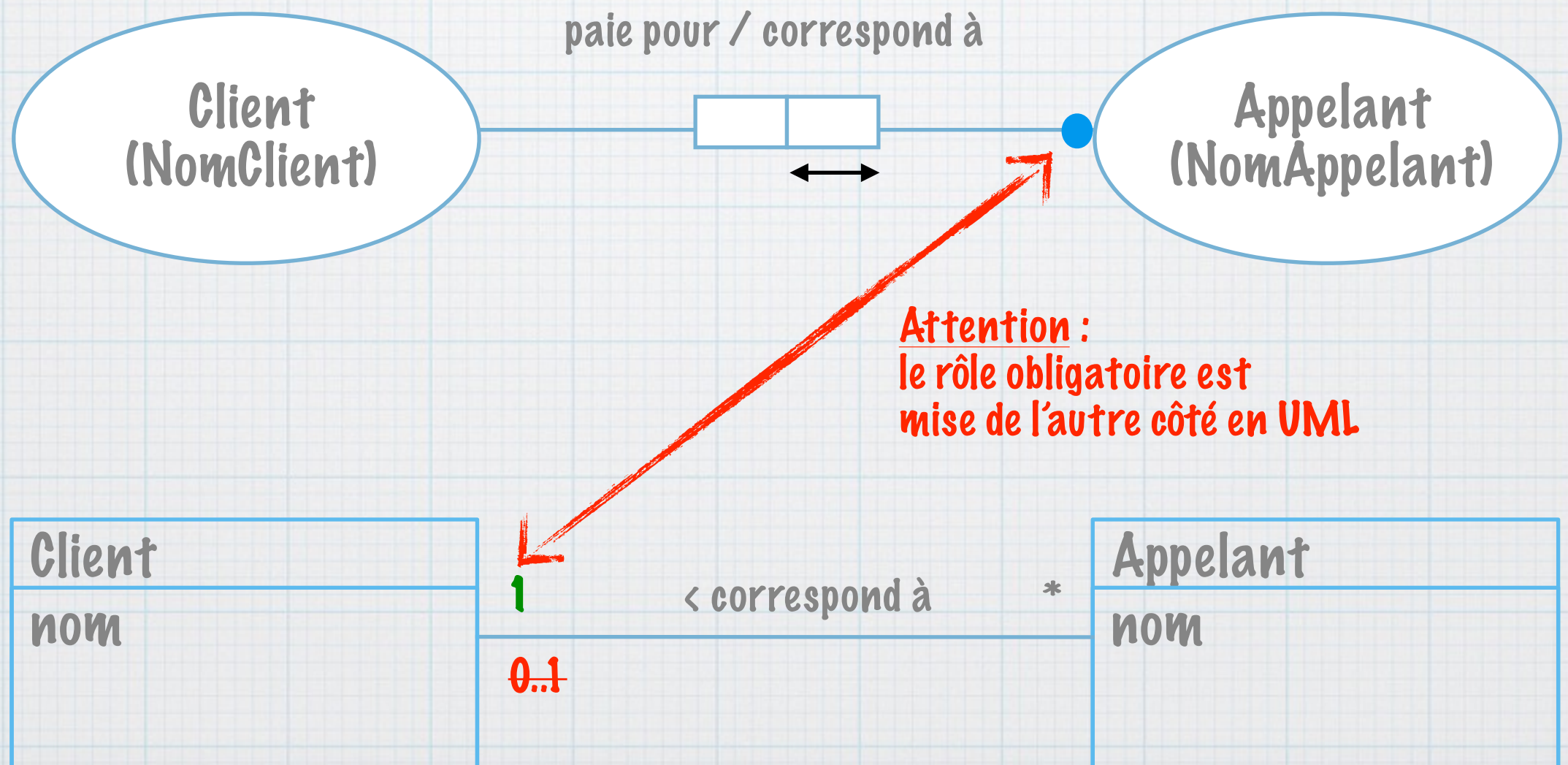


- **Un rôle r est obligatoire pour la population d'une Entité E si**
 - **Chaque instance de cette entité qui est encodée dans la base de données joue ce rôle**
 - **$\text{population}(r) = \text{population}(E)$**

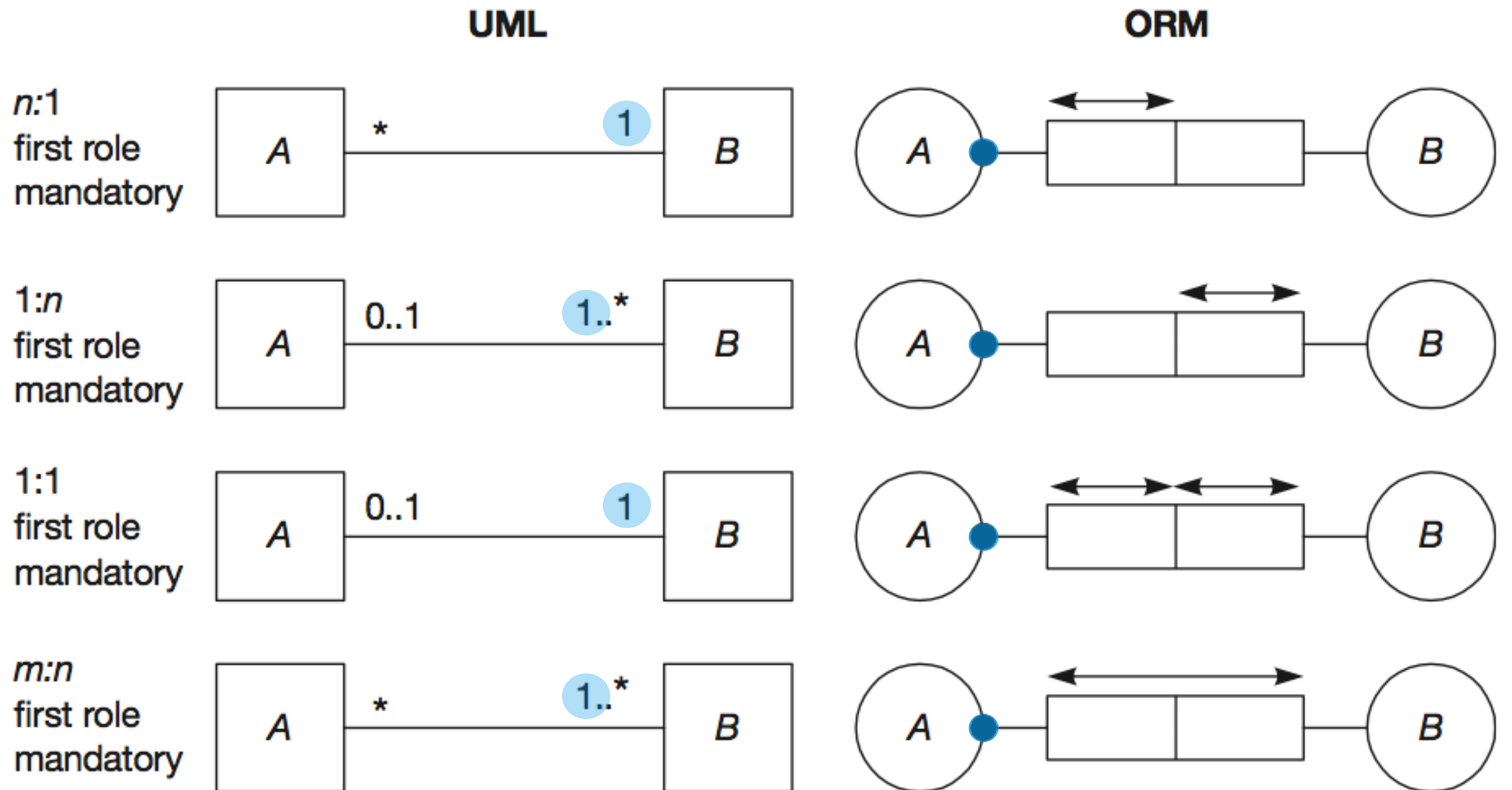


ORM vs. UML

- Comment exprimer:
"chaque appelant doit correspondre à un seul client?"



ORM vs. UML



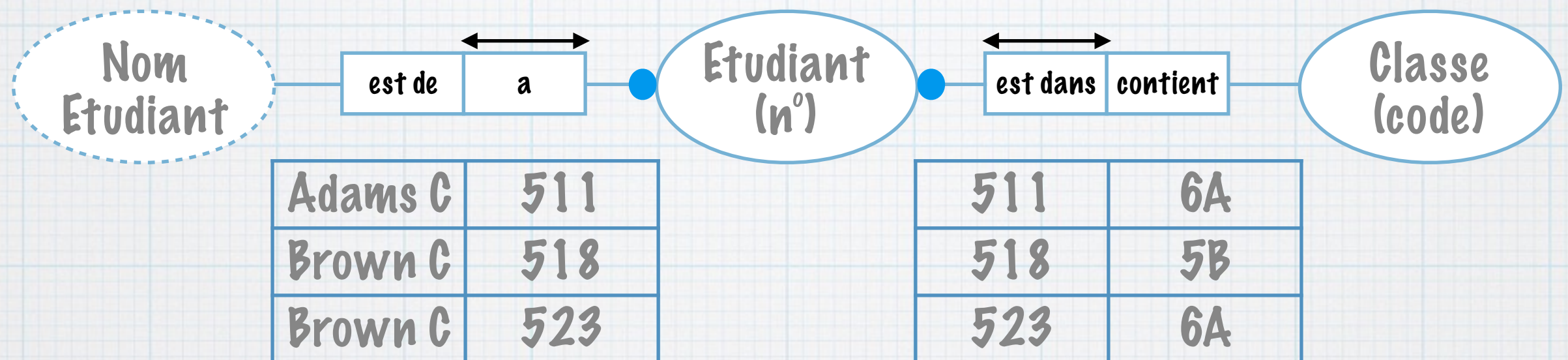
Contrainte d'unicité externe

- Voici quelques données sur les élèves d'une école :

EtudiantN°	Nom	Classe
511	Adams C	6A
518	Brown C	5B
523	Brown C	6A

- Contraintes :
 - Chaque étudiant a exactement un nom
Chaque étudiant est dans exactement une classe
 - On peut avoir deux étudiants avec le même nom à l'école (Brown C)
On ne peut pas avoir 2 étudiants avec le même nom dans une même classe (si cela arrive on ajoute un chiffre à leur nom, p.ex: Brown C2)

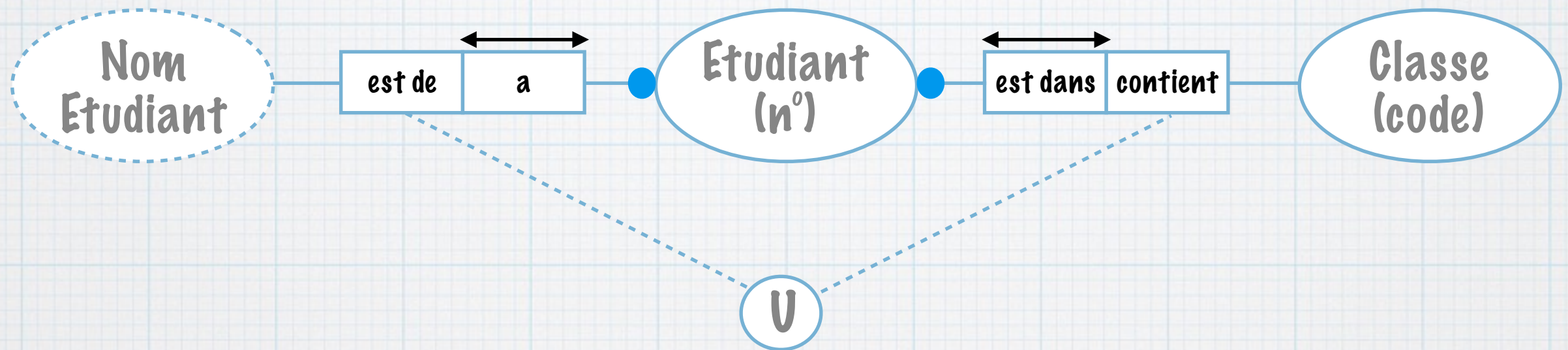
Contrainte d'unicité externe



□ Contraintes :

- Chaque étudiant a exactement un nom
Chaque étudiant est dans exactement une classe
- On peut avoir deux étudiants avec le même nom à l'école (Brown C)
On ne peut pas avoir 2 étudiants avec le même nom dans une même classe (si cela arrive on ajoute un chiffre à leur nom, p.ex: Brown C2)

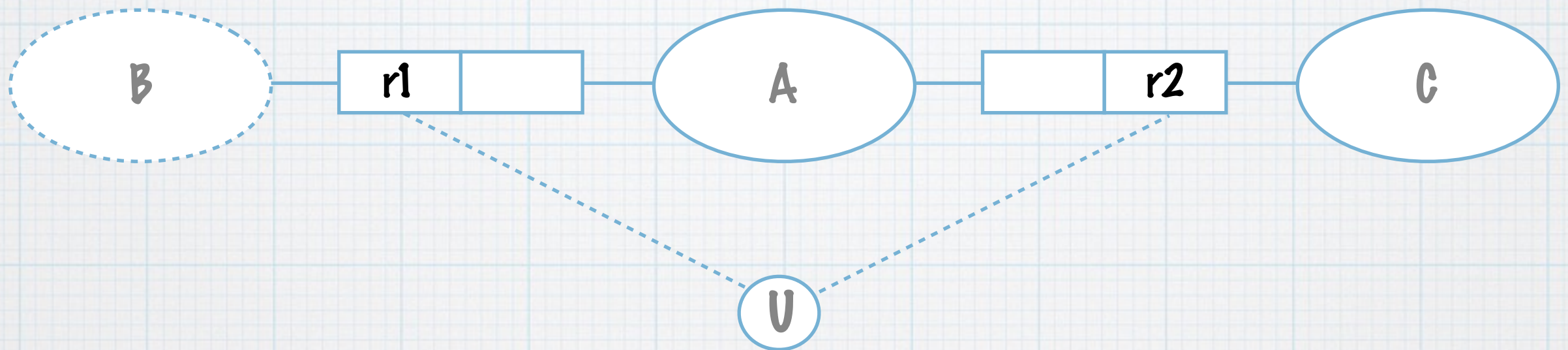
Contrainte d'unicité externe



□ Contraintes :

- Chaque étudiant a exactement un nom
Chaque étudiant est dans exactement une classe
- On peut avoir deux étudiants avec le même nom à l'école (Brown C)
On ne peut pas avoir 2 étudiants avec le même nom dans une même classe (si cela arrive on ajoute un chiffre à leur nom, p.ex: Brown C2)

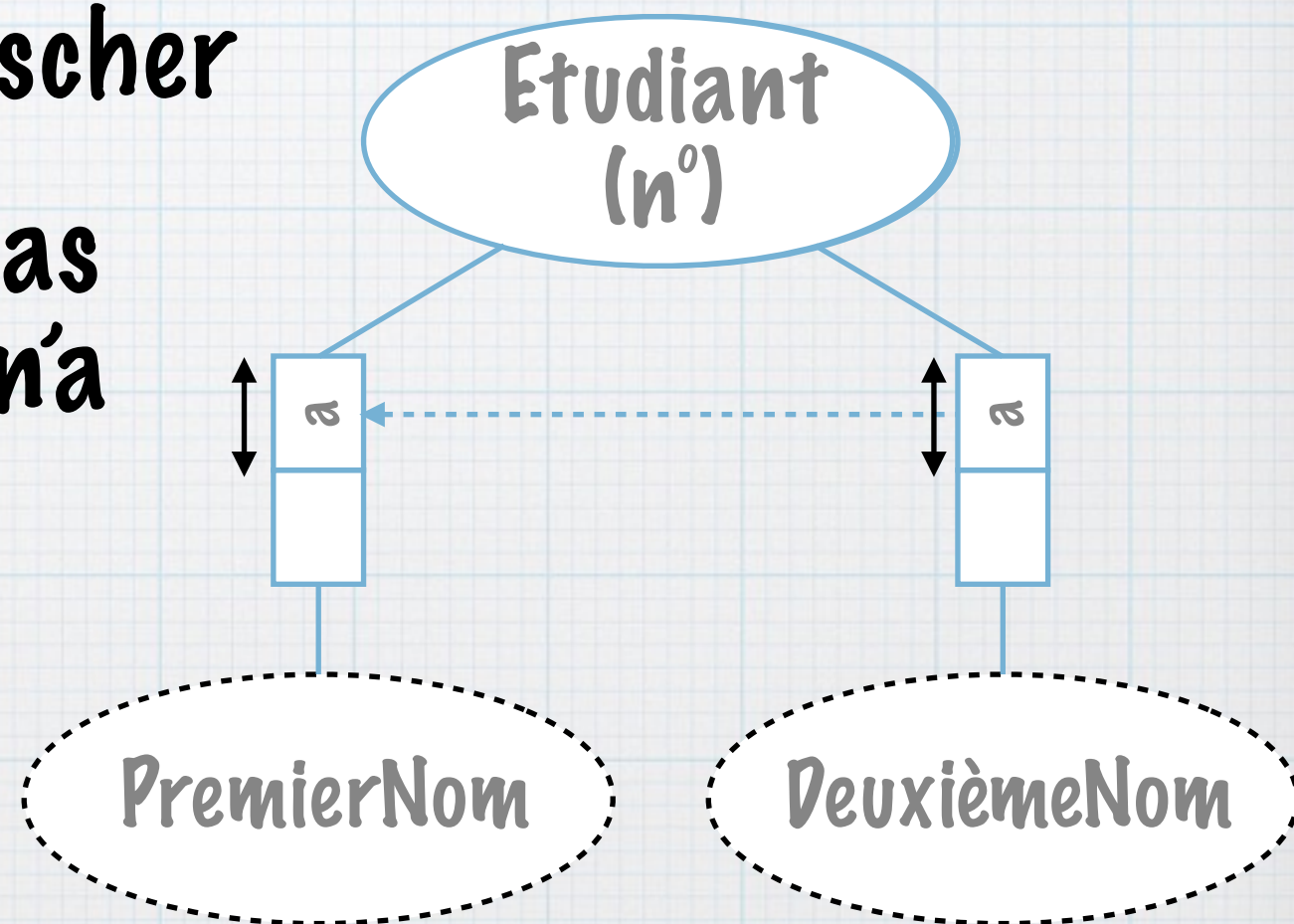
Contrainte d'unicité externe



- La contrainte d'unicité externe entre les 2 rôles r_1 et r_2 veut dire:
 - Chaque combinaison (b, c) correspond à maximum un a
 - Dans notre exemple: la combinaison du nom de l'étudiant et de sa classe détermine l'étudiant de façon unique

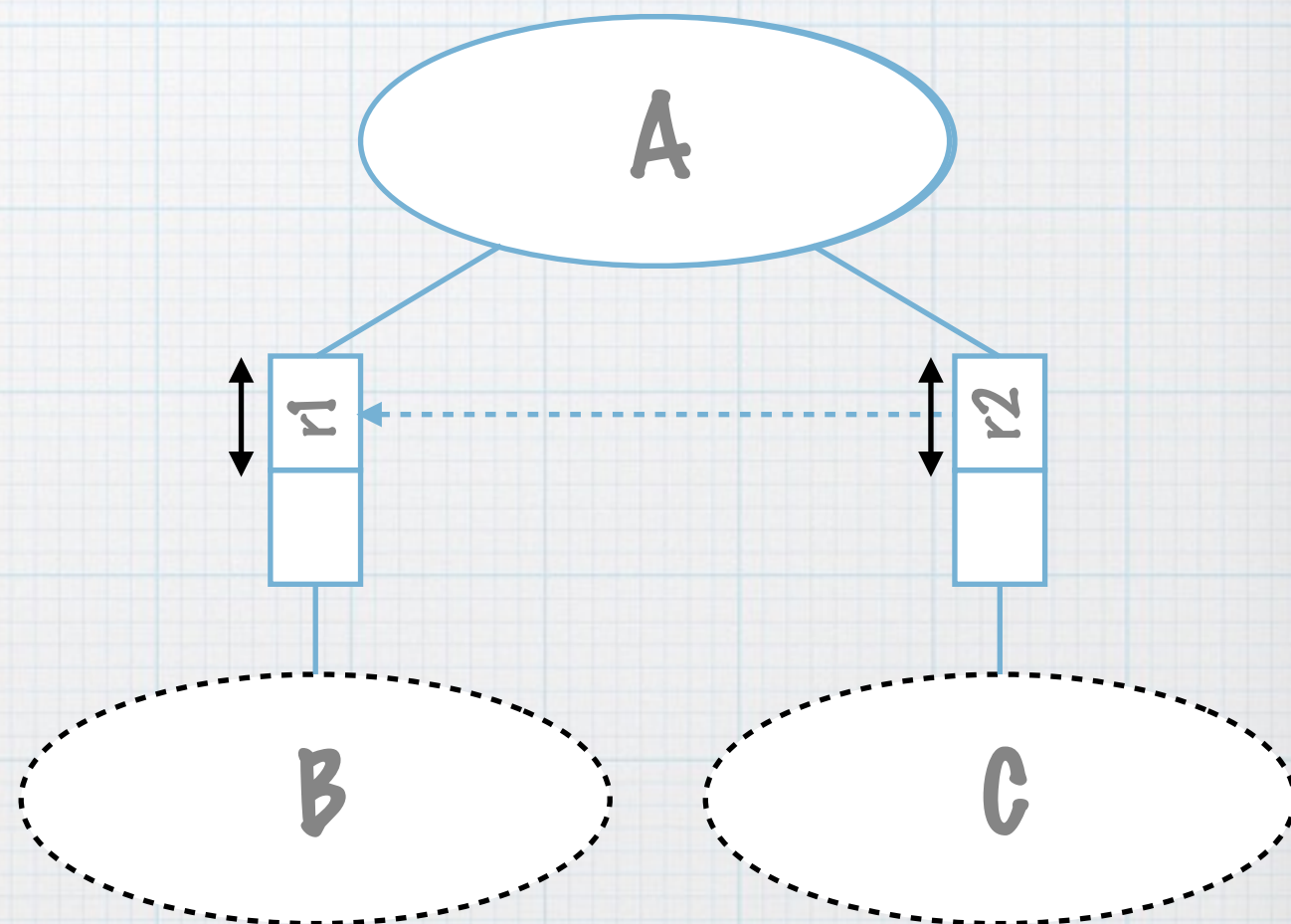
Contrainte de sous-ensemble

- Un étudiant peut avoir deux noms
- par ex. Maurits Cornelis Escher
- mais un étudiant ne peut pas avoir un deuxième nom s'il n'a pas déjà un premier nom

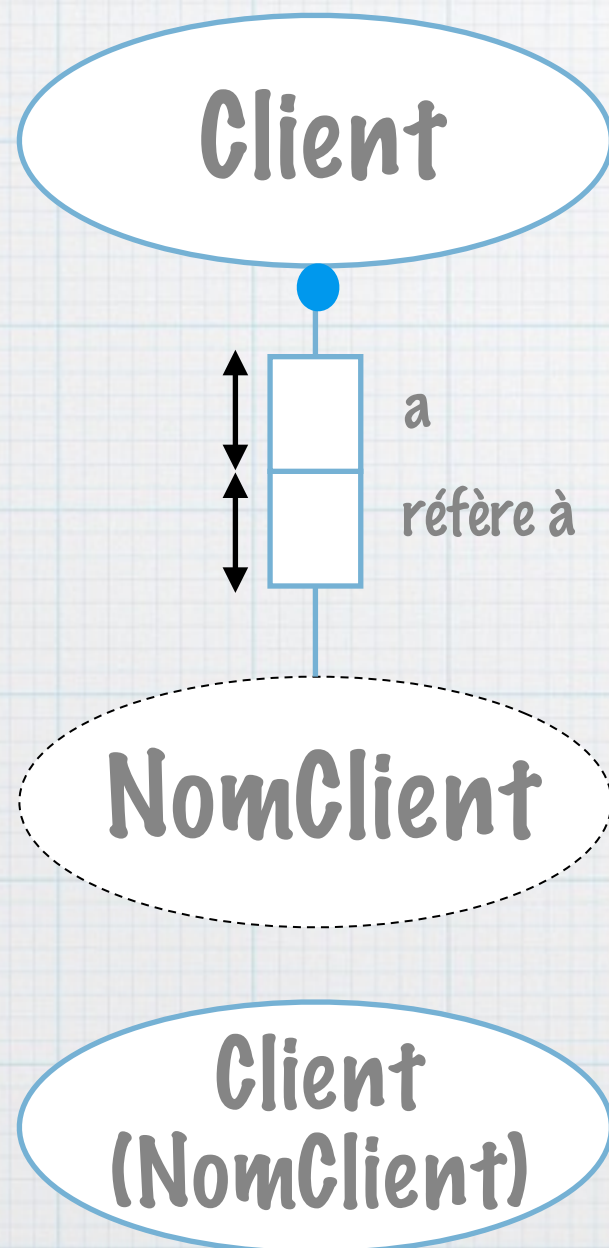


Contrainte de sous-ensemble

□ $\text{population}(r_2) \subseteq \text{population}(r_1)$



Types de référence (avec contraintes)



□ Contraintes :

□ Chaque client a exactement un nom:

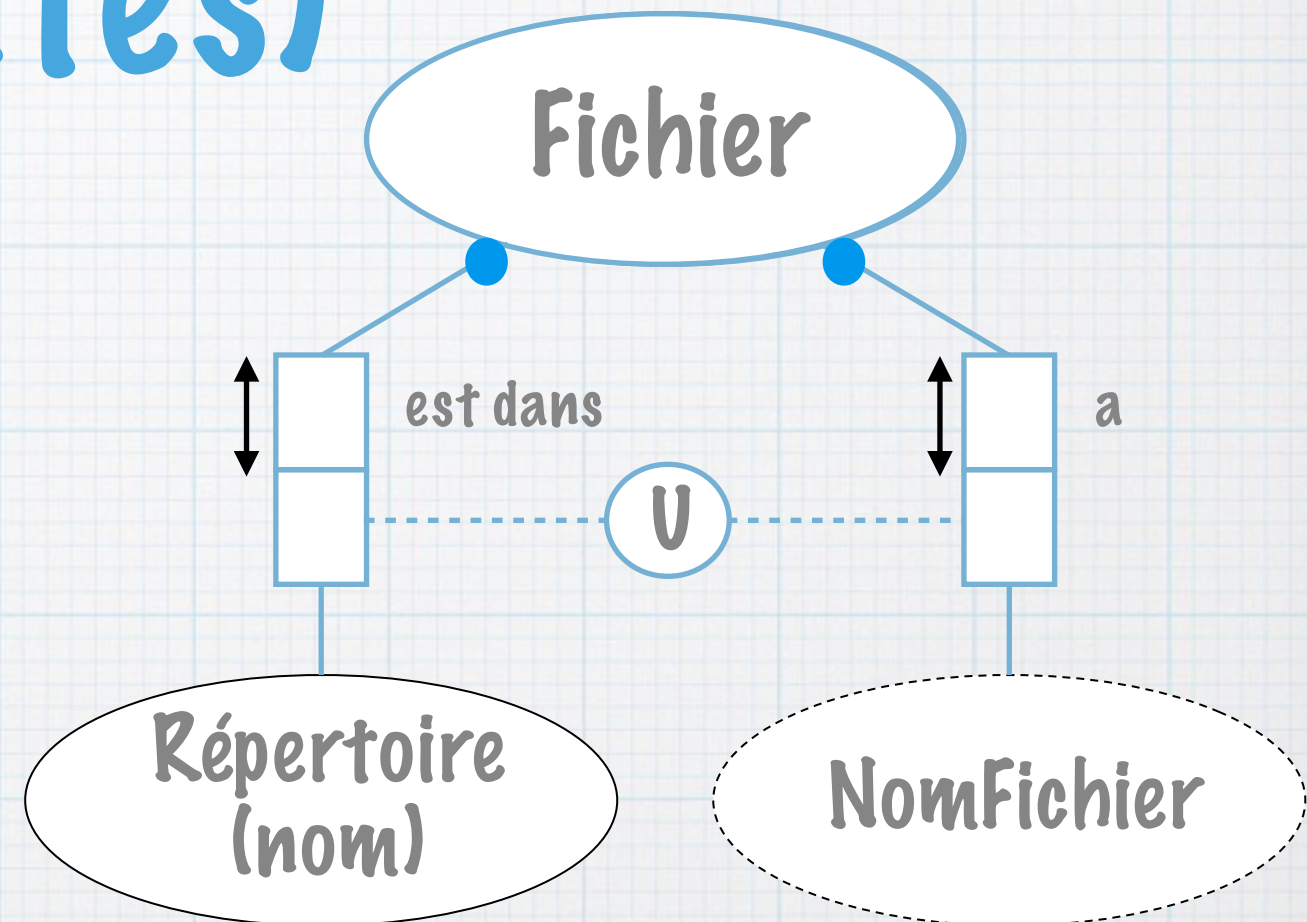
□ contrainte d'unicité interne ↔

□ rôle obligatoire ●—

□ Chaque nom de client correspond à maximum un Client:

□ contrainte d'unicité interne ↔

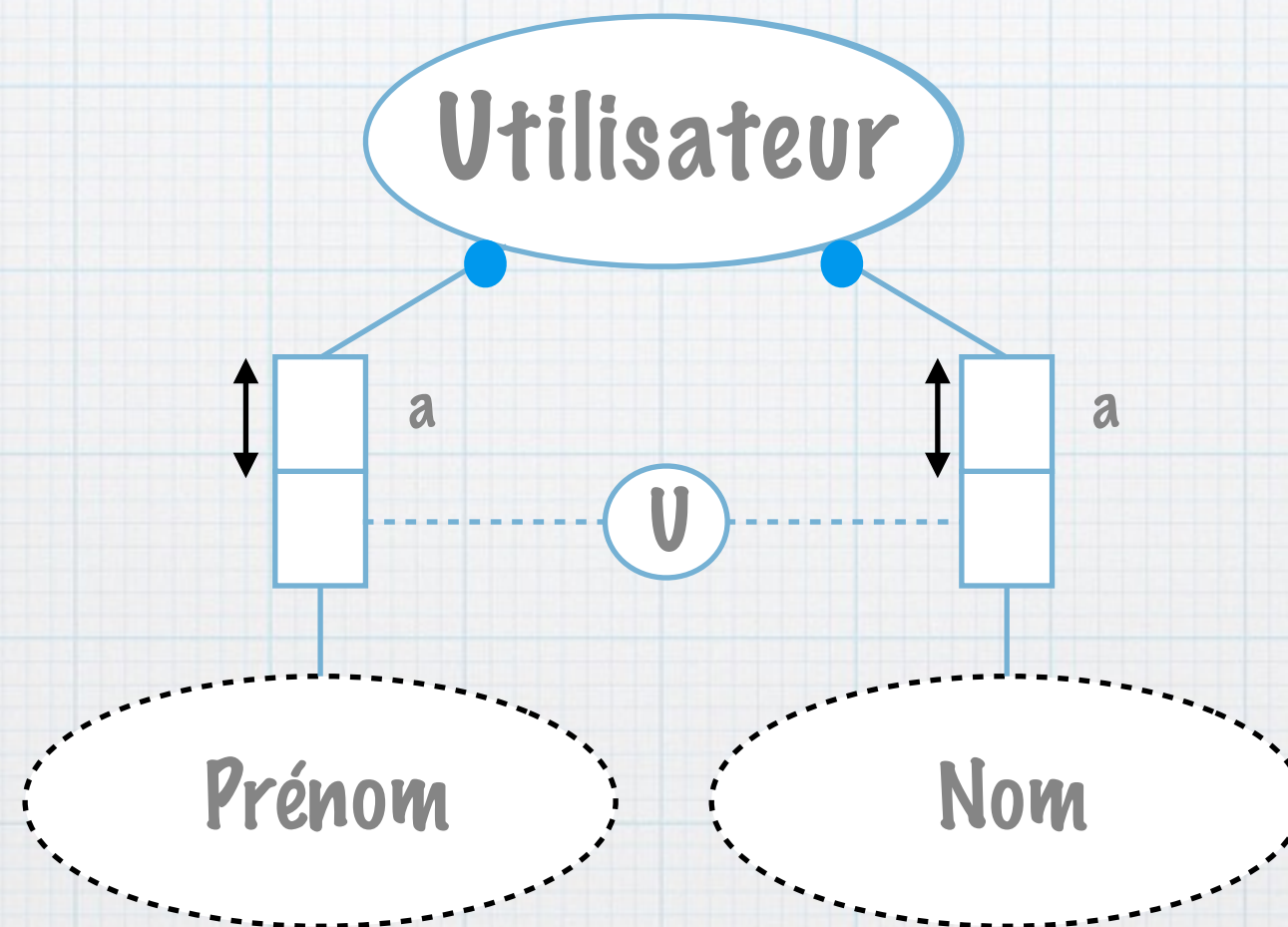
Schéma de référence composé (avec contraintes)



□ Contraintes :

- Chaque fichier est dans exactement un répertoire et a exactement un nom (contrainte d'unicité interne + rôle obligatoire)
- Chaque paire Répertoire, NomFichier correspond à maximum un Fichier (contrainte d'unicité externe)

Schéma de référence composé pour identifier un utilisateur



Un dernier conseil

- ❑ Dans une base de données on utilise souvent des IDs pour les données
- ❑ Dans un schéma conceptuel ORM on ne les met typiquement pas
 - ❑ sauf si ces IDs existent déjà dans l'univers de discours
 - ❑ p.e. le NOMA d'un étudiant, ou le n° ISBN d'un livre
- ❑ l'idée est que le modèle conceptuel reste proche de la réalité