

# Base de données relationnelles

**Walter RUDAMETKIN**

Bureau F011

Walter.Rudametkin@polytech-lille.fr

# Le modèle relationnel

- Un **schéma conceptuel** est très pratique pour la **phase d'analyse et conception**
- Mais un schéma conceptuel présente des limites pour une implémentation ou pour effectuer des requêtes.
- Codd (1970) a inventé le **modèle relationnel** basé sur des concepts simples :
  - ⌞ Facilement implémentable sur un ordinateur
  - ⌞ Facilité pour poser des requêtes
  - ⌞ passage simple : schéma conceptuel → schéma relationnel

# Le modèle relationnel

- Le **modèle relationnel** : ensemble de concepts pour décrire le contenu structuré d'une base de données relationnelles sous **forme tabulaire**.
- L'**algèbre relationnelle** : ensemble des opérateurs de manipulation des tables qui sont les fondements des langages d'interrogation / manipulation relationnels
- La **théorie relationnelle** = modèle + algèbre

## Bibliographie :

- A relational model of data for large shared databases. Comm ACM, 1970
- Further normalization on the database relational model in database systems. Prentice Hall, 1972

# La notion de Relation

Une **relation** = une **table** à 2 dimensions

Une **colonne** = un **attribut**

**En-tête** du tableau = **description** de la relation ou schéma de la relation

Une **ligne** = un **tuple** ou un n-uplet

Ensemble des lignes = contenu de la relation

# Exemple : la relation Fournisseur

	Numéro Fournisseur	Nom Fournisseur	Ville	
	1435	Helfer	Paris	
	678	Giard	Nantes	
	1234	Rolland	Marseille	

# Exemple : la relation Fournisseur

The diagram illustrates a database relation named 'Fournisseur'. It consists of a table with four columns: 'Numéro Fournisseur', 'Nom Fournisseur', and 'Ville'. The first column is highlighted in orange and labeled 'Schéma' (Schema). The second and third columns are also highlighted in orange. The fourth column is highlighted in green and labeled 'Un attribut' (An attribute). The table contains three rows of data: (1435, Helfer, Paris), (678, Giard, Nantes), and (1234, Rolland, Marseille). The third row is highlighted in purple and labeled 'Un tuple' (A tuple). The entire table is labeled 'données' (data).

Numéro Fournisseur	Nom Fournisseur	Ville
1435	Helfer	Paris
678	Giard	Nantes
1234	Rolland	Marseille

# La notion de Domaine

Définition 1 : **Domaine** = Ensemble de valeurs caractérisé par un nom (i.e., le **type**)

- Exemples : Entiers, réels, date, marque voiture, couleur
  - ↳ Marque Voiture : {Renault, Peugeot, Citroen}
  - ↳ Couleur : { blanche, grise }

Définition 2 : **Produit cartésien** d'un ensemble de domaines  $D_1, D_2, \dots, D_n$  noté  $D_1 \times D_2 \times \dots \times D_n$  = ensemble des éléments  $(v_1, v_2, \dots, v_n)$  avec  $v_i \in D_i$ ,  $\forall i \in [1, n]$

# Exemple de produit cartésien

## Jeux de cartes

Valeurs :

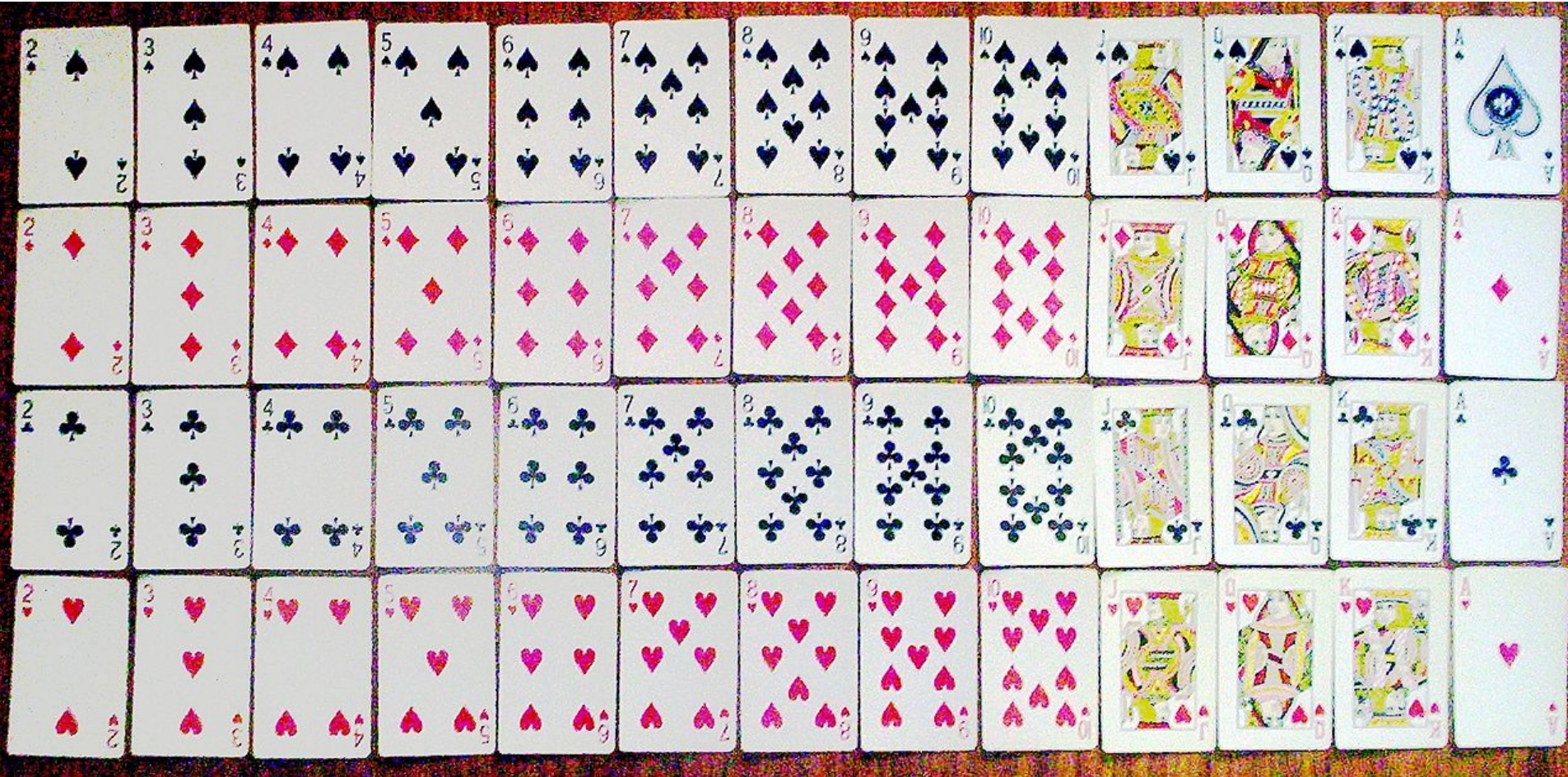
2, 3, 4, 5, 6, 7, 8, 9, 10, Valet, Dame, Roi

Enseignes :

pique ♠, cœur ♥, carreau ♦ et trèfle ♣.



# Exemple de produit cartésien



# La notion de Relation

**Définition 3** : Soient  $D_1, D_2, \dots, D_n$  une liste de domaines, une relation est un sous-ensemble du produit cartésien  $D_1 \times D_2 \times \dots \times D_n$ .

- Une relation est caractérisée par un **nom**
- On parle également **de table** pour une relation
- **Il n'y a pas deux lignes (tuples) égales** (théorie des ensembles)

# La notion d'attribut

Définition 4 : **Attribut** = **Colonne** d'une relation caractérisée par un nom

- Un attribut est **unique par relation** mais plusieurs attributs peuvent avoir le même domaine.
- Significatif pour comprendre le sens des valeurs d'une colonne. Le nom de la table et le nom des attributs permettent de comprendre quels sont les faits du monde réel représentés par la table.
- L'ordre des attributs n'a pas d'importance (il est donc important de nommer les colonnes).

# La notion de schéma

**Définition 5 :** **schéma** = nom de la relation suivi de la liste des attributs qui la composent et éventuellement de la définition de leurs domaines.

- $R(A1 : D1, \dots, An : Dn)$  où  $R$  est le nom de la relation,  $Ai$  les attributs et  $Di$  les domaines associés.
- Exemple :
  - ↳ EMPLOYE (NoEmpl : Entiers, Nom : Caractères, Année : Entiers, Nodept : Entiers)
- Remarque : Les domaines peuvent être omis
  - ↳ EMPLOYE (NoEmpl, Nom, Année, Nodept)

# La notion de clé

La **clé d'une relation** est un ensemble minimal d'attributs dont chaque valeur **détermine un tuple unique** de la relation.

- il ne doit pas exister plusieurs lignes d'une relation avec la même valeur de clé.

Exemple :

EMPLOYE (Nom, Prénom,  
Adresse, Ville)

Durand	Alain	3 rue Rose	Paris
Noël	Anne	19 rue Haute	Paris
Remy	André	46 rue Vilaine	Nantes
Durand	Etienne	10 rue Limite	Nice

Problème: Deux employés existent sous le même nom !

→ la clé 'nom' de la relation 'employé' ne permet pas d'identifier un client de manière unique.



# La notion de clé

Solution : gérer un numéro d'employé ce qui permettra une identification totale d'un employé

EMPLOYEE (NoEmp1, Nom, Prénom, Adresse, Ville)

101	Durand	Alain	3 rue Rose	Paris
102	Noël	Anne	19 rue Haute	Paris
120	Remy	André	46 rue Vilaine	Nantes
131	Durand	Etienne	10 rue Limite	Nice

Chaque valeur de 'noempl' n'est associée qu'à un seul employé.

# Déterminer les clés des relations

Client (numcli, nom, prenom, adresse)

Livre (numlivre, titre, auteur, nbexmpl)

Emprunt (numcli, numlivre, date, retard)

Peut-on identifier un emprunt avec le couple  
(numcli, numlivre) ?

# Déterminer les clés des relations

Client (numcli, nom, prenom, adresse)

Livre (numlivre, titre, auteur, nbexempl)

Emprunt (numcli, numlivre, date, retard)

Peut-on identifier un emprunt avec le couple (numcli, numlivre) ?

Non, si un client a la possibilité d'emprunter deux fois le même livre.



# Déterminer les clés des relations

## Solutions

- Utiliser en plus la date
  - ⌞ Clé d'emprunt: (numcli, numlivre, date)
- On suppose que la date ne suffit pas
  - ⌞ Possibilité d'emprunter deux fois le même livre le même jour
  - ⌞ Identifiant (numemprunt)

Emprunt (**numemprunt**, numcli, numlivre, date, retard)

# Déterminer une seule clef pour une relation

Si plusieurs clefs candidates possibles:

Etudiant (n°ss, nom, prenom, adresse, n°filiere, n°inscription)

– Deux clefs possibles

└ N°inscription: numero de carte d'etudiant

└ N°ss: numéro de sécurité sociale

**Choisir une des clés comme clé primaire** (ou clé principale)

Les autres clés sont alors des clés secondaires

Etudiant (n°ss, nom, prenom, adresse, n°filiere, n°inscription)

↖ *Clé secondaire*

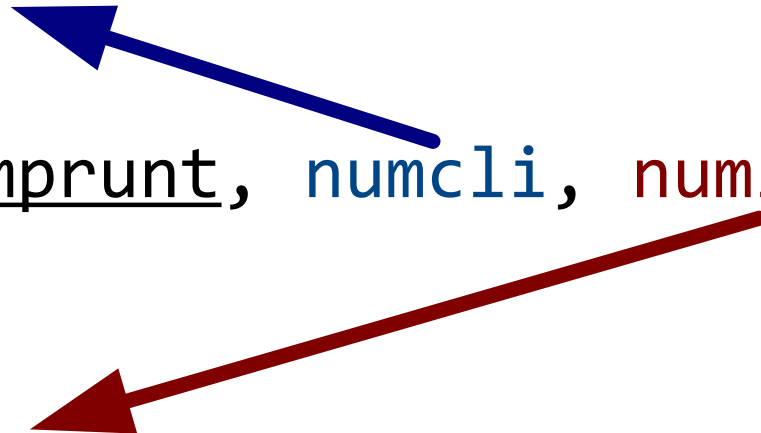
# Rôle des clés

Liens entre les relations

Client (numcli, nom, prenom, adresse)

Emprunt (numemprunt, numcli, numlivre, date, retard)


Livre (numlivre, titre, auteur, nbexmpl)



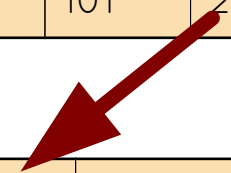
# Rôle des clés

101	Durand	Alain	3 rue Rose	Paris
102	Noël	Anne	19 rue Haute	Paris
120	Remy	André	46 rue Vilaine	Nantes
131	Durand	Etienne	10 rue Limite	Nice

Liens entre les  
tableaux de valeurs



54	120	12	30/11/2014	N
51	102	12	18/04/2013	N
52	101	12	12/10/2012	N
53	101	20	26/04/2014	0



20	Le prince de Sang mêlé	JK Rowling	10
30	Can you keep a secret ?	Sophie Kinsella	4
12	Websphere V3.5 Handbook	Websphere Consulting Team	1

# Contrainte référentielle

Définition 6 : Une **référence** (ou **clé étrangère**) est un attribut (ou un groupe d'attributs) dont les valeurs sont incluses dans l'ensemble des valeurs prises par la clé d'une autre relation.

Exemple:

- Numcli dans Client est appelé clé primaire
- Numcli dans Emprunt est appelé **clé étrangère**
- Toute valeur de l'attribut Numcli dans la relation Emprunt doit se trouver dans la relation Client

# Contrainte référentielle

Définition 6 : Une **référence** (ou **clé étrangère**) est un attribut (ou un groupe d'attributs) dont les valeurs sont incluses dans l'ensemble des valeurs prises par la clé d'une autre relation.

Exemple:

- Numcli dans Client est appelé clé primaire
- Numcli dans Emprunt est appelé **clé étrangère**
- Toute valeur de l'attribut Numcli dans la relation Emprunt doit se trouver dans la relation Client

# Contrainte structurelle

La valeur nulle (**NULL**) est une valeur conventionnellement introduite dans une relation pour représenter une information inconnue ou inapplicable.

Exemple:

Employé (NoEmpl, Nom, NomMarital, Année, Adresse, Téléphone, Nodept)

- Téléphone non connu / (10, Durand, Faure, 1980, Paris, NULL, 75)
- Nom marital pour un homme / (23, Fergio, NULL, 1987, Marseille, 0234565412, 13)

**Contrainte de relation** : Toute relation doit posséder au moins une clé non nulle

**Schéma d'une base de données** : schéma des relations et contraintes d'intégrité structurelles

# **Passage du schéma conceptuel au schéma relationnel**

**Walter RUDAMETKIN**

Bureau F011

Walter.Rudametkin@polytech-lille.fr



# Introduction

## Modèle conceptuel :

- repose sur deux concepts : les classes et les associations → très descriptif, bon support de communication.
- Toutefois, les SGBD usuels sont relationnels

## Passage du modèle conceptuel vers le relationnel

- règles de conversion des entités en relations **3FN**,
- règles de conversion des associations... selon leurs types

# Traduction des Classes

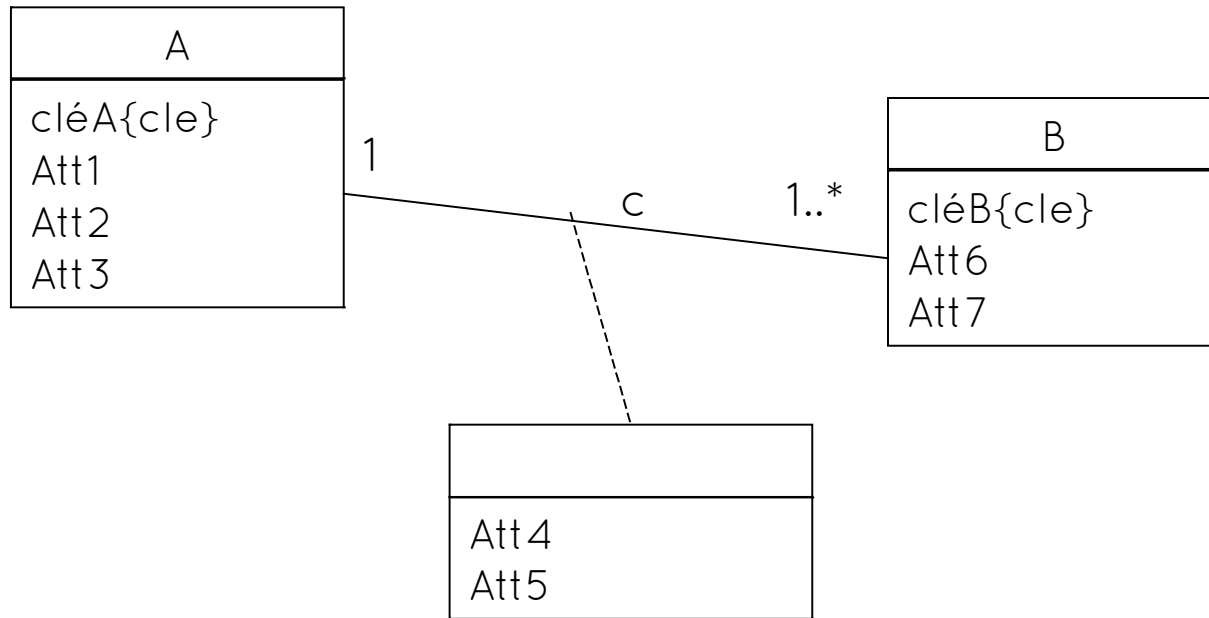
Cas courant

Toute classe devient une relation 3FN dont la clé est l'identifiant de la classe.

A
cléA {clé}
Att1
Att2
Att3

A (cléA, Att1, Att2, Att3)

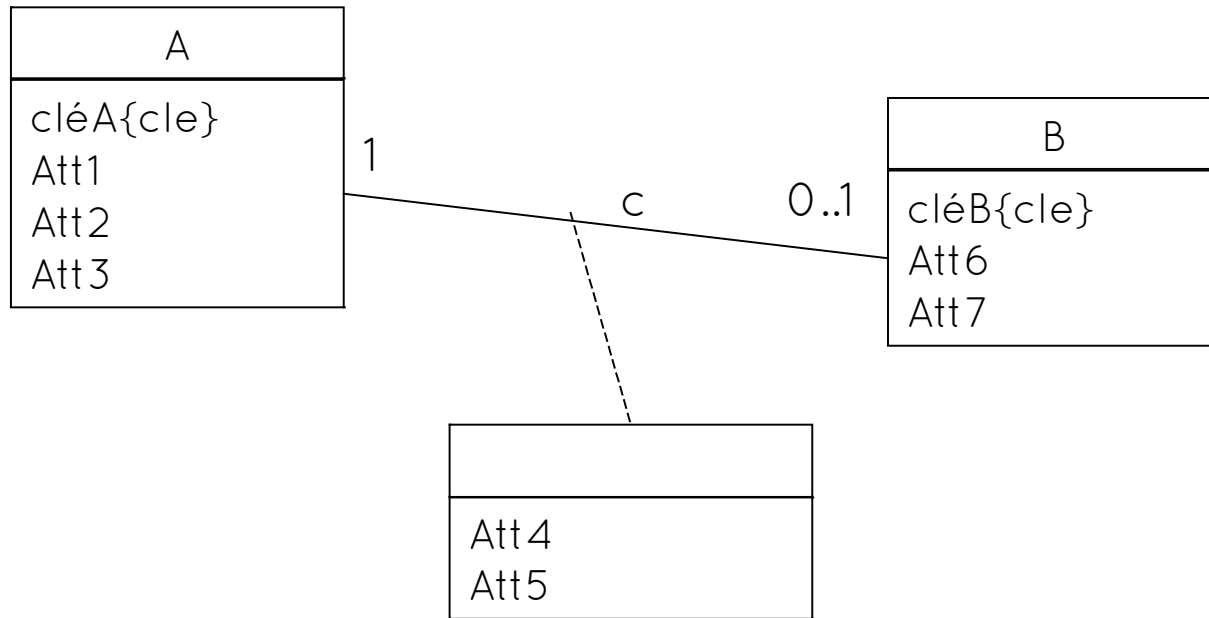
# Traduction des associations / Cardinalités 1,\* - 1



A (CléA, Att1, Att2, Att3)

B (CléB, Att6, Att7, Att4, Att5, **CléA#**)

# Traduction des associations / Cardinalités 1,1 - 0,1

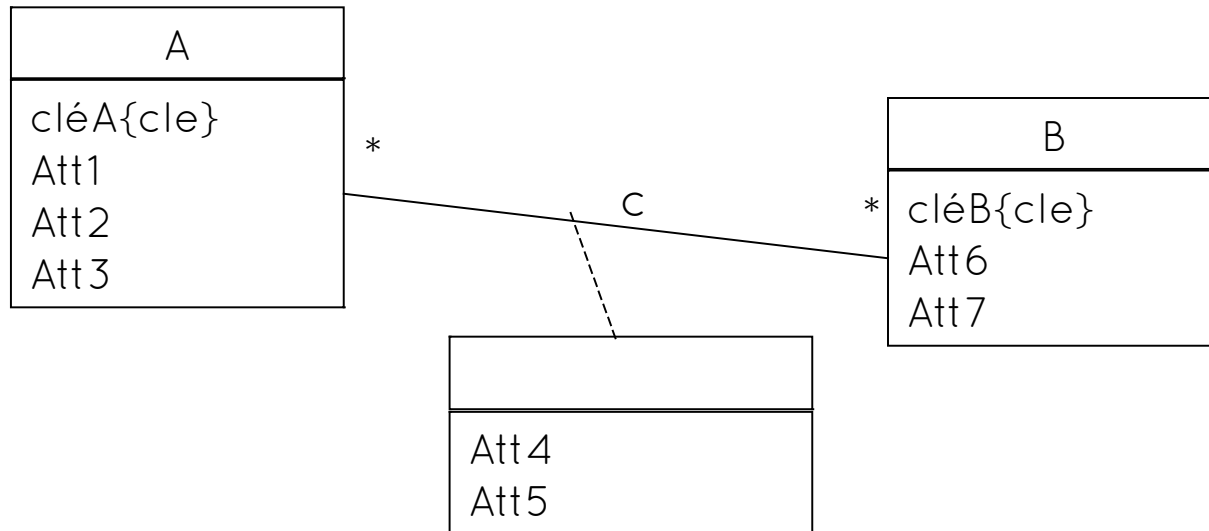


A (CléA, Att1, Att2, Att3)

B (CléB, Att6, Att7, Att4, Att5, CléA#)

# Traduction des associations /

## Cardinalités maximales égales à \* (1,\* ou 0,\*)



A (CléA, Att1, Att2, Att3)

B (CléB, Att6, Att7)

C (CléA#, CléB#, Att4, Att5)

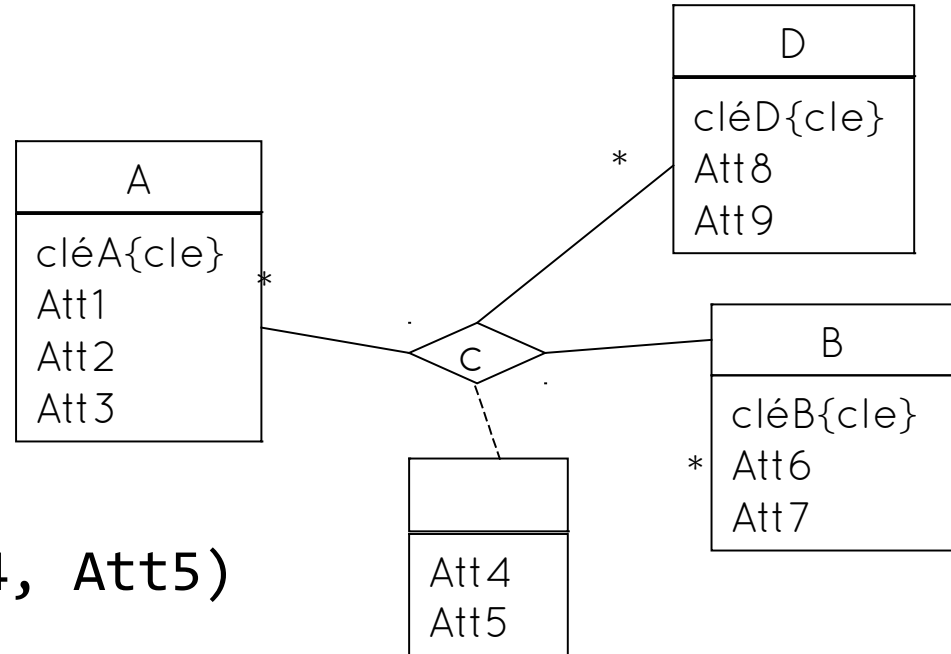
# Traduction des associations / Association ternaire dont les cardinalités sont de type \*

A (CléA, Att1, Att2, Att3)

B (CléB, Att6, Att7)

D (CléD, Att8, Att9)

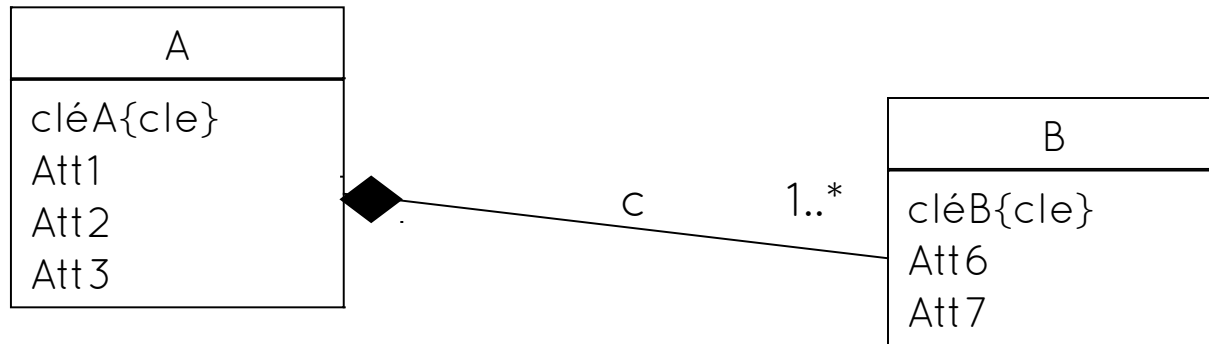
C (CléA#, CléB#, CléD#, Att4, Att5)



SAUF SI une partie de la clé détermine une autre partie de la clé

- Si CléA, CléB → CléD alors C (CléA, CléB, CléD, Att4, Att5)
- Si CléA, CléD → CléB alors C (CléA, CléD, CléB, Att4, Att5)
- Si CléD, CléB → CléA alors C (CléD, CléB, CléA, Att4, Att5)

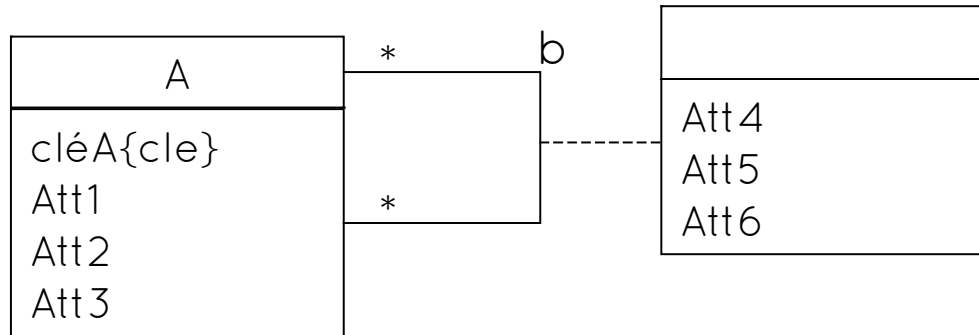
# Traduction des Entités faibles



A (CléA, Att1, Att2, Att3)

B (CléA#, CléB, Att6, Att7)

# Traduction des associations réflexives



A (CléA, Att1, Att2, Att3)

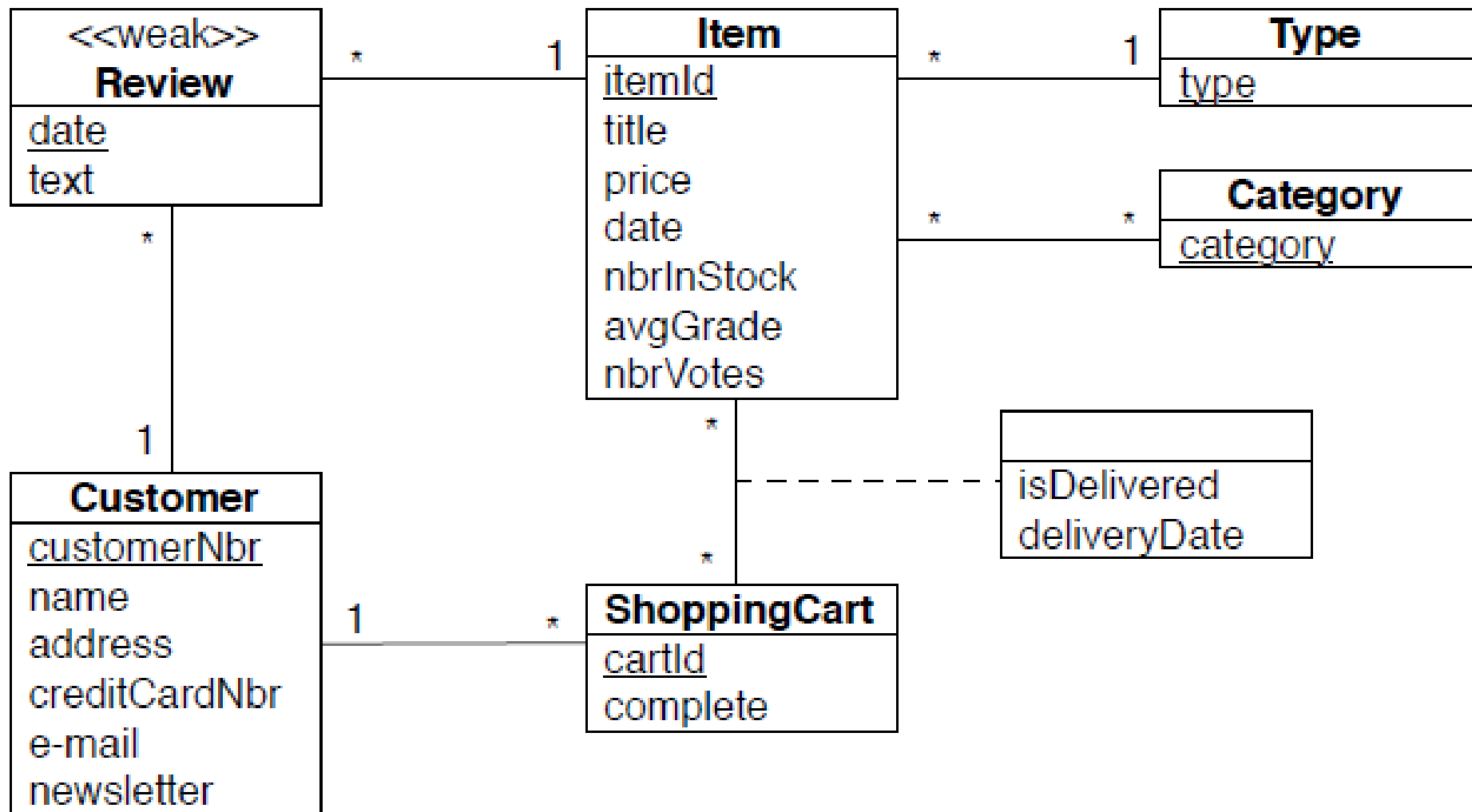
B (CléA#, CléA2#, Att4, Att5, Att6)

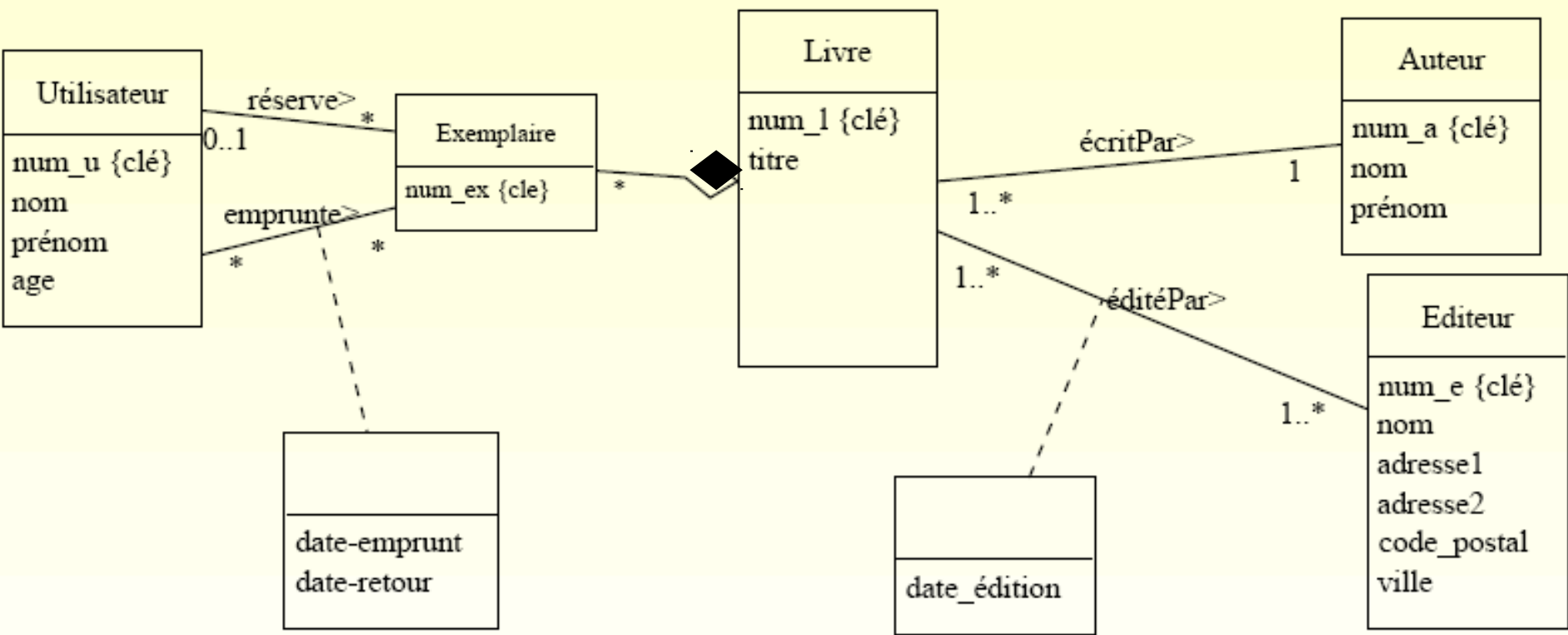


# Exercices

Convertissez le schéma conceptuel en schéma relationnel.

Créez le schéma conceptuel d'abord si nécessaire.





A **calendar program** that allows users to browse each other's calendars and to book common appointments shall be developed. The program has a database which keeps track of the **users** and their **calendars**. You use the calendar to store data concerning **appointments**. An appointment **starts and ends** at a given time on a given day and is described by a **text**. You may specify that you wish to be reminded of an appointment. **Reminders** are of different **kinds**: a signal in the computer's loudspeaker, a pop-up window with the description of the meeting, or an e-mail containing the description. You may, for each reminder, specify **how long before** the appointment that you wish to be reminded. Develop a UML model of the database.