

## 3. Bases de Données Relationnelles

### Le modèle relationnel

- ▶ Un schéma conceptuel **CM1** 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
  - ▶ Facilité de passage : schéma conceptuel → schéma relationnel

68

### 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**

69

### 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

70

## La notion de relation: exemple

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

Diagramme illustrant un schéma de relation. Le schéma est représenté par le tableau ci-dessus. Les attributs sont les colonnes (Numéro Fournisseur, Nom Fournisseur, Ville). Les données sont les valeurs dans les cellules. Un tuple est une ligne du tableau (par exemple, la ligne correspondant à Rolland et Marseille).

71

## La notion de domaine

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

► Exemples : Entiers, réels, date, marque voiture, couleur

► Marque Voiture : {Renault, Peugeot, Citroen}

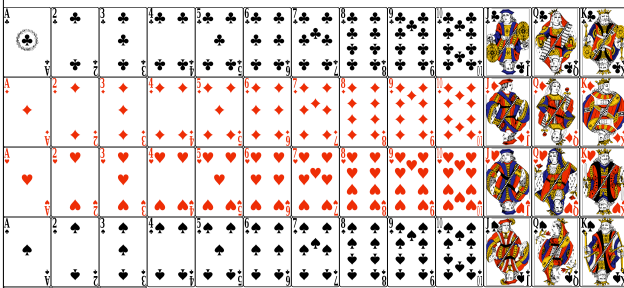
► Couleur : { blanche, grise}

► **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  $v_i$  appartient à  $D_i$ ,  $\forall i$  dans  $[1, n]$

► Exemple: jeu de cartes,  $D_1 = \{2, 3, 4, 5, 6, 7, 8, 9, 10, \text{Valet}, \text{Dame}, \text{Roi}, \text{As}\}$ ,  $D_2 = \{\heartsuit, \spadesuit, \clubsuit, \diamondsuit\}$

72

## La notion de domaine



73

## La notion de relation

► 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)

74

## La notion d'attribut

- ▶ **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).

75

## La notion de schéma

- ▶ Schéma = nom de la relation suivi de la liste des attributs qui la composent et éventuellement de la définition de leurs domaines.
  - ▶  $R(A_1 : D_1, \dots, A_n : D_n)$  où R est le nom de la relation,  $A_i$  les attributs et  $D_i$  les domaines associés.
  - ▶ Exemple :
    - ▶ EMPLOYE (NoEmpl : Entiers, Nom : Caractères, Année : Entiers, Nodept : Entiers)
  - ▶ Remarque : Les domaines peuvent être omis
    - ▶ EMPLOYE (NoEmpl, Nom, Année, Nodept)

76

## 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)

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

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

77

## La notion de clé

- ▶ Solution : gérer un numéro d'employé ce qui permettra une identification totale d'un employé
- ▶ EMPLOYE (NoEmpl, Nom, Prénom, Adresse, Ville)

NoEmpl	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
108	Durand	Etienne	10 rue Limite	Nice

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

78

## 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) ?
- ▶ Non, si un client a la possibilité d'emprunter deux fois le même livre.

79

---

---

---

---

---

---

---

---

## Déterminer les clés des relations

- ▶ Solutions
  - ▶ Utiliser en plus la date
    - ▶ Clé d'emprunt: (numcli, numlivre, date)
  - ▶ Si 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)

80

---

---

---

---

---

---

---

---

## Déterminer 1 seule clé / relation

- ▶ Si plusieurs clés candidates possibles:
  - ▶ Etudiant (n°ss, nom, prenom, adresse, n°filie, n°inscription)
- ▶ Deux clés possibles
  - ▶ n°inscription: numéro de carte d'étudiant
  - ▶ 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°filie, n°inscription)

81

---

---

---

---

---

---

---

---

## Rôle des clés

- ▶ Liens entre les relations
  - ▶ Client (numcli, nom, prenom, adresse)
  - ▶ Emprunt (numemprunt, numcli, numlivre, date, retard)
  - ▶ Livre (numlivre, titre, auteur, nbexmpl)

82

---

---

---

---

---

---

---

---

## Rôle des clés

num_cli	nom	prenom	adresse	ville	num_livre	titre	auteur	nbexempl
101	Durand	Alain	3 rue Rose	Paris	28	Harry Potter...	JK Rowling	10
102	Noël	Anne	19 rue Haute	Paris	30	Can you keep...	S Kinsella	4
128	Remy	André	46 rue Vilaine	Nantes	12	Websphere Handbook	Websphere Team	1
108	Durand	Etienne	10 rue Limite	Nice				

num_emprunt	num_client	num_livre	date	retard
54	128	12	30/11/2022	N
51	102	12	18/04/2022	N
52	101	12	12/10/2022	N
53	101	28	26/04/2022	O

83

## Contrainte référentielle

- ▶ 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:
  - ▶ **num\_cli** dans **Client** est appelé **clé primaire**, **num\_client** dans **Emprunt** est appelé **clé étrangère**
  - ▶ Toute valeur de l'attribut **num\_client** dans la relation **Emprunt** doit se trouver dans la relation **Client**

84

## 4. Du schéma conceptuel au schéma relationnel

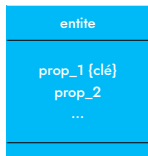
### Introduction

- ▶ **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** (*forme normale*),
  - ▶ règles de conversion des associations... selon leurs types

86

## Traduction des Classes

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



`entite(prop_1, prop_2, ...)`

87

---

---

---

---

---

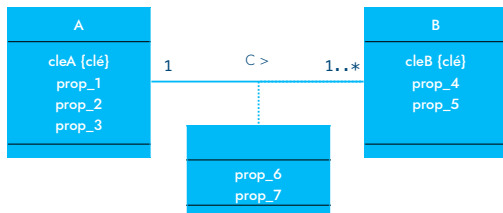
---

---

---

## Traduction des associations

- Cardinalités 1..\* — 1



`A(cleA, prop_1, prop_2, prop_3)`  
`B(cleB, prop_4, prop_5, prop_6, prop_7, cleA#)`

88

---

---

---

---

---

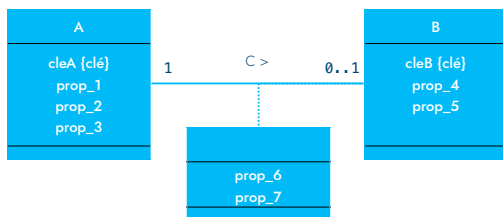
---

---

---

## Traduction des associations

- Cardinalités 0..1 — 1



`A(cleA, prop_1, prop_2, prop_3)`  
`B(cleB, prop_4, prop_5, prop_6, prop_7, cleA#)`

89

---

---

---

---

---

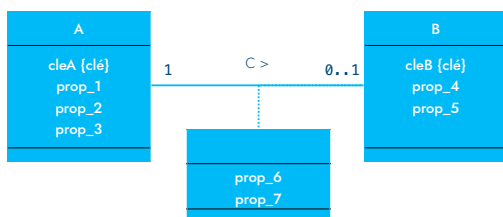
---

---

---

## Traduction des associations

- Cardinalités \* — \*



`A(cleA, prop_1, prop_2, prop_3)`  
`B(cleB, prop_4, prop_5)`  
`C(cleA#, cleB#, prop_6, prop_7)`

90

---

---

---

---

---

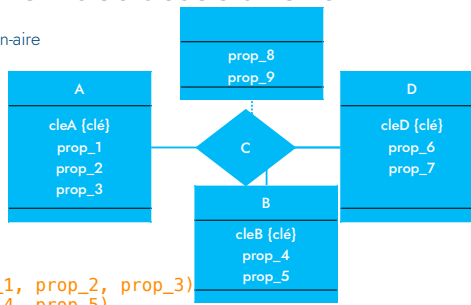
---

---

---

## Traduction des associations

### ▸ Association n-aire



A(cleA, prop\_1, prop\_2, prop\_3)  
 B(cleB, prop\_4, prop\_5)  
 D(cleD, prop\_6, prop\_7)  
 C(cleA#, cleB#, cleD, prop\_8, prop\_9)

91

---

---

---

---

---

---

---

---

---

---

## Traduction des associations

### ▸ Association n-aire

- Si une partie de la clé détermine une autre partie de la clé on peut
- C(cleA#, cleB#, prop\_8, prop\_9, cleD#) si (cleA, cleB) permettent d'identifier cleD
- Idem pour les autres permutations

92

---

---

---

---

---

---

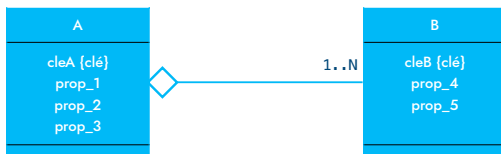
---

---

---

---

## Cas des associations faibles



A(cleA, prop\_1, prop\_2, prop\_3)  
 B(cleA#, cleB, prop\_4, prop\_5)

93

---

---

---

---

---

---

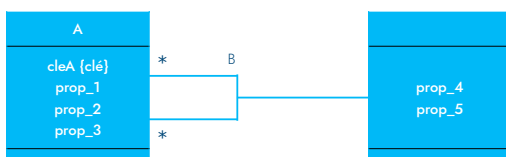
---

---

---

---

## Cas des associations reflexives



A(cleA, prop\_1, prop\_2, prop\_3)  
 B(cleA\_1#, cleA\_2#, prop\_4, prop\_5)

94

---

---

---

---

---

---

---

---

---

---

## Exercices

- ▶ Convertissez le schéma conceptuel en schéma relationnel.
- ▶ Créez le schéma conceptuel d'abord si nécessaire.

95

---

---

---

---

---

---

---

---

## Exercices

- ▶ Convertissez le schéma conceptuel en schéma relationnel.
- ▶ Créez le schéma conceptuel d'abord si nécessaire.

96

---

---

---

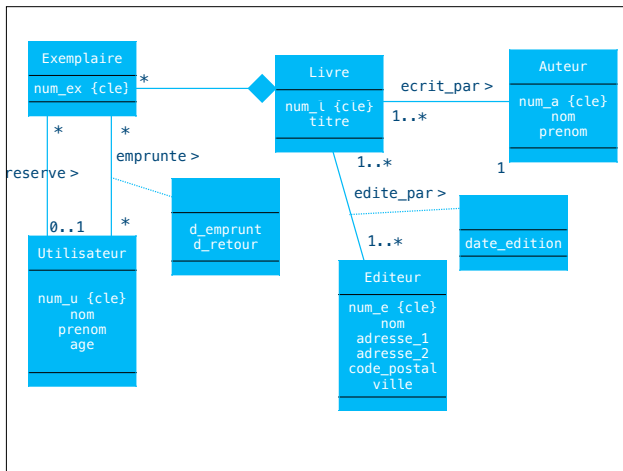
---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

---

---

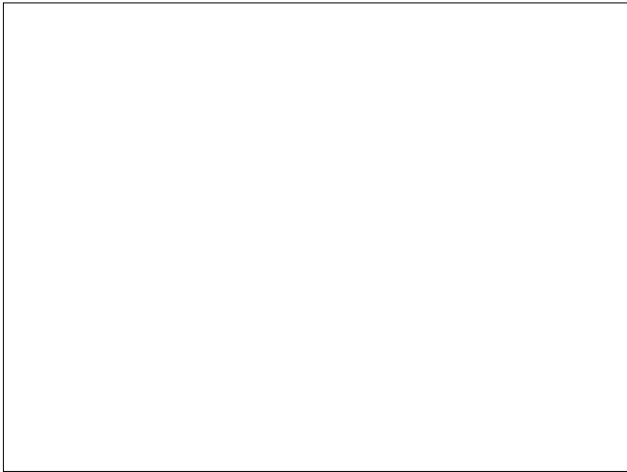
---

---

---

---





---

---

---

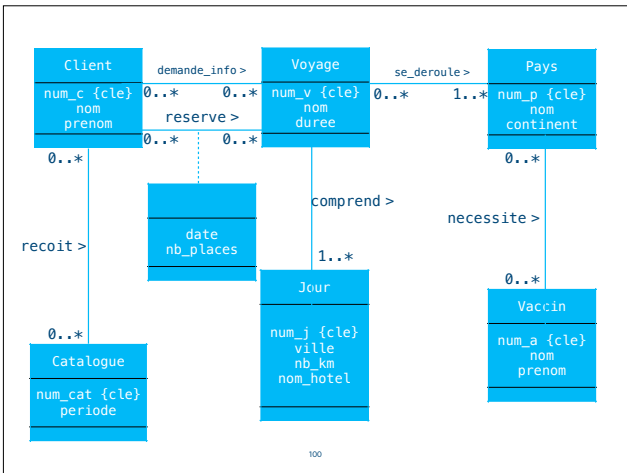
---

---

---

---

---



---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

► 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.

---

---

---

---

---

---

---

---