

# CHAPITRE 2

# LE MODELE RELATIONNEL



# Définitions

## Concepts du langage relationnel

2

**Les objets**

**Les associations**

**La relation**

```
graph TD; A[Les objets] --> C[La relation]; B[Les associations] --> C;
```

# Notion de relation

3

## Légende:

**Relation:** un tableau à deux dimensions, appelé **table**.

<u>NoEtud</u>	Nom	Prénom	Age
125	Amara	Leila	19
203	Ghichi	Faiza	20
65	Bey	Sami	20
365	Amouri	Brahim	21

- Etudiant : le nom de la relation.
- Les entêtes des colonnes NoEtud, Nom, Prénom et Age : les **attributs** de la relation.
- Chaque ligne de la table : une occurrence.
- Par exemple, <203, ghichi, faiza, 20> est un **tuple** qui décrit une occurrence.
- **Identifiant** ou **clé** de la relation : attribut souligné

# Exemple (suite)

4

**Cours**

<u>NomC</u>	Horaire	Prof
Algo	Samedi08-10	Bentounsi
systeme	Mardi 10-12	bouzenada

**Suit**

<u>NoEtud</u>	<u>NomC</u>
365	Algo
65	systeme
125	Algo
<b>125</b>	<b>systeme</b>

Remarque : l'identifiant de la relation « **suit** » (qui traduit un type d'association) est composé des identifiants des deux relations « **Etudiant** » et « **Cours** »

# Notion de domaine

**Définition:** Un **domaine** est un ensemble de valeurs que peut prendre un attribut. Un domaine peut définir un ou plusieurs attributs.

**Exemple :**

- **Domaine(Noetud) : INF + entier[0..300]**
- **Domaine(nom): chaîne de caractère de longueur maximale 20**
- **Domaine(âge) : entier dans [16..60]**

# Définition d'une Relation

6

- Une relation est définie par :
  - Son nom
  - La liste de ses attributs qui est une liste de couples (attribut, domaine)
  - Son identifiant
  - Sa définition qui est représentée par un texte expliquant l'utilité de la relation.

# Schéma de relation

7

## Définition:

Un schéma de relation est composé de son nom, la liste de ses attributs et son identifiant.

Un schéma de relation a la forme suivante :  
**Nom-relation (A1 :D1, A2 :D2, ....., An :Dn)**

## Exemple :

- Etudiant (Noetud : Domaine(Noetud), nom : Domaine(nom), prénom : domaine(nom), age : domaine(age))

# Population de relation

## Définition:

La **population** d'une relation est constituée par l'ensemble de ses tuples. Dans cet ensemble, il n'y a ni double ni ordre.

## Remarque :

on appelle souvent le **schéma** d'une relation son **intention** et sa **population** son **extension**.



# Schéma d'une base de donnée

## Définition:

On appelle **schéma d'une base de données** l'ensemble des schémas de ses relations. De même, on appelle **population d'une base de données**, l'ensemble des populations de ses relations.

# Règles de modélisation des attributs

10

Dans le modèle relationnel, les attributs sont tous simples et monovalués.

Toute valeur prise par un attribut pour un tuple donné est atomique et unique.

Les notions d'attributs complexe ou multivalué n'existe pas dans le modèle relationnel.

# (a) Attribut complexe :

11

- Ajouter l'attribut «adresse » dans la relation Etudiant avec :  
Adresse = numéro + rue +ville + code-postal + pays »
- Adresse : attribut complexe et ne peut être géré par le modèle relationnel.
- **Solution 1** : considérer «adresse » comme une chaîne de caractères.  
L'utilisateur ne peut pas poser des questions sur la ville, le pays ou autre.....
- **Solution 2** : on ajoute les attributs suivants à la relation Etudiant :
  - numéro
  - Rue
  - Ville
  - Code-postal
  - PaysL'utilisateur connaît le détail de l'adresse mais ne connaît pas la notion globale d'adresse.
- La solution est choisie en fonction de l'utilisation de l'attribut

## (b) Attribut multi-valué

12

- Si on veut mémoriser les différents prénoms des étudiants, deux solutions sont possibles :
- **Solution 1** : avoir dans la relation «Etudiant », plusieurs attributs Prénom1, Prénom 2,.....Le problème qui se pose est celui du nombre d'attributs «prénom » à mettre dans la relation
- **Solution 2** : on ne garde que les attributs Noetud, nom, age dans la relation Etudiant et on crée une autre relation supplémentaire « Etudiant-prénoms»
- Etudiant-prénoms :

Noetud	prénom
INF2000	a Y M E N
INF2000	MOHAMED
INF2002	B R A H I M

# Exemple de schéma relationnel :

## DOMAINES :

D-nom : chaîne de caractères de longueur maximale 20

D-Noetud : INF + entier dans [0..300]

D-age : entier dans [16..60]

## RELATION Etudiant :

### ATTRIBUTS :

Noetud : D-Noetud Sans NULL

Nom : D-nom Sans NULL

Age : D-age

**IDENTIFIANT** : (Noetud)

**DEFINITION** : toute personne actuellement inscrite à l'institut.

## RELATION Etudiant-prénoms

### ATTRIBUTS :

Noetud : D-Noetud Sans NULL

Prénom : D-nom Sans NULL

**IDENTIFIANT** : (Noetud + Prénom)

**DEFINITION** : prénom des étudiants

# Dépendances fonctionnelles et clés de relations

14

Les **dépendances fonctionnelles**, notées DFs, est une notion importante utilisée dans la normalisation des relations. La **normalisation** est le processus qui permet de construire des relations **complètes et non redondantes**.

# Dépendance fonctionnelle

15

- **Définition**
- Soit  $R(X, Y, Z)$  un schéma de relation où  $X \cup Y \cup Z$  sont des attributs de  $R$ .
- On note :  $X \xrightarrow{R} Y$  et on dit  $X$  *détermine*  $Y$  dans  $R$  ou  $Y$  *dépend fonctionnellement* de  $X$  dans  $R$ 
  - Si et seulement si : étant donné *une valeur de  $X$* , il lui correspond *une et une seule valeur de  $Y$*  ( $X$  et  $Y$  sont des attributs ou des ensembles d'attributs de  $R$ ).

# Dépendance fonctionnelle (suite)

16

## ■ Dépendance Fonctionnelle (suite)

Soient  $X$  et  $Y$  deux attributs de la relation  $R$ , on dit que l'on a une **DF**  $X \rightarrow Y$  si et seulement si:  
si  $E1$  et  $E2$ , 2  $n$ -uplets de  $R$ , ont les mêmes valeurs pour l'attribut  $X$  alors  $E1$  et  $E2$  ont nécessairement les mêmes valeurs pour l'attribut  $Y$ .

## ■ Utilité

- Contraintes sur les données que contient une relation
- Outil de contrôle de l'intégrité des données (au niveau du schéma)
- Outil mathématique permettant d'expliquer le processus de normalisation
- Indispensable pour re-modeler un schéma de BD qui pose problème



# Exemple

17

- Soit la relation **Gestion-Stock** (code-prod, code-dep, libellé-prod, Pu, Adr-dep, Qte-stock).
- Les dépendances suivantes sont vraies :
  - Code-prod  $\longrightarrow$  libellé-prod
  - Code-dep  $\longrightarrow$  Adr-dep
  - Code-prod + code-dep  $\longrightarrow$  Qte-stock
- Les dépendances suivantes sont fausses :
  - Code-prod  $\longrightarrow$  Qte-stock
  - Code-prod  $\longrightarrow$  Adr-dep

# Propriétés des dépendances fonctionnelles

18

- Des **axiomes et des règles** permettent de **déduire** de nouvelles dépendances fonctionnelles à partir d'un ensemble de dépendances fonctionnelles initial.
- Dans ce qui suit, on désigne « **A** » l'ensemble des attributs de la relation **R**.

# Axiomes d'Armstrong

19

□ *Réflexivité* :

$$\underline{Y} \subseteq X \wedge X \subset A \Rightarrow X \xrightarrow{R} Y$$

□ *Augmentation* :

$$X \xrightarrow{R} Y \Rightarrow XZ \xrightarrow{\quad} YZ$$

□ *Transitivité* :

$$X \xrightarrow{R} Y \wedge Y \xrightarrow{R} Z \Rightarrow X \xrightarrow{R} Z$$

# Règles d'inférence ou dérivées :

20

□ *Union* :

$$X \xrightarrow{R} Y \wedge X \xrightarrow{R} Z \Rightarrow X \xrightarrow{R} Y \cup Z$$

□ *Pseudo-transitivité* :

$$X \xrightarrow{R} Y \wedge Y W \xrightarrow{R} Z \Rightarrow X W \xrightarrow{R} Z$$

□ *Décomposition* :

$$X \xrightarrow{R} Y \wedge Z \subseteq Y \Rightarrow X \xrightarrow{R} Z$$

# Types de dépendances fonctionnelles

21

## □ a) DF triviale

Une DF  $X \xrightarrow{R} Y$  est dite triviale si  $Y \subseteq X$

## □ b) DF canonique

□ Une DF  $X \xrightarrow{R} Y$  est dite canonique si sa partie droite ne comporte qu'un seul attribut (simple) et un ensemble de DFs est dit canonique si chacune de ses DFs est canonique.

## □ Exemple :

La DF code-prod  $\xrightarrow{\quad}$  libellé, Pu n'est pas canonique.

□ On applique la règle de décomposition:

Code-prod  $\xrightarrow{\quad}$  libellé  
Code-prod  $\xrightarrow{\quad}$  Pu

# Types de dépendances fonctionnelles

## (suite)

22

### □ **DF élémentaire**

□ Une DF canonique  $X \xrightarrow{R} Y$  est élémentaire si pour tout  $X' \subset X$ , la DF  $X' \xrightarrow{R} Y$  n'est pas vraie. En d'autres termes, Y ne dépend pas fonctionnellement d'une partie de X

□ La DF Code-prod, code-dep  $\xrightarrow{\quad}$  libellé n'est pas élémentaire car Code-prod  $\xrightarrow{\quad}$  libellé

### □ **d) DF directe**

□ Une DF  $X \xrightarrow{R} Y$  est dite directe si :

□ Elle est élémentaire

□ Y ne dépend pas transitivement de X

On dit qu'une DF  $A \rightarrow B$  est **directe** s'il n'existe aucun attribut  $C$  tel que l'on puisse avoir  $A \rightarrow C$  et  $C \rightarrow B$ .

# Clés de relation

24

- **Clé d'une relation** : on dit qu'un attribut ou une liste d'attributs «  $X$  » est une clé pour la relation  $R(X,Y,Z)$  si
- $X \xrightarrow{R} Y \cup Z$

**exemple :**

dans la relation : **Gestion-Stock** (code-prod, code-dep, libellé-prod, Pu, Adr-dep, Qte-stock), l'ensemble des attributs (code-prod + code-dep + libellé-prod) peut être une clé de Gestion-stock car elle détermine tous les autres attributs.



# Clés de relation (suite)

25

## □ Clé minimale :

□ Une clé  $X$  est dite minimale si  $X \longrightarrow Y \cup Z$  est élémentaire.

□ **Exemple** : la clé (code-prod + code-dep + libellé-prod) n'est pas une clé minimale de Gestion-stock car on a :  
code-prod + code-dep  $\longrightarrow$  Pu

La clé minimale est : (code-prod + code-dep)

# Clés de relation (suite)

26

## Clé primaire/ clé candidate :

- Si une relation possède plusieurs clés, on choisit une parmi elles qui sera appelée *clé primaire* ;
- les autres clés seront appelées *clés candidates* ou *clés secondaires*.
- Exemple : soit la relation **Véhicule** (matricule, marque, type, Nochassis)

Dans cette relation, il existe deux clés matricule et Nochassis. On choisit par exemple « matricule » comme clé primaire et Nochassis comme clé candidate.

# Graphe de dépendance fonctionnelle (1)

27

## □ a) Graphe de DFs

Etant donnée une relation  $r$  et un ensemble de DFs,  $F$ , portant sur les attributs de  $R$ .  $F$  peut être représenté à l'aide d'un graphe dont les nœuds sont les attributs de  $R$  et les arcs sont les dépendances fonctionnelles elles même.

- La construction d'un G.D.F se fait à partir d'un ensemble de DFs canoniques.

# Graphe de dépendance fonctionnelle (2)

28

## □ Graphe de DFs minimum :

Un G.D.F est dit minimum s'il ne contient pas de cycle, ie, il ne contient pas de DFs déduite par transitivité.

□ **Exemple** : Le G.D.F qui représente l'ensemble des DFs suivantes n'est pas minimum.

□  $F = \{X \longrightarrow Y, Y \longrightarrow Z, X \longrightarrow Z\}$

□ Le G.D.F qui représente l'ensemble des DFs suivantes est par contre minimum.  $F = \{X \longrightarrow Y, Y \longrightarrow Z\}$