

IFT3030

Base de données

Chapitre 4

Le modèle relationnel

§Introduction
§Architecture
§Modèles de données
§**Modèle relationnel**
§Algèbre relationnelle
§SQL
§Conception
§Fonctions avancées
§Concepts avancés
§Modèle des objets
§BD à objets

Plan du cours

- Introduction
- Architecture
- Modèles de données
- **Modèle relationnel**
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- Concepts avancés
- Modèle des objets
- BD à objets

Concepts de base

■ Scalaires

- Les valeurs scalaires représentent la plus petite unité sémantique de données.
- Atomiques

- Nom d'une ville
 - 'Montréal' possède une sémantique
 - 'M', 'o', 'n', 't', 'r', 'é', 'a', 'l' n'en possèdent pas

3

Concepts de base

■ Domaine

- Ensemble nommé de valeurs scalaires toutes du même type
- Groupe de valeurs d'où sont extraites les valeurs effectives
 - jours = {lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche}
 - couleurs = {rouge, vert, bleu, jaune, ... , pourpre}
 - âges = [0, 125]
 - salaires = \mathbb{R}^+

4

Concepts de base

■ Relation

– Une relation R sur un ensemble de domaines D_1, D_2, \dots, D_n est constituée de deux parties

• **L'en-tête** : ensemble fixé d'attributs
(<nom-attribut : nom-domaine>)

$\{ \langle A_1 : D_1 \rangle, \langle A_2 : D_2 \rangle, \dots, \langle A_n : D_n \rangle \}$

Chaque attribut A_j prend ses valeurs dans D_j et les noms des attributs sont distincts

5

Concepts de base

■ Relation

• **Le corps** : ensemble de **n-uplets**

(<nom-attribut : valeur-attribut>)

$\{ \langle A_1 : v_{i1} \rangle, \langle A_2 : v_{i2} \rangle, \dots, \langle A_n : v_{in} \rangle \}$ pour le n-uplet i

Tout couple attribut-valeur doit correspondre à un couple attribut-domaine de l'en-tête et

toute valeur $v_{ij} \in D_{ij}$

6

- § Introduction
- § Architecture
- § Modèles de données
- § **Modèle relationnel**
- § Algèbre relationnelle
- § SQL
- § Conception
- § Fonctions avancées
- § Concepts avancés
- § Modèle des objets
- § BD à objets

Concepts de base

■ Attribut

– Colonne d'une relation caractérisée par un nom

■ Relation

– Degré

- nombre d'attributs

– Cardinalité

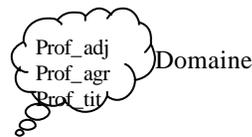
- nombre de n-uplets

- § Introduction
- § Architecture
- § Modèles de données
- § **Modèle relationnel**
- § Algèbre relationnelle
- § SQL
- § Conception
- § Fonctions avancées
- § Concepts avancés
- § Modèle des objets
- § BD à objets

Concepts de base

■ Exemple

Attribut



MAT	NOM	GRADE	COURS	AN_ENT
62945	Gilles	Prof_adj	MRT1111	1997
34560	Myriam	Prof_agr	MRT2221	1993
21539	Claudine	Prof_adj	MRT3331	1999
80200	Bernard	Prof_tit	MRT1112	1982
75902	Yida	Prof_agr	MRT1664	1990

n-uple



Degré

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Concepts de base

■ Relation vs. Table

- Relation : définition d'un objet abstrait (voir plus haut)
- Table : représentation concrète de cet objet abstrait
- Souvent confondues

9

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Concepts de base

■ Propriétés d'une relation

- Pas de duplication des n-uplets
 - ensemble mathématique
- Pas d'ordonnement des n-uplets
 - ensemble mathématique
 - différence avec les modèles précédents
- Pas d'ordonnement des attributs
 - référence par nom

10

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Concepts de base

■ Propriétés d'une relation

- Les valeurs des attributs sont atomiques
 - première forme normale

NF	PRODUIT	
F1	NP	QTE
	P1	100
	P2	350
F2	NP	QTE
	P3	175

NF	NP	QTE
F1	P1	100
F1	P2	350
F2	P3	175

11

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Intégrité des données

■ Introduction

- A tout instant la base représente une partie du monde réel à travers une configuration particulière des valeurs des données
- Certaines configurations des valeurs n'ont pas de sens
 - exemple age d'une personne = -35
- Intégration de règles d'intégrité pour informer le SGBD de certaines contraintes du monde réel

12

Intégrité des données

■ Exemple de contraintes d'intégrité

- Matricule doit être sous la forme nnnnn
- La valeur de GRADE doit appartenir à l'ensemble {Prof_adj, Prof_agr, Prof_tit}
- L'année de recrutement doit être supérieure à 1920 (ou pourquoi pas 1878)
- L'année de recrutement doit être inférieure ou égale à l'année courante
- Tous les professeurs recrutés avant 1994 doivent être au moins agrégés



■ Contraintes spécifiques à une base de données particulière

13

Intégrité des données

■ Propriétés d'intégrité générale

- clés candidates et clés primaires
- clés étrangères

■ Clés candidates

- Ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un n-uplet unique de la relation considérée
- Chaque relation contient au moins une clé candidate

14

Intégrité des données

■ Clés candidates

– Propriétés

Soit R une relation et soit K un ensemble d'attributs (clé candidate),

- Unicité : $\forall t_1 \text{ et } t_2 \in R, K(t_1) \neq K(t_2)$
- Irréductibilité : Il n'existe d'ensemble L tel que $L \subset K$ et L clé candidate

15

Intégrité des données

■ Clés candidates

– Utilité

- mécanisme d'adressage au niveau des n-uplets d'une table

- Exemple

Professeur WHERE MAT = 34560, produit un seul n-uplet en résultat

Professeur WHERE AN_ENT = 1994, va produire un nombre imprévisible de n-uplets en résultat

16

Intégrité des données

■ Clé primaire

- Une clé choisie parmi l'ensemble des clés candidates
- Le choix est arbitraire
- Peut se faire selon des heuristiques
 - Plus simple par exemple
- Exemple
 - Clés candidates MAT et NOM - COURS
 - Clé primaires MAT

17

Intégrité des données

■ Clé primaire

- Exemple
- Etudiant (Code_per, Nom, Prenom, Adresse, Date_nais, Lieu_nais, NASS)
- Quelles sont les clés candidates ?
- Quelle clé primaire choisir ?

18

- Introduction
- Architecture
- Modèles de données
- Modèle relationnel**
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- Concepts avancés
- Modèle des objets
- BD à objets

Intégrité des données

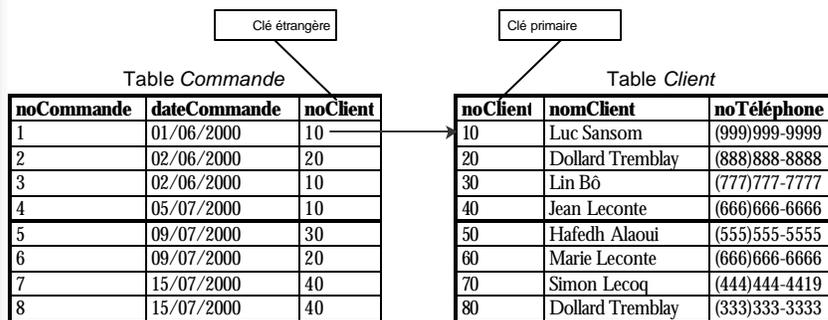
■ Clés étrangères

- Soit R2 une relation, un sous-ensemble d'attributs FK dans R2 est dit clé étrangère dans le cas où
 - * il existe une relation R1 avec une clé candidate CK (R1 peut être R2)
 - * à tout instant, chaque valeur de FK dans R2 est identique à une valeur de CK dans R1

- Introduction
- Architecture
- Modèles de données
- Modèle relationnel**
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- Concepts avancés
- Modèle des objets
- BD à objets

Intégrité des données

■ Clés étrangères



Intégrité des données

■ Clés étrangères

– Propriétés

- Il n'est pas nécessaire d'avoir pour toute valeur de CK dans R1 une valeur identique de FK dans R2
- Une clé étrangère est composée ou simple
- Chaque attribut d'une clé étrangère doit être défini sur le même domaine que son équivalent dans la clé candidate correspondante

21

Intégrité des données

■ Clés étrangères

– Propriétés

- Un attribut d'une clé étrangère peut faire partie d'une clé candidate de sa relation

Employe (Mat, Nom, ...)

Projet (Num, Designation, ...)

Role (Mat, Num, role)

- On dit que R2 référence R1

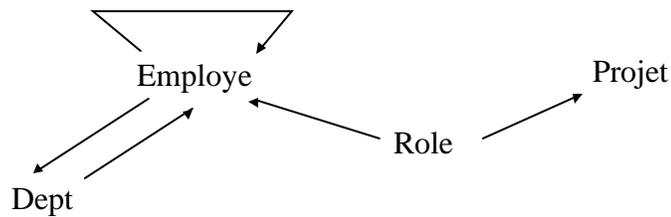
22

Intégrité des données

■ Clés étrangères

– Diagramme référentiel

- représentation graphique des références



Intégrité des données

■ Exemple

Employee (mat# : entier, nom : caracteres, adresse : caracteres, dept : entier, sup : entier)

Projet (num# : entier, designation : caracteres)

Departement (dept# : entier, dir : entier, nom : caracteres)

Role (mat# : entier, num# : entier, role_emp : caracteres)

Intégrité des données

■ Clés étrangères

– Intégrité référentielle

- La base de données ne doit pas contenir une valeur de clé étrangère non « unifiable »
- Cette contrainte est dite référentielle
- Conséquence : Il faut créer et détruire les n-uplets dans un ordre particulier
- Dans certain SGBD (version 92 de SQL), on peut différer la vérification d'une contrainte d'intégrité référentiel (restreinte vs. cascade)

25

Aperçu de déclaration de table en SQL

■ Domaine de base

- Entier : INTEGER
- Décimal : DECIMAL(m,n)
- Réel flottant : FLOAT
- Chaîne de caractères : CHAR(n)
- Date : DATE
- ...

■ Création d'une table (forme simple)

```
CREATE TABLE <nom-table> ( <attribut> <type>,  
  <attribut> <type>, <attribut> <type>, ...)
```

26

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Aperçu de définition de données en SQL

■ Création d'une table (forme simple)

– Exemple

```
CREATE TABLE VIN (
  NV INTEGER,
  CRU CHAR(10),
  MIL INTEGER)
```

■ Contraintes d'intégrité

- NOT NULL
- UNIQUE ou PRIMARY KEY
- FOREIGN KEY
- REFERENCES
- CHECK

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Aperçu de définition de données en SQL

■ Avec contraintes d'intégrité

Exemple (SQL 86)

```
CREATE TABLE COMMANDE (
  NC INTEGER UNIQUE NOT NULL,
  NV INTEGER NOT NULL,
  QUANTITE DECIMAL(6))
```

Exemple (SQL 89)

```
CREATE TABLE COMMANDE (
  NC INTEGER PRIMARY KEY,
  NV INTEGER NOT NULL REFERENCES VIN,
  QUANTITE DECIMAL(6) CHECK(QUANTITE > 0))
```

```
NC INTEGER,
PRIMARY KEY (NC),
NV INTEGER NOT NULL,
FOREIGN KEY (NV) REFERENCES VIN, } SQL 92
```

Aperçu de définition de données en SQL

■ Modification du schéma d'une table

```
ALTER TABLE <table>  
ADD COLUMN <attribut> <type>, <attribut> <type>, ...
```

– Exemple

```
ALTER TABLE COMMANDE  
ADD COLUMN DATE_COM DATE
```

■ Suppression d'une table

```
DROP TABLE <table>
```

– Exemple

```
DROP TABLE COMMANDE
```

Aperçu de définition de données en SQL

■ Exemple (SQL)

Employe (mat# : entier, nom : caracteres,
adresse : caracteres, dept : entier, sup : entier)

Projet (num# : entier, designation : caracteres)

Departement (dept# : entier, dir : entier, nom :
caracteres)

Role (mat# : entier, num# : entier, role_emp :
caracteres)

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Aperçu de définition de données en SQL, exemple

Employe (mat# : entier, nom : caracteres, adresse : caracteres, dept : entier, sup : entier)

```
CREATE TABLE EMPLOYE (
  MAT INTEGER,
  NOM CHAR(12),
  ADRESSE CHAR(25),
  DEPT INTEGER NOT NULL,
  SUP INTEGER NOT NULL,
  PRIMARY KEY (MAT),
  FOREIGN KEY (DEPT) REFERENCES DEPARTEMENT,
  FOREIGN KEY (SUP ) REFERENCES EMPLOYE (MAT));
```

31

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Aperçu de définition de données en SQL, exemple

Projet (num# : entier, designation : caracteres)

```
CREATE TABLE PROJET (
  NUM INTEGER,
  DESIGNATION CHAR(20),
  PRIMARY KEY (NUM));
```

32

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Aperçu de définition de données en SQL, exemple

Departement (dept# : entier, dir : entier, nom : caracteres)

```
CREATE TABLE DEPARTEMENT (
  DEPT INTEGER,
  DIR INTEGER NOT NULL,
  DESIGNATION CHAR(20),
  PRIMARY KEY (DEPT),
  FOREIGN KEY (DIR) REFERENCES EMPLOYE (MAT));
```

33

- §Introduction
- §Architecture
- §Modèles de données
- §**Modèle relationnel**
- §Algèbre relationnelle
- §SQL
- §Conception
- §Fonctions avancées
- §Concepts avancés
- §Modèle des objets
- §BD à objets

Aperçu de définition de données en SQL, exemple

Role (mat# : entier, num# : entier, role_emp : caracteres)

```
CREATE TABLE ROLE (
  MAT INTEGER,
  NUM INTEGER,
  ROLE_EMP CHAR(15),
  PRIMARY KEY (MAT, NUM),
  FOREIGN KEY (MAT) REFERENCES EMPLOYE ,
  FOREIGN KEY (NUM) REFERENCES PROJET);
```

34

- ☒ Introduction
- ☒ Architecture
- ☒ Modèles de données
- ☒ **Modèle relationnel**
- ☒ Algèbre relationnelle
- ☒ SQL
- ☒ Conception
- ☒ Fonctions avancées
- ☒ Concepts avancés
- ☒ Modèle des objets
- ☒ BD à objets

Aperçu de l'insertion de données en SQL

■ Insertion

– Forme

```
INSERT INTO <table> [ (<liste_colonnes>)]  
{VALUES (<liste_valeurs>) | <requete>}
```

– Exemple 1

```
INSERT INTO Employe (mat, nom, adresse, dept, sup)  
VALUES (34098, 'Gilles', '3456, Gaspé, Mtl', 456, 68735)
```