

Le modèle relationnel

Généralités

- ◆ Notions de modèle de données :
 - Un modèle est un ensemble d'outils utilisés pour décrire et manipuler des données
- ◆ Modèle relationnel
 - Créé par CODD (69/70)
 - La majorité des SGBD actuels sont basés sur ce modèle
 - Les données sont organisées sous forme de tables à deux dimensions, une table correspondant à une relation.
 - Les données sont manipulées par des opérateurs de l'algèbre relationnelle
 - L'état de la base est défini par un ensemble de contraintes d'intégrité
- ◆ Normalisation
 - Au modèle relationnel est associé la théorie de la normalisation des relations qui permet de se débarrasser des incohérences au moment de la conception d'une base de données.

1

Concepts du modèle (1)

Table (relation) : Vision tabulaire du relationnel

- les données sont représentées dans une table
- Exemple : la table OUVRAGES décrit un ouvrage

OUVRAGES

Côte	Titre	Editeur	Année	NbExemplaire	Thème
12TA1	Réseaux informatique	Eyrolles	1998	10	Interconnexion,
13G01	Algorithmes génétiques	Eyrolles	1994	5	Informatique,

2

Concepts du modèle (2)

- ◆ Attribut : nom donnée à une colonne d'une table (exemple cote, titre, editeur, etc...). La première ligne de la table comporte ses attributs
- ◆ Nom de la table (ou de la relation) : OUVRAGES
- ◆ Tuple (ou n-uplet) : nom donné à une ligne comportant des valeurs saisies.
- ◆ Extension d'une table : le contenu de la table à savoir tous les tuples
- ◆ Cardinalité : nombre de tuples de la relation. Exemple la cardinalité dans OUVRAGES est 2
- ◆ L'ordre des lignes et des colonnes n'est pas significatif
- ◆ Pas de lignes identiques
- ◆ Une case possède une et une seule valeur

3

Concepts du modèle (3)

- ◆ Attribut
 - Nom donné à une colonne d'une relation
 - Prend ses valeurs dans un domaine
- ◆ Domaine : ensemble de valeurs possibles prises par les attributs
 - Exemple :
 - Entier, reel, chaines de caractères
 - Franc
 - Salaire = {4 000..100 000}
 - Couleur = {bleu, blanc, rouge}

4

Notions de clé primaire

- ◆ Clé primaire : Groupe d'attributs minimum qui détermine un tuple d'une manière unique dans la table
 - Exemple de clés : le numéro de la sécu, le numéro d'étudiant
 - La clé de la table OUVRAGE est l'attribut « cote », car la cote permet de déterminer de façon unique une ligne de la table.
 - ATTENTION : la clé de détermine par rapport à toutes les valeurs possibles de l'attribut (ou les attributs) formant la clé primaire, et surtout pas par rapport aux valeurs déjà saisie
- ◆ Remarque : toute table doit obligatoirement avoir une clé primaire

5

Schéma d'une table

- ◆ Le schéma d'une table, appelé aussi le schéma en intention, comporte le nom de la relation, ses attributs + format et la clé primaire.
- ◆ La clé primaire est souvent soulignée (et/ou mise en gras)
- ◆ Exemple : le schéma de la table OUVRAGES est
 - OUVRAGES (cote : texte, titre : texte, editeur : texte, NbExemplaire, numérique, année : date, thème : texte)

6

Problème de Redondance des données

- ◆ La redondance = répétition des informations
- ◆ Un des objectifs des SGBD est (de nous permettre) de représenter les données avec le moins de redondance possible
- ◆ Pour chaque ouvrage nous désirons avoir ses auteurs dans la base. Ceci nous amène à créer un attribut supplémentaire : Auteur. Or un ouvrage peut avoir plusieurs auteurs. Nous sommes donc obligés de créer des tuples supplémentaires (un pour chaque auteur). → apparition de redondance
- ◆ Comment éliminer ces redondances ?

7

Éliminer les redondances (1)

- ◆ Pour éliminer les répétitions, nous allons dans un premier temps construire une table auteur comportant tous les auteurs
- ◆ La table auteur est décrite par AUTERS (NumAuteur, nomAuteur, PrénomAuteur). Nous avons rajouté l'attribut NumAuteur pour représenter la clé. NumAuteur est un numéro qui peut être donné automatiquement par le SGBD.
- ◆ Dans la table OUVRAGE, nous pouvons remplacer chaque nom d'auteur par son numéro.
- ◆ Cette représentation nous permet effectivement de réduire la table OUVRAGES, il n'y a que le numéro de l'auteur au lieu du nom et du prénom, mais il y a toujours des redondances. La redondance provient du fait qu'un OUVRAGE peut avoir plusieurs auteurs
- ◆ Pour éliminer ces redondances, nous allons construire une table ECRIT qui permet de relier les OUVRAGES et leurs AUTEURS.

8

Eliminer les redondances (2)

- ◆ Rappelons qu'un des intérêts d'un SGBD est sa possibilité de créer des liens entre les objets
- ◆ Le schéma de la table ECRIT est ECRIT (cote, NumAuteur), il suffit donc de prendre les clés primaires des tables OUVRAGES et AUTEURS et former une nouvelle table, en l'occurrence ECRIT.
- ◆ La base de données décrivant les OUVRAGES sera composé des tables suivantes :
 - AUTEURS (NumAuteur, Nom, Prénom)
 - OUVRAGES (cote, Titre, NbExemplaire, Année, Editeur, Theme)
 - ECRIT (cote, NumAuteur)
- ◆ Noter que nous avons supprimé l'attribut NumAuteur de la table OUVRAGES

9

Clé étrangère

- ◆ Les attributs cote et NumAuteur de la table ECRIT proviennent en fait respectivement des tables OUVRAGES et AUTEURS. Ces deux attributs sont clés primaires dans chacune de ces tables
- ◆ Définition : nous appelons Clé étrangère toute clé primaire apparaissant dans une autre table
- ◆ Exemple :
 - NumAuteur est une clé étrangère dans la table ECRIT
 - Cote est aussi une clé étrangère dans ECRIT
 - Par convention, une clé étrangère est soulignée en pointillé (et/ou mise en italique)
- ◆ Attention : la notion de clé est toujours liée à une table. Un attribut (ou groupe d'attributs) est clé primaire, ou clé étrangère dans une table donnée.

10

Contraintes d'intégrités (1)

- ◆ Un des avantages des bases de données par rapport à une gestion de fichiers traditionnelle réside dans la possibilité d'intégrer des contraintes que doivent vérifier les données à tout instant.
- ◆ Exemple : on souhaite poser les contraintes suivantes :
 - Le nombre d'exemplaire de chaque OUVRAGE doit être supérieur à 0 (zéro)
 - Chaque OUVRAGE doit avoir au moins un auteur
 - Tec.
- ◆ Ceci est possible grâce à la notion de contraintes d'intégrité
- ◆ Définition : Des contraintes d'intégrités sont des assertions qui doivent être vérifiées à tout moment par les données contenues dans la base de données.

11

Contraintes d'intégrités (2) Trois type de C.I. obligatoires

- ◆ Contrainte de clé : une relation doit posséder une clé primaire
- ◆ Contrainte d'entité : un attribut d'un clé ne doit pas posséder de valeurs nulles (vides)
- ◆ Contraintes de référence (pour les clés étrangères), c'est une contrainte exprimée entre deux tables. Tout tuple d'une relation faisant référence à une autre relation doit se référer à un tuple qui existe :
 - Intuitivement, cela consiste à vérifier que l'information utilisée dans un tuple pour désigner un autre tuple est valide, notamment si le tuple désigné existe bien
 - En d'autre terme, quand on désigne un attribut comme clé étrangère, les seules valeurs que peut prendre cet attribut sont celles qui sont déjà saisies dans la table qu'il référence.

12

Contraintes d'intégrités (3)

- ◆ Contrainte optionnelle
 - Contrainte de domaine : liée au domaine de définition d'un attribut
 - Exemple : NbExemplaire > 0
- ◆ Les contraintes d'intégrité sont vérifiées (exécutées) à chaque mise à jour de la base de données (ajout, suppression ou modification d'un tuple). Si lors d'une mise à jour une contrainte n'est pas satisfaite, cette mise à jour ne peut pas avoir lieu.

13

Exemple (1)

- ◆ Schéma de la relation AUTEURS
 - AUTEURS (NumAuteur, Nom, Prénom)
- ◆ Schéma de la relation EDITEURS
 - EDITEURS (NumEditeur, Nom, Adresse)
- ◆ Schéma de la relation OUVRAGES (avec la modification de l'exercice précédent)
 - OUVRAGES (cote, Titre, NbExemplaire, Annee, *NumEditeur*, Theme)
 - Clé primaire : cote
 - Contrainte de domaine : NbExemplaire > 0
 - Contrainte référentielle : OUVRAGES.NumEditeur est une clé étrangère et fait référence à EDITEURS.NumEditeur ou encore OUVRAGES.NumEditeur REFERENCE EDITEURS.NumEditeur

14

Exemple (2)

◆ Schéma de la table ECRIT

- ECRIT (NumAuteur, cote)
- Clé primaire : NumAuteur, cote
- Contraintes référentielles :
 - ECRIT.NumAuteur REFERENCE AUTEURS.NumAuteur
 - ECRIT.cote REFERENCE OUVRAGES.cote
 - Le fait d'écrire ECRIT.cote REFERENCE OUVRAGES.cote, c'est-à-dire définir l'attribut cote dans ECRIT comme clé étrangère, implique une contrainte référentielle. Ceci se traduit par ; les seules valeurs que peut prendre cote dans ECRIT sont celles qui sont déjà saisies dans cote d'OUVRAGES.

15

Schéma d'une base de données

- ◆ Le schéma d'une base de données est composée de l'ensemble des schémas des tables (relations) définies dans cette base de données
- ◆ Exemple de schéma de la base de données permettant la gestion de notices bibliographiques :
 - AUTEURS (NumAuteur, Nom, Prénom)
 - OUVRAGES (cote, Titre, NbExemplaire, Annee, *NumEditeur*, Theme)
 - Contrainte de domaine : NbExemplaire > 0
 - Contrainte référentielle : OUVRAGES.NumEditeur REFERENCE EDITEURS.NumEditeur
 - ECRIT (NumAuteur, cote)
 - Contraintes référentielles :
 - ECRIT.NumAuteur REFERENCE AUTEURS.NumAuteur
 - ECRIT.cote REFERENCE OUVRAGES.cote
 - EDITEURS (NumEditeur, Nom, Adresse)

16

Remarques (1)

- ◆ Beaucoup de systèmes de gestion de données (et non de gestion de bases de données) sont vendus comme étant relationnels, souvent parce qu'ils présentent les données sous forme de tables.
- ◆ Un système est dit minimalement relationnel s'il satisfait aux conditions suivantes :
 - Toute information dans la base est représentée par des valeurs dans des tables,
 - Il n'y a pas de pointeurs visibles par l'utilisateur entre les tables,
 - Le système doit supporter au moins les opérateurs relationnels de restriction, projection, jointure naturelle.

17

Remarques (2)

- ◆ Un système est dit complètement relationnel s'il satisfait, en plus, aux conditions suivantes :
 - Il supporte tous les opérateurs de l'algèbre relationnelle,
 - Il supporte la contrainte d'unicité de clé d'une relation
 - Il supporte les contraintes référentielles qui permettent de s'assurer que la valeur d'une donnée d'une relation existe dans une autre relation (notion de foreign key ou clé étrangère)
- ◆ En dépit de sa simplicité et de son élégance le modèle relationnel n'apporte pas une réponse satisfaisante à tous les problèmes des applications . Il faut :
 - Pouvoir prendre en compte des « objets » structurés ainsi que les opérations qui leur sont associées (base de données orientées objet)
 - Prendre en compte des données peu structurées : textes, sons, images, graphiques (bases de données multi-média)
 - Faire le pont avec l'intelligence artificielle afin de pouvoir déduire de nouvelles données à partir de celles existant déjà (bases de données déductives)

18