

Introduction | Architectures | Objectifs | Fonctions | Historique | Principaux types

LES PRINCIPAUX TYPES DE SGBD

<p>♦ SGBD pour usage professionnel:</p> <ul style="list-style-type: none"> ✓ Oracle: propriétaire (Oracle Corporation) ; ✓ DB2: propriétaire (IBM) ; ✓ MS SQL Server : propriétaire (Microsoft) ; ✓ Sybase: propriétaire (Sybase) ; ✓ Ingres: libre ; ✓ Informix: propriétaire (IBM) ; ✓ <u>MySQL: libre ou propriétaire (Oracle Corporation) selon l'utilisation :</u> ✓ PostgreSQL: libre. 	<p>♦ SGBD pour usage personnel:</p> <ul style="list-style-type: none"> ✓ Paradox: propriétaire (Corel) ; ✓ Interbase: propriétaire (Borland Software) ; ✓ MS Access: propriétaire (Microsoft) ; ✓ OpenOffice Base : libre. <p>♦ SGBD Objet:</p> <ul style="list-style-type: none"> ✓ Object Store: propriétaire (Object Design); ✓ Versant: propriétaire (Versant) ; ✓ O2: propriétaire (O2 Technology) ; ✓ DB4O: libre.
---	---

57

Conception des Bases de données relationnelles

Chapitre 3:

Introduction

- ✓ Le modèle relationnel
- ✓ Caractéristiques du modèle relationnel

Structures de données de base

- ✓ Le domaine, la relation et l'attribut
- ✓ Tuple, Arité et Cardinalité
- ✓ Schéma d'une relation
- ✓ Intention et Extension d'une relation
- ✓ Schéma d'une base de données

Contraintes d'intégrité structurelles

- ✓ Clé d'une relation, clé candidate, clé primaire et étrangère
- ✓ Comment choisir une clé primaire
- ✓ Contraintes d'unicité
- ✓ Contraintes de références
- ✓ Contraintes de domaines
- ✓ Contraintes valeurs par défaut

Introduction | STRUCTURES DE DONNÉES DE BASE | CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

INTRODUCTION

La **conception** d'une base de données consiste à déterminer les **structures de données** et les **règles de gestion** que l'on peut appliqué sur les données, afin d'en **assurer l'intégrité et la cohérence** pour répondre aux besoins des clients;

Dans le cadre de ce cours nous étudierons la conception des bases de données qui reposent sur le **Modèle Relationnel**.

59

Introduction | STRUCTURES DE DONNÉES DE BASE | CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

LE MODÈLE RELATIONNEL

- ♦ Le modèle relationnel a été introduit au début des années 70 par Edgar Frank Codd, en publiant un article intitulé: "Un modèle relationnel de données pour les grandes banques de données partagées", actuellement, cet article est l'un des plus influents de l'histoire de l'informatique, parce qu'il a défini les bases du modèle relationnel;

60

LE MODÈLE RELATIONNEL

- ◆ Ce modèle est fondé sur des bases mathématiques solides (logique et théorie des ensembles);
- ◆ C'est le premier modèle de données indépendant des critères de stockage, il consiste à percevoir la BD comme un ensemble de **relations** qui peuvent être représentées sous forme de table à deux dimensions: Les **colonnes** correspondent aux attributs et les **lignes** correspondent aux tuples, ce **modèle utilise un seul concept (une seule structure de données) qui est la relation**, ce qui lui rend plus facile à utiliser même pour un débutant en informatique;
- ◆ Aujourd'hui la majorité des SGBD sont basés sur le modèle relationnel. Il est associé à la théorie de la normalisation des relations dont le but est d'éliminer les comportements anormaux des relations pendant les mises à jour et la redondance des données.

61

CARACTÉRISTIQUES DU MODÈLE RELATIONNEL

- ◆ Les principales caractéristiques du modèle relationnel sont:
 - ✓ **Absence de pointeurs** : les liens (associations) sont stockés dans des nouvelles relations (tables);
 - ✓ **Simplicité** : les interrogations sont écrites dans un langage non procédural (SQL) basé sur l'algèbre relationnelle;
 - ✓ **Séparation entre schéma logique** (« quoi ») **et schéma physique** (« comment »);
 - ✓ **La base de la plupart des SGBD** (Oracle, DB2, Informix, MySQL, SQL Server, Access...).

62

LES STRUCTURES DE DONNÉES DE BASE

- ◆ Le modèle relationnel est fondé sur la **théorie mathématique bien connue des relations**. Cette théorie se construit à partir de la **théorie des ensembles** (un ensemble est une collection non ordonnée d'éléments **distincts**);
- ◆ Trois notions sont importantes pour concevoir les BD relationnelles :
 - ✓ **Domaine** ;
 - ✓ **Relation**;
 - ✓ **Attribut**.

63

LE DOMAINE

Définition:

Domaine: est caractérisé par un nom qui désigne un **ensemble fini ou infini de valeurs atomiques** qui ont un sens pour l'utilisateur.

- ◆ Un domaine est un ensemble nommé dans lequel **les données peuvent prendre leurs valeurs**;
- ◆ **Exemples**:
 - ✓ Domaines des chiffres: D1={0,1,2,3,4,5,6,7,8,9};
 - ✓ Domaine des prénoms : D2 = {Ahmed, Fadoua, Rachida};
 - ✓ Domaines des matières : D3={ Algo, BDR, SI};
 - ✓ Domaine des villes : D4= { Safi, Raba, Marrakech, Tanger };
 - ✓ Domaine des entiers: D5= { 1, 2, 3....}.

64

IntroductionSTRUCTURES DE DONNÉES DE BASECONTRAINTES D'INTÉGRITÉ STRUCTURELLES

LE DOMAINE

Rappel:

Produit cartésien:
Le produit cartésien d'un ensemble de domaines D_1, D_2, \dots, D_n noté $D_1 \times D_2 \times \dots \times D_n$ est l'ensemble des vecteurs $\langle Val_1, Val_2, \dots, Val_n \rangle$ tel que $\forall i, Val_i \in D_i$ pour tout $i=1, 2, \dots, n$.

➤ **Exemple:** prenant les domaines $D_2 = \{Ahmed, Fadoua, Rachida\}$ et $D_3 = \{Algo, BDR, SI\}$, le produit cartésien $D_2 \times D_3$ donne les 9 couples suivants :

D1	D2
Ahmed	Algo
Ahmed	BDR
Ahmed	SI
Fadoua	Algo
Fadoua	BDR
Fadoua	SI
Rachida	Algo
Rachida	BDR
Rachida	SI

65

IntroductionSTRUCTURES DE DONNÉES DE BASECONTRAINTES D'INTÉGRITÉ STRUCTURELLES

LA RELATION

Définition:

Relation:
Une relation R est un sous-ensemble de produit cartésien d'un ensemble de domaine $D_1 \times D_2 \times \dots \times D_n$ désignée par un nom. Elle est notée sous la forme: $R \subset D_1 \times D_2 \times \dots \times D_n$

➤ Une relation est la **structure véritablement centrale du modèle relationnel**;

➤ Une relation est composée de **vecteurs**, une représentation commode de la relation est une **table à deux dimensions (matrice)** où chaque **ligne est un vecteur** et chaque **colonne correspond à un domaine**;

➤ **Exemples:** Soient les domaines $D_2 = \{Ahmed, Fadoua, Rachida\}$, $D_3 = \{Algo, BDR, SI\}$ et $D_4 = \{Safi, Raba, Marrakech, Tanger\}$ On peut avoir une relation Etudiant définie sur les domaines D_2, D_3 , et D_4 où $Etudiant \subset D_2 \times D_3 \times D_4$ avec les lignes suivantes :

La relation Etudiant		
Colonne1	Colonne2	Colonne3
Ahmed	Algo	Safi
Fadoua	Algo	Rabat
Fadoua	BDR	Rabat
Rachida	BDR	Marrakech
Rachida	Algo	Marrakech

66

IntroductionSTRUCTURES DE DONNÉES DE BASECONTRAINTES D'INTÉGRITÉ STRUCTURELLES

L'ATTRIBUT

Définition:

Attribut: est caractérisé par un nom unique dans la relation qui représente une colonne de cette relation, il prend ses valeurs dans un des domaines sur lesquels la relation est définie.

➤ Un attribut est une **donnée qui contribue à la description de l'objet que l'on veut modéliser**;

➤ **Exemples:** dans l'exemple précédent on peut avoir les attributs:

- ✓ attribut A1 = Pre_Etudiant qui a des valeurs dans D_2 ;
- ✓ attribut A2 = Mat qui a des valeurs dans D_3 ;
- ✓ attribut A3 = Ville_Resi qui a des valeurs dans D_4 .

➤ D'où la relation R deviennent :

La relation Etudiant		
Pre_Etudiant	Mat	Ville_Resi
Ahmed	Algo	Safi
Fadoua	Algo	Rabat
Fadoua	BDR	Rabat
Rachida	BDR	Marrakech
Rachida	Algo	Marrakech

67

IntroductionSTRUCTURES DE DONNÉES DE BASECONTRAINTES D'INTÉGRITÉ STRUCTURELLES

L'ATTRIBUT

Remarques:

➤ Un **attribut n'est pas le domaine** (c'est un sous-ensemble de domaine), par exemple $\{Algo, BDR\}$;

➤ L'ordre des attributs **n'a pas d'importance** dans la relation;

➤ Plusieurs attributs peuvent avoir le **même domaine** ce qui permet à un **même domaine d'être présent plusieurs fois dans la même relation**, dans ce cas il faut **donner des noms différents aux attributs**;

Exemples: soit la relation Etudiant définit sur les domaines $D_2 = \{Ahmed, Fadoua, Rachida\}$ et $D_4 = \{Safi, Raba, Marrakech, Tanger\}$ où $Etudiant \subset D_2 \times D_4 \times D_4$ dans laquelle l'attribut A1 = Pre_Etudiant qui représente le prénom qui a des valeurs dans D_2 , attribut A2 = Ville_Resi qui représente la ville de résidence qui a des valeurs dans D_4 et attribut A3 = Ville_Naiss qui représente la ville de naissance qui a des valeurs dans D_4 .

68

TUPLE, ARITÉ ET CARDINALITÉ

Définition:

Tuple(vient du mot anglais *tuple*): est une ligne de la relation correspondant à un enregistrement.

- Un tuple d'une relation désigne une **ligne dans la table** représentant cette relation;

Exemples: dans la relation Etudiant, (Ahmed, Algo, Safi) est un tuple.

Définitions:

- ✓ **Arité:** est le nombre des attributs d'une relation;
- ✓ **Cardinalité:** est le nombre des tuples d'une relation.

- L'arité d'une relation est le **nombre des colonnes** de la table **Exemples:** l'arité de la relation Etudiant, (Pre_Etudiant, Mat, Ville_Resi) est 3;
- La cardinalité d'une relation est le **nombre des lignes de la table**;

69

SCHEMA D'UNE RELATION

Définition:

Schéma de relation: est le nom de la relation suivi de la liste des attributs et de la définition de leurs domaines, il est noté sous la forme: $R(A1 : D1, A2 : D2, ..., An : Dn)$.

Exemple:

Etudiant(Num_Etu: Integer, Pre_Etudiant: VarChar, Ville_Resi: VarChar, Sexe: DS)

où: DS est le domaine des sexes {Masculin, Féminin} et Integer domaine des entiers et VarChar domaine des chaînes de caractères de longueur variable.

Pour simplification nous pouvons écrire le schéma de la manière suivante:

Etudiant(Num_Etu, Pre_Etudiant, Ville_Resi, Sexe)

70

INTENTION ET EXTENSION D'UNE RELATION

Définition:

l'intention d'une relation: représente le schéma d'une relation.

- C'est-à-dire que les **attributs et leurs domaines** qui sont figés au cours du temps;
- L'intention est le **résultat de la description des données**.

Définition:

L'extension d'une relation: appelée aussi **instance** ou **n-uplets**, il représente l'ensemble des tuples d'une relation.

- Les tuples d'une relations **évoluent au cours du temps** (ajout, suppression et modification);
- L'extension est le **résultat des manipulations des données** et représente **un état de la relation dans un moment donné**.

71

SCHEMA D'UNE BASE DE DONNÉES

Définition:

Le schéma d'une base de données: est l'ensemble des intentions des relations de la base de données ou encore l'ensemble des schémas de ses relations.

- Une base de données est **composée de plusieurs relations**;
- Les données qui concernent une application sont divisées en parties, chaque relation stockant une partie des données;
- **Exemples:** gestion des étudiants:
 - ✓ **Etudiant:** stocke les informations sur les étudiants;
 - ✓ **Matières:** stocke les informations sur les matières;
 - ✓ **Notes:** stocke les informations sur les notes.

72

Introduction | STRUCTURES DE DONNÉES DE BASE | CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

RÉSUMÉ

La relation ou la table Etudiant

Pre_Etudiant	Mat	Ville_Resi
Ahmed	Algo	Safi
Fadoua	Algo	Rabat
Fadoua	BDR	Rabat
Rachida	BDR	Marrakech
Rachida	Algo	Marrakech

Attribut ou colonne

Tuple, ligne ou enregistrement

Valeurs de l'attribut Ville_Resi dans le domaine VarChar

- ✓ Schéma ou intention: Etudiant(Pre_Etudiant, Mat, Ville_Resi);
- ✓ L'extension ou les n-uplets: L'ensemble des tuples de la relation.
- ✓ Arité : 3;
- ✓ Cardinalité : 5.

73

Introduction | STRUCTURES DE DONNÉES DE BASE | CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

- ♦ Les **contraintes d'intégrité** ou **règles d'intégrité** sont les conditions (logique) qui doivent être satisfaites par les données contenues dans une base de données, ces contraintes permettent de faciliter la gestion de la cohérence des données:
 - ✓ Vérifier / valider automatiquement (en dehors des applications utilisateurs) les données lors des mises à jour : ajout, modifications, suppression;
 - ✓ Déclencher automatiquement des mises à jour entre les relations pour maintenir la cohérence globale.

74

Introduction | STRUCTURES DE DONNÉES DE BASE | CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

- ♦ La gestion automatique des contraintes d'intégrité est l'un des outils les plus importants d'un SGBD;
- ♦ Principaux contraintes sont:
 - ✓ Contraintes d'unicité;
 - ✓ Contraintes de références;
 - ✓ Contraintes de domaines;
 - ✓ Contraintes valeurs par défaut.

75

Introduction | STRUCTURES DE DONNÉES DE BASE | CONTRAINTES D'INTÉGRITÉ STRUCTURELLES

CLÉ D'UNE RELATION

Définition:

Clé de la relation: est un attribut ou un ensemble d'attributs qui permet d'identifier d'une manière unique chaque tuple de la relation.

- ♦ Une clé est un ensemble minimal d'attributs dont la connaissance des valeurs permet d'identifier un tuple de façon unique au sein d'une relation;
- ♦ Une clé n'est pas forcément un unique attribut, elle pourra être composée de plusieurs attributs.

76

CLÉ CANDIDATE, CLÉ PRIMAIRE ET ÉTRANGÈRE

Définitions:

- ✓ **Clé candidate**: est un attribut ou un ensemble d'attributs est appelé clé candidate s'il peut jouer le rôle de clé dans une relation;
- ✓ **Clé primaire**: lorsqu'on dispose de plusieurs clés candidates dans une relation, il faut retenir une seule clé parmi ces clés candidates, la clé retenue s'appellera alors clé primaire, une clé primaire doit être soulignée dans un schéma de relation;
- ✓ **Clé Étrangère**: est un groupe d'attributs d'une relation qui apparaît comme clé primaire dans une autre relation, une clé étrangère doit être précédée par le symbole # dans un schéma de relation.

- Toute relation doit avoir au moins une clé primaire (équivalent à l'identifiant de l'entité dans le MCD).

77

COMMENT CHOISIR UNE CLÉ PRIMAIRE

- Le choix de la clé primaire est généralement effectué en fonction des deux critères suivants :
 - ✓ On choisit la clé candidate ayant le plus petit nombre d'attributs: il est préférable de minimiser le nombre d'attributs manipulés;
 - ✓ On choisit la clé candidate en fonction de son utilisation pour identifier les tuples : il faut privilégier la clé candidate dont l'usage serait le plus fréquent pour identifier les tuples de la relation et sa valeur reste figée au fil du temps ;
- La clé primaire sert de **référence externe** (clé étrangère), il ne doit jamais être modifié (dans le cas de suppression ou de modification il faut avoir des **répercussions sur les autres relations**).

78

EXERCICE D'APPLICATION

- Soient les relations suivantes:

- ✓ Etudiant(NumInscrEt, CinEt, NomEt, PrenomEt, AdrEt, TeleEt)
- ✓ Matiere(CodMat, Intitule)
- ✓ Note(NumInscrEt, CodMat, Note)

Indiquez les clés primaires et les clés étrangères dans chaque schéma de ces relations?
Justifier votre réponse.

79

SOLUTION

- Les clés primaires:

- ✓ Etudiant(NumInscrEt, CinEt, NomEt, PrenomEt, AdrEt, TeleEt)
- ✓ Matiere(CodMat, Intitule)
- ✓ Note(NumInscrEt, CodMat, Note)

Clé concaténée (composée de deux attributs)

80