

Généralités sur les bases de données

ESI – Structures de fichiers et structures de données (SFSD)
N. EL-ALLIA

1

Plan du cours

- Historique
- Pourquoi utiliser une base de données?
- Définitions
- Objectifs d'un SGBD
- Fonctionnalités d'un SGBD
- Niveaux de représentation des données
- Processus de modélisation d'un BD
- Modèles de structuration des données
- Modèle Entité-Association
- Modèle Relationnel
- Algèbre relationnel

2

Historique

- De tout temps, toute organisation (société) a cherché à structurer ses informations.
- Au début (avant l'avènement de l'informatique): informations sous formes de fiches uniquement, classées par ordre alphabétique, chronologique, ...(qui existent encore !),
- Puis avec les supports informatiques
→ infos mémorisées sur supports magnétiques, ...
Il faut les organiser pour pouvoir travailler dessus (retrouver telle info, voir sa valeur, supprimer, mettre à jour, ...) → notion de **fichiers**.

3

Historique

- Jusqu'aux années 60: organisation classique en fichiers gérés par des **SGF** (système de gestion de fichiers)
- années 60: 1^{ère} génération de SGBD :
Au début de l'apparition des bases de données, le niveau conceptuel est très lié à la représentation des données sur les supports physiques →
 - modèle hiérarchique, modèle réseau
- 1970 - 1980: 2^{ème} génération :
plus indépendant des supports :
 - modèle relationnel

4

Historique

- débuts des années 80: 3^{ème} génération :
 - modèle à objets,

5

Pourquoi utiliser une base de données?

La démarche classique consiste à réaliser pour un type d'information, un programme d'application.

Exemple : Une université doit conserver un volume élevé d'information:

noms, prénom, année d'entrée, adresse de l'étudiant, nom, prénom, année d'embauche, grade des enseignants, et des salariés, ...etc

- Ces informations se retrouvent dans différents systèmes de traitement de fichiers.

6

Pourquoi utiliser une base de données?

- Système de gestion de la scolarité,
- Système de comptabilité (paie),
- Système des emprunts de la bibliothèque,
- Système de gestion de personnel,...etc
- Pour obtenir une information, on doit:
 - 1) déterminer le système à consulter
 - 2) trouver la bonne information concernée.**Perte de temps → fichiers multiples**

7

Pourquoi utiliser une base de données?

- De plus, certaines informations sont souvent conservées en plusieurs endroits.

Duplication de données → redondance

Gaspillage au niveau du volume de fichiers

→ fichiers multiples

8

Définitions

Base de données:

Une base de données est un *gros ensemble* d'informations *structurées mémorisées* sur un *support permanent* et qui peut être *partagée* par *plusieurs applications* et qui est *interrogeable* par le contenu.

L'utilisation de fichiers classiques pourrait sembler pouvoir apporter une solution à ce problème. Mais l'utilisation directe de gros fichiers soulève de gros problèmes .

9

Définitions

- Lourdeur d'accès aux données.
- Manque de sécurité (accès direct aux fichiers)
- Pas de contrôle de concurrence (comment palier aux problèmes d'accès simultané par plusieurs utilisateurs)

Il est donc nécessaire d'avoir recours à un logiciel chargé de *gérer* les fichiers constituant une base de données, de prendre en charge les fonctionnalités de *protection* et de *sécurité* et de fournir les différents types d'interfaces nécessaires à l'accès aux données. Ce logiciel est le *Système de Gestion des Bases de Données* (SGBD).

10

Définitions

Définition d'un SGBD:

Un Système de Gestion de Bases de Données (SGBD) est un logiciel de *haut niveau* permettant aux utilisateurs:

- de *structurer*,
- d'*insérer*,
- de *modifier*,
- de *rechercher* de manière efficace des données spécifiques, au sein d'une grande quantité d'informations, *stockées* sur mémoires secondaires *partagée* de manière transparente par *plusieurs utilisateurs*.

11

Définitions

Les SGBD assurent d'autres fonctions importantes, notamment :

- la *protection* de la base de données

→ La protection implique à la fois:

la *protection du système* contre les pannes logicielles et matérielles
et la *protection sécuritaire* contre les accès illicites ou malveillants.

- et son *entretien* à long terme

→ Une grande base de données peut être utilisées de *nombreuses années*.

Le SGBD doit donc être capable d'*entretenir* et de faire évoluer ses propres structures dans la durée.

12

Objectifs d'un SGBD

1. Indépendance des programmes aux données :

- Dépendance physique.
- Dépendance logique.

2. Simplicité des Manipulations des données :

- Langage non procédural.
- Recherche, Insertion et Mise à jour.

3. Efficacité des accès aux données :

- Temps de réponse.
- Débit global.

13

Objectifs d'un SGBD

4. Partage et sécurité des données :

- Simultanéité lecture/écriture.
- Confidentialité (authentification, droits d'accès, ...).
- Gestion de la concurrence des transactions.
- Restauration après pannes (journal, sauvegarde).

5. Redondance contrôlée des données :

- Sauvegarde des données.

14

Objectifs d'un SGBD

6. Conception facilité des applications :

- Conception visuelle des BD.
- Conception des traitements.
- Dictionnaire de données.

7. Facilité de l'administration système de la BD :

- Outils d'audit et de tuning.
- Visualisation des plans d'accès.
- Élaboration de statistiques

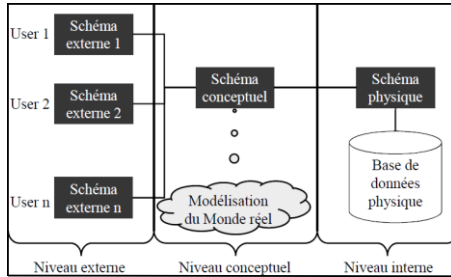
15

Fonctionnalités d'un SGBD

- 1) Définition du schéma de données en utilisant les modèles de données du SGBD.
- 2) Opérations sur les données : recherches, mises-à-jour, etc.
- 3) Partager les données entre plusieurs utilisateurs (mécanismes de transaction).
- 4) Optimiser les performances par le réglage de l'organisation physique des données.

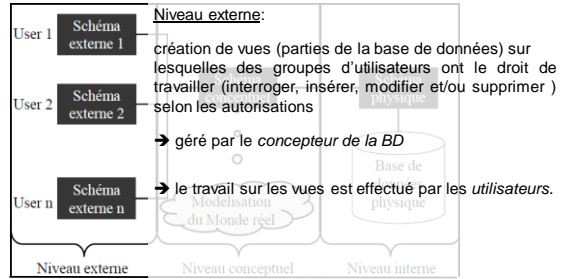
16

Niveaux de représentation des données



17

Niveaux de représentation des données

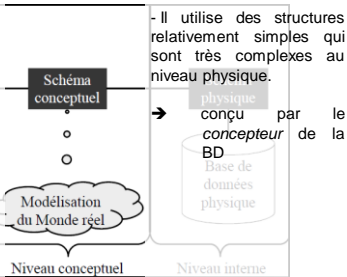


18

Niveaux de représentation des données

Niveau conceptuel ou niveau logique:

- Les données sont considérées de façon sémantique (d'après contenu effectif et les relations qui les relient).
- Il permet de décrire toute la BD:
 - Les entités
 - Les types de données
 - Les relations
 - Les opérations des usagers
 - Les contraintes



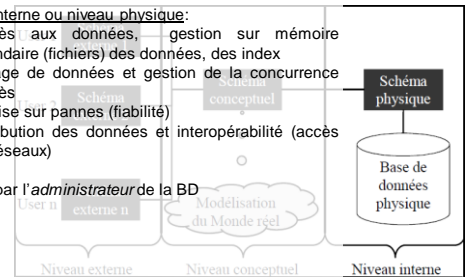
19

Niveaux de représentation des données

Niveau interne ou niveau physique:

- Accès aux données, gestion sur mémoire secondaire (fichiers) des données, des index
- Partage de données et gestion de la concurrence d'accès
- Reprise sur pannes (fiabilité)
- Distribution des données et interopérabilité (accès aux réseaux)

→ géré par l'administrateur de la BD



20

Niveaux de représentation des données

Exemple: Analogie des niveaux avec déclaration de types

Déclaration d'un enregistrement en C :

```
typedef struct
{
    char nom[15];
    char rue[25];
    char ville[15];
}Client;
```

Au niveau externe

L'enregistrement est décrit par des vues :

- vue #1 : tous les clients demeurant dans la ville X
- vue #2 : tous les clients ayant le même nom de famille
- vue #3 : tous les clients demeurant sur la rue Y

Schéma physique

Base de données

21

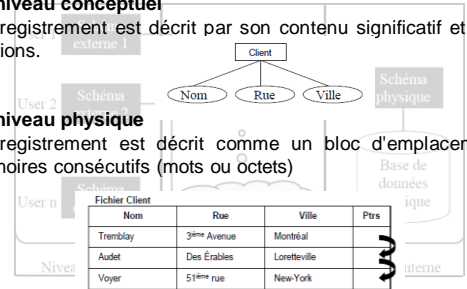
Niveaux de représentation des données

Au niveau conceptuel

L'enregistrement est décrit par son contenu significatif et ses relations.

Au niveau physique

L'enregistrement est décrit comme un bloc d'emplacement mémoires consécutifs (mots ou octets)

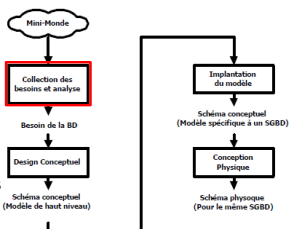


22

Processus de modélisation d'une BD

1) Collection des besoins et analyse

- Collection des besoins des utilisateurs.
 - Spécifiques
 - Détaillés
 - Complets
- Analyse du problème:
 - On se questionne sur quelles informations nous voulons stocker?

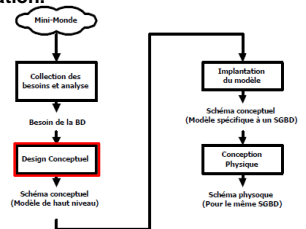


23

Processus de modélisation d'une BD

2) Design Conceptuel

- Création du **schéma conceptuel** basé sur le **modèle Entité-Relation**.
- Description concise des besoins des utilisateurs :
 - des types de données,
 - des relations,
 - des contraintes.
- Facile à comprendre de l'utilisateur.
- Aucune implantation. Il est seulement une Représentation graphique.

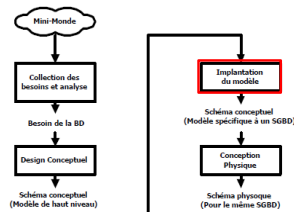


24

Processus de modélisation d'une BD

3) Implantation du modèle

- Déterminer la structure de la BD:
 - relationnelle,
 - réseau
 - hiérarchique.
- Le choix du SGBD dicte donc la structure interne (physique) de la BD.
- Transformation du schéma conceptuel du modèle de données haut niveau au modèle d'implantation de données.

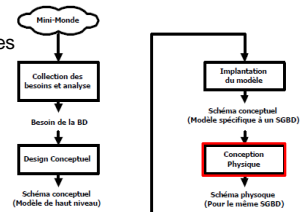


25

Processus de modélisation d'une BD

4) Conception physique

- Spécification des structures de stockage interne et de l'organisation des fichiers de la BD.



26

Processus de modélisation d'une BD

Pourquoi une méthode de conception?

- Plusieurs façons d'aborder la conception d'une BD:
 - Intuition + création directe de la BD
 - Suivre une méthode de conception (MCD, MLD, MPD)
 - Entité/Association (E/A)
 - Merise
 - UML
- Suivre son intuition peut conduire à des erreurs:
 - Redondances
 - Valeurs nulles
 - Difficulté de gestion
 - Impossibilité de répondre à certaines questions
- Une fois la base de données créée, difficile à modifier
- Les outils de conception sont une aide précieuse

27

Processus de modélisation d'une BD

Exemple1 de mauvaise conception

N°	Nom	Prénom	Ville	Pays	Immatriculation	Marque	Couleur
1	Flen	aaa	Alger	ALG	16-106-1000	Renault	Rouge
2	Alt	Alt	Tizi	ALG	15-109-5200	Peugeot	Vert
3	Ben	Ben	Oran	ALG	31-103-1400	Renault	Rouge
4	Flen	aaa	Alger	ALG	16-112-2500	Renault	Jaune

- Redondance des données et incohérence potentielle
 - Personne répétée pour chaque voiture :
ex. Si Flen aaa change de ville et qu'une seule ligne est mise à jour...
 - Redondance Ville/Pays : impact d'une erreur de saisie
- Anomalies de mises à jour et besoin de valeurs nulles.
 - Comment insérer une personne sans voiture ?
 - Sémantique de calculs avec des valeurs nulles...
 - Comment supprimer la dernière voiture d'une personne ?

28

Processus de modélisation d'une BD

Exemple2 de mauvaise conception

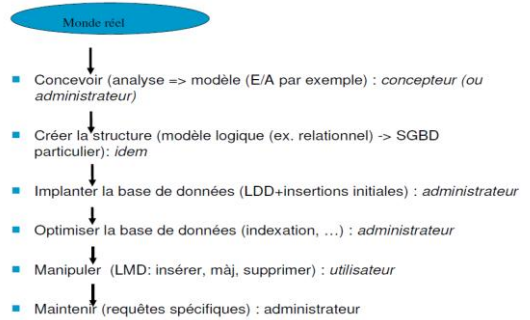
N°	Nom	Prénom	Ville	Pays	Enfant1	Enfant2	Enfant3	NbEnfants
1	Flen	aaa	Alger	ALG	Pipo	Lolo		2
2	Ait	Ait	Tizi	ALG	Sissi			1
3	Ben	Ben	Oran	ALG	Sam	Tit	Yoyo	3
4	Moh	Med	Alger	ALG				0

- Redondance cachée
 - Nombre d'enfants vs enfants
- Difficulté de gestion
 - Comment gérer les personnes ayant plus de 3 enfants !
 - Comment afficher la liste des enfants ?

29

Processus de modélisation d'une BD

En résumé: Cycle de vie d'une base de données



30

Modèles de structuration des données

La modélisation des données permet de décrire correctement l'architecture d'une BD.

- les données;
- leurs relations;
- leur sémantique;
- les contraintes.

Deux grands modèles de structuration :

- Modèle logique orienté objet
- Modèle logique orienté enregistrement

31

Modèles de structuration des données

Modèle logique orienté objet

- Niveau conceptuel et externe
- Grande flexibilité
 - description des structures
 - explication des contraintes
- Très grand nombre de modèles
 - modèle **Entités-Relations** (E-R)

32

Modèles de structuration des données

Modèle logique orienté enregistrement

- Niveau conceptuel et interne de la BD;
- Se rapproche de la structure physique des données;
- Trois modèles les plus connus :
 - **Modèle relationnel**;
 - Modèle hiérarchique;
 - Modèle réseau.

33

Modèle Entité/Association

- Origine: Peter Chen (1976).
- La méthode permet de distinguer les **entités** qui constituent la base de données, et les **associations** (relations) entre ces entités.
- Ces concepts permettent de donner une structure à la base.

Ces concepts de base sont:

- Entité
- Attribut
- Type d'entité
- Association (relation)
- Type d'association
- Cardinalités

34

Modèle Entité/Association

Entité

Une entité est un objet, une chose concrète ou abstraite qui peut être reconnue distinctement et qui est caractérisée par son unicité.

On peut distinguer deux types d'entités :

- **Entités concrètes** (visible, palpable): pièce mécanique, personne
- **Entités abstraites** (non visible): période de temps, action

Exemple:

L'étudiant: Ben Moh, Matricule 89/201
Cours : Structure de Fichiers à l'ESI

35

Modèle Entité/Association

Attributs

- Propriétés qui compose l'entité ou l'association

Exemple:

Etudiant : Matricule, Nom, Prénom, Adresse
Compte CCP : Numéro, Solde

- Comporte des domaines de valeurs autorisées

Exemple:

Nom : chaîne de 12 caractères
Numéro : ensemble d'entiers positifs

- Types d'attributs:

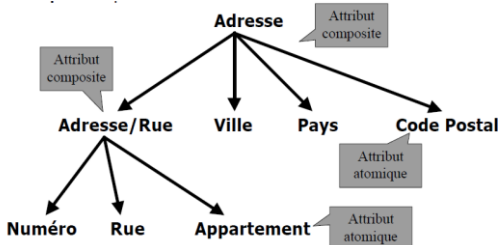
- Valeur simple ou valeurs multiples
- Dérivé
- Simple (atomique) ou composé (composite)

36

Modèle Entité/Association

Attributs

- **Composite**: composé de plusieurs autres attributs de base.
- **Atomique**: pas divisible.



37

Modèle Entité/Association

Attributs

- **Valeur simple**: possède qu'une seule valeur (valeur unique).
Exemple: Âge possède une seule valeur.
- **Valeurs multiples**: possède un ensemble de valeurs pour la même entité
Exemple: Couleur pour une voiture (rouge, verte, bleue, jaune, etc...)
- **Dérivé**: attribut qui est en étroite relation avec un autre attribut.
Exemple: Âge et date de naissance.

38

Modèle Entité/Association

Attributs clés

- Ensemble d'un ou plusieurs attributs qui permet d'identifier de façon unique une entité dans l'ensemble des entités.
Exemple:
Le matricule de l'entité ÉTUDIANT suffit de distinguer un étudiant d'un autre.
- Caractéristiques d'une clé
 - Super clé
 - Clé secondaire ou clé candidate
 - Clé primaire

39

Modèle Entité/Association

Attributs clés

- **Super Clé**: Tout ensemble d'un ou plusieurs attributs qui permet d'identifier de façon unique une entité dans l'ensemble des entités.
Remarque: Chaque entité possède au moins une super clé qui est l'ensemble de tous ses attributs.
- **Clé**: C'est une super clé à laquelle on ne peut plus enlever d'attributs.
Exemple: L'ensemble d'attributs {Matricule, Nom, Age} est une super clé de l'entité ÉTUDIANT, mais pas une clé.
- **Clé Candidate**: Clé possédant les caractéristiques pour être une clé primaire

40

Modèle Entité/Association

Attributs clés

Clé Primaire :

- La clé désignée parmi les clés candidates par le concepteur de la base de données pour identifier de façon unique une entité.
- soulignée dans le schéma de E/A.

Exemple:

ÉTUDIANT{Matricule, Nom, Age}

- Super Clé: {Matricule, Nom, Age} ou {Matricule, Nom}
- Clé: {Matricule}
- Clé candidate: {Matricule}
- Clé primaire: {Matricule}

41

Modèle Entité/Association

Type d'entités

- Ensemble d'entités du même type.

Exemple:

Employé: ensemble des personnes qui travaillent pour une compagnie.

SCHEMA :

EMPLOYÉ
Nom, Age, Salaire

e1.
(John Smith, 55, \$80000)

e2.
(Fred Brown, 40, \$30000)

e3.
(Judy Clark, 25, \$20000)

·
·
·

COMPAGNIE
Nom, Siège Social, Président

c1.
(Sunco Oil, Houston, John Smith)

c2.
(Fast Computer, Dallas, Bob King)

·
·
·

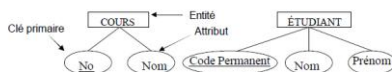
INSTANCES :

42

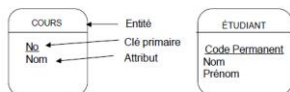
Modèle Entité/Association

- Représentation graphique de l'entité et de l'attribut

1ère méthode:



2ème méthode:



43

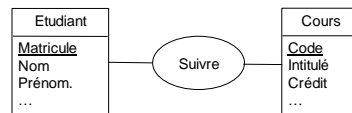
Modèle Entité/Association

Relation

Une relation associe **plusieurs entités**.

Exemple:

L'étudiant Flen « suit » le cours SFSD



44

Modèle Entité/Association

Relation Réflexive

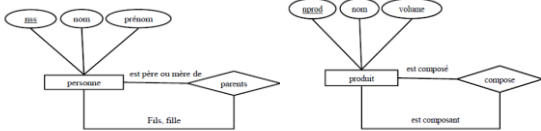
Appelée aussi cyclique ou récursive.

Exemple :

– FILIATION(père: PERSONNE, fils: PERSONNE)

Dans une occurrence d'association une personne peut assumer un des deux rôles père ou fils.

– Compose(composant: PRODUIT, composé: PRODUIT)



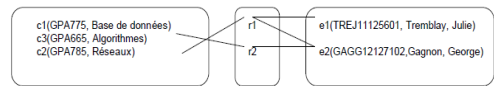
45

Modèle Entité/Association

Type de Relation

Un type de relations est un ensemble de relations qui agissent sur les mêmes types d'entités.

Ici, l'ensemble de relations (r1,r2) représente un type de relation.



46

Modèle Entité/Association

Degré du type relation

Le degré du type de relation est le nombre de types (classes) d'entités participantes.

Exemple:

Le degré du type de relations « Suit » est égale à deux.

- Une relation de **degré deux** est appelé relation **binaire**.
- Une relation de **degré trois** est appelé relation **ternaire**.
- Une relation peut être de n'importe quel degré. En général, elles sont binaires.

47

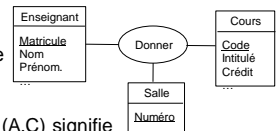
Modèle Entité/Association

Exemple d'une relation ternaire

Cette relation associe trois entités (Enseignant, Cours et Salle), elle signifie qu'un enseignant A donne cours B dans la salle C.

Supposons maintenant que :

- (Enseignant, Cours): l'instance (A,B) signifie l'enseignant A donne le cours B
- (Enseignant, Salle): l'instance (A,C) signifie l'enseignant A enseigne à la salle C
- (Cours, Salle): l'instance (B,C) signifie que le cours B est donné à la salle C
- Les 3 instances (A,C), (A,B) et (C,B) n'implique pas nécessairement que l'instance (A,B,C) existe dans la relation ternaire ABC. (voir explication au tableau)



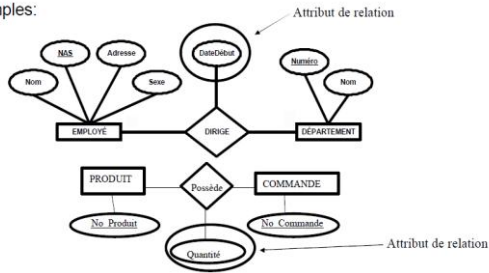
48

Modèle Entité/Association

Attributs d'une relation

Une relation peut posséder des attributs.

Exemples:



49

Modèle Entité/Association

Cardinalité

Exprime le nombre minimum et le nombre maximum de participations de chaque occurrence d'entité à une association (min, max).

50

Modèle Entité/Association

Cardinalité

Les seuls cardinalités admises sont donc :

0,1 : une occurrence du type-entité peut exister tout en étant impliquée dans aucune association et peut être impliquée dans au maximum une association.

0,n : c'est la cardinalité la plus ouverte ; une occurrence du type-entité peut exister tout en étant impliquée dans aucune association et peut être impliquée, sans limitation, dans plusieurs associations.

1,1 : une occurrence du type-entité ne peut exister que si elle est impliquée dans exactement (au moins et au plus) une association.

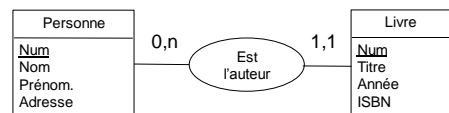
1,n : une occurrence du type-entité ne peut exister que si elle est impliquée dans au moins une association.

51

Modèle Entité/Association

Cardinalité

Exemple:



- Un livre ne peut posséder qu'un seul et unique auteur.

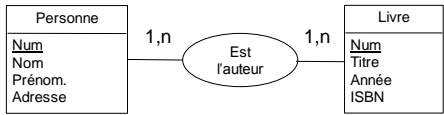
- Une personne peut ne pas être l'auteur d'aucun livre et peut être l'auteur de plusieurs livres

52

Modèle Entité/Association

Cardinalité

Exemple:

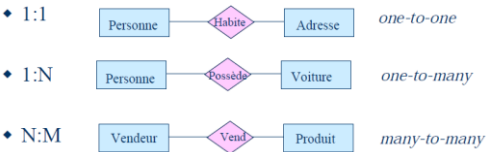


- Un livre possède au minimum un auteur jusqu'à plusieurs auteurs
- Une personne est l'auteur d'au moins un livre jusqu'à plusieurs livres.

53

Modèle Entité/Association

Les ratios de cardinalité

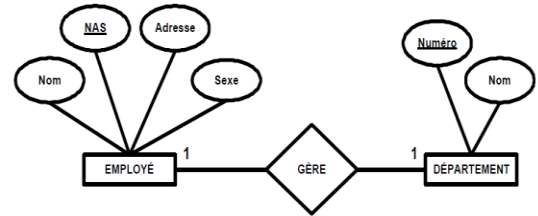


54

Modèle Entité/Association

Les ratios de cardinalité

Relation 1 vers 1

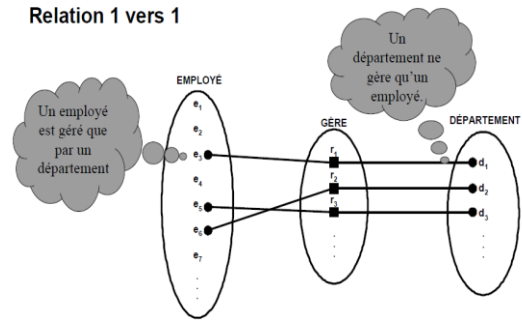


55

Modèle Entité/Association

Les ratios de cardinalité

Relation 1 vers 1

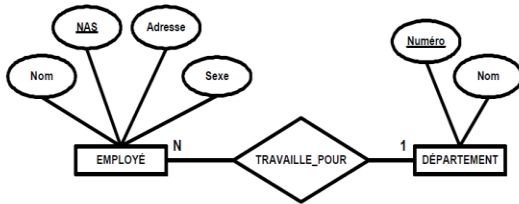


56

Modèle Entité/Association

Les ratios de cardinalité

Relation 1 vers n ou n vers 1



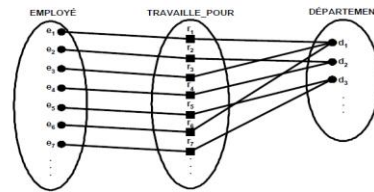
57

Modèle Entité/Association

Les ratios de cardinalité

Relation 1 vers n ou n vers 1

- l'employé doit travailler pour un seul département
- le département peut avoir plusieurs employés.

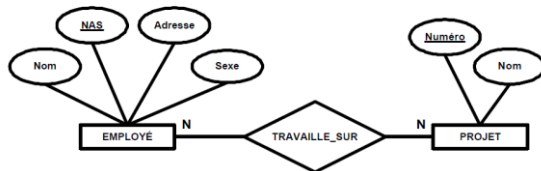


58

Modèle Entité/Association

Les ratios de cardinalité

Relation n vers n



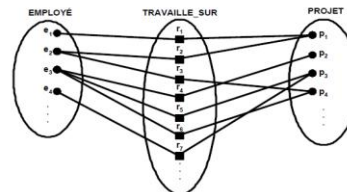
59

Modèle Entité/Association

Les ratios de cardinalité

Relation n vers n

- l'employé peut travailler pour un ou plusieurs projets
- le projet peut avoir un ou plusieurs employés.



60

Modèle Entité/Association

Mise en situation un exemple:

Considérons la base de données d'une université pour maintenir l'information concernant les étudiants et leurs cours.

Chaque **étudiant** **suit** un certain nombre de **cours**.

Un **cours** peut **posséder** certains pré-requis.

Pour chaque cours effectués, un étudiant doit avoir un résultat. (voir au tableau)

61

Modèle Entité/Association

Mise en situation un exemple:

Cette BD pourrait être organisée, en 2 entités et 2 relations, comme suit:

Entité ÉTUDIANT: Information sur chaque étudiant.

Entité COURS: Information sur chaque cours.

Relation RÉSULTAT: Résultat pour les cours des étudiants.

Relation PRÉ-REQUIS: Les pré-requis de chaque cours.

62

Modèle Entité/Association

Mise en situation un exemple:

Pour définir cette BD, il faut spécifier les données (attributs) à stocker dans chaque entité:

ÉTUDIANT: Matricule
 Nom
 Prénom
 Adresse

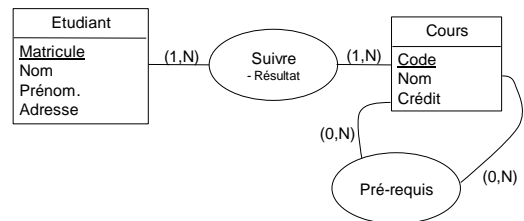
COURS: Code du cours
 Nom du cours
 Crédit

63

Modèle Entité/Association

Mise en situation un exemple:

Le modèle Entité/Association de cette BD est:



64

Modèle Entité/Association

Conclusion sur le modèle E/A

Le modèle E/A est utilisé dans la plupart des méthodes d'analyse/conception : OMT, CASE, MERISE, etc. La syntaxe varie, mais on retrouve toujours les mêmes éléments fondamentaux.

Dans le cadre des bases de données, le modèle E/A est utilisé dans la phase de conception. Il permet de spécifier la structure des informations qui vont être contenues dans la base et d'offrir une représentation abstraite indépendante du modèle logique qui sera choisi ensuite. Le modèle E/A a cependant l'inconvénient majeur de ne pas proposer d'opérations sur les données

65

Modèle Relationnel

1. Introduction
2. Evolution du modèle relationnel
3. Définitions
4. Contraintes relationnelles
5. Opérations sur une BD relationnelles
6. Manipulation des données

66

Modèle Relationnel

1. Introduction

- Développé par Codd en 1970
- Modèle logique orienté enregistrement
- Constitué de tableaux appelés **relations**.
- Simple: la relation est la seule structure du modèle.
- La définition de relation est plus générale qu'au modèle E-R.
- Il définit l'Algèbre Relationnelle sur laquelle est basé SQL (Structured Query Language), le langage standard de manipulation (LMD) et de description des données (LDD) de tous les SGBD Relationnels actuels.

67

Modèle Relationnel

2. Evolution du modèle relationnel

- **Années 1980**
 - Maturation de la technique relationnelle.
 - 1986 : Première Norme SQL (SQL86)
 - 1989 : Standardisation de SQL (SQL89).
- **Années 1990**
 - 1992 : Norme SQL2 (SQL92)
 - Amélioration constante de la technologie relationnelle.
 - Support de la distribution et du parallélisme.
 - 1999 : Le relationnel-objet Norme SQL3 (SQL99)
 - Nouveaux domaines d'application: entrepôts de données et décisionnel, Web, Multimédia, Mobiles, etc.
- **Années 2000**
 - Norme SQL 2003 :
 - SQL Routines and Types Using the Java Programming Language
 - XML-Related Specifications (SQL/XML)
 - Norme SQL 2008

68

Modèle Relationnel

3. Définitions

Dans le modèle relationnel, les données sont stockées dans des *tables* (tableau à deux dimensions).

Chaque colonne de la table est nommée indépendamment de son ordre, qui n'a aucune incidence dans ce modèle. On nomme *attribut* le nom donné à une colonne de la table.

Une ligne de la table contient des valeurs pour chacun des attributs. Chacune de ces lignes est appelée *n-uplet* ou *tuple*.

L'ensemble des n-uplets d'une table se nomme *relation*. Une BD relationnelle est formée d'un ensemble de relations

69

Modèle Relationnel

• Attribut

Les attributs nomment les colonnes d'une relation. Ils servent à la fois à indiquer le contenu de cette colonne, et à la référencer quand on effectue des opérations. Le nom d'un attribut peut apparaître dans plusieurs schémas de relations.

Chaque attribut possède un domaine de valeurs (D_1, D_2, \dots, D_n) .

Exemple:

Les attributs: Matricule, Nom, Prénom, Date de naissance, ... sont des attributs d'étudiant

70

Modèle Relationnel

• Domaine de valeur

Un domaine de valeur est un ensemble d'instances d'un type élémentaire. Exemple : les entiers, les réels, les chaînes de caractères, etc. La notion de 'type élémentaire' s'oppose à celle de type structuré : il est interdit en relationnel de manipuler des valeurs instances de graphes, de listes, d'enregistrements, etc. En d'autres termes le système de types est figé et fourni par le système.

Exemple:

L'ensemble des nombres entiers, l'intervalle de nombres réels $[0;0; 10000;0]$ et l'ensemble {rouge; vert; bleu} constituent des exemples de domaines.

71

Modèle Relationnel

• Schéma de relation

Le schéma d'une relation R est défini par le nom de la relation et la liste des attributs avec pour chaque attribut son domaine :

$R(A_1 : D_1, \dots, A_n : D_n)$ ou $R(A_1, \dots, A_n)$

Exemple:

VEHICULE(Nom:CHAR(20),Type:CHAR(10),Année:ENTIER)

ETUDIANT(Matricule, Nom, Date_naiss, Groupe)

Un schéma de relation définit l'*intention* de la relation

• Schéma de BD

Le schéma d'une base de données est l'ensemble des schémas de ses relations.

72

Modèle Relationnel

- **Relation**

Une relation r de schéma $R(A_1; A_2; \dots; A_n)$ est un sous-ensemble du produit cartésien des domaines de R , c.à.d. $R \subseteq \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$.

On dit aussi que r est une *instance* du schéma de relation R . Une instance de table représente une **extension** de la relation

- **Tuple ou n-uplet**

Un n -uplet correspond à une ligne d'une relation. Il est composé de n éléments :

(v_1, v_2, \dots, v_n) où $v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$

73

Modèle Relationnel

- **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 de n -uplets (ou tuples) $\langle v_1, v_2, \dots, v_n \rangle$ tels que $v_i \in D_i$.

- **Degré d'une relation**

Le degré de la relation est le nombre d'attributs (n) dans son schéma relationnel

Exemple:

Cette relation est de degré 7 :

ÉTUDIANT(Matricule, Nom, Prénom, Date_naiss, Téléphone, Adresse, Promotion)

74

Modèle Relationnel

Exemple

ETUDIANT(Matricule, Nom, Date_naiss, Groupe)

Nom de la relation

Nom des attributs

Table

ETUDIANTS			
Matricule	Nom	Date_naiss	Groupe

Population

Attributs

Tuple ou occurrence

Matricule	Nom	Date_naiss	Groupe
12546	Toto	25/01/81	A1
24323	Tata	23/04/80	C1
43587	Tit	20/09/80	A2

75

Modèle Relationnel

- En résumé:

Concepts	Définition (Terme utilisé)
Relation	Table
Attribut	Colonne
Tuple	Ligne
Domaine	Valeur de la colonne
Schéma de la relation (Intension)	Définition de la table
Extension	Exemples de la table

76

Modèle Relationnel

4. Contraintes relationnelles

- Il existe 4 types de contraintes:
 - de domaines
 - de clés
 - d'intégrité entité
 - d'intégrité de référence
- Elles doivent être vérifiées par chaque instance du schéma.

77

Modèle Relationnel

4. Contraintes relationnelles

4.1 Contraintes de domaines

- La valeur de chaque attribut dans un tuple est atomique (non divisible)
 - Attributs composites ou multivalués non acceptés
- La valeur doit respecter le format des données du domaine de l'attribut (entier, réel, date, caractère).
- Un attribut peut ne pas être renseigné pour un tuple: on dit alors qu'il a une valeur nulle

Exemple : on ne connaît pas la date de naissance de Tata

→ L'attribut n'est pas défini pour le tuple en question

→ La valeur de l'attribut n'est pas encore connue

78

Modèle Relationnel

4. Contraintes relationnelles

4.2 Contraintes de clés

- Chaque tuple dans une relation doit être unique.
- Toute relation doit posséder une clé qui identifie un tuple de façon unique.
- Une relation peut posséder plusieurs clés candidates.
 - A la limite, l'ensemble de tous les attributs constitue une clé.
- La clé choisie est appelé **clé primaire**.
 - Elle est soulignée dans la relation.

Exemple:

ETUDIANT(Nom, Prénom, Matricule, Adresse, DateNaissance)

79

Modèle Relationnel

4. Contraintes relationnelles

4.3 Contraintes d'intégrité entité

- stipulent qu'**aucune clé primaire ne doit être nulle**.

4.4 Contraintes d'intégrité de référence

- contraintes spécifiées entre deux relations et utilisées pour maintenir la consistance entre les tuples de deux relations.
- concept de clé étrangère (Foreign Key).

80

Modèle Relationnel

5. Opérations sur une BD relationnelle

- Langage de définition des données (définition et MAJ du schéma) :
 - création et destruction d'une relation ou d'une base
 - ajout, suppression d'un attribut
 - définition des contraintes (clés, références, ...)
- Langage de manipulation des données
 - saisie des n-uplets d'une relation
 - affichage d'une relation
 - modification d'une relation: insertion, suppression et maj des n-uplets
 - requêtes: consultation d'une ou de plusieurs relations
- Gestion des transactions et Gestion des vues

81

Modèle Relationnel

6. Manipulation des données

Pour manipuler les relations, nous avons besoin d'un langage adapté dont la particularité est de savoir manipuler aisément ces tableaux de données. Ce langage constitue l'**algèbre relationnelle**.

L'algèbre relationnelle est le langage interne d'un SGBD relationnel. Ce langage consiste en un ensemble d'opérations qui permettent de manipuler des relations, considérées comme des ensembles de tuples : on peut ainsi faire l'union ou la différence de deux relations, sélectionner une partie de la relation, effectuer des produits cartésiens ou des projections, etc.

82

Modèle Relationnel

6. Manipulation des données

Une propriété fondamentale de chaque opération est qu'elle prend une ou deux relations en entrée, et produit une relation en sortie. Cette propriété permet de composer des opérations:

on peut appliquer une sélection au résultat d'un produit cartésien, puis une projection au résultat de la sélection et ainsi de suite.

En fait on peut construire des expressions algébriques arbitrairement complexes qui permettent d'exprimer des manipulations sur un grand nombre de relations.

83

Modèle Relationnel

6. Manipulation des données

Une requête est une expression algébrique qui s'applique à un ensemble de relations (la base de données) et produit une relation finale (le résultat de la requête). On peut voir l'algèbre relationnelle comme un langage de programmation très simple qui permet d'exprimer des **requêtes** sur une base de données relationnelle. On parle de langage assertienel car ils permettent de définir les données que l'on souhaite sans dire comment y accéder.

84

Modèle Relationnel

6. Manipulation des données

Deux classes :

- Les langages procéduraux: permettent de définir:
 - l'information
 - une méthode de recherche dans la base
 - la façon d'accéder à l'information

Exemple: algèbre relationnelle

langage opérationnel, une requête s'écrit comme une succession d'opérations effectuées sur des relations

- Les langages non-procéduraux: permettent de définir:
 - l'information désirée, mais le système de BD se charge de la procédure de recherche.

Exemple: calcul relationnel

langage non-procédural permettant d'expliciter le résultat que l'on désire sans spécifier la séquence des opérations à effectuer⁸⁵

Algèbre Relationnelle

L'algèbre relationnelle est le langage interne d'un SGBD relationnel. Elle se compose d'opérateurs de manipulation des relations.

Ces opérateurs sont regroupés en deux familles :

- les opérateurs ensemblistes (union, différence, intersection et produit cartésien)
- et les opérateurs relationnels (sélection, projection, jointure et division).

86

Algèbre Relationnelle

Union U

- Arguments: 2 relations de même schéma :
 $R(A_1, \dots, A_m) \quad S(A_1, \dots, A_m)$
- Notation: $R \cup S$
- Schéma de $T = R \cup S$: $T(A_1, \dots, A_m)$
- Valeur de T : Union ensembliste sur $D_1 \times \dots \times D_m$:
 $T = \{t \mid t \in R \vee t \in S\}$

87

Algèbre Relationnelle

Union U

règles :

- R et S doivent présenter le même nombre d'attributs
- les domaines du $i^{\text{ème}}$ attribut de R et du $i^{\text{ème}}$ attribut de S doivent être identiques
- ne crée pas de doublons (tuples identiques)

Exemple:

R	<table><tr><th>A</th><th>B</th></tr><tr><td>a</td><td>b</td></tr><tr><td>a</td><td>c</td></tr><tr><td>d</td><td>e</td></tr></table>	A	B	a	b	a	c	d	e				
A	B												
a	b												
a	c												
d	e												
S	<table><tr><th>A</th><th>B</th></tr><tr><td>a</td><td>b</td></tr><tr><td>a</td><td>e</td></tr><tr><td>d</td><td>e</td></tr><tr><td>f</td><td>g</td></tr></table>	A	B	a	b	a	e	d	e	f	g		
A	B												
a	b												
a	e												
d	e												
f	g												
\Rightarrow	$R \cup S$												
	<table><tr><th>A</th><th>B</th></tr><tr><td>a</td><td>b</td></tr><tr><td>a</td><td>c</td></tr><tr><td>d</td><td>e</td></tr><tr><td>a</td><td>e</td></tr><tr><td>f</td><td>g</td></tr></table>	A	B	a	b	a	c	d	e	a	e	f	g
A	B												
a	b												
a	c												
d	e												
a	e												
f	g												

88

Algèbre Relationnelle

Différence -

- Arguments : 2 relations de même schéma :
 $R(A_1, \dots, A_m) \quad S(A_1, \dots, A_m)$
- Notation: $R - S$
- Schéma de $T = R - S$: $T(A_1, \dots, A_m)$
- Valeur de T : Différence ensembliste sur $D_1 \times \dots \times D_m$:
 $T = \{t \mid t \in R \wedge t \notin S\}$

La différence signifie: on veut 'rejeter' quelque chose, on 'ne veut pas' des lignes ayant telle propriété.

89

Algèbre Relationnelle

Différence -

Exemple:

R

A	B
a	b
a	c
d	e

S

A	B
a	b
a	e
d	e
f	g

R - S

A	B
a	c

S - R

A	B
a	e
f	g

90

Algèbre Relationnelle

Intersection \cap

- Arguments: 2 relations de même schéma :
 $R(A_1, \dots, A_m) \quad S(A_1, \dots, A_m)$
- Notation: $R \cap S$
- Schéma de $T = R \cap S$: $T(A_1, \dots, A_m)$
- Valeur de T : Intersection ensembliste sur $D_1 \times \dots \times D_m$:
 $T = \{t \mid t \in R \wedge t \in S\}$

Cette opération peut être dérivée de l'opération différence: $R \cap S = R - (R - S) = S - (S - R)$

91

Algèbre Relationnelle

Intersection \cap

$$R \cap S = R - (R - S) = S - (S - R)$$

Exemple:

R

A	B
a	b
a	c
d	e

S

A	B
a	b
a	e
d	e
f	g

R - S

A	B
a	c

$R \cap S = R - (R - S)$

A	B
a	b
d	e

92

Algèbre Relationnelle

Produit cartésien \times

- Arguments : 2 relations quelconques :
 $R(A_1, \dots, A_n) \quad S(B_1, \dots, B_k)$
- Notation: $R \times S$
- Schéma de $T = R \times S : T(A_1, \dots, A_n, B_1, \dots, B_k)$.
- On introduit les règles de renommage suivantes pour lever les éventuelles ambiguïtés sur le schéma de T:
 - Si le produit cartésien est le produit d'une relation avec elle-même alors le nom de la relation est numéroté pour identifier les deux rôles (par 1 et 2).
 - Si les relations ont des attributs en commun, les noms des attributs en commun sont préfixés par le nom de la relation d'origine.

93

Algèbre Relationnelle

Produit cartésien \times

- Valeur de $T = R \times S$: ensemble de tous les n-uplets ayant $n + k$ composants (attributs)
 - dont les n premiers composants forment un n-uplet de R
 - et les k derniers composants forment un n-uplet de S .

94

Algèbre Relationnelle

Produit cartésien \times

Exemple: R

A	B
1	1
1	2
3	4

$|R|$

S

C	D	E
a	b	a
a	b	c
b	a	a

$|S|$

$R \times S$

A	B	C	D	E
1	1	a	b	a
1	1	a	b	c
1	1	b	a	a
1	2	a	b	a
1	2	a	b	c
1	2	b	a	a
3	4	a	b	a
3	4	a	b	c
3	4	b	a	a

$|R \times S|$

95

Algèbre Relationnelle

Sélection/restriction σ

- Sélection est un opérateur unaire
- Sélection avec une condition C sur les attributs d'une relation R : on garde les n-uplets de R dont les attributs satisfont la condition C .
- Notation: $\sigma_C(R)$
- La condition C d'une sélection $\sigma_C(R)$ est une formule logique qui relie des termes de la forme $(A_i \theta A_j)$ ou $(A_i \theta a)$ avec les connecteurs logiques $\text{et } (\wedge)$ et $\text{ou } (\vee)$ où:
 - A_i et A_j sont des attributs de la relation R ,
 - a est un élément (une valeur) du domaine de A_i ,
 - θ est un prédicat de comparaison $(=, <, \leq, >, \geq, \neq)$.

96

Algèbre Relationnelle

Sélection/restriction σ

Exemple:

- a) On sélectionne les n-uplets dans la relation R tels que l'attribut B vaut "b" :

R	A	B	C
a	b	1	
d	a	2	
c	b	3	
a	b	4	
e	e	5	

 $\Rightarrow \sigma_{B="b"}(R)$

A	B	C
a	b	1
c	b	3
a	b	4

97

Algèbre Relationnelle

Sélection/restriction σ

Exemple:

- b) On sélectionne les n-uplets tels que

$$(A = "a" \vee B = "a") \wedge C \leq 3 :$$

R	A	B	C
a	b	1	
d	a	2	
c	b	3	
a	b	4	
e	e	5	

 $\Rightarrow \sigma_{(A="a" \vee B="a") \wedge C \leq 3}(R)$

A	B	C
a	b	1
d	a	2

98

Algèbre Relationnelle

Sélection/restriction σ

Exemple:

- c) On sélectionne les n-uplets tels que la 1^{re} et la 2^e colonne sont identiques :

R	A	B	C
a	b	1	
d	a	2	
c	b	3	
a	b	4	
e	e	5	

 $\Rightarrow \sigma_{A=B}(R)$

A	B	C
e	e	5

99

Algèbre Relationnelle

Projection π

- Projection sur une partie (un sous-ensemble) des attributs d'une relation R.

- Notation : $\pi_{A_1, \dots, A_k}(R)$

A_1, \dots, A_k sont des attributs (du schéma) de la relation R. La projection "élimine" tous les autres attributs (colonnes) de R.

100

Algèbre Relationnelle

Projection π

Exemple:

a) On élimine la colonne C dans la relation R :

R	A	B	C
→	a	b	c
	d	a	b
	c	b	d
→	a	b	e
	e	e	a

 \Rightarrow

$\pi_{A,B}(R)$	A	B
	a	b
	d	a
	c	b
	e	e

Le résultat est une relation (un ensemble) : le n-uplet (a,b) n'apparaît qu'une seule fois dans la relation $\pi_{A,B}(R)$, bien qu'il existe deux n-uplets (a,b,c) et (a,b,e) dans R.

10

Algèbre Relationnelle

Projection π

Exemple:

b) On élimine la colonne B dans la relation R (on garde A et C) :

R	A	B	C
	a	b	c
	d	a	b
	c	b	d
	a	b	e
	e	e	a

 \Rightarrow

$\pi_{A,C}(R)$	A	C
	a	c
	d	b
	c	d
	a	e
	e	a

10

Algèbre Relationnelle

Jointures

1. Jointure naturelle \bowtie

- Notation: $R \bowtie S$
- Arguments: 2 relations quelconques :
 $R(A_1, \dots, A_m, X_1, \dots, X_k)$
 $S(B_1, \dots, B_n, X_1, \dots, X_k)$
 où X_1, \dots, X_k sont les attributs en commun.

- Schéma de $T = R \bowtie S$: $T(A_1, \dots, A_m, B_1, \dots, B_n, X_1, \dots, X_k)$
- Valeur de $T = R \bowtie S$: ensemble de tous les n-uplets ayant $m+n+k$ attributs dont les m premiers et k derniers composants forment un n-uplet de R et les $n+k$ derniers composants forment un n-uplet de S.

10

Algèbre Relationnelle

Jointures

1. Jointure naturelle \bowtie

Exemple:

R	A	B	C
	a	b	c
	d	b	c
	b	b	f
	c	a	d

S	B	C	D
	b	c	d
	b	c	e
	a	d	b
	a	c	c

 $R \bowtie S$

	A	B	C	D
	a	b	c	d
	a	b	c	e
	d	b	c	d
	d	b	c	e
	c	a	d	b

10

Algèbre Relationnelle

Jointures

1. Jointure naturelle \bowtie

Soit $U = \{A_1, \dots, A_m, B_1, \dots, B_n, X_1, \dots, X_k\}$ l'ensemble des attributs des 2 relations et $V = \{X_1, \dots, X_k\}$ l'ensemble des attributs en commun

$$R \bowtie S = \pi_U(\sigma_{\forall X \in V: R.X=S.X}(R \times S))$$

Notation : $R.X$ signifie "l'attribut X de la relation R".

10

Algèbre Relationnelle

Jointures

1. Jointure naturelle \bowtie

Exemple: $R \bowtie S = \pi_U(\sigma_{\forall X \in V: R.X=S.X}(R \times S))$

R	A	B	S	A	B	D		\Rightarrow	$R.A=S.A \wedge R.B=S.B (R \times S))$
	1	a		1	a	b			
	1	b		2	c	b			
	4	a		4	a	a			

R × S	R.A	R.B	S.A	S.B	D				R × S	A	B	D
	1	a	1	a	b					1	a	b
$S.A \wedge R.B \neq S.B \rightarrow$	1	a	2	c	b					4	a	a
$R.A \neq S.A \rightarrow$	1	a	4	a	a							
$R.B \neq S.B \rightarrow$	1	b	1	a	b							
$S.A \wedge R.B \neq S.B \rightarrow$	1	b	2	c	b							
$S.A \wedge R.B \neq S.B \rightarrow$	1	b	4	a	a							
$R.A \neq S.A \rightarrow$	4	a	1	a	b							
$R.A \neq S.A \wedge R.B \neq S.B \rightarrow$	4	a	2	c	b							
	4	a	4	a	a							

10

Algèbre Relationnelle

Jointures

1. Jointure naturelle \bowtie

L'algorithme général:

Pour chaque n -uplet a dans R et pour chaque n -uplet b dans S :

- on concatène a et b et on obtient un n -uplet qui a pour attributs

$$\overbrace{A_1, \dots, A_m, X_1, \dots, X_k}^a \quad \overbrace{B_1, \dots, B_n, X_1, \dots, X_k}^b$$

- on ne le garde que si chaque attribut X_i de a est égal à l'attribut X_i de b : $\forall_{i=1..k} a.X_i = b.X_i$.
- on élimine les valeurs (colonnes) dupliquées : on obtient un n -uplet qui a pour attributs

$$\overbrace{A_1, \dots, A_m}^a \quad \overbrace{B_1, \dots, B_n}^b \quad \overbrace{X_1, \dots, X_k}^{a \text{ et } b}$$

10

Algèbre Relationnelle

Jointures

2. Théta jointure \bowtie_θ

- Arguments: deux relations qui ne partagent pas d'attributs :

$$R(A_1, \dots, A_m) \quad S(B_1, \dots, B_n)$$

- Notation: $R \bowtie_{A_i \theta B_j} S$, $\theta \in \{=, \neq, <, \leq, >, \geq\}$

- Schéma de $T = R \bowtie_{A_i \theta B_j} S$: $T(A_1, \dots, A_m, B_1, \dots, B_n)$

- Valeur de $T = R \bowtie_{A_i \theta B_j} S$: $T = \sigma_{A_i \theta B_j} (R \times S)$

- Équijointure : θ est l'égalité

10

Algèbre Relationnelle

Jointures

2. Théta jointure \bowtie

Exemple Théta jointure: $R \bowtie_{A \leq C} S \rightarrow \sigma_{A \leq C} (R \times S)$

R	A	B
1	a	
1	b	
3	a	

S	C	D	E
1	b	a	
2	b	c	
4	a	a	

$T := R \times S$

A	B	C	D	E
1	a	1	b	a
1	a	2	b	c
1	a	4	a	a
1	b	1	b	a
1	b	2	b	c
1	b	4	a	a
3	a	1	b	a
3	a	2	b	c
3	a	4	a	a

\Rightarrow

$\sigma_{A \leq C}(T)$
 $= R \bowtie_{A \leq C} S$

A	B	C	D	E
1	a	1	b	a
1	a	2	b	c
1	a	4	a	a
1	b	1	b	a
1	b	2	b	c
1	b	4	a	a
3	a	4	a	a

$A > C \rightarrow$
 $A > C \rightarrow$

10

Algèbre Relationnelle

Jointures

2. Théta jointure \bowtie

Exemple Equijointure: $R \bowtie_{A=D} S \rightarrow \sigma_{B=D} (R \times S)$

R	A	B
1	a	
1	b	
3	a	

S	C	D	E
1	b	a	
2	b	c	
4	a	a	

$T := R \times S$

$B \neq D \rightarrow$
 $B \neq D \rightarrow$

A	B	C	D	E
1	a	1	b	a
1	a	2	b	c
1	a	4	a	a
1	b	1	b	a
1	b	2	b	c
1	b	4	a	a
3	a	1	b	a
3	a	2	b	c
3	a	4	a	a

$B \neq D \rightarrow$
 $B \neq D \rightarrow$
 $B \neq D \rightarrow$

\Rightarrow

$\sigma_{B=D}(T)$
 $= R \bowtie_{B=D} S$

A	B	C	D	E
1	a	4	a	a
1	b	1	b	a
1	b	2	b	c
3	a	4	a	a

11

Algèbre Relationnelle

Renommage

- Notation: ρ
- Arguments : 1 relation : $R(A_1, \dots, A_n)$
- Schéma de $T = \rho_{A_i \rightarrow B_i} R : T(A_1, \dots, A_{i-1}, B_i, A_{i+1}, \dots, A_n)$
- Valeur de $T = \rho_{A_i \rightarrow B_i} R : T = R$. La valeur de R est inchangée.

Seul le nom de l'attribut A_i a été remplacé par B_i

11

Algèbre Relationnelle

Division \div

- Arguments: 2 relations :
 $R(A_1, \dots, A_m, X_1, \dots, X_k)$
 $S(X_1, \dots, X_k)$
où tous les attributs de S sont des attributs de R .
 - Notation: $R \div S$
 - Schéma de $T = R \div S : T(A_1, \dots, A_m)$
 - Valeur de $T = R \div S$:
 $R \div S = \{(a_1, \dots, a_m) \mid \forall (x_1, \dots, x_k) \in S : (a_1, \dots, a_m, x_1, \dots, x_k) \in R\}$
- pour qu'un tuple t apparaisse dans le résultat T de la division, les valeurs dans t doivent apparaître en combinaison avec tous les tuples de S .

11

Algèbre Relationnelle

Division ÷

- La division s'exprime en fonction du produit cartésien, de la projection et de la différence :

$$T = R \div S = R_1 - R_2 \quad \text{où}$$

$$R_1 = \pi_{A_1, \dots, A_m}(R) \text{ et } R_2 = \pi_{A_1, \dots, A_m}((R_1 \times S) - R)$$

Remarque : le produit cartésien $T \times S$ doit être contenu dans la table R.

113

Algèbre Relationnelle

Division ÷

Exemple: $R \div S = R_1 - R_2$ avec

$$R_1 = \pi_{A_1, \dots, A_m}(R) \text{ et } R_2 = \pi_{A_1, \dots, A_m}((R_1 \times S) - R)$$

R

A	B	C	D
a	b	x	m
a	b	y	n
a	b	z	o
b	c	x	o
b	d	x	m
c	e	x	m
c	e	y	n
c	e	z	o
d	a	z	p
d	a	y	m

$R_1 = \pi_{A_1, \dots, A_m}(R)$ et $R_2 = \pi_{A_1, \dots, A_m}((R_1 \times S) - R)$

S

C	D
x	m
y	n
z	o

R : S

A	B
a	b
c	e

114

Algèbre Relationnelle

- La sélection a pour effet de supprimer des lignes, mais chaque ligne garde l'ensemble de ses attributs.

La différence «on veut 'rejeter' quelque chose, on 'ne veut pas' des lignes ayant telle propriété)

115

Algèbre Relationnelle

En résumé, cinq de ces huit opérateurs forment les opérateurs de base (ce sont l'union, la différence, le produit cartésien, la restriction et la projection) tandis que les trois autres, appelés opérateurs dérivés, s'obtiennent plus ou moins facilement par combinaison des opérateurs de base :

- $R \cap S = R \setminus (R \setminus S) = (R \cup S) \setminus ((R \setminus S) \cup (S \setminus R))$
- $R \bowtie_P S = \sigma_P(R \times S)$
- $R \div S = \pi_Y(R) \setminus \pi_Y((S \times \pi_Y(R) \setminus R))$

Les cinq opérateurs de base permettent de répondre à toutes les questions que l'on peut poser avec la logique du premier ordre (c'est-à-dire sans les fonctions) : on dit que **l'algèbre relationnelle est complète**.

116

Algèbre Relationnelle

En réalité, nous n'utiliserons dans nos requêtes que les opérateurs les plus maniables : ce sont l'union et la différence pour l'insertion et la suppression de tuples dans la base et la restriction, la projection et la jointure pour la recherche sélective de tuples.

Les opérateurs de l'algèbre relationnelle ne présentent pas seulement un intérêt sur le plan théorique. Leur portée pratique est aussi importante. par exemple, nous en aurons besoin pour optimiser les requêtes au niveau du langage des systèmes de bases de données relationnelles

117

Algèbre Relationnelle

En outre, ils trouvent leur application dans la conception des ordinateurs de base de données : les opérateurs de l'algèbre relationnelle et leurs formes dérivées n'y sont pas mis en œuvre sous forme logicielle, mais implantées directement dans des composants matériels de l'ordinateur.

118

Algèbre Relationnelle

Exemple1:

Soit les trois tables suivantes :

- CLIENTS(ncit, nom, age, adresse)
- PRODS(nprod, design, couleur, volume)
- CMDS(ncit, nprod, qte, date).

- 1.La liste des noms de clients qui ont un age > 20.
- 2.La liste des noms de clients ayant commandés le produit numéro 13.
- 3.La liste des noms de clients ayant commandés un produit de couleur rouge.
- 4.La couleur des produits commandés par monsieur Dupont.
- 5.La liste des noms de clients ayant commandés au moins un produit.
- 6.La liste des noms de clients ayant commandés un produit vert ou rouge.

119

Algèbre Relationnelle

Exemple1:

- 7.La liste des noms de clients ayant commandés un produit vert ou bien rouge.
- 8.La liste des noms de clients ayant commandés au moins deux produits.
- 9.La liste des clients qui ont un age > 50 et qui n'ont pas commandé un produit vert.
- 10.La liste des noms de clients qui ont commandé tous nos produits.
- 11.La liste des noms de clients qui ont commandé tous nos types de pince.
- 12.Le Nombre de clients habitant à paris.
- 13.Le ncit et le nom du ou des clients les plus jeunes de la table client

120

Algèbre Relationnelle

Exemple2 de travail:

ETUDIANT (Num_Etudiant *integer*, Nom *char(20)*,
Adresse *varchar(50)*)

COURS (Num_Cours *integer*, Nom *char(20)*)

PROFS (Num_Prof *integer*, Nom *char(20)*, Adresse
varchar(50))

COURS_SUIVI (Num_Etudiant *integer*, Num_Cours
integer)

COURS_ENSEIGNES (Num_Prof *integer*, Num_Cours
integer)

12