Bases de données

Souhila KACI

Partie 1

Plan du cours

- Introduction aux bases de données
- Modèle Relationnel de données
- Langages de manipulation des données

Bases de données (BD) : Introduction

- Collection d'informations ou de données existantes sur une longue durée et décrivant les activités d'une ou plusieurs organisations.
- Ensemble de données modélisant les objets d'une partie du monde réel et servant de support à une application informatique.
- Un ensemble d'informations structurées et mémorisées sur un support permanent.
- Cet ensemble peut être interrogé et mis à jour par des utilisateurs.

Exemples de cadres d'utilisation de BD

- Gestion d'entreprises : stocks, personnel, clients
- Gestion bancaire : comptes, emprunts
- Systèmes de réservation : avions, trains, spectacles
- Bibliothèques : ouvrages, emprunteurs, prêts

Utilisation d'une base de données pour toute application nécessitant la structuration, le stockage, la manipulation et l'interrogation d'un ensemble conséquent d'informations.

Système de Gestion de Bases de Données (SGBD) (1)

Un SGBD est un logiciel intermédiaire entre la base de données et l'utilisateur (un humain ou un programme).
Un système de gestion de bases de données met à la disposition de l'utilisateur un outil pour

- la création et l'administration d'une BD,
- la sauvegarde (stockage) des données,
- la manipulation (insertion, modification, suppression, interrogation) des données d'une manière efficace.

Système de Gestion de Bases de Données (SGBD) (2)

Un SGBD permet de satisfaire les propriétés suivantes :

- Indépendance.
- Non redondance des données.
- Maintien de la cohérence des données (contraintes d'intégrité).
- Concurrence d'accès.

Univers réel

Modèle conceptuel de données : Modèles sémantiques orientés "Conception" (exemple : Entité-Association)

Modèle logique : Hiérarchique (structure de données “arbre”), Réseau (structure de données “graphe”), **Relationnel (structure de données “tableau de n-uplets”)**, Objet (structure de données “classes, attributs, méthodes”)

Le Modèle Relationnel de Données

Les concepts de base: Les domaines

- **Ensemble de valeurs caractérisé par un nom.**
- Ce sont les ensembles dans lesquels les données prennent leur valeur.
- Sa définition peut être.
  - en extension, en donnant la liste des valeurs composante,
  - en intention, en définissant une propriété caractéristique des valeurs du domaine.
- Il peut être fini (chaîne de moins de 20 caractères) ou infini (les entiers naturels).

Exemple

- les entiers naturels (intention)
- La monnaie : réel avec deux chiffres derrière la virgule (intention)
- `CouleurVin = {'Blanc','Rouge','Rosé'}` (extension)
- `Crus = {'Aix','Sancerre'}` (extension)

Rappel: Produit cartésien

Le produit cartésien d'un ensemble de domaines $D_1, D_2, \ldots, D_n$ est l'ensemble des vecteurs $\langle v_1, v_2, \ldots, v_n \rangle$, où $v_i \in D_i$.

Exemple

Le produit cartésien de
`CouleurVin = {'Rouge','Blanc','Rosé'}` et de
`Crus = {'Aix','Sancerre'}`

'Rouge'	'Aix'
'Rouge'	'Sancerre'
'Blanc'	'Aix'
'Blanc'	'Sancerre'
'Rosé'	'Aix'
'Rosé'	'Sancerre'

`CouleurVin` $\times$ `Crus`

Relation

- Concept central du modèle.
- Sous-ensemble d'un produit cartésien.
- **Une relation est donc un ensemble de n-uplets (ou tuples) de la forme $\langle a_1, a_2, \ldots, a_n \rangle$, avec $a_i \in A_i$.**
- Représentation commode : table à deux dimensions.
- Chaque ligne correspond à un vecteur (un tuple).
- Chaque colonne correspond à un domaine.

Exemple

CouleurCrus	Couleur	Cru
	'Rouge'	'Aix'
	'Blanc'	'Sancerre'
(Tuples) →	'Rosé'	'Aix'
	'Rosé'	'Sancerre'

↑

(Domaine)

Attribut

- **Colonne d'une relation caractérisée par un nom.**
- Porteur de sens : Souvent différent du domaine qui supporte l'attribut.

Exemple

- La relation précédente comportait deux attributs Couleur et Cru de même nom que leur domaine associés.
- Le premier attribut peut avantageusement être remplacé par le nom Type.
- Il faut pouvoir distinguer deux colonnes qui prendraient leurs valeurs dans le même domaine.
- Le choix du nom d'un attribut est aussi important que le choix du nom d'une fonction lorsque l'on programme.

Schéma d'une relation

- **Nom de la relation suivi de la liste des attributs et de la définition de leur domaine.**
- La structure d'une relation caractérisée par les concepts de domaines et d'attributs est un invariant pour une relation (elle ne change pas en fonction du temps).

Exemple

La relation précédente avait comme schéma de relation
R(Type : {'Rouge','Rosé','Blanc'},Cru : {'Aix','Sancerre'})

Définitions complémentaires

- ARITE d'une relation: Nombre fixe de ses attributs (colonnes).
- CARDINALITE d'une relation: Nombre de n-uplets (lignes).
- SCHEMA de base de données relationnelles : ensemble de schéma de relations.

Contraintes d'intégrité (1) : Clé primaire

Des contraintes sur le schéma de la BD (ensemble des schémas des relations) permettant en partie de garantir la cohérence de la BD.

- Toute relation possède une **clé primaire** : un ensemble minimal d'attributs dont les valeurs permettent de distinguer deux tuples de la relation.
- Par convention, on souligne les attributs participant aux clés primaires.
  - **Voiture**(<u>Immatriculation</u>,Couleur,NomModele)
  - **Modele**(<u>NomModele</u>,Marque)
  - **Commande_Produit**(<u>NumCmd,RefProduit</u>,QtéCom)

Clé primaire

- Les valeurs de la clé primaire permettent d'identifier de manière unique un tuple de la table.

- Impossible d'avoir deux tuples avec les mêmes valeurs pour les attributs de la clé primaire.

Voiture(<u>Immatriculation</u>,Couleur,NomModele)

'123XY34'	'Jaune'	'106'
'34UV62'	'Verte'	'106'
'345RT62'	'Verte'	'Megane'
~~'123XY34'~~	~~'Verte'~~	~~'106'~~
'234XU45'	'Bleue'	'Clio'

Contraintes d'intégrité (2) : Clé étrangère

- Clé étrangère : un attribut dont les valeurs appartiennent à l'ensemble des valeurs d'une clé primaire d'une autre relation.
- Une relation peut posséder une ou plusieurs clés étrangères.
- Par convention, on souligne en pointillés les clés étrangères.

Modele(<u>NomModele</u>,Marque)

Voiture(<u>Immatriculation</u>,Couleur,NomModele)

Ici, nous avons une clé étrangère entre les tables Modele et Voiture : l'attribut NomModele de la table Voiture **référence** l'attribut NomModele de la table Modele.

On parle de contrainte d'intégrité référentielle.

Clé étrangère

Modele

NomModele	Marque
'106'	'Peugeot'
'206'	'Peugeot'
'306'	'Peugeot'
'Clio'	'Renault'
'Espace'	'Renault'

Voiture

Immatriculation	Couleur	NomModele
'123XY34'	'Jaune'	'106'
'34UV62'	'Verte'	'106'
~~'345RT62'~~	~~'Verte'~~	~~'Megane'~~
'123XY36'	'Verte'	'106'
'234XU45'	'Bleue'	'Clio'

L'algèbre relationnelle

L'algèbre relationnelle

- L'algèbre relationnelle est une théorie mathématique dans laquelle sont définies les différentes opérations permettant de manipuler les relations.
- Une expression algébrique permet d'exprimer une requête.
- La valeur d'une expression algébrique correspond au résultat de l'évaluation de la requête.

Opérateurs

Les opérateurs ensemblistes traditionnels

- Opérateurs binaires: à partir de 2 relations, ils en génèrent une nouvelle.

➜ Union, Différence, Produit Cartésien

Les opérateurs relationnels

- Répondent à certains besoins spécifiques de l'algèbre relationnelle.
- Ils génèrent également une nouvelle relation.

➜ Projection, Restriction, Jointure

Opérateurs. . .

Opérateurs dérivés

- Opérateurs pouvant étre obtenus par combinaison des opérateurs précédents.
- Utilisés pour plus de commodité.

➜ Intersection, division. . .

Union

Définition

Opération portant sur deux relations de même schéma R et S construisant une relation de même schéma ayant pour tuples ceux appartenant à R ou à S.

Notations

$R \cup S$, $Union(R, S)$, $Append(R, S)$

Lister les n-uplets qui appartiennent à cette relation **OU** à cette autre relation.

Exemple

Deux relations `Vins1` et `Vins2` de même schéma

`(Crus,Mill,Région,Type)`

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Vins2

'Bandol'	1988	'Provence'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Cassis'	2000	'Provence'	'Blanc'
'Aix'	1996	'Provence'	'Rouge'

Vins = Vins1 $\cup$ Vins2

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'
'Bandol'	1988	'Provence'	'Rouge'
'Cassis'	2000	'Provence'	'Blanc'
'Aix'	1996	'Provence'	'Rouge'

- $R \cup S = S \cup R$
- $Card(R \cup S) \leq Card(R) + Card(S)$
- $Card(R \cup S) = Card(R) + Card(S) - Card(R \cap S)$
- $Arite(R \cup S) = Arite(R) = Arite(S)$

Définition

Opération portant sur deux relations de même schéma R et S construisant une relation de même schéma ayant pour tuples ceux appartenant à R mais pas à S.

Notations

$R - S$, $Diff(R, S)$, $Minus(R, S)$

Lister les n-uplets qui appartiennent à cette relation **MAIS PAS** à cette autre relation.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Vins2

'Bandol'	1988	'Provence'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Cassis'	2000	'Provence'	'Blanc'
'Aix'	1996	'Provence'	'Rouge'

Vins = Vins1 - Vins2

'Chenas'	1983	'Beaujolais'	'Rouge'
'Aix'	1995	'Provence'	'Rouge'

Propriétés

- $R - S \neq S - R$
- $Card(R - S) \leq Card(R)$
- $Card(R - S) = Card(R) - Card(R \cap S)$
- $Arite(R - S) = Arite(R) = Arite(S)$

Produit cartésien

Définition

Opération portant sur deux relations de schéma R et S $(Attr(R) \cap Attr(S) = \emptyset)$ construisant une relation ayant pour schéma la concaténation de R et de S et pour tuples toutes les combinaisons de tuples possibles.

Notations

$R \times S$, $Prod(R, S)$

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Année

1988
2000

Vins = Vins1 $\times$ Année

'Chenas'	1983	'Beaujolais'	'Rouge'	1988
'Tokay'	1980	'Alsace'	'Blanc'	1988
'Aix'	1995	'Provence'	'Rouge'	1988
'Chenas'	1983	'Beaujolais'	'Rouge'	2000
'Tokay'	1980	'Alsace'	'Blanc'	2000
'Aix'	1995	'Provence'	'Rouge'	2000

Propriétés

- $R \times S = S \times R$
- $Card(R \times S) = Card(R) \times Card(S)$
- $Arite(R \times S) = Arite(R) + Arite(S)$

Projection

- Opération spécifique aux relations.
- Permet de supprimer des attributs d'une relation.

Définition

Opération portant sur une relation R construisant une relation contenant uniquement les attributs mentionnés en opérande et éliminant les tuples en double.

Notations

- $\Pi_{Att_i, Att_j \ldots}(R)$
- $Project(R, Att_i, Att_j, \ldots)$
- $R[Att_i, Att_j, \ldots]$

Lister **UNIQUEMENT** certaines colonnes de cette relation.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'
'Aix'	2000	'Provence'	'Rouge'
'Cassis'	2000	'Provence'	'Blanc'

$$\text{Vins} = \Pi_{Cru, Region}(\text{Vins1})$$

'Chenas'	'Beaujolais'
'Tokay'	'Alsace'
'Aix'	'Provence'
'Cassis'	'Provence'

Propriétés

- $Card(Proj(R, A_1, A_2, \ldots)) \leq Card(R)$
- $Arite(Proj(R, A_1, A_2, \ldots)) \leq Arite(R)$

Restriction

- C'est également une opération spécifique. Elle sélectionne les tuples suivant un critère.

Définition

Opération portant sur une relation R construisant une relation de même schéma mais comportant uniquement les tuples qui vérifient la condition précisée en argument.

Notations

- $\sigma_{cond}(R)$
- $Select(R, cond)$
- $Restrict(R, cond)$

Lister **UNIQUEMENT** les nuplets **VERIFIANT** cette condition.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'
'Aix'	2000	'Provence'	'Rouge'
'Cassis'	2000	'Provence'	'Blanc'

$$\text{Vins} = \sigma_{Cru=Aix}(\text{Vins1})$$

'Aix'	1995	'Provence'	'Rouge'
'Aix'	2000	'Provence'	'Rouge'

Propriétés

- $Card(\sigma_{cond}(R)) \leq Card(R)$
- $Arite(\sigma_{cond}(R)) = Arite(R)$

- On dit que l'algèbre relationnelle est un langage fermé car chaque opération retourne une relation.
- On peut donc appliquer un opérateur de l'algèbre au résultat d'une autre opération.

Exemple

TouteRégion = Π(Vins1 $\cup$ Vins2,Région)

'Provence'
'Beaujolais'
'Alsace'

Jointure

- Opérateur essentiel à l'algèbre relationnelle.
- Compose deux relations à l'aide d'un critère de jointure.
- Peut être vue comme une extension du produit cartésien avec une condition permettant de comparer les attributs.

Définition

Opération portant sur deux relations R et S et construisant une relation qui contient la concaténation des tuples de R et de S vérifiant la condition de rapprochement.

Notations

$Join(R, S, cond)$, $R \bowtie_{cond} S$

Exemple

Vins1

'Chenas'	'Beaujolais'	'Rouge'
'Tokay'	'Alsace'	'Blanc'
'Aix'	'Provence'	'Rouge'

Année

1988
2000

V = Vins1 $\times$ Année

'Chenas'	'Beaujolais'	'Rouge'	1988
'Tokay'	'Alsace'	'Blanc'	1988
'Aix'	'Provence'	'Rouge'	1988
'Chenas'	'Beaujolais'	'Rouge'	2000
'Tokay'	'Alsace'	'Blanc'	2000
'Aix'	'Provence'	'Rouge'	2000

Vins = $\sigma_{Type='Rouge'}$ (V)
Vins = Vins1 $\bowtie_{Type='Rouge'}$ Année

'Chenas'	'Beaujolais'	'Rouge'	1988
'Aix'	'Provence'	'Rouge'	1988
'Chenas'	'Beaujolais'	'Rouge'	2000
'Aix'	'Provence'	'Rouge'	2000

Propriétés

- $Join(R, S, cond) = Join(S, R, cond)$
- $Card(Join(R, S, cond)) \leq Card(R) \times Card(S)$
- $Arite(Join(R, S, cond)) = Arite(R) + Arite(S)$
- $Join(R, S, cond) = Select(Prod(R, S), cond)$

Jointure naturelle

Définition

- Opération consistant à rapprocher les tuples de deux relations R et S afin de former une relation dont les attributs sont l'union des atrributs de R et S et dont les tuples sont obtenus en composant un tuple de R et un tuple de S ayant même valeurs pour les attributs de même nom.
- Consiste en une opération de jointure spéciale où la condition est l'égalité des attributs de même nom.

Notations

$Join(R, S)$, $R \bowtie S$

Exemple

Vins1

'Chenas'	1983	'Beaujolais'
'Tokay'	1980	'Alsace'
'Aix'	1995	'Provence'

Vins2

'Bandol'	'Rosé'
'Tokay'	'Blanc'
'Cassis'	'Blanc'
'Aix'	'Rouge'

Vins = Vins1 $\bowtie$ Vins2

'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Renommage

Définition

Opération permettant de renommer un attribut de la relation R. Le résultat est la même relation avec un schéma de relation différent. On peut renommer aussi une relation.

Intersection

Définition

Opération portant sur deux relations de même schéma R et S construisant une relation de même schéma ayant pour tuples ceux appartenant à R et à S.

Notations

$R \cap S$, $Inter(R, S)$

Propriétés

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

Lister les n-uplets qui appartiennent à cette relation **ET** à cette autre relation.

Exemple

Vins1

'Chenas'	1983	'Beaujolais'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Aix'	1995	'Provence'	'Rouge'

Vins2

'Bandol'	1988	'Provence'	'Rouge'
'Tokay'	1980	'Alsace'	'Blanc'
'Cassis'	2000	'Provence'	'Blanc'

Vins = Vins1 $\cap$ Vins2

'Tokay'	1980	'Alsace'	'Blanc'

Un petit exercice (1)

Voiture	Immatriculation	Marque	Annee	Prix	IdProprio
	'1111AA01'	'Toyota'	1997	16 000	'Id01'
	'2222BB02'	'Peugeot'	2000	31 200	'Id01'
	'3333CC03'	'Fiat'	1997	2 000	'Id03'
	'4444DD13'	'Fiat'	1995	30 300	'Id02'
	'5555EE62'	'Renault'	1997	21 000	'Id02'
	'6666FF59'	'Opel'	1999	2 900	'Id01'
	'7777ZZ75'	'Ford'	1998	22 222	'Id03'

Personnes	IdProprio	Nom	Prenom	Naissance
	'Id01'	'Martin'	'Paul'	01/02/1967
	'Id02'	'Duval'	'Jean'	03/09/1980
	'Id03'	'Dupond'	'Laurence'	01/01/1945
	'Id04'	'Durand'	'Julie'	03/03/1985

Un petit exercice (2)

1. Liste des immatriculations.
2. Liste des voitures de 1996.
3. Voitures qui appartiennent au proprio Id01.
4. Liste des voitures entre 10000 et 20000 euros.
5. Différentes marques de voitures.
6. Id des propriétaires ayant des voitures de plus de 20000 euros.