Langage SQL

définitions

- Algèbre relationnelle: Ensemble d'opérateurs qui s'appliquent aux relations
- Résultat : nouvelle relation qui peut à son tour être manipulée
- L'algèbre relationnelle permet de faire des recherches dans les relations

Opérateurs de l’algèbre relationnelle

- Opérations unaires (une seule opérande): sélection (noté σ), projection (π), renommage (a)
- Opérations binaires: produit cartésien (x), jointures (|X|), union (∪), intersection (∩), différence (−), division (/)

Sélection (restriction) (1)

T=

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Bob	13	Lyon	Nice
Sam	7	Nice	Nice
Cathy	13	Brest	Brest
Julie	20	Lyon	Brest

Sélection par rapport à une constante

$\sigma_{c_3=''Lyon''}(T)$

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Bob	13	Lyon	Nice
Julie	20	Lyon	Brest

Sélection (restriction) (2)

Sélection par rapport à un critère inter-colonne

$\sigma_{c_3=c_4}(T)$

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Sam	7	Nice	Nice
Cathy	13	Brest	Brest

Sélection à l'aide d'autres opérateurs

$\sigma_{c_2\leq 14}(T)$

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Bob	13	Lyon	Nice
Sam	7	Nice	Nice
Cathy	13	Brest	Brest

Sélection (restriction) (3)

Composition de sélection

$$\sigma_{c_3=''Lyon''}\left(\sigma_{c_2\leq 14}(T)\right)$$

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Bob	13	Lyon	Nice

Conjonction de critères

$$\sigma_{c_3=''Lyon''}\left(\sigma_{c_2\leq 14}(T)\right) \iff \sigma_{c_3=''Lyon'' \text{ et } c_2\leq 14}(T)$$

Projection (1)

T=

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Bob	13	Lyon	Nice
Sam	7	Nice	Nice
Cathy	13	Brest	Brest
Julie	20	Lyon	Brest

Exemple de projection

$\pi_{c_1,c_2}(T)$

c_1(nom)	c_2(âge)
Bob	13
Sam	7
Cathy	13
Julie	20

$\pi_{c_1,c_3}(T)$

c_1(nom)	c_2(adresse)
Bob	Lyon
Sam	Nice
Cathy	Brest
Julie	Lyon

Renumérotation des colonnes

Composition

T=

c_1(nom)	c_2(âge)	c_3(adresse)	c_4 (né à)
Bob	13	Lyon	Nice
Sam	7	Nice	Nice
Cathy	13	Brest	Brest
Julie	20	Lyon	Brest

Quels sont les noms des personnes habitant à Lyon ?

- Algèbre : le résultat d'une opération portant sur des relations est aussi une relation, ce qui rend possible la composition de différentes opérations

Renommage

- Parfois on veut appliquer plusieurs relations à la fois
- Soit on écrit l'opération comme une seule expression relationnelle algébrique en imbriquant les opérations, ou On applique une opération à la fois et on crée des résultats de relations intermédiaires.
- Dans le dernier cas on doit donner des noms aux résultats intermédiaires.
- On peut écrire une seule expression algébrique:
  - PRENOM, NOM, SALAIRE(N°Dép=5(EMPLOYE))
- OU on peut écrire une séquence d'opérations:
  - DEP5_EMP ← N°Dép=5(EMPLOYE)
  - RESULTAT ← PRENOM, NOM, SALAIRE (DEP5_EMP)

Produit cartésien (1)

Homme

c_1 (nom)	c_2 (adresse)
Bob	Lyon
Sam	Nice

Femme

c_1 (nom)	c_2 (adresse)
Cathy	Brest
Julie	Lyon
Linda	Lyon

Quels sont les couples homme-femme ?

Homme × Femme

Quels sont les couples homme-femme d'une même ville ?

$\sigma_{c2=c4}$(Homme × Femme)

Produit cartésien (2)

Homme

c_1 (nom)	c_2 (adresse)
Bob	Lyon
Sam	Nice

Ville

c_1 (nom)	c_2 (nb hab.)	c_3 (départ.)
Nice	340.000	Alp-Mar
Brest	160.000	Finistère
Lyon	420.000	Rhône

Dans quel département habitent les hommes?

$\sigma_{c2=c3}(\text{Homme} \times \text{Ville})$

Jointure (1)

Homme

c_1 (nom)	c_2 (adresse)
Bob	Lyon
Sam	Nice

Femme

c_1 (nom)	c_2 (adresse)
Cathy	Brest
Julie	Lyon
Linda	Lyon

Quels sont les couples homme-femme d'une même ville ?

$$\text{Homme} \bowtie_{c2=c2} \text{Femme} \leftrightarrow \sigma_{c2=c4}(\text{Homme} \times \text{Femme})$$

c_1 (nom)	c_2 (adresse)	c_3 (nom)	c_4 (adresse)
Bob	Lyon	Julie	Lyon
Bob	Lyon	Linda	Lyon

Jointure (2)

Homme

c_1 (nom)	c_2 (adresse)
Bob	Lyon
Sam	Nice

Ville

c_1 (nom)	c_2 (nb hab.)	c_3 (départ.)
Nice	340.000	Alp-Mar
Brest	160.000	Finistère
Lyon	420.000	Rhône

Dans quel département habitent les hommes?

$$\text{Homme} \bowtie_{c2=c1} \text{Ville} \leftrightarrow \sigma_{c2=c3}(\text{Homme} \times \text{Ville})$$

c_1 (nom)	c_2 (adresse)	c_3 (nom)	c_4 (nb hab.)	c_5 (départ.)
Bob	Lyon	Lyon	420.000	Rhône
Sam	Nice	Nice	340.000	Alp-Mar

Union

Homme

c_1 (nom)	c_2 (adresse)
Bob	Lyon
Sam	Nice

Femme

c_1 (nom)	c_2 (adresse)
Cathy	Brest
Julie	Lyon
Linda	Lyon

- Quel est l'ensemble des personnes ?
- Quel est l'ensemble des personnes habitant Lyon ?

Les relations doivent avoir même schéma (même nombre d'attributs dont les domaines correspondent)

Différence

Homme

c_1 (nom)	c_2 (adresse)
Bob	Lyon
Sam	Nice

Femme

c_1 (nom)	c_2 (adresse)
Cathy	Brest
Julie	Lyon
Linda	Lyon

• Quelles sont les villes pour lesquelles on connaît au moins
un homme et aucune femme ?

$$\pi_{c2(Homme)} - \pi_{c2(Femme)}$$

Opérations dérivées

- Intersection
- Division
  - Permet de rechercher les sous-tuples d'une relation qui sont "complétés" par tous ceux d'une autre relation

Vins

Cru	Mill.	Qualité
Volnay	1983	A
Volnay	1979	B
Chablis	1983	A
Chablis	1979	A
Julienas	1986	A

Qualité

Mill.	Qualité
1983	A
1979	A

Division (2) Exemple

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

A

pno
p2

B1

sno
s1
s2
s3
s4

A/B1

pno
p2
p4

B2

sno
s1
s4

A/B2

pno
p1
p2
p4

B3

sno
s1

A/B3

definition

- SQL est acronyme de « Structred Query Language » c’est-à-dire langage d’interrogation structuré
- Un langage qui interroge une base donné… mais pas seulement
- Langage conçu par IBM dans les années 1970 et normalisé après en 1992
- SQL est un langage complet de **gestion** de base de donnés

définition

- La gestion comprend:
  - Interrogation de la base
  - Manipulation de donnée (ajout, mise a jour et suppression
  - Création de la structure (création de table)
  - Contrôle d’accès (qui peut faire quoi)

utilisation

- Le langage SQL est utilisé par la plupart des systèmes de gestion de base de données relationnelles
- Ex: oracle,Informix, sql server, MySql…etc
- Pourtant chaque SGBD a sa variante de sql
- PS: les différences entre la syntaxe de différents SGBD rend difficile la portabilité du code sql

fonctionnement

- Sql désigne les objets (Tables ,colonnes, utilisateurs, etc) par des identificateurs
- Un identificateur est un mot formé de 30 caractère max commençant par une lettre d’alphabet
- Un identificateur peut contenir des lettres, des chiffres est les symbol # $_
- Ne pas utiliser les mot clés SQL comme identificateur (DATE, TYPE, SET…)

Les tables

- Les relations sont stockées sous forme de table composé de lignes et des colonnes
- Exemple de table Etudiant:

CNE	Nom	Prenom
23485	Ali	Drissi
65908	Amine	Alami
75824	Maryam	Alaoui

Les colonnes

- Les données d'une même colonne doivent être du même type
- Le type est spécifié au moment de la création de la table
- Chaque colonne est identifié uniquement au sein de la même table
- Deux colonnes appartenant à la même table ne doivent pas porter le même nom
- Le nom complet de la colonne est *nom_table.nom_colonne*

Types de données

- Types numérique:
  - Nombres entiers: SMALLINT (sur 2 octets, de -32.768 à 32.767), INTEGER (sur 4 octets, de -2.147.483.648 à 2.147.483.647)
  - Nombres décimaux: DECIMAL(p,d) p chiffre significatifs et d chiffre après la virgule (ex: DECIMAL(3,2) => 987,32)
  - Numériques non exactes à virgule flottante: REAL avec 7 chiffre significatives et FLOAT avec 15 chiffres significatives
  - D’autres types numériques peuvent être définie par les SGBD

Types de données

- Chaîne de caractères: deux types
  - CHAR (*longueur*): pour les chaînes de caractères de longueur constante d'une longueur max *longueur* (le reste est remplie par des espaces)
  - VARCHAR(*longueur*): pour les chaînes de caractères de longueur variables et une longueur max de longueur
  - CHAR est en général utilisé pour de courtes chaînes et VARCHAR pour des chaînes longues

Types de données

- Types temporels:
  - DATE:deux chiffres pour le jour, deux chiffres pour le mois et quatre pour l'année (08/11/2000)
  - TIME:heures, minutes et secondes
  - INTERVAL:intervalle de temps
  - TIMESTAMP: permet de spécifier un moment avec date et heur

Types de données

- Types binaires: sert à stocker les images et le son
  - BIT:longueur constante
  - BIT VARYING: longueur variable
- Valeur NULL: absence de valeur et **pas un zéro**

Création de table

- L'ordre CREATE TABLE permet de créer une table en définissant le nom et le type de chacune des colonnes de la table
- Syntaxe: CREATE TABLE *nom_table* (*colonne1 type1, colonne2 type2,* .....)

Clé primaire

- Sur une colonne :

PRIMARY KEY

Exemple: *code smallint primary key*

- Sur plusieurs colonnes

PRIMARY KEY (colonne1, colonne2,...)

Language de manipulation de données

- C’est le language permettant de modifier le contenu de la base de données
- Il est composé des trois commandes suivantes
  - Insert
  - Update
  - Delete

insert

- Insere un nouvel enregistrement dans la table
- Syntaxe: INSERT INTO table (col1,..., coln) VALUES (valeur1,...,valeurn)
- Exemple: insert into personne (CIN, nom, prenom) values ("H34556", "drissi" , "ali ")

update

- Met à jour un enregistrement
- Syntaxe:

  UPDATE table SET col1 = exp1, col2 = exp2, ...

  WHERE condition
- Exemple: update employe set salaire=salaire*1.1

  Where poste = vendeur;

delete

- L’ordre delete permet de supprimer des lignes d’une table
- Syntaxe: DELETE FROM table WHERE condition
- Exemple: delete from employe where date de naissance >= 65;
- PS: la commande delete sans clause where supprime toutes les lignes de la table

interrogation

- L’ordre SELECT permet d’interroger la base de données, il possède les clauses suivantes:
  - SELECT ...
  - FROM ...
  - WHERE ...
  - GROUP BY ...
  - HAVING ...
  - ORDER BY ...
- Seuls SELECT et FROM sont obligatoires
- Les clauses doivent apparaitre avec le même ordre que ci dessus

SELECT

- Cette clause indique les colonnes ou expressions retournés par l'interrogation
- Syntaxe: SELECT [DISTINCT] col1, col2
- Ou SELECT [DISTINCT] expr1 [AS nom1], expr2 [AS nom2]
- Distinct est une option facultative pour afficher les enregistrements différents
- « Select * » affiche toutes les colonnes

FROM

- Détermine les tables concernées par la sélection (il est possible de sélectionner de plusieurs tables)
- FROM table1, table2…

WHERE

- Cette clause permet de sélectionner quelles sont les lignes à sélectionner selon un prédicat (expression logique ayant une valeur logique vrai ou faux) si le résultat est vrai la ligne est sélectionnée
- Syntaxe: WHERE *prédicat*

Les opérateurs logiques

- WHERE exp1 = exp2
- WHERE exp1 != exp2
- WHERE exp1 < exp2
- WHERE exp1 > exp2
- WHERE exp1 <= exp2
- WHERE exp1 >= exp2
- WHERE exp1 BETWEEN exp2 AND exp3
- WHERE exp1 LIKE exp2
- WHERE exp1 NOT LIKE exp2
- WHERE exp1 IN (exp2, exp3,...)
- WHERE exp1 NOT IN (exp2, exp3,...)
- WHERE exp IS NULL
- WHERE exp IS NOT NULL

GROUP BY

- Il est possible de subdiviser la table en groupes, chaque groupe étant l'ensemble des lignes ayant une valeur commune.
- Syntaxe: GROUP BY exp1, exp2,...

HAVING

- sert à préciser quels groupes doivent être sélectionnés.
- Syntaxe: HAVING prédicat
- Elle se place après la clause GROUP BY et porte sur les caractéristiques du groupe

ORDER BY

- Les lignes constituant le résultat d'un SELECT sont obtenues dans un ordre indéterminé. La clause ORDER BY précise l'ordre dans lequel la liste des lignes sélectionnées sera donnée.
- Syntaxe: ORDER BY exp1 [DESC], exp2 [DESC], ...
- Par default l’ordre est croissant l’option DESC doit être mentionnée dans le cas d’un ordre décroissant

Fonctions de groupes

- Les fonctions de groupes peuvent apparaître dans le Select ou le Having
  - AVG moyenne
  - SUM somme
  - MIN plus petite des valeurs
  - MAX plus grande des valeurs
  - VARIANCE variance
  - STDDEV écart type (déviation standard)
  - COUNT(*) nombre de lignes
  - COUNT(col) nombre de valeurs non nulles de la colonne
  - COUNT(DISTINCT col) nombre de valeurs non nulles différentes