Chapitre 4.

Algèbre relationnelle

Algèbre relationnelle

Définitions

- L'algèbre relationnelle a été définie avant SQL.
- SQL est, en quelque sorte, le « *langage pratique* » issu de l'algèbre relationnelle. Autrement dit, SQL est basé sur les concepts de l'algèbre relationnelle.
- Les opérations de base du modèle relationnel forment l'algèbre relationnelle.
- Ces opérations permettent de spécifier des requêtes sur des relations sous forme d'expressions de l'algèbre relationnelle
- Pourquoi étudier l'algèbre relationnelle ?
  - Comprendre conceptuellement comment les résultats sont retournés par les requêtes à partir des tables.
  - Mettre à profit cette compréhension dans l'optimisation des requêtes

Algèbre relationnelle

Opérations unaires : SELECT (Sélection)

- But : retenir certains tuples dans une relation.
- Partitionne horizontalement la relation en deux sous-ensembles. Le premier comprend les tuples qui satisfont la condition de sélection et l’autre, le sous-ensemble des tuples qui ne la satisfont pas.
- Produit en sortie une relation avec le même schéma

Opérations unaires : SELECT (Sélection)

- Syntaxe :

$$\sigma_{\texttt{<Condition de sélection>}}(\texttt{R})$$

- condition élémentaire :

  <attribut> <opérateur comparaison><attribut-ou-constante>

  - Attribut : est un attribut de la relation R
  - Opérateur de comparaison : =, <, >, <=, >=, <>

- les conditions de sélection sont exprimées par combinaison des conditions élémentaires à l'aide des opérateurs logiques AND, OR et NOT
- Exemple : Les employé du département numéro 5

$$\sigma_{\texttt{Dnum=5}}(\texttt{EMPLOYE})$$

Opérations unaires : PROJECT (Projection)

- But : ne retenir que certains attributs d'une relation (en éliminant les doublons).
- Partitionne verticalement la relation en deux relations. La première avec la liste des attributs souhaités et la seconde avec les attributs qui restent.

Algèbre relationnelle

Opérations unaires : PROJECT (Projection)

- Syntaxe :

$$\pi_{\texttt{Liste attributs}}(\texttt{R})$$

avec<liste attributs> est la liste des attributs de R à conserver dans le résultat.

- Exemple : Liste des différents salaires de l'entreprise

$$\pi_{\texttt{salaire}}(\texttt{EMPLOYE})$$

Equivalent en SQL à :

```
SELECT DISTINCT Salaire
FROM EMPLOYE;
```

Algèbre relationnelle

Opérations unaires : Expressions

- Pour effectuer des requêtes, on a besoin, en général, d'une série d'opérations d'algèbre relationnelle l'une après l'autre.
- **Exemple** : liste (nom, prénom et salaire) des employés du département n°5 :

$$\pi_{\text{Nom, Prenom, Salaire}}(\sigma_{\text{Dnum=5}}(\text{EMPLOYE}))$$

- Pour exprimer une opération complexe, on a parfois besoin d'attribuer des noms aux résultats intermédiaires
- **Exemple** : liste (nom, prénom et salaire) des employés du département n°5 :

$$\text{EMPDept5} \leftarrow \sigma_{\text{Dnum=5}}(\text{EMPLOYE})$$
$$\text{RESULTAT} \leftarrow \pi_{\text{Nom, Prenom, Salaire}}(\text{EMPDept5})$$

Algèbre relationnelle

Opérations ensemblistes : l'union, l'intersection et la différence

- On peut définir ces trois opérations sur deux ensembles (union-compatibles) R et S comme suit :
- Union : $R \cup S$ inclut tous les tuples aussi bien ceux de R que ceux de de S en éliminant les doublons
- Intersection : $R \cap S$ inclut les tuples communs à R et à S
- Différence : $R - S$ les tuples de R qui ne sont pas dans S

- L'intersection peut être dérivée de l'opérateur de différence
  - $R \cap S = R - (R - S)$
  - $R \cap S = ((R \cup S) - (R - S)) - (S - R)$

Opérations ensemblistes : le produit cartésien

- But : construire toutes les combinaisons de tuples de deux relations (en général, en vue d'une sélection)
- Syntaxe : $R \times S$
- Chaque tuple de R est combiné avec tous les tuples de S

R

A	B
a	b
b	c
c	b

n tuples

S

C	D	E
c	d	e
b	a	b
a	a	c

m tuples

$\mathbf{R \times S}$

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

n x m tuples

Opérations binaires : la theta-jointure

- Combiner les tuples de R avec les tuples de S mais seulement ceux qui satisfont la condition de jointure

- Syntaxe :

$$R \bowtie_{\text{<condition de jointure>}} S$$

- Condition de jointure :

<condition> AND <condition> AND… <condition>

<condition> a la forme $A_i\ \theta\ B_j$ où A_i est un attribut de R, B_j est un attribut de S et θ est un opérateur de comparaison {=, <=, >=, <, >, <>}

Opérations binaires : équi-jointure et jointure naturelle

- Quand la condition de jointure est exprimée à l'aide de l'opérateur d'égalité seulement, elle est appelée équi-jointure.
- La jointure naturelle dénotée * est équivalente àl'équi-jointure appliquée aux attributs de même nom des deux relations. Si les attributs de même domaine ne sont pas de même nom, il est possible de les renommer (ρ).
- Exemple :

$$\text{PROJETDept} \leftarrow \text{DEPARTEMENT} \bowtie_{<\text{Dnum}=\text{DNumero}>} \text{PROJET}$$

Est équivalente à :

$$\text{DEPT} \leftarrow \rho_{(\text{DNum, Dnom, ChefMAt, DateCreation, DateDebutChef})}\text{DEPARTEMENT}$$

$$\text{PROJETDept} \leftarrow \text{DEPT} * \text{PROJET}$$

Opérations binaires : la division (÷)

- La division est utile pour exprimer certaines requêtes.

Par exemple, liste des employés qui travaillent sur tous les projets sur lesquels travaille l'employé « Rachid Alaoui ».

- Soient $R(A_1, \ldots, A_n)$ et $V(A_1, \ldots, A_m)$

avec $n>m$ et $A_1, \ldots, A_m$ des attributs de même nom dans R et V

$R \div V = \{ <a_{m+1}, a_{m+2}, \ldots, a_n> \,/\, \forall <a_1, a_2, \ldots, a_m> \in V,$
$\exists <a_1, a_2, \ldots, a_m, a_{m+1}, a_{m+2}, \ldots, a_n> \in R\}$

Opérations binaires : la division (÷)

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

Algèbre relationnelle

Arbre algébrique

- Une expression en algèbre relationnelle est représentée sous forme d'un arbre dont les feuilles sont les relations en entrés, les nœuds sont les opérations de l'algèbre relationnelle et dont la racine est l'opération qui fournit le résultat final.

Agrégations et regroupement

- Les fonctions d'agrégation permettent de résumer l'information provenant des tuples. Il est également possible de regrouper ces résumés selon les valeurs de certains attributs.
- l'agrégation est notée par $\mathfrak{I}$ (script F) :

$$_{\text{<attributs\_regroupement>}}\mathfrak{I}_{\text{<liste\_fonctions>}}(R)$$

- Exemple :
  - Nombre d'employés et salaire moyen par département :

$$\rho_{\text{EMPLOYE(Dept, NB\_Emp, Moy\_Salaire)}}(_{\text{Dnum}}\mathfrak{I}_{\text{COUNT Matricule, AVG Salaire}}(\text{EMPLOYE}))$$

Jointure externe

- Jointure externe gauche : ⟕
- Droite : ⟖
- Full : ⟗

Chapitre 5.

Calcul Relationnel

Le calcul des tuples

• En calcul relationnel, une requête de recherche est écrite sous forme d'expression déclarative. On ne décrit ni *comment* ni *dans quel ordre* la requête sera évaluée. C'est pour cela qu'on parle de langage déclaratif pour qualifier le calcul relationnel par opposition à l'algèbre relationnelle qui est elle considérée comme procédurale.

• Le calcul relationnel est important pour deux raisons :

- Il est basé sur la logique.
- Il constitue une partie des fondements de SQL.

Le calcul des tuples

- Le calcul des tuples se fait sur des variables de type tuple dont la portée est généralement une relation de base de données.

- Une requête de calcul relationnel a la forme : `{t|condition(t)}` où `t` est la variable tuple et `condition(t)` est une expression conditionnelle booléenne incluant `t`. Le résultat de cette requête est l'ensemble des tuples `t` qui satisfont la condition `condition(t)`

- Exemple : Les empoyés qui gagnent plus de 5000.

```
{t|EMPLOYE(t) AND t.Salaire>5000}
```

Si on souhaite se limiter aux noms et aux prénoms des employés :

```
{t.Nom, t.Prenom|EMPLOYE(t) AND t.Salaire>5000}
```

Le calcul des tuples

- Exemple : Adresse et date de naissance de l'employé « Rachid ALAOUI »

```
{t.Adresse, t.Dnaiss|EMPLOYE(t) AND
t.Nom='ALAOUI' AND t.Prenom = 'Rachid'}
```

Le calcul des tuples

Expressions et formules

- Une expression en calcul des tuples a la forme :

$$\{t_1.A_j,\ t_2.A_k,\ldots,t_n.A_m \mid \text{Condition}(t_1,t_2,\ldots,t_n,t_{n+1},t_{n+2},\ldots,t_p)\}$$

où t_i sont des variables de tuples et **A_i** sont des attributs des relations sur lesquelles portent ces tuples.

Condition est une formule dont les atomes ont la forme **$R(t_i)$** ou **$t_i.A$ op $t_j.B$** ou **$t_i.A$ op c** où **op** est un opérateur parmi {=,>,>=,<,<=,<>} et **c** est une constante.

Ces atomes sont connectés par les opérateurs logiques **AND**, **OR** et **NOT** pour constituer des formules.

Le calcul des tuples

Quantificateurs existentiel et universel

- Les formules du calcul relationnel supportent aussi les quantificateurs existentiel $\exists$ et universel $\forall$.

Le calcul des tuples

Exemples

- Noms et prénoms des employés du département ‘Informatique’

{t.Nom, t.Prenom|EMPLOYE(t) AND ((∃d) DEPARTEMENT(d) AND d.DNom=‘Informatique’ AND d.Dnumero=t.Dnum)}

- Noms et prénoms des employés et noms et prénoms de leurs supérieurs immédiats

{e.Nom, e.Prenom, s.nom, s.prenom|EMPLOYE(e) AND EMPLOYE(s) AND e.SuperMat=s.Matricule}

Le calcul des tuples

Exemples

- Noms et prénoms des employés qui travaillent pour un projet contrôlé par le département n° 2
{t.Nom, t.Prenom|EMPLOYE(t) AND ((∃p) (∃w) (PROJET(p) AND TRAVAILLE_SUR(w) AND p.DNum= w.PNum AND t.Matricule =w.Emat AND p.DNum = 2)}

- Liste des projets sur lesquels a travaillé l'employé de nom 'Alaoui' aussi bien en tant que collaborateur qu'en tant que chef du département en charge de ces projets

Le calcul des tuples

Exemples

- Liste des employés qui n'ont pas d'enfants
- Liste des chefs de départements qui ont au moins un enfant

Le calcul des domaines

- Le calcul de domaines diffère du calcul des tuples par le type de variables : ce sont des domaines (attributs) et non des tuples

•Une expression en calcul relationnel a la forme :

$$\{x_1, x_2,\ldots,x_n \mid \textbf{Condition}(x_1,x_2,\ldots,x_n,x_{n+1},x_{n+2},\ldots,x_{n+m})\}$$

où x_i sont des variables de domaines et **Condition** est une formule dont les atomes ont l'une des formes :

-$\mathbf{R(x_1 x_2,\ldots,x_j)}$ où R est le nom d'une relation de degré j

- x_i **op** $\mathbf{x_j}$ ou x_i **op c** où **op** est un opérateur parmi {=,>,>=,<,<=,<>} et **c** est une constante.

Comme pour le calcul des tuplrd, ces atomes sont connectées par les opérateurs logiques **AND**, **OR** et **NOT** pour constituer des formules.

Le calcul des domaines

Exemples

-Date de naissance et adresse de l'employé 'ALAOUI Rachid'

$$\{u,v \mid (\exists r)(\exists s)(\exists t)(\exists w)(\exists x)(\exists y)(\exists z) \; AND \; EMPLOYE(rstuvwxyz) \; AND \; s = 'ALAOUI' \; AND \; t = 'Rachid' \}$$

-La même requête peut être exprimée dans une notation plus réduite dans le style QBE :

$$\{u,v \mid EMPLOYE(r,'ALAOUI','Rachid',u,v,w,x,y,z)\}$$