Cours de bases de données, http://sql.bdpedia.fr

Expressions algébriques

Philippe Rigaux

le cnam

Algèbre relationnelle : expressions algébriques

Par composition on exprime avec l'algèbre toutes les requêtes relationnelles que l'on peut aussi exprimer (plus naturellement) avec la logique.

Dans cette session, des exemples d'expressions algébriques courantes

- Composition de sélections
- Jointures
- Différence
- Et même la quantification universelle

Ces diapositives correspondent au support en ligne disponible
sur le site `http://sql.bdpedia.fr/`

Composition de sélection

La conjonction de critères s'obtient en composant des sélections

$$\sigma_{capacité>100}(\sigma_{type='Hôtel'}(Logement))$$

On acceptera de l'écrire

$$\sigma_{capacité>100 \wedge type='Hôtel'}(Logement)$$

L'union est l'équivalent en algèbre de la disjonction logique.

$$\sigma_{capacité>100}(Logement) \cup \sigma_{lieu='Corse'}(Logement)$$

Peut s'écrire

$$\sigma_{capacité>100 \ \vee \ lieu='Corse'}(Logement)$$

Composition de sélection, suite

La différence permet d'exprimer l'opérateur $\neq$.

$$\sigma_{capacité>100}(Logement) - \sigma_{lieu='Corse'}(Logement)$$

On acceptera de l'écrire

$$\sigma_{capacité>100 \wedge \neg(lieu='Corse')}(Logement)$$

En résumé, on peut écrire la sélection en exprimant les critères avec une formule F du calcul propositionnel, $\sigma_F(R)$

Attention à la négation : voir plus loin.

Requêtes conjonctives

Toute requête s'écrivant avec σ, π, $\times$ (et donc $\bowtie$).

Nom des logements en Corse :

$$\pi_{nom}(\sigma_{lieu='Corse'}(Logement))$$

Code des logements où l'on pratique la voile.

$$\pi_{codeLogement}(\sigma_{codeActivité='Voile'}(\text{Activité}))$$

Nom et prénom des clients corses

$$\pi_{\text{nom,prénom}}(\sigma_{\text{région}='Corse'}(Voyageur))$$

Jointures

Nom des clients qui sont allés à Tabriz

$$\pi_{nom}(\text{Voyageur} \bowtie_{idVoyageur=idVoyageur} (\sigma_{codeLogement='ta'}(\text{Séjour})))$$

Quels lieux a visité le client 30

$$\pi_{lieu}(\sigma_{idVoyageur=30}(\text{Séjour}) \bowtie_{codeLogement=code} \text{Logement})$$

Nom des clients qui ont eu l'occasion de faire de la voile. En deux étapes.

$R1 := \texttt{Séjour} \bowtie_{codeLogement=codeLogement} (\sigma_{codeActivité='Voile'}(\texttt{Activité}))$

$\pi_{nom}(\texttt{Voyageur} \bowtie_{idVoyageur=idVoyageur} R1)$

La négation

S'exprime en algèbre avec la différence.

Codes des logements qui ne proposent pas de voile

$$\pi_{code}(\text{Logement}) - \pi_{codeLogement}(\sigma_{codeActivité='Voile'}(\text{Activité}))$$

Raisonnement : je prends tous les logements moins ceux qui proposent de la voile.

Très peu pratique si on compare au `not exists` : la différence
ne s'applique que sur des relations ayant le même schéma.

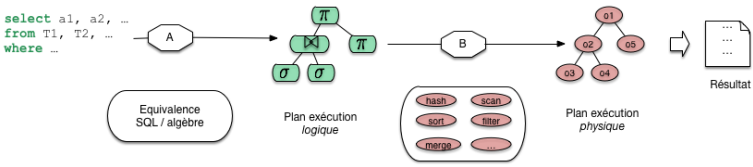
Logique ou fonctionnel, quel langage ?

On soumet une requête SQL, et après ?

On exprime la requête de manière déclarative, le système doit trouver une méthode d'évaluation.

Expression logique, évaluation fonctionnelle

Le système peut produire une expression algébrique pour toute requête SQL.



Il reste à trouver les bons algorithmes, à utiliser correctement les ressources de calcul : c'est pour les administrateurs.

À retenir

L'algèbre est un langage relationnel équivalent à la logique formelle.

- On peut exprimer toutes les requêtes en algèbre.
- Il existe une syntaxe SQL pour toutes les requêtes algébriques
- Le style est celui de la programmation ; moins clair, moins accessible à un non-programmeur, moins proche de l'expression "naturelle" de la requête.
- Peut s'effectuer en plusieurs étapes pour une meilleure clarté.

L'algèbre était conçue à l'origine pour définir l'exécution des requêtes, pas leur expression.