CS3743 Introduction to Databases
Relational Algebra

Weining Zhang

Department of Computer Science
University of Texas at San Antonio

August 15, 2009

Outline

Query Language

- A query language is a formal language design for manipulating and retrieving data from databases
- A query language may be implemented by a DBMS
- Query languages $\neq$ programming languages!
  - Not expected to be "Turing complete".
  - Not intended for complex calculations.
  - Support easy, efficient access to large datasets.

Relation Query Languages

- Relational QLs are simple & powerful.
  - Formal foundation based on set & logic.
  - Allows for much optimization.
- Main relational query languages include
  - Relational algebra. A procedural language in which queries describe how to find answers. Very useful for representing query execution plans.
  - Relational calculi. Declarative languages in which queries describe what answers are, rather than how to find answers.
  - SQL. A standard language implemented by DBMS products

Relational Query

- A query is answered using current relation instances as input, and the result (output) is also a relation instance.
- Schema of output relation is determined from input relations & query language constructs.
- A query is specified against a schema but can be evaluated against any instance.
- Attributes are referenced by names, but can also be referenced by their positions in relation schemas

Relational Algebra

Definition

The relational algebra is a set of relations closed under operations σ (selection), π (projection), $\times$ (cross product), $\cup$ (union), $-$ (set difference), and ρ (rename)

- Closed means both input and output are relations

Example

Let $R_1(A, B, C)$, $R_2(B, C, D)$, $R_3(A, E, F, G)$ be three relations. The following is a relational algebraic expression.

$$\pi_{R_1.A,R_2.D,R_3.G}(\sigma_{R_1.A=R_3.A \wedge R_1.B=R_2.B \wedge R_3.F>5}((\sigma_{B>C}(R_1)) \times R_2 \times R_3))$$

What are these operations? How to compute such an expression?

Selection

- The selection $\sigma_C(R)$ returns the set of tuples of relation R that satisfy condition C, involving only attributes of R
  - Condition C can contain all six comparison operators $=, \neq, >, \geq, <, \leq$, as well as $\neg, \wedge, \vee$

Example

$\sigma_{Major='CS'}(Students)$

Students

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS
678	Ken	3.5	Math

Result

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS

Projection

- The projection $\pi_L(R)$ returns tuples of R with only those attributes listed in L and without any duplicate tuple
  - Duplicate is removed

Example

$\pi_{Major}(Students)$

Students

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS
678	Ken	3.5	Math

Result

Major
CS
Math

Cross Product

- The cross product $R_1 \times R_2$ returns a relation whose schema contains all attributes from R_1 and R_2, and all pairs of tuples $r_1 r_2$, where $r_1 \in R_1$ and $r_2 \in R_2$

Example

$Students \times Awards$

Awards

SID	Amount	Year
456	1500	1998
678	3000	2000

Result

SID	Name	GPA	Major	SID2	Amount	Year
456	John	3.4	CS	456	1500	1998
456	John	3.4	CS	678	3000	2000
457	Carl	3.2	CS	456	1500	1998
457	Carl	3.2	CS	678	3000	2000
678	Ken	3.5	Math	456	1500	1998
678	Ken	3.5	Math	678	3000	2000

Rename

- The rename operation $\rho_S(R)$ (or $\rho_{S(B_1,\ldots,B_k)}(R)$) returns the relation R with name S (and new attribute names B_i)

Example

$\rho_{CS\_Students}(\sigma_{Major='CS'}(Students))$

Students

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS
678	Ken	3.5	Math

CS_Students

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS

Set Operations

- Set operations $R_1 \cup R_2$, $R_1 - R_2$, and $R_1 \cap R_2$, where R_1 and R_2 are union compatible, return a relation that has the schema of R_1(or R_2) and contains tuples according to standard set operations$\cup$, $\cap$, and $-$.

Definition

Two relations are union compatible if their attributes are one-one correspondent, and the corresponding attributes have identical domain.

Set operations automatically remove duplicate tuples

Set Operations

TAs

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS
678	Ken	3.5	Math

RAs

SID	Name	GPA	Major
456	John	3.4	CS
223	Bob	3.0	Ed

$TAs \cup RAs$

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS
678	Ken	3.5	Math
223	Bob	3.0	Ed

$TAs \cap RAs$

SID	Name	GPA	Major
456	John	3.4	CS

$TAs - RAs$

SID	Name	GPA	Major
457	Carl	3.2	CS
678	Ken	3.5	Math

Additional Operations

A number of additional operations were also introduced to make relational algebraic expressions more concise. These operations include

- $\cap$ (Set intersection)
- $\bowtie$ (Inner joins) (there are several types)
- $\div$ (Division)
- Several types of outer joins
- Aggregate functions

Joins

- A join is $R \bowtie_\theta S \equiv \sigma_\theta(R \times S)$, where θ is the join condition containing comparisons of the form $R.A\omega S.B$, $\omega \in \{=, >, \geq, <, \leq\}$ and logic connectives $\neg, \vee, \wedge$
- There are three types of joins
  - Theta-join (as defined above)
  - Equi-join, a theta-join but $\omega \in \{=\}$
  - Natural join, a theta-join with an implicit θ such that
    - for each attribute A shared by R and S, θ contains $R.A = S.A$ and nothing else
    - duplicate attributes are removed in output schema

Examples of Joins

Students

SID	Name	GPA	Age	Prof
456	John	3.4	29	123
457	Carl	3.2	35	123
678	Ken	3.5	25	154

Profs

PID	PName	Age	Dept
123	John	35	CS
154	Scott	28	Ed

- Theta-Join. $Students \bowtie_{Students.Age \leq Profs.Age} Profs$

  Result

SID	Name	GPA	Age	Prof	PID	PName	Age	Dept
456	John	3.4	29	123	123	John	35	CS
457	Carl	3.2	35	123	123	John	35	CS
678	Ken	3.5	25	154	123	John	35	CS
678	Ken	3.5	25	154	154	Scott	28	Ed

Examples of Joins

- Equi-join. $Students \bowtie_{Prof=PID \wedge Name=PName} Profs$

  Result

SID	Name	GPA	Age	Prof	PID	PName	Age	Dept
456	John	3.4	29	123	123	John	35	CS

- Natural join. $Students \bowtie Profs$

  Result

SID	Name	GPA	Age	Prof	PID	PName	Dept
457	Carl	3.2	35	123	123	John	CS

Quizzes on Joins

- Suppose $R_1$and R_2 share no common attribute, what is the result of $R_1 \bowtie R_2$?
- What is the result of $R \bowtie R$?
- Consider two relations.
  Students(SSN, Name, GPA, Major, Age, PSSN)
  Profs(PSSN, Name, Office, Age, Dept)
  - Which type of join can pairs the names of students with the names of their respective advisors?

Division

- The division $R \div S$ requires $schema(S) \subset schema(R)$ and its result is W, such that $schema(W) = schema(R) - schema(S)$ and $W \times S \subseteq R$.

Example:

Takes

SID	CNO
456	CS210
456	CS321
456	CS135
457	CS210
457	CS321
532	CS210
678	CS321

$\div$

CS_Req

CNO
CS210
CS321

$=$

Result

SID
456
457

Aggregate Functions

- The operator ${}_{GL}\mathscr{F}_{FL}(R)$, groups tuples of R by a list of grouping attributes GL and applies a list of functions FL, such as $count()$, $min()$, $max()$, $avg()$, or $sum()$, to each group

Example

${}_{Major}\mathscr{F}_{count(SID),avg(GPA)}(Students)$

Students

SID	Name	GPA	Major
456	John	3.4	CS
457	Carl	3.2	CS
678	Ken	3.5	Math

$\Rightarrow$

Result

Major	$count(SID)$	$avg(GPA)$
CS	2	3.3
Math	1	3.5

Sample Queries in RA

Assume a database schema

```
Students(SID, Name, Age, Major, GPA)
Courses(CNO, Title, Hours, Dept)
Takes(SID, CNO, Semester, Grade)
Departments(Name, Office, Chair)
```

- Primary keys are underlined.
- Course.Dept is a foreign key referencing Department.Name.
- Examples of CNO include 'CS374' and 'MA210'.
- Examples of Major include 'CS' and 'Math'.
- Examples of Semesters include 'F1999'

Sample Queries in RA

- Query: "List titles of courses that were taken by some students in Fall, 1999"
- Solution:

$$\pi_{Title}(\sigma_{Term='F1999'}(Courses \times Takes))$$

or equivalently,

$$\pi_{Title}(Courses \bowtie (\sigma_{Term='F1999'}(Takes)))$$

- How to list titles of courses that were taken by some EE students in Fall, 1999?

Sample Queries in RA

- Query: "Find names of students who have the highest GPA"
- Solution (using temporary variables)
  1. Find students whose GPA is lower than some other students

$$\rho_{LoserSTD}(\pi_{SID}(Students \bowtie_{GPA<S.GPA} (\rho_S(Students))))$$

  2. Subtract from students those who has lower GPAs

$$\rho_{WinerSTD}(\pi_{SID}(Students) - LoserSTD)$$

  3. Find the name of the remaining students

$$\pi_{Name}(Students \bowtie WinerSTD)$$

Sample Queries in RA

- "List IDs of students who take all CS courses"

$$\pi_{SID,CNO}(Takes) \div \pi_{CNO}(\sigma_{Dept='CS'}(Courses))$$

- "For each CS student, list the name, ID and titles of courses that the student takes"

$$\pi_{S.SID,Name,Title}(\rho_S(\sigma_{Major='CS'}(Students)) \bowtie_{S.SID=T.SID} (\rho_T(Takes) \bowtie Courses))$$

- "For each student, list the name and the number of courses the student has taken"

$$_{Name}\mathcal{F}_{count(CNO)}(\rho_S(Students) \bowtie_{S.SID=T.SID} (\rho_T(Takes)))$$

Sample Queries in RA

- "List names of students who took every course that was taken by John Green"
- "List names of students who never had a grade lower than C"
- "List names of departments that offer courses taken by both Mary Smith and John Green"
- "List ID and name of CS students who take CS374"