Database Management Systems

Module 17: Formal Relational Query Languages/2

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Module Recap

- Relational Algebras and its Operations

Module Objectives

- To understand formal calculus-based query language through relational algebra

Module Outline

- Tuple Relational Calculus (Overview only)
- Domain Relational Calculus (Overview only)
- Equivalence of Algebra and Calculus

Formal Relational Query Language

- Relational Algebra
  - Procedural and Algebra based
- Tuple Relational Calculus
  - Non-Procedural and Predicate Calculus based
- Domain Relational Calculus
  - Non-Procedural and Predicate Calculus based

Predicate Logic

Predicate Logic

Predicate Logic or **Predicate Calculus** is an extension of **Propositional Logic** or **Boolean Algebra**.

It adds the concept of predicates and quantifiers to better capture the meaning of statements that cannot be adequately expressed by propositional logic.

Tuple Relational Calculus and **Domain Relational Calculus** are based on **Predicate Calculus**

Predicate

- Consider the statement, "x is greater than 3". It has two parts. The first part, the variable x, is the subject of the statement. The second part, "is greater than 3", is the predicate. It refers to a property that the subject of the statement can have.
- The statement "x is greater than 3" can be denoted by $P(x)$ where P denotes the predicate "is greater than 3" and x is the variable.
- The predicate P can be considered as a function. It tells the truth value of the statement $P(x)$ at x. Once a value has been assigned to the variable x, the statement $P(x)$ becomes a proposition and has a *truth* or *false* value.
- In general, a statement involving n variables $x_1, x_2, x_3, \cdots, x_n$ can be denoted by $P(x_1, x_2, x_3, \cdots, x_n)$. Here P is also referred to as *n*-place predicate or a *n*-ary predicate.

Quantifiers

In predicate logic, predicates are used alongside quantifiers to express the extent to which a predicate is true over a range of elements. Using *quantifiers* to create such propositions is called *quantification*. There are two types of quantifiers:

- **Universal Quantifier**
- **Existential Quantifier**

Universal Quantifier

Universal Quantification: Mathematical statements sometimes assert that a property is true for all the values of a variable in a particular domain, called the **domain of discourse**

- Such a statement is expressed using universal quantification.
- The universal quantification of $P(x)$ for a particular domain is the proposition that asserts that $P(x)$ is *true* for all values of x in this domain
- The domain is very important here since it decides the possible values of x
- Formally, The universal quantification of $P(x)$ is the statement "$P(x)$ for all values of x in the domain".
- The notation $\forall P(x)$ denotes the universal quantification of $P(x)$. Here $\forall$ is called the universal quantifier. $\forall P(x)$ is read as "for all x $P(x)$".
- Example: Let $P(x)$ be the statement "$x + 2 > x$". What is the truth value of the statement $\forall x\ P(x)$?
  Solution: As $x + 2$ is greater than x for any real number, so $P(x) \equiv T$ for all x or $\forall x\ P(x) \equiv T$

Existential Quantifier

Existential Quantification: Some mathematical statements assert that there is an element with a certain property. Such statements are expressed by existential quantification. Existential quantification can be used to form a proposition that is true if and only if $P(x)$ is *true* for at least one value of x in the domain.

- Formally, the existential quantification of $P(x)$ is the statement "There exists an element x in the domain such that $P(x)$".
- The notation $\exists P(x)$ denotes the existential quantification of $P(x)$. Here $\exists$ is called the existential quantifier. $\exists P(x)$ is read as "There is atleast one such x such that $P(x)$"
- Example: Let $P(x)$ be the statement "$x > 5$". What is the truth value of the statement $\exists x P(x)$?
  Solution: $P(x)$ is *true* for all real numbers greater than 5and *false* for all real numbers less than 5. So $\exists x\ P(x) \equiv T$

Tuple Relational Calculus

Tuple Relational Calculus

TRC is a non-procedural query language, where each query is of the form

$$\{t \mid P(t)\}$$

where t = resulting tuples,
P(t) = known as predicate and these are the conditions that are used to fetch t.
P(t) may have various conditions logically combined with OR ($\vee$), AND ($\wedge$), NOT($\neg$).

It also uses quantifiers:
$\exists t \in r(Q(t))$ = "there exists" a tuple in t in relation r such that predicate Q(t) is true.
$\forall t \in r(Q(t))$ = Q(t) is true "for all" tuples in relation r.

- $\{P \mid \exists S \in Students\ and\ (S.CGPA > 8 \wedge P.name = S.sname \wedge P.age = S.age)\}$: returns the name and age of students with a CGPA above 8.

Predicate Calculus Formula

a) Set of attributes and constants
b) Set of comparison operators: $(e.g., <, \leq, =, \neq, >, \geq)$
c) Set of connectives: and $(\wedge)$, or $(\vee)$, not $(\neg)$
d) Implication $(\Rightarrow)$: $x \Rightarrow y$, if x if true, then y is true
   $x \Rightarrow y \equiv \neg x \vee y$
e) Set of quantifiers:
   - $\exists t \in r(Q(t)) \equiv$ "there exists" a tuple in t in relation r such that predicate $Q(t)$ is true
   - $\forall t \in r(Q(t)) \equiv Q$ is true "for all" tuples t in relation r

TRC Example

Student

Fname	Lname	Age	Course
David	Sharma	27	DBMS
Aaron	Lilly	17	JAVA
Sahil	Khan	19	Python
Sachin	Rao	20	DBMS
Varun	George	23	JAVA
Simi	Verma	22	JAVA

Q.1 Obtain the first name of students whose age is greater than 21.

Solution:

$$\{t.Fname \mid Student(t) \wedge t.age > 21\}$$

$$\{t.Fname \mid t \in Student \wedge t.age > 21\}$$

$$\{t \mid \exists s \in Student(s.age > 21 \wedge t.Fname = s.Fname)\}$$

Fname
David
Varun
Simi

TRC Example (2)

Consider the relational schema
student($\underline{rollNo}$, $name$, $year$, $courseId$)
course($\underline{courseId}$, $cname$, $teacher$)

Q.2 Find out the names of all students who have taken the course name 'DBMS'.

- $\{t \mid \exists s \in student\ \exists c \in course(s.courseId = c.courseId \land c.cname = \text{'DBMS'} \land t.name = s.name)\}$
- $\{s.name \mid s \in student \land \exists c \in course(s.courseId = c.courseId \land c.cname = \text{'DBMS'})\}$

Q.3 Find out the names of all students and their rollNo who have taken the course name 'DBMS'.

- $\{s.name, s.rollNo \mid s \in student \land \exists c \in course(s.courseId = c.courseId \land c.cname = \text{'DBMS'})\}$
- $\{t \mid \exists s \in student\ \exists c \in course(s.courseId = c.courseId \land c.cname = \text{'DBMS'}$
  $\land t.name = s.name \land t.rollNo = s.rollNo)\}$

TRC Example (3)

Consider the following relations:
Flights(*flno*, *from*, *to*, *distance*, *departs*, *arrives*)
Aircraft(*aid*, *aname*, *cruisingrange*)
Certified(*eid*, *aid*)
Employees(*eid*, *ename*, *salary*)

Q.4. Find the eids of pilots certified for Boeing aircraft.

RA

$$\Pi_{eid}(\sigma_{aname='Boeing'}(Aircraft \bowtie Certified))$$

TRC

- $\{C.eid \mid C \in Certified \land \exists A \in Aircraft(A.aid = C.aid \land A.aname = \text{'Boeing'})\}$
- $\{T \mid \exists C \in Certified \exists A \in Aircraft(A.aid = C.aid \land A.aname = \text{'Boeing'}$
  $\land T.eid = C.eid)\}$

TRC Example (4)

Consider the following relations:
Flights(*<u>flno</u>, from, to, distance, departs, arrives*)
Aircraft(*<u>aid</u>, aname, cruisingrange*)
Certified(*<u>eid, aid</u>*)
Employees(*<u>eid</u>, ename, salary*)

Q.5. Find the names and salaries of certified pilots working on Boeing aircrafts.

RA

$$\Pi_{ename,salary}(\sigma_{aname=\text{`Boeing'}}(Aircraft \bowtie Certified \bowtie Employees))$$

TRC

$$\{P \mid \exists E \in Employees\ \exists C \in Certified\ \exists A \in Aircraft(A.aid = C.aid \land A.aname = \text{`Boeing'} \land E.eid = C.eid \land P.ename = E.ename \land P.salary = E.salary)\}$$

TRC Example (5)

Consider the following relations:
Flights(*flno*, *from*, *to*, *distance*, *departs*, *arrives*)
Aircraft(*aid*, *aname*, *cruisingrange*)
Certified(*eid*, *aid*)
Employees(*eid*, *ename*, *salary*)

Q.6 Identify the flights that can be piloted by every pilot whose salary is more than \$100,000.

- $\{F.flno \mid F \in Flights \wedge \exists C \in Certified \exists E \in Employees(E.salary > 100,000 \wedge E.eid = C.eid)\}$

Source: Database Management System by Raghu Ramakrishnan.

Safety of Expressions

- It is possible to write tuple calculus expressions that generate infinite relations
- For example, $\{t \mid \neg t \in r\}$ results in an infinite relation if the domain of any attribute of relation r is infinite
- To guard against the problem, we restrict the set of allowable expressions to safe expressions
- An expression $\{t \mid P(t)\}$ in the tuple relational calculus is *safe* if every component of t appears in one of the relations, tuples, or constants that appear in P.
  - NOTE: this is more than just a syntax condition
  - E.g. $\{t \mid t[A] = 5 \vee true\}$ is not safe — it defines an infinite set with attribute values that do not appear in any relation or tuples or constants in P

Domain Relational Calculus

Domain Relational Calculus

- A non-procedural query language equivalent in power to the tuple relational calculus
- Each query is an expression of the form:

$$\{< x_1, x_2, \ldots, x_n > | P(x_1, x_2, \ldots, x_n)\}$$

  - $x_1, x_2, \ldots, x_n$ represent domain variables
  - P represents a formula similar to that of the predicate calculus

Equivalence of Algebra and Calculus

Equivalence of RA, TRC and DRC

Select Operation

$R = (A, B)$

Relational Algebra: $\sigma_{B=17}(r)$

Tuple Calculus: $\{t \mid t \in r \wedge B = 17\}$

Domain Calculus: $\{<a, b> \mid <a, b> \in r \wedge b = 17\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Project Operation

$R = (A, B)$

Relational Algebra:	$\Pi_A(r)$
Tuple Calculus:	$\{t \mid \exists\, p \in r\ (t[A] = p[A])\}$
Domain Calculus:	$\{\langle a \rangle \mid \exists\, b\ (\langle a, b \rangle \in r)\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Combining Operations

R = (A, B)

Relational Algebra:	$\Pi_A(\sigma_{B=17}\ (r))$
Tuple Calculus:	$\{t \mid \exists\ p \in r\ (t[A] = p[A] \wedge p[B] = 17)\}$
Domain Calculus:	$\{<a> \mid \exists\ b\ (<a, b> \in r \wedge b = 17)\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Union

R = (A, B, C)   S = (A, B, C)

Relational Algebra: $r \cup s$

Tuple Calculus: $\{t \mid t \in r \lor t \in s\}$

Domain Calculus: $\{\langle a, b, c\rangle \mid \langle a, b, c\rangle \in r \lor \langle a, b, c\rangle \in s\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Set Difference

R = (A, B, C)   S = (A, B, C)

Relational Algebra: $r - s$

Tuple Calculus: $\{t \mid t \in r \wedge t \notin s\}$

Domain Calculus: $\{<a, b, c> \mid <a, b, c> \in r \wedge <a, b, c> \notin s\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Intersection

R = (A, B, C) S = (A, B, C)

Relational Algebra: $r \cap s$

Tuple Calculus: $\{t \mid t \in r \wedge t \in s\}$

Domain Calculus: $\{<a, b, c> \mid <a, b, c> \in r \wedge <a, b, c> \in s\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Cartesian/Cross Product

$R = (A, B) \qquad S = (C, D)$

Relational Algebra: $r \times s$

Tuple Calculus: $\{t \mid \exists\, p \in r\ \exists\, q \in s\ (t[A] = p[A] \wedge t[B] = p[B] \wedge t[C] = q[C] \wedge t[D] = q[D])\}$

Domain Calculus: $\{<a, b, c, d> \mid <a, b> \in r \wedge <c, d> \in s\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Natural Join

R = (A, B, C, D)   S = (B, D, E)

Relational Algebra: $r \bowtie s$

$\Pi_{r.A, r.B, r.C, r.D, s.E}(\sigma_{r.B = s.B \wedge r.D = s.D}(r \times s))$

Tuple Calculus: $\{t \mid \exists\, p \in r\ \exists\, q \in s\ (t[A] = p[A] \wedge t[B] = p[B] \wedge t[C] = p[C] \wedge t[D] = p[D] \wedge t[E] = q[E] \wedge p[B] = q[B] \wedge p[D] = q[D])\}$

Domain Calculus: $\{<a, b, c, d, e> \mid <a, b, c, d> \in r \wedge <b, d, e> \in s\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Equivalence of RA, TRC and DRC

Division

R = (A, B)   S = (B)

Relational Algebra: $r \div s$

Tuple Calculus: $\{t \mid \exists\, p \in r\ \forall q \in s\ (p[B] = q[B] \Rightarrow t[A] = p[A])\}$

Domain Calculus: $\{\langle a \rangle \mid \langle a \rangle \in r \land \forall \langle b \rangle\ (\langle b \rangle \in s \Rightarrow \langle a, b \rangle \in r)\}$

Source: http://www.cs.sfu.ca/CourseCentral/354/louie/Equiv_Notations.pdf

Module Summary

- Introduced tuple relational and domain relational calculus
- Illustrated equivalence of algebra and calculus

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.
Edited and new slides are marked with "PPD".