Relational Calculus

S Ranbir Singh @ CS344

Chapter 4: Database Management System, 3rd Ed. Ramakrishnan & Gehrke

Sailors(sid: integer, sname: string, rating: integer, age: real)
Boats(bid: integer, bnarne: string, coloT: string)
Reserves(sid: integer, bid: integer, day: date)

Find names of sailors who've reserved boat #103

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

An Instance of Boats

sid		rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

An Instance of Sailors

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

An Instance of Reserves

Relational Algebra: $\pi_{sname}(\sigma_{bid=103}(\text{Reserves} \bowtie Sailors))$

Relational Calculus

- Comes in two flavours:
  - *Tuple relational calculus* (TRC)
  - *Domain relational calculus* (DRC).
- Calculus has *variables, constants, comparison ops*, *logical connectives* and *quantifiers*.
  - *TRC*: Variables range over (i.e., get bound to) *tuples*.
  - *DRC*: Variables range over *domain elements* (= field values).
- Expressions in the calculus are called *formulas.*
- An answer tuple is essentially an assignment of constants to variables that make the formula evaluate to *true*.

Tuple Relational Calculus

- Query: {T|P(T)}
  - T is tuple variable
  - P(T) is a formula that describes T
  - Result, the set of all tuples t for which P(t) evaluates True.
- Ex: Find all sailors with a rating above 7.

$$\{S \mid S \in Sailors \wedge S.rating > 7\}$$

- Atomic formula
  - $R \in \mathrm{Re}\,l$
  - R.a op S.b ,   op is one of $<, >, =, \leq, \geq, \neq$
  - R.a op constant

TRC

- Formula
  - Any atomic formula
  - $\neg p,\ p \wedge q,\ p \vee q,\ p \Rightarrow q$ where p and q are formula
  - $\exists R(p(R))$ where variable R is tuple variable
  - $\forall R(p(R))$ where variable R is tuple variable
- Example: Find the names and ages of sailors with a rating above 7

$$\{P \mid \exists S \in Sailors(S.rating > 7 \wedge P.name = S.name \wedge P.age = S.age)\}$$

Free and Bound Variables

- The use of **quantifiers** $\exists X$ and $\forall X$ in a formula is said to **bind** X in the formula
  - A variable that is **not bound** is **free**
- Query: {T|P(T)}
- The variable that appears on the left side of | should be the only free variable in the formula P(T)

Selection and Projection Operations

- Find all sailors with a rating above 7

$$\{S \mid S \in Sailors \wedge S.rating > 7\}$$

Selection and Projection Operations

- Find all sailors with a rating above 7

$$\{S \mid S \in Sailors \wedge S.rating > 7\}$$

- Find the names and ages of sailors with a rating above 7

$$\{P \mid \exists S \in Sailors(S.rating > 7 \wedge P.name = S.name \wedge P.age = S.age)\}$$

Union Operations

- Find all sailors with a rating above 7 and reserve boat #103

$$\{S \mid S \in Sailors \wedge S.rating > 7 \wedge$$
$$\exists R \in Reserve(R.sid = S.sid \wedge R.bid = \#103)\}$$

- Use of existence join two relations

Division Operations

- Find all sailors who have reserved **all boats**

Find all sailors S such that for all the Boats there exists a tuple in Reserve showing that sailor S has a reservation in R

Division Operations

- Find all sailors who have reserved **all boats**

Find all sailors S such that for all the Boats there exists a tuple in Reserve showing that sailor S has a reservation in R

$$\{S \mid S \in Sailors \wedge$$
$$\forall B \in Boats(\exists R \in \mathrm{Re}\,serve(R.sid = S.sid \wedge B.bid = R.bid))\}$$

Division Operations

- Find all sailors who have reserved **all RED boats**

Find all sailors S such that if there are RED boats they are always reserved by S

Division Operations

- Find all sailors who have reserved **all RED boats**

Find all sailors S such that if there are RED boats they are always reserved by S

$$\{S \mid S \in Sailors \wedge$$
$$\forall B(B \in Boats \wedge$$
$$B.color = RED \Rightarrow \exists R \in \mathrm{Re}\,serve(R.sid = S.sid \wedge B.bid = R.bid))\}$$

Division Operations

- Find all sailors who have reserved **all RED boats**

Find sailors *S* such that, for all boats *B,* either the boat is not red or a Reserves tuple shows that sailor *S* has reserved boat *B.*

$$\{S \mid S \in Sailors \wedge$$
$$\forall B(B \in Boats \wedge$$
$$B.color \neq RED \vee \exists R \in \mathrm{Re}\,serve(R.sid = S.sid \wedge B.bid = R.bid))\}$$

Unsafe Queries, Expressive Power

- It is possible to write syntactically correct calculus queries that have an infinite number of answers! Such queries are called *unsafe*.
  - e.g.,

$$\left\{ S \mid \neg\left(S \in Sailors \right) \right\}$$

- It is known that every query that can be expressed in relational algebra can be expressed as a **safe query** in DRC / TRC; the converse is also true.

Domain Relational Calculus

- *Query* has the form:

$$\left\{ \langle x1, x2, \ldots, xn \rangle \mid p\left(\langle x1, x2, \ldots, xn \rangle \right) \right\}$$

❖ *Answer* includes all tuples $\langle x1, x2, \ldots, xn \rangle$ that make the *formula* $p\left(\langle x1, x2, \ldots, xn \rangle \right)$ be *true*.

❖ *Formula* is recursively defined, starting with simple *atomic formulas* (getting tuples from relations or making comparisons of values), and building bigger and better formulas using the *logical connectives*.

DRC Formulas

- *Atomic formula:*
  - $\langle x1, x2, ..., xn \rangle \in Rname$, or X *op* Y, or X *op* constant
  - *op* is one of $<, >, =, \leq, \geq, \neq$
- *Formula:*
  - an atomic formula, or
  - $\neg p, p \wedge q, p \vee q$, where p and q are formulas, or
  - $\exists X\,(p(X))$, where variable X is *free* in p(X), or
  - $\forall X\,(p(X))$, where variable X is *free* in p(X)

Find all sailors with a rating above 7

$$\left\{ \langle I, N, T, A \rangle \mid \langle I, N, T, A \rangle \in Sailors \wedge T > 7 \right\}$$

- The condition $\langle I, N, T, A \rangle \in Sailors$ ensures that the domain variables *I, N, T* and *A* are bound to fields of the same Sailors tuple.
- The term $\langle I, N, T, A \rangle$ to the left of `|' (which should be read as *such that*) says that every tuple $\langle I, N, T, A \rangle$ that satisfies *T*>7 is in the answer.

Find sailors rated > 7 who've reserved boat #103

$$\left\{ \langle I,N,T,A \rangle \mid \langle I,N,T,A \rangle \in Sailors \land T > 7 \land \right.$$

$$\left. \exists\ Ir, Br, D \left(\langle Ir, Br, D \rangle \in \mathrm{Re}serves \land Ir = I \land Br = 103 \right) \right\}$$

- We have used $\exists\ Ir,\ Br,\ D\ \ (\ldots)$ as a shorthand for

  $$\exists\ Ir \left(\exists\ Br \left(\exists\ D (\ldots) \right) \right)$$

- Note the use of $\exists$ to find a tuple in Reserves that `joins with' the Sailors tuple under consideration.

Find sailors rated > 7 who've reserved a red boat

$$\left\{\left\langle I,N,T,A\right\rangle \mid \left\langle I,N,T,A\right\rangle \in Sailors \wedge T>7 \wedge\right.$$

$$\exists\ Ir,Br,D\left(\left\langle Ir,Br,D\right\rangle \in \mathrm{Re}serves \wedge Ir=I \wedge\right.$$

$$\left.\left.\exists\ B,BN,C\left(\left\langle B,BN,C\right\rangle \in Boats \wedge B=Br \wedge C='red'\right)\right)\right\}$$

Find sailors who've reserved all boats

$$\left\{ \langle I,N,T,A \rangle \mid \langle I,N,T,A \rangle \in Sailors \wedge \right.$$

$$\forall \langle B,BN,C \rangle \in Boats$$

$$\left. \left(\exists \langle Ir,Br,D \rangle \in \mathrm{Re}serves \left(I = Ir \wedge Br = B \right) \right) \right\}$$

Summary

Select Operation

$R = (A, B)$

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

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

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

Project Operation

$R = (A, B)$

Relational Algebra: $\Pi_A(r)$

Tuple Calculus: $\{t \mid \exists\ p \in r\ (t[A] = p[A])\}$

Domain Calculus: $\{<a> \mid \exists\ b\ (\ <a, b> \in r\ )\}$

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: $\{\langle a\rangle \mid \exists\, b\, (\langle a, b\rangle \in r \wedge b = 17)\}$

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: $\{\langle a, b, c, d, e\rangle \mid \langle a, b, c, d\rangle \in r \wedge \langle b, d, e\rangle \in s\}$

Union

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

Relational Algebra: $r \cup s$

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

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

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

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: $\{\langle a, b, c\rangle \mid \langle a, b, c\rangle \in r \wedge \langle a, b, c\rangle \notin s\}$

Cartesian/Cross Product

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

Relational Algebra: $\quad r \times s$

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

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

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

Use of the Universal Quantifier

salary = (employee, salary-amount)

To find the maximum salary-amount:

(Extended) Relational Algebra:

$$\max_{\text{salary-amount}}(\text{salary})$$

Tuple Calculus:

$$\{t \mid \forall p \in \text{salary} \Rightarrow p[\text{salary-amount}] \leq t[\text{salary-amount}]\}$$

Domain Calculus:

$$\{<s> \mid \exists e(<e, s> \in \text{salary} \wedge \forall e1, s1(<e1, s1> \in \text{salary} \Rightarrow s1 \leq s))\}$$