CS354: Database

Relational Calculus

- Declarative query language that describes what is to be retrieved rather than how to retrieve it (nonprocedural)
- Two flavors of relational calculus: Tuple relational calculus (TRC) and Domain relational calculus (DRC)
- Relational calculus and relational algebra are logically equivalent (same logical content)

Relational Calculus

- Calculus has variables, constants, comparison operations, logical connectives, and quantifiers
  - TRC: Variables range over (i.e., get bound to) tuples. Similar to SQL
  - DRC: Variables range over domain elements (field values)
  - Both are simple subsets of first-order Logic
- Expression in calculus are called formulas

Tuple Relational Calculus (TRC)

- Tuple variable: a variable name that represents data tuples in the database
  - Typically denoted using a lower case letter
- Range relation: the relation that is the range for a tuple variable
  - Expression R(t) is evaluated as follows:
    R(t) = true if tuple t is a tuple from the relation R
    R(t) = false if tuple t is not a tuple from the relation R

TRC

- A query in TRC has the form: {t | CONDITION(t) }

- Returns all tuples for which the condition or formula evaluates to true

- Formula is recursively defined, starting with simple atomic formulas and building more complex operators using the logical operators

TRC Formula

- An atomic formula is one of the following:
  - $t \in R$
  - R.a *op* S.b $\qquad <, >=, \geq, \leq, \neq$
  - R.a *op* constant
- A formula can be:
  - An atomic formula
  - NOT p, p AND q, p OR q, where p and q are formulas
  - Special quantifiers

TRC Simple Examples

- {t | Employee(t) AND t.salary > 50000}
  - Retrieve all tuples t such that t is a tuple of the relation EMPLOYEE and their salary amount is greater than 50000
- {t.fname, t.lname | Employee(t) AND t.salary > 50000}
  - Retrieve the first and last name of employees whose salary is greater than 50000
- {t.salary | Employee(t) AND t.fname = 'John' AND t.lname='Smith'}
  - Retrieve the salary of the employee "John Smith"

Special Formula Quantifiers

Two special quantifiers can appear in formulas

- Universal quantifier $(\forall t)\ (\mathrm{Condition}(t))$
  evaluates to true if all tuples t satisfies Condition(t)
  otherwise false

- Existential quantifier $(\exists t)\ (\mathrm{Condition}(t))$
  evaluates to true if there is some (at least one) tuple t
  that satisfies Condition(t)

Free and Bound Variables

- The use of special quantifiers in a formula binds the variable t
  - A variable that is not bound is free
- The variable t that appears to the left of | must be the only free variable in the formula

TRC Example (2)

SAILORS (sid, sname, rating, age)
RESERVES (sid, bid, day)
BOATS (bid, bname, color)

- $\pi_{\text{sname}}(\sigma_{\text{rating}>1}(\text{SAILORS}))$

$$\{t \mid (\exists s)\ (\text{SAILORS}(s)\ \text{AND}\ s.\text{rating} > 1\ \text{AND}\ t.\text{sname} = s.\text{sname})\}$$

free

bound

CONVENTION: the attributes of the free variable t are exactly the ones mentioned in the formula!

TRC Example (3)

Find the department number of the Research department

- $\{d.\text{dno} \mid \text{Department}(d) \text{ AND } d.\text{dname} = \text{'Research'}\}$

$$\{d.\text{dno} \mid \text{Department}(d)$$
$$\text{AND } ($$
$$(\exists\, t)$$
$$(\text{Department}(t)$$
$$\text{AND } t.\text{dname} = \text{'Research'}$$
$$\text{AND } t.\text{dno} = d.\text{dno})$$
$$)\}$$

TRC Example (4)

- List the name and address of all employees who work for the 'Research' department

$$\{t.\text{Fname}, t.\text{Lname}, t.\text{Address} \mid \text{EMPLOYEE}(t) \text{ AND } (\exists d)(\text{DEPARTMENT}(d) \text{ AND } d.\text{Dname} = \text{'Research'} \text{ AND } d.\text{Dnumber} = t.\text{Dno})\}$$

- List the names of employees who work on some projects controlled by department number 5

$$\begin{aligned}\{e.\text{fname}, e.\text{lname} \mid \text{Employee}(e) \\ \text{AND}((\exists p)\ (\exists w) \\ (\text{Project}(p) \text{ AND } \text{Works\_on}(w) \\ \text{AND } p.\text{dnum} = 5 \text{ AND } p.\text{pnumber} = w.\text{pnum} \\ \text{AND } w.\text{essn} = e.\text{ssn}))\}\end{aligned}$$

TRC Example (4)

Employee(e)
e1, John
e2, Kate
e3, Ann

Works_on(w)
e1, p1
e2, p3
e3, p2

Project(p)
p1, 5
p2, 5
p3, 4

- Run through the employee tuples and make the second condition true, we must find tuples such that p is a Project tuple, w is a Works_on tuple, and it matches the 3 conditions with employee number matching.
  - e1 is good since you can find it in all 3 tables and meets the conditions
  - e2 is problematic because p3 = 4, which doesn't match our condition
  - e3 is also output because the combination exists that can make the second condition true

TRC Example (5)

- List the names of employees who work on all the projects controlled by department number 5

- Solution 1: Projects that are either not controlled by department 5 of e is working on

$$\{e.\text{fname}, e.\text{lname} \mid \text{Employee}(e)$$
$$\text{AND } ((\forall x)\ (\text{NOT}(Project(x))$$
$$\text{OR NOT } (x.dnum = 5)$$
$$\text{OR } ((\exists w)\ (\text{Works\_on}(w)$$
$$\text{AND } w.\text{essn} = e.\text{ssn}$$
$$\text{AND } x.\text{pnumber} = w.\text{pno})))\}$$

TRC Example (5)

- List the names of employees who work on all the projects controlled by department number 5

- Solution 2: There is no project controlled by department 5 that e is not working on

$$\{e.\text{fname}, e.\text{lname} \mid \text{Employee}(e) \\ \text{AND } (\text{NOT}(\exists x)(\text{Project}(x) \\ \text{AND } (x.\text{dnum} = 5) \\ \text{AND } (\text{NOT}(\exists w)(\text{Works\_On}(w) \\ \text{AND } w.\text{essn} = e.\text{ssn AND } x.\text{pnumber} = w.\text{pno}))))\}$$

Relational Algebra & Relational Calculus

- (Definition) Expressive power of a query language is the set of all queries that can be written using that query language
- Query language A is more expressive than query language B if the set of all queries written in A is a superset of all queries that can be written in B
- Codd's Theorem: Every relational algebra query can be expressed as a "safe" query in TRC/DRC; the converse is true
  - Relational Algebra and Relational Calculus are equally expressive

Relational Algebra & Calculus: Recap

- Relational Algebra
  - Set Functions
  - Group By Aggregate
- Relational Calculus