CS2102
Database Systems

RELATIONAL CALCULUS

Presented by: Gillian Dobbie (Gill)

Background

- The concept of relational calculus was first proposed by Edgar F. Codd. See http://en.wikipedia.org/wiki/Edgar_F._Codd

- Relational calculus is based on predicate calculus, and operates on relations

- The relational calculus is used to measure the power of relational languages. A language that can be used to produce any relation that can be derived using the relational calculus is said to be *relationally complete*.

Relational Calculus

- Relational calculus is declarative

  can describe a set of answers without being explicit about how they should be computed

  - Describes a relation in terms of one or more database relations
  - Query languages, like SQL, are similar in construction to relational calculus

- Relational algebra is procedural
  - Describes how to build a new relation from one or more relations in the database

Relational Calculus

- Basic concepts: variables, constants, comparison ops, logical connectors, quantifiers
- Two variants
  - Tuple Relational Calculus (TRC)

    - Variables range over tuples

    e.g. $\{C \mid C \in Course \wedge C.credit \geq 4\}$
  - Domain Relational Calculus (DRC)

    - Variables range over attribute domain values

    e.g. $\{<I, N, C> \mid <I, N, C> \in Course \wedge C > 4\}$

Relational Calculus

- Expressions in a calculus are called formulas
- An answer tuple is an assignment of constants to variables that make the formula evaluate to true

e.g. $\{C \mid C \in \text{Course} \wedge C.\text{credit} \geq 4\}$

Tuple Relational Calculus

- Query has the form {T | p(T)}
  - T is a tuple variable
  - p(T) is a formula that describes T
  - Result of query is the set of all tuples *t* for which the formula p(T) evaluates to true with T = *t*

e.g. $\{C \mid C \in \text{Course} \wedge C.\text{credit} \geq 4\}$

Simple TRC Queries

Lecturer

id	name
9876	Stephane Bressan
9865	Mong Li Lee
9843	Wynne Hsu
9821	Ling Tok Wang

- Find all lecturers

  $\{T \mid T \in \text{Lecturer}\}$

- Find courses where the credit is at least 4

  $\{C \mid C \in \text{Course} \wedge C.\text{credit} \geq 4\}$

Course

id	name	credit
1234	Intro to computing	2
2107	Intro to DB	4
2234	Intro to OS	2
3604	Advanced OS	4

Syntax of TRC Queries

- An atomic formula is one of the following:
  - R $\in$ Rel
  - R.*a* op S.*b*
  - R.*a* op constant

where

  - Rel is a relation name
  - R, S are tuple variables with attributes *a* and *b* respectively
  - op is an operator in the set {>, <, =, $\geq$, $\leq$, $\neq$}

e.g. $\{C \mid C \in \text{Course} \wedge C.\text{credit} \geq 4\}$

Syntax of TRC Queries

- A formula is recursively defined as one of the following:
  - Any atomic formula
  - $\neg$ p, $p \wedge q$, $p \vee q$, $p \Rightarrow q$ where p and q are formulas
  - $\exists$ R (p(R)) where R is a tuple variable and p(R) is a formula in which R appears
  - $\forall$ R (p(R)) where R is a tuple variable and p(R) is a formula in which R appears

  e.g. $\{C \mid C \in \text{Course} \wedge C.\text{credit} \geq 4\}$

Combination of Formulas

p	q	$\neg$ p	p $\wedge$ q	p $\vee$ q	p $\Rightarrow$ q
T	T	F	T	T	T
T	F	F	F	T	F
F	T	T	F	T	T
F	F	T	F	F	T

Example Database

- Sailors (sid, name, rating, age)
- Boats (bid, bname, color)
- Reserves (sid, bid, day)

Boats

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

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

Example TRC Queries

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

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

- P is a tuple variable with two fields, name and age
- Result of query is a relation with two fields, name and age
- Atomic formula P.name=S.sname and P.age=S.age give values to the fields of an answer tuple P

Example TRC Queries

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

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

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

name	age
Lubber	55
Andy	25
Rusty	35
Zorbia	16
Horatio	35

Example TRC Queries

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

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

- Compare to SQL:

```
SELECT S.sname, S.age
FROM Sailors S
WHERE S.rating > 7
```

Example TRC Queries

- Find the sailor name, boat id and reservation date for each reservation

$\{ P \mid \exists R \in \text{Reserves} \ \exists S \in \text{Sailors}$

$(R.sid = S.sid \ \wedge \ P.bid = R.bid \ \wedge \ P.date = R.date \ \wedge \ P.sname = S.sname) \}$

- For each Reserves tuple, look for a Sailors tuple with the same sid
- For each pair of such tuples, construct answer tuple P with fields sname, bid and day by copying values from the corresponding fields from the two tuples

Example TRC Queries

- Find the sailor name, boat id and reservation date for each reservation

$$\{ P \mid \exists R \in \text{Reserves}\ \exists S \in \text{Sailors}\ (R.sid = S.sid \wedge P.bid = R.bid \wedge P.date = R.date \wedge P.sname = S.sname) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

bid	date	sname
101	10/10/98	Dustin
102	10/10/98	Dustin
103	10/8/98	Dustin
104	10/7/98	Dustin
102	11/10/98	Lubber
103	11/6/98	Lubber
104	11/12/98	Lubber
101	9/5/98	Horatio
102	9/8/98	Horatio
103	9/8/98	Horatio

Example TRC Queries

- Find the names of sailors who have reserved boat 103

$\{ P \mid \exists S \in Sailors \;\; \exists R \in Reserves$

$(R.sid = S.sid \;\wedge\; R.bid = 103 \;\wedge\; P.sname = S.sname) \}$

- For each Sailors tuple, look for a tuple in Reserves that shows that this sailor has reserved boat 103
- Answer tuple P contains just one field, sname

Example TRC Queries

- Find the names of sailors who have reserved boat 103

$$\{ P \mid \exists S \in Sailors \;\; \exists R \in Reserves$$
$$(R.sid = S.sid \wedge R.bid = 103 \wedge P.sname = S.sname) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

sname
Dustin
Lubber
Horatio

Example TRC Queries

- Find the names of sailors who have reserved boat 103

$\{ P \mid \exists S \in Sailors \;\; \exists R \in Reserves$

$(R.sid = S.sid \;\wedge\; R.bid = 103 \;\wedge\; P.sname = S.sname) \}$

- Compare to SQL:

```
SELECT S.sname
FROM Sailors S, Reserves R
WHERE S.sid= R.sid AND R.bid = 103
```

Example TRC Queries

- Find the names of sailors who have reserved a red boat

$\{ P \mid \exists S \in Sailors \;\; \exists R \in Reserves \;\; \exists B \in Boats$

$(R.sid = S.sid \;\wedge\; R.bid = B.bid \wedge\; B.color = \text{‘red’} \;\wedge$
$P.sname = S.sname) \}$

- Retrieve all sailor tuples S for which there exist tuples R in Reserves and B in Boats such that S.sid=R.sid, R.bid=B.bid and B.color=‘red’

Example TRC Queries

- Find the names of sailors who have reserved a red boat

$$\{ P \mid \exists S \in Sailors \;\; \exists R \in Reserves \;\; \exists B \in Boats$$
$$(R.sid = S.sid \wedge R.bid = B.bid \wedge B.color = \text{'red'} \wedge P.sname = S.sname) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

Boats

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

sname
Dustin
Lubber
Horatio

Example TRC Queries

- Find the names of sailors who have reserved at least two boats

$$\{ P \mid \exists S \in Sailors \;\; \exists R1 \in Reserves \;\; \exists R2 \in Reserves$$
$$(S.sid = R1.sid \wedge R1.sid = R2.sid \wedge R1.bid \neq R2.bid \wedge P.sname = S.sname) \}$$

Example TRC Queries

- Find the names of sailors who have reserved at least two boats

$$\{ P \mid \exists S \in Sailors \ \exists R1 \in Reserves \ \exists R2 \in Reserves$$
$$(S.sid = R1.sid \wedge R1.sid = R2.sid \wedge R1.bid \neq R2.bid \wedge P.sname = S.sname) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

sname
Dustin
Lubber
Horatio

Example TRC Queries

- Find the names of sailors who have reserved all boats

$\{ P \mid \exists S \in Sailors \quad \forall B \in Boats \quad (\exists R \in Reserves$

$(S.sid = R.sid \ \wedge \ R.bid = B.bid \ \wedge P.sname = S.sname)) \}$

- Retrieve sailors S such that for all boats B there is a Reserves tuple showing that sailor S has reserved boat B
- This query was expressed using the division operator in relational algebra

Example TRC Queries

- Find the names of sailors who have reserved all boats

$$\{ P \mid \exists S \in Sailors \quad \forall B \in Boats \quad (\exists R \in Reserves \quad (S.sid = R.sid \wedge R.bid = B.bid \wedge P.sname = S.sname)) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

Boats

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

sname
Dustin

Example TRC Queries

- Find sailors who have reserved all red boats

$\{ S \mid S \in \text{Sailors} \wedge \forall B \in \text{Boats} \quad (B.\text{color} = \text{'red'} \Rightarrow$

$(\exists R \in \text{Reserves} (S.\text{sid} = R.\text{sid} \wedge R.\text{bid} = B.\text{bid}))) \}$

- For each candidate sailor tuple, if a boat is red, then the sailor must have reserved it.
- Return an entire sailor tuple as the answer

Example TRC Queries

- Find sailors who have reserved all red boats

$$\{ S \mid S \in \text{Sailors} \wedge \forall B \in \text{Boats} \ (B.\text{color} = \text{'red'} \Rightarrow (\exists R \in \text{Reserves} (S.\text{sid} = R.\text{sid} \wedge R.\text{bid} = B.\text{bid}))) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

Boats

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

sid	sname	rating	age
22	Dustin	7	45
31	Lubber	8	55

Example TRC Queries

- Find sailors who have reserved all red boats

$\{ S \mid S \in \text{Sailors} \wedge \forall B \in \text{Boats} \quad (B.\text{color} \neq \text{'red'} \vee$

$(\exists R \in \text{Reserves} (S.\text{sid} = R.\text{sid} \wedge R.\text{bid} = B.\text{bid}))) \}$

- 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.

Example TRC Queries

- Find sailors who have reserved all red boats

$$\{ S \mid S \in \text{Sailors} \wedge \forall B \in \text{Boats} \quad (B.\text{color} \neq \text{'red'} \vee (\exists R \in \text{Reserves} (S.\text{sid} = R.\text{sid} \wedge R.\text{bid} = B.\text{bid}))) \}$$

Sailors

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorbia	10	16
74	Horatio	9	35
85	Art	3	25
95	Bob	3	63

Reserves

sid	bid	date
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

Boats

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

sid	sname	rating	age
22	Dustin	7	45
31	Lubber	8	55

Domain Relational Calculus

- Query has the form $\{<x_1, \ldots, x_n> \mid p(x_1, \ldots, x_n)\}$
  - Each x_i is a domain variable or a constant
  - $p(x_1, \ldots, x_n)$ is a DRC formula
  - Result of query is the set of all tuples $<x_1, \ldots, x_n>$ for which the formula evaluates to true

Syntax of DRC Queries

- An atomic formula in DRC is one of the following:
  - $<x_1, \ldots, x_n> \in$ Rel
  - X op Y
  - X op constant

where

- Rel is a relation with n attributes
- Each x_i is a variable or a constant
- X, Y are domain variables
- op is an operator in the set {>, <, =, ≥, ≤, ≠}

Syntax of DRC Queries

- A formula in DRC is recursively defined as one of the following:
  - Any atomic formula
  - $\neg$ p, $p \wedge q$, $p \vee q$, $p \Rightarrow q$ where p and q are formulas
  - $\exists$ X (p(X)) where X is a domain variable and p(X) is a formula in which X appears
  - $\forall$ X (p(X)) where X is a domain variable and p(X) is a formula in which X appears

Example DRC Queries

- Find all sailors with a rating above 7

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

I = id, N = name, T = tel.no, A = address (all of these are columns of some relation)

- Each attribute is given a variable name
- Condition $\langle I, N, T, A\rangle \in Sailors$ ensures that domain variables are restricted to the fields of the same tuple
- Compare to TRC

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

  - Specify T > 7 instead of S.rating > 7
  - Specify tuple $\langle I, N, T, A\rangle$ in result instead of S

Example DRC Queries

- Find the names of sailors who have reserved boat 103

$\{ \langle N \rangle \mid \exists I, T, A \ (\langle I, N, T, A \rangle \in Sailors \wedge$

$\exists I_R, B_R, D \ (\langle I_R, B_R, D \rangle \in Reserves \ \wedge \ I = I_R \wedge B_R = 103)) \}$

- Only the sname field is retained in answer
- Use $\exists I_R, B_R, D$ or $\exists I_R \exists B_R \exists D$ or $\exists \langle I_R, B_R, D \rangle \in Reserves$
- Simplify query to

$\{ \langle N \rangle \mid \exists I, T, A \ (\langle I, N, T, A \rangle \in Sailors \wedge$

$\exists D \ (\langle I, 103, D \rangle \in Reserves)) \}$

Example DRC Queries

- Find the names of sailors who have reserved a red boat

$\{ \langle N \rangle \mid \exists I, T, A \ (\langle I, N, T, A \rangle \in Sailors \ \wedge$

$\exists B, D \ (\langle I, B, D \rangle \in Reserves \wedge$

$\exists BN \ (\langle B, BN, \text{'red'} \rangle \in Boats \ ) \ ) \ ) \}$

- Alternatively,

$\{ \langle N \rangle \mid \exists I, T, A, B, D, BN \ (\langle I, N, T, A \rangle \in Sailors \ \wedge$

$\langle I, B, D \rangle \in Reserves \wedge \langle B, BN, \text{'red'} \rangle \in Boats \ ) \}$

Example DRC Queries

- Find the names of sailors who have reserved at least two boats

$$\{ \langle N \rangle \mid \exists I, T, A \ (\langle I, N, T, A \rangle \in \text{Sailors} \ \wedge \exists B1, B2, D1, D2 \ (\langle I, B1, D1 \rangle \in \text{Reserves} \wedge \langle I, B2, D2 \rangle \in \text{Reserves} \wedge B1 \neq B2)) \}$$

Example DRC Queries

- Find the names of sailors who have reserved all boats

{ <N> | ∃ I, T, A (<I, N, T, A> ∈ Sailors ∧

for all B, BN, C (<B, BN, C> *round e sign* Boats => (*there exists* D (<I, B, D> *round e sign* Reserves)) }

notice that this notation is a bit different. Here, the B, BN, C inside this bracket are not arbitrary, unlike previous examples

Example DRC Queries

- Find sailors who have reserved all red boats

$$\{ \langle N \rangle \mid \exists I, T, A \ (\langle I, N, T, A \rangle \in \text{Sailors} \ \wedge \ \forall (B, BN, C) \in \text{Boats} \ (C = \text{'red'} \Rightarrow \exists \langle I, B, D \rangle \in \text{Reserves})) \}$$

- Find all sailors such that, for every red boat, there is a tuple in Reserves that shows the sailor has reserved it.

Safety of Queries in Relational Calculus

- A query is **safe** if the evaluation result is a finite set of tuples; otherwise, it is **unsafe**.
- Query {S | ¬ (S ∈ Sailors)} is unsafe
  - Ask for all tuples S such that S is not in Sailors
  - Answer set is infinite

Summary

- Relational calculus is a non-procedural formal query language
- Relational calculus has the same expressive power as relational algebra
  - Every query that can be expressed in relational algebra can be expressed as a safe query in TRC/DRC; and vice versa
- A query language (e.g. SQL) is relationally complete if it can express every query that is expressible in relational algebra/calculus

RELATIONAL ALGEBRA	RELATIONAL CALCULUS
It is a procedural method of solving the queries.	It is a non-procedural method of solving the queries.
We specify the sequence of operations to perform a particular request.	We specify the only what is required without bothering about the sequence of operations to perform that request.
It is prescriptive or rigid in nature i.e. it describes steps to perform a given task.	It is descriptive or straightforward in nature i.e. describe desired result.
The evaluation of the query depends upon the order of operations.	It does not depend on the order of operations.
It specifies operations performed on existing relations to obtain new relations.	Operations are directly performed on the relations in form of formulas.
It is more closely asscciated with a programming language.	It is more closely associated with a natural language.
The solution to the database access problem using a relational algebra is obtained by stating what is required and what are the steps to obtain that information.	The solution to the database access problem using a relational calculus is obtained simply by stating what is required and letting the system find the answer.
It is used as a vehicle for implementation of Relational Calculus.	Relational Calculus queries are converted into equivalent relational algebra format by using Codd's Reduction Algorithm and then it is implemented with the help of relational algebra operators.
Relational algebra operators are used as a yardstick for measuring the expressive power of any given language.	A language is said to be complete if it is at least as powerful as the calculus that is, if any relation definable by some expression of the calculus is also definable by some expression of the language in question.
The queries are domain independent.	The queries are domain dependent.

From http://ecomputernotes.com/database-system/rdbms/relational-algebra-and-relational-calculus