

RELATIONAL ALGEBRA & CALCULUS

CS 564- Fall 2015

ACKs: Dan Suciu, Jignesh Patel, AnHai Doan

TODAY

- Relational Algebra
 - Group-By Aggregate
- Relational Calculus
- Examples

GROUP BY AGGREGATE

- is part of the so-called **extended RA**
- helps us to compute counts, sums, min, max, ...

Examples

- What is the average age of the customers?
- How many people bought an iPad?

GROUP BY AGGREGATE

Notation: $\gamma_{X, Agg(Y)}(R)$

- **group by** the attributes in X
- **aggregate** the attribute in Y
 - SUM, COUNT, AVG (average), MIN, MAX
- Output schema: X + an extra (numerical) attribute

EXAMPLE

Person

SSN	name	age
934729837	Paris	24
123123645	John	30
384475687	Arun	21



$\gamma_{AVG(age)}(Person)$

AVG(age)
25

EXAMPLE

Person

SSN	name	age	phoneNumber
934729837	Paris	24	608-374-8422
934729837	Paris	24	603-534-8399
123123645	John	30	608-321-1163
384475687	Arun	21	206-473-8221

↓ $\gamma_{SSN, COUNT(phoneNumber)}(Person)$

SSN	COUNT(phoneNumber)
934729837	2
123123645	1
384475687	1

RELATIONAL CALCULUS

RELATIONAL QUERY LANGUAGES

- **Declarative**: Tuple Relational Calculus (**TRC**)
Domain Relational Calculus (**DRC**)
 - describe what a user wants, rather than how to compute it
- **Procedural**: Relational Algebra (**RA**)
 - operational, useful for representing execution plans

RELATIONAL CALCULUS

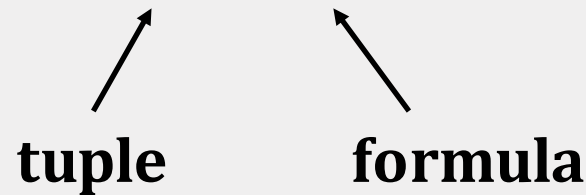
- declarative query language
- simple subset of First-Order Logic
 - **TRC**: variables range over *tuples*
 - **DRC**: variables range over *domain elements*

Example

- RA: $\sigma_{age>24}(Person)$
- TRC: $\{t \mid t \in Person \wedge t.age > 24\}$

TUPLE RELATIONAL CALCULUS

A query in TRC has the form: $\{t \mid \phi(t)\}$



- The query returns all tuples t for which the formula $\phi(t)$ evaluates to **true**
- A formula is **recursively** defined, starting with simple *atomic formulas*, and building more complex formulas using the logical operators

TRC FORMULA

- an **atomic formula** can be:
 - $t \in R$
 - $s.A \text{ op } t.B$
 - $s.A \text{ op } \textit{constant}$
- a formula is **recursively** defined from formulas p, q :
 - $\neg p, p \wedge q, p \vee q, p \implies q$
 - $\exists t (p(t))$
 - $\forall t (p(t))$

TRC EXAMPLE

Sailors (sid, sname, rating, age)

Reserves (sid, bid, day)

Boats (bid, bname, color)

- $\sigma_{rating > 1 \wedge age > 30}(Sailors)$
- $\{t \mid t \in Sailors \wedge t.rating > 1 \wedge t.age > 30\}$



atomic formulas

FREE AND BOUND VARIABLES

Let the TRC query $\{t \mid \phi(t)\}$

- The use of quantifiers $\forall t, \exists t$ in a formula **binds** the variable t
 - a variable that is not bound is **free**
- The variable t that appears to the left of \mid must be the only free variable in the formula $\phi(t)$

TRC EXAMPLE

Sailors (sid, sname, rating, age)

Reserves (sid, bid, day)

Boats (bid, bname, color)

- $\pi_{sname}(\sigma_{rating>1}(Sailors))$
- $\{t \mid \exists s \in Sailors (s.rating > 1 \wedge t.sname = s.sname)\}$

↑ ↘
free **bound**

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

CODD'S THEOREM

- **Codd's Theorem:** Every RA query can be expressed as a **safe** query in TRC/DRC; the converse is also true
- A query language is *relationally complete* if it can express every query that is expressible in RA/RC

MORE EXAMPLES

Sailors (sid, sname, rating, age)

Reserves (sid, bid, day)

Boats (bid, bname, color)

Q1: What are the names of the sailors who have reserved boat #100?

$\{t \mid \exists s \in \text{Sailors}, \exists r \in \text{Reserves} (s.sid = r.sid \wedge r.bid = 100 \wedge t.sname = s.sname)\}$

MORE EXAMPLES

Sailors (sid, sname, rating, age)

Reserves (sid, bid, day)

Boats (bid, bname, color)

Q2: Find the names of the sailors who have reserved **at least two different** boats

Q3: Find the names of the sailors who have reserved **all '470'** boats

MORE EXAMPLES

Product (pid, name, price, category, maker-cid)

Purchase (buyer-ssn, seller-ssn, store, pid)

Company (cid, name, country)

Person (ssn, name, phone, city)

Q4: Find the phone numbers of people who bought iPads from Fred (the salesman)

Q5: Find the names of people who bought products from the USA

MORE EXAMPLES

Product (pid, name, price, category, maker-cid)

Purchase (buyer-ssn, seller-ssn, store, pid)

Company (cid, name, country)

Person (ssn, name, phone, city)

Q6: Find the names of people who bought products from the USA, but not from Greece

Q7: Find the names of people who bought products from the USA, and live in Madison

MORE EXAMPLES (AGGREGATE)

Product (pid, name, price, category, maker-cid)

Purchase (buyer-ssn, seller-ssn, store, pid)

Company (cid, name, country)

Person (ssn, name, phone, city)

Q8: Find the total value of products sold by Fred (the salesman)

Q9: Find the average price of computers made in Greece

MORE EXAMPLES: BEERS

Likes (drinker, beer)

Frequents (drinker, bar)

Serves (bar, beer)

- Find the bars that serve all beers that Fred likes

MORE EXAMPLES: BEERS

Likes (drinker, beer)

Frequents (drinker, bar)

Serves (bar, beer)

- *Average drinker*: frequents some bar that serves some beer they like
- *Prudent drinker*: frequents only bars that serve some beer they like
- *Paranoid drinker*: frequents only bars that serve only beers they like