Database Applications (15-415)

Relational Calculus Lecture 6, January 26, 2016

Mohammad Hammoud



Today...

Last Session:

- Relational Algebra
- Today's Session:
 - Relational calculus
 - Relational tuple calculus

Announcements:

- PS2 is now posted. Due on Feb 07, 2016 by midnight
- PS1 grades are out
- In the next recitation we will practice on relational algebra and calculus



Outline

- Relational Tuple Calculus (RTC)
 - Why?
 - Details
 - Examples
 - Equivalence with relational algebra
 - 'Safety' of expressions



Motivation

Question: What is the main "weakness" of relational algebra?

- Answer: Procedural
 - It describes the steps (i.e., 'how')
 - Still useful, especially for query optimization



Relational Calculus (in General)

- It describes what we want (not how)
- It has two equivalent flavors, 'tuple' and 'domain' calculus
 - We will only focus on relational 'tuple' calculus
- It is the basis for SQL and Query By Example (QBE)
- It is useful for proofs (see query optimization, later)



Relational Tuple Calculus (RTC)

RTC is a subset of 'first order logic':



A "formula" that describes t

Give me tuples 't', satisfying predicate 'P'

Examples:

- Find all students: $\{t \mid t \in STUDENT\}$
- Find all sailors with a rating above 7:

 $\{t \mid t \in Sailors \land t.rating > 7\}$



Syntax of RTC Queries

The allowed symbols:

Quantifiers:

∀, ∃



Syntax of RTC Queries

Atomic "formulas":

 $t \in TABLE$ t.attr op const t.attr op s.attr

Where *op* is an operator in the set $\{<, >, =, \leq, \geq, \neq\}$



Syntax of RTC Queries

- A "formula" is:
 - Any atomic formula
 - If P1 and P2 are formulas, so are

 $\neg P1; \neg P2; P1 \land P2; P1 \lor P2; P1 \Rightarrow P2$

If P(s) is a formula, so are

 $\exists s(P(s)) \\ \forall s(P(s))$



Basic Rules

- Reminders:
 - De Morgan: $P1 \wedge P2 \equiv \neg(\neg P1 \vee \neg P2)$
 - Implication: $P1 \Rightarrow P2 \equiv \neg P1 \lor P2$
 - Double Negation:

 $\forall s \in TABLE \ (P(s)) \equiv \neg \exists s \in TABLE \ (\neg P(s))$

'every human is mortal : no human is immortal'



A Mini University Database

STUDENT			CLASS		
<u>Ssn</u>	Name	Address	<u>c-id</u>	c-name	units
123	smith	main str	15-413	s.e.	2
234	jones	forbes ave	15-412	0.S.	2

TAKES					
<u>SSN</u>	<u>c-id</u>	grade			
123	15-413	Α			
234	15-413	B			



Find all student records

$$\{t \mid t \in STUDENT\}$$

outputtupleof type 'STUDENT'



Find the student record with ssn=123



Find the student record with ssn=123

$$\{t \mid t \in STUDENT \land t.ssn = 123\}$$

This is equivalent to the 'Selection' operator in Relational Algebra!



• Find the **name** of the student with ssn=123

$$\{t \mid t \in STUDENT \land t.ssn = 123\}$$

Will this work?



• Find the **name** of the student with ssn=123

{
$$t \mid \exists s \in STUDENT(s.ssn = 123 \land t.name = s.name)$$
}

This is equivalent to the 'Projection' operator in Relational Algebra!



Get records of both part time and full time students*

$$\{t \mid t \in FT_STUDENT \lor t \in PT_STUDENT\}$$

This is equivalent to the 'Union' operator in Relational Algebra!

* Assume we maintain tables for PT_STUDENT and FT_STUDENT in our Mini University DB



Find students that are not staff*

$\{t \mid t \in STUDENT \land \\ t \notin STAFF\}$

This is equivalent to the 'Difference' operator in Relational Algebra!

* Assume we maintain a table for STAFF in our Mini University DB and that STUDENT and STAFF are union-compatible

جا ہجۃ کارنیجی ہیلوں ہیں ہطر Carnegie Mellon University Qatar

Cartesian Product: A Reminder

Assume MALE and FEMALE dog tables as follows:

			I	M.Name	F.Name
MALE		FEMALE		spike	lassie
<u>name</u>	X	<u>name</u>	=	spike	shiba
spike	∇	lassie		spot	lassie
spot		shiba		spot	shiba

This gives all possible couples!



Examples (Cont'd)

Find all the pairs of (male, female) dogs

$$\{t \mid \exists m \in MALE \land \\ \exists f \in FEMALE \\ (t.m-name = m.name \land \\ t.f-name = f.name)\}$$

This is equivalent to the 'Cartesian Product' operator in Relational Algebra!



Find the names of students taking 15-415

STUDEN	Г			CLA	ASS		
<u>Ssn</u>	Name	Add	dress	<u>c-id</u>		c-name	units
12	3 smith	mai	in str	15-4	113	s.e.	2
23	l jones	for	forbes ave		112	0.S.	2
	Junes		Jes ave	15-2	+ ∠	0.3.	Ľ
			JES ave	13-2	+12	0.3.	L
		TAKES		10-2	+12	0.3.	L
way Join				grade	+12	0.3.	

234 15-413

B

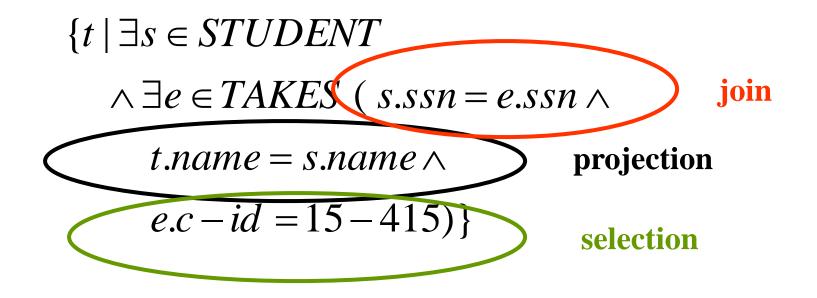
جامعۃ کارنی جے میلوں فی قطر Carnegie Mellon University Qatar

Find the names of students taking 15-415

 $\{t \mid \exists s \in STUDENT \\ \land \exists e \in TAKES \ (s.ssn = e.ssn \land \\ t.name = s.name \land \\ e.c - id = 15 - 415)\}$

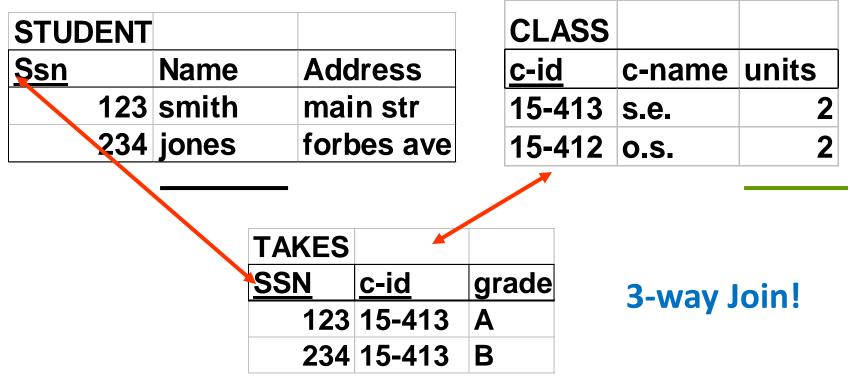


Find the names of students taking 15-415





Find the names of students taking a 2-unit course





• Find the names of students taking a 2-unit course

$$\{t \mid \exists s \in STUDENT \land \exists e \in TAKES \\ \exists c \in CLASS(s,sn = e,sn \land join \\ e,c - id = c,c - id \land \\ t.name = s.name \land \\ c.units = 2)\}$$
 projection selection

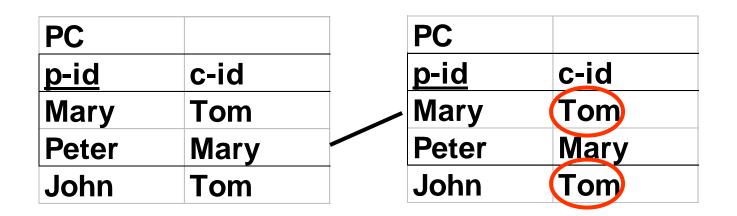
هـ خ صار نبـ حـى مبلون فى قـطر

Carnegie Mellon University Qatar

What is the equivalence of this in Relational Algebra?

More on Joins

Assume a Parent-Children (PC) table instance as follows:



Who are Tom's grandparent(s)? (this is a self-join)



More Join Examples

Find Tom's grandparent(s)

$$t \mid \exists p \in PC \land \exists q \in PC$$

(p.c-id = q.p-id \land
p.p-id = t.p-id \land
q.c-id = "Tom")}

What is the equivalence of this in Relational Algebra?



Harder Examples: DIVISION

Find suppliers that shipped all the bad parts

SHIPMENT					
<u>s#</u> s1	<u>p#</u>		BAD_P		BAD_S
s1	p1		<u>p#</u>		
s2	p1	•		—	<u>s#</u> s1
s1	p2		p1 p2		SI
s3	p1		μΖ		
s5	р3				



Harder Examples: DIVISION

Find suppliers that shipped all the bad parts

 $\{t \mid \forall p (p \in BAD _ P \Longrightarrow) ($ $\exists s \in SHIPMENT($ $t.s \# = s.s \# \land$ $s.p \# = p.p \#)))\}$



General Patterns

There are three equivalent versions:
1) If it is bad, he shipped it

 $\{t \,|\, \forall p(p \in BAD_P \Longrightarrow (P(t))\}$

2) Either it was good, or he shipped it

 $\{t \mid \forall p (p \notin BAD P \lor (P(t)))\}$

3) There is no bad shipment that he missed $\{t \mid \neg \exists p (p \in BAD P \land (\neg P(t)))\}$

More on Division

Find (SSNs of) students who are taking all the courses that ssn=123 is (and maybe even more)

One way to think about this: Find students 's' so that if 123 takes a course => so does 's'



More on Division

Find (SSNs of) students who are taking all the courses that ssn=123 is (and maybe even more)

 $\{o \mid \forall t((t \in TAKES \land t.ssn = 123) \Rightarrow \exists t1 \in TAKES (\\ t1.c - id = t.c - id \land \\ t1.ssn = o.ssn) \}$



'Proof' of Equivalence

Relational Algebra <-> RTC

But...



Safety of Expressions

• FORBIDDEN:



It has infinite output!!

Instead, always use:

 $\{t \mid \dots t \in SOME - TABLE\}$



Summary

- The relational model has rigorously defined query languages — simple and powerful
- Relational algebra is more operational/procedural
 - Useful as internal representation for query evaluation plans
- Relational calculus is declarative
 - Users define queries in terms of what they want, not in terms of how to compute them

Summary

- Several ways of expressing a given query
 - A *query optimizer* should choose the most efficient version
- Algebra and "safe" calculus have the same expressive power
 - This leads to the notion of *relational completeness*



Next Class

SQL-Part I

