

Carnegie Mellon Univ.
Dept. of Computer Science
15-415 - Database Applications

C. Faloutsos Lecture#5: *Relational calculus*



CMU SCS

General Overview - rel. model

- · history
- concepts
- Formal query languages
 - relational algebra
 - rel. tuple calculus
 - rel. domain calculus

Faloutso

CMU SCS 15-415

#2



CMU SC

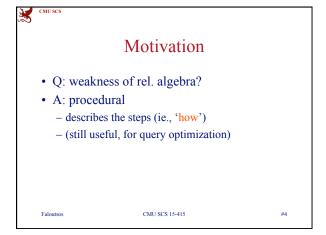
Overview - detailed

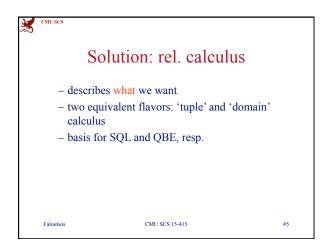
- rel. tuple calculus
 - why?
 - details
 - examples
 - equivalence with rel. algebra
 - more examples; 'safety' of expressions
- re. domain calculus + QBE

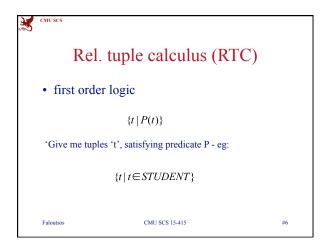
Faloutsos

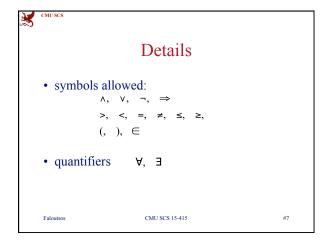
CMU SCS 15-415

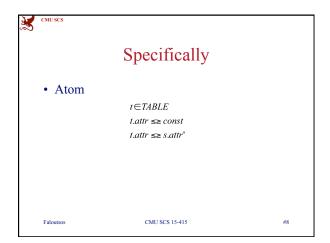
#3

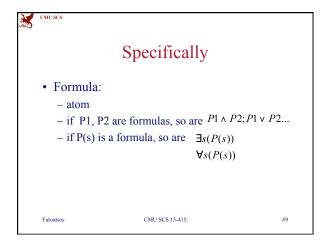


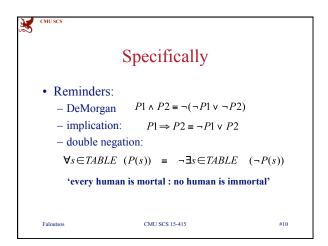


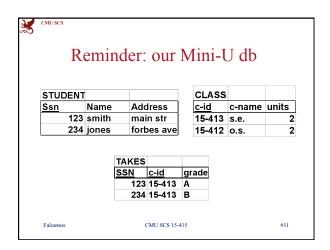


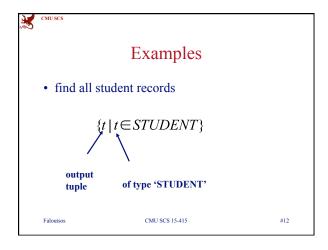










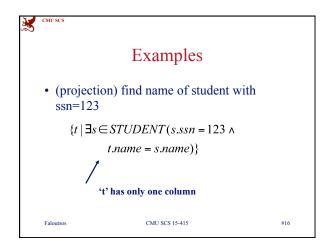


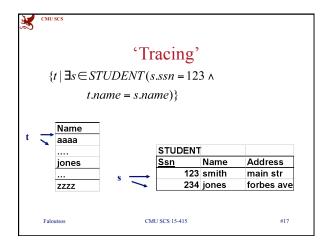
×	CMU SCS
(S)	_ :
	Examples
	• (selection) find student record with ssn=123
	(Selection) find student record with ssir 125
	Faloutsos CMU SCS 15-415 #13
×	CMU SCS
	Examples
	Examples
	• (selection) find student record with ssn=123
	$\{t \mid t \in STUDENT \land t.ssn = 123\}$
	Faloutsos CMU SCS 15-415 #14
×	CMU SCS
	Examples
	Examples
	• (projection) find name of student with
	ssn=123
	$\{t \mid t \in STUDENT \land t.ssn = 123\}$

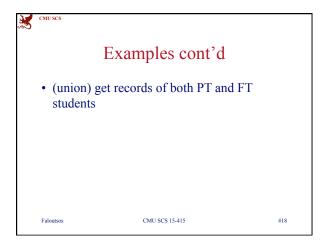
CMU SCS 15-415

Faloutsos

#15







#19

• (union) get is students $\{t \mid t \in F\}$	xamp				
students	records	of both	PT.	d T	
$\{t \mid t \in F\}$				and F	T
	T_ST	UDEN:	T	٧	
$t \in P$	T_ST	UDENT	<i>[</i>]		

CMU SCS 15-415

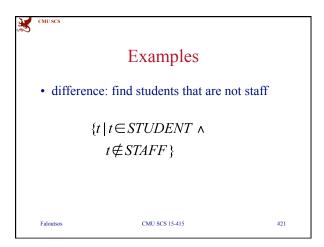
Faloutsos

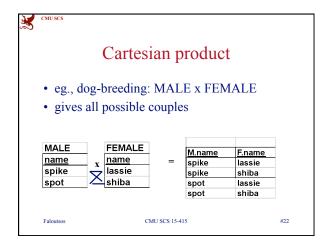
Examples

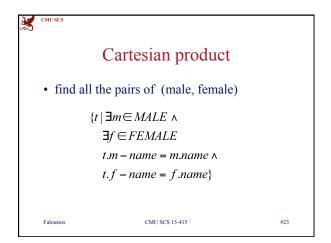
• difference: find students that are not staff

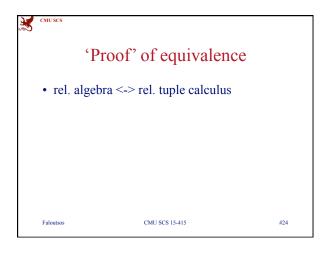
(assuming that STUDENT and STAFF are union-compatible)

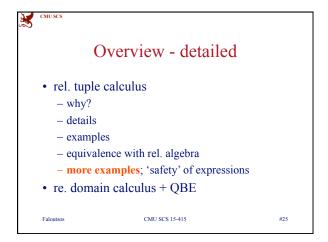
Faloutsos CMU SCS 15-415 #20

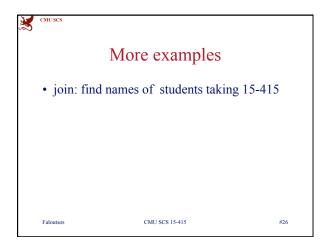


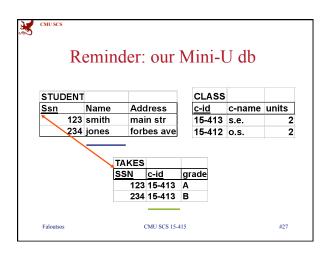


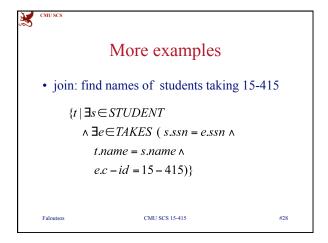






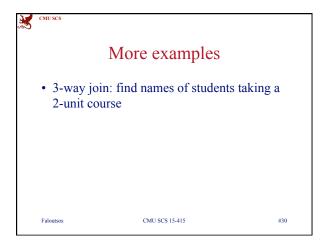


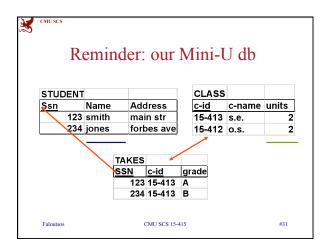


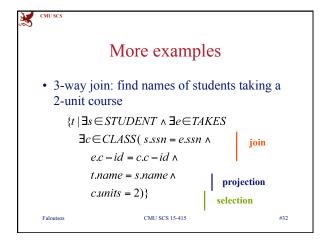


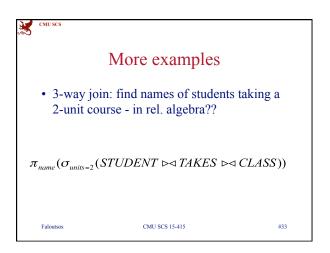
More examples

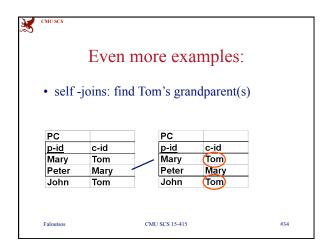
• join: find names of students taking 15-415 $\{t \mid \exists s \in STUDENT \\ \land \exists e \in TAKES \quad (s.ssn = e.ssn \land) \text{ join}$ $t.name = s.name \land \text{ projection}$ $e.c - id = 15 - 415)\}$ selection

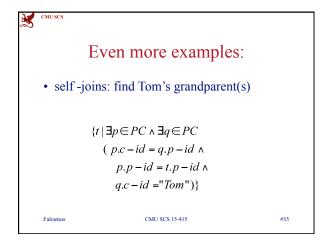


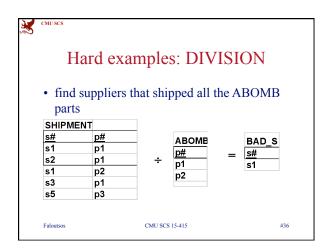


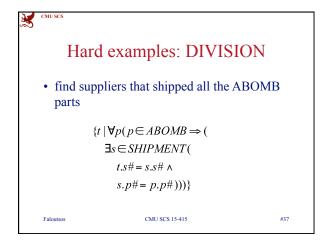








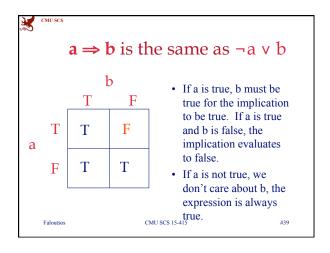




General pattern

• three equivalent versions:

- 1) if it's bad, he shipped it $\{t \mid \forall p(p \in ABOMB \Rightarrow (P(t)))\}$ - 2) either it was good, or he shipped it $\{t \mid \forall p(p \notin ABOMB \lor (P(t)))\}$ - 3) there is no bad shipment that he missed $\{t \mid \neg \exists p(p \in ABOMB \land (\neg P(t)))\}$



#40

#41

#42



More on division

• find (SSNs of) students that take all the courses that ssn=123 does (and maybe even more)

find students 's' so that if 123 takes a course => so does 's'

Faloutsos

CMU SCS 15-415



CMU SCS

More on division

• find students that take all the courses that ssn=123 does (and maybe even more)

```
\{o \mid \forall t((t \in TAKES \land t.ssn = 123) \Rightarrow \\ \exists t 1 \in TAKES(\\ t1.c - id = t.c - id \land \\ t1.ssn = o.ssn)\\)\}
```

Faloutsos

CMU SCS 15-415



CMU SC

Safety of expressions

• FORBIDDEN:



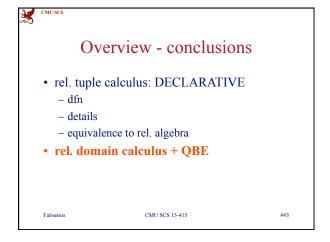
It has infinite output!!

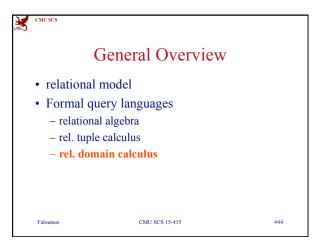
• Instead, always use

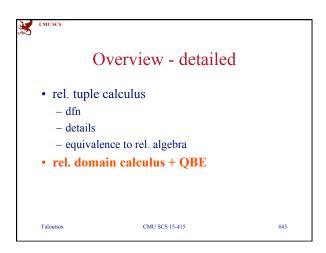
 $\{t \midt \in SOME - TABLE\}$

Faloutsos

CMU SCS 15-415



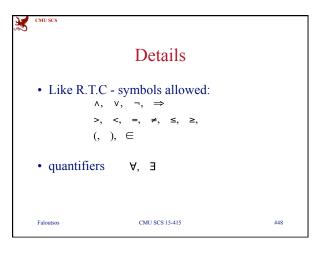


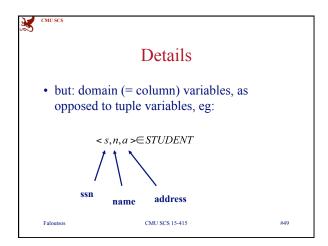


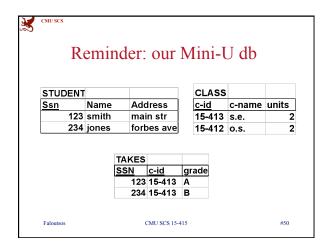
	Rel. domain calculus (RDC)
•	Q: why?
•	A: slightly easier than RTC, although equivalent - basis for QBE.
•	idea: domain variables (w/ F.O.L.) - eg:
•	'find STUDENT record with ssn=123'

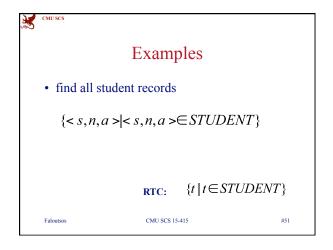
Rel. Dom. Calculus

• find STUDENT record with ssn=123' $\{ < s, n, a > | < s, n, a > \in STUDENT \land s = 123 \}$







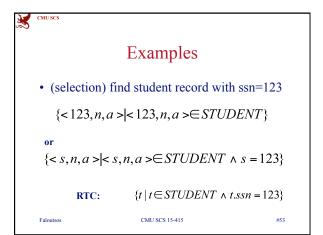


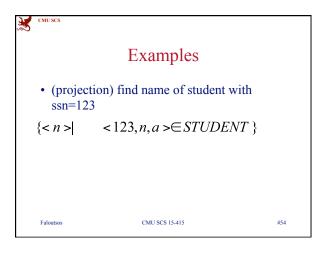
#52

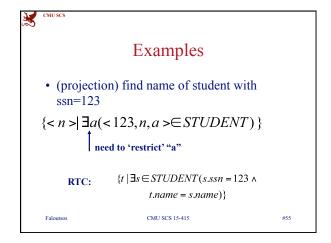
	CMU SCS		
Examples			
	• (selection) find student record with ssn=123		

CMU SCS 15-415

Faloutsos







CMU SC

Examples cont'd

• (union) get records of both PT and FT students

RTC: $\{t \mid t \in FT_STUDENT \lor t \in PT_STUDENT\}$ utsos CMU SCS 15-415 #



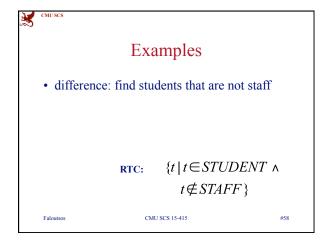
Examples cont'd

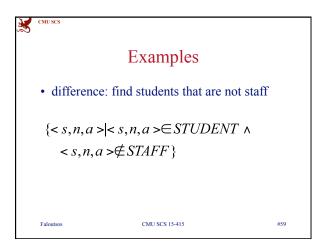
• (union) get records of both PT and FT students

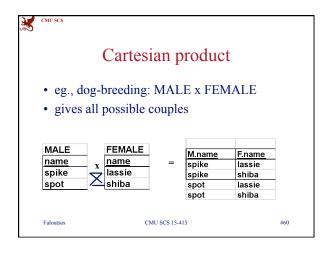
 ${\langle s, n, a \rangle | \langle s, n, a \rangle \in FT_STUDENT \lor }$ ${\langle s, n, a \rangle \in PT_STUDENT \rangle}$

#57

Faloutsos CMU SCS 15-415

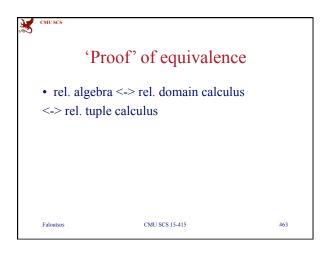


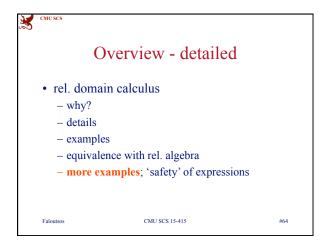


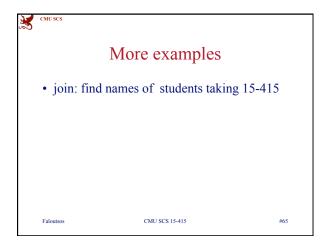


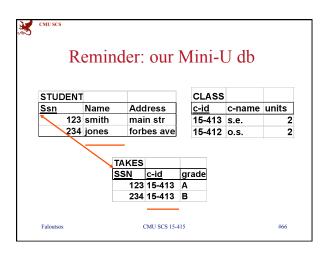
**	CMU SCS			
	Cartesian product			
	• find all the pairs of (male, female) - RTC:			
	$\{t \mid \exists m \in MALE$	٨		
	$\exists f \in FEMALE$			
	$t.m-name=m.name \land$			
	$t.f$ – $name = f.name$ }			
	Faloutsos CMU SCS	15-415	#61	

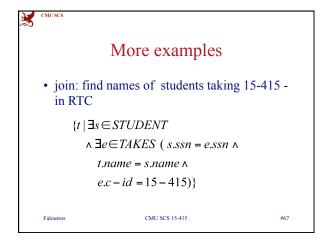
3	Cito Ses				
		Cartesian product			
	• find	d all the pairs of (male, female) - RDC:			
	$\{< m, f > \mid < m > \in MALE \land < f > \in FEMALE\}$				
	Faloutsos	CMU SCS 15-415 #6	62		

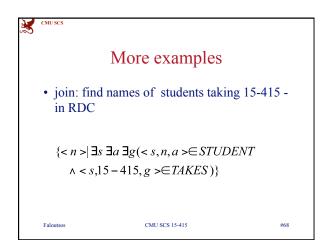


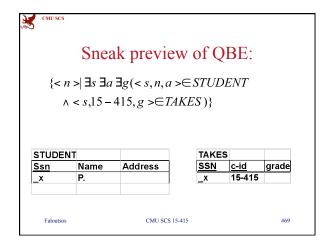


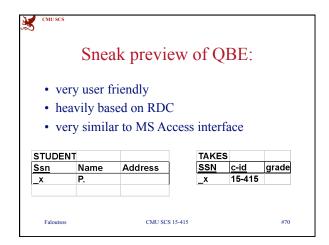


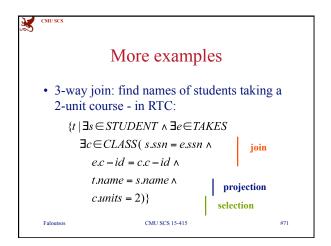


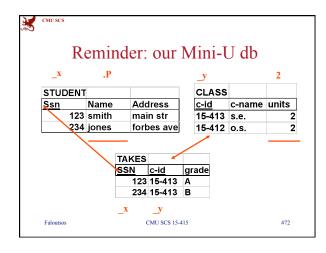


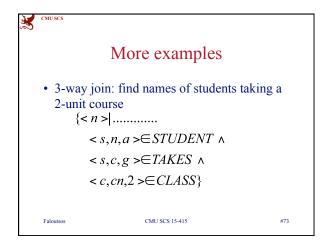


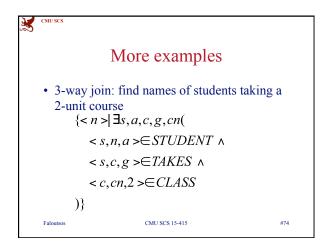


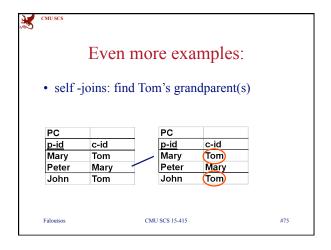










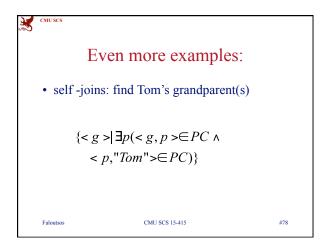


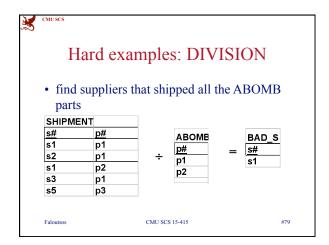
CMU SC	cs .			
Even more examples: • self-joins: find Tom's grandparent(s)				
Falout	sos CMU SCS 15-415	#76		

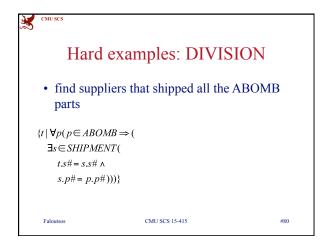
Even more examples:

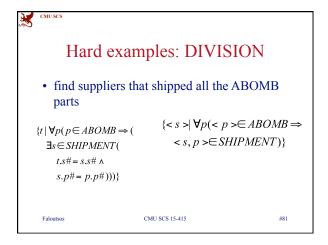
• self -joins: find Tom's grandparent(s) $\{t \mid \exists p \in PC \land \exists q \in PC \\ (p.c - id = q.p - id \land p.p - id = t.p - id \land p.p - id = t.p - id \land q.c - id = "Tom")\}$ Faloutsos

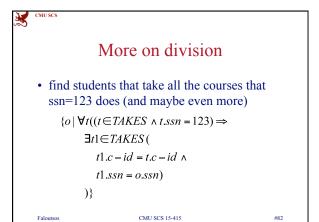
CMU SCS 15-415 $\{ < g > \mid \exists p (< g, p > \in PC \land < p, "Tom" > \in PC) \}$











CMU SC

More on division

• find students that take all the courses that ssn=123 does (and maybe even more)

```
\{ \langle s \rangle | \forall c (\exists g (\langle 123, c, g \rangle \in TAKES)) \Rightarrow \exists g'(\langle s, c, g' \rangle) \in TAKES) \}
```

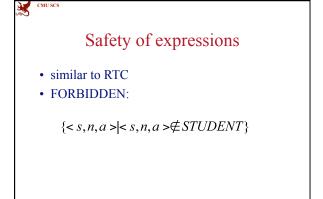
Faloutsos

Faloutsos

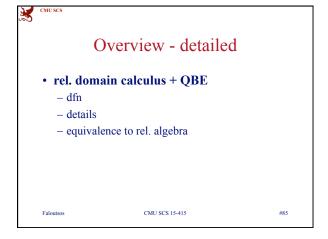
CMU SCS 15-415

#83

#84



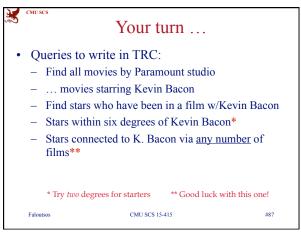
CMU SCS 15-415

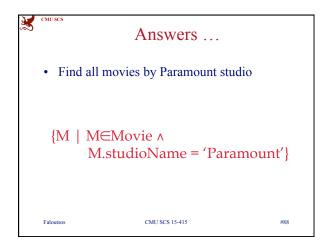


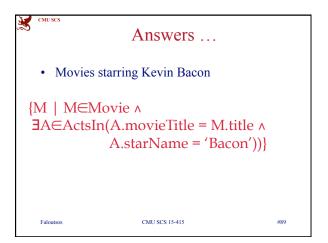
Fun Drill: Your turn ...

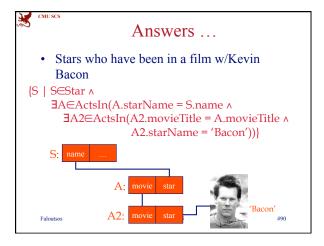
Schema:
Movie(title, year, studioName)
ActsIn(movieTitle, starName)
Star(name, gender, birthdate, salary)

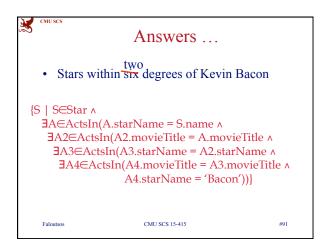
Faloutsos CMU SCS 15-415 #86

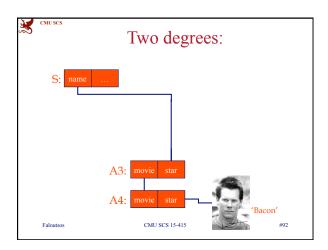


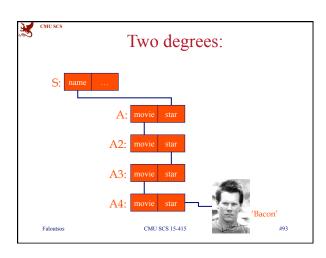














Answers ...

• Stars connected to K. Bacon via any number of films

- Sorry ... that was a trick question
 - Not expressible in relational calculus!!
- What about in relational algebra?
 - We will be able to answer this question shortly ...



Expressive Power

- Expressive Power (Theorem due to Codd):
 - Every query that can be expressed in relational algebra can be expressed as a safe query in DRC / TRC; the converse is also true.
- Relational Completeness:

Query language (e.g., SQL) can express every query that is expressible in relational algebra/calculus. (actually, SQL is more powerful, as we will see...) CMU SCS 15-415

Faloutsos



Question:

- Can we express previous query ('any # steps') in relational algebra?
- A: If we could, then by Codd's theorem we could also express it in relational calculus. However, we know the latter is not possible, so the answer is no.

CMU SCS 15-415



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 it.

Faloutsos CMU SCS 15-415



Summary - cnt'd

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

Faloutsos

CMU SCS 15-415

#98