


CMU SCS

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

Faloutsos CMU SCS 15-415 #2




CMU SCS

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

Motivation

- Q: weakness of rel. algebra?
- A: procedural
 - describes the steps (ie., 'how')
 - (still useful, for query optimization)

Faloutsos CMU SCS 15-415 #4




CMU SCS

Solution: rel. calculus

- describes **what** we want
- two equivalent flavors: 'tuple' and 'domain' calculus
- basis for SQL and QBE, resp.

Faloutsos CMU SCS 15-415 #5



CMU SCS

Rel. tuple calculus (RTC)


- first order logic

$$\{t \mid P(t)\}$$

'Give me tuples 't', satisfying predicate P - eg:

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

Faloutsos CMU SCS 15-415 #6



CMU SCS

Details


- symbols allowed:

$$\wedge, \vee, \neg, \Rightarrow$$

$$>, <, =, \neq, \leq, \geq,$$

$$(,), \in$$
- quantifiers \forall, \exists

Faloutsos CMU SCS 15-415 #7



CMU SCS

Specifically


- Atom

$$t \in \text{TABLE}$$

$$t.attr \leq \text{const}$$

$$t.attr \leq s.attr'$$

Faloutsos CMU SCS 15-415 #8



CMU SCS

Specifically

- Formula:
 - atom
 - if $P1, P2$ are formulas, so are $P1 \wedge P2; P1 \vee P2 \dots$
 - if $P(s)$ is a formula, so are $\exists s(P(s))$
 $\forall s(P(s))$

Faloutsos CMU SCS 15-415 #9

CMU SCS

Specifically

- Reminders:
 - DeMorgan $P1 \wedge P2 \equiv \neg(\neg P1 \vee \neg P2)$
 - implication: $P1 \Rightarrow P2 \equiv \neg P1 \vee P2$
 - 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’

Faloutsos CMU SCS 15-415 #10

CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

Faloutsos CMU SCS 15-415 #11

CMU SCS

Examples


- find all student records

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

output
tuple

of type ‘STUDENT’

Faloutsos CMU SCS 15-415 #12




CMU SCS

Examples

- (selection) find student record with ssn=123

Faloutsos CMU SCS 15-415 #13




CMU SCS

Examples

- (selection) find student record with ssn=123

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

Faloutsos CMU SCS 15-415 #14



CMU SCS

Examples

- (projection) find **name** of student with ssn=123

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

Faloutsos CMU SCS 15-415 #15

CMU SCS

Examples

- (projection) find name of student with ssn=123

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

↗

‘t’ has only one column

Faloutsos CMU SCS 15-415 #16

CMU SCS

‘Tracing’

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

t

Name
aaaa
...
jones
...
zzzz

s

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave


Faloutsos CMU SCS 15-415 #17

CMU SCS

Examples cont’d

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

Faloutsos CMU SCS 15-415 #18




CMU SCS

Examples cont'd

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

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

Faloutsos CMU SCS 15-415 #19




CMU SCS

Examples

- difference: find students that are not staff

(assuming that STUDENT and STAFF are union-compatible)

Faloutsos CMU SCS 15-415 #20



CMU SCS

Examples

- difference: find students that are not staff

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

Faloutsos CMU SCS 15-415 #21

CMU SCS

Cartesian product

- eg., dog-breeding: MALE x FEMALE
- gives all possible couples

MALE		FEMALE	
name	x	name	
spike		lassie	
spot		shiba	

=

M.name	F.name
spike	lassie
spike	shiba
spot	lassie
spot	shiba

Faloutsos CMU SCS 15-415 #22

CMU SCS

Cartesian product

- find all the pairs of (male, female)

$$\{t \mid \exists m \in MALE \wedge \exists f \in FEMALE$$

$$t.m - name = m.name \wedge$$

$$t.f - name = f.name\}$$

Faloutsos CMU SCS 15-415 #23

CMU SCS

'Proof' of equivalence

- rel. algebra \leftrightarrow rel. tuple calculus

Faloutsos CMU SCS 15-415 #24

CMU SCS

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 #25

CMU SCS

More examples

- join: find names of students taking 15-415

Faloutsos CMU SCS 15-415 #26

CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

Faloutsos CMU SCS 15-415 #27

CMU SCS

More examples

- join: find names of students taking 15-415

$$\{t \mid \exists s \in STUDENT$$

$$\wedge \exists e \in TAKES (s.ssn = e.ssn \wedge$$

$$t.name = s.name \wedge$$

$$e.c - id = 15 - 415)\}$$

Faloutsos CMU SCS 15-415 #28

CMU SCS

More examples

- join: find names of students taking 15-415

$$\{t \mid \exists s \in STUDENT$$

$$\wedge \exists e \in TAKES (s.ssn = e.ssn \wedge$$

$$t.name = s.name \wedge$$

$$e.c - id = 15 - 415)\}$$

join
projection
selection

Faloutsos CMU SCS 15-415 #29

CMU SCS

More examples

- 3-way join: find names of students taking a 2-unit course

Faloutsos CMU SCS 15-415 #30

CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

Faloutsos CMU SCS 15-415 #31

CMU SCS

More examples

- 3-way join: find names of students taking a 2-unit course

$$\{t \mid \exists s \in STUDENT \wedge \exists e \in TAKES$$

$$\exists c \in CLASS (s.ssn = e.ssn \wedge$$

$$e.c-id = c.c-id \wedge$$

$$t.name = s.name \wedge$$

$$c.units = 2)\}$$

join
projection
selection

Faloutsos CMU SCS 15-415 #32

CMU SCS

More examples

- 3-way join: find names of students taking a 2-unit course - in rel. algebra??

$$\pi_{name}(\sigma_{units=2}(STUDENT \bowtie TAKES \bowtie CLASS))$$

Faloutsos CMU SCS 15-415 #33

CMU SCS

Even more examples:

- self -joins: find Tom's grandparent(s)

PC		
p-id	c-id	
Mary	Tom	
Peter	Mary	
John	Tom	

PC		
p-id	c-id	
Mary	Tom	
Peter	Mary	
John	Tom	

Faloutsos CMU SCS 15-415 #34

CMU SCS

Even more examples:

- self -joins: find Tom's grandparent(s)

$$\{t \mid \exists p \in PC \wedge \exists q \in PC$$

$$(p.c-id = q.p-id \wedge$$

$$p.p-id = t.p-id \wedge$$

$$q.c-id = "Tom")\}$$

Faloutsos CMU SCS 15-415 #35

CMU SCS

Hard examples: DIVISION

- find suppliers that shipped all the ABOMB parts

SHIPMENT	
s#	p#
s1	p1
s2	p1
s1	p2
s3	p1
s5	p3

ABOMB	
p#	
p1	
p2	

÷

BAD_S	
s#	
s1	

=

Faloutsos CMU SCS 15-415 #36

CMU SCS

Hard examples: DIVISION

- find suppliers that shipped all the ABOMB parts

$$\{t \mid \forall p(p \in ABOMB \Rightarrow (\exists s \in SHIPMENT(t.s\# = s.s\# \wedge s.p\# = p.p\#)))\}$$

Faloutsos CMU SCS 15-415 #37

CMU SCS

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
 $\{t \mid \forall p(p \notin ABOMB \vee (P(t)))\}$
 - 3) there is no bad shipment that he missed
 $\{t \mid \neg \exists p(p \in ABOMB \wedge (\neg P(t)))\}$

Faloutsos CMU SCS 15-415 #38


CMU SCS

$a \Rightarrow b$ is the same as $\neg a \vee b$

		b
		T
a	T	T
	F	F
F	T	T
	F	T

- If a is true, b must be true for the implication to be true. If a is true and b is false, the implication evaluates to false.
- If a is not true, we don't care about b, the expression is always true.

Faloutsos CMU SCS 15-415 #39




CMU SCS

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 \Rightarrow so does 's'

Faloutsos CMU SCS 15-415 #40




CMU SCS

More on division

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

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

Faloutsos CMU SCS 15-415 #41



CMU SCS

Safety of expressions


- FORBIDDEN: ~~$\{t \mid t \notin STUDENT\}$~~

It has infinite output!!

- Instead, always use

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

Faloutsos CMU SCS 15-415 #42




CMU SCS

Overview - conclusions

- rel. tuple calculus: DECLARATIVE
 - dfn
 - details
 - equivalence to rel. algebra
- **rel. domain calculus + QBE**

Faloutsos CMU SCS 15-415 #43




CMU SCS

General Overview

- relational model
- Formal query languages
 - relational algebra
 - rel. tuple calculus
 - **rel. domain calculus**

Faloutsos CMU SCS 15-415 #44




CMU SCS

Overview - detailed

- rel. tuple calculus
 - dfn
 - details
 - equivalence to rel. algebra
- **rel. domain calculus + QBE**


Faloutsos CMU SCS 15-415 #45

 CMU SCS

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’

Faloutsos CMU SCS 15-415 #46


 CMU SCS

Rel. Dom. Calculus

- find STUDENT record with ssn=123’

$$\{ \langle s, n, a \rangle \mid \langle s, n, a \rangle \in STUDENT \wedge s = 123 \}$$

Faloutsos CMU SCS 15-415 #47

 CMU SCS

Details

- Like R.T.C - symbols allowed:
 - $\wedge, \vee, \neg, \Rightarrow$
 - $>, <, =, \neq, \leq, \geq,$
 - $(,), \in$
- quantifiers \forall, \exists

Faloutsos CMU SCS 15-415 #48

CMU SCS

Details

- but: domain (= column) variables, as opposed to tuple variables, eg:

$$\langle s, n, a \rangle \in STUDENT$$

ssn name address

Faloutsos CMU SCS 15-415 #49

CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

Faloutsos CMU SCS 15-415 #50

CMU SCS


Examples

- find all student records

$$\{\langle s, n, a \rangle \mid \langle s, n, a \rangle \in STUDENT\}$$

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


Faloutsos CMU SCS 15-415 #51

 CMU SCS

Examples

- (selection) find student record with ssn=123

Faloutsos CMU SCS 15-415 #52

 CMU SCS

Examples

- (selection) find student record with ssn=123


$$\{ \langle 123, n, a \rangle \mid \langle 123, n, a \rangle \in STUDENT \}$$

or

$$\{ \langle s, n, a \rangle \mid \langle s, n, a \rangle \in STUDENT \wedge s = 123 \}$$

RTC: $\{ t \mid t \in STUDENT \wedge t.ssn = 123 \}$

Faloutsos CMU SCS 15-415 #53

 CMU SCS

Examples

- (projection) find name of student with ssn=123

$$\{ \langle n \rangle \mid \langle 123, n, a \rangle \in STUDENT \}$$

Faloutsos CMU SCS 15-415 #54

CMU SCS

Examples

- (projection) find name of student with $ssn=123$

$$\{ \langle n \rangle \mid \exists a (\langle 123, n, a \rangle \in STUDENT) \}$$

↑
need to 'restrict' "a"

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

Faloutsos CMU SCS 15-415 #55

CMU SCS

Examples cont'd

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

RTC: $\{ t \mid t \in FT_STUDENT \vee t \in PT_STUDENT \}$

Faloutsos CMU SCS 15-415 #56

CMU SCS

Examples cont'd

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

$$\{ \langle s, n, a \rangle \mid \langle s, n, a \rangle \in FT_STUDENT \vee \langle s, n, a \rangle \in PT_STUDENT \}$$

Faloutsos CMU SCS 15-415 #57

CMU SCS

Examples

- difference: find students that are not staff

RTC: $\{t \mid t \in STUDENT \wedge t \notin STAFF\}$

Faloutsos CMU SCS 15-415 #58

CMU SCS

Examples

- difference: find students that are not staff

$\{ \langle s, n, a \rangle \mid \langle s, n, a \rangle \in STUDENT \wedge \langle s, n, a \rangle \notin STAFF \}$

Faloutsos CMU SCS 15-415 #59

CMU SCS


Cartesian product

- eg., dog-breeding: MALE x FEMALE
- gives all possible couples

MALE		FEMALE	
name	x	name	=
spike	⊗	lassie	
spot		shiba	

M.name	F.name
spike	lassie
spike	shiba
spot	lassie
spot	shiba

Faloutsos CMU SCS 15-415 #60


 CMU SCS

Cartesian product

- find all the pairs of (male, female) - RTC:

$$\{t \mid \exists m \in \text{MALE} \wedge \exists f \in \text{FEMALE} \\ t.m - \text{name} = m.\text{name} \wedge t.f - \text{name} = f.\text{name}\}$$

Faloutsos CMU SCS 15-415 #61


 CMU SCS

Cartesian product

- find all the pairs of (male, female) - RDC:

$$\{ \langle m, f \rangle \mid \langle m \rangle \in \text{MALE} \wedge \langle f \rangle \in \text{FEMALE} \}$$


Faloutsos CMU SCS 15-415 #62

 CMU SCS

‘Proof’ of equivalence

- rel. algebra \leftrightarrow rel. domain calculus
- \leftrightarrow rel. tuple calculus


Faloutsos CMU SCS 15-415 #63

 CMU SCS

Overview - detailed

- rel. domain calculus
 - why?
 - details
 - examples
 - equivalence with rel. algebra
 - more examples**; ‘safety’ of expressions


FaloutsosCMU SCS 15-415#64

 CMU SCS

More examples

- join: find names of students taking 15-415

FaloutsosCMU SCS 15-415#65

 CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

FaloutsosCMU SCS 15-415#66

CMU SCS

More examples

- join: find names of students taking 15-415 - in RTC

$$\{t \mid \exists s \in STUDENT$$

$$\wedge \exists e \in TAKES (s.ssn = e.ssn \wedge$$

$$t.name = s.name \wedge$$

$$e.c - id = 15 - 415)\}$$

Faloutsos CMU SCS 15-415 #67

CMU SCS

More examples

- join: find names of students taking 15-415 - in RDC

$$\{ \langle n \rangle \mid \exists s \exists a \exists g (\langle s, n, a \rangle \in STUDENT$$

$$\wedge \langle s, 15 - 415, g \rangle \in TAKES) \}$$

Faloutsos CMU SCS 15-415 #68

CMU SCS

Sneak preview of QBE:

$$\{ \langle n \rangle \mid \exists s \exists a \exists g (\langle s, n, a \rangle \in STUDENT$$

$$\wedge \langle s, 15 - 415, g \rangle \in TAKES) \}$$

STUDENT		
Ssn	Name	Address
_x	P.	

TAKES		
SSN	c-id	grade
_x	15-415	

Faloutsos CMU SCS 15-415 #69

CMU SCS

Sneak preview of QBE:

- very user friendly
- heavily based on RDC
- very similar to MS Access interface

STUDENT		
Ssn	Name	Address
_x	P.	

TAKES		
SSN	c-id	grade
_x	15-415	

Faloutsos CMU SCS 15-415 #70

CMU SCS

More examples

- 3-way join: find names of students taking a 2-unit course - in RTC:

$$\{t \mid \exists s \in STUDENT \wedge \exists e \in TAKES$$

$$\exists c \in CLASS (s.ssn = e.ssn \wedge$$

$$e.c-id = c.c-id \wedge$$

$$t.name = s.name \wedge$$

$$c.units = 2)\}$$

join

projection

selection

Faloutsos CMU SCS 15-415 #71

CMU SCS

Reminder: our Mini-U db

STUDENT		
Ssn	Name	Address
123	smith	main str
234	jones	forbes ave

CLASS		
c-id	c-name	units
15-413	s.e.	2
15-412	o.s.	2

TAKES		
SSN	c-id	grade
123	15-413	A
234	15-413	B

Faloutsos CMU SCS 15-415 #72

CMU SCS

More examples

- 3-way join: find names of students taking a 2-unit course
 $\{ \langle n \rangle \mid \dots \dots \dots \}$
 $\langle s, n, a \rangle \in STUDENT \wedge$
 $\langle s, c, g \rangle \in TAKES \wedge$
 $\langle c, cn, 2 \rangle \in CLASS \}$

Faloutsos CMU SCS 15-415 #73

CMU SCS

More examples

- 3-way join: find names of students taking a 2-unit course
 $\{ \langle n \rangle \mid \exists s, a, c, g, cn($
 $\langle s, n, a \rangle \in STUDENT \wedge$
 $\langle s, c, g \rangle \in TAKES \wedge$
 $\langle c, cn, 2 \rangle \in CLASS$
 $) \}$

Faloutsos CMU SCS 15-415 #74


CMU SCS

Even more examples:

- self -joins: find Tom's grandparent(s)

PC		PC	
p-id	c-id	p-id	c-id
Mary	Tom	Mary	Tom
Peter	Mary	Peter	Mary
John	Tom	John	Tom

Faloutsos CMU SCS 15-415 #75



CMU SCS

Even more examples:

- self -joins: find Tom's grandparent(s)


$$\{t \mid \exists p \in PC \wedge \exists q \in PC$$

$$(p.c - id = q.p - id \wedge$$

$$p.p - id = t.p - id \wedge$$

$$q.c - id = "Tom")\}$$

Faloutsos CMU SCS 15-415 #76



CMU SCS

Even more examples:

- self -joins: find Tom's grandparent(s)

$$\{t \mid \exists p \in PC \wedge \exists q \in PC$$

$$(p.c - id = q.p - id \wedge$$


$$p.p - id = t.p - id \wedge$$

$$q.c - id = "Tom")\}$$

$$\{ \langle g \rangle \mid \exists p (\langle g, p \rangle \in PC \wedge$$

$$\langle p, "Tom" \rangle \in PC) \}$$

Faloutsos CMU SCS 15-415 #77



CMU SCS

Even more examples:

- self -joins: find Tom's grandparent(s)

$$\{ \langle g \rangle \mid \exists p (\langle g, p \rangle \in PC \wedge$$

$$\langle p, "Tom" \rangle \in PC) \}$$

Faloutsos CMU SCS 15-415 #78

CMU SCS

Hard examples: DIVISION

- find suppliers that shipped all the ABOMB parts

SHIPMENT	
s#	p#
s1	p1
s2	p1
s1	p2
s3	p1
s5	p3

÷

ABOMB	
p#	
p1	
p2	

=

BAD_S	
s#	
s1	

Faloutsos CMU SCS 15-415 #79

CMU SCS

Hard examples: DIVISION

- find suppliers that shipped all the ABOMB parts

$$\{t \mid \forall p(p \in ABOMB \Rightarrow (\exists s \in SHIPMENT(\\ t.s\# = s.s\# \wedge \\ s.p\# = p.p\#)))\}$$

Faloutsos CMU SCS 15-415 #80


CMU SCS

Hard examples: DIVISION

- find suppliers that shipped all the ABOMB parts

$$\{t \mid \forall p(p \in ABOMB \Rightarrow (\exists s \in SHIPMENT(\\ t.s\# = s.s\# \wedge \\ s.p\# = p.p\#)))\} \quad \{<s> \mid \forall p(<p> \in ABOMB \Rightarrow \\ <s, p> \in SHIPMENT)\}$$

Faloutsos CMU SCS 15-415 #81




CMU SCS

More on division

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

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

Faloutsos CMU SCS 15-415 #82




CMU SCS

More on division

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

$$\{<s> \mid \forall c(\exists g(<123, c, g> \in TAKES) \Rightarrow \exists g'(<s, c, g'> \in TAKES))\}$$

Faloutsos CMU SCS 15-415 #83




CMU SCS

Safety of expressions

- similar to RTC
- FORBIDDEN:

$$\{<s, n, a> \mid <s, n, a> \notin STUDENT\}$$

Faloutsos CMU SCS 15-415 #84




CMU SCS

Overview - detailed

- **rel. domain calculus + QBE**
 - dfn
 - details
 - equivalence to rel. algebra

Faloutsos CMU SCS 15-415 #85




CMU SCS

Fun Drill: Your turn ...

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

Faloutsos CMU SCS 15-415 #86



CMU SCS

Your turn ...

- Queries to write in TRC:
 - Find all movies by Paramount studio
 - ... movies starring Kevin Bacon
 - Find stars who have been in a film w/Kevin Bacon
 - Stars within six degrees of Kevin Bacon*
 - Stars connected to K. Bacon via any number of films**

* Try *two* degrees for starters ** Good luck with this one!

Faloutsos CMU SCS 15-415 #87

CMU SCS

Answers ...

- Find all movies by Paramount studio

$$\{M \mid M \in \text{Movie} \wedge M.\text{studioName} = \text{'Paramount'}\}$$

Faloutsos CMU SCS 15-415 #88

CMU SCS

Answers ...

- Movies starring Kevin Bacon

$$\{M \mid M \in \text{Movie} \wedge \exists A \in \text{ActsIn}(A.\text{movieTitle} = M.\text{title} \wedge A.\text{starName} = \text{'Bacon'})\}$$

Faloutsos CMU SCS 15-415 #89

CMU SCS

Answers ...

- Stars who have been in a film w/Kevin Bacon

$$\{S \mid S \in \text{Star} \wedge \exists A \in \text{ActsIn}(A.\text{starName} = S.\text{name} \wedge \exists A2 \in \text{ActsIn}(A2.\text{movieTitle} = A.\text{movieTitle} \wedge A2.\text{starName} = \text{'Bacon'}))\}$$

S:


name	...
------	-----

A:

movie	star
-------	------

A2:

movie	star
-------	------



Faloutsos #90

CMU SCS

Answers ...

- Stars within ^{two}~~six~~ degrees of Kevin Bacon

$$\{S \mid S \in \text{Star} \wedge \exists A \in \text{ActsIn}(A.\text{starName} = S.\text{name} \wedge \exists A2 \in \text{ActsIn}(A2.\text{movieTitle} = A.\text{movieTitle} \wedge \exists A3 \in \text{ActsIn}(A3.\text{starName} = A2.\text{starName} \wedge \exists A4 \in \text{ActsIn}(A4.\text{movieTitle} = A3.\text{movieTitle} \wedge A4.\text{starName} = \text{'Bacon'}))\}$$

Faloutsos CMU SCS 15-415 #91

CMU SCS

Two degrees:

S: name ...

A3: movie star

A4: movie star

Faloutsos CMU SCS 15-415 #92

CMU SCS

Two degrees:

S: name ...


A: movie star

A2: movie star

A3: movie star

A4: movie star

Faloutsos CMU SCS 15-415 #93




CMU SCS

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 ...

Faloutsos CMU SCS 15-415 #94




CMU SCS

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...)

Faloutsos CMU SCS 15-415 #95




CMU SCS

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.

Faloutsos CMU SCS 15-415 #96




CMU SCS

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 #97



CMU SCS

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
