CPSC 304
Introduction to Database Systems

Datalog & Deductive Databases

Textbook Reference
Database Management Systems: Sections 24.1 – 24.4

Databases: The Continuing Saga

When last we left databases…

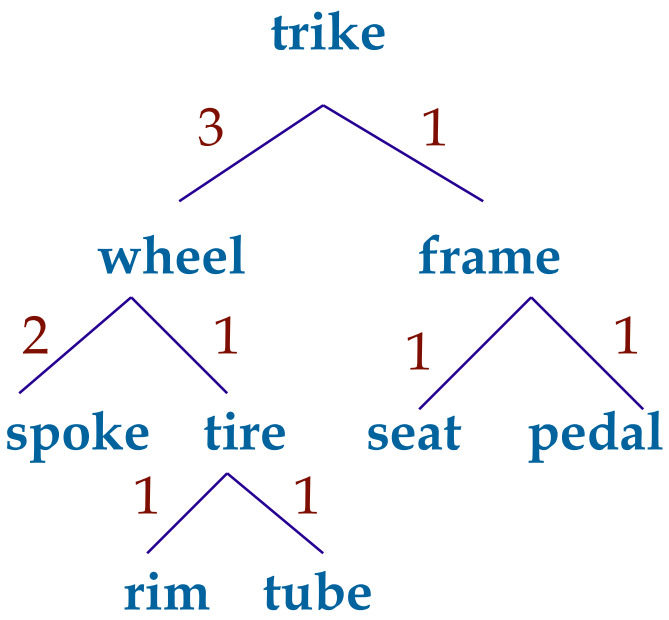
- We had decided they were great things
- We knew how to conceptually model them in ER diagrams
- We knew how to logically model them in the relational model
- We knew how to normalize them
- We learned relational algebra

Let's talk about another database query language – Datalog!

Learning Goals

- Given a set of tuples (an input relation) and rules, compute the output relation for a Datalog program.
- Write Datalog programs to query an input relation.
- Explain why we want to extend query languages with recursive queries. Provide good examples of such queries.
- Explain the importance of safe queries, and what makes a Datalog query safe.

Motivation



part	subpart	qty
trike	wheel	3
trike	frame	1
frame	seat	1
frame	pedal	1
wheel	spoke	2
wheel	tire	1
tire	rim	1
tire	tube	1

Write try to a relational algebra query to find all of the components required for a trike

Datalog

- Based on logic notation (Prolog)
- Can express queries that are not expressible in relational algebra or standard SQL (recursion).
- Uses sets (like RA, unlike SQL)
- Cleaner → convenient for analysis

A nice and easy example to start

From a query perspective: ask a query and get answers.

From a logical perspective: use facts to derive new facts.

Tuples/Initial facts:

Parent(“Dee”, “Jan”)

Parent(“Jan”, “Jamie”)

Parent(“Dee”, “Wally”)

Parent(“Wally”, “Jean”)

Query:

Grandparent(A,C) :- Parent(A,B), Parent(B,C)

Answer/New facts:

Grandparent(“Dee”, “Jamie”)

Grandparent(“Dee”, “Jean”)

Predicates and Atoms

- Relations are represented by predicates
- Tuples are represented by atoms.

  Parent(“Dee”, “Jan”)

Arithmetic comparison atoms:

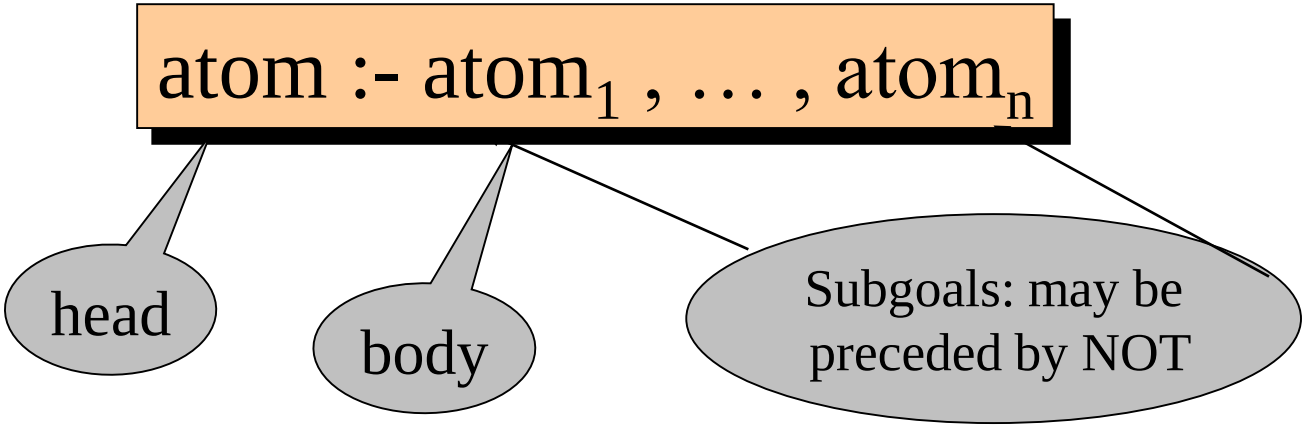
X < 100, X+Y+5 > Z/2, X <> 42

- Negated atoms:

  NOT Parent(“Dee”, “Jean”)

Datalog Definitions

- A Datalog rule:



- E.g.: Grandparent(A,C) :- Parent(A,B), Parent(B,C).
- A comma between the atoms means "and" (sometimes you'll see this as "&")
- Read the rule as "if we know body, then we know head"
- You may also see head ← body, e.g., Grandparent(A,C)← Parent(A,B), Parent(B,C)
- Datalog program = a collection of rules

A single rule can express exactly select-project-join queries.

The Meaning of Datalog Rules

Parent("Dee", "Jan").
Parent("Jan", "Jamie").
Parent("Dee", "Wally").
Parent("Wally", "Jean").
Grandparent(A,C) :- Parent(A,B), Parent(B,C).

Grandparent("Dee", "Jamie")
Grandparent("Dee", "Jean")

Consider every assignment from the variables in the body to the constants in the database. (same variable name means require the same value)

If each atom in the body is in the database, then the tuple for the head is in the result.

Running example

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Projection

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Projection is performed by the variables in the head of the query:

- Find the name of all products:

  RA: π_{name}(Product)

  Datalog: Ans(N):-Product(P,N,PR,C,M)

Projection practice

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Find the countries of all the companies

Ans1(Co):- Company (C, N, S, Co)

– make sure C <> Co

Selection

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Selection is performed by either using the same variable, a constant, or adding an arithmetic comparison:

- Find all purchases with the same buyer and seller:
  RA: $\sigma_{\text{buyer-sin = seller-sin}}$(Purchase)
  Datalog: Ans1(B,B,S,P):-Purchase(B,B,S,P)

- Find all Canadian companies:
  RA: $\sigma_{\text{country='Canada'}}$(Company)
  Datalog: Ans2(C,N,S, 'Canada'):-Company(C,N,S, 'Canada')

Selection practice

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Find all products over $99.99:

RA: $\sigma_{price>99.99}(Product)$

Datalog: Ans(I,N,P,C,M) :- Product(I,N,P,C,M), P>99.99

Find all English companies with stock prices less than $100

Ans1(C,N,S, 'England'):-
Company(C, N, S, 'England'), S < 100

Selection & Projection

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

- Find the names of all products over $99.99:

  RA: $\pi_{name}(\sigma_{price>99.99}(Product))$

  Datalog: Ans(N) :- Product(I,N,P,C,M), P>99.99

Clicker Question

Given the following schema:

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

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

Company (cid, name, stock price, country)

Person(sin, name, phone number, city)

And the Datalog definition:

Ans(C,N) :- Product(I,N,P,C,M), P>99.99

What is the proper translation to RA?

A. $\pi_{name,category}(\sigma_{price>99.99}(Product))$

B. $\pi_{name}(\pi_{category}(\sigma_{price>99.99}(Product)))$

C. $\pi_{category}(\pi_{name}(\sigma_{price>99.99}(Product)))$

D. $\pi_{category,name}(\sigma_{price>99.99}(Product))$

E. None of the above

Clicker Question

Given the following schema:

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

And the Datalog definition:

Ans(C,N) :- Product(I,N,P,C,M), P>99.99

What is the proper translation to RA?

A. $\pi_{name,category}(\sigma_{price>99.99}(Product))$
B. $\pi_{name}(\pi_{category}(\sigma_{price>99.99}(Product)))$
C. $\pi_{category}(\pi_{name}(\sigma_{price>99.99}(Product)))$
D. $\pi_{category,name}(\sigma_{price>99.99}(Product))$
E. None of the above

A – name before category
B,C – can't project name from category & vice versa

D is correct

Selection & Projection and Joins

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Joins are performed by using the same variable in different relations

- Find store names where Fred bought something:
  RA: $\pi_{store}\sigma_{name=\text{“Fred”}}(Person)\bowtie_{sin=buyer\text{-}sin}Purchase$

- Datalog: S(N) :- Person(S, “Fred”,T,C),
  Purchase(S,L,N,P)

Anonymous Variables

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Find names of people who bought from “Gizmo Store”

E.g.:

Ans4(N) :- Person(S, N, _, _), Purchase (S, _,“Gizmo Store”, _)

Each _ means a fresh, new variable

Very useful: makes Datalog even easier to read

Exercise part 1

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Ex #1: Find SINs of people who bought products in the "computers" category.

Ans1(B):-Purchase(B,_,_,P), Product(P,_,_,'Computers',_)

Ex #2: Find the sin of people who bought Canadian products

Ans2(B):- Purchase(B,_,_,P), Product(P,_,_,_,C),
Company(C, _, _, 'Canada')

Clicker exercise – basic Datalog

- Consider Unknown(A,B):
- Compute:
  Secret (A ,B):- Unknown(A,C), Unknown(C,B)
- Which of the following tuples are in Secret(A,B)?

A. (a1,a1)
B. (a2,a3)
C. (a4,a5)
D. Both A & B
E. None of the above

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a5
a2	a1

Clicker exercise – basic Datalog

- Consider Unknown(A,B):
- Compute:
  Secret (A ,B):- Unknown(A,C), Unknown(C,B)
- Which of the following tuples are in Secret(A,B)?

A. (a1,a1) (a1,a2), (a2, a1)
B. (a2,a3) (a2,a1), (a1, a3)
C. (a4,a5) Not an answer
D. Both A & B Correct
E. None of the above

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a5
a2	a1

Clicker exercise – A more meaningful version

- Consider Flight(orig,dest):
- Compute:
  Twohops (orig,final_dest):-
  Flight(orig,mid), Flight(mid,final_dest) (paths of length 2 again)
- Which of the following tuples are in Twohops(orig,final_dest)?

A. (YVR,YVR) (YVR, SEA), (SEA, YVR)
B. (SEA,PIT) (SEA, YVR), (YVR, PIT)
C. (RDU,ITH) One hop
D. Both A & B Correct
E. None of the above

orig	dest
YVR	SEA
YVR	PIT
YVR	RDU
SEA	PIT
PIT	RDU
RDU	ITH
SEA	YVR

Exercise part 2

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Ex #3: Find names of people who bought Canadian products that cost under 50

Ans3(N):-Product(P, _, Pr, _, C), Company(C,_,_, 'Canada'),
Purchase(B, _, _, P), Person(B, N, _, _), Pr < 50

Clicker Question

Consider Unknown(A,B):

Compute:

Secret (A ,B):- Unknown(B,A), Unknown(C,A), C ≠ B.

Which of the following tuples are in Secret(A ,B)?

A. (a2,a3)
B. (a1,a2)
C. (a2,a1)
D. All of the above
E. None of the above

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a2
a2	a1

Answer A explained

Consider Unknown(A,B):

Compute:

Secret (A ,B):- Unknown(B,A), Unknown(C,A), C ≠ B.

Which of the following tuples are in Secret(A ,B)?

A. (a2,a3)
B. (a1,a2)
C. (a2,a1)
D. All of the above
E. None of the above

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a2
a2	a1

A=a2, B=a3

Unknown(a3,a2) is not in the table.

Answer B explained

Consider Unknown(A,B):

Compute:

Secret (A ,B):- Unknown(B,A), Unknown(C,A), C ≠ B.

Which of the following tuples are in Secret(A ,B)?

A. ~~(a2,a3)~~
B. (a1,a2)
C. (a2,a1)
D. All of the above
E. None of the above

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a2
a2	a1

A=a1, B=a2

Unknown(a2,a1) ok

Unknown(C,a1), C ≠ B does not exist

Answer C explained

Consider Unknown(A,B):

Compute:

Secret (A ,B):- Unknown(B,A), Unknown(C,A), C ≠ B.

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a2
a2	a1

Which of the following tuples are in Secret(A ,B)?

A. ~~(a2,a3)~~

B. ~~(a1,a2)~~

C. (a2,a1)

D. All of the above

E. None of the above

A=a2, B=a1

Unknown(a1,a2) ok

Unknown(a4,a2) where C=a4 ok

C is correct

Multiple Datalog Rules

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

- Find names of people that are either buyers or sellers:

  A(N) :- Person(S,N,A,B), Purchase(S,C,D,E)
  A(N) :- Person(S,N,A,B), Purchase(C,S,D,E)

- Multiple rules correspond to union

Exercise part 3

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Ex #4: Find sins of people who bought stuff from a person named Joe or bought products from a company whose stock prices is more than $50.

```
Ans(Sin):-Purchase(Sin,Ssin,_,_), Person(Ssin,"Joe",_,_)
Ans(Sin):-Purchase(Sin,_,_,Pid), Product(Pid,_,_,_,Cid),
                    Company(Cid,_,Sp,_), Sp > 50
```

UBC Department of Computer Science
Undergraduate Events
More details @ https://my.cs.ubc.ca/students/development/events

CS/Life Sciences Panel

Tues. Feb. 27th, 2018
5:30pm-7:30pm
Life Sciences Centre Theatre 3

Blockchain@UBC Monthly Talk

Tues. Feb. 27th, 2018
Noon – 1pm
IKB Dodson Room

Tesla Info Session

Wed. Mar. 14th, 2018
5:30-7:30PM
Scarfe Room 100

UBC CDM Digital Media Capstone Course (6 Credit for CPSC 448)

Apply by: Feb. 28th, 2018
sites.google.com/site/ubccdmcapstone

Outreachy Internship

Apply by: Mar. 22nd, 2018
my.cs.ubc.ca/students/career/outreachy-internship-applications-open

Azure University Tour

Tues. Apr. 3rd, 2018
5:30pm – 8:45pm
Location TBA

Sign up link:
https://aka.ms/AzureTourUBC

Microsoft Imagine Cup

Submit by: Sun. Apr. 8th, 2018
https://imagine.microsoft.com/en-us/canada

Now where were we…

- We'd been discussing Datalog, which as you'll recall, is a logic-based query language
- You'd gotten to the point where you could basically write select-project-join + union queries
- What were those again?

Let's revisit

Our ongoing schema:

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

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

Company (cid, name, stock price, country)

Person(sin, name, phone number, city)

Write queries in relational algebra and Datalog to find the names of all products

Relational algebra: π_{name}(Product)

Datalog: q(n):-Product(_, n, _, _, _)

Let's revisit

Our ongoing schema:

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

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

Company (cid, name, stock price, country)

Person(sin, name, phone number, city)

Write queries in relational algebra and Datalog to find the purchases of products with pid = 42

Relational algebra: $\sigma_{pid=42}$(Purchase)

Datalog: q(b,s,st,p):-Purchase(b, s, st, p), p = 42

Let's revisit

Our ongoing schema:

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

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

Company (cid, name, stock price, country)

Person(sin, name, phone number, city)

Write queries in relational algebra and Datalog to find the names of people who have bought products from themselves

RA: $\pi_{name}((\sigma_{buyer\text{-}sin=seller\text{-}sin}Purchase) \bowtie_{seller\text{-}sin = sin} Person)$

Datalog: q(n):-Person(s,n,_,_), Purchase(s, s,_, _)

Let's revisit

Our ongoing schema:

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

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

Company (cid, name, stock price, country)

Person(sin, name, phone number, city)

Write queries in relational algebra and Datalog to find the SINs of people living in Vancouver or Surrey

RA: $\pi_{sin}(\sigma_{city = \text{'Surrey'} \vee city = \text{'Vancouver'}} Person)$

Datalog: q(s):-Person(s,_, _, "Vancouver")

q(s):-Person(s,_,_, "Surrey")

Great! But surely there's more...

Our ongoing schema:

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

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

Company (cid, name, stock price, country)

Person(sin, name, phone number, city)

Write a query in relational algebra to find SINs of people who have not made a purchase

$\pi_{sin}(Person) - \pi_{buyer\text{-}sin}(Purchase)$

Negation

Find people who live in Vancouver but have not bought anything at “The Bay”

VancouverAntiBay(buyer,seller,product,store) :-

Person(buyer, name, phone,“Vancouver”),

Purchase(buyer, seller, store, product),

not Purchase(buyer, seller, “The Bay”, product)

Note that not has a different semantics than in relational algebra – in Datalog it means “there exists no”

You may also see “NOT” written as “¬ ”

Rule safety

- Every variable in the head of a rule must also appear in the body.

PriceYarts (Part, Price) :- Assembly(Part, Subpart, Qty) , Qty> 2.

Can generate infinite new facts

- Every variable must appear in a relation

Ans(Id) :- Product(Id,Name,Price,Category,Cid), Id < Stock_price

What is the value of stock_price?

- Every variable in the head of the rule must appear in some positive relation occurrence in the body

Ans(Sin):- NOT Person(Sin, 'Joe', Ph, City)

Sin, Ph, and City are unsafe

Clicker Question

Given the following schema:

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

And the following query:

"Find the phone numbers of all customers who bought a computer product from a Canadian company that cost $100"

What is the proper translation into Datalog?

A. Ans(PN):- Purchase(B,_,_,P), Product(P,_,_,'computer',C),
Company(C, _, _, 'Canada'), Person(B,_,PN,_,_),price=100.

B. Ans(PN):- Purchase(B,_,_,P), Product(P,_,100, 'computer',C),
Company(C, _, _, 'Canada'), Person(B,_,PN,_,_).

C. Ans(PN):- Purchase(B,_,_,P), Product(P,_,100, 'computer',C),
Company(C, _, _, country), Person(B,_,_,_,_),country='Canada'.

D. All are correct

E. None of the above

Clicker Question

Given the following schema:

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

A – price not in any atoms
C – PN not in any atoms

And the following query:

"Find the phone numbers of all customers who bought a computer product from a Canadian company that cost $100"

What is the proper translation into Datalog?

A. Ans(PN):- Purchase(B,_,_,P), Product(P,_,_,'computer',C), Company(C, _, _, 'Canada'), Person(B,_,PN,_,_),price=100.

B. Ans(PN):- Purchase(B,_,_,P), Product(P,_,100, 'computer',C), Company(C, _, _, 'Canada'), Person(B,_,PN,_,_).

C. Ans(PN):- Purchase(B,_,_,P), Product(P,_,100, 'computer',C), Company(C, _, _, country), Person(B,_,_,_,_),country='Canada'.

D. All are correct

E. None of the above

B is correct

Exercise part 4

Product (pid, name, price, category, maker-cid)
Purchase (buyer-sin, seller-sin, store, pid)
Company (cid, name, stock price, country)
Person(sin, name, phone number, city)

Ex #5: Find the sins of people who are not named 'Joe'

Ans(s):- Person(s,n,p,c), NOT Person(s, "Joe", p, c)

Note that we can't use anonymous variables here or the variables in the "Joe" person will not be safe

Defining Queries for reuse: Views

VancouverAntiBay(Buyer,Seller,Product,Store) :-

Person(Buyer, “Vancouver”, Phone),

Purchase(Buyer, Seller, Product, Store),

not Purchase(Buyer, Seller, Product, “The Bay”)

Ans6(Buyer) :- VancouverAntiBay(Buyer, “Joe”, Pro, Store)

Ans6(Buyer) :- VancouverAntiBay(Buyer, Sell, Prod, Store),

Product(Prod, Price, Cat, Maker)

Company(Maker, Sp, Country), Sp > 50.

What is returned by Ans6?

Buyers from Vancouver that have never purchased anything from “The Bay” that have either bought from Joe or products that are from companies with SP> 50

Clicker exercise – Datalog with negation

Consider the Unknown(A,B) relation, which is given on the right hand side

Unknown

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a5

Secret (A, B):- Unknown (A,C), Unknown (C,B)
Nameless (A,B):-Secret (A,B), NOT Unknown(A,B)

- Which of the following tuples are in Nameless(A,B)?

A. (a1,a4)
B. (a1,a5)
C. (a4,a5)
D. All of the above
E. None of the above

Clicker exercise – Datalog with negation

Unknown

o	d
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a5

- Consider the Unknown(A,B) relation, which is given on the right hand side

Secret (A, B):- Unknown (A,C), Unknown (C,B)
Nameless (A,B):-Secret (A,B), NOT Unknown(A,B)

- Which of the following tuples are in Nameless(A,B)?

A. (a1,a4)
B. (a1,a5)
C. (a4,a5)
D. All of the above
E. None of the above

In Secret(a1,a3)(a3,a4) and Unknown(a1,a4)

In Nameless, so correct

In Unknown

Clicker exercise – A more meaningful version

Flight

orig	dest
YVR	SEA
YVR	PIT
YVR	RDU
SEA	PIT
PIT	RDU
RDU	ITH

- Consider Flight(orig,dest):
- Compute Indirect_only(orig,dest) defined by:
  Twohops (Orig,Final_dest):- Flight(Orig,Mid),
  Flight(Mid,Final_dest)
  Indirect_only(orig,dest):-Twohops(orig,dest),
  NOT Flight(orig,dest)
- Which of the following tuples are in Indirect_only(orig,dest)?

A. (YVR,RDU) In Twohops(YVR,PIT)(PIT,RDU) and Flight(YVR,RDU)

B. (YVR,ITH) In Indirect_only, so correct

C. (RDU,ITH) In Flight

D. All of the above

E. None of the above

Taking it to the next level

Say you're planning a beach vacation

And you wanted to find if it's possible to get from YVR to OGG (that's on Maui)

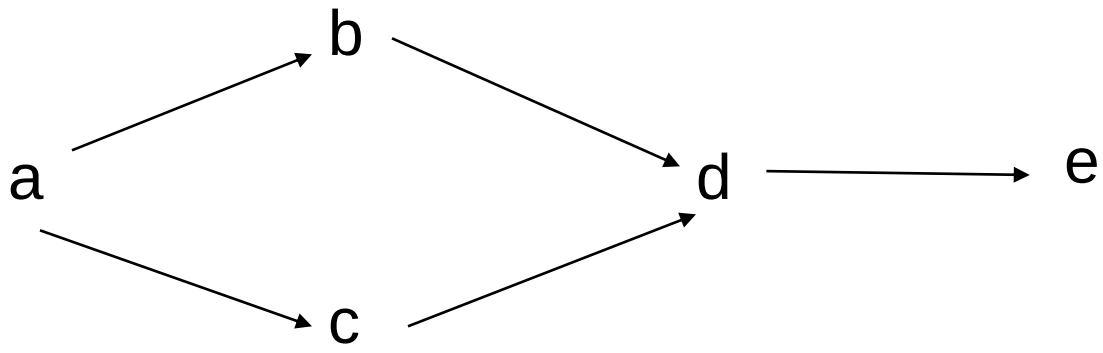
Your available information:

Flight(airline,num,origin,destination)

Now what?

A more general Example: Transitive Closure

Suppose we represent a graph w/ relation *Edge(X,Y):*
Edge(a,b), Edge (a,c), Edge(b,d), Edge(c,d), Edge(d,e)



How can I express the query: *Find all paths*

```
Path(X, Y)   :-   Edge(X, Y).
Path(X, Y)   :-   Path(X, Z),  Path(Z, Y).
```

Evaluating Recursive Queries

Path(X, Y) :- Edge(X, Y).
Path(X, Y) :- Path(X, Z), Path(Z, Y).

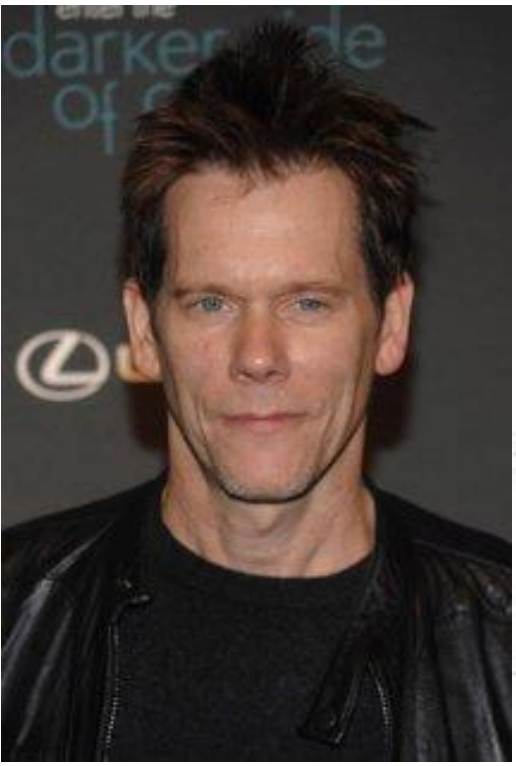
Semantics: evaluate the rules until a *fixed point:*

Iteration #0: Edge: {(a,b), (a,c), (b,d), (c,d), (d,e)}
Path: {}
Iteration #1: Path: {(a,b), (a,c), (b,d), (c,d), (d,e)}
Iteration #2: Path gets the new tuples: (a,d), (b,e), (c,e)
Path: {(a,b), (a,c), (b,d), (c,d), (d,e), (a, d), (b,e), (c, e)}
Iteration #3: Path gets the new tuple: (a,e)
Path: {(a,b), (a,c), (b,d), (c,d), (d,e), (a, d), (b,e), (c, e), (a,e)}
Iteration #4: Nothing changes → Stop.

Note: # of iterations depends on the data. Cannot be anticipated by only looking at the query!

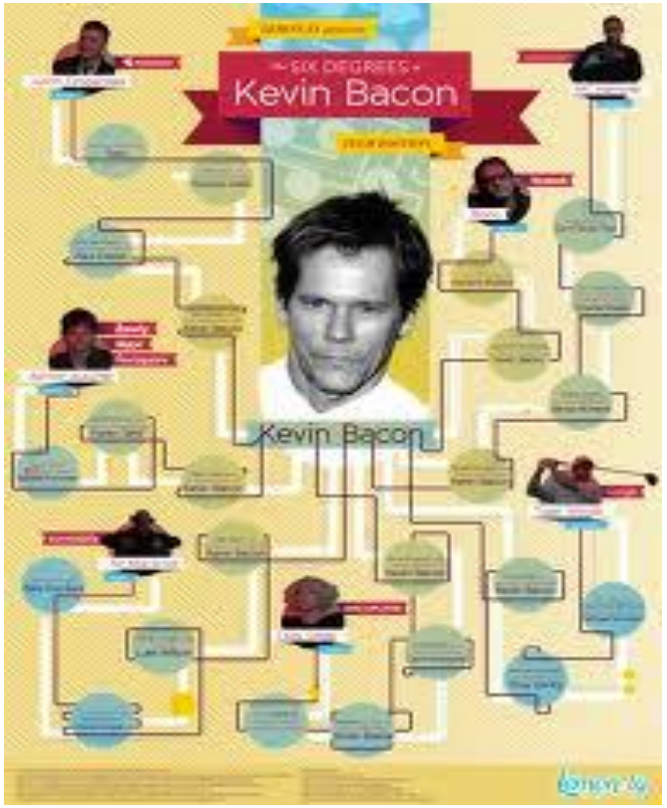
A fun Example

Kevin Bacon



6 degrees of separation

6 degrees of Kevin Bacon



More examples

- Given:
  Movie(id, title)
  Actor(id, name)
  Role(movie-id, actor-id, character)
- Find names of actors who have “Bacon numbers” (assume there’s only one “Kevin Bacon”)

```
CoStars(Aid,Bid):-Role(Mid,Aid,_), Role(Mid,Bid,_)
CoStars(Aid,Bid):- CoStars(Aid,Cid), CoStars(Cid,Bid)
Bacon_N(B):-Actor(Aid, “Kevin Bacon”), CoStars(Aid,Bid),
                Actor(Bid,B)
```

Skip the stuff on Magic Sets

- That's Datalog
- It's simple
- It's based on logic
- It's easy to see the join patterns (especially with anonymous variables)

Learning Goals Revisited

- Given a set of tuples (an input relation) and rules, compute the output relation for a Datalog program.
- Write Datalog programs to query an input relation.
- Explain why we want to extend query languages with recursive queries. Provide good examples of such queries.
- Explain the importance of safe queries, and what makes a Datalog query safe.