CPSC 304 – Administrative notes November 6 and 7, 2024

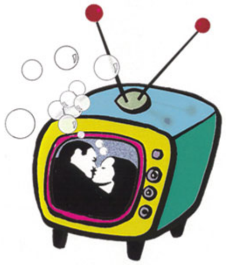
- Project:
  - Make sure you have your end-to-end project tech stack working now!!!
  - Milestone 4: Project implementation – due November 29
    - You cannot change your code after this point!
  - Milestone 5: Group demo – week of December 2
  - Milestone 6: Individual Assessment – Due November 29
- Tutorials: basically project group work time
- Final exam: December 16 at 12pm!
  - Conflict form on Piazza – due November 12
- Fall break! No Tutorials/Classes/Office hours November 11-13

CPSC 304
Introduction to Database Systems

Datalog & Deductive Databases

Textbook Reference
Database Management Systems: Sections 24.1 – 24.4

Databases: The Continuing Saga



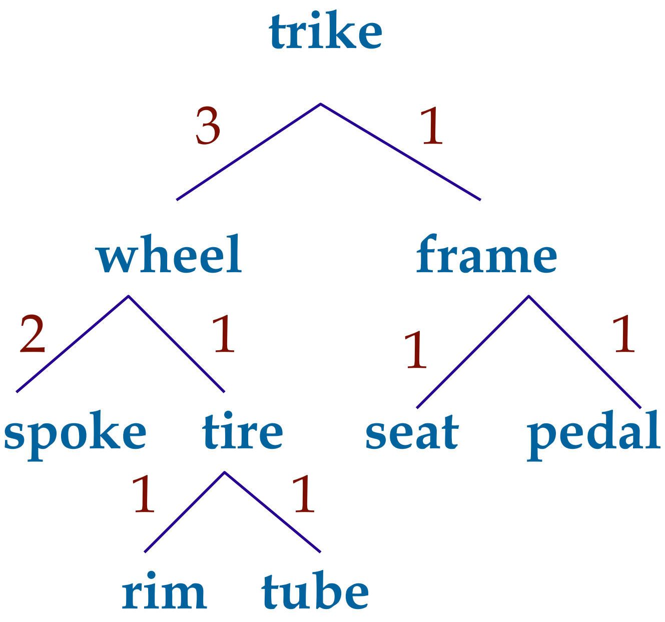
When last we left databases…

- 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
- We learned SQL
- Let’s talk about our final 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



trike

3 1

wheel frame

2 1 1 1

spoke tire seat pedal

1 1

rim tube

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

Try to write 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:

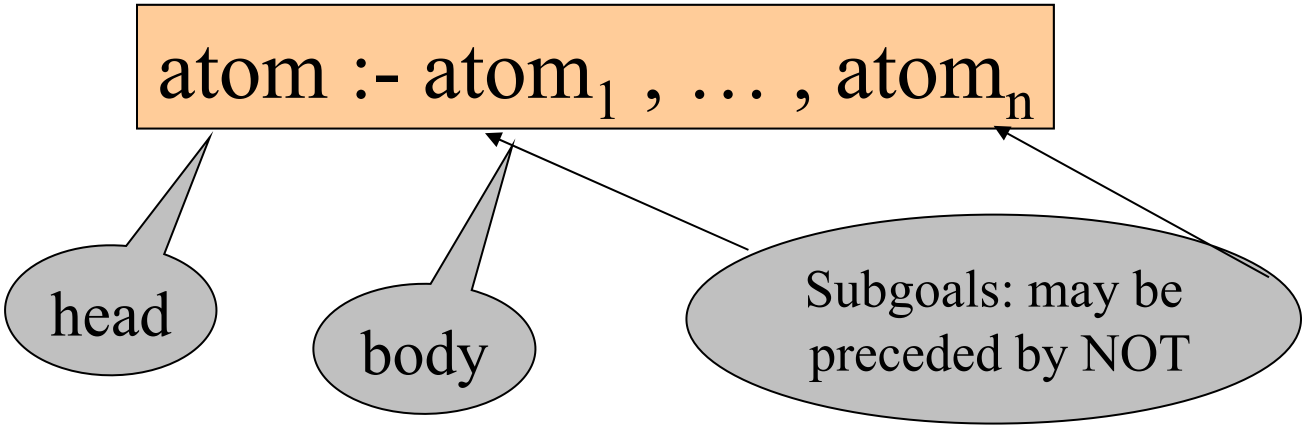
$X < 100$, $X+Y+5 > Z/2$, $X \neq 42$

- Negated atoms:

  NOT Parent(“Dee”, “Jean”)

Datalog Definitions

- A Datalog rule:



atom :- atom$_1$, … , atom$_n$

head

body

Subgoals: may be preceded by NOT

- 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)

Note: 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’)
  Alternate option: Ans2(C,N,S, Co):-Company(C,N,S, Co),
  Co = ‘Canada’

Selection Practice

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 OR
q(b,s,st,42):-Purchase(b, s, st, 42)

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

  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(BSIN, “Fred”,T,C),
  Purchase(BSIN,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

Anonymous Variables

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, _, _, _)

Anonymous Variables

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

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

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 Flight(orig,dest):

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

- 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

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 products sold by a Canadian company that cost under 50

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

Clicker Question

Consider Unknown(o,d): ---------------------------------------→

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

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

C is correct

Tuples explained

Consider Unknown (o,d): --→

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

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

This is asking for cases where there are two different o's with the same d. Be careful about ordering!

B	A
a1	a2
a4	a2
a1	a3
a2	a3
a1	a4
a3	a4

Tuples explained

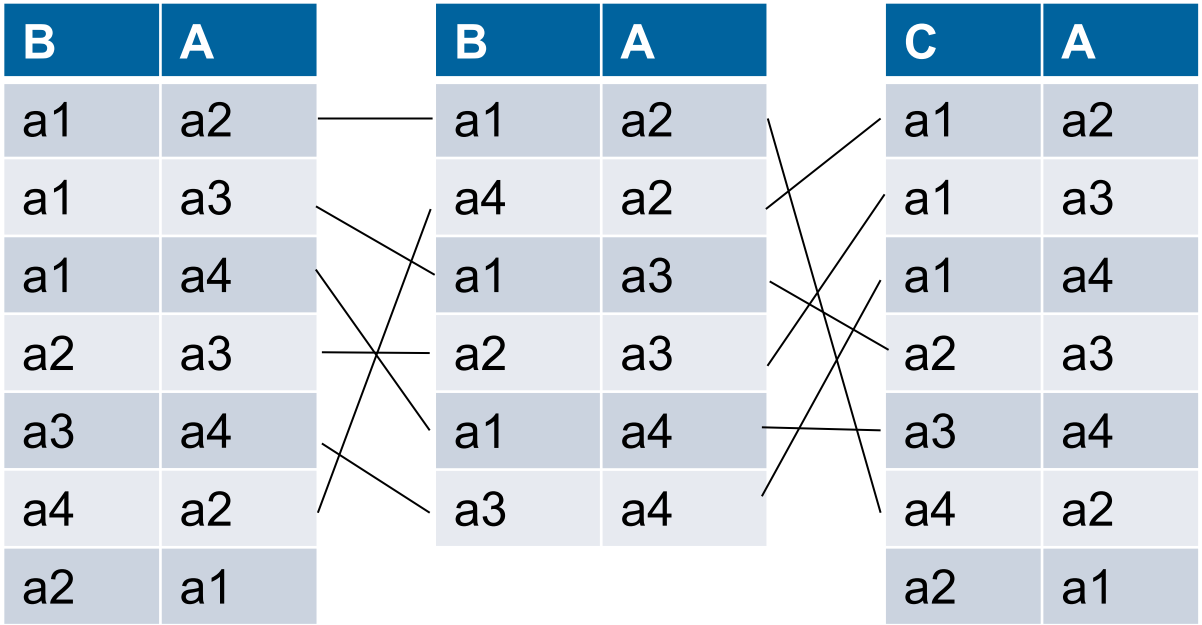
Consider Unknown (o,d): --→

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

Compute:

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

Unknown(B,A) Secret Unknown(C,A)
answers from here columns reversed Witness (C ≠ B)



B	A
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a2
a2	a1

B	A
a1	a2
a4	a2
a1	a3
a2	a3
a1	a4
a3	a4

C	A
a1	a2
a1	a3
a1	a4
a2	a3
a3	a4
a4	a2
a2	a1

Each matched tuple from Unknown(B,A) appears in Secret because of the “witness” in the third table

Answer A explained

Consider Unknown (o,d): ---------------------------------------→

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

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

A=a2, B=a3

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

Answer B explained

Consider Unknown (o,d): --→

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

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

A=a1, B=a2

Unknown(a2,a1) ok

Unknown(C,a1), C ≠ B does not exist

Answer C explained

Consider Unknown (o,d): ---------------------------------------→

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

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

A=a2, B=a1

Unknown(a1,a2) ok

Unknown(a4,a2) where C=a4 ok

A=a2, B=a4

Unknown(a4,a2) ok

C is correct