Introduction to Data Management
CSE 344

Lecture 12: Relational Calculus
Announcements

• Homework 3 due night, 11 pm
• Homework 4 due Thursday, 11pm

• Next web quiz posted by tomorrow, due next Tuesday

• Suggested reading *Query Language Primer*, posted on the website
Announcements

• We will be moving to PAA A114 started this Wednesday
Announcements

• Midterm next Monday!
  – Covers everything up to and including Wednesday’s lecture
  – Extra OH Tuesday 5:00-6:00, CSE 530
Big Picture

• Query languages and data models
  – SQL, SQL, SQL, SQL, …
  – Relational algebra
    – Datalog
    – Relational calculus
  – Xpath and Xquery
Datalog program

Example: Find all actors with Bacon number ≤ 2

B0(x) :- Actor(x,'Kevin', 'Bacon')
B1(x) :- Actor(x,f,l), Casts(x,z), Casts(y,z), B0(y)
B2(x) :- Actor(x,f,l), Casts(x,z), Casts(y,z), B1(y)
Q4(x) :- B0(x)
Q4(x) :- B2(x)

Note: Q4 is the union of B0, B1, and B2
RA to Datalog by Examples

Selection: \( \sigma_{x>100 \text{ and } y='some string'} (R) \)
\[
L(x,y,z) :- R(x,y,z), x > 100, y='some string'
\]

Selection \( x>100 \text{ or } y='some string' \)
\[
L(x,y,z) :- R(x,y,z), x > 100 \\
L(x,y,z) :- R(x,y,z), y='some string'
\]
RA to Datalog by Examples

Equi-join: $R \bowtie_{R.A=S.D \text{ and } R.B=S.E} S$

$J(x,y,z,q) :- R(x,y,z), S(x,y,q)$
RA to Datalog by Examples

Projection

P(x) :- R(x,y,z)
RA to Datalog by Examples

To express difference, we add negation

\[ D(x,y,z) :\neg R(x,y,z), \neg S(x,y,z) \]
Examples

R(A,B,C)
S(D,E,F)
T(G,H)

Translate: $\Pi_A(\sigma_{B=3} (R))$
A(a) :- R(a,3,_{})
Underscore used to denote an "anonymous variable",
a variable that appears only once.
Examples

R(A,B,C)
S(D,E,F)
T(G,H)

Translate: \( \Pi_A(\sigma_{B=3}(R) \bowtie_{R.A=S.D} \sigma_{E=5}(S) ) \)
A(a) :- R(a,3,__), S(a,5,__)
Friend(name1, name2)
Enemy(name1, name2)

More Examples

Find Joe's friends, and Joe's friends of friends.

A(x) :- Friend('Joe', x)
A(x) :- Friend('Joe', z), Friend(z, x)
More Examples

Find all of Joe's friends who do not have any friends except for Joe:

\[
\begin{align*}
\text{JoeFriends}(x) & :\text{- Friend('Joe',x)} \\
\text{NonAns}(x) & : \text{JoeFriends}(x), \text{Friend}(x,y), y \neq 'Joe' \\
\text{A}(x) & : \text{JoeFriends}(x), \text{NOT NonAns}(x)
\end{align*}
\]
More Examples

Find all people such that all their enemies' enemies are their friends

• Q: if someone doesn't have any enemies nor friends, do we want them in the answer?
• A: Yes!

Friend(name1, name2)
Enemy(name1, name2)

Everyone(x) :- Friend(x,y)
Everyone(x) :- Friend(y,x)
Everyone(x) :- Enemy(x,y)
Everyone(x) :- Enemy(y,x)
NonAns(x) :- Enemy(x,y), Enemy(y,z), NOT Friend(x,z)
A(x) :- Everyone(x), NOT NonAns(x)
More Examples

Find all persons x that have a friend all of whose enemies are x's enemies.

\[
\text{Everyone}(x) :- \text{Friend}(x,y) \\
\text{NonAns}(x) :- \text{Friend}(x,y) \text{ Enemy}(y,z), \text{NOT} \text{ Enemy}(x,z) \\
A(x) :- \text{Everyone}(x), \text{NOT} \text{ NonAns}(x)
\]
Datalog Summary

• EDB (base relations) and IDB (derived relations)
• Datalog program = set of rules
• Datalog is recursive
• Pure datalog does not have negation; if we want negation we say “datalog +negation”
• Multiple atoms in a rule mean join (or intersection)
• Multiple rules with same head mean union
Relational Calculus

- Aka *predicate calculus* or *first order logic*

- TRC = Tuple RC
  - See book

- DRC = Domain RC
  - We study only this one
  - Also see: *Query Language Primer*
Relational Calculus

Relational predicate $P$ is a formula given by this grammar:

$$ P ::= \text{atom} | P \land P | P \lor P | P \Rightarrow P | \neg P | \forall x. P | \exists x. P $$

Query $Q$:

$$ Q(x_1, \ldots, x_k) = P $$
Relational Calculus

Relational predicate $P$ is a formula given by this grammar:

$$P ::= \text{atom} \mid P \land P \mid P \lor P \mid P \Rightarrow P \mid \text{not}(P) \mid \forall x. P \mid \exists x. P$$

Query $Q$:

$$Q(x_1, \ldots, x_k) = P$$

Example: find the first/last names of actors who acted in 1940

$$Q(f,l) = \exists x. \exists y. \exists z. (\text{Actor}(z,f,l) \land \text{Casts}(z,x) \land \text{Movie}(x,y,1940))$$

What does this query return?

$$Q(f,l) = \exists z. (\text{Actor}(z,f,l) \land \forall x.(\text{Casts}(z,x) \Rightarrow \exists y.\text{Movie}(x,y,1940)))$$
Likes(drinker, beer)
Frequents(drinker, bar)
Serves(bar, beer)

Important Observation

Find all bars that serve all beers that Fred likes

\[ A(x) = \forall y. \text{Likes}("Fred", y) \Rightarrow \text{Serves}(x,y) \]

• Note: \( P \Rightarrow Q \) (read P implies Q) is the same as \( \neg P \) OR Q
In this query: If Fred likes a beer the bar must serve it \( (P \Rightarrow Q) \)
In other words: Either Fred does not like the beer \( (\neg P) \) OR the bar serves that beer \( (Q) \).

\[ A(x) = \forall y. \neg \text{Likes}("Fred", y) \text{ OR Serves}(x,y) \]
More Examples

Find drinkers that frequent some bar that serves some beer they like.
More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y, z) \land \text{Likes}(x, z) \]
Likes(drinker, beer)
Frequents(drinker, bar)
Serves(bar, beer)

More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y, z) \land \text{Likes}(x, z) \]

Find drinkers that frequent only bars that serves some beer they like.
Likes(drinker, beer)
Frequents(drinker, bar)
Serves(bar, beer)

More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y, z) \land \text{Likes}(x, z) \]

Find drinkers that frequent only bars that serves some beer they like.

\[ Q(x) = \forall y. \text{Frequents}(x, y) \Rightarrow (\exists z. \text{Serves}(y, z) \land \text{Likes}(x, z)) \]
More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y, z) \land \text{Likes}(x, z) \]

Find drinkers that frequent only bars that serves some beer they like.

\[ Q(x) = \forall y. \text{Frequents}(x, y) \Rightarrow (\exists z. \text{Serves}(y, z) \land \text{Likes}(x, z)) \]

Find drinkers that frequent some bar that serves only beers they like.
More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y, z) \land \text{Likes}(x, z) \]

Find drinkers that frequent only bars that serves some beer they like.

\[ Q(x) = \forall y. \text{Frequents}(x, y) \Rightarrow (\exists z. \text{Serves}(y, z) \land \text{Likes}(x, z)) \]

Find drinkers that frequent some bar that serves only beers they like.

\[ Q(x) = \exists y. \text{Frequents}(x, y) \land \forall z. (\text{Serves}(y, z) \Rightarrow \text{Likes}(x, z)) \]
More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y,z) \land \text{Likes}(x,z) \]

Find drinkers that frequent only bars that serves some beer they like.

\[ Q(x) = \forall y. \text{Frequents}(x, y) \Rightarrow (\exists z. \text{Serves}(y,z) \land \text{Likes}(x,z)) \]

Find drinkers that frequent some bar that serves only beers they like.

\[ Q(x) = \exists y. \text{Frequents}(x, y) \land \forall z.(\text{Serves}(y,z) \Rightarrow \text{Likes}(x,z)) \]

Find drinkers that frequent only bars that serves only beer they like.

Likes(drinker, beer)
Frequents(drinker, bar)
Serves(bar, beer)
More Examples

Find drinkers that frequent some bar that serves some beer they like.

\[ Q(x) = \exists y. \exists z. \text{Frequents}(x, y) \land \text{Serves}(y,z) \land \text{Likes}(x,z) \]

Find drinkers that frequent only bars that serves some beer they like.

\[ Q(x) = \forall y. \text{Frequents}(x, y) \Rightarrow (\exists z. \text{Serves}(y,z) \land \text{Likes}(x,z)) \]

Find drinkers that frequent some bar that serves only beers they like.

\[ Q(x) = \exists y. \text{Frequents}(x, y) \land \forall z.(\text{Serves}(y,z) \Rightarrow \text{Likes}(x,z)) \]

Find drinkers that frequent only bars that serves only beer they like.

\[ Q(x) = \forall y. \text{Frequents}(x, y) \Rightarrow \forall z.(\text{Serves}(y,z) \Rightarrow \text{Likes}(x,z)) \]