Tools & Models for Data Science

Relational Calculus

Chris Jermaine & Risa Myers

Rice University

Relational Calculus

- Nothing more than a First Order Logic predicate...
- Embedded within a set constructor

Writing RC Expressions

Start with the easy parts
Recall the syntax $\{t|P(t)\}$

1. Start with $\{\}$
2. and the "such that" bar, |
3. Then add in what you are looking for to the left of the |. This is a description of the tuples you want back
4. Then work on the right hand side
5. Provide a predicate that evaluates to True over all the variables that appear on the left
6. If the predicate evaluates to True, that tuple will be included in the result set

Some Notations and Conventions

Be sure to only include tuples that are in our relation(s)
Denoted as

- FREQUENTS(f)
  OR
- FREQ(f)
  OR
- $f \in FREQ$

Example: Cold Brew Drinkers

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- ? Query: Who goes to a cafe serving Cold Brew?

Example: Cold Brew Drinkers

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Who goes to a cafe serving Cold Brew?

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \exists(s)(\text{SERVES}(s) \wedge s.\text{COFFEE} = \text{'Cold Brew'} \wedge s.\text{CAFE} = f.\text{CAFE})\}$$

Example: Cold Brew Haters

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

? Query: Who has not gone to a cafe serving Cold Brew?

Example:Cold Brew Haters, Common Approach

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Who has not gone to a cafe serving Cold Brew?

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \neg\exists(s)(\text{SERVES}(s) \wedge s.\text{COFFEE} = \text{'Cold Brew'} \wedge s.\text{CAFE} = f.\text{CAFE})\}$$

Example: Cold Brew Haters, Common Approach

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Who has not gone to a cafe serving Cold Brew?

$$\{f.\mathrm{DRINKER} \mid \mathrm{FREQUENTS}(f) \wedge \neg\exists(s)(\mathrm{SERVES}(s) \\ \wedge\ s.\mathrm{COFFEE} = \text{'Cold Brew'} \wedge\ s.\mathrm{CAFE} = f.\mathrm{CAFE})\}$$

- Wrong! This gives us "Who has gone to a cafe that does not serve Cold Brew"
- In parenthesis we have cafes that serve 'Cold Brew' that someone has visited.
- Negating it, and checking for existence, gives us cafes that someone has visited that do NOT serve Cold Brew.

Walk–through Data

FREQUENTS

DRINKER	CAFE
Chris	A
Chris	B
Chris	C
Risa	A
Risa	B

SERVES

CAFE	COFFEE
A	Drip
A	Cold Brew
A	Espresso
B	Drip
C	Espresso

Common Approach

Who has gone to a cafe that does not serve 'Cold Brew'?

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \neg\exists(s)(\text{SERVES}(s) \\ \wedge\ s.\text{COFFEE} = \text{'Cold Brew'} \wedge\ s.\text{CAFE} = f.\text{CAFE})\}$$

For convenience, let's say:

hasCB $= \text{SERVES}(s) \wedge\ s.\text{CAFE} = f.\text{CAFE} \wedge\ s.\text{COFFEE} = \text{'Cold Brew'}$

Common Approach Steps

1. Start with the FREQUENTS table
2. Then look at matches in the SERVES table, where s.CAFE = f.CAFE
3. Evaluate the predicate
4. Is it TRUE?
   - If Yes, then include f.DRINKER in the result set
5. When f.CAFE = ‘B’, Risa gets included in the result set.

DRINKER	CAFE	CAFE	COFFEE	hasCB	¬hasCB	Result Set
Risa	B	B	Drip	F	T	{Risa}

6. When f.CAFE = ‘A’, Risa does NOT get add to the result set

<table>
<tr><th>DRINKER</th><th>CAFE</th><th>CAFE</th><th>COFFEE</th><th>hasCB</th><th>¬hasCB</th><th>Result Set</th></tr>
<tr><td rowspan="3">Risa</td><td rowspan="3">A</td><td>A</td><td>Drip</td><td rowspan="3">T</td><td rowspan="3">F</td><td rowspan="3">{ }</td></tr>
<tr><td>A</td><td>Cold Brew</td></tr>
<tr><td>A</td><td>Espresso</td></tr>
</table>

Common Approach Final Result

- To get the final result set, we union together all the results from the final column

						{ Risa }

- The final result set is { Risa } $\cup$ { } = { Risa }
- However, Risa shouldn't be in the result set, because she frequents Cafe A, which serves Cold Brew
- Issue: We want to look at ALL of the coffees served at ALL of the cafes Risa frequents all at one time

Correct Approach

Who has not gone to a cafe serving 'Cold Brew'?

- To answer this question, we need to introduce a second variable:
  $\{f_1.\text{DRINKER} \mid \text{FREQUENTS}(f_1) \wedge \neg\exists(f_2, s)(FREQ(f_2)$
  $\wedge\ SERVES(s) \wedge f_2.CAFE = s.CAFE$
  $\wedge\ s.COFFEE = \text{'Cold Brew'} \wedge f_1.DRINKER = f_2.DRINKER)\}$
- Again, for convenience, let's say:
  hasCB $= (FREQ(f_2)$
  $\wedge\ SERVES(s) \wedge f_2.CAFE = s.CAFE$
  $\wedge\ s.COFFEE = \text{'Cold Brew'} \wedge f_1.DRINKER = f_2.DRINKER)$

Correct Approach

In this case, by having the second variable, we are able to look at all the data for every place Risa frequents as a whole.

- Here, we have another variable, f_2
- We consider each drinker in turn from the FREQUENTS relation. Basically, we are using this table as our master list of drinkers, and are ignoring the CAFE attribute.
  Again, look just at Risa.
- Now, look at all the combinations of FREQUENTS and SERVES where the CAFE matches and the drinker is f_1.DRINKER

f_1.DRINKER	f_2.DRINKER	CAFE	CAFE	COFFEE	hasCB	¬hasCB	Result Set
Risa	Risa	A	A	Cold Brew	T	F	{ }
	Risa	A	A	Drip			
	Risa	A	A	Espresso			
	Risa	B	B	Drip			

- If there is any tuple where the Coffee is ‘Cold Brew’, we exclude the drinker
- Now, in this case, one of the cafes that Risa frequents does serve Cold Brew, so Risa is not added to the result set

Example: People Who Like to Drink Coffee

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- ? Query: Who goes to a cafe that serves a coffee they like?

Example: People Who Like to Drink Coffee

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Who goes to a cafe that serves a coffee they like?

$$\begin{aligned}\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \exists(s,l)(\text{SERVES}(s) \wedge \text{LIKES}(l) \\ \wedge\ s.\text{COFFEE} = l.\text{COFFEE} \\ \wedge\ s.\text{CAFE} = f.\text{CAFE} \\ \wedge\ l.\text{DRINKER} = f.\text{DRINKER})\}\end{aligned}$$

? We didn't refer to any table more than once. Why not?

Example: People Who Like to Drink Coffee

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Who goes to a cafe that serves a coffee they like?

$$\begin{aligned}\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \exists(s,l)(&\text{SERVES}(s) \wedge \text{LIKES}(l)\\ &\wedge s.\text{COFFEE} = l.\text{COFFEE}\\ &\wedge s.\text{CAFE} = f.\text{CAFE}\\ &\wedge l.\text{DRINKER} = f.\text{DRINKER})\}\end{aligned}$$

- We didn't refer to any table more than once. Why not?
- It wasn't needed since we didn't have any 'Always' or 'Never' predicates
- We were looking for 'Any'

Example: People Who Avoid Bad Cafes

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

? Query: Which people only go to cafes that serve a coffee they like?

Example: People Who Avoid Bad Cafes

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

? Query: Which people only go to cafes that serve a coffee they like?

Example: People Who Avoid Bad Cafes

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Which people only go to cafes that serve a coffee they like?

$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \forall(f_2)(\text{if } f_2 \text{ tells us a cafe that}$
$f.\text{DRINKER goes to then that cafe needs to serve a coffee that}$
$f.\text{DRINKER likes})\}$

Example: People Who Avoid Bad Cafes

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Which people only go to cafes that serve a coffee they like?

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \forall (f_2)(\text{FREQUENTS}(f_2)$$
$$\wedge f.\text{DRINKER} = f_2.\text{DRINKER} \rightarrow \exists (s,l)(\text{SERVES}(s)$$
$$\wedge \text{LIKES}(l) \wedge s.\text{CAFE} = f_2.\text{CAFE} \wedge l.\text{COFFEE} = s.\text{COFFEE}$$
$$\wedge l.\text{DRINKER} = f_2.\text{DRINKER}))\}$$

Implication

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \forall (f_2)(\text{FREQUENTS}(f_2)$$
$$\wedge\, f.\text{DRINKER} = f_2.\text{DRINKER} \rightarrow \exists (s,l)(\text{SERVES}(s)$$
$$\wedge\, \text{LIKES}(l) \wedge\, s.\text{CAFE} = f_2.\text{CAFE} \wedge\, l.\text{COFFEE} = s.\text{COFFEE}$$
$$\wedge\, l.\text{DRINKER} = f_2.\text{DRINKER}))\}$$

? Note: we invariably have a "$\rightarrow$" within a $\forall$ quantifier. Why?

Implication

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \forall(f_2)(\text{FREQUENTS}(f_2)$$
$$\wedge\, f.\text{DRINKER} = f_2.\text{DRINKER} \rightarrow \exists(s,l)(\text{SERVES}(s)$$
$$\wedge\, \text{LIKES}(l) \wedge\, s.\text{CAFE} = f_2.\text{CAFE} \wedge\, l.\text{COFFEE} = s.\text{COFFEE}$$
$$\wedge\, l.\text{DRINKER} = f_2.\text{DRINKER}))\}$$

- Note: we invariably have a "$\rightarrow$" within a $\forall$ quantifier. Why?
  - $\rightarrow$ is a logical IF–THEN statement

Implication

$$\{f.\text{DRINKER} \mid \text{FREQUENTS}(f) \wedge \forall(f_2)(\text{FREQUENTS}(f_2)$$
$$\wedge f.\text{DRINKER} = f_2.\text{DRINKER} \rightarrow \exists(s,l)(\text{SERVES}(s)$$
$$\wedge \text{LIKES}(l) \wedge s.\text{CAFE} = f_2.\text{CAFE} \wedge l.\text{COFFEE} = s.\text{COFFEE}$$
$$\wedge l.\text{DRINKER} = f_2.\text{DRINKER}))\}$$

Likes

Drinker	Coffee
Chris	Drip
Chris	Espresso
RIsa	Cold Brew
Risa	Drip

Frequents

Drinker	Cafe
Chris	A Cafe
Risa	Brew Joint
Chris	Double Trouble
Risa	Double Trouble

Serves

Cafe	Coffee
A Cafe	Espresso
Brew Joint	Espresso
Valhalla	Cold Brew
Valhalla	Espresso

f Drinker	f Coffee	f2 Drinker	f2 Café	f.Drinker = f2.Drinker	$\exists$	$\rightarrow$	$\forall$
Chris	~~Drip~~	Chris	A Cafe	1	1	1	1
		Risa	Brew Joint	0	na	1	
		Chris	Double Trouble	1	1	1	
		Risa	Double Trouble	0	na	1	
Risa	~~Cold Brew~~	Chris	A Cafe	0	na	1	0
		Risa	Brew Joint	1	1	1	
		Chris	Double Trouble	0	na	1	
		Risa	Double Trouble	1	0	0	

Wrap up

1. What is Relational Calculus?
2. Why does it matter?

? How can we use what we learned today?

? What do we know now that we didn't know before?