TAs

Presenter: Yuxi Liu

Q&A TA
Session 1(10:15am - 11:30am): Joyce Wang, Chengyu Wu, Tong Lin, Haibo Xiu
Session 2(1:45pm - 3:00pm): Zhe Wang, Justin Lim, Haibo Xiu
Projects Announcement

- Project group submissions are due in a week, please see pinned Ed post for more details
Check-in

- 01D 9/9 — 10:55-10:59am
  - Code:XXXX
- 02D 9/9 — 2:27-2:31pm
  - Code:XXXX
Roadmap

Two RAs

1. HW1-e (won't provide solution)
2. A problem similar to HW1-f (will provide solution)

You will get X minutes to get yourself familiar with the two problems and have a try at first
Two RAs

1. (hw1-e) Find drinker pairs (drinker1, drinker2) such that drinker1 and drinker2 visit the same bars which serve some beer they both like. [Output columns: drinker1, drinker2. Drinker1 should be the lexicographically smaller one. i.e. don't give results like (Ben, Amy) but (Amy, Ben) instead].

1. Find names of all drinkers who frequent EVERY bars that serve ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)
RA1: read the query

Find drinker pairs \((\text{drinker1, drinker2})\) such that drinker1 and drinker2 visit the same bars which serve some beer they both like. [Output columns: drinker1, drinker2. Drinker1 should be the lexicographically smaller one. i.e. don't give results like (Ben, Amy) but (Amy, Ben) instead].

What do we need?
1. A pair of drinkers
2. Under some conditions...
3. drinker 1 is lexicographically smaller than drinker 2 (like Amy < Ben)
RA1: schema

Find drinker pairs (drinker1, drinker2) such that drinker1 and drinker2 visit the same bars which serve some beer they both like. [Output columns: drinker1, drinker2. Drinker1 should be the lexicographically smaller one. i.e. don't give results like (Ben, Amy) but (Amy, Ben) instead].

What relations do we need?

- `drinker(name, address)`
- `bar(name, address)`
- `beer(name, brewer)`
- `frequents(drinker, bar, times_a_week)`
- `likes(drinker, beer)`
- `serves(bar, beer, price)`
RA1: what conditions?

drinker1 and drinker2 visit the same bars which serve some beer they both like

Let's extract the key words:
1. Drinker 1 visits some bar
2. The bar serves some beer
3. The beer is liked by drinker 1

We try to find a commonality between drinker 1 and drinker 2!

Recall that we always use JOIN to connect two separate information

What is the “separate information” in this scenario? —consider each drinker separately
Drinker 1 visits some bar
2. The bar serves some beer
3. The beer is liked by drinker 1
RA1: how to join them

drinker1 and drinker2 visit the same bars which serve some beer they both like

1. Drinker 1 visits some bar
2. The bar serves some beer
3. The beer is liked by Drinker1

Make these pairs of attribute equal by joining the three relations!
RA1: apply projection over the joined result

drinker1 and drinker2 visit the same bars which serve some beer they both like

But I don’t care times_a_week or price

Notes: projection will eliminate duplicates (this case has no duplicates)
RA1: (drinker, bar, beer)

Drinker1 and drinker2 visit the same bars which serve some beer they both like.

Intermediate results: each row is a triple (drinker, bar, beer)

Meaning: drinker frequent a bar that serves the beer the drinker likes.

How to get the pair of (drinker1, drinker2) then?
RA1: (drinker1, drinker2)

drinker1 and drinker2 visit the same bars which serve some beer they both like.

(join)

Duplicate

Join

frequencies(drinker, bar, times_a_week)
likes(drinker, beer)
serves(bar, beer, price)

drinker 1 is lexicographically smaller than drinker 2 (like Amy < Ben)
RA2: EVERY & ONLY

Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

Consider the counter-example!
—since it is difficult to figure out the valid drinkers directly

Counter-examples: drinkers who frequent SOME bar(s) that serve beer they DON’T like
—use (ALL - counter-examples) to get the valid drinkers
Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

Counter-examples: drinkers who frequent SOME bar(s) that serve beer they DON’T like
RA2: counter examples

Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

Counter-examples: drinkers who frequent SOME bar(s) that serve beer they DON'T like

\[
\text{Set Difference} \setminus \text{diff}
\]

(drinker, bar, beer):

- drinker frequents some bar that serve some beer

(drinker, bar, beer):

- drinker frequents some bar and likes some beer
RA2: counter examples

Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

Counter-examples: drinkers who frequent SOME bar(s) that serve beer they DON’T like

\texttt{\textbackslash project\_\{drinker, bar, beer\} (}
\texttt{frequents \textbackslash join serves)
\texttt{)}
\texttt{diff}
\texttt{\textbackslash project\_\{drinker, bar, beer\} (}
\texttt{frequents \textbackslash join likes)
\texttt{)}
\texttt{(drinker, bar, beer):}
\texttt{drinker frequents some bar that serve some beer the drinker dislikes}
RA2: projection

Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

Counter-examples: drinkers who frequent SOME bar(s) that serve beer they DON'T like

```
\project\_\{drinker\} ( 
  \project\_\{drinker, bar, beer\} ( 
    frequents \join serves 
  )
\diff 
\project\_\{drinker, bar, beer\} ( 
  frequents \join likes 
  )
)
```
RA2: **ALL**

Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

\((\text{ALL} - \text{counter-examples})\) to get the valid drinkers

\[ \text{ALL} = \text{\textbackslash project}_{\text{drinker}} \text{ frequents} \]

Set difference (\textbackslash diff)
RA2: solution

Find names of all drinker(s), EVERY bar frequent by the drinker(s) serves ONLY beer they like. (Notes: Your answer will consider the drinkers that frequent at least one bar. Using frequents, likes, and serves is enough for this problem.)

\[ \text{\textbackslash project\{drinker\} frequents} \]
\[ \text{\textbackslash diff} \]
\[ \text{\textbackslash project\{drinker\} (} \]
\[ \text{\textbackslash project\{drinker, bar, beer\}(frequents \text{\textbackslash join\{frequents.bar = serves.bar\} serves)} \]
\[ \text{\textbackslash diff} \]
\[ \text{\textbackslash project\{drinker, bar, beer\}(frequents \text{\textbackslash join\{frequents.drinker = likes.drinker\} likes)} \]
\]

\[ \text{\textbackslash frequents(drinker, bar, times\_a\_week)} \]
\[ \text{\textbackslash likes(drinker, beer)} \]
\[ \text{\textbackslash serves(bar, beer, price)} \]
Summary

- Get yourself familiar with the operators
- Get yourself familiar with the schema
- Read the query
- Extract the keywords
- Be sensitive to ONLY/EVERY/NOT
- Try to think about the RA query from both directions
  - Valid
  - All - Invalid