I/O Examples + Description = SQL

Chenglong Wang

Observation

- One common way users ask question on online SQL forums is to demonstrate the problem by providing:
  - An short English description
  - An I/O example pair

- These queries perform non-trivial tasks and contain advanced features:
  - Tasks: Moving average, pivot, argmax, ancestor lookup
  - Features: exist, aggregation, subquery

Why description + I/O examples?

- End-users:
  - Description can be partial and simple: do not need to worry if descriptions is complete in describing the query semantics
  - IO help people to verify the correctness of their answer
- Synthesizer:
  - Examples and descriptions provide different constraints in the task
    - Descriptions: critical but partial: express certain relationship between some columns
    - Examples: complete but complex constraints are hard to infer
  - Do not fully depend on quality of the description

Related works – NL to Query

- F.Li et al: Constructing an Interactive Natural Language Interface for Relational Databases
  - Input as natural languages, map parse tree to query tree
  - Interact with users to resolve description ambiguities

- S.Guwalni et al: NLyze paper
  - Pattern based translation from NL to PL

Related works – I/O to Query

- S.Zhang et al, Automatically synthesizing SQL queries from input-output examples
  - Input as I/O examples
  - Synthesize query through heuristic search
  - Support a subset of SQL features:
    - Aggregation, in, having, join, but not sub-query
- Rishabh, Sumit: Learning Semantic String Transformations from Examples
  - Semantically join + syntax manipulation

NL + PL

- Sumit: Compositional Program Synthesis from Natural Language and Examples
  - Syntax manipulation with natural language description
- Manshadi et al: Integrating Programming by Example and Natural Language Programming
  - NLPBE probabilistic model

Our challenges

- Existing works are quite powerful in solving such questions, we need to:
  - Deal with poor and partial English description
    [H.V. Jagadish] [Sumit]
  - Support sub-query, moving average, pivot
    [Sai Zhang]
  - Support join [Rishabh]

A demonstration on
NL+Example → SQL
with NL interaction

Retrieving the last record in each group

description

```
That is, the last record in each group should be returned.
```

input

```
| Id |  Name |  Other_Columns |
|----------------------------|
| 1  |  A    |   A_data_1     |
| 2  |  A    |   A_data_2     |
| 3  |  A    |   A_data_3     |
| 4  |  B    |   B_data_1     |
| 5  |  B    |   B_data_2     |
| 6  |  C    |   C_data_1     |
```

output

```
| col1 | col2 | col3     |
|-----------------------|
| 3    | A    | A_data_3 |
| 5    | B    | B_data_2 |
| 6    | C    | C_data_1 |
```

Step 1: Extract constraints

Description:

- That is, the last record in each group should be returned.
  - Record -> row
  - Each group -> aggregation on group
  - Last -> aggregation function

Example:

- Filter (T, f)

Step 2: Synthesizing candidate programs

```
f r T :=
 r.attr1 =  phi(r.attrlist[a1, a2, a3],
                    agrfun(attr3, attrlist[a1, a2, a3]))
```

attri : some attribute of a row

agrfun: some aggregation function

Step 3: Refine results

f r T :=

r.attr1 = phi(r.attrlist[a1, a2, a3],
agrfun(attr3, attrlist[a1, a2, a3]))

F1 := r.id = phi(r.Name, max(id, Name))

F2 := r.otherColumn = phi(r.Name, max(otherColumn, Name))

Step 4: Interact with users for top candicates

F1 := r.id = phi(r.Name, max(id, Name))

"That is, the last record in each group should be returned."

➔ "That is, the record with max id in each Name group should be returned"

F2 := r.otherColumn = phi(r.Name, max(otherColumn, Name))

"That is, the last record in each group should be returned."

➔ "That is, the record with max otherColumn in each Name group should be returned"

Step 5: translation to SQL

- Filter (T, f r => r.id = phi(r.Name, max(id, Name)))

- Select t1.id, t1.Name, t1.otherColumn
  from T t1
  where t1.id = (
  select max(id)
  from T t2
  where t2.Name = t1.Name
)

Proposal

- Plan A: NL + PL → SQL with NL interaction
  - Need to dive into advanced queries and poor english specification
- Plan B: A framework to synthezie SQL from multiple source
  - Demonstration + I/O example
  - NL + Demonstration
  - NL + I/O
  - NL + old SQL query
  - Too much? How to present it?
- Plan C: New interface for user to program SQL
  - More HCI research?

Some additional examples

Demonstration through I/O and a computation explaination

I am trying to filter out some records but cannot make it.

Take this table as example,

```
ID   Status    Name
1      NULL    ItemA
2      2       ItemB
3      2       ItemA
4      NULL    ItemC
```

I try to show name as result with only NULL status and if for an item with status 2, do not show it. In this case, the result is only `ItemC` because even one record of `ItemA` has NULL status, there is another record of ItemA with status 2.

I am stuck here.

I have a query which joins multiple tables. For eg.

```
Select u.username,r.role,s.salary
from user u, roles r, salary s
where u.userid = s.userid
and r.roleid = u.roleid.
```

Now I need to append the condition to get the data only if the eligibility flag is true, which can be present in two tables (default eligibility and user eligility) . If the data is in user_eligiblity i need to check from user_eligility else i need to get from default eligibility.

```
Select u.username,r.role,s.salary
from user u, roles r, salary s, default_eligility de, user_eligiblity ue
where u.userid = s.userid
and r.roleid = u.roleid
and (if ue has data check if ue.eligible=y and ue.userid = u.userid
     else get data from de.eligibity='y' and de.roleid = u.roleid)
```

User Eligibility may or may not have have rows for the user. If it does not have have to take it from default table.

mysql sql oracle

share improve this question

asked 1 hour ago

googytech

65 ● 1 ● 9

1) tag the appropriate database, not two 2)post some sample data and the expected result – vkp 1 hour ago

Use explicit OUTER JOINs and COALESCE() function. – PM 77-1 1 hour ago

@vkp - The solution to this problem can be written in ANSI SQL and work on both databases. – PM 77-1 1 hour ago

1 Learn how to use explicit join syntax. Aaron Bertrand did some **writting** about it – Juan Carlos Oropeza 1 hour ago

I have this query

```
SELECT
  schedule.JOB_NUMBER, max(schedule_milestone.actual_start_date), milestone.milestor
FROM
  schedule
  inner join
   schedule_milestone on (schedule.schedule_id = schedule_milestone.schedule_id) inr
   milestone on (milestone.milestone_id = schedule_milestone.milestone_id)
--where schedule.job_number = '024MGV002' using this for testing
group by schedule.job_number, milestone.milestone_name
```

I return the jobNumber, MaxDate and milestone name.

What I want it to return is jobnumber, MaxDate, and milestone name, but I only want one line for each jobnumber. I want the milestone that has the most recent date.

So jobnumber 1234 has 4 different milestones. I want the jobnumber, date and milestone name for the milestone that has the Max date.