

JULY 31

Using Provsq. <https://github.com/PierreSenellart/provsq>

installProvsq

data

381 bytes



The ProvSQL system currently supports proper management of provenance attached to SQL queries, in the form of a provenance circuit, suitable both for regular Boolean provenance, arbitrary semiring provenance, with or without monus (m-semiring), in the free m-semiring, or specialized to any m-semiring of choice. It also supports where-provenance and probability computation from the provenance, through a variety of methods.

The following SQL queries are currently supported.

- Regular SELECT-FROM-WHERE queries (aka conjunctive queries with multiset semantics)
- JOIN queries (regular joins only; outer, semijoins, and antijoins are not currently supported)
- SELECT queries with nested SELECT subqueries in the FROM clause
- GROUP BY queries (without aggregation)
- SELECT DISTINCT queries (i.e., set semantics)
- UNION's or UNION ALL's of SELECT queries
- EXCEPT of SELECT queries

Database:

students (
name,
age,

```
    major,  
    gender  
)
```

```
times (  
    name,  
    time  
)
```

Each table in provsql has their own mappings of tuples to provenance tokens.

Table 'students':

```
provsql=# select * from students;  
name | age | major | gender | provsql  
-----+-----+-----+-----+-----  
joe   | 18  | CS    | M      | 23783654-5407-44be-88df-726a2c6ba94e  
jack  | 20  | ECE   | M      | 6208981a-48a6-483a-a4c3-5fc76f00ee0d  
nacy  | 21  | CS    | W      | c05956e6-fbf5-47ec-aa9c-60618d76c163  
neal  | 24  | CS    | M      | 2ac386fb-1606-4017-bd19-6647dbdee948  
(4 rows)
```

'students_mapping': map students' names to tokens. All mapping are created by users according to their needs. I could also create a mapping of gender to tokens.

```
provsql=# select * from students_mapping;
value |
-----+-----
joe   | 23783654-5407-44be-88df-726a2c6ba94e
jack  | 6208981a-48a6-483a-a4c3-5fc76f00ee0d
nacy  | c05956e6-fbf5-47ec-aa9c-60618d76c163
neal  | 2ac386fb-1606-4017-bd19-6647dbdee948
(4 rows)
```

'times' :

```
provsql=# select * from times;
time | name |
-----+-----
10   | joe  | 22decb5e-ec9a-49bc-b38b-ba281e8ebd0c
11   | joe  | 9b3470fc-1bb6-4525-8810-c6d410e8b957
3    | nacy | de6be744-72dc-4097-91ff-5de9ea2bffa2
5    | nacy | 5f62aabd-2cbe-45e7-b94d-2c1d3f5c8f82
21   | jack | ca4d5c7b-a475-451f-9623-7ca4b7e9cd61
27   | jack | bf31ba8b-9b17-48aa-a675-bc12920727d5
16   | neal | 3c2805e0-c60e-4ce8-bc3e-0011cdd5d4ec
17   | neal | c80f47d5-0806-4dee-8908-2b3278f84263
(8 rows)
```

'times_mapping':

```
provsql=# select * from times_mapping;
value |
-----+-----
joe   | 22decb5e-ec9a-49bc-b38b-ba281e8ebd0c
joe   | 9b3470fc-1bb6-4525-8810-c6d410e8b957
nacy  | de6be744-72dc-4097-91ff-5de9ea2bffa2
nacy  | 5f62aabd-2cbe-45e7-b94d-2c1d3f5c8f82
jack   | ca4d5c7b-a475-451f-9623-7ca4b7e9cd61
jack   | bf31ba8b-9b17-48aa-a675-bc12920727d5
neal  | 3c2805e0-c60e-4ce8-bc3e-0011cdd5d4ec
neal  | c80f47d5-0806-4dee-8908-2b3278f84263
(8 rows)
```

Test on Provsql:

```

select formula(provenance(), 'times_mapping'),
s.name
from times as t join students as s
on s.major='CS' and s.name=t.name;

```

```

provsq1=# select formula(provenance(), 'times_mapping'), s.name from times as t join students as
s on s.major='CS' and s.name=t.name;
 formula | name | provsq1
-----+-----+-----
(joe @ 1) | joe | 9ea2c4f3-c928-52bc-aec8-502d08251ee4
(joe @ 1) | joe | 414c0258-dd28-53ff-955c-236fd84167ef
(nacy @ 1) | nacy | bac2c4b7-3f03-54ed-b8a8-11b3bf450f30
(nacy @ 1) | nacy | b26a4bc0-fc00-5a77-87fc-b0c4fc788a47
(neal @ 1) | neal | a70741b2-34d9-53b1-9817-cbd14b6122c0
(neal @ 1) | neal | c1e85054-84ec-5d07-9419-630015401666
(6 rows)

```

This is how-provenance, and it shows exactly which tuple contributes to the output.

The '1' means another table.

It seems that it can only show provenance of one mapping, but this can be fixed by running provenance on each mapping participating in the query.

```

select formula(provenance(),
'students_mapping'), s.name
from times as t join students as s
on s.major='CS' and s.name=t.name;

```

```

provsq1=# select formula(provenance(), 'students_mapping'), s.name from times as t join students
as s on s.major='CS' and s.name=t.name;
 formula | name | provsq1
-----+-----+-----
(1 @ joe) | joe | 9ea2c4f3-c928-52bc-aec8-502d08251ee4
(1 @ joe) | joe | 414c0258-dd28-53ff-955c-236fd84167ef
(1 @ nacy) | nacy | bac2c4b7-3f03-54ed-b8a8-11b3bf450f30
(1 @ nacy) | nacy | b26a4bc0-fc00-5a77-87fc-b0c4fc788a47
(1 @ neal) | neal | a70741b2-34d9-53b1-9817-cbd14b6122c0
(1 @ neal) | neal | c1e85054-84ec-5d07-9419-630015401666
(6 rows)

```

We could combine the two results and get the final how-provenance.

Queries: