B561 Assignment 7

Due Friday, November 2, 2018

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems. Submit a file Assignment7.sql with your solutions.

In the following questions, use the data provided for the student, majors, book, cites, buys relations.

The purpose of this assignment is to work with complex-object relations.

1. Consider following function `setunion` which computes the set union of two sets represented as arrays. Notice that this function is defined poly-morphically.

```
create or replace function setunion(A anyarray, B anyarray)
    returns anyarray as
$$ select ARRAY(select unnest(A)
                union
                select unnest(B));
$$ language sql;
```

   (a) In the style of the `setunion` function, write a function `setintersection` that computes the intersection of two sets.

   (b) In the style of the `setunion` function, write a function `setdifference` that computes the difference of two sets.

   You will need to use these functions in the remaining problems.

   You can also make use of the function `memberof` which verifies if an object x is in a set S. (Again this function is defined poly-morphically.)

```
create or replace function memberof(x anyelement, A anyarray)
   returns boolean as
$$
  select x = SOME(A);
$$ language sql;
```

2. Consider the view `student_books(sid,books)` which associates with each student the set of books he or she buys.

```
create or replace view student_books as
   select s.sid as sid, array(select t.bookno
                              from   buys t
                              where  t.sid = s.sid
                              order by bookno) as books
   from   student s
   order by sid;
```

   Observe that it is possible that a student does not buy any books.

   (a) Define a view `book_students(bookno,students)` which associates with each book the set of students who bought that book. Observe that there may be books that are not bought by any student.

   (b) Define a view `book_citedbooks(bookno,citedbooks)` which associates with each book the set of books that are cited by that book. Observe that there may be books that cite no books.

   (c) Define a view `book_citingbooks(bookno,citingbooks)` which associates with each book the set of books that cite that book. Observe that there may be books that are not cited.

   (d) Define a view `major_students(major,students)` which associates with each major the set of students who have that major. (You can assume that each major has at least one student.)

   (e) Define a view `student_majors(sid,majors)` which associates with each student the set of his or her majors. Observe that there can be students who have no major.

   Test that each of these views work properly. You will need to use them in the subsequent problems.

3. Using the above defined functions, views, and the `book` and `student` relations, specify the following queries in SQL. You can also use array aggregation, the `cardinality` function, and the UNNEST operator.

   Observe that you are **not permitted** to use (expose) the `buys`, `cites`, and `major` relations. (Of course these relations are used, but they are encapsulated (hidden) inside the views.)

   For example, a query such as

```
select t.sid
from   buys t, book b
where  t.bookno = b.bookno and price < 50
```

   is **not** permitted. However, a query such as

```
select sb.sid
from   student_books sb, book b
where  memberof(b.bookno, sb.books) and price < 50
```

   is permitted. (By the way, these queries are actually equivalent.)

   (a) Find the sid of each student who bought precisely 2 books.

   (b) Find the sid of each student who bought all the books bought by the student with sid 1001.

   (c) Find the bookno of each book that cites fewer than 2 books that each cost more than $30.

   (d) Find the bookno and title of each book that was not only bought by students who major both in 'CS' and in 'Math'.

   (e) Find the sid-bookno pairs (s, b) pairs such student s bought book b and such that book b is cited by at least two books that cost less than $50.

   (f) Find the tuple (*students*) where students is the set of students who major in 'CS' and in 'Math'.

   (g) Find each pair (s, majors) where s is the sid of a student who bought none of the books bought by student with sid 1001 and where $majors$ is the set of majors of the student s.

   (h) Find the tuple (*books*) where books is the set of books bought by 'CS' students.

   (i) Find the tuple (*students*) where students is the set of students who bought books that cites at least two books.

   (j) Find the pairs $(b, students)$ where b is a bookno and *students* is the set CS students who bought that book.

(k) Find the sids of students who major in 'CS' and who did not buy any of the books bought by the students who major in 'Math'."

(l) Find the pairs (b_1, b_2) of different booknos of books that are bought by the same students.

(m) Find the pairs (b_1, b_2) of different booknos of books such that there are fewer Math students who bought book b_1 than there are 'CS' students who bought book b_2.

(n) Find the sid of each student who bought all but one book that cost more than \$50.