

Assignments 5

Due Monday, October 23, 2017

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems. Submit a file Assignment5.sql with your solutions and a file OutputAssignment5.txt with your outputs to the queries.

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 polymorphically.

```
create or replace function setunion(A anyarray, B anyarray) returns anyarray as
$$
with
    Aset as (select UNNEST(A)),
    Bset as (select UNNEST(B))
select array( (select * from Aset) union (select * from Bset) order by 1);
$$ 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 polymorphically.)

```
create or replace function memberof(x anyelement, S anyarray) returns boolean as
$$
select x = SOME(S)
$$ 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, 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 cites 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 b.price < 50
```

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

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

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

- (a) Find the bookno of each book that is cited by at least two books that cost less than \$50.

- (b) Find the bookno and title of each book that was bought by a student who majors in CS and in Math.
- (c) Find the bookno of each book that is cited by exactly one book.
- (d) Find the sid of each student who bought all books that cost more than \$50.
- (e) Find the sid of each student who bought no book that cost more than \$50.
- (f) Find the sid of each student who bought only books that cost more than \$50.
- (g) Find the sid of each students who bought exactly one book that cost less than \$50.
- (h) Find the bookno of each book that was not bought by any students who majors in CS.
- (i) Find the Bookno of each book that was not bought by all students who majors in Anthropology.
- (j) Find sid-bookno pairs (s, b) such that not all books bought by student s are books that cite book b .
- (k) Find sid-bookno pairs (s, b) such student s only bought books that cite book b .
- (l) Find the pairs (s_1, s_2) of different sids of students that buy the same books.
- (m) Find the pairs (s_1, s_2) of different sids of students that buy the same number of books.
- (n) Find the bookno of each book that cites all but two books. (In other words, for such a book, there exists only two books that it does not cite.)