Assignment 3

Due September 28, 2017 by 11:59pm

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems, as well as the results of running these statements. Turn in a file Assignment3.sql with your solutions (where necessary, include comments explaining your solutions). Also turn in a file Assignment3.output.txt with the outputs of running your SQL statements.

1 Miscellaneous Problems

For the problems in this section, you can use views (including parameterized view, i.e., user-defined function) but you can not use aggregate functions nor the `GROUP BY` and `HAVING` clause. You can also not use the `INNER JOIN` (or other joins) operators.

1. (10 points) Let $A(x)$ and $B(x)$ be two unary relation schemas that represent a set A and a set B, respectively. (The domain of x in INTEGER.)

   (a) Write a SQL statement that determines whether it is true or not if $A-B$ is empty, $B-A$ is empty, and $A \cap B$ is empty. Make appropriate use of the SQL operators INTERSECT and EXCEPT in your SQL statement.

   (b) Repeat problem 1a but this time you can not use INTERSECT nor EXCEPT. However, you can use IN, NOT IN, EXISTS and NOT EXISTS.

   For example, if $A = \{1, 2, 3\}$ and $B = \{1, 3\}$ then your SQL statements should produce the output:

```
 empty_a_minus_b | empty_b_minus_a | empty_a_intersection_b
-----------------+-----------------+------------------------
 f               | t               | f
(1 row)
```

   because $A - B = \{2\}$, $B - A = \{\}$, and $A \cap B = \{1\}$.

   Your solution should work for arbitrary A and B.

2. Let $A(x)$, $B(x)$ and C be three unary relation schemas that represent sets A, B and C of integers.

   For each of the following subproblems, write two SQL statements to determine the specified property on the sets A, B, and C.

   In the first SQL statement, you should make appropriate use of the set operations `UNION`, `INTERSECT`, or `EXCEPT`.

   In the second SQL statement, you are not allowed to use these set operators. However, you can use the predicates `IN`, `NOT IN`, `EXISTS`, or `NOT EXISTS`.

   (a) (10 points) Determine whether or not $A \cap B \neq \{\}$.
   For example, if $A = \{1, 2\}$ and $B = \{1, 4, 5\}$ then the result of your statement should be

```
 answer
--------
 t
(1 row)
```

   If, however, $A = \{1, 2\}$ and $B = \{3, 4\}$ then the result of your statement should be

```
 answer
--------
 f
(1 row)
```

   (b) (10 points) Determine whether or not $A \cap B = \{\}$.

   (c) (10 points) Determine whether or not $A \subseteq B$.

   (d) (10 points) Determine whether or not $A \not\supseteq B$.

   (e) (10 points) Determine whether or not $A \neq B$.

   (f) (10 points) Determine whether or not $|A - B| < 2$.

   (g) (10 points) Determine whether or not $(A \cap B) \not\supseteq C$.

   (h) (10 points) Determine whether or not $|A \cap (B \cup C)| = 2$.

3. Let $p(x)$ and $q(x)$ be 2 polynomials with integer coefficients. Let P(coefficient, degree) and Q(coefficient, degree) be two binary relations representing $p(x)$ and $q(x)$, respectively. E.g., if $p(x) = 3x^3 - 2x^2 + 5$ then its representation in the relation P is as follows:

P

coefficient	degree
3	3
−2	2
5	0

   (a) (10 points) Write a SQL statement that computes a binary relation representing the addition of $p(x)$ and $q(x)$, i.e., the polynomial $p(x) + q(x)$. Your solution should work for arbitrary polynomials.

   Consider $p(x) = 2x^2 - 5x + 5$ and $q(x) = 3x^3 + x^2 - x$. Then $p(x) + q(x) = 3x^3 + (2+1)x^2 + (-5-1)x + 5 = 3x^3 + 3x^2 - 6x + 5$.

   (b) (10 points) Write a SQL statement that computes a binary relation representing the multiplication of $p(x)$ and $q(x)$, i.e., the polynomial $p(x) * q(x)$. For this problem, you must use the SUM aggregate function. Your solution should work for arbitrary polynomials.

   Consider $p(x) = 2x^2 - 5x + 5$ and $q(x) = 3x^3 + x^2 - x$. Then $p(x) * q(x) = 6x^5 + (2-15)x^4 + (-2-5+15)x^3 + (5+5)x^2 - 5x = 6x^5 - 13x^4 + 8x^3 + 10x^2 - 5x$.

4. Consider the relation schema Point(pid, x, y) of a relation of points in the plane. The attribute pid (of type INTEGER) is the identifier of a point, and the attributes x and y, both of type FLOAT, are its x and y coordinates.

   (a) (10 points) Write a SQL query that returns the (p_1, p_2) pairs of different pids of points that are closest in distance from each other. Recall that if $p_1 = (x_1, y_1)$ and $p_2(x_2, y_2)$ are two points in the plane, then the distance between them is given by the formula

   $$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

   For example if you have the following points,

   ```
   pid | x | y
   ----+---+---
     1 | 0 | 0
     2 | 0 | 1
     3 | 1 | 0
   ```

   Then your answer should be

   ```
   p1  | p2
   ----+---+---
   1   | 2
   2   | 1
   1   | 3
   3   | 1
   ```

   (b) (10 points) Continuing with the setup in problem 4, determine the triples of different points (p_1, p_2, p_3) that are collinear. p_1, p_2, and p_3 are collinear if they lie on the same line.

2 Parameterized views: user-defined function returning sets of records

Use the files student, majors, book, cites, and and buys from Assignment 2. For the problems in this section, you can use views (including parameterized view, i.e., user-defined function) but you can not use aggregate functions nor the `GROUP BY` and `HAVING` clause. You can also not use the `INNER JOIN` (or other joins) operators.

5. (a) (10 points) Write a function `booksBoughtbyStudent(sid int, out bookno int, out title VARCHAR(30), out price integer)` that takes a student sid as input and returns the book information of books bought by that student.

   (b) (10 points) Test this function on the student with sid 1001 and on the student with sid 1015.

   (c) Using this function, write the following queries:

      i. (10 points) "Find the sids and names of students who bought exactly one book that cost less than $50."

      ii. (10 points) "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'."

      iii. (10 points) "Find the pairs of different student sids (s1,s2) such that student s1 and student s2 bought the same books.

6. (a) (10 points) Write a function `studentWhoBoughtBook(bookno int, out student sid, out sname VARCHAR(15))` that takes a bookno as input and returns the student information of students who bought that book.

   (b) (10 points) Test your function on the book with bookno 2001 and that with bookno 2010.

   (c) (10 points) Using this function and the `booksBoughtbyStudent` function from problem 5a write the query "Find the booknos of books bought by a least two CS students who each bought at least one books that cost more that $30."

3 Writing queries using aggregate functions

In the following problem you should use user-defined functions and aggregate functions.

Write the following queries in SQL.

7. (a) (10 points) Write a function `numberOfBooksBoughtbyStudent(sid int)` that returns the number of books bought by a student given his of her sid.

   (b) Using this function write the following queries

      i. (10 points) "Find for each CS student who bought more than 2 books, the number of books bought by that student."

      ii. (10 points) "Find student who bought fewer books than the number of books bought by student who major in CS."

      iii. (10 points) "Find the pairs of different student sids (s1,s2) who bought the same number of books."

8. In the following problems use aggregate functions. You should also consider introducing user-defined functions. Formulate the following queries:

   (a) (10 points) "For each student, find the number of books bought by that student, provided their collective cost is less that $300."

   (b) (10 points) "Find the Bookno's of books that cite at least 2 books and are cited by fewer than 4 books."

   (c) (10 points) "Find the booknos and titles of the cheapest books." Use the MIN aggregate function in this problem.

   (d) (10 points) "Find each (s, b) pair where s is the Sid of a student and b is the Bookno of a book whose price is the cheapest among the books bought by that student.

   Use the MIN aggregate function in this problem.

   (e) (10 points) "Find each department whose majors, collectively, spend the most on books."

   (f) (10 points) Find the Bookno's of books that where bought by all students who major in 'Biology'.

   (g) (10 points) "Find the tuples (s1,s2) where s1 and s2 are different sids of students and such that student s1 and student s2 bought exactly one book in common."

   (h) (10 points) "Write a user-defined function AllbutK(k int) that returns each pair of students (s1,s2) such that s1 buys **all but k** books that are also bought by student s2. " (Notice that when k = 0, this function finds each pair of students such that s1 only buys books that were also bought by s2.)