Assignment 3

Due Thursday February 13, 2020 by 11:45pm EST

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems. Turn in a `.sql` file with your solutions on IUCanvas as well as a .txt with outputs generated by your solutions. (I.e., use the standard turn-in procedure for assignments.) We furthermore encourage, but not require you to include comments that explain your solutions.

1 Queries with expressions and functions; Boolean queries

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

1. 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 is `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 UNION, INTERSECT and/or EXCEPT in your SQL statement. In this problem, you can not use the set predicates of SQL.

   (b) Repeat problem 1a but this time you can not use UNION, INTERSECT and/or EXCEPT. However, you can use the IN, NOT IN, EXISTS and/or 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 $A - B = \{2\}$, $B - A = \emptyset$, and $A \cap B = \{1\}$.

   Your solution should work for arbitrary A and B.

2. SQL uses 3-valued logic where it concerns the treatment of `NULL` (i.e., the value "unknown"). (Read your textbook or search the web for relevant information.) Consider relation schemas p(value), q(value), and r(value) where the type of the attribute `value` is `boolean`. In other words, p, q,

and r represent propositional (boolean) variables. Populate each of these relations with the possible values `true`, `false`, and `NULL` (i.e., the value `unknown`).

In other words, p, q, and r are as follows:

p

value
t
f
NULL

q

value
t
f
NULL

r

value
t
f
NULL

Write a SQL statement that generates the 3-valued truth table for the Propositional Logic formula

$$(p \to q) \to (\neg r)$$

Your statement should return the following answer:

```
 p | q | r | value
---+---+---+-------
 t | t | t | f
 t | t | f | t
 t | t |   |
 t | f | t | t
 t | f | f | t
 t | f |   | t
 t |   | t |
 t |   | f | t
 t |   |   |
 f | t | t | f
 f | t | f | t
 f | t |   |
 f | f | t | f
 f | f | f | t
 f | f |   |
 f |   | t | f
 f |   | f | t
 f |   |   |
   | t | t | f
   | t | f | t
   | t |   |
   | f | t |
   | f | f | t
   | f |   |
   |   | t |
   |   | f | t
   |   |   |
(27 rows)
```

The blank characters in this table represent the `NULL` (unknown) value.

3. 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) 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
```

Notice that the pairs $(2, 3)$ and $(3, 2)$ are not in this answer because the distance between points 2 and 3 is larger than the distance between each pair of points reported in the above answer.

(b) Determine the triples of different points (p_1, p_2, p_3) that are *collinear*. Recall that points p_1, p_2, and p_3 are collinear if they lie on the same straight line.

4. Let $R(A, B, C)$ be a relation schema. The attributes A, B, and C have as their domain `INTEGER`.

   (a) Write a SQL statement that determines whether or not A is a *primary key* for the relation R.

   (b) Provide two R instances where one of these instances has A as primary key and the other does not have A has a primary key. Of course, the determination of the primary key property for these instances needs to be verified by using your SQL statement for this problem.

2 Queries using aggregate functions

In the problems in this section, you will practice working with aggregate functions. As explained in the lecture on aggregate functions and partitioning, there are various approaches to accomplish this:

- the GROUP BY map COUNT method;
- the user-defined COUNT FUNCTION method; and
- the SELECT COUNT-expression method.

To solve the problems in this section, you can freely use any of these methods.

For some of the problems in this section, use the relations `student`, `major`, `book`, and `buys` that can be found in the `data.sql` file.

5. Let M be an $n \times n$ matrix of integers (n is a positive integer).

   For $i, j \in [1, n]$, we will denote by $M[i, j]$ the element in matrix M at row i and column j.

   Given an $n \times n$ matrix M, denotes by M^2 the $n \times n$ the matrix define such that for $i, j \in [1, n]$, row i and column j of M^2 is defined by the formula

$$M^2[i, j] = \sum_{k=1}^{n} M[i, k]M[k, j].$$

   The matrix M can be represented using a ternary relation `M` with schema (`row`, `colmn`, `value`)[1] as follows: for each pair $i, j \in [1, n]$, tuple (i, j, v) in `M` represents that $v = M[i, j]$.

   For example if M is the 3×3 matrix

$$M = \begin{matrix} 1 & 2 & 3 \\ 1 & -3 & 5 \\ 4 & 0 & -2 \end{matrix}$$

   then `M` is the following relation of 9 tuples:

   M

row	colmn	value
1	1	1
1	2	2
1	3	3
2	1	1
2	2	−3
2	3	5
3	1	4
3	2	0
3	3	−2

   Let M be an $n \times n$ matrix represented by the relation `M`.

   Write a SQL query that computes a relation over schema (`row`, `column`, `value`) that represents the matrix M^4, where M^4 is defined as the matrix $(M^2)^2$.

   Your solution should work for any $n \geq 1$.

[1]Notice that we use the attribute name '`colmn`' since the word '`column`' is a reserved word in PostgreSQL.

6. Let $A(x)$ be a unary relation schema that represent a set A of integers. (So the domain of x in `INTEGER`.)

   Define the following equivalence relation on A: for each pair of elements x and y in A, we say that x and y are equivalent if $x \,\texttt{mod}\, 4 = y \,\texttt{mod}\, 4$. (Recall that $x \,\texttt{mod}\, 4$ is the remainder of the integer division of x by 4.)

   Given a relation $A(x)$ storing a set of integers, determine for each possible remainder value, i.e. 0, 1, 2, or 3, the number of elements in A that are equivalent relative to these values.

7. Let $A(x)$ be a unary relation schema that represent a **bag** A of integers. (So the domain of x in `INTEGER`.)

   For example, we may have the following bag:

A
x
5
3
3
2
1
3
5

   Write a SQL query that coerces such a bag into a set which contains the same elements as the bag. (In this problem, you can not use the `DISTINCT` clause.)

   In other words, for the bag above, the query should output the set

x
1
2
3
5

8. Formulate the following queries in SQL:

   (a) "Find the bookno's and titles of books that cost less than \$40 and that where bought by fewer than 3 CS students."

   (b) "For each student, find the sid and name of that student along with the number of books bought by that student, provided that the collective cost of these books is less that \$200."

   (c) "Find the sids and names of the students who spent the most on the books that they bought."

(d) "For each major, specify the combined cost of all the books bought by students with that major."

(e) "Find the pairs of different booknos (b_1, b_2) that were bought by the same number of CS students."

3 Queries with quantifiers using Venn diagrams with conditions

For the following problems, use the relations `student`, `major`, `book`, and `buys` that can be found in the `data.sql` file.

Using the method of Venn diagrams with conditions and without using the `COUNT` function, write SQL queries for the following queries with quantifiers.

In these problems, you must write appropriate views and parameterized views for the sets A and B that occur in the Venn diagram with conditions for these queries. (See the lecture on Queries with Quantifiers.)

9. Find the sid and name of each student who did not buy all the books that cost more than \$50.

10. Find the bookno and title of each book that was not only bought by students who major in 'CS' or in 'Math'.

11. Find the sid and name of each student who bought none of the least expensive books.

    create or replace view leastExpensiveBook as (select b.bookno from book b where b.price = (select min(b1.price) from book b1));

12. Find the pairs of booknos (b_1, b_2) of different books that where bought by the same set of CS students.

4 Queries with quantifiers using Venn diagrams with counting conditions

Using the method of Venn diagram with counting conditions, write SQL queries for the following queries with quantifiers.

In these problems, you should write appropriate views and parameterized views for the sets A and B that occur in the Venn diagrams for these queries. (See the lecture on Queries with Quantifiers Using the COUNT function.)

13. Find sid and name of each CS student who bought fewer than 4 books that cost less that \$50.

14. Find the bookno and title of each book that was bought by an odd number of CS students.

15. Find the sid and name of each student who bought all but 3 books.
16. Find the pairs of booknos (b_1, b_2) of different books such that all the students who bought book b_1 also bought book b_2.