Assignments 4

Due Friday, September 30 2016 by 11:59pm

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems.

1 Part 1: Relational Algebra

Remark: Consider a relation $R(A, B)$ and a relation $S(C)$ and consider the following RA (Relational Algebra) expression F:

$$\pi_A(R) - \pi_A(\sigma_{B=1}(R \bowtie_{B=C} S))$$

Then we can write this query in SQL in a way that closely mimics its RA formulation. This can be done using the WITH statement of SQL as follows.[1]

First, we break the RA expression F up into sub-expressions as follows. (Notice that each sub-expression corresponds to the application of a single RA operation.)

Expression Name	RA expression
E_1	$\pi_A(R)$
E_2	$R \bowtie_{B=C} S$
E_3	$\sigma_{B=1}(E_2)$
E_4	$\pi_A(E_3)$
F	$E_1 - E_4$

Then we write the following SQL query. Notice how the expressions $E1$, $E2$, $E3$, and $E4$ occur as separate queries in the WITH statement and that the final query gives the result for the expression F.[2]

If you use overloading of the relational name, the query becomes

```
WITH
E1 AS (SELECT DISTINCT A FROM R),
E2 AS (SELECT A, B, C FROM (R INNER JOIN S ON (B = C)) e2),
E3 AS (SELECT A, B, C FROM E2 WHERE B = 1),
E4 AS (SELECT DISTINCT A FROM E3)
(SELECT A FROM E1) EXCEPT (SELECT A FROM E4);
```

In your answer to a problem, you should write the resulting RA expression as such an SQL query. (Your SQL query should of course execute correctly in PostgreSQL). In a separate file you should also submit the text for the RA expressions in their standard notation, just as show for the expression F above.

[1] For more information about the WITH statement see `https://www.postgresql.org/docs/9.1/static/queries-with.html`.

[2] For better readability, I have used relational-name overloading. Sometimes, you may need to introduce new attribute names in SELECT clauses using the AS clause. Also, use DISTINCT were needed.

1. (10 points) Let $W(A, B)$ be a relation schema. The domain of A is INTEGER and the domain of B is VARCHAR(5).

   Write a RA expression which returns the A-values of tuples in W if A is a primary key of W. Otherwise, i.e., if A is not a primary key, then your query should return the A-values of tuples in W for which the primary key property is violated.

   W

A	B
1	John
2	Ellen
3	Ann

   Then your query should return the following answer since, in this case, A satisfies the primary property for W.

   ```
   a
   ---
   1
   2
   3
   (3 rows)
   ```

   However, if we have the following relation instance for W

   W

A	B
1	John
2	Ellen
2	Linda
3	Ann
4	Ann
4	Nick
4	Vince
4	Lisa

   then your query should return the following answer because the primary key property of A for W is violated for the A-values 2 and 4.

   ```
   a
   ---
   2
   ```

```
4
(2 rows)
```

2. In the following questions, use the files student.txt, majors.txt, book.txt, cites.txt, and buys.txt that are provided for this assignment. Use the same relations as in Assignment 2.

   Write the following queries as RA expressions. For each such RA expression, write a SQL query (using the WITH statement) that mimics this expression.

   (a) (10 points) Find the Sid and Sname of each student who bought a book that cites another book.

   (b) (10 points) Find the Sid and Sname of each student who has at least two majors.

   (c) (10 points) Find the Sid of each student who bought exactly one book.

   (d) (10 points) Find the Bookno of each book with the lowest price.

   (e) (10 points) Find the Bookno of each book that is not cited by a book that cost more than $50.

   (f) (10 points) Find the Sid of each student who only bought books that cost less than $30.

   (g) (10 points) Find the Bookno of each book that was not bought by all students who major in CS.

   (h) (10 points) Find the pairs (s1,s2) of Sid's of students such that all books bought by student s1 where also bought by student s2.

   (i) (10 points) Find the Bookno of each book that is cited by all but one book.

2 Part 2 : Queries with Aggregate Functions

1. Let $A(x)$ and $B(x)$ be two unary relation schemas that represent a set A and a set B. (The domain of x in INTEGER.) In this problem you should use the COUNT function in your solution.

   For example, suppose that you are asked to write a SQL query that determines whether it is true or not if $A \cap B = \emptyset$. The answer to this problem should be a SQL query of the following form.

```
SELECT (SELECT count(1)
        FROM ((SELECT a.x FROM A a)
               INTERSECT
              (SELECT b.x FROM b)) Q ) = 0 AS "PropertySatisfied";
```

   This query utilizes the fact that $A \cap B = \emptyset$ if and only if $|A \cap B| = 0$. If indeed $A \cap B = \emptyset$, then the output of your query will be

```
 PropertySatisfied
-------------------
 t
(1 row)
```

   Otherwise the output will be

```
 PropertySatisfied
-------------------
 f
(1 row)
```

   (a) (10 points) Write a SQL statement that determines whether it is true or not if $A \subseteq B$.

   (b) (10 points) Write a SQL statement that determines whether it is true or not if $|A| = |B|$.

   (c) (10 points) Write a SQL statement that determines whether it is true or not if $|A \cap B| \geq 2$.

   (d) (10 points) Write a SQL statement that determines whether it is true or not if $|A| \in B$. Notice that $|A|$ is a number and B is a set of numbers.

2. Write the following queries in SQL query using the COUNT aggregate function.

   For each problem, you will be asked to write a SQL query that uses the GROUP BY method, the FUNCTION method, the LATERAL method, or the COUNT expression in SELECT clause method. In your solutions you are NOT permitted to use the EXISTS, NOT EXISTS, IN, NOT IN, ALL, ANY, or SOME set predicates. You are permitted to use the UNION, EXCEPT, and INTERSECT operations where necessary.

   (a) For each student determine the number of books bought by that student whose price is in the range [$20,$40]. (Notice that it is possible that a student does not buy any such book. For such a student you should report the number 0 in your answer.)

      i. (10 points) Write this query using the GROUP BY method.

      ii. (10 points) Write this query using the FUNCTION method.

   (b) Find the bookno of each book that is not cited by any other book.

      i. (10 points) Write this query using the FUNCTION method.

      ii. (10 points) Write this query using the LATERAL method.

   (c) (10 points) Find the Sid and Sname of each students who has at least two majors and who only bought books that were cited by other books.

      Write this query using the COUNT expression in SELECT clause method.

   (d) Find the sid and major of each student who bought at least one book, but who did not buy any books that cost less than $30.

      i. (10 points) Write this query using the GROUP BY method.

      ii. (10 points) Write this query using the LATERAL method.

   (e) (10) Find the tuples (s1,s2) where s1 and s2 are different Sid's of students such that student s1 bought at least as many books as the number of books bought by student s2.

      Write this query using the COUNT expression in SELECT clause method.

      Write this query using the GROUP BY method.

   (f) (10 points) Find the Bookno's of books that where bought buy all but one student.

      Write this query using the LATERAL method.

2. Let A and B be sets whose union $A \cup B$ is not empty. The *Jaccard similarity measure* $J(A, B)$ of A and B is the quantity

$$\frac{|A \cap B|}{|A \cup B|}$$

The Jaccard similarity measure of A and B gives a measure of how similar they are. Indeed, if $J(A, B) = 1$, then $A = B$, and if $J(A, B) = 0$ then the sets are entirely non-similar, i.e. $A \cap B = \emptyset$.[3]

Consider now our `Cites` relation and consider a pair of books (b_1, b_2). Let $C(b_1)$ denote the set of books cited by book b_1 and let $C(b_2)$ denote the set of books cited by book b_2. Then $J(C(b_1), C(b_2))$ gives a measure of how similar books b_1 and b_2 are in terms of the books they cite. If $J(C(b_1), C(b_2)) = 1$ then books b_1 and b_2 cite the same books. If $J(C(b_1), C(b_2)) = 0$ then books b_1 and b_2 do not cite any common book.

Write a SQL function `Jaccard`(l, u), with $0 \leq l \leq u \leq 1$, that returns the relation of pairs of books (b_1, b_2) such that $l \leq J(C(b_1), C(b_2)) \leq u$. (Think of l as a lower bound and u as an upperbound for the Jaccard similarity measure.)

So `Jaccard`$(0, 1)$ should return the set of all pairs of books, `Jaccard`$(0, 0)$ should return the pairs of books that do not cite any books in common, and `Jaccard`$(1, 1)$ should return the pairs of books that cite the same books.

The function `Jaccard` should have the following structure:

```
create or replace function Jaccard(l float, u float)
returns table (book1 integer, book2 integer) AS
$$
  ... code of Jaccard function
$$ LANGUAGE SQL
```

[3] You might want to look up some information about the Jaccard similarity measure and it many applications.

3. Suppose that you have a relation `CourseTopicsDistribution`(sid, percentage, topic) which stores triples of the form (s, p, t) where s is a student Sid, and p is the percentage of courses taken by that student in topic t.

   For example, the data could be as follows:

   `CourseTopicDistribution`

Sid	Percentage	Topic
1	15	CS
1	30	Math
1	20	Biology
1	35	Physics
2	100	CS
2	0	Math
2	0	Biology
2	0	Physics
3	50	CS
3	50	Math
3	0	Biology
3	0	Physics
4	0	CS
4	0	Math
4	5	Biology
4	95	Physics
5	25	CS
5	25	Math
5	25	Biology
5	25	Physics

   In this data, we assume that there are only four topics: CS, Math, Biology, and Physics. In general, however, there could be any number of topics.

   The *Simpson diversity measure* determines for a student the degree of diversity in the distribution of the number of courses taken by that student in in the various topics. For example, student 5 has the highest diversity measure since each topic has the same percentage 25%. Student 1 also has a high diversity. Student student 2 has the lowest diversity because he/she has a 100% in a single topic. Student 4 also has low diversity skewed towards Physics and Student 3 has a diversity skewed towards CS and Math.

   In this data, given a student s, we have 4 tuples, namely (s, p_1, CS), $(s, p_2, Math)$, $(s, p_3, Biology)$, and $(s, p_4, Physics)$. The *Simpson diver-*

sity measure Simpson(s) for student s is given by the formula

$$\frac{4}{3}(1-(\frac{p_1}{100}^2+\frac{p_2}{100}^2+\frac{p_3}{100}^2+\frac{p_4}{100}^2))$$

In this measure, the value 4 corresponds to the number of topics, and the value 3 is the number of topics minus 1. In general, if there where n topics, the above sum would have n terms, and the multiplication factor would be $\frac{n}{n-1}$.

So the general formula for the Simpson diversity measure for a student is

$$\frac{n}{n-1}(1-\Sigma_{i=1}^{n}\frac{p_i}{100}^2)$$

where the p_i's are the n percentages for the n topics associated with the student.

This Simpson diversity measure is between 0 and 1. The closer to 0, the less diversity and the closer to 1, the more diversity.

So, for the student data given in the table above, we get the following table of the Simpson diversity measure for the various students:

Sid	SimpsonMeasure
1	$0.96\cdots$
2	0
3	$0.66\cdots$
4	$0.12\cdots$
5	1

(a) Write a SQLfunction for the Simpson diversity measure. The structure of this function should be

```
create or replace function Simpson(student integer)
returns float AS
$$
... code
$$ LANGUAGE SQL
```

Your solution should be general in these sense that if you have a table with n topics, then your function should work for such data as well. This means that one of the first thing to do in the code is to determine the number n of different topics in the data. Subsequently, the sum in the formula for the Simpson diversity measure should be over n terms. Hint: Use the COUNT and SUM aggregate functions of SQL to write the code for this function.

(b) Write a SQL query that gives the Simpson diversity measures for all students. (Just as shown in the previous table.)

(c) Write a SQL function table DiversityRange(l float, u float) that returns the set of students whose Simpson diversity measure is within the range $[l, u]$ where $0 \leq l \leq u \leq 1$. So $Diversity(1, 1)$ gives the most diverse students and $Diversity(0, 0)$ gives the least diverse students. $Diversity(0.9, 1)$ gives students with high diversity etc.

The structure of your function should be

```
create or replace function SimpsonRange(l float, u float)
returns table(int) AS
$$
... code
$$ LANGUAGE SQL
```

Remark: The Simpson diversity measure has many applications. You may want to look up some information about it. A closely related measure is the *Shannon diversity measure* which measure the average entropy in a distribution. You may want to read about the Shannon diversity measure since it also used in many applications. As such, the Simpson and Shannon diversity measures are used in Data Analytics and Data Science to analyze properties of data distributions.