Assignment 4

Due Wednesday, October 10 2018 by 11:59pm

In this assignment, you will be required to use PostgreSQL. Your solutions should include the PostgreSQL statements for solving the problems. You will use the same data as that used in Assignment 3.

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 variety of ways that closely mimic its RA formulation. One way to write this RA expression in SQL is as follows:

```
SELECT  A
FROM    R
EXCEPT
SELECT A
FROM (SELECT R.A, R.B, S.C
      FROM   R INNER JOIN S ON B=C
      WHERE  B = 1) q;
```

An alternative way is to use the WITH statement of SQL.[1] First, we separate the RA expression F into sub-expressions as follows. (In this case, notice that each sub-expression corresponds to the application of a single RA operation. More generally, one can of course use sub-expressions that can contain multiple RA operations.)

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	$E_1 - E_4$

[1] This is especially so if the RA expression is long and complicated.

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 may write the resulting RA expression with or without the WITH statement. (Your SQL query should of course closely resemble the RA expression it is aimed to express.) In a separate file you should also submit the text for the RA expressions in their standard notation, just as illustrated for the expression F above.

[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. 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 3.

   Write the following queries as RA expressions in the standard RA notation. Submit these queries as a separate document.

   Then, for each such RA expression, write a SQL query (possibly using the WITH statement) that mimics this expression.

   In this document, we will use the following notations:

Student	S, S_1, etc
Book	B, B_1, etc
Cites	C, C_1, etc
Major	M, M_1, etc
Buys	T, T_1, etc

   On occassion we will also use the notation bno as an abbreviation for $bookno$.

   (a) Find the sid and sname of each student who bought a book that cites another book.

   $$\pi_{S.sid,S.sname}(S \bowtie T \bowtie C)$$

   This RA expression can be formulated in SQL as follows:

```
select distinct  sid, sname
from   student
        natural join buys
        natural join cites ;
```

   (b) Find the sid and sname of each student who has at least two majors.

   $$\pi_{S.sid.S.sname}(S \underset{S.sid = M_1.sid}{\bowtie} M_1 \underset{S.sid = M_2.sid \wedge M_1.major \neq M_2.major}{\bowtie} M_2)$$

   This RA expression can be formulated in SQL as follows:

```
select distinct  s.sid, s.sname
from   student s
       inner join major m1 on s.sid = m1.sid
       inner join major m2 on (s.sid = m2.sid and m1.major <> m2.major);
```

(c) Find the sid of each student who bought exactly one book.

$$\pi_{sid}(T) - \pi_{T_1.sid}(T_1 \underset{T_1.sid\, =\, T_2.sid\ \wedge\ T_1.bookno\, \neq\, T_2.bookno}{\bowtie} (T_1 \times T_2))$$

This RA expression can be formulated in SQL as follows:

```
select t.sid
from   buys t
except
select distinct  t1.sid
from   buys t1 inner join buys t2 on (t1.sid = t2.sid and t1.bookno <> t2.bookno);
```

(d) Find the bookno and title of each book that was only bought by the student with sid = 1001.

$$\pi_{bookno,title}(B) - \pi_{bookno,title}(B \bowtie \sigma_{sid \neq 1001}(T))$$

This RA expression can be formulated in SQL as follows:

```
select b.bookno, b.title
from   book b
except
select distinct b.bookno, b.title
from   book b natural join (select bookno from buys where sid <> 1001) t;
```

(e) Find the sid and sname of each student who bought at least two books that cost less than \$50.

Let E be the expression such that

$$E = \pi_{T.sid,T.bookno}(T \underset{T.bookno = B.bookno}{\bowtie} \sigma_{price<50}(B))$$

$$\pi_{S.sid.S.sname}(S \underset{S.sid = T_1.sid}{\bowtie} Q)$$

where

$$Q = E_1 \underset{E_1.sid = E_2.sid \,\wedge\, E_1.bookno \neq E_2.bookno}{\bowtie} E_2$$

This RA expression can be formulated in SQL as follows:

```
with E as
  (select  t.*
   from   buys t inner join (select b.* from book b where price < 50) b on t.bookno = b.bookno and b.price < 50)
select distinct s.*
from   student s
       inner join E e1 on s.sid = e1.sid
       inner join E e2 on (e1.sid = e2.sid and e1.bookno <> e2.bookno);
```

(f) Find the bookno of each book that was not bought by all students who major in CS.

$$\pi_{bookno}(\pi_{sid}(\sigma_{major=CS}(M)) \times \pi_{bookno}(B) - T)$$

This RA expression can be formulated in SQL as follows:

```
select distinct q.bookno
from
   (select m.sid, b.bookno
    from  (select m.sid from major m where major ='CS’) m
           cross join (select b.bookno from book b) b
    except
    select t.sid, t.bookno
    from   buys t) q;
```

(g) Find the bookno of each book that is not cited by a book that cost more than \$50.

$$\pi_{bookno}(B) - \pi_{citedbookno}(C \bowtie \sigma_{price>50}(B))$$

This RA expression can be formulated in SQL as follows:

```
select bookno
from   book
except
select citedbookno
from   cites natural inner join (select bookno from book where price > 50) b;
```

(h) (10 point) Find each pair (s, b) such that s is the sid of a student who bought a book that does not cite the book with bookno b.

Then the answer is

$$\pi_{T.sid,B.bookno}(T \times \pi_{bookno}(B) - \pi_{sid}(S) \times C)$$

This RA expression can be formulated in SQL as follows:

```
select distinct q.ssid, q.bbookno
from   (select t.sid as ssid, t.bookno, b.bookno as bbookno
        from   buys t
               cross join (select bookno from book) b
        except
        select s.sid, bookno, citedbookno
        from   (select sid from student s) s
                cross join cites) q;
```

(i) Find the pair of different booknos (b_1, b_2) that where bought by the same CS students.
We use the following expressions

$$\begin{array}{lcl}
B & = & \pi_{bookno}(Book) \\
\mathbf{T} & = & \pi_{T.sid,T.bookno}(T \ltimes \sigma_{major=CS}(M)) \\
E_1 & = & \pi_{T_1.bno,T_1.sid,B_2.bno}(T_1 \times B_2) \\
E_2 & = & \pi_{B_1.bno,T_2.sid,T_2.bno}(T_2 \times B_1) \\
E_3 & = & \pi_{B_1.bno,B_2.bno}((B_1 \underset{B_1.bno \neq B_2.bno}{\bowtie} B_2) - \pi_{T_1.bno,B_2.bno}(E_1 - E_2)) \\
E_4 & = & \pi_{B_1.bno,B_2.bno}((B_1 \underset{B_1.bno \neq B_2.bno}{\bowtie} B_2) - \pi_{B_1.bno,T_2.bno}(E_2 - E_1))
\end{array}$$

Then the answer is

$$E_3 \cap E_4$$

This RA expression can be formulated in SQL as follows:

```
with  B as (select b.bookno from book b),

      T  as (select  distinct t.sid, t.bookno
             from buys t natural join (select distinct m.sid from major m where m.major = 'CS') m),

      E1 as (select t1.bookno as t1bookno, t1.sid, b2.bookno as b2bookno
             from   T t1 cross join B b2),

      E2 as (select b1.bookno as b1bookno, t2.sid, t2.bookno as t2bookno
             from   T t2 cross join B b1),

      E3 as (select  b1.bookno, b2.bookno
             from    B b1 inner join B b2  on (b1.bookno <> b2.bookno)
             except
             select q.t1bookno, q.b2bookno
             from (select e1.t1bookno, e1.sid, e1.b2bookno from E1 e1
                   except
                   select e2.b1bookno, e2.sid, e2.t2bookno from E2 e2) q),

       E4 as (select  b1.bookno, b2.bookno
              from    B b1 inner join B b2  on (b1.bookno <> b2.bookno)
              except
              select q.b1bookno, q.t2bookno
              from (select e2.b1bookno, e2.sid, e2.t2bookno from E2 e2
                    except
                    select e1.t1bookno, e1.sid, e1.b2bookno from E1 e1) q)


select e3.* from E3 e3
intersect
select e4.* from E4 e4;
```

(j) Find the pairs of different sid (s1,s2) of students such that all books bought by student s1 were also bought by student s2.

Let

$$S = \pi_{S.sid}(Student)$$

$$\begin{aligned} E_1 &= \pi_{T_1.sid, T_1.bno, S_2.sid}(T_1 \times S_2) \\ E_2 &= \pi_{S_1.sid, T_2.bno, T_2.sid}(T_2 \times S_1) \end{aligned}$$

Then the answer is

$$(S_1 \underset{S_1.sid \neq S_2.sid}{\bowtie} S_2) - \pi_{T_1.sid, S_2.sid}(E_1 - E_2)$$

This RA expression can be formulated in SQL as follows:

```
with S as (select s.sid from student s),
     E1 as (select t1.sid as t1sid, t1.bookno, s2.sid as s2sid
            from   buys t1 cross join S s2),
     E2 as (select s1.sid as s1sid, t2.bookno, t2.sid as s2sid
            from   buys t2 cross join S s1)
select  s1.sid, s2.sid
from    S s1 inner join S s2  on (s1.sid <> s2.sid)
except
select q.t1sid, q.s2sid
from (select e1.t1sid, e1.bookno, e1.s2sid from E1 e1
      except
      select e2.s1sid, e2.bookno, e2.s2sid from E2 e2) q;
```

2. Translate each of the following SQL queries in RA expressions using the translation algorithm given in class. Make sure that these RA expressions are formulated in SQL with RA operations so they can be run in PostgreSQL. You are required to show the intermediate steps that you took during the translation.

   In addition, submit separately the RA expressions in the standard RA notation.

   (a) Find the sid and major of each student who bought a book that cost less than $20.

```
select m.sid, m.major
from   major m
where  m.sid in (select t.sid
                 from   buys t, book b
                 where  t.bookno = b.bookno and b.price < 20) order by sid, major;
```

is translated in SQL to

```
select distinct m.sid, m.major
from   major m, buys t, book b
where  m.sid = t.sid and t.bookno = b.bookno and b.price < 20 order by sid, major;
```

is translated SQL to

```
select distinct sid, major
from   (major natural join buys) natural join
                (select bookno from book b where price < 20) q order by sid, major;
```

is translated to RA as

$$\pi_{M.sid,M.major}(M \bowtie T \bowtie \sigma_{price<20}(B))$$

(b) Find each (s, b) pair where s is the sid of a student and where b is the bookno of a book whose price is the cheapest among the books bought by that student.

```
select distinct t.sid, b.bookno
from   buys t, book b
where  t.bookno = b.bookno and
       b.price <= ALL (select b1.price
                       from   buys t1, book b1
                       where t1.bookno = b1.bookno and t1.sid = t.sid) order by sid, bookno;
```

is translated in SQL to

```
select distinct t.sid, b.bookno
from   buys t, book b
where  t.bookno = b.bookno and
       not exists (select b1.price
                       from   buys t1, book b1
                       where b.price > b1.price and t1.bookno = b1.bookno and t1.sid = t.sid) order by sid, bookno;
```

is translated in SQL to

```
select distinct q.sid, q.bookno
from
    (select t.*, b.bookno as bbookno, b.title, b.price
      from buys t, book b
      where t.bookno = b.bookno
      except
      select t.*, b.*
      from  buys t, book b, buys t1, book b1
      where b.price > b1.price and t1.bookno = b1.bookno and t1.sid = t.sid) q order by sid, bookno;
```

is translated in SQL to

```
select distinct q.sid, q.bookno
from
    (select t.*, b.bookno as bbookno, b.title, b.price
      from buys t inner join book b on t.bookno = b.bookno
      except
      select t.*, b.*
      from
            buys t
            inner join buys t1 on t.sid = t1.sid
            inner join book b1 on t1.bookno = b1.bookno
            inner join book b on b.price > b1.price) q order by sid, bookno;
```

is translated in RA to

$$\pi_{T.sid,B.bookno}(\pi_{T.*,B.*}(T \underset{T.bookno = B.bookno}{\bowtie} B) - \pi_{T.*,B*}(E))$$

where

$$E = T \underset{T.sid = T_1.sid}{\bowtie} T_1 \underset{T_1.bookno = B_1.bookno}{\bowtie} B_1 \underset{B.price > B_1.price}{\bowtie} B$$

(c) Find the sid and name of each student who majors in ‘CS’ and who bought a book that is cited by a lower priced book.

```
select s.sid, s.sname
from   student s
where  s.sid in (select m.sid from major m where m.major = 'CS') AND
       exists (select 1
               from   buys t, cites c, book b1, book b2
               where  s.sid = t.sid and t.bookno = c.citedbookno and
                      c.citedbookno = b1.bookno and c.bookno = b2.bookno and
                      b1.price > b2.price) order by sid, sname;
```

is translated in SQL to

```
select    q.sid, q.sname
from
     (select s.*
      from   student s, major m
      where s.sid = m.sid and m.major ='CS'
      intersect
      select  s.*
      from   student s, buys t, cites c, book b1, book b2
      where  s.sid = t.sid and t.bookno = c.citedbookno and
            c.citedbookno = b1.bookno and c.bookno = b2.bookno and b1.price > b2.price) q order by sid, sname;
```

is translated in SQL to

```
select    q.sid, q.sname
from
     (select s.*
      from   student s, major m
      where s.sid = m.sid and m.major ='CS'
      intersect
      select  s.*
      from   student s, buys t, cites c, book b1, book b2
      where  s.sid = t.sid and t.bookno = c.citedbookno and
            c.citedbookno = b1.bookno and c.bookno = b2.bookno and b1.price > b2.price) q order by sid, sname;
```

is translated in SQL to

```
select    q.sid, q.sname
from
     (select s.*
      from   student s natural join (select m.* from major m where major = 'CS') m
      intersect
      select  s.*
      from   student s
                  inner join buys t on s.sid = t.sid
                  inner join cites c on t.bookno = c.citedbookno
                  inner join book b1 on c.citedbookno = b1.bookno
                  inner join book b2 on (c.bookno = b2.bookno and b1.price > b2.price)) q order by sid, sname;
```

is translated in RA to

$$\pi_{S.sid,S.sname}((S \bowtie \sigma_{major=CS}(M)) \cap E)$$

where

$$E = S \underset{S.sid = T.sid}{\bowtie} T \underset{T.bno = C.citedbno}{\bowtie} C \underset{C.citedbno = B_1.bno}{\bowtie} B_1 \underset{C.bno = B_2.bno \wedge B_1.price > B_2.price}{\bowtie} B_2$$

Here *bno* is an abbreviated for *bookno*.

(d) Find the bookno and title of each book that is not bought by all students who major in ‘CS’.

```
select b.bookno, b.title
from   book b
where  exists (select m.sid
               from   major m
               where  m.major = 'CS' and
                      m.sid not in(select t.sid
                                   from   buys t
                                   where  t.bookno = b.bookno)) order by bookno;
```

is translated in SQL to

```
select distinct b.bookno, b.title
from   book b, major m
where m.major = 'CS' and
                      m.sid not in(select t.sid
                                   from   buys t
                                   where  t.bookno = b.bookno) order by bookno;
```

is translated in SQL to

```
select distinct q.bookno, q.title
from (select  b.*, m.*
      from   book b, major m
      where m.major = 'CS'
      except
      select b.*,m.*
      from book b, major m, buys t
      where m.sid = t.sid and  t.bookno = b.bookno) q order by bookno;
```

```
select distinct q.bookno, q.title
from (select  b.*, m.*
      from   book b CROSS JOIN (select m.* from major m where major = 'CS') m
      except
      select b.*,m.*
      from major m
                 inner join  buys t on m.sid = t.sid
                 inner join  book b on t.bookno = b.bookno) q order by bookno;
```

is translated in RA to

$$\pi_{B.bookno,M.major}(\pi_{B.*,M.*}(B\times\sigma_{major=CS}(M))-\pi_{B.*,M.*}(M \underset{M.sid = T.sid}{\bowtie} T \underset{T.bookno = B.bookno}{\bowtie} B))$$

(e) Find the bookno and title of each book that is bought by all students who major in both ‘CS’ and in ‘Math’.

```
select b.bookno, b.title
from   book b
where  not exists (select s.sid
                   from   student s
                   where  s.sid in (select m.sid from major m
                                    where m.major = ’CS’) and
                          s.sid in (select m.sid from major m
                                    where m.major = ’Math’) and
                          s.sid not in (select t.sid
                                        from   buys t
                                        where  t.bookno = b.bookno));
```

This query is equivalent with

```
select q.bookno, q.title
from(
        select b.*
        from  book b
        except
        select p.bookno, p.title, p.price from
           (select m.sid,s.*,b.* from major m, student s, book b
             where s.sid = m.sid and m.major = ’CS’
             intersect
             select m.sid,s.*,b.* from major m, student s, book b
             where s.sid = m.sid and m.major = ’Math’
             except
              select t.sid, s.*,b.*
              from   buys t, student s, book b
              where  t.sid = s.sid and t.bookno = b.bookno) p) q;
```

We use the following expression

$$\begin{aligned}
E_1 &= \pi_{M.sid,S.*,B.*}(M \underset{M.sid = S.sid}{\bowtie} S \times \sigma_{major=CS}(M)) \\
E_2 &= \pi_{M.sid,S.*,B.*}(M \underset{M.sid = S.sid}{\bowtie} S \times \sigma_{major=Math}(M)) \\
E_3 &= \pi_{T.sid,S.*,B.*}(T \underset{T.sid = S.sid}{\bowtie} S \underset{T.bookno = B.bookno}{\bowtie} B)
\end{aligned}$$

Then the answer is

$$\pi_{B.bookno,B.title}(B - \pi_{B.bookno,B.title,B.Price}((E_1 \cap E_2) - E3))$$