Assignment 5: Relational Algebra

For this assignment, you will need to submit 2 files. One such file is a .sql file that contains the SQL code relating to problems requesting the development of such code. A second file with .pdf extension that contains your solution for problems where RA expressions are requested as well ass essay question for some of the problems.

1 Theoretical Problems about RA

1. In the lecture on set joins and semijoins, we specified the RA expressions for the $[\textit{some}, \textit{no}, \textit{not only}, \textit{only}, \textit{not all}, \textit{all}]$ set semijoins:

$$\begin{aligned}
\text{some} &= \pi_{sid}(E \ltimes CS) \\
\text{not only} &= \pi_{sid}(E \,\overline{\ltimes}\, CS)) \\
\text{not all} &= \pi_{sid}((\pi_{sid}(S) \times CS) - (E \ltimes CS)) \\
\text{no} &= \pi_{sid}(S) - \pi_{sid}(E \ltimes CS) \\
\text{only} &= \pi_{sid}(S) - \pi_{sid}(E \,\overline{\ltimes}\, CS) \\
\text{all} &= \pi_{sid}(S) - \pi_{sid}((\pi_{sid}(S) \times CS) - (E \ltimes CS))
\end{aligned}$$

   Using the techniques developed for the set joins in that lecture, show how to derive these RA expressions.

2. Develop an RA expression for the [*all but one*] set semijoin. I.e., for the query "Find the sid of each student who takes all but one CS course."

   What is the time and space complexity for this [*all but one*] set semijoin?

3. Formulate the [*all but one*] set semijoin using the count method. I.e., write an SQL query that uses the `COUNT` aggregate function for this query.

   What is the time and space complexity for this query?

   What does this tell you about using the count method for expressing set semijoins?

4. Consider two RA expressions E_1 and E_2 over the same schema (A,B). Furthermore, consider an RA expression F.

   Consider the following `if-then-else` query:

$$\begin{array}{ll}
\texttt{if } F \neq \emptyset & \texttt{then return } E_1 \\
 & \texttt{else return } E_2
\end{array}$$

   So this query evaluates to the expression E_1 if $F \neq \emptyset$ and to the expression E_2 if $F = \emptyset$.

   (a) Write a RA expression in function of E_1, E_2, and F that expresses this `if-then-else` statement.

(b) What is the time and space complexity of this `if-then-else` statement?

5. Generalize this result for the `case` statement defined as follows: Let E_1 though E_k be $k \geq 2$ expressions over the same schema (A, B) and let F_1 through F_{k-1} be ≥ 2 expressions all over the same schema.

   The `case` statements is the following:

```
case  when $F_1 \neq \emptyset$       then return $E_1$
      when $F_2 \neq \emptyset$       then return $E_2$
      ...
      when $F_{k-1} \neq \emptyset$   then return $E_{k-1}$
      else                            return $E_k$.
```

   It semantics is defined inductively as follows. The case statement correspond to the following `if-then-else` statement:

```
if   $F_1 \neq \emptyset$   then return $E_1$
                            else return case$_{k-1}$
```

   Here `case`$_{k-1}$ is the following statement:

```
case  when $F_2 \neq \emptyset$       then return $E_2$
      when $F_3 \neq \emptyset$       then return $E_k$
      ...
      when $F_{k-1} \neq \emptyset$   return $E_{k-1}$
      else                            return $E_k$.
```

   (a) Write a RA expression in function of E_1 through E_k, and F_1 through F_k expresses this `case` statement.

   (b) What is the time and space complexity if this `case` statement?

2 Formulating Queries in RA

Before you solve the problems in this section, we briefly review how you can express RA expressions in SQL in a way that closely mimics their RA specifications. (For more detail, consult the lectures relating to RA and joins.)

Consider a relation $R(A, B)$ and a relation $S(C)$ and consider the following RA 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 mimics its RA formulation. One way to write this RA expression in SQL is as follows:

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

An alternative way to write this query is to use the `WITH` statement of SQL.[1] To do this, 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	$E_1 - E_4$

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]

```
WITH
E1 AS (SELECT DISTINCT A FROM R),
E2 AS (SELECT DISTINCT A, B, C FROM (R 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);
```

[1] This is especially convenient when the RA expression is long and complicated.

[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.

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 with .pdf extension you should also submit the text for the RA expressions in their standard notation, just as illustrated for the expression F above.

6. In the following questions, we will use the data that you can find in the data.sql file provided for these problems.

   Write the following queries as RA expressions in the standard RA notation. Submit these queries as a separate document. In these expressions, you can use the following notations for the relations:

Student	S, S_1, S_2, etc
Book	B, B_1, B_2 etc
Cites	C, C_1, C_2 etc
Major	M, M_1, M_2, etc
Buys	T, T_1, T_2, etc

   Then, for each such RA expression, write a SQL query (possibly using the WITH statement) that mimics this expression as discussed above. Submit these queries in a .sql file as usual.

   (a) Find the bookno and title of each book that was bought by a student who majors in both CS and in Math.

   (b) Find the sid-bookno pairs (s,b) such that student s bought book b and such that book b is cited by at least two books that cost less than $50.

   (c) Find the triples (s, b_1, b_2) where s is the sid of a student and b_1 and b_2 are the booknos of books such that

   - student s bought both books b_1 and b_2 and
   - book b_1 cites book b_2.

   (d) Find the sid and sname of each student who bought a book that is cited by no other book.

   (e) Find the bookno and title of each book bought by some student who majors in ‘CS’ and which has, among these books, the highest price. (In other words, considering all the books bought by CS students, you need to find those books are the most expensive.)

   (f) Find the bookno and title of each cited book that was only cited by books that cost more that $50.

   (g) Find the bookno and title of each book that was not cited by all books that cost more that $50.

   (h) 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.

(i) Find the pairs of different sid (s_1, s_2) of students such that no book bought by student s_1 is a book bought by student s2.

(j) Find the pair of different booknos (b_1, b_2) that where not bought by the same CS students. (In other words, if $S(b_1)$ is the set of CS students who bought book b_1 and $S(b_2)$ is the set of CS students who bought book b_1, then $S(b_1) \neq S(b_2)$.)