Solutions Assignment 6: Translating and optimizing SQL queries

For this assignment we will be using the student, book, buys, and cites relational database schema.

1. Translate the following SQL queries into equivalent RA expressions and show the various steps that were involved in the translations.

   (a)
   ```
   SELECT  s.sid, b1.bookno
   FROM    student s, buys b1, buys b2
   WHERE   s.sid = b1.sid AND s.sid= b2.sid AND
           b1.bookno <> b2.bookno AND
           s.sname = 'Eric' and b1.bookno <> 2010;
   ```

   $$\pi_{S.sid,B_1.bookno}(\sigma_C(S \times B_1 \times B_2))$$

   where C is the condition

   $$\begin{array}{l} S.sid = B1.sid \wedge S.sid = B2.sid \wedge \\ \quad B1.bookno <> B2.bookno \wedge \\ \quad\quad S.sname = \text{`Eric'} \wedge B1.bookno <> 2010 \end{array}$$

   We now begin with the optimization.

   When looking at the condition C, notice the conditions $S.sname =$ 'Eric' and $B1.bookno <> 2010$. These conditions can be pushed down over $\times$ to the $Student$ and $B1$ relations.

   Furthermore, note that after applying the condition $S.sname =$ 'Eric', we no longer need the sname attribute. We will use SE to denote the expression

   $$\pi_{S.sid}(\sigma_{S.sname=\text{`Eric'}}(S))$$

   The condition $B1.bookno <> B2.bookno$ will induce a join between $B1$ and $B2$. This leads to the expression

   $$\sigma_{B1.bookno<>2010}(B1) \bowtie_{B1.bookno<>B2.bookno} B2$$

Furthermore, we notice that for the rest in the query, we only need to retain the $B1.sid$, $B1.bookno$, and $B2.sid$ attributes, so we get the expression

$$\pi_{B1.sid,B1.bookno,B2.sid}(\sigma_{B1.bookno<>2010}(B1) \bowtie_{B1.bookno<>B2.bookno} B2)$$

Let us call this the expression E. The condition $S.Sid = B1.sid$ and $S.sid = B2.sid$ requires use to join SE and E on these conditions. But after doing this, we only need to retain the $SE.sid$, and $E.B1.bookno$ attributes. So we get the optimized expression

$$\pi_{SE.sid,E.B1.bookno}(SE \bowtie_{SE.sid=B1.sid \wedge SE.sid=B2.sid} E)$$

An alternative, and probably better solution, could be if consider the followin expression which we denote by F

$$\pi_{B1.sid,B1.bookno}(\sigma_{B1.bookno<>2010}(B1) \bowtie_{B1.sid=B2.sid \wedge B1.bookno<>B2.bookno} B2)$$

We then derive at the following optimized expression:

$$\pi_{SE.sid,F.bookno}(SE \bowtie F)$$

What we did to get to this solution is to observe that the condition $S.sid = B1.sid$ and $S.sid = B2.sid$ are equivalent with the conditions $S.sid = B1.sid$ and $B1.sid = B2.sid$.

(b)
```
SELECT  DISTINCT b.bookno, b.title
FROM    book b, student s
WHERE   b.price = SOME(select b1.price
                       from   buys t, book b1
                       where  b1.price > 50 and
                              s.sid = t.sid and
                              t.bookno = b1.bookno);
```

Let E be the expression

$$\pi_{B2.bookno,B2.title,B2.price,S1.sid,S1.sname}(\sigma_C(Buys \times B1 \times B2 \times S1))$$

where C is the condition

$$\begin{array}{l} B1.price > 50 \land S1.sid = Buys.sid \land \\ \quad Buys.bookno = B1.bookno \land B2.price = B1.price \end{array}$$

Then the translation becomes is

$$\pi_{B.bookno,B.title}((B \times S) \ltimes E)$$

We can now turn to optimizing this expression.

In this case, we observe that the expression E will always evaluate to a subset of $(B \times S)$ so the above expression simplifies to

$$\pi_{B.bookno,B.title}(E)$$

Which, looking at E, further simplifies to

$$\pi_{B2.bookno,B2.titleS1.sid}(\sigma_C(Buys \times B1 \times B2 \times S1))$$

By pushing σ down, introducing joins, and pushing π down, we get the optimized expression $E3$ derived as follows:

$$\begin{array}{rcl} E1 & = & \pi_{B1.bookno,B1.price}(\sigma_{B1.price>50}(B1)) \\ E2 & = & \pi_{B1.price}(\pi_{Buys.bookno}(\pi_{S1.sid}(S1) \bowtie Buys) \bowtie E1) \\ E3 & = & \pi_{B2.bookno,B2.title}(E2 \bowtie B2) \end{array}$$

(c)
```
SELECT b.bookno
FROM   book b
WHERE  b.bookno IN (SELECT b1.bookno FROM book b1 WHERE b1.price > 50)
                    UNION
                   (SELECT c.bookno FROM cites c);
```

For the translation, we need to push the variable `b` into the sub-queries. We can then consider expressions $E1$ and $E2$

$$\begin{aligned} E1 &= \pi_{B2.bookno,B2.title,B2.price}(\sigma_{B2.bookno=B1.bookno \wedge B1.price>50}(B1 \times B2)) \\ E2 &= \pi_{B3.bookno,B3.title,B3.price}(\sigma_{B3.bookno=C.bookno}(C \times B3)) \end{aligned}$$

Then the translation is

$$\pi_{B.bookno}(B \ltimes (E1 \cup E2))$$

We now proceed to the optimization. First observe that $(E1 \cup E2) \subseteq B)$, therefore the above expression simplifies to

$$\pi_{B.bookno}(E1 \cup E2)$$

Then since π distributes over $\cup$, we get the expression

$$\pi_{B.bookno}(E1) \cup \pi_{Bookno}(E2)$$

We then proceed to optimize the left and right expressions of this union separately. The result is the following optimized expression:

$$\pi_{B.bookno}(\sigma_{B.price>50}(B)) \cup \pi_{Bookno}(C)$$

(d)
```
SELECT b.bookno FROM book b
WHERE b.price >= 80 and
      NOT EXISTS(SELECT b1.bookno
      FROM book b1
      WHERE b1.Price > b.Price);
```

For the translation strategy to RA see the .sql file. The results is as follows:

Let E be the expression

$$\pi_{B.bookno,B.title,B.price}(\sigma_{B.price \geq 80}(B))$$

Notice that E is equal to

$$\sigma_{B.price \geq 80}(B)$$

Let F be the expression

$$\pi_{B2.bookno,B2.title,B2.price}(\sigma_{B1.price > B2.price}(B1 \times B2))$$

Then the translation is the expression

$$\pi_{E.bookno}(E \,\overline{\ltimes}\, F)$$

This is equal to the expression

$$\pi_{E.bookno}(E - (E \ltimes F))$$

As far as the optimization, the only part we can work on is $E \ltimes F$, i.e., the expression

$$\sigma_{B.price \geq 80}(B) \ltimes \pi_{B2.bookno,B2.title,B2.price}(\sigma_{B1.price > B2.price}(B1 \times B2))$$

But this expression can be rewritten as

$$\pi_{B.bookno,B.title,B.price}(\pi_{B1.price}(B1) \bowtie_{B1.price > B.price} \sigma_{B.price \geq 80}(B))$$

Thus the fully optimized expression is

$$\pi_{B.bookno}(\sigma_{B.price \geq 80}(B) - \\ \pi_{B.bookno,B.title,B.price}(\pi_{B1.price}(B1) \bowtie_{B1.price > B.price} \sigma_{B.price \geq 80}(B)))$$

(e)
```
SELECT s.sid
FROM   Student s
WHERE  EXISTS(SELECT 1
              FROM Book b
              WHERE b.price > 50 AND
                    b.bookno IN (SELECT  t.bookno
                                 FROM    Buys t
                                 WHERE   s.sid = t.sid AND
                                         s.sname = 'Eric'))
```

Let $E1$ be the expression

$$\pi_{B1.bookno,B1.title,B1.price,S2.sid,S2.sname}(\\ \sigma_{s2.sid=t.sid \wedge s2.sname=\text{`Eric'} \wedge b1.bookno=t.bookno}(Buys \times B1 \times S2))$$

Let $E2$ be the expression

$$\pi_{S1.sid,S1.sname}(\sigma_{B.price>50}(B \times S1))$$

Let $E3$ be the expression

$$\pi_{E2.sid,E2.sname}(E2 \ltimes E1)$$

Then the translation becomes

$$\pi_{S.sid}(S \ltimes E3)$$

We now proceed to the optimization. Since $E3 \subseteq S$, we may proceed with optimizing the expression

$$\pi_{S.sid}(E3)$$

Consider $E1$:

$$\pi_{B1.bookno,B1.title,B1.price,S2.sid,S2.sname}(\\ \sigma_{s2.sid=t.sid \wedge s2.sname=\text{`Eric'} \wedge b1.bookno=t.bookno}(Buys \times B1 \times S2))$$

By pushing selection and projection down and introducing join where appropriate, we can rewrite $E1$ into the expression $E1'$

$$\pi_{B1.bookno,B1.title,B1.price,S2.sid,S2.sname}(\\ B1 \bowtie \pi_{Buys.bookno,S2.sid,S2.sname}(Buys \bowtie \sigma_{S2.sname=\text{`Eric'}}(S2)))$$

Consider $E2$:

$$\pi_{S1.sid,S1.sname}(\sigma_{B.price>50}(B \times S1))$$

This can be rewritten to the expression $E2'$

$$\pi_{S1.sid,S1.sname}(\sigma_{B.price>50}(B) \times S1)$$

So now $E3$ has become the expression

$$\pi_{E2.sid,E2.sname}(E2' \ltimes E1')$$

Notice the condition $\sigma_{B.price>50}$. This condition can be pushed inside $E1'$ so that we get the expression $E1''$

$$\pi_{B1.bookno,B1.title,B1.price,S2.sid,S2.sname}(\\ \sigma_{B.price>50}(B1) \bowtie \pi_{Buys.bookno,S2.sid,S2.sname}(Buys \bowtie \sigma_{S2.sname=\text{`Eric'}}(S2)))$$

So the final optimized expression, after pushing projections down and renaming $S2$ to S and $B1$ to B, becomes the following:

$$\pi_{S.sid}(\pi_{B.bookno}(\sigma_{B.price>50}(B)) \bowtie \pi_{Buys.bookno,S.sid}(Buys \bowtie \pi_{S.Sid}(\sigma_{S.sname=\text{`Eric'}}(S)))$$

(f)
```
SELECT s1.sid, s2.sid
FROM student s1, student s2
WHERE s1.sid <> s2.sid AND
      NOT EXISTS(SELECT 1
                 FROM   Buys t1
                 WHERE  t1.sid = s1.sid AND
                        t1.bookno NOT IN (SELECT t2.bookno
                                          FROM   Buys t2
                                          WHERE  t2.sid = s2.sid));
```

The first thing to do with this query it appropriately simulation the NOT IN using NOT EXISTS. This will yield the SQL query

```
SELECT s1.sid, s2.sid
FROM student s1, student s2
WHERE s1.sid <> s2.sid AND
      NOT EXISTS(SELECT 1
                 FROM   Buys t1
                 WHERE  t1.sid = s1.sid AND
                        NOT EXISTS (SELECT 1
                                    FROM   Buys t2
                                    WHERE  t2.sid = s2.sid AND
                                           t1.bookno = t2.bookno));
```

Notice that we need to recursively push down the variables $s1$, $s2$ and $t1$ into the subqueries.

We will now proceed to the translation. We consider the following expressions In my expression I have overload the names for the relation $Buys1$, $S1$ and $S2$. I did this to keep the reasoning and expressions simpler. I also use the notation $R.*$ to list all the attributes of a relation R.

$$
\begin{aligned}
E1 &= \pi_{Buys1.*,S1.*,S2.*}(\sigma_{Buys.sid=S2.sid \wedge Buys1.bookno=Buys.bookno}(\\
&\qquad Buys \times Buys1 \times S1 \times S2)) \\
E2 &= \pi_{Buys1.*,S1.*,S2.*}(\sigma_{Buys1.sid=S1.sid}(Buys1 \times S1 \times S2)) \\
E3 &= \sigma_{E2.S1.sid,E2.S1.sname,E2.S2.sid,E2.S2.sname}(E2 \overline{\ltimes} E1) \\
E4 &= \pi_{S1.*,S2.*}(\sigma_{S1.sid<>S2.sid}(S1 \times S2)) \\
E5 &= \pi_{S1.*,S2.*}(E4 \overline{\ltimes} E3) \\
E6 &= \pi_{S1.sid,S2.sid}(E5)
\end{aligned}
$$

$E6$ is the final result.

We can now begin with the optimization. **I will release this optimization ASAP**

Observe that the names of student do not feature in this query at all. This mean that the sname attributes can be projected out of $S1$ and $S2$. So in the above expression we can replace $S1$ by $\pi_{S1.sid}(S1)$ and $S2$ by $\pi_{S2.sid}(S2)$. Doing this would make the expressions quite tedious to read and rather than doing this, let us agree that from now on $S1$ and $S2$ are a convenient notation for these projections. So we get the expressions:

$$
\begin{aligned}
E1 &= \pi_{Buys1.*,S1.sid,S2.sid}(\sigma_{Buys.sid=S2.sid \wedge Buys1.bookno=Buys.bookno}(\\
&\qquad Buys \times Buys1 \times S1 \times S2)) \\
E2 &= \pi_{Buys1.*,S1.sid,S2.sid}(\sigma_{Buys1.sid=S1.sid}(Buys1 \times S1 \times S2)) \\
E3 &= \sigma_{E2.S1.sid,E2.S2.sid}(E2 \,\overline{\ltimes}\, E1) \\
E4 &= \pi_{S1.sid,S2.sid}(\sigma_{S1.sid<>S2.sid}(S1 \times S2)) \\
E5 &= \pi_{S1.sid,S2.sid}(E4 \,\overline{\ltimes}\, E3) \\
E6 &= \pi_{S1.sid,S2.sid}(E5)
\end{aligned}
$$

We can then push selections down over $\times$ and introduce joins. We can do this for expression $E1$, $E2$ and $E4$. A more subtle notion is that we can push the selection $\sigma_{Buys1.sid=S1.sid}$ inside $E1$. This will not affect the subsequent anti-semijoin. The same can be done recursively for the selection condition $\sigma_{S1.sid<>S2.Sid}$.

After having done this, we can observe that $E1 \subseteq E2$ and $E3 \subseteq E4$, and therefore $E2 \,\overline{\ltimes}\, E1$ can be rewritten as $E2 - E1$ and $E4 \,\overline{\ltimes}\, E3$ can be rewritten as $E4 - E3$.[1] Finally, it should be obvious that in this specific case, we can also get rid of all the $\pi_{S1.sid,S2.sid}$ operations in expressions $E4$, $E5$ and $E6$.[2] And then $E5$ becomes the final result.

[1] Indeed, $E2 \,\overline{\ltimes}\, E2 = E2 - (E2 \ltimes E1)$, and because $E1 \subseteq E2$, we have $E2 \ltimes E1 = E2$, thus $E2 \,\overline{\ltimes}\, E1 = E2 - E1$. A similar argument establishes that $E4 \,\overline{\ltimes}\, E3 = E4 - E3$.

[2] Notice however that in general projections can not be pushed over set differences.

$$
\begin{aligned}
E1 &= \pi_{Buys1.*,S1.sid,S2.sid}((S1 \bowtie_{S1.sid<>S2.Sid} S2) \bowtie_{s1.Sid=Buys1.sid \wedge s2.sid=Buys2.sid} \\
&\qquad (Buys1 \bowtie_{Buys1.bookno=Buys2.bookno} Buys2)) \\
E2 &= \pi_{Buys1.*,S1.sid,S2.sid}((Buys1 \bowtie_{Buys1.sid=S1.sid} (S1)) \bowtie_{S1.sid<>S2.sid} S2) \\
E3 &= E2 - E1 \\
E4 &= S1 \bowtie_{S1.sid<>S2.Sid} S2 \\
E5 &= E4 - E3
\end{aligned}
$$