Assingment 6 Solutions

Prudhvi Vajja

31-3-2020

1. a)Show how such SQL queries can be translated to equivalent RA expressions.

   I) In Case:

$$\pi_{L1(R1)}(\sigma_{C1}(R1) \ltimes (R2 \bowtie_{C2(S1,R2)} S1[\cap| \cup |-]R3 \bowtie_{C3(T1,R3)} T1))$$

   II) Not In Case:

$$\pi_{L1(R1)}(\pi_{R1.*}(\sigma_{C1}(R1)) - \pi_{R2.*}(R2 \bowtie_{C2(S1,R2)} S1[\cap| \cup |-]R3 \bowtie_{C3(T1,R3)} T1))$$

   III) General Case (IN):

$$\pi_{L(R_1,...,R_k)}(\sigma_{C_1}(\mathbf{R_1}) \ltimes (\mathbf{R_2} \bowtie_{C_2(\mathbf{S},\mathbf{R_2})} \mathbf{S}[\cap| \cup |-]\mathbf{R_3} \bowtie_{C_3(\mathbf{T},\mathbf{R_3})} \mathbf{T}))$$

   where

$$\mathbf{R} = R_1 \times ... \times R_k$$
$$\mathbf{S} = S_1 \times ... \times S_m$$
$$\mathbf{T} = T_1 \times ... \times T_n$$

   IV) General Case (NOT IN):

$$\pi_{L(R_1,...,R_k)}(\pi_{R.*}(\sigma_{C_1}(\mathbf{R_1})) - \pi_{R_1.*}((\mathbf{R_2} \bowtie_{C_2(\mathbf{S},\mathbf{R_2})} \mathbf{S}[\cap| \cup |-]\mathbf{R_3} \bowtie_{C_3(\mathbf{T},\mathbf{R_3})} \mathbf{T})))$$

   where

$$\mathbf{R} = R_1 \times ... \times R_k$$
$$\mathbf{S} = S_1 \times ... \times S_m$$
$$\mathbf{T} = T_1 \times ... \times T_n$$

   b)Show how your translation can be improved when the variable âr1â does not occur in the conditions C2 and C3 in the sub-query.

   I) In Case:

$$\pi_{L1(R1)}(\sigma_{C1}(R1) \ltimes (\pi_{S1.B1}(\sigma_{C2}(S1))[\cap| \cup |-]\pi_{T1.C1}(\sigma_{C3}(T1))))$$

   II) Not In Case:

$$\pi_{L1(R1)}(\pi_{R1.*}(\sigma_{C1}(R1)) - \pi_{R2.*}(R2 \ltimes (\pi_{S1.B1}(\sigma_{C2}(S1))[\cap| \cup |-]\pi_{T1.C1}(\sigma_{C3}(T1)))))$$

2. Prove the correctness of the following rewrite rule:

$$\begin{aligned}
\pi_a(R \bowtie_{R.a=S.b \wedge R.b=S.a} S) &= \pi_a(\pi_{a,b}(R) \cap \pi_{b,a}(S)) \\
\pi_a(R \bowtie_{R.a=S.b \wedge R.b=S.a} S) &= \{(a) | \exists a \exists b((R_{a,b,c}) \wedge_{R.a=S.b} \wedge_{R.b=S.a} (S_{b,a,d}))\} \\
&= \{(a) | ((a,b) \in R_{a,b}) \wedge_{R.a=S.b} \wedge_{R.b=S.a} ((b,a) \in S_{b,a}))\} \\
&= \{(a) | \pi_{a,b}(R) \wedge_{R.a=S.b} \wedge_{R.b=S.a} \pi_{b,a}(S)\} \\
&= \{(a) | \pi_{a,b}(R) \bowtie \pi_{b,a}(S)\} \\
&= \{(a) | \pi_{a,b}(R) \cap \pi_{b,a}(S)\} \\
&= \pi_a(\pi_{a,b}(R) \cap \pi_{b,a}(S))
\end{aligned}$$

Similarly we can prove that

$$\pi_a(\pi_{a,b}(R) \cap \pi_{b,a}(S)) = \pi_a(R \bowtie_{R.a=S.b \wedge R.b=S.a} S)$$

Case 1: If there exists a (a,b) pair in R which has a corresponding (b,a) pair in S
then by joining r.a with s.b and r.b with s.a gives the same output from given case 1 conditions.

3. Find the sid and name of each student who majors in âCSâ and who bought a book that cites a higher priced book.

a)

$$\begin{aligned}
\pi_{S.sid,S.sname}(S \bowtie (\sigma_{major='CS'}(M)) \bowtie T) \bowtie_{T.bookno=C.bookno} ((C \bowtie_{C.bookno=B1.bookno} B1) \\
\bowtie_{C.citedbookno=B2.bookno \wedge B1.price<B2.price} B2)
\end{aligned}$$

b) 1. Select only required columns from the table
2. used semi-joins whenever necessary
3. used natural joins instead of joins and cross joins.

$$\begin{aligned}
\pi_{S.sid,S.sname}(S \ltimes \pi_{T.sid}((\pi_{m.sid}(\sigma_{major='CS'}(M)) \bowtie T) \ltimes (\pi_{C.bookno}(\pi_{C.citedbookno,B1.price}(C \bowtie B1) \\
\bowtie_{C.citedbookno=B2.bookno \wedge B1.price<B2.price} (\pi_{B1.bookno,B1.price} B2)))))
\end{aligned}$$

4. Find the sid, name, and major of each student who (i). does not major in CS, (ii).did not buy a book that less than 30, (iii).bought some book(s) that cost less than less than 50.

a)

$$\begin{aligned}
B1 = \pi_{S.*,M.*}((S \bowtie M) \bowtie \pi_{T.sid}(T \bowtie_{b.bookno=t.bookno,b.price<60} B)) - \pi_{S.*,M.*}(S \bowtie_{m.major='CS'} M) \\
\pi_{S.sid,S.sname,M.major}(B1 - pi_{S.*,M.*}((S \bowtie M) \bowtie \neg_{S.sid=T1.sid}(\pi_{T1.sid}(T1 \bowtie_{b2.bookno=t1.bookno,b2.price<30} B2))))
\end{aligned}$$

b) 1.moved Selections inside the functions
2.Used semi joins when ever needed.

$$\begin{aligned}
B1 &= \pi_{T.sid}(T \bowtie_{b.price<30} (\pi_{B.bookno,B.price} B)) \\
B2 &= \pi_{T.sid}(T \bowtie_{b.price<60} (\pi_{B.bookno,B.price} B)) \\
M1 &= \pi_{M1.sid}(M) - \pi_{M2.sid}(\sigma_{M2.major='CS'}(M2)) \\
S1 &= S \bowtie M \\
&\pi_{S1.sid,S1.sname,S1.major}(S1 \ltimes \pi_{M1.sid}(M1 \bowtie (B1 - B2)))
\end{aligned}$$

5. Find each (s, n, b) triple where s is the sid of a student, n is the name of this student, and where b is the bookno of a book whose price is the most expensive among the books bought by that student.

   a)

$$A1 = \pi_{s.sid,s.sname,b.bookno}(S \bowtie_{s.sid=t.sid} T \bowtie_{t.bookno=b.bookno} B)$$
$$A2 = \pi_{s.sid,s.sname,b1.bookno}(S \bowtie_{s.sid=t1.sid} T1 \bowtie_{t1.bookno=b1.bookno} B1 \bowtie_{s.sid=t2.sid} T2 \bowtie_{t2.bookno=b2.bookno \wedge b1.price<b2.price} B2)$$
$$\pi_{a1.sid,a1.sname,a1.bookno}(A1 - A2)$$

   b) 1. remove unused tables
   2. used natural joins

$$A1 = \pi_{s.sid,s.sname,b.bookno}((S \bowtie T))$$
$$A2 = \pi_{t1.sid,t1.bookno,b1.price}(T1 \bowtie (\pi_{b1.bookno,b1.price} B1))$$
$$A3 = \pi_{a2.sid,a2.bookno}(A2 \bowtie_{a2.sid=a21.sid \wedge a1.bookno<>a21.bookno \wedge a2.price<a21.price} A2(a21))$$
$$\pi_{a1.sid,a1.sname,a1.bookno}(A1 - (S \bowtie A3))$$

6. Find the bookno and title of each book that is not bought by all students who major in both CS or in Math.

   a)

$$\pi_{b1.bookno,b1.title}(B \ltimes (\pi_{s.sid,b.*}(S \times B) \bowtie_{s.sid=q.sid} (\pi_{q.sid}(\sigma_{m1.major='CS'}(M1) \cup \sigma_{m2.major='Math'}(M2))q)) - (\pi_{s1.sid,b1.*}(S1 \times B1) \bowtie_{s1.sid=t.sid \wedge t.bookno=b1.bookno} T)$$

   b)

$$A1 = \pi_{m.sid}(\sigma_{m1.major='CS'}(M1) \cup \sigma_{m2.major='Math'}(M2))$$
$$\pi_{b.bookno,b.title}(\pi_{B.*}B \ltimes (\pi_{B1.*,A1.sid}(B1 \times A1) - \pi_{B2.*,T.sid}(B2 \bowtie T)))$$

7. select distinct r1.a from R r1, R r2, R r3 where r1.b = r2.a and r2.b = r3.a;

makerandomR	makerandomS	Q3(in ms)	Q4(in ms)
(1000,600,500)	(600,500)	0.516	0.687
(100,100,1000)	(100,1000)	34.093	1.126
(500,500,25000)	(500,25000)	15281.439	25.061
(1000,1000,50000)	(1000,50000)	22343.522 ms	50.123
(1000,1000,100000)	(1000,100000)	–	101.478

   What conclusions can you draw from the results of these experiments?
   1.Combining two tables using natural join instead of reduces the space complexity for the algorithm to search across the joined table.
   2. Q4 is very fast than Q3(which is cubic in terms of runtime).

8. select ra.a from Ra ra where not exists (select from where r.b Rr r.a = ra.a and r.b not in (select s.b from S s));

makerandomR	makerandomS	Q5(in ms)	Q6(in ms)
(5000,3000,2000)	(3000,2000)	2.125	6.246
(1000,1000,10000)	(1000,10000)	2.484	13.145
(5000,5000,100000)	(5000,100000)	25.388	129.482
(5000,5000,2500000)	(5000,2500000)	458.840	5692.386

What conclusions can you draw from the results of these experiments?
1. Though we are using optimized query without any set predicates it is more time than normal query.
2. Q5 is faster than Q6

9. select ra.a from Ra ra where not exists (select s.b from Ss where s.b not in (select r.b from Rr where r.a = ra.a));

makerandomR	makerandomS	Q7(in ms)	Q8(in ms)
(1000,600,500)	(600,500)	88.380	9.575
(100,100,1000)	(100,1000)	90.138	9.789
(500,500,2500)	(500,2500)	140.551	247.563
(500,500,10000)	(500,10000)	283.841	252.591
(1000,1000,250000)	(1000,250000)	12046.690	1383.375
(5000,5000,2500000)	(5000,2500000)	–	34841.851

What conclusions can you draw from the results of these experiments?
Furthermore, what conclusion can you draw when you compare you experiment with those for the ONLY set semijoin in problem 8?
1. optimized query Q8 is faster than Q7
2. It is opposite to what we observed while using ONLY condition between R and S
3. ONLY is taking less time to execute than All condition between the two tables for same number of inputs.

10. Explain briefly how query Q9 works and how it solves the problem.

– NestedR contains r.a and all the corresponding r.b values in a array.
with NestedR as (select r.a, arrayagg(r.b order by 1) as Bs from Rr group by (r.a)),
–SetS contains all the distinct s.b values.
SetS as (select array(select s.b from S s order by 1) as Bs)
–Returns a if it has all the corresponding B in S
select r.a from NestedR r, SetS s where r.Bs <@ s.Bs union
–Returns all a values from Ra that are not in R
(select ra.a from Ra ra except select r.a from R r);

makerandomR	makerandomS	Q5(in ms)	Q6(in ms)	Q9(in ms)
(5000,3000,2000)	(3000,2000)	2.125	6.246	28.536
(1000,1000,10000)	(1000,10000)	2.484	13.145	33.292
(5000,5000,100000)	(5000,100000)	25.388	129.482	791.055
(5000,5000,2500000)	(5000,2500000)	458.840	5692.386	17647.720

What conclusions can you draw from the results of these experiments?
1. Q9 is slower than both normal and optimized query's Q5 and Q6.

11. Explain briefly how query Q10 works and how it solves the problem.

– NestedR contains r.a and all the corresponding r.b values in a array.
with NestedR as (select r.a, arrayagg(r.b order by 1) as Bs from Rr group by (r.a)),
–SetS contains all the distinct s.b values.
SetS as (select array(select s.b from S s order by 1) as Bs)
–Returns a if it has all the corresponding B in S
select r.a from NestedR r, SetS s where r.Bs <@ s.Bs

a & b)

makerandomR	makerandomS	Q7(in ms)	Q8(in ms)	Q10
(1000,600,500)	(600,500)	88.380	9.575	0.779
(100,100,1000)	(100,1000)	90.138	9.789	0.725
(500,500,2500)	(500,2500)	140.551	247.563	2.399
(500,500,10000)	(500,10000)	283.841	252.591	8.037
(1000,1000,250000)	(1000,250000)	12046.690	1383.375	194.439
(5000,5000,2500000)	(5000,2500000)	–	34841.851	1795.151
(6000,6000,3000000)	(6000,3000000)	–	50557.680	2232.592
(500,500,5000)	(500,5000)	–	269.381	4.255

What conclusions can you draw from the results of these experiments?
1. RunTimes »» Q10 < Q8 < Q7
2. Runtime has decresed drastically by using object relational database of arrays 3. We can observe from the last run(500,500,5000) that even for small inputs optimized query is taking 270 ms whereas object relational query Q10 is taking just 4ms