SAMPLE SOLUTION
Assignment 2
CIS 452/552, Winter 2004

These are the sample solutions that came with the text. I have made some annotations.

Q1. Exercise 4.2 Given two relations \( R_1 \) and \( R_2 \), where \( R_1 \) contains \( N_1 \) tuples, \( R_2 \) contains \( N_2 \) tuples, and \( N_2 > N_1 > 0 \), give the minimum and maximum possible sizes (in tuples) for the resulting relation produced by each of the following relational algebra expressions. In each case, state any assumptions about the schemas for \( R_1 \) and \( R_2 \) needed to make the expression meaningful:

1. \( R_1 \bowtie R_2 \)
2. \( R_1 \cap R_2 \)
3. \( R_1 - R_2 \)
4. \( R_1 \times R_2 \)
5. \( \sigma_{a=b}(R_1) \)
6. \( \pi_a(R_1) \)
7. \( R_1 / R_2 \)

Answer 4.2 See Figure 4.1.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Assumption</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_1 \cup R_2 )</td>
<td>( R_1 ) and ( R_2 ) are union-compatible</td>
<td>( N_2 )</td>
<td>( N_1 + N_2 )</td>
</tr>
<tr>
<td>( R_1 \cap R_2 )</td>
<td>( R_1 ) and ( R_2 ) are union-compatible</td>
<td>0</td>
<td>( N_1 )</td>
</tr>
<tr>
<td>( R_1 - R_2 )</td>
<td>( R_1 ) and ( R_2 ) are union-compatible</td>
<td>0</td>
<td>( N_1 )</td>
</tr>
<tr>
<td>( R_1 \times R_2 )</td>
<td>( R_1 ) and ( R_2 ) are union-compatible</td>
<td>( N_1 \times N_2 )</td>
<td>( N_1 \times N_2 )</td>
</tr>
<tr>
<td>( \sigma_{a=b}(R_1) )</td>
<td>( R_1 ) has an attribute named ( a )</td>
<td>0</td>
<td>( N_1 )</td>
</tr>
<tr>
<td>( \pi_a(R_1) )</td>
<td>( R_1 ) has attribute ( a ), ( N_1&gt;0 )</td>
<td>1</td>
<td>( N_1 )</td>
</tr>
<tr>
<td>( R_1 / R_2 )</td>
<td>The set of attributes of ( R_2 ) is a subset of the set of attributes of ( R_1 )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( R_2 / R_1 )</td>
<td>The set of attributes of ( R_1 ) is a subset of the set of attributes of ( R_2 )</td>
<td>0</td>
<td>( \left[ N_2 / N_1 \right] )</td>
</tr>
</tbody>
</table>

Figure 4.1 Answer to Exercise 4.2.

Q2. Exercise 4.4 Consider the Supplier-Parts-Catalog schema from the previous question.

State what the following queries compute:

1. \( \pi_{\text{name}}(\pi_{\text{sid}}((\sigma_{\text{color}=\text{red}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers}) \)
2. \( \pi_{\text{name}}(\pi_{\text{sid}}((\sigma_{\text{color}=\text{red}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers}) \)
3. \( (\pi_{\text{name}}((\sigma_{\text{color}=\text{red}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers})) \cap (\pi_{\text{name}}((\sigma_{\text{color}=\text{green}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers})) \)
4. \( (\pi_{\text{sid}}((\sigma_{\text{color}=\text{red}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers})) \cap (\pi_{\text{sid}}((\sigma_{\text{color}=\text{green}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers})) \)
5. \( \pi_{\text{name}}(\pi_{\text{sid}}(\pi_{\text{name}}((\sigma_{\text{color}=\text{red}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers})) \cap (\pi_{\text{sid}}(\pi_{\text{name}}((\sigma_{\text{color}=\text{green}} \text{ Parts}) \circ (\sigma_{\text{cost}<100} \text{ Catalog})) \circ \text{ Suppliers})) \)

Answer 4.4 The statements can be interpreted as:
1. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars. [Note: clearly they meant \( \pi_{\text{name}} \).]

2. This Relational Algebra statement does not return anything because of the sequence of projection operators. Once the sid is projected, it is the only field in the set. Therefore, projecting on sname will not return anything.

3. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars. [Actually, no. If there are two different suppliers with the same name, one supplying the red and the other the green, that name will be returned. This is because sname is not a key.]

4. Find the Supplier ids of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

5. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

Q3.

Retrieve the names of all employees in department 5 who work at least 10 hours per week on the ‘ProductX’ project.

RA: \( \pi_{\text{name}} \left( \sigma_{\text{hours} \geq 10} \left( \text{employee join}_{\text{ssn} = \text{essn}} \text{ works on join}_{\text{pno} = \text{pnumber}} \text{ project} \right) \right) \)

[Sorry, I can’t find the join symbol in Word.]

TRC: [The attributes of T will be those referred to in the body of the query.]
\{ T | \exists E \in \text{employee}, \exists W \in \text{works on}, \exists P \in \text{project} \ ( T.\text{name} = E.\text{name} \wedge T.\text{name} = E.\text{name} \wedge E.\text{ssn} = W.\text{ssn} \wedge W.\text{pno} = P.\text{pnumber} \wedge E.\text{dno} = 5 \wedge W.\text{hours} \geq 10 \wedge P.\text{name} = '\text{ProductX}' ) \}

DRC: [We’ll use the abbreviated version where there are implicit \( \exists \) quantifiers.]
\{ <F,L,A> | <F, *, L, s, *, *, *, 5> \in \text{employee} \wedge <s, p, h> \in \text{works on} \wedge <'\text{ProductX}', p, *, *> \in \text{project} \wedge h \geq 10 \}

Give the names of all employees who make less money than Ramesh Narayanan.

RA: \( \pi_{\text{e1.name}, \text{e1.name}} \left( \sigma_{\text{e1.salary} < \text{e2.salary}} \left( \text{e2.fname = 'Ramesh' \&\& e2.lname = 'Narayanan'} \right) \left( \rho_{\text{e1.employee}} \circ \rho_{\text{e2.employee}} \right) \right) \)

TRC: \{ P | \exists E \in \text{employee}, \exists R \in \text{employee} \ (P.\text{name} = E.\text{name} \wedge P.\text{name} = E.\text{name} \wedge E.\text{salary} < R.\text{salary} \wedge R.\text{fname} = 'Ramesh' \wedge R.\text{lname} = 'Narayan' ) \}

DRC: \{ <F,L> | <F, *, L, *, *, *, *, s1, *, *> \in \text{employee} \wedge <'Ramesh', *, 'Narayanan', *, *, *, s2, *, *> \in \text{employee} \}
\[ s_1 < s_2 \]

Retrieve the names of all employees who work on every project.

RA: \[ \pi_{\text{fname}, \text{lname}}(\pi_{\text{essn}, \text{pno}}(\text{works on})/\pi_{\text{pnumber}}(\text{project})) \text{ join}_{\text{essn}=\text{ssn}} \text{ employee} \]

TRC: \{ P \mid \exists E \in \text{employee, } \forall R \in \text{project, } \exists W \in \text{works on} (P.\text{fname}=E.\text{lname} \land P.\text{lname}=E.\text{lname} \land E.\text{ssn}=W.\text{ssn} \land W.\text{pno}=R.\text{pnumber} \}

DRC: [We’ll do the full quantification.]

\{ <F, L> \mid \exists \text{em, ES, eb, ea, ex, el, ep, ed } <F, \text{em,L,es,eb,ea,ex,ep,ed}> \in \text{employee} \land \forall \text{pn, P#, pl, pd } [ <\text{pn,p#,pl,pl}> \in \text{project} \Rightarrow \exists \text{ h } <\text{ES,P#,h}> \in \text{works on} ] \}