Problem 1: Consider two RA expressions E_1 and E_2 over the same schema. Furthermore, consider an RA expression F with a schema that is not necessarily the same as that of E_1 and E_2.

Consider the following if-then-else query:

if $F =$ then $\emptyset$ return E_1
else return E_2

➔
Let

$Q_1 = \pi_{()}(F)$AND $Q_2 = \pi_{(True:arecord)}(Q_1)$ And E_1 *as expression* 1 *and* E_2 *as Expression* 2

And $Q_3 = \pi_*(E_1) - \pi_{E_1.*}(\sigma_{Q_2.arecord=True}(E_1 \times Q_2))$

Then the final query is

$$Q_3 \cup \pi_{E_2.*}(\sigma_{Q_2.arecord=True}(E_2 \times Q_2))$$

Problem 2: Let R(x) be a unary relation that can store a set of integers R. Consider the following boolean SQL query:

```
select not exists(select 1
                  from   R r1, R 2
                  where  r1.x <> r2.x) as fewerThanTwo;
```

This boolean query returns the constant "true" if R has fewer than two elements and returns the constant "false" otherwise.

-→
Let

$Q_1 = \pi_1\left(\sigma_{R_1.x \neq R_2.x}(R_1 \times R_2)\right)$

AND

$Q_2 = \pi_{(():fewerThanTwo)}(Q_1)$

Then the final query is

$$(\pi_{(True:fewerThanTwo)} - \pi_{(True:fewerThanTwo)}(Q_2) \cup \ \pi_{(False:fewerThanTwo)}(Q_2)$$

Problem 3:

A) exists and union

➔ Corresponding RA will be

Let

$$Q_1 = \pi_{L_1.(r_1,\ldots,r_n),L_2.(s_1,\ldots,s_m)} \left(\sigma_{C_1(r_1,\ldots,r_n)\ \theta\ C_2.(s_1,\ldots,s_m,r_1,\ldots,r_n)}(S_1 \times \ldots\ldots S_m \times R_1 \ldots\ldots \times R_n)\right)$$

And

$$Q_2 = \pi_{L_1.(r_1,\ldots,r_n),L_3.(t_1,\ldots,t_k)} \left(\sigma_{C_1(r_1,\ldots,r_n)\ \theta\ C_3.(t_1,\ldots,t_k,r_1,\ldots,r_n)}(T_1 \times \ldots\ldots T_k \times R_1 \ldots\ldots \times R_n)\right)$$

Then the final query be

$= \pi_{L^q.(r_1,\ldots,r_n)}(Q_1 \cup Q_2)$

B) exists and intersect

➔ Corresponding RA will be

Let

$$Q_1 = \pi_{L_1.(r_1,\ldots,r_n),L_2.(s_1,\ldots,s_m)} \left(\sigma_{C_1(r_1,\ldots,r_n)\ \theta\ C_2.(s_1,\ldots,s_m,r_1,\ldots,r_n)}(S_1 \times \ldots\ldots S_m \times R_1 \ldots\ldots \times R_n)\right)$$

And

$$Q_2 = \pi_{L_1.(r_1,\ldots,r_n),L_3.(t_1,\ldots,t_k)} \left(\sigma_{C_1(r_1,\ldots,r_n)\ \theta\ C_3.(t_1,\ldots,t_k,r_1,\ldots,r_n)}(T_1 \times \ldots\ldots T_k \times R_1 \ldots\ldots \times R_n)\right)$$

Then the final query be

$= \pi_{L^q.(r_1,\ldots,r_n)}(Q_1 \cap Q_2)$

C) exists and except

➔ Corresponding RA will be

Let

$$Q_1 = \pi_{L_1.(r_1,\ldots,r_n),L_2.(s_1,\ldots,s_m)} \left(\sigma_{C_1(r_1,\ldots,r_n)\ \theta\ C_2.(s_1,\ldots,s_m,r_1,\ldots,r_n)}(S_1 \times \ldots\ldots S_m \times R_1 \ldots\ldots \times R_n)\right)$$

And

$$Q_2 = \pi_{L_1.(r_1,\ldots,r_n),L_3.(t_1,\ldots,t_k)} \left(\sigma_{C_1(r_1,\ldots,r_n)\ \theta\ C_3.(t_1,\ldots,t_k,r_1,\ldots,r_n)}(T_1 \times \ldots\ldots T_k \times R_1 \ldots\ldots \times R_n)\right)$$

Then the final query be

= $\pi_{L^q.(r_1,,,,,,,,,,,,,,r_n)}(Q_1 - Q_2)$

D) not exists and union

Let

$$Q_1 = \pi_{L_1.(r_1,,,,,,,,,,,,,,r_n)} \; (\sigma_{C_1(r_1,,,,,,,,,,,,,,r_n)} \; (R_1 \ldots . \times R_n))$$

and

$$Q_2 \; = \; \pi_{L_1.(r_1,,,,,,,,,,,,,,r_n),L_2.(s_1,,,,,,,,,,,,,,s_m)} \; (\sigma_{C_1(r_1,,,,,,,,,,,,,,r_n)\;\theta\;C_2.(s_1,,,,,,,,,,,,,,s_m,r_1,,,,,,,,,,,,,,r_n)}(S_1 \; \times \ldots \ldots S_m \times \; R_1 \ldots . \times R_n))$$

And

$$Q_3 \; = \; \pi_{L_1.(r_1,,,,,,,,,,,,,,r_n),L_3.(t_1,,,,,,,,,,,,,,t_k)} \; (\sigma_{C_1(r_1,,,,,,,,,,,,,,r_n)\;\theta\;C_3.(t_1,,,,,,,,,,,,,,t_k,r_1,,,,,,,,,,,,,,r_n)}(T_1 \; \times \ldots \ldots T_k \times \; R_1 \ldots . \times R_n))$$

And

$$Q_4 \; = \; \pi_{L^q.(r_1,,,,,,,,,,,,,,r_n)} \; (Q_2 \; \cup Q_3)$$

Then the final query be

= $\pi^1_{L^q.(r_1,,,,,,,,,,,,,,r_n)}(Q_1 - Q_4)$

E) not exists and intersect

Let

$$Q_1 = \pi_{L_1.(r_1,,,,,,,,,,,,,,r_n)} \; (\sigma_{C_1(r_1,,,,,,,,,,,,,,r_n)} \; (R_1 \ldots . \times R_n))$$

and

$$Q_2 \; = \; \pi_{L_1.(r_1,,,,,,,,,,,,,,r_n),L_2.(s_1,,,,,,,,,,,,,,s_m)} \; (\sigma_{C_1(r_1,,,,,,,,,,,,,,r_n)\;\theta\;C_2.(s_1,,,,,,,,,,,,,,s_m,r_1,,,,,,,,,,,,,,r_n)}(S_1 \; \times \ldots \ldots S_m \times \; R_1 \ldots . \times R_n))$$

And

$$Q_3 \; = \; \pi_{L_1.(r_1,,,,,,,,,,,,,,r_n),L_3.(t_1,,,,,,,,,,,,,,t_k)} \; (\sigma_{C_1(r_1,,,,,,,,,,,,,,r_n)\;\theta\;C_3.(t_1,,,,,,,,,,,,,,t_k,r_1,,,,,,,,,,,,,,r_n)}(T_1 \; \times \ldots \ldots T_k \times \; R_1 \ldots . \times R_n))$$

And

$$Q_4 \; = \; \pi_{L^q.(r_1,,,,,,,,,,,,,,r_n)} \; (Q_2 \; \cap Q_3)$$

Then the final query be

$= \pi^{1}_{L^q.(r_1,,,,,,,,,,,,,,,,r_n)}(Q_1 \ - Q_4)$

F) not exists and except

Let

$Q_1 = \pi_{L_1.(r_1,,,,,,,,,,,,,,,,r_n)} \ (\sigma_{C_1(r_1,,,,,,,,,,,,,,,,r_n)} \ (R_1.\dots.\times R_n))$

and

$$Q_2 \ = \ \pi_{L_1.(r_1,,,,,,,,,,,,,,,,r_n),L_2.(s_1,,,,,,,,,,,,,,,,s_m)} \ (\sigma_{C_1(r_1,,,,,,,,,,,,,,,,r_n) \ \theta \ C_2.(s_1,,,,,,,,,,,,,,,,s_m,r_1,,,,,,,,,,,,,,,,r_n)}(S_1 \ \times\dots\dots S_m \times \ R_1.\dots.\times R_n))$$

And

$$Q_3 \ = \ \pi_{L_1.(r_1,,,,,,,,,,,,,,,,r_n),L_3.(t_1,,,,,,,,,,,,,,,,t_k)} \ (\sigma_{C_1(r_1,,,,,,,,,,,,,,,,r_n) \ \theta \ C_3.(t_1,,,,,,,,,,,,,,,,t_k,r_1,,,,,,,,,,,,,,,,r_n)}(T_1 \ \times\dots\dots T_k \times \ R_1.\dots.\times R_n))$$

And

$Q_4 \ = \ \pi_{L^q.(r_1,,,,,,,,,,,,,,,,r_n)} \ (Q_2 \ - Q_3)$

Then the final query be

$= \pi^{1}_{L^q.(r_1,,,,,,,,,,,,,,,,r_n)}(Q_1 \ - Q_4)$

Problem 4:

RA expression is : $\pi_{a,d} \ (R \bowtie_{c=d} S) \ = \ \pi_{a,d} \ (\pi_{a,c} \ (R) \bowtie_{c=d} \pi_d \ (S))$

➔ Here we can write the LHS as

$\pi_{a,d} \ (R \bowtie_{c=d} S) \ = \{\mathrm{r.a, r.d} | \ \mathrm{R(r)} \wedge \mathrm{S(s)} \wedge \mathrm{r.c = s.d}\}$

$= \{\mathrm{r.a, r.d} | \ \exists \ \mathrm{r} \ \exists \ \mathrm{s} \ \in (\mathrm{R(a,b,c)} \wedge \mathrm{S(d,e)} \wedge \mathrm{r.c = s.d})\}$

$= \{\mathrm{r.a, r.d} | \ \exists \ \mathrm{r} \in \ \mathrm{R(a,b,c)} \wedge \exists \ \mathrm{s} \in \ \mathrm{S(d,e)} \wedge \mathrm{r.c = s.d}\}$

$= \{\mathrm{r.a, r.d} | \ \{\mathrm{r.a, r.c} \ |\mathrm{R(a,b,c)}\} \ \wedge \{\mathrm{s.d} | \ \mathrm{S(d,e)}\} \wedge \mathrm{r.c = s.d}\}$

And this can be translated to RA as

$$= \pi_{\mathrm{R}.a,\mathrm{R}.d} \ (\pi_{\mathrm{R}.a,\mathrm{R}.c} \ (R) \bowtie_{\mathrm{R}.c=\mathrm{S}.d} \pi_{\mathrm{S}.d} \ (S))$$

Hence Proved, LHS=RHS and $\pi_{a,d} \ (R \bowtie_{c=d} S) \ = \ \pi_{a,d} \ (\pi_{a,c} \ (R) \bowtie_{c=d} \pi_d \ (S))$

Problem 5:

RA expression is : $\pi_{a,d}\ (R \bowtie_{c=d} S) = \pi_{a,d}\ (\pi_{a,c}\ (R) \bowtie_{c=d} \pi_d\,(S))$

→ This query can be simplified as

$$\pi_{a,d}\ (R \bowtie_{c=d} S) = \pi_{a,c}(\mathrm{R})$$

Since here S has primary key d and that R has foreign key c referencing this primary key in S. We can use attribute c from relation in r in final projection. This constraint can be denoted by following RA expression.

$$\pi_d\,(S) - \pi_{\mathrm{c}}\,(\mathrm{R}) = \emptyset$$

Problem 6:

Consider the query "*Find the cname and headquarter of each com- pany that employs persons who earn less than 55000 and who do not live in Bloomington.*"

→

Let

$Q_1 = \pi_{P.pid}\ (\sigma_{P.city \neq 'Bloomington'}(P))$

AND

$Q_2 = \pi_{\mathrm{W.pid,W.cname}}(\sigma_{W.salary < 55000}(W))$

And

$Q_3 = \pi_{\,Q_2.\mathrm{cname}}\,(Q_2 \cap \pi_{\,Q_2.\mathrm{cname}}\,((Q_1 \bowtie_{Q_1.pid=Q_2.pid} Q_2)))$

The final query is

$$\pi_{\,C.\mathrm{cname},C.\mathrm{headquarter}}\left(C \cap \pi_{\,C.\mathrm{cname},C.\mathrm{headquarter}}\left(C \bowtie_{C.cname=Q_3.cname} Q_3\right)\right)$$

<u>Optimizations</u>

1. *Pushing projections over semi join*
    a. Q_3 *can be optimized as by pushing projections over join as*
    b. $Q_{Opt} = \pi_{\,Q_2.\mathrm{cname}}\,(Q_2 \cap (\pi_{Q_2.\mathrm{pid}}(Q_2) \ltimes Q_1)))$
2. Idempotence of ∩
    a. $Q_{Opt} = \pi_{\,Q_2.\mathrm{cname}}\,(\pi_{Q_2.\mathrm{pid}}(Q_2) \ltimes Q_1)$
    b. *Thus the final query become like*
    c. $\pi_{\,C.\mathrm{cname},C.\mathrm{headquarter}}\left(C \cap \pi_{\,C.\mathrm{cname},C.\mathrm{headquarter}}\left(C \bowtie_{C.cname=Q_{Opt}.cname} Q_{Opt}\right)\right)$
3. Idempotence of ∩
    a. Final query become like

b. $\left(\pi_{\text{C.cname,C.headquarter}}\left(C \bowtie_{C.cname=Q_{Opt}.cname} Q_{Opt}\right)\right)$

4. *Pushing projections over join*
   a. Final query become like
   b. $C \ltimes Q_{Opt}$

Optimized Query:

$$C \ltimes Q_{Opt}$$

Problem 7: Consider the query "*Find the pid of each person who has all-but-one job skill.*"

-→

Let

$Q_1 = \pi_*(\pi_*(S \times P) - \pi_{S.*,P.*}(\sigma_{pS.pid=P.pid \wedge pS.skill = S.skill}(S \times P \times pS)))$

The final query is

$$\pi_{Q_1.pid}(Q_1 - (\pi_*(\pi_*(\sigma_{S_1.skill \neq S_2.skill}(S_1 \times S_2 \times Q_1)) - \pi_{S_1.*,S_2.*,Q_1.*}(\sigma_{S_1.skill \neq S_2.skill \wedge Q_1.pid=pS.pid \wedge S_1.skill=pS.Skill}(S_1 \times S_2 \times pS \times Q_1))) \cap \pi_*(\pi_*(\sigma_{S_1.skill \neq S_2.skill}(S_1 \times S_2 \times Q_1)) - \pi_{S_1.*,S_2.*,Q_1.*}(\sigma_{S_1.skill \neq S_2.skill \wedge Q_1.pid=pS.pid \wedge S_2.skill=pS.Skill}(S_1 \times S_2 \times pS \times Q_1))))$$

<u>Optimizations</u>

1. *Pushing projections over join*
   a. In case of Q_1 we can push selections over joins such as
   b. $Q_1 = \pi_*(\pi_*(S \times P) - (\pi_{S.*,P.*}(S \bowtie_{pS.skill = S.skill} pS \bowtie_{pS.pid=P.pid} P)))$
2. Since we require only pid and skill of each person, we can further optimize this query by reducing the list of attributes used in projection ---
   a. $Q_1 = \pi_{pid}(\pi_{P.pid,S.skill}(S \times P) - (\pi_{pS.pid,pS.skill}(S \bowtie_{pS.skill = S.skill} pS \bowtie_{pS.pid=P.pid} P)))$

3. Property of primary key and foreign key constraint
   a. Since the relation pS the person skill contains both the skill and pid of each person we can remove the redundant join as
   b. $Q_{Opt} = \pi_{pid}\left(\pi_{P.pid,S.skill}(S \times P) - \left(\pi_{pS.pid,pS.skill}(pS)\right)\right)$
4. Pushing selection over joins
   a. $\pi_{Q_{opt}.pid}(Q_{opt} - (\pi_*(\pi_*(\sigma_{S_1.skill \neq S_2.skill}(S_1 \times S_2 \times Q_{opt})) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_1.skill \neq S_2.skill \wedge Q_{Opt}.pid=pS.pid \wedge S_1.skill=pS.Skill}(S_1 \times S_2 \times pS \times Q_{Opt}))) \cap \pi_*(\pi_*(\sigma_{S_1.skill \neq S_2.skill}(S_1 \times S_2 \times Q_{Opt})) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_1.skill \neq S_2.skill \wedge Q_{Opt}.pid=pS.pid \wedge S_2.skill=pS.Skill}(S_1 \times S_2 \times pS \times Q_{Opt}))))$
   b. $\pi_{Q_{opt}.pid}(Q_{opt} - (\pi_*(\pi_*((S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times Q_{opt})) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_1.skill=pS.Skill}(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times pS \bowtie_{Q_{Opt}.pid=pS.pid} Q_{Opt}))) \cap \pi_*(\pi_*((S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times$

$Q_{opt})) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_2.skill = pS.Skill}(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times pS \bowtie_{Q_{Opt}.pid=pS.pid} Q_{Opt}))))$

5. Relativized De Morgan for ∪ → $(E - F) \cap (E - G) = E - (F \cup G)$
   a. $\pi_{Q_{opt}.pid}(Q_{opt} - (\pi_*(\pi_*((S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times Q_{opt})) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_1.skill = pS.Skill}(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times pS \bowtie_{Q_{Opt}.pid=pS.pid} Q_{Opt}))) \cup \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_2.skill = pS.Skill}(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times pS \bowtie_{Q_{Opt}.pid=pS.pid} Q_{Opt})))$
6. By pushing selections over ∪
   a. $\pi_{Q_{opt}.pid}(Q_{opt} - (\pi_*(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times Q_{opt}) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_1.skill = pS.Skill \vee S_2.skill = pS.Skill}(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times pS \bowtie_{Q_{Opt}.pid=pS.pid} Q_{Opt})$

<u>Optimized Query:</u>

$$\pi_{Q_{opt}.pid}(Q_{opt} - (\pi_*(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times Q_{opt}) - \pi_{S_1.*,S_2.*,Q_{Opt}.*}(\sigma_{S_1.skill = pS.Skill \vee S_2.skill = pS.Skill}(S_1 \bowtie_{S_1.skill \neq S_2.skill} S_2 \times pS \bowtie_{Q_{Opt}.pid=pS.pid} Q_{Opt})$$

Problem 8: Consider the query "*Find the pid and name of each person whoworks for a company located in Bloomington but who does not know any person who lives in Chicago.*"

→

Let

$Q_1 = \pi_{\text{W.pid,W.cname}}\left(\sigma_{W.cname\, =\, cL.cname\, \wedge\, cL.city\, =\, 'Bloomington'}(W \times cL)\right)$

And

$Q_2 = \pi_{P.pid}\ (\sigma_{P.city='Chicago'}(P))$

And

$Q_3 = \pi_{K.pid1}(Q_2 \bowtie_{Q_2.pid\, =\, K.pid2} K))$

The final query is

$$\pi_{P.pid,P.pname}\ ((P \bowtie_{P.pid\, =\, Q_1.pid} Q_1) - (P \bowtie_{P.pid\, =\, Q_1.pid} Q_1 \bowtie_{P.pid\, =\, Q_3.pid1} Q_3))$$

<u>Optimizations</u>

1. Pushing selections over the semi joins (Q_1)
   a. $Q_{opt} = \pi_{\text{W.pid,W.cname}}(\pi_{\text{W.cname}}(W) \ltimes \pi_{\text{cL.cname}}(\sigma_{cL.city\, =\, 'Bloomington'}(cL))$
2. *Removing unnessary projections*

a. $Q_{opt} = \pi_{\text{W.pid}}(\pi_{\text{W.cname}}(W) \ltimes \pi_{\text{cL.cname}}(\sigma_{cL.city = 'Bloomington'}(cL))$

3. Elimination of joins by pushing selections over the semi joins
   a. $Q_{3-opt} = \pi_{K.pid1}(\pi_{K.pid2}(K) \ltimes Q_2)$
4. Elimination of joins by pushing selections over the semi joins
   a. $\pi_{P.pid,P.pname}(\pi_{P.pid}(P) \ltimes Q_{opt}) - \pi_{P.pid,P.pname}(P \bowtie_{P.pid = Q_{opt}.pid} Q_{opt} \bowtie_{P.pid = Q_{3-opt}.pid1} Q_{3-opt}))$
5. Double complementation rule and Relativized De Morgan for ∪
   a. $\pi_{P.pid,P.pname}(\pi_{P.pid}(P) \ltimes Q_{opt}) - \pi_{P.pid,P.pname}(\pi_{P.pid}(P) \ltimes Q_{3-opt}))$

Optimized Query is:

$$\pi_{P.pid,P.pname}(\pi_{P.pid}(P) \ltimes Q_{opt}) - \pi_{P.pid,P.pname}(\pi_{P.pid}(P) \ltimes Q_{3-opt}))$$

Problem 9: Consider the query "*Find the cname and headquarter of each com- pany that (1) employs at least one person and (2) whose work- ers who make at most 70000 have both the programming and AIskills.*"

--→

Let

$Q_1 = \pi_{\text{W.pid,W.cname}}\left(\sigma_{W.salary <= 70000}(W)\right)$

And

$Q_2 = \pi_{pS.pid}(\sigma_{pS.skill='Programming'}(pS))$

And

$Q_3 = \pi_{pS.pid}(\sigma_{pS.skill='AI'}(pS))$

And

$Q_4 = \pi_{C.\text{cname},\text{C.headquarter}}(C \bowtie_{C.cname = W.cname} W)$

And

$$Q_5 = \pi_{Q_1.*,(Q_4.cname:cc),(Q_4.\text{headquarter:hq})}(Q_1 \bowtie_{Q_1.cname = Q_4.cname} Q_4 - \pi_{Q_1.*,Q_4.cname,Q_4.\text{headquarter}}(Q_1 \bowtie_{Q_1.cname = Q_4.cname} Q_4 \bowtie_{Q_1.pid = Q_2.pid} Q_2))$$

$$Q_6 = \pi_{Q_1.*,(Q_4.cname:cc),(Q_4.\text{headquarter:hq})}(Q_1 \bowtie_{Q_1.cname = Q_4.cname} Q_4 - \pi_{Q_1.*,Q_4.cname,Q_4.\text{headquarter}}(Q_1 \bowtie_{Q_1.cname = Q_4.cname} Q_4 \bowtie_{Q_1.pid = Q_3.pid} Q_3))$$

And

$Q_7 = \pi_*(Q_5 \cup Q_6)$

Final query is

$$\pi_*(\pi_*(Q_4) - \pi_{(Q_7.\text{cc:cname}),(Q_7.\text{hq:headquarter})}(Q_7))$$

<u>Optimizations</u>

1. *Pushing selections over semi* − *joins for* Q_4
   a. $Q_{opt-4} = \pi_{C.\text{cname},\text{C.headquarter}}(\pi_{C.\text{cname}}(C) \ltimes \pi_{W.cname}(W))$
2. We can optimize the join $Q_1 \bowtie_{Q_1.cname = Q_4.cname} Q_4$ *by pushing selections over the joins*
   a. $Q_{opt-4} = \pi_{C.\text{cname},\text{C.headquarter},\text{W.pid}}\left(\pi_{C.\text{cname}}(C) \ltimes \pi_{W.cname}\left(\sigma_{P.city='Chicago'}(W)\right)\right)$
3. Relativized De Morgan for ∩
   a. $Q_{opt-7} = \pi_*\left(\pi_{Q_{opt-4}.pid,(Q_{opt-4}.cname:cc),(Q_{opt-4}.\text{headquarter:hq})}(Q_{opt-4}) - \left(\pi_{Q_{opt-4}.pid,(Q_{opt-4}.cname:cc),(Q_{opt-4}.headquarter:hq)}\left(Q_{opt-4} \bowtie_{Q_{opt-4}.pid = Q_2.pid} Q_2\right) \cap \pi_{Q_{opt-4}.pid,(Q_{opt-4}.cname:cc),(Q_{opt-4}.headquarter:hq)}\left(Q_{opt-4} \bowtie_{Q_{opt-4}.pid = Q_3.pid} Q_3\right)\right)\right)$
4. *Pushing selections over semi* − *joins for* Q_{opt-7}
   a. $Q_{opt-7} = \pi_*\left(\pi_{Q_{opt-4}.pid,(Q_{opt-4}.cname:cc),(Q_{opt-4}.\text{headquarter:hq})}(Q_{opt-4}) - \left(\pi_{Q_{opt-4}.pid,(Q_{opt-4}.cname:cc),(Q_{opt-4}.headquarter:hq)}\left(\pi_{Q_{opt-4}.pid}(Q_{opt-4}) \ltimes Q_2\right) \cap \pi_{Q_{opt-4}.pid,(Q_{opt-4}.cname:cc),(Q_{opt-4}.headquarter:hq)}\left(\pi_{Q_{opt-4}.pid}(Q_{opt-4}) \ltimes Q_3\right)\right)\right)$

The optimized query looks like

$$\pi_{Q_{opt-4}.cname,Q_{opt-4}.headquarter}(Q_4) - \pi_{(Q_{opt-7}.\text{cc:cname}),(Q_{opt-7}.\text{hq:headquarter})}(Q_{opt-7}))$$

Problem 10:Consider the following Pure SQL query.

```
select p.pid, exists (select 1
```

from hasManager hm1, hasManager hm2

where hm1.mid = p.pid and hm2.mid = p.pid andhm1.eid <> hm2.eid)

from Person p;

This query returns a pair (p, t) if p is the pid of a person who manages at least two persons and returns the pair (p, f) other- wise.[12]

→

Let

$Q_1 = \pi_{\text{P.pid,True}}(M_1 \bowtie_{M_1.eid \neq M_2.eid} M_2 \bowtie_{M_1.mid = P.pid \wedge M_2.mid = P.pid} P)$

And

$Q_2 = \pi_{P.pid}(P) - \pi_{Q_1.pid}(Q_1)$

Final query is

$\pi_{Q_2.pid,False}(Q_2) \cup Q_1$

<u>Optimizations</u>

1. Primary Key and Foreign Key Constraint
    a. We can optimize the Q_1 as
    b. $Q_{opt-1} = \pi_{M_1.\text{mid,True}}(M_1 \bowtie_{M_1.eid \neq M_2.eid \wedge M_1.mid = M_2.mid} M_2)$
2. $Pushing\ selections\ over\ semi-joins\ for\ Q_{opt-1}$
    a. $Q_{opt-1} = \pi_{M_1.\text{mid,True}}(\pi_{M_1.\text{mid}}(M_1) \ltimes \pi_{M_2.\text{mid}}(\sigma_{M_1.eid \neq M_2.eid}(M_2))$
3. Relativized tautology for ∪
    a. The final query can be optimized like
    b. $\pi_{P.pid,False}(P) \cup Q_{opt-1}$

Optimized Query is:

$$\pi_{P.pid,False}(P) \cup Q_{opt-1}$$