# 15-415 HW2 Solutions

October 1, 2009

## 1 Micro-Blogging and Relational Algebra

Let:

- Person = P
- Follows = F
- Tweets = T
- PersonTweets = PT
- TweetTag = TT

Q1.1 $\Pi_{\text{pname}}(\sigma_{\text{tagname}='\text{Obama}'\wedge \text{PT.tid}=\text{TT.tid}}(PT \times TT))$

Q1.2 $\Pi_{\text{tagname}}(TT)$

Q1.3 $\Pi_{\text{tagname}}(\sigma_{\text{pname}='\text{Bob Smith}'\wedge \text{PT.tid}=\text{TT.tid}}(PT \times TT))$

Q1.4 $\Pi_{\text{tagname}}(\sigma_{\text{pname1}='\text{Bob Smith}'\wedge \text{PT.tid}=\text{TT.tid}\wedge \text{F.pname2}=\text{PT.pname}}(F \times PT \times TT))$

Q1.5 $\text{PWriteTag} = \Pi_{\text{pname,tagname}}(\sigma_{\text{PT.tid}=\text{TT.tid}}(PT \times TT))$
$\text{PReadTag} = \Pi_{\text{pname,tagname}}(\sigma_{\text{PT.tid}=\text{TT.tid}\wedge \text{F.pname2}=\text{PT.pname}}(F \times PT \times TT))$

Full Query:

$\Pi_{\text{pname}}(P)\ -$

$\Pi_{\text{PReadTag.pname}}(\sigma_{\text{PReadTag.pname}=\text{PWriteTag.pname}\wedge \text{PReadTag.tagname}=\text{PWriteTag.tagname}}(\text{PReadTag}\times\text{PWriteTag}))$

Q1.6 $\Pi_{pname}(\sigma_{\text{city}='\text{Pittsburgh}'\wedge \text{tagname}='\text{G20}'}(P \bowtie PT \bowtie TT))$

Q1.7 $\text{AshtonFollowers} = \Pi_{\text{pname1}}(\sigma_{\text{pname2}='\text{Ashton Kutcher}'}(F))$
Full Query:

$$\Pi_{\text{pname,city}}(P \bowtie_{\text{pname1}=\text{pname}} (\text{AshtonFollowers} \bowtie_{\text{AshtonFollowers.pname1}=\text{pname2}} F))$$

Q1.8 $\text{IPhoneTweeters} = \Pi_{\text{PT.pname}}(\sigma_{(\text{T.title}='\text{New IPhone}'\wedge \text{TT.tagname}='\text{IPhone 3GS}')\wedge \text{PT.ts}<'\text{00:00:00 06-19-2009}'}(T \bowtie PT \bowtie TT))$

Full Query:

$$\Pi_{\text{pname1}}(\text{IPhoneTweeters} \bowtie_{\text{pname2}=\text{pname}} F)$$

Q1.9 $\Pi_{\text{pname,city}}(P \bowtie_{\text{pname1}=\text{pname}} (F \div \Pi_{\text{pname2}}(\sigma_{\text{pname1}='\text{Bob Smith}'}(F))))$

Q1.10 Can not be answered using relational algebra. Relational algebra doesn't provide a way of unraveling the recursion involved.

Q1.11 $\Pi_{\text{F.pname2,F1.pname2}}(\sigma_{\text{F.pname1}=\text{F1.pname1}}(F \times \rho_{F1}(F)))$
(Yes - the above query will have mirror and self pairs!)

## 2 Relational Calculus

Q1.2-RTC $\{t|\exists x \in \text{TT}(\text{t.tagname} = \text{x.tagname})\}$

Q1.4-RTC $\{t|\exists tt \in TT \wedge \exists f \in F \wedge \exists pt \in PT(\text{t.tagname} = \text{tt.tagname} \wedge \text{f.pname1} =' \text{Bob Smith}' \wedge \text{pt.pname} = \text{f.pname2} \wedge \text{pt.tid} = \text{tt.tid})\}$

Q1.7-RTC $\{t|\exists f1 \in F \wedge \exists f2 \in F \wedge \exists p \in P(\text{f1.pname2} = \text{f2.pname1} \wedge \text{f2.pname2} =' \text{Ashton Kutcher}' \wedge \text{f1.pname1} = \text{p.pname} \wedge \text{p.pname} = \text{t.pname} \wedge \text{t.city} = \text{p.city})\}$

Q1.5-RDC $\{< \text{n} > | \; \exists i \exists i2 \exists ts \exists ts2 \exists n2 \forall t(< \text{n, i, ts} > \in PT \wedge < \text{n, n2} > \in F \wedge < \text{n2, i2, ts2} > \in PT \wedge (< \text{i, t} > \in TT \Rightarrow < \text{i2, t} > \notin \text{TT}))\}$

Q1.8-RDC $\{< \text{n} > | \; \exists n2 \exists ts \exists ti \exists txt \exists i \exists tag(< \text{n, n2} > \in F \wedge < \text{n2, i, ts} > \in PT \wedge < \text{i, ti, txt} > \in T \wedge < \text{i, tag} > \in TT \wedge ts < '00:00:00\ 06-19-2009' \wedge (\text{tag} = \text{`IPhone 3GS'} \vee \text{ti} = \text{`New IPhone'}))\}$

Q1.9-RDC $\{< \text{n, c} > | \; \forall s(\exists n2(<' \text{BobSmith}', \text{n2} > \in F \Rightarrow < \text{n, n2} > \in F) \wedge < \text{n, c, s} > \in P)\}$

## 3 Perfect Matches

### 3.1 Warm up

NON-EXISTENT-PAIRS = $(\Pi_{pid}(\text{Hobbies}) \times \Pi_{hobby}(\text{Hobbies})) - \text{Hobbies}$
PPH = $\Pi_{\text{pid1,pid2,hobby}}(\text{Hobbies} \bowtie_{\text{Hobbies.hobby=NON-EXISTENT-PAIRS.hobby}} \text{NON} - \text{EXISTENT} - \text{PAIRS})$

### 3.2 Full Query

REVERSE-PPH = $\Pi_{\text{pid1,pid2,hobby}}(\text{NON} - \text{EXISTENT} - \text{PAIRS} \bowtie_{\text{Hobbies.hobby=NON-EXISTENT-PAIRS.hobby}} \text{Hobbies})$

FULL-QUERY
$\sigma_{\text{pid1<pid2}}(\Pi_{\text{Hobbies.pid,H2.pid}}(\text{Hobbies} \bowtie_{\text{Hobbies.hobby=H2.hobby}} \rho_{H2}(\text{Hobbies})) - (\Pi_{\text{pid1,pid2}}(\text{PPH}) \cup \Pi_{\text{pid1,pid2}}(\text{REVERSE} - \text{PPH})))$

**Common Mistake** Students didn't subtract REVERSE-PPH, only PPH. You will still get the correct output for the sample db even if you don't, but it will fail on other test databases.

### 3.3 Verification with SQL

**Q3.3**

```
select * from
(select TT.pid as pid1, hobbies.pid as pid2 from (select * from hobbies) TT, hobbies where
TT.hobby = hobbies.hobby
except
select distinct * from
(select NEP.pid, hobbies.pid from (select * from (select pid
from hobbies) T1, (select hobby from hobbies) T2 except select * from hobbies)
NEP, hobbies where NEP.hobby = hobbies.hobby
union
select hobbies.pid, NEP.pid from hobbies, (select * from (select pid
from hobbies) T1, (select hobby from hobbies) T2 except select * from hobbies)
NEP where NEP.hobby = hobbies.hobby)) ANS
where ANS.pid1 < ANS.pid2;
```

**Q3.4**

| pid1 | pid2 |
|---|---|
| 2 | 3 |
| 21 | 23 |
| 21 | 25 |
| 23 | 25 |
| 33 | 34 |