NYU School of Engineering October 5, 2016
Computer Science and Engineering
CS 6086, Fall 2016

Problem Set #1 – Sample Solutions

Problem 1:

(a)
(pid, mid) is a foreign key from PURCHASE to OFFER.
(cid, pid, mid) is a foreign key in the PURCHASE table.

(b)
PURCHASE has (cid, pid, mid, putime) attributes as primary key because more than one purchase can be made with the by same customer with the same merchant but no two purchases can be made at the exact same time at the same merchant for same product.
If we remove mid from the key, then we will not be able to differentiate two purchases for same product, at same price by a customer. For example, Customer1 buys a $10 shirt at 2016-10-2 10:00:00, with other merchant Customer1 buys same shirt at same time. Since mid is not in key such scenarios will create ambiguity and conflict in the database.
If we remove putime from the key then we cannot between a customer buying same item, from same merchant at different times. For example, Customer1 buying a $10 shirt from Merchant1 at 10am and same customer buying same item at same price at 11am, if putime is not included this will create conflict.

(c)
I.

```
Select c.cid, c.cname
From customer c
Join purchase p on p.cid = c.cid
Where p.price > 50;
```

II.

```
Select c.cid, c.cname
From customer c
Join purchase p on c.cid=p.cid
Join merchant m on m.mid=p.mid
where c.cstate=m.mstate
```

III.

```
Select m.mid, sum(p.puprice)
From merchant m
Left join purchase on m.mid=p.mid
where p.putime is between ‘2015-01-01’ and ‘2015-12-31’
Group by m.mid;
```

IV.

```
Select p.cid
From purchase p
Where p.puprice in (Select max(p.puprice) From purchase);
```

V.

```
Select cid
From (Select cid, count(*)
       From Purchase
       Goup by cid)
where count(*)= (Select max(a.ct)
                  From (Select cid, count(*) as ct
                         From Purchase
                         Group by cid) as a);
```

VI.

```
Select sum(p.puprice), m.mstate
From merchant m
Join purchase p on p.mid=m.mid
Where m.mname=”ABC Sneakers”
Group by m.mstate;
```

VII.

```
Select m.mid, m.mname
From merchant m
Join purchase p on p.mid=m.mid
Join customer c on p.cid=c.cid
Group by p.mid
Having count(distinct c.cstate)=(select count(distinct cstate) from customer);
```

(d)

I.

$\prod_{\text{cid, cname}} (\sigma_{\text{PURCHASE.puprice>50}} (\text{CUSTOMER} \bowtie \text{PURCHASE}))$

II.

$\prod_{\text{cid, cname}} (\sigma_{\text{CUSTOMER.cstate=MERCHANT.mstate}} (\text{CUSTOMER} \bowtie \text{MERCHANT} \bowtie \text{PURCHASE}))$

III.

$\prod_{\text{mid, sum(puprice)}} ({}_{\text{mid}} G_{\text{ sum(puprice)}}(\sigma_{\text{year(PURCHASE.putime)=2015}}(\text{MERCHANT} ⟕ \text{PURCHASE})))$

IV.

$\prod_{\text{pid}} (\sigma_{\text{cid, puprice = max(puprice) (PURCHASE)}} (\text{PURCHASE}))$

V.

$\text{D} \leftarrow G_{\text{max(c) as m}}({}_{\text{cid}}G_{\text{count(cid) as c}} (\text{PURCHASE})))$

$\prod_{\text{cid}} (\sigma_{\text{c=D}}({}_{\text{cid}}G_{\text{count(cid) as c}} (\text{PURCHASE})$

VI.

$\prod_{\text{sum(puprice), mstate}} (\sigma_{\text{mname="ABC Sneakers"}} (G_{\text{mstate}} (\text{MERCHANT} \bowtie \text{PURCHASE})))$

VII.

$\prod_{\text{mid, mname}} (\sigma_{\text{count(CUSTOMER.cstate)=count(MERCHANT.mstate)}} G_{\text{mid}} (\text{PURCHASE} \bowtie \text{MERCHANT} \bowtie \text{CUSTOMER}))$

(e)

I. {<cid,cname> | ∃ccity,cstate (<cid,cname,ccity,cstate> ∈ CUSTOMER) ∧ ∃pid,mid,puprice,putime (<cid,pid,mid,puprice,putime> ∈ PURCHASE ∧ puprice>50))}

II. {<cid,cname>| ∃c(<cid,cname,ccity,cstate>∈CUSTOMER ∧ ∃m(mid,mname,mcity,mstate)∈MERCHANT ∧ c=m))}

III. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are no aggregate functions.

IV. $\{t.\text{cid} \mid t \in \text{PURCHASE}\ \neg(\exists s \in \text{PURCHASE}\ (t.\text{puprice} > s.\text{puprice}))\}$

V. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are no aggregate functions.

VI. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are no aggregate functions.

VII. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are no aggregate functions.

Problem 2 :

(a)

There are many possible approaches for designing the schema for this scenario, we have provided one of them:

USERS (uid, username, firstname, lastname, email, city, state, country)
PROPERTY (pid, uid, title, description, fullAddress, city, state, country, capacity)
RESERVATION (pid, uid, status, startdate, enddate, rating, review, total_cost)
AVAILABLE (pid, date, status, price, cleaningfee, agencyfee)

In USERS, uid is primary key.
In PROPERTY, pid is primary key. uid is a foreign key referencing uid(USERS)
In RESERVATION, (pid,uid) is primary key. pid is a foreign key referencing pid(PROPERTY) and uid is a foreign key referencing uid(USERS)
In AVAILABLE, (pid,date) is primary key. pid is a foreign key referencing pid(PROPERTY). status has 3 possible value: available, unavailable, rented.

(b)

I.

```
select distinct P.ID, P.title, (sum(A.price) + A.clean_fee) * 1.1
from PROPERTY as P, AVAILABLE as A
where P.city = “Chicago” and
not exist (select * from available as C
        where (C.status = “unavailable” or C.status = “rented out”) and (C.date >=
        ‘2017-04-27’ and C.date <= ‘2017-05-06’) and C.PID = P.ID and A.PID = C.PID);
```

II.

```
select P.pid, X.c/Y.c as ratio
from
(select pid, count(*) as c
from AVAILABLE
where date>=‘2015-01-01’ and date<=‘2015-12-31’ and status = ‘rented’
group by pid) as X,
(select pid, count(*) as c
from AVAILABLE
where date>=‘2015-01-01’ and date<=‘2015-12-31’ and (status = ‘rented’ or status =‘available’)
group by pid) as Y
where P.pid = X.pid and P.pid=Y.pid;
```

III.

```
select r.pid as propertyid, avg(r.rating)
from reservation r
group by r.pid
having count(r.rating)>=5;
```

IV.

```
select P.city, sum(R.total_cost)
from Reservation as R, Property as P
where R.ID = P.ID
group by P.Pcity
```