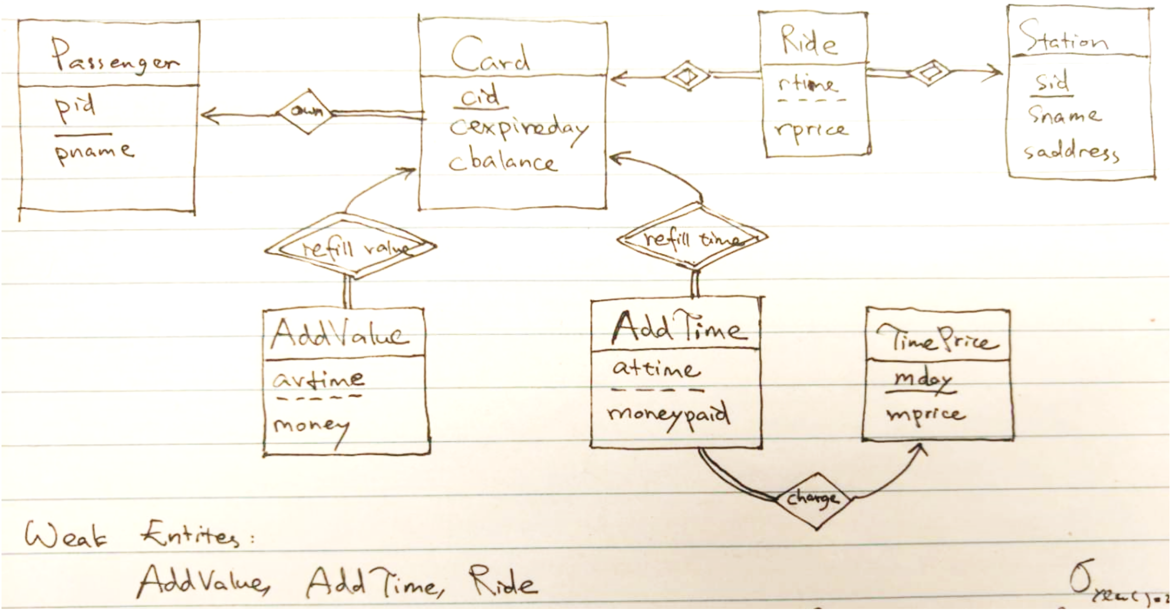
NYU Tandon School of Engineering **Mar 29, 2019**
Computer Science and Engineering
CS 6083, Spring 2019

Problem Set #2 Sample Solution

Problem 1

(a)



(b) Answer:

- Card.pid is a foreign key referencing Passenger.pid
- AddTime.cid, AddValue.cid and Ride.cid are foreign keys referencing Card.cid
- AddValue.mday is a foreign key referencing TimePrice.mday
- Ride.sid is a foreign key referencing Station.sid

(c) Answer:

Passenger (pid, pname);
Ownership (pid, cid);
Card (cid, cexpireday, cbalance);
TimePrice (mday, mprice);
Station (sid, sname, saddress);
AddTime (cid, attime, mday, moneypaid);
AddValue (cid, avtime, money);
Ride (cid, sid, rtime, rprice);

(d) Answer:

Passenger (pid, pname);
Card (cid, pid, cexpireday, cbalance);
TimePrice (mday, mprice);
Station (sid, sname, saddress);
AddTime (cid, attime, mday, moneypaid);
AddValue (cid, avtime, money);
Pricetable (departuresid, arrivalsid, pprice)
Ride (cid, departuresid, arrivalsid, departuretime, arrivaltime);

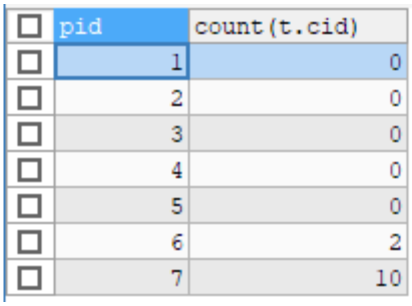
Pricetable.departuresid and Pricetable.arrivalsid are foreign keys referencing Station.sid;
Ride.(departuresid, arrivalsid) is a foreign key referencing Pricetable. (departuresid, arrivalsid).

(e) -------

(f)

(i)

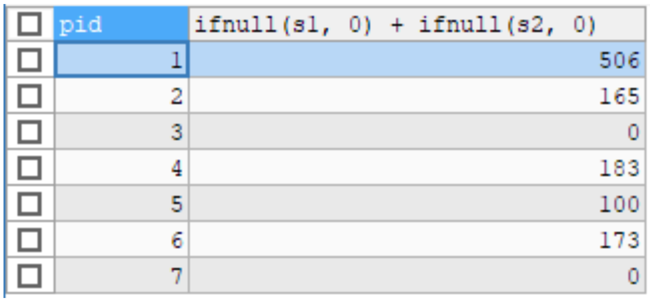
```
select p.pid, count(*)

from Passenger p left outer join (

        select pid, cid, sid

        from Card natural join Ride

        where date(rtime) = '2019-12-25') as t on p.pid = t.pid

group by p.pid
```



☐	pid	count(t.cid)
☐	1	0
☐	2	0
☐	3	0
☐	4	0
☐	5	0
☐	6	2
☐	7	10

(ii)

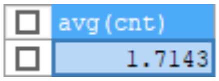
```
create view v1 as
        select p.pid, sum(t.moneypaid) as s1
        from Passenger p left outer join(
                select *
                from Card natural join AddTime
                where year(attime) = '2018') as t on p.pid = t.pid
        group by p.pid;
create view v2 as
        select p.pid, sum(t.money) as s2
        from Passenger p left outer join(
                select *
                from Card natural join AddValue
                where year(avtime) = '2018') as t on p.pid = t.pid
        group by p.pid;
select pid, ifnull(s1, 0) + ifnull(s2, 0)
from v1 natural join v2;
```



□	pid	ifnull(s1, 0) + ifnull(s2, 0)
□	1	506
□	2	165
□	3	0
□	4	183
□	5	100
□	6	173
□	7	0

(iii)

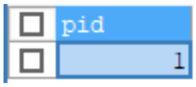
```
select avg(cnt)
from (select count(c.cid) as cnt
         from Passenger p left outer join Card c on p.pid = c.pid
         group by p.pid) as t
```



□	avg(cnt)
□	1.7143

(iv)

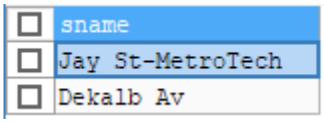
```
select distinct pid

from Card natural join Ride

group by pid, date(rtime)

having count(*) > 10
```



□	pid
□	1

(v)

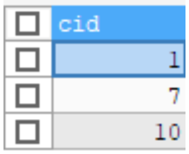
```
create view max_cnt as

            select count(cid) as cnt

            from Ride

            where year(rtime) = '2018'

            group by sid

            order by count(cid) desc

            limit 1;

select sname

from Station s natural join Ride r natural join max_cnt

where year(rtime) = '2018'

group by sid, cnt

having count(r.cid) = max_cnt.cnt;
```



□	sname
□	Jay St-MetroTech
□	Dekalb Av

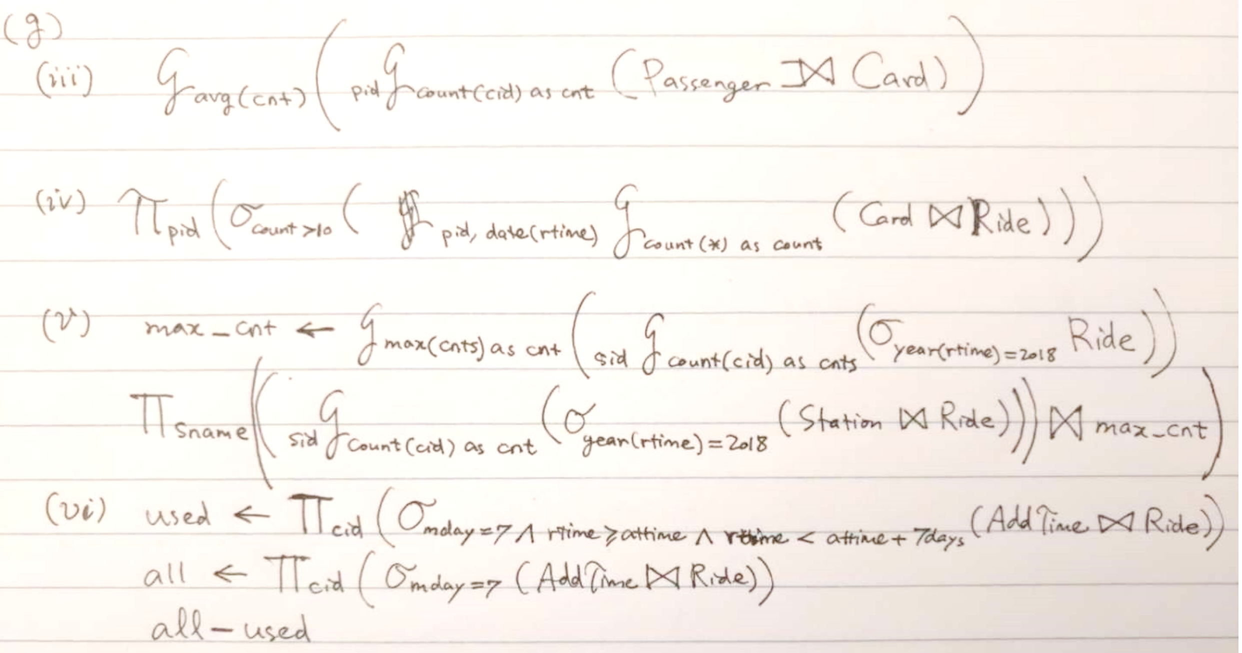
(vi)

```
select distinct cid
from AddTime natural join Ride
where mday = 7 and cid not in (select cid
                              from AddTime natural join Ride
                              where mday = 7 and
                              rtime >= attime and
                              rtime < attime + interval '7' day)
```



□	cid
□	1
□	7
□	10

(g)



(g)

(iii) $\mathcal{G}_{avg(cnt)}\left({}_{pid}\mathcal{G}_{count(cid)\ as\ cnt}(Passenger \bowtie Card)\right)$

(iv) $\Pi_{pid}\left(\sigma_{count>10}\left(\cancel{\mathcal{G}}\ {}_{pid,\ date(rtime)}\mathcal{G}_{count(*)\ as\ count}(Card \bowtie Ride)\right)\right)$

(v) $max\_cnt \leftarrow \mathcal{G}_{max(cnts)\ as\ cnt}\left({}_{sid}\mathcal{G}_{count(cid)\ as\ cnts}(\sigma_{year(rtime)=2018} Ride)\right)$

$\Pi_{sname}\left(\left({}_{sid}\mathcal{G}_{count(cid)\ as\ cnt}\left(\sigma_{year(rtime)=2018}(Station \bowtie Ride)\right)\right) \bowtie max\_cnt\right)$

(vi) $used \leftarrow \Pi_{cid}\left(\sigma_{mday=7 \wedge rtime \geq attime \wedge rtime < attime+7days}(AddTime \bowtie Ride)\right)$

$all \leftarrow \Pi_{cid}\left(\sigma_{mday=7}(AddTime \bowtie Ride)\right)$

$all - used$

(h)

(i)

```
update Card
        set cexpireday = timestampadd(day, 1, cexpireday)
        where datediff(cexpireday, now()) >= 10
```

(ii)

```
delete from Card
        where cbalance = 0 and cexpireday < now()
```

(iii)

```
insert into AddTime
        select 1, now(), mday, mprice
        from TimePrice
        where mday = 30;
update Card
        set cexpireday = date_add(curdate(), INTERVAL 30 day)
```

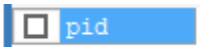
Problem 2

(a)

```
create view ValueOnlyPassengerRide as
select pid, cid, rtime as ttime, rprice as tprice
from Card natural join Ride
where year(rtime) = 2018 and pid not in (
  select distinct pid
  from Card natural join AddTime
  where year(attime) = 2018)
```

(i)

```
select pid
from
    (select mprice
    from TimePrice
    where mday = 30) as a
join
    (select pid, sum(tprice) as actualprice
    from ValueOnlyPassengerRide
    where month(ttime) = 6
    group by pid) as b
where mprice < actualprice
```



```
□ pid
```

(ii)

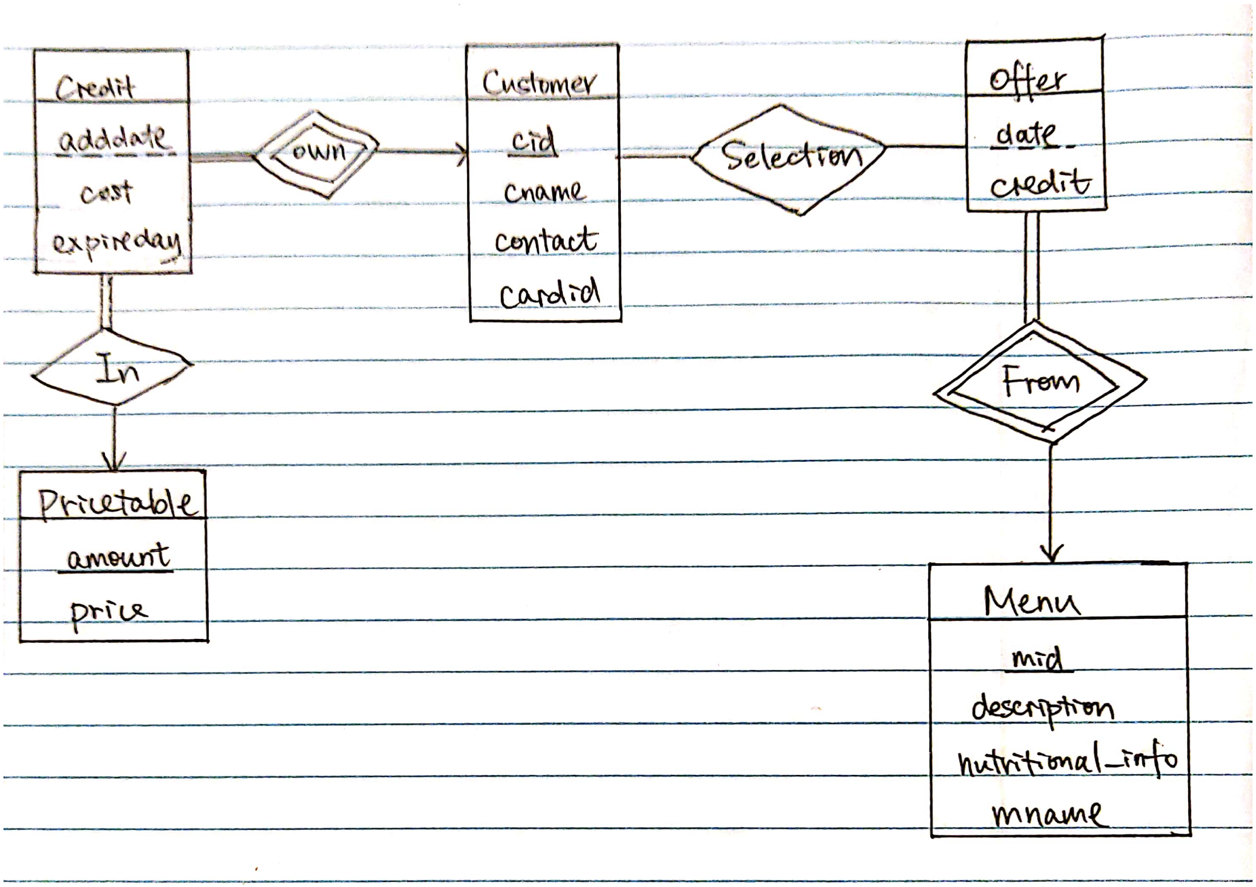
For those passengers who never had an unlimited ride card during 2018, output the distribution of their total cost on month.

```
select month(ttime), sum(tprice)
from ValueOnlyPassengerRide
group by month(ttime)
```

Problem 3

(a) Assumptions:

- Every selection's mid is from the mid that day offer.
- Expireday in Credit will always be the last day of a month.
- A person can only order the same meal once per day.



(b)

Customer (cid, cname, contact, cardid)

Credit (cid, adddate, amount, cost, expireday)

Menu (mid, description, nutritional_info, mname)

Offer (mid, date, credit)

Pricetable (amount, price)

Selection (cid, mid, date)

Foreign keys:

- Credit.cid is a foreign key referencing Customer.cid
- Credit.amount is a foreign key referencing Pricetable.amount
- Offer.mid is a foreign key referencing Menu.mid
- Selection.cid is a foreign key referencing Customer.cid
- Selection.(mid, date) is a foreign key referencing Offer.(mid, date)

(c) (i)

```
create view v1 as
        select mid, count(date) as cnt
        from Offer
        where year(date) = “2018”
        group by mid;
create view v2 as
        select mid, ifnull(v1.cnt, 0) as times_offered
        from Menu left outer join v1 on Menu.mid = v1.mid
create view v3 as
        select mid, count(distinct date) as cnt
        from Selection
        where year(date) = “2018”
        group by mid;
select v2.mid, v2.times_offered, ifnull(v3.cnt, 0) as times_pickup
from v2 left outer join v3 on v2.mid = v3.mid
```

(ii)

```
create view v1
        select cid, sum(amount) as total_credit
        from Credit
        where year(expireday)  = “2018”
        group by cid;
create view v2
        select cid, sum(credit) as used_credit
```

```
                from Selection natural join Offer
                where year(Selection.date) = “2018”
                group by cid;
        select v1.cid
        from v1 left outer join v2 on v1.cid = v2.cid
        where total_credit * 0.8 > used_credit or used_credit is null


(iii)   create view v0
                select mid, date, count(*) as c
                from Selection
                where year(date) = “2018”
                group by mid, date
        create view v
                select v0.mid, v0.date, ifnull(v0.c, 0) as cnt
                from Offer o left outer join v0 on o.mid = v0.mid and o.date = v0.date
        select t1_1.mid as midA, t1_2.mid as midB
        from v v1, v v2
        where v1.mid != v1.mid and v1.date = v2.date and v2.count >= v1.count * 2
```