**NYU Tandon School of Engineering** **March 3, 2017**
**Computer Science and Engineering**
**CS6083, Spring 2017**

# Problem Set #1

**Problem 1:**

a. MENUITEM.rid is foreign key referencing RESTAURANT.rid
   ORDER.cid is foreign key referencing CUSTOMER.cid
   ORDER.rid is foreign key referencing RESTAURANT.rid
b. ORDERDETAIL.oid is foreign key referencing ORDER.oid
   ORDERDETAIL.(rid, itemname) is foreign key referencing MENUITEM.(rid,itemname)

c. (rid, itemname) is primary key because different restaurant can have same itemnames so (rid, itemname) combination has to be unique every time. For example, a restaurant with rid=1 has (1, 'Burger', 20), (1, 'Pizza, 30) while another restaurant with rid=2 can also contain 'Burger': (1, 'Burger', 20), (2, 'Burger', 15).

   If we remove rid from MENUITEM key then we cannot have repeated 'itemname' for any restaurant. For example, (1, 'Pizza', 20), (2, 'Pizza', 20) is not possible with rid removed and since itemname is primary key- just one restaurant can have 'Pizza' in entire table. Similarly, table can have every item occurring just once this will not be practical and will create conflict in the database.

   It's okay to remove rid from ORDERDETAIL key (rid still in the table), because one order could only done by one restaurant and number of same item in one order would show in quantity, thus oid and itemname could determine a tuple. But we can't remove rid from the table. If so, we lost the foreign key constraint (rid, itemname) referencing MENUITEM. In this case, it's possible to insert an order with the item that the restaurant don't offer, which is invalid.

d.
   i. Select distinct cid, cname
      From CUSTOMER natural join ORDER
      Where totalprice>50;

ii.
```
Select distinct cid, cname
From CUSTOMER natural join RESTAURANT natural join ORDER
Where ccity=”Queens” and rcity=”Brooklyn”;
```

iii.
```
Select r.rid, r.rname
From RESTAURANT r
Where r.rid not in
(Select o.rid from ORDER o where Hour(o.deliverytime-o.ordertime) > 1);
```

iv.
```
Select cid, cname, orders
From CUSTOMER natural join
(select cid, count(oid) as orders from ORDER group by cid);
```

v.
```
Select distinct cid, cname
From CUSTOMER natural join
(select cid from ORDER where totalprice in (select max(totalprice) from
ORDER));
```

vi.
```
Select oid, totalprice as paid_price, current_price
from ORDER natural join (select oid, sum(price*quantity) as current_price
from ORDERDETAIL natural join MENUITEM group by oid);
```

vii.
```
Select cid, cname
from CUSTOMER natural join
        (select cid, count(oid) as orders from ORDER natural join
                (select  rid from RESTAURANT where rname=”LittleSheep”)
        group by cid having orders > 1);
```

viii.
```
With CUSTOMER_ORDERS as
(select  cid, cname, cstate, orders from CUSTOMER natural join
(select cid, count (*) as orders from ORDER group by cid))

Select cstate, cid, cname
from CUSTOMER_ORDERS natural join (select cstate, max(orders) as
maxorders from CUSTOMER_ORDERS group by cstate);
Where orders = maxorders;
```

e.

i. $\Pi_{\text{distinct cid,cname}}(\sigma_{\text{totalprice>50}}(\text{CUSTOMER} \bowtie \text{ORDER}))$

ii. $\Pi_{\text{distinct cid, cname}}(\sigma_{\text{ccity="Queens" and rcity="Brooklyn"}}(\text{CUSTOMER} \bowtie \text{ORDER} \bowtie \text{RESTAURANT}))$

iii. $T \leftarrow \Pi_{\text{rid}}(\sigma_{\text{deliverytime-ordertime<=1}}(\text{ORDER}))$

$\Pi_{\text{r.rid, r.rname}}(\sigma_{\text{rid=T}}(\text{RESTAURANT}))$

iv. $\Pi_{\text{c.cid, c.cname, count(oid)}}({}_{\text{cid, cname}}G(\text{CUSTOMER} \bowtie \text{ORDER}))$

v. $\Pi_{\text{c.cid, c.cname}}(\sigma_{\text{max(totalprice)}}(\text{CUSTOMER} \bowtie \text{ORDER}))$

vi. $\Pi_{\text{oid, totalprice as paid_price, current_price}}(\text{ORDER} \bowtie ({}_{\text{oid}}G_{\text{sum(price*quantity) as current_price}}(\text{ORDERDETAIL} \bowtie \text{MENUITEM})))$

vii. $\Pi_{\text{cid, cname}}(\text{CUSTOMER} \bowtie (\sigma_{\text{orders>1}}({}_{\text{cid}}G_{\text{count(cid) as orders}}(\text{ORDER})) \bowtie (\sigma_{\text{rname="Little Sheep"}}(\text{RESTAURANT}))))$

viii. $\text{CUSTOMER\_ORDERS} \leftarrow \Pi_{\text{cid, cname, cstate, orders}}(\text{CUSTOMER} \bowtie ({}_{\text{cid}}G_{\text{count(cid) as orders}}(\text{ORDER}))$

$\Pi_{\text{cid, cname}}(\text{CUSTOMER\_ORDERS} \bowtie {}_{\text{cstate}}G_{\text{max(orders)}}(\text{CUSTOMER\_ORDERS}))$

f.

i. $\{< \text{cid, cname} > \mid \exists < \text{ccity, cstate, cphone} > (< \text{cid, cname, ccity, cstate, cphone} > \in \text{CUSTOMER} \wedge \exists < \text{oid, rid, ordertime, deliverytime, totalprice, deliveryfee} > (< \text{oid, cid, rid, ordertime, deliverytime, totalprice, deliveryfee} > \in \text{ORDER} \wedge \text{totalprice} > 50))\}$

ii. $\{< \text{cid, cname} > \exists < \text{ccity, cstate, cphone} > (< \text{cid, cname, ccity, cstate, cphone} > \in \text{CUSTOMER} \wedge \text{ccity} = \text{"Queens"} \wedge \exists < \text{oid, rid, ordertime, deliverytime, totalprice, deliveryfee} > (< \text{oid, cid, rid, ordertime, deliverytime, totalprice, deliveryfee} > \in \text{ORDER} \wedge \exists < \text{rname, rcity, rstate} > (< \text{rid, rname, rcity, rstate} > \in \text{RESTAURANT} \wedge \text{rcity} = \text{"Brooklyn"})))\}$

iii. {< rid, rname > | ∃ < rcity, rstate > (< rid, rname, rcity, rstate > ∈ RESTAURANT ∧ ∃ < oid, cid, ordertime, deliverytime, totalprice, deliveryfee > (< oid, cid, rid, ordertime, deliverytime, totalprice, deliveryfee > ∈ ORDER ∧ deliverytime − ordertime <= 1))}

iv. Cannot express in TRC or DRC.

v.
{<cid, cname> | ∃ <ccity, cstate, cphone> ((<cid, cname, ccity, cstate, cphone> ∈ CUSTOMER) ^ ∃ <oid, rid, ordertime, deliverytime, totalprice, deliveryfee> ((<oid, cid, rid, ordertime, deliverytime, totalprice, deliveryfee> ∈ ORDER)
=> ∀ <oid2, cid2, rid2, ordertime2, deliverytime2, totalprice2, deliveryfee2> ((<oid2, cid2, rid2, ordertime2, deliverytime2, totalprice2, deliveryfee2> ∈ ORDER) ^ totalprice >= totalprice2)))

vi. Cannot express in TRC or DRC.

vii.
{<cid, cname> | ∃ <ccity, cstate, cphone> ((<cid, cname, ccity, cstate, cphone> ∈ CUSTOMER)
^ ∃ <oid, rid, ordertime, deliverytime, totalprice, deliveryfee> ((<oid, cid, rid, ordertime, deliverytime, totalprice,
deliveryfee> ∈ ORDER)^ ∃ rname ((<rid, rname, ccity, cstate> ∈ RESTAURANT) ^ rname = ‘Little Sheep’ ^ ∃ <oid2, ordertime2, deliverytime2, totalprice2, deliveryfee2 > (<oid2, cid, rid, ordertime2, deliverytime2, totalprice2, deliveryfee2> ∈ ORDER ^ oid != oid2))))

viii. Cannot express in TRC or DRC.

**Problem 2:**

a. PATIENT(pid, pname, page, pgender, height, weight, paddress)

STAFF(sid, sname, sage, sgender, saddress, title, hid)

HOSPITAL(hid, hname, haddress)

STAY(pid, hid, admitted_time, released_time,attending)

TREATMENT(pid,admitted_time,sid)

DISEASEHISTORY(pid, dname,date_first_diagnosed, cured,curetime)

DISEASE(dname, description, contagious)

Primary keys are underlined in above schema.
Foreign keys:

- STAFF.hid is a foreign key referencing HOSPITAL.hid
- STAY.pid is a foreign key referencing PATIENT.pid
- STAY.hid is a foreign key referencing HOSPITAL.hid
- STAY.attending is a foreign key referencing STAFF.sid
- (TREATMENT.pid,TREATMENT.admitted_time) is a foreign key referencing (STAY.pid,STAY.admitted_time)
- TREATMENT.sid is a foreign key referencing STAFF.sid
- DISEASEHISTORY.pid is a foreign key referencing PATIENT.pid
- DISEASEHISTORY.dname is a foreign key referencing DISEASE.dname

b.

i. Select p.pname
From patient p, hospital h, stay s
Where s.hid = h.hid and p.pid = s.pid
and h.hname = “St. Rudolph’s Hospital” and s.admitted_time like “%2016”;

ii. Select D.sname, count(distinct st.pid) as num_patients
From STAY as N
right join (select * from staff s where s.title = 'doctor') as D on N.sid = D.sid
Group by (D.sid,D.sname);

iii.
```
With CONTAGIOUS_PATIENTS as
Select s.pid, s.attending from STAY s, DISEASE d, DISEASEHISTORY dh
Where s.pid= dh.pid and dh.dname = d.dname and d.contagious = 'yes';

Select s.sname
From CONTAGIOUS_PATIENTS c, staff s where s.attending = c.sid
Group by c.sid
Having count(distinct pid) > 10);
```

iv.
```
With BMI_Details as
Select p.pid, (p.weight/(p.height*p.height)) as BMI, dh.did
From PATIENT p, DISEASEHISTORY dh
Where p.pid = dh.pid and p.page >= 18 and cure =’NO’

Select avg(BMI) ,dname
From BMI_Details bd, disease d
Where d.dname = BD.did
Group by bd.dname;
```