Question 1

(a) Identify suitable foreign keys for this schema :

In MENUITEM rid is foreign key which we are referring from RESTAURANT table.
In ORDER table cid and rid are foreign keys which we are referencing from CUSTOMER and RESTAURANT table.
In ORDERDETAIL table oid, rid and itemname are foreign keys referencing from ORDER and MENUITEM table.

(b) In the MENUITEM table, why is (rid, itemname) the primary key? What happens if we remove rid from the key? What happens if we remove rid from the key for ORDERDETAIL?

In MENUITEM table both the attributes i.e rid and itemname are important and should be added as a primary key in MENUITEM. Since there are multiple restaurants available more than one restaurant can have same itemname with different prices thus only itemname is not enough to correctly identify the any menu in MENUITEM.
If we remove rid from MENUITEM it wont be able to correctly identify which menuitem we are referring to.

ORDERDETAIL gives details about which order contains which item , ordered from which restaurant and to which quantity. Since our assumption states that single order can only contain menu items from a single restaurant so even if we remove rid from ORDERDETAIL we will be correctly identify details of any order placed. Even if we want to find from which restaurant order is placed we have oid through which we can traverse in ORDER table and can find the restaurant. However, it is required if we want to check that a particular restaurant actually offers the item, this cane be done via a foreign key integrity constraint, which is where it is important to have both rid and itemname in the ORDERDETAIL. Thus we cannot remove rid from ORDERDETAIL.

(c) Write statements in SQL for the following queries.

1. Output the cid and cname of any customer who has placed an order for more than $50.

```
Select Distinct c.cid, c.cname
From CUSTOMER as c , ORDER as o
Where c.cid=o.cid and
o.totalprice > 50;
```

2. Output the cid and cname for any customer living in “Queens” who has placed an order from a restaurant in “Brooklyn”.

```
Select Distinct c.cid, c.cname
From CUSTOMER as c, ORDER as o, RESTAURANT r
Where c.cid=o.cid And
r.rid=o.cid And
c.ccity=’Queens’ And
r.rcity=’Brooklyn’;
```

3. Output the rid and rname of any restaurant that has delivered every order within 1 hour.

```
Select Distinct r.rid,r.name
From RESTAURANT as r, ORDER as o
Where r.rid=o.rid And
o.deliverytime < DATE_ADD(o.ordertime,INTERVAL 1 HOUR);
```

4. For each customer, output the cid, cname, and number of orders she has placed.

```
Select o.cid, c.cname, count(*) as no_of_order
From CUSTOMER as c , ORDER as o
Where c.cid=o.cid
Group By o.cid;
```

5. Output the cid and cname of the customer(s) who has placed the most expensive order ever.

```
Select c.cid,c.cname
From CUSTOMER as c, ORDER as o
Where c.cid = o.cid And
 o.totalprice =(Select max(totalprice) from ORDER);
```

6. Output for each order the oid, the price that was paid, and the price that the order would now cost if we sum up the current prices of all the items in the order.

```
Select o.oid,o.totalprice,sum(m.price*od.quantity)
From MENUITEM as m, ORDER as o, ORDERDETAIL as od
Where o.oid=od.oid And
m.rid=od.rid And
m.itemname=od.itemname
Group By o.oid;
```

7. Output the cid and cname for any customer who has placed more than one order from a restaurant whose rname is “Little Sheep”.

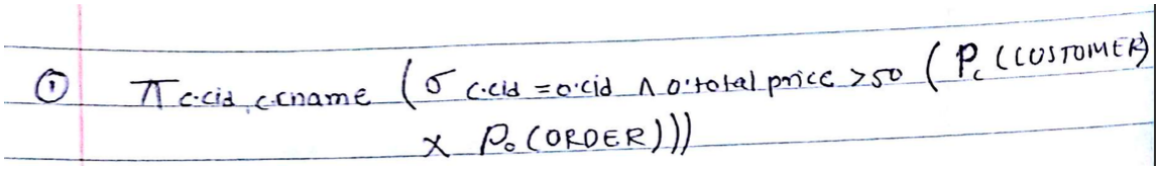
```
Select c.cid,c.cname, count(o.cid) as counter
From CUSTOMER as c, ORDER as o, RESTAURANT as r
Where c.cid=o.cid And
r.rid=o.rid And
r.rname='Little Sheep'
Group By c.cid
Having counter>1;
```

8. For each state, output the cid and cname of the customer(s) living in this state who has placed the most orders.

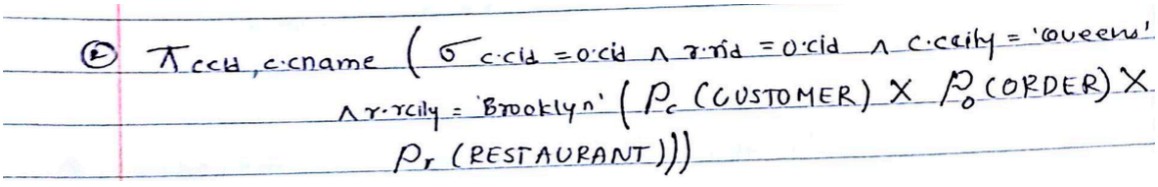
```
Select x.cid, x.cname
 From (Select c.cid, c.cname, c.cstate, count(o.oid) as no_of_orders
          From CUSTOMER as c, ORDER as o
          Where c.cid = o.cid
          Group By c.cid) as x ,
         (Select cstate, max(no_of_orders) as no_of_orders
          From (Select c.cid, c.cname, c.cstate, count(o.oid) as no_of_orders
                   From CUSTOMER as c, ORDER as o
                   Where c.cid = o.cid
                   Group By c.cid) as y
          Group By y.cstate) as y
Where x.cstate = y.cstate And
 x.no_of_orders = y.no_of_orders;
```

(d) Write expressions in Relational Algebra for the above queries.

1.



2.



3.

③ $\Pi_{r.rid, r.name}(\sigma_{r.rid = o.rid \wedge o.deliverytime < date\_add(o.ordertime, interval\ 1\ hour)}(\rho_r(RESTAURANT) \times \rho_o(ORDER)))$

4.

④ $\Pi_{o.cid, c.cname}\ \mathcal{G}_{count(*)\ as\ no\_of\_order}(\sigma_{c.cid = o.cid}(\rho_c(CUSTOMER) \times \rho_o(ORDER)))$

5.

⑤ $\Pi_{c.cid, c.cname}(\sigma_{c.cid = o.cid \wedge o.totalprice = (\mathcal{G}_{max(totalprice)}(ORDER))}(\rho_c(CUSTOMER) \times \rho_o(ORDER)))$

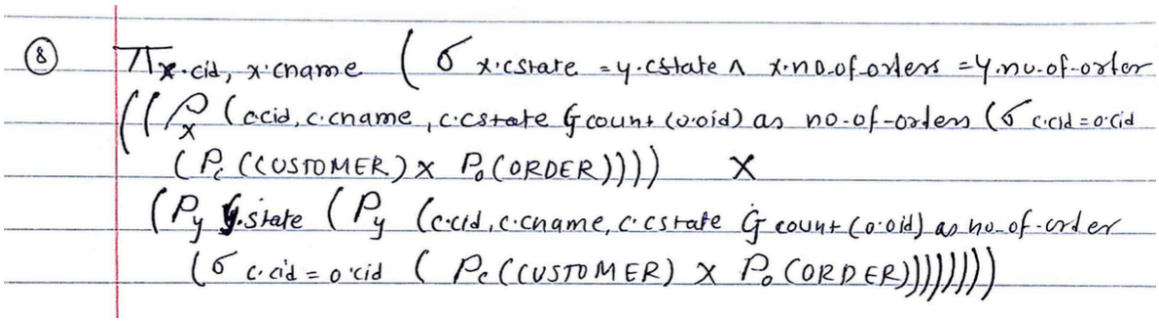
6.

⑥ $\Pi_{o.oid, o.totalprice, \mathcal{G}\ sum(m.price * od.quantity)}(\sigma_{o.oid = od.oid \wedge m.rid = od.rid \wedge m.itemname = od.itemname}(\rho_m(MENUITEM) \times \rho_o(ORDER) \times \rho_{od}(ORDERDETAIL)))$

7.

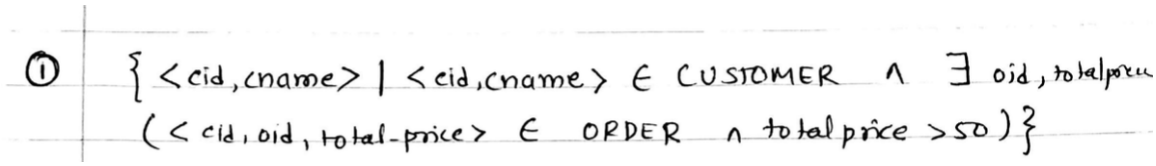
⑦ $\Pi_{c.cid, c.cname}(\sigma_{counter > 1}(c.cid, c.cname\ \mathcal{G}_{count(o.cid)\ as\ counter}(\sigma_{c.cid = o.cid \wedge r.rid = o.rid \wedge r.name = 'Little\ Sheep'}(\rho_c(CUSTOMER) \times \rho_o(ORDER) \times \rho_r(RESTAURANT)))))$

8.

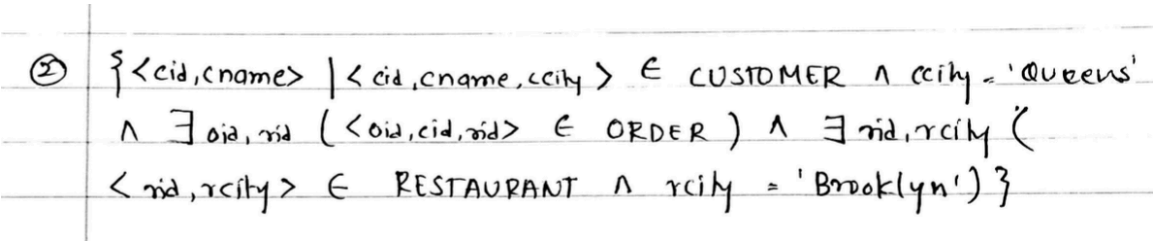


(e) Write either DRC or TRC queries for the above queries. Or explain the reason why you think a particular query cannot be done in DRC or TRC.

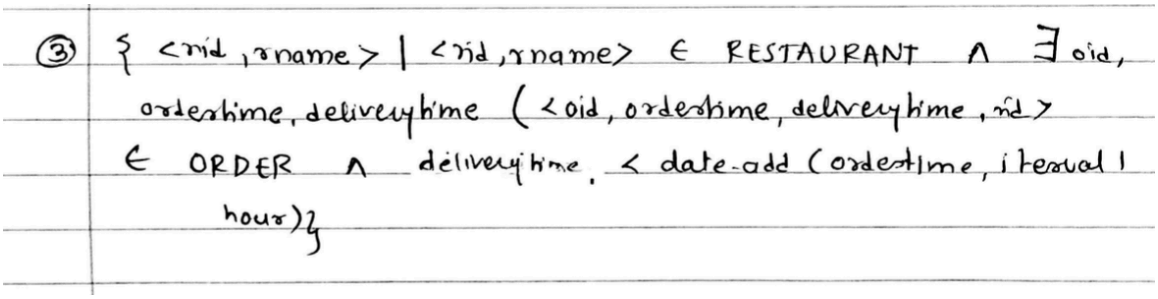
1.



2.



3.



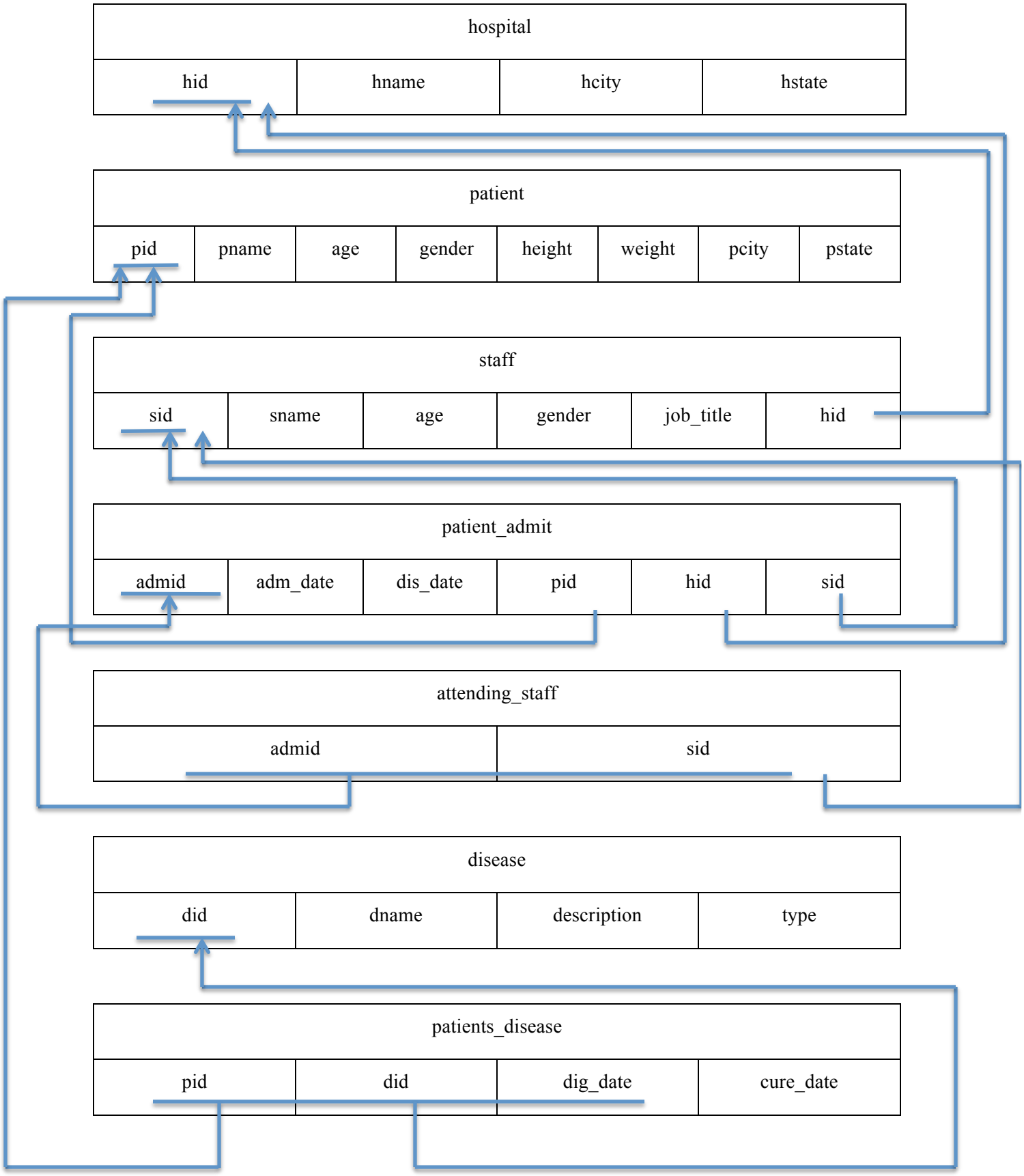
4. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as they do not support aggregate function.

5. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as they do not support aggregate function.
6. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as they do not support aggregate function.
7. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as they do not support aggregate function.
8. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as they do not support aggregate function.
9. Cannot express in Domain Relational Calculus or Tuple Relational Calculus as they do not support aggregate function.

Question 2

a) Schema:

Assumptions: One staff works only in one hospital.


hospital
hid
hname
hcity
hstate
patient
pid
pname
age
gender
height
weight
pcity
pstate
staff
sid
sname
age
gender
job_title
hid
patient_admit
admid
adm_date
dis_date
pid
hid
sid
attending_staff
admid
sid
disease
did
dname
description
type
patients_disease
pid
did
dig_date
cure_date

```
mysql> desc hospital;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| hid    | varchar(10) | NO   | PRI | NULL    |       |
| hname  | varchar(10) | YES  |     | NULL    |       |
| hcity  | varchar(10) | YES  |     | NULL    |       |
| hstate | varchar(10) | YES  |     | NULL    |       |

+--------+-------------+------+-----+---------+-------+


mysql>desc patient;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| pid    | varchar(10) | NO   | PRI | NULL    |       |
| pname  | varchar(10) | YES  |     | NULL    |       |
| age    | varchar(10) | YES  |     | NULL    |       |
| gender | varchar(10) | YES  |     | NULL    |       |
| height | varchar(10) | YES  |     | NULL    |       |
| weight | varchar(10) | YES  |     | NULL    |       |
| pcity  | varchar(10) | YES  |     | NULL    |       |
| pstate | varchar(10) | YES  |     | NULL    |       |

+--------+-------------+------+-----+---------+-------+

mysql> desc staff;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| sid       | varchar(10) | NO   | PRI | NULL    |       |
| sname     | varchar(10) | YES  |     | NULL    |       |
| age       | varchar(10) | YES  |     | NULL    |       |
| gender    | varchar(10) | YES  |     | NULL    |       |
| job_title | varchar(10) | YES  |     | NULL    |       |
| hid       | varchar(10) | YES  | MUL | NULL    |       |

mysql> desc patient_admit;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| admid    | varchar(10) | NO   | PRI | NULL    |       |
| adm_date | date        | YES  |     | NULL    |       |
| dis_date | date        | YES  |     | NULL    |       |
| pid      | varchar(10) | YES  | MUL | NULL    |       |
| hid      | varchar(10) | YES  | MUL | NULL    |       |
| sid      | varchar(10) | YES  | MUL | NULL    |       |

+----------+-------------+------+-----+---------+-------

mysql> desc attending_staff;
+-------+-------------+------+-----+---------+-------+
```

```
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| admid | varchar(10) | NO   | PRI | NULL    |       |
| sid   | varchar(10) | NO   | PRI | NULL    |       |

+-------+-------------+------+-----+---------+-------+
```

```
mysql> desc disease;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| did         | varchar(10) | NO   | PRI | NULL    |       |
| dname       | varchar(10) | YES  |     | NULL    |       |
| description | varchar(10) | YES  |     | NULL    |       |
| type        | varchar(10) | YES  |     | NULL    |       |

+-------------+-------------+------+-----+---------+-------+
```

```
mysql> desc patients_disease;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| pid       | varchar(10) | NO   | PRI | NULL    |       |
| did       | varchar(10) | NO   | PRI | NULL    |       |
| dig_date  | date        | NO   | PRI | NULL    |       |
| cure_date | date        | YES  |     | NULL    |       |

+-----------+-------------+------+-----+---------+-------+
```

1. Output the names of all patients who have been admitted into “St. Rudolph’s Hospital” during 2016.

```
select distinct p.pname
from patient as p join patient_admit as pa on p.pid=pa.pid
join hospital as h on h.hid=pa.hid
where hname='St. Rudolph’s Hospital' and year(adm_date)='2016';
```

2. For each doctor, output their name, and the number of distinct patients for whom they have served as attending physician during a stay.

```
select s.sid,s.sname,count(distinct pid)
from staff as s natural join patient_admit as pa
group by s.sid;
```

3. Output the names of doctors who have been in charge of more than 10 patients who either have or have had a contagious disease.

```
select s.sid,s.sname ,count(distinct pa.pid)
from staff as s natural join patient_admit as pa natural join
```

```
      (select *
       from patients_disease as pd natural join disease as d
       where d.type='c')as a
group by s.sid
having count(distinct pa.pid)>2;
```

4. The BMI (Body Mass Index) is a measure of body fat based on height (h) and weight (w) that applies to adult men and women. Its formula is . For each infectious disease, output the average BMI of patients that currently have this disease.

```
select d.did,d.dname,avg(p.weight/(p.height*p.height))as bmi
from patient as p natural join patients_disease as pd natural join disease as d
where p.age>18 and pd.cure_date is null
group by d.did;
```