DATABASE MANAGEMENT SYSTEM PROJECT

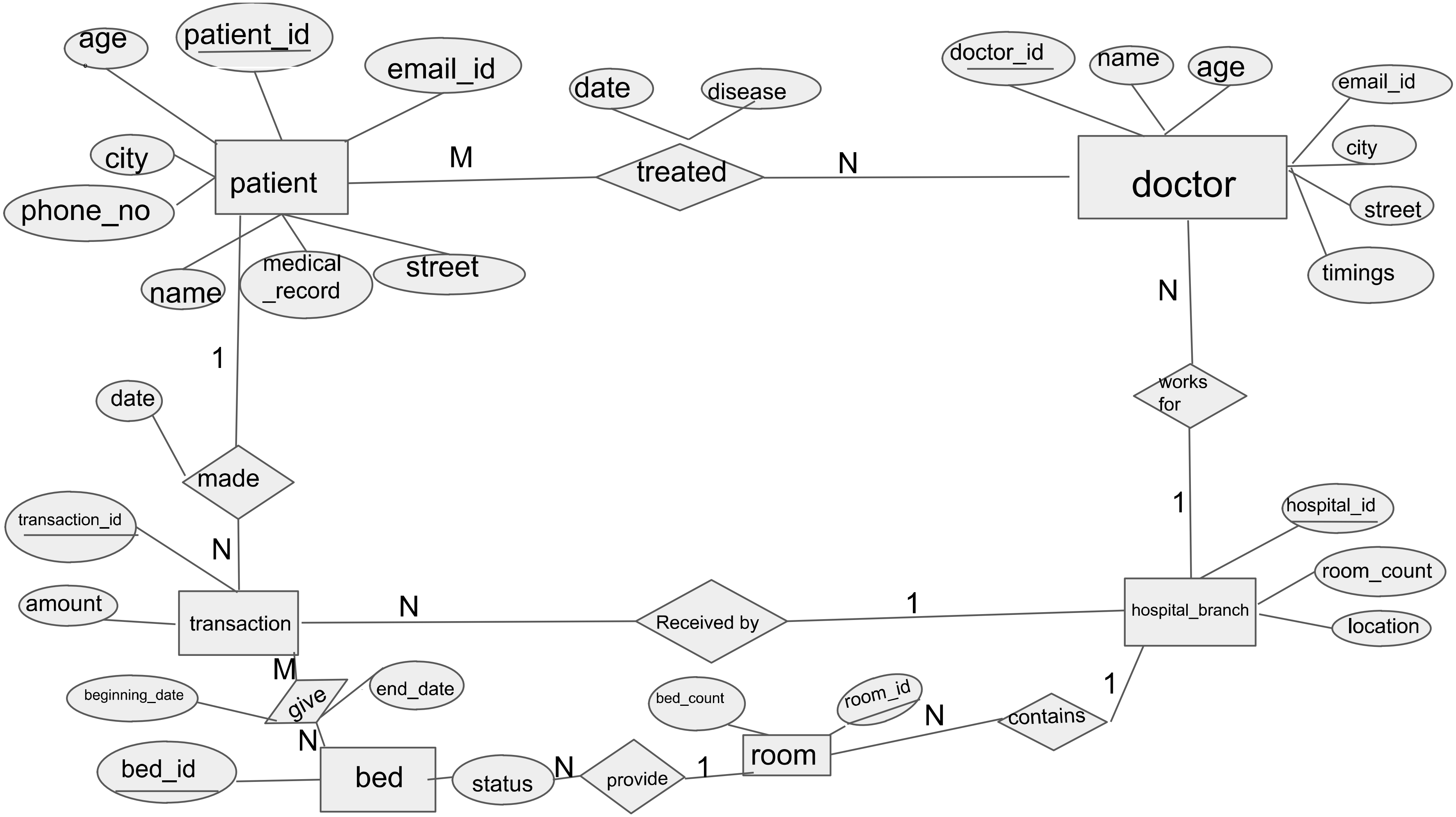
HOSPITAL DATABASE DESIGN

NAME:
KUKUDALA SRISHANTH REDDY -21CHB0B37

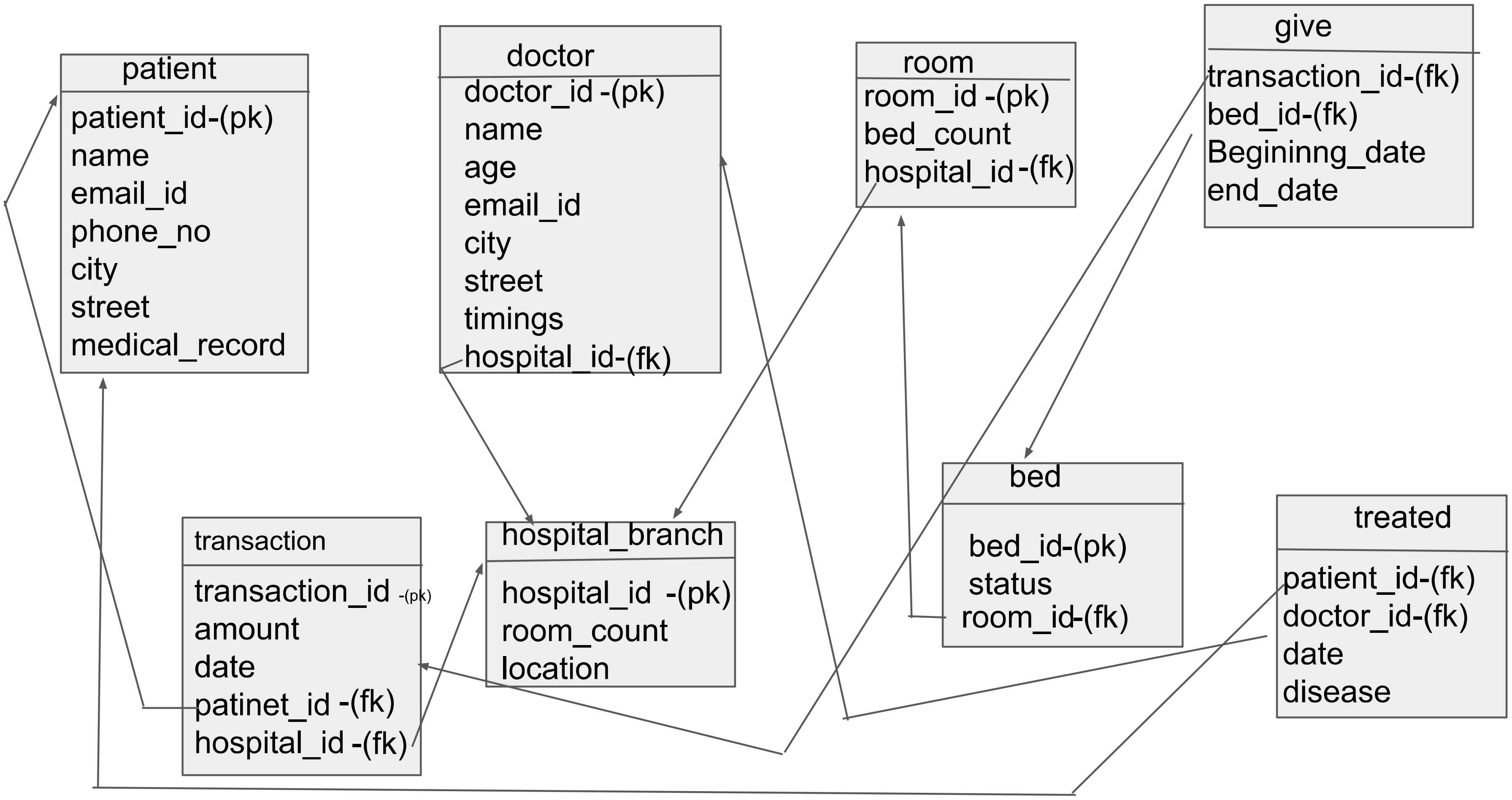
PROBLEM STATEMENT:For this project,our objective is to design a database system for a hospital.The hospital wants to store the information of patients ,doctors and the transaction made by patient and the availability of rooms and beds.The hospital has various branches so they need to store the information of each individual hospital.

CONTENTS:

ER-DIAGRAM:


age
patient_id
email_id
city
patient
phone_no
name
medical
_record
street
M
date
disease
treated
N
doctor_id
name
age
email_id
city
doctor
street
timings
N
works
for
1
1
date
made
N
transaction_id
amount
transaction
N
Received by
1
hospital_id
room_count
hospital_branch
location
M
beginning_date
give
end_date
N
bed_id
bed
status
N
provide
1
bed_count
room_id
room
N
contains
1

RELATION MODEL:



patient
patient_id-(pk)
name
email_id
phone_no
city
street
medical_record

doctor
doctor_id -(pk)
name
age
email_id
city
street
timings
hospital_id-(fk)

room
room_id -(pk)
bed_count
hospital_id -(fk)

give
transaction_id-(fk)
bed_id-(fk)
Begininng_date
end_date

bed
bed_id-(pk)
status
room_id-(fk)

treated
patient_id-(fk)
doctor_id-(fk)
date
disease

transaction
transaction_id -(pk)
amount
date
patinet_id -(fk)
hospital_id -(fk)

hospital_branch
hospital_id -(pk)
room_count
location

ENTITIES:

Patient:This entity holds the information of the patient.The patient_id is the primary key used to identify the unique patients.The other attributes are used to store the patients useful information.

Doctor:This entity stores the information of a doctor working at hospital.The doctor_id is unique and it is used to identify the doctor.

Transaction:This entity stores the information of payment done by a patient.The transaction_id refers to the particular transaction.It is unique.The amount attribute defines the amount that has been paid.

HospitalBranch:This entity stores the information of a particular hospital branch details as they have many branches,each hospital_id uniquely identifies the branch.

Room:This entity stores the information of a particular room and the number of beds it contain.Each room is uniquely identified by room_id.

Bed:This entity stores the information of the bed whether the bed is available is or not.Each bed has a unique id to refer to it.

RELATIONSHIP WITH CARDINALITY:

Patient and Doctor:patient and doctor had a relation i.e doctor treated patient.The cardinality of the relation is M:N because patient may be treated by different doctors at different times because patient can have multiple disease or he/she visited hospital for more than once.so this relation treated has date of treatment and disease patient treated for.

Patient and Transaction:These two entities have a relation i.e patient made transaction.Every patient need to make the transaction for the bill amount.The cardinality of the relation is 1:N.That a patient can make multiple transaction as he/she visited hospital multiple times.But a transaction a can be made by single patient.

Transaction and bed:These two entities have a relation i.e transaction gives bed.The cardinality of the relation is M:N because the with a single transaction we can take many beds same way a single bed can have multiple transaction but not at same time so that the relation has beginning date and end date.

Doctor and Hospital_branch:These two entities have relation i.e doctor works for hospital_branch.The cardinality of the relation is N:1.The doctor only works for single branch.He/She can't work in multiple branches at same time.

Transaction and Hospital_branch:These two entities have relation i.e transaction received by hospital_branch.The cardinality of the relation is N:1.Every transaction is to be received by a single hospital_branch but the hospital_branch can receive multiple transactions.

Hospital_branch and Room:These two entities have relation i.e hospital_branch contains room.The cardinality of the relation is 1:N.Every hospital_branch contains many rooms but a room present in single hospital_branch.the room identified by room_id.

Room and Bed:These two entities have relation i.e room provide bed.The cardinality of the relation is 1:N because every room can have many beds but each bed is present in a single room.

CREATION OF TABLES:

1.**Patient**:

```
    create table patient(patient_id int not null unique,email_id varchar(30),
                         age int,name varchar(50),city varchar(30),street varchar(40),
                         phone_no numeric(10),medical_record varchar(100),primary
                         key(patient_id));
```

2.**doctor**:

```
    create table doctor(doctor_id int not null unique,email_id varchar(30),name varchar(50),
                        age int,city varchar(40),street varchar(40),start_timing time,
                        primary key(doctor_id),foreign key(hospital_id) references
                       hospital_branch(hospital_id),end_time time,hospital_id int);
```

3.**hospital_branch**:

```
    create table hospital_branch(hospital_id int not null unique,room_count int,
                                 Location varchar(90),primary key(hospital_id));
```

4.**transaction**:

```
    create table transaction(transaction_id int not null unique,amount numeric(10,5),
                              patient_id int,foreign key(patient_id) reference patient(patient_id),
                            paid_date date,hospital_id int, foreign key(hospital_id) references
                            hospital_branch(hospital_id),primary key(transaction_id));
```

5.**room**:

```
   create table room(room_id int not null unique,bed_count int,primary key(room_id),
                     hospital_id int, foreign key(hospital_id) references
                     hospital_branch(hospital_id));
```

6.**bed:**

```
        create table bed(bed_id int not null unique,status varchar(10),room_id int,
                              foreign key(room_id) references room(room_id),primary key(bed_id));
```

7.**give**:

```
        create table give(begining_date date,end_date date,transaction_id int,bed_id int,
                              foreign key(transaction_id) references  transaction(transaction_id),
                             foreign key(bed_id) references bed(bed_id),primary
                             key(transaction_id,bed_id,end_date));
```

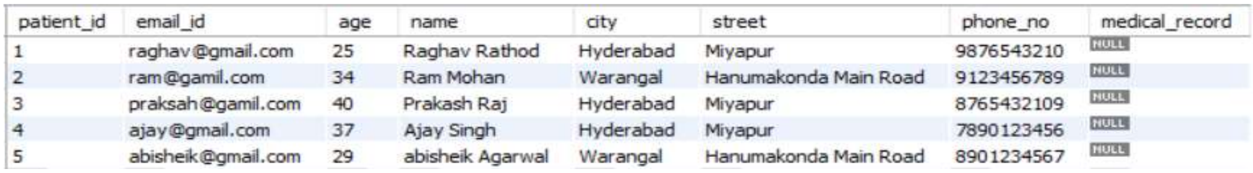
8.**treated**:

```
        create table treated(treated_date date,disease varchar(90),patient_id int, foreign
                                   key(patient_id) references patient(patient_id), doctor_id int,foreign
                                   key(doctor_id) references doctor(doctor_id), primary
                                   key(patient_id,doctor_id,treated_date));
```

INSERTION OF VALUES AND CREATION OF TRIGGERS AND PROCEDURES:

1.patient:

```
              Insert into patient values(1,”raghav@gmail.com”,25,”Raghav Rathod”,”Hyderabad”,
                                                      “Miyapur”,9876543210,null);
              Insert into patient values(2,”ram@gamil.com”,34,”Ram Mohan”,”Warangal”,
                                                      “Hanumakonda Main Road”,9123456789,null);
              Insert into patient values(3,”praksah@gamil.com”,40,”Prakash Raj”,”Hyderabad”,
                                                      “Miyapur”,8765432109,null);
               Insert into patient values(4,”ajay@gmail.com”,37,”Ajay Singh”,”Hyderabad”,
                                                       “Miyapur”,7890123456,null);
               Insert into patient values(5,”abisheik@gmail.com”,29,”abisheik Agarwal”,”Warangal”,
                                                       “Hanumakonda Main Road”,8901234567,null);
```



patient_id	email_id	age	name	city	street	phone_no	medical_record
1	raghav@gmail.com	25	Raghav Rathod	Hyderabad	Miyapur	9876543210	NULL
2	ram@gamil.com	34	Ram Mohan	Warangal	Hanumakonda Main Road	9123456789	NULL
3	praksah@gamil.com	40	Prakash Raj	Hyderabad	Miyapur	8765432109	NULL
4	ajay@gmail.com	37	Ajay Singh	Hyderabad	Miyapur	7890123456	NULL
5	abisheik@gmail.com	29	abisheik Agarwal	Warangal	Hanumakonda Main Road	8901234567	NULL

2.doctor:

```
           Insert into doctor values(101,’arjun@gmail.com’,’Arjun prasad’,36,’Hyderabad’,’Miyapur’,
                                             ‘08:00::00’,200,’20:00:00’);
           Insert into doctor values(102,’Rohit@gamil.com’,’Rohit’,42,’Hyderabad’,’Miyapur’,
                                             ‘08:00:00’,200,’20:00:00’);
           Insert into doctor values(103,’rahul@gamil.com’,’Rahul’,29,’Warangal’,’stationroad’,
                                               ‘08:00:00’,201,’20:00:00’);
           Insert into doctor values(104,’vikraml@gamil.com’,’Vikram’,25,’Warangal’,’stationroad’,
                                               ‘08:00:00’,201,’20:00:00’);
```

Insert into doctor values(105,’rajesh@gamil.com’,’Rajesh’,39,’Warangal’,’stationroad’,
‘08:00:00’,201,’20:00:00’);

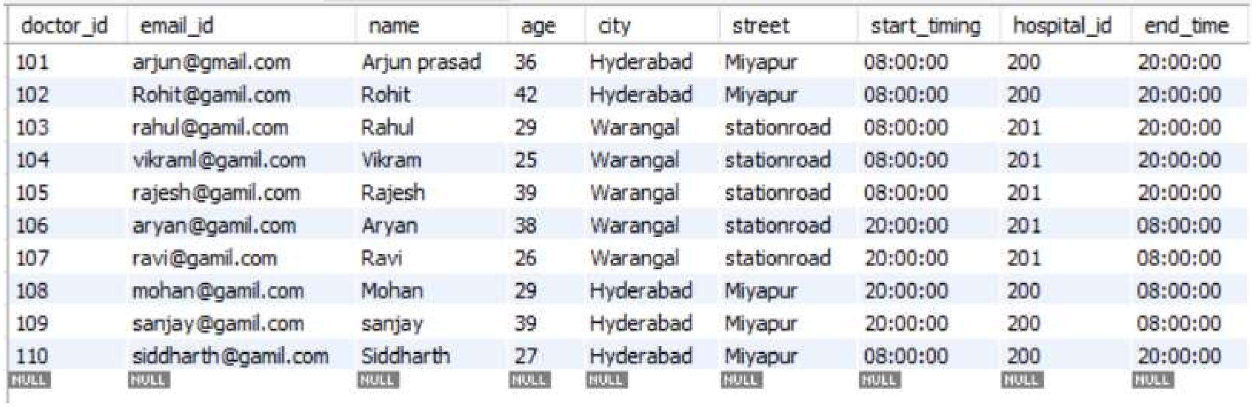
Insert into doctor values(106,’aryan@gamil.com’,’Aryan’,38,’Warangal’,’stationroad’,
’20:00:00’,201,‘08:00:00’);

Insert into doctor values(107,’ravi@gamil.com’,’Ravi’,26,’Warangal’,’stationroad’,
’20:00:00’,201,‘08:00:00’);

Insert into doctor values(108,’mohan@gamil.com’,’Mohan’,29,’Hyderabad’,’Miyapur’,
’20:00:00’,200,‘08:00:00’);

Insert into doctor values(109,’sanjay@gamil.com’,’sanjay’,39,’Hyderabad’,’Miyapur’,
’20:00:00’,200,‘08:00:00’);

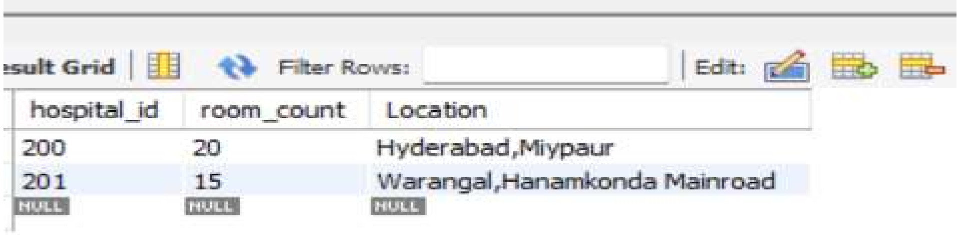
Insert into doctor values(110,’siddharth@gamil.com’,’Siddharth’,27,’Hyderabad’,’Miyapur’,
‘08:00:00’,200,’20:00:00’);



doctor_id	email_id	name	age	city	street	start_timing	hospital_id	end_time
101	arjun@gmail.com	Arjun prasad	36	Hyderabad	Miyapur	08:00:00	200	20:00:00
102	Rohit@gamil.com	Rohit	42	Hyderabad	Miyapur	08:00:00	200	20:00:00
103	rahul@gamil.com	Rahul	29	Warangal	stationroad	08:00:00	201	20:00:00
104	vikraml@gamil.com	Vikram	25	Warangal	stationroad	08:00:00	201	20:00:00
105	rajesh@gamil.com	Rajesh	39	Warangal	stationroad	08:00:00	201	20:00:00
106	aryan@gamil.com	Aryan	38	Warangal	stationroad	20:00:00	201	08:00:00
107	ravi@gamil.com	Ravi	26	Warangal	stationroad	20:00:00	201	08:00:00
108	mohan@gamil.com	Mohan	29	Hyderabad	Miyapur	20:00:00	200	08:00:00
109	sanjay@gamil.com	sanjay	39	Hyderabad	Miyapur	20:00:00	200	08:00:00
110	siddharth@gamil.com	Siddharth	27	Hyderabad	Miyapur	08:00:00	200	20:00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.hospital_branch:

Insert into hospital_branch values(200,20,”Hyderabad,Miypaur”);
Insert into hospital_branch values(201,15,”Warangal,Hanamkonda Mainroad”);

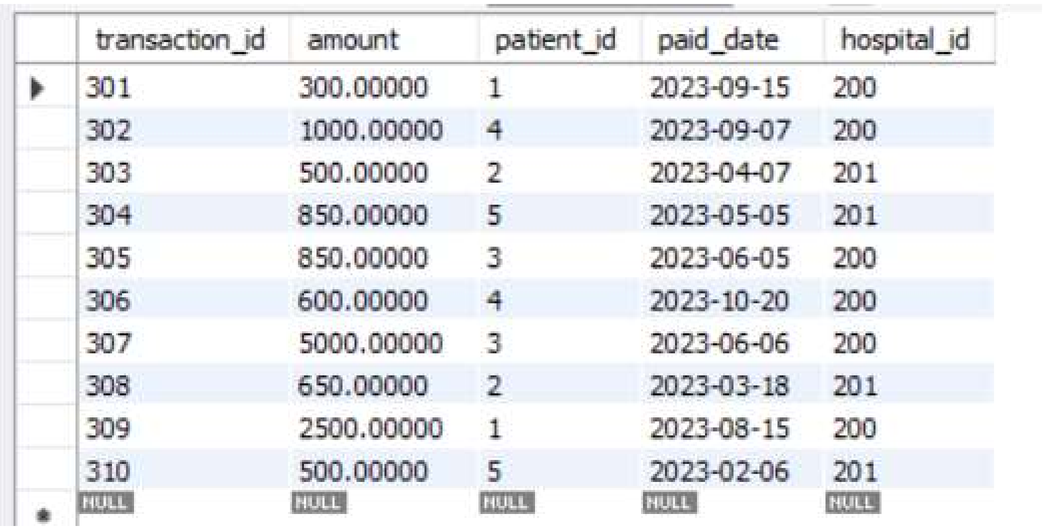


hospital_id	room_count	Location
200	20	Hyderabad,Miypaur
201	15	Warangal,Hanamkonda Mainroad
NULL	NULL	NULL

4.transaction:

insert into transaction values(301,300,1,'2023-09-15',200);
insert into transaction values(302,1000,4,'2023-09-07',200);
insert into transaction values(303,500,2,'2023-04-07',201);
insert into transaction values(304,850,5,'2023-05-05',201);
insert into transaction values(305,850,3,'2023-06-05',200);
insert into transaction values(306,600,4,'2023-10-20',200);
insert into transaction values(307,5000,3,'2023-06-06',200);
insert into transaction values(308,650,2,'2023-03-18',201);

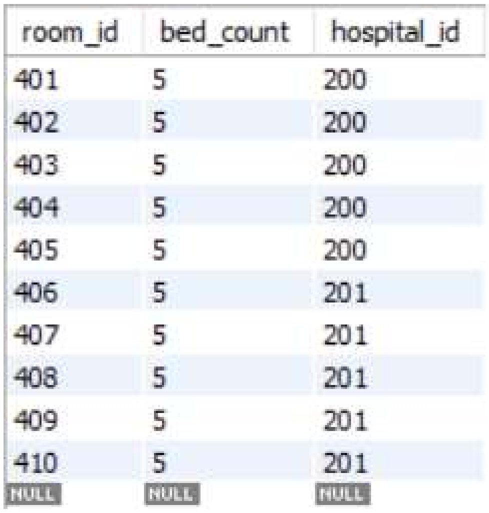
```
insert into transaction values(309,2500,1,'2023-08-15',200);
insert into transaction values(310,500,5,'2023-02-06',201);
```



transaction_id	amount	patient_id	paid_date	hospital_id
301	300.00000	1	2023-09-15	200
302	1000.00000	4	2023-09-07	200
303	500.00000	2	2023-04-07	201
304	850.00000	5	2023-05-05	201
305	850.00000	3	2023-06-05	200
306	600.00000	4	2023-10-20	200
307	5000.00000	3	2023-06-06	200
308	650.00000	2	2023-03-18	201
309	2500.00000	1	2023-08-15	200
310	500.00000	5	2023-02-06	201
NULL	NULL	NULL	NULL	NULL

5.room:

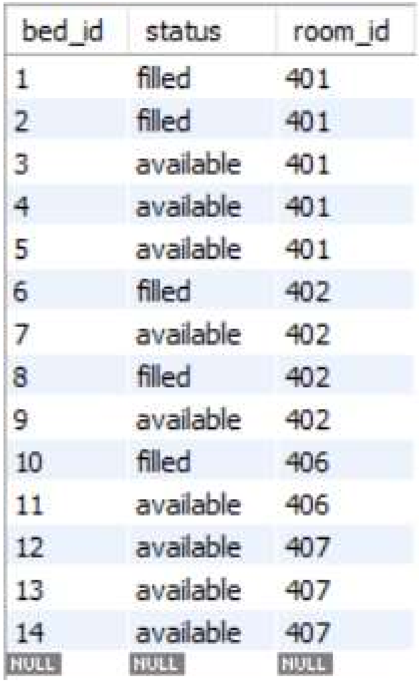
```
insert into room values(401,5,200);
insert into room values(402,5,200);
insert into room values(403,5,200);
insert into room values(404,5,200);
insert into room values(405,5,200);
insert into room values(406,5,201);
insert into room values(407,5,201);
insert into room values(408,5,201);
insert into room values(409,5,201);
insert into room values(410,5,201);
```



room_id	bed_count	hospital_id
401	5	200
402	5	200
403	5	200
404	5	200
405	5	200
406	5	201
407	5	201
408	5	201
409	5	201
410	5	201
NULL	NULL	NULL

6.bed:

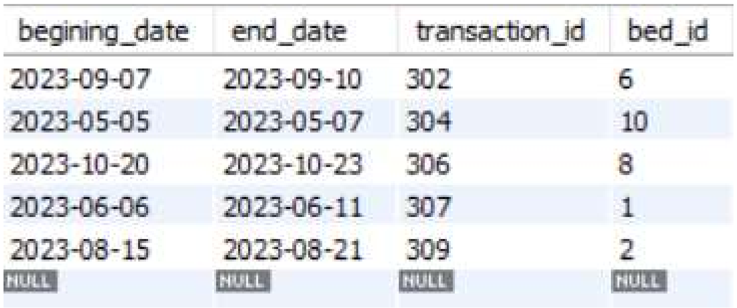
```
insert into bed values(1,’filled’,401);
insert into bed values(2,’filled’,401);
insert into bed values(3,’available’,401);
insert into bed values(4,’available’,401);
insert into bed values(5,’available’,401);
    insert into bed values(6,’filled’,402);
    insert into bed values(7,’available’,402);
    insert into bed values(8,’filled’,402);
    insert into bed values(9,’available’,402);
   insert into bed values(10,’filled’,406);
  insert into bed values(11,’available’,406);
  insert into bed values(12,’available’,407);
  insert into bed values(13,’available’,407);
  insert into bed values(14,’available’,407);
```



bed_id	status	room_id
1	filled	401
2	filled	401
3	available	401
4	available	401
5	available	401
6	filled	402
7	available	402
8	filled	402
9	available	402
10	filled	406
11	available	406
12	available	407
13	available	407
14	available	407
NULL	NULL	NULL

7.give:

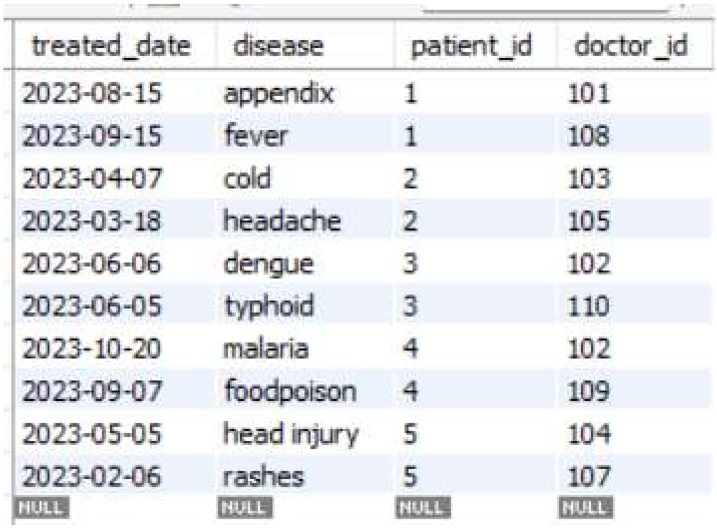
```
 insert into give values(‘2023-06-06’,’2023-06-11’,307,1);
 insert into give values(‘2023-08-15’,’2023-08-21’,309,2);
 insert into give values(‘2023-09-07’,’2023-09-10’,302,6);
 insert into give values(‘2023-05-05’,’2023-05-07’,304,10);
insert into give values(‘2023-10-20’,’2023-10-23’,306,8);
```



begining_date	end_date	transaction_id	bed_id
2023-09-07	2023-09-10	302	6
2023-05-05	2023-05-07	304	10
2023-10-20	2023-10-23	306	8
2023-06-06	2023-06-11	307	1
2023-08-15	2023-08-21	309	2
NULL	NULL	NULL	NULL

8.treated:

```
insert into treated values(‘2023-08-15’,’appendix’,1,101);
insert into treated values(‘2023-09-15’,’fever’,1,108);
insert into treated values(‘2023-09-07’,’foodpoison’,4,109);
insert into treated values(‘2023-04-07’,’cold’,2,103);
insert into treated values(‘2023-05-05’,’head injury’,5,104);
insert into treated values(‘2023-06-05’,’typhoid’,3,110);
insert into treated values(‘2023-10-20’,’malaria’,4,102);
insert into treated values(‘2023-06-06’,’dengue’,3,102);
insert into treated values(‘2023-03-18’,’headache’,2,105);
insert into treated values(‘2023-02-06’,’rashes’,5,107);
```



treated_date	disease	patient_id	doctor_id
2023-08-15	appendix	1	101
2023-09-15	fever	1	108
2023-04-07	cold	2	103
2023-03-18	headache	2	105
2023-06-06	dengue	3	102
2023-06-05	typhoid	3	110
2023-10-20	malaria	4	102
2023-09-07	foodpoison	4	109
2023-05-05	head injury	5	104
2023-02-06	rashes	5	107
NULL	NULL	NULL	NULL

TRIGGERS:

```
 Create tiggers bed_status
After insert on give
For each row
Begin
       Update bed
       Set status=”filled”
      Where new.bed_id=bed.bed_id;
end;
```

This trigger helps when a transaction is made for bed it automatically updates the status of bed to filled.

```
Create tigger update_record
After insert on treated
For each row
Begin
     Update patient
     Set medical_record=concat_ws(“,”,medical_record,new.disease)
    Where new.patient_id=patient.patinet_id
```

```
End;
```

This trigger helps to write the medical_record of the patient.it needs to updated when the patient gets treated for the disease.

PROCEDURES:

```
Create procedure update_status(IN bed_id int,IN end_date date)
Begin
        Update bed
        Set status=”available”
        Where datediff(curdate(),end_date)>=0 and bed_id=bed.bed_id;
End;
```

These procedure helps to update the bed status when the end_date of transaction is completed.we need to call these procedure.

```
Create procedure extend_date(IN new_date date,IN bed_id int)
Begin
       Update give
       Set end_date=new_date
       Where bed_id=give.bed_id;
End;
```

These procedure was created to extend the enddate of the patient bed when the patient needs more rest he/she needs to extend date then these procedure need to be called.

FUNCTIONAL DEPENDENCY AND NORMALIZATION:

The normalization has taken into from the beginning of ER-diagram creation and Relational schema design.

1.patient:

pateint_id→(name,age,email_id,city,street,medical_record,phone_no)

This is the functional dependency as the patient_id is the candidate key so there is no partial dependency and there is no transitive dependency.The table is in BCNF.

2.doctor:

doctor_id→(name,age,email_id,city,street,start_time,end_time,hospital_id)

This is the only functional dependency present in these table.The doctor_id is the candidate key as the closure of the doctor_id gives all the attributes.The table contains no partial dependency of the candidate key and there is no transitive dependency present in the table.The table is in BCNF.

3.hospiatl_branch:

hospial_id→(room_count,location)

This is the functional dependency present in these table.The hospital_id is the candidate key and all the attributes are fully dependent on the candidate key and there is no transitive dependency.The table is in BCNF.

4.transaction:

transaction_id→(amount,patient_id,hospital_id)

This is the functional dependency present in these table.The transaction_id is the candidate key and there is no functional dependency between patient_id and hospital_id.There is no partial dependency and there is no transitive dependency.The table is in BCNF.

5.room:

room_id→(bed_count,hospital_id)

This is the functional dependency present in these table.The room_id is the candidate key for this table.There is no partial dependency and transitive dependency.The table is in BCNF.

6.bed:

bed_id→(status,room_id)

This is the functional dependency present in these table.The bed_id is the candidate key and there is no partial and transitive dependency.The table is in BCNF.

7.give:

transaction_id,bed_id,end_date→(bed_id,transaction_id,begining_date,end_date)

This is the functional dependency present in these table.The candidate key is transaction_id,bed_id,end_date.The prime attributes are transaction_id,bed_id,end_date.There is no partial dependency or there is no dependency on this prime attributes.There is no transitive dependency.The table is in BCNF.

8.treated:

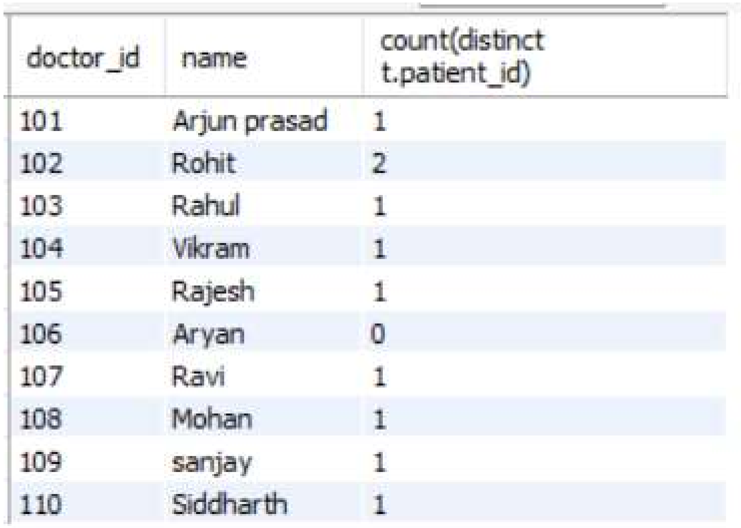
patient_id,doctor_id,treated_date→(patient_id,doctor_id,treated_date,disease)

This is the functional dependency present in these table.The prime attributes are patient_id,doctor_id,treated_date.There is no functional dependency alone on the prime attributes.like with patient_id we can't find the doctor_id treated because the patient may be treated by same doctor on two different dates.The table is in BCNF.

QUERIES:

1.Find the number of patients each doctor has treated.

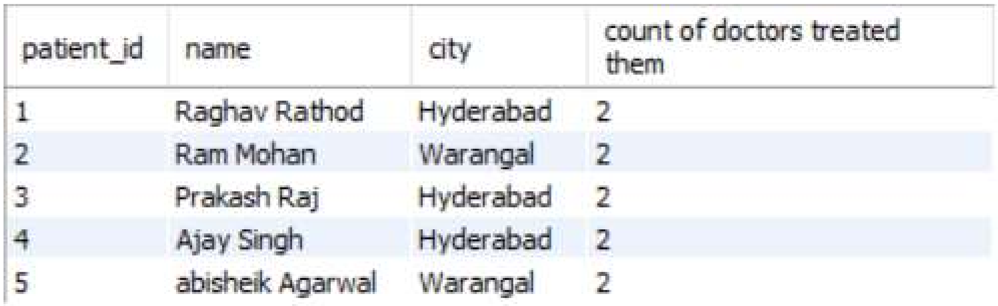
```
select distinct d.doctor_id,d.name,count(distinct t.patient_id)
from doctor d left join treated t on d.doctor_id=t.doctor_id
group by 1
```



doctor_id	name	count(distinct t.patient_id)
101	Arjun prasad	1
102	Rohit	2
103	Rahul	1
104	Vikram	1
105	Rajesh	1
106	Aryan	0
107	Ravi	1
108	Mohan	1
109	sanjay	1
110	Siddharth	1

2.Find the number of doctors who have treated the patient.

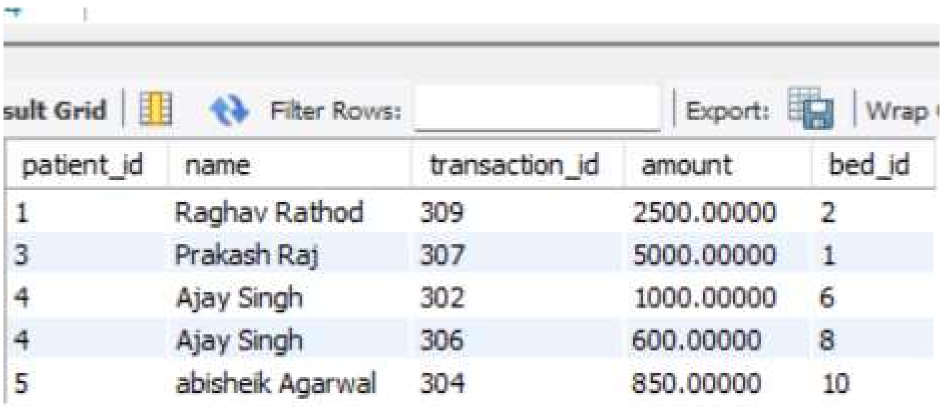
```
select distinct p.patient_id,p.name,p.city,count(distinct t.doctor_id) as "count of doctors treated them"
from patient p left join treated t on p.patient_id=t.patient_id
group by 1;
```



patient_id	name	city	count of doctors treated them
1	Raghav Rathod	Hyderabad	2
2	Ram Mohan	Warangal	2
3	Prakash Raj	Hyderabad	2
4	Ajay Singh	Hyderabad	2
5	abisheik Agarwal	Warangal	2

3.Find the number of patients who made transaction for bed.

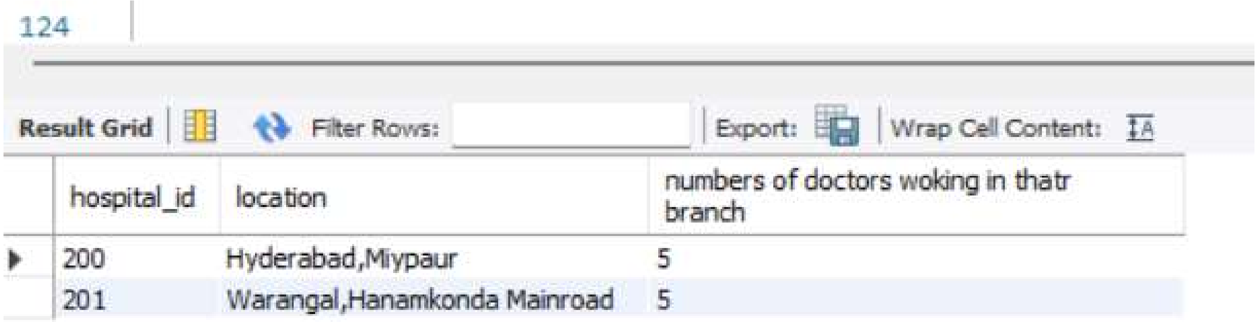
```
select p.patient_id,p.name,t.transaction_id,t.amount,g.bed_id
from patient p join transaction t on p.patient_id=t.patient_id join give g on
t.transaction_id=g.transaction_id
order by 1;
```



patient_id	name	transaction_id	amount	bed_id
1	Raghav Rathod	309	2500.00000	2
3	Prakash Raj	307	5000.00000	1
4	Ajay Singh	302	1000.00000	6
4	Ajay Singh	306	600.00000	8
5	abisheik Agarwal	304	850.00000	10

4.Find the number of doctors each hospital_branch contains.

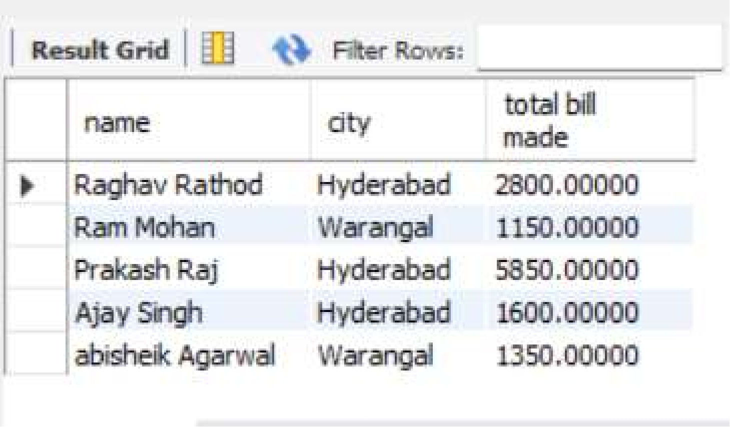
select h.hospital_id,h.location,count(distinct d.doctor_id) as "numbers of doctors woking in thatr branch"
from hospital_branch h join doctor d on h.hospital_id=d.hospital_id
group by 1;



hospital_id	location	numbers of doctors woking in thatr branch
200	Hyderabad,Miypaur	5
201	Warangal,Hanamkonda Mainroad	5

5.Find the bills made by each patient.

select p.name,p.city,sum(t.amount) as "total bill made"
from patient p join transaction t on p.patient_id=t.patient_id
group by p.patient_id;



name	city	total bill made
Raghav Rathod	Hyderabad	2800.00000
Ram Mohan	Warangal	1150.00000
Prakash Raj	Hyderabad	5850.00000
Ajay Singh	Hyderabad	1600.00000
abisheik Agarwal	Warangal	1350.00000