

# HOSPITAL MANAGEMENT SYSTEM

DATABASE MANAGEMENT SYSTEMS (CSC2003)

– PROJECT COMPONENT - REVIEW REPORT

Submitted by

**P.SATHISH KUMAR (17BCS0082)**

**EZIL ARASAN (17BCS0022)**

**AKASH (17BCS0110)**

Submitted To

**Dr. T. Janani**

**Assistant Professor**

**SCHOOL OF COMPUTING SCIENCE AND  
ENGINEERING**



October, 2018

[Table of Contents](#)

<b>1.</b>	<b>Introduction .....</b>	<b>3</b>
<b>2.</b>	<b>Project SCOPE .....</b>	<b>3</b>
<b>3.</b>	<b>Key Contacts and Stakeholders .....</b>	<b>3</b>
<b>4.</b>	<b>Project Resource Requirements.....</b>	<b>3</b>
4.1	Software Resource Requirements.....	3
<b>5.</b>	<b>ER Diagram .....</b>	<b>4</b>
<b>6.</b>	<b>Tables and Constraints .....</b>	<b>5</b>
<b>7.</b>	<b>Queries and Screenshot.....</b>	<b>6</b>
<b>8.</b>	<b>Normalization form of Schema .....</b>	<b>6</b>
<b>9.</b>	<b>Work break down.....</b>	<b>13</b>
<b>10.</b>	<b>Review evaluation.....</b>	<b>13</b>

## 1. INTRODUCTION

Increasing accidents and diseases in India it is mandatory to have more hospitals and doctors. In hospitals it's necessary to maintain the patients records who visited there hospital and the employees who are working in the hospitals. So the hospital management database will solve this problem by storing the information of hospital.

## 2. PROJECT SCOPE [LIST ALL THE FUNCTIONALITIES HERE]

The database can help in generating following information:

1. Patient's general information and medical history;
2. Patient's stay time on hospital and deposit;
3. Information on doctor's general information;
4. Doctor's prescriptions and remarks on patient's health;
5. Information on patient's total expenses for collective billing;
6. Information on tests carried upon patient and medicines purchased on his behalf.

## 3. KEY CONTACTS AND STAKEHOLDERS

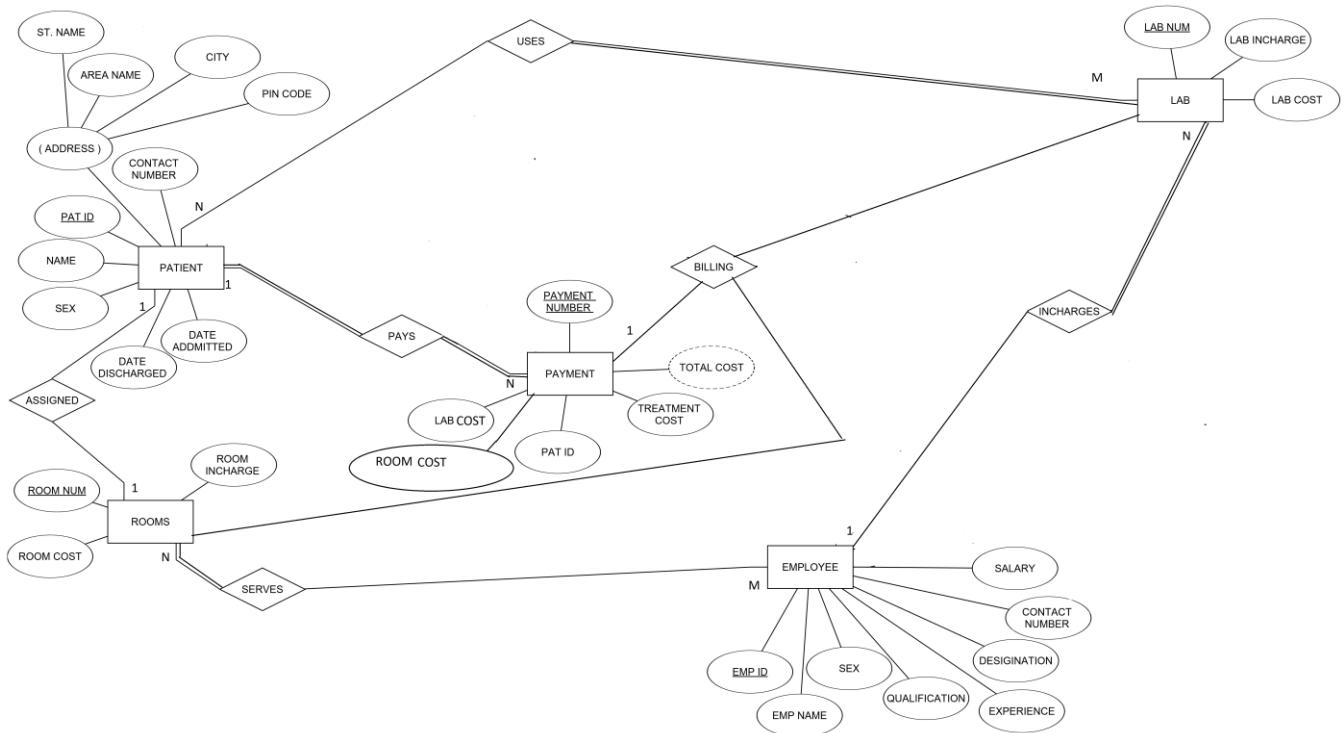
Name	Registration Number	Phone Number
SATHISH KUMAR P	17BCS0082	9486136916
EZHILARASAN P	17BCS0022	9159298650
AKASH R	17BCS0110	8610789194

## 4. PROJECT RESOURCE REQUIREMENTS

### 4.1 Software Resource Requirements

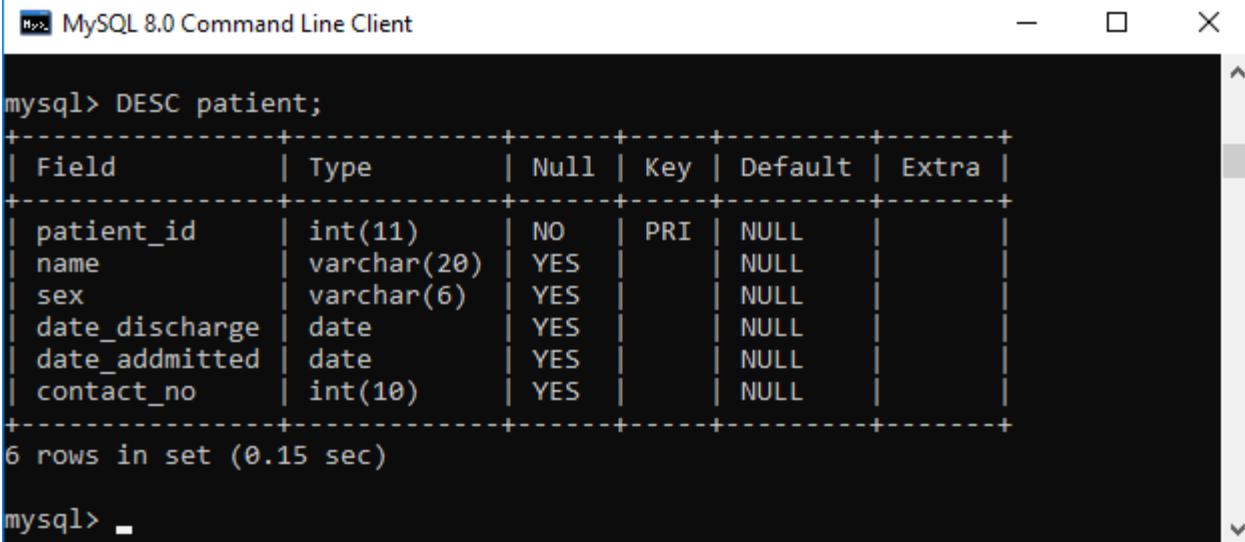
- Windows 10
- mysql

## 5. ER DIAGRAM



## 6. TABLES AND CONSTRAINTS

PATIENT:

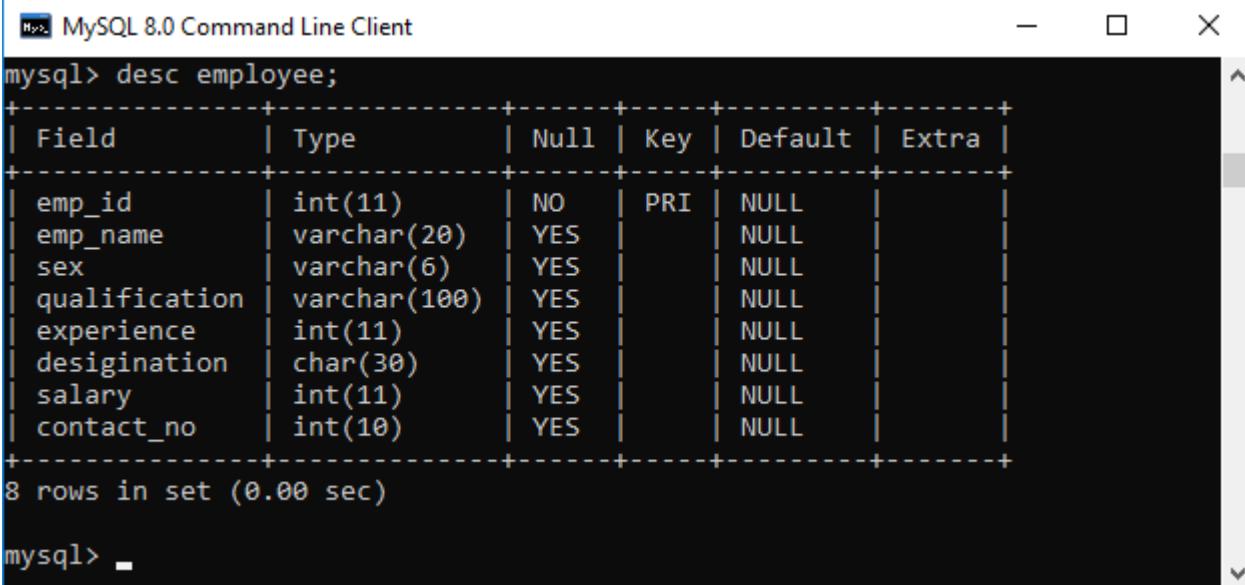


MySQL 8.0 Command Line Client

```
mysql> DESC patient;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| patient_id | int(11) | NO | PRI | NULL |       |
| name | varchar(20) | YES |       | NULL |       |
| sex | varchar(6) | YES |       | NULL |       |
| date_discharge | date | YES |       | NULL |       |
| date_admitted | date | YES |       | NULL |       |
| contact_no | int(10) | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.15 sec)

mysql> -
```

EMPLOYEE:



MySQL 8.0 Command Line Client

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(11) | NO | PRI | NULL |       |
| emp_name | varchar(20) | YES |       | NULL |       |
| sex | varchar(6) | YES |       | NULL |       |
| qualification | varchar(100) | YES |       | NULL |       |
| experience | int(11) | YES |       | NULL |       |
| designation | char(30) | YES |       | NULL |       |
| salary | int(11) | YES |       | NULL |       |
| contact_no | int(10) | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> -
```

## ROOMS:

```
MySQL 8.0 Command Line Client

mysql> desc rooms;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| room_num | int(11) | NO | PRI | NULL |
| room_cost | int(11) | YES | | NULL |
| pat_id | int(11) | YES | MUL | NULL |
| room_incharge | int(11) | YES | MUL | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> -
```

## PAYMENT:

```
MySQL 8.0 Command Line Client

mysql> desc payment;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| payment_id | varchar(10) | NO | PRI | NULL | |
| treatment_cost | int(11) | YES | | NULL |
| lab_cost | int(11) | YES | | NULL |
| room_cost | int(11) | YES | | NULL |
| pat_id | int(11) | YES | MUL | NULL |
| total_cost | int(11) | YES | | NULL | VIRTUAL GENERATED |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## LAB:

```
MySQL 8.0 Command Line Client

mysql> desc lab;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| lab_num | int(11) | NO | PRI | NULL |
| lab_cost | int(11) | YES | | NULL |
| lab_incharge | int(11) | YES | MUL | NULL |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> -
```

## 7. QUERIES AND SCREENSHOT

### QUERIES FOR CREATING TABLES:

```
mysql> create table patient(patient_id int primary key,name varchar(20),sex varchar(6),date_discharge date,date_admitted date,contact_no int(10));
```

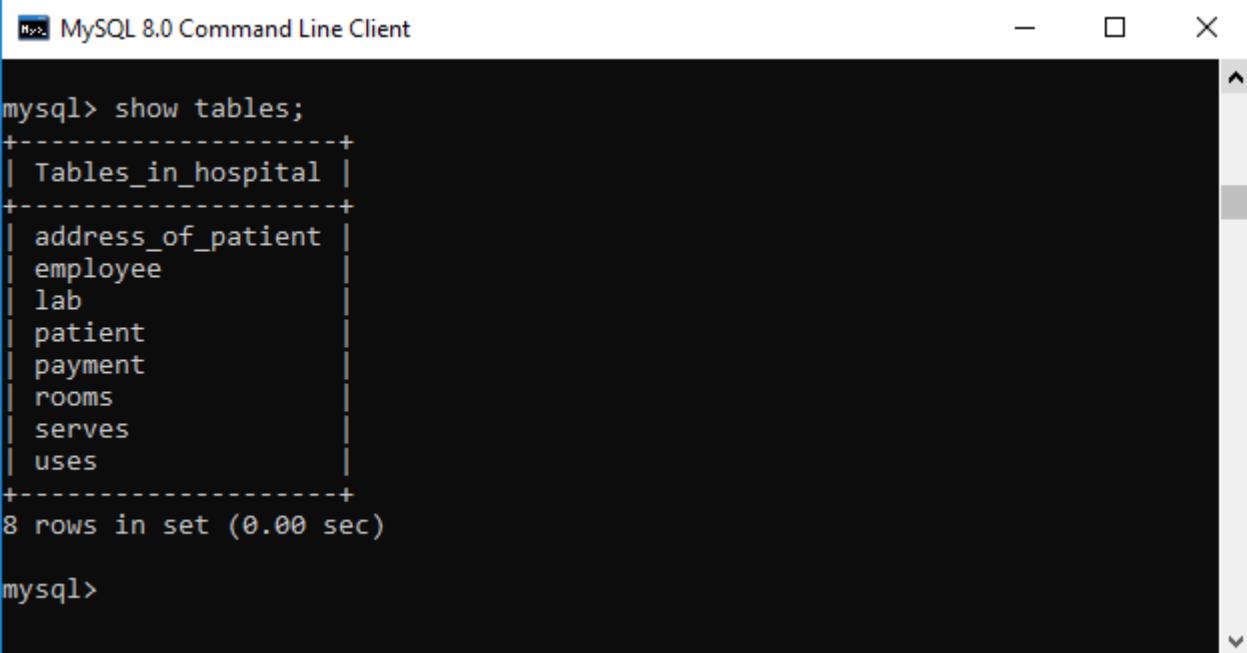
```
mysql> create table employee(emp_id int primary key,emp_name varchar(20),sex varchar(6),qualification varchar(100),experience int,designation char(30),salary int,contact_no int(10));
```

```
mysql> create table rooms(room_num int primary key,room_cost int,room_incharge varchar(20),pat_id int);
```

```
mysql> create table lab(lab_num int primary key,lab_cost int,lab_incharge varchar(20),emp_id int);
```

```
mysql> create table payment(payment_id varchar(10) primary key,treatment_cost int,lab_cost int,room_cost int,pat_id int);
```

```
mysql> alter table payment add column total_cost int as (treatment_cost+lab_cost+room_cost);
```



The screenshot shows the MySQL 8.0 Command Line Client window. The title bar reads "MySQL 8.0 Command Line Client". The main area displays the output of the "show tables;" command. The output is a table with one column labeled "Tables\_in\_hospital" containing eight rows: address\_of\_patient, employee, lab, patient, payment, rooms, serves, and uses. Below the table, it says "8 rows in set (0.00 sec)". At the bottom, there is a prompt "mysql>".

Tables_in_hospital
address_of_patient
employee
lab
patient
payment
rooms
serves
uses

8 rows in set (0.00 sec)

mysql>

## QUERIES FOR ASSIGNING FOREIGN KEYS IN 1:1 RELATION

```
mysql> alter table rooms add foreign key (pat_id) references patient(patient_id);
```

The screenshot shows a MySQL command-line interface window. It displays the following SQL commands and their results:

```

mysql> Select MySQL 8.0 Command Line Client
+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+
| room_num   | int(11)    | NO   | PRI  | NULL    |       |
| room_cost  | int(11)    | YES  |       | NULL    |       |
| room_incharge | varchar(20) | YES  |       | NULL    |       |
| pat_id     | int(11)    | YES  |       | NULL    |       |
+-----+
4 rows in set (0.00 sec)

mysql> alter table rooms add constraint FK_ASSIGNED foreign key (pat_id) references patient(pat_id);
ERROR 3734 (HY000): Failed to add the foreign key constraint. Missing column 'pat_id' for constraint 'FK_ASSIGNED' in the referenced table 'patient'
mysql> desc patient;
+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+
| patient_id | int(11)    | NO   | PRI  | NULL    |       |
| name        | varchar(20) | YES  |       | NULL    |       |
| sex         | varchar(6)  | YES  |       | NULL    |       |
| date_discharge | date      | YES  |       | NULL    |       |
| date_admitted | date      | YES  |       | NULL    |       |
| contact_no  | int(10)    | YES  |       | NULL    |       |
+-----+
6 rows in set (0.00 sec)

mysql> alter table rooms add foreign key (pat_id) references patient(patient_id);
Query OK, 0 rows affected (1.67 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

## QUERIES FOR ASSIGNING FOREIGN KEYS IN 1:N AND N:1 RELATION

```
mysql> alter table lab add foreign key (lab_incharge) references employee(emp_id);
mysql> alter table payment add foreign key (pat_id) references patient(pat_id);
```

The screenshot shows a MySQL command-line interface window. It displays the following SQL commands and their results:

```

mysql> Select MySQL 8.0 Command Line Client
mysql> alter table payment add foreign key (pat_id) references employee(emp_id);
Query OK, 0 rows affected (1.71 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc payment;
+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+
| payment_id | varchar(10) | NO   | PRI  | NULL    |       |
| cost_of_equip | int(11)    | YES  |       | NULL    |       |
| treatment_cost | int(11)    | YES  |       | NULL    |       |
| lab_cost    | int(11)    | YES  |       | NULL    |       |
| room_cost   | int(11)    | YES  |       | NULL    |       |
| total_cost  | int(11)    | YES  |       | NULL    | VIRTUAL GENERATED |
| pat_id      | int(11)    | YES  | MUL  | NULL    |       |
+-----+
7 rows in set (0.08 sec)

mysql> desc lab;
+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+
| lab_num    | int(11)    | NO   | PRI  | NULL    |       |
| lab_cost   | int(11)    | YES  |       | NULL    |       |
| lab_incharge | int(11)    | YES  | MUL  | NULL    |       |
+-----+
3 rows in set (0.00 sec)

mysql> alter table lab add foreign key (lab_incharge) references employee(emp_id);
Query OK, 0 rows affected (1.75 sec)
```

## QUERIES IN RELATION M:N

```
Create table serves(room_num int,emp_id int);
create table uses(pat_id int,lab_num int);
```

```
mysql> desc serves;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| room_num | int(11) | YES | MUL | NULL    |
| emp_id   | int(11) | YES | MUL | NULL    |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> desc uses;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| pat_id | int(11) | YES | MUL | NULL    |
| lab_num | int(11) | YES | MUL | NULL    |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## QUERIES FOR N-ARRAY RELATION RELATION:

Here we have n-array relations with patient,rooms,lab,payment so we create a new table “billing”

```
mysql> create table billing(pat_id int,room_num int,lab_num int,payment_id varchar(8));
```

Connecting the foreign key

```
mysql> alter table billing add foreign key (pat_id) references patient(patient_id);
mysql> alter table billing add foreign key (room_num) references rooms(room_num);
mysql> alter table billing add foreign key (lab_num) references lab(lab_num);
mysql> alter table billing add foreign key (payment_id) references payment(payment_id);
```

```
mysql> desc billing;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| pat_id | int(11) | YES | MUL | NULL    |
| room_num | int(11) | YES | MUL | NULL    |
| lab_num | int(11) | YES | MUL | NULL    |
| payment_id | varchar(8) | YES | MUL | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.14 sec)

mysql>
```

## QURIES FOR MULTIVALUED ATTRIBUTE:

```
mysql> create table address_of_patient(pat_id int,st_name varchar(20),area varchar(20),city varchar(20));
```

The screenshot shows a terminal window titled "Select MySQL 8.0 Command Line Client". It displays the following MySQL session:

```

+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| pat_id | int(11) | YES  |     | NULL    |       |
| treatment_id | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> desc gets;
+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| patient_id | int(11) | NO   | PRI | NULL    |       |
| name        | varchar(20) | YES  |     | NULL    |       |
| sex         | varchar(6)  | YES  |     | NULL    |       |
| date_discharge | date   | YES  |     | NULL    |       |
| date_admitted | date   | YES  |     | NULL    |       |
| contact_no  | int(10) | YES  |     | NULL    |       |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> create table address_of_patient(pat_id int,st_name varchar(20),area varchar(20),city varchar(20));
Query OK, 0 rows affected (0.46 sec)

mysql>

```

## INSERTING VALUES

```
insert into employee (emp_id,emp_name,sex,qualification,experience,desigination,salary,contact_no)
values(115,'ankitha','female','nursing',1,'nurse',1000,95865631);
```

```
mysql> insert into employee (emp_id,emp_name,sex,qualification,experience,desigination,salary,contact_no)
values(111,'chandra','male','mbbs',2,'doctor',10000,96325841);
```

```
mysql> insert into employee (emp_id,emp_name,sex,qualification,experience,desigination,salary,contact_no)
values(112,'prakash','male','mbbs',5,'doctor',5000,95625841);
```

```
mysql> insert into employee (emp_id,emp_name,sex,qualification,experience,desigination,salary,contact_no)
values(113,'ajith','male','mbbs',1,'doctor',10000,95625631);
```

```
mysql> insert into employee (emp_id,emp_name,sex,qualification,experience,desigination,salary,contact_no)
values(114,'ajith','male','nursing',1,'ward boy',1000,95225631);
```

## PATIENT:

```
mysql> insert into patient (patient_id,name,sex,date_admitted,date_discharge,contact_no) values
-> (2,'akash','male','2018/09/08','2018/10/08',564789321);
```

```
mysql> insert into patient (patient_id,name,sex,date_admitted,date_discharge,contact_no) values
-> (3,'ezil','male','2018/10/20','2018/09/07',685321497);
```

```
mysql> insert into patient (patient_id,name,sex,date_admitted,date_discharge,contact_no) values  
-> (4,'arasi','female','2018/09/01','2018/09/22',658932741)  
-> (1,'sathish','male','2018/10/01','2018/10/08',985674231);  
mysql> insert into patient (patient_id,name,sex,date_admitted,date_discharge,contact_no) values  
-> (5,'akask','male','2018/09/20','2018/10/01',65465545);
```

**ROOMS:**

```
mysql> insert into rooms(room_num,room_cost,pat_id,room_incharge) values(101,10000,1,114);  
Query OK, 1 row affected (0.49 sec)
```

```
mysql> insert into rooms(room_num,room_cost,pat_id,room_incharge) values(102,5000,3,115);
```

```
mysql> insert into rooms(room_num,room_cost,pat_id,room_incharge) values(103,15000,4,114);
```

**LAB:**

```
mysql> insert into lab(lab_num,lab_cost,pat_id,lab_incharge) values(201,10000,1,114);
```

```
mysql> insert into lab(lab_num,lab_cost,pat_id,lab_incharge) values(202,15000,4,115);
```

```
mysql> insert into lab(lab_num,lab_cost,pat_id,lab_incharge) values(203,10000,1,114);
```

**PAYMENT:**

```
insert into payment (payment_id,treatment_cost,lab_cost,room_cost,pat_id) values('p01',1000,20000,10000,2);
```

```
mysql> insert into payment (payment_id,treatment_cost,lab_cost,room_cost,pat_id) values('p02',1500,0,0,2);
```

```
mysql> insert into payment (payment_id,treatment_cost,lab_cost,room_cost,pat_id) values('p03',1000,0,0,3);
```

```
mysql> insert into payment (payment_id,treatment_cost,lab_cost,room_cost,pat_id)  
values('p04',1500,15000,15000,4);
```

```
mysql> insert into payment (payment_id,treatment_cost,lab_cost,room_cost,pat_id) values('p05',18000,0,0,5);\
```

## 8. NORMALIZATION FORM OF EACH SCHEMA

1NF:

In this schema we have composite attribute so that is deleted from its entity and made as separate table.

```
mysql> alter table patient drop column address;
mysql> create table address_of_patient(pat_id int,st_name varchar(20),area varchar(20),city varchar(20));
```

The screenshot shows a MySQL command-line interface window. The title bar says "MySQL 8.0 Command Line Client". The main area displays the following text:

```
ERROR 1062 (23000): Duplicate entry 'p02' for key 'PRIMARY'
mysql> desc address_of_patient;
+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+
| pat_id | int(11) | YES  | MUL  | NULL    |       |
| st_name | varchar(20) | YES  |       | NULL    |       |
| area   | varchar(20) | YES  |       | NULL    |       |
| city   | varchar(20) | YES  |       | NULL    |       |
+-----+-----+-----+-----+
4 rows in set (0.12 sec)

mysql>
```

2NF:

In table patient,

Patient\_id is the primary key which determines all the other non-prime attribute,

Patient\_id → Patient\_id

Patient\_id → name

Patient\_id → sex

Patient\_id → date\_discharge

Patient\_id → date\_admitted

Patient\_id → contact\_no

As same as in the employee table,

Emp\_id is the primary key which determines all the other non-prime attribute,

Emp\_id → Emp\_id

Emp\_id → emp\_name

Emp\_id → sex

Emp\_id → qualification

Emp\_id → experience

Emp\_id → designation

Emp\_id → salary

Emp\_id → contact\_no

As the primary attribute in the table is determines all other non-prime attributes respectively.

3NF:

This schema satisfies the both 1NF, 2NF and this schema does not have any transitive dependency. Hence this schema is in 3NF.

**9. WORK BREAK DOWN**

Team Member Registration Number	Name	Work Assigned
17BCS0082	SATHISH KUAMR	DATABASE CREATION, DOCUMENT CREATION
17BCS0022	EZILARASAN	ER DIAGRAM IMPLEMENTATION
17BCS0110	AKASH	NORMALISATION

**10. REVIEW EVALUATION**

COMPONENT	MARKS	MEMBER 1	MEMBER 2	MEMBER 3
REPORT				
ER DIAGRAM				
SCHEMA PREPARATION				
NORMALIZATION OF SCHEMA				
TABLE CREATION				
DATA MANIPULATION				
PRESENTATION				
REPORT PREPARATION				
TOTAL	100 MARKS			

