HOSPITAL MANAGEMENT SYSTEM

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Kavuri Sumana Priya

(AP20110010390)

Under the Guidance of

Prof. Dr Rajiv Senapati

Department of Computer Science Engineering

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[December, 2022]

Certificate

Date: 8-Dec-22

This is to certify that the work present in this Project entitled "**HOSPITAL MANAGEMENT SYSTEM**" has been carried out by **Kavuri Sumana Priya** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Co-supervisor

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

Acknowledgements

I express my sincere gratitude to **Dr. Rajiv Senapati** sir, my professor in charge, for his guidance and support in finishing the Database Management System project. His patience and enormous knowledge helped me to overcome many obstacles which occurred at each and every phase of this project. My overall experience while doing this project was just indescribable. Under his supervision I came to know about many things which I have always been curious about. I could not have imagined a better supervisor.

Table of Contents

Abstract

Value for data is getting increased day by day. And storing the data in an organized way is very crucial in order performs certain operations on it. Nowadays, especially during pandemic, hospitals are dealing with huge amount of data. For the hospitals to function properly and productively, they should store and manage the data related to patients in an efficient way. Hospital Management System project stores the details about hospital and its branches, available doctor's details, patient details, patient's medical history and room details. The steps involved in this project are requirement gathering, E-R diagram based on the details that are collected, conversion of E-R model into relational model, normalizing the tables obtained in the relational model, redrawing the relational model if new tables are created during the normalization process and finally the implementation. Main objective of this project is to maintain the data properly without any inconsistency.

1. Introduction

In order to be productive and to function properly a hospital needs to maintain their data. Maintaining data using outdated methods (File system) is very difficult. And there are many disadvantages of using file system. So, by using a database management system, we can manage the data in an efficient way and a lot of man power will be saved.

One of the applications of Database Management System is Hospital Management System. This project is built by considering the basic requirements for a hospital. Branches of a hospital, Details of doctors corresponding to that particular branch and details of patients along with their medical records are stored in this database. Patients can consult multiple doctors of different departments if they want. And the medical records belonging to different departments of a particular patient are kept separate. If required, then a room can be allocated for a patient. Bill generated for a patient will include doctor charges along with room charges if a room is allocated for him/her.

Methodology

2.1 Entity List

1. Hospital
2. Doctor
3. Patient
4. MedicalRecord
5. Bill
6. Room

2.2 Entities with corresponding attributes

Hospital

ATTRIBUTE	DATATYPE	CONSTRAINT
hospitalID	Varchar(30)	Primary Key
hospitalName	Varchar(100)	Not null
hospoitalPhoneNo	Varchar(15)	Not null

Doctor

ATTRIBUTE	DATATYPE	CONSTRAINT
doctorEmail	Varchar(150)	Primary Key
doctorFName	Varchar(30)	Not null
doctorLName	Varchar(30)	Not null
salary	Int	Not null
doctorGender	Varchar(7)	Not null
department	Varchar(100)	Not null

MedicalRecord

ATTRIBUTE	DATATYPE	CONSTRAINT
recordID	Varchar(30)	Primary Key
examinationDate	Varchar(10)	Not null
recordDept	Varchar(100)	Not null

Patient

ATTRIBUTE	DATATYPE	CONSTRAINT
patientEmail	Varchar(50)	Primary Key
patientFName	Varchar(30)	Not null
patientLName	Varchar(30)	Not null
patientGender	Varchar(7)	Not null
patientDOB	Varchar(10)	Not null
patientBloodGroup	Varchar(12)	Not null
patientAppointmentTime	Varchar(8)	Not null

Bill

ATTRIBUTE	DATATYPE	CONSTRAINT
billNumber	Varchar(30)	Primary Key
doctorCharges	Int	Not null
roomCharges	Int	---

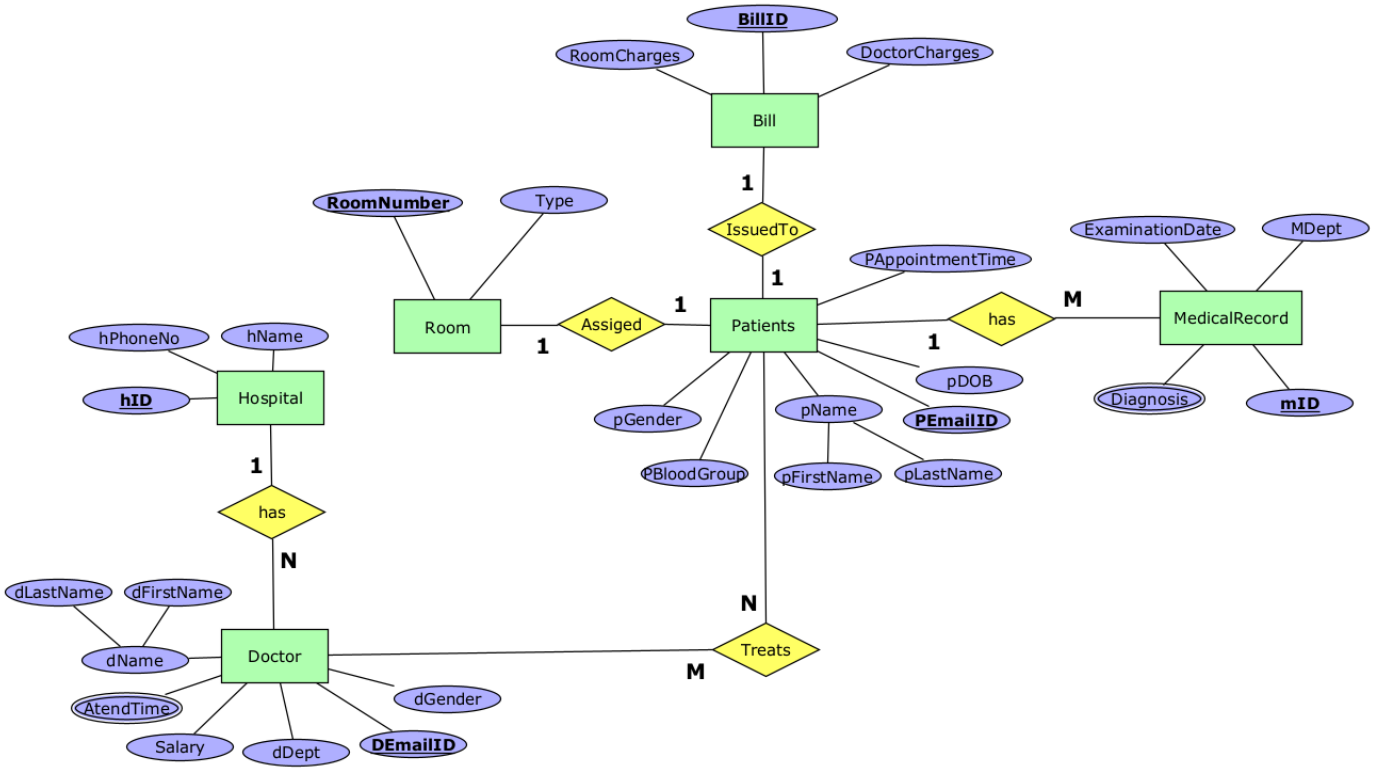
Room

ATTRIBUTE	DATATYPE	CONSTRAINT
roomNumber	Varchar(30)	Primary Key
roomType	Varchar(10)	Not null

2.3 Cardinalities

Entity 1	Entity 2	Relation	Cardinality
Hospital	Doctor	Has	1:M
Doctor	Patient	Treats	M:M
Patient	MedicalRecord	Has	1:M
Bill	Patient	Issued to	1:1
Patient	Room	Assigned	1:1

2.4 Entity Relationship Model (ER Model)



ER Diagram for Hospital Management System

2.5 ER model to Relational model

In relational model data is represented in the form of tables with each table having certain number of rows and columns.

Rules to be followed while converting entities in ER model to tables in relational model are as follows

1. An entity in the ER model is represented by a relational table in the relational model.

2. All the attributes of the ER model are represented in different columns as an attribute in relational model.

3. Primary key attribute of the ER model is represented as the primary key in relational model.

4. Composite key of the ER model is split into different columns in the relational model.

5. Derived attributes must be dropped in the relational model.

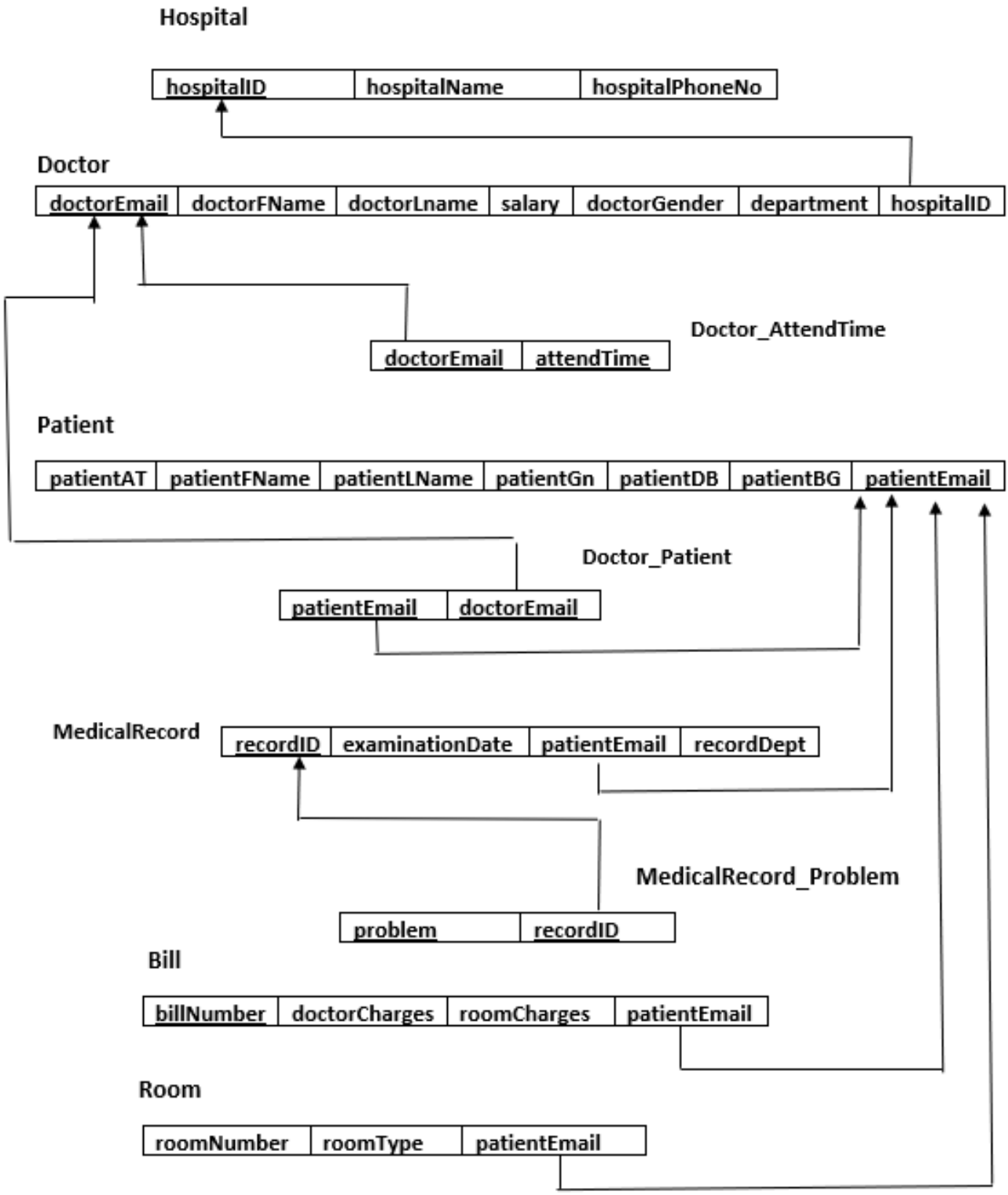
6. Multi valued attributes need to be kept separate in a new table.

Rules to be followed while converting relationships in ER model to tables in relational model are as follows

7. If the relationship between 2 entities is 1:1, then the primary key attribute of one relation becomes foreign key in another relation.

8. If the relationship between 2 entities is 1:M or M:1, then the primary key attribute of one-sided relation becomes foreign key in many-sided relation.

9. If the relationship between 2 entities is M:M, then a new table needs to be created to represent that relation, where the new relation will have primary key of both relations as the foreign keys.


Hospital
hospitalID
hospitalName
hospitalPhoneNo
Doctor
doctorEmail
doctorFName
doctorLname
salary
doctorGender
department
hospitalID
Doctor_AttendTime
doctorEmail
attendTime
Patient
patientAT
patientFName
patientLName
patientGn
patientDB
patientBG
patientEmail
Doctor_Patient
patientEmail
doctorEmail
MedicalRecord
recordID
examinationDate
patientEmail
recordDept
MedicalRecord_Problem
problem
recordID
Bill
billNumber
doctorCharges
roomCharges
patientEmail
Room
roomNumber
roomType
patientEmail

2.6 Normalization

Normalization is a process of analysing and decomposing the complex relations which satisfies some constraints to form a simple relation.

There are 5 types of normal forms:

1. First Normal Form (1NF)

2. Second Normal Form (2NF)

3. Third Normal Form (3NF)

4. Fourth Normal Form (4NF)

5. Fifth Normal Form (5NF)

First Normal Form (1NF)

A relation is said to be in 1NF if all the attributes in the relation are atomic in nature.

- All the attributes in the above relational model are atomic in nature (All the attributes cannot be further sub divided). So, the above relational model is in 1NF.

Second Normal Form (2NF)

A relation in 1NF is said to be in 2NF if it satisfies any one of the following conditions. They are,

1. The primary key consists of only one attribute.

2. There exists no non key attribute.

3. Every non-key attribute present in the relation should functionally depends upon full set of primary key.

- The tables Hospital, Doctor, Patient, MedicalRecord, Bill and Room has a primary key which consists of only one attribute.
- The tables Doctor_AttendTime, Doctor_Patient, MedicalRecord_Problem have no non key attribute.
- All the tables in the relational model are satisfying any one of the above conditions.
- So, the above relational model is in 2NF.

Third Normal Form(3NF)

A relation which is in 2NF is said to be in 3NF, if there exists no transitive functional dependency of any non-key attribute on the set of primary key.

- There are no transitive functional dependencies in the above relational model. So, it is in 3NF.

2.7 SQL Code

/* Creating a database */

```
create database hms;
```

/* Activate the database */

```
use Employee;
```

/* Create 'Hospital' table */

```
create table Hospital(
hospitalID varchar(30) primary key,
hospitalName varchar(100) not null,
hospitalPhoneNo varchar(15) not null
);
```

Field	Type	Null	Key	Default	Extra
hospitalID	varchar(30)	NO	PRI	NULL	
hospitalName	varchar(100)	NO		NULL	
hospitalPhoneNo	varchar(15)	NO		NULL	

/* Insert values into 'Hospital' table */

```
insert into Hospital values("HOS-VIZ-1288","Raghavendra Hospital","0891-
2525617"),
("HOS-HYD-229","Raghavendra Hospital","0891-2526500"),
```

("HOS-DIS-488","Raghavendra Hospital","0891-2704055");

hospitalID	hospitalName	hospitalPhoneNo
HOS-DIS-488	Raghavendra Hospital	0891-2704055
HOS-HYD-229	Raghavendra Hospital	0891-2526500
HOS-VIZ-1288	Raghavendra Hospital	0891-2525617
NULL	NULL	NULL

/* Create 'Doctor' table */

```
create table Doctor(
doctorEmail varchar(150) primary key,
doctorFName varchar(30) not null,
doctorLName varchar(30) not null,
salary int not null,
doctorGender varchar(7) not null,
department varchar(100) not null,
hospitalID varchar(30) not null,
foreign key (hospitalID) references Hospital (hospitalID) on delete cascade
);
```

Field	Type	Null	Key	Default	Extra
doctorEmail	varchar(150)	NO	PRI	NULL	
doctorFName	varchar(30)	NO		NULL	
doctorLName	varchar(30)	NO		NULL	
salary	int	NO		NULL	
doctorGender	varchar(7)	NO		NULL	
department	varchar(100)	NO		NULL	
hospitalID	varchar(30)	NO	MUL	NULL	

/* Insert values into 'Doctor' table */

```
insert into Doctor
values("rama_p@gmail.com","Rama","Perli",100000,"Female","General
Surgery","HOS-VIZ-1288"),
("kumar1205@gmail.com","Kumar","Koduru",100000,"Male","Orthopedics","HOS-
VIZ-1288"),
("ramya02@gmail.com","Ramya","Kavali",120000,"Female","Obstetrics &
Gynaecology","HOS-VIZ-1288");
```

doctorEmail	doctorFName	doctorLName	salary	doctorGender	department	hospitalID
kumar1205@gmail.com	Kumar	Koduru	100000	Male	Orthopedics	HOS-VIZ-1288
rama_p@gmail.com	Rama	Perli	100000	Female	General Surgery	HOS-VIZ-1288
ramya02@gmail.com	Ramya	Kavali	120000	Female	Obstetrics & Gynaecology	HOS-VIZ-1288
NULL	NULL	NULL	NULL	NULL	NULL	NULL

/* Create 'Doctor_AttendTime' table */

```
create table Doctor_AttendTime(
doctorEmail varchar(150) not null ,
attendTime varchar(10) not null,
foreign key (doctorEmail) references Doctor (doctorEmail) on delete cascade,
primary key(doctorEmail,attendTime)
);
```

Field	Type	Null	Key	Default	Extra
doctorEmail	varchar(150)	NO	PRI	NULL	
attendTime	varchar(10)	NO	PRI	NULL	

/* Insert values into 'Doctor_AttendTime' table*/

```
insert into Doctor_AttendTime values ("rama_p@gmail.com","10:55AM"),
("rama_p@gmail.com","12:55PM"),
("kumar1205@gmail.com","11:30AM"),
("ramya02@gmail.com","11:30AM");
```

doctorEmail	attendTime
kumar1205@gmail.com	11:30AM
rama_p@gmail.com	10:55AM
rama_p@gmail.com	12:55PM
ramya02@gmail.com	11:30AM
NULL	NULL

/* Create 'Patient' table */

```
create table Patient(
patientEmail varchar(50) primary key,
patientFName varchar(30) not null,
patientLName varchar(30) not null,
```

patientGender varchar(7) not null,

patientDOB varchar(10) not null,

patientBloodGroup varchar(12) not null,

patientAppointmentTime varchar(8) not null

);

Field	Type	Null	Key	Default	Extra
patientEmail	varchar(50)	NO	PRI	NULL	
patientFName	varchar(30)	NO		NULL	
patientLName	varchar(30)	NO		NULL	
patientGender	varchar(7)	NO		NULL	
patientDOB	varchar(10)	NO		NULL	
patientBloodGroup	varchar(12)	NO		NULL	
patientAppointmentTime	varchar(8)	NO		NULL	

/* Insert values into 'Patient' table*/

insert into Patient values ("priya@gmail.com","Priya","Katuri","Female","03-12-1998","O+ve","10:55Am"),

("kiran@gmail.com","Kiran","Kolli","Male","21-5-1991","B+ve","12:55PM"),

("Devi@gmail.com","Devi","Kolli","Female","15-1-2000","O-ve","11:30Am"),

("ravali@gmail.com","Ravali","Makineni","Female","15-2-1995","AB+ve","11:30AM");

patientEmail	patientFName	patientLName	patientGender	patientDOB	patientBloodGroup	patientAppointmentTime
Devi@gmail.com	Devi	Kolli	Female	15-1-2000	O-ve	11:30Am
kiran@gmail.com	Kiran	Kolli	Male	21-5-1991	B+ve	12:55PM
priya@gmail.com	Priya	Katuri	Female	03-12-1998	O+ve	10:55Am
ravali@gmail.com	Ravali	Makineni	Female	15-2-1995	AB+ve	11:30AM
NULL	NULL	NULL	NULL	NULL	NULL	NULL

/* Create 'Doctor_Patient' table */

create table Doctor_Patient(

doctorEmail varchar(50) not null,

patientEmail varchar(50) not null,

foreign key (doctorEmail) references Doctor (doctorEmail) on delete cascade,

foreign key (patientEmail) references Patient (patientEmail) on delete cascade,

primary key (doctorEmail,patientEmail)

);

	Field	Type	Null	Key	Default	Extra
▶	doctorEmail	varchar(50)	NO	PRI	NULL	
	patientEmail	varchar(50)	NO	PRI	NULL	

/* Insert values into 'Doctor_Patient' table*/

insert into Doctor_Patient values ("rama_p@gmail.com","priya@gmail.com"),

("rama_p@gmail.com","kiran@gmail.com"),

("kumar1205@gmail.com","Devi@gmail.com"),

("ramya02@gmail.com","ravali@gmail.com");

	doctorEmail	patientEmail
▶	kumar1205@gmail.com	Devi@gmail.com
	rama_p@gmail.com	kiran@gmail.com
	rama_p@gmail.com	priya@gmail.com
	ramya02@gmail.com	ravali@gmail.com
*	NULL	NULL

/* Create 'MedicalRecord' table */

create table MedicalRecord(

recordID varchar(30) primary key,

examinationDate varchar(10) not null,

patientEmail varchar(50) not null,

recordDept varchar(100) not null,

foreign key (patientEmail) references Patient (patientEmail) on delete cascade

);

	Field	Type	Null	Key	Default	Extra
▶	recordID	varchar(30)	NO	PRI	NULL	
	examinationDate	varchar(10)	NO		NULL	
	patientEmail	varchar(50)	NO	MUL	NULL	
	recordDept	varchar(100)	NO		NULL	

/* Insert values into 'MedicalRecord' table*/

insert into MedicalRecord values ("M12","08-12-2022","priya@gmail.com","General Surgery"),

("M13","08-12-2022","kiran@gmail.com","General Surgery"),

("M14","08-12-2022","Devi@gmail.com","Orthopedics"),

("M15","08-12-2022","ravali@gmail.com","Obstetrics & Gynaecology");

recordID	examinationDate	patientEmail	recordDept
M12	08-12-2022	priya@gmail.com	General Surgery
M13	08-12-2022	kiran@gmail.com	General Surgery
M14	08-12-2022	Devi@gmail.com	Orthopedics
M15	08-12-2022	ravali@gmail.com	Obstetrics & Gynaecology
NULL	NULL	NULL	NULL

/* Create 'MedicalRecord_Problem' table */

```
create table MedicalRecord_Problem(
problem varchar(50) not null,
recordID varchar(30)  not null,
foreign key (recordID) references MedicalRecord (recordID) on delete cascade,
primary key(problem,recordID)
);
```

Field	Type	Null	Key	Default	Extra
problem	varchar(50)	NO	PRI	NULL	
recordID	varchar(30)	NO	PRI	NULL	

/* Insert values into 'MedicalRecord_Problem' table*/

```
insert into MedicalRecord_Problem values("Abdominal Pain","M12"),
("Hernia","M13"),
("Joint Pain","M14"),
("Adenomyosis","M15");
```

problem	recordID
Abdominal Pain	M12
Hernia	M13
Joint Pain	M14
Adenomyosis	M15
NULL	NULL

/* Create 'Bill' table */

```
create table Bill(
billNumber varchar(30) primary key,
doctorCharges int not null,
roomCharges int,
patientEmail varchar(50) not null,
foreign key (patientEmail) references Patient (patientEmail) on delete cascade
);
```

Field	Type	Null	Key	Default	Extra
billNumber	varchar(30)	NO	PRI	NULL	
doctorCharges	int	NO		NULL	
roomCharges	int	YES		NULL	
patientEmail	varchar(50)	NO	MUL	NULL	

/* Insert values into 'Bill' table*/

```
insert into Bill values ("B12",1000,15000,"priya@gmail.com"),
("B13",1000,10000,"kiran@gmail.com"),
("B14",1000,0,"Devi@gmail.com"),
("B15",1000,0,"ravali@gmail.com");
```

billNumber	doctorCharges	roomCharges	patientEmail
B12	1000	15000	priya@gmail.com
B13	1000	10000	kiran@gmail.com
B14	1000	0	Devi@gmail.com
B15	1000	0	ravali@gmail.com
NULL	NULL	NULL	NULL

/* Create 'Room' table */

```
create table Room(
roomNumber varchar(30) primary key,
roomType varchar(10) not null,
patientEmail varchar(50) not null,
foreign key (patientEmail) references Patient (patientEmail) on delete cascade
);
```

	Field	Type	Null	Key	Default	Extra
▶	roomNumber	varchar(30)	NO	PRI	NULL	
	roomType	varchar(10)	NO		NULL	
	patientEmail	varchar(50)	NO	MUL	NULL	

/* Insert values into 'Room' table*/

insert into Room values("N201","AC","priya@gmail.com"),

("N202","Non AC","kiran@gmail.com");

	roomNumber	roomType	patientEmail
▶	N201	AC	priya@gmail.com
	N202	Non AC	kiran@gmail.com
*	NULL	NULL	NULL

2.8 Queries

Query 1: Get the doctor's first name, patient's first name along with the appointment time.

SQL Statement:

select Doctor.doctorFName, Patient.patientFName, Patient.patientAppointmentTime from Doctor join Doctor_Patient join Patient

where Doctor.doctorEmail=Doctor_Patient.doctorEmail and Doctor_Patient.patientEmail=Patient.patientEmail;

Output:

	doctorFName	patientFName	patientAppointmentTime
▶	Kumar	Devi	11:30Am
	Rama	Kiran	12:55PM
	Rama	Priya	10:55Am
	Ramya	Ravali	11:30AM

Query 2: Get the list of doctor's email ID's who are working in the hospital whose ID is "HOS-VIZ-1288"

SQL Statement:

select Hospital.hospitalID,Doctor.doctorEmail from Hospital join Doctor

where Hospital.hospitalID=Doctor.hospitalID and Hospital.hospitalID="HOS-VIZ-1288";

Output:

hospitalID	doctorEmail
HOS-VIZ-1288	kumar1205@gmail.com
HOS-VIZ-1288	rama_p@gmail.com
HOS-VIZ-1288	ramya02@gmail.com

Query 3: Get the doctor's first name along with the medical records that were examined by them.

SQL Statement:

select Doctor.doctorFName,Doctor.department, MedicalRecord.recordID from Doctor join MedicalRecord

where Doctor.department=MedicalRecord.recordDept;

Output:

doctorFName	department	recordID
Rama	General Surgery	M12
Rama	General Surgery	M13
Kumar	Orthopedics	M14
Ramya	Obstetrics & Gynaecology	M15

References

1. https://www.karexpert.com/blogs/what-is-hospital-management-system/
2. https://www.edrawsoft.com/article/er-diagrams-for-hospital-management-system.html
3. https://www.edrawmax.com/templates/1008526/