Business Data Management Course Project Paper

On

Designing and Developing a Hospital-Based Database System

Department of Information Technology

Submitted By:-

1. Abhinav Singh	169005150
2. Jing Li	168006176
3. Suchendra kuchipudi	168002998

Project documentation submitted to fulfill the requirements for the class of Data Business Management and in Partial fulfillment of the requirements for Master's Degree in Information Technology.

Date: - December 4, 2016

Session: - Fall '16

Table of Contents

PREFACE

Our course project is about Designing and Developing a Hospital-Based Database System. It forms a basic entity of the management of a Hospital. Hence, it is very important for the system to be reliable, user friendly, and should be properly functional for a long time without cropping up of any errors.

To start with the system study we analyzed various hospital case studies as well how database systems in hospitals work and what's the need. We saw these system, studied it and tried to develop a better system. Our system is aimed on achieving an automated system for Hospital Management. This gave us the idea of the different fields that ought to be in a Hospital Management System such as patient registration, his/her advance payment, the records, the details etc. and also how a software system can make the work easy both for the hospital staff and the patients. Moreover, the evaluation helped us to arrive at the conclusion that the automated software is far more superior to the manual ones.

ABSTRACT

Our motive is to develop a database management system that is very much user friendly and easy to gather information in a very short time. We try to make our software reliable and comfortable.

As per our project documentation is on Designing and Developing a Hospital Management System we divide our work into two basic parts:
Designing part and Developing Part.

® We give a flow chart on our work division in project overview part.

Chapter 1 → Introduction

In this chapter we discuss the definition of Database and its usefulness. We also describe the reason to take HOSPITAL MANAGEMENT SYSTEM as our thesis work.

Chapter 2 → Designing the Database System

In this chapter we describe the entities and attributes. We draw the Entity Relationship Diagram (ERD) and Tables. We determine the attributes of tables and its data types. We also find functional dependencies and normalize all the tables.

Chapter 3 → Developing the Database System using MySQL.

In this chapter we implement our database in MySQL and finally execute some complex queries on the system to analyze the information stored and tests it reliability and consistency.

Chapter 4 → Conclusion and Future Work.

We tried to Save, Delete and Update data using MySQL interface and we also tried to use Trigger in SQL Server but we cannot complete them. So we include it as a part of future work.

INTRODUCTION

- ❖ What is a Database?

  - ✓ A Database is a collection of records which are stored on a computer; a database organizes the data according to database models such as a relational model.

- ❖ Why do we need Databases?

  - ✓ Databases collect items on which the user can carry out various operations such as viewing, navigating, creating tables, and searching. Databases can be seen as a symbolic form of the computer age.

We use databases for these reasons. Such as,

1. We use database because we can easily manipulate, edit or delete data.
2. Data are kept organized in a database so we can easily retrieve data.
3. Easy to find out desired data.
4. Data are secured.

Advantages of Database

- ❖ Reduced Data Redundancy.
- ❖ Reduced updating errors and increased consistency.
- ❖ Greater data integrity and independence from applications programs.
- ❖ Improved data access to users through use of host and query languages.
- ❖ Improved data security.
- ❖ Reduced data entry, storage, and retrieval costs.
- ❖ Facilitated development of new application programs.

In our project report of Designing and Developing a Hospital-Based Database System we can see two basic parts.

- ✓ Designing
- ✓ Developing

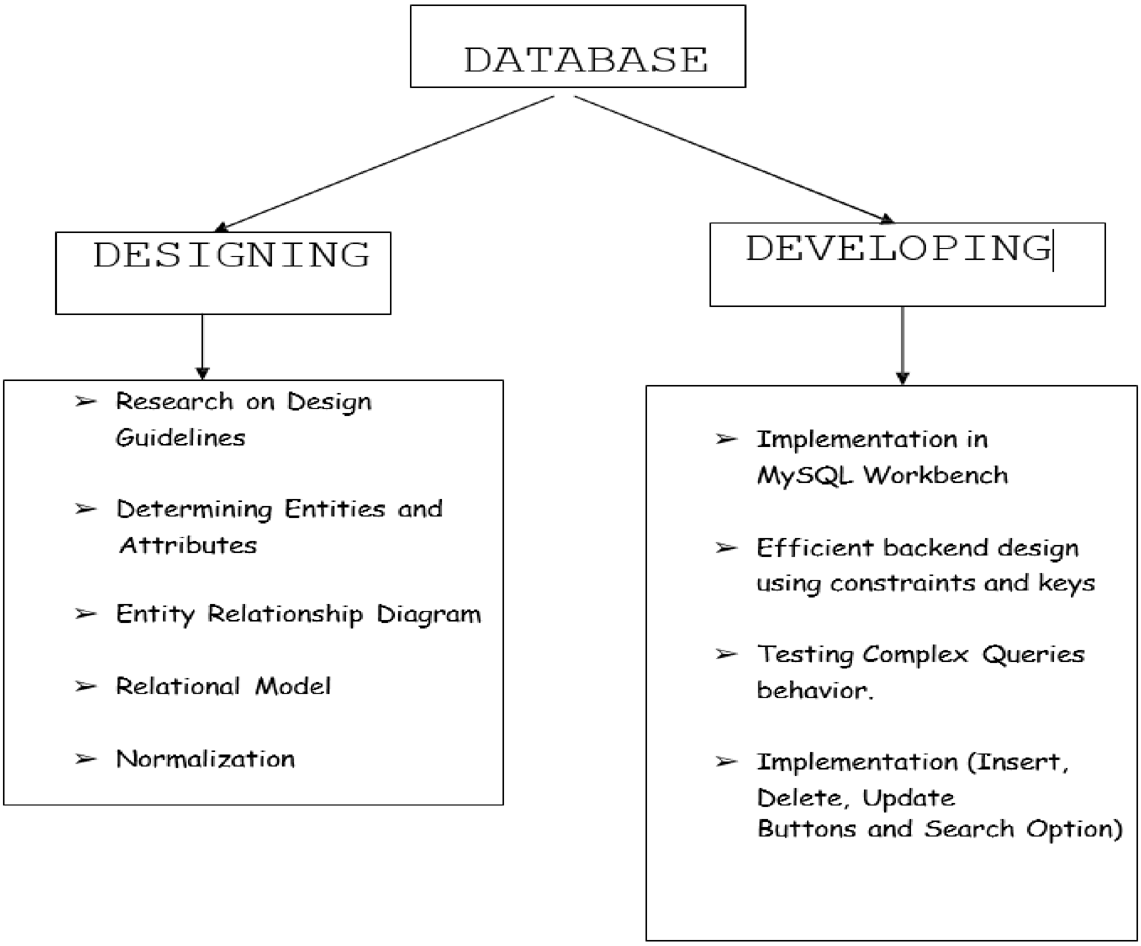
# Choosing Hospital Management System for our thesis

We study and select three systems at first. The systems were:

- Banking System
- Computer Sales Management System
- Hospital Management System

We researched the demos and workings of the respective systems from different sources and all the group members decided to do the project on Hospital Management System because the system is less complex and easy to study. Most Banking Systems and Computer Sales Management Systems are controlled using online based software where users can access from any part of the country. But we are determined to make management system for internal usage desktop based software. So we decided to choose Hospital Management System based on our research and knowledge or organizational database systems. We try our best to make the system efficient and reliable with the help of our design.

# Thesis Overview


DATABASE
DESIGNING
DEVELOPING
Research on Design Guidelines
Determining Entities and Attributes
Entity Relationship Diagram
Relational Model
Normalization
Implementation in MySQL Workbench
Efficient backend design using constraints and keys
Testing Complex Queries behavior.
Implementation (Insert, Delete, Update Buttons and Search Option)

CHAPTER 2

DESIGNING THE DATABASE SYSTEM

2.1 Determining Entities and Attributes

❖ Entity

- ✓ An entity is something that has a distinct, separate existence, though it need not be a material existence. In particular, abstractions and legal fictions are usually regarded as entities. In general, there is also no presumption that an entity is animate. Entities are used in system developmental models that display communications and internal processing of, say, documents compared to order processing.
- ✓ An entity could be viewed as a set containing subsets.
- ✓ A DBMS entity is either a thing in the modeled world or a drawing element in an Entity Relationship Diagram(ERD)

❖ Attribute

- ✓ An attribute is a specification that defines a property of an object, element, or file. It may also refer to or set the specific value for a given instance of such.
- ✓ Attributes should more correctly be considered metadata. It is frequently and generally a property of an entity.
- ✓ An attribute of an object usually consists of a name and a value; of an element, a type or class name; of a file, a name and extension.

❖ Data Type

- ✓ A data type (or datatype): In programming, a classification identifying one of various types of data, as floating-point, integer, or Boolean, stating the possible values for that type, the operations that can be done on that type, and the way the values of that type are stored.
- ✓ We think our best and determine the entities and attributes for our Database System. The Entities and Attributes are given below.

2.2 Entity Relationship Diagram (ERD):

We draw the Entity Relationship Diagram (ERD) very carefully and efficiently for the whole system of Hospital.

We were able to cover all main information required in the HMS in our ERD.

The ERD is given below:

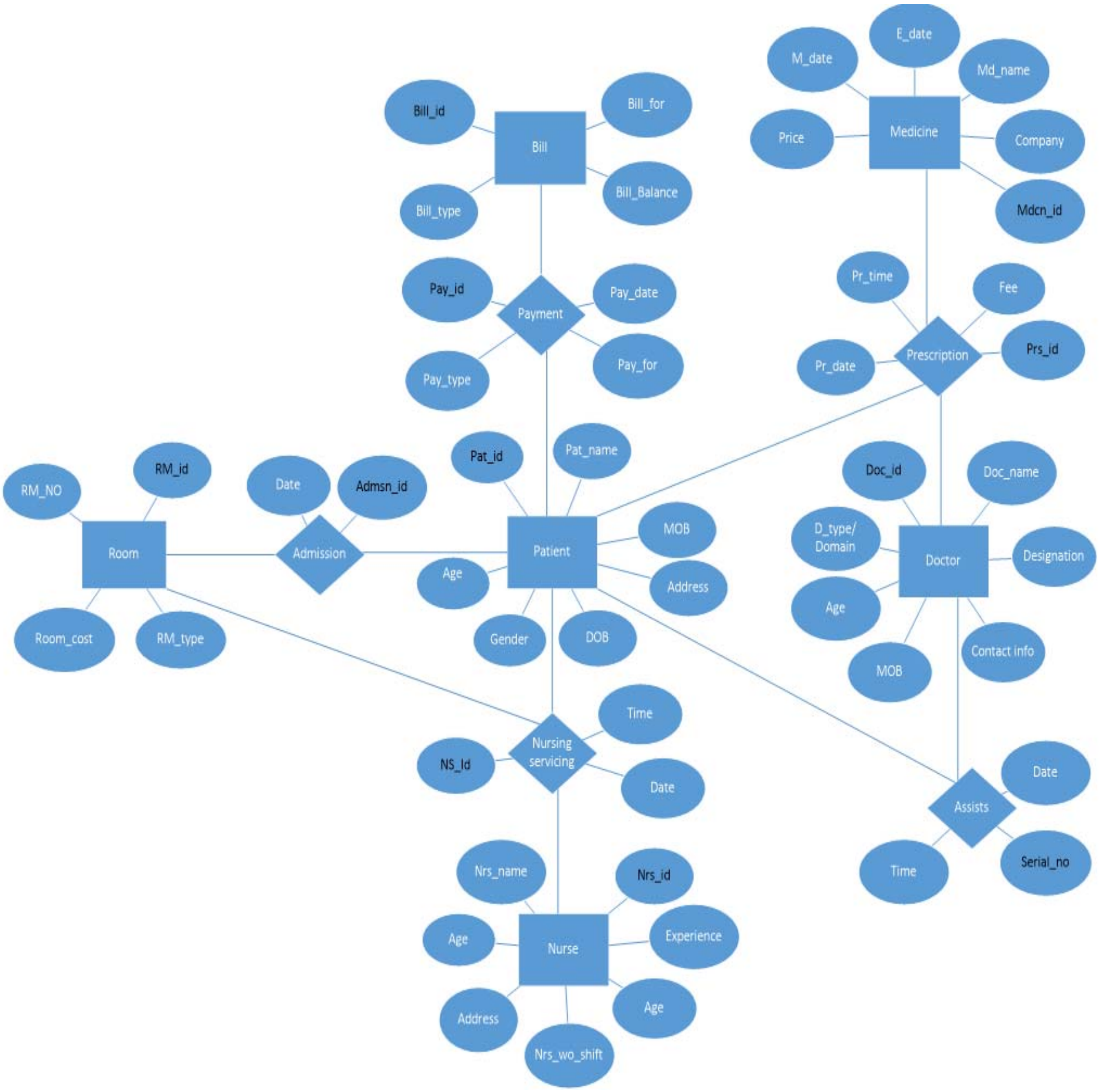


Fig1: Entity Relationship Diagram (ERD).

2.3 Relational Model:

After completing the ERD successfully we made the relational model (table schemas) taking into account all the entities and the relationships.

Patient Table:-

Pat_id	Pat_name	Age	Gender	Address	DOB	MOB

Room Table:-

Room_id	Room_No	Room_type	Room_cost

Admission Table:-
This is a junction table between Patient & Room tables.

Admsn_id	Pat_id	Room_id	date	time

Doctor Table:-

Doc_id	Doc_name	Doc_type	Designation	Passed_from	Age	Address	MOB	Salary

Payment Table:-
This is a junction table between Patient & Bill Tables.

Pay_id	Bill_id	Pat_id	Pay_type	Pay_date

Bill Table:-

Bill_id	Bill_for	Bill_type	Bill_total

Medicine Table:-

Mdcn_id	Mdcn_name	Company	m_date	e_date	price

Prescription Table: -
This is a junction table between Patients, Doctor & Medicine tables.

Prs_id	Doc_id	Mdcn_id	Pat_id	Date	Time	Fee

Assist Table:-
This is a junction table between Patient & Doctor tables.

Serial_no	Pat_id	Doc_id	Time	Date

Nurse Table:-

Nrs_id	Nrs_name	Age	Address	MoB	Nrs_wo_shift	experience	Salary

Nursing_Service Table:-
This is a junction table between Patients, Room & Nurse tables.

Ns_id	Pat_id	Nrs_id	Room_id	Date	Time

2.3.1 Relational Tables' Descriptions

Patient table

Attributes	Data type	Comments
Pat_id	int	Unique id for a Patient
Pat_name	varchar(45)	Patient's Name
Age	int	Patient's Age
Gender	varchar(6)	Patient is Male or Female
Address	varchar(45)	Patient's Address
Dob	Date	Date of Birth
Mob	int	Mobile Number

Room table

Attributes	Data type	Comments
Room_id	int	Unique id for a Room
Room_no	varchar(20)	Room number
Room_type	varchar(20)	Room is VIP or Normal
Room_cost	int	Cost of the Room

Admission table

Attributes	Data type	Comments
Admsn_id	int	Unique id for an Admission
Pat_id	int	Unique id for a Patient

Room_id	int	Unique id for a Room
Date	Date	Date of Admission
Time	Time	Time of Admission

Doctor table:

Attributes	Data type	Comments
Doc_id	int	Unique id for a Doctor
Doc_name	varchar(45)	Doctor’s name
Doc_type	varchar(20)	Doctor’s specialty
Age	int	Doctor’s age
Address	varchar(45)	Doctor’s address
Mob	int	Mobile Number
Designation	varchar(20)	Doctor’s designation
Passed_from	varchar(20)	Doctor is passed from which medical college
Salary	int	Salary of a doctor

Bill table:

Attributes	Data type	Comments
Bill_id	int	Unique id for a Bill
Bill_for	varchar(20)	Purpose of the bill
Bill_type	varchar(20)	Bill either in Cash or Check
Bill_total	int	Total amount

Payment table:

Attributes	Data type	Comments
Pay_id	int	Unique id for a Payment
Bill_id	int	Unique id for a Bill
Pat_id	int	Unique id for a Patient
Pay_type	varchar(20)	Payment in Cash or Check
Pay_date	Date	Date of Payment

Medicine table:

Attributes	Data type	Comments
Mdcn_id	int	Unique id for a Medicine
Mdcn_name	varchar(20)	Medicine’s Name
company	varchar(20)	Medicine’s Company
M_date	Date	Manufacture Date
E_date	Time	Expire Date
price	int	Price of the Medicine

Prescription table:

Attributes	Data type	Comments
Prs_id	int	Unique id for a Prescription
Doc_id	int	Unique id for a Doctor
Mdcn_id	int	Unique id for a Medicine
Pat_id	int	Unique id for a Patient
Date	Date	Date of the Prescription

Time	Time	Time of the Prescription
Fee	int	Prescription Fees

Assist table:

Attributes	Data type	Comments
Serial_no	int	Unique id for an Assisted Test directed to a Patient by a Doctor
Pat_id	int	Unique id for a Patient
Doc_id	int	Unique id for a Doctor
Date	Date	Date of the Assisted Test
Time	Time	Time of the Assisted Test

Nurse table:

Attributes	Data type	Comments
Nrs_id	int	Unique id for a Nurse
Nrs_name	varchar(45)	Nurse's Name
Age	int	Nurse's age
Address	varchar(45)	Nurse's Address
Mob	int	Mobile Number
Nrs_wo_shift	varchar(20)	Nurse working Shift example morning, day, evening, night
Experience	varchar(20)	Nurse's Experience
salary	int	Salary of a Nurse

Nursing_Service table:

Attributes	Data type	Comments
Ns_id	int	Unique id for a Nursing Service
Pat_id	int	Unique id for a Patient
Nrs_id	int	Unique id for a Nurse
Room_id	int	Unique id for a Room
Date	Date	Date of Nursing Service
Time	Time	Time of Nursing Service

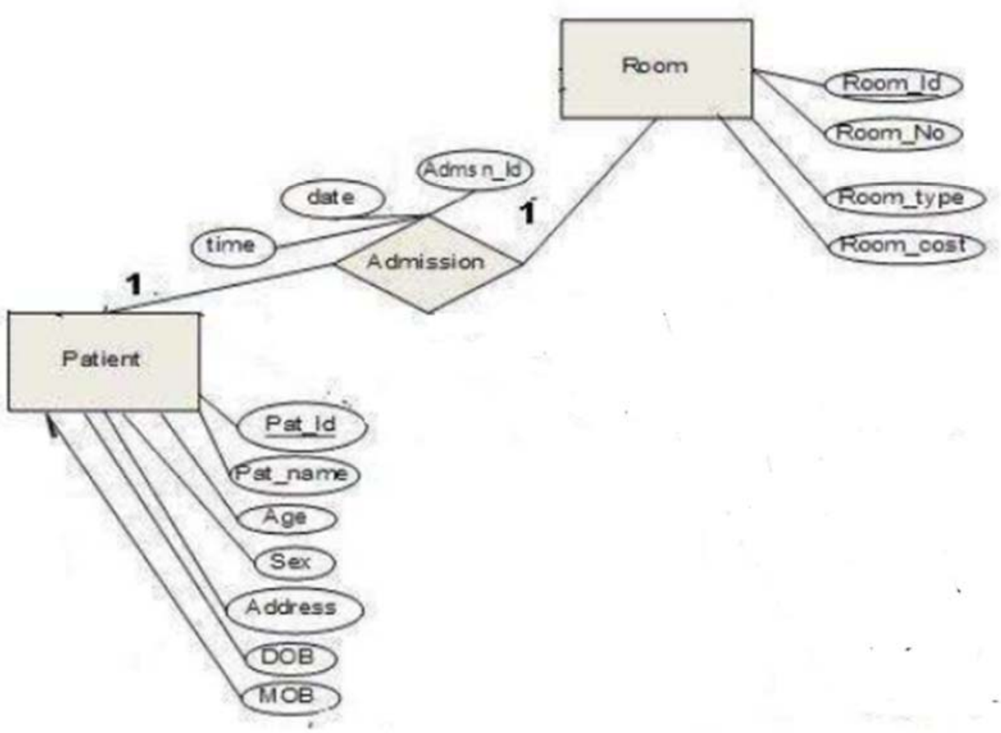
Note:

Gender will accept only "M" and "F" values.

Nrs_wo_shift will accept only "M", "A" and "N" values for the respective shifts

2.3.2 Explanation of Relational Model

#Relationship between Patient and Room Entities in the ER Model:



- ❖ 1 Patient can be admitted in 1 Room in a certain date and time.
- ❖ In 1 Room, only 1 Patient is admitted in a certain date and time.

  So the relationship is a Binary Relationship named Admission (in the diamond) with cardinality ratio from Patient to Room as 1 to 1.
- ❖ **Room here means, private Room for special care. Room doesn't mean ward**

Patient and Room Entities become Patient and Room tables.

Patient Table:-

Pat_id	Pat_name	Age	Sex	DOB	MOB	Address

Room Table:-

Room_id	Room_No	Room_type	Room_cost

The junction Admission also becomes a table.

Admission Table:-

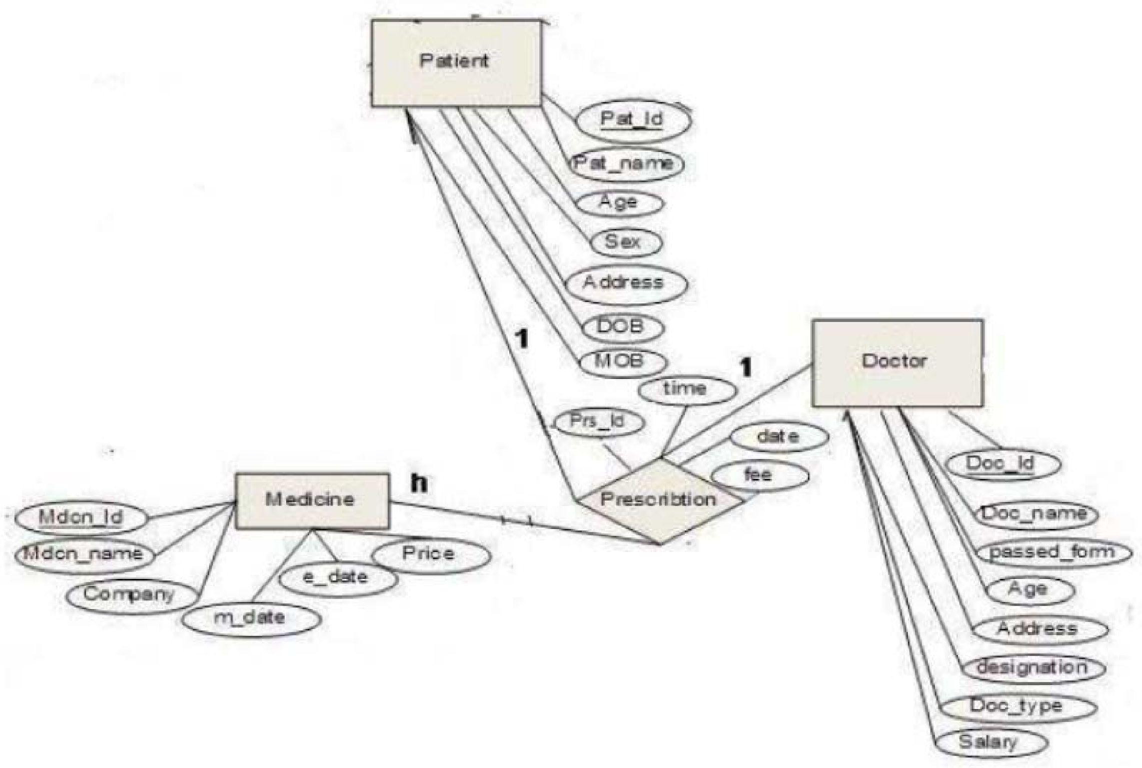
Admsn_id	Pat_id	Room_id	Date	time

- Primary Key of the Patient Table goes to Admission Table as Foreign Key.
- Primary Key of the Room Table goes to Admission Table as Foreign Key.

Since the Cardinality Ratio from Patient to Room is 1 to 1, admsn_id is a Primary key in the Admission Table. Pat_id from Patient Table and Room_id from Room Table become Foreign Keys in the Admission Table.

In a similar way relational tables have been designed for Patient – Assists – Doctor relationships with cardinality ratio 1 to 1.

# Relationship between Patient, Doctor and Medicine Entities in the ER Model:



- ❖ 1 Doctor gives 1 patient 1 or more medicine.
- ❖ 1 patient takes 1 medicine prescribed by 1 doctor.
- ❖ 1 medicine is prescribed by 1 doctor to 1 patient.

So the relationship is a Ternary Relationship named Prescription (in the diamond) with a Cardinality Ratio from Patient to Doctor to Medicine 1 to 1 to many. Patient, Doctor and Medicine Entities become Patient, Doctor and Medicine tables.

Patient Table:-

Pat_Id	Pat_name	Age	Sex	DOB	MOB	Address

Doctor Table:-

Doc_id	Doc_name	Doc_type	Designation	Age	Address	MOB	Passed_from	Salary

Medicine Table:-

Mdcn_id	Mdcn_name	company	m_date	e_date	price

Prescription Table: -
This is a junction table between Patients, Doctor & Medicine Table.

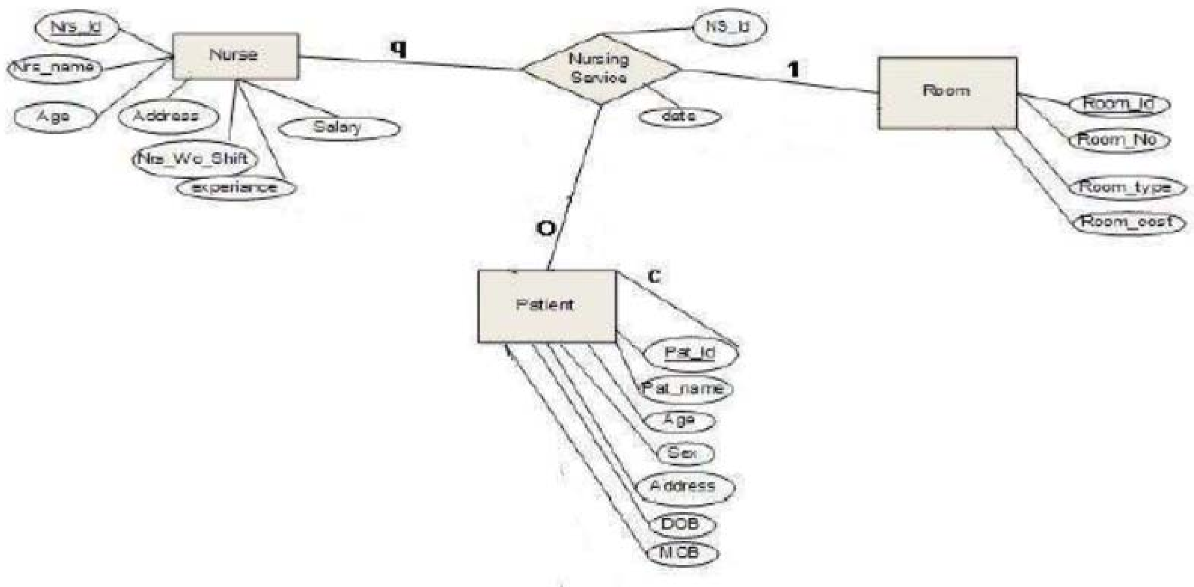
Prs_id	Doc_id	Mdcn_id	Pat_id	date	fee

- Primary Key of the Patient Table goes to Prescription Table as Foreign Key.
- Primary Key of the Doctor Table goes to Prescription Table as Foreign Key.
- Primary Key of the Medicine Table goes to Prescription Table as part of Primary Key.

Since the Cardinality Ratio from Patient to Doctor to Medicine 1 to 1 to M, Prs_id is a Primary key in the Prescription Table. Pat_id from Patient Table, Doc_id from Doctor Table become Foreign key and Mdcn_id from Medicine Table become Primary Key in the Prescription Table.

Same concept can be applied for defining primary and foreign keys in Patient - Payment_Relation – Room where 1 to M many relationships.

# Relationship between Patient, Room & Nurse Entities in the ER Model:-



- ❖ 1 room is fixed for 1 Patient to provide nursing service for 1 or Many nurses in a certain date.
- ❖ 1 patient receives nursing service from 1 Nurse in 1 Room in a certain date.
- ❖ 1 nurse can render proper services in 1 room to many patients in a certain date.

So it is a Ternary Relationship named Nursing Services (in the diamond) with cardinality Ratio from Room to Nurse to Patient 1 to M to M.

Patient Table:-

<u>Pat_id</u>	Pat_name	Age	Sex	DOB	MOB	Address

Room Table:-

<u>Room_id</u>	Room_No	Room_type	Room_cost

Nurse Table:-

Nrs_id	Nrs_name	Age	Address	Mob	Nrs_wo_shift	experience	Salary

Nursing Service Table:-
This is a junction table between Patient, Room and Nurse Table.

Ns_id	Pat_id	Nrs_id	Room_id	Date	Time

- ❖ Primary Key of the Patient Table goes to Nursing Service Table as part of Primary Key.
- ❖ Primary Key of the Nurse Table goes to Nursing Service Table as part of Primary Key.
- ❖ Primary Key of the Room Table goes to Nursing Service Table as Foreign Key.

Since the Cardinality Ratio from Room to Patient to Nurse is 1 to M to M. Ns_id is a Primary key in the Nursing Service Table. Pat_id from Patient Table, Nrs_id from Nurse Table become parts of Primary Key in the Nursing Service Table. Room_id from Room Table becomes Foreign Key in the Nursing Service Table.

2.4 Relational Database Design

Relational databases are the most commonly used database today. It uses the table to structure information so that it can be readily and easily searched through.

To make a Relational database design we have to be clear about two parts:

1. Functional Dependency
2. Normalization

2.4.1 Functional Dependencies

Definition of functional dependencies:

Given a relational schema R (A1, A2, ..., An) and X, Y {A1, ..., An}.
Then X -> Y means that for every extension of R, the following holds:
R contains no two tuples that are equal in all values of X but differ in at least one value of Y.
(Pronunciation: "X determines Y functionally" "Y is functionally dependent of X").

Example:

Student (matNr, name):
{matNr} -> {name}

Definition of full functional dependencies:

Prerequisites as in Definition 1.
Y is said to be fully functionally dependent of X, if there is no proper subset X' ⊂ X, Where X' -> Y.
Notation: X => Y.

Example:
A University Database:-
Class (classId, room, day, pName)
{classId, room} -> {pName}
{classId, day, pName} -> {room}
{classId} => {pName}
{classId} => {room}

2.4.2 Normalization

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

It has mainly two goals:-

- ✓ First goal: eliminate redundant data
  For example, storing the same data in more than one table

- ✓ Second Goal: ensure data dependencies make sense
  For example, only storing related data in a table

Benefits of Normalization:

- ✓ Less storage space
- ✓ Quicker updates
- ✓ Less data inconsistency
- ✓ Clearer data relationships
- ✓ Easier to add data
- ✓ Flexible Structure

Bad database designs results in:

- ✓ Redundancy: inefficient storage.
- ✓ Anomalies: data inconsistency, difficulties in maintenance.

1NF, 2NF, 3NF, BCNF are some of the early forms in the list that address this problem.

Fulfilment of Normal Forms:

Room Table:-

Room_id	Room_no	Room_type	Room_cost

{Room_id} => {Room_no} Functional Dependency Exist

2 different room no's do not correspond to the same Room_id.

{Room_id} => {Room _type} Functional Dependency Exist

2 different room types' do not correspond to the same Room_id

{Room_id} => {Room cost} Functional Dependency Exist

2 different room cost's do not correspond to the same Room_id

Relation :(Room_id, Room_No, Room_type, Room_cost)

Full Functional Dependencies:

{Room_id} => {Room_no}

{Room_id} => {Room_type}

{Room_id} => {Room_cost}

1NF:-
Attributes do not have sub attributes. So the relation is in 1NF.

2NF:-
Every non primary key is Fully Functional Dependent on the primary key. So the relation is in 2NF.

3NF:-
No chain Exists. So the relation is in 3NF.

BCNF:-
No part of the primary key is Fully Functional Dependent on the non- primary keys. So the relation is in BCNF.

Bill Table:-

Bill_id	Bill_for	Bill_type	Bill_ total

{Bill_id} => {Bill_for}   Functional Dependency Exist.

2 different Bill_for's do not correspond to the same Bill_id.

{Bill _id} => {Bill_type}   Functional Dependency Exist.

2 different Bill_type do not correspond to the same Bill_id.

{Bill _id} => {Bill total   Functional Dependency Exist.

2 different Bill total do not correspond to the same Bill_id. \

Relation :(Bill_id, Bill_for, Bill total, Bill_type)

Full Functional Dependency:

{Bill_id} => {Bill_for}
{Bill_id} => {Bill_type}
{Bill _id} => {Bill total}

1NF:-
Attributes do not have sub attributes. So the relation is in 1NF.

2NF:-
Every non primary key is Fully Functional Dependent on the primary key. So the relation is in 2NF

3NF:-
No chain Exists. So the relation is in 3NF.

BCNF:-
No part of the primary key is Fully Functional Dependent on the non-primary key. So the relation is in BCNF.

In a similar way Bill, Doctor, Medicine and Nurse Tables fulfill all the normal forms.

Junction Tables:

Admission Table:-

This is a junction table between Patient and Room Table

Admsn_id	Room_id	Pat_id	Date	Time

Full Functional Dependencies:

{admsn_id} => {Room_id} Functional Dependency Exist
{adsn_id} => {Date}Functional Dependency Exist
{admsn_id} => {Time} Functional Dependency Exist
{admsn_id} => {Pat_id} Functional Dependency Exist

1NF:-
Attributes do not have sub attributes. So the relation is in 1NF.

2NF:-
Every non primary key is Fully Functional Dependent on the primary key. So the relation is in 2NF.

3NF:-
No chain Exists. So the relation is in 3NF.

BCNF:-
No part of the primary key is Fully Functional Dependent on the non- primary keys. So the relation is in BCNF.

Prescription Table:-

This is a junction table between Patients, Medicine & Doctor Table.

Prs_id	Doc_id	Mdcn_id	Pat_id	Date	Fees	Time

Full Functional Dependencies:

{Prs_id, Mdcn_id}=> {Doc_id} Functional Dependency Exist

{Prs_id, Mdcn_id}=> {Pat_id} Functional Dependency Exist

{Prs_id, Mdcn_id}=> {Date, Fees, Time} Functional Dependency Exist

Relation: (Prs_id, Mdcn_id, Doc_id, Pat_id, Date, Fees, Time)

{Prs_id, Mdcn_id}=> {Doc_id}

{Prs_id, Mdcn_id}=> {Pat_id}

{Prs_id, Mdcn_id}=> {Date}

{Prs_id, Mdcn_id}=> {Time}

{Prs_id, Mdcn_id}=> {Fees}

1NF:-
Attributes do not have sub attributes. So the relation is in 1NF.

2NF:-
Every non primary key is Fully Functional Dependent on the primary key. So the relation is in 2NF.

3NF:-
No chain Exists. So the relation is in 3NF.

BCNF:-
No part of the primary key is Fully Functional Dependent on the non-primary keys. So the relation is in BCNF.

In a similar way Assist and Nursing Service tables fulfill all normal forms.

Violation of Normal Form:

Payment Table:-

This is a junction table between Patients and Bill tables.

Pay_id	Pat_id	Bill_id	Pay_type	Pay_date

For Payment relation, the following functional dependencies exist:

{Pay_id}=> {Pay_Type, Pay_date, Pat_id}

Two different patient ids, payment dates and payment types cannot correspond to the same payment id. So Pay_Type, Pay-date and Pat_id are fully functionally dependent on Pay_id.

{Bill_id}=> {Pat_id}

Similarly two different patient ids cannot correspond to the same bill id. So Pat_id are fully functionally dependent on Bill_id. Based on the above functional dependencies:

The relation is in 1NF.

The relation is not in 2NF because all non-primary keys are not fully functionally dependent on the primary key (Pay_id, Bill_id). So we split the relation to make it 2NF.

Payment1 (<u>Pay_id</u>, Pay_Type, Pay_date, Pat_id)

Payment2 (<u>Bill_id</u>, Pat_id)

The relations are now in 2NF.

3NF:

There is no chain. So the relations are in 3NF.

BCNF:

No Part of the primary key (Pay_Id, Bill_Id) is fully functionally dependent on any non-primary key. So the relations are in BCNF.

CHAPTER 3

IMPLEMENTATION IN MYSQL WORKBENCH

After Normalization, we implemented our Database in MySQL Workbench. There were 11 tables and each of them was connected accurately in the MySQL Workbench's Entity Relationship Diagram. Then we entered the data in the corresponding database tables.

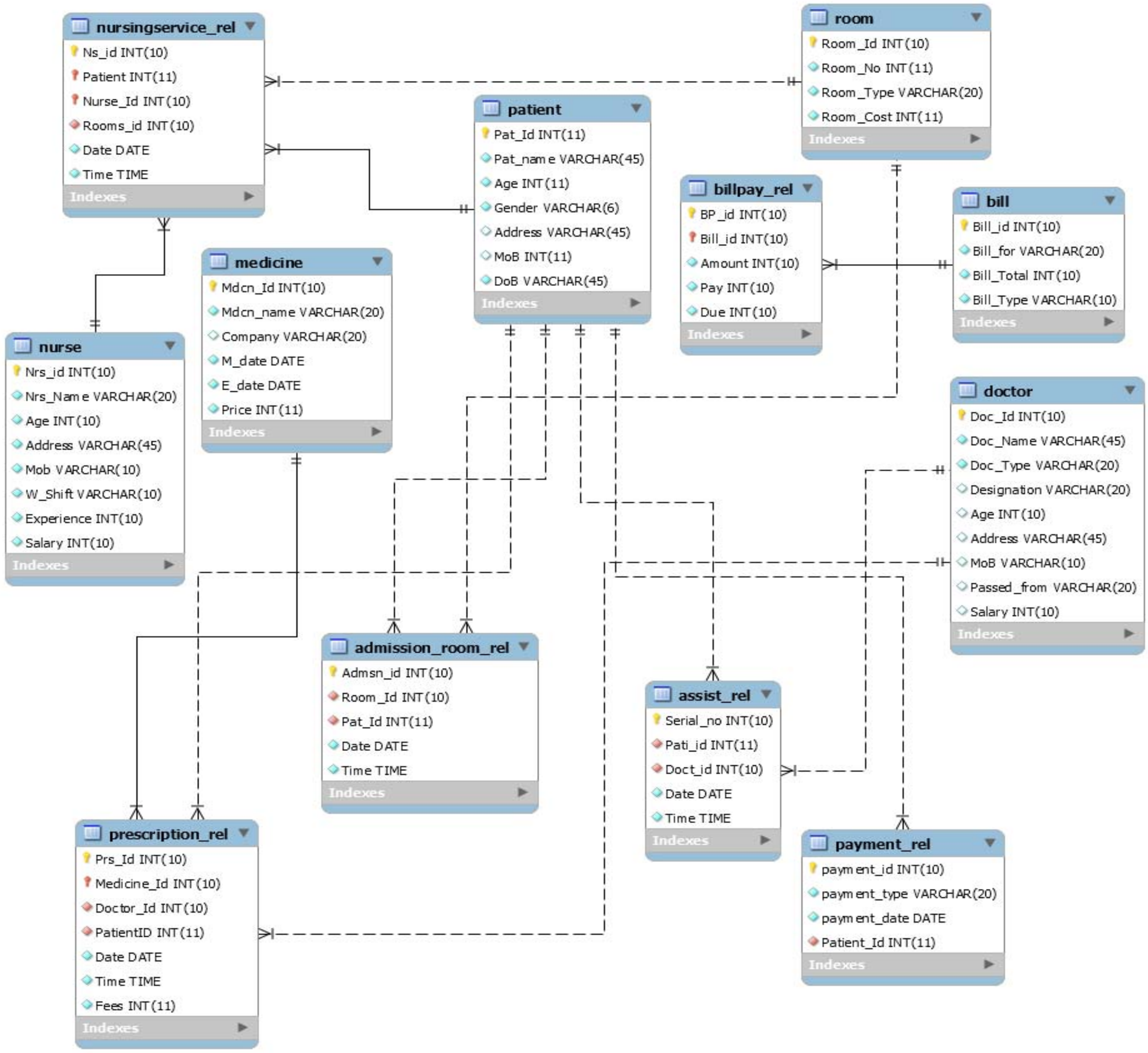


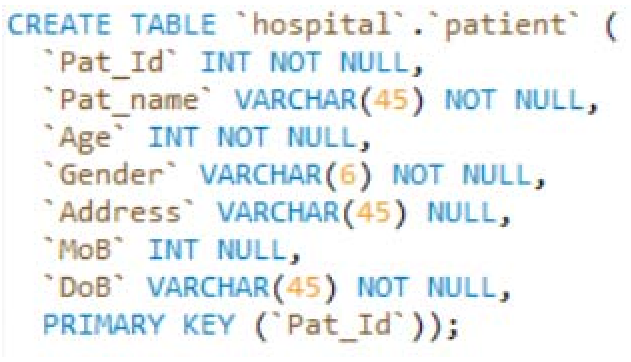
Fig2: Relational model Implementation in MySQL WorkBench.

3.1 Creation of Tables and Insertion of Data:

In our course project, we create tables and insert data using MySQL Workbench and MySQL Language.

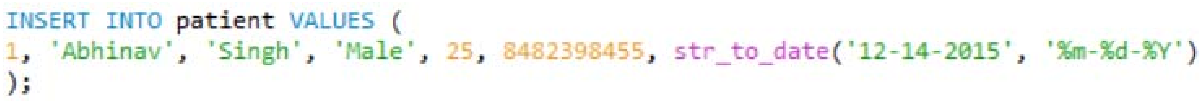
MAIN TABLE

Create Patient Table



```
CREATE TABLE `hospital`.`patient` (
  `Pat_Id` INT NOT NULL,
  `Pat_name` VARCHAR(45) NOT NULL,
  `Age` INT NOT NULL,
  `Gender` VARCHAR(6) NOT NULL,
  `Address` VARCHAR(45) NULL,
  `MoB` INT NULL,
  `DoB` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`Pat_Id`));
```

Insert Values into Patient Table

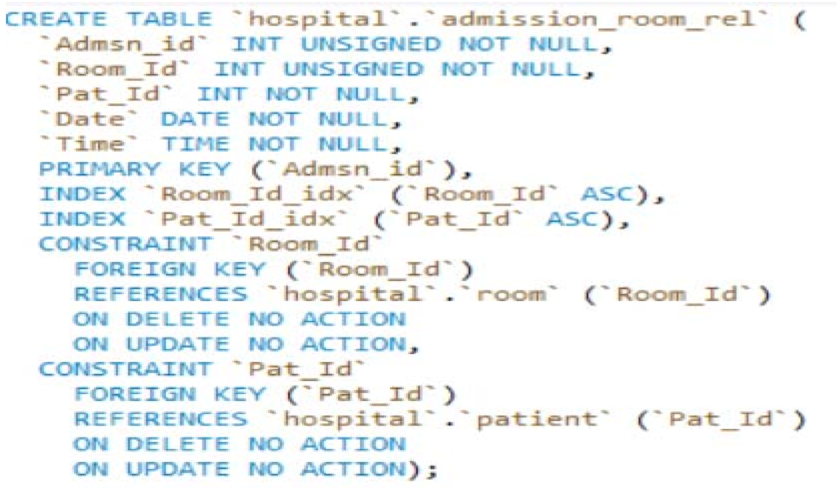


```
INSERT INTO patient VALUES (
1, 'Abhinav', 'Singh', 'Male', 25, 8482398455, str_to_date('12-14-2015', '%m-%d-%Y')
);
```

In this way we create all the main tables and insert data in them.

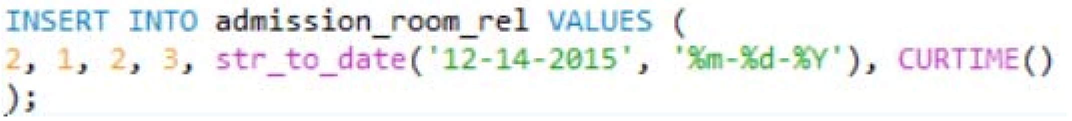
JUNCTION TABLE

Create Admission Table



```
CREATE TABLE `hospital`.`admission_room_rel` (
  `Admsn_id` INT UNSIGNED NOT NULL,
  `Room_Id` INT UNSIGNED NOT NULL,
  `Pat_Id` INT NOT NULL,
  `Date` DATE NOT NULL,
  `Time` TIME NOT NULL,
  PRIMARY KEY (`Admsn_id`),
  INDEX `Room_Id_idx` (`Room_Id` ASC),
  INDEX `Pat_Id_idx` (`Pat_Id` ASC),
  CONSTRAINT `Room_Id`
    FOREIGN KEY (`Room_Id`)
    REFERENCES `hospital`.`room` (`Room_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Pat_Id`
    FOREIGN KEY (`Pat_Id`)
    REFERENCES `hospital`.`patient` (`Pat_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

Insert Values into Admission Table

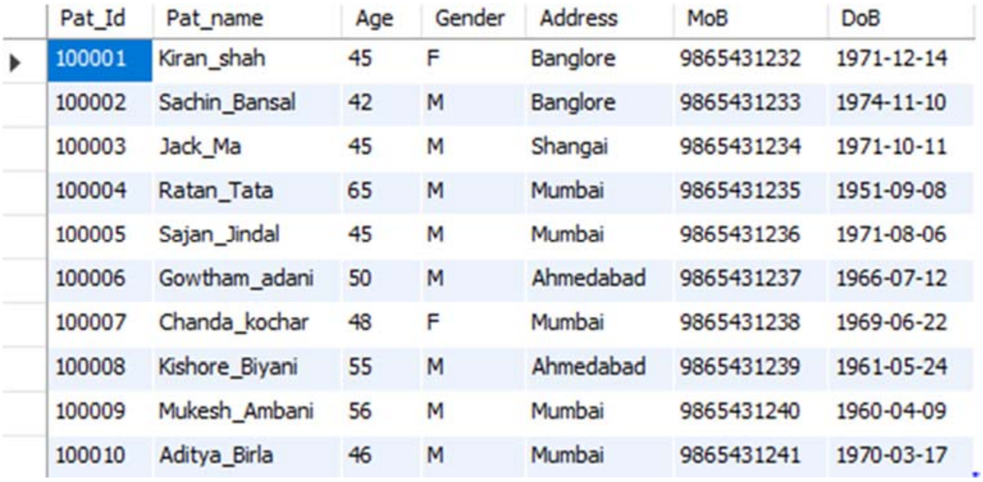


```
INSERT INTO admission_room_rel VALUES (
2, 1, 2, 3, str_to_date('12-14-2015', '%m-%d-%Y'), CURTIME()
);
```

In this way we create all the junction tables and insert data in them.

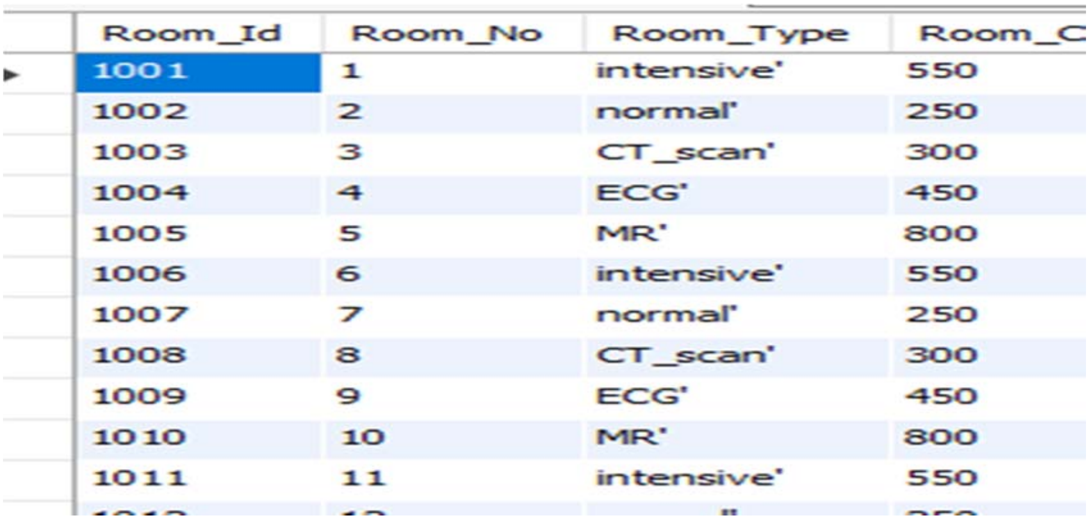
3.2 Sample Data values of Tables

Patient table



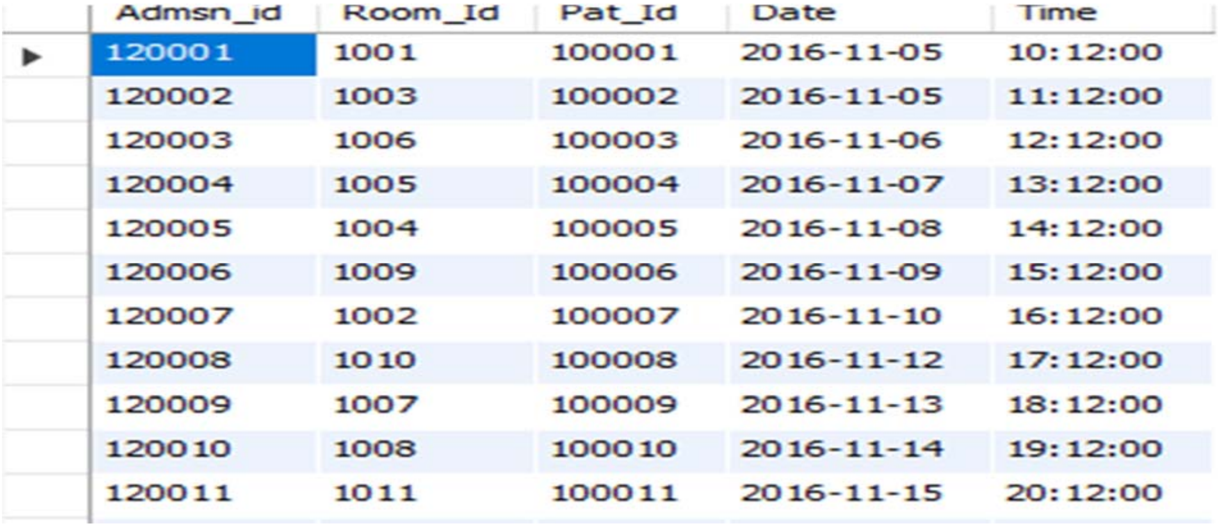
Pat_Id	Pat_name	Age	Gender	Address	MoB	DoB
100001	Kiran_shah	45	F	Banglore	9865431232	1971-12-14
100002	Sachin_Bansal	42	M	Banglore	9865431233	1974-11-10
100003	Jack_Ma	45	M	Shangai	9865431234	1971-10-11
100004	Ratan_Tata	65	M	Mumbai	9865431235	1951-09-08
100005	Sajan_Jindal	45	M	Mumbai	9865431236	1971-08-06
100006	Gowtham_adani	50	M	Ahmedabad	9865431237	1966-07-12
100007	Chanda_kochar	48	F	Mumbai	9865431238	1969-06-22
100008	Kishore_Biyani	55	M	Ahmedabad	9865431239	1961-05-24
100009	Mukesh_Ambani	56	M	Mumbai	9865431240	1960-04-09
100010	Aditya_Birla	46	M	Mumbai	9865431241	1970-03-17

Room table



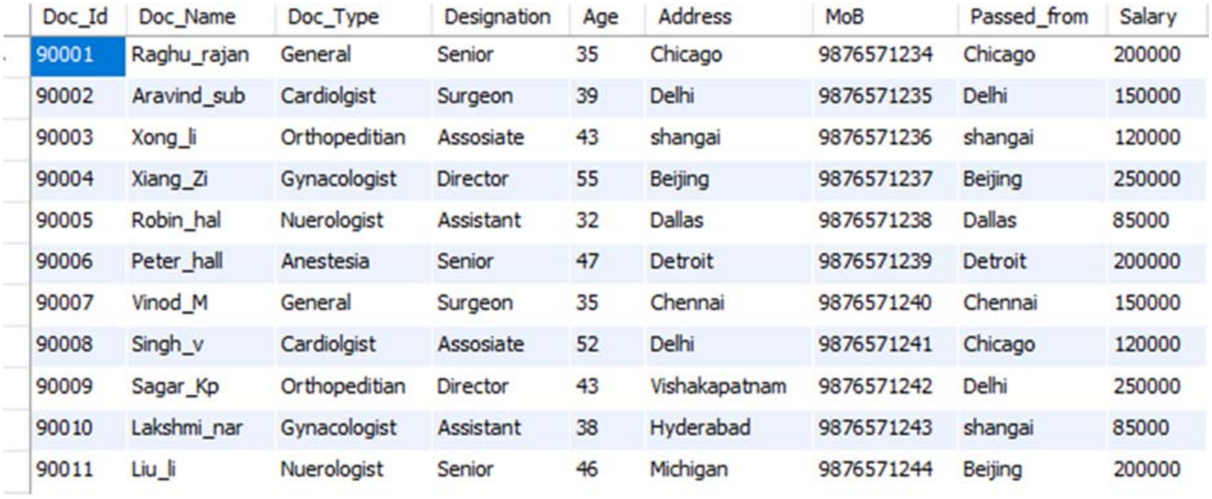
Room_Id	Room_No	Room_Type	Room_C
1001	1	intensive'	550
1002	2	normal'	250
1003	3	CT_scan'	300
1004	4	ECG'	450
1005	5	MR'	800
1006	6	intensive'	550
1007	7	normal'	250
1008	8	CT_scan'	300
1009	9	ECG'	450
1010	10	MR'	800
1011	11	intensive'	550

Admission room table:



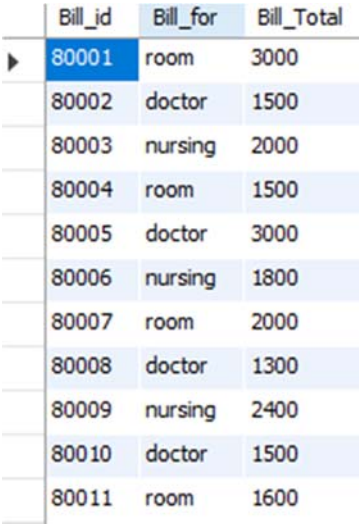
Admsn_id	Room_Id	Pat_Id	Date	Time
120001	1001	100001	2016-11-05	10:12:00
120002	1003	100002	2016-11-05	11:12:00
120003	1006	100003	2016-11-06	12:12:00
120004	1005	100004	2016-11-07	13:12:00
120005	1004	100005	2016-11-08	14:12:00
120006	1009	100006	2016-11-09	15:12:00
120007	1002	100007	2016-11-10	16:12:00
120008	1010	100008	2016-11-12	17:12:00
120009	1007	100009	2016-11-13	18:12:00
120010	1008	100010	2016-11-14	19:12:00
120011	1011	100011	2016-11-15	20:12:00

Doctor table:



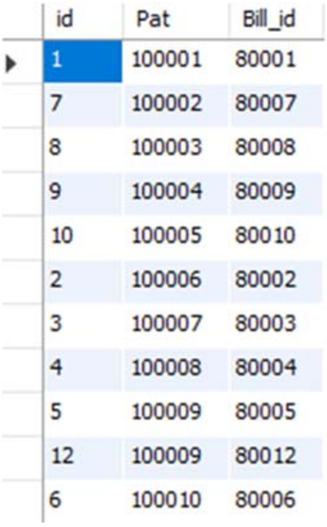
Doc_Id	Doc_Name	Doc_Type	Designation	Age	Address	MoB	Passed_from	Salary
90001	Raghu_rajan	General	Senior	35	Chicago	9876571234	Chicago	200000
90002	Aravind_sub	Cardiolgist	Surgeon	39	Delhi	9876571235	Delhi	150000
90003	Xong_li	Orthopeditian	Assosiate	43	shangai	9876571236	shangai	120000
90004	Xiang_Zi	Gynacologist	Director	55	Beijing	9876571237	Beijing	250000
90005	Robin_hal	Nuerologist	Assistant	32	Dallas	9876571238	Dallas	85000
90006	Peter_hall	Anestesia	Senior	47	Detroit	9876571239	Detroit	200000
90007	Vinod_M	General	Surgeon	35	Chennai	9876571240	Chennai	150000
90008	Singh_v	Cardiolgist	Assosiate	52	Delhi	9876571241	Chicago	120000
90009	Sagar_Kp	Orthopeditian	Director	43	Vishakapatnam	9876571242	Delhi	250000
90010	Lakshmi_nar	Gynacologist	Assistant	38	Hyderabad	9876571243	shangai	85000
90011	Liu_li	Nuerologist	Senior	46	Michigan	9876571244	Beijing	200000

Bill table:



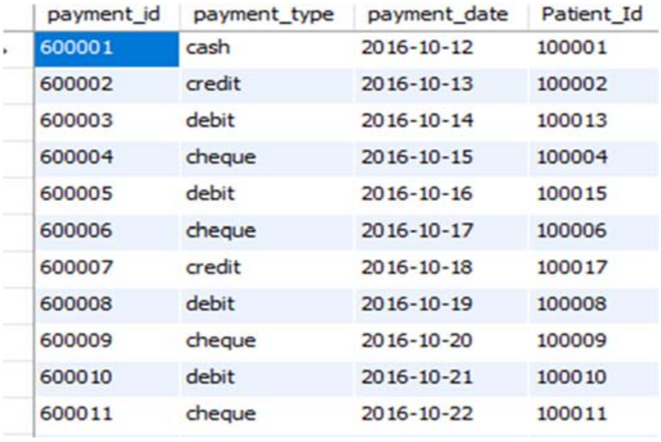
Bill_id	Bill_for	Bill_Total
80001	room	3000
80002	doctor	1500
80003	nursing	2000
80004	room	1500
80005	doctor	3000
80006	nursing	1800
80007	room	2000
80008	doctor	1300
80009	nursing	2400
80010	doctor	1500
80011	room	1600

Pat_Bill table:



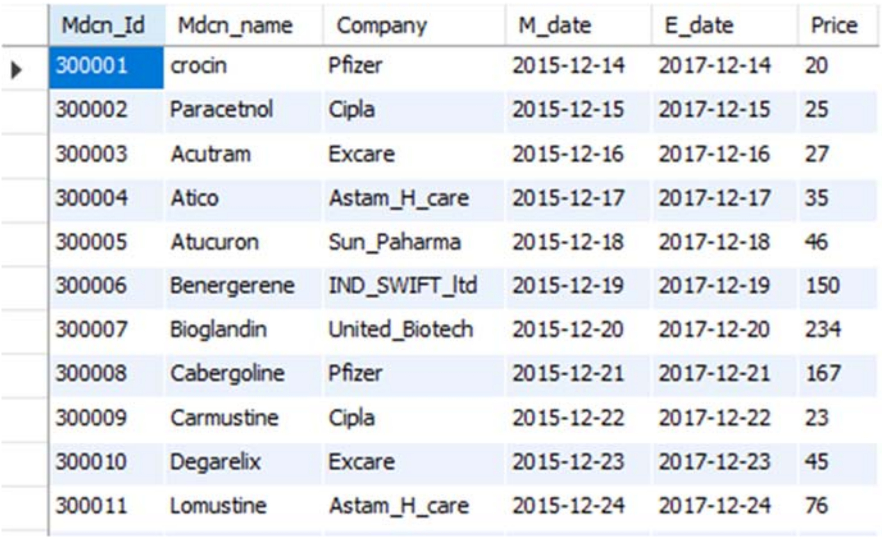
id	Pat	Bill_id
1	100001	80001
7	100002	80007
8	100003	80008
9	100004	80009
10	100005	80010
2	100006	80002
3	100007	80003
4	100008	80004
5	100009	80005
12	100009	80012
6	100010	80006

Payment Table:



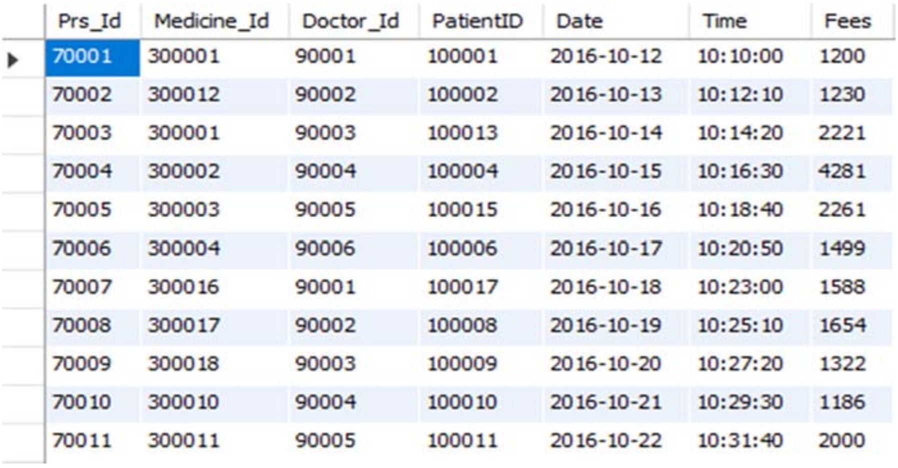
payment_id	payment_type	payment_date	Patient_Id
600001	cash	2016-10-12	100001
600002	credit	2016-10-13	100002
600003	debit	2016-10-14	100013
600004	cheque	2016-10-15	100004
600005	debit	2016-10-16	100015
600006	cheque	2016-10-17	100006
600007	credit	2016-10-18	100017
600008	debit	2016-10-19	100008
600009	cheque	2016-10-20	100009
600010	debit	2016-10-21	100010
600011	cheque	2016-10-22	100011

Medicine table:



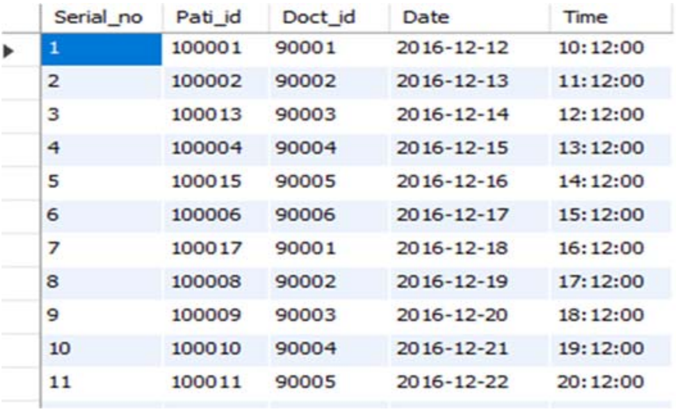
Mdcn_Id	Mdcn_name	Company	M_date	E_date	Price
300001	crocin	Pfizer	2015-12-14	2017-12-14	20
300002	Paracetnol	Cipla	2015-12-15	2017-12-15	25
300003	Acutram	Excare	2015-12-16	2017-12-16	27
300004	Atico	Astam_H_care	2015-12-17	2017-12-17	35
300005	Atucuron	Sun_Paharma	2015-12-18	2017-12-18	46
300006	Benergerene	IND_SWIFT_ltd	2015-12-19	2017-12-19	150
300007	Bioglandin	United_Biotech	2015-12-20	2017-12-20	234
300008	Cabergoline	Pfizer	2015-12-21	2017-12-21	167
300009	Carmustine	Cipla	2015-12-22	2017-12-22	23
300010	Degarelix	Excare	2015-12-23	2017-12-23	45
300011	Lomustine	Astam_H_care	2015-12-24	2017-12-24	76

Prescription table:



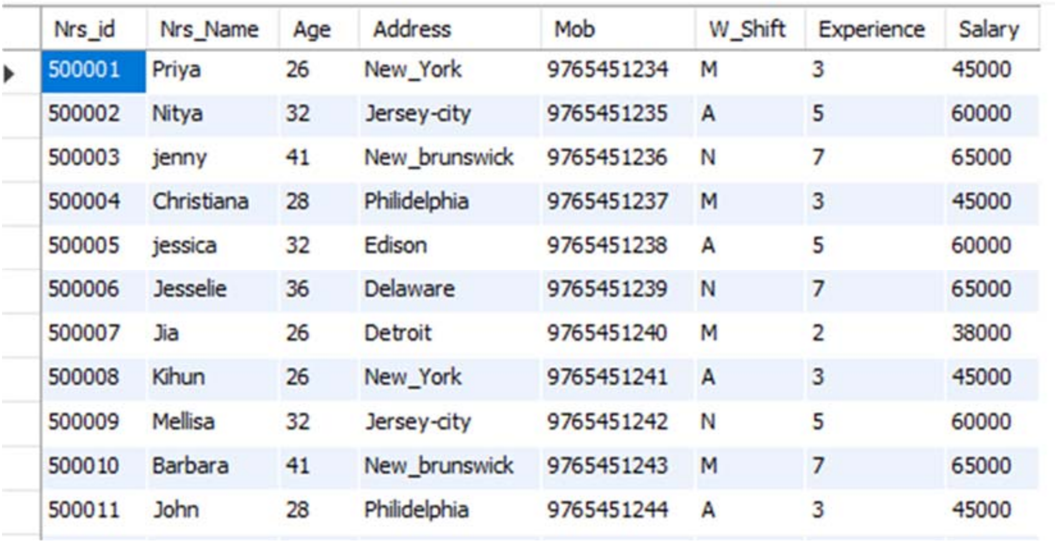
Prs_Id	Medicine_Id	Doctor_Id	PatientID	Date	Time	Fees
70001	300001	90001	100001	2016-10-12	10:10:00	1200
70002	300012	90002	100002	2016-10-13	10:12:10	1230
70003	300001	90003	100013	2016-10-14	10:14:20	2221
70004	300002	90004	100004	2016-10-15	10:16:30	4281
70005	300003	90005	100015	2016-10-16	10:18:40	2261
70006	300004	90006	100006	2016-10-17	10:20:50	1499
70007	300016	90001	100017	2016-10-18	10:23:00	1588
70008	300017	90002	100008	2016-10-19	10:25:10	1654
70009	300018	90003	100009	2016-10-20	10:27:20	1322
70010	300010	90004	100010	2016-10-21	10:29:30	1186
70011	300011	90005	100011	2016-10-22	10:31:40	2000

Assist table:



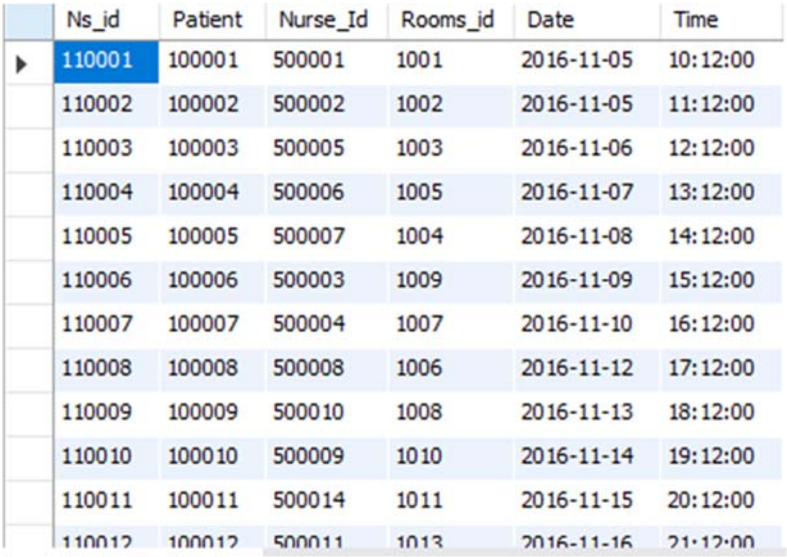
Serial_no	Pati_id	Doct_id	Date	Time
1	100001	90001	2016-12-12	10:12:00
2	100002	90002	2016-12-13	11:12:00
3	100013	90003	2016-12-14	12:12:00
4	100004	90004	2016-12-15	13:12:00
5	100015	90005	2016-12-16	14:12:00
6	100006	90006	2016-12-17	15:12:00
7	100017	90001	2016-12-18	16:12:00
8	100008	90002	2016-12-19	17:12:00
9	100009	90003	2016-12-20	18:12:00
10	100010	90004	2016-12-21	19:12:00
11	100011	90005	2016-12-22	20:12:00

Nurse table:



Nrs_id	Nrs_Name	Age	Address	Mob	W_Shift	Experience	Salary
500001	Priya	26	New_York	9765451234	M	3	45000
500002	Nitya	32	Jersey-city	9765451235	A	5	60000
500003	jenny	41	New_brunswick	9765451236	N	7	65000
500004	Christiana	28	Philidelphia	9765451237	M	3	45000
500005	jessica	32	Edison	9765451238	A	5	60000
500006	Jesselie	36	Delaware	9765451239	N	7	65000
500007	Jia	26	Detroit	9765451240	M	2	38000
500008	Kihun	26	New_York	9765451241	A	3	45000
500009	Mellisa	32	Jersey-city	9765451242	N	5	60000
500010	Barbara	41	New_brunswick	9765451243	M	7	65000
500011	John	28	Philidelphia	9765451244	A	3	45000

Nursing service table:



Ns_id	Patient	Nurse_Id	Rooms_id	Date	Time
110001	100001	500001	1001	2016-11-05	10:12:00
110002	100002	500002	1002	2016-11-05	11:12:00
110003	100003	500005	1003	2016-11-06	12:12:00
110004	100004	500006	1005	2016-11-07	13:12:00
110005	100005	500007	1004	2016-11-08	14:12:00
110006	100006	500003	1009	2016-11-09	15:12:00
110007	100007	500004	1007	2016-11-10	16:12:00
110008	100008	500008	1006	2016-11-12	17:12:00
110009	100009	500010	1008	2016-11-13	18:12:00
110010	100010	500009	1010	2016-11-14	19:12:00
110011	100011	500014	1011	2016-11-15	20:12:00
110012	100012	500011	1013	2016-11-16	21:12:00

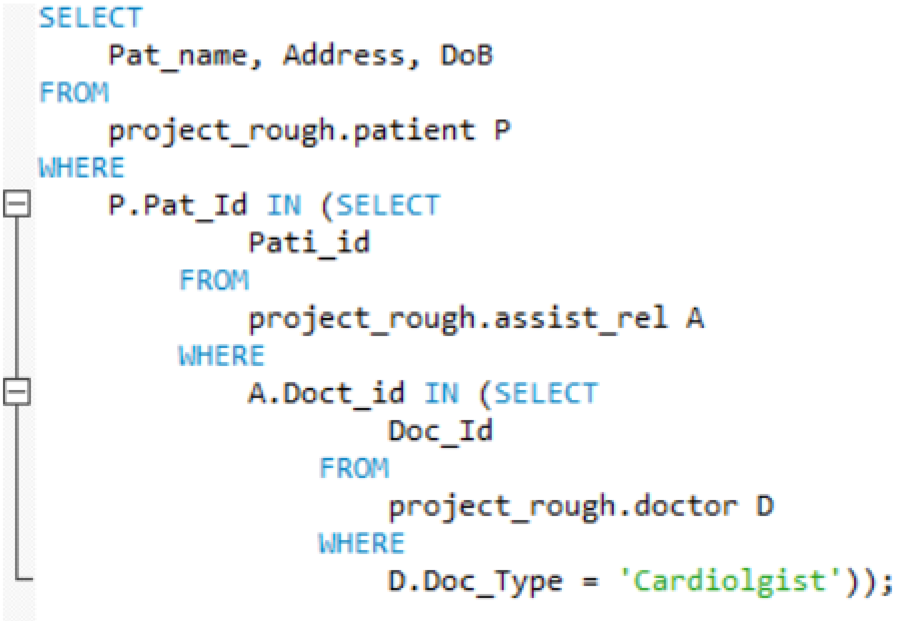
3.3 Complex Queries

After completing the implementation we retrieved different information from the system by joining 2 or more tables of the system. Sample Examples are given below:

Question 1

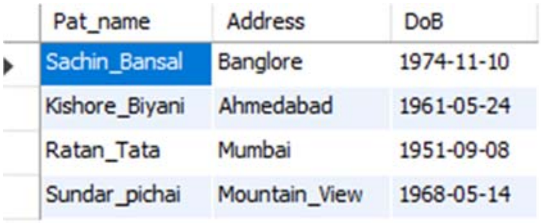
Display the name, Address, Date of birth of the patients who are treated by a Cardiologist?

Query 1:



```
SELECT
    Pat_name, Address, DoB
FROM
    project_rough.patient P
WHERE
    P.Pat_Id IN (SELECT
            Pati_id
        FROM
            project_rough.assist_rel A
        WHERE
            A.Doct_id IN (SELECT
                    Doc_Id
                FROM
                    project_rough.doctor D
                WHERE
                    D.Doc_Type = 'Cardiolgist'));
```

Output:

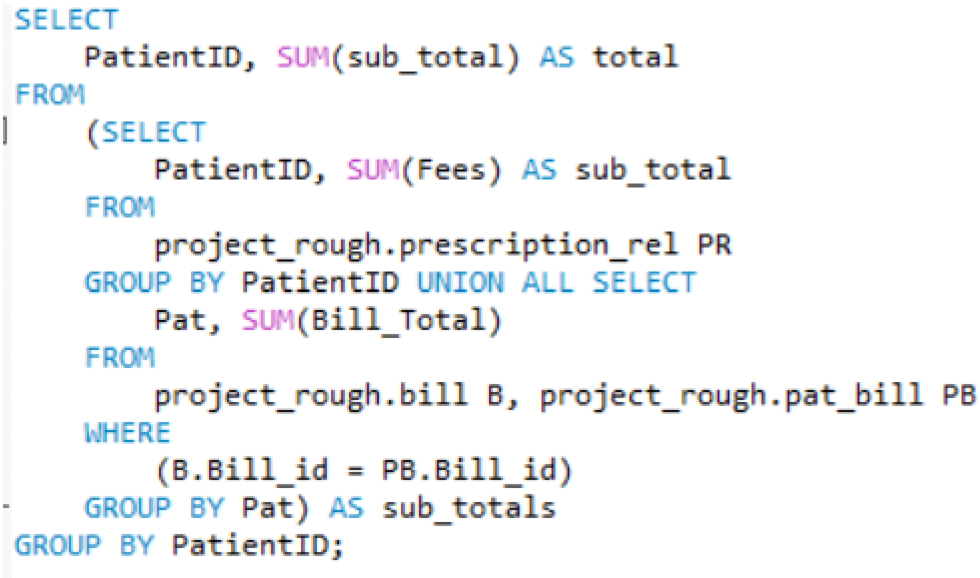


Pat_name	Address	DoB
Sachin_Bansal	Banglore	1974-11-10
Kishore_Biyani	Ahmedabad	1961-05-24
Ratan_Tata	Mumbai	1951-09-08
Sundar_pichai	Mountain_View	1968-05-14

Question 2

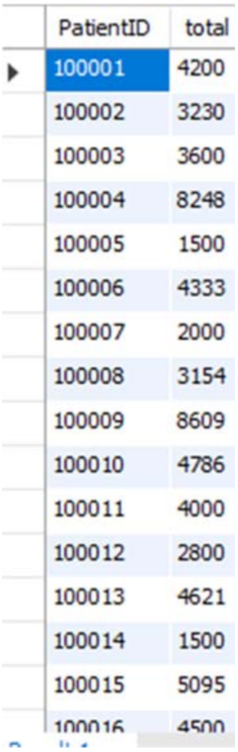
Patient id and total amount of fee paid by each in all departments?

Query 2:



```
SELECT
    PatientID, SUM(sub_total) AS total
FROM
    (SELECT
        PatientID, SUM(Fees) AS sub_total
    FROM
        project_rough.prescription_rel PR
    GROUP BY PatientID UNION ALL SELECT
        Pat, SUM(Bill_Total)
    FROM
        project_rough.bill B, project_rough.pat_bill PB
    WHERE
        (B.Bill_id = PB.Bill_id)
    GROUP BY Pat) AS sub_totals
GROUP BY PatientID;
```

Output 2:

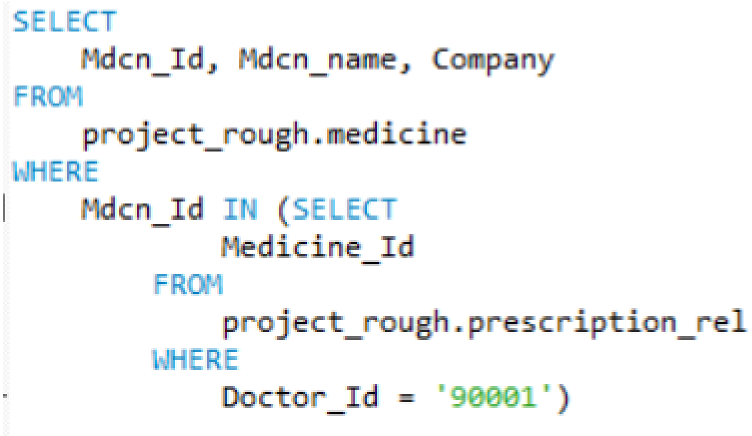


PatientID	total
100001	4200
100002	3230
100003	3600
100004	8248
100005	1500
100006	4333
100007	2000
100008	3154
100009	8609
100010	4786
100011	4000
100012	2800
100013	4621
100014	1500
100015	5095
100016	4500

Question 3

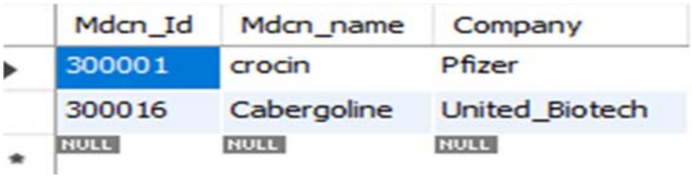
What drugs are prescribed by the doctor with doctor id 90001?

Query 3:



```
SELECT
    Mdcn_Id, Mdcn_name, Company
FROM
    project_rough.medicine
WHERE
    Mdcn_Id IN (SELECT
            Medicine_Id
        FROM
            project_rough.prescription_rel
        WHERE
            Doctor_Id = '90001')
```

Output 3:

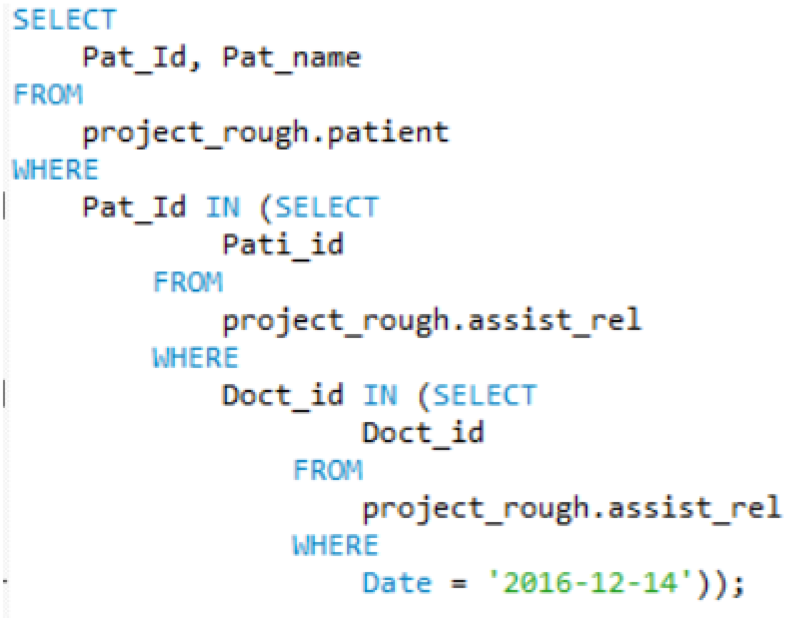


	Mdcn_Id	Mdcn_name	Company
▶	300001	crocin	Pfizer
	300016	Cabergoline	United_Biotech
*	NULL	NULL	NULL

Question 4

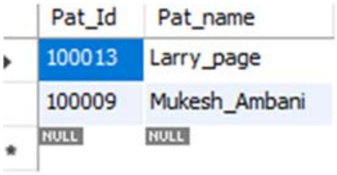
Names of all Patients and their ids treated by the doctor who worked on 2016-12-14?

Query 4:



```
SELECT
    Pat_Id, Pat_name
FROM
    project_rough.patient
WHERE
    Pat_Id IN (SELECT
            Pati_id
        FROM
            project_rough.assist_rel
        WHERE
            Doct_id IN (SELECT
                    Doct_id
                FROM
                    project_rough.assist_rel
                WHERE
                    Date = '2016-12-14'));
```

Output 4:

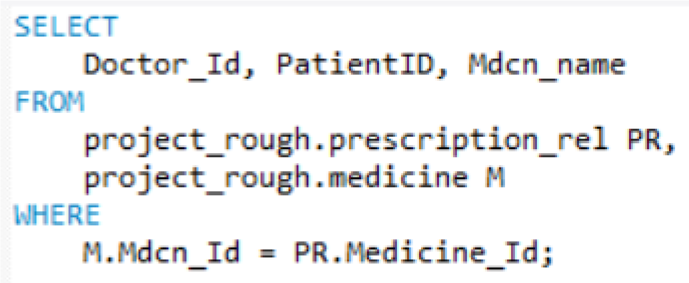


Pat_Id	Pat_name
100013	Larry_page
100009	Mukesh_Ambani
NULL	NULL

Question 5

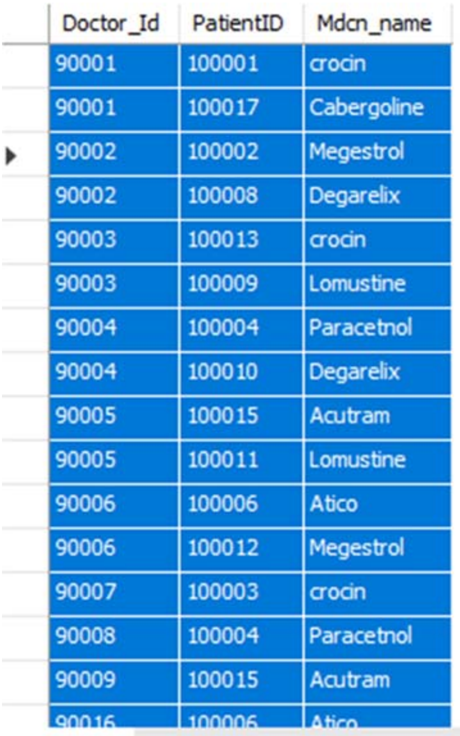
List all the Names of the patient, doctor and the medical prescription?

Query 5:



```
SELECT
    Doctor_Id, PatientID, Mdcn_name
FROM
    project_rough.prescription_rel PR,
    project_rough.medicine M
WHERE
    M.Mdcn_Id = PR.Medicine_Id;
```

Output 5:



Doctor_Id	PatientID	Mdcn_name
90001	100001	crocin
90001	100017	Cabergoline
90002	100002	Megestrol
90002	100008	Degarelix
90003	100013	crocin
90003	100009	Lomustine
90004	100004	Paracetnol
90004	100010	Degarelix
90005	100015	Acutram
90005	100011	Lomustine
90006	100006	Atico
90006	100012	Megestrol
90007	100003	crocin
90008	100004	Paracetnol
90009	100015	Acutram
90016	100006	Atico

CHAPTER4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

With team work, co-operative assistance from each teammates and Professor's Guidance, we have come to the end of our course project report. It is not the work of one day. In fact it took us a 4 Months to complete. The group members worked hard to make it a good and improvised report.

Summing up, we worked on a case study of Hospital Management, designing and storing its information in a sample database of our creation. We designed ER models, Relational Models and Normalized tables of the relational model and finally developed our database in MySQL Workbench.

The second part of this report involved developing the backend implementation and putting the above mentioned concepts in a MySQL based database. We used MySQL Workbench Diagram to create normalized relational schema, filled the server tables with data values and queried different useful information from the database. We researched Backend design Guidelines and applied some of those to our database development design practice. We have taken into account issues of reliability and security too.

4.2 Future Work

While an efficient and normalized MySQL- based backend database has been successfully database has been developed we have in mind some scope for future work involving Guideline View Features and Trigger Features. A direct development can be targeted on designing a user friendly and efficient interface to this backend database.

The DataGridView control provides a customizable table for displaying data. The DataGridView class allows customization of cells, rows, columns, and borders through the use of properties such as DefaultCellStyle, ColumnHeadersDefaultCellStyle, CellBorderStyle, and GridColor.

We can use a DataGridView control to display data with or without an underlying data source. Without specifying a data source, we can create columns and rows that contain data and add them directly to the DataGridView using the Rows and Columns properties. You can also use the Rows collection to access DataGridViewRow objects and the DataGridViewRow.Cells property to read or write cell values directly. The Item indexer also provides direct access to cells.

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for keeping the integrity of the information on the database. For example, when a new record (representing a new worker) is added to the employees table, new records should be created also in the tables of the taxes, vacations, and salaries.

We can write triggers that fire whenever one of the following operations occurs:

1. DML statements (INSERT, UPDATE, DELETE) on a particular table or view, issued by any user.

2. DDL statements (CREATE or ALTER primarily) issued either by a particular schema/user or by any schema/user in the database.

3. Database events, such as logon/logoff, errors, or startup/shutdown, also issued either by a particular schema/user or by any schema/user in the database

Triggers are similar to stored procedures. A trigger stored in the database can include SQL and PL/SQL or Java statements to run as a unit and can invoke stored procedures. However, procedures and triggers differ in the way that they are invoked. A procedure is explicitly run by a user, application, or trigger. Triggers are implicitly fired by Oracle SQL server when a triggering event occurs, no matter which user is connected or which application is being used.

Summary

We can say that Data Grid view is very essential while developing frontend Framework. We can do a lot of things easily and efficiently using Data Grid view. Though the coding is not so easy but it will help us to make user friendly software. On the other hand trigger is a very essential approach in database. We can make a database for functional and efficient using Triggers.

REFERENCES

1. http://www.blurtit.com/q959542.html

2. http://en.wikipedia.org/wiki/Database#Applications

3. http://www.cl500.net/pros_cons.html

4. http://en.wikipedia.org/wiki/Entity

5. http:/en.wikipedia.org/wiki/Attribute_%28computing%29

6. http://en.wikipedia.org/wiki/Data_type

7. Prof. Dorothee Koch, Lecture Notes : HfT Stuuttgart-Normalaization.e.fm

8. Source: SOFTWARE ENGINEERING A Practical Approach 6th Edition McGRAW-HILL INTERNATIONAL EDITION by ROGER S. PRESSMAN (Chapter-26)

9. http://ergo.human.cornell.edu/ahtutorials/interface.html

APPENDIX:

```
#Creating the Database
create schema project_rough;

#Creating Patient Table
CREATE TABLE project_rough.patient
(
  Pat_Id INT NOT NULL auto_increment,
  Pat_name VARCHAR(45) NOT NULL,
  Age INT NOT NULL,
  Gender VARCHAR(6) NOT NULL check (Sex = 'F' or Sex ='M' ),
  Address VARCHAR(45) NULL,
  MoB double NULL,
  DoB date NOT NULL,
  PRIMARY KEY (Pat_Id)
);

#Populating Patient table
insert into project_rough.patient values ( 100001, "Kiran_shah", 45, 'F',
'Banglore', 9865431232,str_to_date('12-14-1971', '%m-%d-%Y'));
insert into project_rough.patient values ( 100002, "Sachin_Bansal", 42,
'M', 'Banglore', 9865431233,str_to_date('11-10-1974', '%m-%d-%Y'));
insert into project_rough.patient values ( 100003, "Jack_Ma", 45, 'M',
'Shangai', 9865431234,str_to_date('10-11-1971', '%m-%d-%Y'));
insert into project_rough.patient values ( 100004, "Ratan_Tata", 65, 'M',
'Mumbai', 9865431235,str_to_date('09-08-1951', '%m-%d-%Y'));
insert into project_rough.patient values ( 100005, "Sajan_Jindal", 45, 'M',
'Mumbai', 9865431236,str_to_date('08-06-1971', '%m-%d-%Y'));
insert into project_rough.patient values ( 100006, "Gowtham_adani", 50,
'M', 'Ahmedabad', 9865431237,str_to_date('07-12-1966', '%m-%d-%Y'));
insert into project_rough.patient values ( 100007, "Chanda_kochar", 48,
'F', 'Mumbai', 9865431238,str_to_date('06-22-1969', '%m-%d-%Y'));
insert into project_rough.patient values ( 100008, "Kishore_Biyani",55,
'M', 'Ahmedabad', 9865431239,str_to_date('05-24-1961', '%m-%d-%Y'));
insert into project_rough.patient values ( 100009, "Mukesh_Ambani", 56,
'M', 'Mumbai', 9865431240,str_to_date('04-09-1960', '%m-%d-%Y'));
insert into project_rough.patient values ( 100010, "Aditya_Birla", 46,'M',
'Mumbai', 9865431241,str_to_date('03-17-1970', '%m-%d-%Y'));
insert into project_rough.patient values ( 100011, "Elon_Musk", 44, 'M',
'Fremont', 98654331242,str_to_date('02-13-1972', '%m-%d-%Y'));
insert into project_rough.patient values ( 100012, "Hugo_Burra",34, 'M',
'Shangai', 9865431243,str_to_date('01-16-1982', '%m-%d-%Y'));
insert into project_rough.patient values ( 100013, "Larry_page", 47, 'M',
'Mountain_View', 9865431244,str_to_date('12-24-1969', '%m-%d-%Y'));
insert into project_rough.patient values ( 100014, "Tim_cook", 53, 'M',
'Cuppertino', 986541245,str_to_date('11-27-1963', '%m-%d-%Y'));
insert into project_rough.patient values ( 100015, "Indra_nuyi", 54, 'F',
'New_York', 9865431246,str_to_date('10-31-1964', '%m-%d-%Y'));
insert into project_rough.patient values ( 100016, "Mark_zukerberg", 36,
'M', 'Menlo_Park', 9865431247,str_to_date('09-15-1980', '%m-%d-%Y'));
insert into project_rough.patient values ( 100017, "Cher_wang", 44, 'F',
'Taipei', 9865431248,str_to_date('08-21-1972', '%m-%d-%Y'));
insert into project_rough.patient values ( 100018, "Satya_Nadella", 52,
'M', 'Redmond', 9865431249,str_to_date('07-26-1965', '%m-%d-%Y'));
insert into project_rough.patient values ( 100019, "Jeff_bezos", 48, 'M',
'Seattle', 9865431250,str_to_date('06-22-1969', '%m-%d-%Y'));
insert into project_rough.patient values ( 100020, "Sundar_pichai", 47,
'M', 'Mountain_View', 9865431251,str_to_date('05-14-1968', '%m-%d-%Y'));
```

```
#Creating Room Table
CREATE TABLE project_rough.room
(
  Room_Id INT UNSIGNED NOT NULL,
  Room_No INT NOT NULL,
  Room_Type VARCHAR(20) NOT NULL,
  Room_Cost INT NOT NULL,
  PRIMARY KEY (Room_Id)
);

#Populating Room Table
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1001', '1', 'intensive\'', '550');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1002', '2', 'normal\'', '250');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1003', '3', 'CT_scan\'', '300');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1004', '4', 'ECG\'', '450');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1005', '5', 'MR\'', '800');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1006', '6', 'intensive\'', '550');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1007', '7', 'normal\'', '250');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1008', '8', 'CT_scan\'', '300');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1009', '9', 'ECG\'', '450');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1010', '10', 'MR\'', '800');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1011', '11', 'intensive\'', '550');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1012', '12', 'normal\'', '250');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1013', '13', 'CT_scan\'', '300');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1014', '14', 'ECG\'', '450');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1015', '15', 'MR\'', '800');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1016', '16', 'intensive\'', '550');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1017', '17', 'normal\'', '250');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1018', '18', 'CT_scan\'', '300');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1019', '19', 'ECG\'', '450');
INSERT INTO project_rough.room (Room_Id, Room_No, Room_Type, Room_Cost)
VALUES ('1020', '20', 'MR\'', '800');

# Creating Nurse table
CREATE TABLE project_rough.nurse (
  Nrs_id INT UNSIGNED NOT NULL,
  Nrs_Name VARCHAR(20) NOT NULL,
  Age INT UNSIGNED NOT NULL,
  Address VARCHAR(45) NOT NULL,
  Mob VARCHAR(10) NOT NULL,
  W_Shift VARCHAR(10) NOT NULL,
  Experience INT UNSIGNED NOT NULL,
  Salary INT UNSIGNED NOT NULL,
  PRIMARY KEY (Nrs_id));

# Populating Nurse table
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500001', 'Priya', '26', 'New_York',
'9765451234', 'M', '3', '45000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500002', 'Nitya', '32', 'Jersey-
city', '9765451235', 'A', '5', '60000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500003', 'jenny', '41',
'New_brunswick', '9765451236', 'N', '7', '65000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500004', 'Christiana', '28',
'Philidelphia', '9765451237', 'M', '3', '45000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500005', 'jessica', '32', 'Edison',
'9765451238', 'A', '5', '60000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500006', 'Jesselie', '36',
'Delaware', '9765451239', 'N', '7', '65000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500007', 'Jia', '26', 'Detroit',
'9765451240', 'M', '2', '38000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500008', 'Kihun', '26', 'New_York',
'9765451241', 'A', '3', '45000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500009', 'Mellisa', '32', 'Jersey-
city', '9765451242', 'N', '5', '60000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500010', 'Barbara', '41',
'New_brunswick', '9765451243', 'M', '7', '65000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500011', 'John', '28',
'Philidelphia', '9765451244', 'A', '3', '45000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500012', 'Tom', '32', 'Edison',
'9765451245', 'N', '5', '60000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500013', 'Mohammed', '36',
'Delaware', '9765451246', 'M', '7', '65000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500014', 'Ling_yu', '26', 'Boston',
'9765451247', 'A', '2', '38000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500015', 'xing_li', '26', 'Dallas',
'9765451248', 'N', '3', '45000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500016', 'Abhisheik', '32',
'New_York', '9765451249', 'M', '5', '60000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500017', 'Arun', '41', 'Jersey-city',
'9765451250', 'A', '7', '65000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500018', 'Sameera', '28',
'New_brunswick', '9765451251', 'N', '3', '45000');
```

```
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500019', 'Sravya', '32',
'Philidelphia', '9765451252', 'M', '5', '60000');
INSERT INTO project_rough.nurse (Nrs_id, Nrs_Name, Age, Address, Mob,
W_Shift, Experience, Salary) VALUES ('500020', 'Jayalalitha', '36',
'Edison', '9765451253', 'A', '7', '65000');

# Creating Doctor Table
CREATE TABLE project_rough.doctor
(
  Doc_Id INT UNSIGNED NOT NULL,
  Doc_Name VARCHAR(45) NOT NULL,
  Doc_Type VARCHAR(20) NOT NULL,
  Designation VARCHAR(20) NULL,
  Age INT UNSIGNED NULL,
  Address VARCHAR(45) NULL,
  MoB VARCHAR(10) NULL,
  Passed_from VARCHAR(20) NULL,
  Salary INT UNSIGNED NULL,
  PRIMARY KEY (Doc_Id)
);

# Polpulating Doctor Table
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90001', 'Raghu_rajan',
'General', 'MD', '35', 'Chicago', '987651234', 'Delhi', '200000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90002', 'Aravind_sub',
'Cardiolgist', 'Surgeon', '39', 'Delhi', '9876571235', 'Delhi', '150000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90003', 'Xong_li',
'Orthopeditian', 'Assosiate', '43', 'shangai', '9876571236', 'shangai',
'120000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90004', 'Xiang_Zi',
'Gynacologist', 'Director', '55', 'Beijing', '9876571237', 'Beijing',
'250000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90005', 'Robin_hal',
'Nuerologist', 'Assistant', '32', 'Dallas', '9876571238', 'Dallas',
'85000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90006', 'Peter_hall',
'Anestesia', 'Senior', '47', 'Detroit', '9876571239', 'Detroit', '200000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90007', 'Vinod_M',
'General', 'Surgeon', '35', 'Chennai', '9876571240', 'Chennai', '150000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90008', 'Singh_v',
'Cardiolgist', 'Assosiate', '52', 'Delhi', '9876571241', 'Chicago',
'120000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90009', 'Sagar_Kp',
'Orthopeditian', 'Director', '43', 'Vishakapatnam', '9876571242', 'Delhi',
'250000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90010', 'Lakshmi_nar',
'Gynacologist', 'Assistant', '38', 'Hyderabad', '9876571243', 'shangai',
'85000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90011', 'Liu_li',
'Nuerologist', 'Senior', '46', 'Michigan', '9876571244', 'Beijing',
'200000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90012', 'John_phillip',
'Anestesia', 'Surgeon', '55', 'Boston', '9876571245', 'Dallas', '150000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90013', 'Joshi_sk',
'General', 'Assosiate', '46', 'Banglore', '9876571246', 'Detroit',
'120000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90014', 'Venkat_K',
'Cardiolgist', 'Director', '31', 'Mumbai', '9876571247', 'Chennai',
'250000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90015', 'Xingfing_zhou',
'Orthopeditian', 'Assistant', '45', 'Shenzen', '9876571248', 'Delhi',
'85000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90016', 'Jessie_M',
'Gynacologist', 'Senior', '51', 'Jersey_city', '9876571249', 'shangai',
'200000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90017', 'Lawrence_wang',
'Nuerologist', 'Surgeon', '32', 'Louisville', '9876571250', 'Beijing',
'150000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90018', 'Christian_S',
'Anestesia', 'Assosiate', '34', 'Ashburn', '9876571251', 'Dallas',
'120000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90019', 'Leanardo_S',
'General', 'Director', '36', 'Los_Anageles', '9876571252', 'Detroit',
'250000');
INSERT INTO project_rough.doctor (Doc_Id, Doc_Name, Doc_Type, Designation,
Age, Address, MoB, Passed_from, Salary) VALUES ('90020', 'Alyssa_lueck',
'Cardiolgist', 'Assistant', '24', 'Toms_river', '9876571253', 'Chennai',
'85000');

# Creating Medicine Table
CREATE TABLE project_rough.medicine (
  Mdcn_Id INT UNSIGNED NOT NULL,
  Mdcn_name VARCHAR(20) NOT NULL,
  Company VARCHAR(20) NULL,
  M_date DATE NOT NULL,
  E_date DATE NOT NULL,
  Price INT NOT NULL,
  PRIMARY KEY (Mdcn_Id));

# Populating Medicine Table
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300001', 'crocin', 'Pfizer', str_to_date('12-14-
2015', '%m-%d-%Y'), str_to_date('12-14-2017', '%m-%d-%Y'), '20');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300002', 'Paracetnol', 'Cipla', str_to_date('12-15-
2015', '%m-%d-%Y'), str_to_date('12-15-2017', '%m-%d-%Y'), '25');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300003', 'Acutram', 'Excare', str_to_date('12-16-
2015', '%m-%d-%Y'), str_to_date('12-16-2017', '%m-%d-%Y'), '27');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300004', 'Atico', 'Astam_H_care', str_to_date('12-
17-2015', '%m-%d-%Y'), str_to_date('12-17-2017', '%m-%d-%Y'), '35');
```

```
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300005', 'Atucuron', 'Sun_Paharma',
str_to_date('12-18-2015', '%m-%d-%Y'), str_to_date('12-18-2017', '%m-%d-
%Y'), '46');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300006', 'Benergerene',
'IND_SWIFT_ltd',str_to_date('12-19-2015', '%m-%d-%Y') , str_to_date('12-19-
2017', '%m-%d-%Y'), '150');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300007', 'Bioglandin',
'United_Biotech',str_to_date('12-20-2015', '%m-%d-%Y') , str_to_date('12-
20-2017', '%m-%d-%Y'), '234');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300008', 'Cabergoline', 'Pfizer',str_to_date('12-
21-2015', '%m-%d-%Y') , str_to_date('12-21-2017', '%m-%d-%Y'), '167');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300009', 'Carmustine', 'Cipla', str_to_date('12-22-
2015', '%m-%d-%Y'), str_to_date('12-22-2017', '%m-%d-%Y'), '23');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300010', 'Degarelix', 'Excare',str_to_date('12-23-
2015', '%m-%d-%Y') , str_to_date('12-23-2017', '%m-%d-%Y'), '45');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300011', 'Lomustine', 'Astam_H_care',
str_to_date('12-24-2015', '%m-%d-%Y'), str_to_date('12-24-2017', '%m-%d-
%Y'), '76');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300012', 'Megestrol', 'Sun_Paharma',
str_to_date('12-25-2015', '%m-%d-%Y'), str_to_date('12-25-2017', '%m-%d-
%Y'), '33');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300013', 'Oxytocin',
'IND_SWIFT_ltd',str_to_date('12-26-2015', '%m-%d-%Y') , str_to_date('12-26-
2017', '%m-%d-%Y'), '64');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300014', 'Benergerene',
'United_Biotech',str_to_date('12-27-2015', '%m-%d-%Y') , str_to_date('12-
27-2017', '%m-%d-%Y'), '599');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300015', 'Bioglandin', 'IND_SWIFT_ltd',
str_to_date('12-28-2015', '%m-%d-%Y'), str_to_date('12-28-2017', '%m-%d-
%Y'), '675');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300016', 'Cabergoline',
'United_Biotech',str_to_date('12-29-2015', '%m-%d-%Y') , str_to_date('12-
29-2017', '%m-%d-%Y'), '123');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300017', 'Degarelix', 'Cipla', str_to_date('12-30-
2015', '%m-%d-%Y'), str_to_date('12-30-2017', '%m-%d-%Y'), '123');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300018', 'Lomustine', 'Excare', str_to_date('12-31-
2015', '%m-%d-%Y'), str_to_date('12-31-2017', '%m-%d-%Y'), '143');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300019', 'Megestrol', 'Astam_H_care',
str_to_date('1-1-2016', '%m-%d-%Y'), str_to_date('1-1-2018', '%m-%d-%Y'),
'24');
INSERT INTO project_rough.medicine (Mdcn_Id, Mdcn_name, Company, M_date,
E_date, Price) VALUES ('300020', 'Degarelix', 'Sun_Paharma',str_to_date('1-
2-2016', '%m-%d-%Y') , str_to_date('1-2-2018', '%m-%d-%Y'), '85');

# Creating Bill Table
CREATE TABLE project_rough.bill
```

```
(
  Bill_id INT UNSIGNED NOT NULL,
  Bill_for VARCHAR(20) NOT NULL check ( Bill_for in
('room','nursing','doctor')),
  Bill_Total INT UNSIGNED NOT NULL,
  PRIMARY KEY (Bill_id)
);

#Populating Bill Table
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80001', 'room', '3000');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80002', 'doctor', '1500');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80003', 'nursing', '2000');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80004', 'room', '1500');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80005', 'doctor', '3000');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80006', 'nursing', '1800');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80007', 'room', '2000');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80008', 'doctor', '1300');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80009', 'nursing', '2400');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80010', 'doctor', '1500');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80011', 'room', '1600');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80012', 'doctor', '3000');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80013', 'nursing', '1800');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80014', 'room', '2000');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80015', 'doctor', '1300');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80016', 'nursing', '2400');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80017', 'doctor', '1500');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80018', 'nursing', '1600');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80019', 'room', '4500');
INSERT INTO project_rough.bill (Bill_id, Bill_for, Bill_Total) VALUES
('80020', 'doctor', '1500');

# Creating Admission_room_rel Table
CREATE TABLE project_rough.admission_room_rel
(
  Admsn_id INT UNSIGNED NOT NULL,
  Room_Id INT UNSIGNED NOT NULL,
  Pat_Id INT NOT NULL,
  Date DATE NOT NULL,
  Time TIME NOT NULL,
  PRIMARY KEY (Admsn_id),
  INDEX Room_Id_idx (Room_Id ASC),
  INDEX Pat_Id_idx (Pat_Id ASC),
  CONSTRAINT Room_Id
    FOREIGN KEY (Room_Id)
    REFERENCES project_rough.room (Room_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT Pat_Id
    FOREIGN KEY (Pat_Id)
    REFERENCES project_rough.patient (Pat_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

# Populating Admission_room_rel Table
insert into project_rough.admission_room_rel values
(120001,1001,100001,str_to_date('11-5-2016
', '%m-%d-%Y'),'10:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120002', '1003', '100002', str_to_date('11-05-2016
', '%m-%d-%Y'), '11:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120003', '1006', '100003', str_to_date('11-06-2016
', '%m-%d-%Y'), '12:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120004', '1005', '100004', str_to_date('11-07-2016
', '%m-%d-%Y'), '13:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120005', '1004', '100005', str_to_date('11-08-2016
', '%m-%d-%Y'), '14:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120006', '1009', '100006', str_to_date('11-09-2016
', '%m-%d-%Y'), '15:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120007', '1002', '100007', str_to_date('11-10-2016
', '%m-%d-%Y'), '16:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120008', '1010', '100008', str_to_date('11-12-2016
', '%m-%d-%Y'), '17:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120009', '1007', '100009', str_to_date('11-13-2016
', '%m-%d-%Y'), '18:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120010', '1008', '100010', str_to_date('11-14-2016
', '%m-%d-%Y'), '19:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120011', '1011', '100011', str_to_date('11-15-2016
', '%m-%d-%Y'), '20:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120012', '1017', '100012', str_to_date('11-16-2016
', '%m-%d-%Y'), '21:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120013', '1012', '100013', str_to_date('11-17-2016
', '%m-%d-%Y'), '22:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120014', '1016', '100014', str_to_date('11-18-2016
', '%m-%d-%Y'), '23:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120015', '1013', '100015', str_to_date('11-19-2016
', '%m-%d-%Y'), '0:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120016', '1018', '100016', str_to_date('11-20-2016
', '%m-%d-%Y'), '1:12:00');
```

```
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120017', '1014', '100017', str_to_date('11-21-2016
', '%m-%d-%Y'), '2:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120018', '1019', '100018', str_to_date('11-22-2016
', '%m-%d-%Y'), '3:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120019', '1015', '100019', str_to_date('11-23-2016
', '%m-%d-%Y'), '4:12:00');
INSERT INTO project_rough.admission_room_rel (Admsn_id, Room_Id, Pat_Id,
Date, Time) VALUES ('120020', '1020', '100020', str_to_date('11-24-2016
', '%m-%d-%Y'), '5:12:00');

# Creating Assist_rel Table
CREATE TABLE project_rough.assist_rel (
  Serial_no INT UNSIGNED NOT NULL,
  Pati_id INT NOT NULL,
  Doct_id INT UNSIGNED NOT NULL,
  Date DATE NOT NULL,
  Time TIME NOT NULL,
  PRIMARY KEY (Serial_no),
  INDEX Pati_id_idx (Pati_id ASC),
  INDEX Doct_id_idx (Doct_id ASC),
  CONSTRAINT Pati_id
    FOREIGN KEY (Pati_id)
    REFERENCES project_rough.patient (Pat_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT Doct_id
    FOREIGN KEY (Doct_id)
    REFERENCES project_rough.doctor (Doc_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

#Populating Assist_rel Table
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('1', '100001', '90001', str_to_date('12-12-2016', '%m-%d-
%Y'), '10:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('2', '100002', '90002', str_to_date('12-13-2016', '%m-%d-
%Y'), '11:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('3', '100013', '90003', str_to_date('12-14-2016', '%m-%d-
%Y'), '12:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('4', '100004', '90004', str_to_date('12-15-2016', '%m-%d-
%Y'), '13:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('5', '100015', '90005', str_to_date('12-16-2016', '%m-%d-
%Y'), '14:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('6', '100006', '90006', str_to_date('12-17-2016', '%m-%d-
%Y'), '15:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('7', '100017', '90001', str_to_date('12-18-2016', '%m-%d-
%Y'), '16:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('8', '100008', '90002', str_to_date('12-19-2016', '%m-%d-
%Y'), '17:12:00');
```

```
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('9', '100009', '90003', str_to_date('12-20-2016', '%m-%d-
%Y'), '18:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('10', '100010', '90004', str_to_date('12-21-2016', '%m-%d-
%Y'), '19:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('11', '100011', '90005', str_to_date('12-22-2016', '%m-%d-
%Y'), '20:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('12', '100012', '90006', str_to_date('12-23-2016', '%m-%d-
%Y'), '21:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('13', '100003', '90007', str_to_date('12-24-2016', '%m-%d-
%Y'), '22:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('14', '100004', '90008', str_to_date('12-25-2016', '%m-%d-
%Y'), '23:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('15', '100015', '90009', str_to_date('12-26-2016', '%m-%d-
%Y'), '0:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('16', '100006', '90016', str_to_date('12-27-2016', '%m-%d-
%Y'), '1:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('17', '100017', '90017', str_to_date('12-28-2016', '%m-%d-
%Y'), '2:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('18', '100018', '90018', str_to_date('12-29-2016', '%m-%d-
%Y'), '3:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('19', '100009', '90019', str_to_date('12-20-2016', '%m-%d-
%Y'), '4:12:00');
INSERT INTO project_rough.assist_rel (Serial_no, Pati_id, Doct_id, Date,
Time) VALUES ('20', '100020', '90020', str_to_date('12-31-2016', '%m-%d-
%Y'), '5:12:00');

# Creating Nursingservice_rel Table
CREATE TABLE project_rough.nursingservice_rel (
  Ns_id INT UNSIGNED NOT NULL,
  Patient INT NOT NULL,
  Nurse_Id INT UNSIGNED NOT NULL,
  Rooms_id INT UNSIGNED NOT NULL,
  Date DATE NOT NULL,
  Time TIME NOT NULL,
  PRIMARY KEY (Ns_id, Patient, Nurse_Id),
  INDEX Patient_idx (Patient ASC),
  INDEX Nurse_Id_idx (Nurse_Id ASC),
  INDEX Room_id_idx (Rooms_id ASC),
  CONSTRAINT Patient
    FOREIGN KEY (Patient)
    REFERENCES project_rough.patient (Pat_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT Nurse_Id
    FOREIGN KEY (Nurse_Id)
    REFERENCES project_rough.nurse (Nrs_id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT Rooms_id
    FOREIGN KEY (Rooms_id)
    REFERENCES project_rough.room (Room_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

# Populating Nursingservice_rel Table
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110001', '100001', '500001', '1001',
str_to_date('11-5-2016', '%m-%d-%Y'), '10:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110002', '100002' ,'500002',
'1002',str_to_date('11-5-2016', '%m-%d-%Y'), '11:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110003',  '100003','500005',
'1003',str_to_date('11-6-2016', '%m-%d-%Y'), '12:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110004',  '100004','500006',
'1005',str_to_date('11-7-2016', '%m-%d-%Y'), '13:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110005', '100005', '500007',
'1004',str_to_date('11-8-2016', '%m-%d-%Y'), '14:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110006',  '100006','500003',
'1009',str_to_date('11-9-2016', '%m-%d-%Y'), '15:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110007',  '100007','500004', '1007',
str_to_date('11-10-2016', '%m-%d-%Y'), '16:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110008', '100008', '500008', '1006',
str_to_date('11-12-2016', '%m-%d-%Y'), '17:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110009',  '100009','500010',
'1008',str_to_date('11-13-2016', '%m-%d-%Y'), '18:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110010',  '100010','500009',
'1010',str_to_date('11-14-2016', '%m-%d-%Y'), '19:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110011',  '100011','500014',
'1011',str_to_date('11-15-2016', '%m-%d-%Y'), '20:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110012', '100012', '500011', '1013',
str_to_date('11-16-2016', '%m-%d-%Y'), '21:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110013',  '100013','500012', '1012',
str_to_date('11-17-2016', '%m-%d-%Y'), '22:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110014',   '100014','500013','1014',
str_to_date('11-18-2016', '%m-%d-%Y'), '23:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110015',  '100015', '500017',
'1015',str_to_date('11-19-2016', '%m-%d-%Y'), '0:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110016',  '100016','500016',
'1017',str_to_date('11-20-2016','%m-%d-%Y') ,'1:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110017',  '100017','500015',
'1019',str_to_date('11-21-2016', '%m-%d-%Y'), '2:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110018',  '100018','500020',
'1018',str_to_date('11-22-2016', '%m-%d-%Y'), '3:12:00');
```

```
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110019',  '100019','500019',
'1020',str_to_date('11-23-2016', '%m-%d-%Y'), '4:12:00');
INSERT INTO project_rough.nursingservice_rel (Ns_id, Patient, Nurse_Id,
Rooms_id, Date, Time) VALUES ('110020',  '100020','500018',
'1016',str_to_date('11-24-2016', '%m-%d-%Y'), '5:12:00');

# Creating Prescription_rel Table
CREATE TABLE project_rough.prescription_rel (
  Prs_Id INT UNSIGNED NOT NULL,
  Medicine_Id INT UNSIGNED NOT NULL,
  Doctor_Id INT UNSIGNED NOT NULL,
  PatientID INT NOT NULL,
  Date DATE NOT NULL,
  Time TIME NOT NULL,
  Fees INT NOT NULL,
  PRIMARY KEY (Prs_Id, Medicine_Id),
  INDEX Doctor_Id_idx (Doctor_Id ASC),
  INDEX PatientID_idx (PatientID ASC),
  INDEX Medicine_Id_idx (Medicine_Id ASC),
  CONSTRAINT Doctor_Id
    FOREIGN KEY (Doctor_Id)
    REFERENCES project_rough.doctor (Doc_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT PatientID
    FOREIGN KEY (PatientID)
    REFERENCES project_rough.patient (Pat_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT Medicine_Id
    FOREIGN KEY (Medicine_Id)
    REFERENCES project_rough.medicine (Mdcn_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

# Populating Prescription_rel Table
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70001', '300001', '90001', '100001',
str_to_date('10-12-2016', '%m-%d-%Y'), '10:10:00', '1200');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70002', '300012', '90002', '100002',
str_to_date('10-13-2016', '%m-%d-%Y'), '10:12:10', '1230');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70003', '300001', '90003', '100013',
str_to_date('10-14-2016', '%m-%d-%Y'), '10:14:20', '2221');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70004', '300002', '90004', '100004',
str_to_date('10-15-2016', '%m-%d-%Y'), '10:16:30', '4281');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70005', '300003', '90005', '100015',
str_to_date('10-16-2016', '%m-%d-%Y'), '10:18:40', '2261');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70006', '300004', '90006', '100006',
str_to_date('10-17-2016', '%m-%d-%Y'), '10:20:50', '1499');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70007', '300016', '90001', '100017',
str_to_date('10-18-2016', '%m-%d-%Y'), '10:23:00', '1588');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70008', '300017', '90002', '100008',
str_to_date('10-19-2016', '%m-%d-%Y'), '10:25:10', '1654');
```

```
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70009', '300018', '90003', '100009',
str_to_date('10-20-2016', '%m-%d-%Y'), '10:27:20', '1322');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70010', '300010', '90004', '100010',
str_to_date('10-21-2016', '%m-%d-%Y'), '10:29:30', '1186');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70011', '300011', '90005', '100011',
str_to_date('10-22-2016', '%m-%d-%Y'), '10:31:40', '2000');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70012', '300012', '90006', '100012',
str_to_date('10-23-2016', '%m-%d-%Y'), '10:33:50', '1500');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70013', '300001', '90007', '100003',
str_to_date('10-24-2016', '%m-%d-%Y'), '10:36:00', '2300');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70014', '300002', '90008', '100004',
str_to_date('10-25-2016', '%m-%d-%Y'), '10:38:10', '1567');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70015', '300003', '90009', '100015',
str_to_date('10-26-2016', '%m-%d-%Y'), '10:40:20', '1234');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70016', '300004', '90016', '100006',
str_to_date('10-27-2016', '%m-%d-%Y'), '10:42:30', '1334');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70017', '300016', '90017', '100017',
str_to_date('10-28-2016', '%m-%d-%Y'), '10:44:40', '1424');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70018', '300017', '90018', '100018',
str_to_date('10-29-2016', '%m-%d-%Y'), '10:46:50', '1289');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70019', '300018', '90019', '100009',
str_to_date('10-30-2016', '%m-%d-%Y'), '10:49:00', '1287');
INSERT INTO project_rough.prescription_rel (Prs_Id, Medicine_Id, Doctor_Id,
PatientID, Date, Time, Fees) VALUES ('70020', '300019', '90020', '100020',
str_to_date('10-31-2016', '%m-%d-%Y'), '10:51:10', '1987');

# Creating Payment_rel Table
CREATE TABLE project_rough.payment_rel
(
  payment_id INT UNSIGNED NOT NULL,
  payment_type VARCHAR(20) NOT NULL check ( payment in ("cash" or "cheque"
or "credit" or "debit")),
  payment_date DATE NOT NULL,
  Patient_Id INT NOT NULL,
  PRIMARY KEY (payment_id),
  INDEX Pat_Id_idx (Patient_Id ASC),
  CONSTRAINT Patient_Id
    FOREIGN KEY (Patient_Id)
    REFERENCES project_rough.patient (Pat_Id)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

# Populating Payment_rel Table
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600001', 'cash', str_to_date('10-12-
2016', '%m-%d-%Y'), '100001');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600002', 'credit',str_to_date('10-13-
2016', '%m-%d-%Y') , '100002');
```

```
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600003', 'debit', str_to_date('10-14-
2016', '%m-%d-%Y'), '100013');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600004', 'cheque',str_to_date('10-15-
2016', '%m-%d-%Y') , '100004');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600005', 'debit', str_to_date('10-16-
2016', '%m-%d-%Y'), '100015');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600006', 'cheque', str_to_date('10-17-
2016', '%m-%d-%Y'), '100006');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600007', 'credit',str_to_date('10-18-
2016', '%m-%d-%Y') , '100017');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600008', 'debit',str_to_date('10-19-
2016', '%m-%d-%Y') , '100008');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600009', 'cheque',str_to_date('10-20-
2016', '%m-%d-%Y') , '100009');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600010', 'debit', str_to_date('10-21-
2016', '%m-%d-%Y'), '100010');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600011', 'cheque',str_to_date('10-22-
2016', '%m-%d-%Y') , '100011');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600012', 'cash', str_to_date('10-23-
2016', '%m-%d-%Y'), '100012');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600013', 'cash', str_to_date('10-24-
2016', '%m-%d-%Y'), '100003');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600014', 'credit',str_to_date('10-25-
2016', '%m-%d-%Y') , '100004');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600015', 'debit',str_to_date('10-26-
2016', '%m-%d-%Y') , '100015');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600016', 'cheque',str_to_date('10-27-
2016', '%m-%d-%Y') , '100006');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600017', 'debit',str_to_date('10-28-
2016', '%m-%d-%Y') , '100017');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600018', 'cheque',str_to_date('10-29-
2016', '%m-%d-%Y') , '100018');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600019', 'cash',str_to_date('10-30-
2016', '%m-%d-%Y') , '100009');
INSERT INTO project_rough.payment_rel (payment_id, payment_type,
payment_date, Patient_Id) VALUES ('600020', 'credit',str_to_date('10-31-
2016', '%m-%d-%Y') , '100020');

# Creating Bill Table
CREATE TABLE project_rough.pat_bill (
    id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Pat INT NOT NULL,
    Bill_id INT UNSIGNED NOT NULL,
    key (Pat, Bill_id),
    CONSTRAINT Pat FOREIGN KEY (Pat)
        REFERENCES project_rough.patient (Pat_Id)
        ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT Bill_id FOREIGN KEY (Bill_id)
        REFERENCES project_rough.bill (Bill_id)
        ON DELETE NO ACTION ON UPDATE NO ACTION
);

# Populating Bill Table
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('1',
'100001', '80001');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('2',
'100006', '80002');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('3',
'100007', '80003');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('4',
'100008', '80004');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('5',
'100009', '80005');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('6',
'100010', '80006');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('7',
'100002', '80007');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('8',
'100003', '80008');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('9',
'100004', '80009');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('10',
'100005', '80010');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('11',
'100017', '80011');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('12',
'100009', '80012');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('13',
'100010', '80013');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('14',
'100011', '80014');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('15',
'100012', '80015');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('16',
'100013', '80016');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('17',
'100014', '80017');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('18',
'100015', '80018');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('19',
'100016', '80019');
INSERT INTO project_rough.pat_bill (id, Pat, Bill_id) VALUES ('20',
'100018', '80020');
```

Role Sheet Template

Students Tasks	Abhinav Singh	Jing Li	Suchendra kuchipudi
Define Requirements	X	X	X
Develop Conceptual Model	X		X
Convert Conceptual Model to Relational Model	X	X	
Normalization of Relational Model	X	X	X
Development of DDL	X	X	X
Development of DML		X	X
Generation of Data samples	X	X	X
Documentation	X	X	X