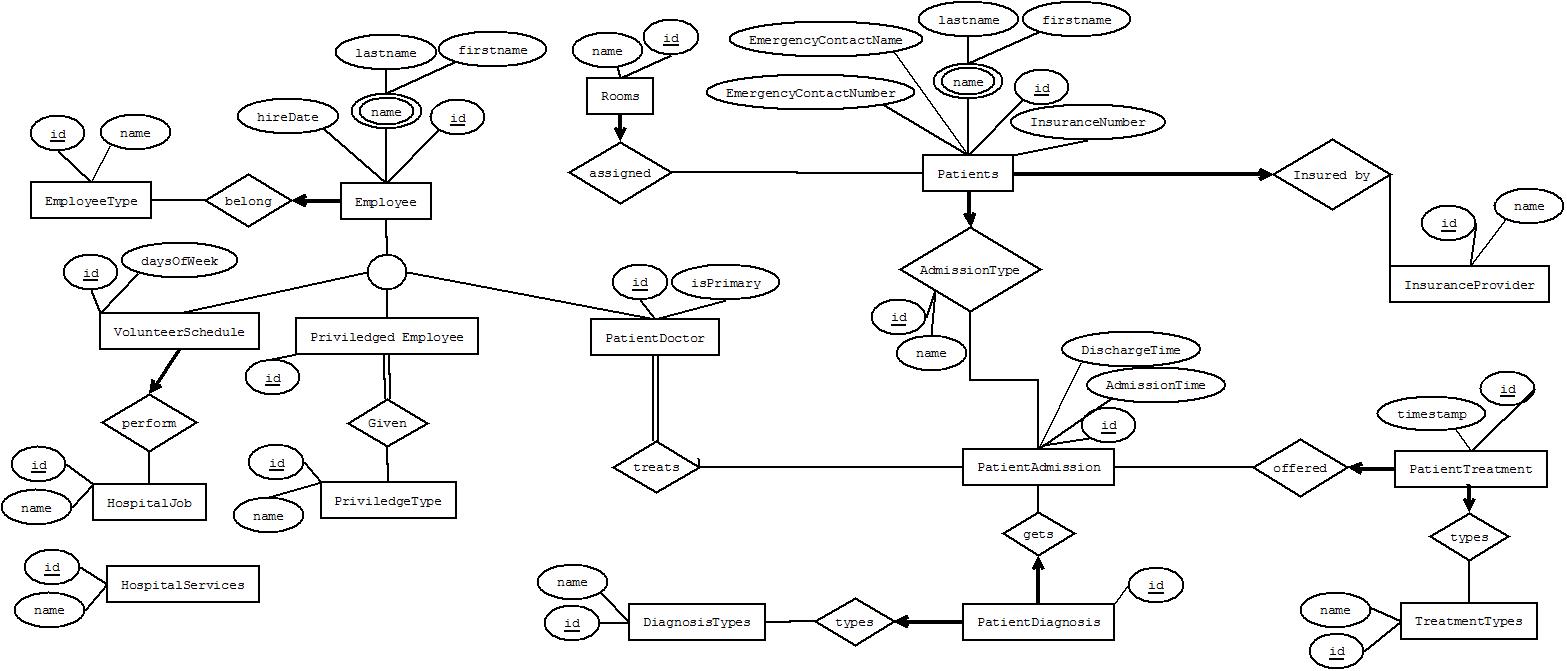
***Database Design and Implementation Project***

*Ravinder Putta (rzp0038)*

1. **Specification of the Conceptual Model**
2. Entity-Relationship Model



1. Assumptions and Constraints

* *Employees* are workers and can be either volunteers *VolunteerSchedule* or *hospital* *employees* i.e. *doctors*, *nurses*, *technicians*, *staff* and *administrators*. They have the following attributes: *firstname, lastname, hireDate and a unique id.* The *Employees* have an *EmployeeType* which states their employment type or designation.
* The *VolunteerSchedule* have a *HospitalJob* and offer services such as giftshop, information desk, snack carts, reading carts. The *VolunteerSchedule* also have a unique *volunteerId* along with their *employeeId*. They also have the attribute *DaysOf Week* which keeps track of their work days.
* The hospital staff work in non-medical areas such as the cafeteria, gift shop, janitorial services etc; which is listed in the *HospitalJob* entity which has a unique *id* and a name.
* The priviledges of the doctor for admitting and consulting services are listed in the *PriviledgeType.* An *PriviledgedEmployee* which is a subtype of *Employee* has at least one *PriviledgeType.* A *PriviledgedEmployee* also has a name and a unique *PriviledgedEmployeeID* along with the *employeeId.*
* All doctors have *consultingPriviledges. S*ome doctors have *admittingPriviledges* that allow them to admit *patients*. This condition depends on the presence of consistent data in the database.
* A *patient* after being admitted to the hospital by a *PriviledgedEmployee* doctor having *admittingPriviledges*  is assigned a unique *patientId*  and is assigned to a *room* which has a *number* and a unique *id.*
* Each *patient* also has the *emergency* *contact’s* *name* and *number* as well as their *InsuranceNumber* along with the details of the *InsuranceProvider*, listed as the attributes. The *InsuranceProvider* details include its *unique id* and *name.* The *Patient* has exactly one *InsuranceProvider.*
* The assignment of the *patient* to a *room* is done by an administrative employee. This condition depends on the presence of consistent data in the database.
* A *room* accommodates exactly one patient and has the *patientId* as its foreign key attribute.
* The doctor who handles admitted patients is their primary doctor(*PatientDoctor)*. The *PatientDoctor.*  The *PatientDoctor* handles atleast one *PatientAdmission*. The *PatientDoctor* gives a *PatientDiagnosis* and offers a *PatientTreatment.*
* When a patient is admitted, we keep track of the admissiontime and dischargetime of the *PatientAdmission.*
* A *PatientDiagnosis* is given to exactly one admitted *patient.* The *PatientDiagnosis has a PatientAdmissionId, a DiagnosisId and exactly one DiagnosisTypes.*
* A *PatientTreatment* is given to exactly one admitted *patient.* The *PatientTreatment has a PatientAdmissionId, a PatientTreatmentID,exactly one TreatmentTypes and timestamp* to keep track of the date and time when the treatment was offered.
* The *PatientDoctor* can assign multiple doctors or staffs to a *PatientAdmission* who can offer PatientTreatment. This condition depends on the presence of consistent data in the database.
* At the time of admission, a *PatientDoctor* provides an initial diagnosis which may change while the patient is receiving inpatient services. This condition depends on the presence of consistent data in the database.
* When a *PatientAdmission* ‘s *PatientDoctor* decides that the necessary course of PatientTreatment has ended, that *PatientAdmission* is discharged from the hospital by an administrative employee. This condition depends on the presence of consistent data in the database.
* Hospital offers Inpatient and Outpatient services and it is administered by hospital employees. This condition depends on the presence of consistent data in the database.

1. **Specification of the Implementation Model**
2. Relational Schema

**EmployeeType** (Id : SERIAL, Name : STRING );

**InsuranceProvider**(Id : SERIAL, Name : STRING );

**HospitalJob** (Id : SERIAL,Name : STRING );

**PrivlidgeType** (Id : SERIAL, : STRING );

**TreatmentType** (Id : SERIAL, Name : STRING );

**DiagnosisType** (Id : SERIAL, Name : STRING );

**AdmissionType** (Id : SERIAL, Name : STRING );

**Employee** (Id : SERIAL, FirstName : STRING , LastName : STRING , EmployeeTypeId : INTEGER (FOREIGN KEY References EmployeeType(Id)), HireDate : DATETIME);

**PrivlidgeEmployee** (Id : SERIAL, PrivlidgeTypeId : INTEGER (FOREIGN KEY REFERENCES PrivlidgeType(Id)), EmployeeId : INTEGER (FOREIGN KEY REFERENCES Employee(Id)) );

**Patient** (Id : SERIAL, FirstName : STRING , LastName : STRING , EContactName : STRING , EContactNumber : STRING , InsuranceNumber : STRING , InsuranceProviderId : INTEGER (FOREIGN KEY REFERENCES InsuranceProvider(Id)));

**PatientAdmission** (Id : SERIAL, PatientId : INTEGER (FOREIGN KEY REFERENCES Patient(Id)), AdmissionTypeId : INTEGER (FOREIGN KEY REFERENCES AdmissionType (Id)), AdmissionTime : DATETIME, DischargeTime : DATETIME);

**Room** (Id : SERIAL, Number : STRING , PatientId : INTEGER (FOREIGN KEY REFERENCES Patient(Id));

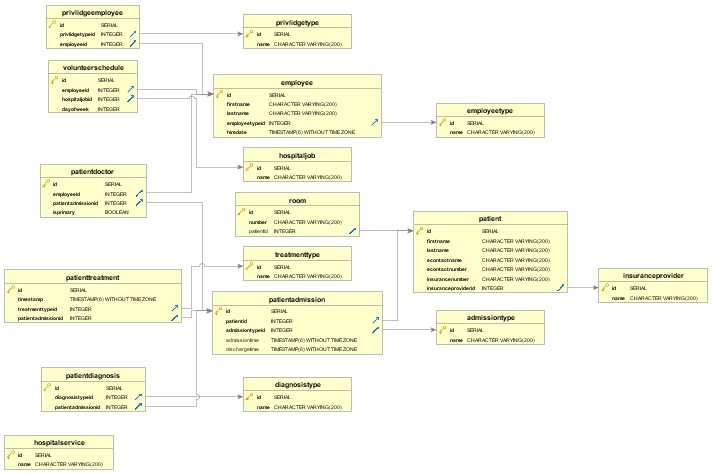
**PatientDoctor** (Id : SERIAL, EmployeeId : INTEGER (FOREIGN KEY REFERENCES Employee(Id)), PatientAdmissionId : INTEGER (FOREIGN KEY REFERENCES PatientAdmission (Id)), IsPrimary : BOOLEAN );

**PatientTreatment** (Id : SERIAL, TimeStamp : DATETIME , TreatmentTypeId : INTEGER (FOREIGN KEY REFERENCES TreatmentType (Id)), PatientAdmissionId : INTEGER (FOREIGN KEY REFERENCES PatientAdmission (Id) ) );

**PatientDiagnosis** (Id : SERIAL, DiagnosisTypeId : INTEGER (FOREIGN KEY REFERENCES DiagnosisType (Id) ), PatientAdmissionId : INTEGER (FOREIGN KEY REFERENCES PatientAdmission (Id)) );

**VolunteerSchedule** (Id : SERIAL,

EmployeeId : INTEGER (FOREIGN KEY REFERENCES Employee (Id)), **HospitalJobId** : INTEGER (FOREIGN KEY REFERENCES HospitalJob (Id)), DayOfWeek : INTEGER );



1. Translation of ER model into the listed schema

* The entities are mapped in as tables.
* The attributes are mapped in as attributes.
* The relationships are mapped as foreign keys.
* For 1:N /N:1 relationships, the primary key from “one” side is the foreign key in the “N” side and any relationship attributes becomes a part of the “N” entity’s table.
* Since its difficult to capture participation scenarios of type “atleast one”, we captured it as “one” cardinality on one side of the relationship set.
* The total participation constraints are captured on “one” cardinality side of the relationship set, by mapping the primary key on the other side as foreign key and assigning it “NOT NULL”.
* Composite attributes are treated as multiple single attributes in relation.
* For mapping superclasses and subclasses, we created a relation for the superclass and a distinct relation for each subclass. The primary key of the superclass becomes a foreign key of each subclass relation.

1. Text description of normalization to BCNF

Relational schemas and their functional dependencies:

1. **EmployeeType**

**EmployeeType**(Id) -> **EmployeeType**(name)

1. **InsuranceProvider**

**InsuranceProvider** (Id) -> **InsuranceProvider** (name)

1. **HospitalJob**

**HospitalJob** (Id) -> **HospitalJob** (name)

1. **HospitalService**

**HospitalService** (Id) -> **HospitalService** (name)

1. **PrivlidgeType**

**PrivlidgeType** (Id) -> **PrivlidgeType** (name)

1. **TreatmentType**

**TreatmentType** (Id) -> **TreatmentType** (name)

1. **DiagnosisType**

**DiagnosisType** (Id) -> **DiagnosisType** (name)

1. **AdmissionType**

**AdmissionType** (Id) -> **AdmissionType** (name)

1. **Employee**

**Employee** (Id) -> FirstName

**Employee** (Id) -> LastName

**Employee** (Id) -> EmployeeTypeId

**Employee** (Id) -> HireDate

1. **PrivlidgeEmployee**

**PrivlidgeEmployee** (Id) -> PrivlidgeTypeId

**PrivlidgeEmployee** (Id) -> EmployeeId

1. **Patient**

**Patient** (Id) -> FirstName

**Patient** (Id) -> LastName

**Patient** (Id) -> EContactName

**Patient** (Id) -> EContactNumber

**Patient** (Id) -> InsuranceNumber

**Patient** (Id) -> InsuranceProviderId

1. **PatientAdmission**

**PatientAdmission** (Id) -> PatientId

**PatientAdmission** (Id) -> AdmissionTypeId

**PatientAdmission** (Id) -> AdmissionTime

**PatientAdmission** (Id) -> DischargeTime

1. **Room**

**Room** (Id) -> Room(Number)

**Room** (Id) -> PatientId

1. **PatientDoctor**

**PatientDoctor** (Id) -> EmployeeId

**PatientDoctor** (Id) -> PatientAdmissionId

**PatientDoctor** (Id) -> IsPrimary

1. **PatientTreatment**

**PatientTreatment** (Id) -> TimeStamp

**PatientTreatment** (Id) -> TreatmentTypeId

**PatientTreatment** (Id) -> PatientAdmissionId

1. **PatientDiagnosis**

**PatientDiagnosis** (Id) -> DiagnosisTypeId

**PatientDiagnosis** (Id) -> PatientAdmissionId

1. **VolunteerSchedule**

**VolunteerSchedule** (Id) -> EmployeeId

**VolunteerSchedule** (Id) -> **HospitalJobId**

**VolunteerSchedule** (Id) -> DayOfWeek

1. Assumption, Constraints and Decisions on 3NF

Our schema is already in BCNF, no additional steps are required to convert database into BCNF.

1. **Database Implementation**
2. Create Table Commands

START TRANSACTION;

-- Contains name information for Employee Types.

CREATE TABLE EmployeeType (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Insurance Providers.

CREATE TABLE InsuranceProvider(

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Hostpital Jobs.

CREATE TABLE HospitalJob (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Hospital Services.

CREATE TABLE HospitalService (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Privlidge Types.

CREATE TABLE PrivlidgeType (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Treatment Types.

CREATE TABLE TreatmentType (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Diagnosis Types.

CREATE TABLE DiagnosisType (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains name information for Admission Types.

CREATE TABLE AdmissionType (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(200) NOT NULL

);

-- Contains information on all Employees, Volunteer or otherwise, in the Hospital.

CREATE TABLE Employee (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(200) NOT NULL,

LastName VARCHAR(200) NOT NULL,

EmployeeTypeId INTEGER NOT NULL,

HireDate DATETIME NOT NULL,

FOREIGN KEY (EmployeeTypeId )

REFERENCES EmployeeType(Id)

);

-- Links the Employee table with PrivlidgeType information.

CREATE TABLE PrivlidgeEmployee (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

PrivlidgeTypeId INTEGER NOT NULL,

EmployeeId INTEGER NOT NULL,

FOREIGN KEY (PrivlidgeTypeId)

REFERENCES PrivlidgeType(Id),

FOREIGN KEY (EmployeeId)

REFERENCES Employee(Id)

);

-- Contains information on all past and present Patients in the Hospital.

CREATE TABLE Patient (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(200) NOT NULL,

LastName VARCHAR(200) NOT NULL,

EContactName VARCHAR(200) NOT NULL,

EContactNumber VARCHAR(200) NOT NULL,

InsuranceNumber VARCHAR(200) NOT NULL,

InsuranceProviderId INTEGER NOT NULL,

FOREIGN KEY (InsuranceProviderId)

REFERENCES InsuranceProvider(Id)

);

-- Contains information on Patients specific to a certain admission.

CREATE TABLE PatientAdmission (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

PatientId INTEGER NOT NULL,

AdmissionTypeId INTEGER NOT NULL,

AdmissionTime DATETIME NULL,

DischargeTime DATETIME NULL,

FOREIGN KEY (PatientId)

REFERENCES Patient(Id),

FOREIGN KEY (AdmissionTypeId)

REFERENCES AdmissionType(Id)

);

-- Contains information on Rooms in the Hospital.

CREATE TABLE Room (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

Number VARCHAR(200) NOT NULL,

PatientId INTEGER NULL,

FOREIGN KEY (PatientId)

REFERENCES Patient(Id)

);

-- Links a PatientAdmission to any medical staff who assisted in that admission.

CREATE TABLE PatientDoctor (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

EmployeeId INTEGER NOT NULL,

PatientAdmissionId INTEGER NOT NULL,

IsPrimary BOOLEAN NOT NULL,

FOREIGN KEY (EmployeeId)

REFERENCES Employee(Id),

FOREIGN KEY (PatientAdmissionId)

REFERENCES PatientAdmission(Id)

);

-- Links a PatientAdmission to the TreatmentType information.

CREATE TABLE PatientTreatment (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

TimeStamp DATETIME NOT NULL,

TreatmentTypeId INTEGER NOT NULL,

PatientAdmissionId INTEGER NOT NULL,

FOREIGN KEY (TreatmentTypeId)

REFERENCES TreatmentType(Id),

FOREIGN KEY (PatientAdmissionId)

REFERENCES PatientAdmission(Id)

);

-- Links a PatientAdmission to the DiagnosisType information.

CREATE TABLE PatientDiagnosis (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

DiagnosisTypeId INTEGER NOT NULL,

PatientAdmissionId INTEGER NOT NULL,

FOREIGN KEY (DiagnosisTypeId)

REFERENCES DiagnosisType(Id),

FOREIGN KEY (PatientAdmissionId)

REFERENCES PatientAdmission(Id)

);

-- A schedule that tracks who works various non-medical stations in the Hospital.

CREATE TABLE VolunteerSchedule (

Id INTEGER NOT NULL AUTO\_INCREMENT PRIMARY KEY,

EmployeeId INTEGER NOT NULL,

HospitalJobId INTEGER NOT NULL,

DayOfWeek INTEGER NOT NULL,

FOREIGN KEY (HospitalJobId)

REFERENCES HospitalJob(Id),

FOREIGN KEY (EmployeeId)

REFERENCES Employee(Id)

);

COMMIT;

1. All data and associated INSERT commands

-- Data Entry Below Here

INSERT INTO AdmissionType (Name) VALUES ('Outpatient'), ('Inpatient');

INSERT INTO EmployeeType (Name) VALUES ('Volunteer'), ('Doctor'), ('Nurse'), ('Technician'), ('Staff'), ('Administrator');

INSERT INTO HospitalService (Name) VALUES ('Outpatient'), ('Inpatient');

INSERT INTO HospitalJob (Name) VALUES ('Cafeteria'), ('Gift shop'), ('Janitorial services'), ('Information desk'), ('Snack carts'), ('Reading carts');

INSERT INTO PrivlidgeType (Name) VALUES ('Consulting privilege'), ('Admitting privilege');

INSERT INTO InsuranceProvider (Name) VALUES ('BlueCross BlueShield'), ('UnitedHealthcare'), ('Humana'), ('aetna');

INSERT INTO TreatmentType (Name) VALUES ('Devliver baby'), ('Insert heart stint'), ('Acetaminphen 100mg'), ('Cabozantinib Tablets 10mg'), ('Rabies Vaccine');

INSERT INTO DiagnosisType (Name) VALUES ('Pregnant'), ('Heart blockage'), ('Headache'), ('Rabies');

INSERT INTO Room(Number) VALUES ('100'), ('101'), ('102'), ('103'), ('200'), ('201'), ('202'), ('300');

INSERT INTO Employee (FirstName, LastName, EmployeeTypeID, HireDate) VALUES

('Jane', 'Thompson', 1, '2012-01-01 08:00'),

('Gordon', 'Smith', 1, '2012-01-02 08:00'),

('Dan', 'Plymouth', 1, '2012-01-06 08:00'),

('Jordan', 'Landis', 2, '2012-01-10 08:00'),

('Gwen', 'Davis', 2, '2015-01-01 09:00'),

('Lee', 'Miller', 2, '2012-07-01 08:00'),

('Tina', 'Ramsey', 3, '2013-01-02 10:00'),

('Patricia', 'Copper', 3, '2012-01-01 08:00'),

('Dot', 'Carpenter', 3, '2012-01-05 08:00'),

('Dennis', 'Port', 4, '2012-04-05 08:00'),

('Weston', 'Black', 4, '2012-01-01 08:00'),

('Patrick', 'Washington', 5, '2012-01-01 08:00'),

('Bobby', 'Jefferson', 5, '2012-04-01 08:00'),

('Rod', 'McPatrick', 5, '2014-01-09 08:00'),

('Hilda', 'Stone', 6, '2012-01-12 08:00'),

('Jim', 'Bob', 2, '2012-06-01 08:00');

INSERT INTO PrivlidgeEmployee (PrivlidgeTypeId, EmployeeId) VALUES

(1, 10), (1, 11), (2, 10), (2, 11);

INSERT INTO Patient (FirstName, LastName,EContactName, EContactNumber, InsuranceNumber, InsuranceProviderId) VALUES

('John', 'Smith', 'Margaret Smith', 1235553487, 20384579, 1),

('George', 'Jones', 'Sally Jones', 1235554589, 25903748, 2),

('Jennifer', 'Smith', 'Daryl Smith', 1235552345, 02987345, 3),

('Tom', 'Lord', 'Sammy Lord', 1235555748, 19287344, 4),

('Wade', 'Davis', 'Ronald McDonald', 1235559921, 55293478, 1),

('John', 'Garth', 'Richard Garth', 1235557123, 11803945, 2),

('Cooper', 'Parnell', 'Kelly Parnell', 1235550043, 22890534, 3),

('Logan', 'Spector', 'Molly Spector', 1235551178, 33593478, 4),

('Logan', 'Parnell', 'Magdalene Chavez', 1235551092, 44023894, 1),

('Sami', 'Yousif', 'Wendy Yousif', 1235559943, 55347895, 2),

('Jordan', 'Yousif', 'Margaret Smith', 1235557271, 66837423, 3),

('Jose', 'Chavez', 'Miguel Chavez', 1235550987, 77547893, 4),

('Austin', 'Rains', 'Mary Rains', 1235551029, 88589034, 1),

('Casey', 'Rains', 'Mary Rains', 1235558823, 99234789, 2),

('Brian', 'Cook', 'Tina Cook', 1235558181, 11384723, 3),

('Shane', 'Fichter', 'Dolly Fichter', 1235556327, 22102983, 4);

INSERT INTO PatientAdmission (PatientId, AdmissionTypeId, AdmissionTime, DischargeTime) VALUES

(1,1,'2015-01-01 08:30', NULL),

(2,1,'2013-01-01 08:30', NULL),

(3,1,'2016-06-01 08:30', NULL),

(4,2,'2015-01-01 08:30', '2015-01-13 10:30' ),

(5,2,'2014-01-01 08:30', '2014-02-03 11:00' ),

(6,2,'2013-01-01 08:30', '2013-06-03 09:05' ),

(6,2,'2016-06-01 08:30', NULL);

INSERT INTO PatientDiagnosis (DiagnosisTypeId, PatientAdmissionId) VALUES

(3,1), (3,2), (3,3), (1,4), (4,5), (4,6), (4,7);

INSERT INTO PatientAdmission (PatientId, AdmissionTypeId, AdmissionTime, DischargeTime) VALUES (3,2,'2016-07-26 7:00:00', NULL);

INSERT INTO PatientTreatment (TimeStamp, TreatmentTypeId, PatientAdmissionId)

VALUES

('2016-07-26 07:00:00', 3, 1),

('2016-07-26 010:00:00', 3, 1),

('2016-07-26 14:00:00', 3, 1),

('2016-07-26 17:00:00', 3, 1),

('2015-01-01 09:30:00', 4, 4);

INSERT INTO PatientAdmission (PatientId, AdmissionTypeId, AdmissionTime, DischargeTime)

VALUES

(5, 2, '2013-01-01 08:30:00', '2013-01-03 08:30:00'),

(5, 2, '2013-01-21 08:30:00', '2013-01-25 08:30:00');

INSERT INTO PatientDiagnosis (DiagnosisTypeId, PatientAdmissionId) VALUES

(1,8), (2,9), (3,10);

INSERT INTO VolunteerSchedule (EmployeeId, HospitalJobId, DayOfWeek) VALUES

(12,1,1), (12,2,2), (12,3,3), (12,1,4), (12,2,5), (12,3,6), (12,1,7),

(13,2,1), (13,3,2), (13,1,3), (13,2,4), (13,3,5), (13,1,6), (13,2,7),

(14,3,1), (14,1,2), (14,2,3), (14,3,4), (14,1,5), (14,2,6), (14,3,7),

(1,4,1), (1,5,2), (1,6,3), (1,4,4), (1,5,5), (1,6,6), (1,4,7),

(2,5,1), (2,6,2), (2,4,3), (2,5,4), (2,6,5), (2,4,6), (2,5,7),

(3,6,1), (3,4,2), (3,5,3), (3,6,4), (3,4,5), (3,5,6), (3,6,7);

INSERT INTO PatientAdmission (PatientId, AdmissionTypeId, AdmissionTime, DischargeTime)

VALUES

(7,1,'2013-01-01 08:30', NULL),

(8,1,'2015-08-01 08:30', NULL),

(9,1,'2013-06-02 08:30', NULL),

(10,2,'2015-01-05 08:30', '2015-01-13 10:30' ),

(11,2,'2014-01-01 08:30', '2014-02-03 11:00' ),

(12,2,'2013-04-01 08:30', '2013-06-03 09:05' ),

(13,2,'2016-06-01 08:30', NULL),

(14,1,'2015-07-04 08:30', NULL),

(15,1,'2013-10-01 08:30', NULL),

(16,1,'2016-06-01 08:30', NULL);

INSERT INTO PrivlidgeEmployee (PrivlidgeTypeId, EmployeeId) VALUES

(1, 4), (1, 5), (2, 4), (2, 5), (1, 6);

INSERT INTO PatientDoctor (EmployeeId, PatientAdmissionId, IsPrimary)

VALUES

(4, 1, true),(4, 2, true),(4, 3, true),(4, 4, true),(4, 5, true),

(4, 6, true),(4, 6, true),(4, 7, true),(4, 8, true),(4, 9, true),

(5, 10, true),(5, 11, true),(5, 12, true),(5, 13, true),(5, 14, true),

(5, 15, true),(5, 16, true),(5, 17, true),(5, 18, true),(5, 19, true),

(16, 1, false), (16, 2, false), (16, 3, false), (16, 4, false), (16, 5, false),

(16, 6, false), (16, 7, false), (16, 8, false), (16, 9, false), (16, 10, false),

(16, 11, false), (16, 12, false), (16, 13, false), (16, 14, false), (16, 15, false),

(16, 16, false), (16, 17, false), (16, 18, false), (16, 19, false), (16, 20, false),

(7, 1, false), (7, 2, false), (7, 3, false), (7, 4, false), (7, 5, false),

(8, 1, false), (8, 2, false), (8, 3, false), (8, 4, false), (8, 5, false),

(9, 1, false), (9, 2, false), (9, 3, false), (9, 4, false), (9, 5, false);

Insert into PatientTreatment (TimeStamp, TreatmentTypeId, PatientAdmissionId)

VALUES

('2016-06-01 09:30:00',3,7),

('2016-07-26 08:00:00',3,8);

INSERT INTO PatientAdmission(PatientId,AdmissionTypeId, AdmissionTime,DischargeTime)

VALUES

(6,2,'2013-05-01 08:30:00', '2013-05-02 08:30:00'),

(6,2,'2013-06-01 08:30:00', '2013-06-02 08:30:00'),

(6,2,'2013-07-01 08:30:00', '2013-07-02 08:30:00'),

(6,2,'2013-08-01 08:30:00', '2013-08-02 08:30:00'),

(6,2,'2013-09-01 08:30:00', '2013-09-02 08:30:00'),

(6,2,'2013-10-01 08:30:00', '2013-10-02 08:30:00'),

(5,2,'2009-05-01 08:30:00', '2009-05-02 08:30:00'),

(5,2,'2010-06-01 08:30:00', '2010-06-02 08:30:00');

UPDATE Room SET PatientID = 13 WHERE Id = 2;

1. Source Code

The source code for our project is present in the **SourceCode** folder.

1. Instructions to install, access and interact with the project
2. The project is hosted at <http://auburn.edu/~hollimi/Home.php>
   1. For each section, there is a menu on the interface. Below are the menus (Right hand side) of the application section wise .
   2. Rooms Utilization 🡪 Rooms Menu
   3. Patient Information 🡪 Patients
   4. Diagnosis & Treatment Information 🡪 Diagnosis & Treatment
   5. Employee Information 🡪 Employees
   6. To execute any query other than those provided in the menu, use the “Custom Query” Menu.

To connect to the database using any MySQL client as an admin, use the below command:

mysql -h acadmysql.duc.auburn.edu -u hollimi -p hollimidb

Database details:

Host Name = acadmysql.duc.auburn.edu

Database name = hollimidb

Username = hollimi

**Password = test21**