Hospital Management Database Design

Database Specification:

- Purpose
- Business Rules
- Design Requirements
- Design Decisions
- ERD diagram.

Snehal Mundhe , Abhishek Bhagwat

Database Purpose:

The purpose of this database is to maintain the data in all the domain and departments of the hospital. The database will be used by the admin, doctors, patients, nurse.

Business Rules:

- Admin may book one or more events.
- Each doctor may access one or more appointments.
- Each patient may check one or more appointments.
- Many patient may have many doctors.
- Multiple doctors may have one or more feedbacks.
- Multiple feedbacks are given to one or more visitors.
- Multiple feedbacks are given to one or more patients.
- Multiple lab report may have one or more inpatients.
- Multiple lab report may have one or more outpatients.
- Each inpatient may have one room.
- Each inpatient may have one or more nurses.
- Each nurse may have one or more patients.
- Each inpatient may have one bill.
- Each outpatient may have one bill.

Design Requirements:

- Use Crow's Foot Notation.
- Specify the primary key fields in each table by specifying PK beside the fields.
- Draw a line between the fields of each table to show the relationships between each table. This line should be pointed directly to the fields in each table that are used to form the relationship.
- Specify which table is on the one side of the relationship by placing a one next to the field where the line starts.
- Specify which table is on the many side of the relationship by placing a crow's feet symbol next to the field where the line ends.

Design Decisions:

Entity Name	Why Entity Included	How Entity is Related to Other
Admin	One of the primary purposes of this database entity is to check availability of doctors and book the appointments for the patients. Admin table is included in the database so as to make decision on fixing the appointment. The admin can access both the Doctors and Patients information for checking the suitable date for the appointment. The appointment ID, name of the doctor and patients along with the appointment date are added by the Admin to the Appointment database. The attributes of the Admin database are the security code, email, password.	Admin entity accesses the Doctor and Patient Entities. Admin adds the doctorID and patientID in the Appointment entity. In addition, it also adds the patientName and doctorName in the appointment table and chooses the available doctor and assigns the availability to the Appointment entity. In this way, the Admin entity is related to Doctor, Patient and Appointment entities. Admin entity may have by default access to all of the entities in the database. Admin and Appointment entity has one-many relationship.
Appointment	The key function of the Appointment table is to accommodate the appointment information added/updated by the admin. It is important to include the Appointment table as both the Patients and Doctors will know about their Appointment information. The appointment information can be accessed by Doctors and Patients.	Appointment entity is accessed by Doctor and Patient entities. For Doctor entities, each doctor may have multiple appointments. Multiple doctors can access the Appointment entity at the same time. Therefore, many-many relationship exists between Doctor and Appointment Entity. Similarly, multiple patients may have multiple appointments and multiple patients can access the Appointment Entity simultaneously. Therefore, Patient entity is related to the Appointment Entity with many-many relationship. In short, there exists many-many relationship between Appointment entity and Doctor & Patient Entities.

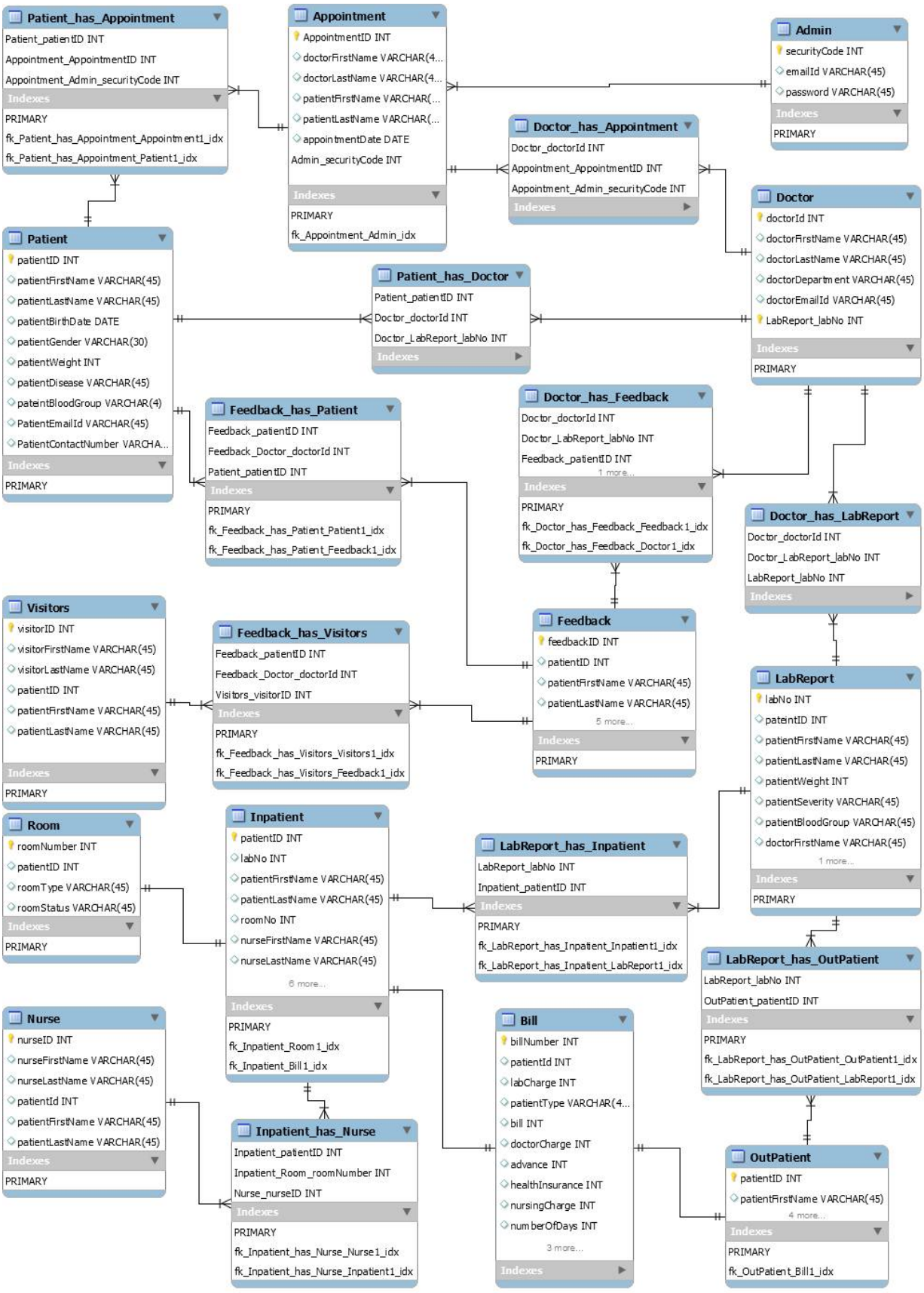
Patient	Patient entity is the most important entity of this database management system. This entity includes patient details such as patientID, patientName, age, gender, weight and so on as these attributes are important for the lab report that is to be generated. This lab report would be verified by the Doctor and then Feedback is provided to the Patients and Visitors.	Patient entity is related to the Appointment Entity. Many patients can access the Appointment Table simultaneously. Therefore, there exists many-many relationship between Patient and Appointment entities. Also, many patient may consult to many doctor at a time. Therefore, there exists many-many relationship between Patient and Doctor entities. In addition, Patient table receives multiple Feedbacks from multiple doctors. Therefore, there exists many-many relationship between Feedback and Patient Database.
Doctor	Doctor table has the key function in this database management system. This entity includes doctor's details. Doctor details are important to know about the doctors and the type of the disease they can treat. This information is useful for appointing doctors to patients according to the disease of the patients. Therefore, storing details of Doctors is important for Admin, for creating the Appointment Table.	Doctor access the Appointment table to check the appointments assigned to them. Many doctors can have many appointments and many doctors can access Appointment entity simultaneously. Therefore, there is many-many relationship between Doctor and Appointment entities. Doctor table is related to the lab report table. The lab report table stores the information about the doctor who generates/checks the lab report for the patient. Many doctors can simultaneously access many lab reports. Therefore, there exists many-many relationship between Doctor and Lab reports. In addition, the many Doctors may give many Feedbacks to each patient. Therefore, there is a many-many relationship between Doctor and Feedback tables.

Feedback	The Feedback table is important for Patients and Doctors. The Feedback table consists of the comments of the doctors about the patient's disease. It is important to include the Feedback about the Patients so that it would be easy for other doctors to evaluate the case of the patient. The major attributes of the Feedback table are name of the patient, comments about the patient, name of the doctor who has given comments about the patient.	Feedback table is associated with the Doctor table. Doctor generates feedback about the patient. Many doctors may have many feedbacks about a patient. Therefore, there exists many-many relationship between Doctor and Feedback. Also, many Feedbacks may be given to many patients and patient's visitors. Therefore, there exists many-many relationship between Feedback and Patient &Visitor entities.
Visitors	The patient's Visitor table is included in the database since in the real situations, it becomes necessary to let patient as well as patient's visitors/relatives know about the condition of the patient. It becomes very important for Hospitals to store information on who visits which patient, that is information about patient's visitors/relatives. Maintaining such information about patients and visitors would ensure safety of patients.	Visitor table forms relationship with the Feedback table. The Feedback generated from the Doctors may be given to Visitors. Multiple Feedbacks are generated from the multiple doctors. Multiple feedbacks about the patient can be given to multiple Visitors. Therefore, there exists many-many relationship between Feedback and Visitors entities.
LabReport	The Lab Report is principal database entity for any Hospital Management system, as it is the means to evaluate the condition of the patient. Apart from all the patient details the table includes the attributes such as severity of the disease, name of the doctor who treated the patient. Lab report is essential to make the decision on if the patient is serious and it should be admitted with emergency. Including Lab Report in the database helps doctor to decide on further steps that should be taken for the treatment of the patient.	Lab Report table is related to the Doctor entity. Doctor generates the Lab Report. Many doctors can generate many lab reports for the patient. Therefore, there exists many-many relationship between Doctor and Lab Report. The Lab report has the important attribute named severity that decides if the patient should be admitted in the Hospital or not. If the severity is high, patient is admitted and if severity is low patient is not admitted. Thus, lab report will carry the information about Inpatient and Outpatients. Thus, many Lab Reports may have many Inpatients. Therefore, there exists many-many relationship between Lab reports and Inpatients. Similarly, based on the severity, many Lab reports may have many Outpatients. Therefore, there exists many-many relationship between Lab Reports and Outpatients.

Inpatient	Inpatient database table is included in the database model as it will give clear idea on how many patients are admitted in the Hospital. It is important to maintain and track the data about the patients that are admitted as it eases the process for the management to provide best services to the patients. The important attributes about the Inpatient database entity are the room, nurse assigned and date of admission, health insurance. Keeping all this information is important for managing, treating, generating billing information, giving better services to the patient.	Inpatient is related to Lab Report. The lab Report decides on the basis of the severity of the disease if the patient is Inpatient or not. Therefore, many lab reports maintain the data on the severity of the patients that decides if the patient is admitted or not. Thus, there exists many-many relationship between Lab Reports as it can give information on many Inpatients. Also, single Inpatient is assigned a single room. Therefore, there exists one-one relationship between Inpatient and Room entity. In addition, the many nurses can be assigned to single Inpatient. Therefore, there exists one-many relationship between Inpatient and Nurses. In addition, Inpatient can generate only one Billing Information. Therefore, there exists one-one relationship between Inpatient and Bill entities.
OutPatient	Outpatient database table is included in the database model as it will give clear idea on how many patients are not admitted in the Hospital. It is important to maintain and track the data about the patients that are not admitted. Maintaining all kinds of data is important for any management.	Outpatient is related to the Lab Report since lab report maintains the data on the severity of the disease of the patient. If the patient has no severe condition then the patient is not admitted to the hospital. Based on the severity, the many lab reports can tell information on many patients that are not to be admitted. Therefore, there exists many-many relationship between Outpatient and Lab Reports. Also, each outpatient can have only one Billing Inoforamtion/Bill. Therefore, there exists one-one relationship between Outpatient and Bill entities.
Nurse	Maintaining all the data about employees of any management is important. Nurse employee holds	Nurse is related to Inpatient entity. Many nurses may be assigned to single Inpatient.

	key role in assisting Doctors in Hospital therefore, including this database entity is important. Nurse helps to assist the doctor in treatment of admitted Inpatients of the Hospital. Maintaining information about which Nurse is assigned to which Romm/Ward and which Inpatient is important as it avoids unnecessary confusion and aids in providing better service.	Therefore, there exists one-many relationship between Inpatient and Nurses.
Room	Room entity has the key function in database as it stores the important information on which patient is admitted to the respective rooms/wards. Therefore, it is necessary to include the information about the rooms and their availabilities for the patients to be admitted and admitted patients.	Room entity is related with Inpatients. It holds all the information about the availability of rooms, the Inpatients that are admitted. A single room can have only one Inpatient therefore, there exits one-one relationship between Inpatient and Room.
Bill	Billing should be included in the database as it will take all the charges including room assigned, medicine, hours of operation and so on. It is important for Hospital to maintain the record on each patient's billing information as it gives idea on hospital's revenue.	Billing is related to Inpatients and outpatients with one-one relationship.

Final ERD Diagram:


Patient_has_Appointment
Patient_patientID INT
Appointment_AppointmentID INT
Appointment_Admin_securityCode INT
Indexes
PRIMARY
fk_Patient_has_Appointment_Appointment1_idx
fk_Patient_has_Appointment_Patient1_idx
Appointment
AppointmentID INT
doctorFirstName VARCHAR(4...
doctorLastName VARCHAR(4...
patientFirstName VARCHAR(...
patientLastName VARCHAR(...
appointmentDate DATE
Admin_securityCode INT
Indexes
PRIMARY
fk_Appointment_Admin_idx
Admin
securityCode INT
emailId VARCHAR(45)
password VARCHAR(45)
Indexes
PRIMARY
Doctor_has_Appointment
Doctor_doctorId INT
Appointment_AppointmentID INT
Appointment_Admin_securityCode INT
Indexes
Doctor
doctorId INT
doctorFirstName VARCHAR(45)
doctorLastName VARCHAR(45)
doctorDepartment VARCHAR(45)
doctorEmailId VARCHAR(45)
LabReport_labNo INT
Indexes
PRIMARY
Patient
patientID INT
patientFirstName VARCHAR(45)
patientLastName VARCHAR(45)
patientBirthDate DATE
patientGender VARCHAR(30)
patientWeight INT
patientDisease VARCHAR(45)
pateintBloodGroup VARCHAR(4)
PatientEmailId VARCHAR(45)
PatientContactNumber VARCHA...
Indexes
PRIMARY
Patient_has_Doctor
Patient_patientID INT
Doctor_doctorId INT
Doctor_LabReport_labNo INT
Indexes
Feedback_has_Patient
Feedback_patientID INT
Feedback_Doctor_doctorId INT
Patient_patientID INT
Indexes
PRIMARY
fk_Feedback_has_Patient_Patient1_idx
fk_Feedback_has_Patient_Feedback1_idx
Doctor_has_Feedback
Doctor_doctorId INT
Doctor_LabReport_labNo INT
Feedback_patientID INT
1 more...
Indexes
PRIMARY
fk_Doctor_has_Feedback_Feedback1_idx
fk_Doctor_has_Feedback_Doctor1_idx
Doctor_has_LabReport
Doctor_doctorId INT
Doctor_LabReport_labNo INT
LabReport_labNo INT
Indexes
Visitors
visitorID INT
visitorFirstName VARCHAR(45)
visitorLastName VARCHAR(45)
patientID INT
patientFirstName VARCHAR(45)
patientLastName VARCHAR(45)
Indexes
PRIMARY
Feedback_has_Visitors
Feedback_patientID INT
Feedback_Doctor_doctorId INT
Visitors_visitorID INT
Indexes
PRIMARY
fk_Feedback_has_Visitors_Visitors1_idx
fk_Feedback_has_Visitors_Feedback1_idx
Feedback
feedbackID INT
patientID INT
patientFirstName VARCHAR(45)
patientLastName VARCHAR(45)
5 more...
Indexes
PRIMARY
LabReport
labNo INT
pateintID INT
patientFirstName VARCHAR(45)
patientLastName VARCHAR(45)
patientWeight INT
patientSeverity VARCHAR(45)
patientBloodGroup VARCHAR(45)
doctorFirstName VARCHAR(45)
1 more...
Indexes
PRIMARY
Room
roomNumber INT
patientID INT
roomType VARCHAR(45)
roomStatus VARCHAR(45)
Indexes
PRIMARY
Inpatient
patientID INT
labNo INT
patientFirstName VARCHAR(45)
patientLastName VARCHAR(45)
roomNo INT
nurseFirstName VARCHAR(45)
nurseLastName VARCHAR(45)
6 more...
Indexes
PRIMARY
fk_Inpatient_Room1_idx
fk_Inpatient_Bill1_idx
LabReport_has_Inpatient
LabReport_labNo INT
Inpatient_patientID INT
Indexes
PRIMARY
fk_LabReport_has_Inpatient_Inpatient1_idx
fk_LabReport_has_Inpatient_LabReport1_idx
LabReport_has_OutPatient
LabReport_labNo INT
OutPatient_patientID INT
Indexes
PRIMARY
fk_LabReport_has_OutPatient_OutPatient1_idx
fk_LabReport_has_OutPatient_LabReport1_idx
Nurse
nurseID INT
nurseFirstName VARCHAR(45)
nurseLastName VARCHAR(45)
patientId INT
patientFirstName VARCHAR(45)
patientLastName VARCHAR(45)
Indexes
PRIMARY
Inpatient_has_Nurse
Inpatient_patientID INT
Inpatient_Room_roomNumber INT
Nurse_nurseID INT
Indexes
PRIMARY
fk_Inpatient_has_Nurse_Nurse1_idx
fk_Inpatient_has_Nurse_Inpatient1_idx
Bill
billNumber INT
patientId INT
labCharge INT
patientType VARCHAR(4...
bill INT
doctorCharge INT
advance INT
healthInsurance INT
nursingCharge INT
numberOfDays INT
3 more...
Indexes
OutPatient
patientID INT
patientFirstName VARCHAR(45)
4 more...
Indexes
PRIMARY
fk_OutPatient_Bill1_idx