Hospital Database Management System

Team 10				
Team Members	Avinash Raikesh	Animesh Shukla	Shruti Manohar	Vandit Kothari
NU Email ID	raikesh.a@northeastern.edu	shukla.an@northeastern.edu	manohar.s@northeastern.edu	kothari.va@northeastern.edu

Preface:

1. OVERVIEW:

Healthcare is a field that demands efficiency when it comes to processing vast amounts of medical data and effectively treating patients. Within the last twenty years or so, many organizations have had to face the difficult task of transferring medical data from file processing systems into a robust relational database that can provide all parties with access to quick and reliable medical as well as financial information. Throughout the entire process, doctors have always played a fundamental role in assessing and prescribing medication, performing surgeries and monitoring patients. As per 2021, there are approximately 6000 hospitals and most of them use a proper hospital database management system. Yet, not all of those databases have been implemented properly and data security is another big issue. At the same time, it is very tough to conclude how effective that management system is for the staff of 6000 hospitals.

In accordance with HIPAA privacy standards, medical personnel are protected from disclosing patients information to fellow medical personnel, which means hospitals handle highly sensitive data that requires strong file security. In today's healthcare setting, this is enough to justify why file processing systems are antiquated. There have been many successful transitions, but some have faced numerous obstacles which have led to difficulties for hospitals, medical practitioners, and other hospital staff.

Choosing an RDBMS that is efficient is crucial when it comes to healthcare because workflow congestion can mean life or death for patients. It is critical to manage patient data, to arrange patient-doctor schedules, and to account for financial concerns of the parties in the process. A database management system will accommodate security concerns and update problems adequately if it is set up in a way that is able to take care of these issues.

2. IDEA OF PROJECT:

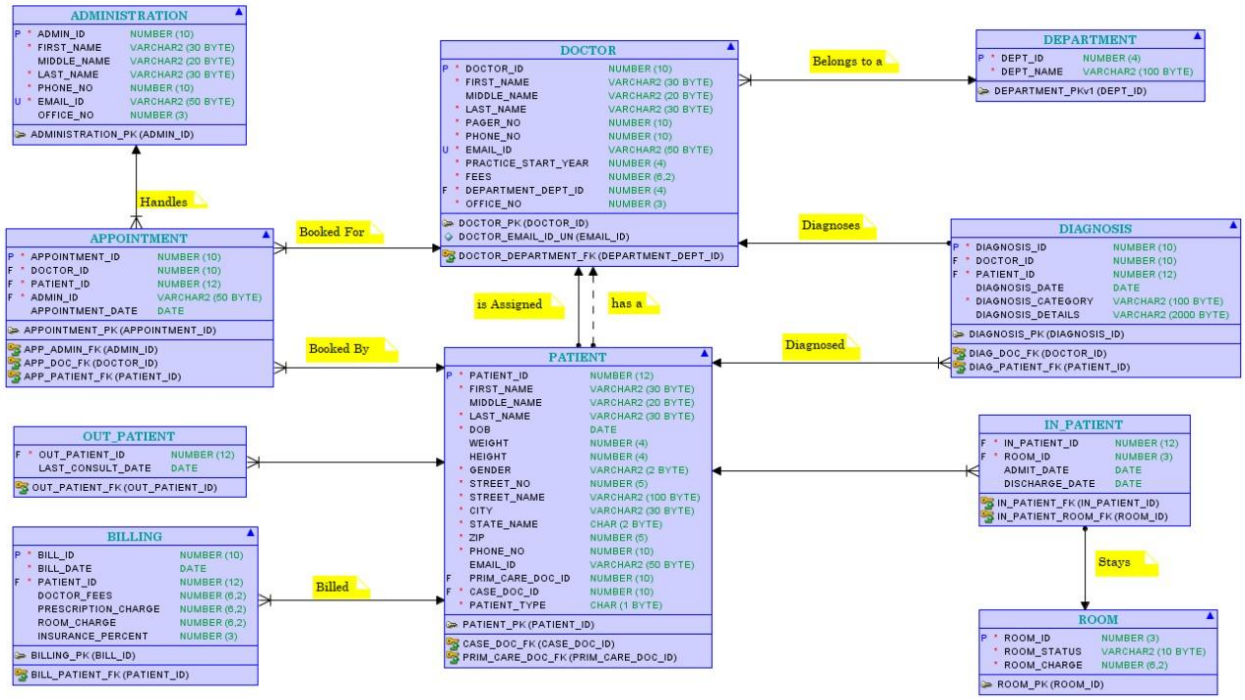
To develop a system that processes medical information quickly, securely, and efficiently, we will be building a relational database. Patients' medical records, appointment schedules, staff directories, room assignment information, billing data, and other details can be stored in the system. This system is easier to use, less time consuming, and more secure than usual file processing programs. In addition to improving healthcare management, the staff-patient relationship would also benefit from enhanced precision by automating the process. With a single secure platform, patients could schedule their appointments, access their billing statements, and manage their medical records. With this approach, you can minimize hospital congestion, which will allow you to get treated and cared for faster. As a result of our system, doctors can also retrieve/update medical notes more quickly, ensuring the best diagnostic practices are being used.

It will comprise doctors, patients, diagnoses, billing information, and plans for inpatient rooms. Administration will be able to view billing statements, assign patients to rooms based on availability, confirm/modify/delete appointments, and update patient/staff details. A doctor will be able to diagnose patients based on symptoms. Medical information is both confidential and subject to HIPAA laws, so all users will simply not have equal access to it. Patients won't have the option to share their records with others, and neither will doctors.

To make our proposal useful, we are assuming our hospital is still using a file processing system. The data intake process is as follows. A medical form is filled out online by the patient in advance. Patient information and medical history (name, address, height, weight, etc.) would be captured on this form by the admin who would then enter this information into a system that would update if the patient's condition changes. If a patient has previously visited a hospital, this information would be updated in their system; otherwise, it would be created if it's their first visit. Appointments with doctors are scheduled ahead of time, following a phone call from the patient. An administrator will determine their availability (offline) and then schedule the appointment. As soon as an appointment takes place, the doctor provides a proper diagnosis to the patient, which is entered into the medical records, updating that patient's profile. A bill will be issued to the patient following the appointment, containing the service charges. The patient's room is assigned by an administrator after they have been admitted. The patient's status is changed from inpatient to outpatient by an administrator after the patient has been discharged. The bill is administered one day after the patient leaves the hospital.

3. **ENTITY RELATIONSHIP DIAGRAM:**

When it comes to designing a database, Entity Relationship (ER) diagrams play a big role. Based on the business idea, we have drafted a representation of the ER model containing tables with relations.



4. TABLES:

Based on the business, we have created an idea of how many tables this system will consist of. These tables will define the database through column name, data type, constraints, and the description of those entities.

a. ADMINISTRATION Table:

This table will store the details of the administrative staff of the hospital. While some of the fields are optional, most of them should not be null.

Column Name	Data Type	Constraints	Description
ADMIN_ID	NUMBER(10)	Primary Key	Administrative ID
FIRST_NAME	VARCHAR(30)	Not Null	First Name of Administrator
MIDDLE_NAME	VARCHAR(30)		Middle Name of Administrator
LAST_NAME	VARCHAR(30)	Not Null	Last Name of Administrator
PHONE_NO	NUMBER(10)	Not Null	Administrator's contact information
EMAIL_ID	VARCHAR(50)	Unique, Not Null	
OFFICE_NO	NUMBER(3)	Not Null	Administrator's Office number

b. DOCTOR Table:

The doctor table will store all of the information of the doctors like name, contact details, years of experience, their departments,and many more.

Column Name	Data Type	Constraints	Description
DOCTOR_ID	NUMBER(10)	Primary Key	Doctor's ID
FIRST_NAME	VARCHAR(30)	Not Null	Doctor's First Name
MIDDLE_NAME	VARCHAR(30)		Doctor's Middle Name
LAST_NAME	VARCHAR(30)	Not Null	Doctor's Last Name
PAGER_NO	NUMER(10)	Not Null	Doctor's contact information
PHONE_NO	NUMBER(10)	Not Null	
EMAIL_ID	VARCHAR(50)	Unique, Not Null	
PRACTICE_START_YEAR	NUMBER(4)	Not Null	The year from which doctor started practicing medicine as a licensed physician
FEES	NUMBER(6,2)	Not Null	Charges for doctor's appointment/services (USD)
DEPT_ID	VARCHAR(10)	Foreign Key (DEPARTMENT – DEPT_ID), Not Null	Medical Department under which doctor works
OFFICE_NO	NUMBER(3)	Not Null	Doctor's Office number

c. PATIENT Table:

This is going to be one of the most important tables of our project as it will consist of all of the details of the patients.

Column Name	Data Type	Constraints	Description
PATIENT_ID	NUMBER(12)	Primary Key	Patient's ID
FIRST_NAME	VARCHAR(30)	Not Null	Patient's First Name
MIDDLE_NAME	VARCHAR(30)		Patient's Middle Name
LAST_NAME	VARCHAR(30)	Not Null	Patient's Last Name
DOB	DATE	Not Null	Patient's Date of Birth
WEIGHT	NUMBER(4)		Patient's weight (lbs)
HEIGHT	NUMBER(4)		Patient's height (cm)
GENDER	VARCHAR(2)	Not Null	Patient's gender (M/F/NA)
STREET_NO	NUMBER(5)		Patient's Address
STREET_NAME	VARCHAR(100)		
CITY	VARCHAR(30)		
STATE	CHAR(2)		
ZIP	NUMBER(5)		
PHONE_NO	NUMBER(10)	Not Null	Patient's contact Information
EMAIL_ID	VARCHAR(50)	Null	
PRIM_CARE_DOC_ID	NUMBER(10)	Foreign Key (DOCTOR – DOCTOR_ID), Nullable	Patient's primary care doctor

CASE_DOC_ID	NUMBER(10)	Foreign Key (DOCTOR – DOCTOR_ID), Not Null	Patient's current case doctor
PATIENT_TYPE	CHAR(1)	Not Null, Check	'I' = inpatient 'O' = outpatient

d. DIAGNOSIS Table:

The diagnosis table will have the data of the patients based on their treatment.

Column Name	Data Type	Constraints	Description
DIAGNOSIS_ID	NUMBER(10)	Primary Key	Patient's diagnosis ID
DOCTOR_ID	NUMBER(10)	Foreign Key (DOCTOR – DOCTOR_ID), Not Null	Doctor responsible for diagnosis
PATIENT_ID	NUMBER(12)	Foreign Key (PATIENT – PATIENT_ID), Not Null	Patient's ID who was diagnosed
DIAGNOSIS_DATE	DATE	Not Null	Date on which patient was diagnosed
DIAGNOSIS_CATEGORY	VARCHAR(100)	Not Null	Description of Medical ailments
DIAGNOSIS_DETAILS	VARCHAR(2000)	Not Null	Details of diagnosis and recommendation moving forward

e. IN_PATIENT Table:

The In patient table will only store the details of those who are currently admitted to the hospital.

Column Name	Data Type	Constraints	Description
IN_PATIENT_ID	NUMBER(12)	Foreign Key (PATIENT – PATIENT_ID), Not Null	Patient's ID who is admitted
ROOM_ID	NUMBER(3)	Foreign Key (ROOM – ROOM_ID), Not Null	Room number where patient is admitted.
ADMIT_DATE	DATE	Not Null	Date on which patient is admitted to the hospital
DISCHARGE_DATE	DATE	Null	Date on which patient is discharged from the hospital

f. OUT_PATIENT Table:

Unlike the In patient table, this table will only store the data of those patients who are discharged from the hospital as well as those who are visiting for the consultation with the doctor.

Column Name	Data Type	Constraints	Description
OUT_PATIENT_ID	NUMBER(12)	Foreign Key (PATIENT – PATIENT_ID), Not Null	Patient's ID who is only consulted and not admitted
LAST_CONSULT_DATE	DATE	Not Null	Date on which patient last consulted with a doctor

g. BILLING Table:

This table holds a special place as it will store the financial or billing details along with room charges, doctor fees, many more attributes.

Column Name	Data Type	Constraints	Description
BILL_ID	NUMBER(10)	Primary Key	Bill ID
BILL_DATE	DATE	Not Null	Date on which patient's bill is sent
PATIENT_ID	NUMBER(12)	Foreign Key (PATIENT – PATIENT_ID), Not Null	Patient's ID
DOCTOR_FEES	NUMBER(6,2)	Not Null	Total amount of fees charged by doctor (USD)
PRESCRIPTION_CHARGE	NUMBER(6,2)	Not Null	Total charges for patient's prescribed medicines
ROOM_CHARGE	NUMBER(6,2)	Not Null	Total charges of the room occupied by patient from admission till discharge
INSURANCE_PERCENT	NUMBER(3)	Not Null	Percentage of total billing charges covered by patient's insurance

h. ROOM Table:

The following table will contain the information about the rooms within the hospital that are dedicated to patients. It will show the availability of the rooms, room number, and its charges

Column Name	Data Type	Constraints	Description
ROOM_ID	NUMBER(3)	Primary Key	Room number where patient is admitted.
ROOM_STATUS	VARCHAR(10)	Not Null Check	Room's status i.e. 'Vacant' or 'Occupied'
ROOM_CHARGE	NUMBER(6,2)	Not Null	Room charges per night

i. APPOINTMENT Table:

Appointment table will contain details like doctor id, patient id, and appointment id and it will be another building block for joining multiple tables.

Column Name	Data Type	Constraints	Description
APPOINTMENT_ID	NUMBER(10)	Primary Key	ID for scheduled appointment
DOCTOR_ID	NUMBER(10)	Foreign Key (DOCTOR – DOCTOR_ID), Not Null	Doctor's ID with whom appointment is booked
PATIENT_ID	NUMBER(12)	Foreign Key (PATIENT – PATIENT_ID), Not Null	Patient's ID for whom appointment is booked
ADMIN_ID	NUMBER(10)	Foreign Key (ADMINISTRATION – ADMIN_ID), Not Null	Administrator's ID who booked the appointment for patient
APPOINTMENT_DATE	DATE	Not Null	Date and time of scheduled appointment

j. DEPARTMENT Table:

In the end, the department table will list the details of all types of departments within that hospital.

Column Name	Data Type	Constraints	Description
DEPT_ID	NUMBER(4)	Primary Key	Medical department's ID
DEPT_NAME	VARCHAR(100)	Not Null	Medical Department Name

CONCLUSION:

With the help of this database management system, the hospital organizations can successfully handle the data of the doctors, staff, patients, billings, and many more entities. With its proper implementation:

1. Our design will specify the total cost of a hospital visit for the patient
2. It will list the doctors unders specific departments
3. The administrative assistant will have the proper data of vacant and occupied rooms
4. Through the name, age, and contact number, the doctors will be able to gather all the medical details of the patient updated in their database
5. Through the database, it will be easy to define the details of the doctors based on their years of experience
6. Patients will be able to book for 3 appointments in a week
7. Through the patient database, the doctors and administrative staff can also determine the most common type of diseases
8. There will be specific days when a doctor can see more than a limited number of patients
9. Patients will be charged a night fee if they are admitted
10. While getting discharged, patients will be issued the bill which will contain all details
11. More functionalities will be dependent on different types of scenarios