JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Low-Country Memorial Hospital Database

Course Section: EN.605.641.81.FA20
Fall, 2020

Prepared by:
Ricca Callis

11/22/2020

Table of Contents

1. INTRODUCTION 3
1.1. Scope and Purpose of Document 4
1.2. Project Objective 4
2. SYSTEM REQUIREMENTS 5
2.1 Hardware Requirements 5
2.2 Software Requirements 5
2.3 Functional Requirements 6
2.4 Database Requirements 6
3. DATABASE DESIGN DESCRIPTION 7
3.1 Design Rationale 7
3.2 E/R Model 9
3.2.1 Entities 9
3.2.1.1 Employees……….9
3.2.2 Relationships 49
3.2.3 E/R Diagram 59
3.3 Relational Model 61
3.3.1 Data Dictionary 61
3.3.2 Integrity Rules 76
3.3.3 Operational Rules 76
3.3.4 Operations 77
3.4 Security 77
3.5 Database Backup and Recovery 77
3.6 Using Database Design or CASE Tool 77
3.7 Other Possible E/R Relationships 77
4. IMPLEMENTATION DESCRIPTION 77
4.1 Data Dictionary 77
4.2 Advanced Features 79
4.3 Queries 79
4.3.2 Print Patient Discharge Information 79
4.3.3 Calculate a Bill 79
5. CRUD MATRIX 82
5.1 List of Entity Types 82
5.2 List of Functions 82
6. CONCLUDING REMARKS 82
APPENDICES 82
REFERENCES 83

1. Introduction

This project created a database for a fictitious hospital, Low-Country Memorial Hospital, modeling its structure based on patient and staff needs at my current place of employment, Beaufort Memorial Hospital (BMH). BMH is a not-for-profit hospital located in Beaufort, South Carolina (Beaufort Memorial Hospital, n.d.). It was founded in 1944 and is licensed for 197 beds (169 acute, 14 rehabilitation, and 14 mental health). BMH is an acute-care hospital, a regional referral center, and the largest medical facility between Savannah, GA, and Charleston, SC. BMH offers medical care and short-term residence for patients with a wide variety of medical needs including: cancer care, critical care, diabetes care, digestive services, emergency services, express care, heart and vascular services, imaging services, infusion and chemotherapy services, laboratory services, LifeFit Wellness Center and community health services, lung and respiratory care, maternity, mental health, neurology, nutrition counseling, occupational health, orthopedics, palliative care, pediatric care, physical therapy and rehabilitation services, primary care, sickle cell clinic, stroke care, urology, women's health, and wound care. Medical care is provided for both inpatient and outpatient services. BMH currently uses Meditech as its Electronic Health Record (EHR) platform for all inpatient services and Athena for all outpatient/primary care services.

As mentioned earlier, the current project is a database system for Low-Country Memorial Hospital, a fictitious place based on BMH. In the present context, a hospital management system was developed for Low-Country Memorial Hospital where a database system was also required. This project focused on the development of the database for the hospital management system. Structure modelling of the database was based on the needs of staff and patients at BMH. This database focused on both inpatient and outpatient hospital records, including all patient services and actions performed by administrative, technician, or medical staff during a patient's stay.

This database project was selected as means to learn the important aspects of a hospital management system. Hospital management systems function to integrate information from multiple disparate sources, which manage all aspects of a hospital's operations, including: medical, financial, administrative, legal, and compliance. It includes EHR, business intelligence, and revenue cycles management. Data through a hospital management system is complex and utilized for many purposes: track/manage patient care (both inpatient and outpatient), track/manage nursing, technician, and admin staff (usually hospital-staffed employees), track/manage physicians/surgeons (usually contracted hospital employees).

All patient activity (visits, admits, procedures, labs, vitals, diagnoses, treatments, imaging/x-rays, medications, and demographic data) must be included in a medical record. Physicians, nurses, technicians, and admin staff must have access to a patient's medical records. Doctors must have quick and easy access to patient records at each visit (and to look at visits over time) so they can better manage patient care, manage compliance, and better communicate and collaborate with the rest of the care-team. Hospital staff use the data for record maintenance, coordination with patients and doctors, and to direct workflow or task management. Hospital Administrators use the data to improve management, performance, and operations of units, staff, physician contracts, pharmacy, lab, imaging, in/out-patient services, maintain inventory/resources/resource allocation, and to minimize waste and improve cost. Specifically, costs, duration of appointments or procedures/length of stay, and total volume will be used by administrative staff to improve procedures and patient care and to determine resource allocation. The same data will be used by the billing department to create insurance claims and patient bills. A proper hospital management system ensures efficient and accurate hospital administration, improved patient monitoring, improved patient care as well as patient-doctor interactions, provides opportunity for improved revenue and better clinical decision-making.

1.1. Scope and Purpose of Document

In general, hospitals record all information about the services patients require (prescriptions and special needs ordered by doctors) and all actions performed by nurses and doctors through a documentation process stored in each patient's EHR. The Low-Country Memorial Hospital database is meant to generalize the recording process across all units. As a result, special equipment required for each unit (like the ICU) is not part of the conceptual design (i.e., intensive care unit special equipment is excluded). Unlike the database systems currently used by BMH, the database system for Low-Country Memorial Hospital includes common medical care services for both inpatient and outpatient recipients (e.g., laboratory services are utilized by both inpatient and outpatient individuals).

The Low-Country Memorial Hospital database focuses on the flow of patient information across both inpatient and outpatient services. Both inpatients and outpatients may require specific prescription medications, will require labs or imaging, will have their vitals checked, and will visit a triage location. As a result, all hospital employees who interact with patients during these shared services must have access to the database system.

As a hospital, the Low-Country Memorial Hospital database system must also track information for patients required to stay overnight or for short-term durations within the hospital. As a result, information relevant to unit, rooms, and nurses or doctors working on site with patients is stored. However, information specific to operating rooms or outpatient surgical centers are excluded from the scope of the database.

The Low-Country Memorial Hospital database system records different types of entities including employees and their types. There are four employee types: Doctor, nurse, technician and administrative. Many other entity types are included and will be explained below (e.g., prescription, medication, patient, room, lab test, medical records, patient bill, etc.). Different types of entities will have specific roles for this database system. By using this database, different hospital operations can be performed, all of which will be aligned with the hospital management systems. Therefore, the scope of this database system will include searching medical records of patients, creating patient records, doctor records and holding specific information about the rooms. Also, billing related records for the patients will be maintained by this database system.

1.2.Project Objective

The main objective of this project is to develop a database system for the fictitious Low-Country Memorial Hospital. This hospital is currently aiming to develop a hospital management system, which will require a relational database management system. This database system will capture all the important information that flows within the hospital management system, including patient information, employee information, previous health records of the patients, total billing for the patient, and much more. By capturing maintaining all important information, this database will play an extremely important role for the hospital management system of Low-Country Memorial Hospital. The Low-Country Memorial Hospital database system aims to create a more efficient system for storing historical records and accessing all the doctor-ordered patient needs at once, including both inpatient and outpatient records.

A clear understanding of the historical patient data and doctor-orders allows the data to be represented as entity sets and relationship sets using the Entity-Relationship (ER) model. The objects that makeup Low-Country Memorial Hospital (entity type definitions), as well as their relationships with one another (relationship type definitions) are used to develop a conceptual database design, a logical design, and the final implementation, which captures all informational aspects of the operations of the hospital employees.

2. System Requirements

In order to develop, implement, access, and utilize the Low-Country Memorial Hospital database system, the following sections will discuss all hardware and software requirements. Generally, users will need: a minimum of 4GB of RAM; 500GB of hard disk drive; Windows 7, 8, 8.1, or 10; Microsoft Office 2016 Professional or Professional Plus, and Microsoft Access 2016.

2.1 Hardware Requirements

For the development of this database system there are some specific hardware requirements. This hardware includes processor, ram and hard disk drive. The minimum requirements for this hardware are provided below.

- **Processor:** 1 Ghz or faster, x86-bit or x64-bit processor with SSE2. The recommended minimum is the Dual Core Intel Pentium Processor from their Xeon series.
- **Ram:** The minimum amount of required RAM is 4GB, but more is recommended.
- **Hard Disk Drive:** The minimum requirement for the hard disk drive is 500GB, but more is recommended but if possible a bigger amount of hard disk drive should be used.
- **Display:** 1024 x 786 resolution
- **Graphics Hardware Acceleration:** DirectX 10 graphics card

2.2 Software Requirements

Some specific software is also required for the development of this database system. Three different types of software packages are required in this case to develop this database system. The first category is the operating system (OS), which hosts or executes the other software applications that are required for developing the database system. The second category is the software application, which is used for running the application. The final category includes other optional uses for the database system, including optional web-based uses.

Although the database can be run using either a Windows or Mac OS, the software application utilized is Microsoft Access 2016 (MS-Access). This application was selected because of its current use within BMH. However, there is no MS-Access designed specifically for the Mac OS. Therefore, users will need Windows 7 or higher either on a Windows OS or through a modification to the Mac OS (e.g., bootcamp or remote desktop). For the operating system Windows 10 will be used. The main software application that will be used for creating this database system is MS Access.

- **Operating Systems:**
  - **Windows:** Windows 7; Windows 8; Windows 10; Windows Vista, Windows Server 2008 R2; Windows Server 2012; .NET Framework 2.0 or higher
  - **Mac:** JAVA runtime Environment (JRE) 1.5 or higher; JAVA servlet container (e.g., Tomcat, JBoss, WebLogic, WebSphere, Jetty)
    - Run Windows Operating System by either:
      - Bootcamp
      - Remote Desktop
- **Software Application:**
  - **MS-Access 2016:**
    - https://www.microsoft.com/en-us/download/details.aspx?id=50040 and
    - https://www.microsoft.com/en-us/microsoft-365/access?legRedir=true&CorrelationId=524351fd-0f5f-454c-be09-847f20e3ce76
  - For difficulty reading accb files, users may utilize:
    - **LibreOffice:** https://www.libreoffice.org
    - **MBD Viewer by Google:** https://mdbviewer.herokuapp.com/
- **Internet Connection:** A persistent internet connection is required, as is an external IP address for the system running the application, and a firewall/proxy access to communicate with external system.

2.3 Functional Requirements

As mentioned above, MS-Access was selected because of its current use within BMH. Functionally, MS-Access is able to import and export data to many formats including: Excel, Outlook, XML, HTML, ASCII, dBase, Paradox, Lotus 1-2-3, FoxPro, MS-SQL Server, Oracle, MySQL, and PostgreSQL. ASP.net web forms can also query a MS-Access database, retrieve records, and display them on the browser. Furthermore, SharePoint Server 2010 allows for MS-Access databases to be published to SharePoint.

Functionally, it is essential that the database be able to import and export files through Excel and Outlook, as they are both heavily utilized within BMH. For the purposes of this project, however, the current database system will provide the functionality of managing and maintaining the patient and hospital employee records, billing records and the medical records of the patients.

2.4 Database Requirements

MS-Access 2016 was used for the development of the Low-Country Memorial Hospital database.

3. Database Design Description

Before designing a physical system to store the data we have to understand how this data will be stored. This is known as Conceptual Database Design. One way to represent the structure of data is using the Entity-Relationship (ER) model. Using this method, we try to represent the structure of the hospital data using two concepts: entity sets, which define real-world “objects” (like people, places, and things); and relationship sets, which define how entities of two or more entity types are related.

Entity types are created to unite entity sets that represent the same object. For example, a “Doctor” entity type is named and defined according to each of the attributes that a doctor contains. Relationship types are also created to define how entity types are related to each other, as well as any other attributes describing how they are related.
In this section, each entity type is defined in detail, including each attribute and its domain. Each relationship is also defined, including: the entity types each relates, the cardinality, and participation. Finally, a diagram that visually represents the conceptual design is presented.

3.1 Design Rationale

The design rationale for any database system includes some important aspects and among them data flow is an important aspect. The specific data flow design has been chosen as it helps the database system to work efficiently. By using this data flow, the present database system will be able to work by utilizing a minimum amount of time. Performance of the database is the main consideration and due to this that specific design has been used for the database.

An outline of the dataflow is presented below:

- Patient has a visit (either scheduled or emergency) at a triage location (emergency dept, surgical center, doctor’s office, lab, imaging)
    - The following will be entered into patient’s medical record:
        - Initial triage info: date, time, location, symptoms (entered by admin; accessed by medical staff and admin)
        - Vital signs (entered by nurse; accessed by medical staff and admin)
        - Orders from doctor: procedures, labs, tests, prescriptions (entered by doctor; accessed by medical staff and admin)
        - Diagnosis, Treatment (entered by doctor; accessed by medical staff and admin)
        - Doctor will determine whether patient needs to be admitted to hospital (entered by doctor; accessed by medical staff and admin)
    - The following will be extracted from DB (by admin, nurse, and doctor)
        - Patient’s prior medical history Symptoms (accessed by medical staff)
- Inpatient Visits:
    - Patients are assigned rooms (by an admin)
    - Patients are monitored by nurses and cared for by doctors

        - Nurses will monitor vital signs, administer medications and special needs
        - Doctors will prescribe medications, perform procedures, order special needs, labs, imaging.
        - Discharge information will be provided to patient
        - Bills track: room fee, length of stay, and fees for services (staff, physician, medication, procedures, labs, or images)
    - Outpatient Visits:
        - Same as above except no fee for hospital stay
- Administrators, Patients, and Medical Staff all interact with the DB for different reasons.
    - Admin:
        - Records data into DB & Extracts data from DB
        - Service List: Entry of Patient (visit, hospital admit), Assign room to a patient, Release of Patient (end of visit, hospital discharge), Update information, Query Information, Generate a Bill, Generate other Reports (finance; population health; operations/costs/waste for hospital departments)
    - Patients:
        - Inpatient:
            - Does not record data into DB; Can extract data from DB; Receives extracted data from DB
            - Service List: Request for medical information, View/Pay Bill
            - Receives: Interaction with admin at initial triage, Room for hospital stay, care from nurses, treatment from doctors (prescriptions, procedures, labs, imaging), discharge information, bill for services
        - Outpatient:
            - Does not record data into DB; Can extract data from DB; Receives extracted data from DB
            - Service List: Request for medical information, View/Pay Bill
            - Receives: Interaction with admin at initial triage, Care from nurses, treatment from doctors (prescriptions, procedures, labs, imaging), discharge information, bill for services
    - Doctor:
        - Enters and extracts data in DB
        - Service List: Request for medical information; Provide treatment/diagnosis, Prescribe medication, Order imaging, Order lab, Order Special Need, Perform Procedure
    - Nurse
        - Enters and extracts data in DB
        - Service List: Request for medical information; Administer special need, administer medication, conduct vital assessments

3.2 E/R Model

Entities describe real world-objects, and are defined by their name and the attributes they contain. The Low-Country Memorial Hospital E/R Model includes important entities such as doctors, nurses, and patients. It also includes the orders given by doctors (prescriptions or special needs) and individual assessments of patients.

Relationships associate two or more entities of the same type. They are defined by which entities they relate, as well as additional attributes describing how entities are related. They also specify constraints that control how many entities are related to each other. Individual entities from different types are related to each other by participating in relationship instances.

3.2.1 Entities

The Low-Country Memorial Hospital Database system uses the following entities: Employee, Doctor, Nurse, Technician, Admin, Prescription, Medication, Room, Vitals, Unit, Special Need, Patient, Medical Record, Procedure, Imaging, Lab Test, Triage, and Patient Bill. The following section will list each entity type and its name, the attributes it defines, the domain constraints of each attribute, fields to be indexed for fast access, and keys, or fields that uniquely identify individual entities in the set.

3.2.1.1 Employee

Entity Name: Employee

Description: An employee is paid by Low-Country Memorial Hospital to work on-site during shifts. Each employee has basic identifying and contract information. Since the scope of the database includes all interactions with patients, employees can be: doctors, nurses, technicians, or administrative (a total, disjoint specialization). Employees will be infrequently inserted and updated by the Human Resources department.

Candidate Keys: EmployeeID, SSN
Primary Key: EmployeeID
Strong/Weak Entity: Strong
Fields to be Indexed: Employee ID, FName, LName, SSN

List of Attributes: EmployeeID, SSN, FName, MInitial, LName, Address, PhoneNumber, EmployeeType

Attribute Descriptions:

Attribute Name	EmployeeID	SSN	FName
Description	Number assigned by Human Resources to each employee; Uniquely identifies each employee; Can be implemented as an auto-increment integer	Social Security Number; Can be used to uniquely identify employees	First Name of employee
Domain/Type	Integer	Integer	String
Value/Range	0-MaxID	000000000-999999999 (any 9-digit integer)	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	Yes	No
Single or Multi-Value	Single	Single	Multi-Value
Simple or Composite	Simple	Simple	Simple

Employee Attributes Continued:

Attribute Name	MInitial	LName	Address

Description	Middle initial of employee	Last name of employee	Street Address, City, State, Zip Code
Domain/Type	String	String	String, String, String, Integer
Value/Range	Any	Any	Any, Any, Any, 00000-99999
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Multi-Value	Multi-Value	Single
Simple or Composite	Simple	Simple	Composite

Employee Attributes Continued:

Attribute Name	PhoneNumber	EmployeeType
Description	Primary contact number	Whether employee is medical (doctor, nurse), technical (e.g., imaging, lab services, etc), or administrative (triage)
Domain/Type	Integer	String
Value/Range	0000000000-9999999999	Any
Default Value	None	None
Null Value Allowed	No	No
Unique	No	No
Single or Multi-Value	Single	Multi-Value
Simple or Composite	Simple	Simple

3.2.1.2 Doctor

Entity Name: Doctor

Description: A doctor is a specialization of an employee. Doctors decide which actions Nurses need to perform on each patient (including when, for how long, and how often) by issuing two general types of orders: Prescriptions (for medications) and Special Needs (for required activities, like assisted showers, bed turning for bed-ridden patients, or wound cleaning for patients with wounds). Doctors will also order the following for patients: laboratory tests (e.g., blood tests, urine analyses, etc), imaging (e.g., MRI, ultrasound, PET, CAT, etc). Finally, doctors perform procedures on patients (e.g., surgeries).

Candidate Keys: LicenseNumber
Primary Keys: LicenseNumber
Strong/Weak Entity: Strong
Fields to be Indexed: LicenseNumber

List of Attributes: LicenseNumber, Specialty, Salary, Fee

Attribute Descriptions:

Attribute Name	LicenseNumber	Specialty	Salary
Description	All South Carolina doctors are given a Medical Doctor license number that uniquely identifies them from all other state doctors. The number has two parts: the letters “MD” and a unique 6-digit number.	The doctor’s specialty field (e.g., neurology, urology, orthopedics, etc)	The doctor’s annual salary
Domain/Type	String, Integer	String	Integer
Value/Range	String “MD” followed by 6-digit integer	Any	0000000-9999999 (up to 7 digit integer)
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Composite	Simple	Simple

Doctor Attributes Continued:

Attribute Name	Fee
Description	The doctor's fee for a specific procedure or duty; This attribute is tallied and used on the patient's bill
Domain/Type	Integer
Value/Range	000000-999999 (up to 6 digit integer)
Default Value	0
Null Value Allowed	No
Unique	No
Single or Multi-Value	Single
Simple or Composite	Simple

3.2.1.3 Nurse

Entity Name: Nurse

Description: A nurse is a specialization of an employee. They perform the actions ordered by doctors and assess how patients respond. During a shift, a nurse works in a specific unit and is assigned to specific rooms, in order to take of the patient who inhabits it. The nurse conducts vital assessments (every four hours) and carries out the activities and administers the medications for their assigned patient, as ordered by the patient's doctor.

Candidate Keys: LicenseNumber
Primary Keys: LicenseNumber
Strong/Weak Entity: Strong
Fields to be Indexed: LicenseNumber

List of Attributes: LicenseNumber, Salary, Fee

Attribute Descriptions:

Attribute Name	LicenseNumber	Salary	Fee
Description	All South Carolina nurses are given a Registered Nurse license number that uniquely identifies them from all other nurses in the state. The number has two parts: the letters "RN" and a unique 6-digit number.	The nurse's annual salary	The nurse's fee for a specific procedure or duty. This information is tallied and used on the patient's bill.
Domain/Type	String, Integer	Integer	Integer
Value/Range	The string "RN" followed by a 6-digit integer	0000000-9999999 (up to 7 digit integer)	000000-999999 (up to 6 digit integer)
Default Value	None	None	0
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Composite	Simple	Simple

3.2.1.4 Technician

Entity Name: Technician

Description: A technician is a specialization of an employee. A technician may work in the laboratory, concluding lipid or urine analysis or may work in the imaging department, conducting MRIs, ultrasounds, etc. They perform the tests or images ordered by doctors.

Candidate Keys: LicenseNumber
Primary Keys: Specialty
Strong/Weak Entity: Strong
Fields to be Indexed: LicenseNumber

List of Attributes: LicenseNumber, Specialty, Salary, Fee

Attribute Descriptions:

Attribute Name	LicenseNumber	Specialty	Salary
Description	A unique 6 to 10 digit number.	The technicians's specialty field (e.g., phlebotomy, MRI, ultrasound, etc)	The technicians's annual salary
Domain/Type	Integer	String	Integer
Value/Range	6 to 10 digit integer	Any	0000000-9999999 (up to 7 digit integer)
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Composite	Simple	Simple

Technician Attributes Continued:

Attribute Name	Fee
Description	The technicians' fee for a specific procedure or duty; This attribute is tallied and used on the patient's bill
Domain/Type	Integer
Value/Range	000000-999999 (up to 6 digit integer)
Default Value	0
Null Value Allowed	No
Unique	No
Single or Multi-Value	Single
Simple or Composite	Simple

3.2.1.5 Admin

Entity Name: Admin

Description: An admin is a specialization of an employee. An admin is an administrative employee who works in triage. This individual will handle all initial patient intake at whatever location the patient is first seen. Admin employees may work in one or many locations (e.g., emergency, primary care, surgery, imaging, laboratory).

Candidate Keys: AdminID
Primary Keys: AdminID
Strong/Weak Entity: Strong
Fields to be Indexed: AdminID

List of Attributes: AdminID, Location

Attribute Descriptions:

Attribute Name	AdminID	Location
Description	Number assigned by Human Resources to each employee; Uniquely identifies each admin employee; Can be implemented as an auto-increment integer	Location of worksite within Low-Country Memorial Hospital
Domain/Type	Integer	String
Value/Range	0-MaxID	Any
Default Value	MaxID+1	None
Null Value Allowed	No	No
Unique	Yes	No
Single or Multi-Value	Single	Multi-Valued
Simple or Composite	Simple	Composite

3.2.1.6 Prescription

Entity Name: Prescription

Description: A prescription contains a medication ordered by a doctor for a single patient, including administration instructions. Prescriptions are related to medications ("has medication" relationship). Doctors prescribe medication to patients based on the patient's needs. For inpatients, nurses ensure patients receive the medication prescribed by the doctors. For inpatients, nurses administer each medication numerous times throughout the patient's hospital stay, based on the "Frequency" and "Start/End Date" fields. Similarly, outpatient patients/individuals self-administer each medication multiple times based on the "Frequency" and "Start/End Date" fields. Prescriptions are inserted as the doctor orders them and are never updated or deleted (only replaced by new prescriptions).

Candidate Keys: PrecriptionID
Primary Keys: PrescriptionID
Strong/Weak Entity: Strong
Fields to be Indexed: PrescriptionID

List of Attributes: PrescriptionID, Dosage, Frequency, StartDate, EndDate, Fee

Attribute Descriptions:

Attribute Name	PrescriptionID	Dosage	Frequency
Description	An auto-increment unique identifier for prescriptions.	Describes the amount that should be administered in a dose.	How often the dose should be administered.
Domain/Type	Integer	String	String
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Prescription Attributes Continued:

Attribute Name	StartDate	EndDate	Fee
Description	The date the medication should first be administered.	The date when the medication should no longer be administered.	The cost of the medication; This attribute is tallied and used on the patient's bill.
Domain/Type	Date	Date	Integer
Value/Range	Any	Any	000000-999999 (up to 6 digit integer)
Default Value	None	None	0
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

3.2.1.7 Medication

Entity Name: Medication

Description: A single medication is associated with each prescription when a doctor prescribes it. The same medication can appear in multiple prescriptions. Each medication has a descriptive name that indicates its contents and a summary of its purpose (e.g., "painkiller", "anti-inflammatory", "antibiotic").

Candidate Keys: MedicationID, MedicationName
Primary Keys: MedicationID
Strong/Weak Entity: Strong
Fields to be Indexed: MedicationID, MedicationName, MedicationPurpose

List of Attributes: MedicationID, MedicationName, MedicationPurpose, Dosage, Frequency, Fee

Attribute Descriptions:

Attribute Name	**MedicationID**	**MedicationName**	**MedicationPurpose**
Description	An auto-increment unique identifier for medication.	The standard name of the medication; Describes the chemical contents of the medication.	Brief description of the function of the medication.
Domain/Type	Integer	String	String
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Medication Attributes Continued:

Attribute Name	Frequency	Fee
Description	How often the dose should be taken.	The cost of the medication; This attribute is tallied and used on the patient's bill.
Domain/Type	String	Integer
Value/Range	Any	000000-999999 (up to 6 digit integer)
Default Value	None	0
Null Value Allowed	No	No
Unique	No	No
Single or Multi-Value	Single	Single
Simple or Composite	Simple	Simple

3.2.1.8 Room

Entity Name: Room

Description: A room houses a patient in the hospital (i.e., houses an inpatient individual). Each room is designed to provide all necessary services to a patient in a unit. Thus, the patient's room allows the patient to be located. Each room has a unique number and a location in a unit. Nurses repeatedly visit a patient's room to administer medication and/or fulfill a special need. Rooms generally do not have to be updated or deleted.

Candidate Keys: RoomID, RoomNumber
Primary Keys: RoomID
Strong/Weak Entity: Strong
Fields to be Indexed: RoomID, RoomNumber

List of Attributes: RoomID, RoomNumber, RoomType, RoomStatus, Fee

Attribute Descriptions:

Attribute Name	RoomID	RoomNumber	RoomType
Description	Auto-increment integer that uniquely identifies a room.	Uniquely identifies a room. Usually contains information about the unit and floor to which it belongs.	Indicates whether the room is a single or shared and whether it is sterile, intensive care, or standard care room.
Domain/Type	Integer	Integer	String, String
Value/Range	0-MaxID	0-999	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	Yes	No
Single or Multi-Value	Single	Single	Multi-Value
Simple or Composite	Simple	Simple	Composite

Room Attributes Continued:

Attribute Name	RoomStatus	Fee
Description	Describes whether the room is occupied or vacant, clean or requires cleaning	The cost of the medication; This attribute is tallied and used on the patient's bill.
Domain/Type	String, String	Integer
Value/Range	Any	000000-999999 (up to 6 digit integer)
Default Value	Any	0
Null Value Allowed	No	No
Unique	No	No
Single or Multi-Value	Multi-Value	Single
Simple or Composite	Composite	Simple

3.2.1.9 Unit

Entity Name: Unit

Description: A unit has a specific location in a hospital and contains a set of rooms, each of which offer the same level of service to the inpatients. Units may include: Intensive Care Unit (ICU), Medical/Surgical Unit, or Non-Intensive Care Units. ICUs may include: Neonatal intensive care units (NICUs); Pediatric Intensive Care units (PICUs); Coronary Care and Cardiothoracic Units (CCUs/CTUs); Surgical Intensive Care Units (SICUs); Medical Intensive Care Units (MICUs); or Long Term Intensive Care Units (LTAC ICUs). Medical/Surgical Units may include: medical units which provide care for conditions like stroke, heart attack, or pneumonia; oncology units which provide care for patients with cancer and immune system disorders; surgical units which provide care for pre and post-surgical patients and/or which may specialize in certain types of surgery (like orthopedic joint surgery). And non-intensive care units may include: Neonatal units which provide care for ill premature infants and neonates; woman and infant health units which provide care before, during, and after childbirth; pediatric units which provide care for children younger than 19 years old; post-critical care (or step down units) which care for patients no longer needing ICU level care; rehabilitation wards which provide care for conditions like stroke, heart attack, or pneumonia; or long-term care wards which provide care to patients for an extended period of time. In most hospitals, the condition of the patient determines the unit to which they are assigned. For the current project, the framework describes services offered to patients belonging to all units. Units have a fixed location in the hospital (e.g., floor or area of a floor). In this way, units can be used to locate rooms as well. New units will rarely need to be inserted.

Candidate Keys: UnitID, UnitName, UnitAbbreviation
Primary Keys: UnitID
Strong/Weak Entity: Strong
Fields to be Indexed: UnitID, UnitName, UnitAbbreviation

List of Attributes: UnitID, UnitName, UnitAbbreviation

Attribute Descriptions:

Attribute Name	UnitID	UnitName	UnitAbbreviation
Description	An auto-increment integer that uniquely identifies a unit.	Identifies (i.e., names) and describes the purpose of the unit (e.g., Intensive Care).	Short name for the unit (e.g., ICU, Med/Surg, NICU).
Domain/Type	Integer	String	String
Value/Range	0-MaxID	Any	String length 10
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	Yes	Yes
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

3.2.1.10 Vitals

Vitals: Assessment ID, Blood pressure, Respiration rate, Heart rate
An assessment is taken by a nurse for a single patient, and describes basic information about their
current condition.

Entity Name: Vitals

Description: Vitals are assessments taken by a nurse for a single patient and describe basic information about their current medical/physical condition. A nurse conducts a basic assessment of a patient's health with some order of frequency (usually every 4 hours). The information included in the vitals assessment is generally the same for all units and includes the following information: blood pressure, respiration rate, heart rate, oxygen saturation, blood glucose levels, high-density lipoprotein (HDL), low-density lipoprotein (LDL), triglycerides, weight, height, date, and time. Note that total cholesterol was excluded as it can be derived from HDL and LDL values. Vital assessments are never updated, but are inserted frequently throughout the day, as they are performed.

Candidate Keys: AssessmentID
Primary Keys: AssessmentID
Strong/Weak Entity: Strong
Fields to be Indexed: AssessmentID

List of Attributes: AssessmentID, BloodPressure, RespirationRate, HeartRate, OxygenSaturation, BloodGlucose, HDL, LDL, Triglycerides, Weight, Height, Date, Time, Fee

Attribute Descriptions:

Attribute Name	AssessmentID	BloodPressure	RespirationRate
Description	An auto-increment integer to uniquely identify vistal assessments.	Systolic and diastolic blood pressure readings in millimeters of mercury.	Rate at which the patient is breathing.
Domain/Type	Integer	Integer, Integer	Integer
Value/Range	0-MaxID	0-300, 0-300	0-300
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Composite	Simple

Vitals Attribute Continued:

Attribute Name	HeartRate	OxygenSaturation	BloodGlucose
Description	The patient's current heart rate in beats per minute.	Measurement taken from a pulse oximeter (pulse ox), which estimates the amount of oxygen in the patient's blood by sending infrared light into capillaries of the finger, toe, or earlobe and then measures how much light is reflected off the gases. Output is represented as a percentage.	Fasting blood sugar test; Following an overnight fast, the patient's blood sample is taken. Results are given in milligrams per deciliter (mg/dL). Normal ranges are less than 100mg/dL, prediabetes ranges between 100-125mg/dL, and diabetes is indicated at 126mg/dL or higher.
Domain/Type	Integer	Integer	Integer
Value/Range	0-300	0-100	0-999
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Vitals Attribute Continued:

Attribute Name	HDL	LDL	Triglycerides
Description	"Good" cholesterol; Results are given in milligrams per deciliter (mg/dL). Good values are 40 mg/dL or higher; low values are less than 40mg/dL	"Bad" cholesterol; Results are given in milligrams per deciliter (mg/dL). Good values are less than 100mg/dL; Borderline values are 130-159mg/dL; High values are 160 mg/dL or higher	Results are given in milligrams per deciliter (mg/dL). Good values are less than 150mg/dL; Borderline values are 150-199mg/dL; High values are 200 mg/dL or higher.
Domain/Type	Integer	Integer	Integer
Value/Range	0-999	0-999	0-999
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Vitals Attribute Continued:

Attribute Name	Weight	Height	Date
Description	Describes the weight of the patient in lbs	Describes the height of the patient in cms	Describes the date the vital assessment was performed.
Domain/Type	Integer	Integer	Date
Value/Range	0-999	0-999	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Vitals Attribute Continued:

Attribute Name	Time	Fee
Description	Describes the time at which the vital assessment was performed.	The cost of the assessment; This attribute is tallied and used on the patient's bill.
Domain/Type	Time	Integer
Value/Range	Any	000000-999999 (up to 6 digit integer)
Default Value	None	0
Null Value Allowed	No	No
Unique	No	No
Single or Multi-Value	Single	Single
Simple or Composite	Simple	Simple

3.2.1.11 Special Need

Entity Name: Special Need

Description: A special need is an activity for which a single patient requires assistance from a nurse. Special needs include the name and description of the activity (e.g., bed turn, wound cleaning, assisted shower, walking). A doctor orders a special need for a patient and then the nurse performs the special need several times a day over a specified time period. Special needs are inserted but never updated or deleted.

Candidate Keys: SpecialNeedID
Primary Keys: SpecialNeedID
Strong/Weak Entity: Strong
Fields to be Indexed: SpecialNeedID

List of Attributes: SpecialNeedID, Frequency, StartDate, EndDate

Attribute Descriptions:

Attribute Name	SpecialNeedID	Frequency	StartDate	EndDate
Description	An auto-increment unique identifier for special needs	Describes how often a patient requires an activity (e.g., “once a day”, “once a week”, “every hour”)	The date the special need should first be performed.	The date when the special need should no longer be performed.
Domain/Type	Integer	String	Date	Date
Value/Range	0-MaxID	Any	Any	Any
Default Value	MaxID+1	Null	None	None
Null Value Allowed	No	Yes	No	No
Unique	Yes	No	No	No
Single or Multi-Value	Single	Single	Single	Single
Simple or Composite	Simple	Simple	Simple	Simple

3.2.1.12 Patient

Entity Name: Patient

Description: A patient is either outpatient or is admitted to the hospital (i.e., patient type is either outpatient or inpatient). Patients which are admitted to the hospital as sent to a unit depending on the severity of their medical condition. Each patient contains basic identifying and contact information, as well as language spoken in case a translator is needed. Additionally, each patient contains relevant health history information including: allergies, chronic diseases, symptoms, and diagnoses. A doctor will prescribe medication for out- and inpatients. Additionally, doctors will order special needs for inpatients. For all inpatients, nurses will conduct ordered special needs, administer prescribed medication, and conduct vital assessments on each patient.

Candidate Keys: PatientID, SSN
Primary Keys: PatintID
Strong/Weak Entity: Strong
Fields to be Indexed: PatientID, FName, LName, SSN, PhoneNumber, InsuranceNumber

List of Attributes: PatientID, SSN, FName, MInitial, LName, Address, PhoneNumber, BirthDate, Gender, Age, MaritalStatus, InsuranceNumber, InsuranceProvider, AdmitDate, AdmitTime, DischargeDate, DischargeTime, DischargeInstructions, LanguageSpoken, Allergies, Weight, Height, EmergencyContact, Symptom, TimeOfRegistration, Diagnosis, ChronicDiseases, PatientType

Attribute Descriptions:

Attribute Name	PatientID	SSN	FName
Description	Number assigned by the hospital to each patient that uniquely identifies them; Can be implemented as an auto-increment integer	Social security number; can be used to uniquely identify patients	First name of patient
Domain/Type	Integer	Integer	String
Value/Range	0-MaxID	000000000-999999999 (Any 9-digit integer)	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	Yes	No
Single or Multi-Value	Single	Single	Single

Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	MInitial	LName	Address
Description	Middle initial of middle name of the patient	Last name of the patient	Street address, city, State, Zip code
Domain/Type	String	String	String, String, String, Integer
Value/Range	Any	Any	Any, Any, Any, 00000-99999
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	PhoneNumber	BirthDate	Gender
Description	Indicates primary method of contact (home or cell) for the patient; represented as an integer.	Indicates date of birth of patient	Indicates sex of patient; single letter that identifies the gender (e.g., M for Male; F for Female).
Domain/Type	Integer	Date	Character
Value/Range	0000000000-9999999999	Any	F/M
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	Age	MaritalStatus	InsuranceNumber
Description	Describes the current age of the patient in years	Describes the marital status of the patient (married, divorced, separated, single)	Describes the patient's insurance policy number
Domain/Type	Integer	String	String
Value/Range	0-999	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	InsuranceProvider	AdmitDate	AdmitTime
Description	Describes the name insurance plan (e.g., Medicare)	Describes the date a patient is admitted to the hospital	Describes the time at which a patient is admitted to the hospital
Domain/Type	String	Date	Time
Value/Range	Any	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	DischargeDate	DischargeTime	DischargeInstructions
Description	Describes the date a patient is discharged from the hospital	Describes the time at which a patient is discharged from the hospital	Describes instructions for the patient to follow once discharged from the hospital
Domain/Type	Date	Time	String
Value/Range	Any	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Multi-Value
Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	LanguageSpoken	Allergies	Weight
Description	Describes the languages spoken by the patient	Describes the patient’s allergies (including medicine, food, seasonal)	Describes the patient’s weight in lbs
Domain/Type	String	String	Integer
Value/Range	Any	Any	0-999
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Multi-Value	Multi-Value	Single
Simple or Composite	Simple	Simple	Simple

Patient Attributes Continued:

Attribute Name	Height	EmergencyContact	Symptom
Description	Describes the height of the patient in cm	Describes the First Name, Last Name, and Phone Number of the patient’s emergency contact	Describes the symptom(s) experienced by the patient at the point of contact.
Domain/Type	Integer	String, String, Integer (10-digit)	String
Value/Range	0-999	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Multi-Value	Multi-Value
Simple or Composite	Simple	Composite	Composite

Patient Attributes Continued:

Attribute Name	TimeOfRegistration	Diagnosis	ChronicDiseases
Description	Indicates the time at which the patient is registered at the triage location	Indicates the diagnosis of the patient’s current condition	Indicates the patient’s current chronic diseases
Domain/Type	Time	String	String
Value/Range	Any	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Multi-valued	Multi-valued
Simple or Composite	Simple	Composite	Composite

Patient Attributes Continued:

Attribute Name	PatientType
Description	Indicates whether the patient is currently inpatient or outpatient
Domain/Type	String
Value/Range	Inpatient/Outpatient
Default Value	None
Null Value Allowed	No
Unique	No
Single or Multi-Value	Single
Simple or Composite	Simple

3.2.1.13 Triage

Entity Name: Triage

Description: Triage is the place at which the hospital encounters a patient. Triage is much like a reception area. At a triage, a hospital admin employee will greet a patient, obtain specific information regarding the patient's purpose/reason for hospital visit, and enter all relevant information into the database. Each triage has a location and an encounter type (Routine Visit, Hospital Admission, Virtual Visit, Episode of Care, Emergency NonAdmit).Triages are location-based areas within the hospital, and therefore will rarely need to be inserted.

Candidate Keys: TriageID
Primary Keys: TriageID
Strong/Weak Entity: Strong
Fields to be Indexed: TriageID

List of Attributes: TriageID, EncounterType, TriageLocation, Evaluation, Comment, Fee, StartDate, StartTime, EndDate, EndTime, Symptom, Diagnosis

Attribute Descriptions:

Attribute Name	TriageID	EncounterType	TriageLocation
Description	Number assigned by the hospital to each triage location that uniquely identifies it; Can be implemented as an auto-increment integer	Encounters are either: Routine Visit, Hospital Admission, Virtual Visit, Episode of Care, Emergency NonAdmit	Describes the location within the hospital. Locations may include: primary care, imaging, laboratory, emergency department, pre-op surgery
Domain/Type	Integer	String	String
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Single	Single

Triage Attribute Continued:

Attribute Name	Evaluation	Comment	Fee
Description	Describes whether the patient has been evaluated. Y indicates the patient has been evaluated. N indicates the patient has not been evaluated	The evaluation comments.	The cost of the triage; This attribute is tallied and used on the patient's bill.
Domain/Type	String	String	Integer
Value/Range	Y/N	Any	0-999999 (up to 6 digit integer)
Default Value	N	None	0
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Triage Attribute Continued:

Attribute Name	StartDate	StartTime	EndDate
Description	Indicates the date at which the patient is seen at triage	Indicates the time at which the patient is seen at triage	Indicates the date at which the patient leaves triage
Domain/Type	Date	Time	Date
Value/Range	Any	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Triage Attribute Continued:

Attribute Name	EndTime	Symptom	Diagnosis
Description	Indicates the time at which the patient leaves triage	Indicates the symptoms with which the patient currently presents at triage	Indicates the diagnosis, if any, given to the patient at triage
Domain/Type	Time	String	String
Value/Range	Any	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Multi-value	Multi-value
Simple or Composite	Simple	Simple	Simple

3.2.1.14 Imaging

Entity Name: Imaging

Description: A doctor orders one or more radiological imaging assessment for a patient. These may include, but are not limited to: MRI, CAT scan, X-Ray, PET scan, Ultrasound, IVP, etc. A hospital technician performs an imaging assessment on a patient, based on the doctor's specified orders. Each imaging assessment is given a radiology report, which describes the observed outcome of the assessment. Images are first conducted by technicians and then updated once the radiologist completes their report. Additionally, images may be inserted frequently.

Candidate Keys: ImageID, ImageName
Primary Keys: ImageID
Strong/Weak Entity: Strong
Fields to be Indexed: ImageID

List of Attributes: ImagingID, ImageName, ImageDate, ImageTime, RadiologyReport, Fee

Attribute Descriptions:

Attribute Name	ImagingID	ImageName	ImageDate
Description	Number assigned by the hospital to each imaging service that uniquely identifies it; Can be implemented as an auto-increment integer	Name and cite location of the image (e.g., X-ray, pelvis)	Date image was taken
Domain/Type	Integer	String, String	Date
Value/Range	0-MaxID	Any, Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Multi-Value	Single
Simple or Composite	Simple	Composite	Simple

Imaging Attribute Continued:

Attribute Name	ImageTime	RadiologyReport	Fee
Description	The time at which the image was taken	The radiologist's report or analysis of the outcome of the image	Indicates the cost of the image to the patient
Domain/Type	Time	String	Integer
Value/Range	Any	Any	0-999999
Default Value	None	None	0
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

3.2.1.15 LabTest

Entity Name: LabTest

Description: A Doctor orders one or more Lab Test for a patient, which may include but is not limited to: lipid analysis, urine analysis, or fecal analysis. A hospital technician conducts the laboratory test on/for the patient, as per the doctor's orders. Lab tests are may be updated by a physician or technician to enter test results or comments.

Candidate Keys: LabID, TestName
Primary Keys: LabID
Strong/Weak Entity: Strong
Fields to be Indexed: LabID, TestName

List of Attributes: LabID, TestName, OrderDate, OrderTime, ReturnDate, ReturnTime, TestResult, Comment, Fee

Attribute Descriptions:

Attribute Name	LabID	TestName	OrderDate
Description	Number assigned by the hospital to each laboratory test that uniquely identifies it; Can be implemented as an auto-increment integer	The name of the laboratory test	The date on which the doctor ordered the laboratory test
Domain/Type	Integer	String	Date
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	Any	Any
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

LabTest Attribute Continued:

Attribute Name	OrderTime	ReturnedDate	ReturnedTime
Description	Number assigned by the hospital to each laboratory test that uniquely identifies it; Can be implemented as an auto-increment integer	The date the test is completed	The time at which the test is completed
Domain/Type	Integer	Date	Time
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

LabTest Attribute Continued:

Attribute Name	TestResult	Comment	Free
Description	The result of the laboratory test	The lab test comments.	The cost of the triage; This attribute is tallied and used on the patient's bill.
Domain/Type	String	String	Integer
Value/Range	Any	Any	0-999999 (up to 6 digit integer)
Default Value	None	None	0
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

3.2.1.16 MedicalRecord

Entity Name: MedicalRecord

Description: Each patient seen at the hospital has a medical record. Each time the patient is seen, the medical record is updated.

Candidate Keys: RecordID
Primary Keys: RecordID
Strong/Weak Entity: Strong
Fields to be Indexed: RecordID

List of Attributes: RecordID, Symptom, Diagnosis, DateOfExamination

Attribute Descriptions:

Attribute Name	RecordID	Symptom	Diagnosis
Description	Number assigned by the hospital to each medical record that uniquely identifies it; Can be implemented as an auto-increment integer	The symptom with which the patient presents	The diagnosis given to the patient, based on their symptoms
Domain/Type	Integer	String	String
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Multi-value	Multi-value
Simple or Composite	Simple	Simple	Simple

3.2.1.17 PatientBill

Entity Name: PatientBill

Description: Patients accrue charges for all services provided by the hospital, including medical staff fees, imaging services fees, laboratory fees, and room fees. The patient bill collects all relevant charges and is dispersed to the patient at the completion of all rendered services.

Candidate Keys: BillID
Primary Keys: BillID
Strong/Weak Entity: Strong
Fields to be Indexed: BillID

List of Attributes: BillID, DoctorCharge, MedicineCharge, ProcedureCharge, RoomCharge, NumberOfDays, NursingCharge, ImagingCharge, LabCharge, TriageCharge, TotalBill

Attribute Descriptions:

Attribute Name	BillID	DoctorCharge	MedicineCharge
Description	Number assigned by the hospital to each patient bill that uniquely identifies it; Can be implemented as an auto-increment integer	Indicates the fee for services provided by a doctor	Indicates the fee for a medication
Domain/Type	Integer	Integer	Integer
Value/Range	0-MaxID	0-999999	0-999999
Default Value	MaxID+1	0	0
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

PatientBill Attribute Continued:

Attribute Name	ProcedureCharge	RoomCharge	NumberOfDays
Description	Cost of medical procedure performed by a doctor	Cost of room for inpatient/hospital admit	Number of days the patient stayed in a specific hospital room
Domain/Type	Integer	Integer	Integer
Value/Range	0-999999	0-999999	0-999999
Default Value	0	0	0
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

PatientBill Attribute Continued:

Attribute Name	NursingCharge	ImagingCharge	LabCharge
Description	Cost of nursing provided to inpatient	Cost of imaging service provided to patient	Cost of laboratory service provided to patient
Domain/Type	Integer	Integer	Integer
Value/Range	0-999999	0-999999	0-999999
Default Value	0	0	0
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

PatientBill Attribute Continued:

Attribute Name	TriageCharge	TotalBill
Description	Cost of triage	Total accrued charges
Domain/Type	Integer	Integer
Value/Range	0-999999	0-999999
Default Value	0	0
Null Value Allowed	No	No
Unique	No	No
Single or Multi-Value	Single	Single
Simple or Composite	Simple	Simple

3.2.1.18 Procedure

Entity Name: Procedure

Description: A doctor performs one or more medical procedures on a patient. Procedures may be conducted on inpatients or outpatients. Each procedure is charged to the patient's bill. Procedures are inserted, updated (end dates/times), but never deleted.

Candidate Keys: ProcedureID, ProcedureName
Primary Keys: ProcedureID
Strong/Weak Entity: Strong
Fields to be Indexed: ProcedureID, ProcedureName

List of Attributes: ProcedureID, ProcedureName, StartDate, StartTime, EndDate, EndTime, Description, Fee

Attribute Descriptions:

Attribute Name	ProcedureID	ProcedureName	StartDate
Description	Number assigned by the hospital to each procedure that uniquely identifies it; Can be implemented as an auto-increment integer	Name of procedure conducted	Date the procedure began
Domain/Type	Integer	String	Date
Value/Range	0-MaxID	Any	Any
Default Value	MaxID+1	None	None
Null Value Allowed	No	No	No
Unique	Yes	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Procedure Attribute Continued:

Attribute Name	StartTime	EndDate	EndTime
Description	Time at which the procedure began	Date at which the procedure ended	Time at which the procedure ended
Domain/Type	Time	Date	Time
Value/Range	Any	Any	Any
Default Value	None	None	None
Null Value Allowed	No	No	No
Unique	No	No	No
Single or Multi-Value	Single	Single	Single
Simple or Composite	Simple	Simple	Simple

Procedure Attribute Continued:

Attribute Name	Description	Fee
Description	Description and/or comments about procedure	Cost of procedure
Domain/Type	String	Integer
Value/Range	Any	0-999999
Default Value	None	0
Null Value Allowed	No	No
Unique	No	No
Single or Multi-Value	Single	Single
Simple or Composite	Simple	Simple

3.2.2 Relationships

Relationships associate two or more entities of the same type. Relationships are defined by which entities they relate, as well as the degree of their relationship and the constraints that control how many entities are related to one another. The following section will define each relationship type: the entity types it relates, the constraints on cardinality and participation, and any additional attributes describing how entities in participating in the relationship set are related.

Briefly, the Employee entity is connected with the entity Doctor, Nurse, Technician, and Admin. They are the subsets of the Employee entity. Doctor prescribes Prescription, Prescription prescribed for Patient, Prescription has Medication, Nurse administers Prescription, Nurse takes vitals, Patient undergoes Assessment, Doctor orders SpecialNeed, Nurse performs SpecialNeed, Patient receives SpecialNeed, Nurse assigned to Patient, Nurse works in Unit, Patient placed Room, Technician takes Lab Test and

Imaging, and Patient has MedicalRecord, Patient has a PatientBill, Admin conducts aTriage, and Doctor performs a Procedure

3.2.2.1 Prescribes

Relationship: Prescribes

Description: Doctors prescribe medications. These prescriptions must be ordered by a single, licensed doctor. Doctors may order multiple prescriptions for multiple patients. At the time of prescription, doctors will also include instructions for the administration of the medication, including the dose, frequency, and duration of it’s use.

Entity Sets Involved: Doctor, Prescription

Mapping Cardinality: 1 N

Participation Constraint: Partial participation for doctor; Partial participation for prescription. All prescriptions must be ordered by a single doctor. Doctors may or may not prescribe medications.

3.2.2.2 Prescribed For

Relationship: Prescribed For

Description: Prescriptions are prescribed by doctors for patients. Each patient can receive none or many prescriptions, because medical conditions may change over time. A prescription is ordered for one patient only, as each prescription has patient-specific dosages, frequencies, and durations.

Entity Sets Involved: Prescription, Patient

Mapping Cardinality: N 1

Participation Constraint: Total participation for prescription; partial participation for patient. Prescriptions must be linked to a patient. Patients, however, may not have any prescriptions.

3.2.2.3 Administers

Relationship: Administers

Description: A nurse administers a prescribed medication. A patient may have multiple nurses administering care in one day.

Entity Sets Involved: Nurse, Prescription

Mapping Cardinality: M N

Descriptive Field: Date/Time. Nurses must record the date and time the medication is administered.

Participation Constraint: Partial participation for nurse; Partial participation for prescription. New prescriptions may not have yet been administered. Nurses can work for the hospital without having administered a prescription.

3.2.2.4 Takes

Relationship: Takes

Description: A nurse will take a vitals assessment. This is usually done every 4 hours.

Entity Sets Involved: Nurse, Assessment

Mapping Cardinality: 1 N

Descriptive Field: Date/Time. A nurse takes a vitals assessment multiple times a day and records the date and time of each assessment.

Participation Constraint: Partial participation for nurse; total participation for assessment. A nurse can work for the hospital without having taken a vitals assessment. A vitals assessment, however, must be taken by a nurse.

3.2.2.5 Undergoes

Relationship: Undergoes

Description: A patient undergoes a vitals assessment. A patient may undergo none or many vitals assessments, but each assessment is for one patient only.

Entity Sets Involved: Patient, Vitals

Mapping Cardinality: 1 N

Participation Constraint: Partial participation for participant; total participation for vitals assessment. Newly admitted patients may not have had vital assessments. A vitals assessment, however, must be linked to a specific patient.

3.2.2.6 Orders

Relationship: Orders

Description: A doctor orders a special need. One doctor many order none or many special needs.

Entity Sets Involved: Doctor, Special Need

Mapping Cardinality: 1 N

Descriptive Field: Date/Time. The doctor orders a special need on a specific date.

Participation Constraint: Partial participation for doctor; total participation for special need. Doctors can be employed by the hospital without having ordered a special need. A special need, however, must be ordered by a doctor.

3.2.2.7 Performs

Relationship: Performs

Description: Nurses perform one or many special needs. These special needs are assigned to a specific patient and may be performed multiple times throughout the day. Nurses perform the special need according to the instructions and frequency provided in the doctor's orders. Many nurses may special the special needs necessary for a specific patient.

Entity Sets Involved: Nurse, Special Need

Mapping Cardinality: M N

Descriptive Field: Date/Time. Special needs must be performed according to the date, time, and frequency assigned by the doctor.

Participation Constraint: Partial participation for nurse; partial participation for special need. Nurses can be employed at the hospital without having performed a special need. A special need may not have yet been performed.

3.2.2.8 Receives (Special Need)

Relationship: Receives (Special Need)

Description: A patient receives a special need order from a doctor. Each special need order is specific to one patient, but a patient can receive multiple special needs.

Entity Sets Involved: Patient, Special Need

Mapping Cardinality: 1 N

Participation Constraint: Partial participation for patient; total participation for special need. Newly admitted patients may not have been given any special needs (and outpatients may not have any special needs). However, special need orders must be linked to a specific patient.

3.2.2.8 (Nurse) Assigned to

Relationship: (Nurse) Assigned To

Description: Nurses are assigned to a patient during their shift. Although one nurse is assigned to one patient during a shift, this occurs over the course of time. Thus, many patients may be assigned to the same nurse and many nurses may be assigned to the same patient.

Entity Sets Involved: Nurse, Patient

Mapping Cardinality: M N

Descriptive Field: StartDate, StartTime

Participation Constraint: Partial participation for nurse; partial participation for patient. Nurses can be employed at a hospital without being assigned to a patient. Patients can be in the hospital without being assigned to a nurse.

3.2.2.9 Works in

Relationship: Works In

Description: A nurse works in a unit within the hospital. Most nurses work within only one unit during their employment, but this is not always the case. Overtime, a nurse may work within multiple units and a unit may employ several nurses.

Entity Sets Involved: Nurse, Unit

Mapping Cardinality: M N

Descriptive Field: StartDate, StartTime, EndDate, EndTime

Participation Constraint: Total participation for nurse; total participation for unit. A nurse must work within a unit and all units must employ nurses.

3.2.2.10 Admitted

Relationship: Admitted

Description: A patient is admitted to a specific hospital room. Not all patients are admitted (some remain outpatients). At a specific point in time, none, one, or two patients may be placed in a room, depending on the room type. Patients can also move from room to room, depending on their health conditions and requirements.

Entity Sets Involved: Patient, Room

Mapping Cardinality: M N

Descriptive Field: AdmitDate, AdmitTime, DischargeDate, DischargeTime

Participation Constraint: Partial participation for patient; partial participation for room. A patient may or may not be admitted to the hospital; a room may or may not have any patients.

3.2.2.11 Belongs To

Relationship: Belongs To

Description: A room belongs to a unit of the hospital. A single unit contains many rooms.

Entity Sets Involved: Room, Unit

Mapping Cardinality: M 1

Participation Constraint: Total participation for room; total participation for unit. All rooms must belong to a unit. All units must contain one or many rooms.

3.2.2.12 Takes Image

Relationship: Takes Image

Description: A hospital technician takes none or many image (e.g., MRI, X-ray, CAT scan). Although a specific image must be taken by one technician, overtime, a technician may take several images and several image types may be taken by multiple technicians. These images are assigned to a specific patient and may be performed multiple times throughout the day. Technicians take the images according to the instructions provided in the doctor's orders.

Entity Sets Involved: Technician, Image

Mapping Cardinality: M N

Participation Constraint: Partial participation for technician; partial participation for image. Technicians can be employed at the hospital without having taken any images. A specific image may not yet be performed by any technician.

3.2.2.13 Has Imaging

Relationship: Has Imaging

Description: A patient has imaging done. A patient has imaging by a technician at the location and specifications set by the patient's doctor. A patient may have none or many images done. A

Entity Sets Involved: Patient, Image

Mapping Cardinality: M N

Participation Constraint: Partial participation for patient; total participation for image. Not all patients will receive imaging services. However, all images must be linked to a specific patient.

3.2.2.14 Orders Imaging

Relationship: Orders Imaging

Description: A doctor orders imaging for a patient. A doctor many order none or many images for a patient. An image may or may not be ordered by a doctor.

Entity Sets Involved: Doctor, Image

Mapping Cardinality: M N

Participation Constraint: Partial participation for doctor; partial participation for image

3.2.2.15 Takes Labs

Relationship: Takes Labs

Description: A hospital technician takes none or many labs (e.g., blood lipid analysis, urine analysis, etc) Although a specific lab must be taken by one technician, overtime, a technician may take several images and several image types may be taken by multiple technicians. These labs are assigned to a specific patient and may be performed multiple times throughout the day. Technicians take the labs according to the instructions provided in the doctor's orders.

Entity Sets Involved: Technician, Lab

Mapping Cardinality: M N

Participation Constraint: Partial participation for technician; partial participation for lab. Technicians can be employed at the hospital without having taken any labs. A specific lab may not yet be performed by any technician.

3.2.2.16 Has Labs

Relationship: Has Labs

Description: A patient has labs done. A patient has labs done by a technician according to the specifications set by the patient's doctor. A patient may have none or many labs done.

Entity Sets Involved: Patient, Lab

Mapping Cardinality: M N

Participation Constraint: Partial participation for patient; total participation for lab. Not all patients will receive labs. However, all labs must be linked to a specific patient.

3.2.2.17 Orders Labs

Relationship: Orders Labs

Description: A doctor orders labs for a patient. A doctor many order none or many images for a patient. A lab may or may not be ordered by a doctor.

Entity Sets Involved: Doctor, lab

Mapping Cardinality: M N

Participation Constraint: Partial participation for doctor; partial participation for lab

3.2.2.18 Performs Procedure

Relationship: Performs Procedure

Description: A doctor performs a medical procedure on a patient. A doctor may perform one or none procedures. A specific medical procedure must be performed by a doctor, but as this may occur over time, a procedure can be performed by many doctors.

Entity Sets Involved: Doctor, Procedure

Mapping Cardinality: M N

Participation Constraint: Partial participation for doctor, partial participation for procedure. Doctors can be employed at the hospital without having performed a procedure. A procedure may not have yet been performed.

3.2.2.19 Receives Procedure

Relationship: Received Procedure

Description: A patient receives a medical procedure from a doctor. Each patient may receive multiple procedures. Overtime, a procedure many be performed on many patients.

Entity Sets Involved: Patient, Procedure

Mapping Cardinality: M N

Participation Constraint: Partial participation for patient; total participation for procedure. Not all patients will receive procedures. However, all procedures must be linked to a specific patient.

3.2.2.16 Treats

Relationship: Treats

Description: Doctors treat none or multiple patients. Multiple doctors may be assigned to treat one or many patients.

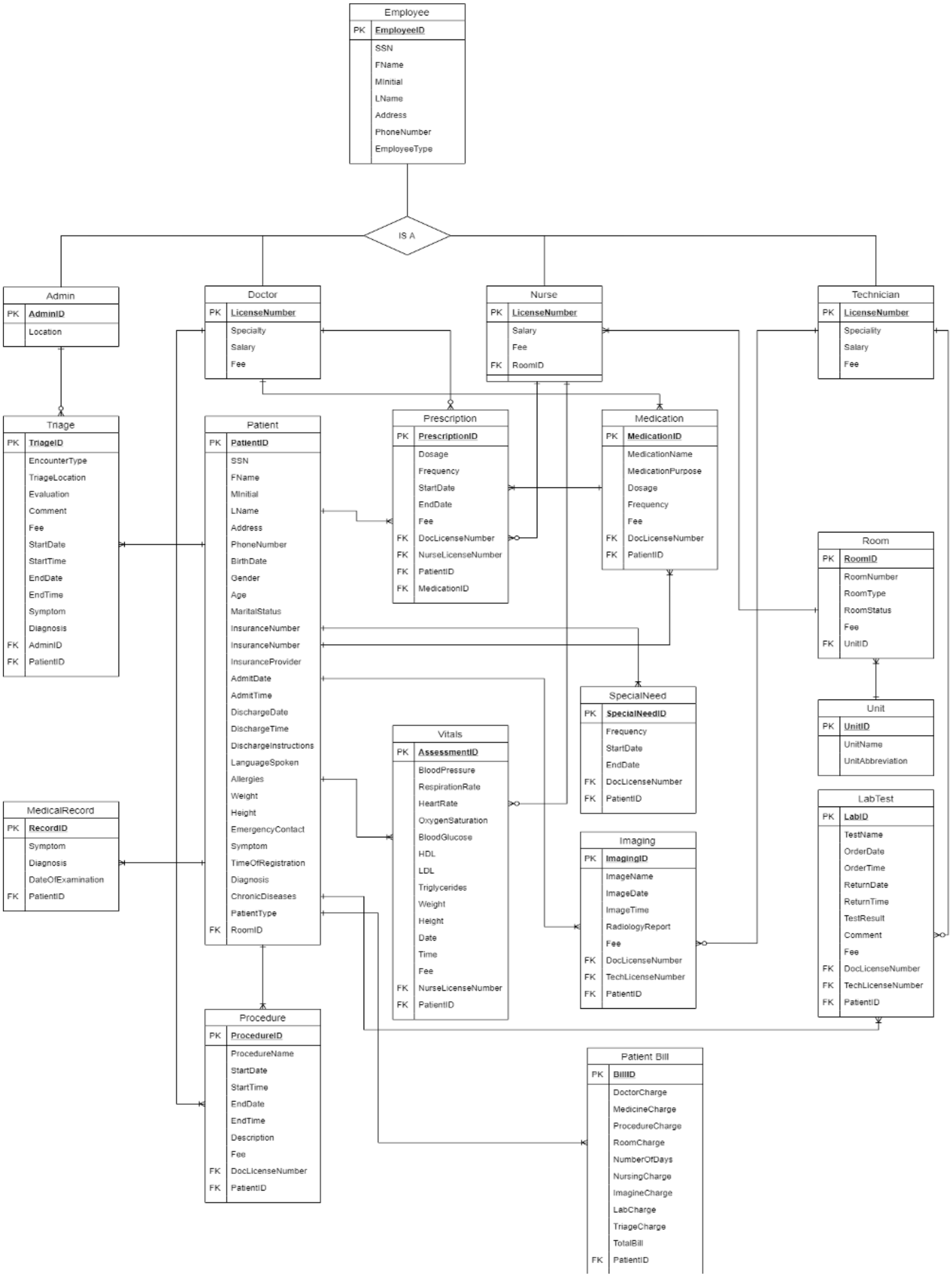
Entity Sets Involved: Doctor, Patient

Mapping Cardinality: M N

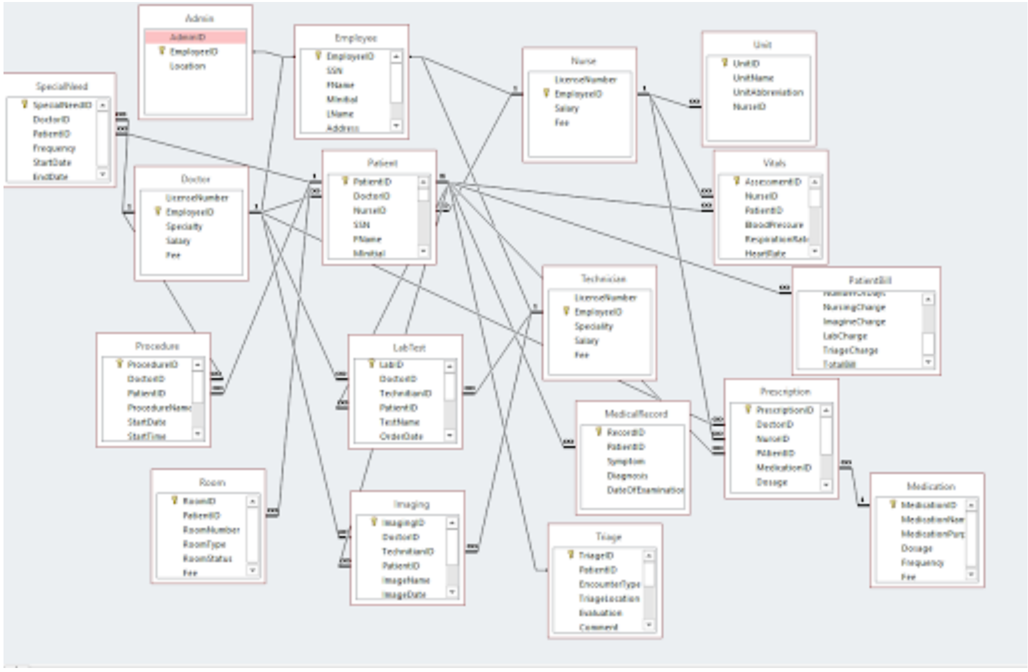
Participation Constraint: Partial participation for doctor; partial participation for patient. Doctors can be employed at a hospital without being assigned to a patient. Patients can be in the hospital without being assigned to a doctor (e.g., triage may not require an assigned doctor).

3.2.3 E/R Diagram

The original E/R diagram was corrected prior to implementing the Low-Country Memorial database in MS-Access and is presented below.



Since MS-Access also produces an E/R diagram, it is included in the report below.



3.3 Relational Model

The relational model is a method for creating logical database designs. Specifically, the relational model reflects how the database will be implemented in software. While the E/R model represents the conceptual design of the database by describing entities, attributions, and relationships, the relational model represents the logical design of the database by describing “the relation”. A relational schema consists of a list of attributes in a tuple. Groups of tuples represent a relation instance and are displayed logically as a table. In this table, the attributes are represented as columns and the tuples are represented as rows.

In order to create the relational model, each strong entity type is converted into one relation schema, which has the same name as the strong entity type. The attributes of the relation schema as the simple, single-value attributes of the entity type along with simple components of the entity type’s composite attributes. Each multi-value attribute of an entity type is represented with a new, separate relation. One key attribute becomes the primary key attribute and the others become candidate keys.

All 1:N relationship types were mapped using a foreign key approach. All M:N relationship types were mapped through the cross reference approach. Superclasses and Subclasses were mapped for the “IsA” Relationship, where entity types are disjoint subclasses of a superclass entity type, or the “HasA” Relationship, where entity types are overlapping subclasses of a superclass entity type.

3.3.1 Data Dictionary

In the data dictionary, we take abbreviation of the attribute names for the column names. It will be used when the full application is developed including the front end parts as well. Listed below are examples of the data dictionaries for select relation schema.

3.3.1.1 Employee Relation Schema

employee(EID, SSN, FName, MInitial, LName, street, city, state, zip, EType, PNum, licType, licNo, salary)

Candidate Keys: empID (primary), SSN

Primary Key/Entity Integrity Constraint: EID must be unique and cannot be NULL

Uniqueness Constraint: SSN must be unique

Derivation From Entity and Relationship Types: Employee, Doctor, Nurse, Technician, Admin
Derived from the Employee entity type. Represents the “is_a” specialization using the single relation with one type attribute method; “licType” is the type attribute for discerning between employee types. The composite “Address” attributes of the Employee Entity Type are broken into their simple components.

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
EID	Employee ID	Integer	0-MaxID	Primary Key	Y	Any alpha-numeric value
SSN	Social Security Number	Integer	000000000 – 999999999	Primary Key	Y	6 numeric digits
FName	First Name	Varchar2	255	Primary Key	Y	Any alphabetic value
MInitial	Middle Initial	Varchar2	255	Primary Key	Y	Any alphabetic value
LName	Last Name	Varchar2	255	Primary Key	Y	Any alphabetic value
Street	Street address	Varchar2	255	Primary Key	Y	Any alpha-numeric value
City	City of home address	Varchar2	255	Primary Key	Y	Any alpha-numeric value
State	State of home address	Varchar2	255	Primary Key	Y	Any alpha-numeric value
Zip	Zip code of home address	Integer	00000 – 99999	Primary Key	Y	5 numeric digits
PNum	Phone Number	Integer	0000000000 - 9999999999	Primary Key	Y	10 numeric digits
EType	Employee Type	Varchar2	255	Primary Key	Y	Any alphabetic value
licType	Licence Type “MD” ,“RD”, “TC", “LB”	Varchar2	255	Primary Key	Y	Any alphabetic value
licNo	License number	Integer	000000 - 999999	Primary Key	Y	6 numeric digits

3.3.1.2 Prescription Relation Schema

prescription(rxID, EID, patientID, MID, DOS, freq, SDate, EDate, fee)

Candidate Keys: rxID (primary)

Primary Key/Entity Integrity Constraint: rxID is unique and cannot be NULL

Referential Integrity Constraint: EID is a foreign key for employee, patientID is a foreign key for patient, MID is a foreign key for medication

Business Constraint: SDate is never greater than EDate; The employee tuple referenced by EID must be for a doctor (licType = MD)

Derivation From Entity and Relationship Types: Prescription, Prescription has medication Medication, Doctor prescribes Prescription, Prescription prescribed for Patient Derived from the Prescription entity type. Represents the N:1 relationships to Doctor, Patient, and Medication using the foreign key approach, since the participation of Prescription is total for all of these relationships.

Column Name	**Description**	**Data Type**	**Size**	**Constraint Type**	**Not Null ?**	**Valid Values**
rxID	Prescription ID	Integer	0-MaxID,	Primary Key	Y	6 numeric digits
EID	Employee ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
patientID	Patient ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
DOS	Dosage	Varchar2	255	Primary Key	Y	Any alphabetic value
Feq	Frequency	Varchar2	255	Primary Key	Y	2 numeric digits
SDate	Start Date	Date		Primary Key	Y	Any Date
EDate	End Date	Date		Primary Key	Y	Any Date
Fee	Fees	Integer		Primary Key	Y	Up to 12 numeric digits
MID	Medication ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value

3.3.1.3 Medication Relation Schema

medication(MID, medName, medPur, DOS, freq, SDate, EDate, fee)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
MID	Medication ID	Integer	0-MaxID	Primary Key	Y	Any numeric value
medName	Medication Name	Varchar2	255	Primary Key	Y	Any alpha-numeric value
medPur	Medication Purpose	Varchar2	255	Primary Key	Y	Any alpha-numeric value
DOS	Dosage	Varchar2	255	Primary Key	Y	Any alphabetic value
Feq	Frequency	Varchar2	255	Primary Key	Y	2 numeric digits
SDate	Start Date	Date		Primary Key	Y	Any Date
EDate	End Date	Date		Primary Key	Y	Any Date
Fee	Fees	Integer	0-999999	Primary Key	Y	Up to 6 numeric digits

Candidate Keys: medID, medName

Primary Key/Entity Integrity Constraint: MID is unique and cannot be NULL

Uniqueness Constraint: medName must be unique

Derivation From Entity and Relationship Types: Medication

Derived from the Medication entity type. All attributes are simple and single-value.

3.3.1.3 Room Relation Schema

room(RID, UID, RNum, RType, RSta, fee)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
RID	Room ID	Integer	0-MaxID	Primary Key	Y	Any alpha-numeric value
RNum	Room Number	Integer	0-999	Primary Key	Y	5 numeric digits
RType	Room Type	Varchar2	255	Primary Key	Y	Any alphabetic value
RSta	Room Status	Varchar2	255	Primary Key	Y	Any alphabetic value
UID	Unit ID	Integer	0-MaxID	Foreign Key	Y	4 numeric digits
Fee	Fee for Room	Integer	0-999999	Primary Key	Y	6 numeric digits

Candidate Keys: RID (primary), [UID, RNum]
*[<key1>,<key2>] denotes a compound key

Primary Key/Entity Integrity Constraint: RID is unique and cannot be NULL

Uniqueness Constraint: The combination of UID, RNum is unique (no two rooms in the same unit have the same number)

Referential Integrity Constraint: UID is a foreign key for unit

Derivation From Entity and Relationship Types: Room, Room belongs to Unit

Derived from the Room entity type. Represents the N:1 relationship with Unit using the foreign key approach since the participation of Room is total.

3.3.1.4 Unit Relation Schema

unit(UID, UName, UAbbrv)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
UID	Unit ID	Integer	0-MaxID	Primary Key	Y	4 numeric digits
UName	Unit Name	Varchar2	255	Primary Key	Y	Any alphabetic value
Uabbrv	Unit Abbreviated Name	Varchar2	10	Primary Key	Y	Any alphabetic value

Candidate Keys: UID (primary), Uname, Uabbrv

Primary Key/Entity Integrity Constraint: UID is unique and cannot be NULL

Uniqueness Constraint: Uname is unique, Uabbrv is unique

Derivation From Entity and Relationship Types: Unit

Derived from the unit entity type. All attributes are simple and single-value.

3.3.1.5 Patient Relation Schema

patient(patientID, SSN, FName, MInitial, LName, street, city, state, zip, PNum, DOB, Gender, insurNum, lang)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
patientID	Patient ID	Integer	0-MaxID	Primary Key	Y	Any alpha-numeric value
SSN	Social Security Number	Integer	000000000 – 999999999	Primary Key	Y	9 numeric digits
FName	First Name	Varchar2	255	Primary Key	Y	Any alphabetic value
MInitial	Middle Initial	Varchar2	255	Primary Key	Y	Any alphabetic value
LName	Last Name	Varchar2	255	Primary Key	Y	4 numeric digits
Street	Home street address	Varchar2	255	Primary Key	Y	Any alphabetic value
City	Home city	Varchar2	255	Primary Key	Y	Any alphabetic value
State	Home state	Varchar2	255	Primary Key	Y	Any alphabetic Value
Zip	Home zipcode	Integer	00000 – 99999	Primary Key	Y	5 numeric digits
PNum	Phone number	Integer	0000000000 – 9999999999	Primary Key	Y	10 numeric digits
Dob	Date of Birth	Date	Date	Primary Key	Y	
Gender	Gender of Patient (either Male or Female)	Character	F or M	Primary Key	Y	
insurNum	Insurance number	Integer	0000000000 – 9999999999	Primary Key	Y	Any numeric value
lang	Language	Varchar2	255	Primary Key	Y	Any alphabetic value

Candidate Keys: patientID, SSN

Primary Key/Entity Integrity Constraint: patientID is unique and cannot be NULL

Uniqueness Constraint: SSN is unique

Derivation From Entity and Relationship Types: Patient

Derived from the Patient entity type. The composite "Name" and "Address" attributes are broken into simple component attributes.

3.3.1.6 PatientAdmitted Relation Schema

patientAdmitted(patientID, AdmitDate, DischargeDate)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
patientID	Patient ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
AdmitDate	Start Date	Date		Primary Key	Y	Any
DischargeDate	End Date	Date		Primary Key	Y	Any

Candidate Keys: [patientID, AdmitDate]

Primary Key/Entity Integrity Constraint: The combination of patientID and AdmitDate is unique and cannot be NULL

Referential Integrity Constraint: patientID is a foreign key for room

Business Constraint: SDate cannot be greater than EDate

Derivation From Entity and Relationship Types: Patient

Derived from the multi-value attributes AdmitDate and DischargeDate in the Patient entity type.

3.3.1.7 Placed Relation Schema

placed(patientID, RID, SDate, EDate)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
patientID	Patient ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
RID	Room ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
SDate	Start Date	Date		Primary Key	Y	Any
EDate	End Date	Date		Primary Key	Y	Any

Candidate Keys: [patientID, RID, SDate]

Primary Key/Entity Integrity Constraint: The combination of patientID, RID, SDate is unique and cannot be NULL

Referential Integrity Constraint: RID is a foreign key for room

Business Constraint: SDate cannot be greater than EDate

Derivation From Entity and Relationship Types: Patient placed Room

Derived from the M:N Patient placed Room relationship type using the cross-reference approach; placed is a “relationship relation” with foreign keys for Patient and Room.

3.3.1.8 Assessment Relation Schema

assessment(asmtID, EID, patientID, bpSystolic, bpDiastolic, respRate, heartRate, date)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
asmtID	Assessment ID	Integer	0-MaxID	Primary Key	Y	Any numeric value
EID	Employee ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
patientID	Patient ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
bpSystolic	Systolic Blood Pressure	Integer	0-300	Primary Key	Y	Any numeric value
bpDiastolic	Diastolic Blood Pressure	Integer	0-300	Primary Key	Y	Any numeric value
respRate	Respiration rate	Integer	0-300	Primary Key	Y	Any numeric value
heartRate	Heart Rate	Integer	0-300	Primary Key	Y	Any numeric value
date	timestamp			Primary Key	Y	Any numeric value

Candidate Keys: asmtID, [EID, date], [patientID, date]

Primary Key/Entity Integrity Constraint: asmtID is unique and cannot be NULL

Referential Integrity Constraint: EID is a foreign key for employee, patientID is a foreign key for patient

Business Constraint: None

Derivation From Entity and Relationship Types: Assessment, Nurse takes Assessment, Patient undergoes Assessment

Derived the Assessment entity type. The composite attribute “Blood Pressure” is broken down into simple component attributes. Represents the N:1 relationships to Nurse and

Patient using the foreign key approach, since the participation of Assessment in both relationships is total.

3.3.1.9 SpecialNeed Relation Schema

specialNeed(SNID, EID, patientID, freq, SDate, EDate, date)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
SNID	Assessment ID	Integer	0-MaxID	Primary Key	Y	Any numeric value
EID	Employee ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
patientID	Patient ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
freq	Frequency of special need	Varchar2	255	Primary Key	Y	Any alpha-numeric value
SDate	Start Date	Date		Primary Key	Y	Any
EDate	End Date	Date		Primary Key	Y	Any
date	Timestamp	Date		Primary Key	Y	Any

Candidate Keys: SNID

Primary Key/Entity Integrity Constraint: SNID is unique and cannot be NULL

Referential Integrity Constraints: EID is a foreign key for employee, patientID is a foreign key for patient

Business Constraints: SDate cannot be greater than EDate; The employee tuple referenced by EID must be a doctor (“licType” = “MD”)

Derivation From Entity and Relationship Types: Special Need, Special Need has activity Activity, Doctor orders Special Need

Derived from the Special Need entity type. All attributes are simple and single-value. Represents the N:1 relationships with Doctor and Activity using the foreign key approach, since the participation of Special Need in both relationships is total.

3.3.1.10 Administers Relation Schema

administers(EID, rxID, date)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
EID	EmployeeID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
rxID	Prescription ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
Date	Timestamp			Primary Key	Y	Any

Candidate Keys: [EID, date], [rxID, date]

Primary Key/Entity Integrity Constraint: The combination of EID, date must be unique and cannot be NULL

Business Constraint: The employee tuple referenced by EID is a nurse (“licType” = “RN”)

Referential Integrity Constraints: EID is a foreign key for employee, rxID is a foreign key for prescription

Derivation From Entity and Relationship Types: Nurse administers Prescription
Derived from the M:N relationship between Nurse and Prescription using the cross-reference approach; administers is a “relationship relation.”

3.3.1.11 Assigned To Relation Schema

assignedTo(EID, patientID, SDate, EDate)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
EID	EmployeeID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
patientID	PatientID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
SDate	Start Date	Date		Primary Key	Y	Any
EDate	End Date	Date		Primary Key	Y	Any

Candidate Keys: [EID, patientID, SDate]

Primary Key/Entity Integrity Constraint: The combination of EID, patientID, SDate must be unique and cannot be NULL

Referential Integrity Constraints: EID is a foreign key for employee, patientID is a foreign key for patient

Business Constraint: SDate cannot be greater than EDate

The employee tuple referenced by EID is a nurse (“licType” = “RN”)

Derivation From Entity and Relationship Types: Nurse assigned to Patient
Derived from the M:N relationship between Nurse and Patient using the cross-reference approach; assignedTo is a “relationship relation.”

3.3.1.12 Performs Relation Schema

Performs(EID, SNID, date)

Column Name	Description	Data Type	Size	Constraint Type	Not Null?	Valid Values
EID	EmployeeID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
SNID	Special Need ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
Date	Timestamp	Date		Primary Key	Y	Any

Candidate Keys: [EID, date], [SNID, date]

Primary Key/Entity Integrity Constraint: The combination of EID, date must be unique and cannot be NULL

Referential Integrity Constraints: EID is a foreign key for employee, SNID is a foreign key for specialNeed

Business Constraint: SDate cannot be greater than EDate

The employee tuple referenced by EID is a nurse (“licType” = “RN”)

Derivation From Entity and Relationship Types: Nurse performs Special Need
Derived from the M:N relationship between Nurse and Special Need using the cross-reference approach; performs is a “relationship relation.”

3.3.1.13 WorksIn Relation Schema

worksIn(EID, UID, SDate, EDate)

Column Name	Description	Data Type	Size	Constraint Type	Not Null ?	Valid Values
EID	Employee ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
UID	Unit ID	Integer	0-MaxID	Foreign Key	Y	Any numeric value
SDate	Start Date	Date		Primary Key	Y	Any
EDate	End Date	Date		Primary Key	Y	Any

Candidate Keys: [EID, UID, SDate]

Primary Key/Entity Integrity Constraint: The combination of EID, UID, SDate must be unique and cannot be NULL

Referential Integrity Constraints: EID is a foreign key for employee, UID is a foreign key for unit

Business Constraints: SDate cannot be greater than EDate

The employee tuple referenced by EID is a nurse (“licType” = “RN”)

Derivation From Entity and Relationship Types: Nurse works in Unit
Derived from the M:N relationship between Nurse and Unit using the cross-reference approach; worksIn is a “relationship relation.”

3.3.1.14 Others Relation Schema

For brevity, the remaining relation schema will not be discussed (e.g., triage, imaging, labtest, medicalrecord, patientbill, procedure).

3.3.2 Integrity Rules

There are two types of integrity rules that have been followed for this database. First is the entity integrity and second one is the referential integrity. Entity integrity is important for forming a part of the candidate key of any specific relation. The differential integrity is important as it defines the possible actions that can be used when any deletions, instructions and updates are made to the database. Examples of both entity integrity and referential integrity were outlined in section 3.3.1.

3.3.3 Operational Rules

Examples of operational rules were outlined in section 3.3.1. However, they will also be reviewed here. Specifically, operational constraints include the following: start dates must never be greater than end dates for all medications or doctor orders; similarly, hospital admittance date cannot be greater than hospital discharge date; an employee prescribing a medication must have a license type which includes "MD"; similarly, an employee ordering a special need must have a license type which includes "MD"; an employee administering a medication must have a license type which includes "RD"; an employee performing a special need must have a license type which includes "RD"; an employee performing a procedure must have a license type which includes "MD"; patients must be assigned to employees with either license types which include "MD" or "RD"; employees conducting laboratory testing must be "technicians"; employees conducting imaging services much be "technicians"; employees conducting triage interviews must be "admins"; an employee who works in a unit must have the license type which includes "RD".

3.3.4 Operations

Operations for each entity type were described in Section 3.2.1.

3.4 Security

This database system requires two types of security. First is physical security and second one is data security. For the physical security a separate and sealed room will be used for the hardware infrastructure. For the data security data encryption will be implemented within the database.

Within MS-Access itself, security tools are provided to encrypt the database with a password and/or provide a workgroup administrator. Given more time, it would be nice to split the database into a front-end (form) and back-end (tables) format to prevent users from altering the structure of the database itself.

3.5 Database Backup and Recovery

For the backup and recovery of this database, a log based recovery system will be implemented. In this case all the database transactions will be stored securely and they will be restored if required. For optimal protection, a stored copy of the backed up database will be kept in an offsite location and updated on a periodic basis. Database backups will be encrypted, as they contain sensitive information.

3.6 Using Database Design or CASE Tool

The important CASE tools that will be used in this case are the Information Engineering Facility and Oracle Designer.

3.7 Other Possible E/R Relationships

Semantic database design has been considered as an alternative during the design of the database.

4. Implementation Description

4.1 Data Dictionary

Column Name	Describe
AdmitDate	Date patient was admitted to hospital

DischargeDate	Date patient was discharged from hospital
bpSystolic	Systolic Blood Pressure
bpDiastolic	Diastolic Blood Pressure
respRate	Respiration Rate
heartRate	Heart Rate
EID	It describes ID of employee
SSN	It holds the Social Security Number
FName	First Name of employee or patient is recorded
MInitial	Middle Initial of employee or patient is recorded
LName	Last Name of employee or patient is recorded
Street	Street address of employee or patient is recorded
City	City of home address of employee or patient
Zip	Zipcode of home address of employee or patient
PNum	Phone Number of employee or patient is recorded
EType	Job role of employee is recorded
licType	License type of employee
licNo	License number of employee
Dob	Date of birth
Gender	Gender (F or M)
PID	ID of prescription is recorded
patientID	Patient ID is recorded
DOS	Medicine dosage is recorded
Date	Timestamp for a specific date
Fee	Fee or charge is recoded
Feq	Dosage frequency is recorded
insurNum	Insurance number
SDate	Start Date for the dosage
EDate	End Date for the dosage
Fee	Fees of the patient
Lang	Language of patient
MID	Medication ID for the patient
medName	Medication Name applied to the patient
medPur	Purpose of medication is recorded
RID	Room ID is recorded
RNum	Room Number is recorded
RType	Room Type is recorded
RSta	Room Status is recorded
rxID	Prescription ID is recored
SNID	Special Need ID is recorded
UID	Unit ID is recorded
UName	Unit Name is recorded
Uabbrv	Unit Abbreviation

4.2 Advanced Features

There are two important advanced features for this database system: bill calculation and gathering of multiple patient records. Bill calculation includes room charge, procedure charge, imaging charge, prescription charges, physician charge, and lab charge. In order to gather multiple patient records, the database takes the records from symptoms, diagnosis by procedure, length of stay, total charges, room status and other important data from the actual database.

4.3 Queries

All the following queries are specific to MS-Access only.

4.3.1 Read Patient Record

The Result can be obtained by simply using select statement

```
Select * from patient;
```

4.3.2 Print Patient Discharge Information

The result can be obtained by using select statement and retrieving discharge related details

```
Select PatientID, FName, MIntial, Lname, DischargeDate,
DischargeTime, Discharge instructions from patient;
```

4.3.3 Calculate a Bill

4.3.3.1.1 Room charge

```
SELECT p.patientID, r.Fee
FROM room r
INNER JOIN patient p ON r.PatientID = p.patientID;
```

4.3.3.1.2 Procedure charge

```
SELECT p.patientID, pro.Fee
FROM procedure pro
INNER JOIN patient p ON pro.PatientID = p.patientID;
```

4.3.3.1.3 Prescription charge

```
SELECT p.patientID, pre.Fee
FROM prescription pre
INNER JOIN patient p ON pre.PatientID = p.patientID;
```

4.3.3.1.4 Imaging charge

```
SELECT p.patientID, i.Fee
FROM image i
INNER JOIN patient p ON i.PatientID = p.patientID;
```

4.3.3.1.5 Lab charge

```
SELECT p.patientID, lt.Fee
FROM LabTest lt
INNER JOIN patient p ON lt.PatientID = p.patientID;
```

4.3.3.1.6 Physician charge

Since, Physician charge is the same as doctor fee, we will display the doctor fee in the result (not the sum of doctor salary and fee)

```
SELECT p.patientID, d.Fee
FROM doctor d
INNER JOIN patient p ON d.PatientID = p.patientID;
```

4.3.4 Gather multiple patient records

4.3.4.1 Length of stay by diagnoses

```
SELECT DateDiff("d", [AdmitDate], [DischargeDate])
FROM Patient;
```

4.3.4.2 Length of stay by procedure

```
SELECT DateDiff("d", [StartDate], [EndDate])
FROM Procedure;
```

4.3.4.3 Current census (or, current census by physician)

```
SELECT count(PatientID)
FROM Patient
GROUP BY DoctorID;
```

4.3.4.4 Diagnoses by treatment

```
SELECT PatientID, Diagnosis
From patient
Order by ChronicDiseases ASC;
```

4.3.4.5 Total charges by diagnoses

```
SELECT p.PatientID, p,Diagnosis, pb.TotalBill
FROM PaientBill pb
INNER JOIN patient p ON pb.PatientID = p.patientID
ORDER by p.Diagnosis;
```

4.3.4.6 Total charges by procedure

```
SELECT p.PatientID, pro.ProceduerID , pb.TotalBill
FROM PaientBill pb
INNER JOIN patient p ON pb.PatientID = p.patientID
INNER JOIN procedure pro ON p.PatientID = pro.patientID
ORDER by pro.ProcedureID;
```

4.3.4.7 List length of visit for outpatients at a specific triage location within a specific date range

```
SELECT count(PatientID)
From Triage
Where TriageLocation = “LocationA” and
EndDate Between #01/07/2020# AND #31/07/2020#
```

4.3.4.8 List all diagnoses for all current inpatients

```
Select PatientID, Diagnosis
From Patient
WHERE DischargeDate IS NULL;
```

4.3.4.9 List all current inpatients waiting for a procedure

```
Select PatientID
from Patients
Where PatientID NOT IN (Select PatientID from Procedure);
```

4.3.4.10 List all nurses assigned to 2 or more patients in the same time period

```
Select NurseID
From Patient
Where count(PatienyID)>=2
Group By NurseID;
```

4.3.4.11 List the medications that all current inpatients have in common

```
Select MedicationID
From Prescription
WHERE count(PatientID) = (Select
MAX(Count(PatientID)) from prescription Group By
MedicationID)
Group By MedicationID;
```

4.3.4.12 List all patients who received the same special needs order as “John Doe”

```
Select p.PatientID
from Patient p
Inner Join SpecialNeed sn ON p.PatientID = sn.PatientID
Where SpecialNeedID = (Select s. SpecialNeedID
from Patient pa
Inner Join SpecialNeed s ON pa.PatientID = s.PatientID
 WHERE pa.FName = “John” and pa.LName = “Doe”);
```

4.3.4.13 List all nurses who have administered one of “Jane Doe’s” prescriptions

```
Select NurseID from Patient
Where FName = “John” and LName = “Doe”;
```

4.3.4.14 List all the image services requested by physician “John Doe” for a specified diagnosis

```
Select i.ImagingID
From Employee e
Inner Join Doctor d ON e.EmployeeID = d.EmployeeID
Inner Join Imaging i ON d.employeeID = i.DoctorID
Where e.FName = “John” e.LName = “Doe”;
```

4.3.4.15 List the allergies of all currently admitted patients

```
Select PatientID, Allergies
From Patient
WHERE DischargeDate IS NULL;
```

5. CRUD Matrix

5.1 List of Entity Types

- E1: Employee
- E2: Doctor
- E3: Nurse
- E4: Technician
- E5: Admin
- E6: Prescription
- E7: Medication
- E8: Room
- E9: Vitals
- E10: Unit
- E11: Special Need
- E12: Patient
- E13: Medical Record
- E14: Procedure
- E15: Imaging
- E16: Lab Test
- E17: Triage
- E18: Patient Bill

5.2 List of Functions

- F1: Insert/Update/Delete entities
- F2: Calculate a Bill
- F3: Gather patient records

Function/ Entity Interaction	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18
F1	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD
F2		R	R			R	R	R				R		R	R	R	R	CU
F3	R	R										R	R		R	R		R

6. Concluding Remarks

This database project was designed to help students, enrolled in a Fundamentals of Database Principles course, understand the conceptual and logical design of relational databases, as well as database implementation. Here, a relational database was created to mimic a real-world hospital, but included the functionality needed for both inpatient and outpatient services. The Low-Country Memorial Hospital database project includes all the information of the patients, doctors, nurses, technicians, admins, lab test, vitals, visits, prescription, medicines, procedures, and bills.

References

Ab Rahman, N. H., Glisson, W. B., Yang, Y., & Choo, K. K. R. (2016). Forensic-by-design framework for cyber-physical cloud systems. *IEEE Cloud Computing*, *3*(1), 50-59. DOI: 10.1109/MCC.2016.5

Abd Malik, S., Nordin, A., & Al-Ehaidib, R. N. (2019). Requirements Engineering (RE) Process for The Adaptation of The Hospital Information System (HIS). DOI: https://core.ac.uk/download/pdfr/325990671.pdf

Amaechi James, C., Agbasonu Valerian, C., & Nwawudu Sixtus, E. (2018). Design and Implementation of a Hospital Database Management System (HDMS) for Medical Doctors. *International Journal of Computer Theory and Engineering*, *10*(1). DOI: http://www.ijcte.org/vol10/1190-D3004.pdf

Arnetz, J. E., Hamblin, L., Ager, J., Luborsky, M., Upfal, M. J., Russell, J., & Essenmacher, L. (2015). Underreporting of workplace violence: comparison of self-report and actual documentation of hospital incidents. *Workplace Health & Safety*, *63*(5), 200-210. DOI: https://doi.org/10.1177/2165079915574684

Beaufort Memorial Hospital. (n.d.). About Us: Beaufort Memorial Hospital: Beaufort, SC. Retrieved November 22, 2020, from https://www.bmhsc.org/about-us/

Bonci, A., Pirani, M., Dragoni, A. F., Cucchiarelli, A., & Longhi, S. (2017, July). The relational model: In search for lean and mean CPS technology. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)* (pp. 127-132). IEEE. DOI: 10.1109/INDIN.2017.8104758

Gope, P., & Hwang, T. (2015). BSN-Care: A secure IoT-based modern healthcare system using body sensor network. *IEEE sensors journal*, *16*(5), 1368-1376. DOI: 10.1109/JSEN.2015.2502401

Handayani, P. W., Hidayanto, A. N., Pinem, A. A., Hapsari, I. C., Sandhyaduhita, P. I., & Budi, I. (2017). Acceptance model of a hospital information system. *International journal of medical informatics*, *99*, 11-28. DOI: https://doi.org/10.1016/j.ijmedinf.2016.12.004

Hou, J. L., & Yeh, K. H. (2015). Novel authentication schemes for IoT based healthcare systems. *International Journal of Distributed Sensor Networks*, *11*(11), 183659. DOI: https://doi.org/10.1155/2015/183659

Howdeshell, A. (2020). The Digital Archives Handbook: A Guide to Creation, Management, and Preservation. *The American Archivist*, *83*(1), 208-211. DOI: https://doi.org/10.17723/0360-9081-83.1.208

Hu, L., Qiu, M., Song, J., Hossain, M. S., & Ghoneim, A. (2015). Software defined healthcare networks. *IEEE Wireless Communications*, *22*(6), 67-75. DOI: 10.1109/MWC.2015.7368826

Jia, T., Zhao, X., Wang, Z., Gong, D., & Ding, G. (2016, June). Model transformation and data migration from relational database to MongoDB. In *2016 IEEE International Congress on Big Data (BigData Congress)* (pp. 60-67). IEEE. DOI: 10.1109/BigDataCongress.2016.16

Li, C., Hu, X., & Zhang, L. (2017). The IoT-based heart disease monitoring system for pervasive healthcare service. *Procedia computer science*, *112*, 2328-2334. DOI: https://doi.org/10.10r16/j.procs.2017.08.265

Lino, A., Rocha, Á., Macedo, L., & Sizo, A. (2019). Application of clustering-based decision tree approach in SQL query error database. *Future Generation Computer Systems*, *93*, 392-406. DOI: https://doi.org/10.1016/j.future.2018.10.038

Mezghani, E., Exposito, E., & Drira, K. (2017). A model-driven methodology for the design of autonomic and cognitive IoT-based systems: Application to healthcare. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *1*(3), 224-234. DOI: 10.1109/TETCI.2017.2699218

Son, Y., Kim, M., Kim, S., Yeom, H. Y., Kim, N. S., & Han, H. (2018). Design and Implementation of SSD-assisted Backup and Recovery for Database Systems. *IEEE Transactions on Knowledge and Data Engineering*. DOI: 10.1109/TKDE.2018.2884466

Thakkar, S. (2015). A case study for MySQL and Microsoft SQL server database reporting. DOI: http://hdl.handle.net/10211.3/158648