



Electronic Medical Record System

REQUIREMENT SPECIFICATIONS

11.21.2020

Jonathan Helms

Contents

Introduction.....	2
Purpose.....	2
Overview.....	2
Scope.....	2
Functional Requirements.....	2
UML Diagram.....	5
ER Diagram.....	6
Description.....	6
Tools Used.....	7
Tables.....	7
Table Contents.....	7
Future Revisions.....	8

Introduction

Purpose

Electronic Medical Records (EMRs) provide hospitals with a way to keep track of all the pertinent data. It's important for hospitals to have these systems because of the severity of the work that is done and for the health and safety of all the patients and staff. The quality of care improves immensely as technology advances, but with these comes more data that needs to be tracked and accessed frequently, which is where EMRs come in.

Overview

This project includes a design of a database system, EMRs, which is used in a hospital to keep track of the data. The system works by when a patient is admitted to the hospital, they will be placed into one of the various departments and given a primary doctor who administers their care. Based on which department they are sent to will determine the room they stay in. The room table will update to express that it is occupied. At the end of their stay, an invoice will be created which will be updated by off what supplies and medicines were used for the patient. The primary focus of this system is to keep a track of who the patients are, which departments were they placed in, and which doctor provides the patient with care. The secondary function is to keep track of what supplies are used during the visit and have a detailed list of supplies which also places importance on an adequate amount of items for each supply.

Scope

The goal of this project is to design a database system which will maintain and hold all the information related to a patient being admitted to a hospital. It will be designed in such a way that the patient will have relationships between what doctor treated them, the room they stayed in, and which department that room was in.

Functional Requirements

There are 9 tables that holds all the data in the system and they include Patients, Employees, Addresses/Contact Info, Insurance Companies, Invoices, Rooms, Departments, Supplies, and Suppliers.

Patients

The patient table holds relevant information for patients such as patientID, which is used as the primary key, the patient's name as pname. The foreign keys in this table are insuranceID which identifies the insurance company they use, addressID to connect to the address, docID to identify who their doctor is, and roomNum to connect to the room table. The patient table is the main entity in this design, so many of the other tables branch of it.

Employees

The employee table holds employees ID, their name, their job/role, which department they work in, and a check for whether they are a doctor or not. Only the role determines the value for the is_doc check, with only 'Doctor' as the position that receives this verification, and a trigger was added to check for this when employees are inserted. The id for doctors is used to place them with certain patients.

Addresses/Contact Info

Instead of leaving contact information in the patient table, this table will hold all that information, including street address. Suppliers' address and contact information will also be on this table, intended to have the information available when new supplies need to be purchased. The address_id is the primary key and used to connect to patients and suppliers.

Insurance Companies

InsuranceComp holds the companies name and their id number, which is used as the primary key. The id number is also used for the patient table to get which company they have insurance with.

Invoices

This table is used to hold any of the information that would be important for billing the patients after their care. The data included is an invoice number, which is the primary key, the amount due, the patient's id, and the date of the billing.

Departments

There are a variety of departments in a hospital and the data held in them include the title of the department and the department ID number. It holds relationships with the employee table and the room table to indicate which department they are a part of.

Rooms

The room table has a one-to-many relationship with the department table. It holds the number of the room, what department is a part of, and whether is it currently occupied with a patient. A trigger is used to control the value of `is_occupied` for when patients are placed in their respective rooms.

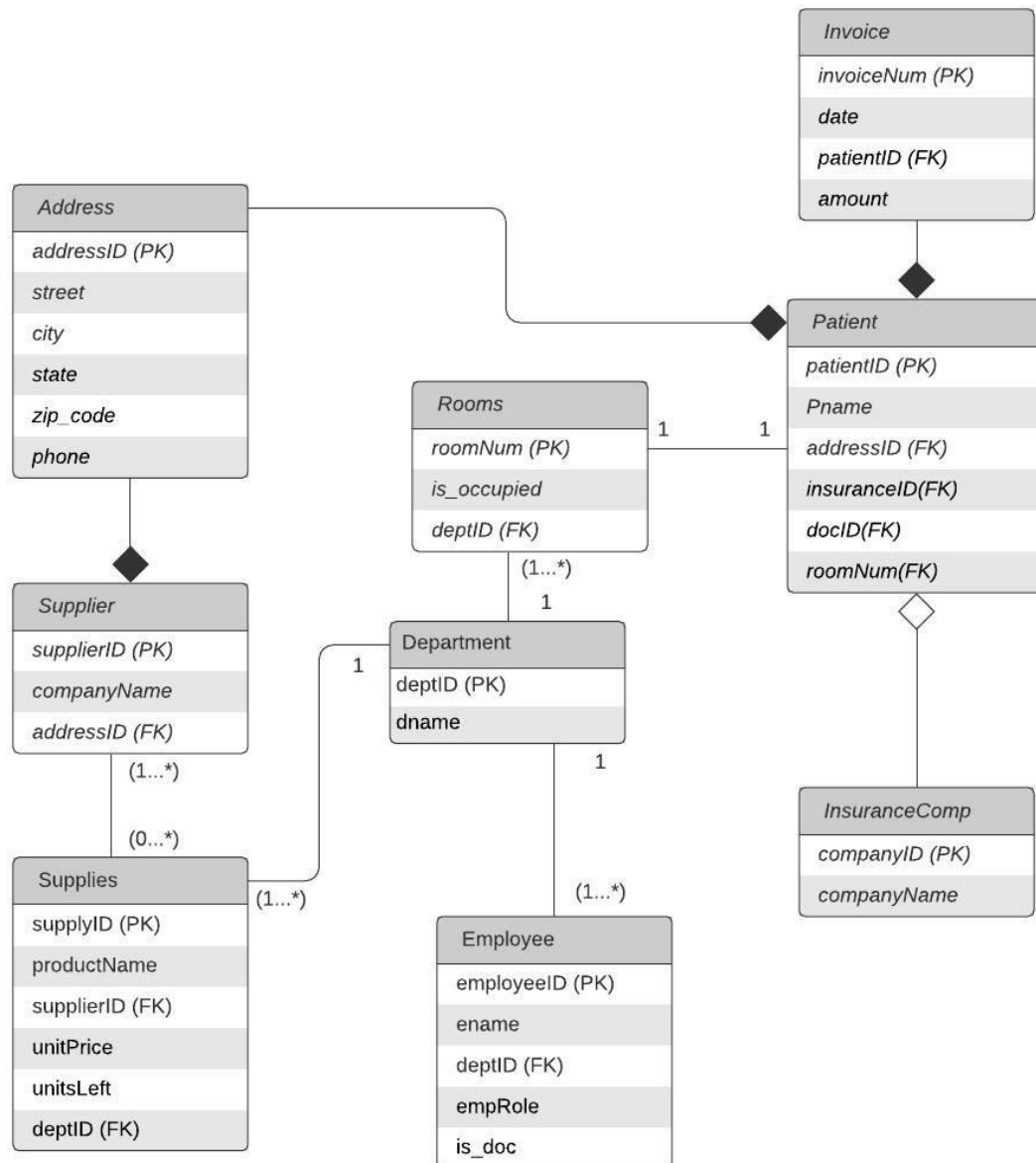
Supplies

Each supply has a supply ID number as a primary key, their name, supplier ID number as its foreign key, the units available, and the cost per unit. A trigger is implemented to add more items to the inventory when they fall below 100 units.

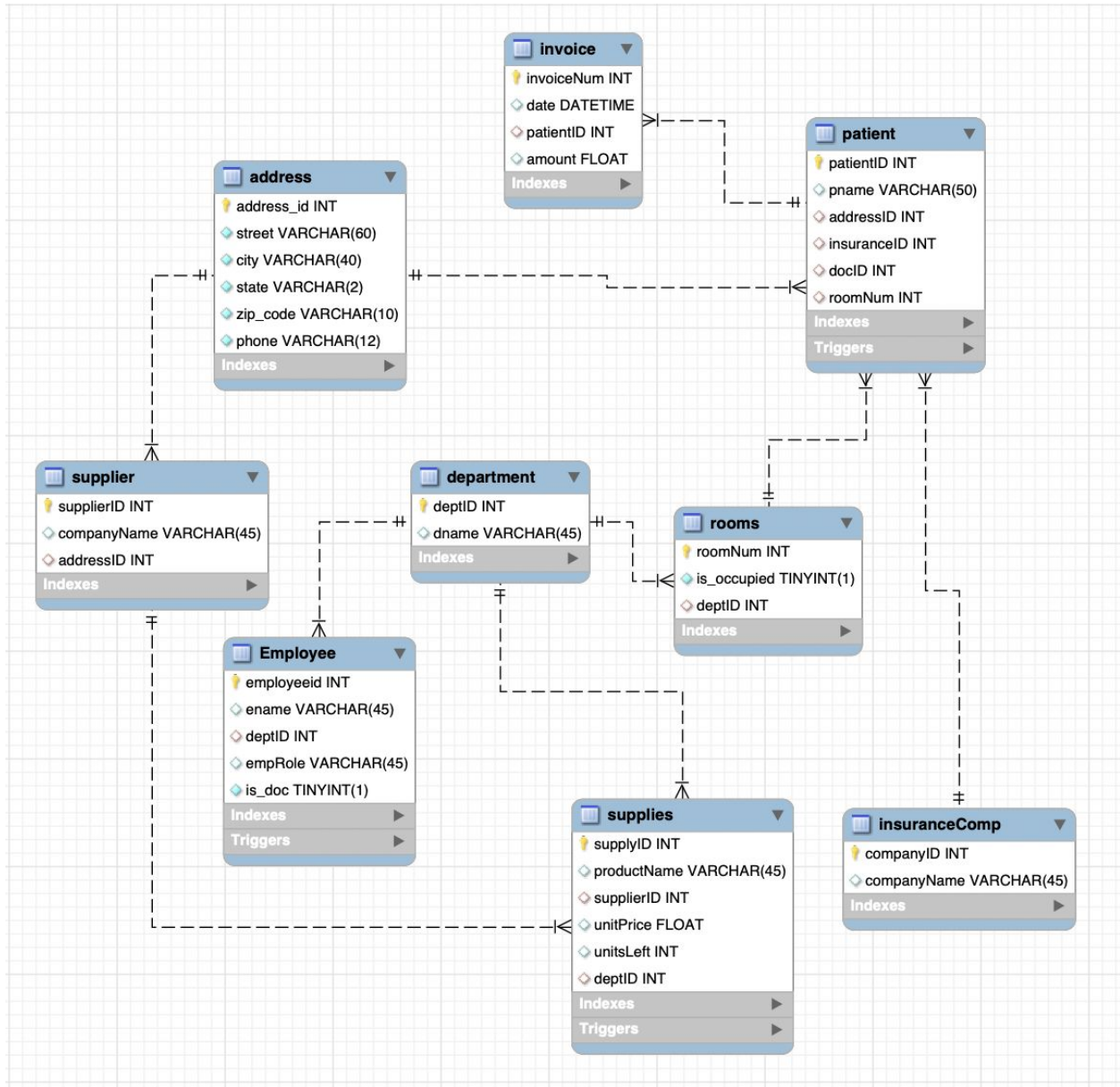
Suppliers

The supplier table is used to have an up to date list of where items are purchased from. They hold the ID number of the supplier as the primary key, the name of the supplier, and the address ID as the foreign key. The address and supplier tables have a one-to-one relationship and help to reference contact information for the supplier when new orders for items need to be placed.

UML Diagram



ER Diagram



Description

The database is designed with 9 tables which are used to retrieve data. There are relationships between the tables to functionally allow for selecting, updating, inserting, or deleting specific pieces of data. Foreign keys were used to create these relationships.

Tools Used

- MySQL WorkBench
- filldb.info/dummy to generate dummy data

Tables

- Patients
- Employees
- Addresses
- Insurance Company
- Invoices
- Rooms
- Departments
- Supplies
- Suppliers

Table Contents

Patients columns:

- PatientID INT, which is the primary key
- Pname VARCHAR
- AddressID INT
- InsuranceID INT
- DocID INT
- RoomNum INT

Employees columns:

- EmployeeID INT, which is the primary key
- Ename VARCHAR
- DeptID INT
- EmpRole INT
- Is_doc BOOLEAN

Address columns:

- Address_ID INT, which is the primary key
- Street VARCHAR
- City VARCHAR

- zip_code VARCHAR
- phone VARCHAR

Insurance Company columns:

- CompanyID INT, which is the primary key
- CompanyName VARCHAR

Invoices columns:

- InvoiceNum INT, which is the primary key
- date DATETIME
- patientID INT
- amount FLOAT

Rooms columns:

- RoomNum INT, which is the primary key
- Is_occupied BOOLEAN
- deptID INT, is the foreign key

Departments columns:

- DeptID INT, is the primary key
- Dname VARCHAR

Supplies columns:


- SupplyID INT, which is the primary key
- productName VARCHAR
- SupplierID INT, is a foreign key
- unitPrice FLOAT
- unitsLeft INT
- deptID INT, is a foreign key

Suppliers columns:

- SupplierID INT, which is the primary key
- companyName VARCHAR
- AddressID INT, is a foreign key

Future Revisions

The Supplies table is currently limited on how it can be used other than keeping track of items. An update to the design and implementation of the database would be to track when certain items are used for certain patients and have the total cost of those items to be added to the invoice at the end of the patients stay. The table should also be separated into multiple tables; the first table can include all the basic medical supplies such as



bandages, gauze, syringes, etc, but another table could be used for specific medications. The reason this would help is the way these items are kept track such as doses and milligrams for medicines versus individual units for basic supplies will add more functionality to the system.

Other tables or fields should also be added to include more information about the treatment that the patient receives during the visit. These can include doctors' diagnosis, medicine given, amount of medicine, and length of stay. A new table could accomplish a lot of these requirements and they would give more data points for information on the patients' stay.