Final Project Submission

May 8, 2019

Team Number 3

Conner Houtman, Andy Silva, Catherine Onumajuru, Kyle Jackson, Sainath Surapaneni

Introduction:

The goal of our team is to create a functioning relational database to accommodate hospital information which will be a mixture of real-world and student-generated data. This database will have data for individuals who visit a particular hospital within Maryland, the doctors who treat them, and other relevant information which would aid in the daily operations within the hospital to keep it running. This is something that is extremely vital to not only keeping a hospital operating efficiently but also being able to do so in a timely manner, which is a big concern with healthcare today.

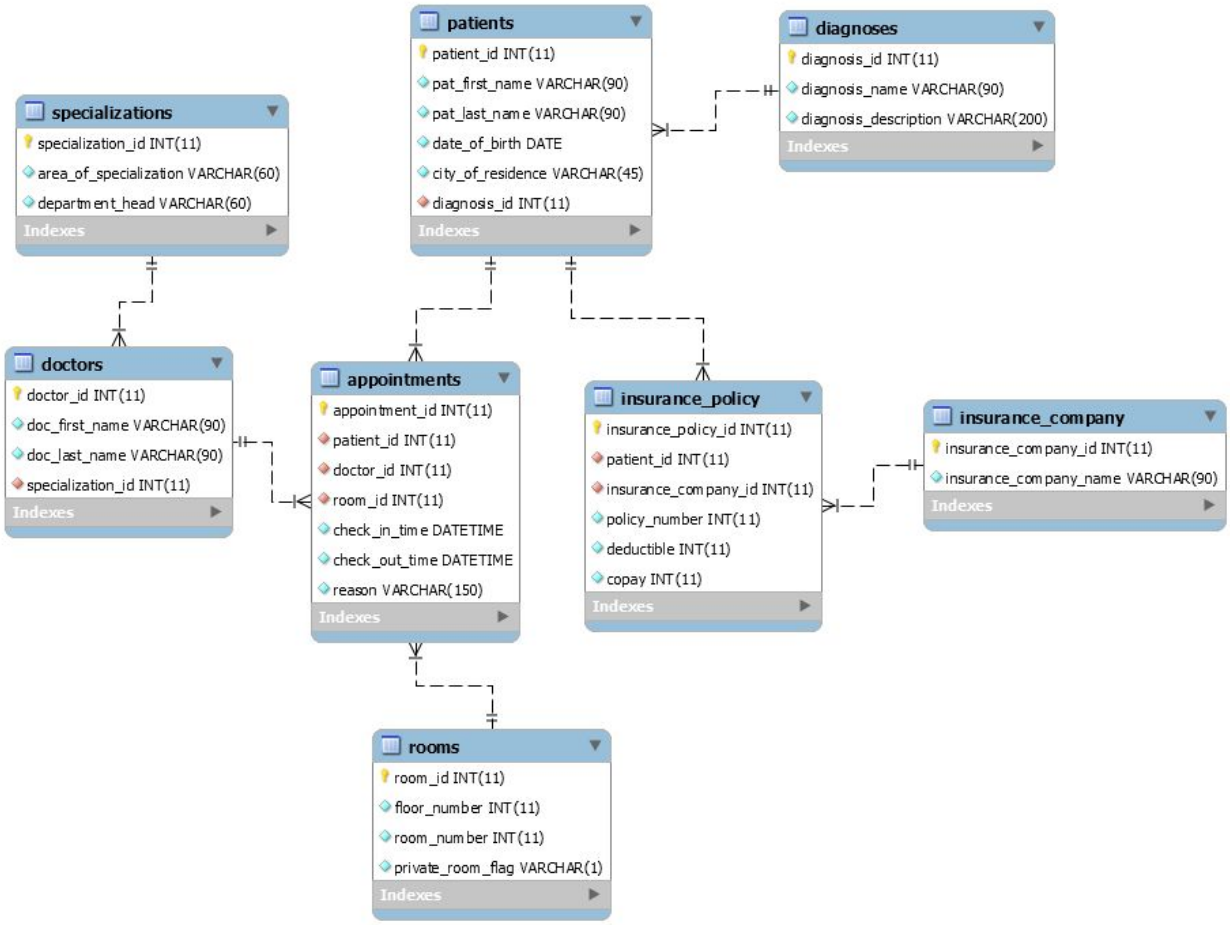
Creating a relational database for a fictional hospital is intriguing to our group because of the complex and typically loaded relationships that will be present. Additionally, patient privacy laws will likely make it difficult to acquire real-world patient information, so we will be generating information for things like names in most cases since we can't use real names of patients and doctors. A well-organized database for a hospital is vital to keeping it running in an organized and efficient manner. A few functions of the database will be tracking patient history in different hospital rooms, the doctors treating different patients, and other useful information to aid in the process of keeping the hospital operating.

Our database will contain the entities: patients, rooms, doctors, insurance company, insurance policy, diagnosis, specializations, and appointments. The number of doctors will range from 10-15 and the number of patients around 30. An important goal for our team in this project

was to limit scope creep and utilize scrum techniques to remain on track for the duration of the project.

Database Description

Logical Design (ERD):



Physical Database:

In creating our physical database, we had to figure out which tables would be core tables, lookup tables, and join tables. Our core tables were patients, doctors, doctors, insurance_policy, rooms. Additionally, appointments was a joining table and specializations, diagnoses, and insurance_company were lookup tables. Patient and doctor names were randomly generated

using an online random generator, while insurance policy information such as policy number was also randomly generated. Information such as deductible as copay was selected from a base that had the most common values for these items.

When inserting our sample data into the database, we started with the tables without foreign keys - specializations, diagnoses, insurance_company, and rooms. Then, we uploaded tables with one foreign key and are connected to those uploaded in the first round - doctors, patients, and insurance_policy. Then the last round of uploading was tricky. We ran into a problem where none of the rows from our appointments CSV were able to get imported into the database, but after but we re-assessed the situation and figured out what was causing it, we managed to upload appointments into our database.

Sample Data:

While gathering the data for our hospital database, we focused on keeping the information as accurate and real as possible, while at the same time not using real people's names or information. When it came to the diagnoses of the patients and specializations of the doctors we researched online common causes for why people would be visiting the hospital. Then allocated different amounts of doctors to each of these depending on the commonality of the conditions of each patient. Within the insurance tables and the rooms/appointments, we created this data rather than using real insurance companies in order to make it more applicable to an actual hospital.

Views/Queries:

There are eight queries in our database as views. Query_1 will show patients treated by doctor Ray Norris or Ella Walker. Query_2 will display insurance companies with the largest deductibles bigger than 1000. Query_3 shows the days in which most patients checked in. In Query_4, the most and least popular insurance companies among patients are found. Query_5 shows a view helping determine if more doctors should be hired in a certain specialization. In Query_6, shows instances where copays are higher than 100. With Query_7, patients with a copay of $50 are displayed. In Query_8, patients treated for general issues are displayed.

View Name	Req. A	Req. B	Req. C	Req. D	Req. E
Query_1	X	X		X	
Query_2	X	X			
Query_3	X	X	X	X	X
Query_4	X	X	X		X
Query_5	X		X	X	
Query_6	X	X		X	
Query_7	X	X		X	
Query_8	X	X		X	

Changes from Original Proposed Design or Scope:

For the most part, we stayed with what we originally planned. However, we did create two changes to the original design taking a table out, and changing data from one table and adding into another table. The table we decided to delete was procedures because it started to get complicated, the focus of the project was to create a database not to be as factually correct as possible. In addition, the check-in and check-out data time data we had in the patient's table was moved to the appointments to simplify the structure of the data. Also, it made more sense to implement that data into appointments then it was in patients. Besides those changes, for the most part, the scope and purpose of our project stayed relatively the same. We encountered small issues with our code but they were simple like syntax errors nothing complicated that we ourselves couldn't figure out.

Lessons Learned:

We would say the most important lesson learned was the value of teamwork and how to work together to fix the issues that we encountered. For example, when we were constructing our database, we would sometimes have issues with our ERD, such as when we were forward engineering it. We would each work closely together to see what exactly the issue was, and as

soon as we identified the issue we would discuss the best method of fixing it. In addition to this, we learned how important communication throughout the team is. One of the main issues we encountered for all of us was a lack of communication, which in turn could slow down how much progress we work making progress. Once we realized this, we all began to respond in a quicker manner, which in turn allowed us to act more effectively as a team.

Strengths and Limitations:

The main strength of our database is how it involves data for the entirety of the hospital population. For example, we have information about not only the patients but also the doctors, which in turn allows to analyze more data and find more trends related to the hospital. In addition, the number of columns in each table allows for us to have an extensive database, thus contributing to the scope of our information. However, our information is somewhat limited by HIPPA laws, which makes it hard to go into detail regarding patient diagnosis information. This does make it hard for us to analyze trends. Another limitation was our team's knowledge of medical information. Although not required, our team still worked extremely hard to make all information as medically accurate as possible.

Additions or Changes for Future Revisions:

For future revisions, the thing that stands out the most as something that we would change is the data we have for the diagnosis of the patient and the specializations of the doctors. We wouldn't have everything that could possibly be wrong with someone, however, we would add more diagnoses as well as specializations so that they would match up more specifically with each other. Also, we would absolutely pay more attention and be more descriptive in naming different columns within our tables.