CS 350

Information and Knowledge Management

Dr. Scott Spetka

Mini-Project

Tucker Mogren

Fall 2017

SUNY Polytechnic Institute

The Idea

For my mini-project was to create a database that could be used to hold information related to doctors, hospitals, patients, and the prescriptions that were prescribed to the patient by the doctors. The database is to then to be accessed via a Python program that will make additions and retrieval of the data easier and more user friendly.

The Database

To create my database, I wanted to use a service that would be accessible from any computer, not just either dogNet or my local machine. Instead of SSH tunneling in order to get MySQL from the CS servers to work in my program I decided that the next best resource would be to use a third party piece of software.

I decided to use a service called Amazon Web Services or, AWS for short. AWS has a sub-service called AWS-RDS for relational database systems. AWS has a client version of AWS-RDS for MySQL. This would allow for me to create a AWS MySQL instance that will allow for me to connect to this instance from any computer. Once the database was set up, all the tables were added via MySQL Workbench, which is a GUI application used to manipulate database instances and makes the task of adding tables and constraints, etc.. much easier than using a Unix based environment.

The Tables

It was determined that I needed to create four different tables within my database. Hospitals, Doctors, Patients, and prescriptions. The database schema for the tables is seen below.

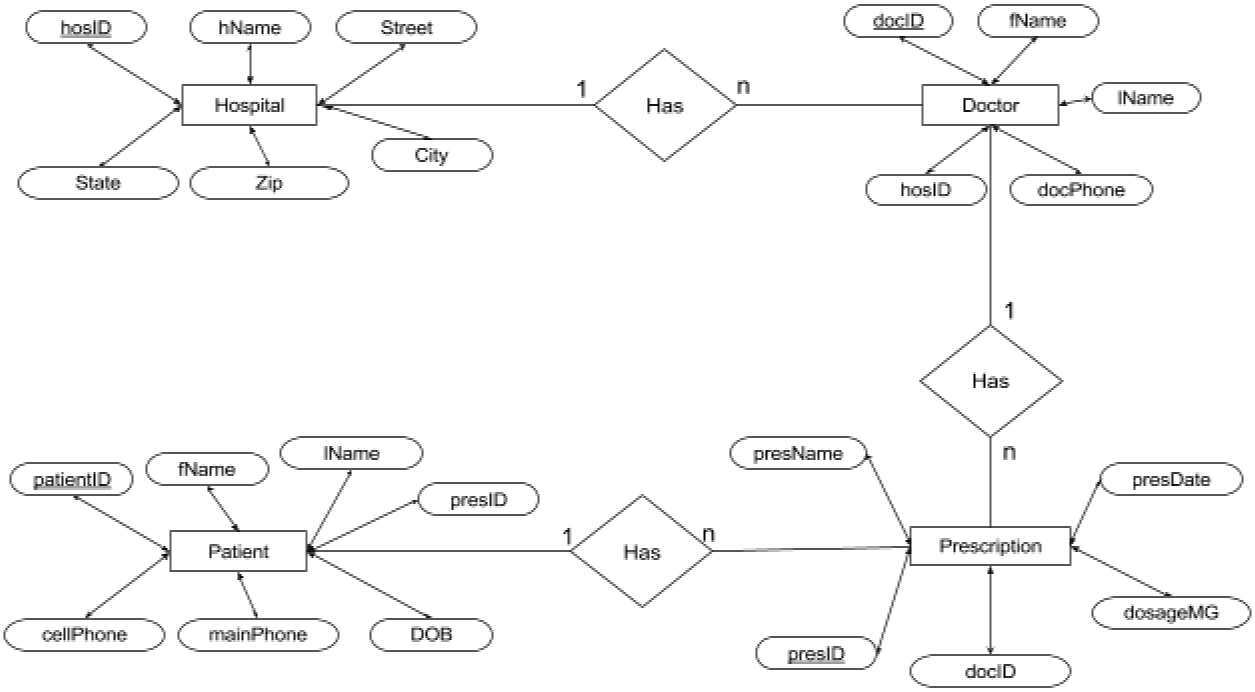


Table Schema

hospital(hosID, hName, Street, City, State, Zip)

Doctor(docID, fName, lName, docPhone, hosID)

Prescription(presID, presName, presDate, dosageMG, docID)

Patient(patientID, fName, lName, DOB, cellPhone, mainPhone, presID)

The Program

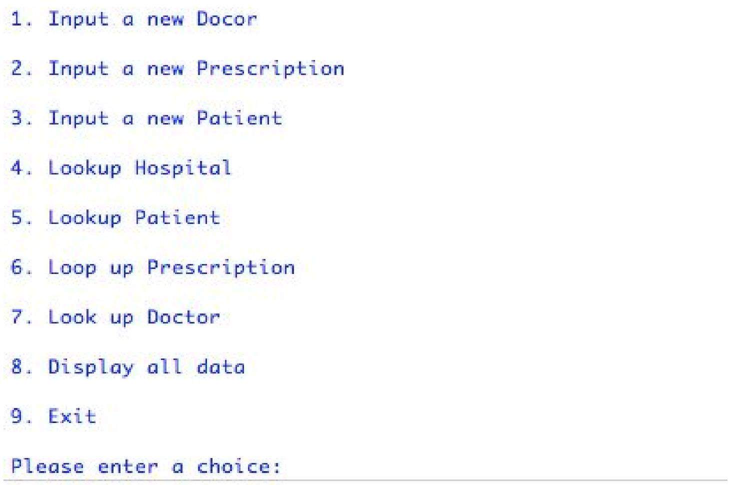
I decided to use the programming language Python in order to code my program. The reason behind this was due to Python's ease of use. Downloading the library in order to connect to a MySQL database was quite easy, and the needed files to accomplish that are located in my GitHub repository. The program starts by promoting the user to enter in the database password[1]. This is so that the database password is not hard coded into the program. I did not want to have it hardcoded into the program because this code is available on GitHub as a public repository and I did not want unauthorized access to my database because the connection string is visible in the code. In the event the password is entered incorrect, the program displays a message that reads “Incorrect Password” and then the program quits. This was set up this way so that if the user does not have access to the database password, they can not access the main menu of the program.

Next, the main menu of the program is available[2] and here the user has multiple different options. Options 1-3 allow for the addition of information into the database. This is accomplished via numerous prompts to the user were the needed information is inputted, then all the data is pushed back to the database/table via an INSERT INTO statement through Python sent to the MySQL server. Options 4-7 allow for data to be retrieved from the database. This is accomplished by the Python program asking the user for a piece of uniquely identifying information, this piece of information had to be unique enough so it wouldn't produce duplicate results but had to be known by the user. Once the piece of information is gathered from the user, it is used within a SELECT statement, passed back to the database, and the information is then send back to the Python program and then formated in a way that is user friendly. Option 8 on the main menu lets the user see all the data that is in the Database, this is done through the use of a JOIN statement, connecting all the Primary/Foreign keys together. This WILL NOT result in a cartesian product of all the tables. Again, all the results are formatted nicely within Python so the output is easily read by the user. The last option, option 9, exits the program gracefully.

The Source Code

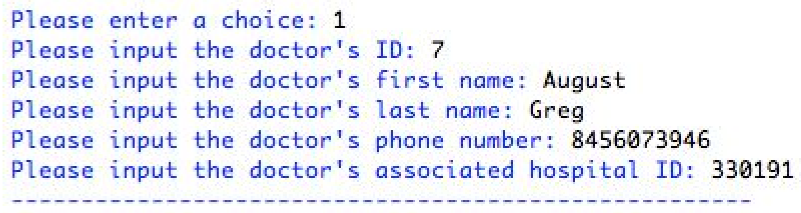
See https://web.cs.sunyit.edu/~mogrent/CS350/Mini_project.html Python Code 1 and Python Code 2 also see hospitals.csv for information related to where the data came from for the hospitals table.

Screenshots



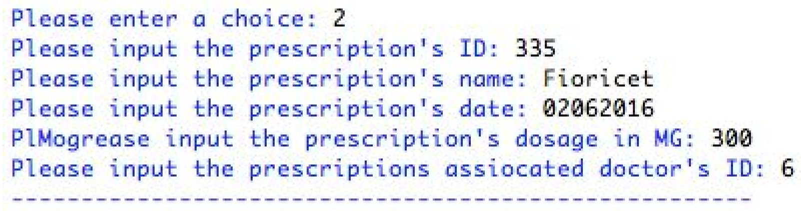
```
1. Input a new Docor

2. Input a new Prescription

3. Input a new Patient

4. Lookup Hospital

5. Lookup Patient

6. Loop up Prescription

7. Look up Doctor

8. Display all data

9. Exit

Please enter a choice:
```

This is a screenshot of the main menu, this screen appears after the database password is entered correctly.



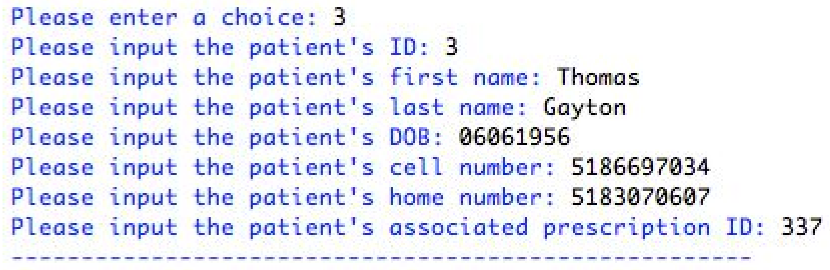
```
Please enter a choice: 1
Please input the doctor's ID: 7
Please input the doctor's first name: August
Please input the doctor's last name: Greg
Please input the doctor's phone number: 8456073946
Please input the doctor's associated hospital ID: 330191
----------------------------------------------------
```

Screenshot shows Data about a doctor being entered in the database from inside the python application.



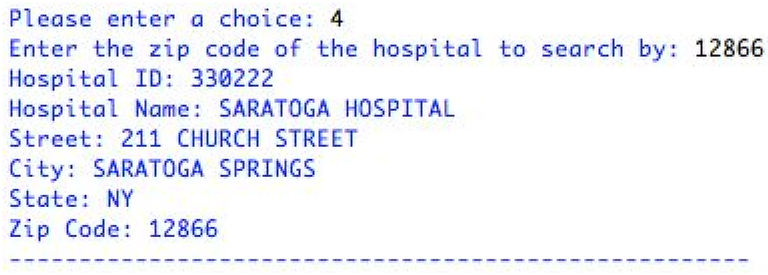
```
Please enter a choice: 2
Please input the prescription's ID: 335
Please input the prescription's name: Fioricet
Please input the prescription's date: 02062016
PlMogrease input the prescription's dosage in MG: 300
Please input the prescriptions assiocated doctor's ID: 6
----------------------------------------------------
```

Screenshot shows Data about a prescription being entered in the database from inside the python application.



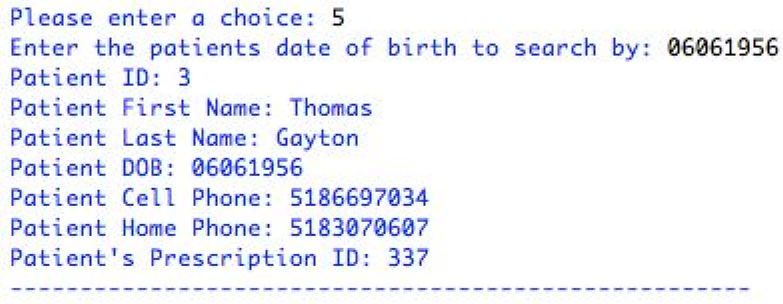
```
Please enter a choice: 3
Please input the patient's ID: 3
Please input the patient's first name: Thomas
Please input the patient's last name: Gayton
Please input the patient's DOB: 06061956
Please input the patient's cell number: 5186697034
Please input the patient's home number: 5183070607
Please input the patient's associated prescription ID: 337
----------------------------------------------------
```

Screenshot shows Data about a patient being entered in the database from inside the python application.



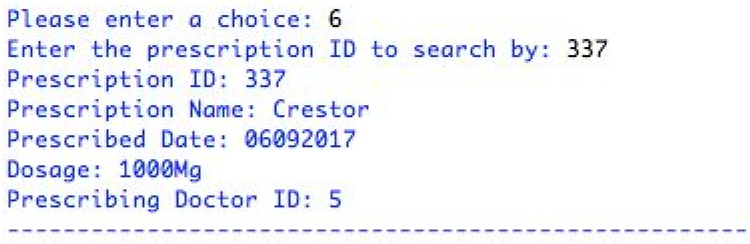
```
Please enter a choice: 4
Enter the zip code of the hospital to search by: 12866
Hospital ID: 330222
Hospital Name: SARATOGA HOSPITAL
Street: 211 CHURCH STREET
City: SARATOGA SPRINGS
State: NY
Zip Code: 12866
-----------------------------------------------------
```

Screenshot shows that the database can be searched through from the application. Shows that if the user wants to search for a hospital all they have to do is enter in a zip code and it will return data from the database about the hospital that is located at that zip code location.



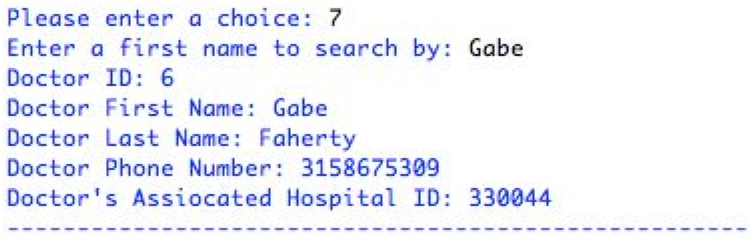
```
Please enter a choice: 5
Enter the patients date of birth to search by: 06061956
Patient ID: 3
Patient First Name: Thomas
Patient Last Name: Gayton
Patient DOB: 06061956
Patient Cell Phone: 5186697034
Patient Home Phone: 5183070607
Patient's Prescription ID: 337
-----------------------------------------------------
```

Screenshot shows that the database can be searched through from the application. Shows that if the user wants to search for a patient all they have to do is enter in a date of birth and it will return data from the database about the patient that has that specific birthday.



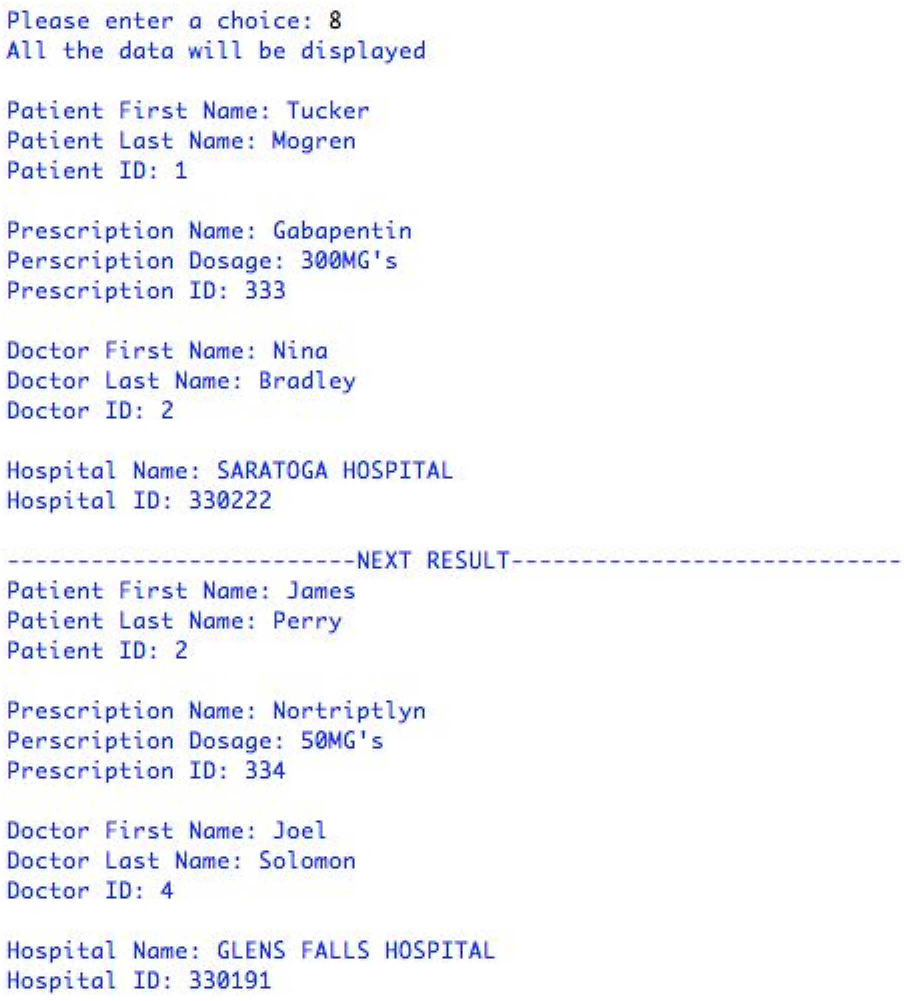
```
Please enter a choice: 6
Enter the prescription ID to search by: 337
Prescription ID: 337
Prescription Name: Crestor
Prescribed Date: 06092017
Dosage: 1000Mg
Prescribing Doctor ID: 5
-----------------------------------------------------
```

Screenshot shows that the database can be searched through from the application. Shows that if the user wants to search for a prescription all they have to do is enter in a prescription ID and it will return data from the database about the prescription that has the prescription ID that was entered.



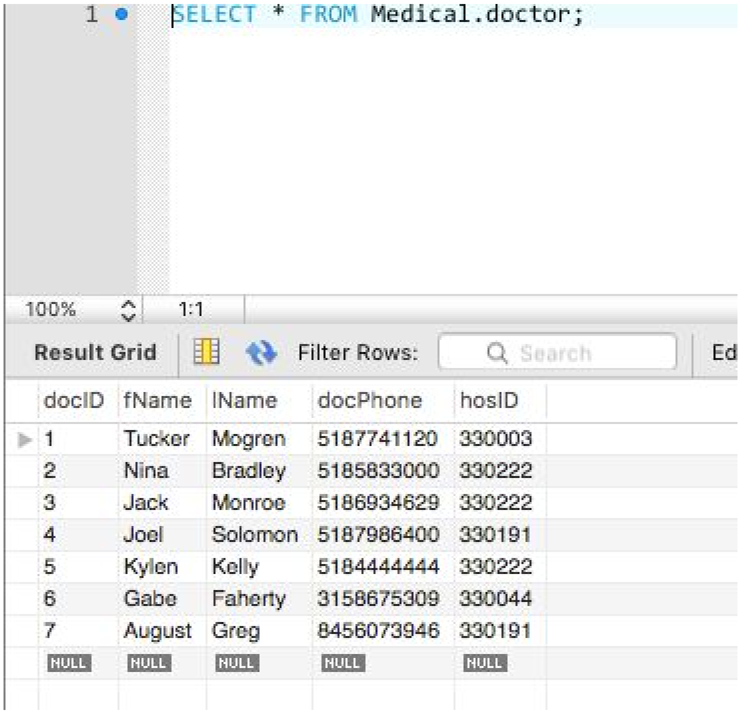
```
Please enter a choice: 7
Enter a first name to search by: Gabe
Doctor ID: 6
Doctor First Name: Gabe
Doctor Last Name: Faherty
Doctor Phone Number: 3158675309
Doctor's Assiocated Hospital ID: 330044
-----------------------------------------------------
```

Screenshot shows that the database can be searched through from the application. Shows that if the user wants to search for a doctor all they have to do is enter the name of the doctor and it will return data from the database about the doctor who has that name.



```
Please enter a choice: 8
All the data will be displayed

Patient First Name: Tucker
Patient Last Name: Mogren
Patient ID: 1

Prescription Name: Gabapentin
Perscription Dosage: 300MG's
Prescription ID: 333

Doctor First Name: Nina
Doctor Last Name: Bradley
Doctor ID: 2

Hospital Name: SARATOGA HOSPITAL
Hospital ID: 330222

-------------------------NEXT RESULT----------------------------
Patient First Name: James
Patient Last Name: Perry
Patient ID: 2

Prescription Name: Nortriptlyn
Perscription Dosage: 50MG's
Prescription ID: 334

Doctor First Name: Joel
Doctor Last Name: Solomon
Doctor ID: 4

Hospital Name: GLENS FALLS HOSPITAL
Hospital ID: 330191
```

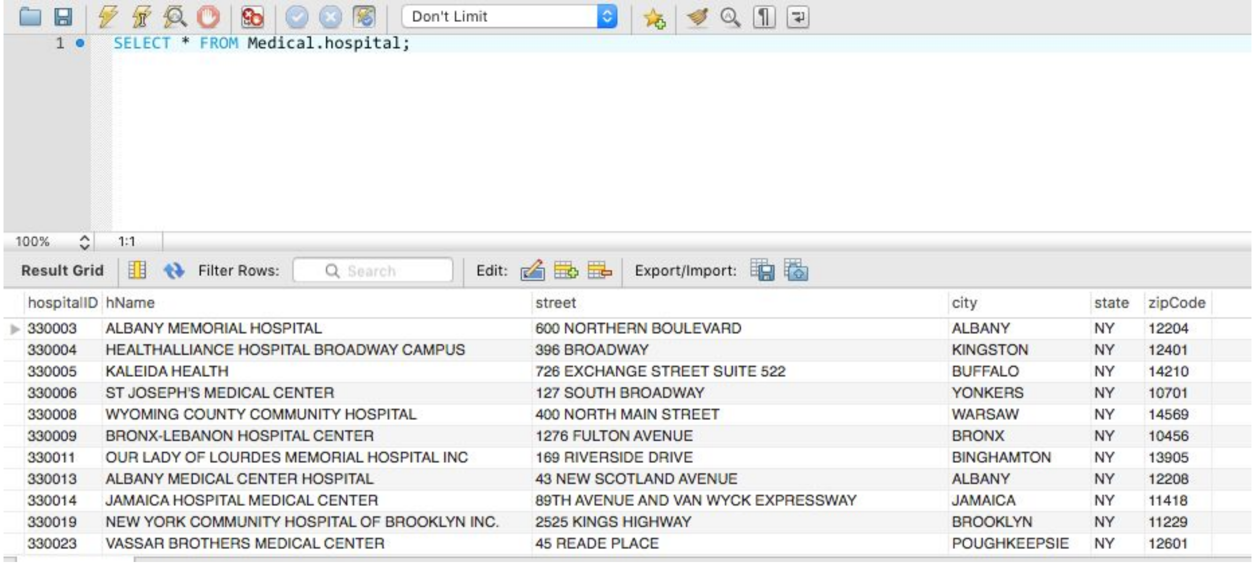
This screenshot above shows that you can display all the data from the database where all the data is joined together and then specific aspects from each table are displayed to the user.



```sql
SELECT * FROM Medical.doctor;
```

docID	fName	lName	docPhone	hosID
1	Tucker	Mogren	5187741120	330003
2	Nina	Bradley	5185833000	330222
3	Jack	Monroe	5186934629	330222
4	Joel	Solomon	5187986400	330191
5	Kylen	Kelly	5184444444	330222
6	Gabe	Faherty	3158675309	330044
7	August	Greg	8456073946	330191
NULL	NULL	NULL	NULL	NULL

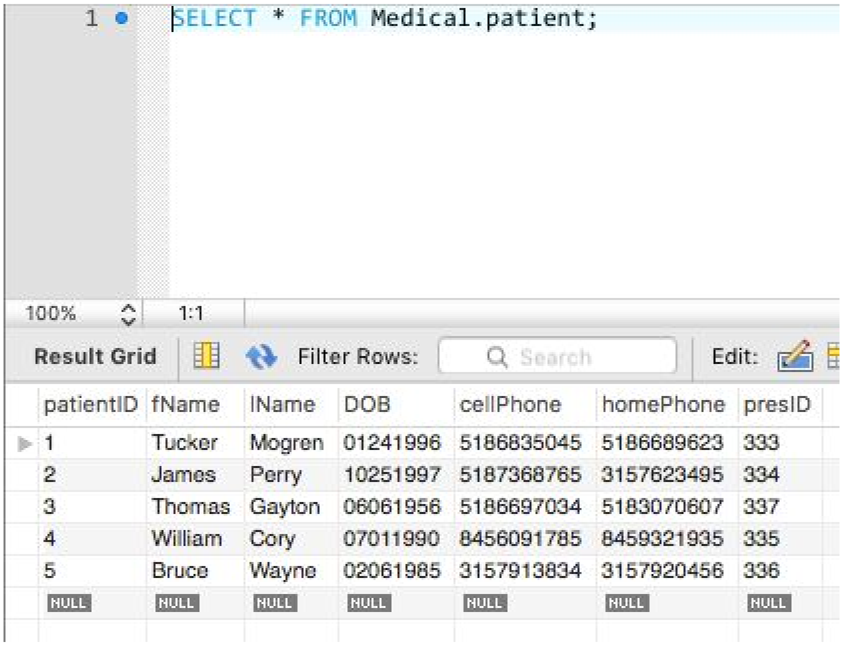
Doctor table, view in MySQL Workbench, shows that data that was entered via the program was pushed to the database correctly.



```sql
SELECT * FROM Medical.hospital;
```

hospitalID	hName	street	city	state	zipCode
330003	ALBANY MEMORIAL HOSPITAL	600 NORTHERN BOULEVARD	ALBANY	NY	12204
330004	HEALTHALLIANCE HOSPITAL BROADWAY CAMPUS	396 BROADWAY	KINGSTON	NY	12401
330005	KALEIDA HEALTH	726 EXCHANGE STREET SUITE 522	BUFFALO	NY	14210
330006	ST JOSEPH'S MEDICAL CENTER	127 SOUTH BROADWAY	YONKERS	NY	10701
330008	WYOMING COUNTY COMMUNITY HOSPITAL	400 NORTH MAIN STREET	WARSAW	NY	14569
330009	BRONX-LEBANON HOSPITAL CENTER	1276 FULTON AVENUE	BRONX	NY	10456
330011	OUR LADY OF LOURDES MEMORIAL HOSPITAL INC	169 RIVERSIDE DRIVE	BINGHAMTON	NY	13905
330013	ALBANY MEDICAL CENTER HOSPITAL	43 NEW SCOTLAND AVENUE	ALBANY	NY	12208
330014	JAMAICA HOSPITAL MEDICAL CENTER	89TH AVENUE AND VAN WYCK EXPRESSWAY	JAMAICA	NY	11418
330019	NEW YORK COMMUNITY HOSPITAL OF BROOKLYN INC.	2525 KINGS HIGHWAY	BROOKLYN	NY	11229
330023	VASSAR BROTHERS MEDICAL CENTER	45 READE PLACE	POUGHKEEPSIE	NY	12601

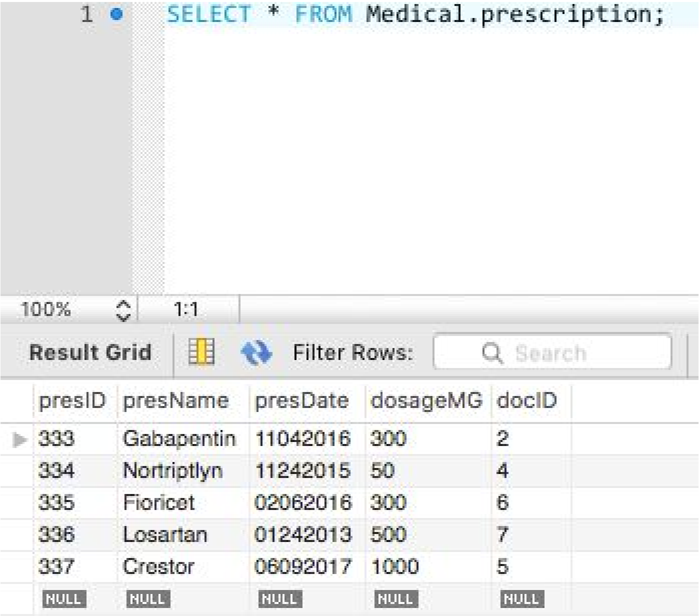
Hospital table, view in MySQL Workbench, this data came from a CSV file that was imported into the database from an online service that has information related to hospitals in New York state.



```sql
SELECT * FROM Medical.patient;
```

patientID	fName	lName	DOB	cellPhone	homePhone	presID
1	Tucker	Mogren	01241996	5186835045	5186689623	333
2	James	Perry	10251997	5187368765	3157623495	334
3	Thomas	Gayton	06061956	5186697034	5183070607	337
4	William	Cory	07011990	8456091785	8459321935	335
5	Bruce	Wayne	02061985	3157913834	3157920456	336
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Patient table, view in MySQL Workbench, shows that data that was entered via the program was pushed to the database correctly.



```sql
SELECT * FROM Medical.prescription;
```

presID	presName	presDate	dosageMG	docID
333	Gabapentin	11042016	300	2
334	Nortriptlyn	11242015	50	4
335	Fioricet	02062016	300	6
336	Losartan	01242013	500	7
337	Crestor	06092017	1000	5
NULL	NULL	NULL	NULL	NULL

Prescription table, view in MySQL Workbench, shows that data that was entered via the program was pushed to the database correctly.