Samantha Sanchez
Professor Yanilda Peralta Ramos
CIS 344
18 December 2023

Final Project Report

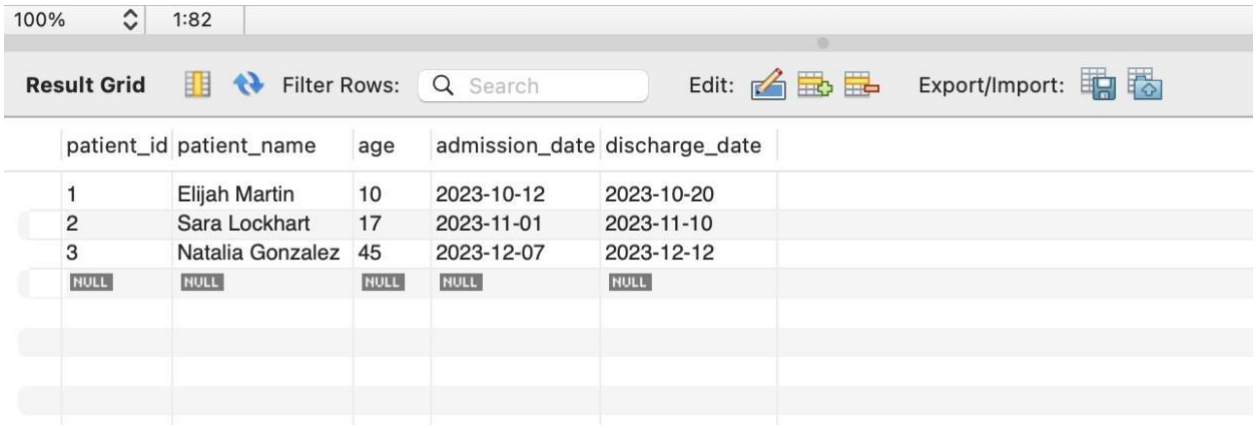
For the final project I started off with completing the MySQL to-do-list. The first step was creating a database named “hospital_portal”.

```
Create database hospital_portal;

Use hospital_portal;
```

The next step was to develop a table named patients that have the following attributes: patient_id (int, not null, unique, auto_increment, primary key), patient_name (varchar(45), not null), age (int, not null), admission_date (date), and discharge_date (date). Another table named Appointments was to be assigned with appointment_id (int, not null, unique, auto_increment, primary key), patient_id (int, not null), doctor_id (int, not null), appointment_date (date, not null), appointment_time (decimal, not null).

```
5  Create table Patients (
6  patient_id int not null unique auto_increment primary key,
7  patient_name varchar(45) not null,
8  age int not null,
9  admission_date date,
10 discharge_date date
11 );
12
13 Create table Appointments (
14 appointment_id int not null unique auto_increment primary key,
15 patient_id int not null,
16 doctor_id int not null,
17 appointment_date date not null,
18 appointment_time decimal not null,
19 Foreign key (patient_id) references Patients(patient_id)
20 );
21
```

I inserted values into the patients table (3 entries) using the “insert into” MySQL command statement. This allowed me to input data into the columns/and rows within the table.



patient_id	patient_name	age	admission_date	discharge_date
1	Elijah Martin	10	2023-10-12	2023-10-20
2	Sara Lockhart	17	2023-11-01	2023-11-10
3	Natalia Gonzalez	45	2023-12-07	2023-12-12
NULL	NULL	NULL	NULL	NULL

After that I created two stored procedures. One served the purpose of scheduling appointments and the other for discharging patients.

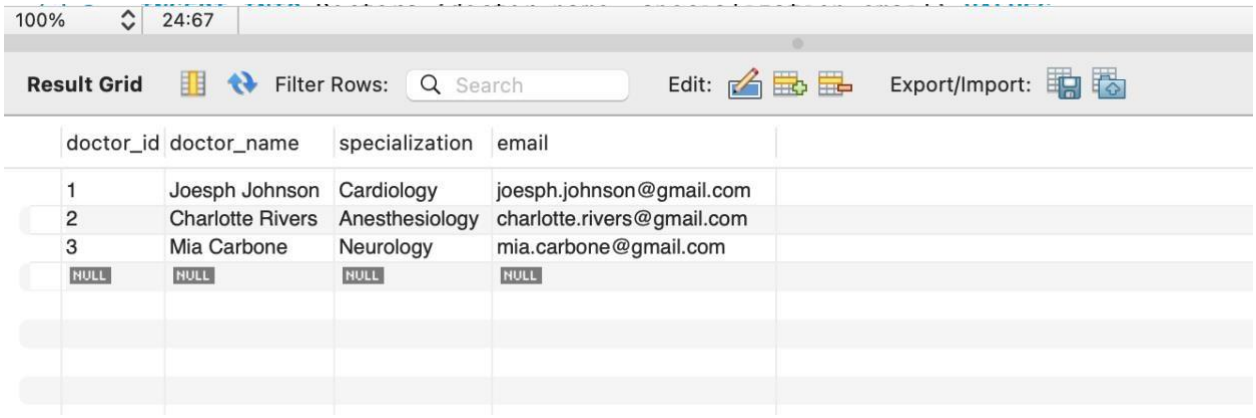
```
DELIMITER //

Create procedure ScheduleAppointment(
    IN patient_id INT,
    IN doctor_id INT,
    IN appointment_date DATE
)
BEGIN
   INSERT INTO appointments (patient_id, doctor_id, appointment_date)
   VALUES (patient_id, doctor_id, appointment_date);
END //


CREATE PROCEDURE DischargePatient(
    IN patient_id INT
)
BEGIN

    UPDATE patients
    SET patient_discharge_status = 'Discharged'
    WHERE patient_id = patient_id;

END //

DELIMITER ;
```

Stored procedures are useful because the code can be reused as often as one would like. Delimiters help with organizing data. They are very important and necessary when creating stored procedures.

The next step was creating a table named Doctors and inserting values that related to Doctors.

```
INSERT INTO Doctors (doctor_name, specialization,email) VALUES
('Joesph Johnson', 'Cardiology', 'joesph.johnson@gmail.com'),
('Charlotte Rivers', 'Anesthesiology', 'charlotte.rivers@gmail.com'),
('Mia Carbone', 'Neurology', 'mia.carbone@gmail.com');
```

Using Select * from Doctors; displayed the following data:



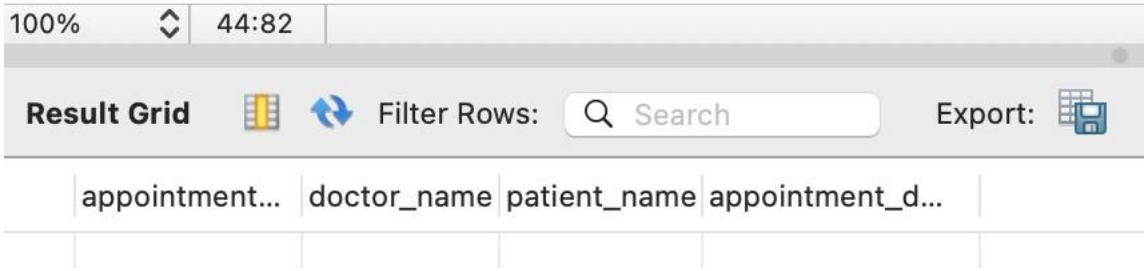
doctor_id	doctor_name	specialization	email
1	Joesph Johnson	Cardiology	joesph.johnson@gmail.com
2	Charlotte Rivers	Anesthesiology	charlotte.rivers@gmail.com
3	Mia Carbone	Neurology	mia.carbone@gmail.com
NULL	NULL	NULL	NULL

The last step of the to-do-list was creating a view that joined the three tables created (Doctors, Appointments, and Patients).

```
69 •    CREATE VIEW DoctorAppointmentPatientView AS
70      SELECT
71          a.appointment_id,
72          d.doctor_name,
73          p.patient_name,
74          a.appointment_date
75      FROM
76          appointments a
77      JOIN
78          doctors d ON a.doctor_id = d.doctor_id
79      JOIN
80          patients p ON a.patient_id = p.patient_id;
```

The join clause is also very useful in SQL because it combines records from tables in databases.
View is said to be a virtual table and does not have/hold data.

```
81
82 •   SELECT * FROM DoctorAppointmentPatientView;
```

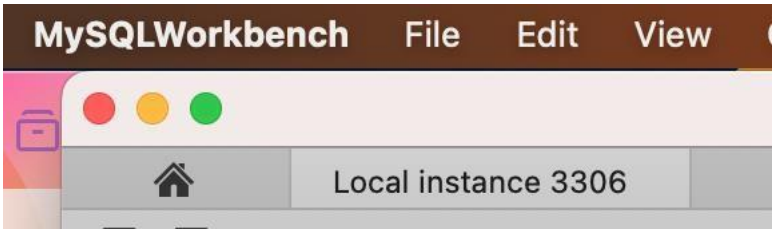


After MySQL, was the integration of Python and MySQL

I knew I had to download MySQL connector
This would have been the command ❼ pip install mysql-connector-python But
I had trouble trying to get it to download.

I know the next step after it would have download was to open one of the starter codes called portalDatabase.py and add MySQL server credentials.

```
import mysql.connector
from mysql.connector import Error

class Database():
    def __init__(self,
                 host="localhost",
                 port="3306",
                 database="hospital_portal",
                 user='root',
                 password='BTS4l7fe'):
```




IDLE Shell 3.12.1
Python 3.12.1 (v3.12.1:2305ca5144, Dec 7 2023, 17:23:39) [Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: /Users/samanthaa.sanchez/Downloads/portalDatabase.py ========
Traceback (most recent call last):
File "/Users/samanthaa.sanchez/Downloads/portalDatabase.py", line 1, in <module>
import mysql.connector
ModuleNotFoundError: No module named 'mysql'
>>>