HOSPITAL DATABASE MANAGEMENT SYSTEM (HDMS)

Stephen Fabeyo

EASTERN UNIVERSITY : M.S. Data Science (Applied Capstone Project)

April 2022

Table of Contents

Contents

Introduction

The Hospital Database Management System (HDMS) is a database system specifically intended for Family Doctors (Primary physicians) to facilitate data entry for hospital or clinical tasks. The system would eliminate all manual data entry while also offering a suitable structure for workflow efficiency, storing, maintaining, and tracking critical health statistics such as labs, finances, health records, bills and payments, patient identity, and so on.

Project Objective

The purpose of this project is to design and develop a simple database system that will meet the needs of a simple Hospital Database Management System (HDMS). This project's goal was to be finished in seven weeks, and it was divided into the following sections:

1. Data Description and Functional Requirements: This section describes what information will be saved in the database and how it will be stored, as well as the relationship between different types of information and any associated business rules that will be applied.
2. The ER model: this section deals with the ER diagram architecture, which will mirror the specifications specified in the prior section to contain all of the ER diagram's components.
3. Relational Schema: This will be the database's final blueprint, representing the data to be inputted and specifying how that data is structured in tables and relationships. The schema explains the relationship between the tables. Finally, this phase will convert the ER diagram into a collection of related schemas.
4. SQL Scripts: This section is divided into subsections and will cover the construction of SQL statements that generate the relationships defined in the schema diagram as well as change the data stored in the tables. Sql statements include SQL DDL, SQL DML Queries, Functions/Triggers/Stored Procedures, and other sql queries as requested for the project.
5. Application Code: This part will include the web application code that interacts with the database. The purpose of this code is to create a simple web page for the patient table. It is written in Python and HTML and will do the following:
    - Insert some data into the database's patient table and display it to user b.
    - Delete any data from the database's patient table.
    - Search for records in the patient table of the database.

Data Specification and Description

The information regarding various clinical functions will be contained in tables, which are linked to all other tables. There will be eight entities, each with its own set of traits. Each entity will have its own primary keys, restrictions, and associated relationships with other related entities. Fake data will be used to populate the tables; each table will include at least 20 rows of information, with the exception of the hospital and room tables, which will have just 5 and 10 rows, respectively. The entities that are considered in the design of this system include the following:

1. Hospital: The Hospital Table will be used to store hospital information.
2. Doctor: This table is used to store and manage information about the doctors.
3. Patient : This patient table is used to record and manage data for each patient in the hospital.
4. Lab_Test: This table stores and manages data about doctor-recommended lab tests..
5. Appointment: This table stores and manages doctor appointments for each patient.
6. Recommendation: This table is used to manage and record information about the patient's diagnoses and recommendations when the lab test is done.
7. Room: This table records the hospitalised patients and their allotted rooms. This table only contains information for patients who have been recommended for admission.
8. Bill: This table is used to manage and record information on aftercare expenses resulting from hospital consultations.

Functional Requirement / Business Rules

Based on the proposed design, the following business rules are the functional requirements of the Hospital Database Management System (HDMS):

- All of the doctors' and patients' appointments are held at the same hospital, despite the fact that it has multiple locations. The hospital is identified by the hospital id, location, phone number, and email address.
- Each appointment is between one doctor and one or more patients, depending on the circumstances, and the appointment is characterized by appointment id, date of schedule, date of appointment, doctor id, and patient id.
- All of the doctors work at the same hospital. Each family doctor can be assigned to one or more patients and recommend one to multiple lab tests to their patient(s). Following each test, the doctor makes further suggestions, either recommending hospitalisation or discharging patients. The doctors are identified by a doctor id, first name, last name, gender, email, phone, and hospital id relating to the hospital location.
- Each patient must have appointments at the hospital and be assigned a family doctor. Each patient is given one or more lab tests to determine the severity of their illness, following which they receive diagnoses and recommendations from doctors, and in the case of critical diagnoses, the patients are hospitalized in a room. Finally, each patient will be invoiced for hospital consultation. A patient id, first name, last name, address, phone, email, age, gender, and doctor id are used to identify the patients.
- Each patient can have one or more lab tests completed as directed by their primary care physician. A lab id, test description, lab date, doctor id, and patient id are used to identify lab tests.

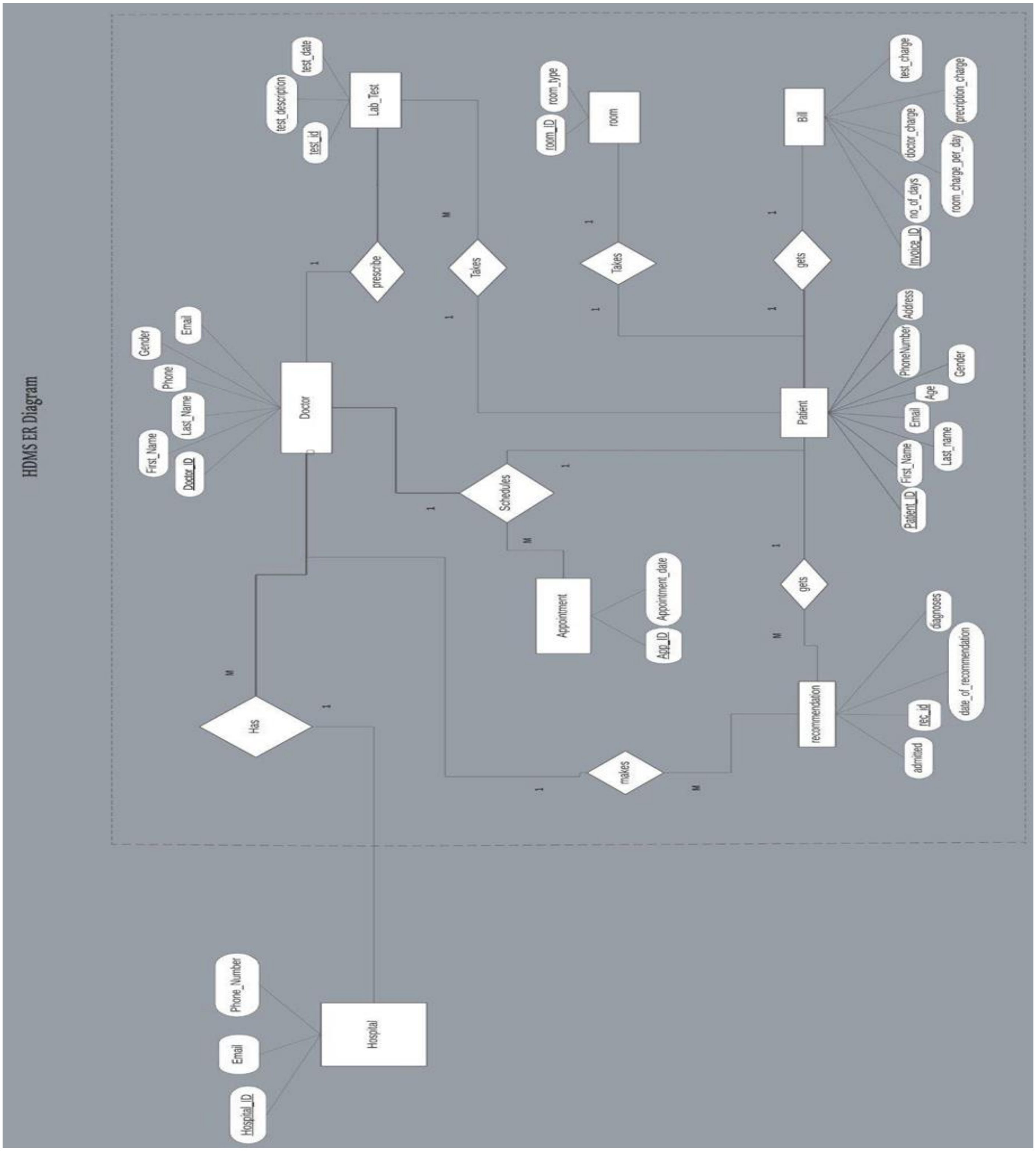
- If a patient is admitted to the hospital following a diagnosis, he or she can be assigned a room. There are numerous room options for the patient, and each one is identified by a room id, room type, and patient id.
- Finally, each patient incurs aftercare costs as a result of their hospital consultations. Each patient's bill is recognised by an invoice id, room charge per day, number of days, test charge, doctor charge, prescription charge, and patient id.

Software

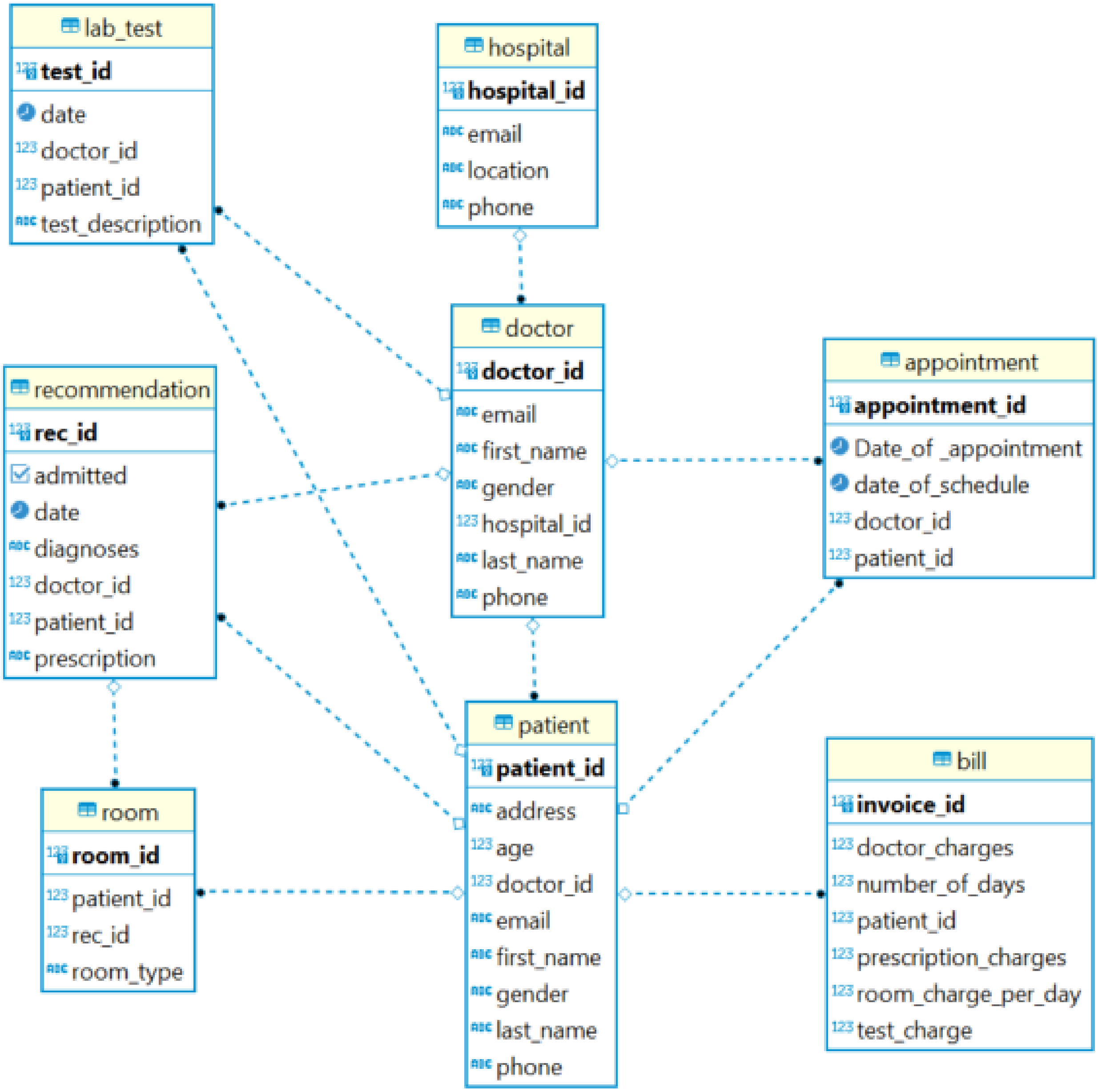
This project made use of the following software, tools, and languages:

1. Postgres Database (pgAdmin 4): the database is Postgres-based, and PGAdmin 4 was utilised as the RDBMS.
2. SQL: The Postgresql SQL variant was used to create the DDL and DML queries, as well as the stored procedures, triggers, functions, and other additional queries for this project.
3. HTML: This code is used to organise a simple web page and its content in accordance with the needs of the hospital database management system.
4. Flask & Python: The HDMS database is accessed using the web application with the power of Flask. Flask is a Python web framework that provides useful tools and features for developing web applications in the Python language. For this project, the web application will connect to a database server, insert data into the patients table, delete data from the patients table, and search for data in the patients' table.
5. DBeaver: This is a free, open source multi platform database management software that was used for the forward (ER to Database sql) and reverse (tables to ER) ER diagrams for this project.
6. Lucid Chart: This is a web-based platform that is used to create the ERD.
7. Pyscopg2: The most widely used PostgreSQL database adapter for the Python programming language is pyscopg2. Psycopg is used in the web app to convert a result of that data type to a Python list automatically.
8. Microsoft Office & Zoom: The functional requirements and other project paperwork would be prepared in Microsoft Word, while the video presentation would be created in Zoom.

ERD Diagram of the HDMS



Relation Schema of the HDMS



DDL Script for the HDMS

```
---Creating the Database HDMS
   CREATE DATABASE "HDMS";

------creating the table definitions

----Hospital Table
   Drop table if Exists Hospital;
   CREATE TABLE Hospital (
   hospital_id Integer NOT NULL GENERATED ALWAYS AS IDENTITY,
   location varchar(12),
   phone varchar(10) NOT NULL,
   email varchar(30) NOT NULL,
   CONSTRAINT hospital_pk PRIMARY KEY (hospital_id) );

----Doctor Table
   Drop table if Exists Doctor;
   CREATE TABLE doctor (
   doctor_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
   first_name varchar(20) NOT NULL,
   last_name varchar(20) NOT NULL,
   gender varchar(10) NOT NULL,
   email varchar(35) NOT NULL,
   phone varchar(10) NOT NULL,
   hospital_id integer NOT NULL,
   CONSTRAINT doctor_pk PRIMARY KEY (doctor_id),
   CONSTRAINT doctor_fk FOREIGN KEY (hospital_id)
   REFERENCES hospital(hospital_id)
   ON DELETE CASCADE
   ON UPDATE CASCADE  );
```

```
----patient Table
  Drop table if Exists patient;
  CREATE TABLE patient (
  patient_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
  first_name varchar(15) NOT NULL,
  last_name varchar(15) NOT NULL,
  address varchar(40) NOT NULL,
  phone varchar(10) NOT NULL,
  email varchar(35) NOT NULL,
  age smallint NOT NULL,
  gender varchar(10) NULL,
  doctor_id integer NOT NULL,
  CONSTRAINT patient_pk PRIMARY KEY (patient_id),
  CONSTRAINT patient_fk FOREIGN KEY (doctor_id)
  REFERENCES doctor(doctor_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE  );

----Lab_test Table
  Drop table if Exists Lab_test;
  CREATE TABLE lab_test (
  test_id int NOT NULL GENERATED ALWAYS AS IDENTITY,
  test_description varchar(100) NOT NULL,
  "date" date NOT NULL,
  doctor_id int NOT NULL,
  patient_id int NOT NULL,
  CONSTRAINT lab_test_pk PRIMARY KEY (test_id),
  CONSTRAINT lab_test_fk FOREIGN KEY (doctor_id)
  REFERENCES doctor(doctor_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT lab_test_fk_1 FOREIGN KEY (patient_id)
  REFERENCES patient(patient_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE  );

----Appointment Table
  Drop table if Exists Appointment;
  CREATE TABLE appointment (
  appointment_id int NOT NULL GENERATED ALWAYS AS IDENTITY,
  date_of_schedule date NOT NULL,
  "Date_of _appointment" date NOT NULL,
  doctor_id int NOT NULL,
  patient_id int NOT NULL,
  CONSTRAINT appointment_pk PRIMARY KEY (appointment_id),
  CONSTRAINT appointment_fk FOREIGN KEY (doctor_id)
  REFERENCES doctor(doctor_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT appointment_fk_1 FOREIGN KEY (patient_id)
  REFERENCES patient(patient_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE  );

----Recommendation Table
  Drop table if Exists Recommendation;
  CREATE TABLE recommendation (
  rec_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
  diagnoses varchar(100) NULL,
  prescription varchar(100) NOT NULL,
  admitted bool NOT NULL,
  "date" date NOT NULL,
  patient_id integer NOT NULL,
  doctor_id integer NOT NULL,
  CONSTRAINT recommendation_pk PRIMARY KEY (rec_id),
  CONSTRAINT recommendation_fk FOREIGN KEY (doctor_id)
  REFERENCES doctor(doctor_id)
  ON DELETE CASCADE ON
  UPDATE CASCADE,
  CONSTRAINT recommendation_fk_1 FOREIGN KEY (patient_id)
  REFERENCES patient(patient_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE  );

----Room Table
  Drop table if Exists Room;
  CREATE TABLE public.room (
  room_id int NOT NULL GENERATED ALWAYS AS IDENTITY,
  room_type varchar(30) NOT NULL DEFAULT 'Single Bedroom',
  patient_id int NOT NULL,
  rec_id int NOT NULL,
  CONSTRAINT room_pk PRIMARY KEY (room_id),
  CONSTRAINT room_fk FOREIGN KEY (rec_id)
  REFERENCES public.recommendation(rec_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT room_fk_1 FOREIGN KEY (patient_id)
  REFERENCES patient(patient_id)
  ON DELETE CASCADE
  ON UPDATE CASCADE  );
```

```sql
----Bill Table
    Drop table if Exists Bill;
    CREATE TABLE public.bill (
    invoice_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
    room_charge_per_day numeric(6,2) NOT NULL DEFAULT 0,
    number_of_days integer NOT NULL DEFAULT 0,
    doctor_charges numeric(6,2) NOT NULL,
    test_charge numeric(6,2) NOT NULL DEFAULT 0,
    prescription_charges numeric(6,2) NOT NULL DEFAULT 0,
    patient_id integer NOT NULL,
    CONSTRAINT bill_pk PRIMARY KEY (invoice_id),
    CONSTRAINT bill_fk FOREIGN KEY (patient_id)
    REFERENCES patient(patient_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE  );
```

INSERT Scripts for the HDMS

```
--- Hospital
INSERT INTO hospital ("location",phone,email)
   VALUES ('East','268197056','east@tclinic.com');
INSERT INTO hospital ("location",phone,email)
   VALUES ('West','268197057','west@tclinic.com');
INSERT INTO hospital ("location",phone,email)
   VALUES ('North','268197058','north@tclinic.com');
INSERT INTO hospital ("location",phone,email)
   VALUES ('South','268197059','south@tclinic.com');
INSERT INTO hospital ("location",phone,email)
   VALUES ('Central','268197060','central@tclinic.com');



----Doctor
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Stephen','Toussaint','Male','stoussaint@tclinic.com','5875559161',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Michael','Adedeji','Male','madedeji@tclinic.com','5875559162',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Zoey','Miller','Female','zmiller@tclinic.com','5875559163',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Susan','Williams','Female','swilliams@tclinic.com','5875559164',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Jack','Laurent','Male','jlaurent@tclinic.com','5875569084',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Abi','Gomez','Female','agomez@tclinic.com','5875569085',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
   VALUES ('Liam','Smith','Male','lsmith@tclinic.com','5875569086',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
```

```
    VALUES ('Alice','Martinez','Female','amartinez@tclinic.com','5875569087',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Bob','Wilson','Male','bwilson@tclinic.com','5875551652',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Abbad','Hassan','Male','ahassan@tclinic.com','5875551653',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Tom','Jones','Male','tjones@tclinic.com','5875551654',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Julia','Anderson','Female','janderson@tclinic.com','5875551655',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Sophia','Pereira','Female','spereira@tclinic.com','5875563386',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Amelia','Taylor','Female','ataylor@tclinic.com','5875563387',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Nathan','Brown','Male','nbrown@tclinic.com','5875563388',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Fouad','Fadel','Male','ffadel@tclinic.com','5875563390',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Daniela','Perez','Female','dperez@tclinic.com','5875557543',5);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Abimbola','Adebisi','Female','aadebisi@tclinic.com','5875557544',5);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Levi','Moore','Male','lmoore@tclinic.com','5875557545',5);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('James','Martin','Male','jmartin@tclinic.com','5875557546',5);


------------Patient
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Olivia','Young','10 Canlish Road, Millwoods, ON','2635149121','Oliviay@yahoomail.com',45,'Female',3);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Anthony','Rivera','1109 Cyrville Road, Orleans, ON','5875167985','riverafamily@live.com',31,'Male',4);
```

```
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Anne','Rivera','1109 Cyrville Road, Orleans, ON','5875167986','riverafamily@live.com',29,'Female',4);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Melody','Rivera','1109 Cyrville Road, Orleans, ON','5875167987','riverafamily@live.com',7,'Female',4);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Joshua','Attah','43 Russel Drive, Alta Vista, ON','2639832139','attahj@rogers.com',60,'Male',1);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Frank','Zucco','984 Amroth Avenue, York, ON','5875198755','zuccof@hotmail.com',45,'Male',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Emma','Zucco','984 Amroth Avenue, York, ON','5875198753','emmazuco321@hotmail.com',43,'Female',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Alex','Zucco','984 Amroth Avenue, York, ON','5875198789','azucco@gmail.com',17,'Male',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Anthony','Zucco','984 Amroth Avenue, York, ON','5875198198','tonyzucco@live.com',14,'Male',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Frank','Rossi','157 Ogilvie Road, Gloucester,
ON','3639008267','frankyRossi2015@yahoomail.com',55,'Male',5);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Jayden','Scott','10, Au Large Boulevard, Orleans,
ON','5814198798','scottjayden@yahoomail.com',23,'Male',7);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Avery','Burton','123 Beacon Hill Drive, Winchester, ON','2635908410','avery4u@rogers.com',29,'Female',6);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Benjamin','Walker','319 Briarscross Boulevard, Kannata,
ON','2635190901','benwalker2010@hotmail.com',20,'Male',9);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Debbie','Christian','1093 Ogilvie Road, Gloucester, ON','2635781122','debbie4u@live.com',34,'Female',8);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Elizabeth','Balogun','12 Bickerton Crescent, Essex,
ON','7175683327','lizzybalogun@live.com',26,'Female',12);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
   VALUES ('Dylan','Robinson','27 Adamson Street, Kannata, ON','6857684128','drobinson@rogers.com',40,'Male',11);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
```

```
   VALUES ('Nathan','Clarke','78 Briarscross Boulevard, Kannata,
ON','7175549090','nathanclarke123@yahoomail.com',44,'Male',15);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Melanie','Pascua','4, Bill Cameron Lane, Barrhaven,
ON','5875990169','pascuafamily@live.com',69,'Female',13);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Emmanuel','Pascua','4, Bill Cameron Lane, Barrhaven,
ON','5814103477','pascuafamily@live.com',75,'Male',13);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Henry','Diaz','59 Augusta Avenue, Gloucester, ON','7175553322','henryd@gmail.com',18,'Male',16);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Binita','Kumar','237 Sussex Drive, York, ON','5814323298','kumarbi@yahoomail.com',21,'Female',14);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Akpan','Bassey','107 Briarscross Boulevard, Kannata,
ON','6856171717','abassey1990@live.com',32,'Male',19);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Anna','Soriano','237 Bridgetown Drive, York, ON','3639002197','anna123@gmail.com',53,'Female',17);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Ben','Jackson','23 Beacon Hill Drive, Winchester, ON','3639091207','benjackson@rogers.com',33,'Male',20);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Ruby','Jackson','23 Beacon Hill Drive, Winchester,
ON','7175171986','rubyjackson@rogers.com',33,'Female',20);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Caleb','Philips','18 Augusta Avenue, Gloucester, ON','3639016545','cphil@rocketmail.com',34,'Male',3);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Ryan','Green','134, Au Large Boulevard, Orleans,
ON','5875961234','ryangreen2000@rogers.com',44,'Male',18);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Andrea','Green','134, Au Large Boulevard, Orleans,
ON','5875964231','andreagreen@rogers.com',42,'Female',18);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Ashley','Green','134, Au Large Boulevard, Orleans,
ON','5875961299','ashleygreen2015@rogers.com',16,'Female',18);

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)

   VALUES ('Amir','Houssein','148 Cyrville Road, Orleans, ON','3639070255','amirh@rocketmail.com',24,'Male',10);
```

```
--------Appointment
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-12','2022-03-03',3,1);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-12','2022-10-03',4,2);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-12','2022-10-03',4,3);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-12','2022-11-03',4,4);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-12','2022-03-03',1,5);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-11','2022-03-02',2,6);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-11','2022-03-02',2,7);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-11','2022-03-02',2,8);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-11','2022-03-02',2,9);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2022-01-01','2021-12-02',5,10);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-02','2021-12-02',7,11);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-02','2022-03-07',6,12);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-12-02','2022-03-07',9,13);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-11-30','2022-03-07',8,14);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-10-12','2022-03-07',12,15);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
```

```
   VALUES ('2021-10-12','2022-03-03',11,16);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-10-12','2021-07-03',15,17);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-01-01','2021-12-02',13,18);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-01-01','2021-12-02',13,19);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-11-12','2022-07-03',16,20);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-11-12','2021-07-03',14,21);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2022-01-01','2022-03-03',19,22);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-09','2021-12-02',17,23);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2022-01-01','2021-12-02',20,24);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2022-01-01','2022-02-27',20,25);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-09','2022-02-27',3,26);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-10','2022-02-27',18,27);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-10','2022-02-27',18,28);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2021-03-10','2022-02-27',18,29);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
   VALUES ('2022-01-01','2022-03-03',10,30);
```

```
--------Lab test table
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Covid PCR Test','2021-03-12',3,1);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('General Routine Check up','2021-03-12',4,2);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('General Routine Check up','2021-03-12',4,3);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('General Routine Check up','2021-03-12',4,4);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood test for Malaria Parasites','2021-03-12',1,5);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Covid Test (PCR)','2021-12-11',2,6);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Covid Test (PCR)','2021-12-11',2,7);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Covid Test (PCR)','2021-12-11',2,8);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Covid Test (PCR)','2021-12-11',2,9);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Salmonella Typhi Test','2022-01-01',5,10);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Dix-Hallpike Test for Vertigo','2021-12-02',7,11);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup','2021-12-02',6,12);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup','2021-12-02',9,13);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup','2021-11-30',8,14);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood count test, measurement of electrolytes and kidney function test','2021-10-12',12,15);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
```

```
   VALUES ('Covid Test (PCR)','2021-10-12',11,16);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Salmonella Typhi Test','2021-10-12',15,17);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood count test, measurement of electrolytes and kidney function test','2021-01-01',13,18);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Dix-Hallpike Test for Vertigo','2021-01-01',13,19);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood test for Malaria Parasites','2021-11-12',16,20);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Covid Test (PCR)','2021-11-12',14,21);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup','2022-01-01',19,22);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood test for Malaria Parasites','2021-03-09',17,23);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood count test, measurement of electrolytes and kidney function test','2022-01-01',20,24);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Blood count test, measurement of electrolytes and kidney function test','2022-01-01',20,25);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Salmonella Typhi Test','2021-03-09',3,26);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup','2021-03-10',18,27);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup','2021-03-10',18,28);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Routine Medical Checkup, Blood Count test','2021-03-10',18,29);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
   VALUES ('Dix-Hallpike Test for Vertigo','2022-01-01',10,30);
```

```
--------Recommendation
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','complication, admission recommended',true,'2021-03-12',1,3);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','None, , Isolation recommended due to close contact',false,'2021-03-12',2,4);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','None, Isolation recommended due to close contact',false,'2021-03-12',3,4);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','complication, admission recommended',true,'2021-03-12',4,4);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Malaria','complication, admission recommended',true,'2021-03-12',5,1);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Positve Covid-19 Test','Isolation and medications recommended',false,'2021-12-11',6,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-11',7,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-11',8,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-11',9,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Typhoid','Admission recommended',true,'2022-01-01',10,5);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test, Vertigo','Isolation, rest and medications recommended',false,'2021-12-02',11,7);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-02',12,6);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','Isolation and medications recommended',true,'2021-12-02',13,9);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Malaria Parasite Detected','Isolation and medications recommended',true,'2021-11-30',14,8);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Inconclusive, further test required','Refer to specialist for further test',false,'2021-10-12',15,12);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
```

```
   VALUES ('Check up Ok','None',false,'2021-10-12',16,11);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Mild Vertig & Typhoid','Admission and treatment',true,'2021-10-12',17,15);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Further test required','Severe complications, admission recommended, further specialist tests',true,'2021-01-01',18,13);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Check up OK','None',false,'2021-01-01',19,13);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Vertigo','Further test required, re-schedule appointment',false,'2021-11-12',20,16);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Check up OK','None',false,'2021-11-12',21,14);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Postive Covid-19 Test','Medications and Isolation',false,'2022-01-01',22,19);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Malaria','admission and treatment',true,'2021-03-09',23,17);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('General Check up OK','None',false,'2022-01-01',24,20);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Malaria Parasite Detected','Medications',false,'2022-01-01',25,20);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Typhoid','Admission & treatment',false,'2021-03-09',26,3);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Postive Covid-19 Test','Isolation and rest recommended',false,'2021-03-10',27,18);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Mild Vertigo','None',false,'2021-03-10',28,18);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Negative Covid-19 Test','None',false,'2021-03-10',29,18);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
   VALUES ('Postive Covid-19 detected','severe complications, Admission',true,'2022-01-01',30,10);
```

```
------room------
INSERT INTO room (patient_id,rec_id)
   VALUES (1,1);
INSERT INTO room (patient_id,rec_id)
   VALUES (4,4);
INSERT INTO room (patient_id,rec_id)
   VALUES (5,5);
INSERT INTO room (patient_id,rec_id)
   VALUES (10,10);
INSERT INTO room (room_type,patient_id,rec_id)
   VALUES ('Family Room',13,13);
INSERT INTO room (patient_id,rec_id)
   VALUES (14,14);
INSERT INTO room (patient_id,rec_id)
   VALUES (17,17);
INSERT INTO room (room_type,patient_id,rec_id)
   VALUES ('Doubles',18,18);
INSERT INTO room (patient_id,rec_id)
   VALUES (23,23);
INSERT INTO room (patient_id,rec_id)
   VALUES (30,30);


------Billing------
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (100,5,1000,100,100,1);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (400,100,2);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (250,100,3);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (100,3,600,100,150,4);
```

```
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (100,14,2500,300,380,5);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (150,100,80,6);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (150,100,7);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (150,100,8);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (100,85,9);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (100,10,1500,500,550,10);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (150,100,50,11);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (150,100,12);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (150,1,200,100,80,13);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (300,150,100,14);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (150,550,15);
INSERT INTO bill (doctor_charges,patient_id)
   VALUES (100,16);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (100,5,450,150,150,17);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
   VALUES (120,30,4500,2000,1000,18);
INSERT INTO bill (doctor_charges,patient_id)
   VALUES (150,19);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
   VALUES (150,125,20);
```

```
INSERT INTO bill (doctor_charges,patient_id)
    VALUES (150,21);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,100,70,22);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,3,250,350,500,23);
INSERT INTO bill (doctor_charges,patient_id)
    VALUES (150,24);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,120,80,25);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,10,800,450,300,26);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,27);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,28);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,29);
INSERT INTO bill (room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,2,300,100,50,30);
```

Functions Scripts for the HDMS

```
----FUNCTIONS
```

- Function 1: The goal of this function is to compute the total room charge for each patient who has been admitted and allotted a room.

```
CREATE OR REPLACE function TOTAL_ROOM_CHARGES(INVOICE_ID INTEGER)
  RETURNS INTEGER
  LANGUAGE PLPGSQL
  AS
  $$
  DECLARE
    ROOM_CHARGE INTEGER;
  BEGIN
     SELECT
     (ROOM_CHARGE_PER_DAY * NUMBER_OF_DAYS) as ROOM_CHARGE_TOTAL
     INTO ROOM_CHARGE
     FROM BILL
     WHERE BILL.INVOICE_ID = TOTAL_ROOM_CHARGES.INVOICE_ID;
     RETURN (ROOM_CHARGE);
  END;
  $$;

  --TESTING FUNCTION

     SELECT TOTAL_ROOM_CHARGES(1) ;
     SELECT TOTAL_ROOM_CHARGES(10) ;
     SELECT TOTAL_ROOM_CHARGES(13) ;
```

```
---FUNCTION 2
```

- Function 2: The goal of this function is to calculate the total hospital bills for each patient.

```
CREATE OR REPLACE function TOTAL_BILL (INVOICE_ID INTEGER)
   returnS INTEGER
   language PLPGSQL
   AS
   $$
  DECLARE
      OUTSTANDING_BALANCE INTEGER;
  BEGIN
       SELECT
       (DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +   (ROOM_CHARGE_PER_DAY *
NUMBER_OF_DAYS))
       INTO OUTSTANDING_BALANCE
       FROM bill
       WHERE BILL.INVOICE_ID = TOTAL_BILL.INVOICE_ID;
       RETURN (OUTSTANDING_BALANCE);
   END;
   $$;

   --TESTING FUNCTION

       SELECT TOTAL_BILL(1) ;
       SELECT TOTAL_BILL(5) ;
       SELECT TOTAL_BILL(10) ;
       SELECT TOTAL_BILL(15) ;
```

```
---------FUNCTION 3
```

- Function 3: This function's objective is to provide a more detailed calculation of each patient's total hospital bills..

```
    CREATE OR REPLACE FUNCTION TOTAL_OUTSTANDING_BILLS(INVOICE_ID integer)
    RETURNS table ( PATIENT_ID INTEGER,
                PATIENT_NAME TEXT,
                TOTAL_BILL numeric(8,2))
     LANGUAGE plpgsql
     AS
     $$
     BEGIN
     RETURN QUERY
         SELECT
         B.PATIENT_ID,
         TRIM(CONCAT(A.FIRST_NAME,' ', A.LAST_NAME))as PATIENT_NAME,
         (DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY *
NUMBER_OF_DAYS)) as TOTAL_BILL
         FROM BILL B
         INNER JOIN PATIENT A USING (PATIENT_ID)
         WHERE B.INVOICE_ID = TOTAL_OUTSTANDING_BILLS.INVOICE_ID;
      END;
      $$;

--TESTING FUNCTION
SELECT TOTAL_OUTSTANDING_BILLS(3);
SELECT TOTAL_OUTSTANDING_BILLS(6);
SELECT TOTAL_OUTSTANDING_BILLS(9);
SELECT TOTAL_OUTSTANDING_BILLS(12);
SELECT TOTAL_OUTSTANDING_BILLS(15);
```

View Definition Scripts for The HDMS

```
----Views
```

View 1: The purpose of this view is to present a more broad overview of each female patient and their assigned doctors, including a calculation of the patient's birth year.

```
CREATE OR REPLACE VIEW Female_patients_doctors
AS
SELECT
CONCAT(D.FIRST_NAME,' ', D.LAST_NAME)as DOCTOR_NAME,
D.GENDER AS DOCTOR_GENDER,
CONCAT(P.FIRST_NAME,' ', P.LAST_NAME)as PATIENT_NAME,
D.EMAIL AS DOCTORS_EMAIL ,
P.EMAIL AS PATIENTS_EMAIL,
P.ADDRESS AS PATIENT_ADDRESS,
EXTRACT (YEAR FROM CURRENT_DATE) - P.AGE AS PATIENTS_YEAR_OF_BIRTH
FROM DOCTOR D
INNER JOIN PATIENT P ON D.DOCTOR_ID=P.DOCTOR_ID
WHERE P.GENDER='Female';
```

Testing View

```
select * from Female_patients_doctors
ORDER BY PATIENTS_YEAR_OF_BIRTH
```

View 2: The purpose of this view is to present a more broad overview of each male patient and their assigned doctors, including a calculation of the patient's birth year.

```
CREATE OR REPLACE VIEW Male_patients_doctors
AS
SELECT
CONCAT(D.FIRST_NAME,' ', D.LAST_NAME)as DOCTOR_NAME,
D.GENDER AS DOCTOR_GENDER,
CONCAT(P.FIRST_NAME,' ', P.LAST_NAME)as PATIENT_NAME,
D.EMAIL AS DOCTORS_EMAIL ,
P.EMAIL AS PATIENTS_EMAIL,
P.ADDRESS AS PATIENT_ADDRESS,
EXTRACT (YEAR FROM CURRENT_DATE) - P.AGE AS PATIENTS_YEAR_OF_BIRTH
FROM DOCTOR D
INNER JOIN PATIENT P ON D.DOCTOR_ID=P.DOCTOR_ID
AND P.GENDER='Male';
```

Testing View

```
select * from Male_patients_doctors
```

Trigger Function & Trigger Scripts for the HDMS

For this trigger, the Patient Audit Logs database is first created in order to audit patient logs. When a new patient data is added to the Patients table, the trigger is set to insert a record into the Patient Audit Logs table.

```
    DROP TABLE IF EXISTS Patient_Audit_Logs;
    CREATE TABLE  Patient_Audit_LOGS (
    Patient_ID INTEGER NOT NULL,
    FIRST_NAME VARCHAR(20) NOT NULL,
    LAST_NAME VARCHAR(20) NOT NULL,
    EMAIL VARCHAR(50) NOT NULL,
    PATIENT_ADDITION_TIME timestamp NOT NULL);

  ---Trigger Function

    CREATE OR REPLACE FUNCTION patient_insert_trigger_fnc()
    RETURNS TRIGGER
    LANGUAGE PLPGSQL
    AS
    $$
      BEGIN
        --trigger logic
     INSERT INTO Patient_Audit_Logs (patient_id, first_name, last_name,email,patient_Addition_time)
VALUES(NEW.patient_id,NEW.first_name,NEW.last_name,NEW.email,current_date);
     RETURN NEW;
      END;
    $$;
```

```
---------Trigger statement

          CREATE TRIGGER patient_insert_trigger
          AFTER INSERT
          ON PATIENT
          FOR EACH ROW

---------Calling Trigger statement

         EXECUTE PROCEDURE patient_insert_trigger_fnc();
```

Stored Procedure Scripts for the HDMS

---Stored Procedure

The stored procedure that was created for this project is called "combinedate." The aim of this stored procedure is to generate a billing summary for each patient ID that includes the patient's full name, email address, and total bill.

```
    CREATE OR REPLACE PROCEDURE  CombineData(PATIENT_ID INTEGER)
    LANGUAGE PLPGSQL
    AS
    $$
      DECLARE NEWDATA RECORD;
      BEGIN
        SELECT
         A.PATIENT_ID,
         A.EMAIL,
         TRIM(CONCAT(A.FIRST_NAME,' ', A.LAST_NAME))as PATIENT_NAME,
         (DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY * NUMBER_OF_DAYS)) as TOTAL_BILL
         INTO NEWDATA
         FROM BILL AS B
         INNER JOIN PATIENT AS A USING (PATIENT_ID)
         WHERE A.PATIENT_ID = CombineData.PATIENT_ID;

      RAISE NOTICE 'Patient Info and Total Bill is %' , NEWDATA;
      END;
    $$;
---------Calling The Stored Procedure
CALL CombineData(10)
CALL CombineData(16)
CALL CombineData(29)
```

Additional SQL DML Queries

```
----DML Scripts
```

```
--1. ----SELECT
```

Except for "single bedroom," the code below selects all other room types.

```
SELECT *
FROM ROOM
WHERE ROOM_TYPE <>'Single Bedroom';
```

```
--2.-----INSERT
```

The codes below populate the doctor and patient tables with data.

```
INSERT INTO DOCTOR(first_name,last_name,gender,email,phone,hospital_id)
 VALUES ('Gabriella','Mendes','Female','gmendes@tclinic.com','512654891',3);

INSERT INTO DOCTOR(first_name,last_name,gender,email,phone,hospital_id)
 VALUES ('Ian','Murphy','Male','imurphy@tclinic.com','522659186',1);

 INSERT INTO PATIENT (first_name,last_name,address,phone,email,age,gender,doctor_id)
 VALUES ('Melissa','Burgh','15 Smyth Road, Innes, ON','6335768113','meb123@gmail.com',25,'Female',21);

 INSERT INTO PATIENT (first_name,last_name,address,phone,email,age,gender,doctor_id)
 VALUES ('Bob','Cameron','10 Coventry Avenue, Innes, ON','6332670943','cameronb@hotmail.com',65,'Male',22);
```

--3. -----UPDATE

The code below modifies the email address in the doctor table for doctor ID 22.

```
UPDATE DOCTOR
SET EMAIL='johnmurphy@tclinic.com',
    FIRST_NAME ='John'
WHERE DOCTOR_ID=22;
```

--4. ------JOINS

The following code joins the doctor and patient tables, displaying only the physicians' and patients' complete names and email addresses.

```
SELECT
TRIM(CONCAT(DC.FIRST_NAME,' ', DC.LAST_NAME))as DOCTOR_NAME,
DC.EMAIL AS DOCTOR_EMAIL,
TRIM(CONCAT(PT.FIRST_NAME,' ', PT.LAST_NAME))as PATIENT_NAME,
PT.EMAIL AS PATIENT_EMAIL
FROM DOCTOR AS DC
INNER JOIN PATIENT AS PT ON DC.DOCTOR_ID=PT.DOCTOR_ID;
```

--5. -----AGGREGATION (COUNT)

The following code below displays the number of patients assigned to each of the doctors.

```
SELECT
DC.DOCTOR_ID,
DC.EMAIL AS DOCTOR_EMAIL,
COUNT(PT.PATIENT_ID) AS NO_OF_PATIENTS
FROM DOCTOR DC
INNER JOIN PATIENT AS PT
ON DC.DOCTOR_ID=PT.DOCTOR_ID
GROUP BY DC.DOCTOR_ID
ORDER BY COUNT(PT.PATIENT_ID) DESC;
```

```
---6. ---DELETE ---
```

The following code deletes the patient id 33 record.

```
DELETE
FROM PATIENT
WHERE PATIENT_ID =33;
```

```
--7 CASE STATEMENT ---
```

The following code used a case statement to identify individuals with Covid-19-related illnesses and labelled the record before sending it to the provincial healthcare system.

```
SELECT
PT.PATIENT_ID,
CONCAT(PT.FIRST_NAME,' ', PT.LAST_NAME)as PATIENT_NAME,
PT.GENDER,
PT.AGE,
PT.EMAIL,
RM.DIAGNOSES,
CASE
WHEN RM.DIAGNOSES LIKE '%Covid%'
THEN 'SEND COVID RECORD TO PROVINCIAL HEALTHCARE'
ELSE 'NOT COVID RELATED: IGNORE'
END  AS COVID19_POLICY
FROM PATIENT AS PT
```

```
    INNER JOIN RECOMMENDATION RM
    ON PT.PATIENT_ID=RM.PATIENT_ID


---8 AGGREGATION (SUM)
```

The following code is being used to calculate the total of each patient's bill.

```
SELECT
P.PATIENT_ID,
CONCAT(P.FIRST_NAME,' ', P.LAST_NAME)as PATIENT_NAME,
SUM(DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY * NUMBER_OF_DAYS)) AS TOTAL_BILL
FROM PATIENT P
INNER JOIN BILL B
ON P.PATIENT_ID=B.PATIENT_ID
GROUP BY P.PATIENT_ID
ORDER BY  TOTAL_BILL
```

Flask & Web Application Scripts for The HDMS

Flask is a Python module and web framework for building lightweight online applications. A Flask-powered web application is used to connect to the HDMS database.. The web application created for this project will connect to a database server, insert data into the patients table, delete data from the patients table, and search the patients table for data for this project. The following steps were first used for installing flask and creating the web application that interacts with the database.

I. Make a directory for the project:

```
mkdir <project name>
```

II. Install Virtual Environment

```
py -2 -m pip install virtualenv
```

III. Create a virtual environment in Python 3 in the project directory:

```
py -3 -m venv <name of environment>
```

IV. Before installing Flask, activate the virtual environment in the project directory.

```
<name of environment>\Scripts\activate
```

V. Install Flask and psycopg2

- pip install Flask
- pip install psycopg2

VI. Create the full app.py python scripts

VII. Create an environment override file .flaskenv to set the Set the web app environment variable. ine the file created, type in the following command;

```
FLASK_ENV=development

FLASK_APP=app.py
```

VIII. Install the .flaskenv file created in step (VI.) with the following command;

```
pip install python-dotenv
```

IX. Create a templates folder in the original project folder that will house the HTML scripts

X. Create the following HTML scripts in the template folder

- index.html
- layout.html
- edit.html

XI. Run the the flask command on the windows command (CMD) line with the syntax “flask run” and navigate to the URL provided on the command line to access the web application

XII. Enter a new patient information, search for patients records or delete patients record

Flask Python Code (app.py)

This is the main python code for the HDMS. The Python app instance is in charge of dealing with incoming http requests and responding to the user.

```
 #app.py
from flask import Flask, render_template, request, redirect, url_for, flash
import psycopg2 #pip install psycopg2
import psycopg2.extras
app = Flask(__name__)
app.secret_key = "cairocoders-ednalan"
DB_HOST =  "localhost"
DB_NAME = "HDMS"
DB_USER = "postgres"
DB_PASS = 12345
conn = psycopg2.connect(dbname=DB_NAME, user=DB_USER, password=DB_PASS, host=DB_HOST)
@app.route('/')
def Index():
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    s = "SELECT * FROM room"
    cur.execute(s) # Execute the SQL
    list_users = cur.fetchall()
    return render_template('index.html', list_users = list_users)
```

```
@app.route('/add_room', methods=['POST'])
def add_room():
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    if request.method == 'POST':
        room_type = request.form['room_type']
        patient_id = request.form['patient_id']
        rec_id = request.form['rec_id']
        cur.execute("INSERT INTO room (room_type, patient_id, rec_id) VALUES (%s,%s,%s)", (room_type, patient_id, rec_id))
        conn.commit()
        flash('Record(s) Added successfully')
        return redirect(url_for('Index'))
@app.route('/edit/<id>', methods = ['POST', 'GET'])
def get_room(room_id):
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    cur.execute('SELECT * FROM room WHERE room_id = %s', (room_id))
    data = cur.fetchall()
    cur.close()
    print(data[0])
    return render_template('edit.html', room = data[0])
```

```
@app.route('/update/<id>', methods=['POST'])
def update_room(room_id):
    if request.method == 'POST':
        room_type = request.form['room_type']
        patient_id = request.form['patient_id']
        rec_id = request.form['rec_id']

        cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
        cur.execute("""
            UPDATE room
            SET room_type = %s,
                patient_id = %s,
                rec_id = %s
            WHERE room_id = %s
        """, (room_type, patient_id, rec_id))
        flash('Record Updated Successfully')
        conn.commit()
        return redirect(url_for('Index'))

@app.route('/delete/<string:id>', methods = ['POST','GET'])
```

```
def delete_room(room_id):

    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)



    cur.execute('DELETE FROM room WHERE id = {0}'.format(room_id))

    conn.commit()

    flash('Record Removed Successfully')

    return redirect(url_for('Index'))



if __name__ == "__main__":

    app.run(debug=True)
```

.flaskenv Code

This is used to start the web app in development mode using the FLASK ENV environment variable:

```
FLASK_ENV=development

FLASK_APP=app.py
```

HTML codes

1. edit.html codes

```
//templates/edit.html

{% extends "layout.html" %}

{% block body %}

  <div class="row">

    <div class="col-md-4 offset-md-4">

      <div class="card card-body">

        <form action="/update/{{patient.patient_id}}" method="POST">

          <div class="form-group">

            <input type="text" name="first_name" value="{{patient.first_name}}" class="form-control">

          </div>

          <div class="form-group">

            <input type="text" name="last_name" value="{{patient.last_name}}" class="form-control">

          </div>

          <div class="form-group">

            <input type="text" name="address" value="{{patient.address}}" class="form-control">

          </div>

          <div class="form-group">
            <input type="text" name="phone" value="{{patient.phone}}" class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="email" value="{{patient.email}}" class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="age" value="{{patient.age}}" class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="gender" value="{{patient.gender}}" class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="doctor_id" value="{{patient.doctor_id}}" class="form-control">
          <div class="form-group">
            <button type="submit" class="btn btn-primary btn-block">
             Update
            </button>
          </div>
        </form>
      </div>
    </div>
```

```
  </div>

{% endblock %}
```

2. index.html codes

```
//templates/index.html
{% extends "layout.html" %}
{% block body %}
 <div class="row"><h3> Patient information </h3></div>
  <div class="row">
    <div class="col-md-4">
      {% with messages = get_flashed_messages()  %}
      {% if messages %}
      {% for message in messages %}
      <div class="alert alert-success alert-dismissible fade show" role="alert">
        {{ message }}
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
          <span aria-hidden="true">×</span>
        </button>
      </div>
      {% endfor %}
      {% endif %}
      {% endwith %}
      <div class="card card-body">
        <form action="{{url_for('add_patient')}}" method="POST">
          <div class="form-group">
            <input type="text" class="form-control" name="first_name" placeholder="FIRST NAME">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="last_name" placeholder="LAST NAME">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="address" placeholder="ADDRESS">
          </div>
```

```
   <div class="form-group">
    <input type="text" class="form-control" name="phone" placeholder="PHONE NUMBER">
   </div>
   <div class="form-group">
    <input type="text" class="form-control" name="email" placeholder="EMAIL ADDRESS">
   </div>
   <div class="form-group">
    <input type="text" class="form-control" name="age" placeholder="AGE">
   </div>
   <div class="form-group">
    <input type="text" class="form-control" name="gender" placeholder="GENDER">
   </div>
   <div class="form-group">
    <input type="text" class="form-control" name="doctor_id" placeholder="DOCTOR ID">
   </div>
   <button class="btn btn-primary btn-block">
    Save
   </button>
  </form>
 </div>
</div>
<div class="col-md-8">
 <table id="example" class="table table-striped table-bordered" style="width:100%">
  <thead>
   <tr>
    <td>PATIENT ID</td>
    <td>FIRST NAME</td>
    <td>LAST NAME</td>
    <td>ADDRESS</td>
    <td>PHONE NUMBER</td>
    <td>EMAIL ADDRESS</td>
    <td>AGE</td>
```

```
            <td>GENDER</td>
            <td>DOCTOR ID</td>
            <td> </td>
          </tr>
        </thead>
        <tbody>
          {% for row in list_users %}
          <tr>
            <td>{{row[0]}}</td>
            <td>{{row[1]}}</td>
            <td>{{row[2]}}</td>
            <td>{{row[3]}}</td>
            <td>{{row[4]}}</td>
            <td>{{row[5]}}</td>
            <td>{{row[6]}}</td>
            <td>{{row[7]}}</td>
            <td>{{row[8]}}</td>
            <td width="130">
              <a></a>
              <a href="/delete/{{row[0]}}" class="btn btn-danger btn-delete btn-sm">delete</a>
            </td>
          </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>
{% endblock %}
```

3. index.html codes

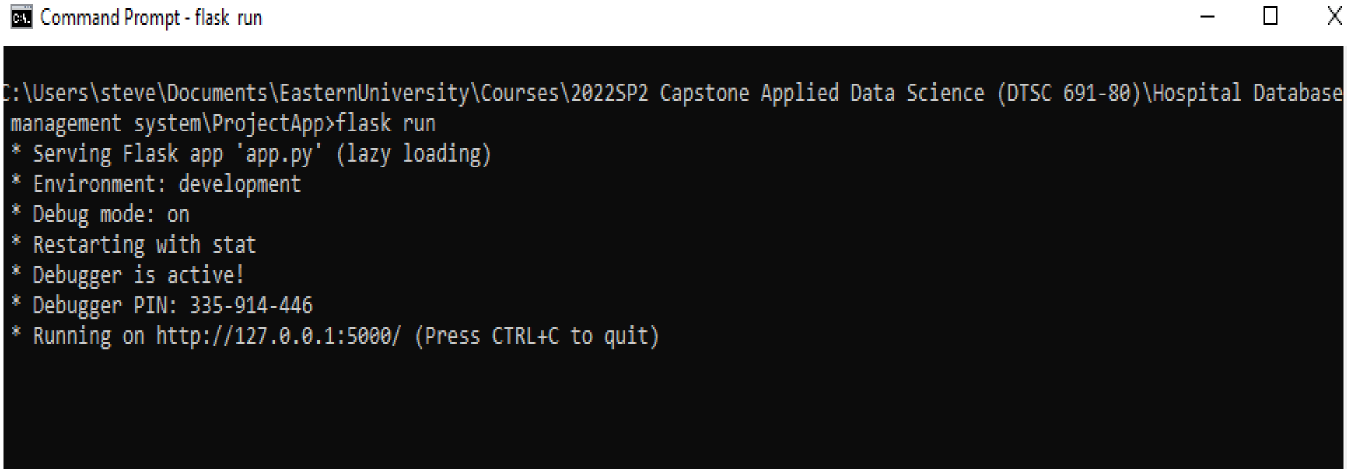
//templates/layout.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8">

    <title>HDMS Capstone Project</title>

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>


<script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>

<script src="https://cdn.datatables.net/1.10.16/js/dataTables.bootstrap4.min.js"></script>

</head>

<body>

    <nav class="navbar navbar-dark bg-dark">

      <a class="navbar-brand" href="/">Hospital Database Management System (HDMS)</a>

    </nav>


    <div class="container pt-4">

      {% block body %}

      {% endblock %}

    </div>


<script>

const btnDelete= document.querySelectorAll('.btn-delete');

if(btnDelete) {

  const btnArray = Array.from(btnDelete);

  btnArray.forEach((btn) => {

    btn.addEventListener('click', (e) => {
      if(!confirm('Are you sure you want to delete it?')){
        e.preventDefault();
      }
    });
  })
}

$(document).ready(function() {
    $('#example').DataTable({
      "aLengthMenu": [[3, 5, 10, 25, -1], [3, 5, 10, 25, "All"]],
        "iDisplayLength": 3
       }
    );
} );

</script>
  </body>
</html>
```

How to Run the Python App

Change directory to the folder containing the app.py on the Windows Command Prompt. Then, in the command prompt, type "flask run"

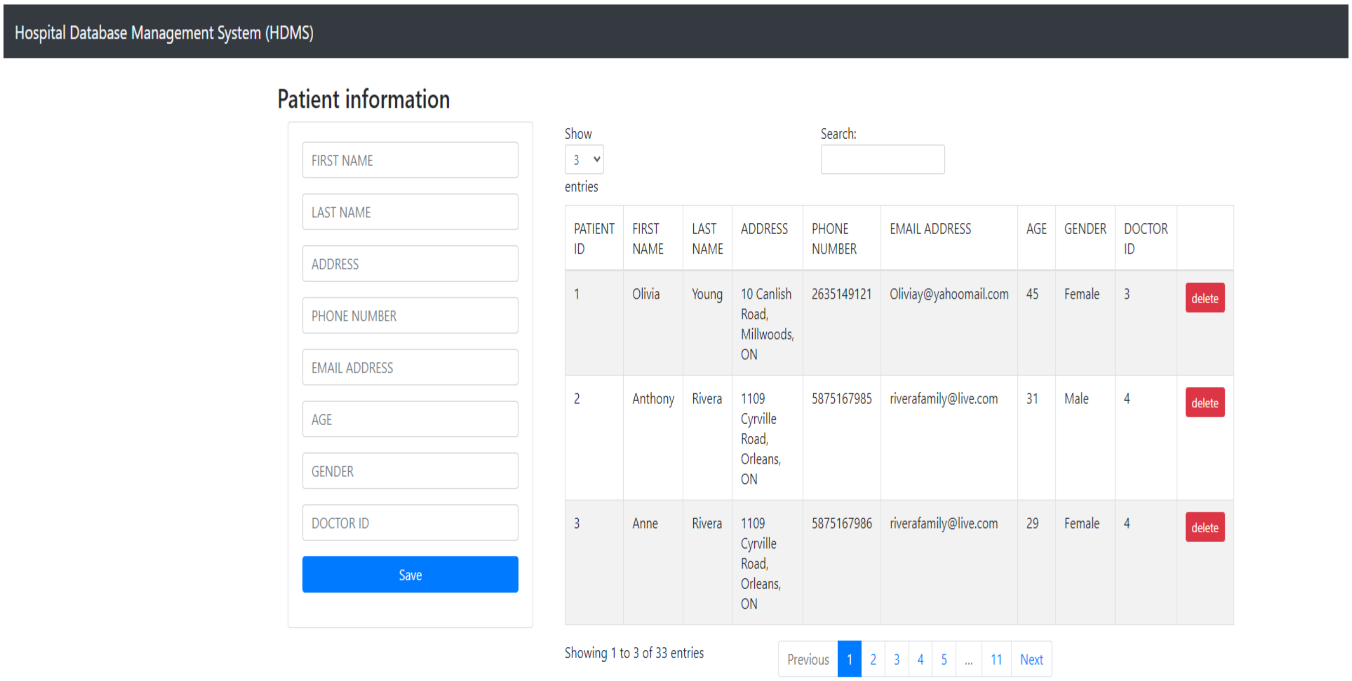
This should appear in your terminal or command prompt.



```
C:\Users\steve\Documents\EasternUniversity\Courses\2022SP2 Capstone Applied Data Science (DTSC 691-80)\Hospital Database
management system\ProjectApp>flask run
 * Serving Flask app 'app.py' (lazy loading)
 * Environment: development
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 335-914-446
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

The most important part is where it reads Running on http://127.0.0.1:5000/. The underlying idea is that 127.0.0.1 and localhost refer to the same computer. Go to that address, and you should see the web app interface, where you may enter, search for, and delete records as needed.

Overview of the HDMS Web App Interface



PATIENT ID	FIRST NAME	LAST NAME	ADDRESS	PHONE NUMBER	EMAIL ADDRESS	AGE	GENDER	DOCTOR ID	
1	Olivia	Young	10 Canlish Road, Millwoods, ON	2635149121	Oliviay@yahoomail.com	45	Female	3	delete
2	Anthony	Rivera	1109 Cyrville Road, Orleans, ON	5875167985	riverafamily@live.com	31	Male	4	delete
3	Anne	Rivera	1109 Cyrville Road, Orleans, ON	5875167986	riverafamily@live.com	29	Female	4	delete

Showing 1 to 3 of 33 entries

Analysis Plan

The following are the detailed actions that were followed by Week:

Week Activity Details

1. Make the project Proposal: Create the project proposal: The proposal was drafted during the first week, and any changes based on the mentor's advice were made before the proposal was submitted for approval.
2. Analyzed and determined the scope and needs of the project:
    - The project's business rule is established.
    - Create the ERD Diagram with all entities, attributes and relationships
    - Create the Relational Schema Diagram.
    - The final relational schema was created to contain all primary and foreign key constraints as well as the relationship between the tables.
3. SQL DDL Scripts:
    - The DDL scripts were written and fine-tuned for the project, with all of the relevant constraints included.
    - DML scripts and other queries were created.
    - According to the project specifications, additional SQL scripts were written for other deliverables such as stored procedures, triggers, and so on.
4. Create and Review Application Code: During this phase, a web-based application code was created to interact with the database. The goal of this project was to build a single front-end web page for inputting data and deleting records in one of the database's tables (patient table).
    - Virtual env setup

- flask & pyscopg2 installation
- python code
- html codes

5. Create a project outline and standardise all written deliverables.
    - Project presentation. A video walkthrough was made of the full project.
    - Project designs and deliverables were reviewed, and video recordings were edited.
    - Completed write-ups and documentation, and prepared for submission

Weekly Goals

Week 1

- Submit a project proposal and gather project materials.
- Create the proposal and submit it for approval.

Week 2

- Analyze and establish the project's functional requirements.
- Create an ERD Diagram that includes all entities, attributes, and relationships.
- Create a Relational Schema that includes all primary and foreign key constraints as well as the relationships.

Week 3

- Write the DDL and DML Scripts.
- Write the stored procedure, triggers etcs as required by the project
- Perform tests on the database's various functions.

Week 4 & 5

- Complete any additional SQL queries
- Finish the Flask course and start writing application code as needed.
- In preparation for submission, go over the codes and test the web interface with the database.

Week 6 & 7

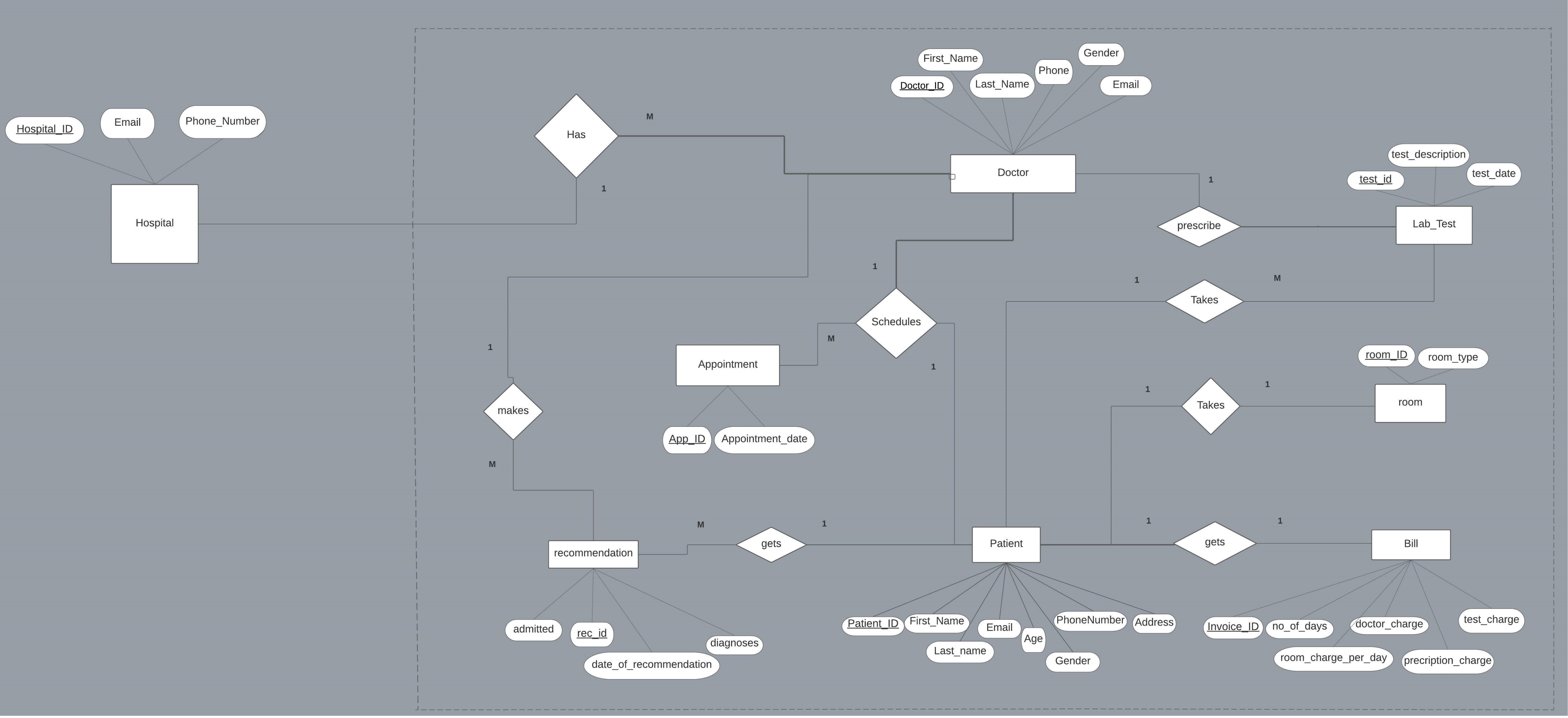
- Create the video presentation walkthrough video and the powerpoint presentation
- Create all project documentation and test all SQL and python scripts
- Review project and prepare all documents for submission
- Final project submission, DDL, DML, & Python codes to be submitted via zip file

Presentation Plan

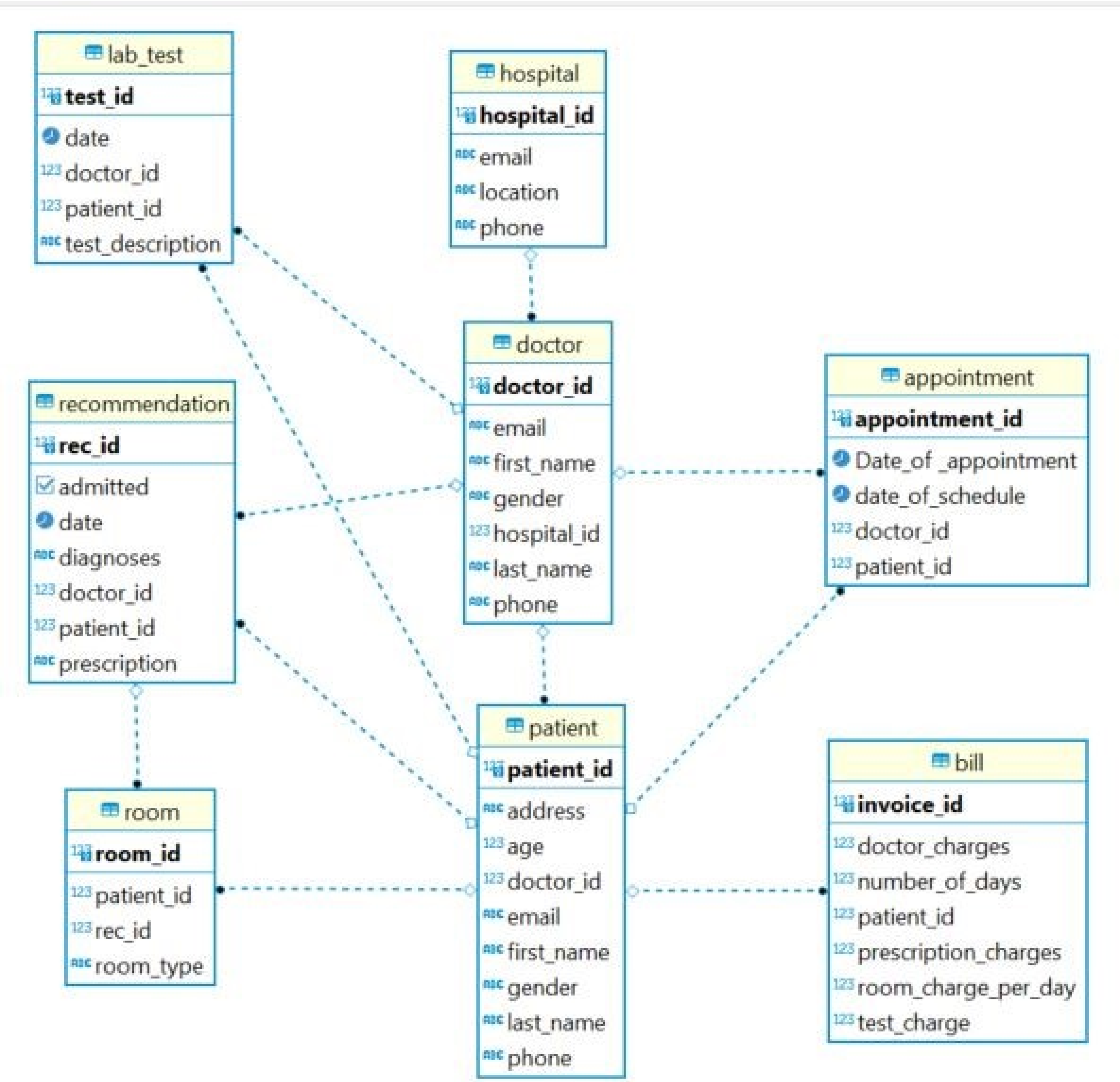
The project submissions include a 30-minute (or less) video walkthrough of the whole project for the final project presentations, a PDF or MS Word document, as well as all scripts used in a zip file.

ERD

HDMS ER Diagram



RELATIONAL SCHEMA



lab_test
test_id
date
doctor_id
patient_id
test_description

hospital
hospital_id
email
location
phone

doctor
doctor_id
email
first_name
gender
hospital_id
last_name
phone

appointment
appointment_id
Date_of _appointment
date_of_schedule
doctor_id
patient_id

recommendation
rec_id
admitted
date
diagnoses
doctor_id
patient_id
prescription

room
room_id
patient_id
rec_id
room_type

patient
patient_id
address
age
doctor_id
email
first_name
gender
last_name
phone

bill
invoice_id
doctor_charges
number_of_days
patient_id
prescription_charges
room_charge_per_day
test_charge

SQL SCRIPTS

```
---Creating the Database HDMS

    CREATE DATABASE "HDMS";


------creating the table definitions


----Hospital Table

    Drop table if Exists Hospital;
    CREATE TABLE Hospital (
    hospital_id Integer NOT NULL GENERATED ALWAYS AS IDENTITY,
    location varchar(12),
    phone varchar(10) NOT NULL,
    email varchar(30) NOT NULL,
    CONSTRAINT hospital_pk PRIMARY KEY (hospital_id)
);


----Doctor Table


    Drop table if Exists Doctor;
    CREATE TABLE doctor (
    doctor_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
    first_name varchar(20) NOT NULL,
    last_name varchar(20) NOT NULL,
    gender varchar(10) NOT NULL,
    email varchar(35) NOT NULL,
    phone varchar(10) NOT NULL,
    hospital_id integer NOT NULL,
    CONSTRAINT doctor_pk PRIMARY KEY (doctor_id),
    CONSTRAINT doctor_fk FOREIGN KEY (hospital_id)
    REFERENCES hospital(hospital_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);



----patient Table


    Drop table if Exists patient;
    CREATE TABLE patient (
    patient_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
    first_name varchar(15) NOT NULL,
    last_name varchar(15) NOT NULL,
    address varchar(40) NOT NULL,
    phone varchar(10) NOT NULL,
    email varchar(35) NOT NULL,
    age smallint NOT NULL,
    gender varchar(10) NULL,
    doctor_id integer NOT NULL,
    CONSTRAINT patient_pk PRIMARY KEY (patient_id),
    CONSTRAINT patient_fk FOREIGN KEY (doctor_id)
    REFERENCES doctor(doctor_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);


----Lab_test Table


    Drop table if Exists Lab_test;
    CREATE TABLE lab_test (
    test_id int NOT NULL GENERATED ALWAYS AS IDENTITY,
    test_description varchar(100) NOT NULL,
    "date" date NOT NULL,
    doctor_id int NOT NULL,
    patient_id int NOT NULL,
    CONSTRAINT lab_test_pk PRIMARY KEY (test_id),
    CONSTRAINT lab_test_fk FOREIGN KEY (doctor_id)
    REFERENCES doctor(doctor_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT lab_test_fk_1 FOREIGN KEY (patient_id)
    REFERENCES patient(patient_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);



----Appointment Table


    Drop table if Exists Appointment;
    CREATE TABLE appointment (
    appointment_id int NOT NULL GENERATED ALWAYS AS IDENTITY,
    date_of_schedule date NOT NULL,
    "Date_of _appointment" date NOT NULL,
    doctor_id int NOT NULL,
    patient_id int NOT NULL,
    CONSTRAINT appointment_pk PRIMARY KEY (appointment_id),
    CONSTRAINT appointment_fk FOREIGN KEY (doctor_id)
    REFERENCES doctor(doctor_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT appointment_fk_1 FOREIGN KEY (patient_id)
    REFERENCES patient(patient_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);



----Recommendation Table


    Drop table if Exists Recommendation;
    CREATE TABLE recommendation (
    rec_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
    diagnoses varchar(100) NULL,
    prescription varchar(100) NOT NULL,
    admitted bool NOT NULL,
    "date" date NOT NULL,
    patient_id integer NOT NULL,
    doctor_id integer NOT NULL,
    CONSTRAINT recommendation_pk PRIMARY KEY (rec_id),
    CONSTRAINT recommendation_fk FOREIGN KEY (doctor_id)
    REFERENCES doctor(doctor_id)
    ON DELETE CASCADE ON
    UPDATE CASCADE,
    CONSTRAINT recommendation_fk_1 FOREIGN KEY (patient_id)
    REFERENCES patient(patient_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);



----Room Table


    Drop table if Exists Room;
    CREATE TABLE public.room (
    room_id int NOT NULL GENERATED ALWAYS AS IDENTITY,
    room_type varchar(30) NOT NULL DEFAULT 'Single Bedroom',
    patient_id int NOT NULL,
    rec_id int NOT NULL,
    CONSTRAINT room_pk PRIMARY KEY (room_id),
    CONSTRAINT room_fk FOREIGN KEY (rec_id)
    REFERENCES public.recommendation(rec_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT room_fk_1 FOREIGN KEY (patient_id)
    REFERENCES patient(patient_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);



----Bill Table


    Drop table if Exists Bill;
    CREATE TABLE public.bill (
    invoice_id integer NOT NULL GENERATED ALWAYS AS IDENTITY,
    room_charge_per_day numeric(6,2) NOT NULL DEFAULT 0,
    number_of_days integer NOT NULL DEFAULT 0,
    doctor_charges numeric(6,2) NOT NULL,
    test_charge numeric(6,2) NOT NULL DEFAULT 0,
    prescription_charges numeric(6,2) NOT NULL DEFAULT 0,
    patient_id integer NOT NULL,
    CONSTRAINT bill_pk PRIMARY KEY (invoice_id),
    CONSTRAINT bill_fk FOREIGN KEY (patient_id)
    REFERENCES patient(patient_id)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
---insert Scripts

--- Hospital

INSERT INTO hospital ("location",phone,email)
    VALUES ('East','268197056','east@tclinic.com');
INSERT INTO hospital ("location",phone,email)
    VALUES ('West','268197057','west@tclinic.com');
INSERT INTO hospital ("location",phone,email)
    VALUES ('North','268197058','north@tclinic.com');
INSERT INTO hospital ("location",phone,email)
    VALUES ('South','268197059','south@tclinic.com');
INSERT INTO hospital ("location",phone,email)
    VALUES ('Central','268197060','central@tclinic.com');


----Doctor

INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Stephen','Toussaint','Male','stoussaint@tclinic.com','5875559161',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Michael','Adedeji','Male','madedeji@tclinic.com','5875559162',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Zoey','Miller','Female','zmiller@tclinic.com','5875559163',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Susan','Williams','Female','swilliams@tclinic.com','5875559164',1);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Jack','Laurent','Male','jlaurent@tclinic.com','5875569084',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Abi','Gomez','Female','agomez@tclinic.com','5875569085',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Liam','Smith','Male','lsmith@tclinic.com','5875569086',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Alice','Martinez','Female','amartinez@tclinic.com','5875569087',2);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Bob','Wilson','Male','bwilson@tclinic.com','5875551652',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Abbad','Hassan','Male','ahassan@tclinic.com','5875551653',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Tom','Jones','Male','tjones@tclinic.com','5875551654',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Julia','Anderson','Female','janderson@tclinic.com','5875551655',3);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Sophia','Pereira','Female','spereira@tclinic.com','5875563386',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Amelia','Taylor','Female','ataylor@tclinic.com','5875563387',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Nathan','Brown','Male','nbrown@tclinic.com','5875563388',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Fouad','Fadel','Male','ffadel@tclinic.com','5875563390',4);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Daniela','Perez','Female','dperez@tclinic.com','5875557543',5);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Abimbola','Adebisi','Female','aadebisi@tclinic.com','5875557544',5);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
    VALUES ('Levi','Moore','Male','lmoore@tclinic.com','5875557545',5);
INSERT INTO doctor (first_name,last_name,gender,email,phone,hospital_id)
```

```
    VALUES ('James','Martin','Male','jmartin@tclinic.com','5875557546',5);



------------Patient

INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Olivia','Young','10 Canlish Road, Millwoods, 
ON','2635149121','Oliviay@yahoomail.com',45,'Female',3);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Anthony','Rivera','1109 Cyrville Road, Orleans, 
ON','5875167985','riverafamily@live.com',31,'Male',4);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Anne','Rivera','1109 Cyrville Road, Orleans, 
ON','5875167986','riverafamily@live.com',29,'Female',4);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Melody','Rivera','1109 Cyrville Road, Orleans, 
ON','5875167987','riverafamily@live.com',7,'Female',4);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Joshua','Attah','43 Russel Drive, Alta Vista, 
ON','2639832139','attahj@rogers.com',60,'Male',1);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Frank','Zucco','984 Amroth Avenue, York, 
ON','5875198755','zuccof@hotmail.com',45,'Male',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Emma','Zucco','984 Amroth Avenue, York, 
ON','5875198753','emmazuco321@hotmail.com',43,'Female',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Alex','Zucco','984 Amroth Avenue, York, 
ON','5875198789','azucco@gmail.com',17,'Male',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Anthony','Zucco','984 Amroth Avenue, York, 
ON','5875198198','tonyzucco@live.com',14,'Male',2);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Frank','Rossi','157 Ogilvie Road, Gloucester, 
ON','3639008267','frankyRossi2015@yahoomail.com',55,'Male',5);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Jayden','Scott','10, Au Large Boulevard, Orleans, 
ON','5814198798','scottjayden@yahoomail.com',23,'Male',7);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Avery','Burton','123 Beacon Hill Drive, Winchester, 
ON','2635908410','avery4u@rogers.com',29,'Female',6);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Benjamin','Walker','319 Briarscross Boulevard, Kannata, 
ON','2635190901','benwalker2010@hotmail.com',20,'Male',9);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Debbie','Christian','1093 Ogilvie Road, Gloucester, 
ON','2635781122','debbie4u@live.com',34,'Female',8);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Elizabeth','Balogun','12 Bickerton Crescent, Essex, 
ON','7175683327','lizzybalogun@live.com',26,'Female',12);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Dylan','Robinson','27 Adamson Street, Kannata, 
ON','6857684128','drobinson@rogers.com',40,'Male',11);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Nathan','Clarke','78 Briarscross Boulevard, Kannata, 
ON','7175549090','nathanclarke123@yahoomail.com',44,'Male',15);
```

```
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Melanie','Pascua','4, Bill Cameron Lane, Barrhaven,
ON','5875990169','pascuafamily@live.com',69,'Female',13);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Emmanuel','Pascua','4, Bill Cameron Lane, Barrhaven,
ON','5814103477','pascuafamily@live.com',75,'Male',13);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Henry','Diaz','59 Augusta Avenue, Gloucester,
ON','7175553322','henryd@gmail.com',18,'Male',16);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Binita','Kumar','237 Sussex Drive, York,
ON','5814323298','kumarbi@yahoomail.com',21,'Female',14);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Akpan','Bassey','107 Briarscross Boulevard, Kannata,
ON','6856171717','abassey1990@live.com',32,'Male',19);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Anna','Soriano','237 Bridgetown Drive, York,
ON','3639002197','anna123@gmail.com',53,'Female',17);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Ben','Jackson','23 Beacon Hill Drive, Winchester,
ON','3639091207','benjackson@rogers.com',33,'Male',20);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Ruby','Jackson','23 Beacon Hill Drive, Winchester,
ON','7175171986','rubyjackson@rogers.com',33,'Female',20);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Caleb','Philips','18 Augusta Avenue, Gloucester,
ON','3639016545','cphil@rocketmail.com',34,'Male',3);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Ryan','Green','134, Au Large Boulevard, Orleans,
ON','5875961234','ryangreen2000@rogers.com',44,'Male',18);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Andrea','Green','134, Au Large Boulevard, Orleans,
ON','5875964231','andreagreen@rogers.com',42,'Female',18);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Ashley','Green','134, Au Large Boulevard, Orleans,
ON','5875961299','ashleygreen2015@rogers.com',16,'Female',18);
INSERT INTO patient (first_name,last_name,address,phone,email,age,gender,doctor_id)
    VALUES ('Amir','Houssein','148 Cyrville Road, Orleans,
ON','3639070255','amirh@rocketmail.com',24,'Male',10);



--------Appointment

INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-12','2022-03-03',3,1);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-12','2022-10-03',4,2);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-12','2022-10-03',4,3);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-12','2022-11-03',4,4);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-12','2022-03-03',1,5);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-12-11','2022-03-02',2,6);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
```

```
    VALUES ('2021-12-11','2022-03-02',2,7);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-12-11','2022-03-02',2,8);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-12-11','2022-03-02',2,9);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2022-01-01','2021-12-02',5,10);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-12-02','2021-12-02',7,11);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-12-02','2022-03-07',6,12);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-12-02','2022-03-07',9,13);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-11-30','2022-03-07',8,14);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-10-12','2022-03-07',12,15);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-10-12','2022-03-03',11,16);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-10-12','2021-07-03',15,17);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-01-01','2021-12-02',13,18);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-01-01','2021-12-02',13,19);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-11-12','2022-07-03',16,20);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-11-12','2021-07-03',14,21);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2022-01-01','2022-03-03',19,22);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-09','2021-12-02',17,23);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2022-01-01','2021-12-02',20,24);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2022-01-01','2022-02-27',20,25);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-09','2022-02-27',3,26);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-10','2022-02-27',18,27);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-10','2022-02-27',18,28);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2021-03-10','2022-02-27',18,29);
INSERT INTO appointment (date_of_schedule,"Date_of _appointment",doctor_id,patient_id)
    VALUES ('2022-01-01','2022-03-03',10,30);

--------------------------------------------------------------
--------Lab test table

INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid PCR Test','2021-03-12',3,1);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('General Routine Check up','2021-03-12',4,2);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('General Routine Check up','2021-03-12',4,3);
```

```
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('General Routine Check up','2021-03-12',4,4);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood test for Malaria Parasites','2021-03-12',1,5);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid Test (PCR)','2021-12-11',2,6);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid Test (PCR)','2021-12-11',2,7);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid Test (PCR)','2021-12-11',2,8);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid Test (PCR)','2021-12-11',2,9);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Salmonella Typhi Test','2022-01-01',5,10);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Dix-Hallpike Test for Vertigo','2021-12-02',7,11);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup','2021-12-02',6,12);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup','2021-12-02',9,13);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup','2021-11-30',8,14);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood count test, measurement of electrolytes and kidney function test','2021-10-
12',12,15);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid Test (PCR)','2021-10-12',11,16);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Salmonella Typhi Test','2021-10-12',15,17);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood count test, measurement of electrolytes and kidney function test','2021-01-
01',13,18);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Dix-Hallpike Test for Vertigo','2021-01-01',13,19);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood test for Malaria Parasites','2021-11-12',16,20);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Covid Test (PCR)','2021-11-12',14,21);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup','2022-01-01',19,22);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood test for Malaria Parasites','2021-03-09',17,23);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood count test, measurement of electrolytes and kidney function test','2022-01-
01',20,24);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Blood count test, measurement of electrolytes and kidney function test','2022-01-
01',20,25);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Salmonella Typhi Test','2021-03-09',3,26);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup','2021-03-10',18,27);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup','2021-03-10',18,28);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
    VALUES ('Routine Medical Checkup, Blood Count test','2021-03-10',18,29);
INSERT INTO lab_test (test_description,"date",doctor_id,patient_id)
```

```
    VALUES ('Dix-Hallpike Test for Vertigo','2022-01-01',10,30);

--------------------------------------------------------------
--------Recommendation


INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','complication, admission recommended',true,'2021-03-12',1,3);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','None, , Isolation recommended due to close
contact',false,'2021-03-12',2,4);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','None, Isolation recommended due to close
contact',false,'2021-03-12',3,4);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','complication, admission recommended',true,'2021-03-12',4,4);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Malaria','complication, admission recommended',true,'2021-03-12',5,1);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Positve Covid-19 Test','Isolation and medications recommended',false,'2021-12-
11',6,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-
11',7,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-
11',8,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-
11',9,2);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Typhoid','Admission recommended',true,'2022-01-01',10,5);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test, Vertigo','Isolation, rest and medications
recommended',false,'2021-12-02',11,7);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','Isolation recommended due to close contact',false,'2021-12-
02',12,6);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','Isolation and medications recommended',true,'2021-12-
02',13,9);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Malaria Parasite Detected','Isolation and medications recommended',true,'2021-11-
30',14,8);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Inconclusive, further test required','Refer to specialist for further
test',false,'2021-10-12',15,12);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Check up Ok','None',false,'2021-10-12',16,11);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Mild Vertig & Typhoid','Admission and treatment',true,'2021-10-12',17,15);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Further test required','Severe complications, admission recommended, further
specialist tests',true,'2021-01-01',18,13);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Check up OK','None',false,'2021-01-01',19,13);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
```

```
    VALUES ('Vertigo','Further test required, re-schedule appointment',false,'2021-11-12',20,16);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Check up OK','None',false,'2021-11-12',21,14);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','Medications and Isolation',false,'2022-01-01',22,19);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Malaria','admission and treatment',true,'2021-03-09',23,17);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('General Check up OK','None',false,'2022-01-01',24,20);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Malaria Parasite Detected','Medications',false,'2022-01-01',25,20);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Typhoid','Admission & treatment',false,'2021-03-09',26,3);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 Test','Isolation and rest recommended',false,'2021-03-10',27,18);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Mild Vertigo','None',false,'2021-03-10',28,18);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Negative Covid-19 Test','None',false,'2021-03-10',29,18);
INSERT INTO recommendation (diagnoses,prescription,admitted,"date",patient_id,doctor_id)
    VALUES ('Postive Covid-19 detected','severe complications, Admission',true,'2022-01-
01',30,10);



---------------------------------------------------------
------room------

INSERT INTO room (patient_id,rec_id)
    VALUES (1,1);
INSERT INTO room (patient_id,rec_id)
    VALUES (4,4);
INSERT INTO room (patient_id,rec_id)
    VALUES (5,5);
INSERT INTO room (patient_id,rec_id)
    VALUES (10,10);
INSERT INTO room (room_type,patient_id,rec_id)
    VALUES ('Family Room',13,13);
INSERT INTO room (patient_id,rec_id)
    VALUES (14,14);
INSERT INTO room (patient_id,rec_id)
    VALUES (17,17);
INSERT INTO room (room_type,patient_id,rec_id)
    VALUES ('Doubles',18,18);
INSERT INTO room (patient_id,rec_id)
    VALUES (23,23);
INSERT INTO room (patient_id,rec_id)
    VALUES (30,30);


---------------------------------------------------------
------Billing------


INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,5,1000,100,100,1);
```

```
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (400,100,2);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (250,100,3);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,3,600,100,150,4);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,14,2500,300,380,5);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,100,80,6);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,7);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,8);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (100,85,9);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,10,1500,500,550,10);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,100,50,11);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,12);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,1,200,100,80,13);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (300,150,100,14);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,550,15);
INSERT INTO bill (doctor_charges,patient_id)
    VALUES (100,16);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,5,450,150,150,17);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (120,30,4500,2000,1000,18);
INSERT INTO bill (doctor_charges,patient_id)
    VALUES (150,19);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,125,20);
INSERT INTO bill (doctor_charges,patient_id)
    VALUES (150,21);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,100,70,22);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,3,250,350,500,23);
INSERT INTO bill (doctor_charges,patient_id)
    VALUES (150,24);
INSERT INTO bill (doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (150,120,80,25);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
```

```
    VALUES (100,10,800,450,300,26);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,27);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,28);
INSERT INTO bill (doctor_charges,test_charge,patient_id)
    VALUES (150,100,29);
INSERT INTO bill
(room_charge_per_day,number_of_days,doctor_charges,test_charge,prescription_charges,patient_id)
    VALUES (100,2,300,100,50,30);
```

```
----DML Scripts

--1. ----SELECT

        SELECT *
        FROM ROOM
        WHERE ROOM_TYPE <>'Single Bedroom';


--2.-----INSERT

        INSERT INTO DOCTOR(first_name,last_name,gender,email,phone,hospital_id)
         VALUES ('Gabriella','Mendes','Female','gmendes@tclinic.com','512654891',3);

        INSERT INTO DOCTOR(first_name,last_name,gender,email,phone,hospital_id)
         VALUES ('Ian','Murphy','Male','imurphy@tclinic.com','522659186',1);

         INSERT INTO PATIENT (first_name,last_name,address,phone,email,age,gender,doctor_id)
         VALUES ('Melissa','Burgh','15 Smyth Road, Innes,
ON','6335768113','meb123@gmail.com',25,'Female',21);

         INSERT INTO PATIENT (first_name,last_name,address,phone,email,age,gender,doctor_id)
         VALUES ('Bob','Cameron','10 Coventry Avenue, Innes,
ON','6332670943','cameronb@hotmail.com',65,'Male',22);


--3. -----UPDATE

        UPDATE DOCTOR
        SET EMAIL='johnmurphy@tclinic.com',
            FIRST_NAME ='John'
        WHERE DOCTOR_ID=22;


--4. ------JOINS

        SELECT
        TRIM(CONCAT(DC.FIRST_NAME,' ', DC.LAST_NAME))as DOCTOR_NAME,
        DC.EMAIL AS DOCTOR_EMAIL,
        TRIM(CONCAT(PT.FIRST_NAME,' ', PT.LAST_NAME))as PATIENT_NAME,
        PT.EMAIL AS PATIENT_EMAIL
        FROM DOCTOR AS DC
        INNER JOIN PATIENT AS PT ON DC.DOCTOR_ID=PT.DOCTOR_ID;


--5. -----AGGREGATION (COUNT)

        SELECT
        DC.DOCTOR_ID,
        DC.EMAIL AS DOCTOR_EMAIL,
        COUNT(PT.PATIENT_ID) AS NO_OF_PATIENTS
        FROM DOCTOR DC
        INNER JOIN PATIENT AS PT
        ON DC.DOCTOR_ID=PT.DOCTOR_ID
        GROUP BY DC.DOCTOR_ID
        ORDER BY COUNT(PT.PATIENT_ID) DESC;
```

```
---6. ---DELETE ---

        DELETE
        FROM PATIENT
        WHERE PATIENT_ID =33;


--7 CASE STATEMENT ---

        SELECT
        PT.PATIENT_ID,
        CONCAT(PT.FIRST_NAME,' ', PT.LAST_NAME)as PATIENT_NAME,
        PT.GENDER,
        PT.AGE,
        PT.EMAIL,
        RM.DIAGNOSES,
        CASE
        WHEN RM.DIAGNOSES LIKE '%Covid%'
        THEN 'SEND COVID RECORD TO PROVINCIAL HEALTHCARE'
        ELSE 'NOT COVID RELATED: IGNORE'
        END  AS COVID19_POLICY
        FROM PATIENT AS PT
        INNER JOIN RECOMMENDATION RM
        ON PT.PATIENT_ID=RM.PATIENT_ID


---8 AGGREGATION (SUM)

SELECT
P.PATIENT_ID,
CONCAT(P.FIRST_NAME,' ', P.LAST_NAME)as PATIENT_NAME,
SUM(DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY *
NUMBER_OF_DAYS)) AS TOTAL_BILL
FROM PATIENT P
INNER JOIN BILL B
ON P.PATIENT_ID=B.PATIENT_ID
GROUP BY P.PATIENT_ID
ORDER BY  TOTAL_BILL
```

```
----Views


-----view 1

CREATE OR REPLACE VIEW Female_patients_doctors
AS
SELECT
CONCAT(D.FIRST_NAME,' ', D.LAST_NAME)as DOCTOR_NAME,
D.GENDER AS DOCTOR_GENDER,
CONCAT(P.FIRST_NAME,' ', P.LAST_NAME)as PATIENT_NAME,
D.EMAIL AS DOCTORS_EMAIL ,
P.EMAIL AS PATIENTS_EMAIL,
P.ADDRESS AS PATIENT_ADDRESS,
EXTRACT (YEAR FROM CURRENT_DATE) - P.AGE AS PATIENTS_YEAR_OF_BIRTH
FROM DOCTOR D
INNER JOIN PATIENT P ON D.DOCTOR_ID=P.DOCTOR_ID
WHERE P.GENDER='Female';

select * from Female_patients_doctors
ORDER BY PATIENTS_YEAR_OF_BIRTH



-----view 2

CREATE OR REPLACE VIEW Male_patients_doctors
AS
SELECT
CONCAT(D.FIRST_NAME,' ', D.LAST_NAME)as DOCTOR_NAME,
D.GENDER AS DOCTOR_GENDER,
CONCAT(P.FIRST_NAME,' ', P.LAST_NAME)as PATIENT_NAME,
D.EMAIL AS DOCTORS_EMAIL ,
P.EMAIL AS PATIENTS_EMAIL,
P.ADDRESS AS PATIENT_ADDRESS,
EXTRACT (YEAR FROM CURRENT_DATE) - P.AGE AS PATIENTS_YEAR_OF_BIRTH
FROM DOCTOR D
INNER JOIN PATIENT P ON D.DOCTOR_ID=P.DOCTOR_ID
AND P.GENDER='Male';

select * from Male_patients_doctors
```

```
----FUNCTIONS


---FUNCTION 1

---CALCULATE THE TOTAL ROOM CHARGE FOR EACH PATIENT THAT WERE ASSIGNED A ROOM

CREATE OR REPLACE function TOTAL_ROOM_CHARGES(INVOICE_ID INTEGER)
    RETURNS INTEGER
    LANGUAGE PLPGSQL
    AS
    $$

    DECLARE
        ROOM_CHARGE INTEGER;

    BEGIN

         SELECT
         (ROOM_CHARGE_PER_DAY * NUMBER_OF_DAYS) as ROOM_CHARGE_TOTAL
         INTO ROOM_CHARGE
         FROM BILL
         WHERE BILL.INVOICE_ID = TOTAL_ROOM_CHARGES.INVOICE_ID;


         RETURN (ROOM_CHARGE);
    END;

    $$;


    --TESTING FUNCTION

         SELECT TOTAL_ROOM_CHARGES(1) ;
         SELECT TOTAL_ROOM_CHARGES(10) ;
         SELECT TOTAL_ROOM_CHARGES(13) ;






---fUNCTION 2

---CALCULATE THE TOTAL HOSPITAL BILLS FOR EACH PATIENTS

CREATE OR REPLACE function TOTAL_BILL (INVOICE_ID INTEGER)
    returnS INTEGER
    language PLPGSQL
    AS
    $$

    declare
        OUTSTANDING_BALANCE INTEGER;

    BEGIN
```

```
         SELECT
         (DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY * 
NUMBER_OF_DAYS))
         INTO OUTSTANDING_BALANCE
         FROM bill
         WHERE BILL.INVOICE_ID = TOTAL_BILL.INVOICE_ID;


         RETURN (OUTSTANDING_BALANCE);
    END;

    $$;


    --TESTING FUNCTION

         SELECT TOTAL_BILL(1) ;
         SELECT TOTAL_BILL(5) ;
         SELECT TOTAL_BILL(10) ;
         SELECT TOTAL_BILL(15) ;






---------fUNCTION 3

---CALCULATE THE TOTAL HOSPITAL BILLS FOR EACH PATIENTS (eXTENSIVE)

          CREATE OR REPLACE FUNCTION TOTAL_OUTSTANDING_BILLS(INVOICE_ID integer)
          RETURNS table ( PATIENT_ID INTEGER,
                         PATIENT_NAME TEXT,
                         TOTAL_BILL numeric(8,2))

          LANGUAGE plpgsql
          AS
          $$

          BEGIN

          RETURN QUERY
                 SELECT
                 B.PATIENT_ID,
                 TRIM(CONCAT(A.FIRST_NAME,' ', A.LAST_NAME))as PATIENT_NAME,
                 (DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY * 
NUMBER_OF_DAYS)) as TOTAL_BILL
                 FROM BILL B
                 INNER JOIN PATIENT A USING (PATIENT_ID)
                 WHERE B.INVOICE_ID = TOTAL_OUTSTANDING_BILLS.INVOICE_ID;
            END;

            $$;


--TESTING FUNCTION
```

```
SELECT TOTAL_OUTSTANDING_BILLS(3)
SELECT TOTAL_OUTSTANDING_BILLS(6)
SELECT TOTAL_OUTSTANDING_BILLS(9)
SELECT TOTAL_OUTSTANDING_BILLS(12)
SELECT TOTAL_OUTSTANDING_BILLS(15)









         SELECT TOTAL_BILL(20) ;
```

```
---------------------------------------------------------------------------
---Stored Procedure 1


    CREATE OR REPLACE PROCEDURE  AddDoctor()
        LANGUAGE PLPGSQL
        AS
        $$

            BEGIN

             INSERT INTO DOCTOR (first_name,last_name,gender,email,phone,hospital_id)
            VALUES ('Stephen','Bent','Male','sbent@tclinic.com','4159780934',3);

            END
        $$;



        CALL AddDoctor()

---------------------------------------------------------------------------
---Stored Procedure 2

        CREATE OR REPLACE PROCEDURE  CombineData(PATIENT_ID INTEGER)
        LANGUAGE PLPGSQL
        AS
        $$
            DECLARE NEWDATA RECORD;
            BEGIN

                 SELECT
                 A.PATIENT_ID,
                 A.EMAIL,
                 TRIM(CONCAT(A.FIRST_NAME,' ', A.LAST_NAME))as PATIENT_NAME,
                 (DOCTOR_CHARGES + TEST_CHARGE + PRESCRIPTION_CHARGES +  (ROOM_CHARGE_PER_DAY *
NUMBER_OF_DAYS)) as TOTAL_BILL
                 INTO NEWDATA
                 FROM BILL AS B
                 INNER JOIN PATIENT AS A USING (PATIENT_ID)
                 WHERE A.PATIENT_ID = CombineData.PATIENT_ID;

            RAISE NOTICE 'Patient Info and Total Bill is %' , NEWDATA;
            END;


        $$;


CALL CombineData(8)
CALL CombineData(10)
CALL CombineData(16)
CALL CombineData(29)
CALL AddDoctor()
```

```
        DROP TABLE IF EXISTS Patient_Audit_Logs;
        CREATE TABLE  Patient_Audit_LOGS (
        Patient_ID INTEGER NOT NULL,

        FIRST_NAME VARCHAR(20) NOT NULL,

        LAST_NAME VARCHAR(20) NOT NULL,

        EMAIL VARCHAR(50) NOT NULL,

        PATIENT_ADDITION_TIME timestamp NOT NULL);



    ---Trigger Function

        CREATE OR REPLACE FUNCTION patient_insert_trigger_fnc()
        RETURNS TRIGGER
        LANGUAGE PLPGSQL
        AS
        $$

           BEGIN
              --trigger logic
          INSERT INTO Patient_Audit_Logs (patient_id, first_name,
last_name,email,patient_Addition_time)

          VALUES(NEW.patient_id,NEW.first_name,NEW.last_name,NEW.email,current_date);

          RETURN NEW;

           END;
        $$;



---------Trigger Estatement

            CREATE TRIGGER patient_insert_trigger
            AFTER INSERT
            ON PATIENT
            FOR EACH ROW

            EXECUTE PROCEDURE patient_insert_trigger_fnc();
```

PYTHON & HTML SCRIPTS

```python
 #app.py
from flask import Flask, render_template, request, redirect, url_for, flash
import psycopg2 #pip install psycopg2
import psycopg2.extras


app = Flask(__name__)
app.secret_key = "cairocoders-ednalan"

DB_HOST =  "localhost"
DB_NAME = "HDMS"
DB_USER = "postgres"
DB_PASS = 12345

conn = psycopg2.connect(dbname=DB_NAME, user=DB_USER, password=DB_PASS, host=DB_HOST)


@app.route('/')
def Index():
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    s = "SELECT * FROM patient"
    cur.execute(s) # Execute the SQL
    list_users = cur.fetchall()
    return render_template('index.html', list_users = list_users)

@app.route('/add_patient', methods=['POST'])
def add_patient():
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    if request.method == 'POST':
        first_name = request.form['first_name']
        last_name= request.form['last_name']
        address = request.form['address']
        phone = request.form['phone']
        email = request.form['email']
        age = request.form['age']
        gender = request.form['gender']
        doctor_id = request.form['doctor_id']
        cur.execute("INSERT INTO patient (first_name, last_name,address,phone,email,age,gender, doctor_id) VALUES (%s,%s,%s,%s,%s,%s,%s,%s)", (first_name, last_name,address,phone,email,age,gender, doctor_id))
        conn.commit()
        flash('Record(s) Added successfully')
        return redirect(url_for('Index'))

@app.route('/delete/<string:patient_id>', methods = ['POST','GET'])
def delete_patient(patient_id):
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    cur.execute('DELETE FROM patient WHERE patient_id = {0}'.format(patient_id))
    conn.commit()
    flash('Record Removed Successfully')
    return redirect(url_for('Index'))

if __name__ == "__main__":
    app.run(debug=True)

```

```
FLASK_ENV=development

FLASK_APP=app.py

```

```
//templates/index.html
{% extends "layout.html" %}
{% block body %}
 <div class="row"><h3> Patient information </h3></div>
  <div class="row">
    <div class="col-md-4">
      {% with messages = get_flashed_messages()  %}
      {% if messages %}
      {% for message in messages %}
      <div class="alert alert-success alert-dismissible fade show" role="alert">
        {{ message }}
        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
          <span aria-hidden="true">×</span>
        </button>
      </div>
      {% endfor %}
      {% endif %}
      {% endwith %}
      <div class="card card-body">
        <form action="{{url_for('add_patient')}}" method="POST">
          <div class="form-group">
            <input type="text" class="form-control" name="first_name" placeholder="FIRST NAME">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="last_name" placeholder="LAST NAME">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="address" placeholder="ADDRESS">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="phone" placeholder="PHONE NUMBER">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="email" placeholder="EMAIL ADDRESS">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="age" placeholder="AGE">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="gender" placeholder="GENDER">
          </div>
          <div class="form-group">
            <input type="text" class="form-control" name="doctor_id" placeholder="DOCTOR ID">
          </div>
          <button class="btn btn-primary btn-block">
            Save
          </button>
        </form>
      </div>
    </div>
    <div class="col-md-8">
      <table id="example" class="table table-striped table-bordered" style="width:100%">
        <thead>
          <tr>
            <td>PATIENT ID</td>
            <td>FIRST NAME</td>
            <td>LAST NAME</td>
            <td>ADDRESS</td>
            <td>PHONE NUMBER</td>
            <td>EMAIL ADDRESS</td>
            <td>AGE</td>
            <td>GENDER</td>
            <td>DOCTOR ID</td>
            <td> </td>
          </tr>
        </thead>
        <tbody>
          {% for row in list_users %}
          <tr>
            <td>{{row[0]}}</td>
            <td>{{row[1]}}</td>
            <td>{{row[2]}}</td>
            <td>{{row[3]}}</td>
            <td>{{row[4]}}</td>
            <td>{{row[5]}}</td>
            <td>{{row[6]}}</td>
            <td>{{row[7]}}</td>
            <td>{{row[8]}}</td>
            <td width="130">
              <a></a>
              <a href="/delete/{{row[0]}}" class="btn btn-danger btn-delete btn-sm">delete</a>
            </td>
          </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>

{% endblock %}

```

```
//templates/edit.html
{% extends "layout.html" %}
{% block body %}
  <div class="row">
    <div class="col-md-4 offset-md-4">
      <div class="card card-body">
        <form action="/update/{{patient.patient_id}}" method="POST">
          <div class="form-group">
            <input type="text" name="first_name" value="{{patient.first_name}}"
            class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="last_name" value="{{patient.last_name}}"
            class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="address" value="{{patient.address}}"
            class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="phone" value="{{patient.phone}}"
            class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="email" value="{{patient.email}}"
            class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="age" value="{{patient.age}}" class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="gender" value="{{patient.gender}}"
            class="form-control">
          </div>
          <div class="form-group">
            <input type="text" name="doctor_id" value="{{patient.doctor_id}}"
            class="form-control">
          <div class="form-group">
            <button type="submit" class="btn btn-primary btn-block">
              Update
            </button>
          </div>
        </form>
      </div>
    </div>
  </div>
{% endblock %}

```

```
//templates/layout.html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HDMS Capstone Project</title>
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
<script src="https://cdn.datatables.net/1.10.16/js/dataTables.bootstrap4.min.js"></script>
</head>
<body>
    <nav class="navbar navbar-dark bg-dark">
      <a class="navbar-brand" href="/">Hospital Database Management System (HDMS)</a>
    </nav>

    <div class="container pt-4">
      {% block body %}
      {% endblock %}
    </div>

<script>
const btnDelete= document.querySelectorAll('.btn-delete');
if(btnDelete) {
  const btnArray = Array.from(btnDelete);
  btnArray.forEach((btn) => {
    btn.addEventListener('click', (e) => {
      if(!confirm('Are you sure you want to delete it?')){
        e.preventDefault();
      }
    });
  })
}

$(document).ready(function() {
    $('#example').DataTable({
      "aLengthMenu": [[3, 5, 10, 25, -1], [3, 5, 10, 25, "All"]],
        "iDisplayLength": 3
       }
    );
} );

</script>
  </body>
</html>
```