

Please past below

(1) the SQL code you have used to create the schema of your database (only create table and alter table statements (if any), not statements for inserting values)

-- We have written 2 databases (sql_HOSPITAL_OLD, sql_HOSPITAL_NEW) which the new one is the modified schema and it is in the optimization part.

```
DROP DATABASE IF EXISTS `sql_HOSPITAL_OLD`;
CREATE DATABASE `sql_HOSPITAL_OLD`;
USE `sql_HOSPITAL_OLD`;
```

```
DROP TABLE IF EXISTS `Physician`;
CREATE TABLE `Physician` (
  `EmployeeID` BIGINT NOT NULL,
  `Name` CHAR(255) NOT NULL,
  `Position` CHAR(255) NOT NULL,
  `SSN` BIGINT NOT NULL,
  CONSTRAINT `pk_physician` PRIMARY KEY(`EmployeeID`)
```

```
DROP TABLE IF EXISTS `Department`;
CREATE TABLE `Department` (
  `DepartmentID` BIGINT NOT NULL,
  `Name` CHAR(255) NOT NULL,
  `Head` BIGINT NOT NULL,
  CONSTRAINT `pk_Department` PRIMARY KEY(`DepartmentID`),
  CONSTRAINT `fk_Department_Physician_EmployeeID` FOREIGN KEY(`Head`)
REFERENCES `Physician`(`EmployeeID`)
);
```

```
DROP TABLE IF EXISTS `Affiliated_With`;
CREATE TABLE `Affiliated_With` (
  `Physician` BIGINT NOT NULL,
  `Department` BIGINT NOT NULL,
  `PrimaryAffiliation` BOOLEAN NOT NULL,
  CONSTRAINT `fk_Affiliated_With_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`),
  CONSTRAINT `fk_Affiliated_With_Department_DepartmentID` FOREIGN KEY(`Department`)
REFERENCES `Department`(`DepartmentID`),
  PRIMARY KEY(`Physician`, `Department`)
);
```

```
DROP TABLE IF EXISTS `Procedures`;
CREATE TABLE `Procedures` (
  `Code` BIGINT PRIMARY KEY NOT NULL,
  `Name` CHAR(255) NOT NULL,
  `Cost` REAL NOT NULL
);
```

```
DROP TABLE IF EXISTS `Trained_In`;
CREATE TABLE `Trained_In` (
  `Physician` BIGINT NOT NULL,
  `Treatment` BIGINT NOT NULL,
  `CertificationDate` DATETIME NOT NULL,
  `CertificationExpires` DATETIME NOT NULL,
  CONSTRAINT `fk_Trained_In_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`),
  CONSTRAINT `fk_Trained_In_Procedures_Code` FOREIGN KEY(`Treatment`)
REFERENCES `Procedures`(`Code`),
  PRIMARY KEY(`Physician`, `Treatment`)
);
```

```
DROP TABLE IF EXISTS `Patient`;
CREATE TABLE `Patient` (
  `SSN` BIGINT PRIMARY KEY NOT NULL,
  `Name` CHAR(255) NOT NULL,
  `Address` CHAR(255) NOT NULL,
  `Phone` CHAR(255) NOT NULL,
  `InsuranceID` CHAR(255) NOT NULL,
  `PCP` BIGINT NOT NULL,
  CONSTRAINT `fk_Patient_Physician_EmployeeID` FOREIGN KEY(`PCP`) REFERENCES
`Physician`(`EmployeeID`)
);
```

```
DROP TABLE IF EXISTS `Nurse`;
CREATE TABLE `Nurse` (
  `EmployeeID` BIGINT PRIMARY KEY NOT NULL,
  `Name` CHAR(255) NOT NULL,
  `Position` CHAR(255) NOT NULL,
  `Registered` BOOLEAN NOT NULL,
  `SSN` BIGINT NOT NULL
);
```

```
DROP TABLE IF EXISTS `Appointment`;
CREATE TABLE `Appointment` (
```

```

`AppointmentID` BIGINT PRIMARY KEY NOT NULL,
`Patient` BIGINT NOT NULL,
`PrepNurse` BIGINT,
`Physician` BIGINT NOT NULL,
`Start` DATETIME NOT NULL,
`End` DATETIME NOT NULL,
`ExaminationRoom` LONGTEXT NOT NULL,
CONSTRAINT `fk_Appointment_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES
`Patient`(`SSN`),
CONSTRAINT `fk_Appointment_Nurse_EmployeeID` FOREIGN KEY(`PrepNurse`)
REFERENCES `Nurse`(`EmployeeID`),
CONSTRAINT `fk_Appointment_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`)
);

```

```

DROP TABLE IF EXISTS `Medication`;
CREATE TABLE `Medication` (
`Code` BIGINT PRIMARY KEY NOT NULL,
`Name` CHAR(255) NOT NULL,
`Brand` CHAR(255) NOT NULL,
`Description` CHAR(255) NOT NULL
);

```

```

DROP TABLE IF EXISTS `Prescribes`;
CREATE TABLE `Prescribes` (
`Physician` BIGINT NOT NULL,
`Patient` BIGINT NOT NULL,
`Medication` BIGINT NOT NULL,
`Date` DATETIME NOT NULL,
`Appointment` BIGINT,
`Dose` CHAR(255) NOT NULL,
PRIMARY KEY(`Physician`, `Patient`, `Medication`, `Date`),
CONSTRAINT `fk_Prescribes_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`),
CONSTRAINT `fk_Prescribes_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES
`Patient`(`SSN`),
CONSTRAINT `fk_Prescribes_Medication_Code` FOREIGN KEY(`Medication`)
REFERENCES `Medication`(`Code`),
CONSTRAINT `fk_Prescribes_Appointment_AppointmentID` FOREIGN KEY(`Appointment`)
REFERENCES `Appointment`(`AppointmentID`)
);

```

```

DROP TABLE IF EXISTS `Block`;

```

```
CREATE TABLE `Block` (  
  `BlockFloor` BIGINT NOT NULL,  
  `BlockCode` BIGINT NOT NULL,  
  PRIMARY KEY(`BlockFloor`, `BlockCode`)  
);
```

```
DROP TABLE IF EXISTS `Room`;  
CREATE TABLE `Room` (  
  `RoomNumber` BIGINT PRIMARY KEY NOT NULL,  
  `RoomType` CHAR(255) NOT NULL,  
  `BlockFloor` BIGINT NOT NULL,  
  `BlockCode` BIGINT NOT NULL,  
  `Unavailable` BOOLEAN NOT NULL,  
  CONSTRAINT `fk_Room_Block_PK` FOREIGN KEY(`BlockFloor`, `BlockCode`)  
REFERENCES `Block`(`BlockFloor`, `BlockCode`)  
);
```

```
DROP TABLE IF EXISTS `On_Call`;  
CREATE TABLE `On_Call` (  
  `Nurse` BIGINT NOT NULL,  
  `BlockFloor` BIGINT NOT NULL,  
  `BlockCode` BIGINT NOT NULL,  
  `OnCallStart` DATETIME NOT NULL,  
  `OnCallEnd` DATETIME NOT NULL,  
  PRIMARY KEY(`Nurse`, `BlockFloor`, `BlockCode`, `OnCallStart`, `OnCallEnd`),  
  CONSTRAINT `fk_OnCall_Nurse_EmployeeID` FOREIGN KEY(`Nurse`) REFERENCES  
`Nurse`(`EmployeeID`),  
  CONSTRAINT `fk_OnCall_Block_Floor` FOREIGN KEY(`BlockFloor`, `BlockCode`)  
REFERENCES `Block`(`BlockFloor`, `BlockCode`)  
);
```

```
DROP TABLE IF EXISTS `Stay`;  
CREATE TABLE `Stay` (  
  `StayID` BIGINT PRIMARY KEY NOT NULL,  
  `Patient` BIGINT NOT NULL,  
  `Room` BIGINT NOT NULL,  
  `StayStart` DATETIME NOT NULL,  
  `StayEnd` DATETIME NOT NULL,  
  CONSTRAINT `fk_Stay_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES  
`Patient`(`SSN`),  
  CONSTRAINT `fk_Stay_Room_Number` FOREIGN KEY(`Room`) REFERENCES  
`Room`(`RoomNumber`)  
);
```

```

DROP TABLE IF EXISTS `Undergoes`;
CREATE TABLE `Undergoes` (
  `Patient` BIGINT NOT NULL,
  `Procedures` BIGINT NOT NULL,
  `Stay` BIGINT NOT NULL,
  `DateUndergoes` DATETIME NOT NULL,
  `Physician` BIGINT NOT NULL,
  `AssistingNurse` BIGINT,
  PRIMARY KEY(`Patient`, `Procedures`, `Stay`, `DateUndergoes`),
  CONSTRAINT `fk_Undergoes_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES
`Patient`(`SSN`),
  CONSTRAINT `fk_Undergoes_Procedures_Code` FOREIGN KEY(`Procedures`)
REFERENCES `Procedures`(`Code`),
  CONSTRAINT `fk_Undergoes_Stay_StayID` FOREIGN KEY(`Stay`) REFERENCES
`Stay`(`StayID`),
  CONSTRAINT `fk_Undergoes_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`),
  CONSTRAINT `fk_Undergoes_Nurse_EmployeeID` FOREIGN KEY(`AssistingNurse`)
REFERENCES `Nurse`(`EmployeeID`)
);

```

-- For the optimization part-1 (materialized view) which is described further.

```

CREATE TABLE table_patient_procedure AS
  SELECT patient.SSN AS PatientId,
    patient.Name AS Patient_Name,
    patient.Address AS Patient_Address,
    patient.Phone AS Patient_Contact,
    patient.InsuranceID AS Patient_Insurance,
    undergoes.DateUndergoes AS Procedure_Date,
    procedures.Name AS Procedure_Name,
    physician.Name AS Physician_Name,
    physician.Position AS Physician_Position
FROM patient, undergoes, procedures, physician
WHERE patient.SSN = undergoes.Patient
  AND procedures.Code = undergoes.Procedures
  AND undergoes.Physician = physician.EmployeeID
  AND undergoes.DateUndergoes > '2020-04-01'
  AND undergoes.DateUndergoes < '2020-04-30';

```

```

CREATE TABLE table_patient_doctors_nurse AS
SELECT patient.SSN AS PatientId,

```

```

patient.Name AS Patient_Name,
patient.Address AS Patient_Address,
patient.Phone AS Patient_Contact,
patient.InsuranceID AS Patient_Insurance,
appointment.Start AS Appointment_Date,
physician.Name AS Physician_Name,
physician.Position AS Physician_Position,
nurse.Name AS Nurse_Name,
nurse.Position AS Nurse_Position
FROM patient, appointment, physician, nurse
WHERE patient.SSN = appointment.Patient
      AND appointment.Physician = physician.EmployeeID
      AND appointment.PrepareNurse = nurse.EmployeeID
      AND appointment.Start >= '2020-04-01'
      AND appointment.End <= '2020-04-30';

```

```

CREATE TABLE table_patient_prescribed_medicines AS
SELECT patient.SSN AS PatientId,
      patient.Name AS Patient_Name,
      patient.Address AS Patient_Address,
      patient.Phone AS Patient_Contact,
      patient.InsuranceID AS Patient_Insurance,
      appointment.Start AS Appointment_Date,
      medication.Name AS Medication_Name,
      prescribes.Dose AS Dosage
FROM patient, appointment, prescribes, medication
WHERE patient.SSN = appointment.Patient
      AND appointment.AppointmentID = prescribes.Appointment
      AND prescribes.Medication = medication.code
      AND appointment.Start >= '2020-04-01'
      AND appointment.End <= '2020-04-30';

```

```

CREATE TABLE table_patient_appointment_procedure AS
SELECT patient.Name AS Patient_Name,
      patient.Address AS Patient_Address,
      patient.Phone AS Patient_Contact,
      appointment.Start AS Appointment_Date,
      procedures.Name AS Procedure_Name
FROM patient,
      appointment,
      undergoes,
      procedures
WHERE appointment.Patient = undergoes.Patient

```

```
AND procedures.Code = undergoes.Procedures
AND patient.SSN = appointment.Patient
AND patient.SSN = undergoes.Patient ;
```

-- For the optimization part 2 (changing query and Indexing)

```
ALTER TABLE Appointment
add key covered(Patient, PrepNurse, Physician);
ALTER TABLE Nurse
ADD KEY Nurse_Name_Registered(Name, Registered);
```

(2) the SQL code of the queries (possibly with an explanation)

```
USE sql_HOSPITAL;
-- q1
-- Availability of rooms in each block in each floor
SELECT blockfloor AS "Floor",
       blockcode AS "Block",
       count(*) "Number of available rooms"
FROM Room
WHERE unavailable='false'
GROUP BY blockfloor,
         blockcode
ORDER BY blockfloor,
         blockcode;
```

```
-- Q-2
-- Find out details of prescribed medicines for patients between the 1st to 5th of April.
-- Medication name
-- Dosage
-- Patient details
```

```
SELECT Medication.Name AS Medication_Name,
       Prescribes.Dose AS Dosage,
       Patient.SSN AS PatientId,
       Patient.Name AS Patient_Name,
       Appointment.Start AS Appointment_Date,
       Patient.Address AS Patient_Address,
       patient.Phone AS Patient_Contact,
       patient.InsuranceID AS Patient_Insurance
FROM patient,appointment,prescribes,medication
WHERE patient.SSN = appointment.Patient
      AND appointment.AppointmentID = prescribes.Appointment
      AND prescribes.Medication = medication.code
```

```
AND appointment.Start >= '2020-04-01'
AND appointment.End <= '2020-04-05';
```

```
-- q3
-- Physicians who did the undergoe but their certification expiration had been passed
-- Patient, Physician, UndergoesDate, Certification_End_Date
```

```
SELECT
P.Name AS PatientName,
U.physician as PhysicianID,
U.DateUndergoes AS UndergoesDate,
IFNULL(T.CertificationExpires,'VERIFIED') AS CERTIFICATE_EXPIRES
FROM undergoes U
LEFT JOIN patient P
      ON U.Patient=P.SSN
LEFT JOIN trained_in T
      USING(physician)
ORDER BY DateUndergoes;
```

```
-- q4
-- Names of all the physicians,
-- Their procedures which were NOT certified for THAT procedure.
-- Date when the procedure was carried out and
-- Name of the patient on which procedure have been carried out
```

```
SELECT p.name AS "Physician",
       pr.name AS "Procedure",
       u.DateUndergoes,
       pt.name AS "Patient"
FROM Physician p,
     Undergoes u,
     Patient pt,
     Procedures pr
WHERE u.patient = pt.SSN
     AND u.Procedures = pr.Code
     AND u.physician = p.EmployeeID
     AND NOT EXISTS
       ( SELECT *
         FROM Trained_In t
         WHERE t.treatment = u.Procedures
               AND t.physician = u.physician );
```

```
-- q5
```


-- Patients who had at least two appointment
-- Nurse who prepared the appointment which was a registered nurse
-- And the physician who has carried out primary care.

```
SELECT pt.name AS "Patient",
       p.name AS "Primary Physician",
       n.name AS "Nurse"
FROM Appointment a
JOIN Patient pt ON a.patient=pt.ssn
JOIN Nurse n ON a.prepnurse=n.employeeid
JOIN Physician p ON pt.pcp=p.employeeid
WHERE a.patient IN
      (SELECT Patient
       FROM Appointment a
       GROUP BY a.patient
       HAVING count(*)>=2)
      AND n.registered='true'
ORDER BY pt.name;
```

-- q6
-- All the nurses who have ever been on call for room 102.

```
SELECT n.name
FROM nurse n
WHERE employeeid IN
      ( SELECT oc.Nurse
        FROM on_call oc,
             room r
        WHERE oc.blockfloor = r.blockfloor
              AND oc.blockcode = r.blockcode
              AND r.roomnumber = 102);
```

-- Q-7
-- Finding all On call nurses,
-- Patients in different blocks
-- And their rooms
-- Between dates 01-04-2020 and 05-04-2020.

```
SELECT nurse.Name as NurseName,
       nurse.Position as NursePosition,
       on_call.OnCallStart,
       on_call.OnCallEnd,
       patient.Name AS Patient_Name,
       patient.Address AS Patient_Address,
```

```

        patient.phone AS Patient_Contact,
        patient.InsurancID AS Patient_Insuranceid
FROM on_call, nurse, patient
WHERE patient.SSN != nurse.SSN
      AND on_call.Nurse = nurse.EmployeeID
      AND on_call.OnCallStart >= '2020-04-01'
      AND on_call.OnCallEnd <= '2020-04-30'
      AND nurse.Registered=1
ORDER BY BlockFloor ASC;

```

```

-- Q-8
-- Physicians and department relations with certification start and end dates.
-- Physician name
-- Position
-- Department
-- Start and End of their certification

```

```

SELECT physician.Name AS Physician_Name,
       physician.Position AS Physician_Position,
       department.Name AS Department_Name,
       trained_in.CertificationDate,
       trained_in.CertificationExpires
FROM affiliated_with, department, physician, trained_in
WHERE affiliated_with.Physician = physician.EmployeeID
      AND affiliated_with.Department = department.DepartmentID
limit 100;

```

```

-- Q-9
-- Patients who STAYED but did NOT UNDERGO on any procedure between two dates (using
left join)
-- Patients details with Start and End date of stay

```

```

SELECT patient.SSN AS PatientId,
       patient.Name AS Patient_Name,
       patient.Address AS Patient_Address,
       patient.Phone AS Patient_Contact,
       stay.StayStart,
       stay.StayEnd
FROM stay
      left JOIN undergoes ON stay.Patient != undergoes.Patient
      left JOIN patient ON stay.Patient = patient.SSN
WHERE stay.StayStart >='2020-04-20'
      AND stay.StayEnd<='2020-04-23';

```

-- Q-10

-- Patients who got APPOINTMENTS, but had NOT been PRESCRIBED.

-- PATIENT details

```
SELECT distinct patient.Name as Patient_Name,
                patient.Address as Patient_Address,
                patient.Phone as Patient_Contact,
                physician.Name as Physician_Name,
                physician.Position as Physician_Position,
                appointment.Start as Appointment_Date

FROM patient,
        appointment, prescribes,
        physician
WHERE  appointment.AppointmentID <> prescribes.Appointment
        AND physician.EmployeeID=appointment.Physician
        AND appointment.patient=patient.SSN;
```

-- Q-11

-- Details of patients

-- Undergoing particular procedure on "2020-04-25"

-- With doctor's name and procedure name.

```
SELECT patient.SSN as PatientId,
        patient.Name as Patient_Name,
        patient.Address as Patient_Address,
        patient.Phone as Patient_Contact,
        patient.InsuranceID as Patient_Insurance,
        undergoes.DateUndergoes as Procedure_Date,
        procedures.Name as Procedure_Name,
        physician.Name as Physician_Name,
        physician.Position as Physician_Position
FROM patient, undergoes, procedures, physician
WHERE patient.SSN = undergoes.Patient
        AND procedures.Code = undergoes.Procedures
        AND undergoes.Physician=physician.EmployeeID
        AND undergoes.DateUndergoes >= '2020-04-25'
        AND undergoes.DateUndergoes < '2020-04-26';
```

-- Q-12

-- Find out the appointment details

-- Of patients between 01-04-2020 to 05-04-2020

-- With different doctors with their attending nurse name.

```
SELECT patient.SSN as PatientId,
```

```

        patient.Name as Patient_Name,
        patient.Address as Patient_Address,
        patient.Phone as Patient_Contact,
        patient.InsuranceID as Patient_Insurance,
        appointment.Start Appointment_Date,
        physician.Name as Physician_Name,
        physician.Position as Physician_Position,
        nurse.Name as Nurse_Name,
        nurse.Position as Nurse_Position
FROM patient, appointment, physician, nurse
WHERE patient.SSN = appointment.Patient
      AND appointment.Physician = physician.EmployeeID
      AND appointment.PrepareNurse = nurse.EmployeeID
      AND appointment.Start >= '2020-04-01'
      AND appointment.End <= '2020-04-05';

-- Q-13
-- PATIENTS THAT HAD APPOINTMENTS WHICH PROCEEDED TO AN UNDERGOE

```

```

SELECT patient.Name AS Patient_Name,
       appointment.Start AS Appointment_Date,
       procedures.Name AS Procedure_Name,
       patient.Phone AS Patient_Contact
FROM patient, appointment, undergoes, procedures
WHERE appointment.Patient = undergoes.Patient
      AND procedures.Code=undergoes.Procedures
      AND patient.SSN=appointment.Patient
      AND patient.SSN=undergoes.Patient;
-- Patient, Physician, UndergoesDate, Certification_End_Date

```

(3) the SQL code used for query optimization for HW2. For each query, indicate the un-optimized version and the optimized one. In case the optimization has been realized through indexes, insert the SQL code for the index creation; in case you have modified the schema (e.g. changed the domain of a field, or constructed a new materialized table, etc.), insert the code you have used for this modification.

```

-- OPTIMIZATION SECTION
-- Part-1 USING Materialized Views
-- In this Part we USE materialized views for optimizing performance.
-- For materialized views a separate table needs to be created.

```

-- OPTIMIZATION I (query 11)

```
CREATE TABLE table_patient_procedure AS
    SELECT patient.SSN AS PatientId,
    patient.Name AS Patient_Name,
    patient.Address AS Patient_Address,
    patient.Phone AS Patient_Contact,
    patient.InsuranceID AS Patient_Insurance,
    undergoes.DateUndergoes AS Procedure_Date,
    procedures.Name AS Procedure_Name,
    physician.Name AS Physician_Name,
    physician.Position AS Physician_Position
FROM patient, undergoes, procedures, physician
WHERE patient.SSN = undergoes.Patient
    AND procedures.Code = undergoes.Procedures
    AND undergoes.Physician = physician.EmployeeID
    AND undergoes.DateUndergoes > '2020-04-01'
    AND undergoes.DateUndergoes < '2020-04-30';
```

-- Optionally we can add index(es) for the queries we want to speed up.

```
CREATE INDEX index_procedure_date
    ON table_patient_procedure (Procedure_Date);
```

```
CREATE VIEW patient_and_procedure AS
    SELECT * FROM table_patient_procedure;
```

```
/* On large database we can see
    query executes 10x faster than the first query */
```

-- SLOW QUERY

```
SELECT patient.SSN as PatientId,
    patient.Name as Patient_Name,
    patient.Address as Patient_Address,
    patient.Phone as Patient_Contact,
    patient.InsuranceID as Patient_Insurance,
    undergoes.DateUndergoes as Procedure_Date,
    procedures.Name as Procedure_Name,
    physician.Name as Physician_Name,
    physician.Position as Physician_Position
FROM patient, undergoes, procedures, physician
WHERE patient.SSN = undergoes.Patient
```

```
AND procedures.Code = undergoes.Procedures
AND undergoes.Physician=physician.EmployeeID
AND undergoes.DateUndergoes >= '2020-04-25'
AND undergoes.DateUndergoes < '2020-04-26';
```

```
SHOW STATUS like "Last_Query_Cost";
```

```
-- Fast query
```

```
EXPLAIN SELECT *
FROM patient_and_procedure
WHERE Procedure_Date >= '2020-04-25'
      AND Procedure_Date < '2020-04-26';
```

```
SHOW STATUS like "Last_Query_Cost";
```

```
-- OPTIMIZATION II (query 12)
```

```
CREATE TABLE table_patient_doctors_nurse AS
SELECT patient.SSN AS PatientId,
patient.Name AS Patient_Name,
patient.Address AS Patient_Address,
patient.Phone AS Patient_Contact,
patient.InsurancID AS Patient_Insurance,
appointment.Start AS Appointment_Date,
physician.Name AS Physician_Name,
physician.Position AS Physician_Position,
nurse.Name AS Nurse_Name,
nurse.Position AS Nurse_Position
FROM patient, appointment, physician, nurse
WHERE patient.SSN = appointment.Patient
      AND appointment.Physician =physician.EmployeeID
      AND appointment.PrepareNurse = nurse.EmployeeID
      AND appointment.Start >= '2020-04-01'
      AND appointment.End   <= '2020-04-30';
```

```
CREATE INDEX index_appointment_date
      ON table_patient_doctors_nurse (Appointment_Date);
```

```
CREATE VIEW patient_and_doctors_nurse AS
SELECT * FROM table_patient_doctors_nurse;
```

/*-- Slow Query --*/

```
SELECT patient.SSN as PatientId,  
       patient.Name as Patient_Name,  
       patient.Address as Patient_Address,  
       patient.Phone as Patient_Contact,  
       patient.InsuraneID as Patient_Insurance,  
       appointment.Start Appointment_Date,  
       physician.Name as Physician_Name,  
       physician.Position as Physician_Position,  
       nurse.Name as Nurse_Name,  
       nurse.Position as Nurse_Position  
FROM patient, appointment, physician, nurse  
WHERE patient.SSN = appointment.Patient  
      AND appointment.Physician =physician.EmployeeID  
      AND appointment.PrepareNurse = nurse.EmployeeID  
      AND appointment.Start >= '2020-04-01'  
      AND appointment.End <= '2020-04-05';
```

SHOW STATUS like "Last_Query_Cost";

-- Slow query Cost : 21.913

-- Fast Query

Explain

Select *

FROM patient_and_doctors_nurse

WHERE Appointment_Date >= "2020-04-01"

AND Appointment_Date<="2020-04-05";

SHOW STATUS like "Last_Query_Cost";

-- Fast query Cost : 10.749

-- OPTIMIZATION - III - query 2

```
CREATE TABLE table_patient_prescribed_medicines AS
SELECT patient.SSN AS PatientId,
       patient.Name AS Patient_Name,
       patient.Address AS Patient_Address,
       patient.Phone AS Patient_Contact,
       patient.InsuranceID AS Patient_Insurance,
       appointment.Start AS Appointment_Date,
       medication.Name AS Medication_Name,
       prescribes.Dose AS Dosage
FROM patient,appointment,prescribes,medication
WHERE patient.SSN = appointment.Patient
      AND appointment.AppointmentID = prescribes.Appointment
      AND prescribes.Medication = medication.code
      AND appointment.Start >= '2020-04-01'
      AND appointment.End <= '2020-04-30';
```

```
CREATE INDEX index_appointment_date
ON table_patient_prescribed_medicines (Appointment_Date);
```

```
CREATE VIEW patient_prescribed_medicines AS
SELECT * FROM table_patient_prescribed_medicines;
```

-- Slow Query

```
SELECT medication.Name AS Medication_Name,
       prescribes.Dose AS Dosage,
       patient.SSN AS PatientId,
       patient.Name AS Patient_Name,
       appointment.Start AS Appointment_Date,
       patient.Address AS Patient_Address,
       patient.Phone AS Patient_Contact,
       patient.InsuranceID AS Patient_Insurance
FROM patient,appointment,prescribes,medication
WHERE patient.SSN = appointment.Patient
      AND appointment.AppointmentID = prescribes.Appointment
      AND prescribes.Medication = medication.code
      AND appointment.Start >= '2020-04-01'
      AND appointment.End <= '2020-04-05';
```

```
SHOW STATUS like "Last_Query_Cost";
```

-- Slow query Cost : '27.544'


```
-- Fast Query
Explain
SELECT *
FROM patient_prescribed_medicines
WHERE Appointment_Date >= "2020-04-01"
AND Appointment_Date<="2020-04-05";
```

```
SHOW STATUS like "Last_Query_Cost";
-- Fast query Cost : 10.249
```

```
-- OPTIMIZATION - IV - query 13
```

```
CREATE TABLE table_patient_appointment_procedure AS
    SELECT patient.Name AS Patient_Name,
           patient.Address AS Patient_Address,
           patient.Phone AS Patient_Contact,
           appointment.Start AS Appointment_Date,
           procedures.Name AS Procedure_Name
    FROM patient,
         appointment,
         undergoes,
         procedures
    WHERE appointment.Patient = undergoes.Patient
          AND procedures.Code = undergoes.Procedures
          AND patient.SSN = appointment.Patient
          AND patient.SSN = undergoes.Patient ;

CREATE INDEX index_appointment_date
    ON table_patient_appointment_procedure (Appointment_Date);
```

```
CREATE VIEW patient_appointment_procedure AS
    SELECT * FROM table_patient_appointment_procedure;
```

```
-- Slow Query
SELECT patient.Name AS Patient_Name,
       patient.Address AS Patient_Address,
       patient.Phone AS Patient_Contact,
       appointment.Start AS Appointment_Date,
       procedures.Name AS Procedure_Name
FROM patient, appointment, undergoes, procedures
WHERE appointment.Patient = undergoes.Patient
```

```

AND procedures.Code=undergoes.Procedures
AND patient.SSN=appointment.Patient
AND patient.SSN=undergoes.Patient;

```

SHOW STATUS like "Last_Query_Cost";

-- Slow query Cost : '67.013'

-- Fast Query

Explain

SELECT *

FROM patient_appointment_procedure;

-- WHERE Appointment_Date >= "2020-04-01"

-- AND Appointment_Date<="2020-04-05";

SHOW STATUS like "Last_Query_Cost";

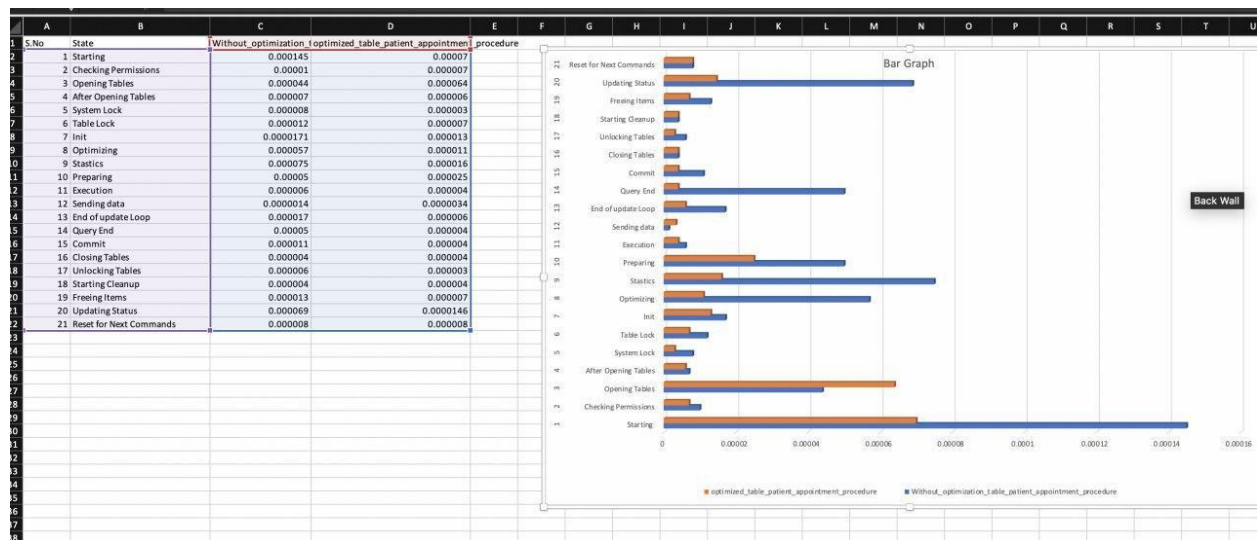
-- Fast query Cost : 0.949

-- Analytics:

-- This is the query analysis through profiling in MySQL which we have visualized.

-- The blue lines are without optimization cost of query 13

-- The orange lines are optimized cost of query 13



/* -- Part 2 Changing Query Statement and Indexing */

-- OPTIMIZATION - 5 - query 5

```
/*-- Slow Query --*/
```

```
EXPLAIN SELECT pt.name AS "Patient",
               p.name AS "Primary Physician",
               n.name AS "Nurse"
FROM Appointment a
JOIN Patient pt ON a.patient=pt.ssn
JOIN Nurse n ON a.prepnurse=n.employeeid
JOIN Physician p ON pt.pcp=p.employeeid
WHERE a.patient IN
      (SELECT Patient
       FROM Appointment a
       GROUP BY a.patient
       HAVING count(*)>=2)
AND n.registered='true'
ORDER BY pt.name;
```

```
show status like "last_query_cost";
-- Slow query Cost : '72.498000'
```

```
-- Fast query:
-- 1
```

```
ALTER TABLE Appointment
add key covered(Patient, PrepNurse, Physician);
ALTER TABLE Nurse
ADD KEY Nurse_Name_Registered(Name, Registered);
```

```
EXPLAIN SELECT pt.name AS "Patient",
               p.name AS "Primary Physician",
               n.name AS "Nurse"
FROM Appointment a
JOIN (select SSN,Name,PCP from Patient order by Name) pt ON a.patient=pt.ssn
JOIN (select EmployeeID,Name from Nurse where Registered = "true") n ON
a.prepnurse=n.employeeid
JOIN (select EmployeeID,Name from Physician) p ON pt.pcp=p.employeeid
WHERE a.patient IN
      (SELECT Patient
       FROM Appointment a
       GROUP BY a.patient
       HAVING count(*)>=2);
```

```
show status like "last_query_cost";
```

```

-- Fast Query cost = '62.498000'
-- 2
EXPLAIN SELECT pt.name AS "Patient",
              p.name AS "Primary Physician",
              n.name AS "Nurse"
FROM (SELECT Patient, PrepNurse, Physician
      FROM Appointment a
      WHERE Patient in (select Patient from Appointment a group by a.patient having
count(*)>=2)) a
JOIN (select SSN, Name, PCP from Patient order by Name) pt ON a.patient=pt.ssn
JOIN (select EmployeeID, Name from Nurse where Registered = "true") n ON
a.prepnurse=n.employeeid
JOIN (select EmployeeID, Name from Physician) p ON pt.pcp=p.employeeid;

show status like "last_query_cost";

-- Fast Query cost = '62.498000'

```

/* Part 3 Changing Schema */

-- For this part we have modified the schema known as sql_HOSPITAL_NEW

```

DROP DATABASE IF EXISTS `sql_HOSPITAL_NEW`;
CREATE DATABASE `sql_HOSPITAL_NEW`;
USE `sql_HOSPITAL_NEW`;

```

```

DROP TABLE IF EXISTS `Physician`;
CREATE TABLE `Physician` (
  `EmployeeID` SMALLINT(4) NOT NULL,
  `Name` VARCHAR(30) NOT NULL,
  `Position` VARCHAR(30) NOT NULL,
  `SSN` INT(9) NOT NULL,
  CONSTRAINT `pk_physician` PRIMARY KEY(`EmployeeID`)
);

```

```

DROP TABLE IF EXISTS `Department`;
CREATE TABLE `Department` (
  `DepartmentID` TINYINT(2) NOT NULL,
  `Name` VARCHAR(30) NOT NULL,
  `Head` SMALLINT(4) NOT NULL,

```

```
CONSTRAINT `pk_Department` PRIMARY KEY(`DepartmentID`),  
CONSTRAINT `fk_Department_Physician_EmployeeID` FOREIGN KEY(`Head`)  
REFERENCES `Physician`(`EmployeeID`)  
);
```

```
DROP TABLE IF EXISTS `Affiliated_With`;  
CREATE TABLE `Affiliated_With` (  
  `Physician` SMALLINT(4) NOT NULL,  
  `Department` TINYINT(2) NOT NULL,  
  `PrimaryAffiliation` BOOLEAN NOT NULL,  
  CONSTRAINT `fk_Affiliated_With_Physician_EmployeeID` FOREIGN KEY(`Physician`)  
REFERENCES `Physician`(`EmployeeID`),  
  CONSTRAINT `fk_Affiliated_With_Department_DepartmentID` FOREIGN KEY(`Department`)  
REFERENCES `Department`(`DepartmentID`),  
  PRIMARY KEY(`Physician`, `Department`)  
);
```

```
DROP TABLE IF EXISTS `Procedures`;  
CREATE TABLE `Procedures` (  
  `Code` TINYINT(2) PRIMARY KEY NOT NULL,  
  `Name` VARCHAR(30) NOT NULL,  
  `Cost` REAL NOT NULL  
);
```

```
DROP TABLE IF EXISTS `Trained_In`;  
CREATE TABLE `Trained_In` (  
  `Physician` SMALLINT(4) NOT NULL,  
  `Treatment` TINYINT(2) NOT NULL,  
  `CertificationDate` TIMESTAMP NOT NULL,  
  `CertificationExpires` TIMESTAMP NOT NULL,  
  CONSTRAINT `fk_Trained_In_Physician_EmployeeID` FOREIGN KEY(`Physician`)  
REFERENCES `Physician`(`EmployeeID`),  
  CONSTRAINT `fk_Trained_In_Procedures_Code` FOREIGN KEY(`Treatment`)  
REFERENCES `Procedures`(`Code`),  
  PRIMARY KEY(`Physician`, `Treatment`)  
);
```

```
DROP TABLE IF EXISTS `Patient`;  
CREATE TABLE `Patient` (  
  `SSN` INT(9) PRIMARY KEY NOT NULL,  
  `Name` VARCHAR(30) NOT NULL,  
  `Address` VARCHAR(60) NOT NULL,  
  `Phone` VARCHAR(16) NOT NULL,
```

```
`InsuranceID` VARCHAR(15) NOT NULL,  
`PCP` SMALLINT(4) NOT NULL,  
CONSTRAINT `fk_Patient_Physician_EmployeeID` FOREIGN KEY(`PCP`) REFERENCES  
`Physician`(`EmployeeID`)  
);
```

```
DROP TABLE IF EXISTS `Nurse`;  
CREATE TABLE `Nurse` (  
  `EmployeeID` SMALLINT(4) PRIMARY KEY NOT NULL,  
  `Name` VARCHAR(30) NOT NULL,  
  `Position` VARCHAR(30) NOT NULL,  
  `Registered` BOOLEAN NOT NULL,  
  `SSN` INT(9) NOT NULL  
);
```

```
DROP TABLE IF EXISTS `Appointment`;  
CREATE TABLE `Appointment` (  
  `AppointmentID` INT(8) PRIMARY KEY NOT NULL,  
  `Patient` INT(9) NOT NULL,  
  `PrepNurse` SMALLINT(4),  
  `Physician` SMALLINT(4) NOT NULL,  
  `Start` TIMESTAMP NOT NULL,  
  `End` TIMESTAMP NOT NULL,  
  `ExaminationRoom` TINYTEXT NOT NULL,  
  CONSTRAINT `fk_Appointment_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES  
  `Patient`(`SSN`),  
  CONSTRAINT `fk_Appointment_Nurse_EmployeeID` FOREIGN KEY(`PrepNurse`)  
  REFERENCES `Nurse`(`EmployeeID`),  
  CONSTRAINT `fk_Appointment_Physician_EmployeeID` FOREIGN KEY(`Physician`)  
  REFERENCES `Physician`(`EmployeeID`)  
);
```

```
DROP TABLE IF EXISTS `Medication`;  
CREATE TABLE `Medication` (  
  `Code` TINYINT PRIMARY KEY NOT NULL,  
  `Name` VARCHAR(30) NOT NULL,  
  `Brand` VARCHAR(30) NOT NULL,  
  `Description` VARCHAR(30) NOT NULL  
);
```

```
DROP TABLE IF EXISTS `Prescribes`;  
CREATE TABLE `Prescribes` (  
  `Physician` SMALLINT(4) NOT NULL,
```

```

`Patient` INT(9) NOT NULL,
`Medication` TINYINT NOT NULL,
`Date` TIMESTAMP NOT NULL,
`Appointment` INT(8),
`Dose` VARCHAR(15) NOT NULL,
PRIMARY KEY(`Physician`, `Patient`, `Medication`, `Date`),
CONSTRAINT `fk_Prescribes_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`),
CONSTRAINT `fk_Prescribes_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES
`Patient`(`SSN`),
CONSTRAINT `fk_Prescribes_Medication_Code` FOREIGN KEY(`Medication`)
REFERENCES `Medication`(`Code`),
CONSTRAINT `fk_Prescribes_Appointment_AppointmentID` FOREIGN KEY(`Appointment`)
REFERENCES `Appointment`(`AppointmentID`)
);

```

```

DROP TABLE IF EXISTS `Block`;
CREATE TABLE `Block` (
  `BlockFloor` TINYINT(1) NOT NULL,
  `BlockCode` TINYINT(2) NOT NULL,
  PRIMARY KEY(`BlockFloor`, `BlockCode`)
);

```

```

DROP TABLE IF EXISTS `Room`;
CREATE TABLE `Room` (
  `RoomNumber` SMALLINT(4) PRIMARY KEY NOT NULL,
  `RoomType` VARCHAR(30) NOT NULL,
  `BlockFloor` TINYINT(1) NOT NULL,
  `BlockCode` TINYINT(2) NOT NULL,
  `Unavailable` BOOLEAN NOT NULL,
  CONSTRAINT `fk_Room_Block_PK` FOREIGN KEY(`BlockFloor`, `BlockCode`)
REFERENCES `Block`(`BlockFloor`, `BlockCode`)
);

```

```

DROP TABLE IF EXISTS `On_Call`;
CREATE TABLE `On_Call` (
  `Nurse` SMALLINT(3) NOT NULL,
  `BlockFloor` TINYINT(1) NOT NULL,
  `BlockCode` TINYINT(2) NOT NULL,
  `OnCallStart` TIMESTAMP NOT NULL,
  `OnCallEnd` TIMESTAMP NOT NULL,
  PRIMARY KEY(`Nurse`, `BlockFloor`, `BlockCode`, `OnCallStart`, `OnCallEnd`),

```

```

    CONSTRAINT `fk_OnCall_Nurse_EmployeeID` FOREIGN KEY(`Nurse`) REFERENCES
`Nurse`(`EmployeeID`),
    CONSTRAINT `fk_OnCall_Block_Floor` FOREIGN KEY(`BlockFloor`, `BlockCode`)
REFERENCES `Block`(`BlockFloor`, `BlockCode`)
);

```

```

DROP TABLE IF EXISTS `Stay`;
CREATE TABLE `Stay` (
    `StayID` SMALLINT(4) PRIMARY KEY NOT NULL,
    `Patient` INT(9) NOT NULL,
    `Room` SMALLINT(4) NOT NULL,
    `StayStart` TIMESTAMP NOT NULL,
    `StayEnd` TIMESTAMP NOT NULL,
    CONSTRAINT `fk_Stay_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES
`Patient`(`SSN`),
    CONSTRAINT `fk_Stay_Room_Number` FOREIGN KEY(`Room`) REFERENCES
`Room`(`RoomNumber`)
);

```

```

DROP TABLE IF EXISTS `Undergoes`;
CREATE TABLE `Undergoes` (
    `Patient` INT(9) NOT NULL,
    `Procedures` TINYINT(2) NOT NULL,
    `Stay` SMALLINT(4) NOT NULL,
    `DateUndergoes` TIMESTAMP NOT NULL,
    `Physician` SMALLINT(4) NOT NULL,
    `AssistingNurse` SMALLINT(4),
    PRIMARY KEY(`Patient`, `Procedures`, `Stay`, `DateUndergoes`),
    CONSTRAINT `fk_Undergoes_Patient_SSN` FOREIGN KEY(`Patient`) REFERENCES
`Patient`(`SSN`),
    CONSTRAINT `fk_Undergoes_Procedures_Code` FOREIGN KEY(`Procedures`)
REFERENCES `Procedures`(`Code`),
    CONSTRAINT `fk_Undergoes_Stay_StayID` FOREIGN KEY(`Stay`) REFERENCES
`Stay`(`StayID`),
    CONSTRAINT `fk_Undergoes_Physician_EmployeeID` FOREIGN KEY(`Physician`)
REFERENCES `Physician`(`EmployeeID`),
    CONSTRAINT `fk_Undergoes_Nurse_EmployeeID` FOREIGN KEY(`AssistingNurse`)
REFERENCES `Nurse`(`EmployeeID`)
);

```

```

-----
/* Slow Query */
-- Optimization for q10

```



```

USE sql_HOSPITAL_OLD;
SELECT distinct patient.Name as Patient_Name,
                patient.Address as Patient_Address,
                patient.Phone as Patient_Contact,
                physician.Name as Physician_Name,
                physician.Position as Physician_Position,
                appointment.Start as Appointment_Date
FROM patient,
        appointment, prescribes,
        physician
WHERE     appointment.AppointmentID <> prescribes.Appointment
        AND physician.EmployeeID=appointment.Physician
        AND appointment.patient=patient.SSN;

```

```

SHOW STATUS LIKE "last_query_cost";
-- cost 1082.47

```

/* Faster Query */

```

USE sql_HOSPITAL_NEW;
SELECT distinct patient.Name as Patient_Name,
                patient.Address as Patient_Address,
                patient.Phone as Patient_Contact,
                physician.Name as Physician_Name,
                physician.Position as Physician_Position,
                appointment.Start as Appointment_Date
FROM patient,
        appointment, prescribes,
        physician
WHERE     appointment.AppointmentID <> prescribes.Appointment
        AND physician.EmployeeID=appointment.Physician
        AND appointment.patient=patient.SSN;

```

```

SHOW STATUS LIKE "last_query_cost";
-- cost 1080.56

```

/* Additional Information */

-- We have also put the database on the Heroku platform which response to our queries using RESTful API which we have developed using python and Flask micro framework.

-- Sample API response in Browser and also in Postman:

```
[[{"EmployeeID": 1, "Name": "Naïda Schultz", "Position": "Cardiology", "SSN": 375019270}, {"EmployeeID": 2, "Name": "Tate Armstrong", "Position": "Phrenology", "SSN": 951100813}, {"EmployeeID": 3, "Name": "Eloa Kane", "Position": "Cardiology", "SSN": 544043010}, {"EmployeeID": 4, "Name": "Nasim McIntyre", "Position": "Gastroenterology", "SSN": 244599760}, {"EmployeeID": 5, "Name": "Kadeem Langley", "Position": "Hematology", "SSN": 994785963}, {"EmployeeID": 6, "Name": "Grace Wolf", "Position": "Pulmonology", "SSN": 74007250}, {"EmployeeID": 7, "Name": "Patience Salas", "Position": "Hematology", "SSN": 51737684}, {"EmployeeID": 8, "Name": "Hilda Shields", "Position": "Gastroenterology", "SSN": 203133100}, {"EmployeeID": 9, "Name": "Lionel Sanchez", "Position": "Gastroenterology", "SSN": 954042600}, {"EmployeeID": 10, "Name": "Caleb Larson", "Position": "Cardiology", "SSN": 895978419}, {"EmployeeID": 11, "Name": "Isabelle Salinas", "Position": "Phrenology", "SSN": 998509517}, {"EmployeeID": 12, "Name": "Noble Franks", "Position": "Infectious Diseases", "SSN": 804141722}, {"EmployeeID": 13, "Name": "Jerry Ward", "Position": "Endocrinology", "SSN": 338329095}, {"EmployeeID": 14, "Name": "Frank Pierce", "Position": "Pulmonology", "SSN": 734040343}, {"EmployeeID": 15, "Name": "Moyt Stafford", "Position": "Hematology", "SSN": 775007760}, {"EmployeeID": 16, "Name": "Lucian Briggs", "Position": "Hematology", "SSN": 534086299}, {"EmployeeID": 17, "Name": "Joshua Silva", "Position": "Pulmonology", "SSN": 513760923}, {"EmployeeID": 18, "Name": "Milyari Atkinson", "Position": "Cardiology", "SSN": 522823713}, {"EmployeeID": 19, "Name": "Hu Hines", "Position": "Cardiology", "SSN": 614175233}, {"EmployeeID": 20, "Name": "Clark Emerson", "Position": "Pulmonology", "SSN": 273991165}, {"EmployeeID": 21, "Name": "Arsenio Mejia", "Position": "Phrenology", "SSN": 102887967}, {"EmployeeID": 22, "Name": "Charissa Gallegos", "Position": "Oncology", "SSN": 75030631}, {"EmployeeID": 23, "Name": "Sydney Wiggins", "Position": "Endocrinology", "SSN": 522345623}, {"EmployeeID": 24, "Name": "Moses Blanchard", "Position": "Oncology", "SSN": 299206368}, {"EmployeeID": 25, "Name": "Dillon Livingston", "Position": "Cardiology", "SSN": 963487947}, {"EmployeeID": 26, "Name": "Dalton Calderon", "Position": "Family Physician", "SSN": 80277655}, {"EmployeeID": 27, "Name": "Nasoa Schmidt", "Position": "Pulmonology", "SSN": 357002532}, {"EmployeeID": 28, "Name": "Ifeoma Murphy", "Position": "Phrenology", "SSN": 743235384}, {"EmployeeID": 29, "Name": "Jacob Bryan", "Position": "Rheumatology", "SSN": 383791023}, {"EmployeeID": 30, "Name": "Moses Warner", "Position": "Hematology", "SSN": 339070138}, {"EmployeeID": 31, "Name": "Denton Long", "Position": "Rheumatology", "SSN": 209841455}, {"EmployeeID": 32, "Name": "Ashrodit Mosley", "Position": "Pulmonology", "SSN": 294261612}, {"EmployeeID": 33, "Name": "Brady Leach", "Position": "Rheumatology", "SSN": 206378777}, {"EmployeeID": 34, "Name": "Jerry Hack", "Position": "Oncology", "SSN": 103416281}, {"EmployeeID": 35, "Name": "Aladdin Bradford", "Position": "Pulmonology", "SSN": 247004018}, {"EmployeeID": 36, "Name": "Serina Gordon", "Position": "Phrenology", "SSN": 184950289}, {"EmployeeID": 37, "Name": "Stacy Diaz", "Position": "Infectious Diseases", "SSN": 979516208}, {"EmployeeID": 38, "Name": "Brett Lancaster", "Position": "Rheumatology", "SSN": 316247062}, {"EmployeeID": 39, "Name": "Dane Burnett", "Position": "Hematology", "SSN": 446033782}, {"EmployeeID": 40, "Name": "Ramona Best", "Position": "Hematology", "SSN": 409422418}, {"EmployeeID": 41, "Name": "Leonard Neuf", "Position": "Cardiology", "SSN": 393015047}, {"EmployeeID": 42, "Name": "Josiah Wilson", "Position": "Hematology", "SSN": 828120653}, {"EmployeeID": 43, "Name": "Faith Carroll", "Position": "Hematology", "SSN": 69353325}, {"EmployeeID": 44, "Name": "Bradley Rios", "Position": "Rheumatology", "SSN": 469845076}, {"EmployeeID": 45, "Name": "Ulla Farmer", "Position": "Oncology", "SSN": 884506463}, {"EmployeeID": 46, "Name": "Lucius Sawyer", "Position": "Phrenology", "SSN": 839908841}, {"EmployeeID": 47, "Name": "Wylie Knox", "Position": "Gastroenterology", "SSN": 677318007}, {"EmployeeID": 48, "Name": "Zachary Bentley", "Position": "Hematology", "SSN": 485887892}, {"EmployeeID": 49, "Name": "Morgan Carr", "Position": "Phrenology", "SSN": 689913553}, {"EmployeeID": 50, "Name": "Darryl Garrison", "Position": "Cardiology", "SSN": 216797526}, {"EmployeeID": 51, "Name": "Lana Neal", "Position": "Family Physician", "SSN": 416532210}, {"EmployeeID": 52, "Name": "Sebastian Bradford", "Position": "Phrenology", "SSN": 793565986}, {"EmployeeID": 53, "Name": "Kerry Hutchinson", "Position": "Cardiology", "SSN": 107477676}, {"EmployeeID": 54, "Name": "Quincy Odum", "Position": "Endocrinology", "SSN": 520893086}, {"EmployeeID": 55, "Name": "Demetrius Sims", "Position": "Rheumatology", "SSN": 347430066}, {"EmployeeID": 56, "Name": "Cullen Perez", "Position": "Hematology", "SSN": 898286199}, {"EmployeeID": 57, "Name": "Zeph Levy", "Position": "Phrenology", "SSN": 709820841}, {"EmployeeID": 58, "Name": "Cora Simmons", "Position": "Pulmonology", "SSN": 379811491}, {"EmployeeID": 59, "Name": "Ursula Simon", "Position": "Pulmonology", "SSN": 556282776}, {"EmployeeID": 60, "Name": "Priscilla West", "Position": "Cardiology", "SSN": 82071712}, {"EmployeeID": 61, "Name": "Linda Andrew", "Position": "Endocrinology", "SSN": 46341500}, {"EmployeeID": 62, "Name": "Patrick Kramer", "Position": "Rheumatology", "SSN": 147338779}, {"EmployeeID": 63, "Name": "Nathan Wolfe", "Position": "Rheumatology", "SSN": 28172739}, {"EmployeeID": 64, "Name": "Brody Strong", "Position": "Pulmonology", "SSN": 352013789}, {"EmployeeID": 65, "Name": "Edward Gallegos", "Position": "Phrenology", "SSN": 65880866}, {"EmployeeID": 66, "Name": "Forrest Skinner", "Position": "Oncology", "SSN": 453791181}, {"EmployeeID": 67, "Name": "Talin Alvarado", "Position": "Gastroenterology", "SSN": 435707371}, {"EmployeeID": 68, "Name": "Debra Goodwin", "Position": "Endocrinology", "SSN": 505720294}, {"EmployeeID": 69, "Name": "Brenna Rivers", "Position": "Gastroenterology", "SSN": 217548885}, {"EmployeeID": 70, "Name": "Bertha Alvarado", "Position": "Oncology", "SSN": 749899207}, {"EmployeeID": 71, "Name": "Dawn Reilly", "Position": "Hematology", "SSN": 694715211}, {"EmployeeID": 72, "Name": "Ann Glenn", "Position": "Rheumatology", "SSN": 949374087}, {"EmployeeID": 73, "Name": "Carly Jones", "Position": "Cardiology", "SSN": 848052361}, {"EmployeeID": 74, "Name": "Karielsh Haynes", "Position": "Phrenology", "SSN": 62488741}, {"EmployeeID": 75, "Name": "Ian Snider", "Position": "Cardiology", "SSN": 38771015}, {"EmployeeID": 76, "Name": "Jessica Montoya", "Position": "Oncology", "SSN": 889175566}, {"EmployeeID": 77, "Name": "Elliana Sweeney", "Position": "Rheumatology", "SSN": 146714555}, {"EmployeeID": 78, "Name": "Cameron Collins", "Position": "Endocrinology", "SSN": 408629477}, {"EmployeeID": 79, "Name": "Lyndra Sweeney", "Position": "Phrenology", "SSN": 525934213}, {"EmployeeID": 80, "Name": "Carol Justice", "Position": "Oncology", "SSN": 135652803}, {"EmployeeID": 81, "Name": "Sylvia Maynard", "Position": "Pulmonology", "SSN": 356065359}, {"EmployeeID": 82, "Name": "Geoffrey Richardson", "Position": "Rheumatology", "SSN": 765057284}, {"EmployeeID": 83, "Name": "Alan Benson", "Position": "Hematology", "SSN": 345844438}, {"EmployeeID": 84, "Name": "Martina Wilcox", "Position": "Pulmonology", "SSN": 156113484}, {"EmployeeID": 85, "Name": "Ashlan Farrell", "Position": "Rheumatology", "SSN": 506430973}, {"EmployeeID": 86, "Name": "Rebekah Clay", "Position": "Rheumatology", "SSN": 865663871}, {"EmployeeID": 87, "Name": "Igor Harmon", "Position": "Gastroenterology", "SSN": 242114546}, {"EmployeeID": 88, "Name": "Jeannette Dickerson", "Position": "Infectious Diseases", "SSN": 299357284}, {"EmployeeID": 89, "Name": "Odessa Dillard", "Position": "Hematology", "SSN": 88461343}, {"EmployeeID": 90, "Name": "Brooke Horn", "Position": "Oncology", "SSN": 821818593}, {"EmployeeID": 91, "Name": "Gay Byers", "Position": "Endocrinology", "SSN": 886489569}, {"EmployeeID": 92, "Name": "Keely Lott", "Position": "Pulmonology", "SSN": 450464377}, {"EmployeeID": 93, "Name": "Orson Franco", "Position": "Rheumatology", "SSN": 37172668}, {"EmployeeID": 94, "Name": "Joel Warner", "Position": "Rheumatology", "SSN": 41616151}, {"EmployeeID": 95, "Name": "Armando Johnston", "Position": "Pulmonology", "SSN": 875624549}, {"EmployeeID": 96, "Name": "Jayme Mendoza", "Position": "Rheumatology", "SSN": 579981367}, {"EmployeeID": 97, "Name": "Paki Oliver", "Position": "Rheumatology", "SSN": 745419330}, {"EmployeeID": 98, "Name": "Dexter Garcia", "Position": "Gastroenterology", "SSN": 346846355}, {"EmployeeID": 99, "Name": "Chae Alston", "Position": "Pulmonology", "SSN": 955243738}, {"EmployeeID": 100, "Name": "Chester Velez", "Position": "Gastroenterology", "SSN": 145312883}, {"EmployeeID": 101, "Name": "Elliana O. Bond", "Position": "Vascular Surgery", "SSN": 599257020}, {"EmployeeID": 102, "Name": "Joshua Acevedo", "Position": "Pulmonology", "SSN": 672301146}, {"EmployeeID": 103, "Name": "Ivory Z. Albert", "Position": "Pediatric Surgery", "SSN": 206182960}, {"EmployeeID": 104, "Name": "Harold O. Lancaster", "Position": "Obstetrician", "SSN": 606784593}, {"EmployeeID": 105, "Name": "Macon V. Mercer", "Position": "Obstetrician", "SSN": 525063910}, {"EmployeeID": 106, "Name": "Ursula E. Parks", "Position": "Rheumatology", "SSN": 606999507}, {"EmployeeID": 107, "Name": "Mash T. Sloan", "Position": "Rheumatology", "SSN": 37712452}, {"EmployeeID": 108, "Name": "Marah C. Rojas", "Position": "Family Physician", "SSN": 473931945}, {"EmployeeID": 109, "Name": "Forrest D. Leblanc", "Position": "Family Physician", "SSN": 597331433}, {"EmployeeID": 110, "Name": "Taylor R. Armstrong", "Position": "Pulmonology", "SSN": 768728078}, {"EmployeeID": 111, "Name": "Zephania Finch", "Position": "Obstetrician", "SSN": 164640861}, {"EmployeeID": 112, "Name": "Mamix Nash", "Position": "Rheumatology", "SSN": 38483713}, {"EmployeeID": 113, "Name": "Acton H. Benton", "Position": "Surgical Oncology", "SSN": 350873439}, {"EmployeeID": 114, "Name": "Bernard Sheppard", "Position": "Family Physician", "SSN": 879325318}, {"EmployeeID": 115, "Name": "Dexter Spears", "Position": "Pediatric Surgery", "SSN": 858889697}, {"EmployeeID": 116, "Name": "Yeo L. Workman", "Position": "Family Physician", "SSN": 758360853}, {"EmployeeID": 117, "Name": "Seth M. Farrell", "Position": "Vascular Surgery", "SSN": 237384421}, {"EmployeeID": 118, "Name": "Merrill D. Delacruz", "Position": "Dermatologist", "SSN": 893447030}, {"EmployeeID": 119, "Name": "Kelsey Baldwin", "Position": "Obstetrician", "SSN": 498036805}, {"EmployeeID": 120, "Name": "Dolan Bradford", "Position": "Surgical Oncology", "SSN": 428631012}, {"EmployeeID": 121, "Name": "Akeem Matthews", "Position": "Rheumatology", "SSN": 339216750}, {"EmployeeID": 122, "Name": "Daniel Huber", "Position": "Vascular Surgery", "SSN": 679297715}, {"EmployeeID": 123, "Name": "Allistair S. Bowman", "Position": "Family Physician", "SSN": 768162497}, {"EmployeeID": 124, "Name": "Stuart Helman", "Position": "Family Physician", "SSN": 644807390}, {"EmployeeID": 125, "Name": "Zenaida T. Becker", "Position": "Vascular Surgery", "SSN": 202136668}, {"EmployeeID": 126, "Name": "Simon Cohen", "Position": "Pulmonology", "SSN": 714706077}, {"EmployeeID": 127, "Name": "Alec M. Bishop", "Position": "Otolaryngologist", "SSN": 744843511}, {"EmployeeID": 128, "Name": "Elton S. Albert", "Position": "Pediatric Surgery", "SSN": 735189957}, {"EmployeeID": 129, "Name": "Rhea M. Duran", "Position": "Family Physician", "SSN": 542362416}, {"EmployeeID": 130, "Name": "Gastor Talley", "Position": "Obstetrician", "SSN": 304519020}, {"EmployeeID": 131, "Name": "Camden O. Peck", "Position": "Vascular Surgery", "SSN": 566096016}, {"EmployeeID": 132, "Name": "Aileen D. Hopkins", "Position": "Rheumatology", "SSN": 607908980}, {"EmployeeID": 133, "Name": "Rylee O. Davenport", "Position": "Pediatric Surgery", "SSN": 295777481}, {"EmployeeID": 134, "Name": "Cassius M. Wagner", "Position": "Obstetrician", "SSN": 57735189}, {"EmployeeID": 135, "Name": "Nissin Duran", "Position": "Surgical Oncology", "SSN": 293505422}, {"EmployeeID": 136, "Name": "Bruck Ballard", "Position": "Pediatric Surgery", "SSN": 357841034}, {"EmployeeID": 137, "Name": "Troy H. Johns", "Position": "Dermatologist", "SSN": 995956084}, {"EmployeeID": 138, "Name": "Signe Barker", "Position": "Pediatric Surgery", "SSN": 782367319}, {"EmployeeID": 139, "Name": "Jeannette I. Battle", "Position": "Surgical Oncology", "SSN": 954525943}, {"EmployeeID": 140, "Name": "Anthony T. Payne", "Position": "Otolaryngologist", "SSN": 678225964}, {"EmployeeID": 141, "Name": "Barbara U. Vaughn", "Position": "Otolaryngologist", "SSN": 57702465}, {"EmployeeID": 142, "Name": "Robby Hillis", "Position": "Otolaryngologist", "SSN": 248186254}, {"EmployeeID": 143, "Name": "Abdul Gili", "Position": "Obstetrician", "SSN": 167795380}, {"EmployeeID": 144, "Name": "Garrett F. Moon", "Position": "Family Physician", "SSN": 350387520}, {"EmployeeID": 145, "Name": "Lucius I. Sexton", "Position": "Obstetrician", "SSN": 666285854}, {"EmployeeID": 146, "Name": "Latifah Myers", "Position": "Rheumatology", "SSN": 373629795}, {"EmployeeID": 147, "Name": "Ciara T. Vaughn", "Position": "Pediatric Surgery", "SSN": 163586895}, {"EmployeeID": 148, "Name": "Isaac Short", "Position": "Hematology", "SSN": 825906073}, {"EmployeeID": 149, "Name": "Lucas Z. Hogan", "Position": "Vascular Surgery", "SSN": 415186660}, {"EmployeeID": 150, "Name": "Sybill O. Peterson", "Position": "Family Physician", "SSN": 665796523}, {"EmployeeID": 151, "Name": "Carlos Weaver", "Position": "Pediatric Surgery", "SSN": 684931244}, {"EmployeeID": 152, "Name": "Bassia G. Hopkins", "Position": "Otolaryngologist", "SSN": 226085014}, {"EmployeeID": 153, "Name": "Sage P. Estes", "Position": "Vascular Surgery", "SSN": 256829231}, {"EmployeeID": 154, "Name": "Dermot Rogers", "Position": "Family Physician", "SSN": 916665870}, {"EmployeeID": 155, "Name": "Penelope T. Russo", "Position": "Surgical Oncology", "SSN": 529145232}, {"EmployeeID": 156, "Name": "August Carpenter", "Position": "Vascular Surgery", "SSN": 836770981}, {"EmployeeID": 157, "Name": "Anthony Dickerson", "Position": "Rheumatology", "SSN": 176997180}, {"EmployeeID": 158, "Name": "Deacon D. Willis", "Position": "Dermatologist", "SSN": 115042776}, {"EmployeeID": 159, "Name": "Hedda Franco", "Position": "Vascular Surgery", "SSN": 679597683}, {"EmployeeID": 160, "Name": "Palmer B. Maxwell", "Position": "Obstetrician", "SSN": 80792823}, {"EmployeeID": 161, "Name": "Naomi O. Dixon", "Position": "Dermatologist", "SSN": 68310807}, {"EmployeeID": 162, "Name": "Ferris O. Frazier", "Position": "Family Physician", "SSN": 994785963}]]
```

```
POSTMAN
mag8 22:37
No Environment
Send Save
localhost:5000/doctors
GET localhost:5000/doctors
Params Authorization Headers (0) Body Pre-request Script Tests Settings
KEY VALUE DESCRIPTION
Key Value Description
Body Cookies Headers (0) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "EmployeeID": 1,
3   "Name": "Naïda Schultz",
4   "Position": "Cardiology",
5   "SSN": 375019270
6 },
7 {
8   "EmployeeID": 2,
9   "Name": "Tate Armstrong",
10  "Position": "Phrenology",
11  "SSN": 951100813
12 },
13 {
14   "EmployeeID": 3,
15   "Name": "Eloa Kane",
16   "Position": "Cardiology",
17   "SSN": 544043010
18 },
19 {
20   "EmployeeID": 4,
21   "Name": "Nasim McIntyre",
22   "Position": "Gastroenterology",
23   "SSN": 244599760
24 },
25 {
26   "EmployeeID": 5,
27   "Name": "Kadeem Langley",
28   "Position": "Hematology",
29   "SSN": 994785963
30 },
31 {
32   "EmployeeID": 6,
33   "Name": "Grace Wolf",
34   "Position": "Pulmonology",
35   "SSN": 74007250
36 },
37 {
38   "EmployeeID": 7,
39   "Name": "Patience Salas",
40   "Position": "Hematology",
41   "SSN": 51737684
42 },
43 {
44   "EmployeeID": 8,
45   "Name": "Hilda Shields",
46   "Position": "Gastroenterology",
47   "SSN": 203133100
48 },
49 {
50   "EmployeeID": 9,
51   "Name": "Lionel Sanchez",
52   "Position": "Gastroenterology",
53   "SSN": 954042600
54 },
55 {
56   "EmployeeID": 10,
57   "Name": "Caleb Larson",
58   "Position": "Cardiology",
59   "SSN": 895978419
60 },
61 {
62   "EmployeeID": 11,
63   "Name": "Isabelle Salinas",
64   "Position": "Phrenology",
65   "SSN": 998509517
66 },
67 {
68   "EmployeeID": 12,
69   "Name": "Noble Franks",
70   "Position": "Infectious Diseases",
71   "SSN": 804141722
72 },
73 {
74   "EmployeeID": 13,
75   "Name": "Jerry Ward",
76   "Position": "Endocrinology",
77   "SSN": 338329095
78 },
79 {
80   "EmployeeID": 14,
81   "Name": "Frank Pierce",
82   "Position": "Pulmonology",
83   "SSN": 734040343
84 },
85 {
86   "EmployeeID": 15,
87   "Name": "Moyt Stafford",
88   "Position": "Hematology",
89   "SSN": 775007760
90 },
91 {
92   "EmployeeID": 16,
93   "Name": "Lucian Briggs",
94   "Position": "Hematology",
95   "SSN": 534086299
96 },
97 {
98   "EmployeeID": 17,
99   "Name": "Joshua Silva",
100  "Position": "Pulmonology",
101  "SSN": 513760923
102 },
103 {
104   "EmployeeID": 18,
105   "Name": "Milyari Atkinson",
106   "Position": "Cardiology",
107   "SSN": 522823713
108 },
109 {
110   "EmployeeID": 19,
111   "Name": "Hu Hines",
112   "Position": "Cardiology",
113   "SSN": 614175233
114 },
115 {
116   "EmployeeID": 20,
117   "Name": "Clark Emerson",
118   "Position": "Pulmonology",
119   "SSN": 273991165
120 },
121 {
122   "EmployeeID": 21,
123   "Name": "Arsenio Mejia",
124   "Position": "Phrenology",
125   "SSN": 102887967
126 },
127 {
128   "EmployeeID": 22,
129   "Name": "Charissa Gallegos",
130   "Position": "Oncology",
131   "SSN": 75030631
132 },
133 {
134   "EmployeeID": 23,
135   "Name": "Sydney Wiggins",
136   "Position": "Endocrinology",
137   "SSN": 522345623
138 },
139 {
140   "EmployeeID": 24,
141   "Name": "Moses Blanchard",
142   "Position": "Oncology",
143   "SSN": 299206368
144 },
145 {
146   "EmployeeID": 25,
147   "Name": "Dillon Livingston",
148   "Position": "Cardiology",
149   "SSN": 963487947
150 },
151 {
152   "EmployeeID": 26,
153   "Name": "Dalton Calderon",
154   "Position": "Family Physician",
155   "SSN": 80277655
156 },
157 {
158   "EmployeeID": 27,
159   "Name": "Nasoa Schmidt",
160   "Position": "Pulmonology",
161   "SSN": 357002532
162 },
163 {
164   "EmployeeID": 28,
165   "Name": "Ifeoma Murphy",
166   "Position": "Phrenology",
167   "SSN": 743235384
168 },
169 {
170   "EmployeeID": 29,
171   "Name": "Jacob Bryan",
172   "Position": "Rheumatology",
173   "SSN": 383791023
174 },
175 {
176   "EmployeeID": 30,
177   "Name": "Moses Warner",
178   "Position": "Hematology",
179   "SSN": 339070138
180 },
181 {
182   "EmployeeID": 31,
183   "Name": "Denton Long",
184   "Position": "Rheumatology",
185   "SSN": 209841455
186 },
187 {
188   "EmployeeID": 32,
189   "Name": "Ashrodit Mosley",
190   "Position": "Pulmonology",
191   "SSN": 294261612
192 },
193 {
194   "EmployeeID": 33,
195   "Name": "Brady Leach",
196   "Position": "Rheumatology",
197   "SSN": 206378777
198 },
199 {
200   "EmployeeID": 34,
201   "Name": "Jerry Hack",
202   "Position": "Oncology",
203   "SSN": 103416281
204 },
205 {
206   "EmployeeID": 35,
207   "Name": "Aladdin Bradford",
208   "Position": "Pulmonology",
209   "SSN": 247004018
210 },
211 {
212   "EmployeeID": 36,
213   "Name": "Serina Gordon",
214   "Position": "Phrenology",
215   "SSN": 184950289
216 },
217 {
218   "EmployeeID": 37,
219   "Name": "Stacy Diaz",
220   "Position": "Infectious Diseases",
221   "SSN": 979516208
222 },
223 {
224   "EmployeeID": 38,
225   "Name": "Brett Lancaster",
226   "Position": "Rheumatology",
227   "SSN": 316247062
228 },
229 {
230   "EmployeeID": 39,
231   "Name": "Dane Burnett",
232   "Position": "Hematology",
233   "SSN": 446033782
234 },
235 {
236   "EmployeeID": 40,
237   "Name": "Ramona Best",
238   "Position": "Hematology",
239   "SSN": 409422418
240 },
241 {
242   "EmployeeID": 41,
243   "Name": "Leonard Neuf",
244   "Position": "Cardiology",
245   "SSN": 393015047
246 },
247 {
248   "EmployeeID": 42,
249   "Name": "Josiah Wilson",
250   "Position": "Hematology",
251   "SSN": 828120653
252 },
253 {
254   "EmployeeID": 43,
255   "Name": "Faith Carroll",
256   "Position": "Hematology",
257   "SSN": 69353325
258 },
259 {
260   "EmployeeID": 44,
261   "Name": "Bradley Rios",
262   "Position": "Rheumatology",
263   "SSN": 469845076
264 },
265 {
266   "EmployeeID": 45,
267   "Name": "Ulla Farmer",
268   "Position": "Oncology",
269   "SSN": 884506463
270 },
271 {
272   "EmployeeID": 46,
273   "Name": "Lucius Sawyer",
274   "Position": "Phrenology",
275   "SSN": 839908841
276 },
277 {
278   "EmployeeID": 47,
279   "Name": "Wylie Knox",
280   "Position": "Gastroenterology",
281   "SSN": 677318007
282 },
283 {
284   "EmployeeID": 48,
285   "Name": "Zachary Bentley",
286   "Position": "Hematology",
287   "SSN": 485887892
288 },
289 {
290   "EmployeeID": 49,
291   "Name": "Morgan Carr",
292   "Position": "Phrenology",
293   "SSN": 689913553
294 },
295 {
296   "EmployeeID": 50,
297   "Name": "Darryl Garrison",
298   "Position": "Cardiology",
299   "SSN": 216797526
300 },
301 {
302   "EmployeeID": 51,
303   "Name": "Lana Neal",
304   "Position": "Family Physician",
305   "SSN": 416532210
306 },
307 {
308   "EmployeeID": 52,
309   "Name": "Sebastian Bradford",
310   "Position": "Phrenology",
311   "SSN": 793565986
312 },
313 {
314   "EmployeeID": 53,
315   "Name": "Kerry Hutchinson",
316   "Position": "Cardiology",
317   "SSN": 107477676
318 },
319 {
320   "EmployeeID": 54,
321   "Name": "Quincy Odum",
322   "Position": "Endocrinology",
323   "SSN": 520893086
324 },
325 {
326   "EmployeeID": 55,
327   "Name": "Demetrius Sims",
328   "Position": "Rheumatology",
329   "SSN": 347430066
330 },
331 {
332   "EmployeeID": 56,
333   "Name": "Cullen Perez",
334   "Position": "Hematology",
335   "SSN": 898286199
336 },
337 {
338   "EmployeeID": 57,
339   "Name": "Zeph Levy",
340   "Position": "Phrenology",
341   "SSN": 709820841
342 },
343 {
344   "EmployeeID": 58,
345   "Name": "Cora Simmons",
346   "Position": "Pulmonology",
347   "SSN": 379811491
348 },
349 {
350   "EmployeeID": 59,
351   "Name": "Ursula Simon",
352   "Position": "Pulmonology",
353   "SSN": 556282776
354 },
355 {
356   "EmployeeID": 60,
357   "Name": "Priscilla West",
358   "Position": "Cardiology",
359   "SSN": 82071712
360 },
361 {
362   "EmployeeID": 61,
363   "Name": "Linda Andrew",
364   "Position": "Endocrinology",
365   "SSN": 46341500
366 },
367 {
368   "EmployeeID": 62,
369   "Name": "Patrick Kramer",
370   "Position": "Rheumatology",
371   "SSN": 147338779
372 },
373 {
374   "EmployeeID": 63,
375   "Name": "Nathan Wolfe",
376   "Position": "Rheumatology",
377   "SSN": 28172739
378 },
379 {
380   "EmployeeID": 64,
381   "Name": "Brody Strong",
382   "Position": "Pulmonology",
383   "SSN": 352013789
384 },
385 {
386   "EmployeeID": 65,
387   "Name": "Edward Gallegos",
388   "Position": "Phrenology",
389   "SSN": 65880866
390 },
391 {
392   "EmployeeID": 66,
393   "Name": "Forrest Skinner",
394   "Position": "Oncology",
395   "SSN": 453791181
396 },
397 {
398   "EmployeeID": 67,
399   "Name": "Talin Alvarado",
400   "Position": "Gastroenterology",
401   "SSN": 435707371
402 },
403 {
404   "EmployeeID": 68,
405   "Name": "Debra Goodwin",
406   "Position": "Endocrinology",
407   "SSN": 505720294
408 },
409 {
410   "EmployeeID": 69,
411   "Name": "Brenna Rivers",
412   "Position": "Gastroenterology",
413   "SSN": 217548885
414 },
415 {
416   "EmployeeID": 70,
417   "Name": "Bertha Alvarado",
418   "Position": "Oncology",
419   "SSN": 749899207
420 },
421 {
422   "EmployeeID": 71,
423   "Name": "Dawn Reilly",
424   "Position": "Hematology",
425   "SSN": 694715211
426 },
427 {
428   "EmployeeID": 72,
429   "Name": "Ann Glenn",
430   "Position": "Rheumatology",
431   "SSN": 949374087
432 },
433 {
434   "EmployeeID": 73,
435   "Name": "Carly Jones",
436   "Position": "Cardiology",
437   "SSN": 848052361
438 },
439 {
440   "EmployeeID": 74,
441   "Name": "Karielsh Haynes",
442   "Position": "Phrenology",
443   "SSN": 62488741
444 },
445 {
446   "EmployeeID": 75,
447   "Name": "Ian Snider",
448   "Position": "Cardiology",
449   "SSN": 38771015
450 },
451 {
452   "EmployeeID": 76,
453   "Name": "Jessica Montoya",
454   "Position": "Oncology",
455   "SSN": 889175566
456 },
457 {
458   "EmployeeID": 77,
459   "Name": "Elliana Sweeney",
460   "Position": "Rheumatology",
461   "SSN": 146714555
462 },
463 {
464   "EmployeeID": 78,
465   "Name": "Cameron Collins",
466   "Position": "Endocrinology",
467   "SSN": 408629477
468 },
469 {
470   "EmployeeID": 79,
471   "Name": "Lyndra Sweeney",
472   "Position": "Phrenology",
473   "SSN": 525934213
474 },
475 {
476   "EmployeeID": 80,
477   "Name": "Carol Justice",
478   "Position": "Oncology",
479   "SSN": 135652803
480 },
481 {
482   "EmployeeID": 81,
483   "Name": "Sylvia Maynard",
484   "Position": "Pulmonology",
485   "SSN": 356065359
486 },
487 {
488   "EmployeeID": 82,
489   "Name": "Geoffrey Richardson",
490   "Position": "Rheumatology",
491   "SSN": 765057284
492 },
493 {
494   "EmployeeID": 83,
495   "Name": "Alan Benson",
496   "Position": "Hematology",
497   "SSN": 345844438
498 },
499 {
500   "EmployeeID": 84,
501   "Name": "Martina Wilcox",
502   "Position": "Pulmonology",
503   "SSN": 15611348
```