CS 6360 Project Phase 3

1. All tables are normalized into their third Normal Form
2. Chaitrik………….Deeni chesey ra
3. CREATE TABLE Treatment (

Treatment\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Description VARCHAR(300),

Start\_Date DATE,

End\_Date DATE,

Outcome VARCHAR(20)

);

CREATE TABLE Schedule (

Schedule\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Day VARCHAR(10),

Opening\_Time TIME,

Closing\_Time TIME

);

CREATE TABLE Prescription (

Prescription\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Dosage VARCHAR(10),

Refill\_Information VARCHAR(10),

Prescription\_Date DATE

);

CREATE TABLE Pharmacy (

Pharmacy\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Pharmacy\_Name VARCHAR(50),

Pharmacy\_Address VARCHAR(200),

Business\_Ph\_no VARCHAR(10)

);

CREATE TABLE Person (

P\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Person\_BDate DATE,

First VARCHAR(30),

Middle VARCHAR(30),

Last VARCHAR(30),

Street VARCHAR(100),

City VARCHAR(30),

State VARCHAR(50),

ZIP INT,

Person\_Gender VARCHAR(30),

Age INT

);

CREATE TABLE Healthcare\_Employee (

Healthcare\_Employee\_ID INT AUTO\_INCREMENT,

P\_ID INT,

Specialization VARCHAR(50),

Department VARCHAR(30),

Contact\_Information VARCHAR(50),

Working\_Hours DECIMAL(10,2),

Years\_of\_Experience DECIMAL(10,2),

PRIMARY KEY (Healthcare\_Employee\_ID, P\_ID),

FOREIGN KEY (P\_ID) REFERENCES Person(P\_ID)

);

CREATE TABLE Patient (

Patient\_ID INT AUTO\_INCREMENT,

P\_ID INT,

Contact\_Number VARCHAR(100),

Email\_ID VARCHAR(50),

PRIMARY KEY (Patient\_ID, P\_ID),

FOREIGN KEY (P\_ID) REFERENCES Person(P\_ID)

);

CREATE TABLE Billing\_Information (

Billing\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Date DATE,

Patient\_ID INT,

Hospital\_Stays VARCHAR(300),

Procedures VARCHAR(100),

Total\_Cost DECIMAL(10,2),

Installment\_Plans VARCHAR(100),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE Payment\_Method (

Payment\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Billing\_ID INT,

Insurance\_ID INT,

Insurance\_Company\_Name VARCHAR(50),

Amount DECIMAL(10,2),

FOREIGN KEY (Billing\_ID) REFERENCES Billing\_Information(Billing\_ID),

UNIQUE (Payment\_ID, Billing\_ID)

);

CREATE TABLE Student (

S1 INT AUTO\_INCREMENT PRIMARY KEY

);

CREATE TABLE Instructor (

I1 INT AUTO\_INCREMENT PRIMARY KEY

);

CREATE TABLE Nurses (

Healthcare\_Employee\_ID INT,

PRIMARY KEY (Healthcare\_Employee\_ID),

FOREIGN KEY (Healthcare\_Employee\_ID) REFERENCES Healthcare\_Employee(Healthcare\_Employee\_ID)

);

CREATE TABLE Administrative\_Staff (

Healthcare\_Employee\_ID INT,

PRIMARY KEY (Healthcare\_Employee\_ID),

FOREIGN KEY (Healthcare\_Employee\_ID) REFERENCES Healthcare\_Employee(Healthcare\_Employee\_ID)

);

CREATE TABLE Doctor (

Healthcare\_Employee\_ID INT,

PRIMARY KEY (Healthcare\_Employee\_ID),

FOREIGN KEY (Healthcare\_Employee\_ID) REFERENCES Healthcare\_Employee(Healthcare\_Employee\_ID)

);

CREATE TABLE Medical\_Record (

Medical\_Record\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Patient\_ID INT,

Current\_Medication VARCHAR(50),

Past\_Diagnosis VARCHAR(50),

Treatment\_History VARCHAR(500),

Lab\_Results VARCHAR(50),

Allergies VARCHAR(50),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE Comments (

Comment\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Pharmacy\_ID INT,

Patient\_ID INT,

Comment\_Contents VARCHAR(100),

Rating\_Score INT,

FOREIGN KEY (Pharmacy\_ID) REFERENCES Pharmacy(Pharmacy\_ID),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE Inventory\_System (

Inventory\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Pharmacy\_ID INT,

Medication\_Name VARCHAR(100),

Stock\_Level INT,

Supplier\_Information VARCHAR(100),

FOREIGN KEY (Pharmacy\_ID) REFERENCES Pharmacy(Pharmacy\_ID)

);

CREATE TABLE Medication (

Medication\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Pharmacy\_ID INT,

Medication\_Name VARCHAR(100),

FOREIGN KEY (Pharmacy\_ID) REFERENCES Pharmacy(Pharmacy\_ID)

);

CREATE TABLE Specifies (

Specifies\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Prescription\_ID INT,

Medication\_ID INT,

FOREIGN KEY (Prescription\_ID) REFERENCES Prescription(Prescription\_ID),

FOREIGN KEY (Medication\_ID) REFERENCES Medication(Medication\_ID)

);

CREATE TABLE Sell (

Sell\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Medication\_ID INT,

Pharmacy\_ID INT,

Price DECIMAL(10,2),

FOREIGN KEY (Medication\_ID) REFERENCES Medication(Medication\_ID),

FOREIGN KEY (Pharmacy\_ID) REFERENCES Pharmacy(Pharmacy\_ID)

);

CREATE TABLE Provides (

Provides\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Healthcare\_Employee\_ID INT,

P\_ID INT,

Treatment\_ID INT,

FOREIGN KEY (Healthcare\_Employee\_ID, P\_ID) REFERENCES Healthcare\_Employee(Healthcare\_Employee\_ID, P\_ID),

FOREIGN KEY (Treatment\_ID) REFERENCES Treatment(Treatment\_ID)

);

CREATE TABLE Gets (

Gets\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Patient\_ID INT,

Treatment\_ID INT,

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID),

FOREIGN KEY (Treatment\_ID) REFERENCES Treatment(Treatment\_ID)

);

CREATE TABLE HasBill (

HasBill\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Billing\_ID INT,

Patient\_ID INT,

FOREIGN KEY (Billing\_ID) REFERENCES Billing\_Information(Billing\_ID),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE ModeofPayment (

ModeofPayment\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Billing\_ID INT,

Payment\_ID INT,

FOREIGN KEY (Billing\_ID) REFERENCES Billing\_Information(Billing\_ID),

FOREIGN KEY (Payment\_ID) REFERENCES Payment\_Method(Payment\_ID)

);

CREATE TABLE Appointment (

Appointment\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Patient\_ID INT,

Healthcare\_Employee\_ID INT,

Date DATE,

Purpose VARCHAR(100),

Time TIME,

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID),

FOREIGN KEY (Healthcare\_Employee\_ID) REFERENCES Healthcare\_Employee(Healthcare\_Employee\_ID)

);

CREATE TABLE Administers (

Administers\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Treatment\_ID INT,

Prescription\_ID INT,

FOREIGN KEY (Treatment\_ID) REFERENCES Treatment(Treatment\_ID),

FOREIGN KEY (Prescription\_ID) REFERENCES Prescription(Prescription\_ID)

);

CREATE TABLE Receives (

Receives\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Patient\_ID INT,

Prescription\_ID INT,

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID),

FOREIGN KEY (Prescription\_ID) REFERENCES Prescription(Prescription\_ID)

);

CREATE TABLE M1\_Person (

M1\_Person\_ID INT AUTO\_INCREMENT PRIMARY KEY,

P\_ID INT,

P\_Mobile\_Number VARCHAR(15),

FOREIGN KEY (P\_ID) REFERENCES Person(P\_ID)

);

CREATE TABLE M1\_Patient (

M1\_Patient\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Patient\_ID INT,

Contact\_Number VARCHAR(100),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE M2\_Patient (

M2\_Patient\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Patient\_ID INT,

Email\_ID VARCHAR(50),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE M1\_Billing\_Information (

Billing\_ID INT AUTO\_INCREMENT PRIMARY KEY NOT NULL,

Date DATE,

Patient\_ID INT,

Hospital\_Stays VARCHAR(300),

Procedures VARCHAR(100),

Total\_Cost DECIMAL(10,2),

Installment\_Plans VARCHAR(100),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

CREATE TABLE M1\_Medical\_Record (

M1\_Medical\_Record\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Medical\_Record\_ID INT,

Lab\_Results VARCHAR(50),

FOREIGN KEY (Medical\_Record\_ID) REFERENCES Medical\_Record(Medical\_Record\_ID)

);

CREATE TABLE M2\_Medical\_Record (

M2\_Medical\_Record\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Medical\_Record\_ID INT,

Allergies VARCHAR(50),

FOREIGN KEY (Medical\_Record\_ID) REFERENCES Medical\_Record(Medical\_Record\_ID)

);

CREATE TABLE Working\_Hours (

Working\_Hours\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Schedule\_ID INT,

Healthcare\_Employee\_ID INT,

Pharmacy\_ID INT,

FOREIGN KEY (Schedule\_ID) REFERENCES Schedule(Schedule\_ID),

FOREIGN KEY (Healthcare\_Employee\_ID) REFERENCES Healthcare\_Employee(Healthcare\_Employee\_ID),

FOREIGN KEY (Pharmacy\_ID) REFERENCES Pharmacy(Pharmacy\_ID)

);

CREATE TABLE TrainingCenter (

Training\_Center\_ID INT AUTO\_INCREMENT PRIMARY KEY

);

CREATE TABLE Courses (

Course\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Training\_Center\_ID INT,

FOREIGN KEY (Training\_Center\_ID) REFERENCES TrainingCenter(Training\_Center\_ID)

);

CREATE TABLE Enroll (

Enroll\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Course\_ID INT,

Student\_ID INT,

FOREIGN KEY (Course\_ID) REFERENCES Courses(Course\_ID),

FOREIGN KEY (Student\_ID) REFERENCES Student(S1)

);

CREATE TABLE Teaches (

Teaches\_ID INT AUTO\_INCREMENT PRIMARY KEY,

Course\_ID INT,

Instructor\_ID INT,

FOREIGN KEY (Course\_ID) REFERENCES Courses(Course\_ID),

FOREIGN KEY (Instructor\_ID) REFERENCES Instructor(I1)

);

1. 1.

CREATE VIEW VIP\_Patient\_View AS

SELECT p.First, p.Last, p.Person\_BDate AS Enrollment\_Date

FROM Person p

JOIN Patient pat ON p.P\_ID = pat.P\_ID

JOIN Appointment app ON pat.Patient\_ID = app.Patient\_ID

WHERE MONTH(app.Date) = MONTH(CURRENT\_DATE()) - 1

GROUP BY p.First, p.Last, p.Person\_BDate

HAVING COUNT(app.Patient\_ID) > 3;

2.

CREATE VIEW Preferred\_Healthcare\_Professional\_View AS

SELECT p.First, p.Last, he.Specialization, COUNT(app.Patient\_ID) AS Patient\_Interaction

FROM Healthcare\_Employee he

JOIN Person p ON he.P\_ID = p.P\_ID

JOIN Appointment app ON he.Healthcare\_Employee\_ID = app.Healthcare\_Employee\_ID

WHERE app.Date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 2 MONTH) AND CURRENT\_DATE()

GROUP BY p.First, p.Last, he.Specialization

ORDER BY Patient\_Interaction DESC;

3.

CREATE VIEW Critical\_Treatment\_View AS

SELECT t.Treatment\_ID, t.Description, COUNT(\*) AS Treatment\_Count

FROM Treatment t

JOIN Gets g ON t.Treatment\_ID = g.Treatment\_ID

JOIN Appointment app ON g.Patient\_ID = app.Patient\_ID

WHERE app.Date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 1 MONTH) AND CURRENT\_DATE()

GROUP BY t.Treatment\_ID, t.Description

HAVING COUNT(\*) > 5;

4.

CREATE VIEW Potential\_VIP\_Patient\_View AS

SELECT p.First, p.Last, p.P\_ID, pat.Contact\_Number, COUNT(\*) AS Visit\_Count

FROM Person p

JOIN Patient pat ON p.P\_ID = pat.P\_ID

JOIN Appointment app ON pat.Patient\_ID = app.Patient\_ID

WHERE app.Date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 2 MONTH) AND CURRENT\_DATE()

GROUP BY p.First, p.Last, p.P\_ID, pat.Contact\_Number

HAVING Visit\_Count > 3;

5.

CREATE VIEW Top\_Staff\_View AS

SELECT p.First, p.Last, he.Specialization, COUNT(\*) AS Processed\_Count

FROM Person p

JOIN Healthcare\_Employee he ON p.P\_ID = he.P\_ID

JOIN Appointment app ON he.Healthcare\_Employee\_ID = app.Healthcare\_Employee\_ID

WHERE app.Date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 1 MONTH) AND CURRENT\_DATE()

GROUP BY p.First, p.Last, he.Specialization

ORDER BY Processed\_Count DESC

LIMIT 1;

e)

1.

SELECT Specialization, COUNT(\*) AS Total\_Count

FROM Healthcare\_Employee

GROUP BY Specialization;

2.

SELECT p.First, p.Last

FROM Person p

JOIN Healthcare\_Employee he ON p.P\_ID = he.P\_ID

JOIN Patient pat ON p.P\_ID = pat.P\_ID;

3.

SELECT AVG(Visit\_Count) AS Average\_Visits

FROM (

SELECT COUNT(\*) AS Visit\_Count

FROM Appointment

WHERE Patient\_ID IN (

SELECT Patient\_ID

FROM (

SELECT Patient\_ID, COUNT(\*) AS Visit\_Count

FROM Appointment

GROUP BY Patient\_ID

ORDER BY Visit\_Count DESC

LIMIT 5

) AS Top\_VIP\_Patients

)

) AS Top\_VIP\_Patient\_Visits;

4. SELECT p.First, p.Last, AVG(c.Rating\_Score) AS Avg\_Rating

FROM Person p

JOIN Healthcare\_Employee he ON p.P\_ID = he.P\_ID

JOIN Appointment app ON he.Healthcare\_Employee\_ID = app.Healthcare\_Employee\_ID

JOIN Comments c ON app.Patient\_ID = c.Patient\_ID

GROUP BY p.First, p.Last

ORDER BY Avg\_Rating DESC

LIMIT 1;

5.

SELECT m.Medication\_Name, COUNT(\*) AS Prescription\_Count

FROM Medication m

JOIN Specifies s ON m.Medication\_ID = s.Medication\_ID

JOIN Prescription p ON s.Prescription\_ID = p.Prescription\_ID

WHERE p.Prescription\_Date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 1 MONTH) AND CURRENT\_DATE()

GROUP BY m.Medication\_ID

ORDER BY Prescription\_Count DESC

LIMIT 1;

6.

SELECT he.Department, COUNT(\*) AS Total\_Treatments

FROM Healthcare\_Employee he

JOIN Provides pr ON he.Healthcare\_Employee\_ID = pr.Healthcare\_Employee\_ID

GROUP BY he.Department;

7.

SELECT p.First, p.Last, he.Specialization

FROM Person p

JOIN Healthcare\_Employee he ON p.P\_ID = he.P\_ID

JOIN Working\_Hours wh ON he.Healthcare\_Employee\_ID = wh.Healthcare\_Employee\_ID

JOIN Schedule s ON wh.Schedule\_ID = s.Schedule\_ID

WHERE s.Day IN ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')

AND TIME\_TO\_SEC(s.Closing\_Time) - TIME\_TO\_SEC(s.Opening\_Time) >= 8 \* 3600 -- Assuming working hours per day is 8 hours

AND DATE\_SUB(CURRENT\_DATE(), INTERVAL 1 WEEK) <= CURRENT\_DATE()

GROUP BY p.First, p.Last, he.Specialization

HAVING COUNT(DISTINCT s.Day) = 7;

8.

SELECT COUNT(\*) AS Patient\_Count

FROM Appointment

WHERE Healthcare\_Employee\_ID = (

SELECT Healthcare\_Employee\_ID

FROM (

SELECT Healthcare\_Employee\_ID, COUNT(\*) AS Appointment\_Count

FROM Appointment

GROUP BY Healthcare\_Employee\_ID

ORDER BY Appointment\_Count DESC

LIMIT 1

) AS Most\_Popular\_Professional

);

9.

SELECT \*

FROM Medical\_Record

WHERE Medical\_Record\_ID IN (

SELECT Medical\_Record\_ID

FROM Appointment

WHERE Healthcare\_Employee\_ID = (

SELECT Healthcare\_Employee\_ID

FROM Healthcare\_Employee

ORDER BY Years\_of\_Experience DESC

LIMIT 1

)

);

10.

SELECT p.First, p.Last

FROM Person p

JOIN Healthcare\_Employee he ON p.P\_ID = he.P\_ID

JOIN Patient pat ON p.P\_ID = pat.P\_ID

WHERE pat.Patient\_ID IN (

SELECT Patient\_ID

FROM Appointment

WHERE Date BETWEEN he.Years\_of\_Experience AND DATE\_ADD(he.Years\_of\_Experience, INTERVAL 1 MONTH)

)

GROUP BY p.P\_ID;

11.

SELECT he.Department, COUNT(DISTINCT app.Patient\_ID) AS Unique\_Patient\_Visits

FROM Healthcare\_Employee he

JOIN Appointment app ON he.Healthcare\_Employee\_ID = app.Healthcare\_Employee\_ID

GROUP BY he.Department

ORDER BY Unique\_Patient\_Visits DESC

LIMIT 1;

12.

SELECT p.First, p.Last

FROM Person p

JOIN Patient pat ON p.P\_ID = pat.P\_ID

WHERE pat.Patient\_ID IN (

SELECT Patient\_ID

FROM Appointment

GROUP BY Patient\_ID

HAVING COUNT(\*) > 60 -- Assuming there are 12 months in a year, so 5 years = 60 months

);

13.

SELECT \*

FROM Appointment

WHERE Patient\_ID IN (

SELECT Patient\_ID

FROM Appointment

GROUP BY Patient\_ID

HAVING COUNT(\*) > 12 -- Assuming there are 12 months in a year

)

AND Date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 1 YEAR) AND CURRENT\_DATE();