**MEDICAL RECORD MAINTANANCE**

**Patient Table:**

Patient\_ID (primary key)

First\_Name

Last\_Name

Date\_of\_Birth

Age

Gender

Address

Street

City

Pincode

Phone\_Number

Email

**Caregiver Table:**

Patient\_ID(foreign key referencing from patient table)

Caregiver\_ID

First\_Name

Last\_Name

Phone\_Number

Email

**Medical History:**

Patient\_ID(foreign key referencing from patient table)

Med\_ID(Primary key)

Surgeries varchar(),

Medication,

Conditions,

Date/time,

**Visit Table:**

Visit\_ID (primary key)

Patient\_ID (foreign key referencing Patient table)

Doctor\_ID (foreign key referencing Doctor table)

Visit\_Date

Visit\_End time(primary key)

Visit\_Start time(primary key)

Treatment

Remarks

**Doctor Table:**

Doctor\_ID (primary key)

First\_Name

Last\_Name

Specialty

Address

Phone\_Number

Email\_Address

**Prescription Table:**

Prescription\_ID (primary key)

Visit\_ID (foreign key referencing Visit table)

Medication\_Name

Dosage

Frequency

Start\_Date

End\_Date

**Bill:**

Patient\_ID(foreign key referencing from patient table)

Bill\_NO(primary key)

Pay

Date

The ER diagram would consist of four entities: Patient, Visit, Doctor, and Prescription.

The Patient entity would have attributes such as Patient\_ID, First\_Name, Last\_Name, Date\_of\_Birth, Gender, Address, Phone\_Number, and Email\_Address. The Patient\_ID would serve as the primary key for the entity.

The Visit entity would have attributes such as Visit\_ID, Patient\_ID, Doctor\_ID, Visit\_Date, Diagnosis, Treatment, and Remarks. The Visit\_ID would serve as the primary key for the entity. The Patient\_ID and Doctor\_ID would be foreign keys referencing the Patient and Doctor entities, respectively.

The Doctor entity would have attributes such as Doctor\_ID, First\_Name, Last\_Name, Specialty, Address, Phone\_Number, and Email\_Address. The Doctor\_ID would serve as the primary key for the entity.

The Prescription entity would have attributes such as Prescription\_ID, Visit\_ID, Medication\_Name, Dosage, Frequency, Start\_Date, and End\_Date. The Prescription\_ID would serve as the primary key for the entity, and the Visit\_ID would be a foreign key referencing the Visit entity.

**CODE:**

CREATE TABLE PatientTable (

Patient\_ID INT PRIMARY KEY,

First\_Name VARCHAR(20),

Last\_Name VARCHAR(20),

Date\_of\_Birth DATE,

Gender VARCHAR(20),

Address VARCHAR(20),

Phone\_Number VARCHAR(20),

Email\_Address VARCHAR(20)

);

CREATE TABLE Caregiver (

Patient\_ID INT,

First\_Name VARCHAR(20),

Last\_Name VARCHAR(20),

Gender VARCHAR(20),

Address VARCHAR(20),

Phone\_Number VARCHAR(20),

Email\_Address VARCHAR(20),

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

);

CREATE TABLE MedicalHistory (

Patient\_ID INT,

Medical\_History VARCHAR(20),

Surgeries VARCHAR(20),

Medication VARCHAR(20),

Conditions VARCHAR(20),

MedicalHistory\_Date DATE,

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

);

drop table MedicalHistory;

CREATE TABLE VisitTable (

Patient\_ID INT,

Doctor\_ID INT,

Visit\_ID INT PRIMARY KEY,

Visit\_Date DATE,

Visit\_Endtime TIMESTAMP,

Visit\_Starttime TIMESTAMP,

Diagnosis VARCHAR(20),

Treatment VARCHAR(20),

Remarks VARCHAR(20),

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

FOREIGN KEY (Doctor\_ID) REFERENCES DoctorTable(Doctor\_ID)

);

CREATE TABLE DoctorTable (

Doctor\_ID INT PRIMARY KEY,

First\_Name VARCHAR(20),

Last\_Name VARCHAR(20),

Specialty VARCHAR(20),

Address VARCHAR(20),

Phone\_Number VARCHAR(20),

Email\_Address VARCHAR(20)

);

CREATE TABLE Prescription (

Prescription\_ID INT PRIMARY KEY,

Visit\_ID INT,

Medication\_Name VARCHAR(20),

Dosage VARCHAR(20),

Frequency VARCHAR(20),

Start\_Date DATE,

End\_Date DATE,

FOREIGN KEY (Visit\_ID) REFERENCES VisitTable(Visit\_ID)

);

CREATE TABLE Bill(

BILL\_NO INT PRIMARY KEY,

Pay INT,

Date date,

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

);

DESC PatientTable;

DESC Caregiver;

DESC MedicalHistory;

DESC VisitTable;

DESC DoctorTable ;

DESC Prescription;











