Table: Doctors

The Doctors table stores information about healthcare professionals. Each doctor has a unique ID, along with personal details such as first name, last name, date of birth, gender, contact number, and email address.

```
CREATE TABLE Doctors (
    doctor_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    dob DATE,
    gender VARCHAR(10),
    phone_number VARCHAR(15),
    email VARCHAR(100)
);
```

Table: Doctor_Licenses

The Doctor_Licenses table records licensing information for each doctor. It includes a unique license ID, the associated doctor ID, license number, issue and expiry dates, and the name of the issuing authority.

```
CREATE TABLE Doctor_Licenses (
    license_id INT PRIMARY KEY,
    doctor_id INT,
    license_number VARCHAR(50),
    issue_date DATE,
    expiry_date DATE,
    issuing_authority VARCHAR(100),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

Table: Specializations

The Specializations table contains a list of medical specialization areas such as Cardiology or Neurology. Each entry includes a unique specialization ID, a name, and a description.

```
CREATE TABLE Specializations (
    specialization_id INT PRIMARY KEY,
    name VARCHAR(100),
    description TEXT
);
```

Table: Doctor_Specializations

The Doctor_Specializations table defines a many-to-many relationship between doctors and their specializations using doctor IDs and specialization IDs.

```
CREATE TABLE Doctor_Specializations (
    doctor_id INT,
    specialization_id INT,
```

```
    PRIMARY KEY (doctor_id, specialization_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id),
    FOREIGN KEY (specialization_id) REFERENCES Specializations(specialization_id)
);
```

Table: Hospitals

The Hospitals table keeps track of different hospital entities. It includes a unique hospital ID, name, address, contact number, and website URL.

```
CREATE TABLE Hospitals (
    hospital_id INT PRIMARY KEY,
    name VARCHAR(100),
    address TEXT,
    phone_number VARCHAR(15),
    website_url VARCHAR(100)
);
```

Table: Departments

The Departments table represents various departments within hospitals. Each department has a unique ID and is linked to a hospital. It also includes the department name and the floor number it is located on.

```
CREATE TABLE Departments (
    department_id INT PRIMARY KEY,
    hospital_id INT,
    name VARCHAR(100),
    floor_number INT,
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id)
);
```

Table: Doctor_Hospital_Assignments

This table manages the relationship between doctors and the hospitals they work in. It stores the doctor ID, hospital ID, department ID, and the start and end dates of the assignment.

```
CREATE TABLE Doctor_Hospital_Assignments (
    doctor_id INT,
    hospital_id INT,
    department_id INT,
    start_date DATE,
    end_date DATE,
    PRIMARY KEY (doctor_id, hospital_id, department_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id),
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id),
    FOREIGN KEY (department_id) REFERENCES Departments(department_id)
);
```

Table: Patients

The Patients table stores personal details of patients including a unique patient ID, name, date of birth, gender, contact number, and email address.

```
CREATE TABLE Patients (
    patient_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    dob DATE,
    gender VARCHAR(10),
    contact_number VARCHAR(15),
    email VARCHAR(100)
);
```

Table: Appointments

The Appointments table logs each appointment scheduled between a doctor and a patient. It includes appointment ID, doctor ID, patient ID, hospital ID, date and time of the appointment, reason for visit, and appointment status.

```
CREATE TABLE Appointments (
    appointment_id INT PRIMARY KEY,
    doctor_id INT,
    patient_id INT,
    hospital_id INT,
    appointment_date DATETIME,
    reason TEXT,
    status VARCHAR(20),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (hospital_id) REFERENCES Hospitals(hospital_id)
);
```