Programming Assignment Unit 5 – Solution

The following SQL statements are used to create the **Specialty**, **Doctor**, **Patient**, **Allergy**, **PatientAllergy**, **Appointment**, **Medicine**, and **PatientMedicine** tables.

As you can see Not Null constraints are added using the "Not Null" clause. We can define constraints in two ways:

1) A constraint can directly be defined in Create Table statement.
2) A constraint can be defined by altering an existing table with a constraint as in altering Doctor Table.

Using dates and times could be confusing in SQL. IBM DB2 and OpenOffice BASE use "date" and "time" data types separately. However, date and time could be used together with Oracle and Microsoft SQL Server.

Specialty Table

```
Create Table Specialty(
    SpecialtyNumber char(10) Not Null,
    SpecialtyName char (40),
CONSTRAINT pk_sn PRIMARY KEY (SpecialtyNumber),
CONSTRAINT uq_sn Unique (SpecialtyNumber)
);
```

Doctor Table

```
Create Table Doctor(
    DoctorID char(10) Not Null,
    Name char (40) Not Null,
    Phone char (20) Not Null,
    SpecialtyNumber char(10) Not Null,
    Supervisor char(10),
CONSTRAINT pk_did PRIMARY KEY (DoctorID),
CONSTRAINT uq_did Unique (DoctorID)
);

ALTER TABLE Doctor
ADD CONSTRAINT fk_Specialty FOREIGN KEY (SpecialtyNumber)
REFERENCES Specialty(SpecialtyNumber);
```

Patient Table

```
Create Table Patient(
    PatientID char(10) Not Null,
    Name char(40) Not Null,
    Phone char(20) Not Null,
    Email char(40),
    Address char(100) Not Null,
    AddedDate date Not Null,
    DoctorID char(10) Not Null,
CONSTRAINT pk_pid PRIMARY KEY (PatientID),
CONSTRAINT uq_pid Unique (PatientID),
CONSTRAINT fk_doctor FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
);
```

Allergy Table

```
Create Table Allergy(
    AllergyID char(10) Not Null,
    AllergyName char(40) Not Null,
CONSTRAINT pk_aid PRIMARY KEY (AllergyID),
CONSTRAINT uq_aid Unique (AllergyID)
);
```

PatientAllergy Table

```
Create Table PatientAllergy(
    AllergyID char(10) Not Null,
    PatientID char(10) Not Null,
CONSTRAINT pk_apid PRIMARY KEY (AllergyID,PatientID),
CONSTRAINT uq_apid Unique (AllergyID,PatientID),
CONSTRAINT fk_allergy FOREIGN KEY (AllergyID) REFERENCES Allergy(AllergyID),
CONSTRAINT fk_patient FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)
);
```

Appointment Table

```
Create Table Appointment(
    AppointmentID char(10) Not Null,
    PatientID char(10) Not Null,
    DoctorID char(10) Not Null,
    AppointmentDate date Not Null,
    BloodPressure  smallint Not Null,
    Weight decimal(5,2) Not Null,
    TreatmentNotes char(255) Not Null,
CONSTRAINT pk_appid PRIMARY KEY (AppointmentID),
CONSTRAINT uq_appid Unique (AppointmentID),
CONSTRAINT fk_doctor_app FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID),
CONSTRAINT fk_patient_app FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)
);
```

Medicine Table

```
Create Table Medicine(
    MedicineID char(10) Not Null,
    MedicineName char (40) Not Null,
CONSTRAINT pk_mid_pk PRIMARY KEY (MedicineID),
CONSTRAINT uq_mid Unique (MedicineID)
);
```

PatientMedicine Table

```
Create Table PatientMedicine(
    AppointmentID char(10) Not Null,
    MedicineID char(10) Not Null,
CONSTRAINT pk_pmid PRIMARY KEY (AppointmentID,MedicineID),
CONSTRAINT uq_pmid Unique (AppointmentID,MedicineID),
CONSTRAINT fk_appointment FOREIGN KEY (AppointmentID) REFERENCES
Appointment(AppointmentID),
CONSTRAINT fk_medicine FOREIGN KEY (MedicineID) REFERENCES Medicine(MedicineID)
);
```