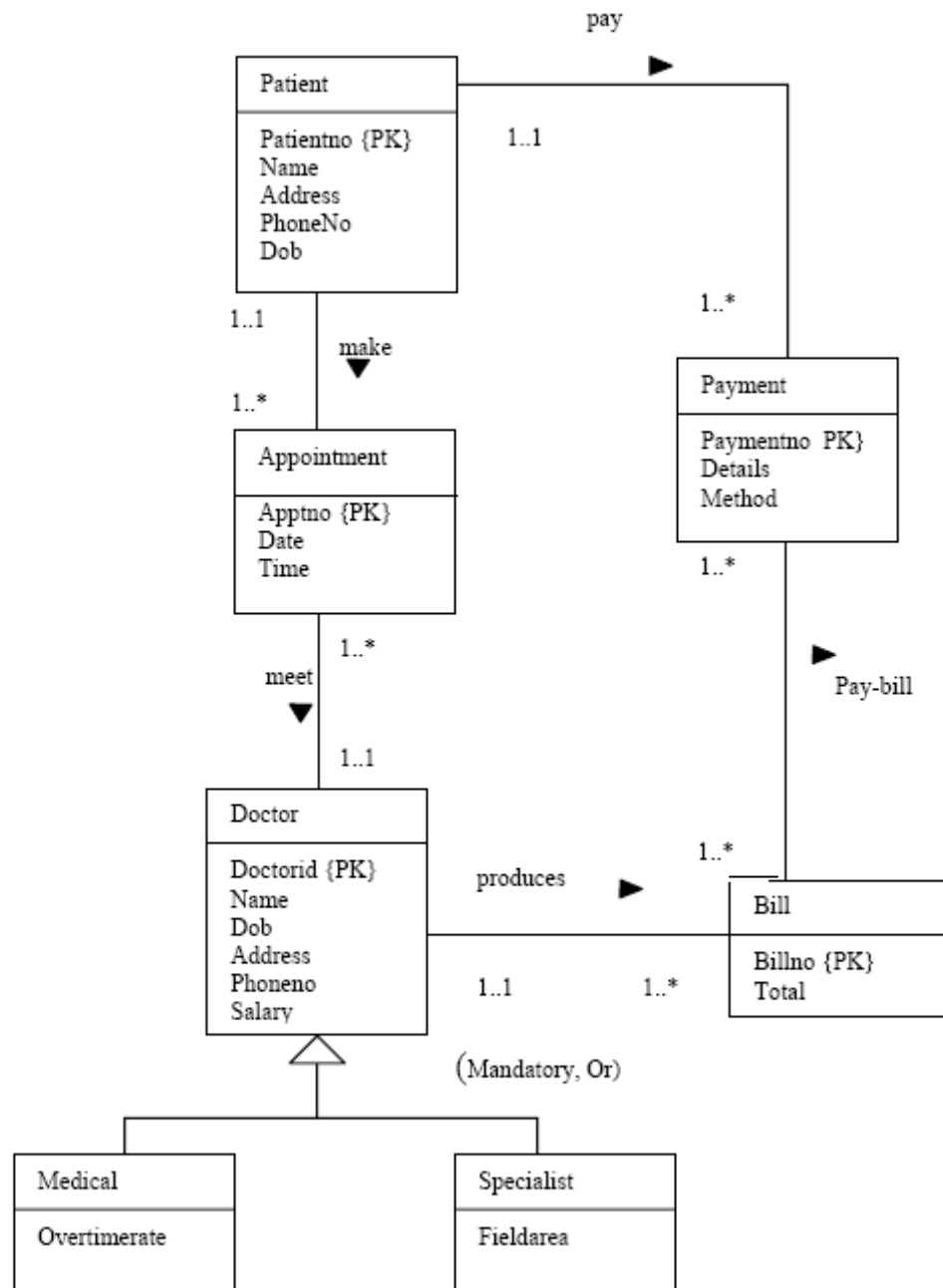


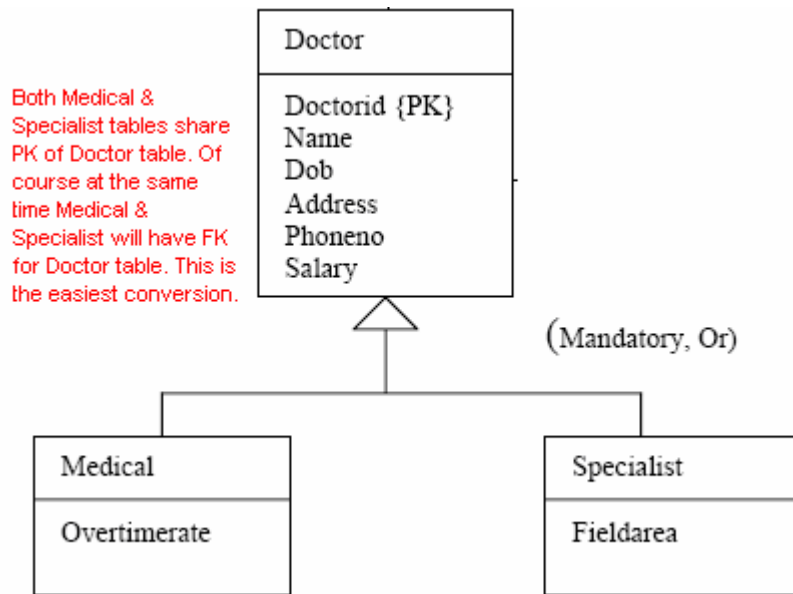
Enhanced Entity Relationship: From ERD to Relational Model – Exercise & Answer

The following is Entity Relationship Model of ABC Medical Centre. Map the Entity Relationship Model into Relational Model.



Need to study and properly understand the 7 rules (from mannino book) in converting the ERD to Table schemas (relational model). The following SQL script try to cover the whole ERD diagram.

Firstly we analyze the generalization part. It is a straight forward and the easiest conversion.



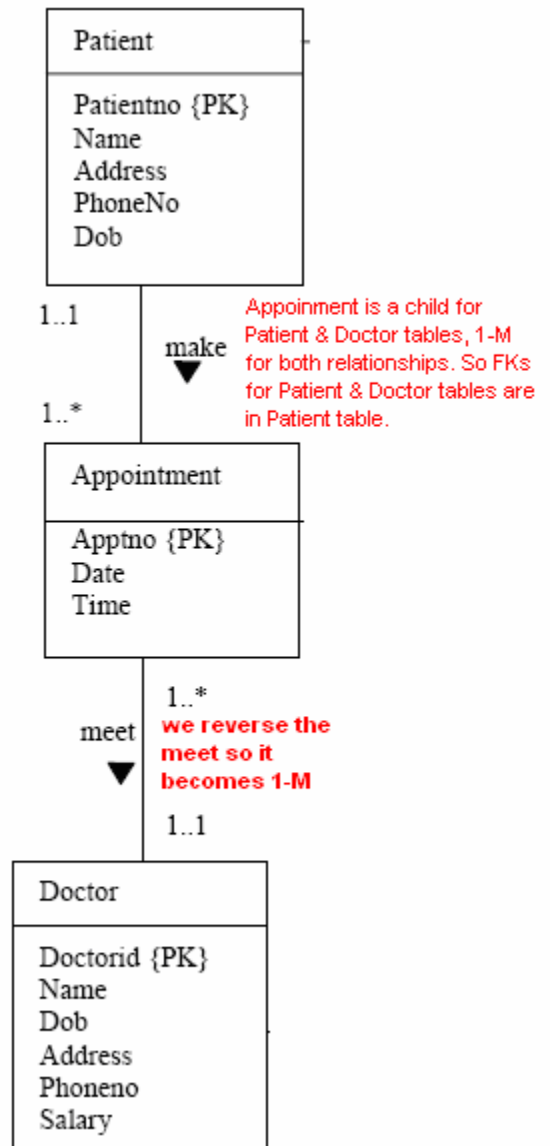
```

CREATE TABLE Doctor(
    Doctorid      INTEGER,
    Name          VARCHAR(30),
    Dob           DATE,
    Address       VARCHAR(50),
    Phoneno       VARCHAR(20),
    Salary        NUMERIC(20,2),
    CONSTRAINT PKDoctor PRIMARY KEY (Doctorid)
)

CREATE TABLE Medical(
    Doctorid      INTEGER,
    Overtime      NUMERIC(4,2),
    CONSTRAINT PKMedical PRIMARY KEY (Doctorid),
    CONSTRAINT FKMedical FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid) ON DELETE
    CASCADE
)

CREATE TABLE Specialist(
    Doctorid      INTEGER,
    Fieldarea     VARCHAR(30),
    CONSTRAINT PKSpecialist PRIMARY KEY (Doctorid),
    CONSTRAINT FKSpecialist FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid) ON
    DELETE CASCADE
)

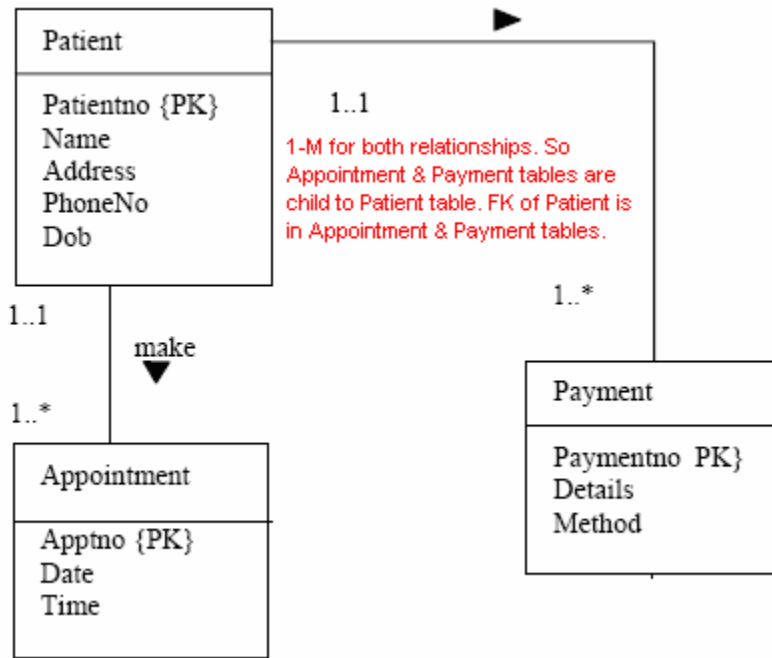
```



```

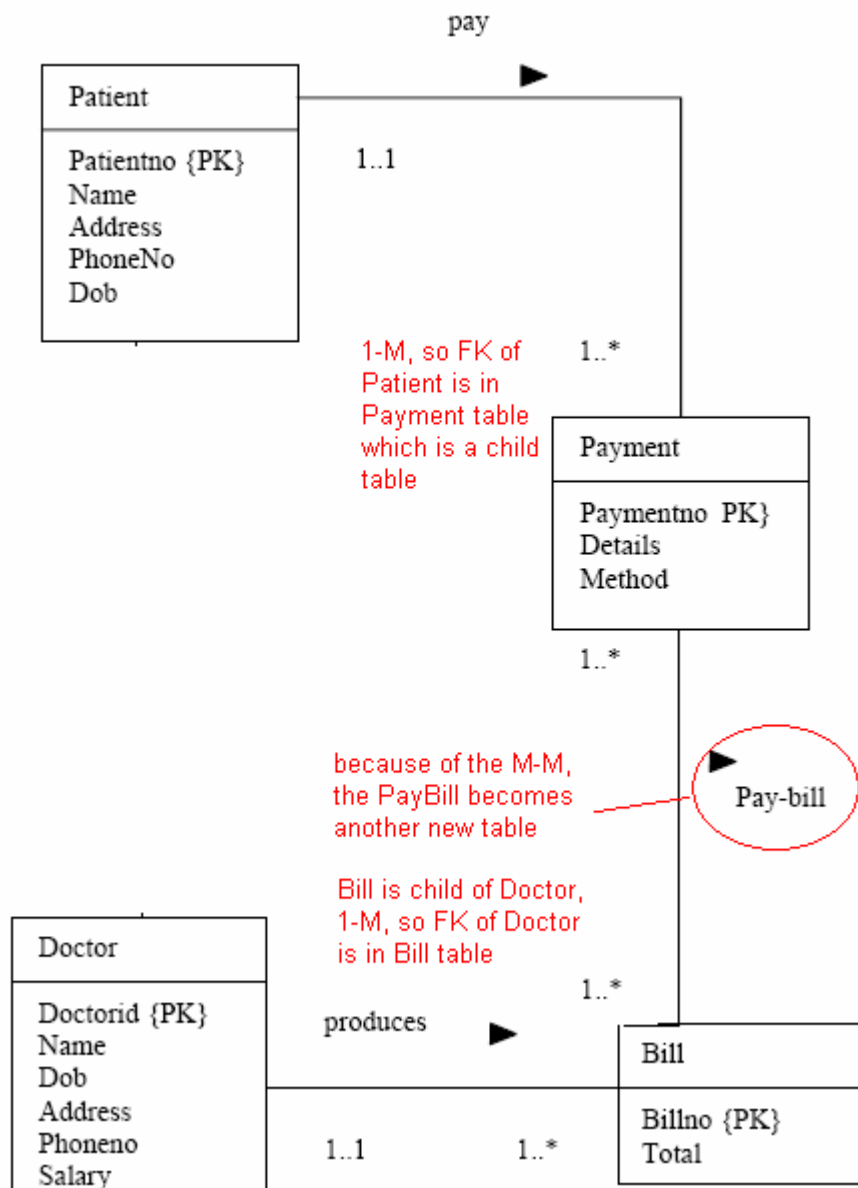
CREATE TABLE Appointment(
    Apptno          INTEGER,
    Date            DATE,
    Time            DATETIME,
    Doctorid        INTEGER,
    Patientno       INTEGER,
    CONSTRAINT PKAppointment PRIMARY KEY (Apptno),
    CONSTRAINT FKAppointment FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid),
    CONSTRAINT FKAppointment1 FOREIGN KEY (Patientno) REFERENCES Patient(Patientno)
)

```



```

CREATE TABLE Patient(
    Patientno    INTEGER,
    Name         VARCHAR(40),
    Address      VARCHAR(50),
    PhoneNo     VARCHAR(20),
    Dob          DATE,
    CONSTRAINT PKPatient PRIMARY KEY (Patientno)
)
  
```



```
CREATE TABLE Payment(
    Paymentno    INTEGER,
    Details      VARCHAR(60),
    Method       VARCHAR(20),
    Patientno    INTEGER,
    CONSTRAINT PKPayment PRIMARY KEY (Paymentno),
    CONSTRAINT FKPayment FOREIGN KEY (Patientno) REFERENCES Patient(Patientno)
)
```

```
CREATE TABLE Bill(
    Billno       INTEGER,
    Total        NUMERIC(10,2),
    Doctorid     INTEGER,
    CONSTRAINT PKBill PRIMARY KEY (Billno),
    CONSTRAINT FKBill FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid)
)
```

```
CREATE TABLE Pay_Bill(
    Paymentno    INTEGER,
```

```

Billno          INTEGER,
CONSTRAINT PKPay_Bill PRIMARY KEY (Paymentno, Billno),
CONSTRAINT FKPaymentno FOREIGN KEY (Paymentno) REFERENCES Payment(Paymentno),
CONSTRAINT FKBillno FOREIGN KEY (Billno) REFERENCES Bill(Billno)
)

```

The cumulative tables are shown below. These are generic SQL scripts. The bold notations are the answer that should be used in exam as relational model. It is just a Functional Dependency (FD) notation.

```

CREATE TABLE Doctor(
  Doctorid      INTEGER,
  Name          VARCHAR(30),
  Dob           DATE,
  Address       VARCHAR(50),
  Phoneno       VARCHAR(20),
  Salary        NUMERIC(20,2),
  CONSTRAINT PKDoctor PRIMARY KEY (Doctorid)
)

```

Doctor(Doctorid, Name, Dob, Address, Phoneno, Salary)
Primary Key Doctorid

```

CREATE TABLE Medical(
  Doctorid      INTEGER,
  Overtimerate  NUMERIC(4,2),
  CONSTRAINT PKMedical PRIMARY KEY (Doctorid),
  CONSTRAINT FKMedical FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid) ON DELETE
  CASCADE
)

```

Medical(Doctorid, Overtimerate)
Primary Key Doctorid
Foreign Key Doctorid references Doctor(Doctorid)

```

CREATE TABLE Specialist(
  Doctorid      INTEGER,
  Fieldarea     VARCHAR(30),
  CONSTRAINT PKSpecialist PRIMARY KEY (Doctorid),
  CONSTRAINT FKSpecialist FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid) ON
  DELETE CASCADE
)

```

Specialist(Doctorid, Fieldarea)
Primary Key Doctorid
Foreign Key Doctorid references Doctor(Doctorid)

```

CREATE TABLE Appointment(
  Apptno        INTEGER,
  Date          DATE,
  Time          DATETIME,
  Doctorid      INTEGER,
  Patientno     INTEGER,
  CONSTRAINT PKAppointment PRIMARY KEY (Apptno),
  CONSTRAINT FKAppointment FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid),
  CONSTRAINT FKAppointment1 FOREIGN KEY (Patientno) REFERENCES Patient(Patientno)
)

```

Appointment(Appno, Date, Time, Doctorid, Patientno)

Primary Key Appno

Foreign Key Doctorid references Doctor(Doctorid)

Foreign Key Patientno references Patient(Patientno)

```
CREATE TABLE Patient(  
    Patientno    INTEGER,  
    Name         VARCHAR(40),  
    Address      VARCHAR(50),  
    PhoneNo     VARCHAR(20),  
    Dob          DATE,  
    CONSTRAINT PKPatient PRIMARY KEY (Patientno)  
)
```

Patient(Patientno, Name, Address, Phoneno, Dob)

Primary Key Patientno

```
CREATE TABLE Payment(  
    Paymentno    INTEGER,  
    Details      VARCHAR(60),  
    Method       VARCHAR(20),  
    Patientno    INTEGER,  
    CONSTRAINT PKPayment PRIMARY KEY (Paymentno),  
    CONSTRAINT FKPayment FOREIGN KEY (Patientno) REFERENCES Patient(Patientno)  
)
```

Payment(Paymentno, Details, Method, Patientno)

Primary Key Paymentno

Foreign Key Patientno references Patient(Patientno)

```
CREATE TABLE Bill(  
    Billno       INTEGER,  
    Total        NUMERIC(10,2),  
    Doctorid     INTEGER,  
    CONSTRAINT PKBill PRIMARY KEY (Billno),  
    CONSTRAINT FKBill FOREIGN KEY (Doctorid) REFERENCES Doctor(Doctorid)  
)
```

Bill(Billno, Total, Doctorid)

Primary Key Billno

Foreign Key Doctorid references Doctor(Doctorid)

```
CREATE TABLE Pay_Bill(  
    Paymentno    INTEGER,  
    Billno       INTEGER,  
    CONSTRAINT PKPay_Bill PRIMARY KEY (Paymentno, Billno),  
    CONSTRAINT FKPaymentno FOREIGN KEY (Paymentno) REFERENCES Payment(Paymentno),  
    CONSTRAINT FKBillno FOREIGN KEY (Billno) REFERENCES Bill(Billno)  
)
```

Pay_Bill(Paymentno, Billno)

Primary Key Paymentno, Billno

Foreign Key Paymentno references Payment(Paymentno)

Foreign Key Billno references Bill(Billno)