

**Washington State University**  
**School of Electrical Engineering and Computer Science**  
**CptS 451 – Introduction to Database Systems**  
**Spring 2018**

Dr. Sakire Arslan Ay

**Homework-2**  
**ER to Relational Model**

Due Date: Tuesday, February 13 <sup>st</sup> , 05:00pm
--

Name: \_\_\_\_\_ Solution Key \_\_\_\_\_

Student Number: \_\_\_\_\_

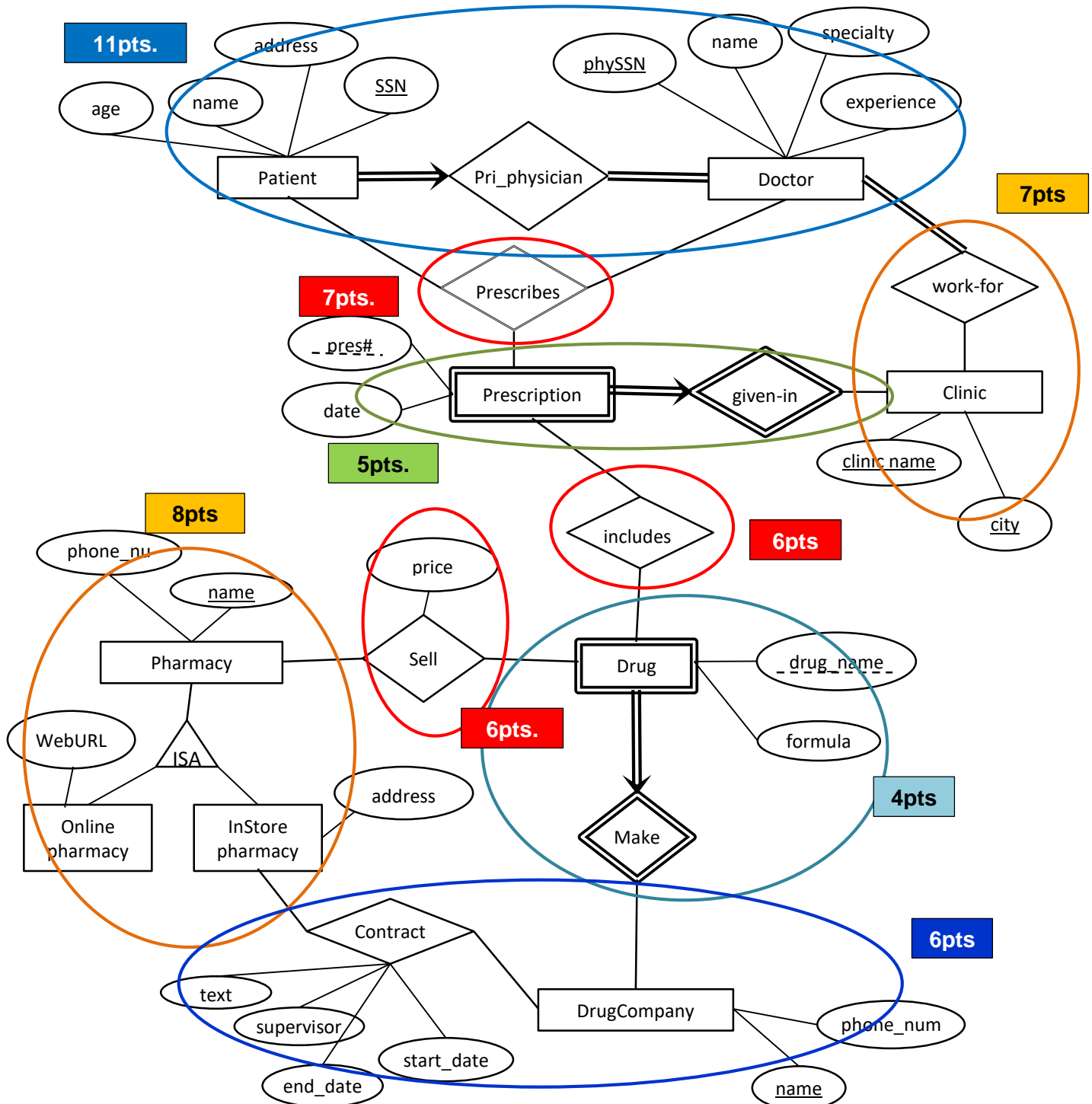
Question:	Max points:	Score:
1	60	
2	40	
Total	100	

### Question 1.

Consider the ER diagram illustrated in the following figure (this is similar to the ER diagram from HW1 - couple modifications have been made to make the diagram simpler.)

Convert the ER diagram to relations and write SQL DDL statements for creating the tables for those relations. Pick suitable data types for each attribute. For string attributes pick reasonable lengths. Include the appropriate constraints (domain, primary key, foreign key, UNIQUE, and NULL constraints) in your SQL DDL statements.

**Note :** For the one-to-many binary relations, combine the relation with the many side. For the superclass-subclass entity sets, translate them using the ER-approach.



```

CREATE TABLE Doctor (
    phySSN      CHAR(9) ,
    name        VARCHAR(30) ,
    speciality   VARCHAR (30) ,
    experience   integer
    PRIMARY KEY (phySSN)
);

CREATE TABLE Patient (
    patientSSN  CHAR(9) ,
    phySSN      CHAR(9) NOT NULL,  --enforces total participation
    name        VARCHAR(30) ,
    age         INTEGER,
    address     VARCHAR(100) ,
    PRIMARY KEY (patientSSN)
    FOREIGN KEY (phySSN) REFERENCES Doctor (phySSN)
);

-- Note: Can't enforce total participation of Doctor in "Pri_physician" relationship.

CREATE TABLE Clinic (
    clinic_name VARCHAR(30) ,
    city         VARCHAR(30) ,
    PRIMARY KEY(clinic_name,city)
);

CREATE TABLE Work-for (
    clinic_name VARCHAR(30) ,
    city         VARCHAR(30) ,
    phySSN      CHAR (9) ,
    PRIMARY KEY(phySSN,clinic_name,city) ,
    FOREIGN KEY(phySSN) REFERENCES Doctor (phySSN) ,
    FOREIGN KEY(clinic_name,city) REFERENCES Clinic(clinic_name,city)
);

-- Note: Can't enforce total participation of Work-for in "Clinic" relationship.

CREATE TABLE Prescription (
    pres_num     INTEGER,
    clinic_name   VARCHAR(30) ,
    city          VARCHAR(30) ,
    date         DATE,
    PRIMARY KEY(pres_num,clinic_name,city) ,
    FOREIGN KEY(clinic_name,city) REFERENCES Clinic(clinic_name,city)
);

CREATE TABLE Prescribes (
    pres_num     INTEGER,
    clinic_name   VARCHAR(30) ,
    city          VARCHAR(30) ,
    patientSSN    CHAR(9) ,
    phySSN        CHAR(9) ,
    PRIMARY KEY (pres_num,clinic_name,city,patientSSN,phySSN) ,
    FOREIGN KEY (pres_num,clinic_name,city) REFERENCES

```

```

        Prescription (pres_num,clinic_name,city),
FOREIGN KEY (patientSSN) REFERENCES Patient(patientSSN),
FOREIGN KEY (phySSN) REFERENCES Doctor(phySSN)
);

CREATE TABLE DrugCompany (
    company_name    VARCHAR(20),
    phone_num       CHAR(10),
    PRIMARY KEY (company_name)
);

CREATE TABLE Drug (
    drug_name       VARCHAR(20),
    company_name    VARCHAR(20),
    formula         VARCHAR(255),
    PRIMARY KEY (drug_name,company_name),
    FOREIGN KEY (company_name) REFERENCES DrugCompany(company_name)
);

CREATE TABLE Includes (
    pres_num        INTEGER,
    clinic_name     VARCHAR(30),
    city            VARCHAR(30),
    drug_name       VARCHAR (20),
    company_name    VARCHAR (20),
    PRIMARY KEY (pres_num,clinic_name,city,drug_name,company_name),
    FOREIGN KEY (pres_num,clinic_name,city)
        REFERENCES Prescription(pres_num,clinic_name,city),
    FOREIGN KEY (drug_name,company_name)
        REFERENCES Drug (drug_name,company_name)
);

CREATE TABLE Pharmacy (
    phar_name       VARCHAR(20),
    phone_num       CHAR(10),
    PRIMARY KEY (phar_name)
);

CREATE TABLE OnlinePharmacy (
    phar_name       VARCHAR(20),
    WebURL          VARCHAR(255),
    PRIMARY KEY (phar_name)
    FOREIGN KEY (phar_name) REFERENCES Pharmacy(phar_name)
);

CREATE TABLE InStorePharmacy (
    phar_name       VARCHAR(20),
    address         VARCHAR(100),
    PRIMARY KEY (phar_name)
    FOREIGN KEY (phar_name) REFERENCES Pharmacy(phar_name)
);

```

```

CREATE TABLE Sell (
    phar_name      VARCHAR (20),
    drug_name      VARCHAR (20),
    company_name   VARCHAR (20),
    price          FLOAT,
    PRIMARY KEY (phar_name, drug_name, company_name),
    FOREIGN KEY (phar_name) REFERENCES Pharmacy (phar_name),
    FOREIGN KEY (drug_name, company_name) REFERENCES
        Drug (drug_name, company_name)
);

CREATE TABLE Contract (
    phar_name      VARCHAR (20),
    company_name   VARCHAR (20),
    text           VARCHAR (255),
    supervisor     VARCHAR (30),
    start_date     DATE,
    end_date       DATE,
    PRIMARY KEY (phar_name, company_name),
    FOREIGN KEY (phar_name) REFERENCES InStorePharmacy (phar_name),
    FOREIGN KEY (company_name) REFERENCES DrugCompany (company_name)
);

```

---

## Question 2.

Consider the following relations:

Relation R1

	<u>A</u>	<u>B</u>	C
1	a1	b10	s100
2	a2	b10	s300
3	a2	b20	s200
4	a3	b10	s500
5	a4	b20	s100

Relation R2

	<u>D</u>	E	F
1	d10	50	100
2	d20	125	200
3	d30	150	300
4	d40	75	400
5	d50	100	200

Relation R3

	<u>M</u>	<u>N</u>	<u>O</u>	P
1	a1	b10	d10	25
2	a1	b10	d20	5
3	a2	b10	d20	20
4	a2	b20	d20	15
5	a3	b10	d40	15
6	a4	b20	d40	5
7	a4	b20	d50	10

Relation R4

	<u>J</u>	<u>K</u>	L
1	s200	d20	22
2	s500	d50	55

Relation R5

	<u>S</u>	T	U
1	s100	20	555
2	s200	20	333
3	s300	30	111
4	s400	30	555
5	s500	40	444

Primary Keys (refer to these as PK1 through PK5):

1. Relation **R1**: A,B
2. Relation **R2**: D
3. Relation **R3**: M,N,O
4. Relation **R4**: J,K
5. Relation **R5**: S

The following foreign key constraints are given for relations R1, R2, R3, R4 and R5:  
(refer to these as FK1 through FK2)

1. **R3(MN)** references **R1(AB)**
2. **R3(O)** references **R2(D)**
3. **R1(C)** references **R5(S)**
4. **R4(J)** references **R5(S)**
5. **R4(K)** references **R2(D)**

- a) (10pts) Write the CREATE TABLE (SQL DDL statements) to define relations R1,R2,R3,R4,R5. Include the appropriate constraints (primary key and foreign key) in your statements. For all foreign keys:
- define "CASCADE" policy for delete operations, and
  - define "SET NULL" policy for update operations.

```
CREATE TABLE R5 (  
    S      CHAR(4) ,  
    T      INTEGER ,  
    U      INTEGER ,  
    PRIMARY KEY (S)  
);
```

```
CREATE TABLE R1 (  
    A      CHAR(2) ,  
    B      CHAR(3) ,  
    C      CHAR(4) ,  
    PRIMARY KEY (A,B) ,  
    FOREIGN KEY (C) REFERENCES R5 (S)  
        ON DELETE CASCADE ON UPDATE SET NULL  
);
```

```
CREATE TABLE R2 (  
    D      CHAR(3) ,  
    E      INTEGER ,  
    F      INTEGER ,  
    PRIMARY KEY (D)  
);
```

```
CREATE TABLE R3 (  
    M      CHAR(2) ,  
    N      CHAR(3) ,  
    O      CHAR(3) ,  
    P      CHAR(3) ,  
    PRIMARY KEY (M,N,O) ,  
    FOREIGN KEY (M,N) REFERENCES R1 (A,B)  
        ON DELETE CASCADE ON UPDATE SET NULL ,  
    FOREIGN KEY (O) REFERENCES R2 (D)  
        ON DELETE CASCADE ON UPDATE SET NULL  
);
```

```

CREATE TABLE R4 (
    J      CHAR(4) ,
    K      CHAR(3) ,
    L      INTEGER,
    PRIMARY KEY (J, K) ,
    FOREIGN KEY (J) REFERENCES R5 (S)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (K) REFERENCES R2 (D)
        ON DELETE CASCADE ON UPDATE SET NULL
);

```

**b) (10pts)** For the operations given below, indicate whether execution of the operation would violate some “primary key” or “foreign key constraints”. If your answer is yes, specify the constraints (from the above list) that would be violated (e.g. violates FK1) Make the changes on the original tables for each operation below.

i) Insert tuple ('a4', 'b10', 's100') into R1.

No violation.

ii) Insert tuple ('s200', 'd30', 11) into R4.

No violation.

iii) Insert tuple ('a4', 'b10', 'd50', 5) into R3.

Violates FK-1 ( i.e., R3(M,N) references R1(AB))

iv) Insert tuple ('s200', 10, 111) into R5.

Violates PK-5

(Can't insert a duplicate value for the primary key attribute S ('s200' already exists))

**c) (15pts)** For the operations given below, indicate whether execution of the operation would violate any “foreign key constraints”. If your answer is yes, specify the constraints (from the above list) that would be violated. Apply “CASCADE” policy for delete operations, and apply “SET NULL” policy for update operations. Update the tables after applying those policies. (For deletions, give the relation name and row-number(s) of the tuple(s) that will be deleted. For updates, if the update is possible rewrite the changed tables. Otherwise explain why update can't be performed.) Make the changes on the original tables for each operation below.

i) Delete tuple ('s400', 30, 555) from R5.

No FK violation, therefore no additional tuples are deleted.

ii) Delete tuple ('a3', 'b10', 's500') from R1.

Will violate FK-1.

In R3 line#5 will be deleted.

iii) Delete tuple ('d20', 125, 200) from R2.

Will violate FK-2 and FK-5.

In R3 lines #2, #3 and #4 will be deleted.

In R4 line #1 will be deleted.

**iv)** Update tuple ('a1', 'b10', 's100') in R1 with values ('a1', 'b20', 's100')

Will violate FK-1.

Since (M,N) is the primary key for R3, M and N can't be set to NULL. Therefore the update will be rejected. No tuples in R3 will be updated.

**v)** Update tuple ('s100', 20, 555) in R5 with values ('s600', 60, 666)

Will violate FK-3.

In R1 lines #1, #5 will be updated.

Line#1: ('a1', 'b10', 's100') will be updated as ('a1', 'b10', NULL)

Line#5: ('a4', 'b20', 's100') will be updated as ('a4', 'b20', NULL)

No updates in R4.

**d) (5pts)** If all tuples in R5 are deleted, what tuples will R2 and R3 contain?

If all tuples in R5 are deleted, all tuples in R1 and R4 will be deleted. Since all tuples in R1 are deleted, all tuples in R3 will be deleted as well.

R2 doesn't have any foreign keys. So no tuples will be deleted from R2.

So R3 will be empty, R2 will stay as it is.