Washington State University

School of Electrical Engineering and Computer Science

CptS 451 – Introduction to Database Systems

Online

Dr. Sakire Arslan Ay

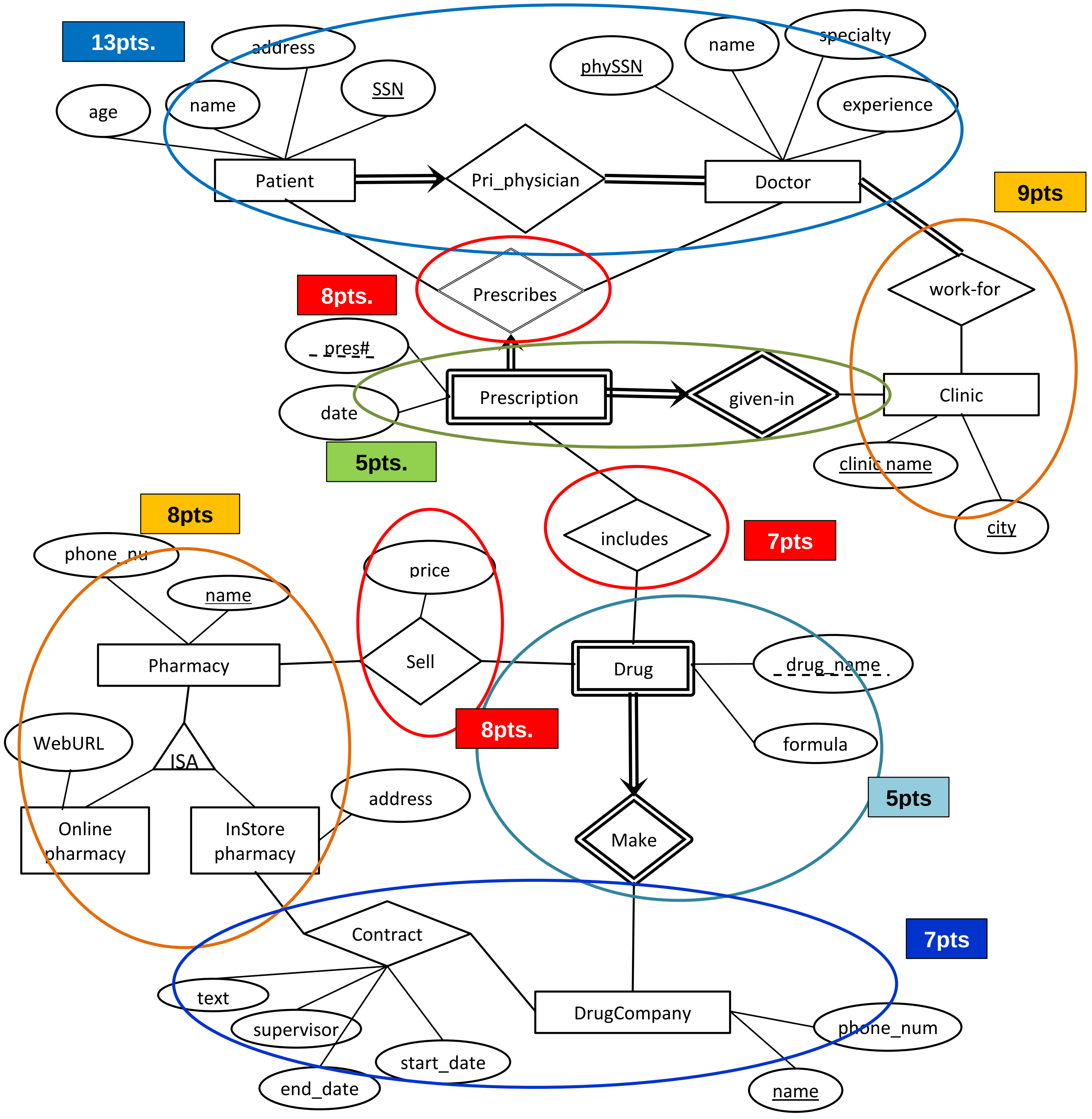
Homework-2
ER to Relational Translation

Name: __________Solution Key___________

Student Number:_____________________

Question:	Max points:	Score:
1	70	
2	30	
Total	100	

Question 1 (70 pts)
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.*

```
-- In the below solution combine Prescribes relation with Prescription
(many-one-one relationship.
-- I will also accept solutions which include separate relations for
Prescribes and Prescription.
   CREATE TABLE Prescription (
      pres_num      INTEGER,
      clinic_name   VARCHAR(30),
      city          VARCHAR(30),
      patientSSN    CHAR(9),
      phySSN        CHAR(9),
      date          DATE,
      PRIMARY KEY (pres_num,clinic_name,city),
      FOREIGN KEY (clinic_name,city) REFERENCES Clinic(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 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

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

Relation R2

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

Relation R3

	M	N	O	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

	J	K	L
1	s200	d20	22
2	s500	d50	55

Relation R5

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

Primary Keys :

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:

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)**

Assume

- "CASCADE" policy for delete operations, and
- "SET NULL" policy for update operations.

a) **(10pts)** For the operations given below, indicate whether execution of the operation would violate some "primary key" or "integrity constraints". <u>If your answer is yes, specify the constraints (from the above list) that would be violated.</u>

i) Insert tuple (a1,b10, d20, 35) into R3.
Violates the PK constraint for R3. (M,N,O) is the primary key for R3 and (M,N,O) value ('a1','b10','d20') already exists in R3. Since duplicate primary key values are not allowed, the insertion will be rejected.

ii) Insert tuple (s500, d20, 75) into R4.
No violation.
R4 has 2 foreign keys (FKs): R4(J) references R5(S) ; R4(K) references R2(D)
The first FK is not violated, since 's500' already appears in R5[S].
The second FK is also not violated since 'd20' already appears in R2[D]

b) **(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. (You may either redraw the tables or mention which tuples/attributes are deleted/updated at each table. For updates rewrite the updated tuples.) <u>Make the changes on the original tables for each operation below.</u>

i) Delete tuple (d30, 150, 300) from R2.
R2 has 2 references: R3(O) references R2(D) ; and R4(K) references R2(D)
Since 'd30' doesn't appear in R3[O] or R4[K], there won't be any violation in FKs and no tuples will be deleted in R3 and R4.

ii) Update tuple (s400,30,555) in R5 with values (6000,60,66)
R5 has 2 references: R1(C) references R5(S); and R4(J) references R5(S)
Since 's400' doesn't appear R1[C] and R4[J], there won't be any FK violation. No tuples in R1 and R4 will be updated.

iii) Update tuple (s100, 20, 555) in R5 with values (6000,60,666)
R5 has 2 references: R1(C) references R5(S); and R4(J) references R5(S).
Since 's100' doesn't appear R4[J], no tuples in R4 will be updated.
However, 's100' appears in R1[C] on lines #1 and #5. R1 will be updated as follows:

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

c) **(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 R1and 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.