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: CHARLES NGUYEN

Student Number: 011606177

Question	Max Points	Score
1	70	
2	30	
Total	100	

Question 1 (70 pts)

Entity Sets

1. Doctor

```
CREATE TABLE Doctor(
        phySSN      CHAR(11) PRIMARY KEY,
        doctor_name VARCHAR(50),
        specialty   VARCHAR(256),
        experience  INTEGER,
        clinic_name VARCHAR(50) NOT NULL,
        clinic_city VARCHAR(50) NOT NULL,
        FOREIGN KEY (clinic_name, clinic_city)
                 REFERENCES Clinic(clinic_name, clinic_city)
)
```

2. Patient

```
CREATE TABLE Patient(
        SSN          CHAR(11) PRIMARY KEY,
        patient_name VARCHAR(50),
        doctor_name  VARCHAR(50) NOT NULL,
        age          INTEGER,
        address      VARCHAR(100),
        FOREIGN KEY (doctor_name) REFERENCES Doctor(doctor_name)
)
```

3. Clinic

```
CREATE TABLE Clinic(
        clinic_name  VARCHAR(50),
        clinic_city  VARCHAR(50),
        PRIMARY KEY  (clinicName, clinicCity)
)
```

4. Pharmacy

```
CREATE TABLE Pharmacy(
        pharm_name   VARCHAR(50) PRIMARY KEY,
        phone_num    CHAR(11),
        online       VARCHAR(100),
        instore      VARCHAR(100),
)
```

5. DrugCompany

```
CREATE TABLE DrugCompany(
        comp_name    VARCHAR(50) PRIMARY KEY,
        phone_num    CHAR(11)
)
```

6. Drug

```
CREATE TABLE Drug(
        drug_name    VARCHAR(50),
        formula      VARCHAR(50),
        comp_name    VARCHAR(50),
        PRIMARY KEY  (drug_name, comp_name)
        FOREIGN KEY (comp_name) REFERENCES DrugCompany(comp_name)
)
```

7. Prescription

```
CREATE TABLE Prescription(
        pres_num     BIGINT,
        patient_ssn  CHAR(11)    NOT NULL,
        doctor_name  CHAR(50)    NOT NULL,
        clinic_name  VARCHAR(50),
        clinic_city  VARCHAR(50),
        date         DATE,
        PRIMARY KEY (pres_num, clinic_name, clinic_city),
        FOREIGN KEY (clinic_name, clinic_city)
                 REFERENCES Clinic(clinic_name, clinic_city)
)
```

Relationship Sets

DrugMarket

```
CREATE TABLE DrugMarket(
        pharm_name   VARCHAR(50),
        drug_name    VARCHAR(50),
        comp_name    VARCHAR(50),
        price        FLOAT,
        PRIMARY KEY (pharm_name, drug_name, comp_name),
        FOREIGN KEY (pharm_name) REFERENCES Pharmacy(pharm_name),
        FOREIGN KEY (drug_name, company)
                 REFERENCES Drug(pharm_name, comp_name)
)
```

Contract

```
CREATE TABLE Contract()
        pharm_name       VARCHAR(50),
        comp_name        VARCHAR(50),
        text_link        VARCHAR(100),
        supervisor       VARCHAR(50),
        start_date       DATE,
        end_date         DATE,
        PRIMARY KEY      (pharm_name, comp_name),
        FOREIGN KEY (pharm_name) REFERENCES Pharmacy(pharm_name),
        FOREIGN KEY (comp_name) REFERENCES DrugCompany(comp_name)
)
```

Question 2 (30 pts)

a. (10 pts)

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

i) Insert tuple (a1, b10, d20, 35) into R3

- violates key constraint in R3
  - an instance of PRIMARY KEY (M=a1, N=b10, O=d20) already exists

ii) Insert tuple (s500, d20, 75) into R4

- legal

b. (15 pts)

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.)

Make the changes on the original tables for each operation below.

i) Delete tuple (d30, 150, 300) from R2

- Premises
  - R2: D
  - R2(D): referenced by R3(O)
- Result
  - Only R2 is altered, R3 does not contain any key instance referenced from R2

D	E	F
d10	50	100
d20	125	200
d40	75	400
d50	100	200

ii) Update tuple (s400, 30, 555) in R5 with values (6000, 60, 66)

- Premises
  - R5: S
  - R5(S): referenced by
    - R1(C)
    - R4(J)
- Evaluation
  - R1(C) references R5, but does not contain the instance C→S(s400)
  - R4(J) references R5, but does not contains the instance J→S(s400)

    - the value (6000) seems to violate the domain constraint on attribute S
        - does not start with the character 's'
- Result
    - Assuming that S=6000 is legal
    - only R5 is changed

S	T	U
s100	20	555
s200	20	333
s300	30	111
6000	**60**	**66**
s500	40	444

iii) Update tuple (s100, 20, 555) in R5 with values (6000, 60, 666)

- Premises
    - R5: S
    - R5(S): referenced by
        - R1(C)
        - R4(J)
- Evaluation
    - R1(C) references R5, and contain the instance C→S(s100)
    - R4(J) references R5, but does not contains the instance J→S(s100)
    - the value (6000) seems to violate the domain constraint on attribute S
        - does not start with the character 's'
- Result
    - Assuming that S=6000 is legal
        - R5 is changed
        - R1 is changed

relation R5	S	T	U
	s100	20	555
	s200	20	333
	s300	30	111
	6000	**60**	**66**

	s500	40	444

relation R1	A	B	C
	a1	b10	**6000**
	a2	b10	s300
	a2	b20	s200
	a3	b10	s500
	a4	b20	**6000**

c.(5 pts)

If all tuples in R5 are deleted, what tuples will R2 and R3 contain?

R2 and R3 remain unchanged because they do not reference R5.