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: Nam Jun Lee

Student Number: 11606459

Question 1:

Relation: Doctor (phySSN, doc_name, specialty, experience)

Primary Key: phySSN

```
CREATE TABLE Doctor (
    phySSN       CHAR (11) PRIMARY KEY,
    doc_name    VARCHAR (25),
    specialty     VARCHAR (20),
    experience   INTEGER
);
```

Relation: Patient (SSN, age, pat_name, address, phySSN)

Primary Key: SSN

Foreign Key:

Patient (phySSN) REFERENCES Doctor (phySSN) – Total participation

```
CREATE TABLE Patient (
    SSN           CHAR (11) PRIMARY KEY,
    age            INTEGER,
    pat_name    VARCHAR (25),
    address       VARCHAR (35),
    phySSN       CHAR (11) NOT NULL,
    FOREIGN KEY (phySSN) REFERENCES
    Doctor (phySSN)
);
```

Relation: Clinic (clinic_name, city)

Primary Key: city

Unique Key: clinic_name

```
CREATE TABLE Clinic (
    city            VARCHAR (35) PRIMARY KEY,
    clinic_name VARCHAR (35) UNIQUE,
);
```

Relation: Prescription (presN, date, clinic_name, city, SSN, phySSN)

Primary Key: presN, clinic_name, city)

Foreign Key:

Prescription (SSN) REFERENCES Patient (SSN)

Prescription (phySSN) REFERENCES Doctor (phySSN)

Prescription (clinic_name, city) REFERENCES Clinic (clinic_name, city)

```
CREATE TABLE Prescription (
    presN          BIGINT,
    date            DATE,
    clinic_name VARCHAR (35),
    city            VARCHAR (35),
    SSN             CHAR (11),
    phySSN        CHAR (11),
    PRIMARY KEY (presN, clinic_name, city),
    FOREIGN KEY (SSN) REFERENCES Patient (SSN),
    FOREIGN KEY (phySSN) REFERENCES Doctor (phySSN),
    FOREIGN KEY (clinic_name, city) REFERENCES Clinic (clinic_name, city)
);
```

Relation: work_for (phySSN, clinic_name, city)

Primary Key: phySSN, clinic_name, city

Foreign Key:

work_for (phySSN) REFERENCES Doctor (phySSN)

work_for (clinic_name, city) REFERENCES Clinic (clinic_name, city)

```
CREATE TABLE work_for (
    phySSN         CHAR (11),
    clinic_name    VARCHAR (35),
    city           VARCHAR (35),
    PRIMARY KEY (phySSN, clinic_name, city),
    FOREIGEN KEY (phySSN) REFERNECES Doctor,
    FOREIGEN KEY (clinic_name, city) REFERENCES Clinic
);
```

Relation: DrugCompany (DCname, phone_num)

Primary Key: DCname

```
CREATE TABLE DrugCompany (
    DCname       VARCHAR (25) PRIMARY KEY,
    phone_num    INTEGER
);
```

Relation: Drug (Dcname, drug_name, formula)

Primary Key: DCname, drug_name

Foreign Key:

Drug (DCname) REFERENCES DrugComany (DCname)

```
CREATE TABLE Drug (
    DCname        VARCHAR (25),
    drug_name     VARCHAR (30),
    formula       VARCHAR (100),
    PRIMARY KEY (DCname, drug_name),
    FOREIGN KEY (DCname) REFERENCES DrugCompany (DCname)
);
```

ER Approach

Relation: Pharmacy (PHname, phone_num)
Online_pharmacy (PHname, WebURL)
InStore_pharmacy (PHname, address)

Primary Key: PHname for all relations

Foreign Keys:
Online_pharmacy (PHname) REFERENCES Pharmacy (PHname)
InStore_pharmacy (PHname) REFERENCES Pharmacy (PHname)

```
CREATE TABLE Pharmacy (
    PHname        VARCHAR (25) PRIMARY KEY,
    phone_num    INTEGER
);

CREATE TABLE Online_pharmacy (
    PHname      VARCHAR (25) PRIMARY KEY,
    WebURL      VARCHAR (50),
    FOREIGN KEY (PHname) REFERENCES Pharmacy (PHname)
);

CREATE TABLE InStore_pharmacy (
    PHname    VARCHAR (25) PRIMARY KEY,
    address     VARCHAR (50),
    FOREIGN KEY (PHname) REFERENCES Pharmacy (PHname)
);
```

Relations: Sell (PHname, DCname, drug_name, price)

Primary Key: PHname, DCname, durg_name

Foreign Key:

Sell (PHname) REFERENCES Pharmacy (PHname)

Sell (DCname, drug_name) REFERENCES Drug (DCname, drug_name)

```
CREATE TABLE Sell (
    PHname    VARCHAR (25),
    DCname    VARCHAR (25),
    drug_name VARCHAR (30),
    price         FLOAT,
    PRIMARY KEY (PHname, DCname, drug_name),
    FOREIGN KEY (PHname) REFERENCE Pharmacy (PHname),
    FOREIGN KEY (DCname, drug_name) REFERNECE Drug (DCname, drug_name)
);
```

Relation: Contract (PHname, DCname, text, supervisor, end_date, start_date)

Primary Key: PHname, DCname

Foreign Key:

Contract (PHname) REFERENCES Pharmacy (PHname)

Contract (DCname) REFERENCES DrugCompany (DCname)

```
CREATE TABLE Contract (
    PHname       VARCHAR (25),
    DCname       VARCHAR (25),
    text         VARCHAR (100),
    supervisor   VARCHAR (50),
    end_date     DATE,
    start_date   DATE,
    PRIMARY KEY (PHname, DCname),
    FOREIGN KEY (PHname) REFERENCES InStore_pharmacy (PHname),
    FOREIGN KEY (DCname) REFERENCES DrugCompany (DCname)
);
```

Question 2 (a)

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

Yes, it violates the primary key constraint. The primary keys of the R3 relation are M, N, and O, and tuple (a1, b10, d20) already exist in the third row of R3(M, N, O). Duplicate default key values are not allowed and cannot be inserted.

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

No, the primary key to the R4 relation is J and K, and R4 J and K is the foreign key of each R5(S) and R2(D). R4(J) may be inserted into R5(S) because s500 already exists, and R4(K) may be inserted into R2(D) because d20 already exists, so that tuples (s500, d20, 75) may be inserted into R4.

Question 2 (b)

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

R2(D) is referred to in R3(O) and R4(K). Since d30 does not exist in R3(O) and d30 does not exist in R4(K), it does not violate the foreign key constraints. This means that nothing is deleted from R3 and R4.

Relation R2 (Deleted)

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)**

R5(S) is referred to in R1(C) and R4(J). Since s400 does not exist in R1(C) and R4(J), it does not violate the foreign key constraints. This means that nothing is updated in R3 and R4.

Relation R5 (Updated)

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)**

R5(S) is referred to in R1(C) and R4(J). Since s100 does not exist in R4(J), R4 is not updated because it does not violate the foreign key. However, since s100 exists in R1(C), the "SET NULL" policy of the update operation is used because it violates the foreign key constraints.

Relation R5 (Updated)

S	T	U
6000	60	666
s200	20	333
s300	30	111
s400	30	555
s500	40	444

Relation R1 (Updated 'SET NULL')

A	B	C
a1	b10	NULL
a2	b10	s300
a2	b20	s200
a3	b10	s500
a4	b20	NULL

Question 2 (c)

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

If all the tuples of R5 are deleted, R2 will remain unchanged, and all the tuples of R3 will be empty. Because R1(C) and R4(J) have foreign key constraints in each R5(S), when R5 is deleted, all tuples of R1 and R4 are deleted. Since R3(M, N) has foreign key constraints in R1(A, B), all tuples of R3 will also be deleted. On the other hand, R2 does not have any foreign key, so no tuple is deleted.