Cpts 451 HW 2

Zach Fechko (011711215)

2/1/23

Question 1

Checklist

Entities

- ☒ Patient
- ☒ Doctor
- ☒ Clinic
- ☒ Presciption
- ☒ Drug
- ☒ Drug Company
- ☒ Pharmacy (online or physical)

Relationships

- ☒ Works for (doctor – clinic)
- ☒ Prescribes (doctor – presciption)
- ☒ Includes (drug – presciption)
- ☒ Sells (pharmacy – drug)
- ☐ given in (prescription – clinic)
- ☐ produces (drug – drug company)
- ☒ contract (drug company – pharmacy)

```
CREATE TABLE patient(
    pat_SSN char(9) PRIMARY KEY,
    phys_SSN char(9) NOT NULL, --enforces total participation
    age int,
    first_name varchar(20),
```

```
    last_name varchar(20),
    address varchar(50)
    FOREIGN KEY (phys_SSN) REFERENCES doctor(phys_SSN)
);

CREATE TABLE doctor(
    phys_SSN char(9) PRIMARY KEY,
    first_name varchar(20),
    last_name varchar(20),
    specialty varchar(20),
    experience int
);

CREATE TABLE clinic(
    clinic_name varchar(20),
    city varchar(20)
    PRIMARY KEY (clinic_name, city)
);

CREATE TABLE works_for( --relation between doctor and clinic
    phys_SSN char(9),
    clinic_name varchar(20),
    city varchar(20),
    PRIMARY KEY (phys_SSN, clinic_name, city),
    FOREIGN KEY (phys_SSN) REFERENCES doctor(phys_SSN),
    FOREIGN KEY (clinic_name, city) REFERENCES clinic(clinic_name, city)
);

CREATE TABLE presciption(
    pres_num int,
    pres_date date,
    clinic_name varchar(20),
    city varchar(20)
    PRIMARY KEY (pres_num, clinic_name, city),
    FOREIGN KEY (clinic_name, city) REFERENCES clinic(clinic_name, city)
);

CREATE TABLE prescribes( --relation between doctor and presciption
    pres_num int,
    clinic_name varchar(20),
    city varchar(20),
```

```
    pat_SSN char(9),
    phys_SSN char(9),
    PRIMARY KEY (pres_num, clinic_name, city, pat_SSN, phys_SSN),
    FOREIGN KEY (pres_num, clinic_name, city) REFERENCES presciption(pres_num, clinic_name
    FOREIGN KEY (pat_SSN) REFERENCES patient(pat_SSN),
    FOREIGN KEY (phys_SSN) REFERENCES doctor(phys_SSN)
)

CREATE TABLE drug_company(
    company_name varchar(20) PRIMARY KEY,
    phone_number char(10)
);

CREATE TABLE drug(
    drug_name varchar(20),
    company_name varchar(20),
    formula varchar(20),
    PRIMARY KEY (drug_name, company_name),
    FOREIGN KEY (company_name) REFERENCES drug_company(company_name)
);

CREATE TABLE includes( --relation between drug and prescription
    pres_num int,
    clinic_name varchar(20),
    city varchar(20),
    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 presciption(pres_num, clinic_name
    FOREIGN KEY (drug_name, company_name) REFERENCES drug(drug_name, company_name)
);

CREATE TABLE pharmacy( --pharmacy superclass
    pharmacy_name varchar(20),
    phone_num varchar(10),
    PRIMARY KEY (pharmacy_name)
);

CREATE TABLE online_pharmacy( --online pharmacy subclass
    pharmacy_name varchar(20) PRIMARY KEY,
    phone_num char(10),
```

```
    pharm_url varchar(50),
    FOREIGN KEY (pharmacy_name, phone_num) REFERENCES pharmacy(pharmacy_name, phone_num)
);

CREATE TABLE physical_pharmacy ( --physical pharmacy subclass
    pharmacy_name varchar(20) PRIMARY KEY,
    phone_num char(10),
    pharm_address varchar(50),
    FOREIGN KEY (pharmacy_name, phone_num) REFERENCES pharmacy(pharmacy_name, phone_num)
);

CREATE TABLE sells ( --relationship between pharmacy and drug
    pharmacy_name varchar(20),
    drug_name varchar(20),
    company_name varchar(20),
    price money,
    PRIMARY KEY (pharmacy_name, drug_name, company_name),
    FOREIGN KEY (pharmacy_name) REFERENCES pharmacy(pharmacy_name),
    FOREIGN KEY (drug_name, company_name) REFERENCES drug(drug_name, company_name)
);

CREATE TABLE contract( --relationship between drug company and physical pharmacy
    pharmacy_name varchar(20),
    company_name varchar(20),
    con_text varchar(255),
    supervisor varchar(30),
    start_date date,
    end_date date,
    PRIMARY KEY (pharmacy_name, company_name),
    FOREIGN KEY (pharmacy_name) REFERENCES physical_pharmacy(pharmacy_name),
    FOREIGN KEY (company_name) REFERENCES drug_company(company_name)
);
```

Question 2

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.

2.a

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

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

2.b

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 `SET NULL` policy for update operations. Update the tables after applying those policies.

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

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

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

2.c

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

If all the tuples in R5 get deleted, it will cause the tuples in R1 and R4 to be delteed as well. Because R3 references R1, the tuples in R3 will be deleted as well. The tuples in R2 will not be deleted because R2 does not contain any foreign keys.