CS2102: Database Systems

Tutorial #3: *Entity Relationship Model*

Week 5 Guide

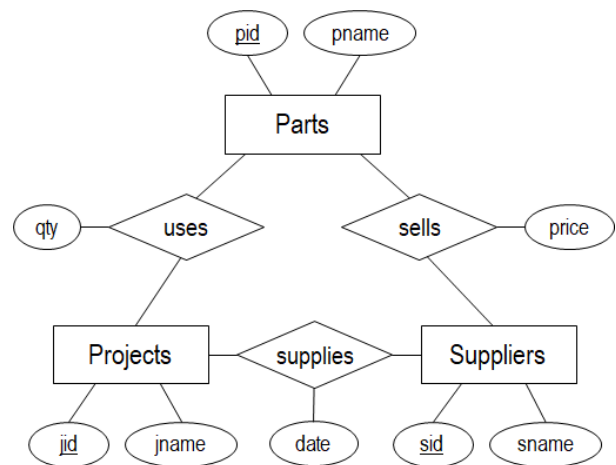AY 2022/23 Sem 2

1 Discussions

The following questions are to be discussed during tutorial. All answers will be released with explanation.

1. **(ER Analysis)** Consider an application about the parts (*with identifier* `pid` *and name* `pname`) supplied by suppliers (*with identifier* `sid` *and name* `sname`) to projects (*with identifier* `jid` *and name* `jname`) where `price` represents the unit price of a part, `qty` represents the quantity of a part and `date` represents the date of transaction.
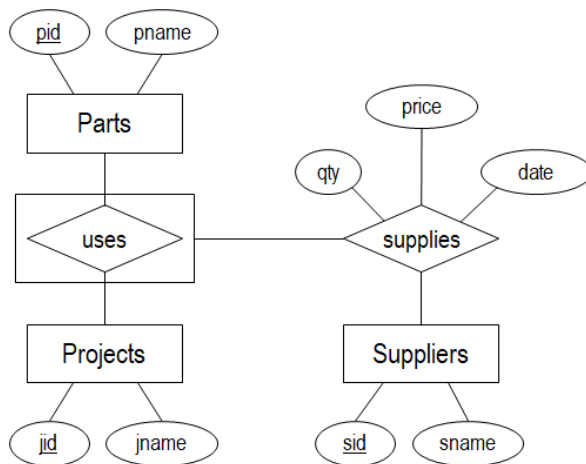
   The following are three different ER designs for the application:
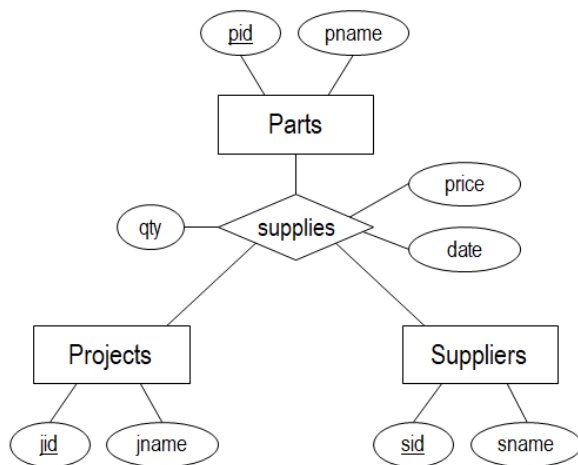
   ERD : Design A (*ternary*)   ERD : Design B (*binary*)

ERD : Design C (*aggregate*)

(a) Discuss whether the designs are *equivalent* in the sense of capturing the same constraints of the application.

(b) Translate each of the ER diagram into a relational schema. Assume appropriate domains for the attributes.

Suggested Guide:

(a) In Design A, if (S, P, J) is in `Supplies` relationship set, the supplier S supplies part P to project J. Moreover, the value of each of the relationship attributes `date`, `qty` and `price` depends on the (S, P, J) triple. For example, supplier S could be selling part P to project J for \$10 each BUT the same supplier S could be selling the same part P to another project J' for \$15 each.

When the ternary relationship in Design A is replaced by a collection of binary relationship in Design B, the semantics are not the same. First, each of the value of the relationship attributes `date`, `qty` and `price` in Design B depends only on a pair of entities. For example, the price of a part P sold by supplier S is fixed for *all projects*.

Second, even if:

- (S, P) is in `Sells` relationship set (*i.e., supplier S sells part P*).
- (S, J) is in `Supplies` relationship set (*i.e., supplier S supplies to project J*).
- (J, P) is in `Uses` relationship set (*i.e., project J uses part P*).

The three facts do **NOT** necessarily mean that supplier S sells part P to project J.

Design C is more *general* than Design A as the former (*i.e.*, Design C) has an additional `Uses` relationship set that captures the parts that are used by projects independent of whether a part is being supplied by any supplier.

> **Note:**
> For the described application, Design A is sufficient.

(b) **Design A**

```
1   DROP TABLE IF EXISTS
2       Parts, Projects, Suppliers, Supplies CASCADE;
3
4   CREATE TABLE Parts (
5     pid    INTEGER PRIMARY KEY,
6     pname  TEXT
7   );
8
```

```
 9  CREATE TABLE Projects (
10    jid    INTEGER PRIMARY KEY,
11    jname  TEXT
12  );
13
14  CREATE TABLE Suppliers (
15    sid    INTEGER PRIMARY KEY,
16    sname  TEXT
17  );
18
19  CREATE TABLE Supplies (
20    pid     INTEGER REFERENCES Parts,
21    jid     INTEGER REFERENCES Projects,
22    sid     INTEGER REFERENCES Suppliers,
23    price   NUMERIC,
24    qty     INTEGER,
25    t_date  DATE,
26    PRIMARY KEY (pid, jid, sid)
27  );
```

Design B

```
 1  DROP TABLE IF EXISTS
 2    Parts, Projects, Suppliers, Uses, Sells, Supplies
 3  CASCADE;
 4
 5  CREATE TABLE Parts (
 6    pid    INTEGER PRIMARY KEY,
 7    pname  TEXT
 8  );
 9
10  CREATE TABLE Projects (
11    jid    INTEGER PRIMARY KEY,
12    jname  TEXT
13  );
14
15  CREATE TABLE Suppliers (
16    sid    INTEGER PRIMARY KEY,
17    sname  TEXT
18  );
19
20  CREATE TABLE Uses (
21    pid     INTEGER REFERENCES Parts,
22    jid     INTEGER REFERENCES Projects,
23    qty     INTEGER,
24    PRIMARY KEY (pid, jid)
25  );
26
27  CREATE TABLE Sells (
28    pid     INTEGER REFERENCES Parts,
29    sid     INTEGER REFERENCES Suppliers,
30    price   NUMERIC,
31    PRIMARY KEY (pid, sid)
32  );
33
34  CREATE TABLE Supplies (
35    jid     INTEGER REFERENCES Projects,
36    sid     INTEGER REFERENCES Suppliers,
37    t_date  DATE,
38    PRIMARY KEY (jid, sid)
39  );
```

Design C

```sql
DROP TABLE IF EXISTS
  Parts, Projects, Suppliers, Uses, Supplies
CASCADE;

CREATE TABLE Parts (
  pid    INTEGER PRIMARY KEY,
  pname  TEXT
);

CREATE TABLE Projects (
  jid    INTEGER PRIMARY KEY,
  jname  TEXT
);

CREATE TABLE Suppliers (
  sid    INTEGER PRIMARY KEY,
  sname  TEXT
);

CREATE TABLE Uses (
  pid     INTEGER REFERENCES Parts,
  jid     INTEGER REFERENCES Projects,
  PRIMARY KEY (pid, jid)
);

CREATE TABLE Supplies (
  pid     INTEGER,
  jid     INTEGER,
  sid     INTEGER REFERENCES Suppliers,
  price   NUMERIC,
  qty     INTEGER,
  t_date  DATE,
  PRIMARY KEY (pid, jid, sid),
  FOREIGN KEY (pid, jid) REFERENCES Uses
);
```
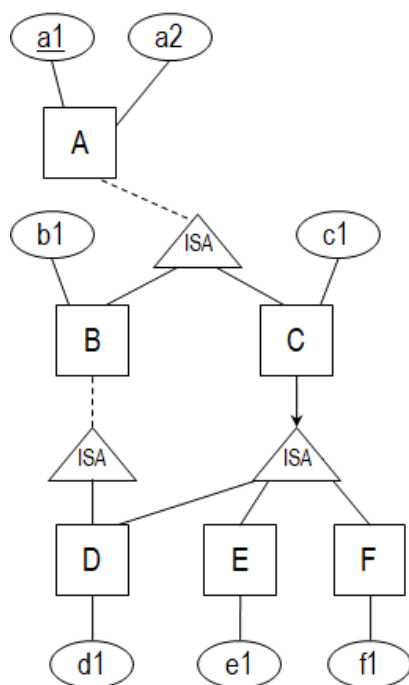
2. **(ER Translation)** Consider the ER diagrams shown below.
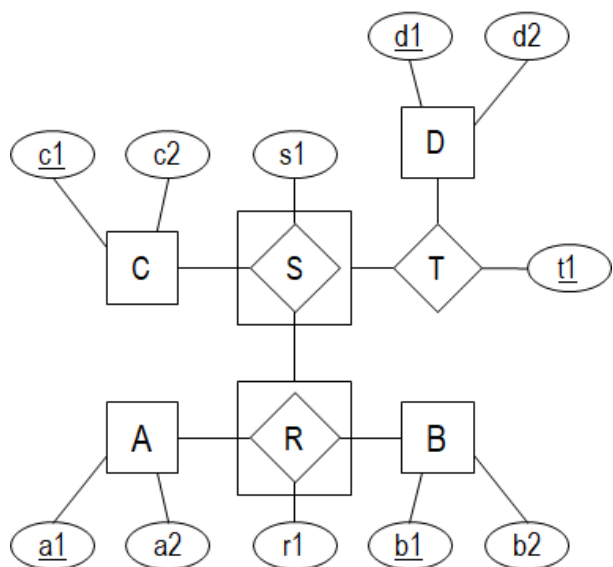
ERD : ER 1
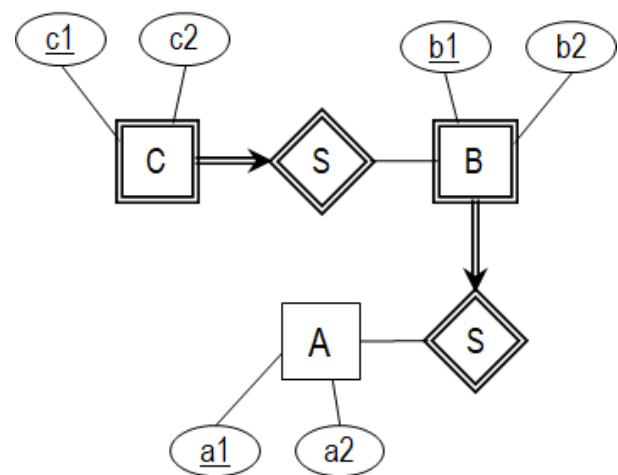
ERD : ER 3

ERD : ER 2

Translate each ER diagram into a relational schema. In your translation

- enforce as many constraints captured in the ER diagram as possible
- do not enforce additional constraints that are not captured in the ER diagram
- assume that all the attributes have integer domain

Suggested Guide:

The idea is to start translating the ER diagram from the construct (*entity set, relationship set, or aggregate*) with the *least* number of foreign key. Once this is done, we select from the remaining constructs, the next smallest number of foreign key.

The process can be repeated until all constructs are translated. If there are *circular foreign key*, break the dependency and use `ALTER TABLE` after all relations are translated.

(a) **ER 1**

```
1   DROP TABLE IF EXISTS A, B, C, D, R, S, T CASCADE;
2
3   CREATE TABLE A (
4     a1  INTEGER PRIMARY KEY,
5     a2  INTEGER
6   );
7
8   CREATE TABLE B (
9     b1  INTEGER PRIMARY KEY,
10    b2  INTEGER
11  );
12
13  CREATE TABLE C (
14    c1  INTEGER PRIMARY KEY,
15    c2  INTEGER
16  );
17
18  CREATE TABLE D (
19    d1  INTEGER PRIMARY KEY,
20    d2  INTEGER
21  );
22
23  CREATE TABLE R (
24    a1  INTEGER REFERENCES A,
25    b1  INTEGER REFERENCES B,
26    r1  INTEGER,
27    PRIMARY KEY (a1, b1)
28  );
29
30  CREATE TABLE S (
31    a1  INTEGER,
32    b1  INTEGER,
33    c1  INTEGER REFERENCES C,
34    s1  INTEGER,
35    PRIMARY KEY (a1, b1, c1),
36    FOREIGN KEY (a1, b1) REFERENCES R
37  );
38
39  CREATE TABLE T (
40    a1  INTEGER,
41    b1  INTEGER,
42    c1  INTEGER,
43    d1  INTEGER REFERENCES D,
44    t1  INTEGER,
45    PRIMARY KEY (a1, b1, c1, d1, t1),
46    FOREIGN KEY (a1, b1, c1) REFERENCES S
47  );
```

(b) **ER 2**

```
1   DROP TABLE IF EXISTS A, B, C, D, E, F CASCADE;
2
3   CREATE TABLE A (
4     a1  INTEGER PRIMARY KEY,
5     a2  INTEGER
6   );
7
8
9
```

```
10   CREATE TABLE B (
11     a1   INTEGER PRIMARY KEY REFERENCES A
12           ON DELETE CASCADE ,
13     b1   INTEGER
14   );
15
16   CREATE TABLE C (
17     a1   INTEGER PRIMARY KEY REFERENCES A
18           ON DELETE CASCADE ,
19     c1   INTEGER
20   );
21
22   CREATE TABLE D (
23     a1   INTEGER PRIMARY KEY REFERENCES B REFERENCES C
24           ON DELETE CASCADE ,
25     d1   INTEGER
26   );
27
28   CREATE TABLE E (
29     a1   INTEGER PRIMARY KEY REFERENCES C
30           ON DELETE CASCADE ,
31     e1   INTEGER
32   );
33
34   CREATE TABLE F (
35     a1   INTEGER PRIMARY KEY REFERENCES C
36           ON DELETE CASCADE ,
37     f1   INTEGER
38   );
```

(c) **ER 3**

```
1    DROP TABLE IF EXISTS A , B , C CASCADE;
2
3    CREATE TABLE A (
4      a1   INTEGER PRIMARY KEY ,
5      a2   INTEGER
6    );
7
8    CREATE TABLE B (
9      a1   INTEGER REFERENCES A
10           ON DELETE CASCADE ,
11     b1   INTEGER ,
12     b2   INTEGER ,
13     PRIMARY KEY (a1, b1)
14   );
15
16   CREATE TABLE C (
17     a1   INTEGER ,
18     b1   INTEGER ,
19     c1   INTEGER ,
20     c2   INTEGER ,
21     PRIMARY KEY (a1, b1, c1),
22     FOREIGN KEY (a1, b1) REFERENCES B
23           ON DELETE CASCADE
24   );
```

2 Challenge

The answers to the following questions is given without explanation. Please discuss them on Canvas.

1. The Prescription-R-X chain of pharmacies has offered to give you a free lifetime supply of medicine if you design its database. Given the rising cost of health care, you agree. Here's the information that you gather:

    1. Patients are identified by an SSN. Their names, addresses and ages must also be recorded.
    2. Doctors are identified by an SSN. Their name, specialty and years of experience must also be recorded.
    3. Each pharmaceutical company is identified by name and has a phone number.
    4. For each drug, the *trade name* and formula must be recorded.
    5. Each drug is sold by a given pharmaceutical company and the trade name identifies a drug *uniquely* from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
    6. Each pharmacy has a name, address and phone number.
    7. Every patient has a primary physician.
    8. Every doctor is the primary physician of at least one patient.
    9. Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies and the price could vary from one pharmacy to another.
    10. Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
    11. There is exactly one contract between a pharmacy and a pharmaceutical company if and only if that pharmacy sells some drug that is made by that pharmaceutical company. For each contract, you have to store a start date, an end date and the text of the contract.

    Questions:

    (a) Consider the ER diagram shown on the next page for Prescriptions-R-X. What are the constraints that are not captured by this design? Modify the ER design to capture as many of the constraints as possible.
    (b) Translate your ER design in part (a) into a relational schema using SQL (*assume reasonable data types for the domain constraints*). Your solution should capture as many of the application's constraints as possible. Identify any constraints that are not captured by your relational schema.
    (c) How would your design in part (a) change if each drug must be sold at a fixed price by all pharmacies?
    (d) How would your design in part (a) change if the design requirements change as follows: "*if a doctor prescribes the same drug for the same patient more than once, several such prescriptions may have to be stored*".
    (e) Suppose that pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract but the contract supervisor can change over the lifetime of the contract. Supervisors are identified by an SSN and their start dates must be recorded. Modify your ER design in part (a) to capture this additional requirement.

(f) Translate your ER design in part (e) into a relational schema. Identify any constraints that are not captured by your relational schema.
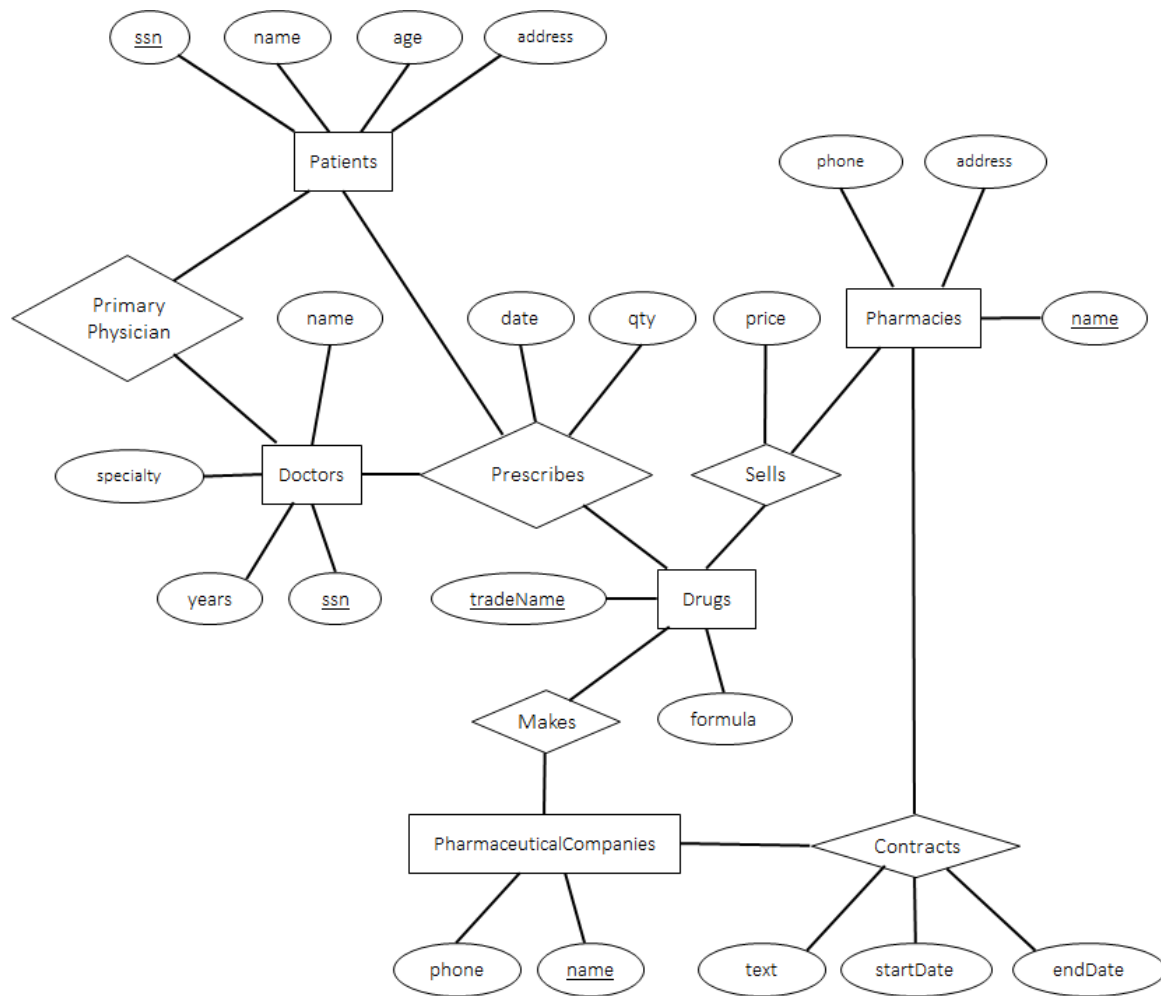
Suggested Guide:

(a) The following constraints are not captured by the ER design:

C1. `Drugs` is a weak entity set that is dependent on `PharmaceuticalCompanies` (*i.e.*, *the owning entity set is PharmaceuticalCompanies*).

C2. The key and total participation constraints on `Patients` w.r.t. `PrimaryPhysician` relationship.

C3. The total participation constraint on `Doctors` w.r.t. `PrimaryPhysician` relationship.

C4. The constraint that each pharmacy sells more than one drug.

C5. The constraint that there is exactly one contract between a pharmaceutical company and a pharmacy if and only if that pharmacy sells some drug that is made by that pharmaceutical company.
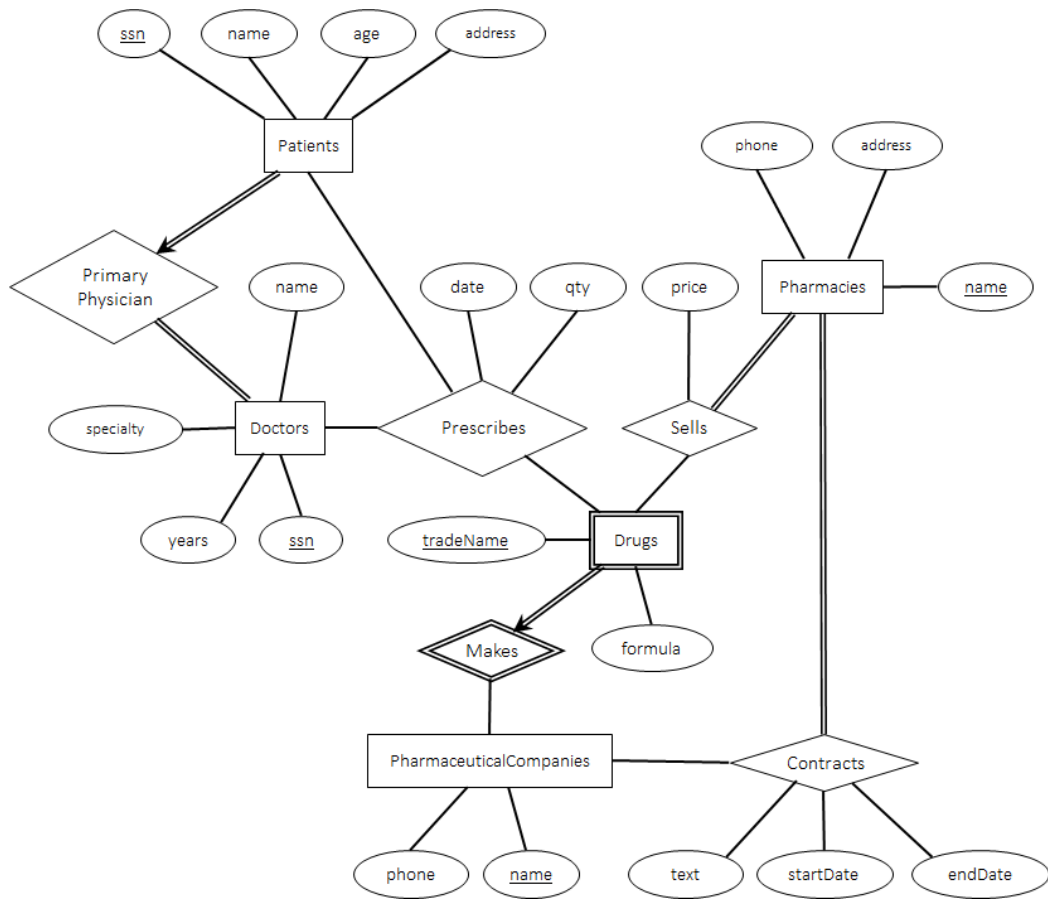
(b) Relational schema

```
1   DROP TABLE IF EXISTS
2     Doctors, Patients, Pharmacies, Drugs, Sells,
3     PharmaceuticalCompanies, Prescribes, Contracts
4   CASCADE;
5
6   CREATE TABLE Doctors (
7     ssn        TEXT PRIMARY KEY,
8     name       TEXT,
9     specialty  TEXT,
10    years      INTEGER
11  );
12
13  CREATE TABLE Patients (
14    ssn        TEXT PRIMARY KEY,
15    physician  TEXT NOT NULL REFERENCES Doctors,
16    name       TEXT,
17    address    TEXT,
18    age        INTEGER
19  );
20  CREATE TABLE Pharmacies (
21    name       TEXT PRIMARY KEY,
22    phone      TEXT,
23    address    TEXT,
24  );
25
26  CREATE TABLE PharmaceuticalCompanies (
27    name       TEXT PRIMARY KEY,
28    phone      TEXT
29  );
30
31  CREATE TABLE Drugs (
32    pcname     TEXT REFERENCES PharmaceuticalCompanies
33                  ON DELETE CASCADE,
34    tradename  TEXT,
35    formula    TEXT,
36    PRIMARY KEY (pcname, tradename)
37  );
38
39  CREATE TABLE Prescribes (
40    dssn       TEXT REFERENCES Doctors,
41    pssn       TEXT REFERENCES Patients,
42    pcname     TEXT REFERENCES PharmaceuticalCompanies
43                  ON DELETE CASCADE,
44    tradename  TEXT,
45    pdate      DATE,
46    qty        INTEGER,
47    PRIMARY KEY (dssn, pssn, pcname, tradename),
48    FOREIGN KEY (pcname, tradename) REFERENCES Drugs
49  );
50
51  CREATE TABLE Contracts (
52    pname      TEXT REFERENCES Pharmacies,
53    pcname     TEXT REFERENCES PharmaceuticalCompanies
54                  ON DELETE CASCADE,
55    start_date DATE,
56    end_date   DATE,
57    comments   TEXT,
58    PRIMARY KEY (pname, pcname)
59  );
```

```
60   CREATE TABLE Sells (
61     pname       TEXT REFERENCES Pharmacies,
62     pcname      TEXT REFERENCES PharmaceuticalCompanies
63                   ON DELETE CASCADE,
64     tradename   TEXT,
65     price       NUMERIC,
66     PRIMARY KEY (pname, pcname, tradename),
67     FOREIGN KEY (pname, pcname) REFERENCES Contracts,
68     FOREIGN KEY (pcname, tradename) REFERENCES Drugs
69   );
```

The above relational schema does not enforce constraints C3 and C4. Constraint C5 is enforced only partially: the foreign key constraint from `Sells` to `Contracts` guarantees that if a pharmacy sells some drug made by a pharmaceutical company, then there is a contract between that pharmacy and pharmaceutical company. However, it is possible for a contract to exist in the database between a pharmacy and a pharmaceutical company where that pharmacy does not sell any drug made by that pharmaceutical company.

(c) Instead of modeling `price` as an attribute of `Sells` relationship set, we model `price` as an attribute of `Drugs` entity set.

(d) Let `Key(R)` denote the primary key of the relation `R`. The `date` attribute of `Prescribes` is changed to a key attribute. Thus, `Key(Prescribes)` consists of the following attributes: `Key(Patients)`, `Key(Drugs)` and `date`, where `Key(Patients)` = {ssn} and `Key(Drugs` = {name, tradename}.

(e) Consider the following ER design which introduces a new *ternary* relationship set called "Supervises" to related `Pharmacies`, `PharmaceuticalCompanies` and a new entity set "Supervisors". Note that if a pharmacy P and a supervisor S is related by a `Supervises` relationship, then it is implicitly means that S is appointed by P. Since each pharmacy has at least one contract, a total participation constraint on `Pharmacies` w.r.t. `Supervises` is required.
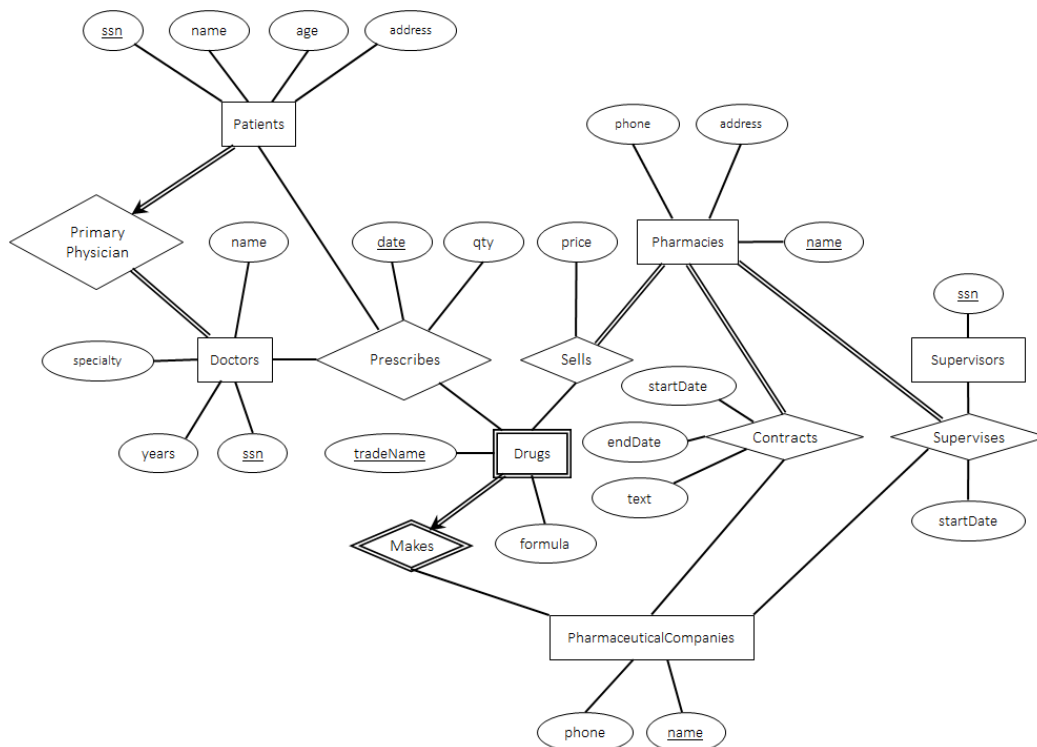
The revised ER design does not completely capture the new requirement: the existence of (P, PC, S) in `Supervises` relationship does not necessarily mean that there exist a contract between pharmacy P and a pharmaceutical company PC. Furthermore, this Er design also does not capture the constraint that each contract is supervised by only one contractor at any time. Finally, this ER design also does not allow a supervisor to supervise the same contract multiple times (*over a different time periods*).

An alternative ER design (*shown below*) is to model `Supervises` as a binary relationship between `Contract` (*as an* **aggregation**) and `Supervisors`. Also, the attribute `start_date` (*startDate in the ER design*) is changed to be part of the key of `Supervises`. This ensures that each contract will be supervised by some supervisor and a supervisor could supervise the same contract multiple times.
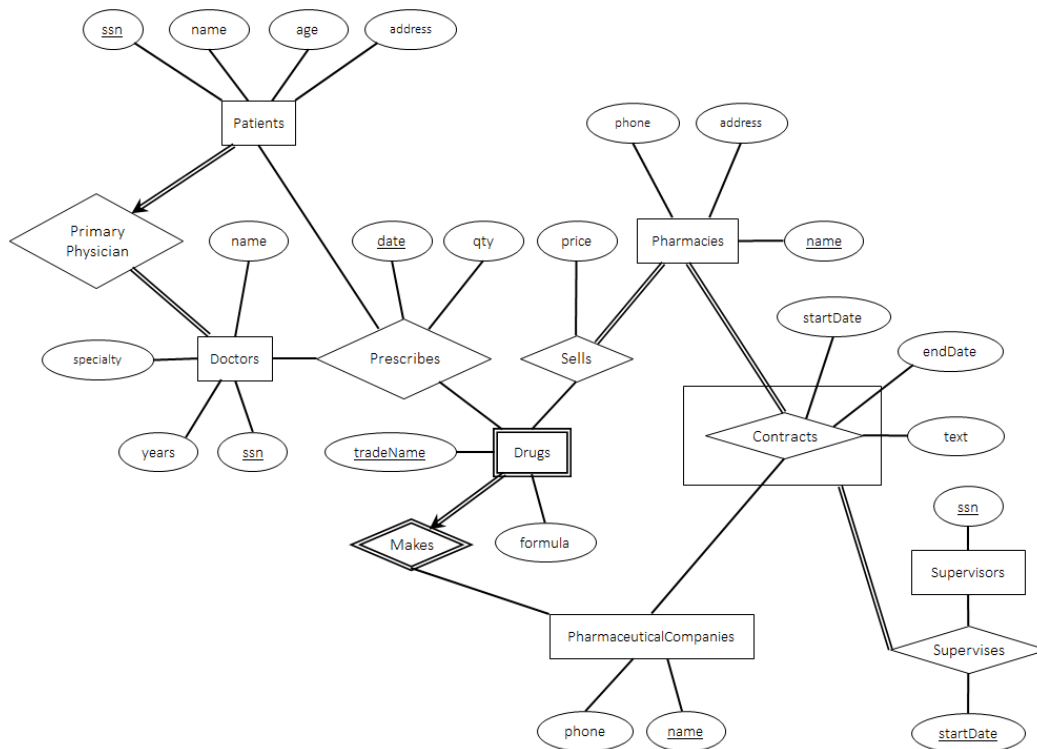
Similar to the previous design, this design does not capture the constraint that each contract should be supervised by only one supervisor at any time.

(f) The following shows the additional relational tables derived from the last aggregation-based ER diagram.

```
1    CREATE TABLE Supervisors (
2      ssn          TEXT PRIMARY KEY
3    );
4
5    CREATE TABLE Supervises (
6      pcname       TEXT,
7      pname        TEXT,
8      ssn          TEXT NOT NULL REFERENCES Supervisors,
9      start_date DATE,
10     PRIMARY KEY (pname, pcname, start_date),
11     FOREIGN KEY (pname, pcname) REFERENCES Contracts
12   );
13
```

The primary key of `Supervises` ensures that there can't be more than one supervisor supervising a contract at the same time. The foreign key constraint from `Supervises` to `Contracts` ensures that every supervised pair of pharmacy and pharmaceutical company indeed has a contract between them. However, the above schema does not enforce the total participation constraint of `Contracts` w.r.t. `Supervises`.