CS2102 Tutorial 3: ER Model

Wk 5, Sem 2, 2021/22

The questions that will be discussed are 1(a), 3(a), 3(b), 3(d), 3(e), and 2.

1. **Solution:**

(a) In Design A, if `(S,P,J)` is in Supplies relationship set, then 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 relationships 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, `(S,J)` is in Supplies relationship set, and `(J,P)` is in Uses relationship set, the fact that supplier S sells part P, S supplies to project J, and J uses P does not necessarily mean that supplier S sells part P to project J.

Design C is more general than Design A as the former 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. For the described application, Design A suffices.

(b)

```
-- Design A

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 Supplies (
    pid integer references Parts,
    jid integer references Projects,
    sid integer references Suppliers,
```

```
    price numeric,
    qty integer,
    t_date date,
    primary key (pid,jid,sid)
);


-- Design B

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,
    qty integer,
    primary key (pid,jid)
);

create table Sells (
    pid integer references Parts,
    sid integer references Suppliers,
    price numeric,
    primary key (pid,sid)
);

create table Supplies (
    jid integer references Projects,
    sid integer references Suppliers,
    t_date date,
    primary key (jid,sid)
);
```

```
-- Design C


create table  Parts  (
    pid integer primary key,
    pname text
);

create table  Projects  (
    jid integer primary key,
    jname text
);

create table  Uses  (
    pid integer references Parts,
    jid integer references Projects,
    primary key (pid,jid)
);

create table  Suppliers  (
    sid integer primary key,
    sname text
);

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
);
```

Note that in Design C, it is not necessary to specify a foreign key constraint for each of the attributes `pid` and `jid` in table `Supplies` as these foreign key constraints are already specified indirectly via the foreign key constraint specification for `(pid, jid)` (e.g., Supplies.pid indirectly references Parts.pid via Supplies.pid references Uses.pid and Uses.pid references Parts.pid).

2. **Solution:**

(a)

```
create table A (
    a1 integer primary key,
    a2 integer
);

create table B (
    b1 integer primary key,
    b2 integer
);

create table R (
    a1 integer references A,
    b1 integer references B,
    r1 integer,
    primary key (a1,b1)
);

create table C (
    c1 integer primary key,
    c2 integer
);

create table S (
    a1 integer,
    b1 integer,
    c1 integer references C,
    s1 integer,
    primary key (a1,b1,c1),
    foreign key (a1,b1) references R
);

create table D (
    d1 integer primary key,
    d2 integer
);


create table T (
    a1 integer,
    b1 integer,
    c1 integer,
    d1 integer references D,
    t1 integer,
```

```
    primary key (a1,b1,c1,d1,t1),
    foreign key (a1,b1,c1) references S
);
```

Note that since relationship S involves relationship R as an aggregation, the schema for table R must be defined before the schema for table S. Similarly, since relationship T involves relationship S as an aggregation, the schema for table S must be defined before the schema for table T.

(b)

```
create table A (
    a1 integer primary key,
    a2 integer
);

create table B (
    a1 integer primary key
        references A on delete cascade,
    b1 integer
);


create table C (
    a1 integer primary key
        references A on delete cascade,
    c1 integer
);

create table D (
    a1 integer primary key
        references B on delete cascade
        references C on delete cascade,
    d1 integer
);

create table E (
    a1 integer primary key
        references C on delete cascade,
    e1 integer
);


create table F (
    a1 integer primary key
        references C on delete cascade,
```

```
    f1 integer
);
```

(c)

```
create table A (
    a1 integer primary key,
    a2 integer
);

create table B (
    a1 integer references A
        on delete cascade,
    b1 integer,
    b2 integer,
    primary key (a1,b1)
);


create table C (
    a1 integer,
    b1 integer,
    c1 integer,
    c2 integer,
    primary key (a1,b1,c1),
    foreign key (a1,b1) references B
        on delete cascade
);
```

Note that since C is a weak entity set that is dependent on B, and B is itself a weak entity set that is dependent on A, the schema for table A must be defined first, followed by the schema for table B, and then the schema for table C.

3. **Solution:**

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

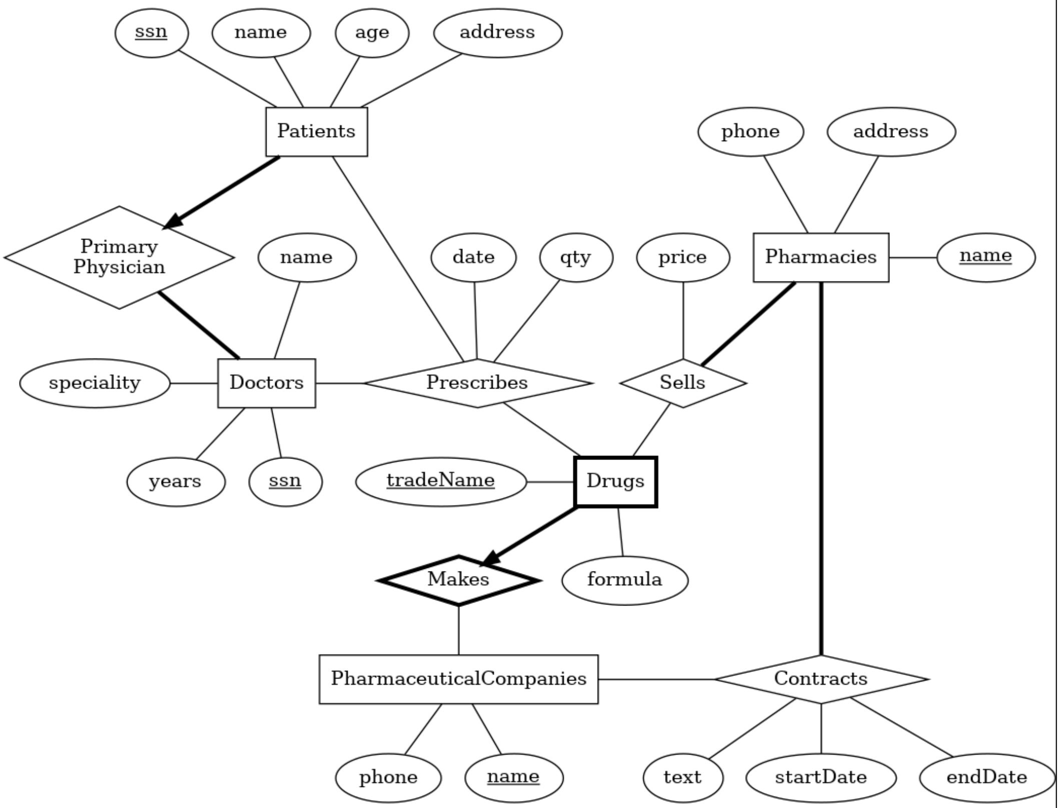
C1. Drugs is a weak entity set that is dependent on the owner entity set PharmaceuticalCompanies.

C2. The key and total participation constraints of Patients w.r.t. Primary-Physician relationship.

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

C4. The contraint 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.



The revised ER design shown above captures constraints C1, C2 and C3. Constraint C4 is captured only partially (total participation of Pharmacies w.r.t. Sells relationship) but it does not require that each pharmacy sells more than one drug. Constraint C5 is not fully captured as the ER design merely specifies that every pharmacy P must have at least one contract with some pharmaceutical company PC but the ER design does not require that P must sell some drug that is made by PC.

(b)

```
create table Doctors (
    ssn text primary key,
    name text not null,
    specialty text not null,
    years integer not null
);

create table Patients (
    ssn text primary key,
```

```
    primary_physician text not null
        references Doctors,
    name text not null,
    address text not null,
    age integer not null
);

create table Pharmacies (
    name text primary key,
    phone text,
    address text
);

create table PharmaceuticalCompanies (
    name text primary key,
    phone text
);


create table Drugs (
    pcname text references PharmaceuticalCompanies
        on delete cascade,
    tradename text,
    formula text not null,
    primary key (pcname, tradename)
);

create table Prescribes (
    dssn text references Doctors,
    pssn text references Patients,
    pcname text,
    tradename text,
    pdate date,
    qty integer,
    primary key (dssn, pssn, pcname, tradename),
    foreign key (pcname, tradename) references Drugs
        on delete cascade
);

create table Contracts (
    pcname text references PharmaceuticalCompanies(name)
        on delete cascade,
    pname text references Pharmacies(name),
    start_date date,
    end_date date,
```

```
    comments text,
    primary key (pname, pcname)
);

create table Sells (
    pcname text,
    pname text,
    tradename text,
    price numeric,
    primary key (pname, pcname, tradename),
    foreign key (pname, pcname) references Contracts,
    foreign key (pcname, tradename) references Drugs
        on delete cascade
);
```

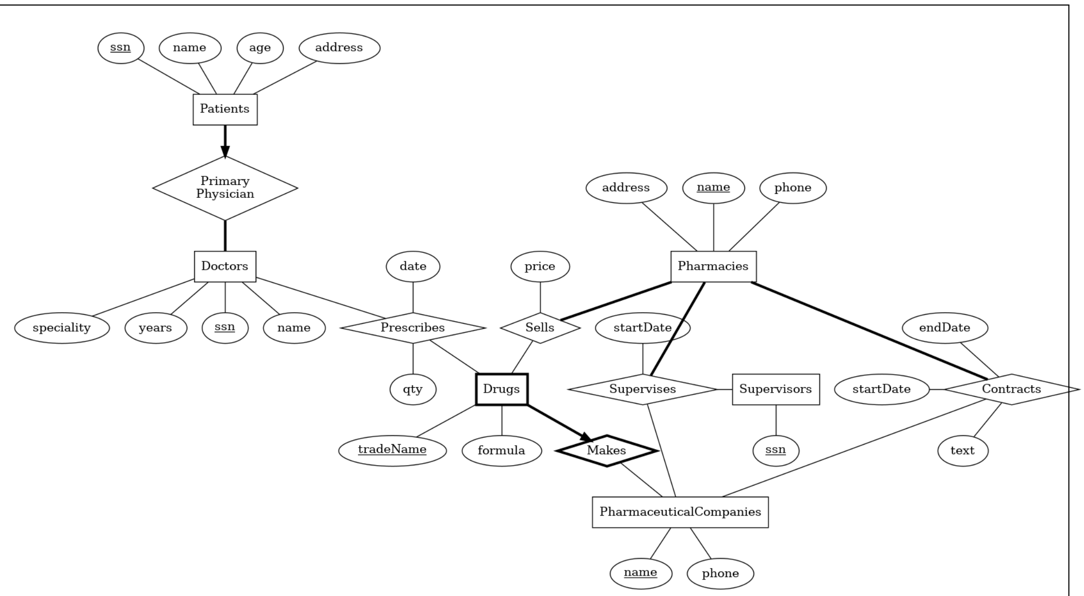
The above relational schema does not enforce constraints C3 and C4. Constraint C5 is enforced only partially: the foreign key constraint from Sells to Contracts guarantees that if a pharmacy sells some drug made by a pharmaceutical company, then there is a contract between that pharmacy and pharmaceutical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

The given answer assumes that if a pharmaceutical company is deleted, then all the records associated with it are also deleted.

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

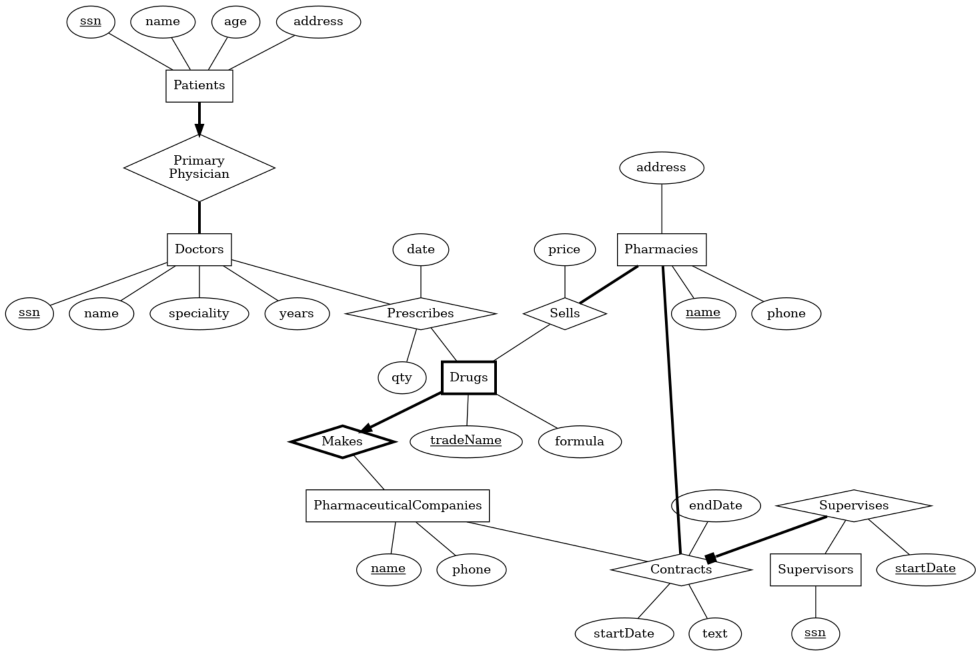
(d) The `date` attribute of `Prescribes` is changed to a key attribute. Thus, Key(Prescribes) consists of the following attributes: Key(Patients), Key(Drugs), and `date`, where Key(Patients) = {ssn} and Key(Drugs) = {name, tradename}.

(e) Consider the following ER design which introduces a new ternary relationship set "Supervises" to relate 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 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 doesn't completely capture the new requirment: the existence of (P, PC, S) in a Supervises relationship does not necessarily mean that there exists a contract between pharmacy P and pharmaceutical company PC. Furthermore, this ER design also does not capture the constraint that each contract is supervised by only one contract supervisor at any time. Finally, this ER design also does not allow a supervisor to supervise the same contract multiple times (over different time periods).

An alternative ER design (shown below) is to model `Supervises` as a binary relationship between `Contracts` (as an aggregation) and Supervisors. Also, the attribute `startDate` 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.

```
create table Supervisors (
    ssn text primary key
);

create table Supervises (
    pcname text,
    pname text,
    ssn text not null references Supervisors,
    start_date date,
    primary key (pname, pcname, start_date),
    foreign key (pname, pcname) references Contracts
);
```

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`.