2.3

Entity-Relationship Diagram:

- **Professor** (attributes: ssn, age, rank, specialty)
- **Department** (attributes: number, name, office)
- **Project** (attributes: pid, budget, start, end, sponsor)
- **Graduate** (attributes: ssn, name, age, program)

Relationships:
- **works** (time percentage) — between Professor and Department
- **runs** — between Professor and Department
- **works in** — Professor and Project
- **manages** — Professor and Project
- **supervises** — Professor and Graduate
- **major** — Department and Graduate
- **work project** (ssn) — Project and Graduate
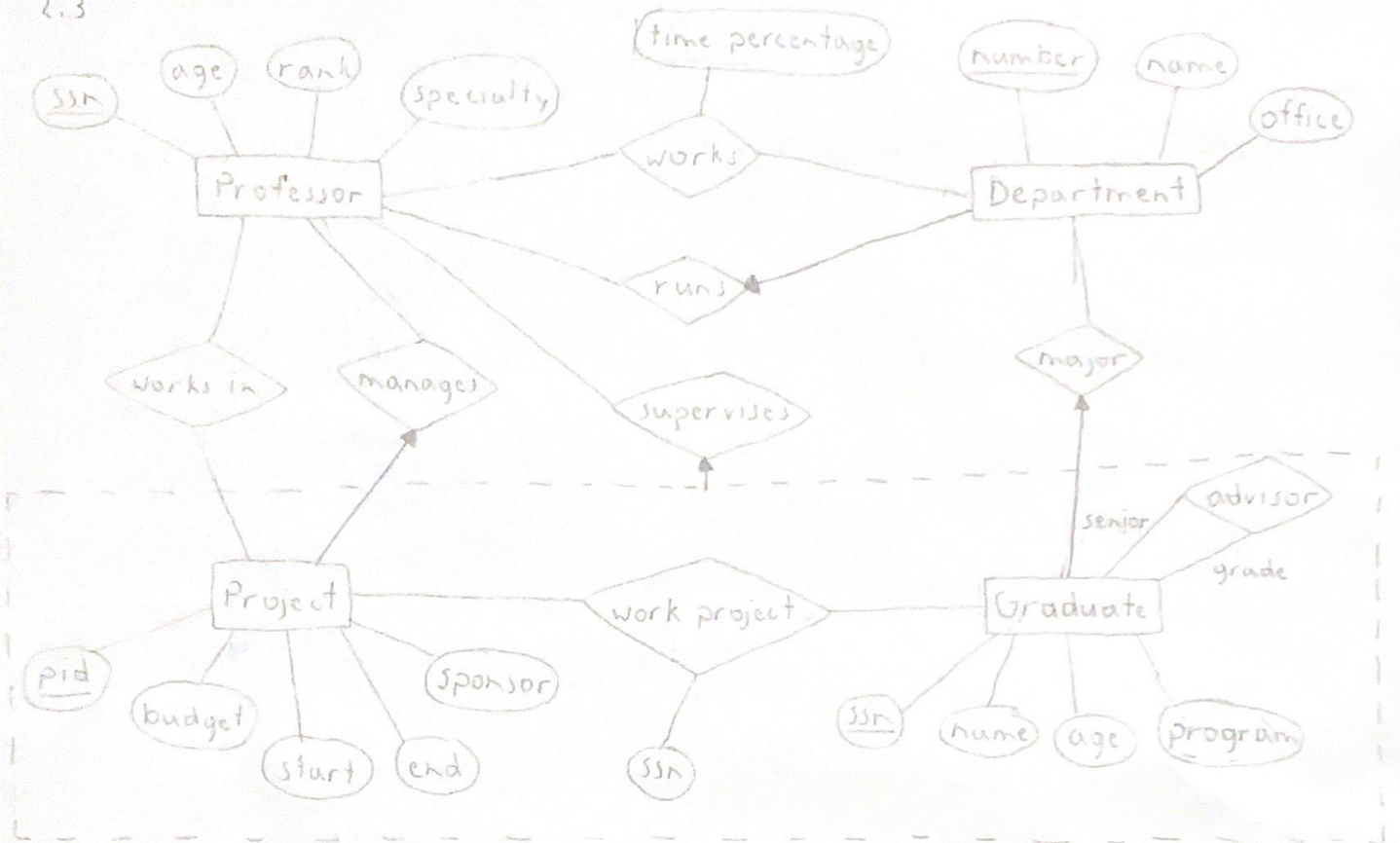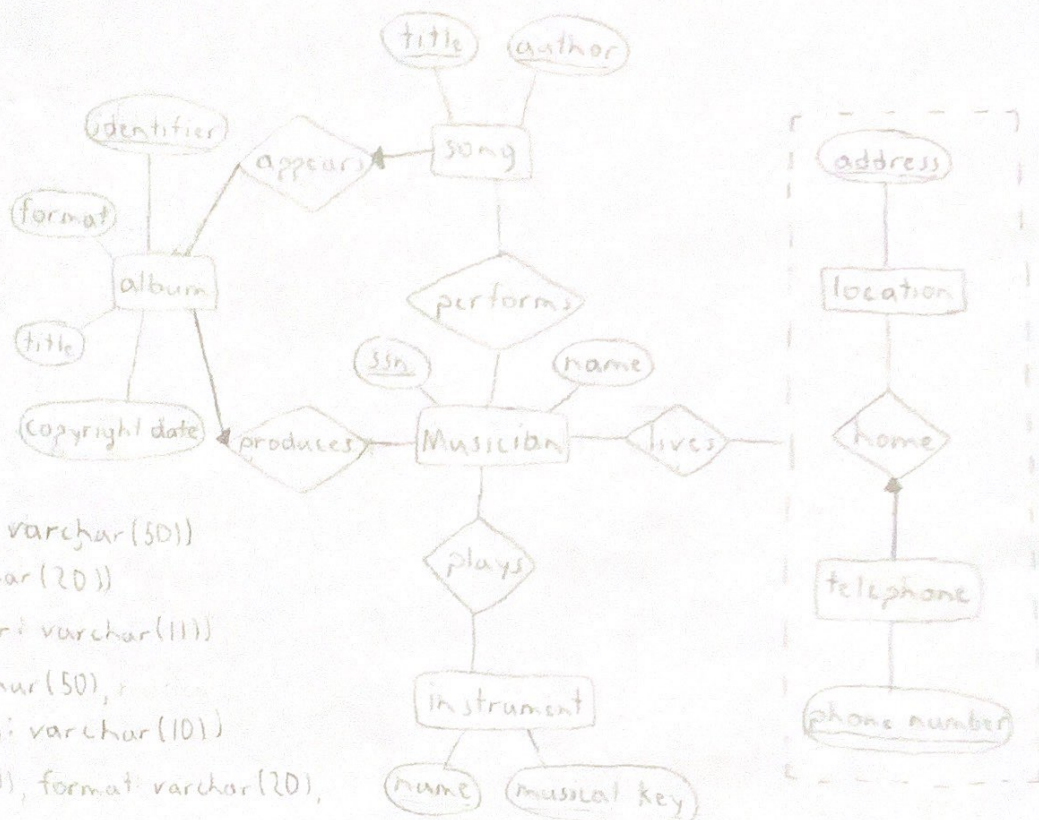- **advisor** (grade), **senior** — Graduate

2.5

Musician(ssn: int, name: varchar(50))

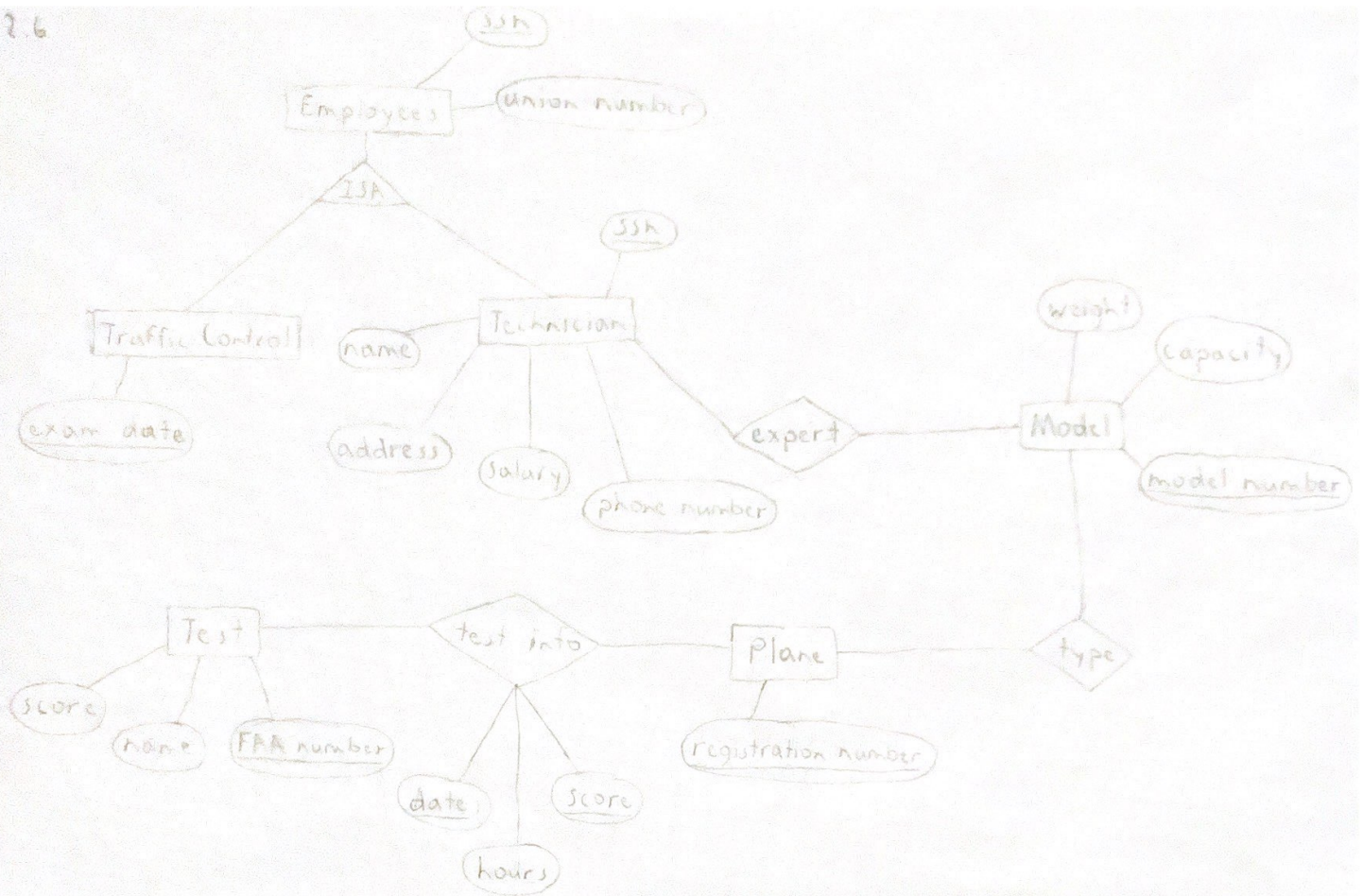Location (address: varchar(20))

Telephone(phone number: varchar(11))
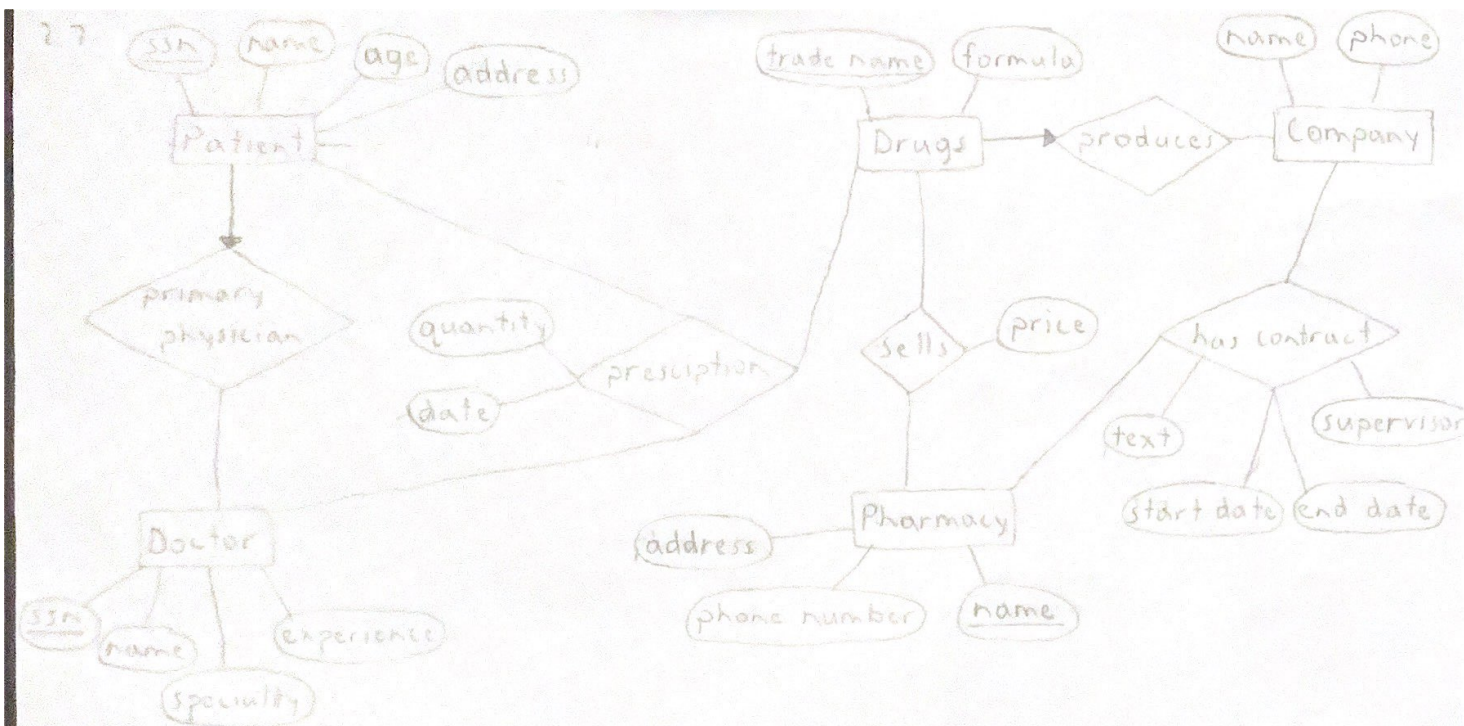
Instrument(name: varchar(50),
                musical key: varchar(10))

Album(title: varchar(50), format: varchar(20),
        copyright date: date, identifier: varchar(20))

Song(title: varchar(50), author(varchar(20))

2.6

SSn

Employees — Union number

ISA

SSn

Traffic Control — name — Technician — expert — Model — weight — Capacity

exam date — address — salary — phone number — model number

Test — test info — Plane — type

score — name — FAA number — date — score — registration number

hours

2.7

ER Diagram:

- Patient entity with attributes: ssn, name, age, address
- Drugs entity with attributes: trade name, formula
- Company entity with attributes: name, phone
- "produces" relationship between Drugs and Company
- "primary physician" relationship (Patient to Doctor)
- "presciption" relationship with attributes: quantity, date
- "Sells" relationship with attribute: price
- "has contract" relationship with attributes: text, supervisor, start date, end date
- Doctor entity with attributes: ssn, name, experience, specialty
- Pharmacy entity with attributes: address, phone number, name

2. We would add a "price" attribute to the Drugs entity set.

3. It would look like:

- prescription details entity with attributes: quantity, pid, date
- "presciption" relationship
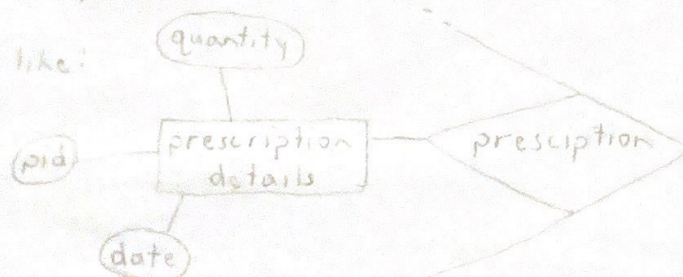
3.2. 22, cardinality, of number = number of tuples.

3.4. Candidate keys are the minimal set of attributes that uniquely identifies an element. Primary keys is the chosen candidate key(s) that uniquely identifies the record. Super keys are the set of attributes used to uniquely identify a tuple.

3.5. 1. Candidate keys are the attributes that can uniquely identify an element. Name alone isn't a candidate key because Smith appears twice. Also age isn't one because 18 appears twice so it's not enough to tell on their own. They would have to be tied together to be viable.

2. sID and emails are examples of candidate keys since each student will have unique IDs and emails.

3.6. A foreign key is an attribute in an entity that is a primary key in a different one. They are important because they refer to one table to another, implying a referential constraint between tables. Referial integrity means that when a foreign key exists, it needs to have a reference to a primary key from a different table.

3.7 1. In Enrolled: sid, cid
     In Teachers: fid, cid
     In Meets: cid, rno

   2. Standardize length of sid, cid, and fid. and putting
      limits on credit, salary, and capacity.

3.12 Professor (SSN: int, primary key(ssn))
     Courses (courseid: varchar(11), primary key (courseid))
     Teaches (SSN: int, courseid: varchar(11),
              semester: varchar(10), primary key (SSN, courseid)
              foreign key (ssn) references Professor,
              foreign key (courseid) references (courses))

3.15 Musician (ssn: int, name: varchar (50), primary key (ssn))
     Location (address: varchar(20))
     Telephone (phone number: varchar(11))
     Instrument (name: varchar(50),
                 musical key: varchar (10)
     Album (title: varchar (50), format: varchar (20),
            copyright date: date, identifier: varchar(20))
     Song (title: varchar(20), author: varchar (20))
     Plays (ssn: int, primary key(ssn),
            foreign key (ssn) represents Musician
     Home (phone number: varchar (11),
           address: varchar(20), primary key(phone #, location)
           foreign key (address) represents Location
           foreign key (phone number) represents Telephone)

Performs (ssn: int, title: varchar(50),
            author: varchar(20), primary key (title, author),
            foreign key (ssn) represents Musician
            foreign key (title) represents Song
            foreign key (author) represents Song)
Produces (ssn: int, identifier: varchar(20),
            primary key (ssn, identifier),
            foreign key (ssn) represents Musician
            foreign key (identifier) represents Album)