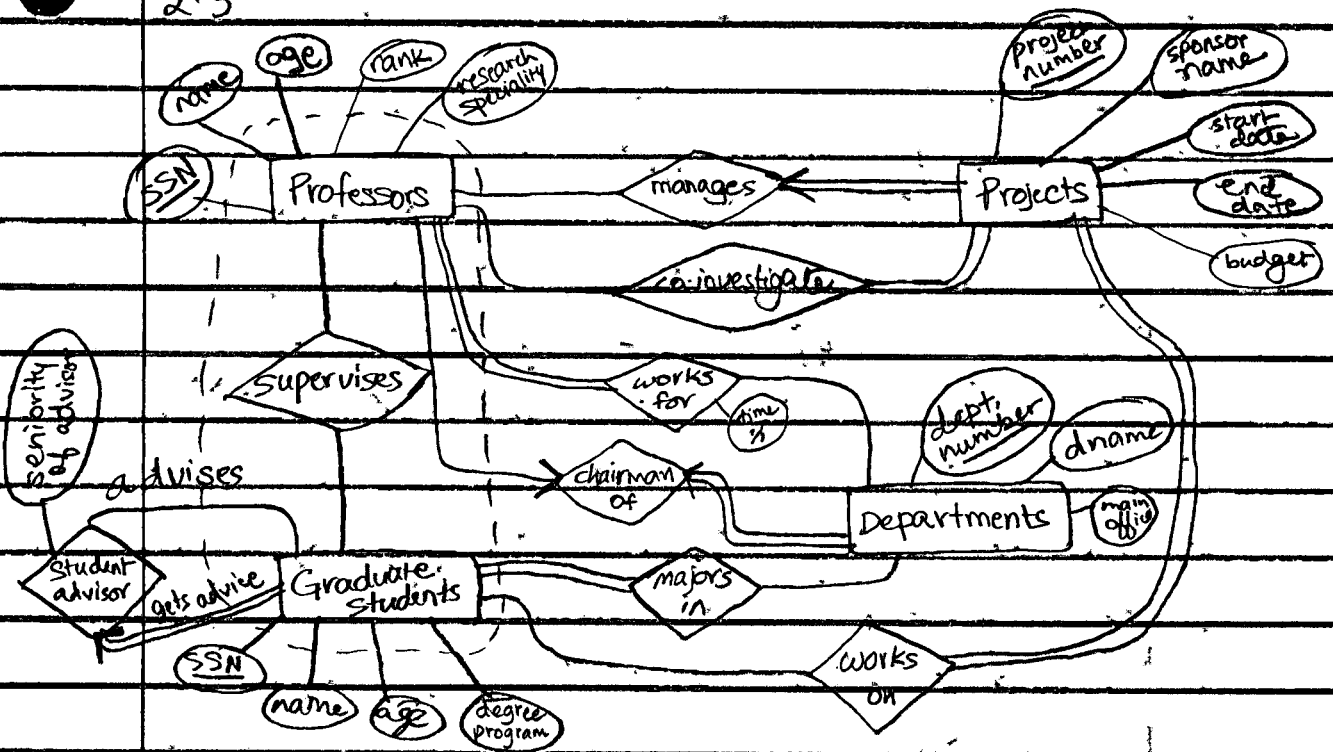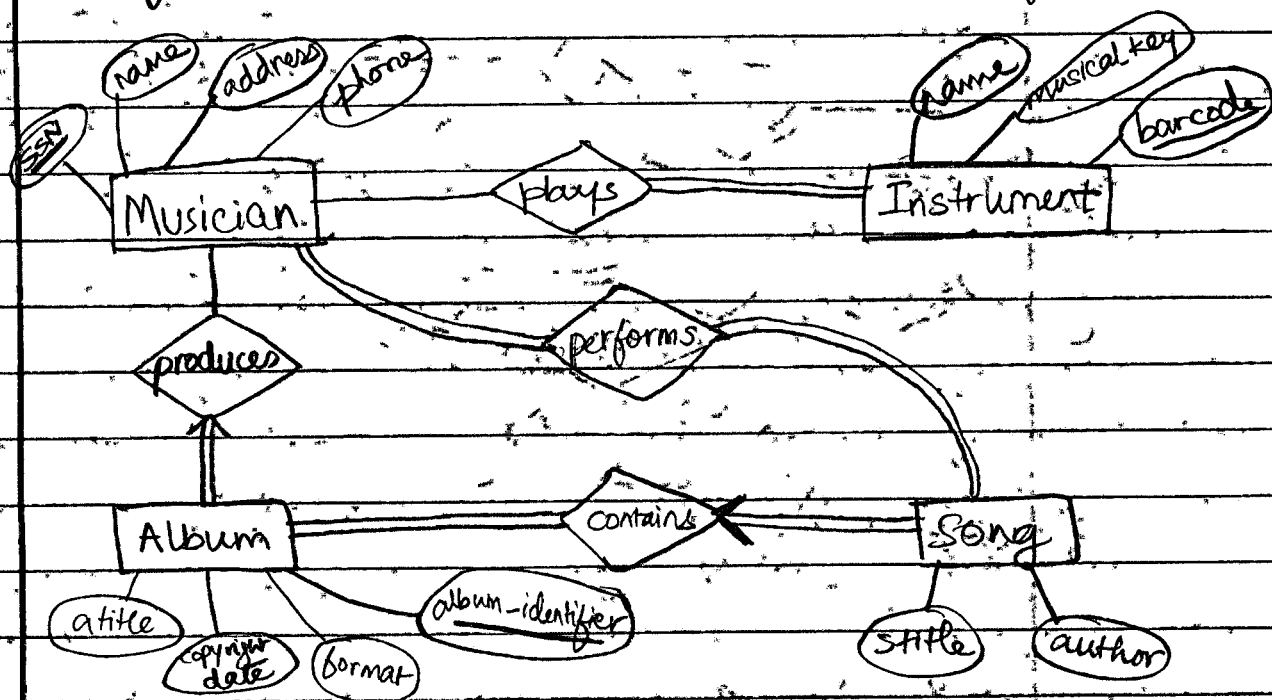2.3

Assumptions:

- Each department has exactly one chairman.
- A professor can only be chairman of one department.
- It is not compulsory for every grad. student to work on a project.
- A grad. student can have more than one major.
- Each grad. student has exactly one student advisor.
- A student advisor can advise multiple grad. students
- The student advisor relationship has an attribute called "seniority of advisor" to check that the student is eligible to advise other students based on seniority.
- Egregation of Professor supervises Grad. Student to show that if a grad. student works on a project, he/she

must have a professor supervising him/her.

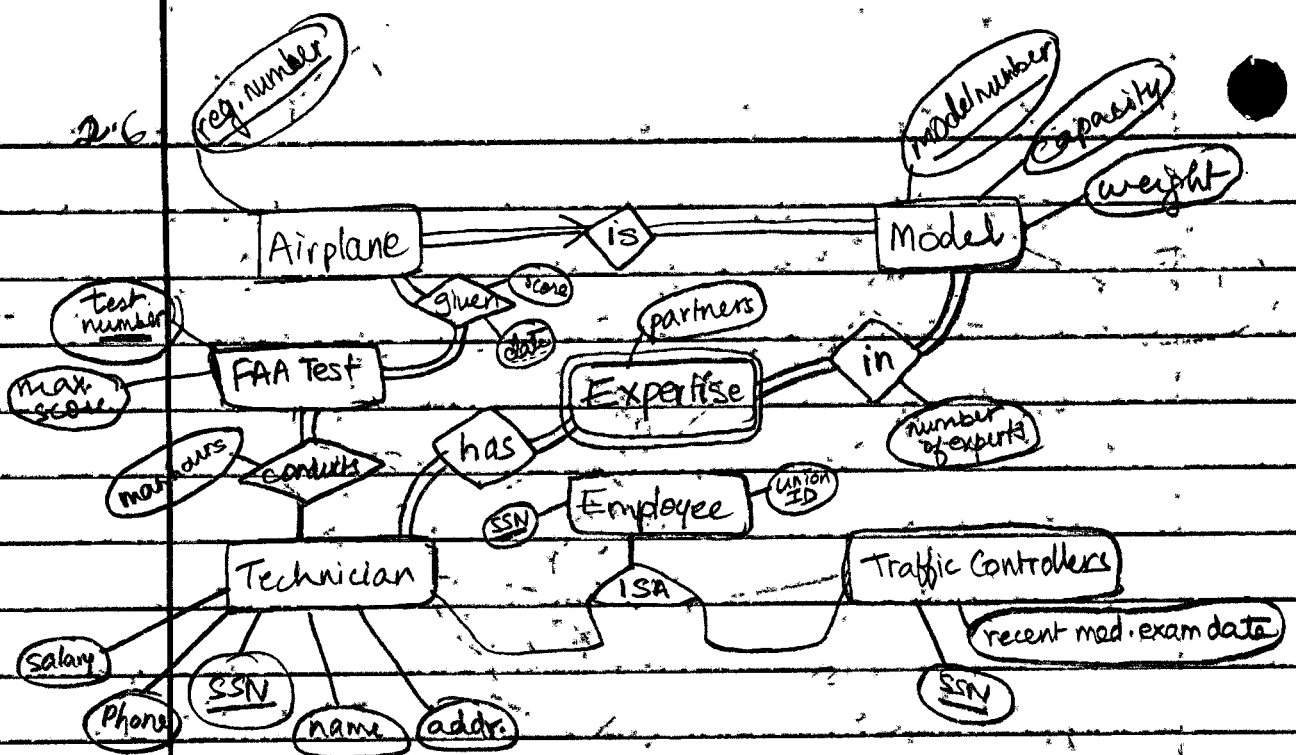|	
2.5	Conceptual Schema:

- Musician (ssn: string, name: string, address: string, phone: string, primary key (ssn))
- Instrument (barcode: string, name: string, musical key: string, primary key (barcode))
- Album (atitle: string, copyright_date: datetime, format: string, album-identifier: string, primary key (album-identifier))
- Song (stitle: string, author: string, primary key (stitle, author))
- Plays (ssn: string, barcode: string, primary key (ssn, barcode), foreign key (ssn) references Musician foreign key (barcode) references Instrument)
- Contains (album-identifier: string, stitle: string, author: string, primary key (stitle, author), foreign key (album-identifier) references Album, foreign key (stitle) references Song, foreign key (author) references Song)
- Performs (ssn: string, stitle: string, author: string, primary key (ssn, stitle, author), foreign key (ssn) references Musician, foreign key (stitle, author) references Song)
- Produces (album-identifier: string, ssn: string, primary key (album-identifier), foreign key (album

identifier) references Album, foreign key (ssn) references Musician)

Assumptions:
- It is not compulsory for every musician to play an instrument.
- Each musician has only one phone number.
- Each musician must perform at least one song.
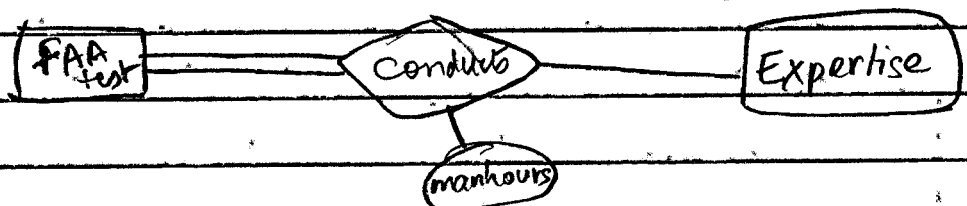- It is not compulsory for every musician to produce an album.

2.6

An ER diagram with entities: Airplane (attribute: reg. number) connected via "is" relationship to Model (attributes: model number, capacity, weight). FAA Test (attributes: test number, max. score) with "given" relationship (attributes: score, date). Expertise (attribute: partners) connected via "in" relationship to Model (attribute: number of experts). FAA Test connected via "conducts" relationship (attribute: manhours) to Technician. "has" relationship connects Expertise to Employee (attribute: union ID, SSN). Employee via ISA to Technician and Traffic Controllers. Technician (attributes: salary, phone, SSN, name, addr.). Traffic Controllers (attributes: recent med. exam date, SSN).

1. Overlapping constraints for partners who have the same expertise. (partners can be NULL if there is only one expert).

   total number of "partners" attribute depends on "number of experts" attribute

   Covering constraints: An employee can be either a traffic controller or a technician, but not both.

2. If the FAA wants test to be conducted by an expert, it would be the same diagram but we would attach the "conducts" relationship to the Expertise (instead of Technician)

A diagram showing FAA test connected via "Conducts" relationship (attribute: manhours) to Expertise.

2.9 UML - Unified Modeling Language is a
standard way to develop, specify, visualize
and document databases.

Use case diagrams - diagrams which specify
external functionality of the system. i.e. the actions
to respond to user requests & the people involved
in those actions.

Statechart diagrams - diagrams which describe
dynamic interactions between system objects, especially
in business modelling.

Class diagrams - they model application entities,
data entities, and their relationships.

database diagrams - show the structure of databases,
representation of classes in them, and constraints
and indexes.

Component diagrams - describe storage aspects
and application interfaces of the DB.

deployment diagrams - describe the hardware
aspects of the DBMS.

The Conceptual Database Modeling of the
UML is similar to ER design. Therefore,
UML constructs parallel ER constructs and
both can be drawn as diagrams.

3.1 relation schema — it specifies the relation's name, the name of each field / column / attribute, and the domain of each field.

relational database schema — the collection of schemas for the relations in the database

domain — It is the set of possible values associated with a field depending on its datatype

relation instance — a set of tuples in which each tuple has the same # of fields as the relation schema

relation cardinality — the number of tuples in a relation instance (i.e. # of rows)

relation degree — the number of fields in a relation instance (i.e. # of columns)

3.2 There are 22 distinct tuples in the relation instance. (Because cardinality = 22 and there are no duplicate tuples allowed)

3.4 Candidate key is the minimal set of fields that uniquely identify a tuple.

Primary key is a candidate key which is given by the DB designer.

Super key is a set of fields that contains a key.

3.5

1. Not a candidate key — age, gpa, name.
2. Candidate key — sid or login

3.6

foreign key constraint — a set of fields
that reference a primary key of another
table or itself.
It is important because if one relation is
modified, the other must be checked, and
perhaps modified, to keep the data consistent.
Referential Integrity — the foreign key of one
table must refer to a valid existing primary
key of another table (or its own PK), or it
can contain a NULL value (depending on the
design of the DB).

3.7

1. • for Enrolled: foreign key (sid) references Students
   foreign key (cid) references Courses
   • for Teaches: foreign key (fid) references Faculty,
   foreign key (cid) references Courses
   • for Meets-In: foreign key (cid) references Courses,
   foreign key (rno) references Rooms.

2. StudyGroup (sgnumber: int, numberOfstudents: int, grp_Leader: string,
   Primary key (sgnumber))