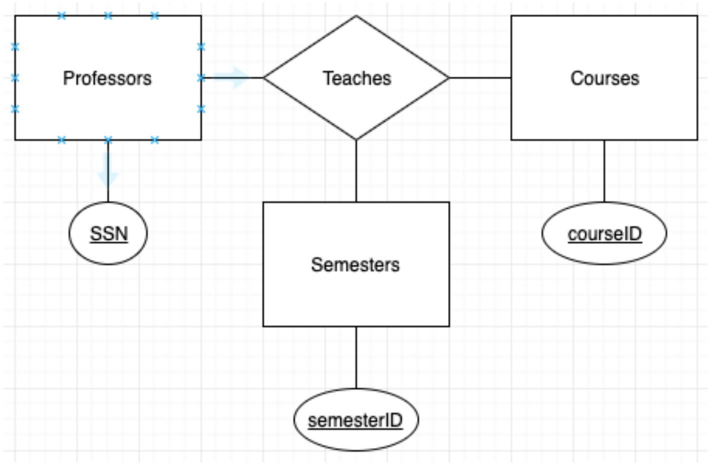
Yu-Hon Lin
yl1220
CS336
Homework 1

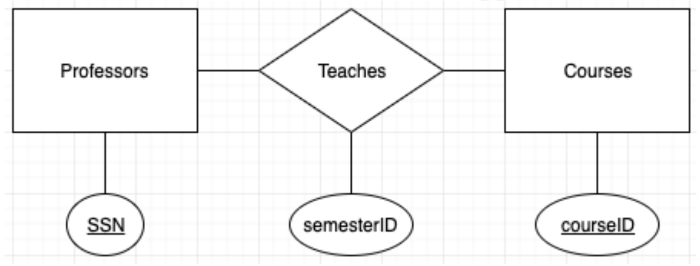
1) Exercise 2.2 from your textbook

Exercise 2.2 A university database contains information about professors (identified by social security number, or SSN) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teachers relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).

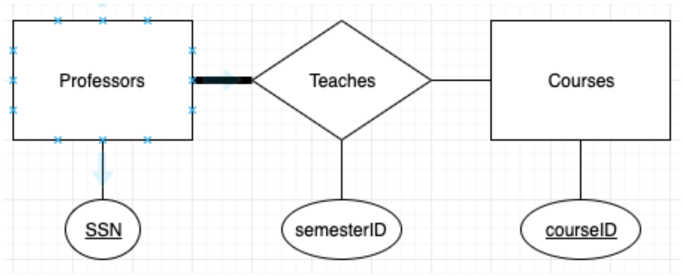
1. Professors can teach the same course in several semesters, and each offering must be recorded.



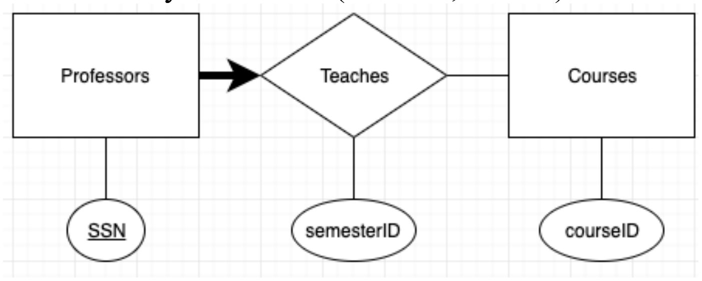
2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies in all subsequent questions.)



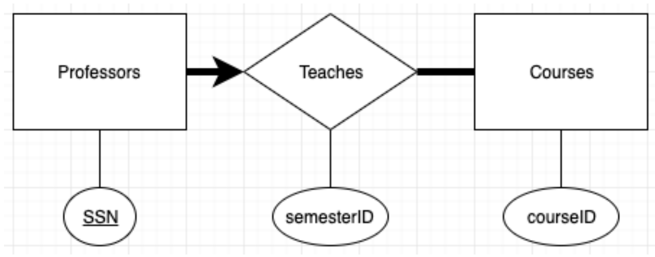
3. Every professor must teach some course.



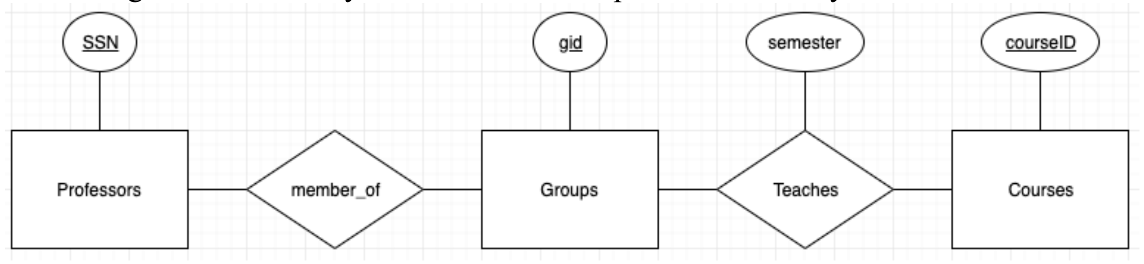
4. Every professor teaches exactly one course (no more, no less).



5. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.



6. Now suppose that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course. Model this situation, introducing additional entity sets and relationship sets if necessary.



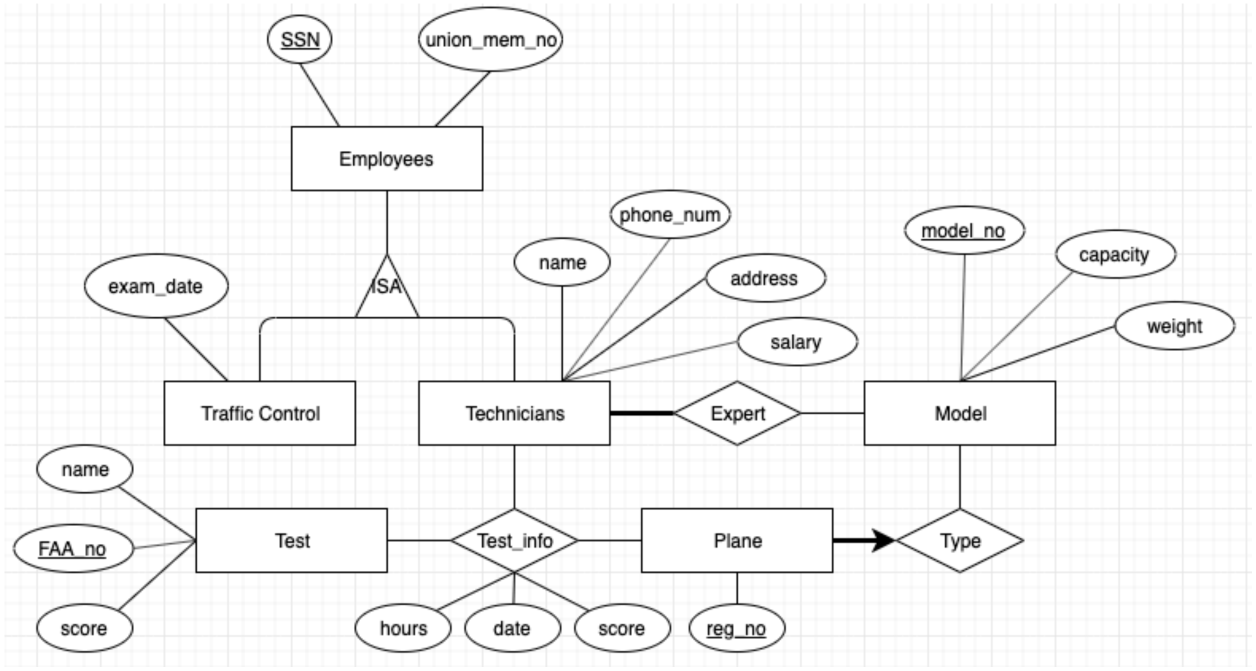
2) Exercise 2.6 from your textbook

Exercise 2.6 Computer Sciences Department frequent fliers have been complaining to Dane County Airport officials about the poor organization at the airport. As a result, the officials decided that all information related to the airport should be organized using a DBMS, and you have been hired to design the database. Your first task is to organize the information about all the airplanes stationed and maintained at the airport. The relevant information is as follows:

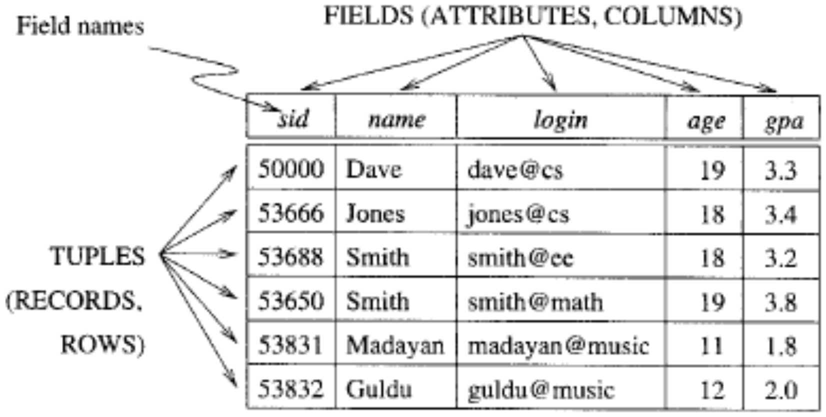
- Every airplane has a registration number, and each airplane is of a specific model.
- The airport accommodates a number of airplane models, and each model is identified by a model number (e.g.., DC_10) and has a capacity and a weight.
- A number of technicians work at the airport. You need to store the name, SSN, address, phone number, and salary of each technician.
- Each technician is an expert on one or more plane model(s), and his or her expertise may overlap with that of other technicians. This information about technicians must also be recorded.
- Traffic controllers must have an annual medical examination. For each traffic controller, you must store the date of the most recent exam.

- All airport employees (including technicians) belong to a union. You must store the union membership number of each employee. You can assume that each employee is uniquely identified by a social security number.
- The airport has a number of tests that are used periodically to ensure that airplanes are still airworthy. Each test has a Federal Aviation Administration (FAA) test number, a name, and a maximum possible score.
- The FAA requires the airport to keep track of each time a given airplane is tested by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the score the airplane received on the test.

1. Draw an ER diagram for the airport database. Be sure to indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Specify any necessary overlap and covering constraints as well (in English).
   Ans: Since all airline employees belong to a union, there is a covering constraint on the Employees ISA hierarchy.
2. The FAA passes a regulation that tests on a plane must be conducted by a technician who is an expert on that model. How would you express this constraint in the ER diagram? If you cannot express it, explain briefly.
   Ans: You cannot note the expert technician constraint required by the FAA in an ER diagram. There is no notation for equivalence in an ER diagram, so the Expert relation must be equivalent to the Type relation.



3) Exercise 3.5 from your textbook
Exercise 3.5 Consider the instance of the Students relation shown in Figure 3.1.


Field names
FIELDS (ATTRIBUTES, COLUMNS)
TUPLES (RECORDS, ROWS)

sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Figure 3.1 An Instance *S*1 of the Students Relation

1. Give an example of an attribute (or set of attributes) that you can deduce is *not* a candidate key, based on this instance being legal.
   Ans: Examples of non-candidate keys include {name}, {age}
2. Is there any example of an attribute (or set of attributes) that you can deduce *is* a candidate key, based on this instance being legal?
   Ans: You cannot determine a key of a relation given only one instance of the relation. The fact that the instance is "legal" is immaterial. The instance shown is just one possible "snapshot" of the relation. At other times, the same relation may have an instance (or snapshot) that contains a totally different set of tuples, and we cannot make predictions about those instances based only upon the instance that we are given.

4) Exercise 3.7 from your textbook
Exercise 3.7 Consider the relations Students, Faculty, Courses, Rooms, Enrolled, Teaches, and Meets_In defined in Section 1.5.2.

1. List all the foreign key constraints among these relations.
   Ans: There is no reason for a foreign key constraint on the Students, Faculty, Courses, or Rooms relations. These four relations must be free-standing, and data must be entered into these base relations cautiously. In the Enrolled relation, *sid* and *cid* should both have foreign key constraints placed on them. In the Teaches relation, *fid* and *cid* should both have foreign key constrains placed on them. In the Meets_In relation, *cid* and *rno* should both have foreign key constraints placed on them.
2. Give an example of a (plausible) constraint involving one or more of these relations that is not a primary key or foreign key constraint.
   The length of *sid*, *cid*, and *fid* should be standardized; checksums could be added to these identification numbers; limits could be placed on the size of the numbers entered into the credits, capacity, and salary fields; an enumerated type should be assigned to the grade field to limit the grade range from A to F; etc.

5) Provide the Relational Database Schema for the diagram of exercise 2.2

1. Professors can teach the same course in several semesters, and each offering must be recorded.

```
CREATE TABLE Professors (   ssn       char(10),
                            PRIMARY KEY (ssn) )
CREATE TABLE Courses (      courseID       INTEGER,
                            PRIMARY KEY (courseID) )
CREATE TABLE Semesters (    semesterID     char(10),
                            PRIMARY KEY (semesterID) )
CREATE TABLE Teaches (      ssn            CHAR(10),
                            coursed        INTEGER,
                            semesterID     char(10),
                            PRIMARY KEY (ssn, courseID, semesterID),
                            FOREIGN KEY (ssn) REFERENCES Professors,
                            FOREIGN KEY (coursed) REFERENCES Courses,
                            FOREIGN KEY (semester) REFERNCES Semester )
```

2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies in all subsequent questions.)

```
CREATE TABLE Professors (   ssn       char(10),
                            PRIMARY KEY (ssn) )
CREATE TABLE Courses (      courseID       INTEGER,
                            PRIMARY KEY (courseID) )
CREATE TABLE Teaches (      ssn            CHAR(10),
                            coursed        INTEGER,
                            semesterID     char(10),
                            PRIMARY KEY (ssn, courseID),
                            FOREIGN KEY (ssn) REFERENCES Professors,
                            FOREIGN KEY (coursed) REFERENCES Courses )
```

3. Every professor must teach some course.
The tables created for the previous part to this question are the best we can do without using check constraints or assertions in SQL. The participation constraint cannot be captured using only primary and foreign key constraints because we cannot ensure that every entry in Professors has an entry in Teaches.

4. Every professor teaches exactly one course (no more, no less).

```
CREATE TABLE Professor_teaches (   ssn            CHAR(10),
                                   coursed        INTEGER,
                                   semesterID     char(10),
                                   PRIMARY KEY (ssn),
                                   FOREIGN KEY (coursed) REFERENCES Courses )
CREATE TABLE Courses (      courseID       INTEGER,
                            PRIMARY KEY (courseID) )
```

Observe that the table for Professors can be omitted.

5. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.

```
CREATE TABLE Professor_teaches (  ssn          CHAR(10),
                                  coursed      INTEGER,
                                  semesterID   char(10),
                                  PRIMARY KEY (ssn),
                                  FOREIGN KEY (coursed) REFERENCES Courses )
```

Observe that the table for Professors and Courses and both be omitted.

6. Now suppose that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course. Model this situation, introducing additional entity sets and relationship sets if necessary.

```
CREATE TABLE Professors (   ssn    CHAR(10),
                            PRIMARY KEY (ssn) )

CREATE TABLE Groups (       gid    INTEGER,
                            PRIMARY KEY (gid) )

CREATE TABLE Courses (      courseID    INTEGER,
                            PRIMARY KEY (courseID) )

CREATE TABLE MemberOf (     ssn    CHAR(10),
                            gid    INTEGER,
                            PRIMARY KEY (ssn, gid),
                            FOREIGN KEY (ssn) REFERENCES Professors,
                            FOREIGN KEY (gid) REFERENCES Groups )

CREATE TABLE Teaches (      gid           INTEGER,
                            courseID      INTEGER,
                            semesterID    char(10),
                            PRIMARY KEY (gid, courseID),
                            FOREIGN KEY (gid) REFERENCES Groups,
                            FOREIGN KEY (coursed) REFERENCES Courses )
```

6) Provide the Relational Database Schema for the diagram of exercise 2.6

```
CREATE TABLE Employees (  ssn               CHAR(11),
                          union_mem_no      INTEGER,
                          PRIMARY KEY (ssn) )

CREATE TABLE Technicians ( ssn          CHAR(11),
                          name          CHAR(20),
                          address       CHAR(20),
                          phone_no      CHAR(14),
                          PRIMARY KEY (ssn),
                          FOREIGN KEY (ssn) REFERENCES Employees ON DELETE CASCADE )

CREATE TABLE Traffic_Control (   ssn           CHAR(11),
                                 exam_date     DATE,
                                 PRIMARY KEY (ssn),
                                 FOREIGN KEY (ssn)
                                      REFERENCES Employees ON DELETE CASCADE )

CREATE TABLE Models (     model_no      INTEGER,
                          capacity      INTEGER,
                          weight        INTEGER,
                          PRIMARY KEY (model_no) )

CREATE TABLE Experts (    ssn           CHAR(11),
                          model_no      INTEGER,
                          PRIMARY KEY (ssn, model_no),
                          FOREIGN KEY (ssn) REFERENCES Technicians,
                          FOREIGN KEY (model_no) REFERENCES Models )
```

(The participation constraint cannot be captured in the Expert table.)

```
CREATE TABLE Plane_Type ( reg_no        INTEGER,
                          model_no      INTEGER,
                          PRIMARY KEY (reg_no),
                          FOREIGN KEY (model_no) REFERENCES Models )

CREATE TABLE Tests ( FAA_no        INTEGER,
                     name          CHAR(20),
                     score         INTEGER,
                     PRIMARY KEY (FAA_no) )
```

```
CREATE TABLE Test_Info (    FFA_no          INTEGER,
                            ssn             CHAR(11),
                            reg_no          INTEGER,
                            hours           INTEGER,
                            date            DATE,
                            score           INTEGER,
                            PRIMARY KEY (ssn, reg_no, FFA_no),
                            FOREIGN KEY (reg_no) REFERENCES Plane_Type,
                            FOREIGN KEY (FAA_no) REFERENCES Test,
                            FOREIGN KEY (ssn) REFERENCES Technicians )
                            CONSTRAINT      MODEL
                            CHECK (         SELECT * FROM Experts, Type
                                            WHERE Experts.ssn = ssn AND
                                            Experts.model_no = Type.model_no AND
                                            Type.reg_no = reg_no )
```