Name: _____Yujie Zhang_____ **Section:** _____ **Mailbox:** _206|_

Purpose: To gain experience with Creating Tables

What to do: Please solve the following problems.

Below is the Research University domain descriptions from HW1, which may be used in the problems below. These are solely for reference.

Research University Scenario:
A research university is looking to create a new database to better track their current research projects and those who are working on them. A professor, with a social security number (SSN), name, age, rank, and research specialty, can run multiple projects. A project has a number, a sponsor name (for example, NSF), a start date, an end date, and a budget.

Research universities also use graduate students in their research projects. Graduates students have an SSN, a name, an age, and a degree program (such as M.S. or Ph. D.). One or more graduate students, work on a project. A graduate student may work on multiple research projects. When a graduate student is working on a research project they must be supervised by a professor for that project. As graduate students could work on multiple projects, they can have different supervisors for each project they work on. Graduate students also have another, more senior, graduate student who advises them on what courses to take. This person is a student advisor. Graduate students in their final year may not have a student advisor assigned to them.

The research university is split into several departments. A department has a number, a name, and a main office (represented by the office number). Each department has a chairman, who is a professor who runs the department. A professor may work in one or more departments, and for each department they work for there is a time percentage that dictates how much time is spent in each department. Graduate students belong to a single department in which they are working on their degree.

Below is the relation schema for the Research University problem:

Person(<u>SSN</u>, Age, Name)

Professor(<u>SSN</u>, Rank, Specialty)

GraduateStudent(<u>SSN</u>, DegreeProgram, AdvisorSSN, DepartmentNumber)

Project(<u>Number</u>, StartDate, EndDate, Budget, SponsorOrganizationName)

Department(<u>Number</u>, Name, Abbreviation, MainOffice, ChairmanSSN)

WorksFor(<u>ProfessorSSN</u>, <u>DepartmentNumber</u>, TimePercentage)

WorksOn(<u>GraduateStudentSSN</u>, <u>ProjectNumber</u>, SupervisingProfessorSSN)

You must support the following foreign keys (→ means "references"):

Professor(SSN) → Person(SSN)
GraduateStudent(SSN) → Person(SSN)
GraduateStudent(AdvisorSSN) → GraduateStudent(SSN)
GraduateStudent(DepartmentNumber) → Department(Number)
WorksFor(ProfessorSSN) → Professor(SSN)
WorksFor(DepartmentNumber) → Department(Number)
WorksOn(GraduateStudentSSN) → GraduateStudent(SSN)
WorksOn(ProjectNumber) → Project(Number)
WorksOn(SupervisingProfessorSSN) → Professor(SSN)
Department(ChairmanSSN) → Professor(SSN)

1) Write SQL statements to create tables for each of the relations above (there are sections for each one below). Be sure to include appropriate primary key and foreign key constraints, as well as any domain constraints you feel are necessary. You may attach additional paper if necessary; if you do so, please indicate where your work can be found.

a) Person

```
CREATE TABLE Person (
        SSN   CHAR (11)
        Age   int,
        Name  VARCHAR (20),
        Primary Key (SSN)
        CHECK (SSN LIKE '[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]')
);
```

← suppose SSN is 11 bit string, in the form of 'AAA-BB-CCCC'

← suppose name max is 20 string with length 20.

b) Professor

```
CREATE TABLE Professor (
        SSN        CHAR (11),
        Rank       VARCHAR (20),
        Specialty  VARCHAR (20),
        Primary Key (SSN),
        Foreign Key (SSN) references Person (SSN)
);
```

← suppose Rank is varchar

c) GraduateStudent

```sql
CREATE TABEL GraduateStudent (
        SSN  CHAR (11)  PRIMARY KEY,
        DegreeProgram  VARCHAR (20),
        Advisor SSN  CHAR(11),
        DepartmentNumber  int,
        Foreign Key (SSN) references Person (SSN),
        Foreign Key (AdvisorSSN) references GraduateStudent (SSN),
        Foreign Key (DepartmentNumber) references Department (Number)
);
```

Suppose DepartmentNumber is an integer

d) Project

```sql
CREATE TABEL Project (
        Number  VARCHAR (20)  PRIMARY KEY,
        StartDate  DATE,
        EndDate  DATE,
        Budget  money,
        SponsorOrganizationName  VARCHAR (20)
);
```

suppose project number is a string, varchar (20)

suppose budget is a money type.

e) Department

```
CREATE TABLE Department (
        Number       VARCHAR (20),
        Name         VARCHAR (50),
        Abbreviation VARCHAR (10),
        Main Office  VARCHAR (20),
        ChairmanSSN  CHAR (11),
        Primary Key (Number)
        Foreign Key (ChairmanSSN) references Professor (SSN)
);
```

suppose number is a string

f) WorksFor

```
CREATE TABLE WorksFor (
        ProfessorSSN      CHAR (11),
        DepartmentNumber  VARCHAR (20),
        TimePercentage    float,
        CHECK (TimePercentage <= 100.00),
        Primary Key (ProfessorSSN, DepartmentNumber),
        Foreign Key (ProfessorSSN) references Professor (SSN),
        Foreign Key (DepartmentNumber) references Department (Number)
);
```

suppose timepercentage is a float,
Ex: 55.0 means 55.0 %
    53.8 means 53.8 %

g) WorksOn

```sql
CREATE TABLE WorksOn (
        GraduateStudentSSN     CHAR(11),
        ProjectNumber          VARCHAR(20),
        SupervisingProfessorSSN    CHAR(11),
        Primary Key ( GraduateStudentSSN, ProjectNumber ),
        Foreign Key ( GraduateStudent SSN) references GraduateStudent (SSN)
        Foreign Key ( ProjectNumber ) references Project (Number)
        Foreign Key ( SupervisingProfessor SSN) references Professor (SSN)
);
```