**Name: Zijie Yu**

**Database Management Systems (CSC434)**

**Final Project**

**1) - Define the information content of your database.**

**a)-Define a set of entities and appropriate attributes for each entity. Minimum 10 entities.**

Note: The first attributes are primary key, the Blue color entity means entity created automaticly by Visual Paradigm because of many to many relationship (intermediate table).

* **Students:** Student\_ID; First\_Name; Last\_Name; Country; Age; Phone; Email; Expected\_graduation\_year; Address;
* **Major**: Major\_Name; Department; Minimum\_GPA; DepartmentId
* **Students\_Major:** Students\_StudentId; Major\_MajorId
* **Course:** CourseId; Course\_Name; Seats Avaliable; Professor\_ProfessorId; SubjectId; ClassSceduleId;
* **Student\_Course:** Students\_StudentId; Course\_CourseId
* **Classroom:** ClassroomId; Classroom\_Name; Buildings; Location; Capacity;
* **Classroom\_Course:** Classroom\_ClassroomId; Course\_CourseId
* **Course\_Scedule**: CourseSceduleId; Meeting\_Day; Time\_Start; Time\_End
* **Subject:** SubjectId; Subject\_Name; DepartmentId
* Professors: ProfessorId; First\_Name; Last\_Name; Email; Phone; Address
* **Subject\_Professors:** Subject\_SubjectId; Professors\_ProfessorId
* **Titles:** TitleId; Type
* **Titles:** Titles\_Titled; Professors\_ProfessorId
* **Manager:** ManagerId; First\_Name; Last\_Name; Phone; Address
* **Advisor:** AdvisorId; First\_Name; Last\_Name; Phone; Address
* **Department:** DepartmentId; Name; AdvisorId; ManagerId

**b)-Define a set of relationships that might exist between/among entities and attributes. Such relationships may include one-to-one, one-to-many and many-to-many associations.**

**Students – Course:** Many to many

(One student can take many course, and one course can have many students)

**Students – Major:** Many to many

(One student can have one or more majore, and one major can have many students.)

**Major – Department:** Many to one

( One major must belong to one department, but one department can have many major)

**Department – Manager:** One to one

(One department can only have one manager, and one manager can only manage one specific department)

**Department – Advisor:** One to one

(One department can only have one advisor, and one advisor can only manage one specific department)

**Department – Subject:** One to many

(One deparment can have many subjects, but one subject can only belong to one department.)

**Subject – Course:** One to many

(One subject can have many different course, but one course must have one subject)

**Course – Course\_Schedule:** Many to one

(One course\_Schedule can have many different course, but one course can only have on specific schedule.)

**Course – Classroom:** Many to many

(One course may have one or more classroom, and one classroom can have many different course.)

**Professor – titles:** Many to many

(One professor can have one or more titles, and one titles can have many different professor.)

**Professor – subject:** Many to many

(One professor can teach many different subject course, and one subject course can have many different professor.)

**c)-Define a set of constraints that may be imposed on data.**

* One department can only have 1 manager.
* One department can only have 1 advisor.
* Student must take at least one course
* Every Professor must have proper title, sush as Ph.d, Acaemic or assistanmt
* One major can only belong to one department.
* One course can only belong to one subject, such as CSC or MATH.
* A course must have at least one classroom.

**2) - Define an E-R Diagram for your database design.**

**3) - Define a relational schema for your database design.**

**Make sure that you have both one-to-many and many-to-many associations.**

**a)-Define one or more realistic key(s) for every relation scheme. Use both simple and composite keys.**

* **Student:** The primary key should be SutdentId, since StudentId is unque.
* **Students\_Course:** Both Students\_StudentId and Course\_CourseId are foreign keys.
* **Course:** CourseId should be primary key, because it is unique. And Professors\_professorId, SubjectId, CourseScheduleId are foregin keys.
* **Students\_Major:** Both Students\_StudentId and Major\_MajorName are foreign keys.
* **Major:** MajorName is primary key because it is unique. The DepartmentId is foregn key.
* **Department:** The primary key should be DepartmentId, the foregn keys are AdvisorId and ManageId.
* **Manager:** ManagerId should be primary key since it is unique.
* **Advisor:** AdvisorId should be primary key.
* **Course\_Schedule:** The primary key should be CourseScheduledId since it is unique.
* **Subject:** SubjectId should primary key and the DepartmentId should be foreign key.
* **Ttitles:** The titleId should be primary key.
* **Titles\_Professor:** Both Titles\_TtitleId and Professor\_ProfessorId are foreign keys.
* **Professors:** ProfessorId should be primary key since it is unique.
* **Subject\_Professors:** Both Subject\_SubjectId and Professors\_ProfessorId are foreign keys.
* **Classroom:** The primary key should be classroomId since it is unique.
* **Classroom\_Course:** Both Classroom\_ClassroomId and Course\_CourseId are foreign keys.

**b)-Define a realistic set of Functional / Multi-Valued Dependencies (when appropriate) for every relation scheme.**

**C-Check whether your relational schema is in 2NF, 3NF, BCNF, 4NF.**

**d)-Put your relational schema in the highest normal form that is possible.**

**Note that, every relation scheme should be in a specific normal form in order to have the relational schema in that normal form.**

**NOTE: Please provide a detailed explanation for every question when appropriate.**