Solution to CS532 Homework 2

1. [70%] Transform the instructor's ER diagram for the Student Registration System (see the Homework 1 folder on blackboard) to relations using the algorithm discussed in class. For composite attributes, use Method 1 (i.e., use the more specific attributes only) to perform the transformation. For each relation obtained, underscore the key, specify other candidate keys (if any) and foreign keys (if any), and specify the constraints associated with this relation (including all constraints that are described in the Requirements Document).

Answer: The ER Diagram is transformed into the following relations. The constraints are given below each relation.

Students(sid, firstname, lastname, status, gpa, email)
Other candidate key: email
Constraints: valid values for status: {freshman, sophomore, junior, senior, graduate}
0 <= gpa <= 4.0

Courses(dept_code, course#, title, credits, deptname)
Constraints: 100 <= course# <= 799
valid values for credits: {3, 4}
if course# < 500, then credits is 3 and if course# >= 500, then credits is 4
deptname is a foreign key referencing deptname of Departments

Prerequisites(dept_code, course#, pre_dept_code, pre_course#)
Both (dept_code, course#) and (pre_dept_code, pre_course#) are foreign keys referencing (dept_code, course#) of Courses.
Courses and prerequisite courses do not form a cycle.

Classes(dept_code, course#, sect#, year, semester, start_time, end_time, limit, size, classroom, capacity, fid)
Constraints: starttime < endtime
size <= limit <= capacity
No faculty can teach classes with overlapping class times
(dept_code, course#) is foreign keys referencing (dept_code, course#) of Courses
fid is a foreign key referencing the fid of Faculty

Classes_days(dept_code, course#, sect#, year, semester, day)
valid values for day: {Monday, Tuesday, Wednesday, Thursday, Friday}.
(dept_code, course#, sect#, year, semester) is a foreign key referencing (dept_code, course#, sect#, year semester) of Classes

Faculty(fid, firstname, lastname, rank, office, email, deptname)
Other candidate keys: office, email
Constraints: valid values for rank: {lecturer, assistant professor, associate professor, professor}

belong_to(<u>fid, deptname</u>)
There's no other candidate key.
fid is a foreign key referencing fid in Faculty and deptname is a foreign key referencing deptname of Departments.

Departments(<u>deptname</u>, office, phone#)
Other candidate key: office

Enrollment(<u>sid, dept_code, course#, sect#, year, semester</u>, lgrade, ngrade)
Constraints: valid values for lgrade: {A, B, C, D, F, I, null}
valid values for ngrade: {0, 1, 2, 3, 4, null}
lgrade and ngrade must match: A ←→ 4, B ←→ 3, C ←→ 2, D ←→ 1, F ←→ 0, I ←→ null, null ←→ null
No student can take the same course more than once
A student must have completed all prerequisite courses with a grade of at least C in order to enroll into a class
Students cannot enroll into classes with overlapping class times
sid is a foreign key referencing sid of Students
(dept_code, course#, sect#, year, semester) is a foreign key referencing (dept_code, course#, sect#, year, semester) of Classes
Each student can only enroll into between 1 and 5 classes
Each class must have at least 5 students

Student_majors(<u>sid, deptname</u>)
sid is a foreign key referencing sid of Students
deptname is a foreign key referencing deptname of Departments
Each student must have either 1 or 2 majors.

What types of constraints can be represented in the ER diagram but cannot be represented in the relational data model? Also provide an example.

Answer: Cardinality/connectivity related constraints cannot be represented in the relational model. For example, the constraint "Each student can only enroll into between 1 and 5 classes." cannot be represented in the relational model.

2. [20%] Let A and B be the only attributes of a relation R.
   1) Suppose neither A nor B is a key of R. Does the combination of these two attributes, (A, B), form a key of R? Why or why not?
   2) Suppose the combination of these two attributes, (A, B), is a key of R. Can either A or B be a superkey of R? Why or why not?

Answer:
1) Yes. First, (A, B) as the only attributes of R form a superkey of R (implied by the Unique Row Rule). Second, since neither A nor B is a key of R, this means both attributes A and B are necessary to make (A, B) a superkey of R. Thus (A, B) satisfies the minimality requirement and is a (candidate) key of R.

2) No, neither A nor B can be a superkey of R. If A (or B) were a superkey, then it would also be a key because it is already minimal; if A or B were a key, then (A, B) could not be a key because it does not satisfy the minimality requirement.

3. [10%] Method 1 for transforming IS_A hierarchy discussed in class (see Lecture Notes) can preserve the IS_A relationships among the entities in the entity sets in an IS_A hierarchy. What are the advantages of Method 1 in transforming ER diagrams to relations comparing to Method 2?

Answer: The main advantage and impact of preserving the IS_A relationship is that the difficult attributes (multi-valued attributes and composite attributes) and relationships (many-to-many and ternary relationships) of the super entity set need not be passed on to the sub entity sets when transforming the ER diagram to relations. This can significantly simplify the transformation of the sub entity sets and reduce the number of resultant relations.