CS532 : Homework 2 - ER to Relation Transform

Name - Shahu S Ronghe
B-Number - B00908937

---

1. **Transform the instructor's ER diagram for the Student Registration System (see the Homework 1 folder in Brightspace) to relations using the approach discussed in Chapter 4 of the Lecture Notes. When transforming the IS_A hierarchy, use Method 1 (i.e., sub entity set only explicitly inherits the key from the super entity set). For composite attributes, use Method 1 (i.e., use the more specific attributes only) to perform the transformation and rename the component attributes to avoid ambiguity as needed. For each relation obtained, (i) underscore the primary key, (ii) specify (using words) other candidate keys (if any), (iii) specify (using words) all foreign keys (if any), and (iv) list all other constraints (except the primary key, candidate key(s) and foreign key(s)) that are described in the Requirements Document). Finally, list those constraints that involve attributes from multiple relations (except foreign keys).**

   Answer -

   Please find below Relational schema for the provided E-R diagram -

   - **Students**
     - Relational – B#, firstname, lastname, gpa, level, email, bdate
     - Candidate Keys – email
     - Constraints – level *{freshman, sophomore, junior, senior, graduate)*, gpa *{0 to 4}*

   - **Courses**
     - Relational – dept_code, course#**,** title, credits
     - Foreign Keys – dept_name *[Departments.deptname]*
     - Constraints - course# *{100 to 799}*, credits *{3,4}*

   - **Prerequisites**
     - Relational - dept_code, course#, c_dept_code, c_course#
     - Foreign Keys - c_dept_code *[Courses.dept_code]*, c_course# *[Courses.course#]*
     - Constraints - no cycle in prerequisites and courses

   - **Departments**
     - Relational - deptname, office, phone#
     - Candidate Keys - office

   - **Faculty**
     - Relational - B#, firstname, lastname, title, office, email, phone#
     - Candidate Keys - office, email
     - Foreign Keys - deptname *[Departments.deptname]*
     - Constraints - title *{adjunct, lecturer, assistant professor, associate professor, professor}*

- **Classes**
  - Relational - dept_code, course#, sec#, year, semester, days, start_time, end_time, limit, size, room, B#
  - Foreign Keys - dept_code *[Courses.dept_code]*, course# *[Courses.course#]*, B# *[Faculty.B#]*
  - Constraints - semester *{Spring, Fall, Winter, Summer 1, Summer 2}*

- **Enrollment**
  - Relational - B#, dept_code, course#, sec#, year, semester, lgrade, score
  - Foreign Keys - B# *[Students.B#]*, dept_code *[Classes.dept_code]*, course# *[Courses.course#]*, sec# *[Courses.sec#]*, year *[Courses.year]*, semester *[Courses.semester]*
  - Constraints - lgrade *{A, A-, B+, B, B-, C+, C, C-, D, F, I}*, score *{0,100}*

- **Major**
  - Relational - B#, deptname
  - Foreign Keys - B# *[Students.B#]*, deptname *[Department.deptname]*

2. **Give three examples where constraints in the Requirements Document that can be explicitly represented in the ER diagram cannot be explicitly represented in the relational data model.**

   Answer: We cannot show weak entity, composite attributes and cardinality in relational data model.

   a. Courses has up to 3 prerequisites courses and has unary relation with it. But showing that is not possible in relational data model
   b. Courses have 0 or more section in semester
   c. Undergraduate and graduate students capacity of 10 and 6 students respectively.

3. **Consider a relation R with 3 attributes A, B and C. What is the maximum number of possible keys R could have? What they may look like (list all possibilities)? Justify your answer.**

   Answer: The total combination of keys are (A,B), (B,C), (A,C), (A,B,C). This is the only combination of keys. So there can be a maximum of 3 keys.

4. **Let A be the primary key of relation R1 and B be the primary key of R2. If A is declared as a foreign key referencing B and B is also declared a foreign key referencing A. Use the Entity Integrity Rule and the Referential Integrity Rule to explain that with foreign keys declared in this way, no tuple can be inserted into either R1 or R2. (You may assume that R1 and R2 are initially empty.)**

   Answer: Using the Entity Integrity Rule and Referential Integrity Rule, there will be violation of rules and thus, we cannot add tuple into R1 or R2.