1. Create an ER diagram with a weak entity set for the following description.

A department has a single attribute—its unique name. Departments offer courses, but each course is offered by only one department. A course has only one attribute, its number (e.g. 101, 142). Several departments can offer courses with the same number (e.g. both Math and Physics can offer a course numbered 101).

2. Using the standard mapping for weak entity sets, map your ER diagram in Exercise 1 to relational schemas. Express your result as SQL create-table statements. Include in your create-table statements primary keys and foreign-key references.

3. Create an ER diagram with ISA hierarchies for the following description.

Students have a name and address and a unique ID number. All students are either undergraduate students or graduate students, but not both. Every graduate student is either an MS student or a PhD student or both an MS student and a PhD student. Some undergraduate students are work-study students and some are scholarship students (some are neither work-study students nor scholarship students). A university policy prevents work-study students from being scholarship students and prevents scholarship students from being work-study students. Work-study students have an hourly rate. Scholarship students have either full or partial scholarships. A faculty member belongs to a single department and has a name, which is also a unique identifier for the faculty member. Some faculty members are currently teaching students, some are currently advising grad students, some are doing both, and some are doing neither. Each grad student has one and only one advisor.

Your ER diagram should have the following entity sets: Student, UndergradStudent, GradStudent, WorkStudyStudent, ScholarshipStudent, PartialScholarshipStudent, FullScholarshipStudent, MSSStudent, PhDStudent, FacultyMember, Instructor, and Advisor. It should also have two relationship sets: one for teaching between Student and Instructor and one for advising between GradStudent and Advisor. And it should have the following attributes: ID, StudentName, Address, Department, FacultyMemberName, and HourlyRate.

4. Map your ER diagram in Exercise 3 to relational schemas. Although there are a number of reasonable mappings, choose to generate four schemas: one for undergraduate students, one for graduate students, one for faculty members, and one for teaching. Append a question mark to nullable attributes. Add clarifying comments for attributes that should be Boolean or have categorical data values.

5. Create an ER diagram with complex attributes for the following description.

A ticket is purchased for a concert. A ticket has basic date-time-place information (date, time, and seat number) that identifies a ticket; a ticket also has a price, and sales commission. A concert has a group name, which identifies the concert, and it also has a list of one or more stars who perform in the group. Make Ticket and Concert be entity sets. Make DateTimePlace be a compound attribute consisting of Date, Time, and SeatNumber. Price is a regular attribute for a ticket, but the SalesCommission attribute is computed (as 1% of the ticket price). Make Star be a multivalued attribute of Concert.

6. Map your ER diagram in Exercise 5 to relational schemas. Assume that the sales commission need not be stored, but make a note saying how it is computed. Be on the lookout for redundant schemas: after mapping your ER diagram to relational schemas (to show that
you know the mapping rules), draw a line through any schema that is redundant, and make a note saying why the schema is redundant. For any schema that could be redundant, but isn’t, make a note explaining why.

7. Reverse-engineer the following database schema into an ER model instance. Your resulting ER model instance must be such that if you apply a standard algorithm to map it to relational schemas, you would arrive at this database schema.

    Book(ISBN, Title, HardOrSoftCover, PublisherName)
    BookAuthors(ISBN, Author)
    Publisher(PublisherName, City, State)
    University(UniversityName, City, State)
    HasAdoptedAsOfDate(UniversityName, ISBN, Date)

HardOrSoftCover is a categorical attribute that has only two values: “Hard Cover Book” and “Soft Cover Book”. To model this, use generalization/specialization (i.e., an is-a relationship) and add an appropriate constraint to the generalization/specialization triangle.