HOMEWORK 7	Normalization for Relational Databases
Due Wed, Nov 18 at 11:30 pm	**Objectives:** **Basics of Functional Dependencies and Normalization**

14.19. Suppose that we have the following requirements for a university database that is used to keep track of students' transcripts:

a. The university keeps track of each student's name (Sname), student number (Snum), Social Security number (Ssn), current address (Sc_addr) and birth date (Bdate), sex (Sex), class (Class) ('freshman', 'sophomore', ... ,'graduate'), major department (Major_code), minor department (Minor_code) (if any), and degree program (Prog) ('b.a.', 'b.s.', ... , 'ph.d.'). Both Ssn and student number have unique values for each student.

b. Each department is described by a name (Dname), department code (Dcode), office number (Doffice), office phone (Dphone), and college (Dcollege). Both name and code have unique values for each department.

c. Each course has a course name (Cname), description (Cdesc), course number (Cnum), number of semester hours (Credit), level (Level), and offering department (Cdept). The course number is unique for each course.

d. Each section has an instructor (Iname), semester (Semester), year (Year), course (Sec_course), and section number (Sec_num). The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the total number of sections taught during each semester.

e. A grade record refers to a student (Ssn), a particular section, and a grade (Grade). Design a relational database schema for this database application. First show all the functional dependencies that should hold among the attributes. Then design relation schemas for the database that are each in 3NF or BCNF. Specify the key attributes of each relation. Note any unspecified requirements, and make appropriate assumptions to render the specification complete.

14.20. What update anomalies occur in the EMP_PROJ and EMP_DEPT relations of Figures 14.3 and 14.4?

14.21. In what normal form is the LOTS relation schema in Figure 14.12(a) with respect to the restrictive interpretations of normal form that take *only the primary key* into account? Would it be in the same normal form if the general definitions of normal form were used?

14.29. Consider the following relations for an order-processing application database at ABC, Inc.
ORDER (O#, Odate, Cust#, Total_amount)
ORDER_ITEM(O#, I#, Qty_ordered, Total_price, Discount%)
Assume that each item has a different discount. The Total_price refers to one item, Odate is the date on which the order was placed, and the Total_amount is the amount of the order. If we apply a natural join on the relations ORDER_ITEM and ORDER in this database, what does the resulting relation schema look like? What will be its key? Show the FDs in this resulting relation. Is it in 2NF? Is it in 3NF? Why or why not? (State assumptions, if you make any.)

14.35. Consider the relation:

BOOK (Book_Name, Author, Edition, Year) with the data:

Book_Name	Author	Edition	Year
DB_fundamentals	Navathe	4	2004
DB_fundamentals	Elmasri	4	2004
DB_fundamentals	Elmasri	5	2007
DB_fundamentals	Navathe	5	2007

a. Based on a common-sense understanding of the above data, what are the possible candidate keys of this relation?
b. Justify that this relation has the MVD { Book }□□{ Author } | { Edition, Year }.
c. What would be the decomposition of this relation based on the above MVD? Evaluate each resulting relation for the highest normal form it possesses.

14.36. Consider the following relation:
TRIP (Trip_id, Start_date, Cities_visited, Cards_used)
This relation refers to business trips made by company salespeople. Suppose the TRIP has a single Start_date, but involves many Cities and salespeople may use multiple credit cards on the trip.Make up a mock-up population of the table.
a. Discuss what FDs and/or MVDs exist in this relation.
b. Show how you will go about normalizing it.