1. Consider the following relational schema. Primary keys are underlined and foreign keys are in italics (you may infer what they refer to).

   PERSON: ssn, fname, lname, address, phone, bdate
   EMPLOYEE: ssn, job_title, date_hired
   SAILOR: ssn, rating
   BOATMODEL: model#, model_name, manufacturer
   BOAT: reg#, bname, color, model#
   RESERVATION: sailor_ssn, date_reserved_for, boat_reg#, emp_ssn

Provide SQL answering the following queries:

a) List the names and addresses of all employees who are not sailors.

   SELECT p.fname, p.lname, p.address
   FROM person p JOIN employee e ON p.ssn=e.ssn
   WHERE p.ssn NOT IN
     (SELECT ssn FROM sailor)

b) Determine if there are two different boats with the same name (give the name and two registration numbers in each row returned). Do not worry about repeated information in other rows returned.

   SELECT b1.bname, b1.reg#, b2.reg#
   FROM boat b1 INNER JOIN boat b2 USING (bname)
   WHERE b1.reg# != b2.reg#;

c) List the first and last name of all customers and the model name and color of the boat they have reserved, for all reservations for the month of June.

   SELECT p.fname, p.lname, bm.model_name, b.color
   FROM person p JOIN sailor s ON p.ssn=s.ssn
   JOIN reservation r ON s.ssn=r.sailor_ssn
   JOIN boat b ON r.boat_reg#=b.reg#
   JOIN boatmodel bm ON b.model#=bm.model#
   WHERE MONTHNAME(r.date_reserved_for)='June'

   d) For each customer, count the number of reservations made by that customer. List only those customers who have less than 3 reservations, and include those with zero.

   SELECT p.fname, p.lname, COUNT(r.boat_reg#) AS numRes
   FROM person p JOIN sailor s ON p.ssn=s.ssn
   LEFT JOIN reservation r ON s.ssn=r.sailor_ssn
   HAVING numRes<3

   *note:* in d) we can get away with the one LEFT join due to the way MySQL associates joins. This is viewed as ((p join s) left join r). Otherwise, we would have to be more explicit about the join order, which might be safer.

2. Design an ER diagram for a library as described below. Show relevant
constraints.
• Each employee has an ssn, fname, lname, and address.
• An employee is classified into one of three categories: managerial, research, and floor. Floor employees are paid by the hour and have an hourly wage rate. The other two categories have a salary. Research workers have a specialty, while managerial workers have a job title.
• Customers are identified by their card number, and also have a fname, lname, and address.
• Each book is identified by its LCN (Library of Congress Number). It has a title, and one or more authors.
• An author has as a key an author code, since fname, lname does not suffice. We also keep track of their birth date and date of death.
• Customers may check out books. We keep track of the date it was checked out, as well as the date of return, if it has been returned.
• Each time a book is checked out, we want to track which employee was involved in that transaction. Checking out a book can be handled by floor or research staff, but not managerial staff.
• Each member of the floor staff has exactly one member of the managerial staff as a supervisor.
3. Derive a relational schema based on the attached ER diagram (not shown here). Indicate all foreign keys and NOT NULL constraints.

**branch**: branch_id, branch_address

**securityBox**: sec_box_num, branch_id, acct_num
FK: branch_id refers to branch not null
FK: acct_num refers to account

**securityBoxUsage**: sec_box_use_datestamp, sec_box_num, branch_id
FK: (sec_box_num, branch_id) refers to securityBox not null
account: account_num, account_type, primary_cust_num, second_cust_num
FK: (primary_cust_num) refers to customer(cust_num) not null
FK: (second_cust_num) refers to customer(cust_num)

customer: cust_num, cust_name, cust_status

check: check_num, account_num, check_amount
FK: (account_num) refers to account not null