1. Consider the following relational schema. Primary keys are underlined and foreign keys are in *italics*.

```
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 boat names and model names of all boats that have no reservations.

```
a) SELECT bname, model_name
   FROM boat JOIN boatmodel USING(model#)
   WHERE reg# NOT IN (SELECT boat_reg# FROM reservation)
```

   b) Count the number of boats made by each manufacturer. List them in decreasing order of the number of boats.

```
b) SELECT manufacturer, count(*) as total
    FROM boat join boatmodel using (model#)
    GROUP BY manufacturer
    ORDER BY total DESC
```

   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 in November 2016.

```
c) SELECT p.fname, p.lname, b.color, m.model_name
    FROM person p JOIN sailor s using(ssn)
    JOIN reservation r on s.ssn=r.sailor_ssn
    JOIN boat b using(reg#)
    JOIN boatmodel m using(model#)
    WHERE MONTHNAME(date_reserved_for)=’November’ AND
    YEAR(date_reserved_for)=2016
```

   d) List the names of all sailors whose reservation was handled by any of the three most least busy employees (an employee is considered busy or not busy by counting the number of reservations they have handled). You may assume that all employees have different numbers of reservations (no ties not a concern).

```
d) (note: limit works in a subquery in the from-clause)
SELECT p.fname, p.lname
FROM person p JOIN sailor s using(ssn)
JOIN reservation r ON s.ssn=r.sailor_ssn
JOIN (SELECT boat_reg#, COUNT(*) total
     FROM reservation
     GROUP BY emp_ssn
     ORDER BY total
     LIMIT 3) n ON r.emp_ssn=n.emp_ssn

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.

Here are two solutions, one using the Chen ER notation from the text, and the other done in MySQL Workbench.
3. **[15 points]** Derive a relational schema based on the attached ER diagram. Indicate all foreign keys and NOT NULL constraints.
Note: Any attribute below that is part of the primary key (underlined) must be NOT NULL.

**branch**: branch_id, branch_address

**securityBox**: sec_box_num, branch_id, cust_num
FK: branch_id refers to branch not null
FK: cust_num refers to premiumCustomer

**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_address
premiumCustomer: cust_num, prem_status
FK: cust_num refers to customer(cust_num) not null

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