1. Consider the AIRCRAFT database described by the following Crowsfoot diagram.

Write SQL queries for this schema which

a) Show all models (code and name) of aircraft that have been chartered (that is, rented) by a customer who has area code 541.

```sql
SELECT m.mod_code, m.mod_name
FROM MODEL m JOIN AIRCRAFT a USING(mod_code)
   JOIN CHARTER ch ON ch.AIRCRAFT_ac_number=a.ac_number
   JOIN CUSTOMER c ON ch.CUSTOMER_cus_code=c.cus_code
WHERE c.cus_areacode = 541;
```
b) For each pilot, list the pilot’s name and the number of charters that person has been hired for (include those with zero).

```
SELECT e.emp_fname, e.emp_lname, COUNT(c.char_trip) AS number_of_charters
FROM EMPLOYEE e JOIN PILOT p USING(emp_num)
    LEFT JOIN CHARTER c ON c.PILOT_emp_num=p.emp_num
GROUP BY p.emp_num;
```

c) List all the pilots (name and rating) that be hired from September 1, 2017 to November 7, 2017 and qualified to fly “Boeing 747” (name of model).

```
SELECT e.emp_fname, e.emp_lname, p.pil_ratings
FROM MODEL m JOIN QUALIFIED q ON m.mod_code = q.MPDEL_mod_code
    JOIN PILOT p ON q.PILOT_emp_num = p.emp_num
    JOIN EMPLOYEE e USING(emp_num)
WHERE m.mod_name = 'Boeing 747' AND e.emp_hire_date >= '09/01/2017' AND e.emp_hire_date <= '11/07/2017';
```

(Date format can also be '2017-09-01' or '20170901'; using “BETWEEN AND” in where clause is correct too.)

d) Show all charters whose assigned pilot is NOT qualified to fly the model of aircraft which is assigned to that charter.

```
SELECT c.char_trip
FROM CHARTER c JOIN PILOT p ON c.PILOT_emp_num = p.emp_num
    JOIN AIRCRAFT a ON c.AIRCRAFT_ac_number = a.ac_number
    JOIN MODEL m ON m.mode_code = a.mode_code
    LEFT JOIN QUALIFIED q ON q.PILOT_emp_num = p.emp_num AND
        q.MODEL_mod_code = m.mod_code
WHERE q.PILOT_emp_num IS NULL;
```

OR

```
SELECT c.char_trip
FROM CHARTER c JOIN PILOT p ON c.PILOT_emp_num = p.emp_num
    JOIN AIRCRAFT a ON c.AIRCRAFT_ac_number = a.ac_number
    JOIN MODEL m ON m.mode_code = a.mode_code
WHERE (p.emp_num, m.mode_code) NOT IN
    (SELECT PILOT_emp_num, MODEL_mod_code FROM QUALIFIED)
```

e) (551) List all charters that use an aircraft of the most common model. (That is, look at the number of aircraft for each model. The most common model has the most aircraft. List charters that use one of those models. Include ties and give the model name.)
2. Draw an ER Diagram for the CAR_INSURANCE problem.

- There are several insurance companies, with a unique code, they also have a name and an address.
- There are many customers, with customer id, first and last names, address, phone, and email.
- Customers can own many several cars (with license, vin, make, model, year), but a car is owned by exactly one customer.
- Customers purchase policies from the companies for their cars. A policy has a text, a policy number, and a cost. A policy involves one car, one customer, and one company.
- Two different companies may issue a policy with the same policy number (so policy is weak).
- Accidents need to be recorded. Accidents have a date (partial key), a description, and are charged to (and owned by) a policy.

Be sure to use the Chen ER notation from the text or the Crowsfoot style as used by MySQLWorkbench.
3. Convert the GRANT ER Diagram to a relational schema. Indicate primary keys and foreign keys.

RESEARCHER: first_name, last_name, email, institution

ADVISORYBOARDMEMBER: first_name, last_name
  first_name, last_name foreign key to RESEARCHER

GRANT: proposal_num, title, abstract, amount, first_name, last_name
  first_name, last_name foreign key to ADVISORYBOARDMEMBER

QUESTION: ques_num, ques_text

PROPOSES: proposal_num, first_name, last_name
  first_name, last_name foreign key to RESEARCHER
  proposal_num foreign key to GRANT
REVIEW: first_name, last_name, proposal_num
first_name, last_name foreign key to RESEARCHER
proposal_num foreign key to GRANT

ANSWER: first_name, last_name, proposal_num, ques_num, score
first_name, last_name, proposal_num foreign key to REVIEW
ques_num foreign key to QUESTION