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).

```sql
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) 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.mod_code=a.mod_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.mod_code=a.mod_code
WHERE (p.emp_num, m.mod_code) NOT IN
    (SELECT PILOT_emp_num, MODEL_mod_code FROM QUALIFIED)
```


d) (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.)

```
SELECT c.char_trip, a.mod_code
FROM CHARTER c JOIN AIRCRAFT a ON c.AIRCRAFT_ac_number=a.ac_number
    JOIN (SELECT m1.mod_code
            FROM AIRCRAFT a1
                JOIN MODEL m1 USING(mod_code)
            GROUP BY mod_code
            HAVING COUNT(*)>=ALL
                (SELECT COUNT(*)
                    FROM AIRCRAFT a2
                        JOIN MODEL m2 USING(mod_code)
                    GROUP BY mod_code)) AS p USING(mod_code)
```

2. Draw an ER Diagram for the TOUR_OF_OREGON problem. Here we are designing a model for a bicycle race which goes from city to city in Oregon.

- We need to keep track of all **people**, who are identified by a `p_id`. We also want to keep their **name** and **email**.
- There are also two types of people: **employee** and **rider**. For employees we track their **role**, and for riders their **bib_num**.
- There are several **cities** on the tour. For each we have a `city_code`, **name**, and **elevation**. Each employee is assigned to a single city.
- There are a number of **legs**, each with a unique `leg_num`. A leg starts in one city and finishes in another city (which need to be in the database) and has a **distance**.
- We want to track the time of a rider on a leg, which we call **leg_instance**. It will be owned by rider and leg, and have a **time**.

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

(Note: ques_num is a partial key to question.)

**STUDENT:** stud_id, stud_name

**ASSIGNMENT:** asst_num, date_due

**STUD ASST:** stud_id, asst_num, date_turned_in
(stud_id) foreign key to STUDENT
(asst_num) foreign key to ASSIGNMENT

**QUESTION:** asst_num, ques_num, ques_text, ques_max_score
(asst_num) foreign key to ASSIGNMENT

**ANSWER:** stud_id, asst_num, ques_num, ans_score, ans_text
(asst_num, ques_num) foreign key to QUESTION
(stud_id) foreign key to STUDENT