“Complicated” SQL Practice

```
CREATE TABLE Class (
    dept   VARCHAR(50),
    number INT,
    title  VARCHAR(50),
    PRIMARY KEY (dept, number));

CREATE TABLE Instructor (
    username   VARCHAR(50) PRIMARY KEY,
    fname      VARCHAR(50),
    lname      VARCHAR(50),
    started_on CHAR(10));

CREATE TABLE Teaches(
    username VARCHAR(50) REFERENCES Instructor,
    dept     VARCHAR(50),
    number   INT,
    PRIMARY KEY (username, dept, number),
    FOREIGN KEY (dept, number) REFERENCES Class);
```

1. How many classes are being taught by at least one instructor?

2. Which instructors teach more than 1 class? Give the username, first name, and last name of these instructors. Do NOT use a correlated subquery (although that is a good place to start).

3. Which CSE courses do neither Dr. Levy (username ‘levy’) nor Dr. Wetherall (username ‘djw’) teach? Give the department, number, and title of these courses.

RA

2. SQL to Relational Algebra. Write an expression in the form of a logical query plan (i.e., draw a tree) that is equivalent to each of the SQL query below:

a. Clinic(<u>cid</u>, name, street, state)

Equipment(<u>eid</u>, type, model)

Assignment(cid, eid)

```
SELECT COUNT(*)
FROM Clinic C
WHERE NOT EXISTS (
      SELECT *
      FROM Assignment A, Equipment E
      WHERE C.cid = A.cid AND A.eid = E.eid
                  AND E.type = 'Fridge' AND E.model = 1004 );
```

b. Item(oid, category, price)

Gift(pid, rid, oid)

```
SELECT O1.category, MAX(ABS(O1.price - O2.price))
FROM Gift G1, Gift G2, Item O1, Item O2
WHERE G1.pid = G2.rid
  AND G2.pid = G1.rid
  AND O1.oid = G1.oid
  AND O2.oid = G2.oid
  AND O1.category = O2.category
GROUP BY O1.category
HAVING count(*) > 5;
```

3. RA

Winter 2016 #2a, b

Consider the following database about a picture tagging website:

```
Member(mid, name, age)

Picture(pid, year)

Tagged(mid, pid)
```

(a) Write a Relational Algebra expression in the form of a logical query plan (i.e., draw a tree) that is equivalent to the SQL query below. Your query plan does not have to be "optimal": however, points will be taken off for overly complex solutions.

```
SELECT w.year, MAX(w.c) AS m
FROM (select x.name, z.year, COUNT(*) AS c
      FROM Member x, Tagged y, Picture z
      WHERE x.mid = y.mid AND y.pid = z.pid AND age < 20
      GROUP BY x.name, z.year) w
GROUP BY w.year
HAVING SUM(w.c) > 100;
```