CSE 344 Section 3 Worksheet

```
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,
    FOREIGN KEY (dept, number) REFERENCES Class);
```

1. Return the first and last name of all instructors who are teaching at least one class.

   ```
   SELECT I.fname, I.lname
     FROM Instructor AS I, Teaches AS T
    WHERE I.username = T.username
    GROUP BY I.username, I.fname, I.lname;
   ```

   Note: we must group by the Instructor's first name and last name along with the username in order to select those attributes. Also, by nature of the Teaches table, any instructor that appears in the table must teach at least one class. The GROUP BY allows us to get the distinct usernames and return the associated first and last name.

2. For each instructor, return their username and the number of classes they teach.

   ```
   SELECT I.username, COUNT(T.number) AS count
     FROM Instructor AS I LEFT OUTER JOIN Teaches AS T
          ON I.username = T.username
    GROUP BY I.username;
   ```

3. Return the first and last name of the newest instructor(s) (who started on the latest date). Assume `Instructor.started_on` uses YYYY-MM-DD format. If there are multiple instructors, list all of them.

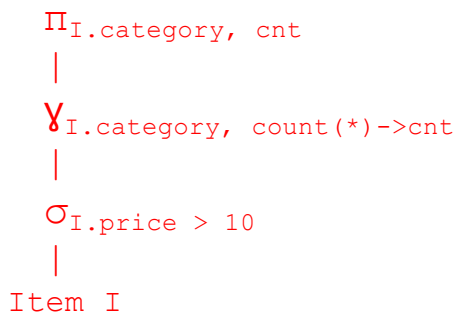
   Example of the witnessing problem that doesn't require subqueries. Here is one solution.

   ```
   SELECT I1.fname, I1.lname
     FROM Instructor AS I1, Instructor AS I2
    GROUP BY I1.username, I1.fname, I1.lname, I1.started_on
   HAVING I1.started_on = MAX(I2.started_on);
   ```

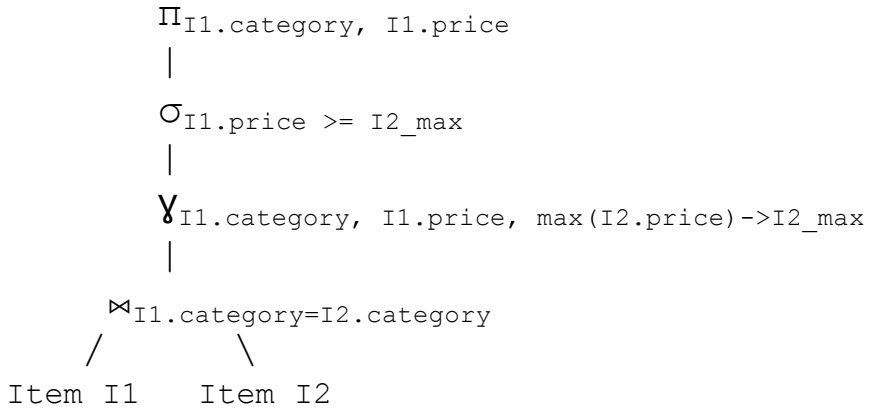
```
Item(iid, category, price)
```

4. Write a logical query plan (i.e., draw a tree) that is equivalent to the SQL query below:

```
SELECT I.category, COUNT(*)
  FROM Item I
 WHERE I.price > 10
 GROUP BY I.category;
```



5. Write a SQL query that is equivalent to the logical query plan below:




```
SELECT I1.category, I1.price
  FROM Item I1, Item I2
 WHERE I1.category = I2.category
 GROUP BY I1.category, I1.price
HAVING I1.price >= MAX(I2.price);
```