1) How do I get started with question #4

   Using a single SQL statement display ssn, fname, lname, age/2 of all the students whose first name begins with the letter J and age is greater than 25 and are taking any class that contains '**Intro'** in its description (Have to convert the dob into a number). Order the results by age/2 in descending order. Use an alias for the order by clause

   A: Notice that you will be displaying information from the students table only so the outer SQL statement will need to address only the SQL statement.

   Have to convert dob into age using the months_between function

   For class descriptions that contain the word Intro, you have to use the like clause. Class description resides in the class table. This would mean that we have to use the subquery mechanism to connect the tables. For example to connect the tables patient, patient_disease and the disease table we need to use the following structure

   Select … from patient where ssn in (select ssn from patient_disease where code = (select code from disease where …));

   Note how the tables are connecting to one another. The order by happens at the end of the SQL statement outside the parentheses because it corresponds to the outer query.

2) The queries look complicated. How can simplify the this process?
   A: Break up the problem into parts. Do each part and confirm that it works. Once you have confirmed each part, then you can begin to combine them piece by piece. You will be spending a lot more time on these problems if you try to attempt the entire problem at once as opposed to breaking it down

3) Can I use joins instead of subqueries?
   A: No, you have to use subqueries. Joins are covered in the next chapter

4) How should I deal with null
   A: Be sure to not use = or !- for filtering for null. You have to use is or is not