CS 155A Spring 2016 | Assignment 06

Due Date: March 09, 2016 11:00 PM **Points:** 35 point max

General Directions

This assignment uses the tables from the vets database.
The goal of the assignment is to get you to think about joins- inner join, outer joins, and subqueries that do the work of joins by working with data from two tables.

If you do a join- where you have the names of two or more tables in the From clause, then you must use the condition join syntax or the column name join. If you do the join in the Where clause, you will not get any credit for that task. That join is referred to in the notes as the Legacy Comma Join- do not use that join for assignments.
Take care that you do not accidentally do a Cartesian product. If your result set contains a thousand rows (or even a hundred rows for this assignment) you have probably done a Cartesian product and you need to correct the query.

The tasks require specific techniques to be used. If you do not use the technique specified, you will get no points for that task.
Do not assume that these will all require outer joins because that was the topic for the week. Use an inner join if that will solve the task; use an outer join only if it is required.

Use the fewest tables possible for the queries. For example if I ask you to find animals with no exams, you need to use the exam headers table. But you do *not* need the exam details table. Sometimes adding extra tables makes your query less efficient; other times it makes your query incorrect.

The use of meaningful table aliases is required in assignments. The table names are fairly long and when you need to qualify a column, that expression gets longer and it can be harder to read. You define the table aliases within in each query. It can help to have a consistent set of table aliases to use. These are the ones I use; you can use them or different aliases that suggest the table.

clients	cl
animals	an
exam headers	eh
exam details	ed

The use of declared variables is allowed if it makes the query easier to write.

Tasks

For tasks 1-3, do not display duplicate rows; but think carefully about this and do not use Distinct if it is not required by the logic of the query.

Task 01: Use a join to display the client id and last name for each client who does not have an animal. Do this using a join.

Task 02: Use a subquery to display the client id and last name for each client who does not have an animal. Do not use any join for this task- just the subquery.

Task 03: Use a subquery to display the client id and last name for each client who does not have a reptile. This will include clients with no animals. Do not use any join for this task- just the subquery.

Task 04: For each client in the client table, display the following: client id, client last name, animal name and animal type. If the client has no animals, then the columns for animal name and animal type will both display the message "no animals". If the client has an animal with no name, then the column for animal name will display the message "no name". Suggestion- look at the columns for the animal table and see which are nullable.
The following are examples of possible rows.

```
+--------+--------------+-----------------+------------+
| cl_id  | cl_name_last | an_name         | an_type    |
+--------+--------------+-----------------+------------+
|      1 | Adams        | no animals      | no animals |
|      2 | Baker        | Pinkie          | bird       |
|      2 | Baker        | no name         | cat        |
|      3 | Carl         | Max             | hedgehog   |
|      3 | Carl         | Morritz         | hedgehog   |
```

For Tasks 05 and 06 use joins. Do not use any subquery for these tasks. Missing data can be displayed as a null.

Task 05: We want to find animals for which we have no exam records (use the exam header table). Show the client id, and last name and the an_id, name and type for every animal that does not have any exam records in the exam table. Sort by the client id and animal id. For this query, do not display clients who have no animals.

Task 06: For this task use the same logic as in Task 05 except that your result will include clients with no animals.

For the rest of the tasks, you can use either joins or subqueries or both.

Task 07: Display the client id and client last name and the animal id, name, and date of birth for all animals owned by a client who lives in New York or Massachusetts but do not display rows for any of the dogs, cats or birds.

Task 08: Display the animal id and name for any animal that has at least one exam in the current year. Do not hard code the year.

Task 09: Display data about services we have in the services table that have a list price of $100 or more and that have not been used on any exam. Display the service id, list price, description and service type.