Due Date:	Sunday 2013-06-30 1100 p.m.
Points:	40 points max
Turn In:	The zipped file containing the script and spool files.

General Directions

This assignment uses the tables associated with 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 column name join syntax or the condition join syntax. 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 you have done a Cartesian product and you need to correct the query.

Task 01, 02, 03, 04 specify a technique to use. For the other tasks you can decide on the technique. Do not assume that these will all require outer joins because that was the topic for the unit. Use an inner join if that will solve the task; use an outer join only if it is required. You may use subqueries if you wish.

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 table makes your query less efficient; other times it makes your query incorrect.

The use of meaningful table aliases is encouraged in this and future assignments. If you are working on a case-sensitive system (linux type system) the table names and the table aliases are case specific.

One of the tasks requires the use of a variable; the use of variables for other tasks is allowed if it makes the query easier to write.

For this assignment the term "reptile" is defined as the animal types: snake, chelonian, crocodilian, and lizard; the term "rodent" is defined as the animal types: hamster, capybara, porcupine and dormouse.
An animal that does not have a name is still an animal; if the name is null then just display the system default for a null.

Tasks

Task 01: Use an outer join to display the client id and last name for any client who does not have an animal. Do not use a subquery for this task.

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

Task 03: Display the client id and last name any client who has at least one animal. Display the client's data only once even if they have more than one animal. Do this using a join. Do not use a subquery for this task.

Task 04: Display the client id and last name any client who has at least one animal. Display the client's data only once even if they have more than one animal. Do this using a subquery; do not use a join.

Task 05: We want to see the id and last names of all clients; if they have any animals, include the name and type of their animals. If the client has no animals, the result will display nulls for the animal name and type. Sort by client id and animal type.

Task 06: Display the client id and last name for any client who has at least one animal that is a reptile. Display the client's data only once.

Task 07: Display the client id and last name for any client who has at least one animal but does not have a reptile. Display the client's data only once.

Task 08: Display the client id and last name for any client who has at least one animal that is not a reptile. Display the client's data only once.

Task 09: Some of the animals owned by client 1825 have an infectious disease. The vet wants to see data (follow the sample header shown here) for all animals owned by a client who lives in the same location as client 1825. Include the data for animals owned by client 1825.
The third column is the year of the animal's exam' if they have any exam record.
For this task location is defined as the City and State.
Define one variable for the client ID; assign the value 1825 and use that variable in the query.
Use only that one variable.
Sample rows

```
+-------+-----------+----------+
| an_id | an_type   | ExamYear |
+-------+-----------+----------+
| 58785 | cat       |     2013 |
| 85456 | dog       |     NULL |
| 97451 | dog       |     2012 |
| 78787 | dormouse  |     2013 |
```

THE END