CSCI 466 Assignment 5

SQL - DDL (50pts)

Introduction

For this assignment, you must write an SQL script. An SQL script is a text file that contains a sequence of SQL commands to be run. Yours should include commands for each of the following, in order:

1. Run a statement that will remove all of the tables that will be created below. This will allow the script to be run again without any errors caused by existing tables.
2. Create a table called `dogs` with a dog id, a breed, and a name. The id of the dog should be the primary key, and should be automatically assigned the next available value when inserting a new row into the table.
3. Put **five** rows into the `dog` table with example data. Make up the data yourself.
4. Run the command `DESCRIBE dogs;`
5. Run the command `SELECT * FROM dogs;`
6. Create a table called `visits` that contains a visit id as primary key (this should take the next available key value when a new row is added). It should also have a foreign key that references a row in the `dog` table, and an attribute to hold the date that the visit took place.
7. Put at least **eight** new rows into the `visits` table. Since there are only five dogs, this means that some dogs will have multiple visits.
8. Run the command `DESCRIBE visits;`
9. Run the command `SELECT * FROM visits;`
10. Add a column to the `visits` table to hold the time of the visit.
11. Change the value for the newly-added attribute in several of the existing rows. There are several date/time functions available in SQL, and you can choose to use any that are appropriate.
12. Run the command `SELECT * FROM visits;` again.

If a data type is not specified in this document, anything *reasonable* should be fine. Use your discretion.

You should include comments in the script that identify which commands match which of the requirements. Comments in SQL are written with a `--` before them.

What to turn in?

Submit, through Blackboard, the following:

1. A text file containing the script you wrote that accomplishes the above. It should be suitable for running with the `\.` command in the SQL client.
2. Another text file containing the output generated after running the script your wrote. You can use `\T filename` to start logging to a file called `filename`, and `\t` to stop logging when you are done. It is also possible, though not recommended, to get the output by copying and pasting from the terminal.

Make sure to choose names for these two files that make it easy for the TA to determine which is which without having to open the file.