Dog Registration Database

Timeline

- Due Date: April 25, 2018
- Cutoff Date: April 30, 2018
- Late points will be deducted for projects submitted after the due date. Five points will be deducted each calendar day submitted after the due date.
- No projects will be accepted after the cutoff date.

Objective

- Create a database to manage NYC dog registration
- Create reports

Dog database requirements

- Dog owner information including name, full address, email and phone. Owners can have multiple phone numbers.
- Dog owners can own many dogs. Track all dogs owned, including current and previously owned.
- Dog information including dog name, breed, gender, weight, date of birth, age, owner.
- A dog has one current owner, but can have many previous owners.
- A dog has many photos. Identify the photo name and date of photo. For instance *rover_running.jpg*
- Dog owners can receive tickets with violations. Tickets include the owner, dog, violation type, fine, date of violation and current status of violation. Dog owners can receive many tickets.
- Tickets are limited to specific types of violations. For instance, violation 1 (*no leash*); violation 2 (*loud barking*), etc.

Data

You must enter at least the following data

- At least 20 dog owners.
- At least 50% of the owners will own more than one dog. Create one owner with at least 10 dogs. One dog can’t be simultaneously owned by the same owner.
- At least 5 violation types
- At least 10 tickets
- Import at least 1,000 dog records from NYC Open Data

Questions

- Generate SQL to answer the following queries.
- Replace underlined items with values of your own choosing. For instance, replace three months with your own date range.
- Create data to insure all questions generate output.
- Format all output. For instance, all numbers will display with commas, dollar values will display with a $ prefix and create descriptive labels for all columns.

1. Identify dogs without violations in the last year. Display the owner name, dog name, breed and email. Use a nested select to answer this question.
2. Identify zip codes without registered pit bulls now. Display the zip code. Remove duplicate zip codes. Use a nested select to answer this question.
3. Identify dogs without owners now. Display the dog name, gender, breed and age. Use a nested select to answer this question.
4. Identify current dog owners with a residence near Queens College. Display the owner name, dog name and breed.
5. Identify pictures of female poodles less than five years old. Display the dog name, age and all photos.
6. Identify dogs owned by Sally Smith with violations in the last year. Display the owner name, dog name, violation, date of violation and fine.
7. Identify the number of dogs by gender. Display two columns and one row for each gender. The two output columns are gender and number of dogs with that gender. Use a function to answer this question.
8. Identify zip codes with the most pit bulls. Display two columns and one row for each zip code. The two output columns are zip code and number of dogs in that zip code. Display the zip code with the most pit bulls first. Use a function to answer this question.
9. Identify the number of total fines by owner in the last two years. Display three columns and one row for each owner. The three columns are owner name, number of violations and total dollar amount of fines. Display owners with the most fines first. Use a function to answer this question.
10. Increase all fines by 20%. Identify the SQL to perform this operation and the fine before and after this operation.
11. Display the structure of all tables using the SQL Describe operation.
12. Display the Oracle version by entering

```
select *
from product_component_version;
```

Additional Design Requirements

- Create your database using Oracle. Utilizing other databases requires prior instructor approval.
- Normalize your database to third normal form.
- Output for all questions must include at least one row displayed.
- All multi value columns must be saved to their own table.
- Identify and create primary keys for each table.
- Create foreign keys to enforce referential integrity. For instance, you must have foreign keys with references to ***at least*** the following:
    a. Dogs and owners
    b. Owner and phones
    c. Owner and tickets
    d. Ticket and violation
- Include the question, SQL command to answer the question and output from the SQL command.
- Include all SQL commands to create your database and answer questions including create tables, search, update, insert data, alter column names and alter column types.
- Create descriptive column labels for all output.
- Clearly label each question and answer.

Formatting

- Your project must include the question and SQL operations to answer the question
- The column output should be displayed in a `non-proportional font such as courier`. This will display the columns vertically straight.
- All columns in your search must display on one line. Don't wrap columns to two lines.
- Your project must be typed.
- All pages of your output must include your name, class, date and project number in the header of each page.
- The first page of your project must include your name, the last four digits of your student id, class, the submission date and the project number.

Submission

- All pages of your project must be combined into one MS Word or one Adobe PDF file. Files not submitted in this format will be rejected. Don’t submit SQL code and output in separate documents.
- An electronic copy of your project will be submitted to Blackboard on the due date. The file name will be in the format: [last name] [first name] Project2.docx or [last name] [first name] Project2.pdf. For example, *Smith Sally Project2.pdf.* Submit one MS Word or one Adobe PDF file. Files not submitted in this format will be rejected.
- Do not submit hardcopies of the project.
- Projects will not be accepted after the cutoff date.
- Late points will be deducted for projects submitted after the due date. Five points will be deducted each calendar day submitted after the due date.

Academic Integrity

Projects and examinations must represent your own work. Group projects and exams are not permitted. Although you are encouraged to ask other students for information, you should neither copy another student's project nor permit another student to see your work. You can be asked to perform specific procedures and operations in the presence of the instructor. A student who submits a project that is too similar to another student's work will receive a ZERO for the project. Additional penalties may be imposed. Students found guilty of any form of academic dishonesty such as plagiarism or cheating on an exam or computer project are subject to discipline, including, but not limited to, failure in the course and suspension or dismissal from the College. You are required to comply with the CUNY Policy on Academic Integrity available at
http://www.cuny.edu/about/administration/offices/la/Academic_Integrity_Policy.pdf