

Lab 01: Pokemon

1 Objective

In this assignment you will gain experience using the Linux command line and the command-line PostgreSQL client. You will write SQL statements for creating a table and querying its data.

2 Setup

You need two pieces of software to work on this lab: a text editor and an SSH client. There are many options for each.

2.1 Text Editors

A text editor is a program that lets you edit simple text files. This is different from a word processor, which contains a lot of non-textual formatting. An IDE like Eclipse is built around a text editor, but since the text we write will not be in the Java language it would probably not be a good one to choose.

- Every operating system comes with a built-in text editor. On Windows that would be `Notepad`; on MacOS it is `TextEdit`; on most distributions of Linux it is `gedit`.
- There are a variety of more powerful but free text editors that you might choose to download and install. `Atom` is a popular choice that runs on any operating system. `Notepad++` is another nice editor for Windows.
- You might already be using `Visual Studio Code` for other courses, and it also makes a perfectly serviceable text editor.

2.2 SSH Clients

An SSH client is a program that allows you to connect to a remote computer, send it commands, and see the output of those commands.

- An SSH client is built-in to MacOS and Linux; you can use it simply by opening a terminal. The most recent version of Windows has this too.
- For older versions of Windows, if you have previously installed the Windows Subsystem For Linux you should also have a built-in SSH client for Windows that works similarly. If not, you could install WSL.
- The free tool `putty` is small and can be downloaded and run from an older Windows computer without requiring installation.

- Regardless of your operating system, you may already be using **Visual Studio** to connect to computers in the Linux lab through SSH. If so, you could also set it up to connect to `csci366.millersville.edu` instead.

Using your chosen text editor, create a new file named `366-lab01.sql` or something similar. SQL files use a pair of consecutive hyphens to start single-line comments, so write a first line that is a comment with your name.

Using your chosen SSH client, connect to `csci366.millersville.edu`. For most clients this means typing `ssh [your username]@csci366.millersville.edu` into a terminal, replacing `[your username]` with the username that Millersville has provided you. While connected, any commands you type will be executed on the server. You should start by creating a directory there for files related to this course (perhaps calling it `366`) and making it your working directory. If you aren't sure how to do that, you can run the command `mkdir 366` followed by `cd 366`.

We would now like to copy a file we are providing to you into that directory. To do so, run the command `cp /home/faculty/chogg/Public/366/pokemon.csv ./`. You can either close your SSH connection or keep it open for future use, although it will be automatically closed for you if you do not use it for some period of time. Whenever you re-connect, change your working directory back into the one where you copied that file.

3 Assignment

For each question below, you are going to write your SQL code into your text file. (Please use a comment to label each answer with its question number.) You will then use your SSH client to run that code in a database. You will then also copy the results back into your text file. Details on how to use the SSH client will come after the first question.

We are going to be creating a database of monsters from the Pokemon universe, with various statistics about them.

1. (20 points) Write an SQL statement that creates a table named `pokemon`. It needs to have the following columns, with the provided names in the provided order:
 - `name`, the name of this type of Pokemon.
 - `type_1`, a category into which this type of Pokemon can be classified.
 - `type_2`, an optional second category into which this type of Pokemon can be classified.
 - `total`, the sum of all this type of Pokemon's other numeric statistics.
 - `hit_points`, the amount of damage this type of Pokemon can take before dying.
 - `attack`, the strength of this type of Pokemon when attacking.
 - `defense`, the strength of this type of Pokemon when defending.
 - `special_attack`, the strength of this type of Pokemon when making special attacks.
 - `special_defense`, the strength of this type of Pokemon when defending against special attacks.
 - `speed`, the speed of this type of Pokemon.

- **generation**, the generation in which this type of Pokemon was created.
- **legendary**, whether or not this type of Pokemon is legendary.

For example, one Pokemon has name Bulbasaur, first type Grass, second type Poison, total 318, hit points 45, attack 49, defense 49, special attack 65, special defense 65, speed 45, generation 1, and legendary false. Another Pokemon has name Entei, first type Fire, no second type, total 580, hit points 115, attack 115, defense 85, special attack 90, special attack 75, speed 100, generation 2, and legendary true. A third Pokemon has name Foongus, first type Grass, second type Poison, total 294, hit points 69, attack 55, defense 45, special attack 55, special defense 55, speed 15, generation 5, and legendary false.

You will need to choose an appropriate type for each column. You will also need to choose an appropriate primary key for the table. You do not need to write any not-null or check constraints.

After you have written your code for creating the table, it will be time to run it. If you are not connected to the database server through SSH, reconnect. Remember to change to the directory where you put the `pokemon.csv` file. Once you are connected to the database server, you will want to run the PostgreSQL client. To do so, type `psql pokemon_[your username]`, again replacing `[your_username]` with your username.

You can then copy your code for creating the table from your text editor and paste it into your SSH client. Run the code. If it was not successful, fix your code and try again. Once it is successful, copy PostgreSQL's response and paste it back into your text editor as a comment.

If you decide later that you made a mistake, you can drop your table and re-create it.

2. (0 points) We would like our table to contain information about a great many Pokemon types. Rather than type the code to insert them one at a time, we will copy information from a file that has been pre-filled with this information – the file that you copied at the end of the Setup section. You can do this by running the following command in the PostgreSQL client: `\copy pokemon FROM 'pokemon.csv' CSV`
3. (10 points) Write (in your text file, of course) an SQL command to create a new Pokemon type with the name Marauder, the first type Fighting, no second type, total 400, hit points 100, attack 70, defense 50, special attack 100, special defense 30, speed 50, generation 6, legendary true.

Copy-paste the code to the PostgreSQL client, then copy-paste the result back into a comment in your text editor.
4. (10 points) Write an SQL command to find the names of all legendary Pokemon from the first generation. For this and all future questions, continue to write your code in the text file, copy-paste it to run in the PostgreSQL client, then copy-paste the result back into your text file. For long results, you can put them in `/*` and `*/` to make multi-line comments.
5. (10 points) Write an SQL command to find the names of all Pokemon with a speed over 150 whose attack is higher than their defense.

6. (10 points) Write an SQL command to find the first types and second types of all Pokemon who have a speed that is strictly greater than 40 but strictly less than 45.
7. (10 points) Write an SQL command to find the total number of Pokemon.
8. (10 points) Write an SQL command to find the highest hit points that any Pokemon from the third generation has.
9. (10 points) Write an SQL command to find the names of all Pokemon whose names start with the letter Z.
10. (10 points) Write an SQL command to delete the Pokemon named Marauder.

4 Submission

Please submit your completed `366-lab01.sql` file (or whatever you named it) to autolab. You can submit as many times as you like. The file will not be autograded.