PostgreSQL Instructions

PostgreSQL Documentation

You'll find documentation for postgreSQL at the **postgreSQL home page (https://www.postgresql.org/)** including information on how to download it for your own platform, should you choose (you are welcome to complete all coursework on the CS Teaching Labs without installing software on your own machine). Since our labs are running postgres version 9.5, that's the version you should install.

- **PostgreSQL 9.5 Documentation (https://www.postgresql.org/docs/9.5/static/index.html)**
- **PostgreSQL Tutorial (http://www.postgresqltutorial.com/)**

Using PostgreSQL on the CS Teaching Labs

We encourage you to use the CS Teaching Labs environment as your main computing environment for the SQL portion of this course. This is a brief tutorial on how to connect to and use PostgreSQL on our labs, for your work on exercises and assignments.

You should be logged in to the CS Teaching Lab (https://www.teach.cs.toronto.edu//resources/cdf_account_management.html), either in a lab or connected remotely, before starting this tutorial.

Part 1: Connecting to a Database

Every student has a separate database hosted on our database server. These instructions will connect you to your database via PostgreSQL.

1. Open a terminal window.
2. ssh in to dbsrv1, the database server machine:

```
> ssh <your_cs_teaching_lab_username>@dbsrv1.teach.cs.toronto.edu
```

   (Substitute your actual CS Teaching Lab username in where shown above, and do not type the angle brackets.)

3. Connect to your database by using the command `psql`:

```
> psql csc343h-<your_cs_teaching_lab_username>
```

   (Again, substitute your actual CS Teaching Lab username in where shown above, and do not type the

(Again, substitute your actual CS Teaching Lab username in where shown above, and do not type the angle brackets.) You should see the following output, ending with a new psql prompt:

```
psql (9.5.9, server 9.5.8)
Type "help" for help.

csc343h-...=>
```

4. To exit psql type `\q`.

`psql` is a terminal-based front-end to PostgreSQL that enables you to type in queries interactively, issue them to PostgreSQL, and see the query results.

Here are some useful psql commands:

- `\d` : Show description of a specific table - use 'q' to go back to the prompt. If no table is specified, this command displays all the tables.
- `\q`: quit psql
- `\i <filename>`: Run the SQL commands in the specified file

You can find more details about psql online in the following link: https://www.postgresql.org/docs/9.5/static/app-psql.html.

Part 2: Loading a Sample Database

Let's actually load in a database.

1. Create a new directory ("mkdir") to store the sql files, and then go into ("cd") that directory.

```
> mkdir csc343db
> cd csc343db
```

2. Copy ("cp") the database file from us to the directory you just went into. We'll discuss the contents of the file in Part 4 below.

```
> cp ~csc343h/fall/public_html/in_class/w4/world.sql .
```

Notice the blank and the dot at the end of the line. These really are necessary. If you are having difficulty, cut and paste the command onto the linux command line.

3. Connect to your personal database using psql:

```
psql csc343h-<your_cs_teaching_lab_username>
```

4. Execute the following command inside psql to load the sample database in your database:

4. Execute the following command inside psql to load the sample database in your database:

```
=> \i world.sql
```

5. Since users often store multiple database schemata, set the search path to the "World" schema to indicate which schema we're working with. Then, use the `\d` command to display all tables:

```
=> SET search_path TO World;
=> \d
```

If everything worked, you should be able to see two tables (**country** and **countrylanguage**).

Part 3: Making queries in the PostgreSQL shell

There are two ways to make queries using PostgreSQL: via the interactive shell `psql`, and by writing the queries in a file and importing them to be executed.

To practice the first method, run this:

```
=> SELECT * FROM country;
```

You should see a table containing the results, which you can scroll through (Press `q` to return to the psql shell). Any valid SQL query can be input in this way.

Part 4: Making queries by importing them into PostgreSQL

Next, exit `psql`, and open your favourite text editor to create a new file with the following SQL commands:

```
SET search_path TO World;
SELECT * FROM countrylanguage WHERE countrycode = 'CAN';
```

Save this in a file called `sample_query.sql` in your csc343db directory. Then, you can run the query by re-entering psql and using the following command:

```
> psql csc343h-<your_cs_teaching_lab_username>

=> \i sample_query.sql
```

The output of the query will be printed to the console (again, press `q` to return to the shell).

Part 5: Anatomy of the SQL file (preview for weeks 6-7)

Open up `world.sql` in your favourite text editor. This file is divided into three parts, although it is common to separate the schema definition and table insertion into separate files.

1. Lines 1-28 are the **definition** of the schema: they define the tables and attributes. Note that the type of each attribute is specified.
2. Lines 31-1257 populate each of the tables with tuples.
3. Lines 1260 and onwards specifies the key and foreign key constraints on the tables.

One quick reminder about syntax: keywords and identifiers are not case-sensitive, but string literals are, and require *single*-quotes.