PostgreSQL Instructions

We encourage you to use the CDF 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 CDF, for your work on exercises and assignments.

You should be on CDF, 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 a CDF 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 dbsrv1.cdf.toronto.edu
   ```

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

   ```
   > psql csc343h-<your_cdf_username>
   ```

   You should see the following output, ending with a new psql prompt:

   ```
   psql (9.1.14)
   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: http://www.postgresql.org/docs/9.1/static/app-psql.html.

Part 2: Loading a Sample Database

Let's actually load in a database.

1. Create a new directory to store the sql files.

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

2. Get the database file from us. We'll go over the contents of the file below.

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

3. Connect to your database using psql:

   ```
   psql csc343h-<your_cdf_username>
   ```

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

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

5. Since users often store multiple database schemata, we need to 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

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

For the first method, connect to the database (running `psql` as usual), and run:

```
=> SET search_path TO World;
=> SELECT count(*) FROM country;
```

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

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 is a file called `sample_query.sql` in the current directory. Then, you can run the query using the following command:

```
> psql csc343h-<your CDF>

=> \i sample_query.sql
```

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

Part 4: 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 note about syntax: keywords and identifiers are not case-sensitive, but string literals are, and require *single*-quotes.