National University of Singapore
School of Computing

CS3223: Database Systems Implementation
Semester 2, 2015/16

Assignment 0: Getting started with PostgreSQL (1 mark)

Due: 11:59pm January 22, 2016 (Friday)

1 Introduction

PostgreSQL is an open-source object-relational database management system that started out as a research project called POSTGRES at the University of California at Berkeley.

In this warm-up assignment, you will learn how to install `PostgreSQL` and run SQL queries using `PostgreSQL`'s client program called `psql`. This is an auto-graded individual assignment: you will receive 1 mark if you submit a log file capturing a successful `psql` session; no credit will be given if you did not submit this assignment.

2 SoC's Compute Cluster

You will be using SoC's Compute Cluster for all the assignments with `PostgreSQL`.

1. You will need to use your SoC account to access the SoC Compute Cluster. If you do not have a SoC account, proceed to `https://mysoc.nus.edu.sg/~newacct` to create one.

2. With your SoC account, proceed to `https://mysoc.nus.edu.sg/~myacct/services.cgi` to enable your SoC Compute account. Click "Enable" under "The SoC Compute Cluster" to enable your compute cluster account.

3. You can now login to one of the access nodes in the Angsana Compute Cluster (with a prefix of "sma" for the hostname) using your SoC account as follows:

   ```
   $ ssh YOUR_SOC_USERID@angsana.comp.nus.edu.sg
   ```

   Each of the Angsana access nodes has the following configuration: 2 x Quad-Core Xeon E5520 2.2GHz with 24GB RAM running Centos Linux 5.5.

2.1 Usage Policy for Compute Cluster Access Nodes

Please take note of the following usage policy for Compute Cluster access nodes extracted from `https://docs.comp.nus.edu.sg/node/1818`.

Access nodes are nodes designated for users to log in, do their coding, compiling, run minor test runs of their codes and just about everything except running their programs for a long long time. Users who need to run long processes are strongly encouraged to move their processes to the Compute Nodes.

To ensure that the limited system resources are given to the designated purposes, the following policies are imposed on all access nodes:

- User processes that has consumed more than 60 minutes of CPU time will have reduced priority. This is to ensure that users who use the nodes for login and interactive purposes continue to enjoy good response time.
- User processes that have run for more than 15 days will be terminated.
- User processes that have been running for three days will be suspended. Suspended processes will subsequently be terminated.

3 Installing PostgreSQL

We will be using version 9.4.5 of `PostgreSQL` for all programming assignments. It is important that you install and use the right version!

You can install `PostgreSQL` using the following steps:

```
$ ssh YOUR_SOC_USERID@angsana.comp.nus.edu.sg
$ cd $HOME
$ wget http://www.comp.nus.edu.sg/~cs3223/install-pgsql-linux.sh
$ chmod u+x install-pgsql-linux.sh
$ ./install-pgsql-linux.sh
```

The installation process may take a while. On successful execution of the shell script, you will see a message similar to the following:

```
Success. You can now start the database server using:
    /home/a/alice/pgsql/bin/postgres -D /home/a/alice/pgdata
or
    /home/a/alice/pgsql/bin/pg_ctl -D /home/a/alice/pgdata -l logfile start
```

The installed files are in $HOME/pgsql (instead of the default /usr/local/pgsql directory) and a database cluster directory for storing database files is created at $HOME/pgdata. For more details on the installation, refer to `$HOME/postgresql-9.4.5/INSTALL` or `http://www.comp.nus.edu.sg/~cs3223/postgresql/doc/html/installation.html`.

For convenience, you should update the `PATH`, `MANPATH`, and `PAGER` environment variables as well as PostgreSQL's `PGDATA` variable as follows:

```
$ wget http://www.comp.nus.edu.sg/~cs3223/init-profile.sh
$ bash init-profile.sh
$ source ~/.bash_profile
```

The above will update your ~/.bash_profile file to add ~/pgsql/bin to `PATH`, add ~/pgsql/share/man to `MANPATH`, set `PGDATA` to ~/pgdata, and set `PAGER` to the pager command `less`.

4 Starting and stopping the database server

To start the database server, use the `pg_ctl` command as follows:

```
$ pg_ctl start -l logfile
```

This will start a database server process called `postmaster` which listens for client connections. The server log output will be written to the file named *logfile*.

To stop the database server, use the `pg_ctl` command as follows:

```
$ pg_ctl stop
```

To learn more about `pg_ctl`, refer to its documentation or man page.

You should remember to stop the database server before you log off.

5 Creating and querying a database

You are now ready to create your first `PostgreSQL` database and run some SQL queries.

First, start the `PostgreSQL` database server if you have not already done so. To create a database named `assign0`, use the `createdb` command as follows:

```
$ createdb assign0
```

This will create a database named `assign0`. You can learn more about `createdb` by referring to its documentation or man page.

You can now connect to your previously created database using `psql`, which is `PostgreSQL`'s interactive SQL client program as follows:

```
$ psql assign0
```

You will receive the following welcome message when `psql` is started:

```
psql (9.4.5)
Type "help" for help.

assign0=#
```

Using the interactive terminal, you can now enter SQL commands to create tables, insert data into tables, query the tables, etc. You can also read and execute SQL commands from a specified text file using the `\i <filename>` command. For example, try the following:

```
\i postgresql-9.4.5/src/tutorial/basics.source
```

The tutorial file *~/postgresql-9.4.5/src/tutorial/basics.source* contains a sequence of SQL commands to create, populate, query, update, and delete two tables "cities" and "weather". To quit from the interactive terminal, enter `\q`.

Alternatively, you can also redirect the input/output of `psql` from the command line as follows:

```
$ psql assign0 < postgresql-9.4.5/src/tutorial/basics.source | less
```

By default, psql's pager option is set to on so that the pager program (configured with the environment variable PAGER) will be used when viewing long output on the terminal. For the `less` pager program, the two basic commands are (1) press SPACE to view the next page of output and (2) press Q to quit.

You can learn more about `psql` and its commands by referring to its documentation or man page.

6 What & How to Submit

In this assignment, your task is to use `PostgreSQL` to query a small database.

1. Log in to your SoC Compute Cluster account if you've not already done so.
2. `cd $HOME`
3. Start the `PostgreSQL` server if you've not already done so.
4. `wget http://www.comp.nus.edu.sg/~cs3223/assign/assign0.zip`
5. `unzip assign0.zip`
6. `cd assign0`
   The `assign0` directory contains two files. `Resale-Flat-Prices-By-Registraion-Date-From-Mar-2012.csv`[1] is a data file containing recent resale flat prices in Singapore. Each record describes a resale flat transaction consisting of 10 attributes: month, town, flat_type, block, street_name, storey_range, floor_area_sqm, flat_model, lease_commence_date, and resale_price. `load-data.sql` is a script to create a relation *resales* using the data file.
7. Execute the commands in `load-data.sql` to create the *resales* relation:

   ```
   $ psql assign0 < load-data.sql
   ```

8. Answer at least one of the following three queries using SQL on the resales relation.
   (a) **Query 1**: Compute the following four statistics for each town: (1) the number of resale flat transactions, (2) the maximum price per square metre (psm) for the town, (3) the average psm for the town, and (4) the minimum psm for the town. The *psm* metric is defined as the ratio of *resale_price* to *floor_area_sqm*. All psm values should be rounded up to the nearest integer values, and the query result should be sorted in descending order of average psm. The following table shows part of the query result.

town	count	max_psm	avg_psm	min_psm
⋮	⋮	⋮	⋮	⋮
QUEENSTOWN	1834	9810	6556	4394
⋮	⋮	⋮	⋮	⋮
CLEMENTI	1649	8367	5465	3750
⋮	⋮	⋮	⋮	⋮

[1] Source: `https://data.gov.sg/dataset/resale-flat-prices`

(b) **Query 2**: Find all resale transactions for the town 'BUKIT TIMAH' in the month '2012-05'. Output the following 9 attributes: (1) all the attributes in the table's schema excluding town and month, and (2) a new attribute named 'rank' which denote the rank of the output tuple in terms of its resale_price value among all the output tuples that have the same lease_commence_date value (i.e., among tuples with the same value for lease_commence_date, a tuple with a higher resale_price value has a lower rank value). The results should be sorted in descending order of resale_price as shown below:

flat_type	block	street_name	storey_range	floor_area_sqm	flat_model	lease_commence_date	resale_price	rank
5 ROOM	1	QUEEN'S RD	21 TO 25	123.00	Standard	1974	890000	1
5 ROOM	14	TOH YI DR	06 TO 10	122.00	Improved	1988	731000	1
5 ROOM	16	TOH YI DR	06 TO 10	122.00	Improved	1988	695000	2
4 ROOM	14	TOH YI DR	06 TO 10	104.00	Model A	1988	608800	3
4 ROOM	16	TOH YI DR	01 TO 05	104.00	Model A	1988	605000	4
4 ROOM	6	FARRER RD	06 TO 10	91.00	Improved	1974	585000	2
3 ROOM	4	QUEEN'S RD	06 TO 10	74.00	Improved	1974	433000	3

(c) **Query 3**: For each year, find the the months with lowest total number of resales transactions for that year and the total number of transactions for that month. The results should be sorted in ascending order of year. The following table shows part of the query result.

year	min_month	min_month_total
2012	12	1493
⋮	⋮	⋮

The year and month components of the *month* attribute can be extracted using the string functions SUBSTRING(month,1,4) and SUBSTRING(month,6,2), respectively.

9. Generate your submission file (named after your student number) containing your queries and their results as follows:

```
$ psql -af query.sql assign0 > YOUR-STUDENT-NUMBER.txt
```

Here, `query.sql` refers to a file containing your SQL queries.

10. Upload your submission file `YOUR-STUDENT-NUMBER.txt` to CS3223 IVLE `Submission-Assignment-0` workbin. To copy a file from your Compute Cluster account to another machine, you can use the `scp` or `sftp` command.

7 Documentation and Resources

The following are some useful resources to learn more about `PostgreSQL`:

- PostgreSQL website
- PostgreSQL 9.4.5 documentation (also available at ∼/pgsql/share/doc)