Using psql for Question 2 of Tutorial 3

This handout describes how to attempt Question 2 of Tutorial 3 using `psql`. This exercise will also help you to get familiar with the procedure for preparing, testing and submitting your answers for Assignment 3 (to be released in Week 5). The submission process for Assignment 3 will be done on the `sunfire` server (and not using IVLE).

It is important that you try out this exercise to identify and resolve any SoC/PostgreSQL acccount administrative/configuration issues before Week 5 when Assignment 3 is released.

Your submission for this tutorial will not be graded as this is just for your own practice and to get you familiar with the submission process.

1 How to set up

1. If you've not already configured your `sunfire` environment as described in the "Using psql" handout, download the psql.pdf handout from IVLE and follow the instructions there (including the instructions in Section 4).

2. Login to `sunfire` server.

3. Download the file http://www.comp.nus.edu.sg/~cs2102/cs2102-tut3.zip as follows.

```
$ cd ~
$ wget http://www.comp.nus.edu.sg/~cs2102/cs2102-tut3.zip
$ unzip cs2102-tut3.zip
```

   The unzipped directory cs2102-tut3/ contains the following files.

   - check.sh - bash script to show the values of your `psql` environment variables.
   - setup.sh - bash script to update the `PATH` environment variable to include the directory `/home/course/cs2102/bin`.
   - tut3.sql - template SQL script for you to fill in your answers for Question 2.
   - data/ - directory containing 5 CSV files for a database instance.
   - loaddata.sql - SQL script to create database schema and load database instance.
   - solution-files/ - directory containing the solution output files for the provided database instance.
   - test.sql - bash script to compare the outputs of your SQL answers against the provided solution files.

4. Check your `psql` environment variable configurations. The following illustrates the output for a user with `PostgreSQL` account "alice" assigned to `PostgreSQL` server "psql0".

```
$ cd cs2102-tut3
$ bash check.sh
PGUSER=alice
PGHOST=psql0
PGDATABASE=cs2102
PGSQL_EDITOR=/usr/local/bin/vim
```

   If the values shown are not the expected values, refer to Step 1.

5. Update the `PATH` environment variable as follows.

```
$ bash setup.sh
$ source ~/.bash_profile
$ echo $PATH
```

   You should see the directory `/home/course/cs2102/bin` included in the value of `PATH`. Note that you do not need to repeat this step for subsequent logins.

6. Create the database schema and load the provided database instance as follows.

```
$ psql < loaddata.sql
```

2 How to prepare & check your answers

Edit the provided SQL script tut3.sql to fill in your answers for Question 2. Your answer to each SQL query will be specified as a view. A view is a virtual relation created by a `CREATE VIEW` statement that is similar to the `CREATE TABLE` statement, and views can be queried similar to conventional relations.

As an example, consider the query "Find all restaurants located in the West area". We could define a view for this query as follows.

```
create view q0 (rname) as
        select rname from Restaurants where area = 'West'
;
```

The results of the query can be computed from the view `q0` as follows.

```
select * from q0;
```

By specifying your answers as views, we can easily generate the outputs of your SQL queries and compare them against the provided solution files.

For each part of Question 2, say part 2(X), a view has been defined named `qX` in `tut3.sql` with a dummy definition consisting of a single line (e.g., `SELECT 1`) that has the comment "replace this line". To input your answer for Question 2(X), simply search for the view definition `qX` in `tut3.sql`, and replace the single line commented with "replaced this line" with your SQL answer.

You can compare the outputs of your SQL answers against the provided solutions in `solution-files/` as follows.

```
$ bash test.sh
```

The `test.sh` script will first execute your `tut3.sql` script to create the views for Question 2. Next, for each view definition qX, it will generate the output of your view qX in a file named `your-solution/qX.txt` and compare it against the corresponding solution file named `solution-files/qX.txt` using the `diff` command. The `diff` command will create an output file named `diff-dir/qX.txt` which will be empty if your solution file matches the provided solution file; otherwise, the output file will contain the differences between the two solution files. Refer to https://en.wikipedia.org/wiki/Diff_utility for an overview of how to interpret the output created by the `diff` command.

Edit `tut3.sql` and run `test.sh` again if necessary to fix any errors.

3 How to submit

Submit your SQL script `tut3.sql` as follows.

```
$ submit-tut3
```

You may make multiple submissions using the `submit-tut3` command and only the latest submission will be recorded.

As the objective of this exercise is to get you familiar with the submission procedure for Assignment 3, your submission for this tutorial will not be graded.