Assignment 2
Testing

Last updated: **Wednesday 1st March 3:48pm**
Most recent changes are shown in red ... older changes are shown in brown.
[Specification] [Database] [Examples] **[Testing]** [Submitting]

This document contains details of how you/we test your scripts.

We provide a testing script `check` and a set of test files to run tests on the same set of examples in the Examples page. To obtain the tests, first ensure that you are in the directory containing your `dirby`, `releases`, etc. scripts. Make sure that this directory does *not* contain files or directories called `check`. Then run the command:

```
unzip ~cs3311/web/22T3/assignments/ass2/testing/testing.zip
```

This adds a shell script called `check` into your directory along with a new subdirectory called `tests`, which contains (surprise!) testing files (e.g. expected outputs). The `tests` directory contains a *file* called `tests` that has a list of all the test being conducted. Note that some of the `movie` tests require us to push a number into the script via a pipe.

To run the tests, use the following command on a machine where a PostgreSQL server is running:

```
sh check
```

which will hopefully produce

```
Test 01 PASSED
Test 02 PASSED
Test 03 PASSED
Test 04 PASSED
Test 05 PASSED
... etc etc ...
```

If it fails all the tests make sure that your PostgreSQL server is running. Also, make sure the `helpers.py` file is present, if you're using it.

It works out whether a test passed by comparing an expected output file against the output from your script. Both the observed and expected output for each test are located in the `tests` directory.

The `check` script uses several different variants of the `diff` command to try to work out why your output is different to the expected. Generally, you will have to examine the differences between the expected and observed outputs (which you can find in the `tests` directory) to work out any problems.