COMP9315 19T2 DBMS Implementation

Assignment 1

Testing Resources #2

Last updated: **Monday 1st July 8:15pm**
Most recent changes are shown in red;
older changes are shown in brown.

The following collection of testing data and scripts allows you to test your `email.c` and `email.source` in a way somewhat similar to how we'll be marking it. However, the actual marking process has significantly more test cases.

You can set up the testing environment as follows:

```
$ ssh grieg
... logs into server where testing works
$ source /srvr/$USER/env
... set up PostgreSQL environment
$ cd /srvr/$USER/
... go to your /srvr/ directory
$ tar xfz /web/cs9315/19T2/assignments/ass1/testing2/testing.tgz
... unpack the testing scripts and data
$ cd testing/
... change to the directory created by the previous step
```

In the `testing/` directory, you'll find the following files:

`run-tests`

A shell script that tuns the tests

`schema.sql`

SQL commands to set up a database (see schema below)

`email.data`

SQL commands to populate the database

`check.php` (and `lib/db.php`)

A script to check the sizes of data files within the database

`queries1.sql`, `queries2.sql`, `queries3.sql`

Queries to test aspects of the `EmailAddr` data type

Each time you run the testing, two more files will be created:

`testing.out`

The output from the `run-tests` script

`pg_log`

The PostgreSQL log file created from the server instance that runs the tests

The test scripts make use of a simple database with two tables:

```
Users(username, realname)
Sessions(id, username, loggedin)
```

The `email.data` file contains 111002 `Sessions` tuples and 68988 `Users` tuples.

You run the tests as follows:

```
$ ssh grieg
... logs into server where testing works
$ source /srvr/$USER/env
... set up PostgreSQL environment
$ cd /srvr/$USER/testing/
... go to your testing directory
$ sh run-tests
... run tests, writing all output to testing.out
$ more testing.out
... find out what happened
```

Here's what the `run-tests` script does:

- stops any currently running PostgreSQL server
- goes to your `postgresql-11.3/src/tutorial/` directory
- cleans out any old copy of `email.o` via `make clean`
- runs `make email` to rebuild `email.o` and `email.sql`
- re-starts your PostgreSQL server, and runs `psql -l` to check
- creates a new database `email-test`
- loads the `EmailAddr` data type from `email.sql`
- loads the schema (`schema.sql`) and data (`email.data`)
- checks the size of the data files for `Users` and `Sessions`
- runs some SQL tests (e.g. manipulating invalid email addresses)
- runs more SQL tests (e.g. checking the operators)
- creates a hash "index" and runs queries to see if it exists

As noted above, all of the output from the above commands goes into the `testing.out` file. The script is potentially a little brittle (doesn't handle compile failures well ... just continues on, even though the compilation failed ... and then fails all subsequent tests).

While it is not possible to give definitive correct outputs for many of the tests, here are some things to look for:

- when you load the data, you should see two lines: `COPY 111002, COPY 68988`
- the size of the `Users` file should be around 5MB and the size of the `Sessions` file should be around 8MB; if they're substantially larger (e.g. 19MB and 30MB), then you don't have an optimal data structure
- the timing values for most of the queries should be way less than 100ms; although the **group by.. having** query might be longer than 100ms
- in the query plan for checking hashing, the node type should be `"Bitmap Heap Scan"`

You'll need to use a little thought in interpreting the testing output.

Enjoy, *jas*