attes

a testing tool ersatz for SQL

www.bidmotion.com

May 2017

What and why?

- A basic but flexible test framework for SQL queries
  - Several suites with multiple tests
  - Can be used to test scenarios
- Why another framework?
  - Testing SQL can be cumbersome with general frameworks
  - Easy to use frameworks did not meet my simple requirements
- Still a toy framework
  - In good old Perl 5 for good old people
  - No programming needed: describe your tests with text files

The attes directory

- git repository

  https://github.com/Hydrane/attes

- License: 3-Clause BSD

- `ls attes/`

```
Attes/              # attes Perl modules
attes.conf          # attes configuration file
rescompare.pl*      # Compare results from queries using regexes
attes.pl*           # The attes launch script
```

Launching attes — the `attest.pl` script

- Launching attes

```
./attes.pl [options] <list_of_suite_names>
```

- Options

```
-c, --config        # path to configuration file
-d, --dry           # 'dry run' mode, no test is actually run
--no-colors         # for a boring black and white look
-s, --srcdir        # path to additional SQL files to test
-t, --suitedir      # path to the test suite directory
-v, --verbose       # display more information
-h, --help          # display usage information and quit
```

The `attes.conf` configuration file

- Configuration file in INI file format
- The fields **`in bold red`** are mandatory

```
[attes]
#
# Setting dry to non-zero will enable the so-called 'dry mode' which
# only lists the test suites that would be run and does nothing else.
#
dry = 0
#
# Path to compare command.
#
compare = rescompare.pl
```

The `attes.conf` configuration file

```
[suites]
    #
    # Available test suites should be listed here following the
    # convention:
    #
    #     <suite_name>.path     = <path/to/test/directory>
    #     <suite_name>.priority = <integer>
    #
    # The priority is used to order the test suites should a suite
    # depends on others.
    #
    predictions.path     = predictions
    predictions.priority = 1

    reports.path         = /home/totoro/work/testsuites/reports
    reports.priority     = 2
```

The `attes.conf` configuration file

```
[rdbms]
    #
    # Options pertaining to the database management system used
    # to test the SQL queries.
    #
    type     = mysql       # mysql or postgresql for the time being
    host     = 127.0.0.1
    port     = 3306
    user     = totoro
    password = nashi
    #
    # The command used to start and stop the RDBMS server should
    # be mentioned here as they are OS dependent.
    #
    command.start = mysql.server start
    command.stop  = mysql.server stop
```

The `attes.conf` configuration file

```
[colors]
    #
    # Colors are used by default and can be customised here using the
    # format [rgb]RGB with R, G and B in 1..5, e.g. rgb311 or 311.
    #
    success      = rgb242
    failure      = rgb411
    suite_info   = rgb124
    suite_header = rgb441
    suite_number = rgb441
    test_extra   = rgb111
```

Anatomy of an test suite directory

- `ls my_test_suite/`

```
tests.txt          # list the tests to perform in the correct order
my_first_test/     # a test directory
my_second_test/    # another test directory
my_third_test/     # yet another one
```

- `cat my_test_suite/tests.txt`

```
my_first_test     # comments starting with # are allowed
my_second_test    # the order of the tests matters!
my_first_test     # same test may be run multiple times
my_third_test
```

Anatomy of a test directory

- `mandatory`, `optional`, and `generated` files
- `ls suite_a/my_first_test`

```
command.txt        #  test description
expected.txt       #  reference or regex file defining checks
query.txt          #  query file to execute
result.txt         #  produced output of the test
errors.txt         #  check errors if any
```

The `command.txt` file

- `cat my_test_suite/my_first_test/command.txt`

```
exe      = mysql                  # path to RDBMS binary
database = test_db                # database name
action   = source_external_file   # type of test to perform
file     = queries/reports.sql    # query file
check    = regex_file             # type of check to perform
user     = chihiro                # if different from config
password = ''                     # if different from config
if_fails = continue               # continue with next test?
```

The `command.txt` file

- `action:`

```
start_rdbms
stop_rdbms
create_database
drop_database
source_file                # queries from file (relative to test directory)
source_external_file       # queries from file in source directory
```

- `check:`

```
regex_file                 # compare result to regex file
trace_regex_file           # compare trace to regex file
compare                    # compare result using external program
none                       # no check, i.e. automatic success
```

The `command.txt` file

- `file`:

  if not specified, will default to `./query.txt`

- `if_fails`:

```
continue          # continue with the next test in the suite
stop              # stop testing this suite and go to the next one
exit              # completely stop testing and exit
jump to <test>    # jump to specified test in the same suite
```

The `expected.txt` file as a reference file

- A field-separated value file giving the result set of the queries
- Regexes are allowed between slashes `/like th(is|at)/`
- `cat reading_test_suite/my_fav_books/expected.txt`

```
1<tab>War and peace<tab>1869<tab>Russian
2<tab>The catcher in the rye<tab>/\d+/<tab>English
3<tab>/To kill a .*bird/<tab>1960<tab>English
```

- Configuration key `attes.compare` to use any tool with usage:

```
/path/to/tool <result_file> <reference_file>
```

- Provided tool, script `rescompare.pl`

The `expected.txt` file as a regex file

- `cat reading_test_suite/my_fav_books/expected.txt`

```
# check regex on each line
line_should_match: catcher\s*in\s*(the)?\s*rye
line_should_not_match: ERROR

# check regex on whole file content
content_should_match : peace[^\n]+\n.+catcher
content_should_not_match : (R|r)eboot in \d+ seconds?
```

- All regex criterion should be satisfied for the test to succeed

Because pretty colors are fun...

```
[jem@khaos attes]$ ./attes.pl -c attes.conf -s ~/dev/sql/predictions/
------------------------------------------------------------------------------
1 test suite(s) to run:
------------------------------------------------------------------------------
    [1] suite 'predictions'

------------------------------------------------------------------------------
[Suite 1/1] Running test suite 'predictions'
------------------------------------------------------------------------------
    Suite 'predictions' consists in 4 test(s).
        [1] test drop_database
        [2] test create_database
        [3] test create_tbl_mc_prediction_methods
        [4] test drop_database

    [1/4] Test 'drop_database'... OK
    [2/4] Test 'create_database'... OK
    [3/4] Test 'create_tbl_mc_prediction_methods'... OK
    [4/4] Test 'drop_database'... OK

------------------------------------------------------------------------------
Ran 1 test suites
>>  All test suites ran successfully
[jem@khaos attes]$
```

... but not all the time

```
[jem@khaos attes]$ ./attes.pl -c attes.conf -s ~/dev/sql/predictions/
------------------------------------------------------------------------------
1 test suite(s) to run:
------------------------------------------------------------------------------
    [1] suite 'predictions'

------------------------------------------------------------------------------
[Suite 1/1] Running test suite 'predictions'
------------------------------------------------------------------------------
    Suite 'predictions' consists in 4 test(s).
        [1] test drop_database
        [2] test create_database
        [3] test create_tbl_mc_prediction_methods
        [4] test drop_database

    [1/4] Test 'drop_database'... OK
    [2/4] Test 'create_database'... OK
    [3/4] Test 'create_tbl_mc_prediction_methods'... FAIL
          Result file: /Users/jem/dev/perl/attes/predictions/create_tbl_mc_prediction_methods/result.txt
          Error file:  /Users/jem/dev/perl/attes/predictions/create_tbl_mc_prediction_methods/errors.txt

>> Test suite 'predictions' failed
------------------------------------------------------------------------------
Ran 1 test suites
>> 1 test suites failed
[jem@khaos attes]$
```

... even with more debug messages

```
[jem@khaos attes]$ ./attes.pl -c attes.conf -s ~/dev/sql/predictions/ -v
------------------------------------------------------------------------------
1 test suite(s) to run:
------------------------------------------------------------------------------
    [1] suite 'predictions'

------------------------------------------------------------------------------
[Suite 1/1] Running test suite 'predictions'
------------------------------------------------------------------------------
    Suite 'predictions' consists in 4 test(s).
        [1] test drop_database
        [2] test create_database
        [3] test create_tbl_mc_prediction_methods
        [4] test drop_database

    [1/4] Test 'drop_database'...

             exe = mysql
        database = prediction_test_database
          action = drop_database
            file = query.txt
           check = regex_file

        echo "DROP DATABASE IF EXISTS prediction_test_database;" | mysql --host 127.0.0.1 --port 3306 --user root

        checking regexes from '/Users/jem/dev/perl/attes/predictions/drop_database/expected.txt' on file '/Users/
jem/dev/perl/attes/predictions/drop_database/result.txt'
            [v] dont_match: [^\s]*

    [1/4] Test 'drop_database'... OK
    [2/4] Test 'create_database'...

             exe = mysql
        database = prediction_test_database
          action = create_database
            file = query.txt
           check = regex_file

        echo "CREATE DATABASE IF NOT EXISTS prediction_test_database;" | mysql --host 127.0.0.1 --port 3306 --use
r root
```

Odds and ends

- Developed in the hope that it will be useful...
- But no hard feelings should it be sent to `/dev/null`

  I will still use it anyway ^-^
- Still a rough tool — ample room for improvement