

Programming Assignment 3
Measuring Performance Relational DB, SQL (Cloud)

Task 1: You will get world earthquake data, import into SQL and with a web interface allow users to find out (query) interesting information about those earthquakes. You will measure performance.

Your assignment is to measure performance on SQL tables: creating, querying, modifying data (tuples).

Starting with (fairly) large, well structured data at:
<https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php>
(all earthquakes for the last 30 days)

If earthquakes make you nervous, equally large (or larger) data exists on:
<https://www2.census.gov/programs-surveys/popest/datasets/>
descriptions on: <https://www2.census.gov/>

Or weather data at:
<https://www.ncdc.noaa.gov/data-access/quick-links#loc-clim>

Create a SQL table, calculate time to create the table (and indexes).
Allow a user to specify on a web interface:

1. A number of random queries (up to 1000 queries of random tuples in the dataset)
2. A restricted set of queries, similar to previous (1.) but where selection is restricted (ie only occurring in CA, or within N<100 km of a specified lat,long location.
Or: a time range, or a magnitude range.
3. Measure time expended to perform these queries.
4. Show results.

Users of this service will interact with your performance service through web page interfaces, all processing and web service hosting is (of course) cloud based.

You will use some type of RDB SQL to store and retrieve earthquake information. and (of course) a friendly web UI.

You should (always) handle conditions such as: missing data (fields, attributes), and similar.

Task 2:

To improve performance (time) to do queries, buffer/cache some previously made query results in memory. This saves time and money (database actions cost.)

5. Then, installing and using either memcache or Redis repeat steps 1 through 4.

Please, submit in Canvas. Work must be individualized, but may be done in a group.

You must submit this lab, working (or partially) by the due date.

Your program should be well commented and documented, make sure the first few lines of your program contain your name, this course number, and the lab name and number.

Your comments should reflect your design and issues in your implementation.
Your design and implementation should address error conditions.