# GXDB Regression

## Scope

The intention of this document to elaborate GeoGraphix(GGX) schema regression requirement and resolution.

## Requirement Verify data set after importing spreadsheet and DefCon2? Verify data set after operations performed on database?

The above two questions answer is   
  
A system to ensure database invariants by comparing two GGX Sybase schema. e.g gxdb.db, gxdb\_production.db etc.

## Resolution

“GXDBRegression” is a bespoke GGX’s regression system baseline on python. It comprises comprehensive sets SQL scripts. It concurrently processes and executes SQL scripts on Sybase server. Later, it validates the generated schema with respect to schema provided for verification. It can support maximum 9 concurrent servers.

## Configuration

Regression configurations are defined in configuration.xml. These settings outline Sybase credentials (uid, pwd and dbf), requisite directories paths (Source, Expected Results and log) and performance parameters (semaphore, debug). The performance parameters are for internal debugging purposes (don’t change). The log setting should point to user accessible path so log file generates flawlessly.

Configuration file can be found at

GXDBRegression/Configuration/configuration.xml

## Workflow

GXDBRegression is run for gxdb.db and gxdb\_production.db. It starts up multiple Sybase servers(max =9) to run groups of tests in parallel. It process each SQL scripts (<Source> parameter) and feed to dbserver (<dbisql> parameter). Corresponding to each SQL test, Sybase generates XML formatted results set (<Result> parameter). The generated XML files are validated with respect to verified provided files.

## Running the Tests

The very first run of GXDB regression results in all test cases failures as there is no verified result set. User analyse and verifies first generated dataset. After verification, this result set may set as benchmark for subsequent run.

During the next run, the <Expected> parameter directs points to verified dataset. The regression system validates generated result set with benchmarked. Regression command is as follows

python.exe regression.py –c “configuration.xml”

## Metrics

### Coverage

Number of SQL Scripts = 92  
Coverage = All Well base views e.g. Survey, Survey Stations etc.

### Performance

**ChuckJ**92 passed out of 92  
Total elapsed time(secs) 61.64

**EagleFord**92 passed out of 92  
Total elapsed time(secs) 12078.87

### Solidity

Single maximum size file = 5.2GB  
EagleFord GXDB Schema Size = 20.6GB

## ER Model

GXDBRegression package contains a conceptual ER model. It’s modelled in MS Visio 2013.

## Unittest

GXDBRegression package contains framework unit tests.

## Install

Launch dos command pattern.  
Run (path-to-GXDBRegression)\regression\Scripts\activate.bat  
results in activation of regression environment, noticeable on command line.  
Run easy\_intall sqlparse(installs SqlParse module)  
python regression.py –c Configuration\configuration.xml

## Dependencies

Python 3.4.2 ([https://www.python.org](https://www.python.org/)), Licence (PSFL)  
SQLParse([https://github.com/andialbrecht/sqlparse),](https://github.com/andialbrecht/sqlparse),%20)  Licence (BSD2)  
Dbisql.com(Installed as part of GGX installation).