ETL Testing Framework

Goals

- fail early
- automated
- reproducible
- quantifiable coverage
- quantify data accuracy and completeness
- readible and accessable results
- simple useable api

Fail Early

“Fail” means:

- log and report warnings (ie non-breaking issues)
- kill a doomed run (ie don’t let broken runs continue)

“Early” means:

- early in the ETL runtime
  - *a run should fail as soon a breaking issue is detected*
- early in the data onboarding process
  - *invalid source data should be detected at upload time*
- early in pipeline development process
  - *pipeline bugs should be detected by pipeline team not the customer*

Automated

Testing should be triggered automatically

- when new data is recieved
- when pipeline code changes

Reproducible

Tests must be repeatable.

- results must include metadata that can be used to easily rerun (ie parameterize) the same test and get the same results.
    - date
    - path, hash, size of each input source file/table ingested
    - path, hash, size of each output table produced (before/after)

Quantifiable Coverage of Data (source data)

Coverage measures the proportion of data/code that is tested vs assumed correct.

- proportion of source files validated before they are ingested
  - check for existance of expected/required files before pipeline starts
  - row count
  - validate schema of source files
- proportion of source columns profiled before they are ingested
  - histogram
    - null/blank count
    - distinct count
  - type/format (eg YYYY-MM-DD vs MM/DD/YYYY; zero padding)
- proportion of source columns contraints checked before they are ingested
  - referential integrity: validate foreign keys
    - confirm expected 1..m and 1..1 relationships

Quantifiable Coverage of Data (output data)

Similarily, proportion of RDM tables/columns/constraints tests before published to client side.

Overall effectiveness of testing effort can be monitored (by management folk) by tracking number of customer reported bugs vs internally discovered bugs.

Quantifiable Coverage of Pipeline

The proportion of transformations that are tested

- confirm it was executed and data actually changed

Also, the proportion of customer found issues attributed to pipeline vs source data.

Quantify Data Completeness and Accuracy (Domain Specific)

- detection of incomplete data requires data testing coverage (see slide: Quantifiable Coverage of Data)

Checks Domain Specific

- testing data accuracy requires domain specific checks to be defined
    - relative comparisons
        - chronological checks (eg. order date <= ship date)
    - absolute checks (eg. item price can't be negative)

Quantify Data Completeness and Accuracy (Generic)

Generic Profile Regression Testing

- automically generate histogram for all source and RDM tables

something like:

```
SELECT COUNT(count_column_name)
FROM table_name
GROUP BY group_by_column_name;
```

column name	column type
test_date	date
table_name	string
count_column_name	string
group_by_column_name	string
count	int
... meta data	

Readible and Accessable Results

Readible

- test results should include a simple “at a glance” summary
- verbosity should be configurable
  - succeed quietly, fail loudly

Accessible

- emailed whenever ran
- detailed/drillable/queriable results should be available when debugging
- compatible with Stuko

Simple Useable Api

- easy to add new tests