**Data Quality APP**

Implementing Data Quality checks are essential part of designing Data Pipelines. Data Quality ensures dataset meets criteria for accuracy, completeness, validity, consistency, uniqueness, timeliness, and fitness for purpose, and it is critical to all data governance initiatives within an organization. This APP uses Python’s Great Expectation library for Data Quality Validation and Snowflake’s SnowPark API for data frames. Once APP gets onboarded into Snowflake account, Data Quality Checks can be implemented by calling stored procedures with relevant parameters. Below is the list of objects used in this framework. Installing APP on Snowflake account creates Tables, Sequences & Stored Procs required for this framework to function.

**-- Sequence List --**  
  
GE\_FUNCTION\_RULE\_MAP\_SEQ  
DQ\_VIOLATIONS\_SEQ  
DQ\_RULES\_SEQ  
PROCESS\_METADATA\_SEQ  
  
**-- Table List --**  
  
DATASETS  
ATTRIBUTES  
DQ\_RULES  
GE\_FUNCTION\_RULE\_MAP  
DQ\_VIOLATIONS  
SENSITIVE\_ATTRIBUTES  
PROCESS\_METADATA  
  
DATASETS - Stores Table level information.  
Attributes - Stores Column Level information.  
DQ\_RULES - Stores Data Quality Rules Information.  
GE\_FUNCTION\_RULE\_MAP - Great Expectation Function & Data Quality Rule Mapping information.  
DQ\_VIOLATIONS - Stores Failed Records in Json Format  
SENSITIVE\_ATTRIBUTES - Stores Attributes to be identified as PI & PII information.  
PROCESS\_METADATA - Stores Process Metadata information  
  
**--Stored Procs List--**  
  
create\_metadata(db VARCHAR, schema VARCHAR, tbl\_name VARCHAR);  
create\_dq\_rule(table\_name varchar,column\_name varchar,rule\_id varchar,expression varchar);  
create process\_metadata(db varchar,schema varchar,tbl\_name varchar);  
create\_metadata - > First Stored Proc called. This makes entries in Dataset and Attributes Tables.  
create\_dq\_rule -> Call Stored Proc with Data Quality Rules. Please refer GE\_FUNCTION\_RULE\_MAP for samples.  
  
**Example :**  
  
Call create\_metadata(db\_name,schema\_name,table\_name);  
Call dq\_rule(table\_name,column\_name,’expect columns to be null’,’NA’);  
  
**-- Populate Datasets and Attributes Table--**  
  
call create\_metadata('DEMO\_DB1','RAW\_SCHEMA','EMPLOYEE\_DEPT\_TABLE');  
  
**-- Populate DQRules Table--**  
  
call create\_dq\_rule('EMPLOYEE\_DEPT\_TABLE','DEPT\_ID','expect\_column\_values\_to\_be\_increasing','NA');  
call create\_dq\_rule('EMPLOYEE\_DEPT\_TABLE','DEPT','column\_values\_be\_unique','NA');  
call create\_dq\_rule('EMPLOYEE\_DEPT\_TABLE','NAME','column\_values\_be\_unique','NA');

**-- Run DQRules. Populates DQ\_Violations Table--**

call rundq\_rules('demo\_db1','raw\_schema','employee\_dept\_table');

**-- Populate Process Metadata Table--**

call collect\_processmetadata('demo\_db','test\_schema','SAMPLE\_STAGE\_TABLE');