Data Quality Framework

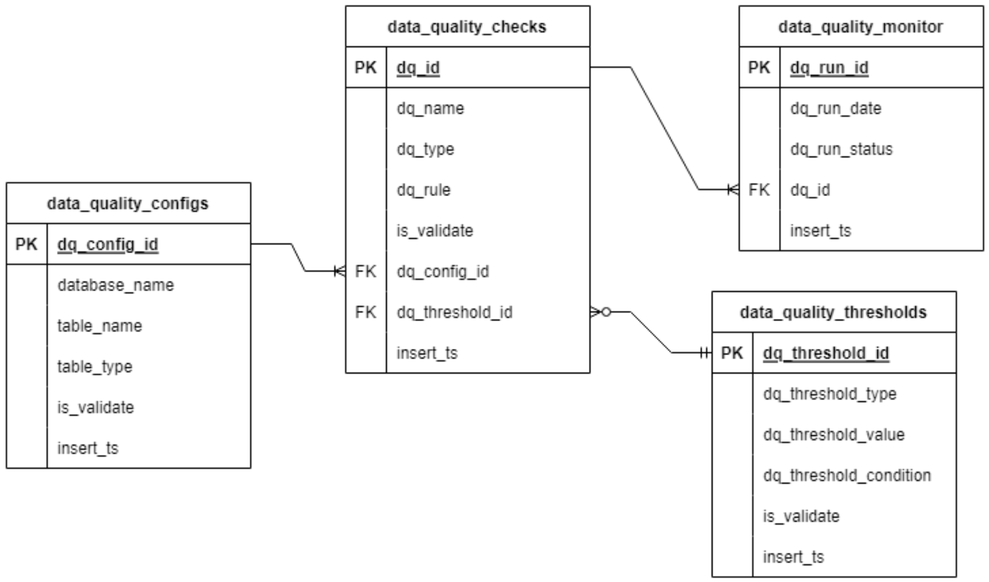
Data Quality Design

This page explains the data quality framework design and it's practices for Data Warehouse project.

Data Quality Entities

ERD Diagram

- the below models illustrates the data quality entities and it's relations



data_quality_configs

- this table holds all the list of delta tables that has to be applied with generic data qulaity tests
- there must be an entry in this config table in order to qualify for data quality
- the configuration entry must be set to {is_validate='Y'}
- each target table can have only single entry

· unique "{dq_config_id}" can be assigned for each target table

Column Name	Description
dq_config_id	unique *id for each entry
database_name	database/schema name
table_name	table name
table_type	table/view
is_validate	Yes (Y) or No (N)
insert_ts	row insert timestamp

data_quality_checks

- this table holds all the list of data quality checks entries for generic tests
- there must be an entry in this table in order to qualify for data quality generic tests
- each type of generic tests can have it's own entry
- the configuration entry must be set to {is_validate='Y'}

unique "{dq_id}" can be assigned for each target table

Column Name	**Description**
dq_id	unique *id for each entry

dq_name	data quality name
dq_type	data quality type
dq_rule	data quality rule
dq_config_id	id from data_quality_configs
dq_threshold_id	id from data_quality_thresholds
is_validate	Yes (Y) or No (N)
insert_ts	row insert timestamp

data_quality_monitor

- this table get inserted with the status of each data quality check that performed on target table
- this is transaction tables logs the data quality status
- this table get inserted for both FAIL and PASS data qulaity check tests

· unique "{dq_run_id}" can be assigned for each target table

Column Name	**Description**
dq_run_id	unique *id for each entry
dq_id	id from data_quality_checks
dq_run_status	data quality run status
dq_run_date	data quality run date
insert_ts	row insert timestamp

data_quality_thresholds

- this table holds all the list of data quality threshold entris for specific table or columns
- there must be an entry in this table in order to qualify for data quality thresholds
- data quality thresholds are optional
- the configuration entry must be set to {is_validate='Y'}

unique "{dq_id}" can be assigned for each target table

Column Name	**Description**
dq_threshold_id	unique *id for each entry
dq_threshold_type	threshold type
dq_threshold_value	threshold value
dq_threshold_condition	threshold condition
is_validate	Yes (Y) or No (N)
insert_ts	row insert timestamp

Generic Data Quality Checks

RefTableConfig_test

- this generic test is used to check the target table configuration entries in "data_quality_configs" table
- if the configuration entries exist here then only it move forward with generic tests

NaturalKey_test

- this is generic test is to check the natural key uniqueness
- this test retrive the configurations from "data_quality_configs" and "data_quality_checks"
- it retrive the natural key columns from configurations and check the uniqueness
- the test FAILS if the expected configurations are missing

NullOrEmptyStringColumns_test

- this is generic test is to check the null or empty string in any of columns
  - it is only limited to test if all the values of specific colums contains NULL or EMPTY STRING
- this test retrive the configurations from "data_quality_configs" and "data_quality_checks"
- if column list exists in configs - it retrive the list of columns from configurations and check the uniqueness
- if column list exists in configs - it retrive the columns from databricks catalog for that table

EmptyString_test

- this is generic test is to check the empty string in any of columns
- it tests if specific colums contains EMPTY STRING
- atleast one value has empty strings to get qualified for this test
- this test retrive the configurations from "data_quality_configs" and "data_quality_checks"
- if column list exists in configs - it retrive the list of columns from configurations and check the uniqueness
- if column list exists in configs - it retrive the columns from databricks catalog for that table

LeadingTrailingSpaces_test

- this is generic test is to check the white spaces in string in any of columns on any side (leading/trailing)
- this test retrive the configurations from "data_quality_configs" and "data_quality_checks"
- if column list exists in configs - it retrive the list of columns from configurations and check the uniqueness
- if column list exists in configs - it retrive the columns from databricks catalog for that table

RowCount_test

- this is generic test is to check the row count of table
- this test retrive the configurations from "data_quality_configs" and "data_quality_checks"
- if table is empty then it fails the test

AuditColumns_test

- this is generic test is to check audit columns exist in target table
- this test retrive the configurations from "data_quality_configs" and "data_quality_checks"
- if any of the audit column is missing in target table then it fails the test

Data Quality Notebook

Data Quality framework is a dynamic databricks notebook which runs all the configured generic tests on a selected delta table.

Key	**Value**
notebook_name	data_quality_framework
notebook_location	Bitbucket location
notebook_description	this framework notebook used to run a generic data quality checks on a given table
notebook_use	1. have the below steps as part of a workflow task after the load notebook task (or) 2. add below command to the load notebook at end of it in a separate cell o notebook_info = DataQualityNotebookTesting(DatabaseName = DatabaseName, TableName = TableName, InsertQuery = Query) o RunDataQualityNotebookTests(notebook_info)

Data Quality Tables DDL

```
CREATE TABLE data_quality_configs (
  dq_config_id INT,
  database_name STRING,
  table_name STRING,
  table_type STRING,
  is_validate STRING,
  insert_ts TIMESTAMP
  );
```

```
insert into data_quality_configs values (1, 'db_name', 'table_name', 'table', 'Y', current_timestamp());
```

```
CREATE TABLE data_quality_checks (
  dq_id INT,
  dq_config_id INT,
  dq_name STRING,
  dq_type STRING,
  dq_rule STRING,
  dq_threshold_id INT,
  is_validate STRING,
  insert_ts TIMESTAMP
  );
```

```
insert into data_quality_checks values (1, 1, 'natural_key_uniqueness', 'NaturalKey_test', 'col1, col2', NULL, 'Y', current_timestamp());
insert into data_quality_checks values (1, 1, 'null_or_empty_strings', 'NullOrEmptyStringColumns_test', 'all_columns', NULL, 'Y',
current_timestamp());
insert into data_quality_checks values (1, 1, 'empty_strings', 'EmptyString_test', 'all_columns', NULL, 'Y', current_timestamp());
insert into data_quality_checks values (1, 1, 'leading_trailing_spaces', 'LeadingTrailingSpaces_test', 'all_columns', NULL, 'Y', current_timestamp());
insert into data_quality_checks values (1, 1, 'table_row_count', 'RowCount_test', 'all_columns', NULL, 'Y', current_timestamp());
insert into data_quality_checks values (1, 1, 'audit_columns_count', 'AuditColumns_test', 'audit_col1, audit_col2', NULL, 'Y', current_timestamp());
```