

QC Master procedure

Created by Mike Timchenko

Last updated: Jan 07, 2022 by Lakshman Kammili • 3 min read •  12 people viewed

In case a lot of tests there is an opportunity to save all tests parameters in rule-based tables and run the master procedure as a set of tests.

Clusters: **<Cluster Name Generic>**

Schema: **QC**

Procedure name: **proc_qc_master**

Parameters:

| Parameter name | Datatype | Description |
|----------------|--------------|--------------------------|
| p_test_group | varchar(200) | Name of a group of tests |
| p_test_time | timestamp | {{now}} |
| p_order_run | varchar(50) | {{CurrentOrderRunId}} |

Rule-Based Tables

primary_key_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|--|--|
| test_group | varchar(200) | schema_name | varchar(100) | | |
| test_description | varchar(1000) | table_name | varchar(100) | | |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

uniqueness_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|--|--|
| test_group | varchar(200) | schema_name | varchar(100) | | |
| test_description | varchar(1000) | table_name | varchar(100) | | |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

not_null_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|--|--|
| test_group | varchar(200) | schema_name | varchar(100) | | |
| test_description | varchar(1000) | table_name | varchar(100) | | |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

allowed_increment_test_set

| | | | | | |
|------------------|---------------|-------------|--------------|--------------------|-----|
| test_group | varchar(200) | schema_name | varchar(100) | delta_record_count | int |
| test_description | varchar(1000) | table_name | varchar(100) | | |
| test_action | varchar(100) | | | | |

data_match_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|--------------------|--------------|
| test_group | varchar(200) | schema_name | varchar(100) | match_schema_name | varchar(100) |
| test_description | varchar(1000) | table_name | varchar(100) | match_table_name | varchar(100) |
| test_action | varchar(100) | column_names | varchar(200) | match_column_names | varchar(200) |
| | | skip_errors | int | mode | varchar(20) |

prior_match_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|--------------|--------------|
| test_group | varchar(200) | schema_name | varchar(100) | schema_prior | varchar(100) |
| test_description | varchar(1000) | table_name | varchar(100) | mode | varchar(20) |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

value_match_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|-----------|--------------|
| test_group | varchar(200) | schema_name | varchar(100) | value_set | varchar(500) |
| test_description | varchar(1000) | table_name | varchar(100) | mode | varchar(20) |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

window_match_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|-------------|--------------|
| test_group | varchar(200) | schema_name | varchar(100) | date_column | varchar(100) |
| test_description | varchar(1000) | table_name | varchar(100) | window_days | int |
| test_action | varchar(100) | column_names | varchar(200) | mode | varchar(20) |
| | | skip_errors | int | | |

condition_check_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|-----------|---------------|
| test_group | varchar(200) | schema_name | varchar(100) | condition | varchar(1000) |
| test_description | varchar(1000) | table_name | varchar(100) | | |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

aggregate_match_test_set

| | | | | | |
|------------------|---------------|------------------|--------------|------------------------|--------------|
| test_group | varchar(200) | schema_name | varchar(100) | match_schema_name | varchar(100) |
| test_description | varchar(1000) | table_name | varchar(100) | match_table_name | varchar(100) |
| test_action | varchar(100) | column_names | varchar(200) | match_column_names | varchar(200) |
| | | skip_errors | int | match_groupby_names | varchar(200) |
| | | groupby_names | varchar(200) | match_having_condition | varchar(500) |
| | | having_condition | varchar(500) | mode | varchar(20) |

custom_query_test_set

| | | | | | |
|------------------|---------------|--------------|--------------|-------|---------------|
| test_group | varchar(200) | schema_name | varchar(100) | query | varchar(4000) |
| test_description | varchar(1000) | table_name | varchar(100) | | |
| test_action | varchar(100) | column_names | varchar(200) | | |
| | | skip_errors | int | | |

Example of usage

1. Create a new step to call master qc procedures in a new or existing node. The example of the way to call procedures:

```
call qc.proc_qc_master('car_t_auto_qc','{{now}}','{{CurrentOrderRunId}}');
```

2. Create a step to get all result messages with result code = -1 and save it into a jinja variable

```
select listagg(test_type||': '||result_message,' ') within group (order by starttime)
from qc.test_results
where order_run = '{{CurrentOrderRunId}}'
and result_code = -1;
```

Also, value of a ninja-variable now and name of a group of test could be used to identify recent test results.

3. Create a step to get all result messages with result code = 0 and save it into a jinja variable
4. Create tests to check rows with result_code = 0 or -1

—

Result of usage

1. Run a variation and open results. It contains messages about all errors and warnings.

—

2. Use DataGrip to access results and examples of wrong data in the table

QC.TEST_RESULTS

```
1 select replace(listagg(test_type||'/n'||result_message||'/n'||result_error_data||'/n',
2     within group (order by null), '/n', chr(10))
3 from qc.test_results
4 where order_run = '93bb0d70-93ee-11eb-a0e6-5e8110fc3037'
5 and result_code = 0
```



⌋

 Like Be the first to like this

No labels 