Quality Gates™

3.3.5

User guide
# Table of Content

1 - Introduction .................................................................................................................. 4

2 - Navigation ..................................................................................................................... 5
   2.1 Navigation tool bar ...................................................................................................... 5
   2.2 Navigation tree ........................................................................................................... 5
   2.3 Folder Tree ................................................................................................................ 6
   2.4 Test history ................................................................................................................. 7

3 - Dashboard ..................................................................................................................... 9
   3.1 Overview .................................................................................................................... 9
   3.2 Dashboard components ............................................................................................. 9
   3.3 Old Dashboard overview .......................................................................................... 12

4 - Settings ....................................................................................................................... 13
   4.1 Connection setup ....................................................................................................... 13
   4.2 Connection naming convention ................................................................................. 15
   4.3 SQL Parameter ......................................................................................................... 15

5 - Defining a test ................................................................................................................ 18
   5.1 Create a new test ...................................................................................................... 18
   5.2 Test tabs .................................................................................................................... 19
      5.2.1 - DataSet tab – ................................................................................................. 19
      5.2.2 - Second tab – ................................................................................................. 21
      5.2.3 - Settings tab - ............................................................................................... 21
      5.2.4 - Preview & Execute Result tab - ................................................................. 24
5.2.5 - Documentation tab ................................................................. 25
5.2.6 – CMD Parameter ................................................................. 25
6 - Source to target tests........................................................................ 27
  6.1 When to use ............................................................................... 27
  6.2 Check Sum - tabular comparison.................................................... 27
  6.3 Check Sum - OLAP comparison...................................................... 29
  6.4 Compare table ........................................................................... 30
7 – Data validation.................................................................................. 33
  7.1 When to use ............................................................................... 33
  7.2 Data Validation ........................................................................... 33
  7.3 KPI Validation ............................................................................ 35
8 – Other tests ..................................................................................... 37
  8.1 Duplicate Key ............................................................................ 37
  8.2 Custom SQL .............................................................................. 38
9 – Referential Integrity ....................................................................... 39
  9.1 When to use ............................................................................... 39
  9.2 Referential Integrity .................................................................... 39
  9.3 Referential Integrity with complete .............................................. 40
10 – Execution Flow ............................................................................. 42
  10.1 When to use ............................................................................ 42
  10.2 Execution flow .......................................................................... 42
1 - Introduction

The purpose of this guide is to give the beginner and advanced BI developer the training needed to work and perform tests with Quality Gates™.

The document explains how to work and set up tests with Quality Gates tool.

If you have any question or a problem, you are welcome to contact one of our Quality Gates experts or open a ticket in the Quality Gates support site: [http://www.quality-gates.com/support/](http://www.quality-gates.com/support/)
2 - Navigation

2.1 Navigation tool bar
The navigation tool bar enables the user to navigate between the Quality Gates working area and the administration settings. A user will be able to see the navigation tool bar only if he has administrator capabilities.

On the top right the name of the user who logged in will appear.

2.2 Navigation tree
The navigation tree enables to organize all tests in a logic tree which is sorted according to business models, loading procedures and target tables. In the navigation tree there are three basic functions (expand, open, option):
• Click “Expand” – little black triangle
  o Folder: Opens the folder to its sub folders or tests
  o Test: Opens the test history tab
• Click “Open” – name of the folder/test
  o Opens the folder/test on the dashboard
• Click “Options” – little arrow button
  o “New” – Open new folder, test or sub folder
  o “Rename” – Gives a new name to a folder or test
  o “Delete” – Terminates folder or test
  o “Move” – Moves folder(and all of its tests) or test to a different tab in the navigation tree
  o “Copy” – Duplicates tests only. Not available for folder.
  o “Commit” – deploy a test to a deployment environment, such as prod or test.
  o “Move up”/ “Move down” – Move the test or folder up or down in the order of display.

2.3 Folder Tree
The tree object is highly flexible; however it is very important to define the tree structure in a good way. Issues that need to be taken into consideration:

1. Folder execution – it is possible to execute all folder tests
   a. Executing at the end of the DWH load?
   b. If users want to execute at the end of Staging – Define as a folders.
2. Dashboard – The dashboard overview is by folder
3. Users security - Security is by folders – Each user can have one parent folder

The folders are usually divided by layers or users. Below there are 2 examples of folder distribution:

<table>
<thead>
<tr>
<th>Tree structure</th>
<th>Sample</th>
</tr>
</thead>
</table>

2.4 Test history

The "Test History" feature enables the user to get details about past execution. When clicking on test history it opens as a mini dashboard containing number of runs, failures and average run time. In the bottom of the screen you will see a summary table with all of the relevant information: when it was executed, environment, duration, result and if an alert was sent.
3 - Dashboard

3.1 Overview

The dashboard allows a management, developer or business user to overview of the status of the system.
From Version 3.3.3 the default dashboard has changed to allow better control for the users.

3.2 Dashboard components

The dashboard contains 5 sections: Folder Overview, Folder Failure Trends, Folder Tests Details, System Health and Test History.

1 – Folder Overview

All the folders under the selected folder are displayed. With details about number of tests, number of executions, number of successes, number of failed tests and a general indicator.

The indicator has 3 colors:

- Green – If all tests under the same folder have successfully passed
- Yellow – If at least one test failed, but none of the test that failed is high severity
- Red – If at least one high severity test has failed

Clicking on the sub folders in this section will change the data displayed in the other sections.

2 – Folder Failure Trends

This section shows the percentage of failure for each folder over time.

- The folders and sub folder displayed in this section will be changed according to the folder selected in the folder overview section
- The default time line is week but it can be changed to day or month by selecting the period on the top bar.

- It is possible to hide some folders line by clicking on the line in the legend. Another click on the line will bring it back.

3 – Folder Tests Details
The folder tests details section shows all the tests that are available under a specific folder. By changing the folder selected in the folder overview, the tests on the bottom will be displayed.

<table>
<thead>
<tr>
<th>Edit</th>
<th>Test Name</th>
<th>Objectkey</th>
<th>Executions</th>
<th>Failed</th>
<th>Success</th>
<th>Fail With High Severity</th>
<th>Health Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sales by Country</td>
<td>25673</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales Detailed Comparison</td>
<td>25675</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- Click on the Edit pencil will open the test to review or change the definition.
- Clicking on a specific test (anywhere besides the pencil) will show the history of this test in the test history section.

4 – System Health

This pie gives a percentage for success, fail, and fails with high severity tests. The numbers reflect the folder selected in the Folder overview section.

5 – Test History

The test history section shows all the executions done for a specific test. It shows: Source A and B duration (in sec), number of rows returned from source A and B, and whether the test failed or not.

Filter Area – top bar

On the top of the dashboard there is a selection bar that allows better defining the parameters:

- Last Execution – if this checkbox is checked, the dashboard will show only the last execution results. Otherwise it takes all history into consideration.
- From Date to Date – defines the start and end date for the dashboard. By default it shows last month.

- Period – what level of period will be displayed in the trend section (Month, Week, Day). By default it is week.

- Refresh dashboard – Click this button to refresh the dashboard details. Attention! The dashboard is not being refreshed automatically.

- Execute – Execute all the tests in the folder of the dashboard.

### 3.3 Old Dashboard overview

In order to change back to the old dashboard view use the button on the bottom right to go back to old dashboard view

This dashboard presents the numbers of tests available in the system, number of executed tests, and failed tests. It is possible to drill to a specific test directly from the folder and view detailed results. The dashboard enables you to filter information by choosing environments, user or time period.
4 - Settings

4.1 Connection setup

How to define a new connection –

1. Expand the settings tree
2. Expand the connections tree
3. Go to "Options" and choose “New”

4. Choose one of the following:
   a. OLEDB connection for relational DB.
   b. SSAS.Catalog for MSAS OLAP connection.
   c. Connection Folder for managing the connection in the connection navigation tree.
* For setting up file connection check see the admin guide.
5. Choose a name for the connection.

![Connection Setup](image)

6. Click “OK” button
7. The connection will be open on the main window on the right.

![Database Configuration](image)

8. Choose one of the DB type.
9. Insert the Data Source: server name
10. In the initial catalog type: provide the DB name
11. Enter the user name and password
12. Press test connection – you should now see a success confirmation

13. Advanced: Use the advanced section to paste a connection string. By modifying the connection template, the parameters chosen above will not affect the connection string pasted.

**Advanced parameters –**

At the bottom of the connection setup there are 2 check boxes:

- [ ] Many Tables in DB? (Will load the tables only after filter)
- [ ] Load to cache? [Load cache]

1. Many tables in DB? – Check this check box when there are many tables in DB, and there is need to narrow the search for a small set of tables. The search itself is done in the data set definition in the test.
2. Load to cache? – In case the time to fetch the tables from the DB takes too long, there is an option to load the structure of the table into QG. This saves time in fetching the tables, but needs to be refreshed once in a while.
4.2 Connection naming convention

Consider the following when deciding on a new connection name:

1. Recommend `<SourceSystem>_<DatabaseName>`
2. Do not use server name – It will be different in the prod

4.3 SQL Parameter

SQL parameter enables the user to create generic parameters in different data source and reuse these parameters in different tests.

How to define a new sql parameter –

1. Expand the settings tree
2. Expand the parameters folder tree
3. Go to "Options" and choose “New”

4. Choose the Execution Parameter
5. Choose a name for the execution parameter.

6. Click “OK” button
7. The parameter window will be open on the main window on the right.

Format of the parameter window:

Example Param  Execution Parameter  #26207

Parameter Name: Example Param
Connection: 
Query: 

Default Value: [Get Default]
When Result Empty: when the parameter result is empty, the query might not be valid

Parameter Name - Specifies the name of the parameter
Connection - the data source from which a parameter will be selected
Query - the query you want to make. Make sure it is written in the DB language you chose
Default Value - displays the default result from the query
When Result Empty - in case the query result is null or empty, this value will replace it
An Example of the parameter window (Query Returns the current date in last month):

**sql last month**  Execution Parameter  #26125

<table>
<thead>
<tr>
<th>Parameter Name:</th>
<th>sql last month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection:</td>
<td>Adventure works</td>
</tr>
<tr>
<td>Query:</td>
<td>SELECT DATEADD(MONTH, -1, GETDATE())</td>
</tr>
<tr>
<td>Default Value:</td>
<td>2013-03-14 18:53:07</td>
</tr>
<tr>
<td>When Result Empty:</td>
<td>1</td>
</tr>
</tbody>
</table>
5 - Defining a test
In this section the basic rules to define a test will be demonstrated.

5.1 Create a new test

In order to define a new test right click on one of the folder or click the option arrow. Choose “new” and a popup window will appear.

The description on the right will change based on the test chosen. The description will assist in choosing the most appropriate test for your need.

Choose a meaningful name for the test and click the OK button.
5.2 Test tabs

Depending on the type of test and data source chosen, 4 or 5 tabs will be available.

5.2.1 - DataSet tab –
This tab is the first one that opens immediately after the user creates a test. Here the user can select the DB tables and filter them in order to suit the purpose of the test.

Selecting data source - Clicking on the arrow button in the right side of theDatasource will open a list of all of the Data sources. In case there are Data Source A and B, it is recommended to put data source A as the target and data source B as the source.

Selecting database tables- There are two options, either selecting a table or creating a free command:

- Table/view – Clicking on the arrow button in the right side of the table line will open a list of all of the tables/views. Select the table you want. It is also possible to define a “where” clause. Click on the “columns” to view the list of possible columns in the table.
- SQL Query – check the free command, and paste the free command.

Selecting CSV file
First a connection need to be set up to load file (additional information is available in the admin guide)
Follow the following steps:

1. Choose a connection which supports CSV\txt file upload.
2. Click on the "Upload New File" button, afterward the "File Upload" screen will be opened.
3. Select the file which will be uploaded.
4. Enter a friendly name or use an existing friendly name (which is derived from the template).
5. Choose the type of the delimiter, which exist in the selected file.
6. Choose the code of the file (the encoding).
7. Choose specification for the file upload:
   - Header in first row- The first row in the file will become the column name
   - Load all files in template- When QG tests will be executed, all the data in the files (according to the specification in the template) will be unified into a single file
   - Get only statistics- Only number of rows in the file will be loaded to Quality Gates.

**Selecting Excel file**

First a connection need to be set up to load file (additional information is available in the admin guide)

Follow the following steps:

1. Choose a connection which supports Excel file upload.
2. Click on the "Upload New File" button, afterward the "File Upload" screen will be opened.
3. Select the excel file which will be uploaded.
4. Enter a friendly name or use an existing friendly name (which is derived from the template).
5. Will upload the excel file and all the sheets.
6. Select a sheet which will be uploaded.
7. Will upload the sheet to a table in the DB.

**Generating Preview**

Clicking on the “Preview!” button pops up a new window where you have to click “Execute” in order to see a preview of the data that was requested.

5.2.2 - Second tab –
This tab differs between type of test and source of data. Explanation about this tab will be given in each type of test. Normally this tab is the place to define column level mapping.
In some cases this tab will not be available.

5.2.3 - Settings tab -
In this tab it is possible to set Mail alerts and Failure Options:
Failure Options

Use the following to define the threshold and failing behavior:

*Fail test when different value ratio is:* Choose Greater, Equal or below the value written on the right to fail the test.

*Then:* Choose the number or the percent of the test to fail on.

*Fix Percent:* Decided whether to use Percent or Fix number for comparison.

*On failure:*

In this section decide what to do when the test fails:

- Alert and continue
- Alert and stop execution

This is important to be used usually when the test is part of an ETL process and we would like the ETL to fail or to continue on failure. The test will return 1 on success or on failure if the settings are set to Alert and continue, and will return 0 if the settings are set to Alert and stop execution.
Log tables

The log table option allows writing the result into a log table for future use.

**Data source**: Choose the data source of the DB to write to

**Table name**: Define the table name to write the log. The default naming is: `QG_Log_{TestName}` which is a name based on the test name. It is possible to create a specific name by replacing this name to a name required.

**Log Pivot**: Will create an additional log table with the data pivoted (only in Check sum).

**Log table**: Check this box to activate the log writing.

**Save Log History**: Will enable the log to save multiple results of log in the same execution table.

Notice! The log structure varies from test to test.

Mail Alerts

In this section, it is possible to enter a mailing list for success or failure of the test. Make sure to check the two boxes at the bottom, to ensure that the recipient receives both the data result and the test queries.

Result preview

It is possible to decide what will be the results displayed in the result execution tab, based on the checkbox available in the settings. By default all the check boxes are marked, meaning that all the results will be displayed.

The result preview will be different from test to test depending on the relevance.

**Show different results** – Will show the keys or columns that have different results between the sources.

**Show exist in a/b and not b/a** – Will show keys which exist only in one source and not in the other.

**Show Same** – Will display the status equal (identical results) and pass ok (different results, but did not pass the threshold).
5.2.4 - Preview & Execute Result tab -

In this tab you can set the execution properties:

- **Top** – How many rows will be displayed
- **Preview Execute** – Executes the test without sending mail alerts and without keeping log documentation.
- **Stop** – Stops the execution.
- **Data** – Presents the result of the test.
- **SQL** – Presents the SQL query that was executed
- **Execute** – Executes the test and sends mail alerts while keeping log documentation.
- **Env** – Presents the environment where the test was executed.

Additional information:

- Time and Duration of last run.
- Number of rows and columns returned.
- Test results.

![Check Sum Training Check Sum #22881](image)

<table>
<thead>
<tr>
<th>Compare Result</th>
<th>Source</th>
<th>PromotionKey</th>
<th>SUM_OrderQty</th>
<th>SUM_SalesAmount</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Different</td>
<td>Total</td>
<td>1</td>
<td>(0 %) - 8</td>
<td>(0 %) - 4 884887070...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Datasour... 1</td>
<td>25589</td>
<td>13875617 9165227</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Datasour... 1</td>
<td>25579</td>
<td>13875617 9165231</td>
</tr>
<tr>
<td>▶ Different</td>
<td>Total</td>
<td>2</td>
<td>1018</td>
<td>(0 %) - 1 979090471...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Datasour... 2</td>
<td>1018</td>
<td>9740589305469398</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Datasour... 2</td>
<td>1018</td>
<td>9740589305469398</td>
</tr>
<tr>
<td>▶ Equal</td>
<td>Total</td>
<td>13</td>
<td>B</td>
<td>5838.8</td>
</tr>
<tr>
<td>▶ Equal</td>
<td>Total</td>
<td>14</td>
<td>4</td>
<td>9536.28</td>
</tr>
</tbody>
</table>

**The test failed!**

Number of return rows is 3.
This test defined to fail when return rows (greater than 0). The compare ratio is set to 0.001 to exit tab for more definitions.
5.2.5 – Documentation tab -

In this tab you can write comments regarding the test that can be seen by any other user.

5.2.6 – CMD Parameter -

CMD Parameter enables the user to create generic templates for all tests in Quality Gates and allows passing parameters form external executions.

The CMD Parameter will appear in the top section of the test screen. The parameters are displayed by clicking on the check box, as shown in this example:
The CMD parameter will be used in the free command section of the test, as shown in the following example:

CMD parameters can be used from:

- Single test execution
- Folder execution
- Execution flow (when editing a test)
6 - Source to target tests

6.1 When to use
The sources to target type of tests include the following: Check Sum and Compare Table. Those types of tests allow comparing 2 different sources.

Use check sum to compare aggregation of data: Number of rows, Amount of sales, compares OLAP to Tabular data and compares data over time.

Use Compare table for detailed comparison: Test that all fields in one table have the same values, during a migration project. Compare table is replacing the traditional Row by Row test. It has better performance and smarter indicator of the exact value that is problematic.

6.2 Check Sum - tabular comparison

1 - Data Set tab –
Choose the 2 tables or queries to compare.

2 - Check Sum tab –
On the left and right it is possible to see the metadata of sources A and B.

Drag the values to the lower window to compare and choose the aggregation function (Sum, Count, Max, etc.). Coordinate the values (columns) from source A to values from source B.

In the Group by window drag the keys of the results break down. In this example the country is the breakdown key and will show whether the values are equal or different based on the country.
4 – Preview & Execute Results tab

The results include the following columns:

1. Compare results
   Displays the result for the comparison. Can be one of the following:
   a. Equal – Both source and target is equal.
   b. Different – There is a gap between source A and B that is above the threshold.
   c. Exists in A and not in B – The Break down key does not exist in data source B.
   d. Exists in B and not in A – The Break down key does not exist in data source A.
   e. OK – There is a gap between source A and B, but did not exceed the threshold.

2. Source
   Can be one of the following options:
   a. Total – Gap between the values if exist or the aggregation value if equal.
   b. Datasource A/B – The aggregation value for each data source.

3. Break down keys (Group by)
   can be 0 or more.

4. Aggregation columns
   can be 1 or more. Indicate the aggregation function and column name.
### 6.3 Check Sum - OLAP comparison

**1 - Data Set tab**

In order to create an OLAP comparison, either to compare to a tabular data or to another OLAP source, first choose an OLAP Datasource.

Immediately when choosing an OLAP data source there will be 2 changes:

1. The Check Sum tab will disappear.
2. The text *Number of Joins* will appear allowing to choose number of columns to join.
MDX guidelines: Make sure the measures are on 0 and keys are on 1.

SQL guidelines: Make keys appear first and then the measures.

Number of joins: Number of joins should be 0 or more based on the number of keys that are available.

All keys/ Members and measures appear in the MDX and SQL will take part in the comparison.

**6.4 Compare table**

This test is intended to compare the data in two different tables based upon a certain key. The test enables you to define which columns to compare and what is the primary key.

1 - **Data Set tab**

Choose the 2 tables or queries to compare.

2 – **Define Relation**
When opening this tab the columns of data set A will appear.

1. In case Data source A and B have the same column names, it is possible to use the “Auto Complete Columns”.

2. If the names are not equal, go to source B and choose the coordinate column.

3. Check the most left check box next to each column for the columns that need to be compared.

4. Check the check box next to the Primary Key column.

5. Click on define ratio setting by clicking on the relevant column to define the ratio:

   - **Compare Success When Difference**
   - **Type:** Not Between (check greater and smaller)
   - **Fixed:** Fixed
   - **Ratio Value:** 0.95
   - **Enabled:**

   In this line you can choose the condition of comparison (greater, lower, between, not between) of the difference between A and B. You can also choose the form of comparison: fixed - numbers, percent - percentage. Do not forget to check the Enabled and then you will be able to see that the “ratio setting "button turned green."

Example of usage:
To allow gaps on sales columns of more or less 1000 use:
- **Type:** Not Between (check greater and smaller)
- **Fixed:** Fixed – check a fixed number and not percentages
- **Ratio Value:** 1000
- **Enable:** V

6. Use the Function to wrap the value of a specific column. Let us consider this case: column source named *client* which contains null values and a target column *client_ID* that handles the null values by replacing it with the value -1. In that case we do not want to see an alert for this gap.
We can add in the function next to the source column client the following expression: `isnull (client, -1)`.

7. Use the settings on the bottom to improve performance:
   a. Stop After X Sample Rows – The default value is 100,000. It is possible to modify it according to the requirements.
   b. Execute on Server (Same DS or LinkServer) – If both tests are on the same DB it is allowed to make the comparison on the same server instead of loading the data to QG and compare it there.
   c. Upload only different rows to memory – This option will reflect on what is saved to the repository.

4 – Preview & Execute Result –

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Scanned</th>
<th>Pass</th>
<th>Failed</th>
<th>Equivalence Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keys In A And Not B</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Keys In B And Not A</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>80%</td>
</tr>
<tr>
<td>Client_ID</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Sale_Date</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>50%</td>
</tr>
<tr>
<td>Keys</td>
<td>DS_A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data A Id</td>
<td>01/01/1901 00:00:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data B Id</td>
<td>01/01/2011 12:00:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sale_Amount</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>75%</td>
</tr>
</tbody>
</table>

The test failed! Number of return rows is 4.
The test defined to fail when return rows [Greater Than 0]. The compare ratio is set to 99.999%. (Go to Settings tab for more definitions.)

1. The results inform us for column we decided to compare and also for: Keys in A and not in B, Keys in B and not in A.
2. Columns marked in red means there are gaps.
3. By expanding the column A it is possible to see the Keys that have differences and both values exist in data sets A and B.
7 – Data validation

7.1 When to use

In order to validate what values exist inside the DB, we would like to define some rules and check if the values are valid. There are 2 levels of validation: row level and aggregate level.

Use Data Validation to check at the row level the possible range of value, format and relation between different columns.

Use KPI Validation to check the aggregate level result and define possible range of values.

7.2 Data Validation

1 - Data Set tab

Choose 1 specific table to validate the values inside.

2 – Table values tab

1. Tick one or more checkbox(es) for columns that define the key.
2. Click on a column to define a validation rule. On the top right it is possible to see the column name that was chosen.
3. Click on Add Rule icon to add a new rule. A column can have more than one rule defined.
4. Click on the green square to edit the rule properties. A pop up window will appear.
5. Fill the Rule Name and Rule Description.
6. Click on Add Constraint and choose the column to make the constraint. It is possible to add several constraints on several columns and this way check the relation between 2 columns.

7. Choose the operator under column “Is” (Greater, Lower, Between, Not Between, Equal, Not Equal, Is Null, Is Date, Is Numeric). The purpose of this rule is to define when the test fails.

8. Click Validate button to make sure the test is valid and then click save.

---

4 – Preview & Execute Result

The results open in a collapse and expand view.

It includes the following columns:

1. Column Name
   presents all the columns that have rules.
   By expanding the column that has a problem, it is possible to click on the rule name and see the exact values that have failed.

2. Pass
   In collapse mode shows the number of rules that pass.
   In expand mode shows the number of rows that pass.

3. Failed
   In collapse mode shows the number of rules that failed.
   In expand mode shows the number of rows that failed.

4. Equivalence Percent
   In collapse mode shows the percentage of rules that failed.
   In expand mode shows the percentage of rows that failed.
7.3 KPI Validation

1 - Data Set tab

Choose 1 specific table or write a query/MDX for aggregated or row data.

2 –Define aggregation tab

In case the data set is row data:
- Drag the aggregation level to upper box ‘Group By’.
- Drag the measures to the bottom box and select the aggregation function.

In case the data is already aggregated:
- Check the ‘Query as a source’ check box.
- Drag the aggregation level to upper box ‘Key’
- Drag the measures to the bottom box.

For MDX this tab does not exist.

Pay attention! Since the rules are defined for a key and a measure, once the aggregation is defined, it cannot be changed.

4 – Preview & Execute Result

As opposed to other tests, most of the action is done in this tab.

Follow those steps:

1. Click Execute to get the query/aggregation results.
2. The results will appear with a pencil next to each measure. If there is a pencil next to an attribute, go back to the data set tab, and define correctly the number of attributes.

3. Click on the pencil to define the range for the different indicators behavior.

4. Fill the values in the following way:
   
   a. Respect the following order of values:
      Min Value Warning << Min Value OK << Max Value OK << Max Value Warning.
   
   b. The values can be defined for either sides or only one side: Min or Max or Min & Max.
   
   c. The values can be defined for OK & Warning or OK only.
   
   d. At least one of the OK is required in order to have an indicator.

5. 2 types of check appear:

   a. When key is missing: In case one of the keys attribute (in the example above: country name) with rule definition does not appear. It is possible to make the whole KPI test fail or pass. In case the checkbox is on ‘Fail’ and the key is missing, the test will fail.

   b. The check box ‘When value is null or empty’ – when the checkbox ‘Fail’ is checked, it will show the indicator ✗ in case of empty or null value.

6. By click on the Save button, one of the following icons will appear: 

36
7. Click on the ‘All rules’ button shows all the rules.
   a. Each row presents key range definition for a specific measure. In the example above the key is country BRZ and column is Sale Amount and the Rule defined is in the Rule column.
   b. The message column and the color of the row provide information about the indicator.
   c. By click on the X button on the right it is possible to remove a rule.

8 – Other tests

8.1 Duplicate Key

This test is intended to identify multiple listings based on logic key (one or more).

1 - Data Set tab

Choose 1 specific table to validate the uniqueness of logic key inside.

2 – Business Key tab

Select the business key by checking the check box on the left next to the relevant column.

4 – Preview& Execute Result

The results include the following columns:

1. All the columns that define the logic key.
2. Cnt – the number of duplicate keys.
8.2 Custom SQL

This test allows creating a custom test according to specific user needs. All the logic of the test will be defined in a free command. The test will fail depending on presence or absence of values.

1 - Data Set tab

Choose 1 specific table or create a free command with some logic that returns rows or not.

2 – Define Relation tab

The columns of the table or the select statement will appear. Choose the relevant columns to view in the result.

3 – Settings tab

Change the Failure Options based on the required behavior. The default values are: 1) Fail test when number of return rows is: Lower. 2) Than: 0

This behavior will cause the test to fail when the query/table returns rows.
9 – Referential Integrity

9.1 When to use

The Referential Integrity type of tests includes the following: Referential Integrity and Referential Integrity with complete.

This set of tests intends to ensure that no data is missing in a referential table. In star schema, this test can test dimensions values compared to fact values, or other tables and their reference tables.

Use Referential Integrity to alert on missing value in referential table.

Use Referential Integrity with complete to complete missing value in referential table with default values.

Those set of tests can be used before processing cube or after loading fact tables.

9.2 Referential Integrity

Use Referential Integrity to alert on missing value in referential table.

1 - Data Set tab

Define the Fact table and the Dimension table that need to be compared.

2 – Define relation tab

Define the key relation between the Fact and the Dimension. This key can be based on 1 or more column(s).
Clicks below the Fact Column and Dimension column to open the drop down list. Choose from this list the relevant column name.

<table>
<thead>
<tr>
<th>Fact Column</th>
<th>Fact Column Type</th>
<th>Dimension Column</th>
<th>Dimension Column Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client_ID</td>
<td>Integer</td>
<td>Client_ID</td>
<td>Integer</td>
</tr>
<tr>
<td>Delivery_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery_Date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client_Country_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Num_Days_Late</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country_Name</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4 – Preview & Execute Result

The results include all the keys that are missing in the referential table.

9.3 Referential Integrity with complete

Use Referential Integrity with complete, to complete missing value in referential table with default values.

The complete is possible only with SQL Server DB. In this test all the relations of a dimension to its many fact tables can be defined.

1 - Data Set tab

1. Choose the Dimension table. This table will be the reference to all other facts.
2. When choosing the table all the columns, their data types and default values (based on data type) will be displayed.
3. Check the primary key (ID).
4. In Default Value column will appear the values that will be completed in the dimension in case a key is missing. The Default Value column is completed automatically based on the values decided per data type in the company. It is possible to change the value for every field (row).
5. To change system default value based on data type click on Open default values and modify the values. Click the Save to keep the value.

2 – Define Relation tab
1. Define relation between the fact/facts to the dimension.
2. Check the complete value column.
3. Choose source and fact table. The fact Query will appear automatically. It is possible to modify this query.
4. Define the relation by clicking the relation button – a pop up window will appear.
5. Click the refresh button and choose the fact column field that is a foreign Key.

4 – Preview& Execute Result

By clicking the execute button the missing values in the Reference table will be completed.

Under column Failed it is possible to see the number of missing records with keys that match.
10 – Execution Flow

10.1 When to use

It is possible to create an execution scenario in the style of “what if”: Creating dependencies between tests we created and executing them only if a previous test was failed or succeeded.

By using execution flow we gain:

1. Narrow the problem and find the exact source of problem.
2. Take action immediately when a test failed or succeeded.
3. Lower the number of executions by launching only the necessary ones.

10.2 Execution flow

1. Click Add Quality Gate to add a defined QG test to the sequence.
2. Click Add SQL Execute to add a manual SQL or Store Procedure.
3. Right clicking on a component shows the following:
   a. Edit – Edit the settings of the test and connect it to an existing Quality Gates test.
   b. Delete – Delete the component.
   c. Open – Open the tests.
   d. History – Show the history of the test.
4. Use the From, Type and To create a connection:
   a. From: Source object that was added to the sequence.
   b. Type: In which case the dependent object will be executed. There are the following options: On Complete, On Fail, and On Success.
   c. To: dependent object which was added to the sequence.
   d. Create connection – to create the connection.
5. Use the drop down list Start: Choose the start object which will be the first in the execution flow.
6. Execute: Click to execute the test.