



PUBLIC

How To Create a DQM (Data Quality Management) Rule to Check Product Classification Dimension Assignment

APPLICABLE RELEASES: SAP MASTER DATA GOVERNANCE ON SAP S/4HANA 2020 AND HIGHER

Table of contents

Introduction	3
Preparation: classification and test records	3
Create a new class and characteristic with transaction cl01	3
Create test record	5
Rule Creation – Create the DQM Rule via Fiori app.....	7
Solution concept.....	7
Create dummy rule on A USP.....	7
Implement a DQM rule to check class and characteristics assignment	11
Scope Expression.....	12
Condition Expression	13
Test the solution.....	20

Introduction

With SAP Master Data Governance on SAP S/4HANA 1909 SAP introduced a central rule repository for master data. It is quite easy to configure simple rules like "for all FERT products a material group must be defined" but you might also want to check if a specific scope of products has a classification characteristic (sometimes also called dimension) assigned. This requirement looks quite similar, but you will recognize that it is not that easy to define this rule because the classification data is not stored in your base table MARA and even the classification characteristics are stored in another table than KSSK. In this blog I will show you how to configure such a rule. BTW: Configuration means that you do not need to write any line of code; it is pure configuration in a Web UI.

The overall steps are:

1. Preparation – You will create the classification with characteristics and the test records.
2. Rule Creation – Create the DQM Rule via Fiori app.
3. Test – Execute a DQM evaluation run and check if the result is as expected.

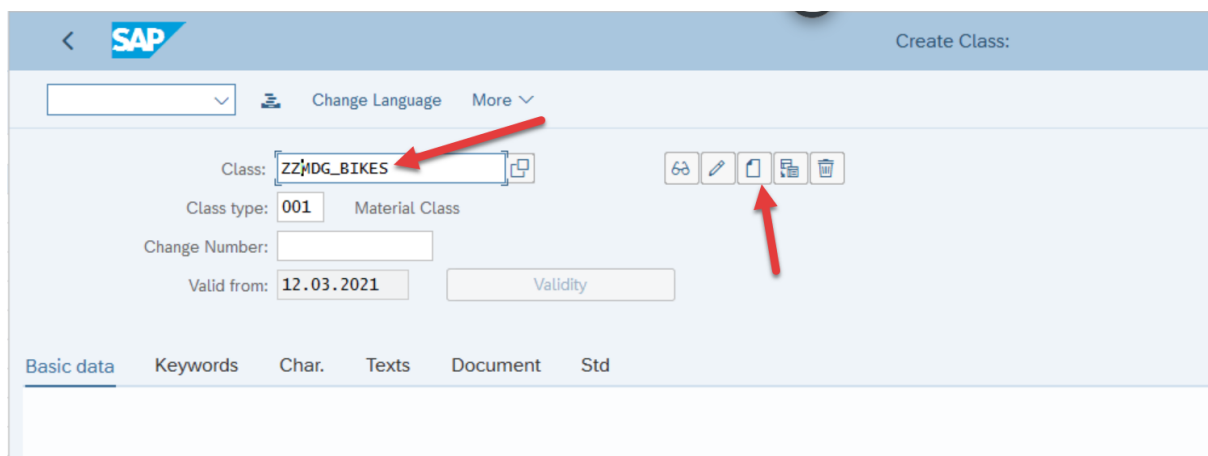
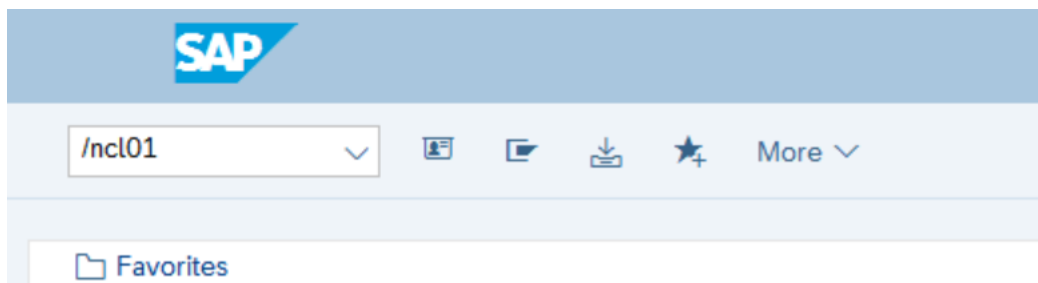
Preparation: classification and test records

Because classifications and their characteristics are typically different from system to system or customer to customer it is not that easy to choose an example which you can rebuild in your own system. As a preparation I have done the following:

1. Create a new class and characteristics using SAP GUI
2. Create test records to double check at the end of the development if your rule works as expected.

Create a new class and characteristic with transaction cl01.

- a. In SAP GUI start transaction cl01 and enter a class name like "ZZMDG_BIKES"



- b. Enter description and click on Char. Tab
- c. Enter a Char. Like “ZZ_MDG_COLOR”

SAP Create Class: Screenshot showing the 'Char.' tab. The Class is ZZMDG_BIKES, Class type is 001, and Valid from is 12.03.2021. The table below shows the characteristics:

Char.	Description	Data ...	Nu...	De...	Unit	R...	Org. Areas	Std C...	O... I...	Origin	P...	S...	D...	I...
<input type="checkbox"/> ZZ_MDG_COLOR														
<input type="checkbox"/>														
<input type="checkbox"/>														
<input type="checkbox"/>														
<input type="checkbox"/>														

- d. Hit enter and the system will ask you if you want to create this characteristic. Click Yes
- e. Enter a description like “ZZ_MDG_COLOR.”
- f. Define Data Type CHAR (10) and Choose Multiple Values. Do not select “Entry Required.”

SAP Create Characteristic: Screenshot showing the 'Basic data' and 'Format' tabs. The Characteristic is ZZ_MDG_COLOR, Valid From is 12.03.2021, and Status is 1 Released. The 'Format' tab shows the Data Type as CHAR Character Format, Number of Chars as 10, and Case Sensitive as unchecked. The 'Value Assignment' tab shows Multiple Values as selected, Restrictable as unchecked, and Entry Required as unchecked.

- g. Click on Values and add 3 values like BLUE; RED; BLACK.

Characteristic:

Change Number:

Valid From:

Basic data Descriptions **Values** Addnl data Restrictions

☐ Additional Values

Allowed Values

Char. Value	Description	D	O	S
<input type="checkbox"/> RED	RED	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> BLUE	BLUE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> BLACK	BLACK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

h. Save the class and characteristics.

Create test record

You will create 2 new records via transaction mm01. Both records will be assigned to the newly created class, but one record will not have the characteristics “ZZ_MDG_COLOR.” Ideally you would use SAP MDG Central Governance to create new materials/products but for simplification reasons we use the backend transaction because the focus is not on “governance of new master data” in this blog.

a. Start transaction mm01.

Class Edit Goto Extras Environment System Help

b. Enter Industry Sector and Material Type and click on “Select Views.”

Material Edit Goto Defaults System Help

< SAP

Select View(s) Org. Levels Data More ▾

Material:

Industry Sector: A Automotive ▾

Material type: FERT Finished Product ▾

Change Number:

- c. Select only Basic data and classification.
- d. Enter a description and BUOM.
- e. Enter class "ZZMDG_BIKES" and select value for the characteristic ZZ_MDG_COLOR
- f. Save your new record.
- g. Result:

Classification

Object

Material: 4010

KOTTER ALBUCH Test Bike

Class Type: 001

Material Class:

Assignments

Class	Description	Stan...	S...	Icon	Itm
<input type="checkbox"/> ZZMDG_BIKES	ZZMDG_BIKES	<input type="checkbox"/>	1	✓	10

Entry: 1 / 1

Values for Class ZZMDG_BIKES - Object 4010

General

Characteristic Description	Value
<input type="checkbox"/> ZZ_MDG_COLOR	BLUE

h. Repeat the steps again but this time do not enter a value for the characteristic. Result:

The screenshot shows the SAP Classification configuration interface. At the top, the title bar says "Classification". Below it, there's a search bar and a "More" dropdown. The "Object" section shows "Material: 4012" with the description "KOTTER ALBUCH Test Bike 2". Below that, "Class Type: 001" and "Material Class:" are visible. The "Assignments" section contains a table with columns: Class, Description, Stan..., S..., Icon, and Itm. The first row shows "ZZMDG_BIKES" for both Class and Description, with a status of 1, a green checkmark icon, and a value of 10. Below the table are navigation icons and an "Entry: 1 / 1" indicator. The bottom section, "Values for Class ZZMDG_BIKES - Object 4012", has a "General" tab. Under this tab, there's a table with "Characteristic Description" and "Value". The first row shows "ZZ_MDG_COLOR" with an empty value field.

Rule Creation – Create the DQM Rule via Fiori app

Solution concept

To make sure that the MARA_AUSP_PRC is filled during the runtime of an evaluation run you may need to create a dummy rule with the base table “AUSP”. The dummy rule is only needed if you do not already have another DQM rule created with base table AUSP. In upcoming releases, it is also planned to overcome this workaround.

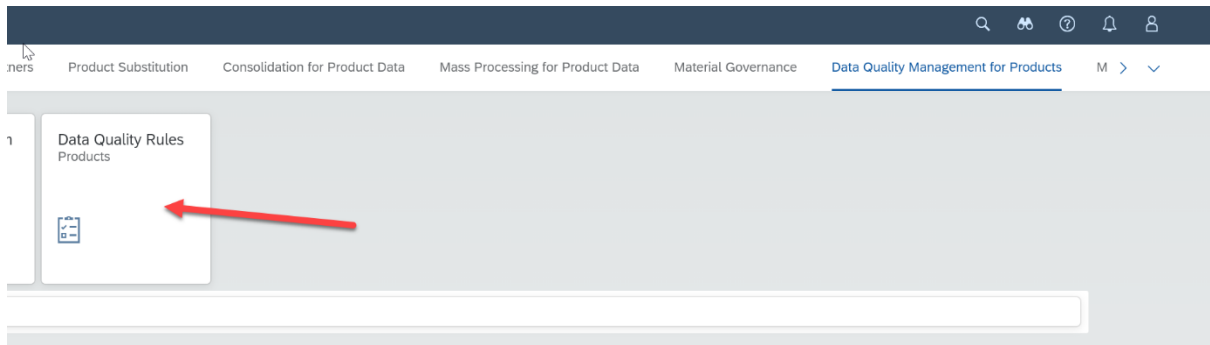
In a second rule you will define the base table “KSSK”. Within the condition expression you will do a DB-lookup to the MARA_AUSP_PRC to check the characteristic assignment for the record in scope related to MARA_KSSK_PRC.

This is the configuration process:

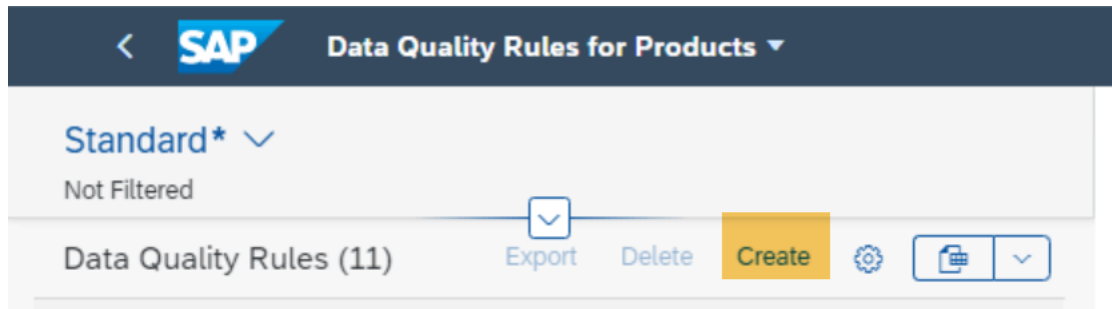
1. Create a dummy DQM rule with base table “MARA_AUSP”.
2. Implement a new DQM rule with base table “MARA_KSSK” and field to be checked “Class Type.”
3. Define the scope expression: Only product records which have the ZZMDG_BIKES assigned.
4. Define condition expression: define a DB lookup to MARA_AUSP_PRC table and check if for the given sourceID the entry for ZMDG_BIKE_COLOR exists.

Create dummy rule on AUSP

- a. Start the app “Data Quality Rules – Products.”



b. Create a new Rule.



c. Enter the mandatory fields:

Z_DEMO_AUSP
Dummy AUSP

Header General Information Usage Implementation Dimensions Additional Information Administrative Data

Rule ID: * Z_DEMO_AUSP Base Table: Characteristics (MARA_AUSP) Status: New
Rule Name: * Dummy AUSP Checked Field: MARA_AUSP-ATWRT

General Information

d. Click Create

e. Add the usage "DQM."

Usage

Add rule usage to proceed with rule implementation.

Add Delete

Usage	Status	Usage Action
		Please assign a usage.

Implementation

Scope Condition

Data Quality Evaluation
Check in Change Requests
Check in Consolidation
Check in Mass Processing

- f. Click on Prepare
- g. Define the BRF+ dummy settings for both expressions:

Usage

Usage	Status	Usage Action
<input type="radio"/> Data Quality Evaluation	Disabled	Enable

Implementation

Check and activate expressions in BRFplus.

Scope	Condition
Expression: Z_DEMO_AUSP_SCP	Expression: Z_DEMO_AUSP_CON
Status: Initial	Status: Initial

- h. The dummy expression for the scope is “false eq false” which is always true. This means that for all records the MARA_AUSP_PRC is filled during runtime.

SAP BRFplus Workbench

Workbench Tools

✓ Objects saved and activated

Repository Catalog

Show: Search Result Search

Search Result

> Z_DEMO_AUSP_SCP

Boolean: Z_DEMO_AUSP_SCP

Back Display Check Save Activate Delete More

General

Detail

Template Invert Condition Result Context Overview Start Simulation

Boolean Logic Schema: <1>

If

False is equal to False Change Edit Operand

Then Boolean is true, else it is false

- i. Same settings must be done for the condition expression:

SAP BRFplus Workbench

Workbench Tools

✓ Objects saved and activated

Repository Catalog

Show: Search Result Search

Search Result S..

> Z_DEMO_AUSP_CON

Boolean: Z_DEMO_AUSP_CON

Back Display Check Save Activate Delete More

General

Detail

Template Invert Condition Result Context Overview Start Simulation

Boolean Logic Schema: <1>

If

False is equal to False Change Edit Operand

Then Boolean is true, else it is false

j.
k. Now you just need to Approve and enable the rule and dummy rule is created.

Z_DEMO_AUSP

Dummy AUSP

Edit Delete Ctrl+E

Base Table: Characteristics (MARA_AUSP) Status

Checked Field: Characteristic Value (MARA_AUSP-ATWRT) New

General Information Usage Implementation Dimensions Additional Information Administrative Data

Business Details

Business Description: -

Business Reason: -

Scope: -

Link: -

Contacts

Rule Owner: -

Implementation Expert: -

Business Contact: -

Data Owner: -

Usage

Add Delete

Usage	Status	Usage Action
<input type="radio"/> Data Quality Evaluation	Disabled	Enable

Implementation

Refresh

Scope

Expression: Z_DEMO_AUSP_SCP

Status: Active

Condition

Expression: Z_DEMO_AUSP_CON

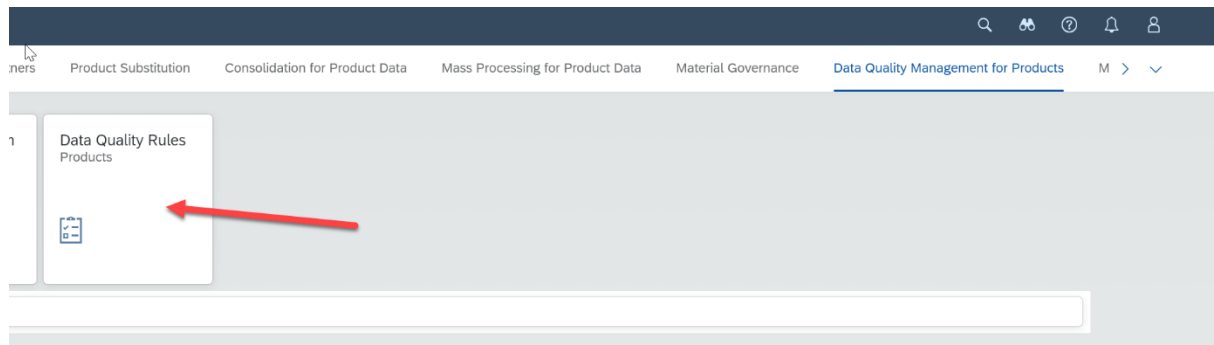
Status: Active

Dimensions

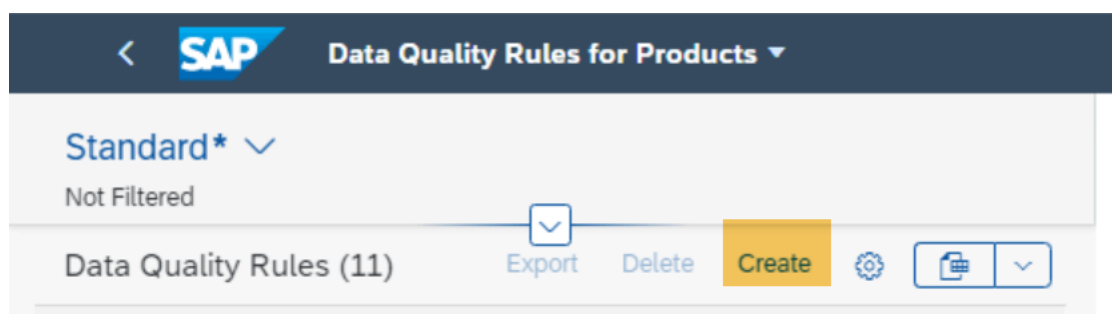
Approve Send for Implementation

Implement a DQM rule to check class and characteristics assignment

- a. Start the app “Data Quality Rules – Products.”



- b. Create a new Rule.



- c. Enter the mandatory fields:

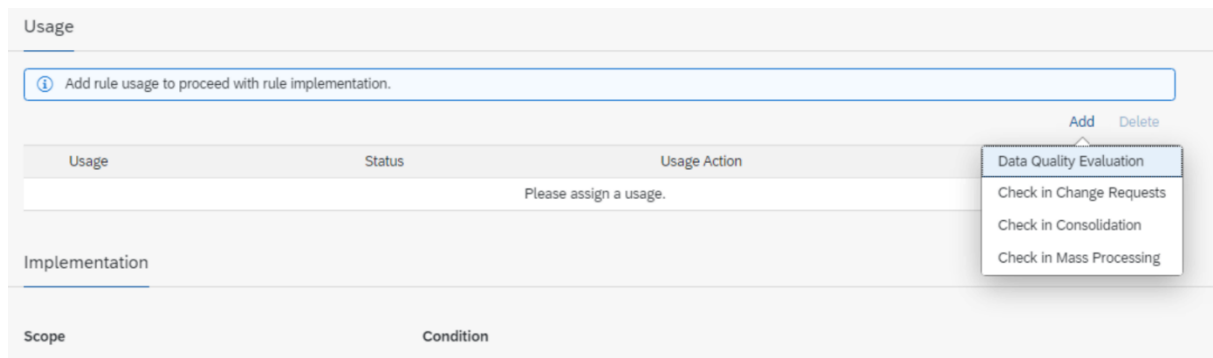
Z_DEMO_KSSK_BIKE
Bikes need a color

Header General Information Usage Implementation Dimensions Additional Information Administrative Data

Rule ID: *	Base Table:	Status:
<input type="text" value="Z_DEMO_KSSK_BIKE"/>	<input type="text" value="Class Assignment (MARA_KSSK)"/>	<input type="text" value="New"/>
Rule Name: *	Checked Field:	
<input type="text" value="Bikes need a color"/>	<input type="text" value="MARA_KSSK-KLART"/>	

General Information

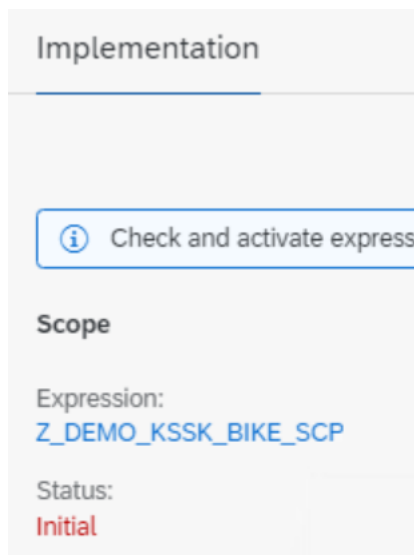
- d. Click Create
- e. Add the usage “DQM.”



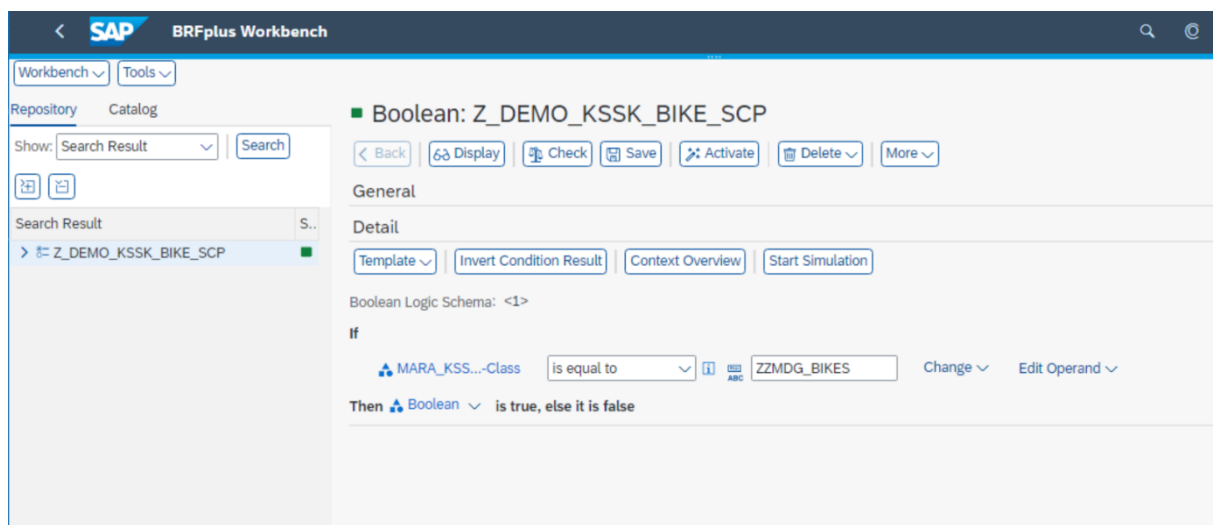
- f. Click on Prepare

Scope Expression

- a. Define the scope expression: You want to check only products which have the class “ZZMDG_BIKES” assigned:
- b. Click on the link on the scope expression to navigate into the BRF+ Workbench.



- c. Click on Edit and define the rule:



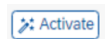
- d. You will find the class attribute in the context. Search for it in this way:

The screenshot shows the 'Context Query' dialog box. It has a 'Search Criteria' section with three rows of search criteria, each with a dropdown for the field, a dropdown for the operator (all set to 'is equal to'), and a text input for the value. The first row has 'Data Object ...' as the field, 'Any' as the value. The second row has 'Name' as the field and '*class*' as the value. The third row has 'Text' as the field and '*' as the value. There are plus and minus icons to the right of each row. Below the search criteria is a 'Maximum Number of Results' field set to '200'. There are 'Search', 'Clear', and 'Reset' buttons. Below this is a 'Result list: 2 objects found' section. It has 'Previous', 'Next', 'Show All Objects', and 'Where Used' buttons. Below these buttons is a table with the following data:

Object	Status	Type	Application
▼ MARA_KSSK Key and Attribute Structure	■	Structure	ZMDQ_194
● Class	■	Text	ZMDQ_194
○			
○			
○			
○			
○			
○			
○			
○			

At the bottom right of the dialog are 'Ok' and 'Cancel' buttons.

- e. Activate the scope expression.

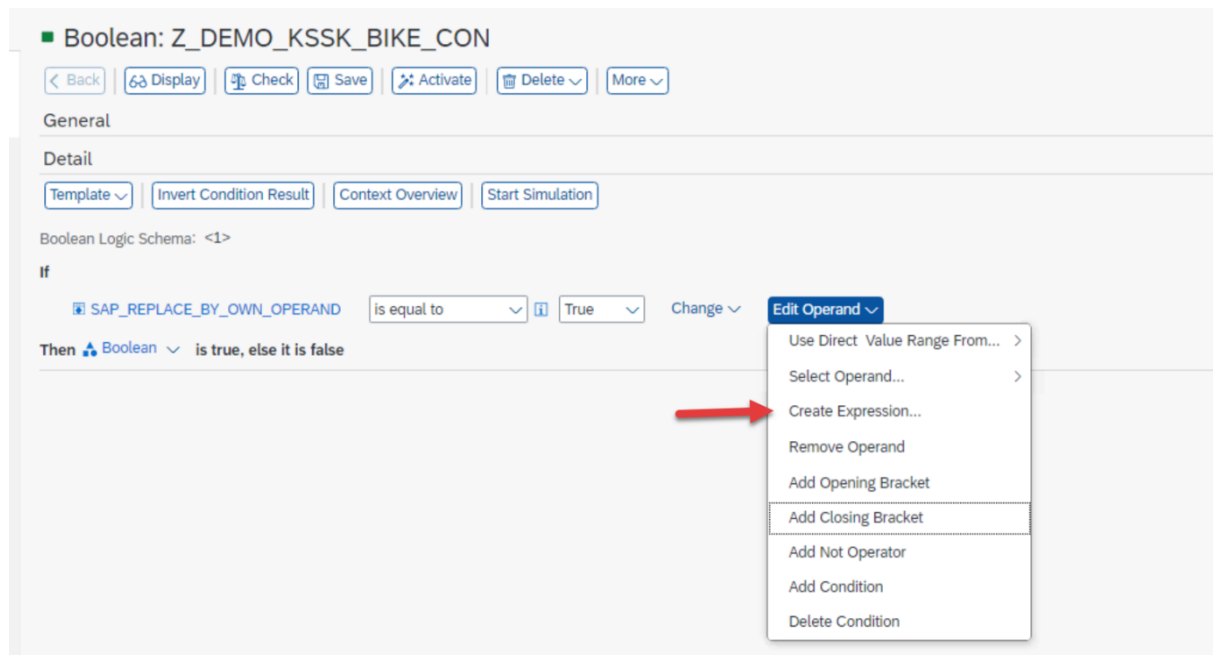


Condition Expression

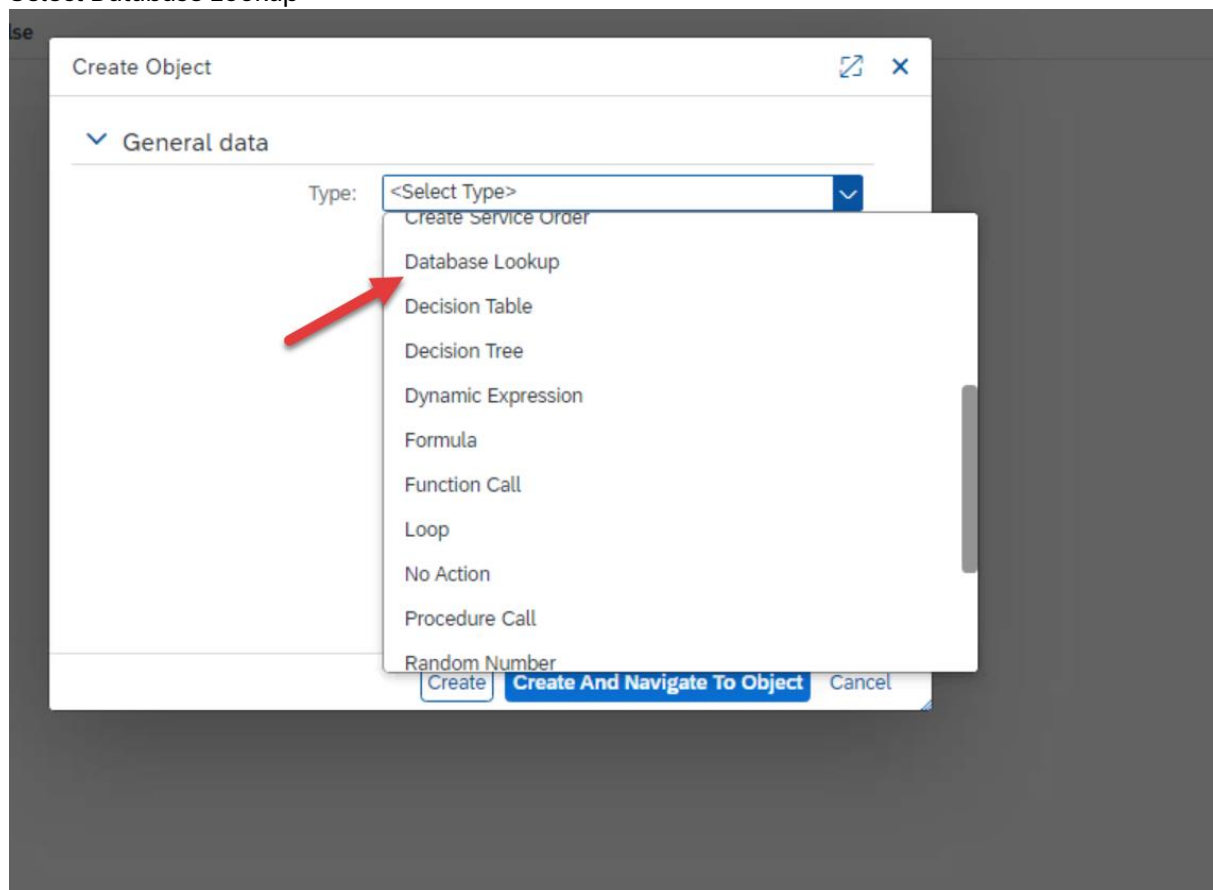
- a. Go back to the DQM Rule and open the BRF+ Workbench for the condition expression:

The screenshot shows the 'Condition' dialog box. It has a label 'Condition' and a text input field for the 'Expression:' containing 'Z_DEMO_KSSK_BIKE_CON'. Below this is a 'Status:' label with a text input field containing 'Initial'.

- b. Create a new Expression as shown in the next screenshot.



c. Select Database Lookup



d. Define a name:

lse

Create Object

General data

Type: Database Lookup

* Name: MARA_AUSP_PRC_CHECK

Short Text:

Text:

Application: ZMDQ_194

Is Reusable: ☒

Create

Create And Navigate To Object

Cancel

- e. Create and Navigate to the Object
- f. Define the DB Lookup as mentioned on the screenshot:

◆ Database Lookup: MARA_AUSP_PRC_CHECK

[Back](#) | [Display](#) | [Check](#) | [Save](#) | [Activate](#) | [Delete](#) | [More](#)

General

Detail

Selection Mode: Existence Check [Context Overview](#) [Start Simulation](#)

If at Least One Entry Exists In Table MARA_AUSP_PRC [Process Records Of OTC 194, Table MARA_AUSP](#)

With Condition: <Not assigned> [Change](#)

Then Boolean

- Insert include condition >
- Insert exclude condition >

- CLIENT (Client)
- PROCESS_ID (Master Data Change Process Identifier)
- PROCESS_STEP_NO (Master Data Change Process Step Number)
- SOURCE_SYSTEM (Source System)
- SOURCE_ID (Master Data Consolidation: ID of a Source Objects)
- ATZHL (Characteristic value counter)
- MAFID (Indicator: Object/Class)
- KLART (Class Type)
- ADZHL (Internal Counter for Archiving Objects by ECM)
- Select other element

Now, it will look like this:

◆ Database Lookup: MARA_AUSP_PRC_CHECK

[Back](#) | [Display](#) | [Check](#) | [Save](#) | [Activate](#) | [Delete](#) | [More](#)

General

Detail

Selection Mode: Existence Check [Context Overview](#) [Start Simulation](#)

If at Least One Entry Exists In Table MARA_AUSP_PRC [Process Records Of OTC 194, Table MARA_AUSP](#)

With Condition: PROCESS_ID is equal to ABC [Change](#)

and

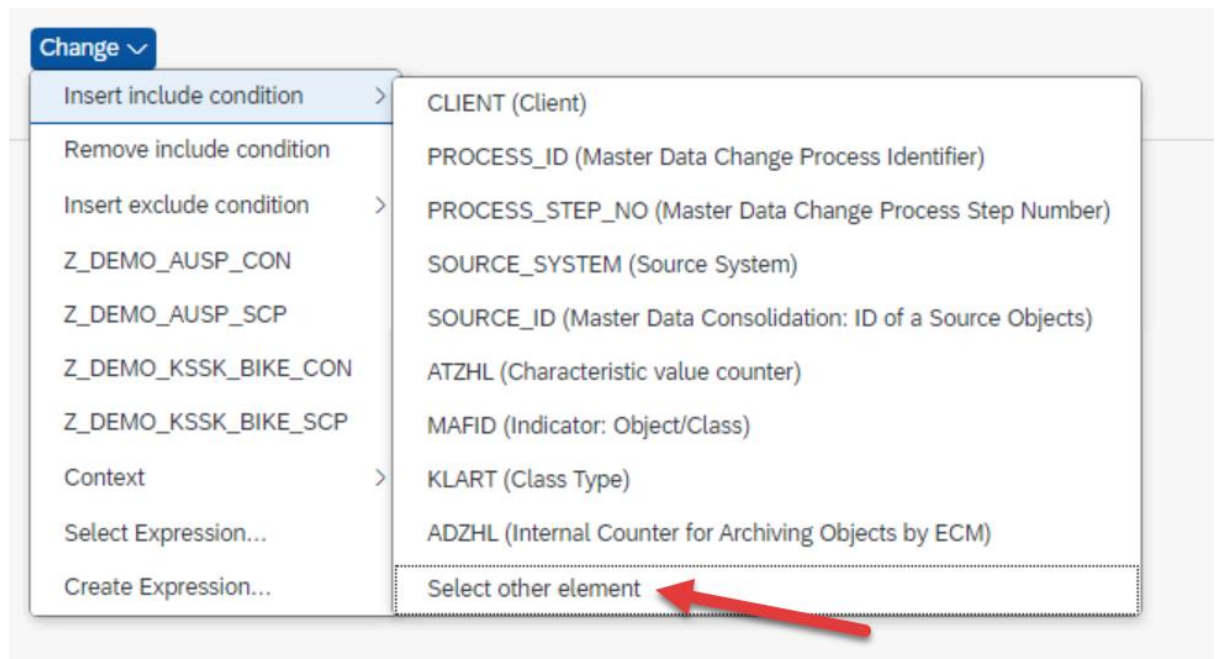
PROCESS_STEP_NO is equal to ABC [Change](#)

and

SOURCE_ID is equal to ABC [Change](#)

Then Boolean Is True, Otherwise It Is False.

g. Now also add the SOURCE_ATINN: You will find it within “Select Other element.”



h. Now enter these values:

- a. PROCESS_ID -> from context
- b. SOURCE_ID -> from context
- c. SOURCE_ATINN = "ZZ_MDG_COLOR"
- d. PROCESS_STEP_NO = "0"

♦ Database Lookup: MARA_AUSP_PRC_CHECK

[Back](#) |
 [Display](#) |
 [Check](#) |
 [Save](#) |
 [Activate](#) |
 [Delete](#) |
 [More](#)

General

Detail

Selection Mode: [Existence Check](#) | [Context Overview](#) | [Start Simulation](#)

If at Least One Entry Exists In Table MARA_AUSP_PRC Process Records Of OTC 194, Table MARA_AUSP

With Condition: PROCESS_ID is equal to Change

and

PROCESS_STEP_NO is equal to Change

and

SOURCE_ATINN is equal to Change

and

SOURCE_ID is equal to Change

Then Boolean Is True, Otherwise It Is False.

Change

- Insert include condition
- Remove include condition
- Insert exclude condition
- Z_DEMO_AUSP_CON
- Z_DEMO_AUSP_SCP
- Z_DEMO_KSSK_BIKE_CON
- Z_DEMO_KSSK_BIKE_SCP
- Context
- Select Expression...
- Create Expression...

- Changed by
- Complete status
- Created by
- Created On
- DF client level
- Last Change
- Maint. status
- Material type
- Source ID
- Source System
- More...

i. Result:

◆ Database Lookup: MARA_AUSP_PRC_CHECK

[Back](#)
[Display](#)
[Check](#)
[Save](#)
[Activate](#)
[Delete](#)
[More](#)

General

Detail

Selection Mode: [Existence Check](#) [Context Overview](#) [Start Simulation](#)

If at Least One Entry Exists In Table [MARA_AUSP_PRC](#) [Process Records Of OTC 194, Table MARA_AUSP](#)

With Condition:

PROCESS_ID [is equal to](#) [Evaluati...-Process ID](#) [Change](#)

and

PROCESS_STEP_NO [is equal to](#) [0](#) [Change](#)

and

SOURCE_ATINN [is equal to](#) [ZZ_MDG_...](#) [Change](#)

and

SOURCE_ID [is equal to](#) [MARA Key...-Source ID](#) [Change](#)

Then [Boolean](#) [Is True, Otherwise It Is False.](#)

- j. Activate the Expression
- k. Go back to the BRF+ rule and delete the predefined line. It should look like this:

■ Boolean: Z_DEMO_KSSK_BIKE_CON

[Back](#)
[Display](#)
[Check](#)
[Save](#)
[Activate](#)
[Delete](#)
[More](#)

General

Detail

[Template](#)
[Invert Condition Result](#)
[Context Overview](#)
[Start Simulation](#)

Boolean Logic Schema: <1>

If

[MARA_AUSP_PRC_CHECK](#)

Then [Boolean](#) [is true, else it is false](#)

Explanation: This condition expression gives back TRUE if where is an entry in MARA_AUSP_PRC (related to the process id, source or productid, Process Step and characteristic). This is done via the database lookup to the table which was filled by the dummy rule.

- l. Now you just need to Approve and enable the rule and dummy rule is created.

Z_DEMO_KSSK_BIKE
Bikes need a color

Base Table: Class Assignment (MARA_KSSK) Status
Checked Field: Class Type (MARA_KSSK-KLART) **New**

General Information Usage Implementation Dimensions Additional Information Administrative Data

Business Details

Business Description:
—

Business Reason:
—

Scope:
—

Link:
—

Contacts

Rule Owner:
—

Implementation Expert:
—

Business Contact:
—

Data Owner:
—

Usage

Add Delete

Usage	Status	Usage Action
<input type="radio"/> Data Quality Evaluation	Disabled	Enable

Implementation

Refresh

Scope

Expression:
Z_DEMO_KSSK_BIKE_SCP

Status:
Active

Condition

Expression:
Z_DEMO_KSSK_BIKE_CON

Status:
Active

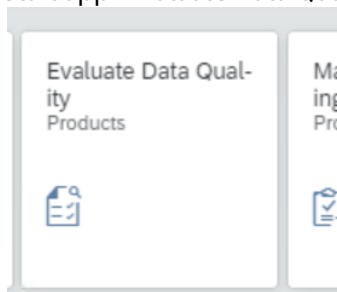
Dimensions

Approve Send for Implementation

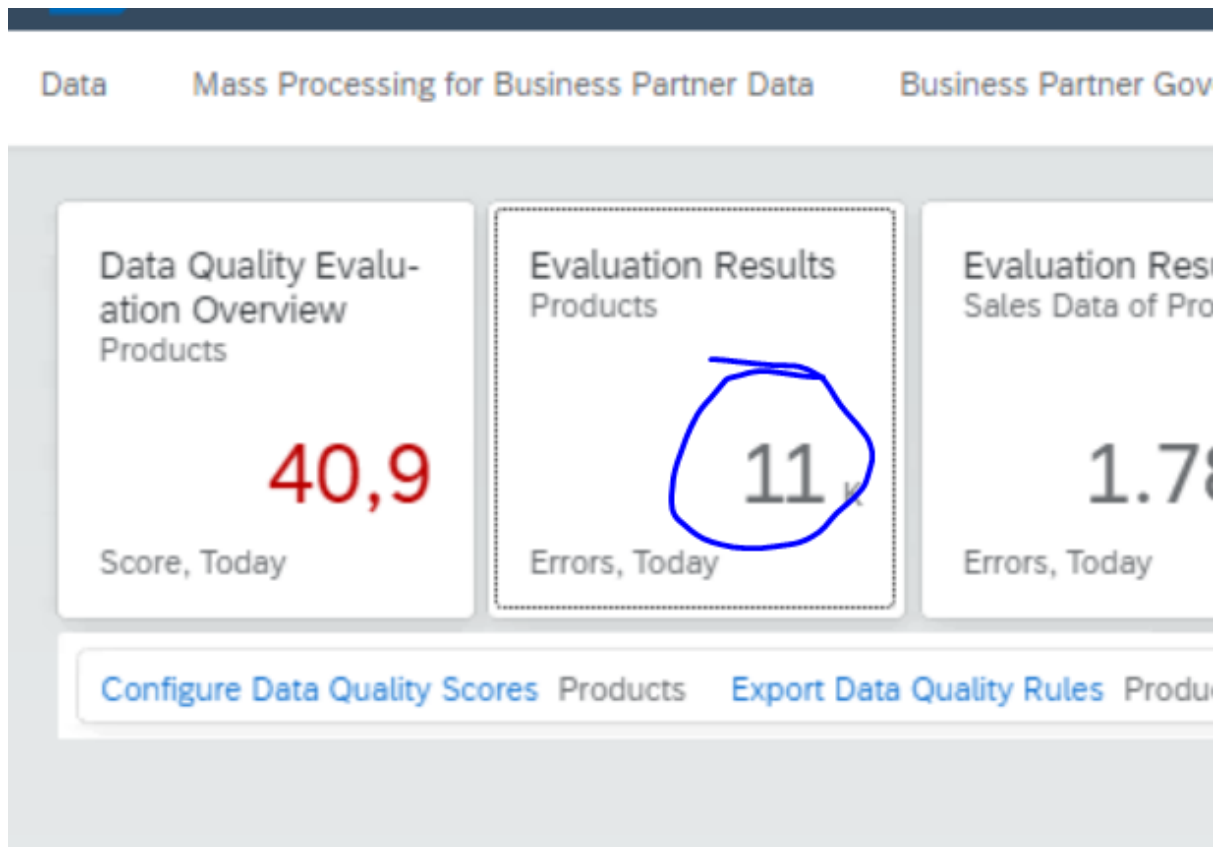
Test the solution

Execute an DQM evaluation run and check if the result is as expected:

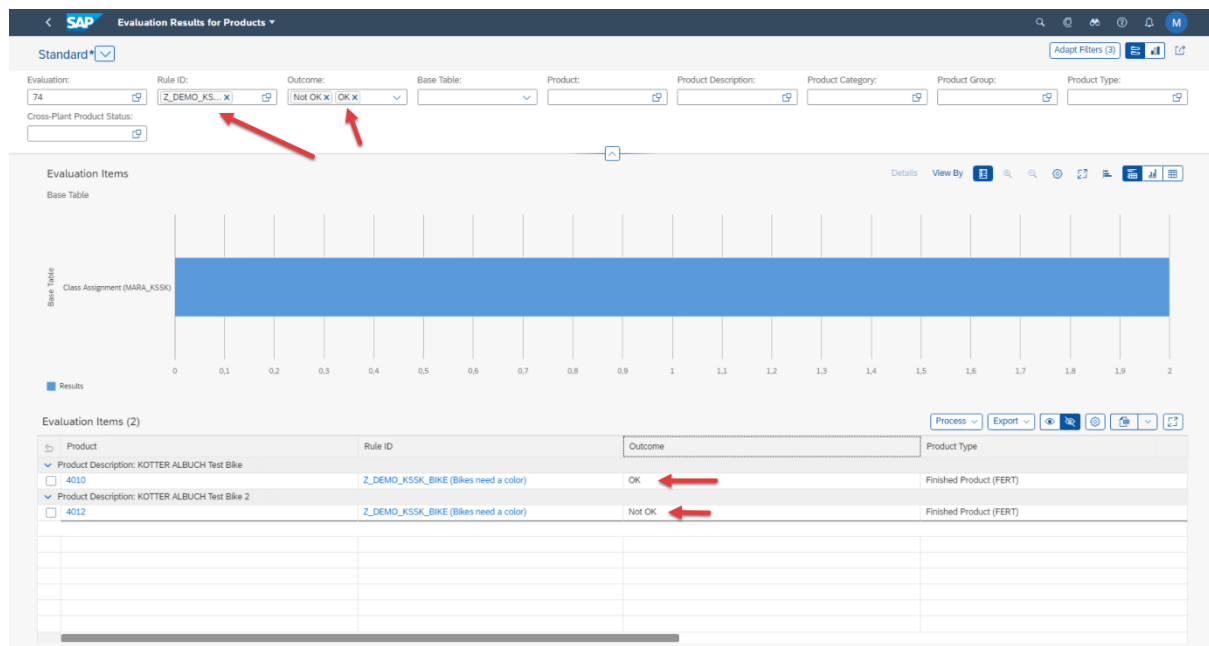
- Start app “Evaluate Data Quality for Products.”



- Start the evaluation and wait until the run is completed.
- Start the app “Evaluation Results Products.”



d. Filter on “Base Table” and/or “Rule ID”:



If you followed the guide from the beginning, you would see the 2 product records: one record is “ok” and the other is “Not Ok”. This is the expected result.