PUBLIC

How to Enable Master Data Rule Mining with Classification Data

Applicable Releases:

From SAP Master Data Governance on SAP S/4HANA 1909 and newer

Version 1.1

September 2024

Document History

Document Version	Description
1.0	First official release of this guide
1.1	Format update

Table of contents

1. Introduction

Data Quality Management with SAP Master Data Governance achieves the best quality for master data in customer's SAP S/4HANA system(s) and their entire enterprise application landscape by extending and integrating with all MDG-related processes, in the most efficient way, at the lowest possible implementation efforts and TCO, by following an extensible packaged-application approach.

MDG Rule Mining supports the business / master data expert to analyze existing master data and propose the found patterns as rules by leveraging machine learning technology.

Classification data is essential for companies to manage their flexible data structure in important master data e.g. product. This document describes how you could enable the rule mining for your master data combining classification data by using "product" as example.

Classification data is not included in the SAP standard delivery in rule mining for products and business partners. This is because of the generic data structure in classification. This document explains how the classification data can be transformed to function as a normal data source and combined with master data (For example, Product) to apply to rule mining.

2. Implementation

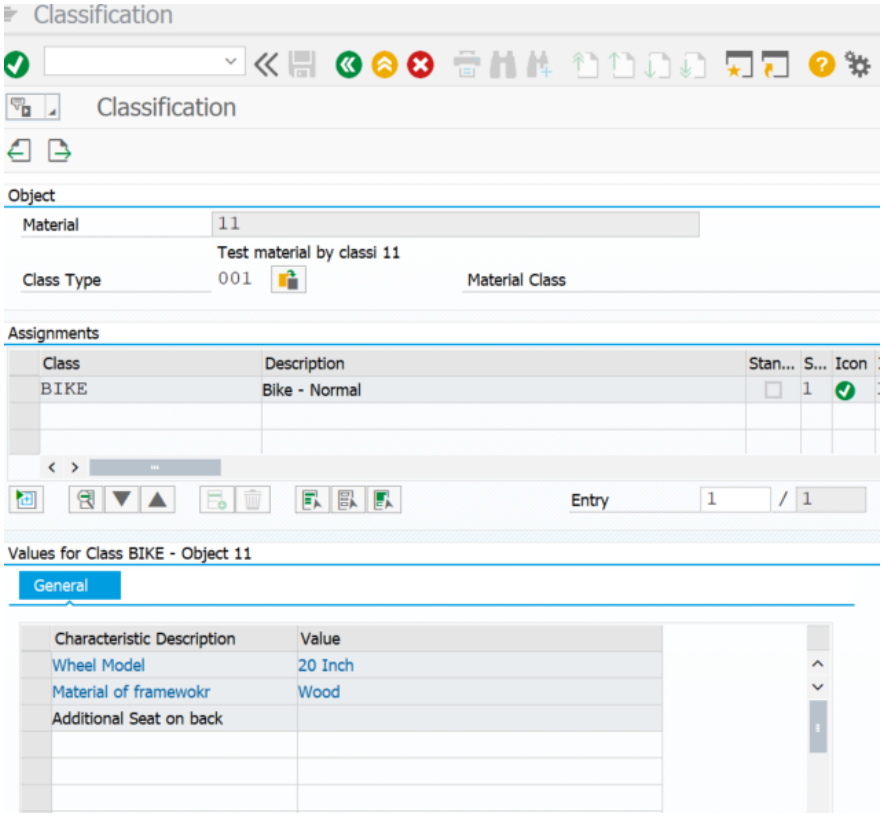
This chapter explains all necessary technical steps to implement the solution.

2.1. Build a CDS View to transform Classification Data

To be able to use rule mining in a Master Data model, it is necessary to transform the generic classification data to an understandable data source such as a normal table, here we choose an ABAP CDS view which is standard and easiest for modelling in our environment.

2.1.1. Supported Characteristics

Because characteristics must be transformed into normal attributes for master data, only Single Value and Character Format is supported. This screenshot shows an example a product classification with 3 characteristics with Single Values.



2.1.2. Create CDS View

You need to use the ABAP Development tool in Eclipse to create a CDS view in your system, be aware that you need developer authorization in your system to create it in the “Z” namespace.

Annotation @AbapCatalog.preserveKey must be set to True.

Here is an example of a CDS view which you could copy and use for your own classification transformation.

```
@AbapCatalog.sqlViewName: 'ZMATCLS01'

@AbapCatalog.compiler.compareFilter: true

@AbapCatalog.preserveKey: true

@AccessControl.authorizationCheck: #CHECK

@EndUserText.label: 'Classficaiton Data as extension of MARA'

define view ZKF_PROD_CLS01

 as  select distinct

 key klah.class  as  Class ,

 key cast( RTRIM( SUBSTRING( CLASS.objek , 1  , 40  ) , ' ' ) as matnr )  as  Material  ,

 @EndUserText.label: 'Additional Seat on back'

cast ( AUSP0001.atwrt as zseat2 )  as  ZSEAT2  ,

 @EndUserText.label: 'Additional Seat on back description'

VALD0001._CharcValueDesc( P_KeyDate: $session.system_date )[1:Language = 'E' ].CharcValueDescription  as
ZSEAT2_Desc  ,

 @EndUserText.label: 'Material of framewokr'
```

```
cast ( AUSP0002.atwrt as zmaterial )   as  ZMATERIAL  ,
 @EndUserText.label: 'Material of framewokr description'
VALD0002._CharcValueDesc( P_KeyDate:  $session.system_date  )[1:Language = 'E' ].CharcValueDescription  as
ZMATERIAL_Desc  ,
 @EndUserText.label: 'Wheel Model'
cast ( AUSP0003.atwrt as zwheel )   as  ZWHEEL  ,
 @EndUserText.label: 'Wheel Model description'
VALD0003._CharcValueDesc( P_KeyDate: $session.system_date  )[1:Language = 'E' ].CharcValueDescription  as
ZWHEEL_Desc
  from  kssk  as  CLASS
  inner join  klah  as  KLAH  on  klah.clint  = CLASS.clint
                              // and klah.class  = 'BIKE'

  inner join  cabn  as  CABN0001  on  CABN0001.atnam  = 'SEAT2'
  left outer join  ausp  as  AUSP0001  on  AUSP0001.mandt  = CLASS.mandt
                                       and AUSP0001.klart  = CLASS.klart
                                       and AUSP0001.mafid  = CLASS.mafid
                                       and AUSP0001.objek  = CLASS.objek
                                       and AUSP0001.atinn  = CABN0001.atinn
  left outer join  I_ClfnCharcValForKeyDateTP ( P_KeyDate :  $session.system_date  )  as  VALD0001
               on  VALD0001.CharcInternalID  = CABN0001.atinn
               and VALD0001.CharcValue  = AUSP0001.atwrt
  inner join  cabn  as  CABN0002  on  CABN0002.atnam  = 'MATERIAL'
  left outer join  ausp  as  AUSP0002
               on  AUSP0002.mandt  = CLASS.mandt
               and AUSP0002.klart  = CLASS.klart
               and AUSP0002.mafid  = CLASS.mafid
               and AUSP0002.objek  = CLASS.objek
               and AUSP0002.atinn  = CABN0002.atinn
  left outer join  I_ClfnCharcValForKeyDateTP ( P_KeyDate :  $session.system_date )  as  VALD0002
                on  VALD0002.CharcInternalID  = CABN0002.atinn and VALD0002.CharcValue  = AUSP0002.atwrt
  inner join  cabn  as  CABN0003  on  CABN0003.atnam  = 'WHEEL'
  left outer join  ausp  as  AUSP0003
               on  AUSP0003.mandt  = CLASS.mandt
               and AUSP0003.klart  = CLASS.klart
               and AUSP0003.mafid  = CLASS.mafid
               and AUSP0003.objek  = CLASS.objek
               and AUSP0003.atinn  = CABN0003.atinn
```

```
left outer join  I_ClfnCharcValForKeyDateTP ( P_KeyDate :  $session.system_date  )  as  VALD0003
              on  VALD0003.CharcInternalID  = CABN0003.atinn
              and VALD0003.CharcValue  = AUSP0003.atwrt


where  CLASS.mandt  = $session.client
       and CLASS.mafid  = 'O'
       and CLASS.klart  = '001'    // classtype 001
       and CLASS.statu = '1';
```

2.1.3. Test CDS View

Once you have created the CDS view, you should test in your environment to see if the data is displayed correctly.

ZKF_PROD_CLS01

Raw Data

Filter pat... **8 rows retrieved - 13 ms** SQL Console Number of Entries S

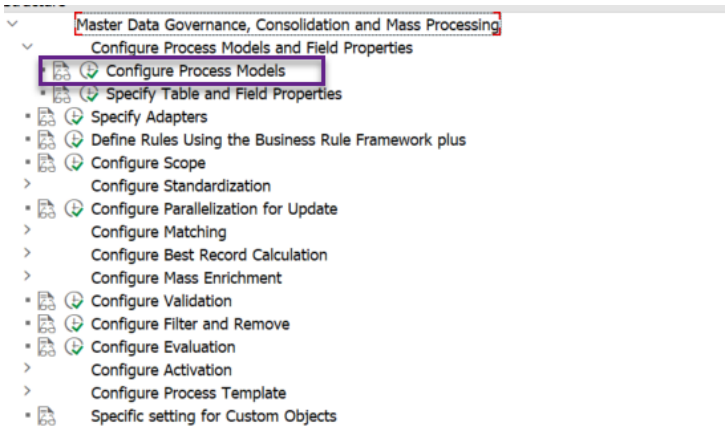
Class	Material	ZSEAT2	ZSEAT2_Desc	ZMATERIAL	ZMATERIAL_Desc	ZWHEEL	ZWHEEL_Desc
BIKE	000000000000000001	0	No	02	Wood	001	20 Inch
BIKE	000000000000000011			02	Wood	001	20 Inch
BIKE	000000000000000012	0	No	02	Wood	002	28 Inch
BIKE	000000000000000091	0	No	03	Plastic	001	20 Inch
BIKE	000000000000000092	1	Yes	01	Steel	002	28 Inch
BIKE	000000000000000093	0	No			001	20 Inch
BIKE	000000000000000101						
MOTOR	000000000000000101						

2.2. Extend Data Model with Classification Data

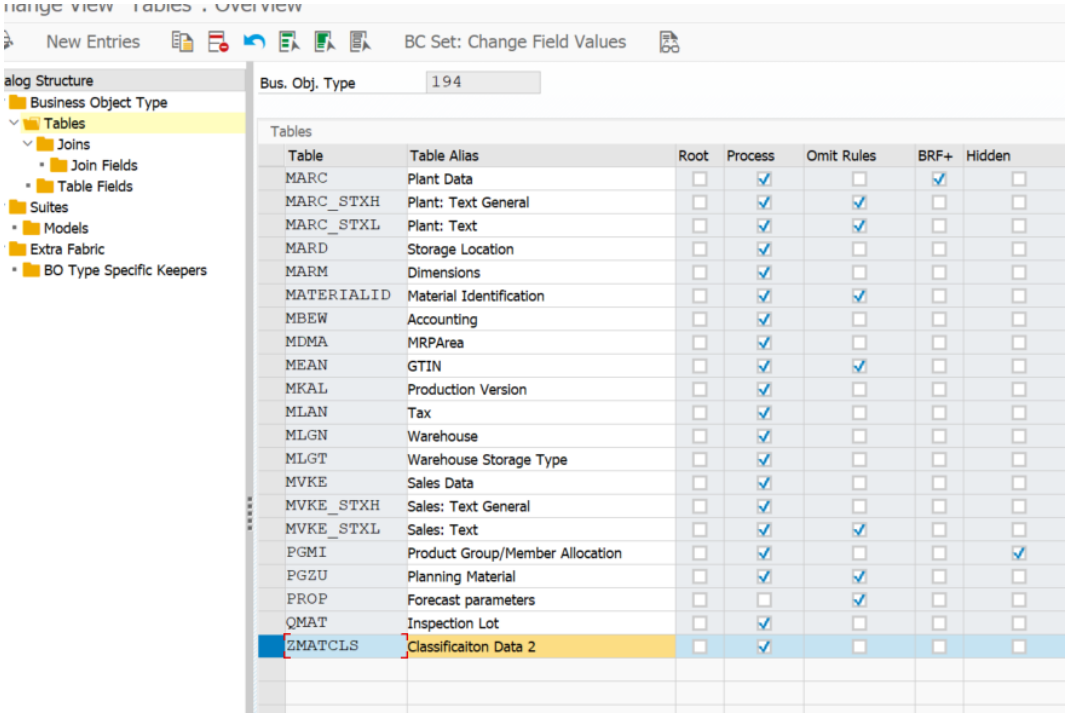
Next you need to connect the new CDS view to the rule mining data model.

2.2.1. Data Model Extension

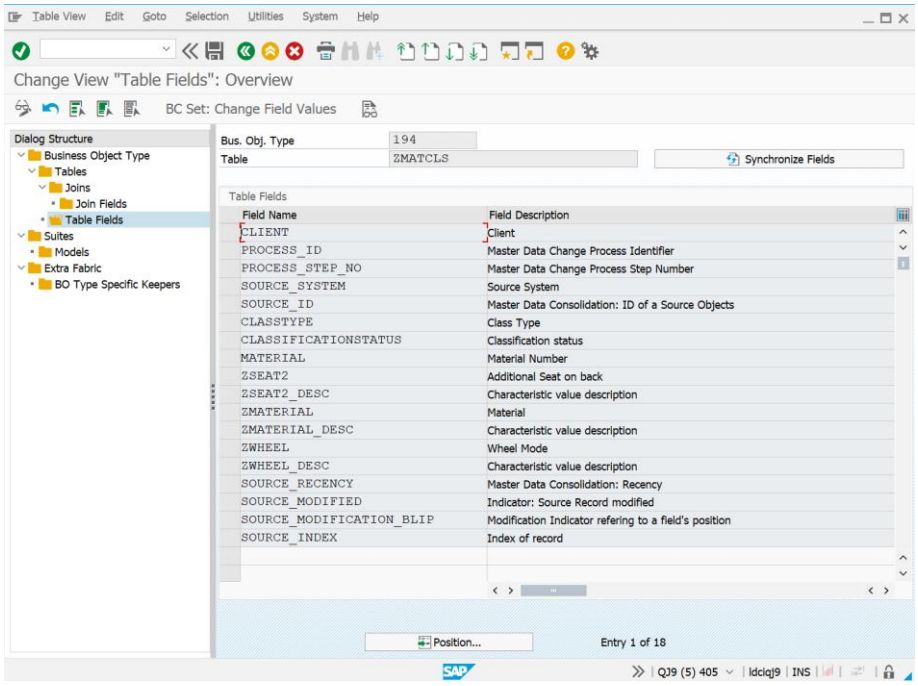
Go to transaction MDCIMG, *Configure Process Models and Field Properties* → *Configure Process Models*,



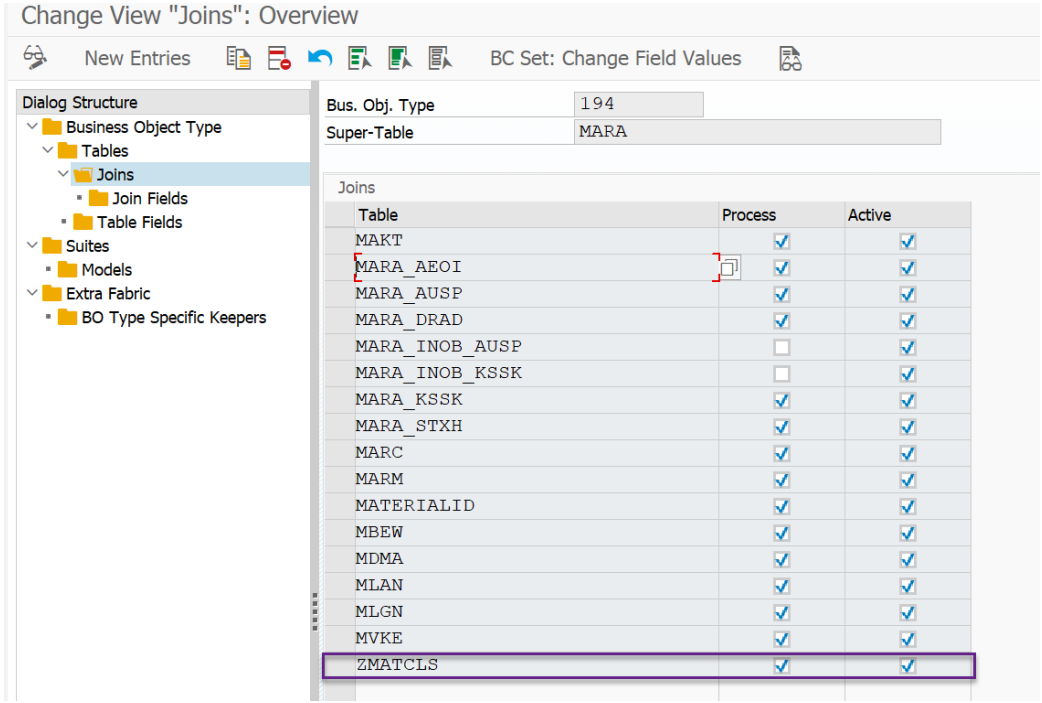
In the Tables view enter the CDS SQL view name, check the “Process” flag and Save.

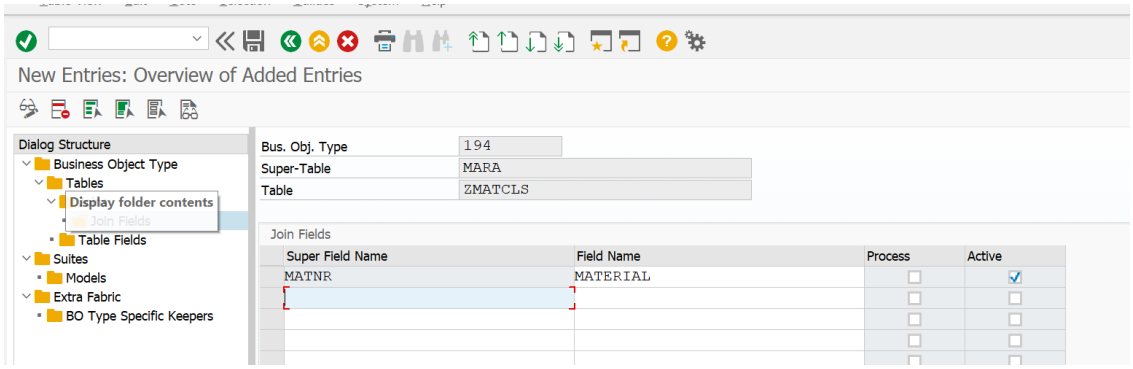


Go to Table – Fields, and choose the Synchronize Fields button.



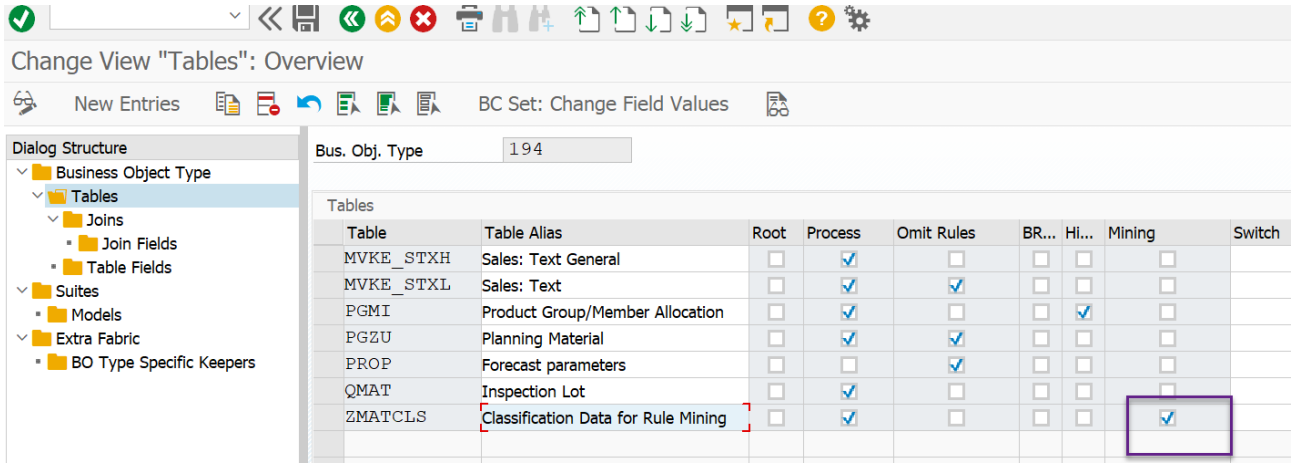
Choose the MARA table and enter the *Joins view. E*nter the view name and check the *Process* and *Active* flags. Navigate to *Joins-> Join Field* -> enter the MATNR and the fieldname of material in your CDS view. Choose Save.





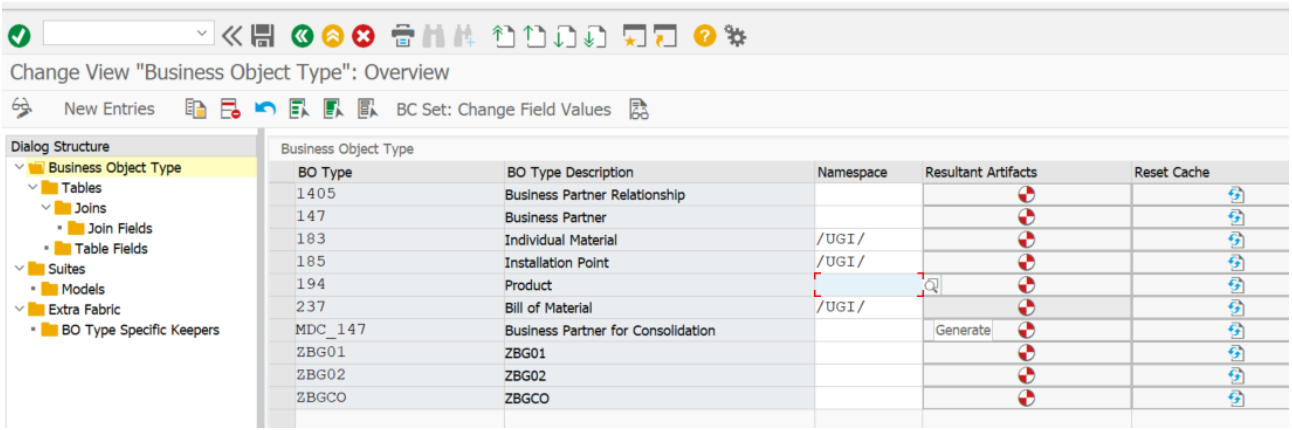
Go back to the Tables view, and for your CDS view, check the *Mining* flag.

Choose Save.

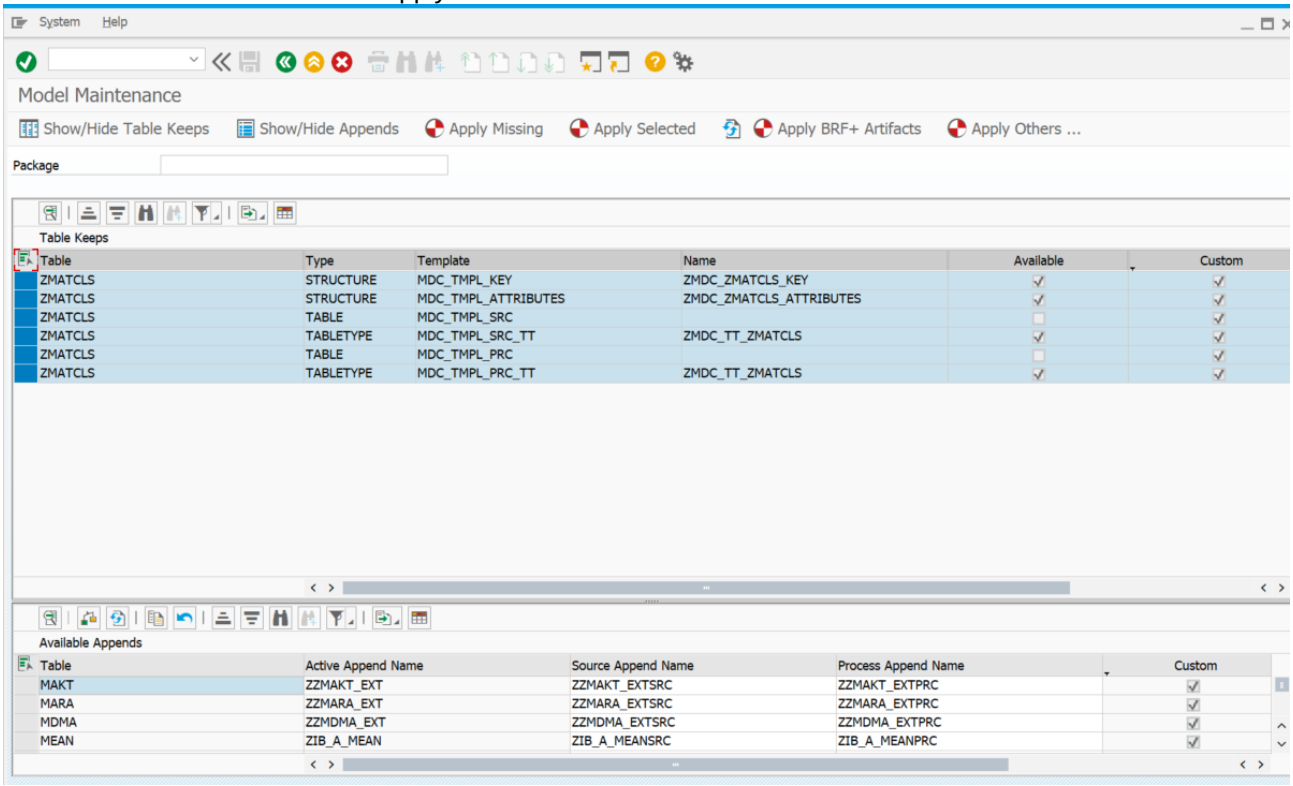


2.2.2. Generate Artifacts

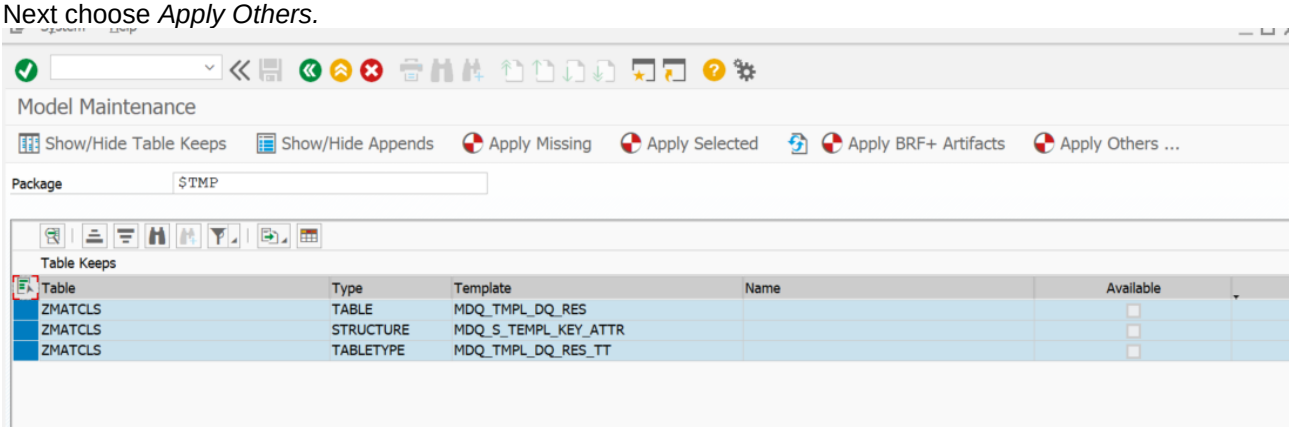
To generate resultant artifacts for your new table, navigate to Business Object Type, and choose *Resultant Artifacts.*



Mark all the entries and choose Apply Selected.



Next choose *Apply Others.*



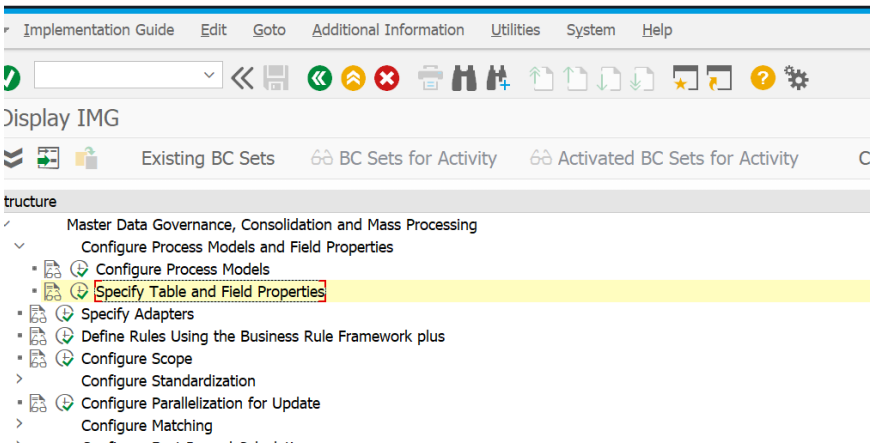
Mark all the entries with your CDS view, and choose *Apply Selected*

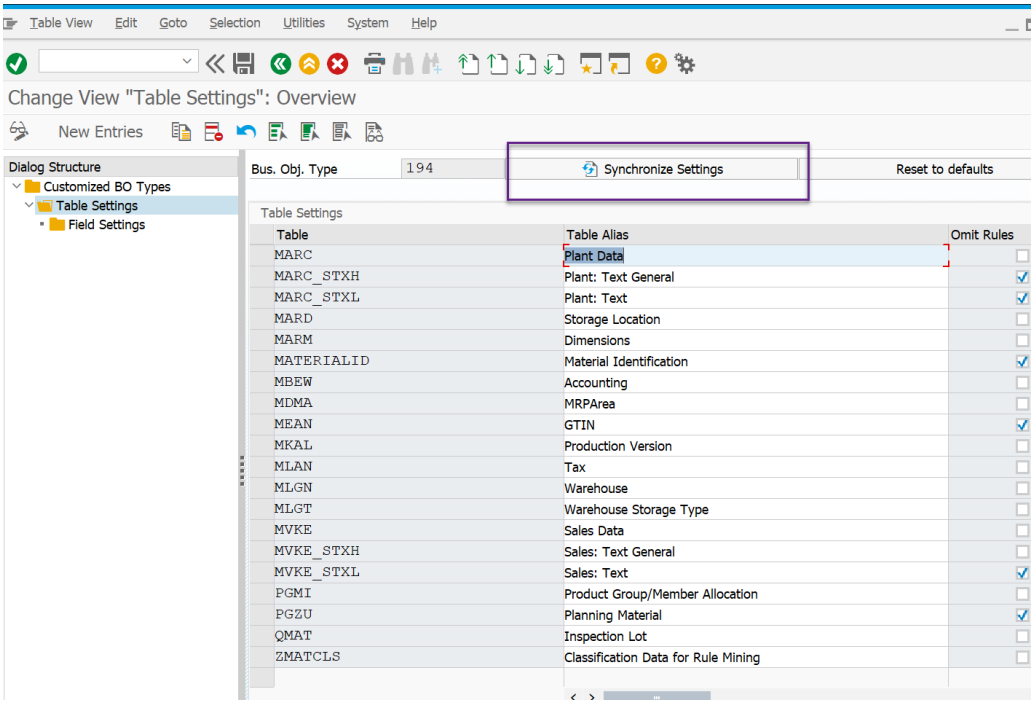
Go to SE11 to review the 3 generated tables.

ZMATCLS_DQ_RES	Master Data Quality Results of OTC 194, Ta
ZZMATCLS_PRC	Process Records Of OTC 194, Table ZMATCLS
ZZMATCLS_SRC	Source Records Of OTC 194, Table ZMATCLS

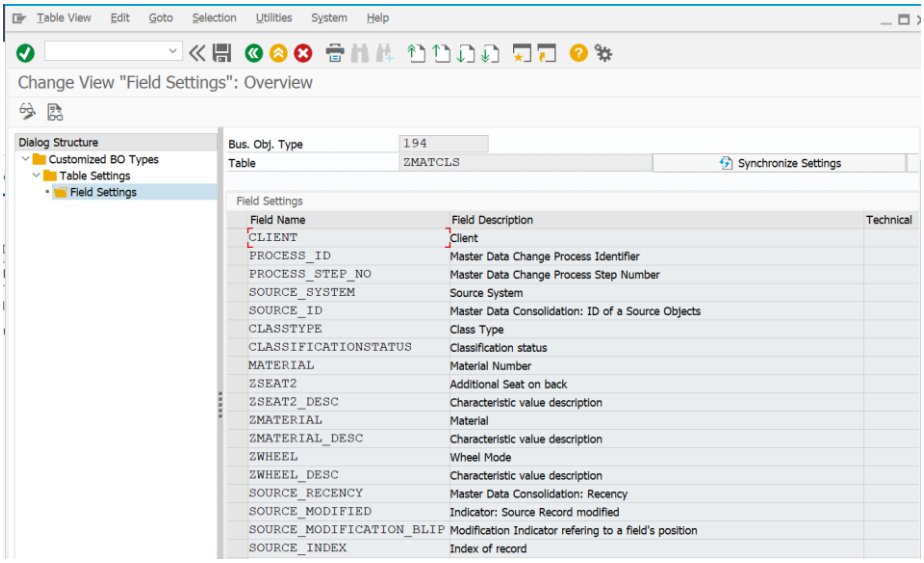
2.2.3. Specify Table and Field Properties

Go to Business Object 194 (if it is not there yet, add it), and choose Synchronize Settings.





Mark your CDS view, go to the *Field Settings view*, choose Synchronize Settings and Save.



The implementation is now complete.

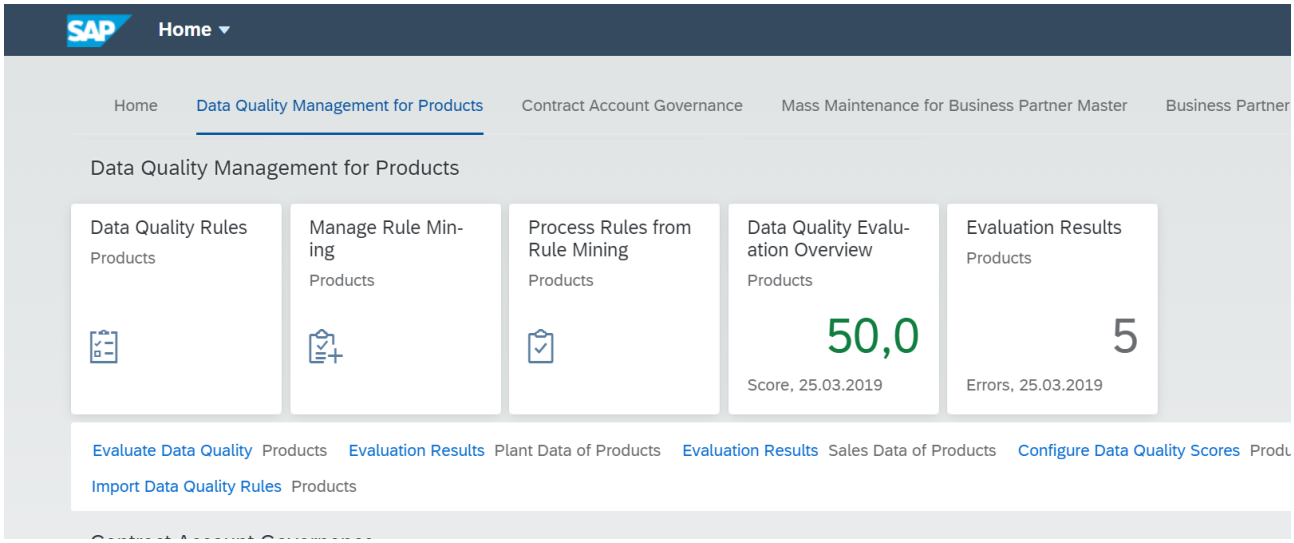
3. Process Test

3.1. Preparation

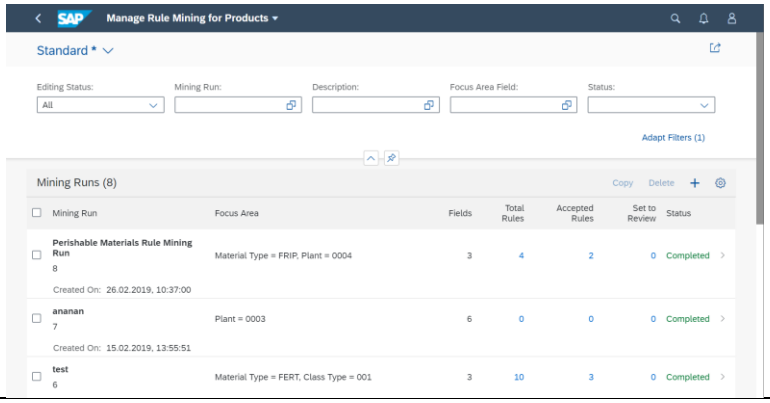
You must have the appropriate data quality authorization role and Fiori Launchpad role.

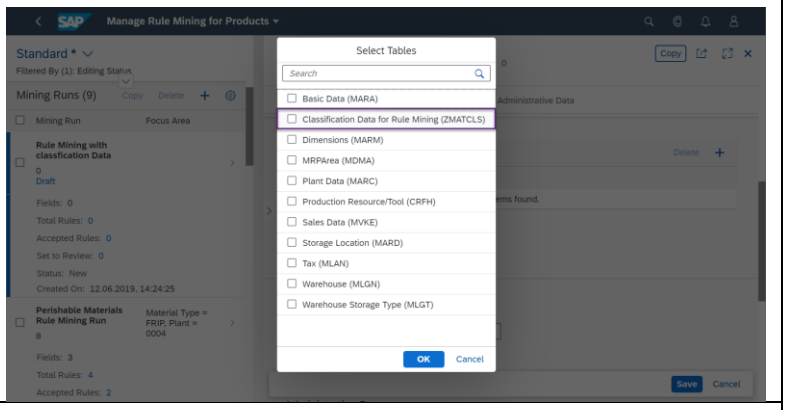
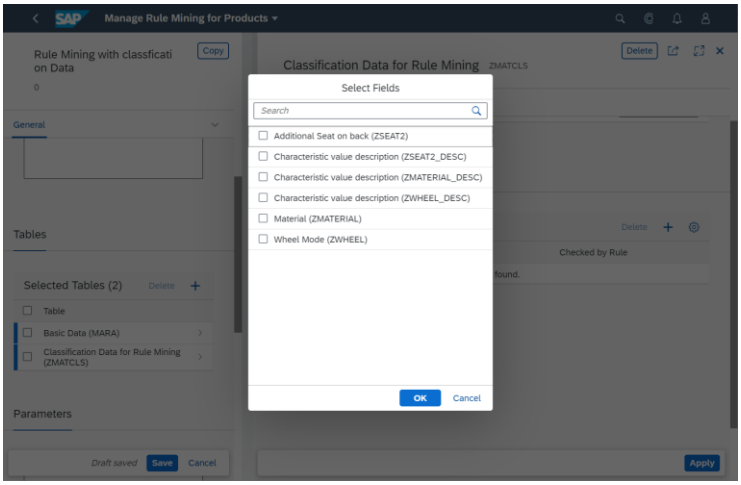
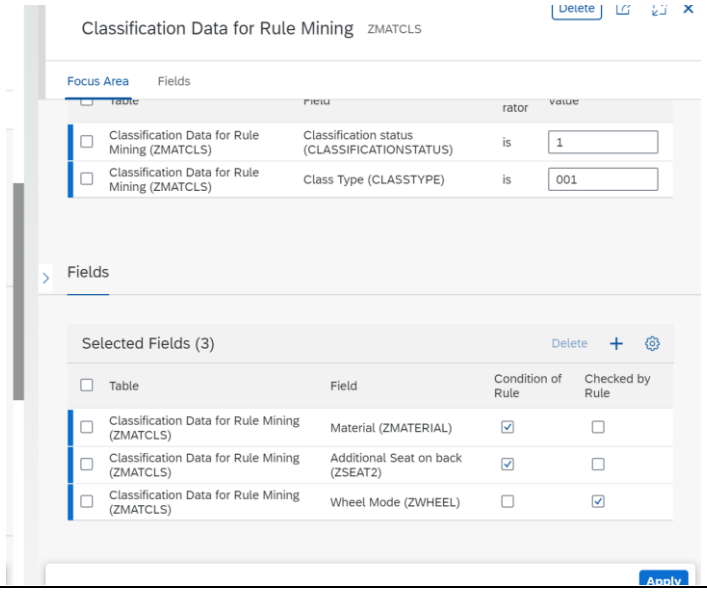
In this document, we are using Product master data, so you need to have the Fiori role SAP_BR_PRODMASTER_STEWARD assigned to your user. Follow the standard S/4 authorization setup for getting the authorization to run the rule mining solution. Basically, you will need the authorization objects *MDQRM_MNDR, MDQRM_MNNG and MDQRM_RULE.*

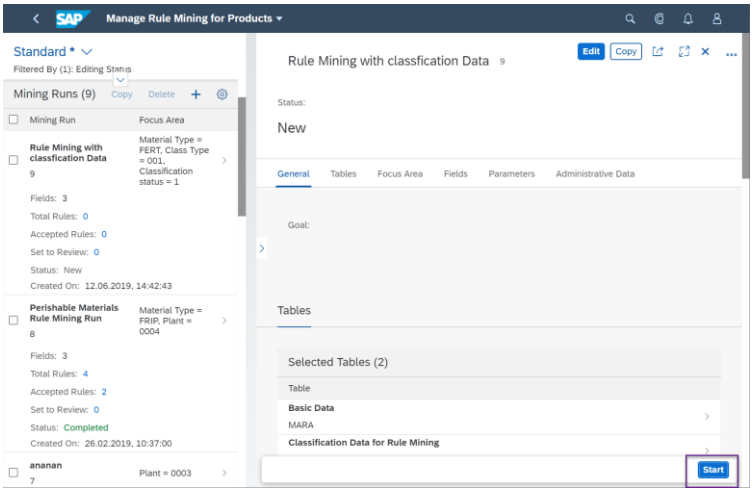
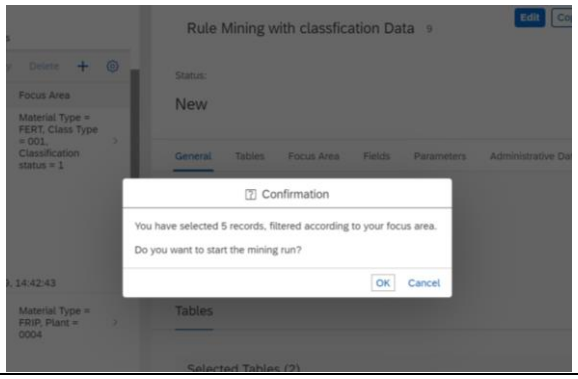
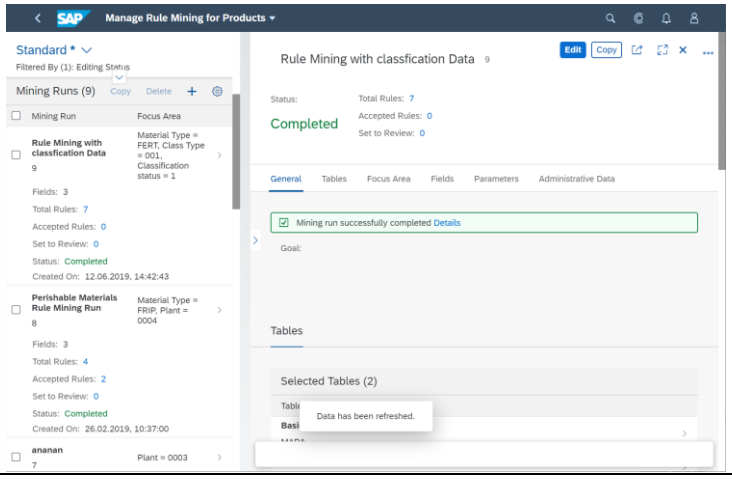
Once you have the correct roles assigned, log on to the Fiori launchpad and go to Data Quality Managements for Products.

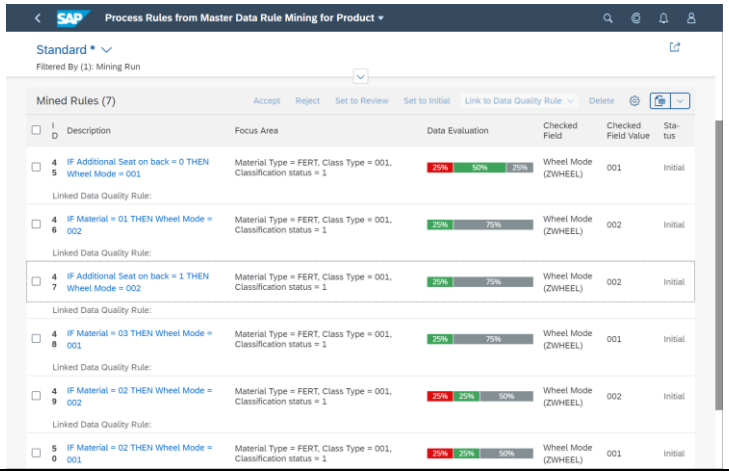


3.2. Rule Mining

In the Manage Rule Mining app, choose New Mining run	

Enter a description. In the Table section, choose **+** to add a table. CDS View should appear in the list on the popup. Mark it and choose OK.	
After you get the CDS view added in the list, the details page will appear. On the details page you can add a filter on the focus area which is used to restrict the data for the mining. Choose **+** to add a filter on your data if you wish. On the Fields section of details page, you can add your characteristics as fields which you want to discover rules. Choose **+**, on the popup, the characteristics which you have exposed in your CDS view should appear on the list. Add several fields.	
Mark at least one of your added fields with the *Checked by Rule flag.*	

Save the Mining Run and start it.	 
When the mining run finishes, you can navigate to the results by choosing the number after *Total Rules* on the header.	

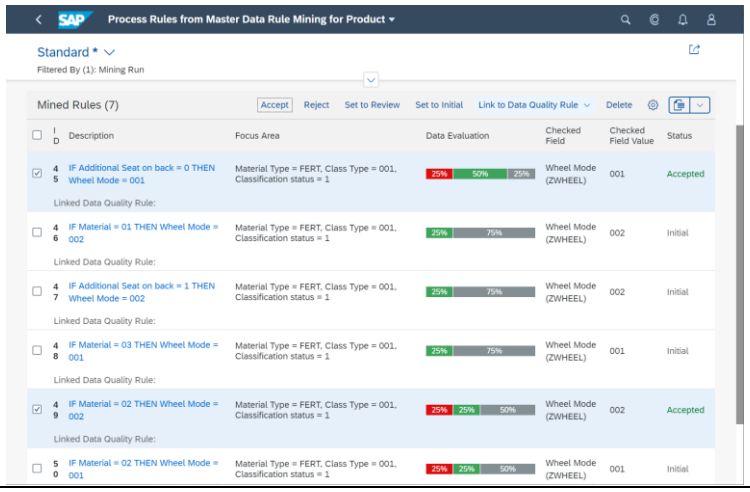
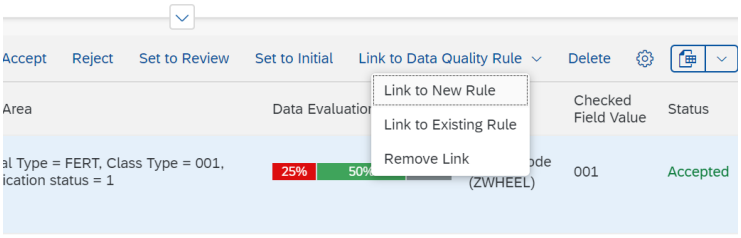
In the Mined Rule, the characteristics are part of rule as if they were normal attributes of the product.	

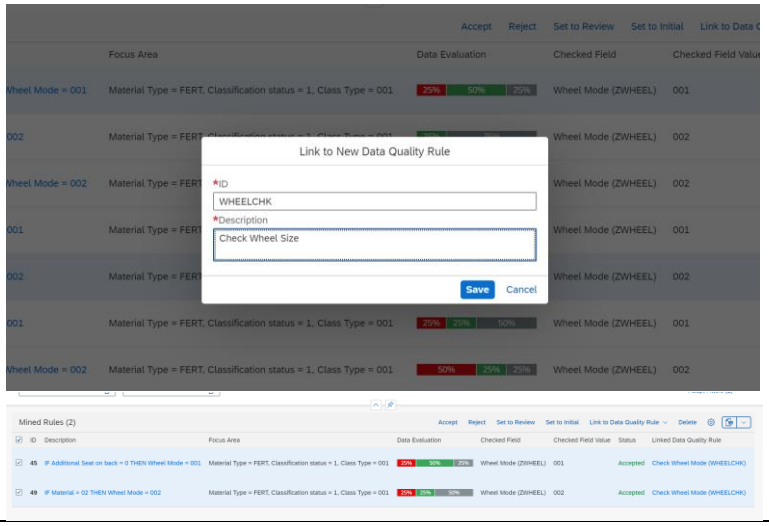
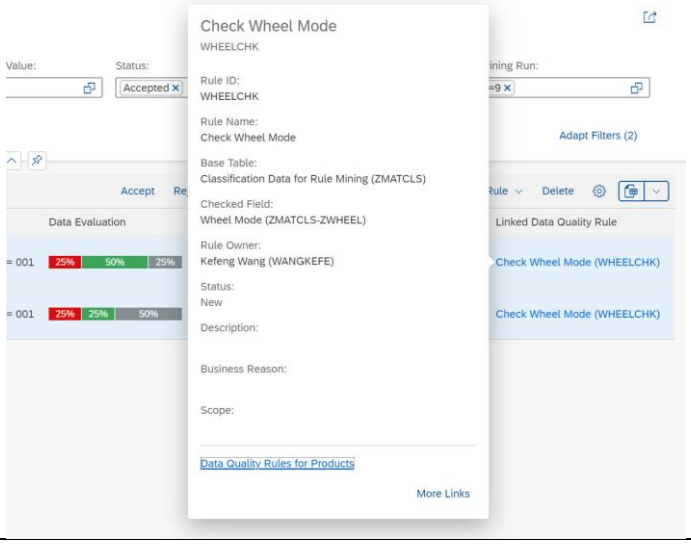
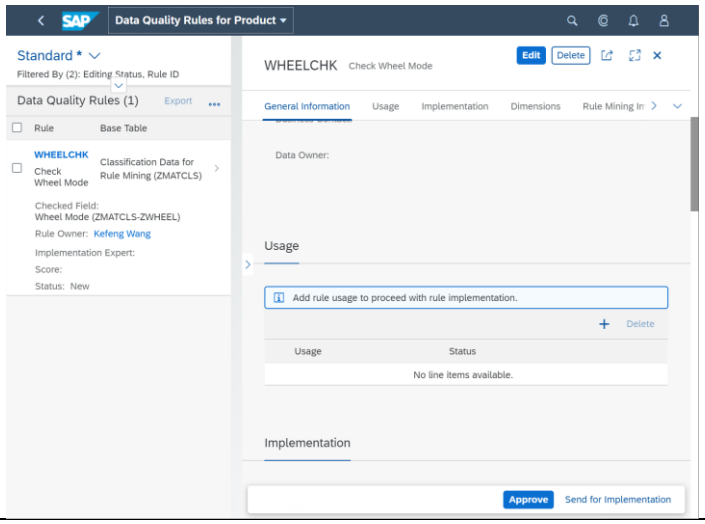
3.3. Data Quality Evaluation (optional)

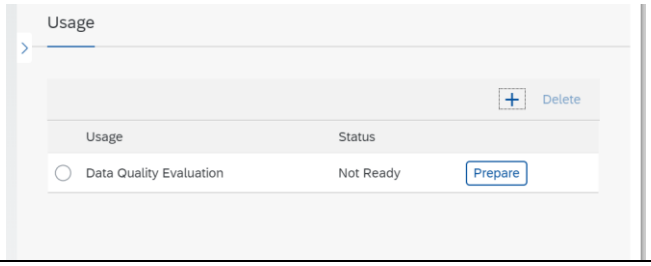
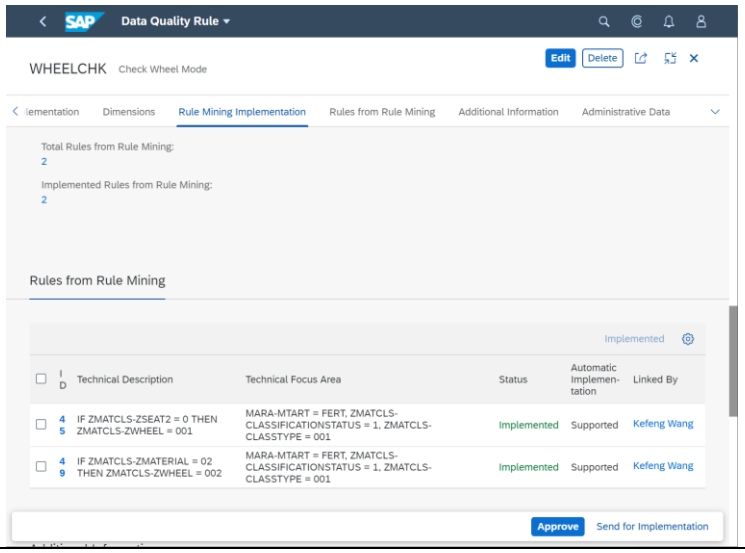
If you decide to take a mined rule as a data quality rule, you can use it for a data evaluation scenario. Firstly, we create data quality rule out of the mined rules, and then evaluate the data by using this rule.

3.3.1. Data Quality Rule Setup

Start from mined rule and create a data quality rule.

On the Mined Rule list, mark several rules that have the same checked field and approve them.	
Mark one or more approved rules, and choose *Link to Data Quality Rule -> Link to New Rule.* On the popup, enter a rule and a name.	

	
Go to the new Data Quality rule by choosing the Linked Data Quality Rule and choosing the link on the popover.	
Add the usage “Data Quality Evaluation” and choose Prepare. Now the system creates the BRFplus objects and decision tables by applying the mined rules. This could take several seconds.	

	
After completing these preparations, you should be able to see the Status ‘Implemented’ for the rules in the Rules from Rule Mining section. Now choose Approve.	
To edit the rule, go back the Usage section. You should be able to choose the Enable button on the Data Quality Evaluation.	
This data quality rule is ready for data quality evaluation	

3.3.2. Data Quality Evaluation

You can follow application help document to use this rule for data evaluation
https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/6d52de87aa0d4fb6a90924720a5b0549/0badfce1f7b041b2a7bccacda7e797f2.html