**Installation steps for Talend Studio 7.3.1**

Table of Contents

[1. Steps to set environment variables for JAVA: 2](file:///P:\MISDEV\BDAP\Talend\Installation%20steps%20Talend%20Studio%207.3.1.docx#_Toc40718522)

[2. Steps to Install Studio: 4](file:///P:\MISDEV\BDAP\Talend\Installation%20steps%20Talend%20Studio%207.3.1.docx#_Toc40718523)

[3. Connect to On-prem SQL Server using Active Directory. 10](file:///P:\MISDEV\BDAP\Talend\Installation%20steps%20Talend%20Studio%207.3.1.docx#_Toc40718524)

[4. Create personal access token for Azure Devops 10](file:///P:\MISDEV\BDAP\Talend\Installation%20steps%20Talend%20Studio%207.3.1.docx#_Toc40718525)

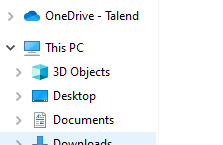
[5. Start Talend Studio for the first time 10](file:///P:\MISDEV\BDAP\Talend\Installation%20steps%20Talend%20Studio%207.3.1.docx#_Toc40718526)

# Steps to set environment variables for JAVA:

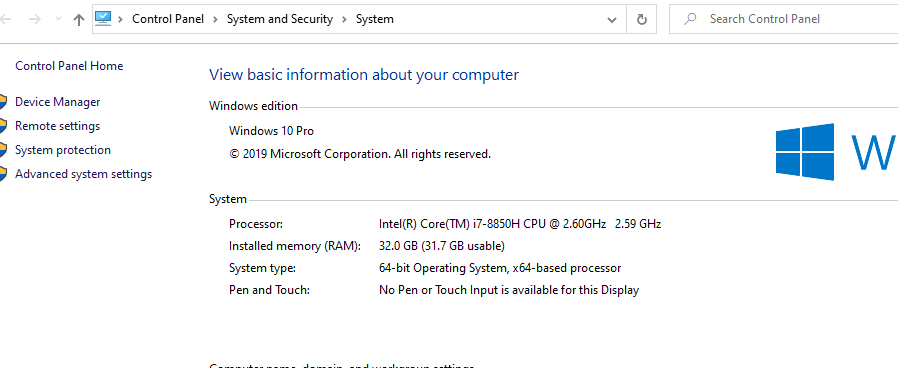
Download and install Java open JDK version 11, if you have not done already.

Make sure correct java version is installed on laptop and JAVA\_HOME and Path are set.

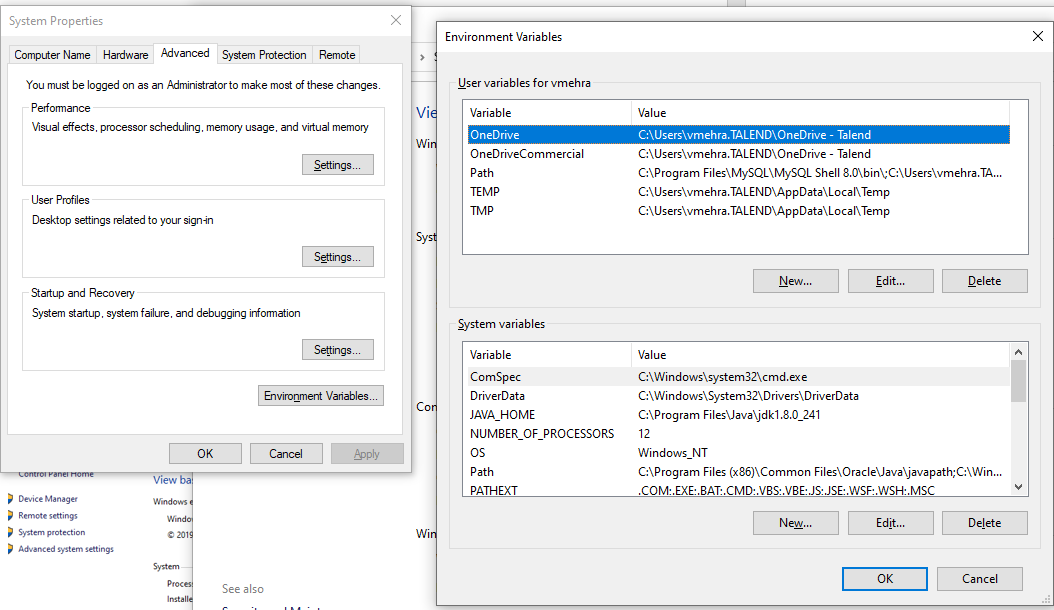
1. Right click on “This PC and click on Properties



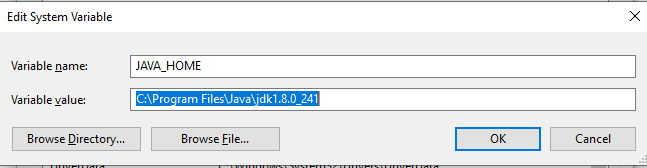
1. Click on Advanced system settings



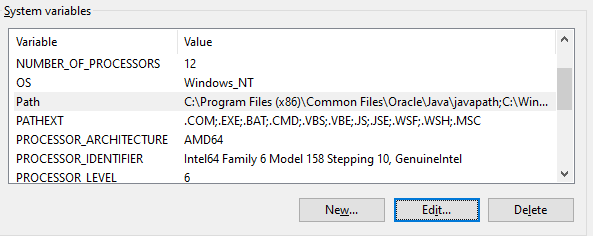
1. Click on Environment Variables and click on New under System Variables



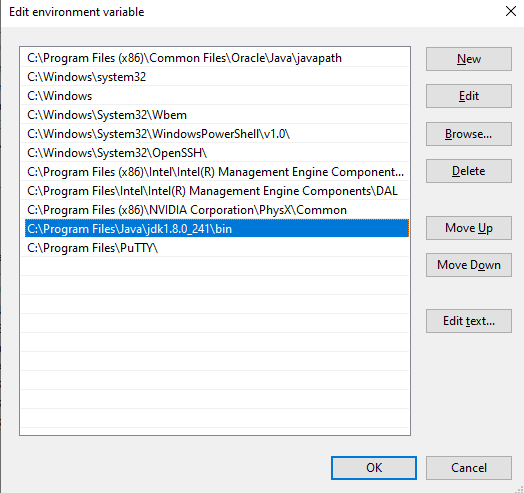
1. Add JAVA\_HOME and add the variable value as per your java version.



1. Select Path variable under System Variables and click on Edit

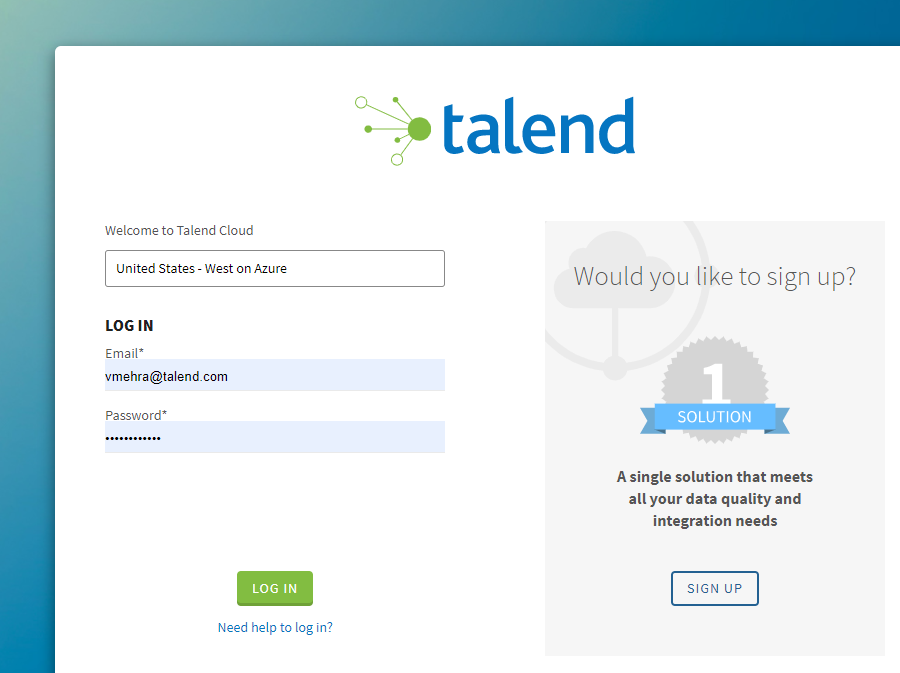


1. Add Java bin directory and click on OK to save

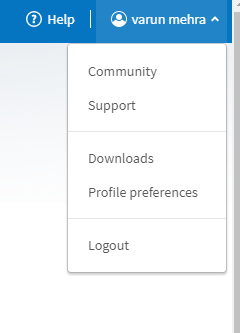


# Steps to Install Studio:

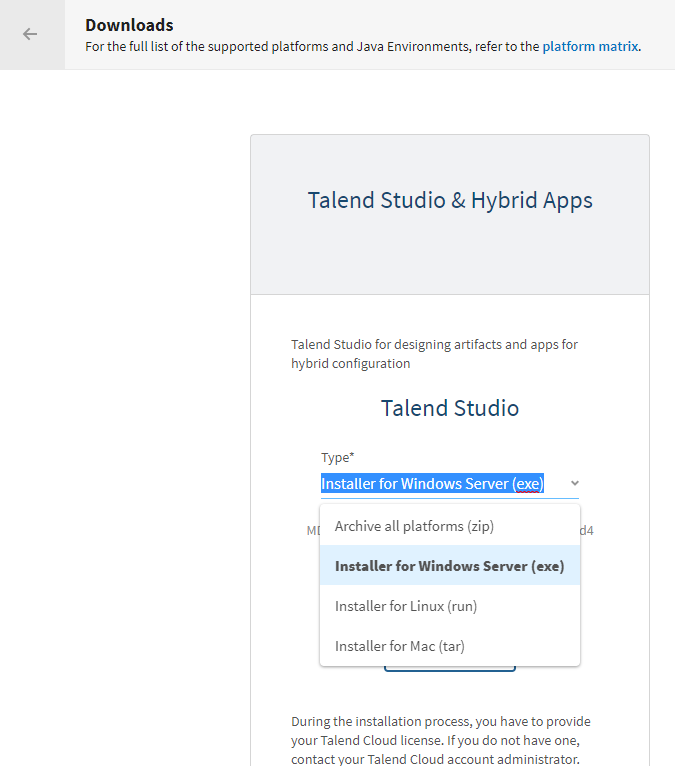
1. Click on the below link and select “Unites States – West on Azure”. Login to talend <https://iam.us-west.cloud.talend.com/idp/login>



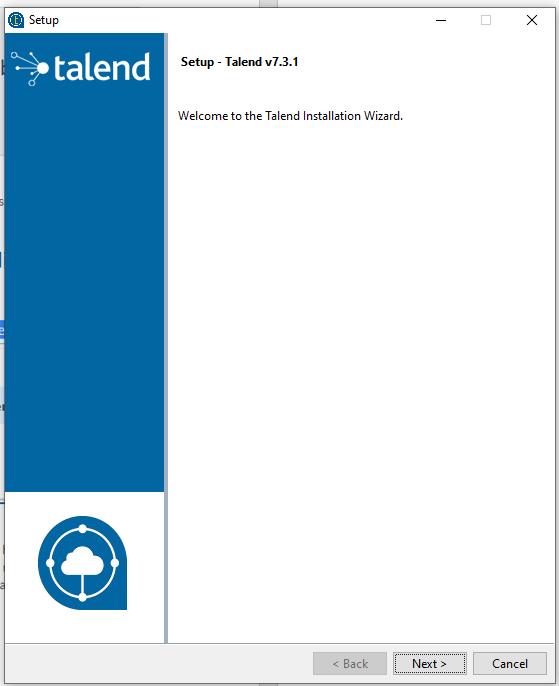
1. Click on the user name and select Downloads



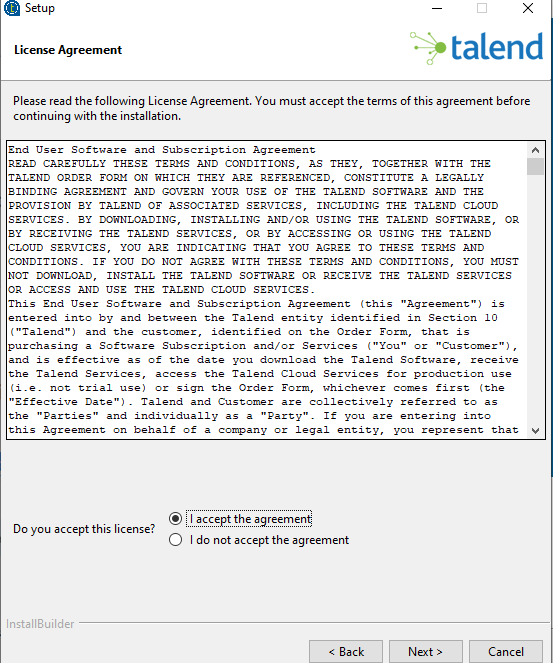
1. Under Talend Studio select Installer for Windows Server (exe).



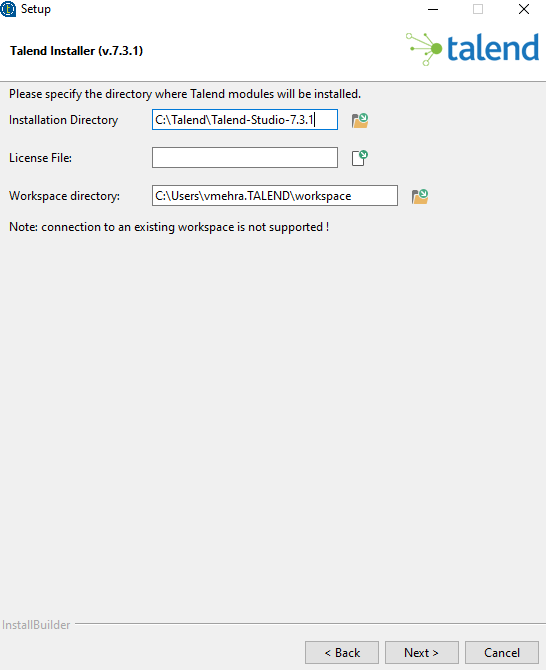
1. Right click the .exe file and click Run as Administrator.
2. Click Yes if you get a security pop-up from Microsoft defender.
3. Click on Next on the following screen



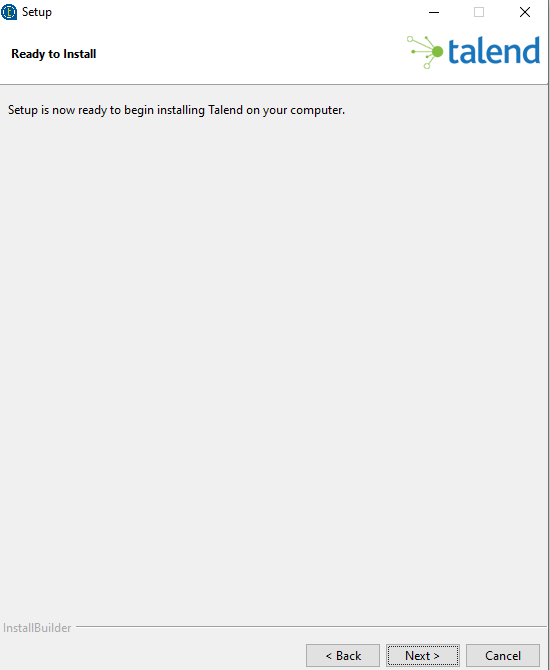
1. Click on radio button “I accept the agreement” and click on Next



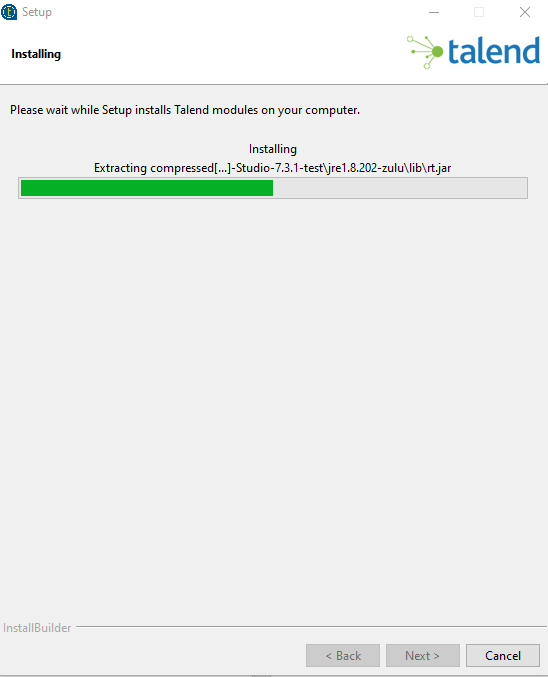
1. Provide the install directory, its best to have the install directory path simple like “C:\Talend\Talend-Studio-7.3.1” however talend doesnot restrict the user and any user defined path can be used.
2. Keep the License file as blank and select a workspace directory and click on next.

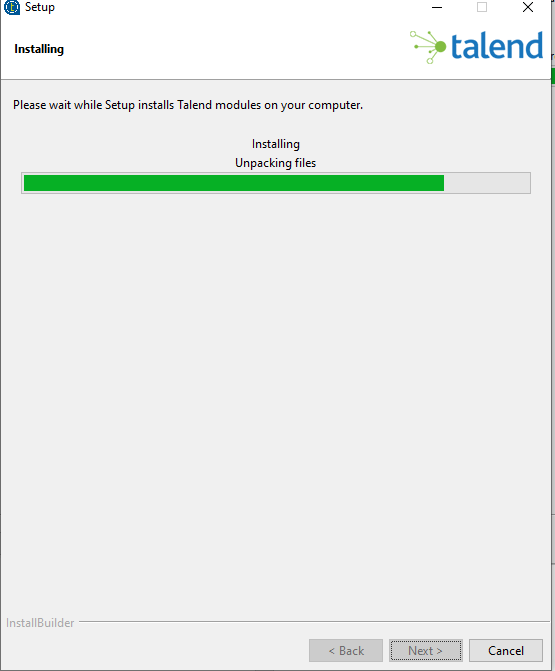


1. Click on Next on the following screen

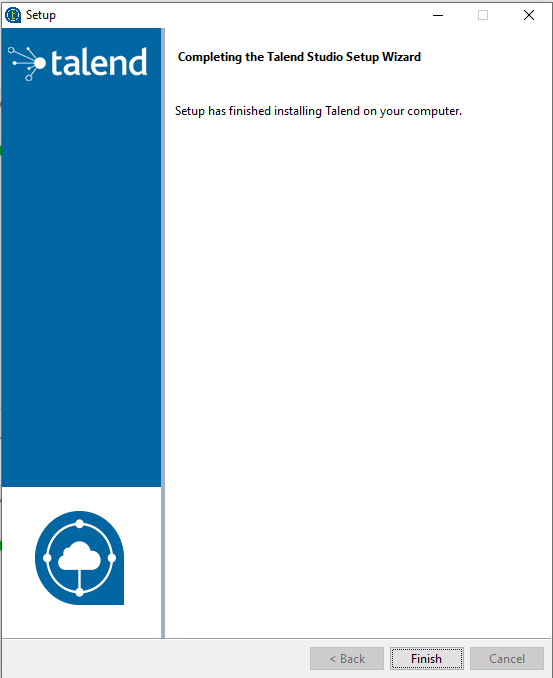


1. Talend Studio 7.3.1 should get installed in the installed directory





1. Click on Finish to complete the installation.



# Connect to On-prem SQL Server using Active Directory.

1. Download the JTDS driver, extract the contents, and find the ntlmauth.dll. The .ddl for x64 bit computer is attached below:



1. Move the ntlmauth.dll file to one (or both) of these locations: C:\Program Files\Java\JDK\_version\bin\ntlmauth.dll or the Windows folder (C:\Windows\System32\ntlmauth.dll).
2. Copy jtds-1.3.1.jar file to talend studio plugin folder i.e “C:\Talend\Talend-Studio-7.3.1\studio\plugins”



# Create personal access token for Azure Devops

1. Please follow the steps from the link below to generate the access token

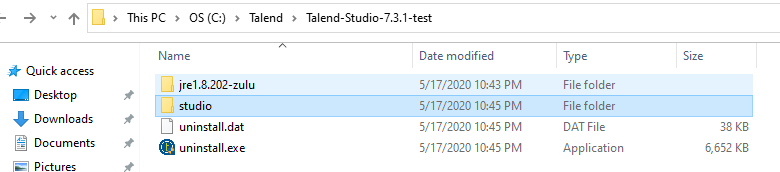
<https://docs.microsoft.com/en-us/azure/devops/organizations/accounts/use-personal-access-tokens-to-authenticate?view=azure-devops&tabs=preview-page>

1. Please copy the token and keep it safe we will need this when we try to start the studio and create a workspace that is linked to Azure repo.

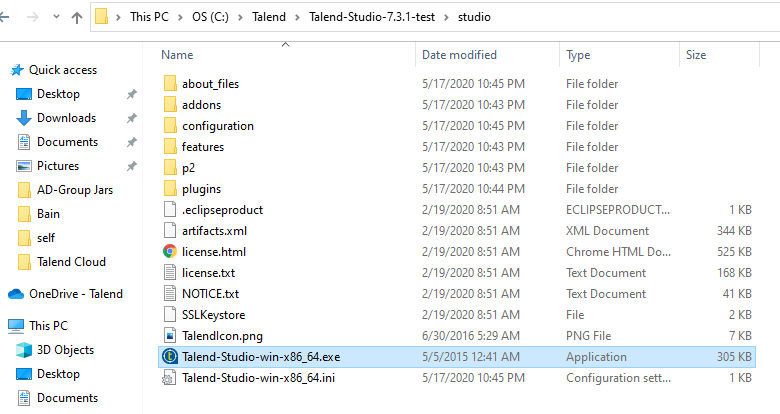
# Disable Netskope client in laptop (this is temporary till ISO team approves SSL for Talend Studio)

# Start Talend Studio for the first time

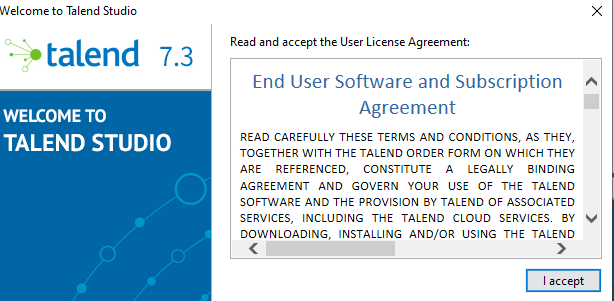
1. Click on “Studio” directory under the talend install directory



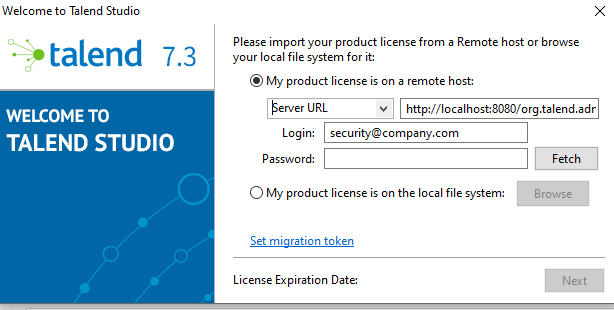
1. Double Click on Talend-Studio-win-x86\_64.exe to start Talend Studio 7.3.1.



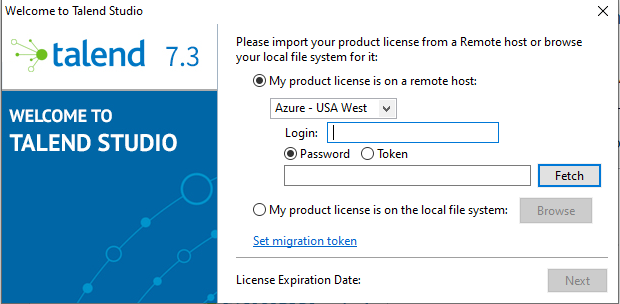
1. Click on “I Accept” on the welcome screen



1. To get the necessary license select the radio button “My product license is on a remote host”



1. From the Server URL drop down select “Azure – USA West” and enter the TMC login credentials under login and password and click on “Fetch”



1. Clink on “Next”, if there is repository associated with the project you will get another pop-up to provide repository credentials. Please note the repo can be on Git Hub or in this case its on Azure Devops.
2. Provide the Git login id and paste the token created in the previous step in pace of password.

**PIVOT CONcept:(table rows to columns)**

DECLARE @ColumnsTable TABLE ([ColumnName] VARCHAR(50));

INSERT INTO @ColumnsTable ([ColumnName])

SELECT DISTINCT '[' + CONVERT(VARCHAR(48), [questioncode]) + ']'

FROM PeopleAnalytics.factSurvey;

DECLARE @PivotColumns VARCHAR(MAX), @SQL VARCHAR(MAX)

SET @PivotColumns = (SELECT STUFF((SELECT DISTINCT ', ' + CONVERT(VARCHAR(50), [ColumnName])

FROM @ColumnsTable

FOR XML PATH('')), 1, 2, ''));

SET @SQL = 'SELECT \*

FROM (SELECT employeecode,response,questioncode,responseId

FROM PeopleAnalytics.factSurvey

where [surveyCode] IN ( ''SV\_cAuMLek39r8F0XA'' )) AS t

PIVOT (MAX([response])

FOR [questioncode] IN (' + @PivotColumns + ')) AS p;';

DECLARE @tbl AS TABLE

(

employeecode VARCHAR(MAX) NULL,

responseId VARCHAR(MAX) NULL,

Q1 VARCHAR(MAX) NULL,

Q11 VARCHAR(MAX) NULL,

Q12\_1 VARCHAR(MAX) NULL,

Q12\_2 VARCHAR(MAX) NULL,

Q12\_3 VARCHAR(MAX) NULL,

Q12\_5 VARCHAR(MAX) NULL,

Q12\_8 VARCHAR(MAX) NULL,

Q13\_10 VARCHAR(MAX) NULL,

Q13\_3 VARCHAR(MAX) NULL,

Q13\_4 VARCHAR(MAX) NULL,

Q13\_6 VARCHAR(MAX) NULL,

Q2 VARCHAR(MAX) NULL,

Q4 VARCHAR(MAX) NULL,

Q5\_1 VARCHAR(MAX) NULL,

Q5\_2 VARCHAR(MAX) NULL,

Q5\_3 VARCHAR(MAX) NULL,

Q5\_5 VARCHAR(MAX) NULL,

Q5\_6 VARCHAR(MAX) NULL,

Q5\_7 VARCHAR(MAX) NULL,

Q6 VARCHAR(MAX) NULL,

Q7 VARCHAR(MAX) NULL

)

INSERT INTO @tbl EXEC (@sql);

With CTE AS

(

SELECT

tbl.[responseId] AS [responseId],

dl.[employeeCode] AS [employeeCode]

,dl.[firstName] AS [firstName]

,dl.[lastName] AS [lastName]

,dl.capability AS [caseCapability]

,dl.[caseCode] AS [caseCode]

,dl.[caseManager] AS [caseManager]

,dl.[caseManagingOffice] AS [caseManagingOffice]

,dl.[caseOVP] AS [caseOVP]

,dl.departmentName AS [department]

,dl.[homeOffice] AS [homeOffice]

,dl.industry AS [caseIndustry]

,dl.[jobCategory] AS [jobCategory]

,dl.[operatingOffice] AS office

,dl.position AS [position]

,o.region AS [region]

,dl.serviceLine AS [serviceLine]

,dl.staffedVsUnstaffed AS [staffed]

,tbl.Q1 AS [NPSQuantitative]

,tbl.Q2 AS [caseNPS]

,tbl.Q12\_1 AS [inspiration1]

,tbl.Q12\_2 AS [inspiration2]

,tbl.Q12\_3 AS [inspiration3]

,tbl.Q12\_5 AS [inspiration4]

,tbl.Q12\_8 AS [inspiration5]

,CASE

WHEN LTRIM(RTRIM(tbl.Q4))='' THEN tbl.Q4

ELSE Substring(tbl.Q4,1,1)

END AS [capacity]

,tbl.Q13\_3 AS [connection1]

,tbl.Q13\_4 AS [connection2]

,tbl.Q13\_10 AS [connection3]

,tbl.Q13\_6 AS [connection4]

,tbl.Q5\_1 AS [LTsupport1]

,tbl.Q5\_2 AS [LTsupport2]

,tbl.Q5\_3 AS [LTsupport3]

,tbl.Q5\_6 AS [LTsupport4]

,tbl.Q5\_7 AS [LTsupport5]

,tbl.Q5\_5 AS [LTsupport6]

,CASE

WHEN LTRIM(RTRIM(tbl.Q6))='' THEN tbl.Q6

WHEN tbl.Q6 in ('N/A = I do not know') THEN '6'

ELSE Substring(tbl.Q6,1,1)

END AS [outlook]

,tbl.Q7 AS [improvementActions]

,getdate() AS [createddatetime]

FROM

[peopleAnalytics].[distributionList] dl

left join @tbl tbl on tbl.employeecode=dl.employeecode

left join dbo.office o on o.region=dl.region and hierarchyType='REV'

where dl.surveyId in (Select surveyId from PeopleAnalytics.dimsurvey where surveyCode='SV\_cAuMLek39r8F0XA')

)

INSERT INTO [PeopleAnalytics].[CLSUReporting]

from CTE

select @@ROWCOUNT AS INSERTCNT

**PIVOT: columns to rows**

CREATE PROCEDURE [peopleAnalytics].[spSurveyDetailsDynRefresh] @surveycode VARCHAR(50),@surveyname VARCHAR(100)

AS

BEGIN

/\* Declare dynamic Table name and Columns names to be as Pivot \*/

DECLARE

@table NVARCHAR(257) = N'staging.dynSurveyStageLoad',

@key\_column SYSNAME = N'ResponseID,ResponseSet,IPAddress,StartDate,EndDate,RecipientLastName,RecipientFirstName,RecipientEmail,ExternalDataReference';

/\* delcare dynamic Table\*/

DECLARE @tblStage AS TABLE

(

ResponseID VARCHAR(MAX) NULL, ResponseSet VARCHAR(MAX) NULL, IPAddress VARCHAR(MAX) NULL,

StartDate VARCHAR(MAX) NULL, EndDate VARCHAR(MAX) NULL, RecipientLastName VARCHAR(MAX) NULL,

RecipientFirstName VARCHAR(MAX) NULL, RecipientEmail VARCHAR(MAX) NULL, ExternalDataReference VARCHAR(MAX) NULL,

QCodes NVARCHAR(MAX) NULL,Response NVARCHAR(MAX) NULL

)

/\* Get Only Q Codes from Dyn Stage Table \*/

DECLARE

@sql NVARCHAR(MAX) = N'',

@cols NVARCHAR(MAX) = N'';

SELECT @cols += ', ' + QUOTENAME(name)

FROM sys.columns

WHERE [object\_id] = OBJECT\_ID(@table)

AND name not in( @key\_column) AND

RIGHT(name,1) LIKE '%[0-9]%' and

LEFT(name,1) = 'Q' ;

/\* Pivot the Dyn Stage Table Data\*/

SELECT @sql = N'SELECT ' + @key\_column + ',pivotColumn AS QCodes,pivotValue as Response

FROM

(

SELECT ' + @key\_column + @cols + '

FROM ' + @table + '

) AS cp

UNPIVOT

(

pivotValue FOR pivotColumn IN (' + STUFF(@cols, 1, 1, '') + ')

) AS up;' ;

INSERT INTO @tblStage EXEC sp\_executesql @sql;

/\* Get Strutured survey data from dyn Stage Table \*/

WITH cteqs AS

(

SELECT \* FROM @tblStage

)

INSERT INTO [staging].[survey] ([description],startdate,enddate,EmployeeCode,employeeemail,response,questionCode,surveyCode,surveyName,responseKey)

SELECT

ques.Response AS [description],res.startdate,res.enddate,res.ExternalDataReference ,

res.RecipientEmail,res.Response,res.QCodes,@surveycode,@surveyname,res.ResponseID

FROM cteqs res

LEFT JOIN cteqs ques ON ques.QCodes = res.QCodes

WHERE res.ResponseID != 'ResponseID' AND lower(res.ResponseID) NOT LIKE '%importid%'

AND ques.ResponseID = 'ResponseID';

/\* insert If New Survey found then capture into dimSurvey Table \*/

MERGE peopleanalytics.dimSurvey AS TARGET

USING etl.[surveyDetails] AS SOURCE

ON (

TARGET.surveyCode = SOURCE.surveyId AND

TARGET.[description] = SOURCE.surveyName

)

WHEN NOT MATCHED BY TARGET

THEN INSERT (

surveyCode,

[description]

) VALUES (

SOURCE.surveyId,

SOURCE.surveyName

) ;

/\* insert If New Survey details found then capture into dimQuestion Table \*/

MERGE peopleanalytics.dimQuestion AS TARGET

USING (

SELECT ds.surveyid,dq.[description],dq.surveyCode,questionCode FROM

(SELECT DISTINCT [description],surveyCode,questionCode FROM staging.survey) dq

LEFT JOIN peopleanalytics.dimSurvey ds ON ds.surveyCode = dq.surveyCode

) AS SOURCE

ON (

TARGET.surveyCode = SOURCE.surveyCode AND

TARGET.questionCode = SOURCE.questionCode AND

TARGET.[description] = SOURCE.[description] AND

TARGET.surveyid = SOURCE.surveyid

)

WHEN NOT MATCHED BY TARGET

THEN INSERT (

surveyid,

[description],

surveyCode,

questionCode

) VALUES (

SOURCE.surveyid,

SOURCE.[description],

SOURCE.surveyCode,

SOURCE.questionCode

) ;

**MASKING Steps:**

* Created a table [testing].[employeeMask]

CREATE TABLE [testing].[employeeMask]

(

       [employeeCode] [nvarchar](255) NULL,

       [employeeName] [nvarchar](255) NULL,

       [employeeStatus] [nvarchar](255) NULL,

       [departmentName] [nvarchar](255) NULL,

       [Level] [nvarchar](255) NULL

)

* Loaded Data to table
* Added masking functions to the columns of the table

ALTER TABLE [testing].[employeeMask]

ALTER COLUMN [employeeName] [nvarchar](255) MASKED WITH (FUNCTION = 'default()');

GO

ALTER TABLE [testing].[employeeMask]

ALTER COLUMN [employeeStatus] [nvarchar](255) MASKED WITH (FUNCTION = 'default()');

GO

ALTER TABLE [testing].[employeeMask]

ALTER COLUMN [departmentName] [nvarchar](255) MASKED WITH (FUNCTION = 'default()');

GO

ALTER TABLE [testing].[employeeMask]

ALTER COLUMN [Level] [nvarchar](255) MASKED WITH (FUNCTION = 'default()');

**Other Masking Options**

MASKED WITH (FUNCTION = 'Email()');

MASKED WITH (FUNCTION='random(1,9)');

MASKED WITH (FUNCTION= 'partial(3,"XXXX",3)');

* Below Query will list out all the tables and columns with their masking details.

SELECT TBLS.NAME           AS TableName,  
       MC.NAME             ColumnName,  
       MC.is\_masked        IsMasked,  
       MC.masking\_function MaskFunction  
FROM   sys.masked\_columns AS MC  
       JOIN sys.tables AS TBLS  
         ON MC.object\_id = TBLS.object\_id  
WHERE  is\_masked = 1;

* Grant permission to Unmask to User

GRANT UNMASK TO [Alison.Lawler@bain.com]

**For Revoking Unmask**

REVOKE UNMASK TO [Alison.Lawler@bain.com]

* Impersonate a user to see if the masking is working Fine

Execute as USER = 'alisa.forney@bain.com'

**For Reverting to your userid use below command**

Revert

* All the Administrators By default have Unmask ability. Apart from the Administrators users with **db\_owner** role in database have unmask ability by default. Below query will give all the **db\_owner** in the database.

SELECT DP1.name AS DatabaseRoleName,  
isnull (DP2.name, 'No members') AS DatabaseUserName  
FROM sys.database\_role\_members AS DRM  
RIGHT OUTER JOIN sys.database\_principals AS DP1  
ON DRM.role\_principal\_id = DP1.principal\_id  
LEFT OUTER JOIN sys.database\_principals AS DP2  
ON DRM.member\_principal\_id = DP2.principal\_id  
WHERE DP1.type = 'R' and DP1.name = 'db\_owner'  
ORDER BY 2

* Below Query give the data of all the users who have Unmask ability other than the **administrators** and **db\_owners**.

SELECT [princ.NAME](http://princ.name/),  
       princ.type\_desc,  
       perm.permission\_name,  
       perm.state\_desc,  
       perm.class\_desc,  
       Object\_name(perm.major\_id)  
FROM   sys.database\_principals princ  
       LEFT JOIN sys.database\_permissions perm  
              ON perm.grantee\_principal\_id = princ.principal\_id  
WHERE  perm.permission\_name = 'UNMASK'

In essence, one 3 user categories would be able to unmask the data -

* DB Administrator
* users with db\_owner role on the database
* users given specific permission to unmask data by an Administrator

Only DB Adminstrators can execute the query as a different user. There are no other ways to bypass this security.

If a user has access to unmask data, they can see it unmasked on Tableau as well. However, if the user logged into Tableau does not have permission to view unmasked data, then they can only see it as masked on Tableau as well.

**TEST cases:**

Env - QA ODS

DataFlow → HistoricalEmployeeTransactions → employeeHistoryMonthly

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test S.No** | **Test Description** | **Test Steps** | **Expected Result** | **Actual Result** | **Defects** |
| **Test S.No** | **Test Description** | **Test Steps** | **Expected Result** | **Actual Result** | **Defects** |
| 1. | Validate for every Active ecode & employeename a snapshot is taken for 1st and 15th of every month. | 1. Login into QA ODS. 2. Validate ecode & employeename is added to the table. 3. Validate for all active employee a snapshot is taken on every 1st and 15th of every month. | For active ecode, effectiveyear and effective month snapshot should be present in the table on 1st and 15th of each month. |  |  |
| 2. | Validate if effective date is same for records then completedDate is considered to bring in the latest record for that time duration. | 1. Login into QA ODS. 2. Validate based on the latest effective date records are pulled into the table. 3. Validate if the effective date is same for the records then completed date is considered. | If effective date is same for records then completed date should be considered. |  |  |
| 3. | Check for the WWELL, WWBAI, WSANT, 40568, 40565,40564 ecode in the table | 1. Login into QA ODS. 2. Validate the sample ecode and check for the data populated from HistoricalEmployeeTransactions table. | for the given ecode the business logic should work as expected |  |  |
| 4. | Validate DEI Flag & average cal are deleted from the table. | 1. Login into QA ODS. 2. Validate that DEI flag is deleted from the table. | DEI flag should be deleted from the table. |  |  |
| 5. | Validate TerminationCategorization are added to the table. | 1. Login into QA ODS. 2. Validate TerminationCategorization is added to the table from vwAttritionDetail. | TerminationCategorization should be added to the table from vwAttritionDetail |  |  |
| 6 | Validate employeetype is added in the table | 1. Login into QA ODS. 2. Validate employeetype is added from historicalemployeetransaction table. | Employeetype is added from historicalemployeetransaction table. |  |  |
| 7. | Validate Termination Code, Rating Code and Categorization Code are deleted from the table | 1. Login into QA ODS. 2. Validate Termination Code, Rating Code and Categorization Code are deleted from the table. | Termination Code, Rating Code and Categorization Code should be deleted from the table. |  |  |
| 8. | Validate all the active employee from dbo.employee tables are present in the table. | 1. Login into QA ODS. 2. Validate all the active employees from dbo.employee table are present in the table. | All the active employees should have a record in table. |  |  |

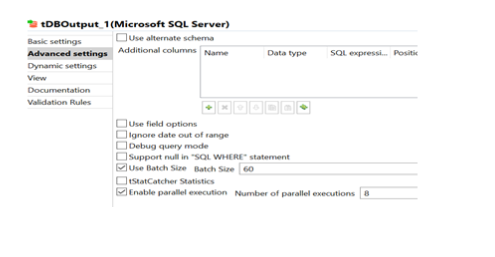
**Note -**

44375,40970 and 36408 for ecode as effectivedate,completed Date and last modified dates are same, initiated date will be added in future sprint to handle such scenario.

**Tuning Methods with Huge data processing in Talend**

**Parallel Execution:**

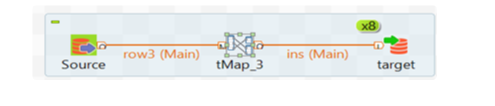
·       Select this check box to perform high-speed data processing, by treating multiple data flows simultaneously. Note that this feature depends on the database or the application ability to handle multiple inserts in parallel as well as the number of CPU affected.



**DB to DB Data Insertion with Parallel process:**

·       Extracts data from the table based on an SQL query

·       Writes data it receives from the preceding component into a table of an MSSQL database.

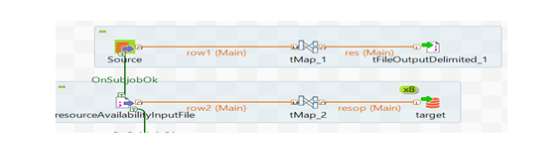


**DB to File Extraction followed by Data Insertion to DB:**

·       Extracts data from the table based on an SQL query and storing into a csv file format.

·       And observation is here to storing is bit faster and extracting from file as well.

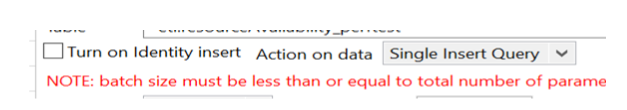
·       Writes data it receives from the preceding component into a table of an MSSQL database.



**Data Insertion Tuning:**

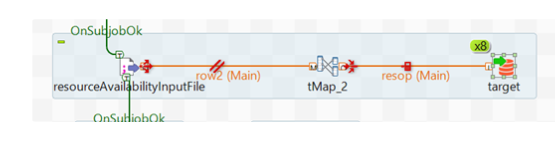
·       Add entries to the table in a batch.

·       And Single Insert Query option will allow max 2200 parameters.



**Partitions on Insertion process:**

·       Parallelization of data flows means to partition an input data flow of a subJob into parallel processes and to simultaneously execute them, so as to gain better performance. These processes are executed always in a same machine



The implementation of the parallelization requires four key steps as explained as follows:

1. Partitioning

: In this step, the Studio splits the input records into a given number of threads.

1. Collecting

: In this step, the Studio collects the split threads and sends them to a given component for processing.

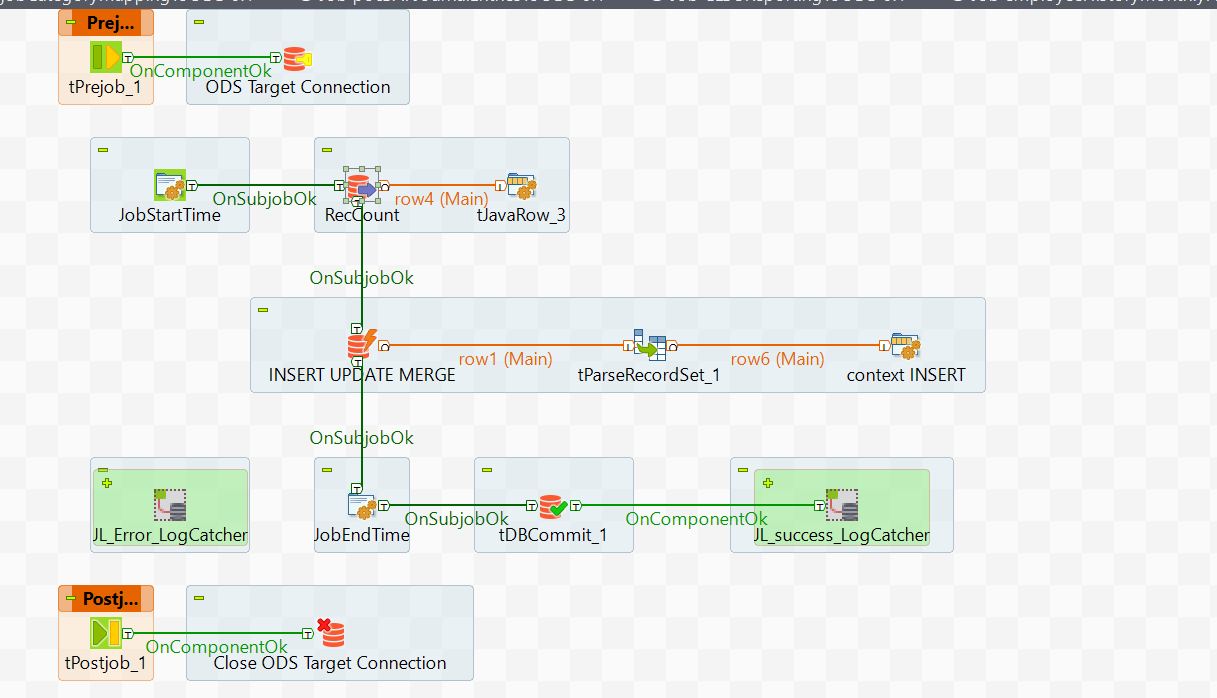
1. De-partitioning

: In this step, the Studio groups the outputs of the parallel executions of the split threads.

1. Recollecting

: In this step, the Studio captures the grouped execution results and outputs them to a given component.

JOB Sample:



**Sample code:**

**--select \* from [peopleAnalytics].[vwReviewWithRatings] where RevieweeEmployeeCode ='39321'**

**----drop view if exists [peopleAnalytics].[vwReviewWithRatings]**

**--select count(\*) from [peopleAnalytics].[vwReviewWithRatings] where RevieweeEmployeeCode ='01MPY'and ReviewPeriodName='Fall 2020 cycle (Boston)'**

**--and review\_id='AD26B131-A619-4519-9AE3-97EAAE14D275' and ratingName='Operating principles'**

**----RevieweeEmployeeCode ='01MPY' and ReviewerEmployeeCode='12YGU'**

**----2331857**

**--2349094**

**----11638850**

**CREATE VIEW [peopleAnalytics].[vwReviewWithRatings]**

**AS**

**WITH empHistory AS**

**(**

**select**

**employeeCode,**

**pdGradeProposed,**

**effectiveDate,**

**tenureCalcEndDate,**

**ROW\_NUMBER() OVER (PARTITION BY employeeCode,effectiveDate ORDER BY lastModifiedDate DESC) rn**

**FROM Workday.historicalEmployeeTransactions**

**),**

**emp as**

**(SELECT employeeCode,**

**employeeName,**

**startDate,**

**ROW\_NUMBER() OVER(PARTITION BY employeecode ORDER BY startdate DESC) rn**

**FROM dbo.employee**

**)**

**SELECT**

**RWR.review\_id,**

**RWR.RevieweePeriodOffice,**

**RWR.ReviewPeriodName,**

**RWR.ReviewPeriodStartDate,**

**RWR.ReviewPeriodEndDate,**

**RWR.RevieweeEmployeeCode,**

**RWR.RevieweeName,**

**RWR.ReviewTypeName,**

**RWR.CaseCode,**

**RWR.ReviewerEmployeeCode,**

**RWR.ReviewerName,**

**RWR.CurrentReviewStatusName,**

**RWR.ReviewStatusDate,**

**RWR.RatingName,**

**RWR.Rating,**

**RWR.CaseStartDate,**

**RWR.CaseEndDate,**

**RWR.officeCode,**

**RWR.[current\_Reviewer\_L/G],**

**RWR.[current\_Reviewee\_L/G],**

**RWR.[PIT\_Reviewee\_Level/Grade],**

**RWR.[PIT\_Reviewer\_Level/Grade],**

**RWR.reviewOfficeName,**

**RWR.ClientName,**

**RWR.Comments,**

**pl.pd\_level\_name AS Expected\_PD\_LevelName\_PIT**

**FROM (SELECT rm.review\_id,**

**o.officeName AS RevieweePeriodOffice,**

**rp.review\_period\_name AS ReviewPeriodName,**

**rp.review\_period\_start\_date AS ReviewPeriodStartDate,**

**rp.review\_period\_end\_date AS ReviewPeriodEndDate,**

**rm.employee\_code AS RevieweeEmployeeCode,**

**e.employeeName AS RevieweeName,**

**rt.review\_type\_name AS ReviewTypeName,**

**CASE**

**WHEN rt.review\_type\_code IN ( '101', '102', '201', '202' ) THEN CONVERT (VARCHAR, cm.oldCaseCode)**

**ELSE 'no case code'**

**END AS CaseCode,**

**reviewer\_employee\_code AS ReviewerEmployeeCode,**

**eer.employeeName AS ReviewerName,**

**rs.review\_status\_description AS CurrentReviewStatusName,**

**rm.review\_status\_date AS ReviewStatusDate,**

**fit.form\_item\_type\_name AS RatingName,**

**fiv.form\_item\_value AS Rating,**

**cm.startDate AS CaseStartDate,**

**cm.endDate AS CaseEndDate,**

**o.officecode AS officeCode,**

**ess.pdGrade AS [current\_Reviewer\_L/G],**

**es.pdGrade AS [current\_Reviewee\_L/G],**

**het.pdGradeProposed AS [PIT\_Reviewee\_Level/Grade],**

**het1.pdGradeProposed AS [PIT\_Reviewer\_Level/Grade]**

**,CASE**

**WHEN review\_status\_description = 'Pending' AND**

**review\_type\_name IN ('Principal Professional Development Review', 'Professional Development Review')**

**THEN o.officeName**

**ELSE NULL**

**END AS reviewOfficeName,**

**CASE**

**WHEN review\_status\_description = 'Pending' AND**

**review\_type\_name IN ('Assoc. Partner Case Review', 'Case Review')**

**THEN cm.clientname**

**ELSE NULL**

**END AS ClientName,**

**CASE**

**WHEN review\_status\_description = 'Pending' AND**

**review\_type\_name IN ('Assoc. Partner Case Review', 'Case Review')**

**THEN rm.comments**

**ELSE NULL**

**END AS Comments**

**FROM hcpd.review\_master rm WITH (nolock)**

**LEFT JOIN hcpd.employee\_ratings er WITH (nolock)**

**ON rm.review\_id = er.review\_id**

**LEFT JOIN revenue.caseWithConfidential cm WITH (nolock)**

**ON rm.case\_code = cm.caseCode**

**AND rm.client\_code = cm.clientCode**

**LEFT JOIN hcpd.form\_item\_type fit WITH (nolock)**

**ON er.form\_item\_type\_code = fit.form\_item\_type\_code**

**LEFT JOIN hcpd.review\_type rt WITH (nolock)**

**ON rm.review\_type\_code = rt.review\_type\_code**

**LEFT JOIN hcpd.review\_period rp WITH (nolock)**

**ON rp.review\_period\_code = rm.review\_period\_code**

**LEFT JOIN dbo.office o**

**ON rp.review\_period\_office\_code = o.officecode**

**AND o.hierarchyType = 'REV'**

**LEFT JOIN hcpd.review\_status rs WITH (nolock)**

**ON rs.review\_status\_code = rm.review\_status\_code**

**LEFT JOIN hcpd.form\_item\_values fiv WITH (nolock)**

**ON er.form\_item\_type\_code = fiv.form\_item\_type\_code**

**AND er.form\_item\_value = fiv.form\_item\_value\_code**

**LEFT JOIN [peopleAnalytics].[employeeSensitive] es**

**ON rm.employee\_code=es.employeeCode**

**LEFT JOIN [peopleAnalytics].[employeeSensitive] ess**

**ON rm.reviewer\_employee\_code=ess.employeeCode**

**LEFT JOIN empHistory het**

**ON rm.employee\_code=het.employeeCode**

**AND rm.review\_status\_date BETWEEN het.effectivedate AND het.tenurecalcenddate**

**AND het.rn=1**

**LEFT JOIN empHistory het1**

**ON rm.reviewer\_employee\_code=het1.employeeCode**

**AND rm.review\_status\_date BETWEEN het1.effectivedate AND het1.tenurecalcenddate**

**AND het.rn=1**

**LEFT JOIN emp e**

**ON rm.employee\_code = e.employeecode**

**AND e.rn = 1**

**LEFT JOIN emp eer**

**ON eer.employeeCode = rm.reviewer\_employee\_code**

**AND eer.rn = 1**

**WHERE rs.review\_status\_description NOT IN ('Inactive','Initiated', 'Active','Pending')**

**or (review\_type\_name NOT IN ('Asset Building','Asset Building Assessment','PDR kick-off','Self Assessment') and**

**review\_status\_description = 'Pending')**

**)RWR**

**LEFT JOIN (SELECT CONCAT(level\_code,'',grade\_code) AS PDGrade,pd\_level FROM [HCPD].[employee\_PD\_level]) epl**

**ON LEFT(RWR.[PIT\_Reviewee\_Level/Grade],CHARINDEX('-',RWR.[PIT\_Reviewee\_Level/Grade])-1)=epl.PDGrade**

**LEFT JOIN [HCPD].[pd\_level] pl**

**ON epl.pd\_level=pl.pd\_level**

**GO**

**DROP TABLE IF EXISTS #archive**

**CREATE TABLE #archive**

**(**

**ActionType VARCHAR(50),**

**employeeCode VARCHAR(20),**

**effectiveMonth int,**

**effectiveYear int,**

**day int**

**);**

**MERGE peopleanalytics.employeeHistoryMonthly AS TARGET**

**using (SELECT employeeCode,**

**employeeName,**

**effectiveMonth,**

**effectiveYear,**

**[day],**

**terminationCategorization,**

**positionnameproposed AS position,**

**departmentProposed AS department,**

**jobCategoryProposed AS jobCategory,**

**pdGrade,**

**businessProcessName,**

**leadershipTag,**

**seniorLeadershipTag,**

**homeOfficeCodeProposed AS homeOfficeCode,**

**employeeTypeProposed AS employeeType**

**FROM**

**(SELECT etran.employeecode,**

**emp.employeeName,**

**Month(cal.[date]) AS effectiveMonth,**

**Year(cal.[date]) AS effectiveYear,**

**Day(cal.[date]) AS [day],**

**termRes.categorization AS terminationCategorization,**

**etran.positionnameproposed,**

**LEFT(etran.pdgradeproposed, Charindex('-', etran.pdgradeproposed) - 1) AS pdGrade,**

**etran.businessprocessname,**

**etran.homeOfficeCodeProposed,**

**etran.departmentProposed,**

**etran.jobCategoryProposed,**

**etran.businessprocessreason,**

**etran.employeeTypeProposed,**

**lg.leadershipTag,**

**lg.seniorLeadershipTag,**

**etran.completedDate,**

**ROW\_NUMBER()**

**OVER (**

**partition BY etran.employeecode, Month(cal.[date]), Year(cal.[date]), Day(cal.[date])**

**ORDER BY etran.effectiveDate DESC, etran.completedDate DESC ) rnk**

**FROM (SELECT [date],**

**CASE**

**WHEN Day([date]) = 1 THEN DATEFROMPARTS (Year([date]), Month([date]), 1)**

**WHEN Day([date]) = 15 THEN DATEFROMPARTS (Year([date]), Month([date]), 15)**

**END AS PeriodStartDate,**

**CASE**

**WHEN Day([date]) = 1 THEN DATEFROMPARTS (Year([date]), Month([date]), 14)**

**WHEN Day([date]) = 15 THEN EOMONTH(date)**

**END AS PeriodEndDate**

**FROM dbo.[calendar]**

**WHERE Day([date]) IN ( 1, 15 )) cal**

**LEFT JOIN [workday].[historicalEmployeeTransactions] etran**

**ON ( cal.date >= etran.effectiveDate AND cal.date <= etran.tenureCalcEndDate )**

**LEFT JOIN (SELECT DISTINCT pdgrade,**

**leadershiptag,**

**seniorleadershiptag**

**FROM workday.levelgradeprecedence) lg**

**ON LEFT(etran.pdgradeproposed, Charindex('-', etran.pdgradeproposed) - 1) = lg.pdgrade**

**LEFT JOIN [peopleAnalytics].[terminationReasonType] termRes**

**ON TRIM(Substring(etran.businessprocessreason, Charindex('>', etran.businessprocessreason)**

**+ 1, 250)) = termRes.terminationReason**

**LEFT JOIN (SELECT DISTINCT employeeCode,**

**employeeName**

**FROM dbo.employee) emp**

**ON etran.employeeCode = emp.employeeCode**

**WHERE**

**etran.workdayId IS NOT NULL**

**AND (etran.businessProcessReason NOT LIKE ('Educational%') OR etran.businessProcessReason is NULL)**

**AND etran.employeeTypeProposed = 'Regular'**

**AND etran.businessProcessType NOT LIKE ('%Contingent%')**

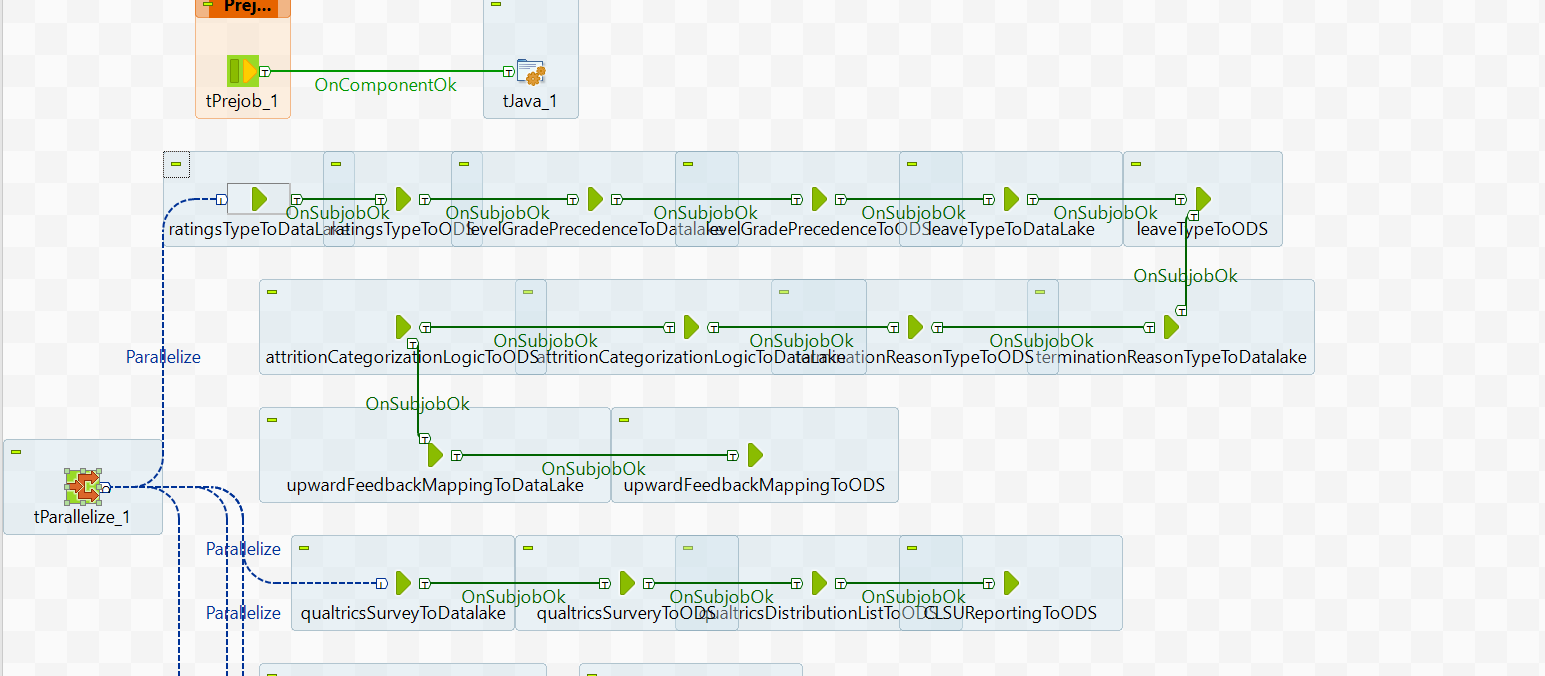
**AND etran.businessprocesstype != 'Termination'**

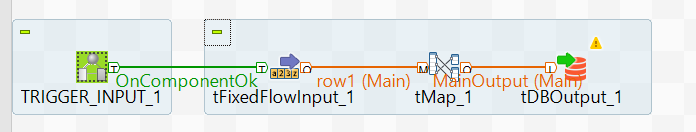
**AND etran.attritionInclusionFlag != 'Exclude'**

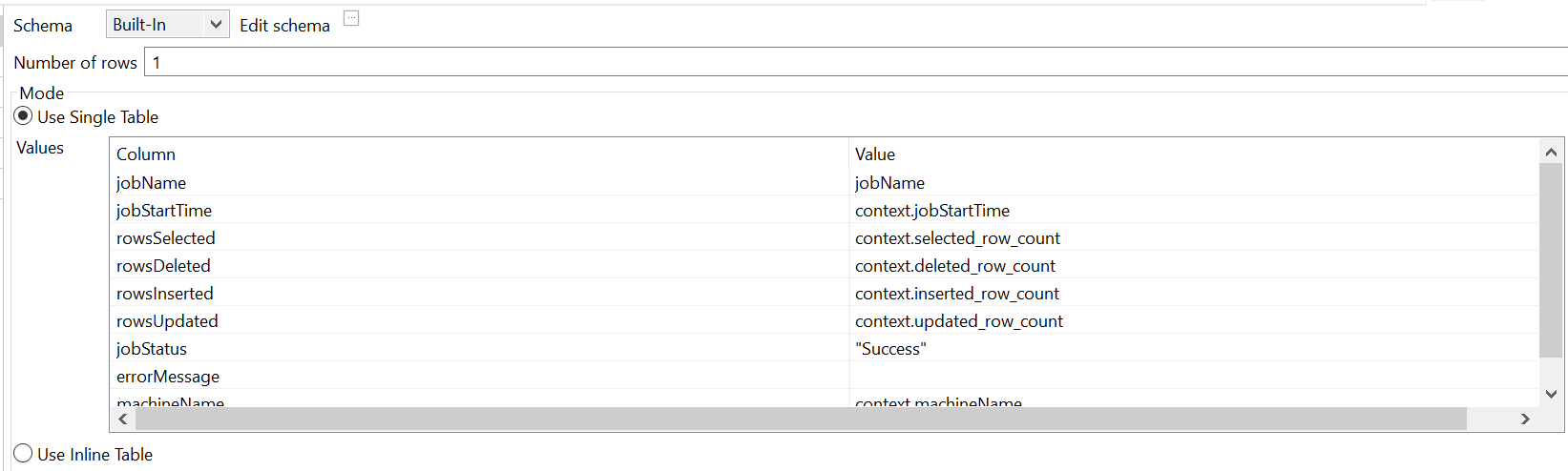
**) qry**

**WHERE qry.rnk = 1)**

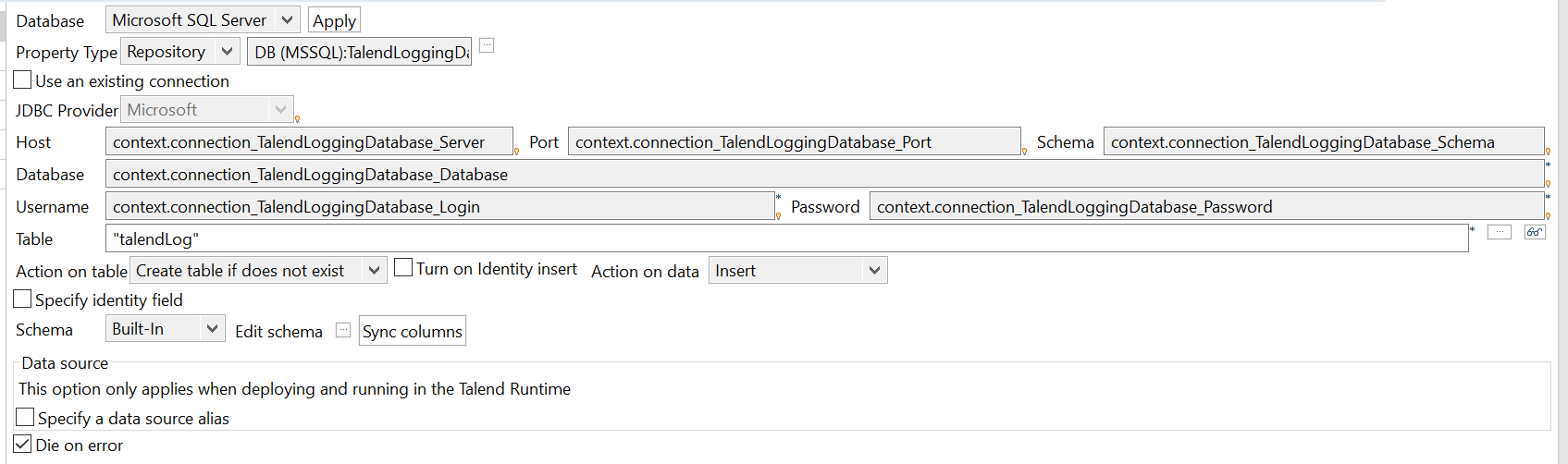
**Master sample:**

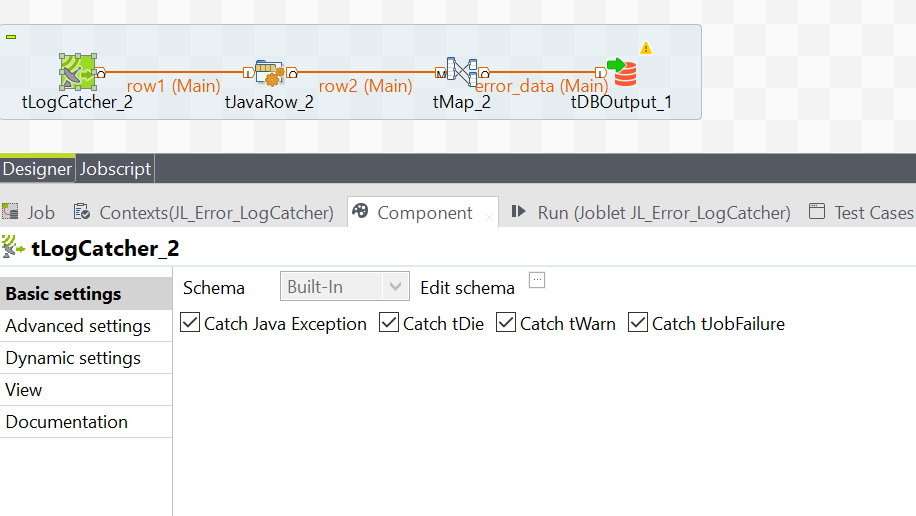


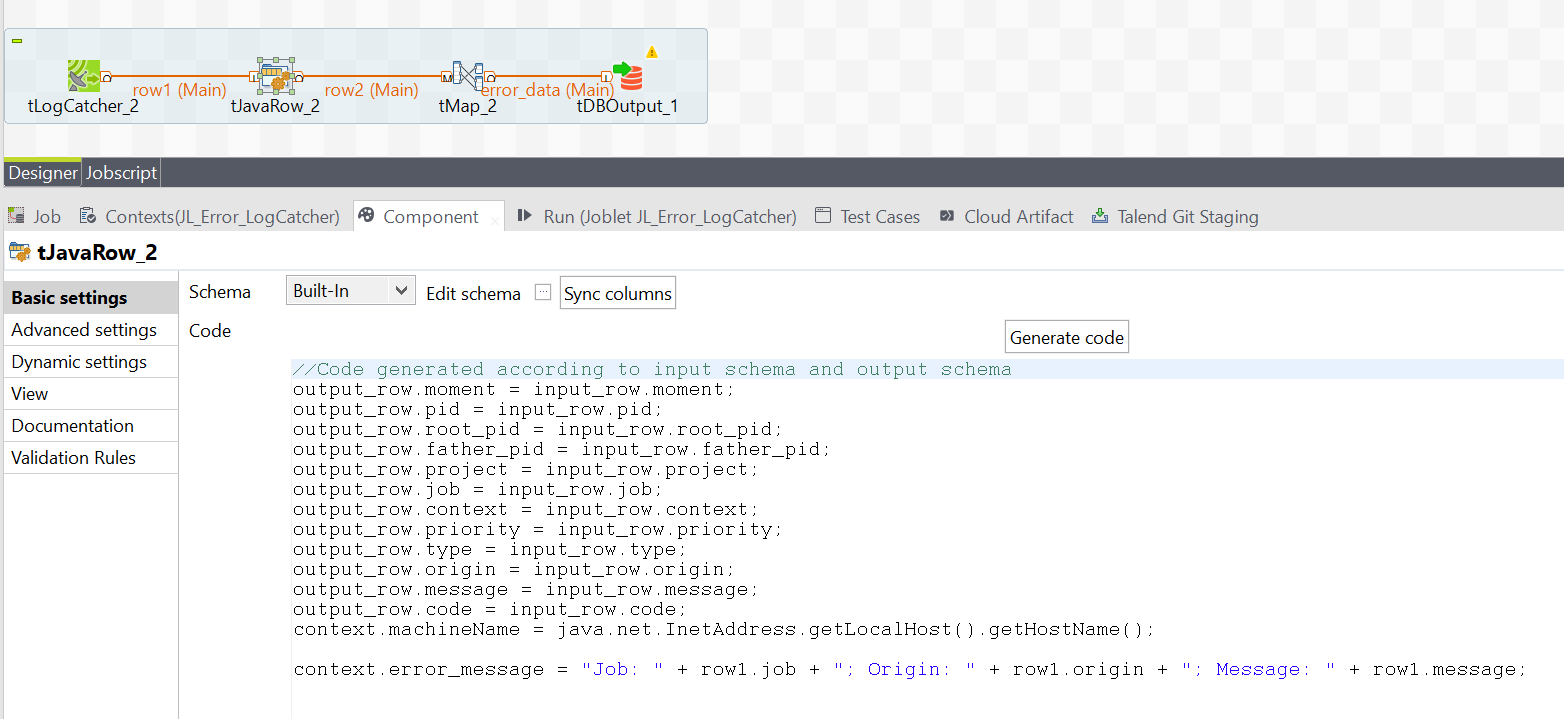




TalendDate.parseDate("dd-MM-yyyy HH:mm:ss",TalendDate.getDate("dd-MM-yyyy HH:mm:ss"))—jobenddate

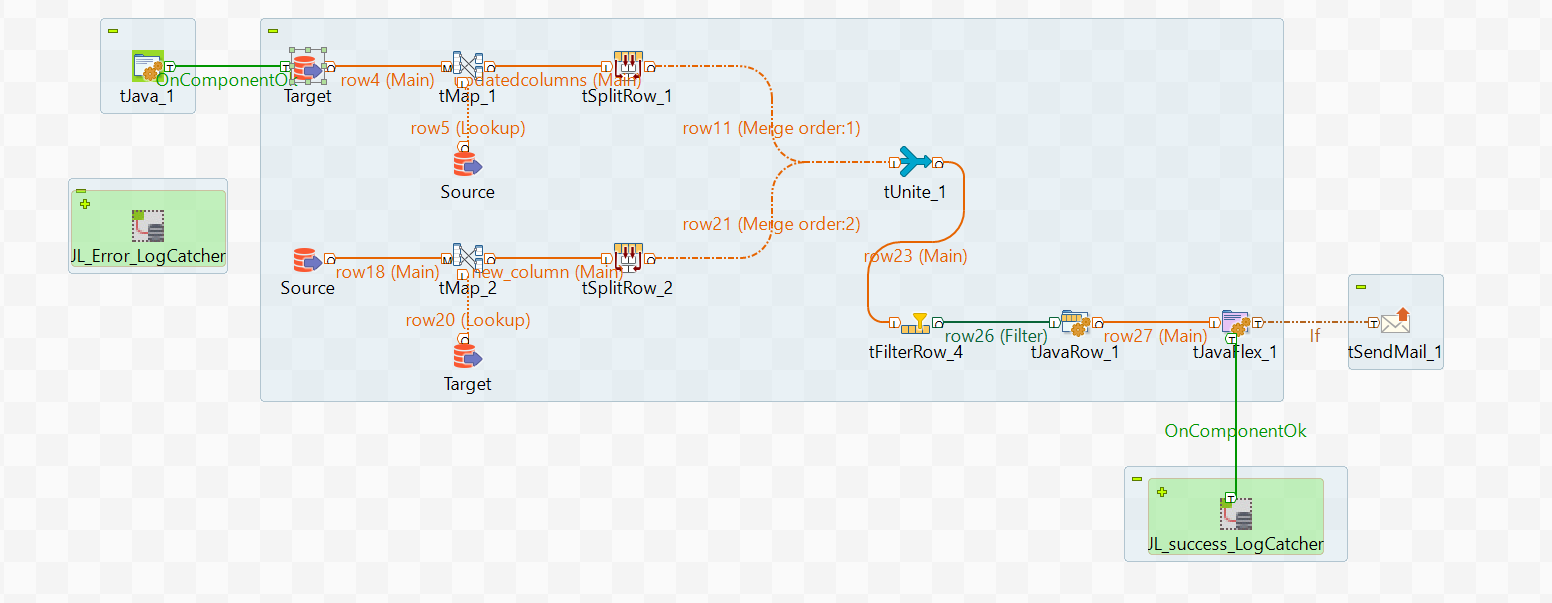






context.jobStartTime = TalendDate.getCurrentDate();

context.machineName = java.net.InetAddress.getLocalHost().getHostName();



"SELECT

[serverName]

,[databaseName]

,[tableName]

,[columnName]

,[dataType]

,[collationName]

,[createdDatetime]

FROM etl.schemaCheck

WHERE isRequiredToCheck ='Y' and tablename in "+context.TableList

"

SELECT @@SERVERNAME as 'ServerName',

DB\_NAME() AS 'DatabaseName',

OBJECT\_NAME(C.object\_id) AS TableName,

c.name 'ColumnName',

UPPER(t.name) +

CASE WHEN t.name IN ('CHAR', 'VARCHAR','NCHAR','NVARCHAR') THEN '('+

CASE WHEN c.max\_length=-1 THEN 'MAX'

ELSE CONVERT(VARCHAR(4),

CASE WHEN t.name IN ('NCHAR','NVARCHAR')

THEN c.max\_length/2 ELSE c.max\_length END )

END +')'

WHEN t.name IN ('DECIMAL','NUMERIC')

THEN '('+ CONVERT(VARCHAR(4),c.precision)+','

+ CONVERT(VARCHAR(4),c.Scale)+')'

ELSE '' END +' '+CASE WHEN c.is\_nullable = 0 then 'NOT NULL' else 'NULL' end

as 'DataType',

C.Collation\_name as 'CollationName',

getdate() as createdDatetime

FROM

sys.columns c

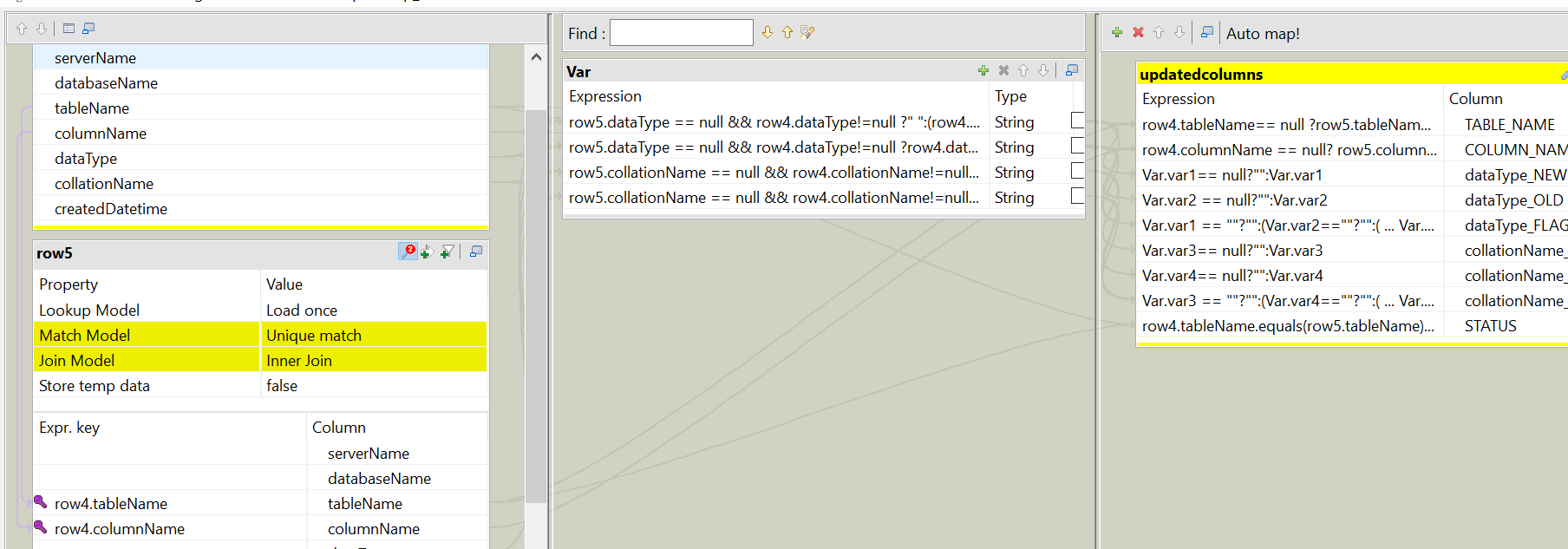
INNER JOIN

sys.types t ON c.user\_type\_id = t.user\_type\_id

WHERE OBJECT\_NAME(C.object\_id) in

"+context.TableList+"

and c.name not in "+context.restrictColumns



//Code generated according to input schema and output schema

String msg = ((String)globalMap.getOrDefault("mailMessage", "<head><style>table { font-family: arial, sans-serif; border-collapse: collapse; width: 50%;}td, th { border: 1px solid #0d0c0c; text-align: left; padding: 8px;}tr:nth-child(even) { background-color: #dddddd;}</style></head><TABLE BGCOLOR=#DCDCDC CELLPADDING=0 border=3 cellspacing=1px" + "<TR><TD><b>TABLE\_NAME</b></TD><TD><b>COLUMN\_NAME</b></TD><TD><b>OLD\_VALUE</b></TD><TD><b>NEW\_VALUE</b></TD><TD><b>STATUS</b></TD></TR>" ));

msg = msg + " <td>"+StringHandling.CHANGE(row23.TABLE\_NAME, "'", "")+"</td> <td>"+row23.COLUMN\_NAME+"</td> <td>"+row23.OLD\_VALUE+"</td> <td>"+row23.NEW\_VALUE+"</td> <td>"+row23.STATUS+"</td></tr>";

globalMap.put("mailMessage", msg );

output\_row.TABLE\_NAME = row23.TABLE\_NAME;

output\_row.STATUS = row23.STATUS;

output\_row.COLUMN\_NAME = row23.COLUMN\_NAME;

**tjavaflex:**

// start part of your Java code

java.util.List<String> list1 = **new** java.util.ArrayList<String>();

**main code:**

// here is the main part of the component,

// a piece of code executed in the row

// loop

**if**(!row27.STATUS.matches("(?i).\*NEW\*.\*")){

list1.add(row27.TABLE\_NAME.toLowerCase());

Set<String> s = **new** LinkedHashSet<String>(list1);

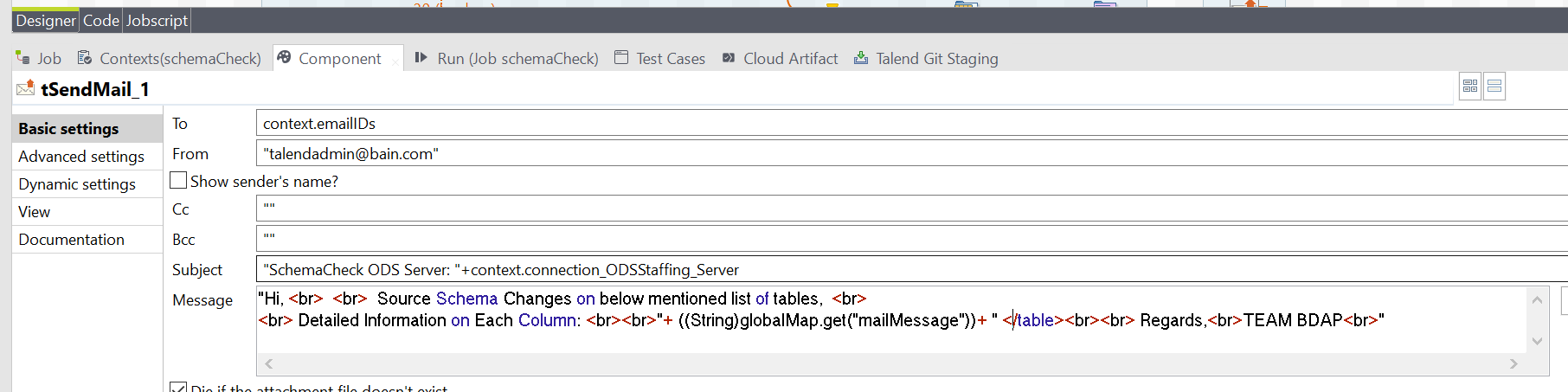
list1.clear();

list1.addAll(s);

globalMap.put("tablelist",list1);

System.out.println("list:"+globalMap.get("tablelist"));

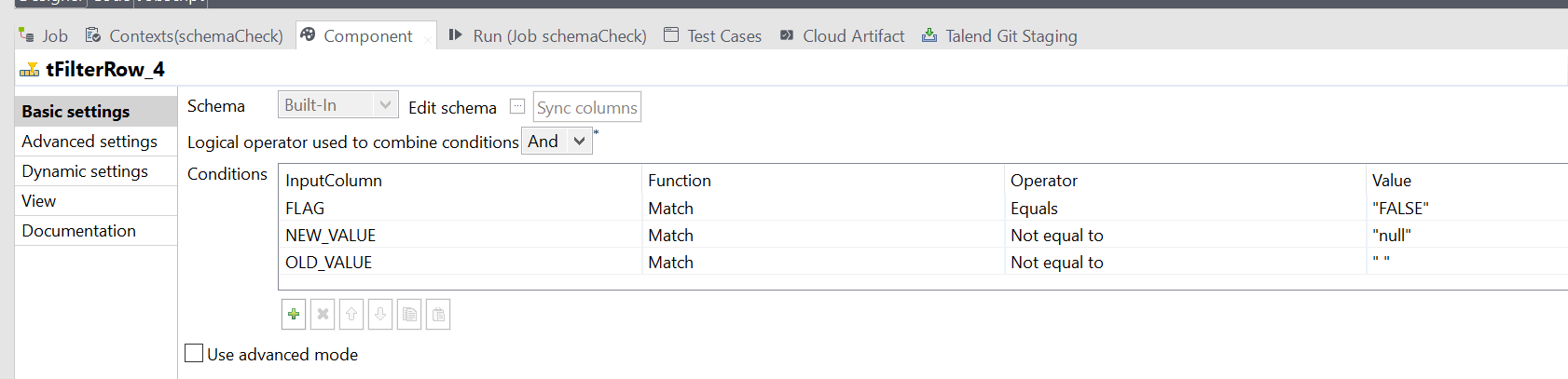
}

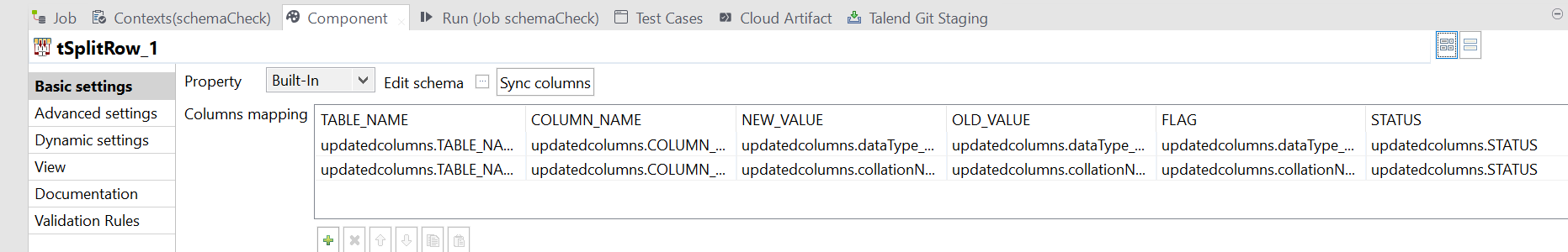


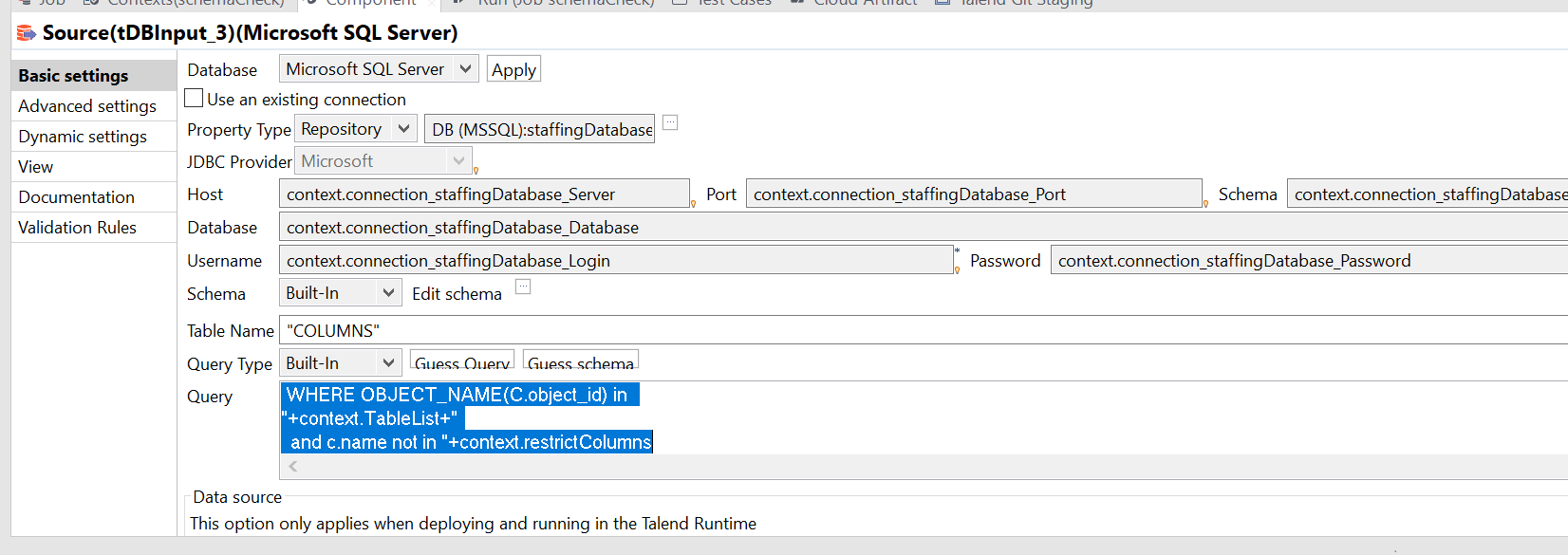
"Hi, <br> <br> Source Schema Changes on below mentioned list of tables, <br>

<br> Detailed Information on Each Column: <br><br>"+ ((String)globalMap.get("mailMessage"))+ " <

/table><br><br> Regards,<br>TEAM BDAP<br>"







**Lookup:**

"SELECT

[serverName]

,[databaseName]

,[tableName]

,[columnName]

,[dataType]

,[collationName]

,[createdDatetime]

FROM etl.schemaCheck

WHERE isRequiredToCheck ='Y' and tablename in "+context.TableList

Main:

"

SELECT @@SERVERNAME as 'ServerName',

DB\_NAME() AS 'DatabaseName',

OBJECT\_NAME(C.object\_id) AS TableName,

c.name 'ColumnName',

UPPER(t.name) +

CASE WHEN t.name IN ('CHAR', 'VARCHAR','NCHAR','NVARCHAR') THEN '('+

CASE WHEN c.max\_length=-1 THEN 'MAX'

ELSE CONVERT(VARCHAR(4),

CASE WHEN t.name IN ('NCHAR','NVARCHAR')

THEN c.max\_length/2 ELSE c.max\_length END )

END +')'

WHEN t.name IN ('DECIMAL','NUMERIC')

THEN '('+ CONVERT(VARCHAR(4),c.precision)+','

+ CONVERT(VARCHAR(4),c.Scale)+')'

ELSE '' END +' '+CASE WHEN c.is\_nullable = 0 then 'NOT NULL' else 'NULL' end

as 'DataType',

C.Collation\_name as 'CollationName',

getdate() as createdDatetime

FROM

sys.columns c

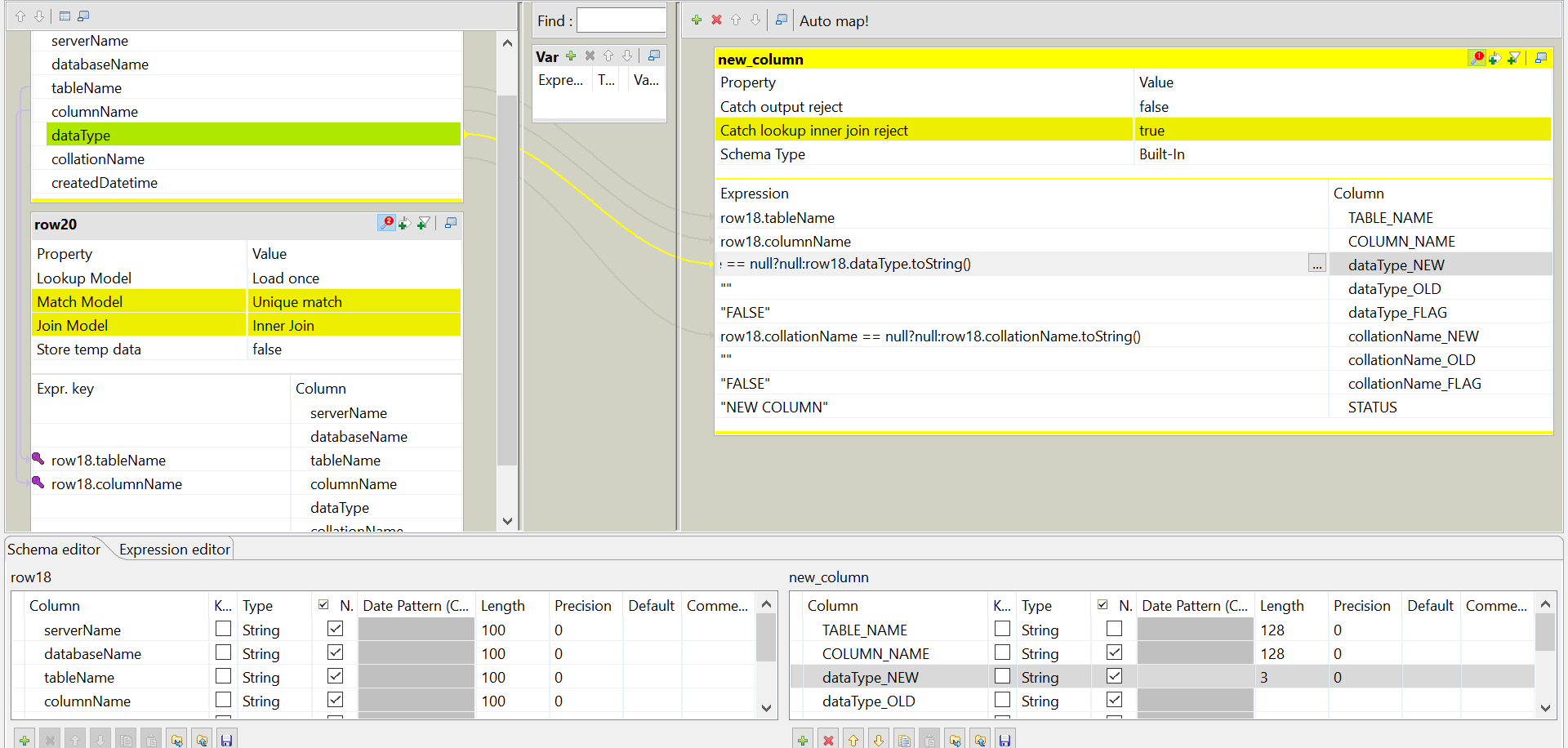
INNER JOIN

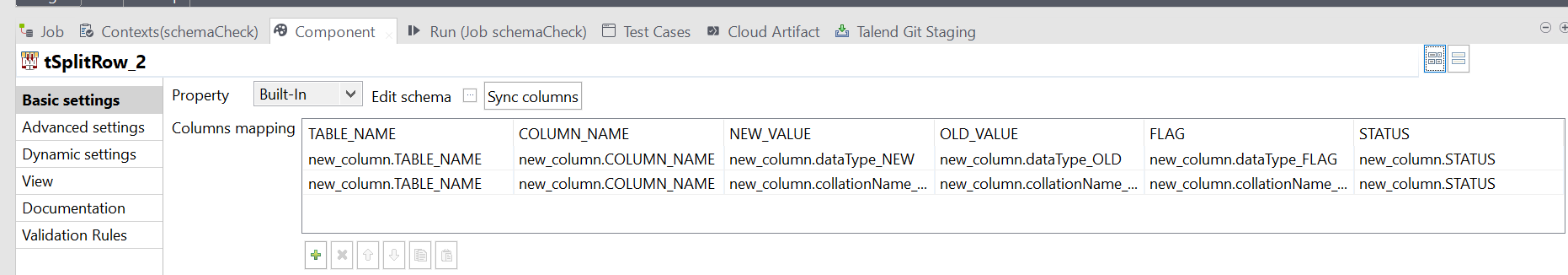
sys.types t ON c.user\_type\_id = t.user\_type\_id

WHERE OBJECT\_NAME(C.object\_id) in

"+context.TableList+"

and c.name not in "+context.restrictColumns





**1st subjob tmap:**

