PLEXA - TRANSFORMATION STUDIO

Contents

1. Overview

1.1 About Transformation Studio

Transformation studio is a tool which extracts huge volumes of data from variety of sources, cleanse, transform and enrich that data to produce a single version of truth. It works with structured, semi-structured and unstructured data.

1.2 Advantages of using Transformation Studio

- Connects to traditional RDBMSs and Big Data systems
- Supports database tables, delimited files, JSON, XML, AVRO, parquet and many other data formats
- Apply transformations on the data coming in from various sources to bring it to the desired shape for analysis
- Cleanse the data to validate the values at the attribute level and take corrective actions if invalid values are found
- Cleanse the data to standardize the values at the attribute level
- Apply business rules to the data
- Store the post processed data in a data repository for Analytics
- Simple drag-and-drop interface with zero coding
- Ability to develop and test jobs in multiple technologies
- Easy migration to different platform versions
- Ability to manage multiple job versions
- Seamlessly integrates with other Plexa studios

2. Jobs

2.1 Steps to create and execute a job

Below are the steps to create a job in Transformation Studio:

1. Add data objects for the inputs and outputs required for a job by providing data connections and authentication for each of these data objects
2. Create a new job in the required mode and version, select the server where the job should reside.
3. Create a job flow that reads the appropriate sources, performs the required transformations, and loads the target data store with the desired information

4. Provide the job authorization to execute the job
5. Save and Execute the job

2.2 Working with Modes

To simplify the process of migrating the jobs from one platform to another without the hassle of re-developing the code, Transformation Studio has different modes available. Each mode generates appropriate backend source code for each of the transformations in the job. Available mode is Spark with Scala

For migrating jobs from one mode to another refer to *Migrating a Job* below

Migrating a Job

Follow the below steps to migrate a job from one platform to another:

1 Click on the “Migrate Job” button on the top panel

2 Enter the below information on the “Migrate Job” window:

- Mode – Select the mode to which the job must be migrated
- Version – Select the version of the mode selected in above step

3 Instance – Select the instance (server) where the job will be stored. More than one instance can be selected

4 New Job Name – Enter the name of the new job

5 Choose the Metadata Folder under “New Folder Name” by clicking on the Browse button. This will open a “Select Folder” window that shows the list of folders available on the left side. Below are the options available in this window:

- Select an existing folder
    - Double Click on the desired existing folder on the left side of the “Select Folder” window and click select. The selected folder location will appear next to the Browse button
- Create a new folder
    - Click on the + symbol on the top right corner of the “Select Folder” window to add a new folder. This will open a “Create New Folder” Window
    - Specify the name of the new folder in “Create New Folder” Window under “Folder Name”. This will create a folder under the main folder called “Plexa”

- To create a sub folder, click on the parent folder and then repeat the above two steps

- Delete an existing folder
  - Click on the desired folder to be deleted
  - Click on the delete icon on the top right corner of the “Select Folder” window. This displays “Delete Selected Folder” window
  - Click Delete to confirm or Close to cancel the action

- View Options
  - Click on the view button present next to the delete icon on the top right corner of the “Select Folder” window. Available options are “List View” and “Tile View”

6 Click on Migrate Button to migrate the job. “Success” window will pop up stating “Job has been submitted for Migration”. Click on “Close” button

Note

This will only save the new job with the new mode in the selected folder but the new job will not be opened

7 Close the old job

8 Open the new migrated job. For steps to open an existing job, refer to *Open a Job*

9 Check out the job to edit. Refer to Check-Out Job section in *Versioning* for more detailed steps.

10 Change the job authentication of the new job to reflect the mode selected for migration. Refer to Job Authentication section in *Authorization* for more detailed steps.

2.3 Working with Jobs

2.3.1 *Create a New Job*

Click on the New Job wizard on the user interface to create an empty job and then follow the below steps.

1 Select the Mode

Select the desired Mode of the job. The available modes are

- Greenplum

- Amazon Redshift
- Spark Streaming
- Spark with Scala
- PostgreSQL

2 Select the Version

The available versions are:

- 4.3.7 (Greenplum)
- 4.20.2016 (Amazon Redshift)
- 1.6 and 2.0 (Spark Streaming)
- 1.6 and 2.0 (Spark with Scala)
- 9.6 (PostgreSQL)

3 Select Instance

Select the Instance(server) where the job will be stored. More than one instance can be selected.

4 Follow the below steps to save the job

- Click on the “Save” button on the top menu bar. This will open a “Save File” Window.
- Select the desired folder location on the left pane.
- Enter the job name in the “Job Name” textbox below.
- Click Save

5 Adding Data Objects to the job

- Using an existing data object - Under the Data section on the left side, drag the data objects to read from and write to and place them at separate ends of the canvas
- Add a new data object – Refer to *Create a new data object*

6 Authorization

7 Authentication

After the data object is authorized, double click the data object. This window has Authentication details in the Authentication tab.

7.1 Authentication Name

Name of the authentication credential

8 Adding Data Reader/Writer nodes to the job

- Under Access category in Tasks section, drag the appropriate reader node required to access data from the data object. For example, if the data object type is a delimited file (also known as, comma separated values…csv), use the delimited file reader
- Under Access category in Tasks section, drag the writer node specific to the file to be written to.

9 Adding Transformations to the job

Under Transformations category in the Tasks section, select the desired transformations to be applied.

9.1 Connections

Different nodes in the job can be connected by dragging the output port of one node to the input port of the next node in the job flow. These are called connections. Double click the connection to open the properties of the connection. This window has below tabs:

1 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. ID

2. Name

3. Description

2 Nodes

This tab has details about nodes in the connection:

Source Node Id

Identifier of the Source Node

Source Port Name

Name of the port connected to the source node

Target Node Id

Identifier of the Source Node

Target Port Name

Name of the port, target mode is connected to

3 *Condition Tag*

10 Save

Click on Save button on the top panel to save the changes made to the job

2.3.2 *Open a Job*

There are two ways to open an existing job:

- Click on any of the recent jobs on the “Recent Jobs” panel
- Click on “Open Job” icon on the top panel. This will open a “Open File” window from which the folder can be selected on the left side and click on the desired job.

2.3.3 *Save/ Save As Job*

If there are changes made to the job in any way, it must be saved to commit the changes by clicking the “Save” button on the top panel or save it with a different name by using “Save As” button.

2.3.4 *Delete a Job*

To Delete a job, open the job and click on “Delete Job” button

2.3.5 *Additional Features*

Each job window has three tabs – Diagram, Source Code and Log. The following table describes the purpose of each of these tabs

Tab	Description
Diagram	Depicts the job flow.
Source Code	Enables to view the code generated for all the transformations used in the job flow
Log	Enables to view the log of the job operation

2.3.6 *Job Properties*

The job properties window enables to view or update the metadata for a job. Double Click the Job Properties icon on the right top corner of the canvas of any job. The job properties pop up window has the below tabs

1 General

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

- *ID* - The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier
- *Name* - This section of the properties allows the user to assign a name to the object.
- *Description* - This section of the properties allows the user to enter the detailed description of the object.

2 Audit

This section of the properties will have the metadata of flow. The user will not have the ability to make any changes in this section. The details in the audit section gets populated automatically by the system

- Created Date - This section of the properties will have the details of when the object was created
- Created By - This section of the properties will have the name of the user of who created the object
- Modified Date -This section of the properties will have the date of the last modification to the object
- Modified By -This section of the properties will have the name of the user of who last modified the object

3 Organization

This section of the properties will have the details about the object location

1 Folder Path

Click on the “Browse” button. This will open a “Select Folder” window that has the below options

1.1 Select an existing folder

Double Click on the desired existing folder on the left panel of the “Select Folder” window and click select. The selected folder location will appear next to the Browse button

1.2 Create a new folder

1. Click on the + symbol on the top right corner of the “Select Folder” window to add a new folder. This will open a “Create New Folder” Window

2. Specify the name of the new folder in “Create New Folder” Window under “Folder Name”. This will create a folder under the main folder called “Plexa”
3. To create a sub folder, click on the parent folder and then repeat the above steps 1 and 2

1.3 Delete an existing folder

1. Click on the desired folder to be deleted
2. Click on the delete icon on the top right corner of the “Select Folder” window. This displays “Delete Selected Folder” window
3. Click Delete to confirm or Close to cancel the action

1.4 View Options

Click on the view button next to the delete icon on the top right corner of the “Select Folder” window. Available options are “List View” and “Tile View”

4 Version Control

This is a read only tab which displays versioning details of the job.

1 Version Number

Version Number of the job.

2 Status

Status of the object in version control.

3 Checkout Date

Date when the job was checked out.

4 User

User who checked out the job.

5 Checkin Date

Date when the job was checked in.

6 Branch Name

Name of the branch in version control.

7 Current

Flag which specifies if the job is current.

8 Main Job ID

ID of the initial job in the version control chain.

9 Parent Job ID

ID of the immediate parent of the current version.

10 Comment

Comment associated with the version.

11 Final

Flag which specifies if the version is final. No more branching is allowed if the version is Final.

5 Security

This tab provides details about Plexa Users who have access to the database system. There are various levels of access as well as precedence of access. Plexa User ID can be specified to one of the several roles described below.

1. Users Grant Read Access

Specifies the list of Users who can see the Database System and all its relevant files and folders.

2. Users Grant Write Access

In addition to roles of Grant Read Access, this role allows the Users to make changes to the Database System and all its files and folders.

3. Users Grant Execute Access

In addition to roles of Grant Write Access, this role allows Users to run a program or a function utilizing the data from the Database System.

4. Users Grant Admin Access

In addition to roles of Grant Execute Access, this role allows Users to control every aspect of the connection including modifying the Login credentials to the database as well as modifying access privileges to other Users. Privileges of Granting Admin Access is like that of a Super user

5. Users Deny Read Access

Specifies the Users who cannot access the Database System and all its files. If is User has any of the Grant Accesses (Read, Write, Execute or Admin) as well as any of the Deny Access (Read, Write, Execute or Admin), then Deny Access will have higher precedence over others.

6. Users Deny Write Access

Specifies the list of Users who can view the files in a Database System but are prohibited from making changes to it.

7. Users Deny Execute Access

Specifies the list of Users who have the permission to view and edit the Database System but blocks access to executing a task.

8. Users Deny Admin Access

Specifies the list of Users who cannot make changes to the Database System such as modifying the User ID and Password to the Database System or inclusion/exclusion of Users/User groups to access the Database System. However, the list of Users can still possess the rights of Users – Grant Execute Access

9. Group Grant Read Access

This field specifies a group of Users who can view the Database Systems and all its files. By adding a Group to this field, every User within that User Group will have access to viewing the contents of the Database System.

10. Group Grant Write Access

This field specifies a group of Users who can modify the contents of the Database System.

11. Group Grant Execute Access

This field specifies a group of Users who can run a task using the Database System.

12. Group Grant Admin Access

This field specifies a group of Users who have all the privileges of Group – Grant Execute Access along with access to modify the connection of the Database Systems and inclusions/exclusions of Users/Users Groups from the various Security levels.

13. Group Deny Read Access

This field specifies a group of Users who cannot view a specific Database System. “Deny” has higher precedence over “Grant”. Therefore, if a User/User Group has Group – Grant Admin Access and one of the Deny Access, then the User/User Group cannot access the specific functionality of the Database System.

14. Group Deny Write Access

This field specifies the group of Users who can view the contents of Database System but cannot make changes to them.

15. Group Deny Execute Access

This field specifies the group of Users who can read and edit the contents of a Database System but cannot run a task.

16. Group Deny Admin Access

This field specifies the group of Users who cannot make changes to the connections or adding/removing Users/User Groups to the Database System.

6 Authentication

This is a read only tab auto populated with Authentication provided for the job.

1 Authentication Name

Name of the authentication credential.

7 Mode Details

This is a read only tab auto populated with Mode and Mode version selected for the job.

1 Mode

Selected mode of the job.

2 Mode Version

Version of the mode selected for the job.

8 Instance Details

This is a read only tab auto populated with Instances selected for the job.

1 Instance name(s)

Instance associated with the selected mode of the job.

2.3.7 *Authorization*

"Authorization" is one of the three panels available on the left panel of the transformation Studio. To execute any job, the data objects and the job should be authorized. Authorizations are organized in a folder structure on the "Authorization" panel.

Authorization must be applied on two levels:

1. Data Object Authentication

Drag the authorization credential based on the Data Server location of the data object. For example, if the data object is stored in a Redshift Server, drag the Redshift authorization onto the data object

2. Job Authentication

Drag the authorization credential based on Mode initially selected. For example, if Mode selected is Greenplum, drag the Greenplum authorization onto the authentication icon available on the right side of the canvas

2.3.8 *Versioning*

Versioning enables the user to see previous versions of the job.

1 Check-In Job

To commit the changes made to the job, the job must be checked in.

2 Check-Out Job

This feature enables the user to check out a job to edit if it is checked in.

3 Branch-Out Job

If multiple users need to work on the same job, branch out option is available to create a different version of the job which is specific to the user. This is the working copy of the checked in job. This will prevent multiple users locking the true version of the job.

4 Job Version

To view previous versions of the job:

Click “Job Version” icon on the panel

Select the desired version to work on

2.3.9 *Executing a Job*

Follow the below steps to execute a job:

- Click the Execute Job icon on the panel.
    - If the changes are saved, the job will be submitted for execution.
    - If there are any unsaved changes, “Save Job and Execute” window will pop up
        - To save the changes, Click on “Save and Execute”. This will open the “Check-in” Job window. In the Check in Job window pop up:
            - If the job is the final version, click True otherwise click False to indicate work is still in progress. This is a mandatory field.

            Note

            If True is selected, the job cannot be edited further
            - To check in the job, click Check in and Execute
            - To continue working on the job, click Execute Job
        - To revert the changes click on “Execute Prior Saved Version”. This will revert the changes made and executes the prior saved version of the job.

2.3.10 *Migrating a Job*

Follow the below steps to migrate a job from one platform to another:

1 Click on the “Migrate Job” button on the top panel

2 Enter the below information on the “Migrate Job” window:

- Mode – Select the mode to which the job must be migrated
- Version – Select the version of the mode selected in above step

3 Instance – Select the instance (server) where the job will be stored. More than one instance can be selected

4 New Job Name – Enter the name of the new job

5 Choose the Metadata Folder under “New Folder Name” by clicking on the Browse button. This will open a “Select Folder” window that shows the list of folders available on the left side. Below are the options available in this window:

- Select an existing folder
    - Double Click on the desired existing folder on the left side of the “Select Folder” window and click select. The selected folder location will appear next to the Browse button
- Create a new folder
    - Click on the + symbol on the top right corner of the “Select Folder” window to add a new folder. This will open a “Create New Folder” Window
    - Specify the name of the new folder in “Create New Folder” Window under “Folder Name”. This will create a folder under the main folder called “Plexa”
    - To create a sub folder, click on the parent folder and then repeat the above two steps
- Delete an existing folder
    - Click on the desired folder to be deleted
    - Click on the delete icon on the top right corner of the “Select Folder” window. This displays “Delete Selected Folder” window
    - Click Delete to confirm or Close to cancel the action

- View Options
    - Click on the view button present next to the delete icon on the top right corner of the “Select Folder” window. Available options are “List View” and “Tile View”

6 Click on Migrate Button to migrate the job. “Success” window will pop up stating “Job has been submitted for Migration”. Click on “Close” button

Note

This will only save the new job with the new mode in the selected folder but the new job will not be opened

7 Close the old job

8 Open the new migrated job. For steps to open an existing job, refer to *Open a Job*

9 Check out the job to edit. Refer to Check-Out Job section in *Versioning* for more detailed steps.

10 Change the job authentication of the new job to reflect the mode selected for migration. Refer to Job Authentication section in *Authorization* for more detailed steps.

2.3.11 *Check In Job*

This pop-up window appears when the User is about to Check In a Job. The purpose of this window is to prevent conflicts in an environment where multiple Users might want to edit the same Job. Check-in describes the process of adding a new or modified item or file into a repository to replace the previous version. The options in this pop-up window are:

1. Final - This drop-down list can be selected to either “True” or “False”. By selecting True, the User confirms that the Job is the Final Version. It is mandatory for the User to select an option from this window.
2. Comments - This field provides a text box for the User to enter any additional information about the job. This field is optional to enter.

3. Tasks

3.1 Access Tasks

Transformation studio has set of specific reader tasks that are designed to read data from disparate data sources such as delimited files, fixed length files, database tables, JSON and XML and writer tasks to load data to delimited files, fixed length files, database tables, JSON and XML. Every source data object should be connected to a Reader task to extract data. Readers

can filter data while extracting. Every target data object should be connected to a Writer task to load the data. Writers can create or use existing target structures. Writers can also append the data or overwrite existing data.

3.1.1 *Reader*

This node is used to read data from databases. The user has the option to add filters to the data sources and to set deployment mode specific parameters.

Follow the below steps to use a reader:

1 Using the Task

- Create an empty job
- Drag a table data object from the data panel on the left on to the job canvas
- Select and drag or Double click the reader task from the Access category

2 Setting the properties

Double click on the table reader task and enter the below information:

3.1.1.1 ***General***

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.1.1.2 ***Spark Read Options***

Parsing Library

Commons : The Apache Commons CLI library provides an API for parsing command line options passed to programs. It's also able to print help messages detailing the options available for a command line tool. (Default value)

Univocity : uniVocity-parsers is a suite of extremely fast and reliable parsers for Java. It provides a consistent interface for handling different file formats, and a solid framework for the development of new parsers.

Mode

Determines the parsing mode. By default it is PERMISSIVE. Possible values are:

Permissive : Tries to parse all lines: nulls are inserted for missing tokens and extra tokens are ignored.

Dropmalformed: Drops lines which have fewer or more tokens than expected or tokens which do not match the schema.

Failfast: Aborts with a RuntimeException if encounters any malformed line.

Ignore Leading White Spaces? :
True : Ignore the leading white space of a value.
False: Accept defaults

Ignore Trailing White Spaces? :
True: Ignore the trailing white space of a value.
False: Accept defaults

3.1.1.3 Temporary Storage
Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.1.2 *Writer*
This node is used to write files to the target data objects. The user has the option to add filters to the incoming data and to set deployment mode specific parameters.

Follow the below steps to use a File Writer task:

1 Using the Task

2 Setting the properties

Double click on the Writer task and enter the below information:

3.1.2.1 ***General***

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.1.2.2 ***Filter Conditions***

Add Filter Conditions as a writing option for your data.

Add Filter Conditions Rules: Once pressed, expandable "Group" will appear where you can add multiple filter conditions.

3.1.2.3 ***Column Selection***

Lets you select the columns to be written. select "_all" if you want all the columns to be written.

3.1.2.4 ***Spark Write Options***

Options for writing the data.

Write Options when data already exists:

Overwrite : Overwrite existing table/file with new data.
Append: Append to the existing table/file.
Ignore: Ignore writing data.
Error If Exists: Throw an error if the file/table exists.

Truncate Target Table?

True: Truncate the table before writing data.
False: Do not truncate the table.

3.1.2.5 Temporary Storage

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.1.3 Data Shell

3.1.3.1 ***General***

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.1.3.2 ***Type of Data Object***

Allow the users to select the type of the data object they want to read.

- Delimited File
- Table
- JSON

3.1.3.3 ***Data Connections***

Display the Properties of the database or service hosting the data object. Data object has to be dragged and dropped on the data shell node automatically populate the following fields.

Data Server Type : Type of the Hosting environment.
Data Server : Name of the hosting environment

Data Location : Location of the data object
Physical Object Name : Name of the data object.

3.1.3.4 ***Authentication***
Authentication Name: Authentication selected according to the environment. Automatically populated when user drags the authentication on to the node.

3.1.3.5 ***Data***
Writing options for the data object.

Delimiter : By default columns are delimited using ,, but delimiter can be set to any character.
Has Headers? : When set to true, the header (from the schema in the DataFrame) will be written at the first line.
Quote Character : By default the quote character is ", but can be set to any character.

3.1.3.6 ***Advanced***

Escape Character : By default the escape character is \, but can be set to any character. Escaped quote characters are ignored.
Character Encoding : Defaults to 'UTF-8' but can be set to other valid charset names.
Comment Character : Skip lines beginning with this character. Default is "#". Disable comments by setting this to null.
Null Value : Specifies a string that indicates a null value, nulls in the DataFrame will be written as this string.
String Representation of Non-Number Value:
String Representation of Positive Infinity :
String Representation of Negative Infinity :
Date & Time Format : Specifies a string that indicates the date format to use writing dates or timestamps. Custom date formats follow the formats at java.text.SimpleDateFormat. This applies to both DateType and TimestampType. If no dateFormat is specified, then "yyyy-MM-dd HH:mm:ss.S".
Maximum Number of Columns :
Maximum Number of Characters in a Field :
Replacement Character :
Retain Quotes within a data value :

3.2 Transform Tasks

3.2.1 Append

This task enables to add records to an existing dataset. The user can append records from two or more different data sources into one single file. The user has the option to add user defined

columns and build expressions for it. It also has options to set deployment mode specific parameters. Follow the below steps to use Append task:

Follow the below steps to use a Append:

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that needs to be appended.
- Select and drag or Double click the Append task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Append task.

2 Setting the properties

Double click on the table reader task and enter the below information:

3.2.1.1 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.1.2 *Order of Columns in Input Files*

Are the Columns in All the Input Files in the same Order? :

True : Indicates that columns are in the same order as input files.

False : Columns are not in the same order.

3.2.1.3 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type

Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.2 *Union*

This task enables to combine records from two or more different datasets. The output will include distinct records from all data sources. It also has options to set deployment mode specific parameters.

Follow the below steps to use Union task:

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be combined.
- Select and drag or Double click the Union task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Union task.

2 Setting the properties

Double click on the table reader task and enter the below information:

3.2.2.1 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.2.2 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.3 Lookup

This task enables to look up a data object based on conditions and returns data, typically from a master data object to a transaction data object. To lookup, both data objects must share a key (common attribute). This task also has options to set deployment mode specific parameters.

Follow the below steps to use Lookup task:

1 Using the Task

- Create an empty job
- Drag two source data objects from the data panel on the left on to the job canvas.
- Select and drag or Double click the Lookup task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Lookup task.

2 Setting the properties

Double click on the table Looup task and enter the below information:

3.2.3.1 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.3.2 ***Lookup Column Selection***
Columns from Lookup File : Select the columns from the lookup file

3.2.3.3 ***Lookup Condition***
Input File Column : Columns from the Input file for lookup
Lookup File Column : Columns from the Lookup file

3.2.3.4 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the "Temporary Storage" Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.4 Custom Code

This task allows the user to write custom code and perform any transformations to the incoming data. Follow the below steps to use Custom Code task:

1 Using the Task

- Create an empty job
- Drag a source data object from the data panel on the left on to the job canvas.

- Select and drag or Double click the Custom Code task from the Transform category under Tasks tab on the left panel.
- Connect the source data object to the Custom Code task.

2 Setting the properties

Double click on the Custom Code task and enter the below information:

3.2.4.1 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.4.2 **Code Input**

Based on the mode of the job, custom code can be written in the Code input text box.

Reset

This will reset the code input to the previously saved version.

Code Input for Spark with Scala

Enter the Spark custom code to transform the data.

3.2.4.3 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type

Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.5 *Rank*

This task ranks one or more variables in the source dataset and stores the rank in the target dataset. Rank task is not available in “Spark With Scala” and “Spark Streaming” modes. Follow the below steps to use Rank task:

1 Using the Task

- Create an empty job
- Drag a source data object from the data panel on the left on to the job canvas that need to be ranked.
- Select and drag or Double click the Rank task from the Transform category under Tasks tab on the left panel.
- Connect the source data object to the Rank task.

2 Setting the properties
Double click on the Rank task and enter the below information:

3.2.5.1 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.5.2 ***Rank By Column***

This tab specifies the columns by which rank is calculated.

Reset
This will reset the rank by columns to the previously saved version.

Rank By Columns
After adding the desired columns by which rank should be calculated, the order of the columns can be changed by dragging and dropping the columns to the desired position.

Sort Type
Rank By columns can be sorted by ascending or descending order.

3.2.5.3 ***Partition By Columns***
Select Partition by columns

3.2.5.4 ***Rank Parameters***
Rank Type : Select the type of rank
- Rank
- Dense Rank
- Percent Rank
- Ntile
- Row Number

Result Variable : Name new rank column

3.2.5.5 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.6 Intersect

This task returns the common records from two or more data objects. It also has options to set deployment mode specific parameters. Follow the below steps to use Intersect task:

1 Using the Task

- Create an empty job
- Drag two source data objects from the data panel on the left on to the job canvas.
- Select and drag or Double click the Intersect task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Intersect task.

2 Setting the properties

Double click on the Intersect task and enter the below information:

3.2.6.1 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.6.2 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type

Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.7 Deduplicate

This task returns the distinct records based on the columns selected. Deduplicate task only allows one input dataset and generate only two datasets - one with distinct records and another with duplicate records. It also has options to set deployment mode specific parameters. Follow the below steps to use Deduplicate task:

1 Using the Task

- Create an empty job
- Drag a source data object from the data panel on the left on to the job canvas.
- Select and drag or Double click the Deduplicate task from the Transform category under Tasks tab on the left panel.
- Connect the source data object to the Deduplicate task.

2 Setting the properties
Double click on the Deduplicate task and enter the below information:

3.2.7.1 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.7.2 ***Deduplicate Columns***

Select the columns for deduplication.

3.2.7.2 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.8 *SCD*

This task captures the slowly changing dimensions. To perform SCD function, two inputs are required: the original dataset and the change file to compare with. This task is not applicable in “Spark Streaming” mode. Follow the below steps to use SCD task:

1 Using the Task

- Create an empty job
- Drag two source data objects from the data panel on the left on to the job canvas.
- Select and drag or Double click the SCD task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the SCD task. Connect the source file to the first input port and change file to the second input port.

2 Setting the properties

Double click on the SCD task and enter the below information:

3.2.8.2 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.8.1 *SCD Type*

There are two types of SCD to choose from.

Type 1

Replaces the Source value (current version) with the Target Value (latest version), that is obtained by performing the SCD. If Type 1 is selected, the source file must contain the following columns:

- Column to identify whether a record is current or old using binary code (1 – Yes/ 0 – No). E.g. 'current flag'
- Column to identify whether a record is deleted or not in the latest file using binary code (1 – Yes/ 0 – No). e.g. 'current flag'. E.g. 'delete flag'

Type 2

Stores both historical and current data in different rows.

Reset

This tab also has the reset button on the top left corner which resets the SCD type to the previously saved version.

3.2.8.2 *Source & Change File Mapping*

This tab allows users to map columns between source file and Change file.

AutoMap
After populating desired fields to the right side, Click on Auto Map to populate the mapping from source to target. For the mapping to be populated, the field names should exactly match but are not case sensitive.

Clear Mapping
Click on Clear Mapping to clear all the mappings from source to target.

Source File Name
Name of the source object. It is auto populated and cannot be altered by the user.

Source File Column Names
Column Names in the source object.

Change File Name
Name of the File being used to compare changes.

Change File Column Names
Column Names in the Change File.

3.2.8.3 *SCD Properties*

This tab specifies how to make the changes of tracked columns in the target. The following options are available in this tab based on the type of the SCD selected under the “SCD type” tab:

1 Primary Key in Source File
Select the primary key columns in the source file by clicking on “Add keycolumns” button. This option is applicable for both SCD Type 1 and Type 2.

2 Delete Flag Column from Source File
Select the delete flag column in the source file (which identifies whether a record is soft deleted) from the drop down. This option is applicable for both SCD Type 1 and Type 2.

3 Modified Date Column from Source File
Select the modified date column in the source file (which identifies whether a record is modified) from the drop down. This option is applicable for only SCD Type 1.

4 Tracked Columns
Select the columns that need to be tracked in the target with changes. If there are any changes identified in these columns, a new version of this record will be created in the target with the captured changes. This option is applicable for only SCD Type 2.

5 Update Columns
Select the columns that need to be updated in the target with changes. If there are any changes identified in these columns, the latest version of this record in the target will be updated with the captured changes. This option is applicable for only SCD Type 2.

6 Start Date
Specify the start date column from the source file to track the new version of the records. This option is applicable for only SCD Type 2.

7 End Date
Specify the end date column from the source file to track the new version of the records. This option is applicable for only SCD Type 2.

8 Current Flag
Specify the current flag column from the source file which specifies whether there are changes in any of the columns. This option is applicable for only SCD Type 2.

9 Name of Change Type Indicator (New Column)
User can assign a name for the new change type column. This column will be added to the target. Once the name is specified in this tab, the “Column Mapping” tab will reflect this name as a read only column. This option is applicable for both SCD Type 1 and Type 2.

10 Can the missing Rows in Change File be Assumed as Deleted
True/False can be selected in this option. True indicates that the change file has all the records regardless of any new changes. False indicates that the change file has only changed records. This option is applicable for both SCD Type 1 and Type 2.

11 Delete Flag Column in Change File
If the delete assumption in the above option is selected as False, then specify the delete column in the change file from which deletion can be inferred. This option is applicable for both SCD Type 1 and Type 2.

3.2.8.4 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.8.5 ***Connecting the SCD Task***

SCD task has four output ports - Output File, Insert Records Temp File, Update Records Temp File and Upsert Records Temp File. Connect the Output File port to the job flow to track all the inserts, updates and deletes. Connect the Insert Records Temp File port to the job flow to track

only inserts. Connect the Update Records Temp File port to the job flow to track only updates. Connect the Upsert Records Temp File port to the job flow to track inserts and updates.

3.2.9 Minus

This task returns all records that exist in the first dataset and not in the second dataset (including common records). It also has options to set deployment mode specific parameters. Follow the below steps to use Minus task:

1 Using the Task

- Create an empty job
- Drag two source data objects from the data panel on the left on to the job canvas.
- Select and drag or Double click the Minus task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Minus task.

2 Setting the properties

Double click on the Minus task and enter the below information:

3.2.9.2 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.9.3 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.10 Filter

This task enables to filter records from a dataset based on specified conditions. The user has the option to add user defined columns and build expressions for it. It also has options to set deployment mode specific parameters. Follow the below steps to use Filter task:

1 Using the Task

- Create an empty job
- Drag a source data object from the data panel on the left on to the job canvas that need to be filtered.
- Select and drag or Double click the Filter task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Filter task.

2 Setting the properties
Double click on the Filter task and enter the below information:

3.2.10.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.10.3 *Filter Conditions*

This tab is to add or modify the conditions to filter the data. Simple or complex expressions can be built in this tab .

3.2.10.3 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.11 Serial Number Generator

This task generates a surrogate key column to the input dataset. This task is useful when a key column is not identifiable in a dataset. This task is not applicable in “Spark Streaming” mode. Follow the below steps to use Serial Number Generator task:

1 Using the Task

- Create an empty job

- Drag a source data object from the data panel on the left on to the job canvas that needs a surrogate key.
- Select and drag or Double click the Key Generator task from the Transform category under Tasks tab on the left panel.
- Connect the source data object to the Key Generator task.

2 Setting the properties
Double click on the Serial Number Generator task and enter the below information:

3.2.11.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.11.3 *New Column Details*
New Column Name : Enter the name of the new key column
Start From Value : Enter the start value for the key
Incremental Value : Enter the value to be incremented.

3.2.11.4 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.12 Expression Builder

This node enables to build expression on any columns from the input dataset. It also has options to set deployment mode specific parameters. Follow the below steps to use Expression Builder task:

1 Using the Task

- Create an empty job
- Drag a source data object from the data panel on the left on to the job canvas that needs an expression.
- Select and drag or Double click the Expression Builder task from the Transform category under Tasks tab on the left panel.
- Connect the source data object to the Expression Builder task.

2 Setting the properties
Double click on the Expression Builder task and enter the below information:

3.2.12.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.12.3 *Create Expression*

New Column Name: Enter the name for new expression column.
Column Type: Enter the Type for the new column.
Expression: Click the field to open the "Expression Builder" window to add expressions

3.2.12.4 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the "Temporary Storage" Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.13 *Join*

This task enables to join two or more different datasets and select one or more columns from either dataset. It also has options to set deployment mode specific parameters. Join task accepts only two datasets as input. If there are more than two datasets to merge, more than one merge task must be used. Follow the below steps to use Merge task:

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be merged.
- Select and drag or Double click the Join task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Join task.

2 Setting the properties

Double click on the Join task and enter the below information:

3.2.13.2 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.13.3 *Column Selection*

Columns from Left File : Columns to be selected from the left file.
Columns from Right File : Columns to be selected from the right file.

3.2.13.4 *Join Type*

This tab specifies the type of join to merge the data objects.

1 Type of Join

This specifies the type of join to use to merge the data objects. Available options are:

- Left Join
- Right Join
- Inner Join
- Full Outer Join
- Cross Join
- Natural Join

3.2.13.3 *Join Condition*

Left File Column : Columns from the left file.
Right File Column : Columns from the right file.

3.2.13.4 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.14 Offset

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be merged.
- Select and drag or Double click the Offset task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Offset task.

2 Setting the properties
Double click on the Offset task and enter the below information:

3.2.14.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.14.2 *OffSet Parameters*

Input Variable : The input column which offset is performed.

Offset Type : Indicates the type of Offset to be performed

Lead
Lag
First Value
Last Value

Relative Row Number : Indicates the relative row number from which the values have to be looked up.

Result Variable : Indicates the name of the new variable.

Value to Replace Null : Indicates the user entered value to replace null while calculating lead and lag.

3.2.14.3 *Sort By Columns*

Sort By Columns : The columns by which the data is to be sorted.

Sort Type : Indicates if the selected column has to be sorted in an ascending or descending way.

3.2.14.4 *Partition By columns*

Columns by which the data has to be partitioned for ranking.

3.2.14.5 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type

Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name

Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.15 Drop Columns

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be merged.
- Select and drag or Double click the Drop Columns task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Drop Columns task.

2 Setting the properties
Double click on the Drop Columns task and enter the below information:

3.2.15.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.15.2 *Columns to Drop*
Columns to be dropped

3.2.15.3 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.16 Rename Columns

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be merged.
- Select and drag or Double click the Rename Columns task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Rename Columns task.

2 Setting the properties
Double click on the Rename Columns task and enter the below information:

3.2.16.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.16.2 *Columns to Rename*
Columns to be renamed.

3.2.16.3 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the "Temporary Storage" Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.17 *Pivot*

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be merged.
- Select and drag or Double click the Pivot task from the Transform category under Tasks tab on the left panel.
- Connect the source data objects to the Pivot task.

2 Setting the properties

Double click on the Pivot task and enter the below information:

3.2.17.2 *General*

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.18.3 *Pivot Parameters*

Group By Columns : Columns by which the data will be grouped.
Pivot Columns : Columns which will be actually pivoted
Specific Values in the Pivot Column : Optional parameter to choose only specific values from the the Pivot Column.

3.2.17.4 *Value Columns*

Value Columns : Value column associated with the pivot column.
Aggregation : Aggregation associated with the value column.

3.2.17.5 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.2.17 Select Specific Columns

1 Using the Task

- Create an empty job
- Drag two or more source data objects from the data panel on the left on to the job canvas that need to be merged.
- Select and drag or Double click the Select Specific Columns task from the Transform category under Tasks tab on the Select Specific Columns panel.
- Connect the source data objects to the Select Specific Columns task.

2 Setting the properties
Double click on the Select Specific Columns task and enter the below information:

3.2.18.2 *General*
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.2.18.3 *Columns to Select*
Select the Columns

3.2.18.4 *Temporary Storage*

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.3 Quality Tasks

Checks are specific rules of validation or standardization that will ensure the quality of the data. The checks must be predefined in the Checks Studio. These checks are imported to the Transformation Studio and mapped to the corresponding columns in the input data. Multiple checks can be combined into a Checkset with the order of priorities for batch execution. One check can be executed in one or more data jobs.

3.3.1 Data Quality

This node is used to perform data quality checks on the input data object. Follow the below steps to use a Data Quality task:

1 **Using The Task**
Create a job having source data objects with respective reader tasks and desired transform tasks

Select and drag or Double click the Data Quality task from the Quality category under Tasks tab on the left panel.

Connect the transform task to the Data Quality task.

2 **Setting the properties**
Double click on the Data Quality task and enter the below information:

3.3.1.1 ***General***
This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.3.1.2 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.4 Analyze Tasks

Analyze tasks are available to collect informative summaries about the data. They help to get various metrics on the variables within the dataset.

3.4.1 Summary Statistics

This node is used to generate summary statistics on the columns from the source dataset. Follow the below steps to use a Summary Statistics task:

1 Using The Task
1 Create a job having source data objects with respective reader tasks and desired transform tasks 2 Select and drag or Double click the Summary Statistics task from the Analyze category under Tasks tab on the left panel. 3 Connect the transform task to the Summary Statistics task.

2 Setting the properties

Double click on the Summary Statistics task and enter the below information:

3.4.1.1 **General**

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.4.1.2 ***Summary Parameters***

Aggregation

This is the aggregation method that will be applied on the analysis variable chosen. The different aggregations allowed in this drop-down menu are:

- Sum
- Count
- Count of Unique Values
- Average
- Minimum
- Maximum
- Standard Deviation
- Variance
- Range
- Count of missing values
- Count of non-missing values

3.4.1.3 Group By Columns

Columns by which the data has to be grouped for summarization.

3.3.1.4 Temporary Storage

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.

Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type
Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name
Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name
Name of the temporary file name. It is auto populated and cannot be altered by the user.

3.5 Rules Tasks

A rule specifies conditions to be evaluated and actions to be taken if those conditions are satisfied. They define business conditions to constraint outcomes of different business scenarios. One condition is always resolved to two outputs, true or false, followed by corresponding actions. The rules must be predefined in the Business Rules Studio. These rules are imported to the Transformation Studio and mapped to the corresponding columns in the input data. Multiple rules can be combined into a Rulesets with the order of priorities for batch execution. One rule can be executed in one or more data jobs.

3.5.1 Business Rules

This node is used to validate business rules. Follow the below steps to use a Business Rules task:

1 Using The Task
1 Create a job having source data objects with respective reader tasks and desired transform tasks 2 Select and drag or Double click the Business Rules task from the Rules category under Tasks tab on the left panel. 3 Connect the transform task to the Business Rules task.

2 Setting the properties
Double click on the Business Rules task and enter the below information:

3.5.1.1 **General**

This tab contains basic information about the objects such as the Name of the Object, additional information about the object and an auto generated field called ID, which uniquely identifies each object

1. *ID* : The object will be assigned a system generated unique identifier, user will not have the ability to make any changes to this identifier

2. *Name*: This section of the properties allows the user to assign a name to the object.

3. *Description*: This section of the properties allows the user to enter the detailed description of the object.

3.5.1.2 ***Temporary Storage***

Any task in the transformation studio can have multiple input ports with each port associated with a temporary file name that stores input data and multiple output ports with each port associated with a temporary file name that stores transformed data.
Input temporary file names cannot be edited but all the output temporary file names can be edited in the “Temporary Storage” Tab.

Port Type

Represents input or output port. It is auto populated and cannot be altered by the user.

Port Name

Name of the port. It is auto populated and cannot be altered by the user.

Temp File Name

Name of the temporary file name. It is auto populated and cannot be altered by the user.