20th September 2023



Data Integration Layer

Framework

Contents

[**Introduction** 4](#_Toc146128577)

[**How to Open Data Integration Layer in Lumenore?** 5](#_Toc146128578)

[**Modules** 5](#_Toc146128579)

[**Features** 6](#_Toc146128580)

[**Workspace** 7](#_Toc146128581)

[**Source** 8](#_Toc146128582)

[**New Connection:** 9](#_Toc146128583)

[**Data Grid:** 10](#_Toc146128584)

[**Excel:** 11](#_Toc146128586)

[**File Reader:** 11](#_Toc146128587)

[**Fixed Width:** 11](#_Toc146128588)

[**JSON Reader:** 11](#_Toc146128589)

[**Table Input:** 11](#_Toc146128590)

[**Target** 11](#_Toc146128591)

[**Dim load:** 11](#_Toc146128592)

[**Encrypt Load:** 12](#_Toc146128593)

[**Lumenore Data Loader:** 12](#_Toc146128594)

[**Vertica Bulk Loader:** 12](#_Toc146128595)

[**Table Output:** 13](#_Toc146128596)

[**Target File:** 13](#_Toc146128597)

[**Generic Bulk Loader:** 13](#_Toc146128598)

[**Transformation** 13](#_Toc146128599)

[**Abort:** 14](#_Toc146128600)

[**Aggregator:** 14](#_Toc146128601)

[**Column Split:** 15](#_Toc146128602)

[**Data Pivot:** 16](#_Toc146128603)

[**Data Transpose:** 18](#_Toc146128604)

[**Data Unpivot:** 19](#_Toc146128605)

[**Editor:** 20](#_Toc146128606)

[**Filter:** 22](#_Toc146128607)

[**Set Variable & Get Variable:** 23](#_Toc146128608)

[**Job Executor:** 25](#_Toc146128609)

[**Joiner:** 27](#_Toc146128610)

[**Preview:** 28](#_Toc146128611)

[**Query Executor:** 28](#_Toc146128612)

[**Remove Duplicates:** 30](#_Toc146128613)

[**Selector:** 30](#_Toc146128614)

[**Sequencer:** 31](#_Toc146128615)

[**Simple Evaluation:** 32](#_Toc146128616)

[**Sorting:** 33](#_Toc146128617)

[**String Operations:** 34](#_Toc146128618)

[**Table Migration:** 35](#_Toc146128619)

[**Union:** 36](#_Toc146128620)

[**Wait:** 36](#_Toc146128621)

[**Rest API:** 38](#_Toc146128622)

[**JavaScript Executer** 39](#_Toc146128623)

[**Graph:** 40](#_Toc146128624)

[**Job:** 40](#_Toc146128625)

[**Linear Regression:** 41](#_Toc146128626)

[**Predictor** 42](#_Toc146128627)

[**Pre-processor:** 43](#_Toc146128628)

Data Integration Layer

# **Introduction**

The Data Integration Layer is a user interface (UI) based module that facilitates data integration by enabling users to extract data from various source systems, transform it, and deliver it in a presentable way. The available features of the Data Integration Layer are as follows:

* Data extraction: This involves extracting data from homogeneous or heterogeneous sources.
* Data transformation: This stage involves cleaning and transforming data into a proper storage format/structure for querying and analysis.
* Data loading: This stage describes the insertion of data into the final target database such as an operational data store, a data mart, a data lake, or a data warehouse.

# **How to Open Data Integration Layer in Lumenore?**

To open the Data Integration Layer, adhere to the steps listed below:

**Step 1:** Open the URL,

[www.lumenore.com](http://www.lumenore.com)

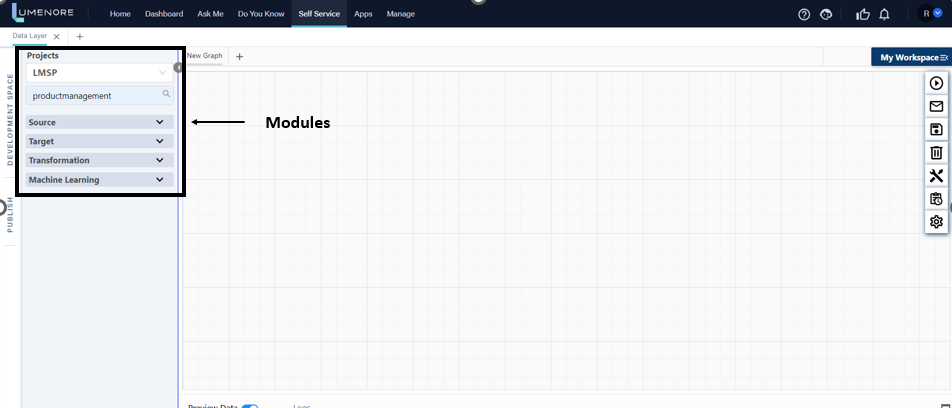
**Step 2:** Sign in with your credentials or sign up to create a new account.

**Step 3:** After opening Lumenore, go to Select Self Service > **Data Integration Layer**.A screenshot of a computer

Description automatically generated

# **Modules**

In the Data Integration Layer window, four modules are shown in the below screenshot:



**Source**: The source section reads data from the local file system or pre-built connectors. Additionally, you can extract information from a table, database, or cloud connector.

**Target**: After performing any task on the data extracted from the source, Target assists in loading the data into the database, file system, or cloud storage.

**Transformation**: A raw data source is transformed into a clean, valid, and usable format through data transformation. For instance, we can use transformation to conduct activities like data cleansing, validation, and null removal. The screenshot of the change and the several operations it permits are shown below.

**Machine Learning:** Machine learning is a subset of artificial intelligence that focuses on the development of algorithms and statistical models that enable computer systems to learn and improve from experience without being explicitly programmed. Machine Learning in Lumenore offers features: Pre-processor, Linear Regression, and Predictor.

# **Features**

The following user-feasibility options are available on the right panel of the Data Integration Layer window:

A screenshot of a computer

Description automatically generated

**Run:** Run the current steps, graphs, or jobs.

**E-mail:** The user can set up alert emails with messages for the Start, Success, and Failure of current graphs, or jobs.

A screenshot of a computer

Description automatically generated

**Save:** Save the existing connections, graphs, or jobs with this option.

**Clear:** Removing the existing connections, graphs, or jobs.

**Tools:**

* **Jobs:** The user can find job details (name, occurrence, and action).
* **Graphs:** Here user can find graph details (name, occurrence, and action).
* **Logs:** Displays the graphs and jobs history.

**Scheduler History:** It tracks and maintains a historical record of scheduled jobs over time.

**Settings:** This tab allows the user to alter the number of rows displayed in the preview data.

# **Workspace**

The workspace area contains the user's saved **connections**, **graphs**, and **jobs**. The user can also view his work history.

A screen shot of a graph

Description automatically generated

After clicking on My Workshop, the user can see three options:

* **Connections**
* **Graphs**
* **jobs**

A screenshot of a computer

Description automatically generated

# **Source**

The source section retrieves data from the local file system. Alternatively, you have the option to utilize pre-configured connectors for data acquisition. Additionally, data extraction from tables, databases, or cloud connectors is also supported.

A screenshot of a computer

Description automatically generated

You can load the data from the several import options under the source section.

**New Connection:** Within the new connection option you can import data from a range of connectors such as Athena, Azure Table, Google Spanner, and more. This broadens the data import capabilities, catering to different departments like Big Data & No SQL, Marketing, Accounting, and others. The connections section empowers you to establish connections with databases and allows you to create custom connections that can be used in various graphs.

**How to use Connections?**

* Simply click on the newly created connection under connections.
* Search and choose from any database, such as Lumenore, MySQL, etc.
* To connect to the database, complete all the fields.
* Use the Test Connection button to verify the connection.
* Then select Add a Connection.
* The connections panel ought to add your connection.
* Additionally, you can edit and delete connections from the tree.

A screenshot of a computer

Description automatically generated

**Data Grid:** Data can be imported according to your requirements.

When the user clicks on the data grid option, the user will find the following fields:

* **Step Name:** Here, you can assign a name to its file.
* **Data Type:** Specify the data type, such as long, integer, double, etc. It serves as a row in an array of your data.
* **Column Name:** You can use this field to name the columns in its data.

Top of Form

**A screenshot of a computer

Description automatically generated**

**Excel:** Import data from Excel files. Supported formats are .XLSX, .XLS, .XLSM, and .XLSB.

**File Reader:** Import data from the file reader. Supported formats are .csv, .txt.

**Fixed Width:**

**JSON Reader:** A JSON reader, also known as a JSON parser, is a software component or library that reads JSON files and provides column-table output. It reads JSON data and converts it into a structured format that can be manipulated and processed by software, facilitating data interchange and integration in various applications.

**Table Input:** Allows data to be extracted from structured data sources for further processing, transformation, and loading into target systems. It provides flexibility and customization options to meet specific data extraction requirements.

# **Target**

Once any operation on the data that has been extracted from the source is completed, the target function allows to conveniently transfer the data into a database, save it to a file system, or store it in a cloud storage system of choice.

A screenshot of a computer

Description automatically generated

When you click on Target, the options listed below will appear as shown in the above screenshot:

**Dim load:** "Dim load" refers to the process of populating data into dimension tables, which are one of the two primary table types within a data warehouse, the other being fact tables.

* Dimension tables contain descriptive attributes that provide context for the numerical measures stored in the fact table. For example, in a sales data warehouse, a time dimension table would include attributes like date, month, quarter, and year.
* A product dimension table would have feature attributes like product name, category, and brand, offering descriptive details about the products.
* Loading data into dimension tables entails inserting or updating the dimension records based on the source data. This operation is typically carried out using an Extract, Transform, Load (ETL) tool or process.
* The ETL tool may execute various transformations, including data cleansing, mapping, and validation, to ensure the accuracy and consistency of the data within the data warehouse.
* Once the data has been successfully loaded into the dimension tables, users can leverage it for reporting, analysis, and data mining purposes to gain valuable insights from the data.

**Encrypt Load:** Encrypting data is a crucial method for safeguarding data security during the ETL (Extract, Transform, Load) process. Encryption transforms data into an unreadable format, making it accessible only with the correct decryption key. This robust security measure creates a formidable barrier, making it challenging for unauthorized individuals to either access or manipulate the data.

* Data encryption is deployed at the load step of an ETL process. This indicates that the data is encrypted before being stored in the final system, such as a data warehouse.
* A built-in encryption function or a call to an external encryption library are two alternative ways that encryption can be implemented depending on the ETL tool being used.

**Lumenore Data Loader:** With the Lumenore data loader feature, you can easily create and name datasets along with the columns within them. This flexibility allows to compile datasets from various sources. For instance, they can connect a database and an Excel file using the Joiner function, and then use this combined data to build the dataset's structure and relationships (known as schema and joins).

**Vertica Bulk Loader:** The Vertica bulk loader is a specialized tool designed for high-speed loading of large amounts of data into Vertica, which is a type of data warehouse known for its column-based storage. This loader is finely tuned for top-notch performance and can load data into Vertica at blazing speeds, reaching rates of several gigabytes per second.

* The Vertica bulk loader employs multiple strategies for fast data loading, such as parallelization and compression. It breaks data into smaller pieces and loads them simultaneously, resulting in impressive throughput rates.
* Typically, the Vertica bulk loader is employed when there is a need to load substantial amounts of data rapidly and effectively into Vertica. For instance, it is useful for loading data from sources like log files or sensor data streams into Vertica for analysis. It is also a common choice in data warehousing situations where large datasets must be regularly imported into Vertica regularly.

**Table Output:** Data is stored and arranged using tables. The source system's data is transformed and processed to produce the output tables in ETL.

The ETL process's transformation rules and the business requirements determine the output tables' structure and content. In general, the downstream applications' reporting or analytical needs should be taken into consideration while designing the output tables.

**Target File:** The "target file" is the file or database table where the transformed data will be loaded or placed. The user can download the data in CSV/Excel or other forms by using the target file.

Data is extracted from one or more sources—such as databases, flat files, or APIs—and then transformed during the ETL process so that it complies with the destination data model or schema. A database table or flat file used for reporting or analysis could be the target file.

**Generic Bulk Loader:** A generic bulk loader is a tool or component used in data integration and ETL (Extract, Transform, Load) processes to efficiently load large volumes of data into a target database or data warehouse. It is often a part of ETL or data integration software and is designed to handle various data formats and databases in a generic or versatile way.

# **Transformation**

Data transformation means converting raw data into a clean, verified, and immediately usable format. This can involve actions like cleaning up the data, checking it for accuracy, removing null values, or other similar operations. Here is a screenshot that shows the transformation tool and the different actions you can perform with it.

A grid of graph paper

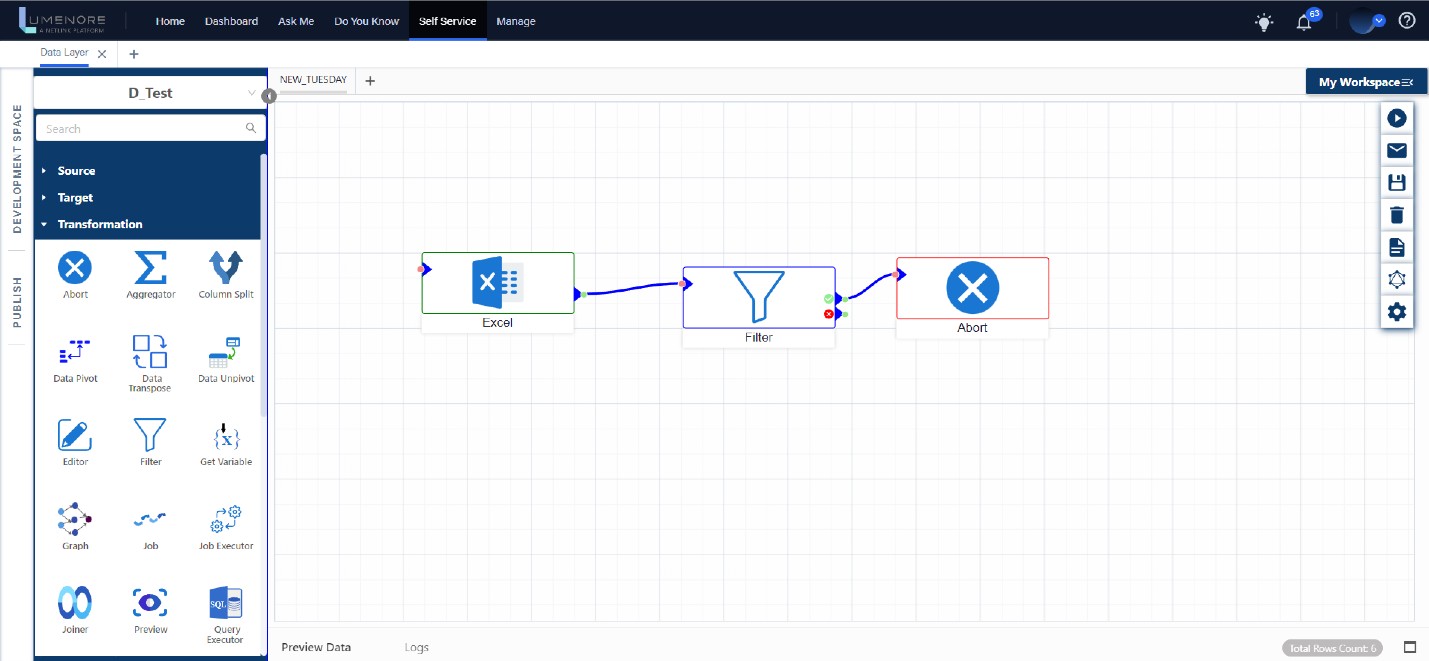
Description automatically generated with medium confidence

To use the transformation feature, follow these steps:

1. Begin by creating a data source for your input data.
2. Next, expand the transformation options from the component tree, and choose the transformation you want to apply.
3. You can simply click on the transformation step or drag it from the component tree to select it.
4. Connect the output node of your data source with the transformation’s input node.

Now, let us explore the various functions that Transformation provides.

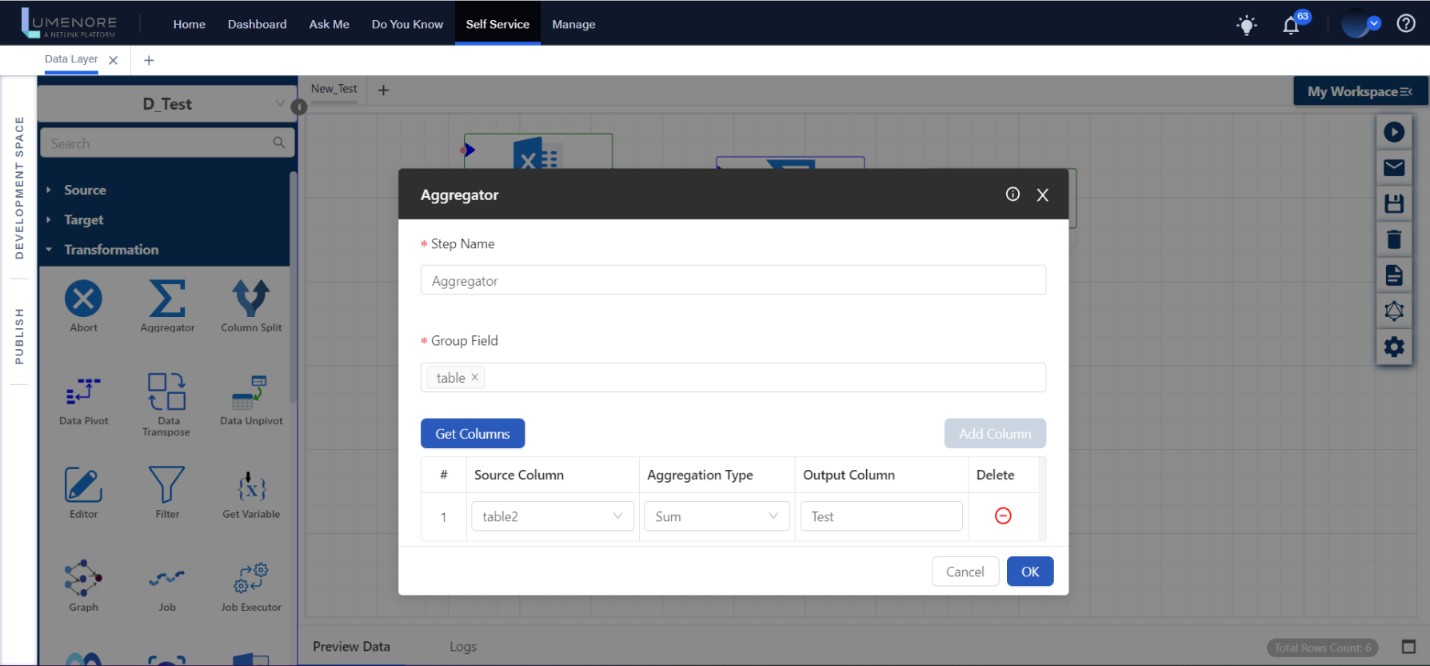
**Abort:** The abort step is useful to stop the execution of the graph at any stage. Using abort, we can stop execution after checking any condition.



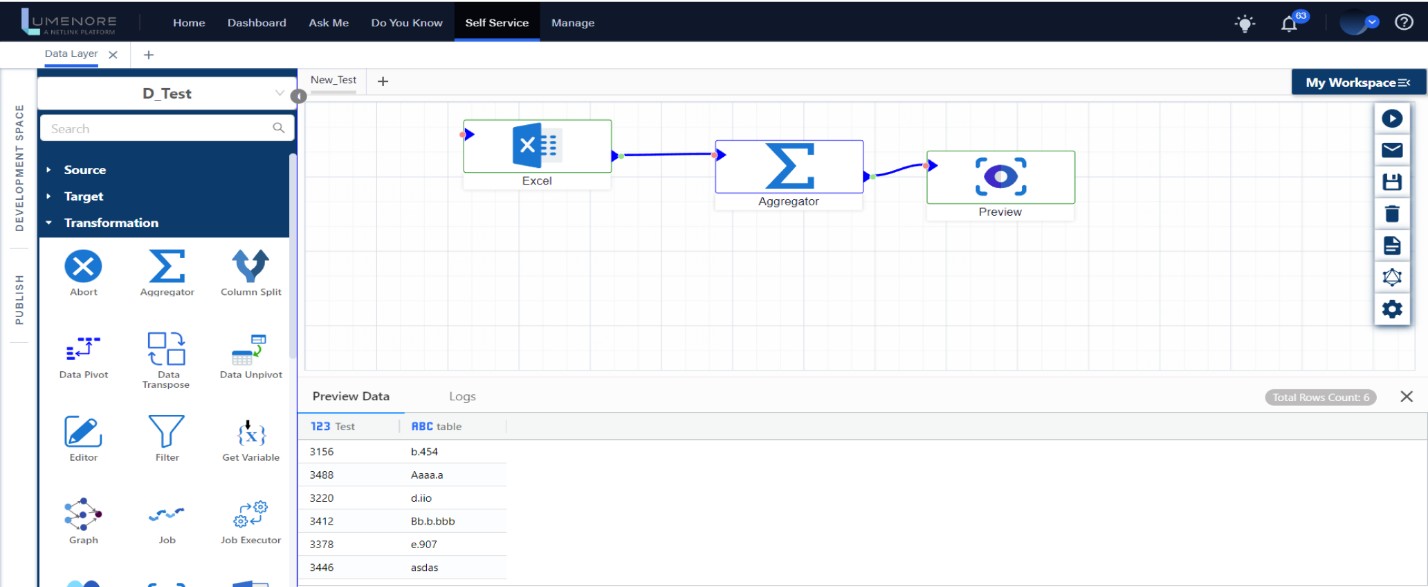
**Aggregator:** All aggregation operations like MAX, MIN, AVG, COUNT, SUM, PROFIT, etc. are performed in the aggregator stage.

**How to use an aggregator?**

* To use the aggregator function, start by selecting your input data source. In this example, use an Excel file as a source. Import the data from this Excel file, and then apply the aggregator function to perform the desired task.



* In the "Step Name" field, give a name to the file of your choice.
* In the "Group" field, import different columns you want to group for performing the aggregator function on that data set.
* For the "Source Column," choose the column where you want to apply the aggregate function. In the "Aggregation Type," decide on the specific type of aggregation you need, such as SUM, MAX, MIN, etc. In the "Output Column," name the resulting column as you prefer, and then click "OK."
* Finally, click the "Run" button to view the output.



**Column Split:** It is used to split the data within a column into multiple columns or parts based on a specified delimiter or pattern. This function is particularly useful when dealing with structured data where information is stored in a single column, and you need to extract or split it into separate components for further processing or analysis.

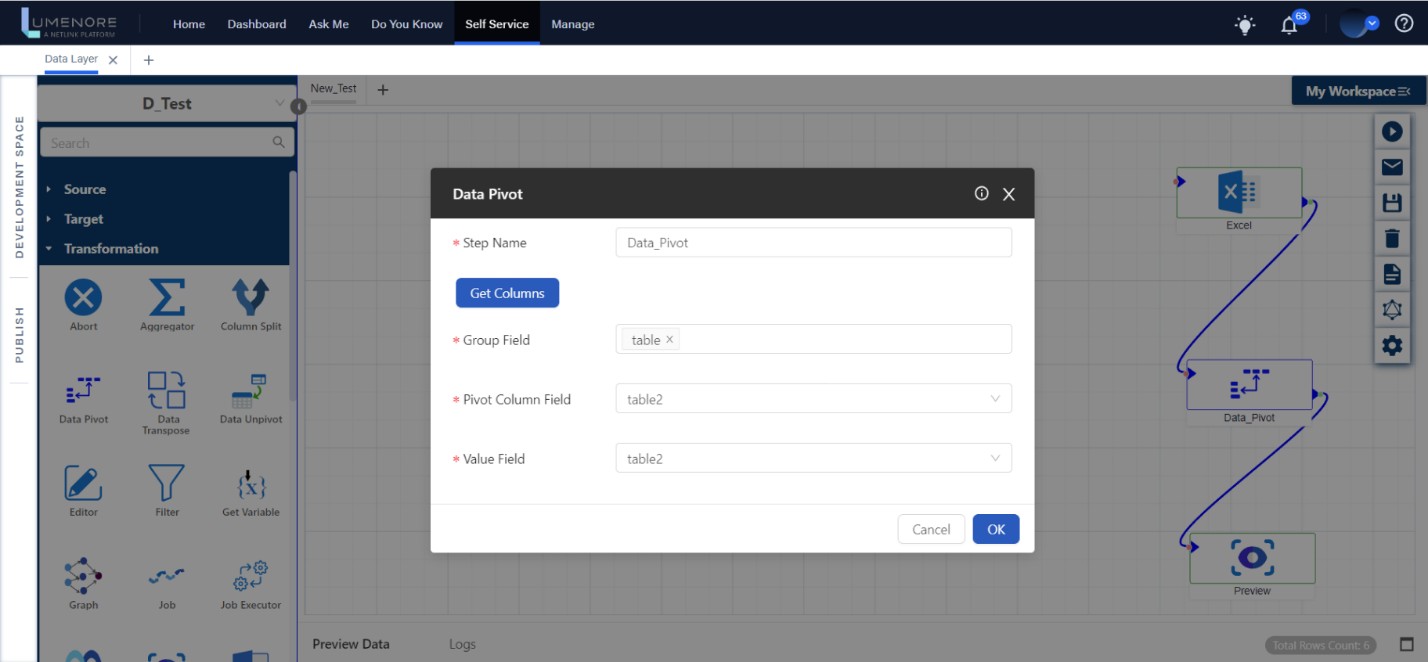
A screenshot of a computer

Description automatically generated

**How to use Column Split?**

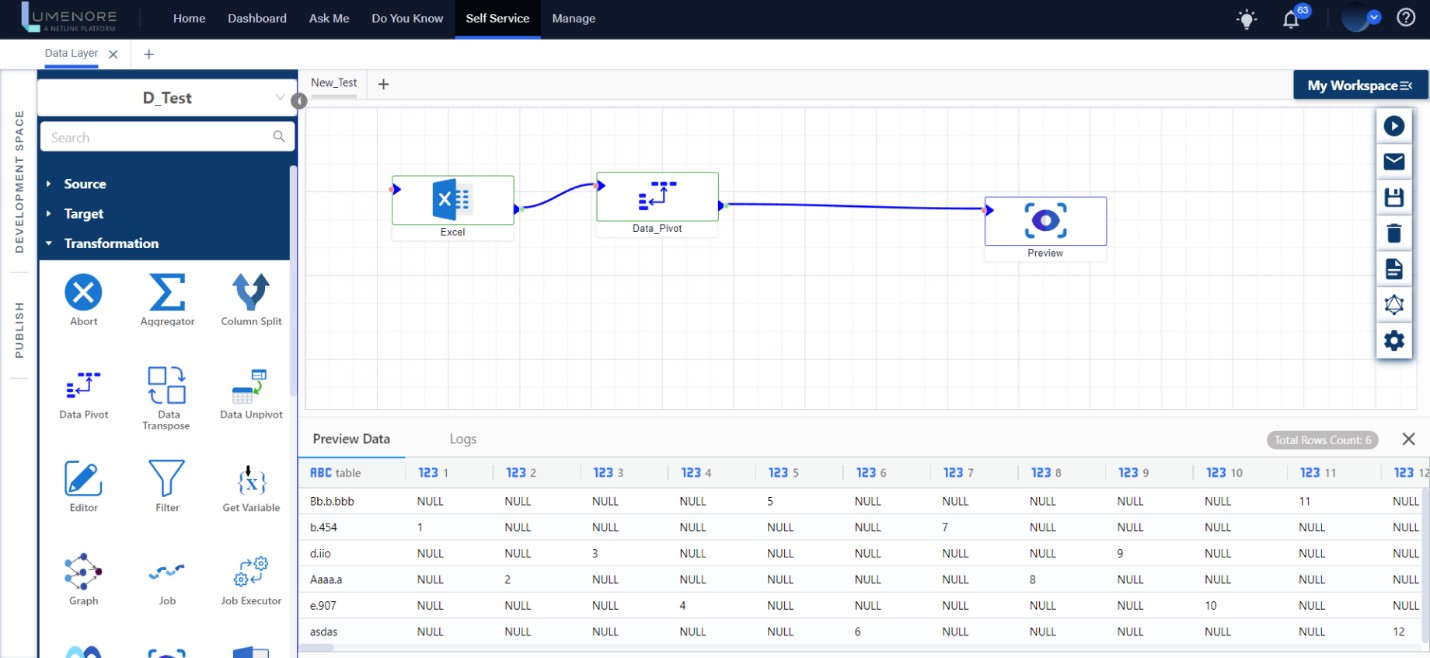
* In the "Step Name," you can give your file a custom name of your choice.
* In the "Split Column" field, you can provide the number of columns you want to split the data into.
* In the "Delimeter" field, specify the separator character that separates the data within the source column. Common delimiters include commas, semicolons, tabs, spaces, pipes (|), and more, depending on the format of your data.
* You can also add a column name by clicking the “Add Column Name” button.

**Data Pivot:** The Data Pivot Step serves to pivot a column, which can be understood as transforming a column into a row. After inputting our data through the source file, we achieve the desired output.

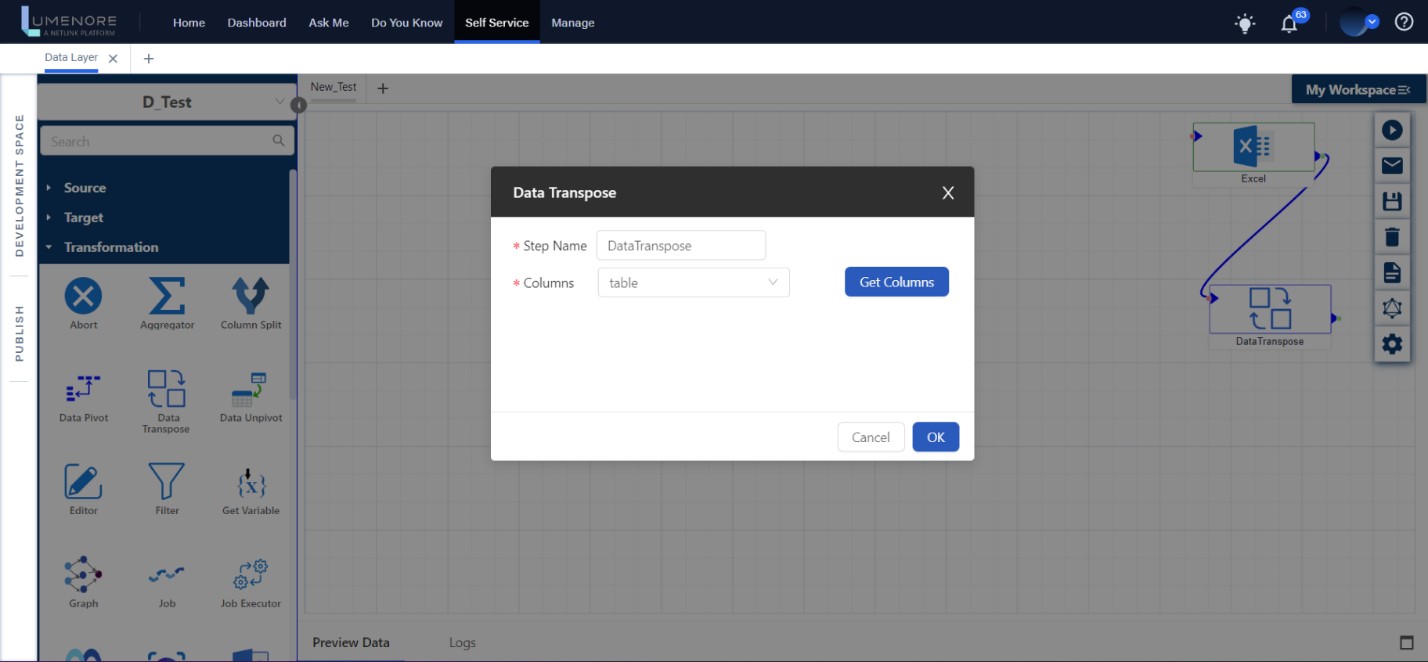


**How to use data pivot?**

* In the "Step Name," you can give your file a custom name of your choice.
* In the "Group" field, you select the specific columns you want to group to perform the aggregator function on that data.
* In the "Pivot Column Field," you will select columns other than those that have been grouped. This choice determines how the pivoted data will be displayed with that specific information.
* In the "Value Field," you can use the same column you selected in the "Pivot Column Field" to represent the data.

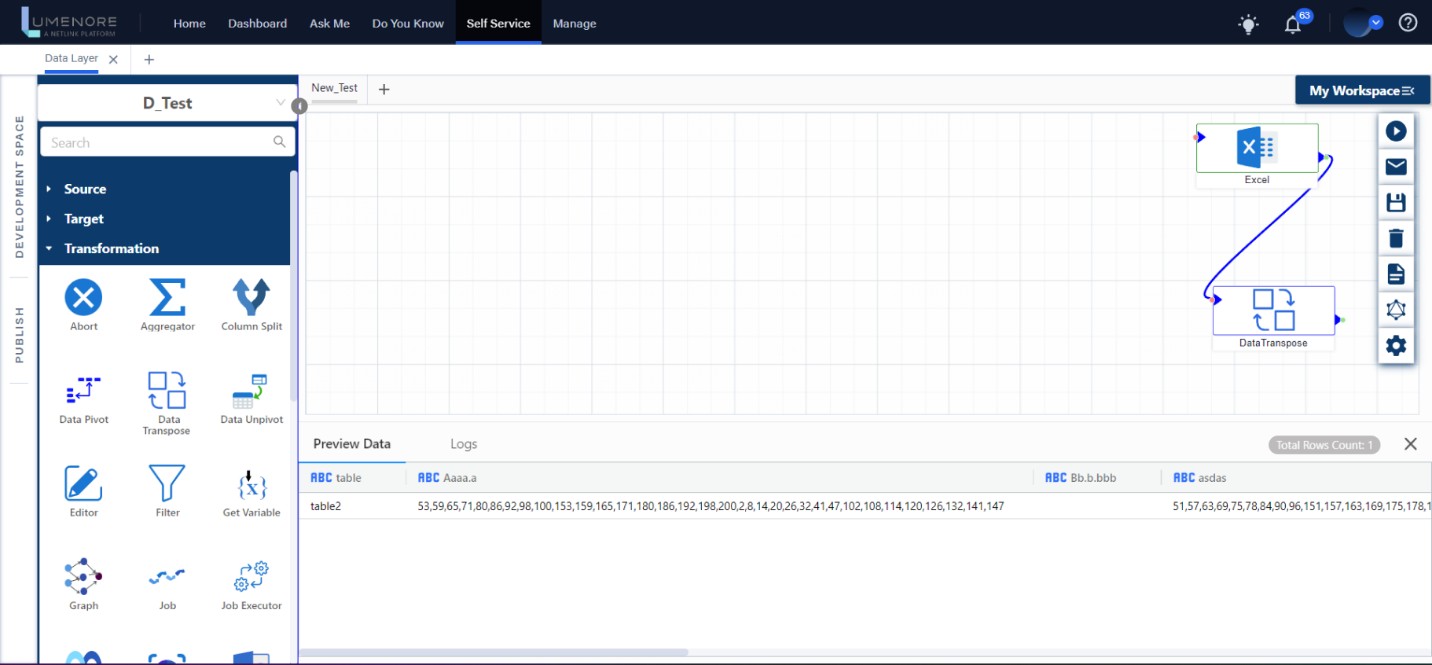


**Data Transpose:** Contrary to Data Pivot, the entire table is being transposed.



**How to use data transpose?**

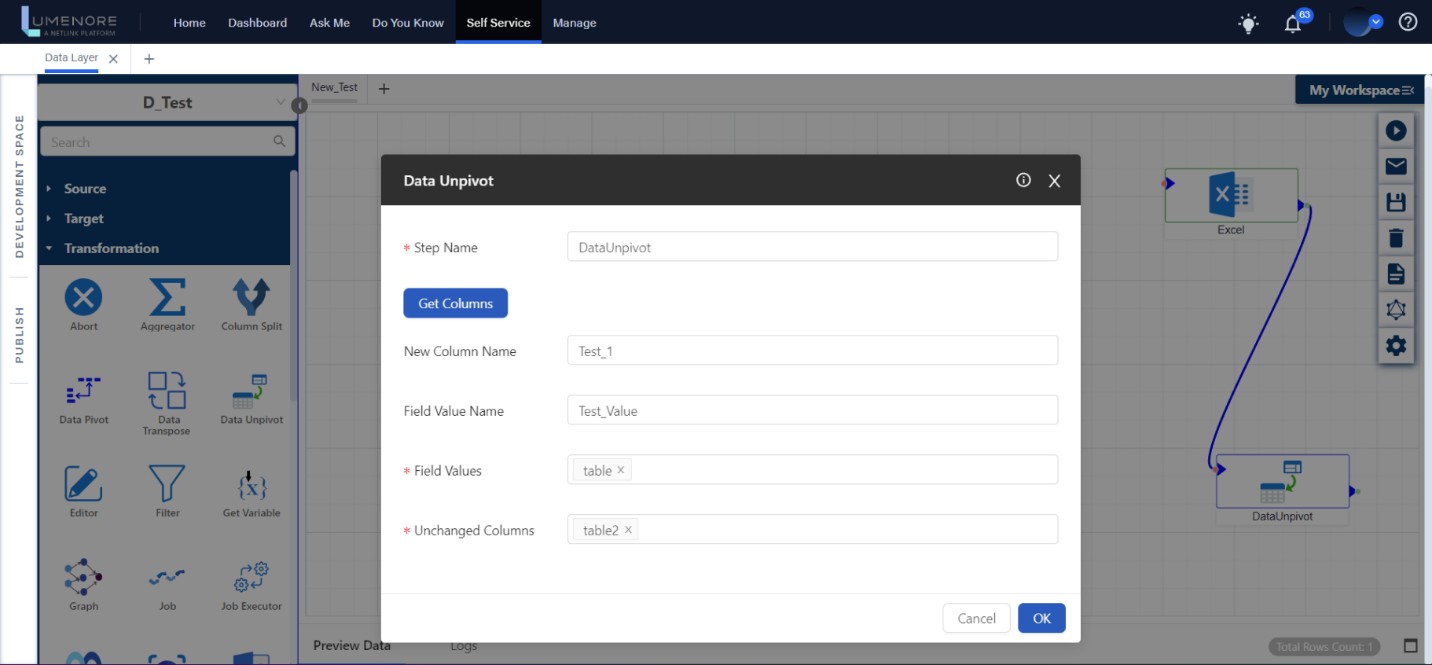
* In the "Step Name," you can assign a name to the file as you prefer.
* Next, click on "Get Column" to automatically import the data. From the dropdown menu, choose the specific column you want, and then click "OK."
* Finally, clicking on the "Run" option will generate the desired output.

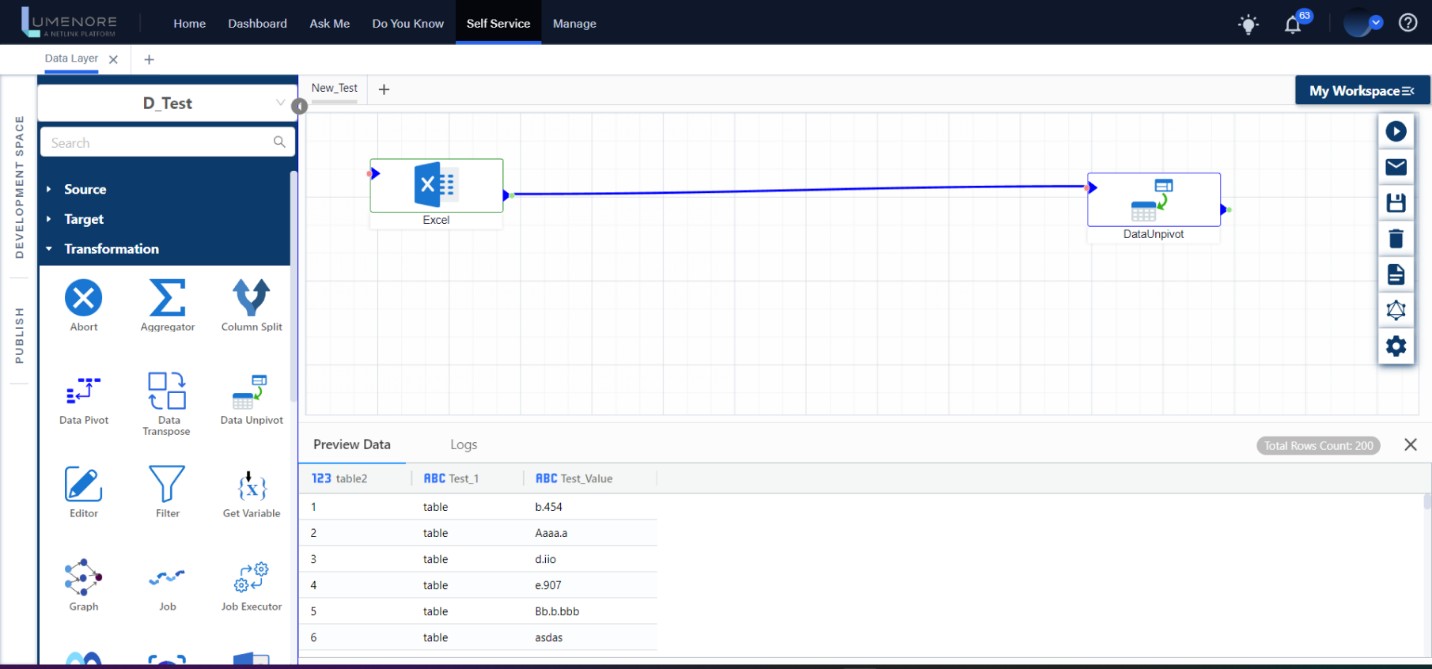


**Data Unpivot:** It operates oppositely compared to data pivot. While data pivot changes rows into columns, data pivot does the reverse, converting columns into rows. Think of it as if you used the data pivot transformation to load data into your system, and now you need data unpivot to revert it to its original form.

**How to use data unpivot?**

* In the "Step Name," you can provide a name for the file as you like. Then, click on "Get Column" to load your data.
* For "New Column Name" and "Field Value Name," simply enter the name you prefer for the column.
* In the "Field Value," select the column you want to transform and indicate the "Unchanged Columns" for those you want to keep as they are. Then, click "OK."
* Finally, execute the graph by clicking on "Run" to obtain the desired result.

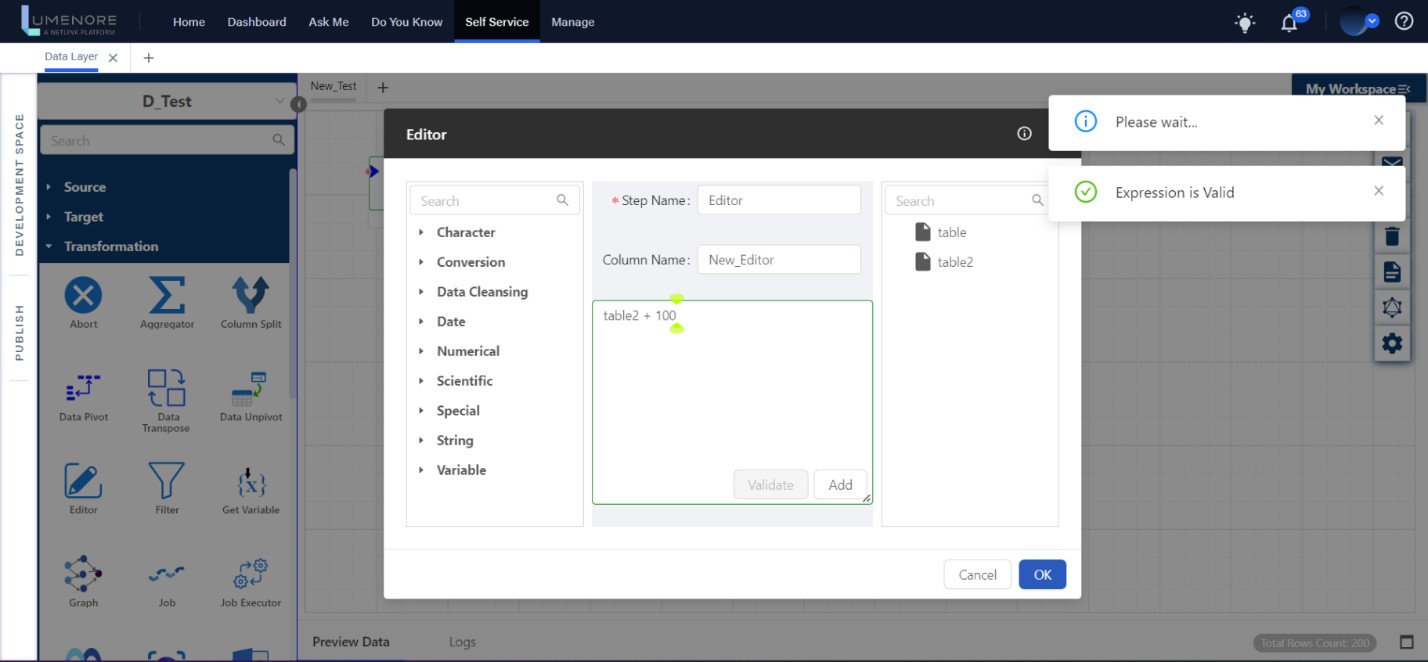




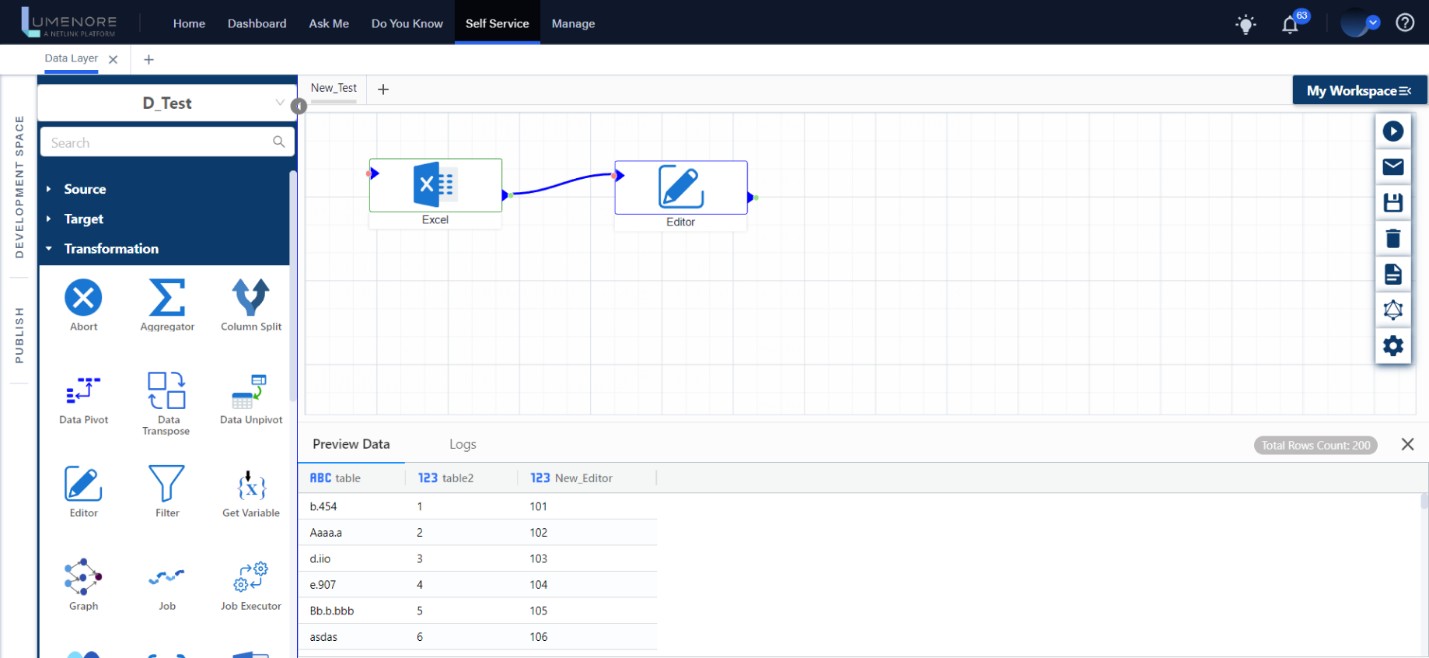
**Editor:** The editor step provides various functions for working with characters, strings, numbers, unique characters, and more. Each category offers numerous operations tailored to its specific type.

**How to use Editor?**

* To use the editor and create a new column with edited values, follow these steps:
* After selecting the editor operation, click "Get Column" to choose the necessary column from the dataset. In this example, we are selecting "Table 2."
* Next, pick the column where you want to act. The description will display it. You can perform actions like ADD, SUBTRACT, DATA CLEANSING, etc. For instance, in this case, we're using "Table 2" +100.
* Specify the desired column name in the "Column Name" field.
* After that, validate the expression using the "Validate" option. Once you receive a correct expression notification, click "Add" to include the column, and then click "OK" to proceed.



* After running the expression, the result will be the creation of a new column with the edited values, as illustrated below.



**Filter:** The filter step serves the purpose of selecting columns based on a provided expression. This step is employed for the specific task of filtering columns based on a given expression.

**How to use Filter?**

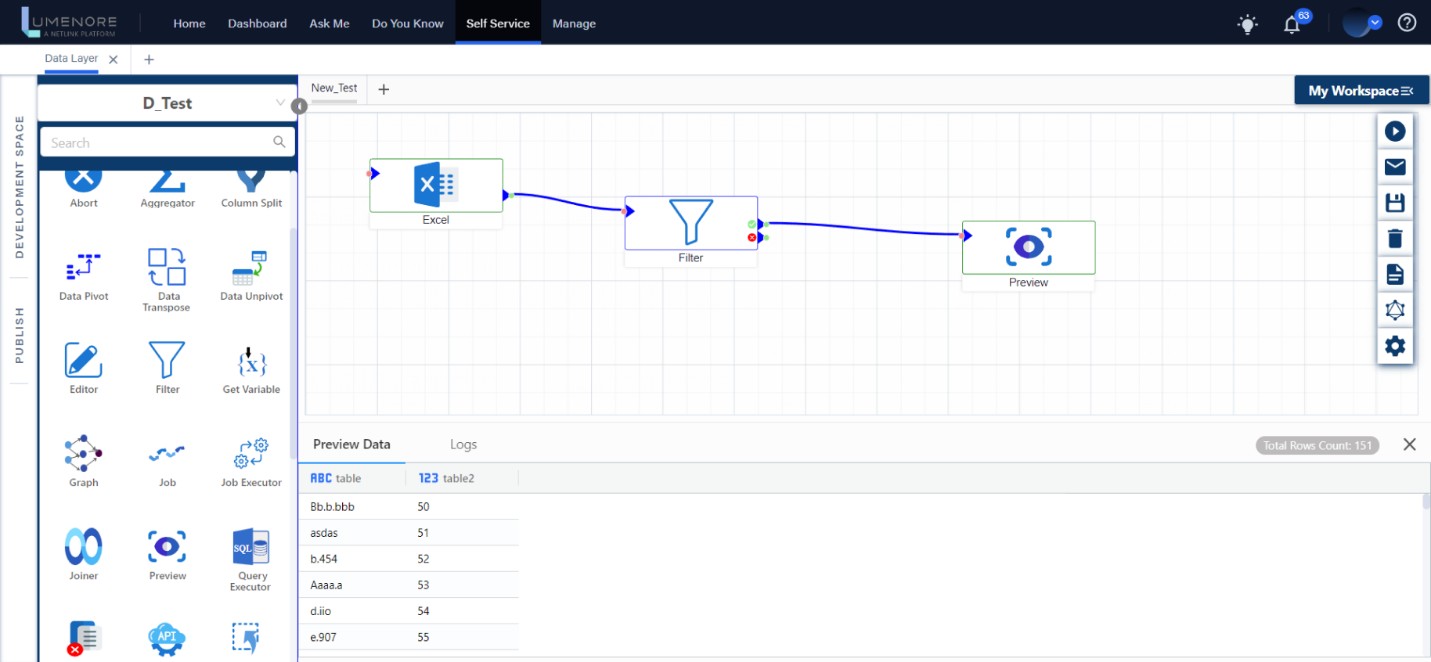
* Begin by selecting your data source, which can be Excel or any other preferred option. Then, use the filter transformation to execute an action based on a specified condition.
* Click on the Filter transformation, and after connecting it, provide a name for the expression. In this example, we have named it "FILTER\_NEW."
* Next, use the "Get Column" function to access and select different columns based on the given criteria.
* In the expression box, enter your desired condition, and then validate to ensure the expression is correct.
* Once you receive the correct notification, choose the "Update" option to add the desired code.
* Finally, click "OK" to apply the changes.

Top of Form

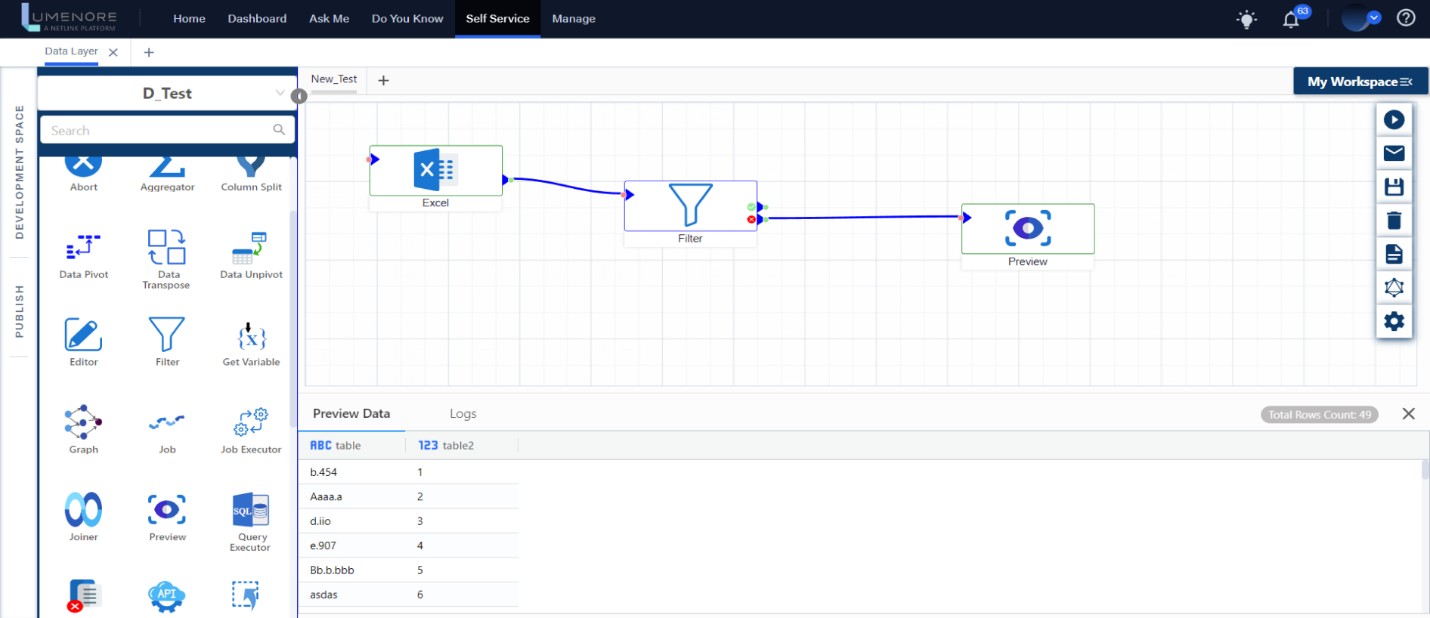


* As observed, the filter consists of two nodes: one is green, and the other is red.
* Green signifies that we want to apply the given expression when it evaluates to true. In this case, we aim to display data when Table 2 has values greater than or equal to 50.
* On the other hand, red signifies that we want to apply the expression when it evaluates to false. In this instance, we are interested in situations when Table 2 has values less than or equal to 50.
* After setting these conditions, you can run the expression to obtain the desired result.

**GREEN**



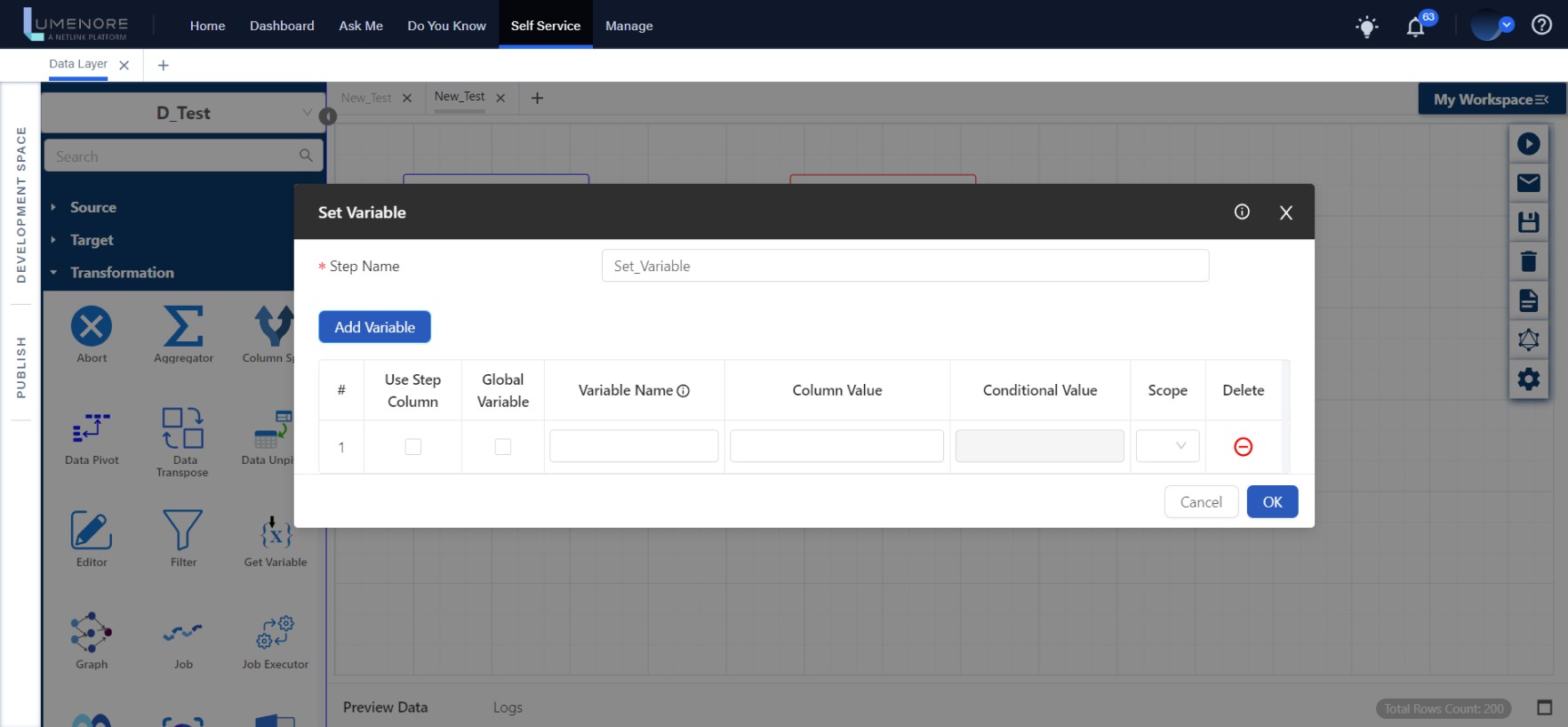
**RED**



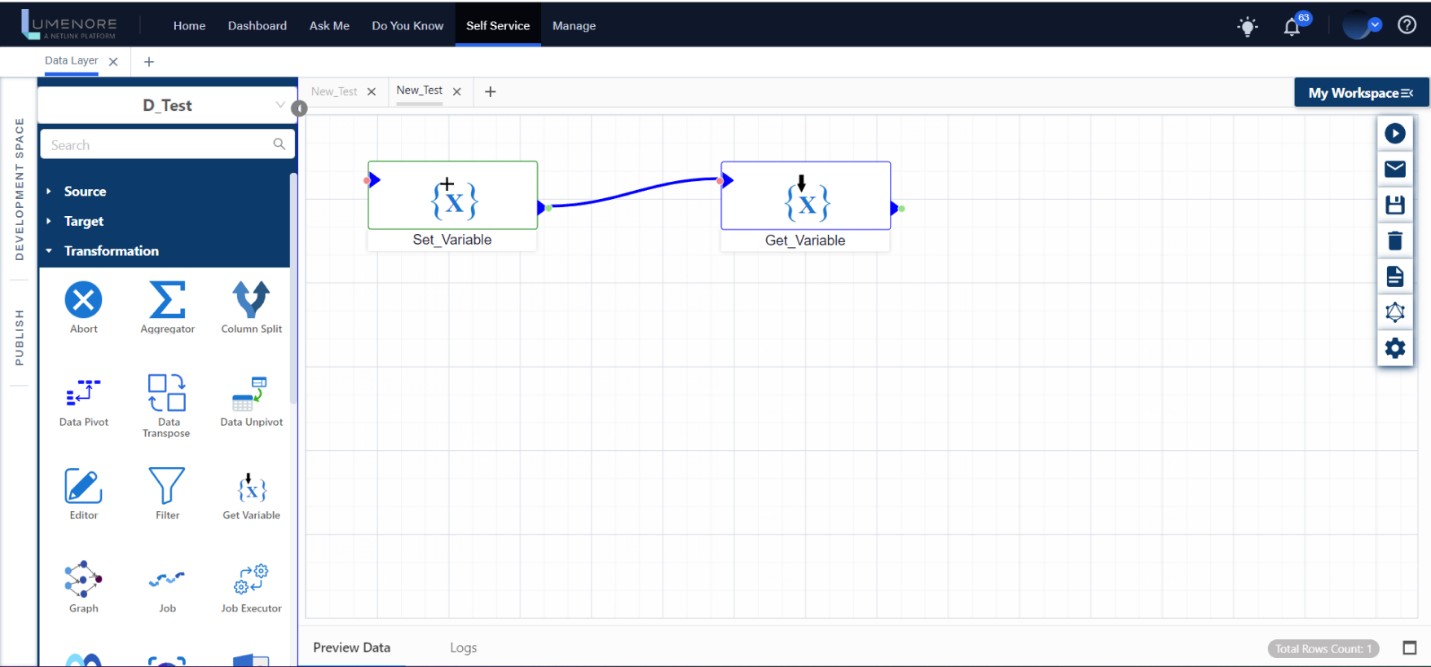
**Set Variable & Get Variable:** It is used to manually set any variable value per our choice, and the get variable is used to get the same value as the output.

**How to use Set and Get Variable?**

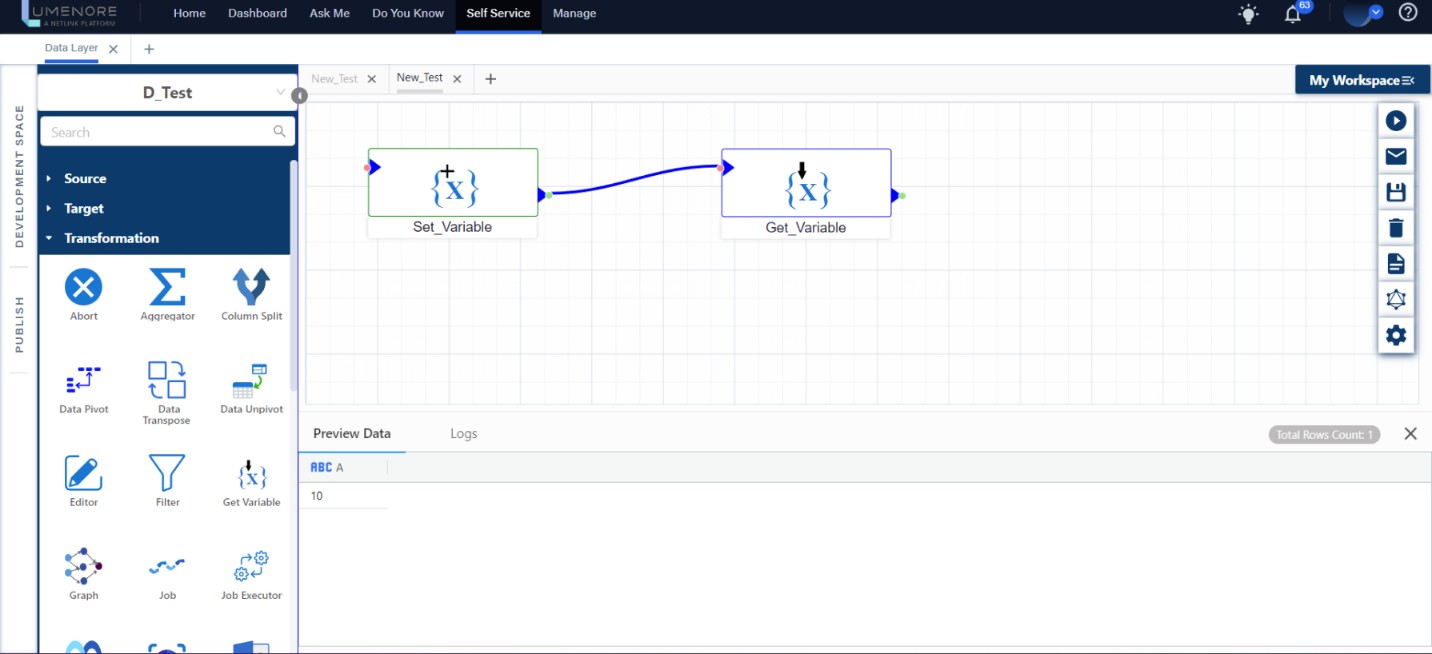
* Begin by selecting "Set Variable" from the transformation panel.
* Next, specify the desired name for this step in the "Step Name" field.
* Click "Add Variable" to include the column as needed.



* Start by providing a variable name of your choice, for example, "A." Then, specify the desired column value, which in this case is 10 (based on your preference). For the scope, select "graph" since we are applying the entire transformation with output.
* Click "OK," and then proceed by clicking on "Get Variable" and connecting both nodes.



* After selecting "Get Variable," specify the preferred name for this step.
* Ensure that the "Variable Name" and "Scope" match the values you previously provided in the "Set Variable" step to avoid errors.
* Finally, click "OK" to obtain the desired output and run the expression.

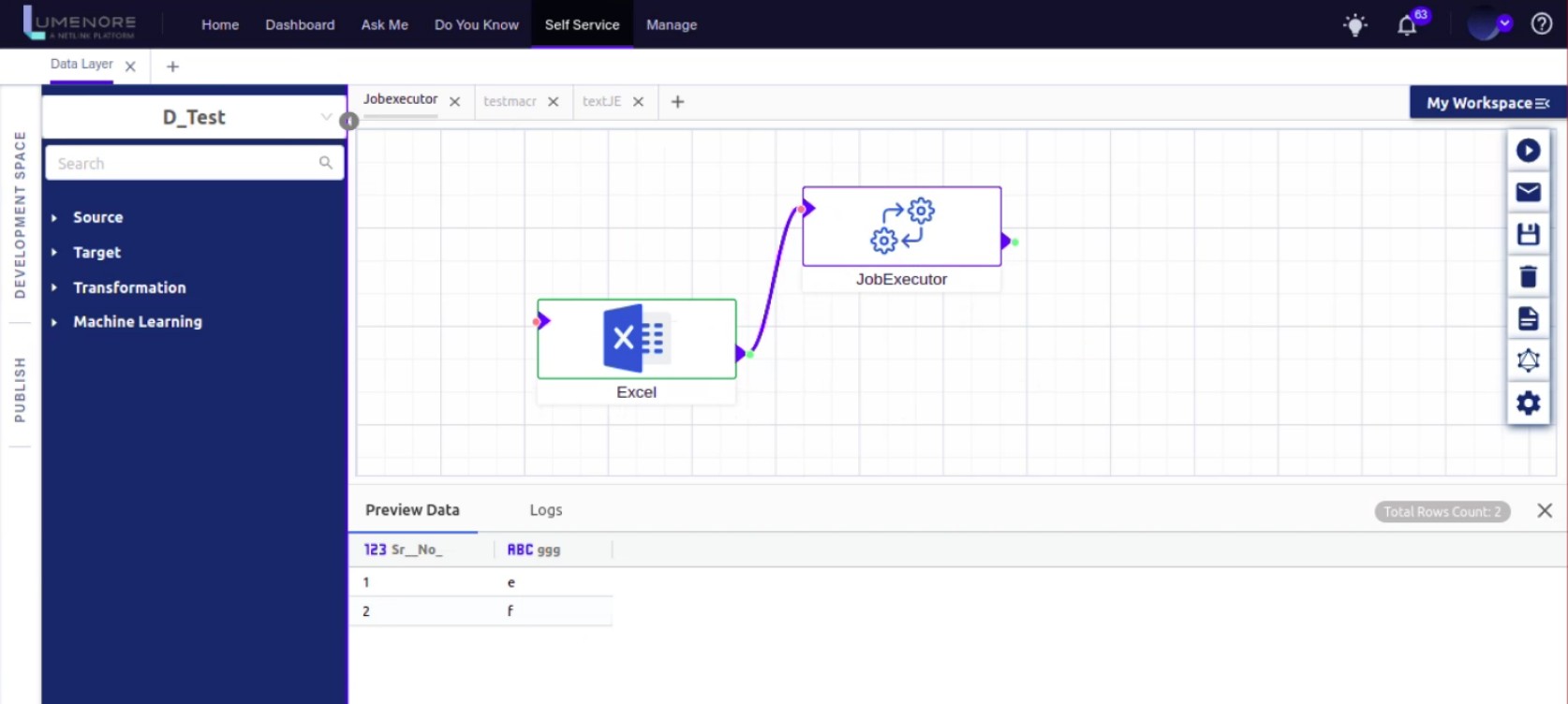


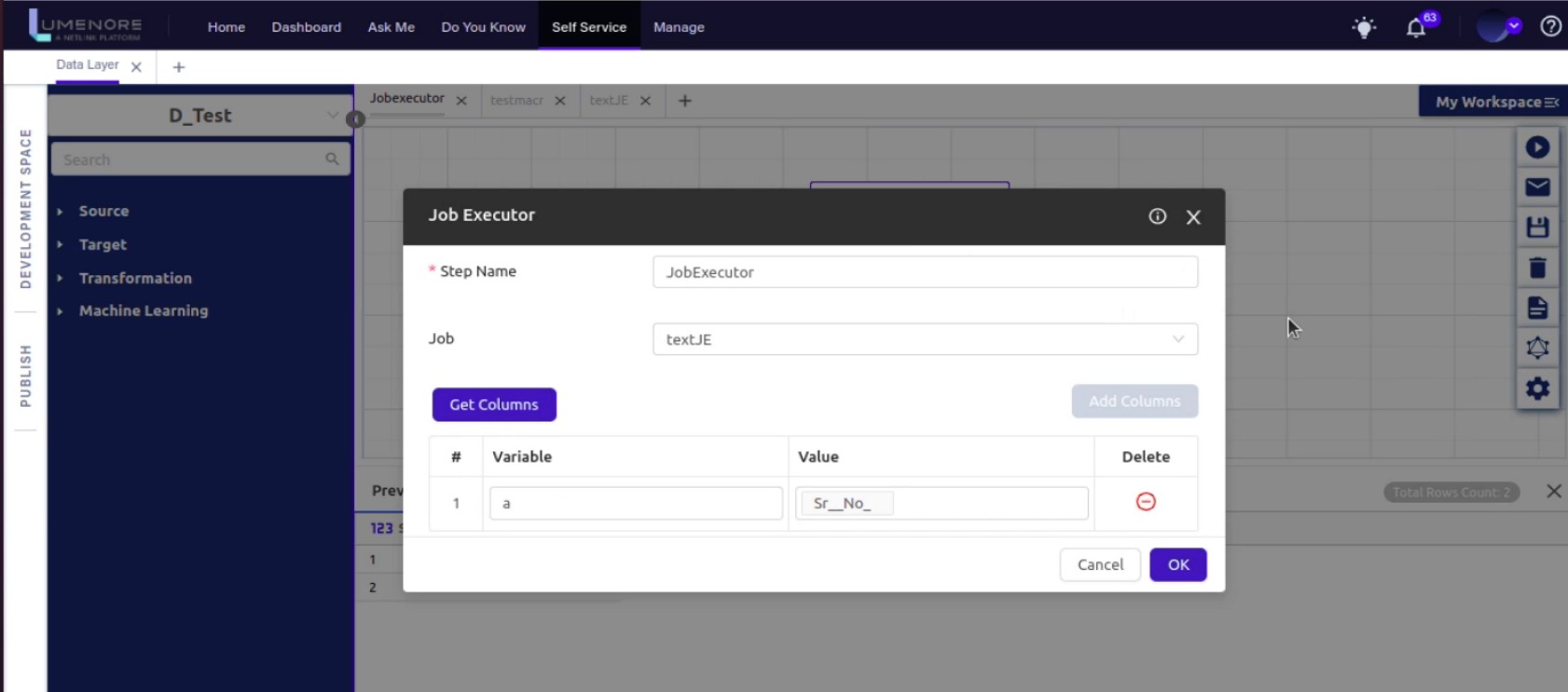
* You can apply the same steps to set variable details according to your requirements, even when using Excel or any other input source before the "Set Variable" step.

**Job Executor:** The Job Executor step facilitates the execution of a job by passing a variable with new data for each row of the source data.

**How to use Job Executor?**

* Start by selecting the source to which you want to apply the job executor.
* This function will run as many times as there are rows in the Excel file.
* Next, you can provide a name of your choice in the "Step Name."
* Then, create the variable by specifying its name and value according to your preferences.
* Finally, execute the query.

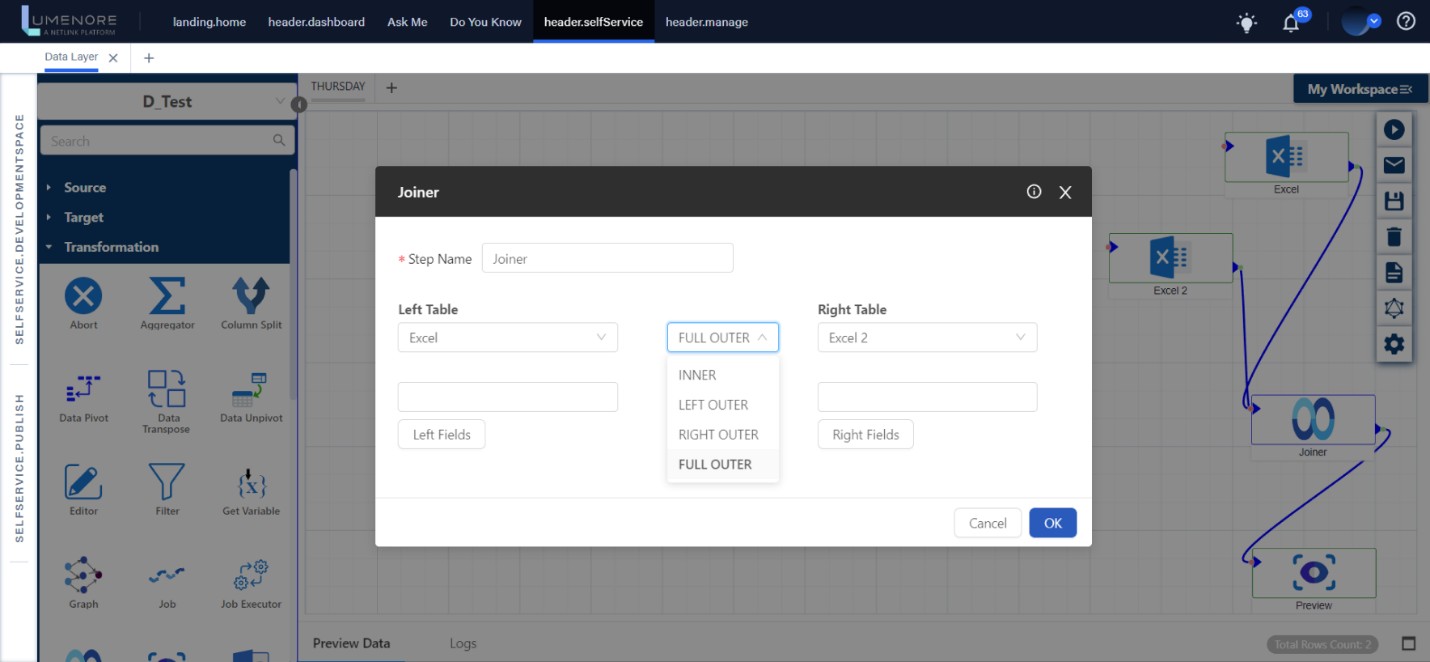


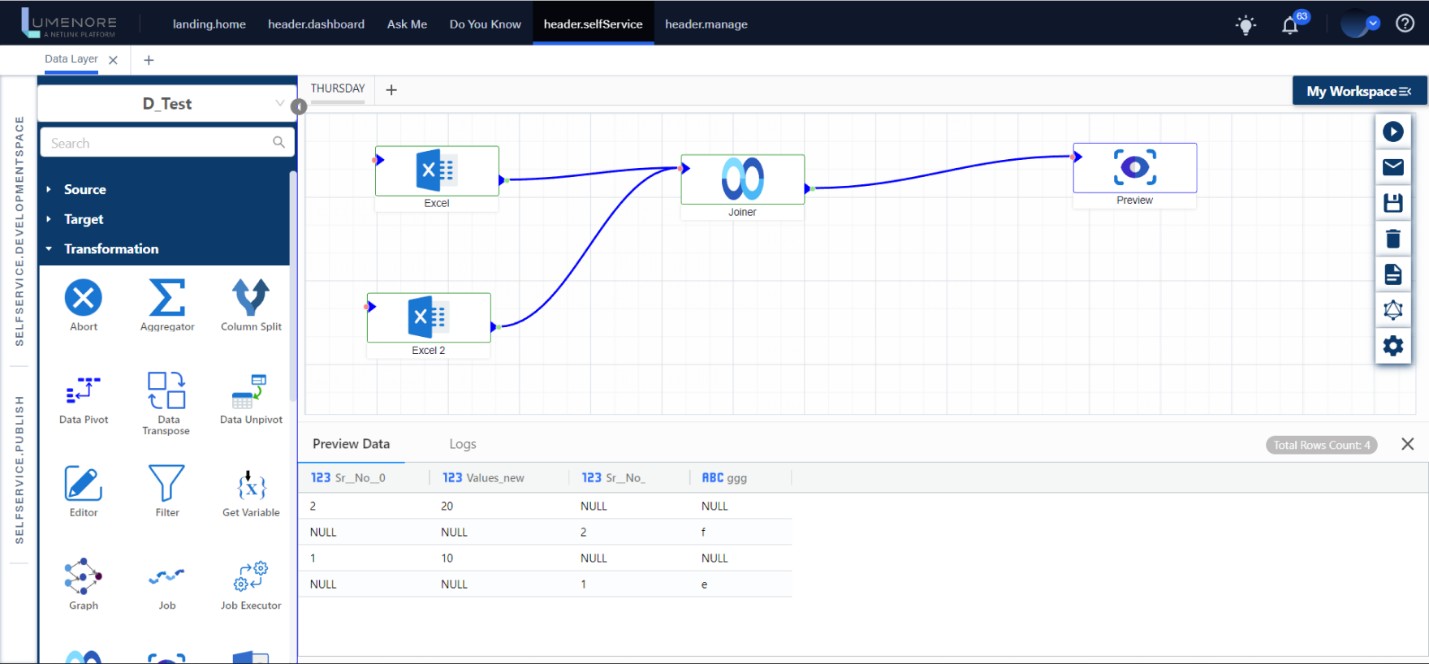


**Joiner:** This step conducts a join operation, merging columns from two tables. Users can choose from several types of joins listed, and the "Left Fields" and "Right Fields" display columns from their respective input tables. To illustrate, let us consider an example where we join two distinct Excel files and execute the operation.

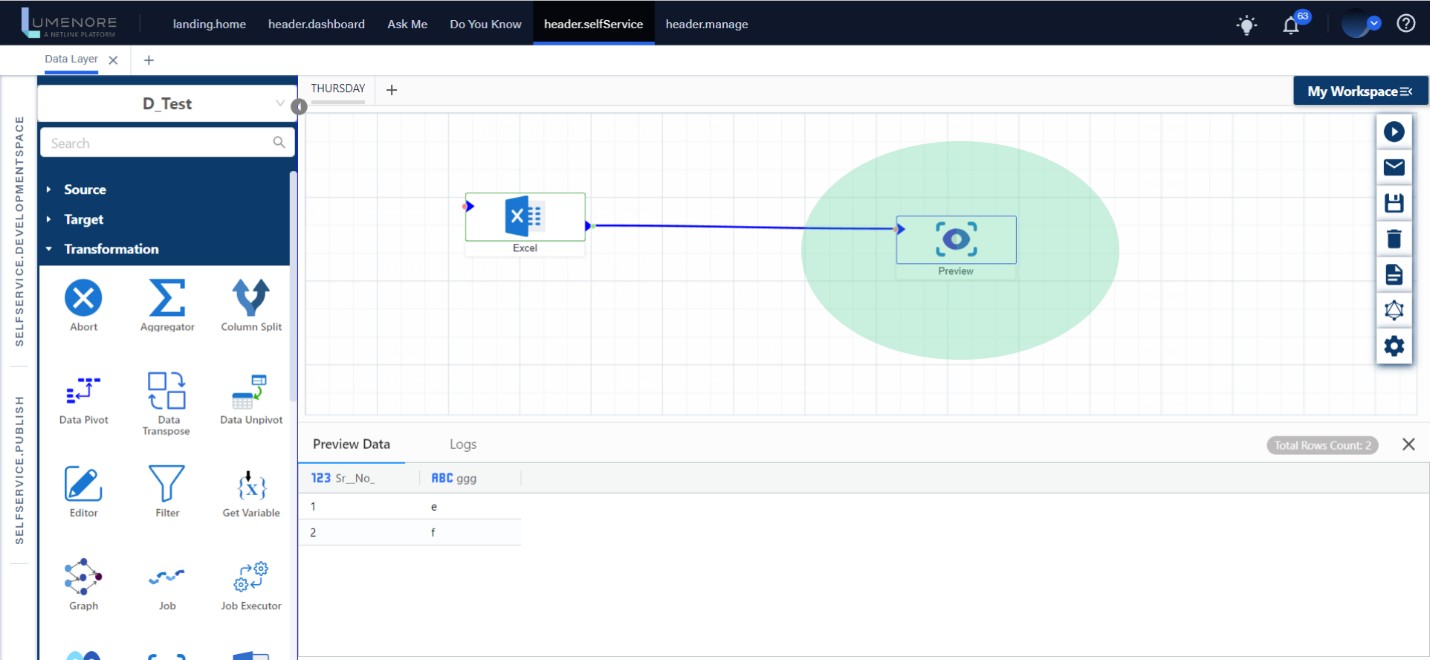
**How to use Joiner?**

* Begin by selecting the Excel files and using the Joiner step. To proceed, double-click on the Joiner.
* In the "Step Name" field, give the file a name of your choice.
* In the "Left Table" and "Right Table," choose different Excel files as needed.
* Next, select the join type from the drop-down menu.
* Then, click on "Left Fields" and "Right Fields" to access the sheets from the Excel files.
* Finally, click on "Run" to execute the operation.





**Preview:** The Preview Step makes it easier to see the Source's data.



**Query Executor:** This step is designed for executing SQL queries. Users can follow these steps:

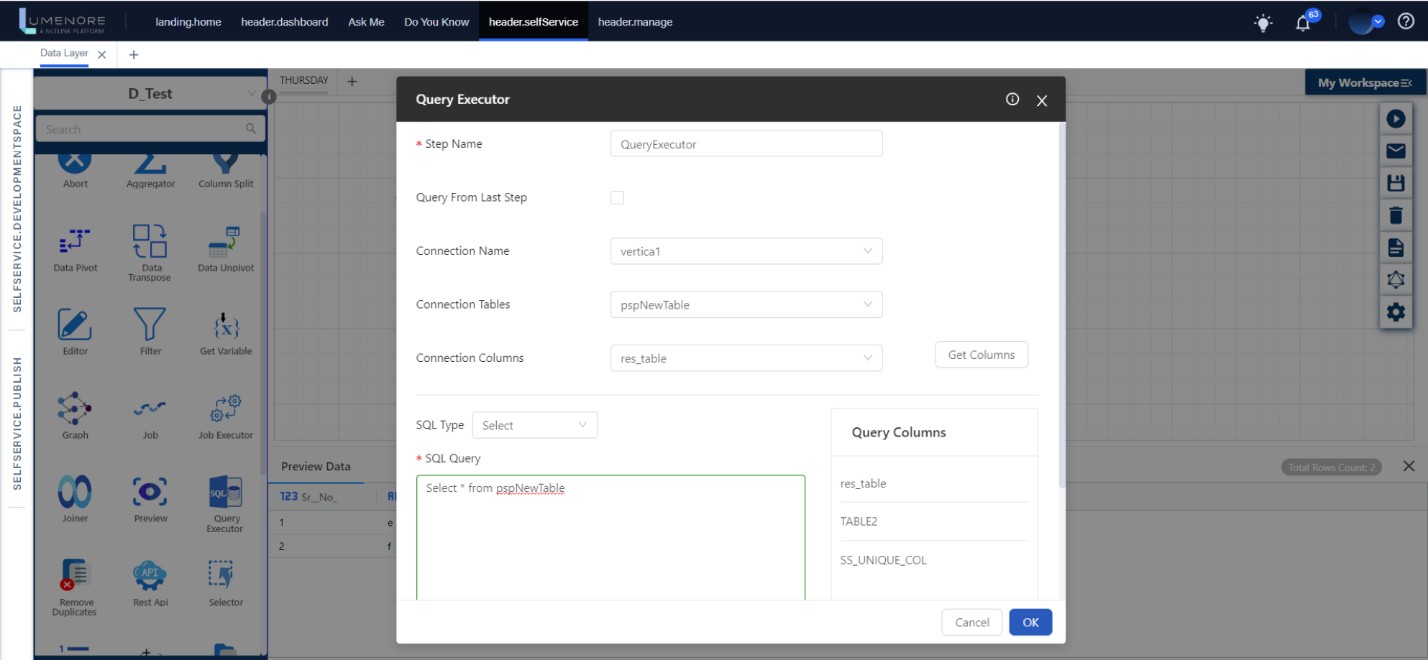
1. Select a connection table and view the respective table columns by clicking the "Get Columns" button.
2. Choose the SQL type, which can be either "select" or "non-select."
3. In the text area, validate and input the SQL query.
4. Retrieve the output column based on the query.

This step allows users to perform SQL queries by selecting a connection table, viewing table columns, selecting the query type, validating, and entering the query, and fetching the query's output column.

**How to use Query Executor?**

* Start by opening the connection from your workspace and then access the query executor.
* The query executor enables you to execute SQL queries based on the available connections.
* In the "Step Name" field, provide a name as per your preference.
* From the "Connection Name" dropdown, select the desired connection type. In this example, we're selecting "vertica1."
* Choose the appropriate "Connection Table" from the dropdown.
* Click on "Get Column" to retrieve the connection columns.
* Select the SQL type, either "Select" or "Non-select."
* Write your SQL code, validate it, and fetch the columns.
* Observe the number of columns available in the "Query Columns" panel.
* After selecting from the "Query Columns" panel, proceed with "OK."
* Finally, run the graph to execute the SQL query.

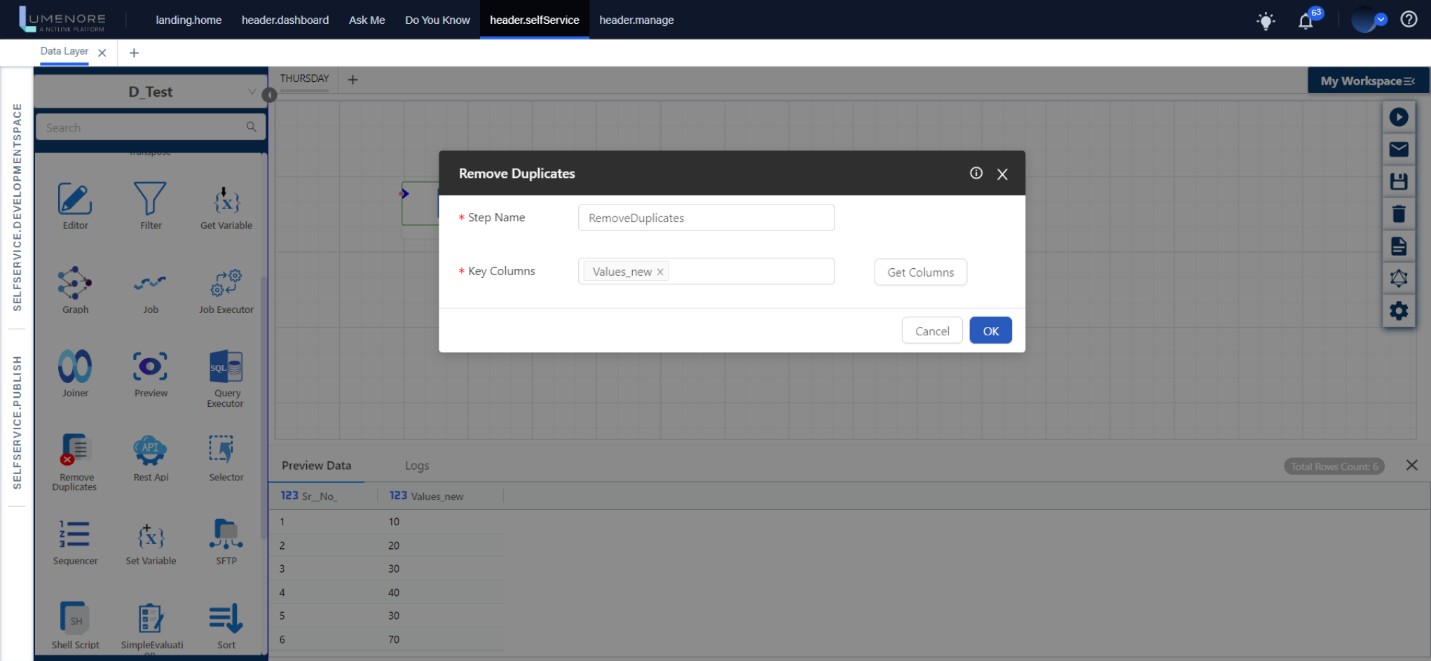
Top of Form



**Remove Duplicates:** This step eliminates duplicate records from a column and offers the functionality for multi-selection.

**How to use Remove Duplicates?**

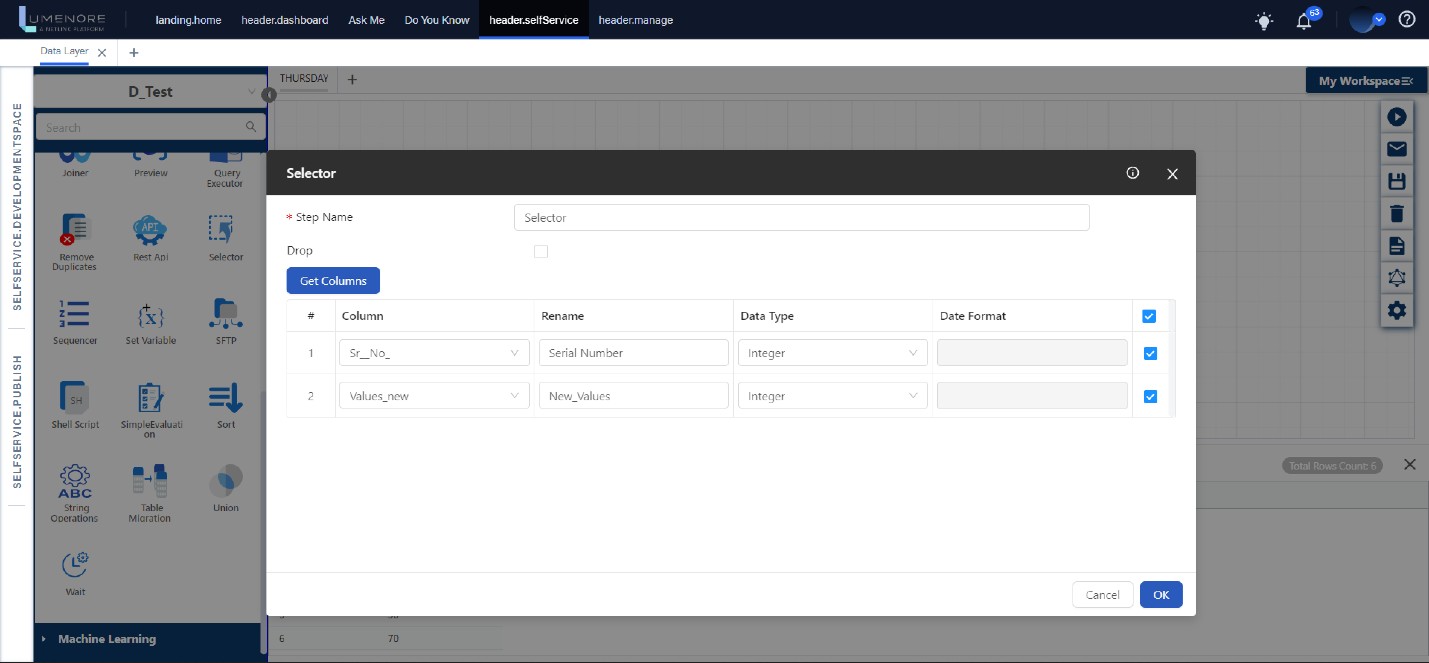
* Import the "Remove Duplicates" step from the transformation panel.
* In the "Step Name" field, give it a name of your choice.
* Use the "Get Column" function to extract and select the columns to be used as key columns.
* Finally, run the query to remove duplicates based on the specified key columns.



**Selector:** The Selector step allows you to choose multiple columns that you need for the subsequent ETL process. You can use the add (+) and subtract (-) buttons to either include a new column or remove a column that has already been selected.

**How to use Selector?**

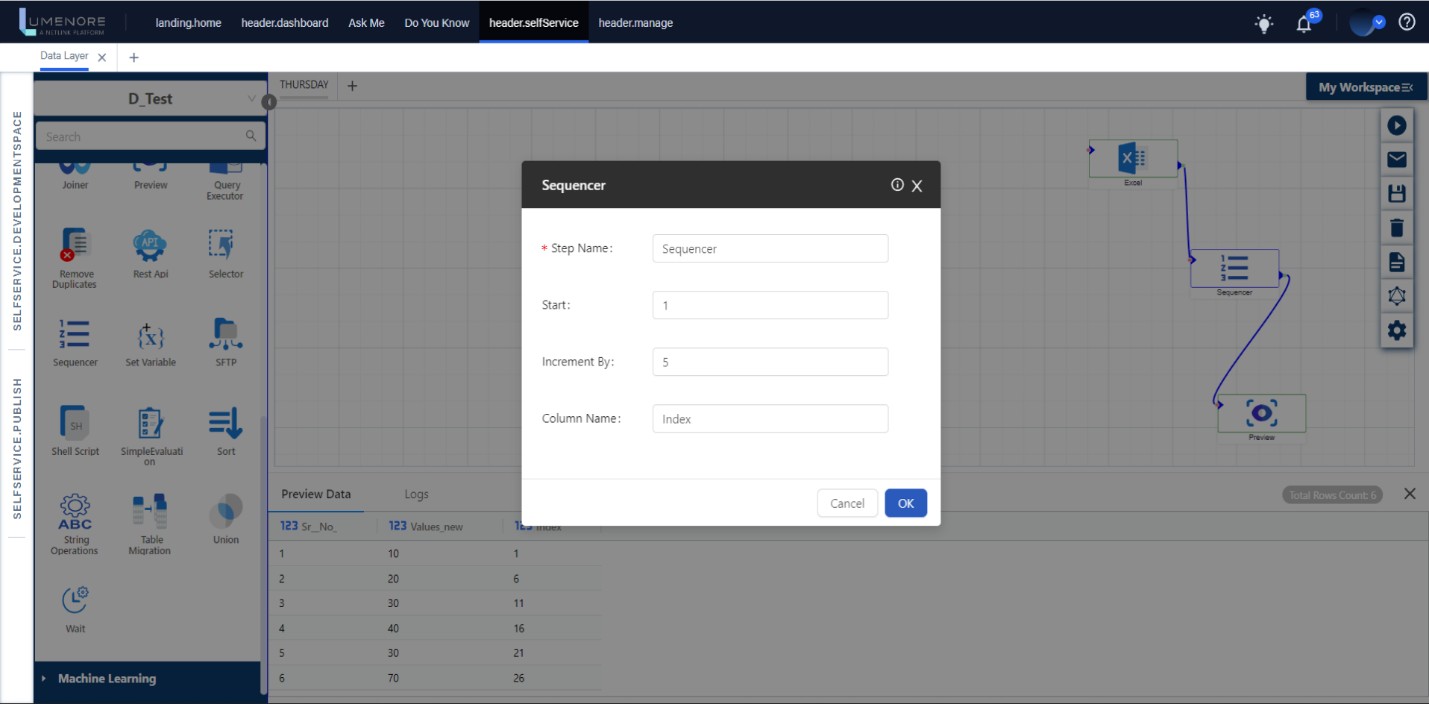
* Click on the Selector step, and in the "Step Name" field, provide a name of your choice.
* Use the "Get Column" function to extract columns from the source file.
* You can then rename and change the data type of these columns as per your preferences, such as Integer, Long, String, etc.
* Click "OK" to confirm your selections, and then run the query to obtain the desired output.



**Sequencer:** The Sequencer step offers a way to input data with a specific increment or sequence.

**How to use Selector?**

* **Start Value:** This is where you specify the initial value from which you want the sequence to begin. For example, if you want it to start at one, you'd enter "1."
* **Increment Step:** Here, you define the amount by which the sequence should increase with each step. For instance, if you want to increment by 2, your sequence will be 1, 3, 5, 7, and so on.
* **Column Name:** This is where you provide a name for the column where the sequence will be generated, as chosen by the user.

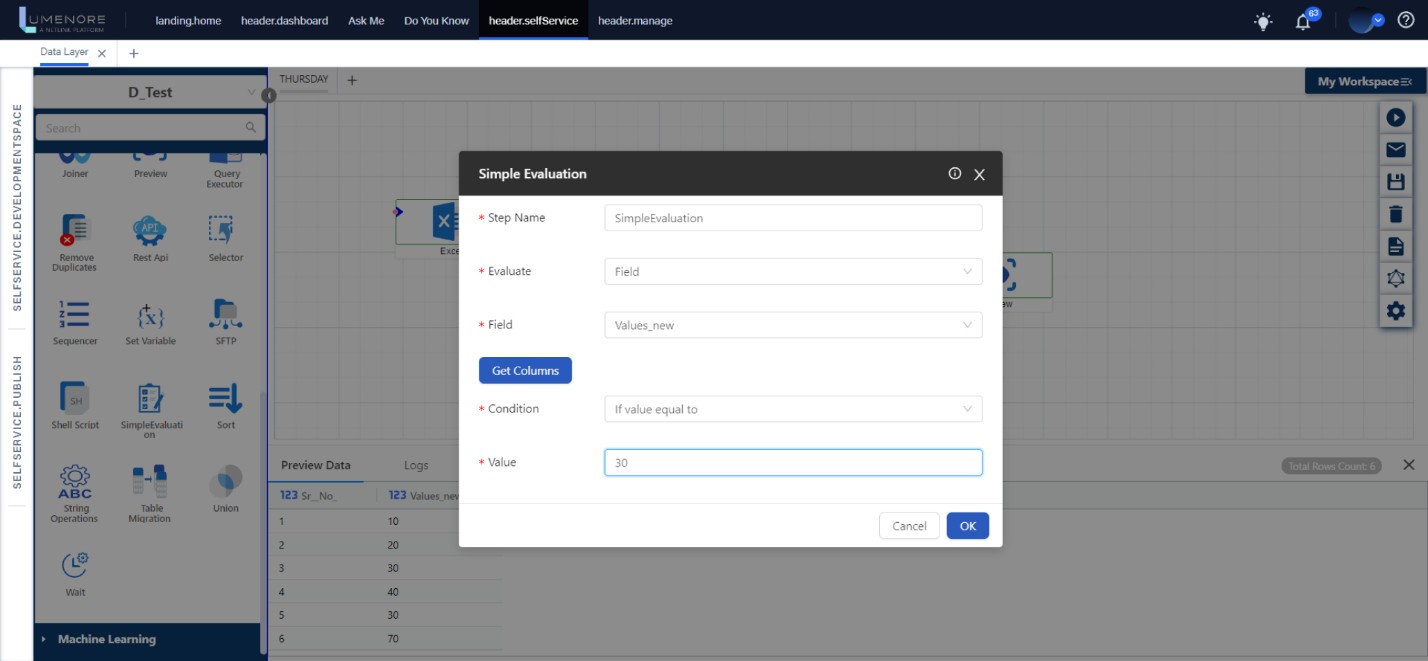


**Simple Evaluation:** The Simple Evaluation step is designed to assess straightforward conditions. It provides the following features:

* You can apply conditions to both variables and column values.
* It offers a variety of conditions to choose from within the "Condition Select" option.
* You can specify a value for comparison using the "Value" option.

**How to use Simple Evaluation?**

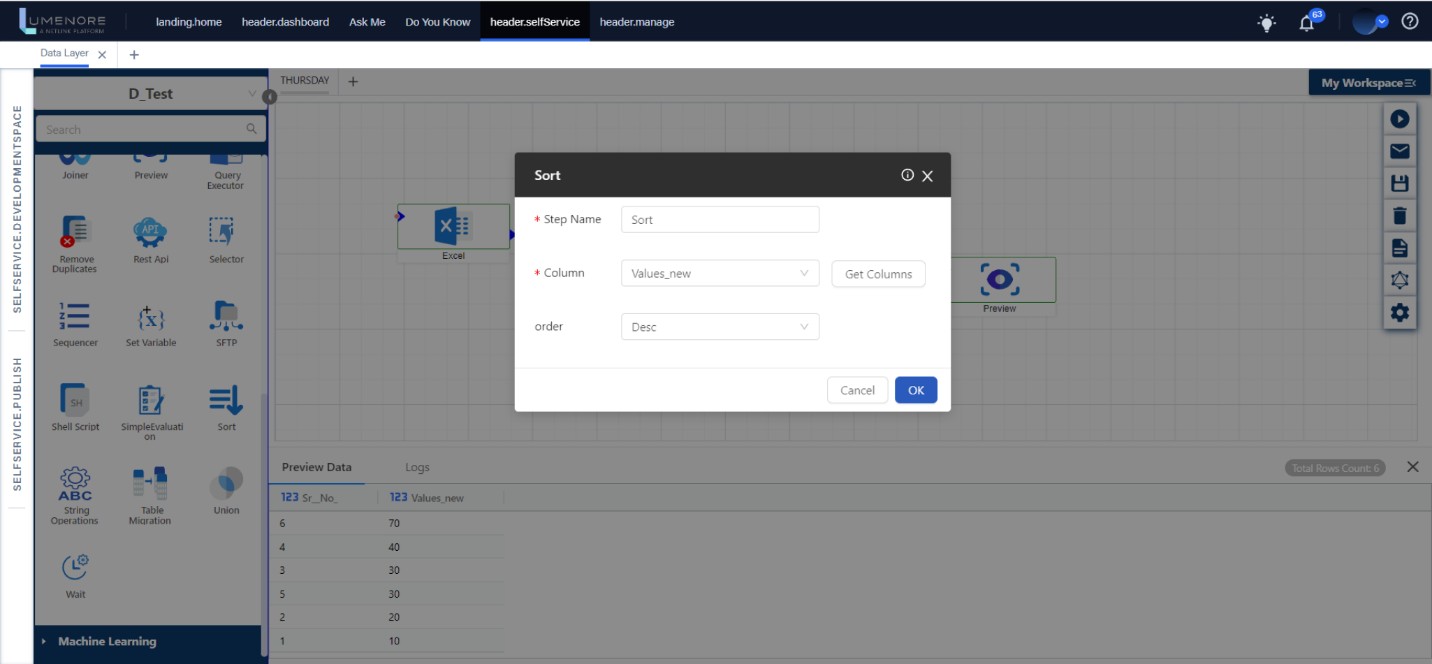
* In the "Step Name," provide a name of your choice.
* In the "Evaluate" section, you have the flexibility to use either a field or a variable.
* In the "Field" section, specify the column from the previous source that you want to examine.
* Then, define the condition and input the value you want to check against.



**Sorting:** The Sort step is employed to arrange a column either in ascending or descending order. Users can access columns by simply clicking the "Get Columns" button.

**How to use Sorting?**

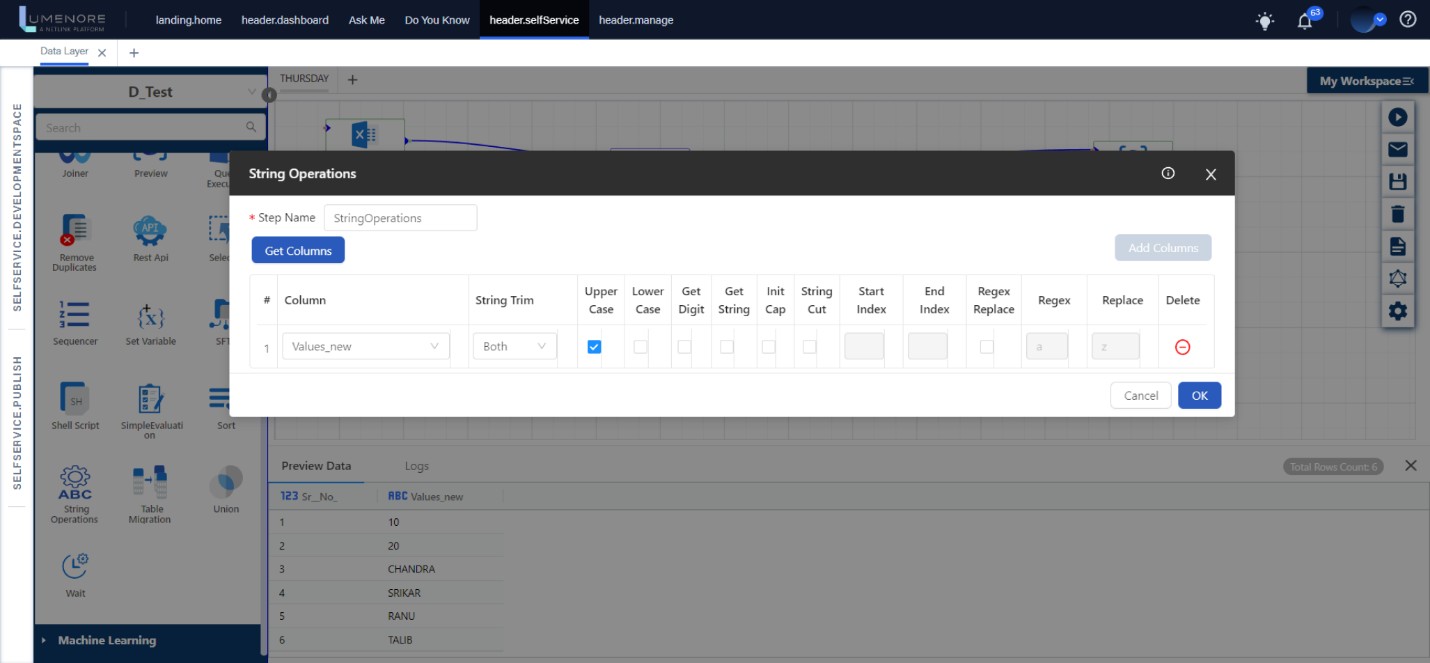
* After selecting the "Sorting" step in the "Step Name," you can specify the data according to your preference.
* Click on "Get Column" to retrieve the columns from the source.
* Lastly, determine the order in which you want to display your data. After providing this input, execute the query accordingly.



**String Operations:** The String Operations step allows you to apply various string functions to a single column individually. These functions include InitCap(), UPPER(), LOWER(), Digits only, and String only. Users can select the checkboxes based on their specific requirements.

**How to use String Operations?**

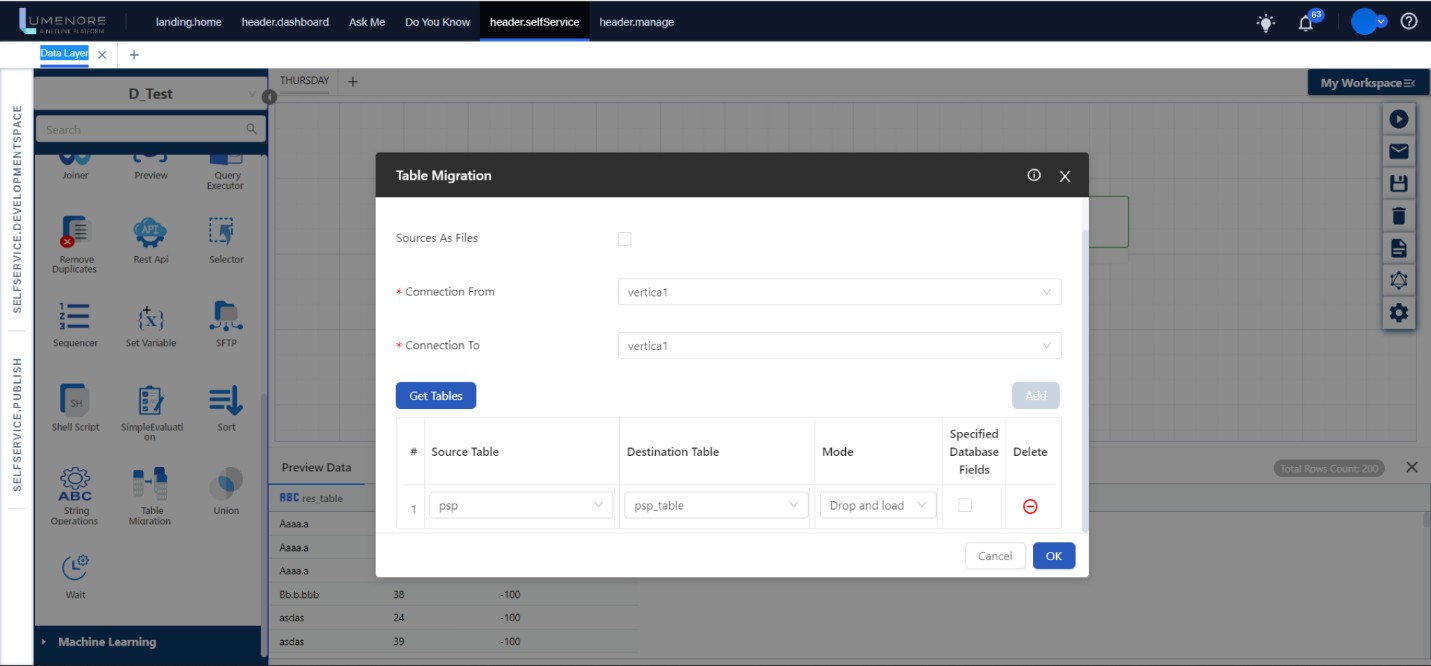
* After selecting "String Operations" in the "Step Name," you can assign it a name of your choice.
* By clicking on "Get Column" and "Add Column," you can select the column that contains string data and add it to the step.
* In the "Column" section, select the column on which you want to perform the specified action.
* You can then apply various functions such as String Trim, Upper Case, Lower Case, etc. Once you've made your selections, click "OK" and run the query.



**Table Migration:** It migrates the complete table from one database to another.

**How to use Table Migration?**

* In the "Step Name," you can give it a name of your choice.
* Specify the source and destination tables using the "Connection From" and "Connection To" options.
* To add the table for extraction, click on "Get Table" and then "Add."
* Select the table you want to extract data from and load it into the destination table.
* Finally, execute the query to obtain the desired output.



**Union:** The Union step functions just like the UNION operator in SQL, allowing you to merge the result sets of two or more SELECT statements.

**How to use Union?**

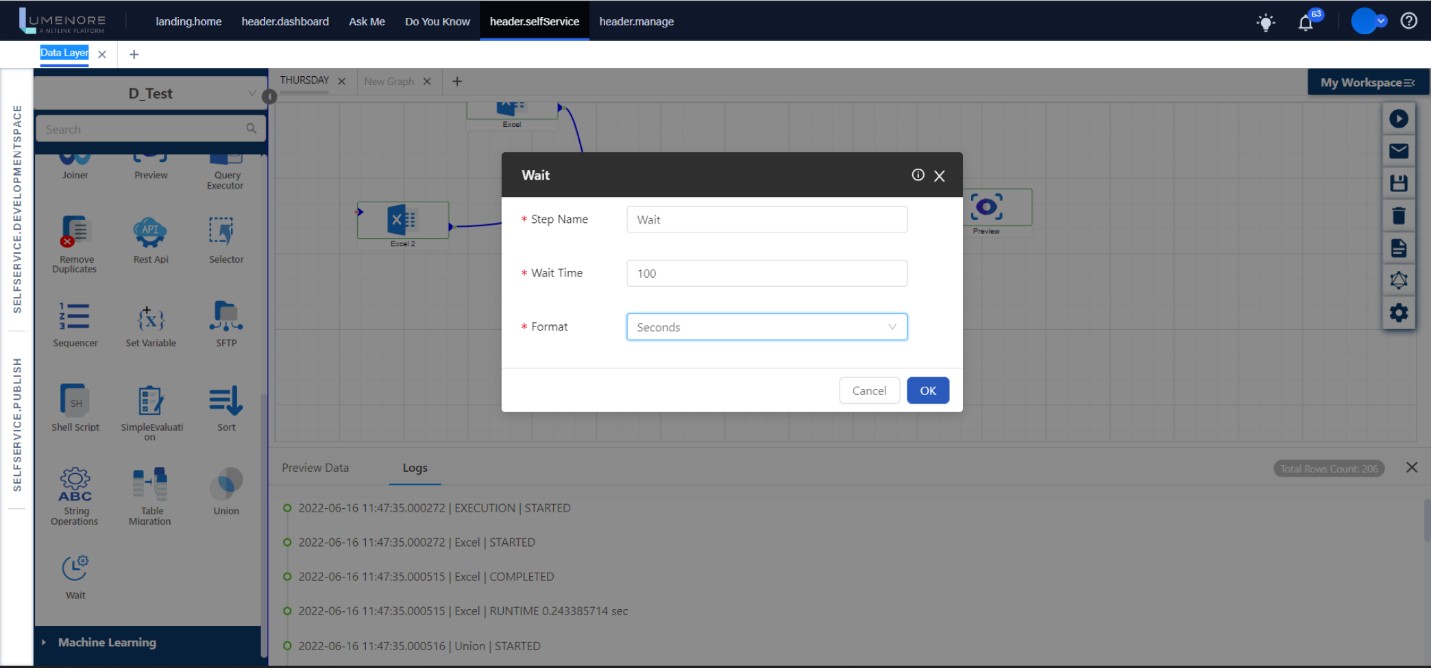
* To perform this action, you need two or more data sources.
* After selecting the "Union" step, specify a name in the "Step Name" field according to your preference.
* In the "Distinct" option, you can choose between "True" or "False."
* You can determine whether you want to prioritize the top or bottom table based on your needs.
* Add or retrieve the columns as necessary, input the details according to the data type, and then run the operation to execute it.

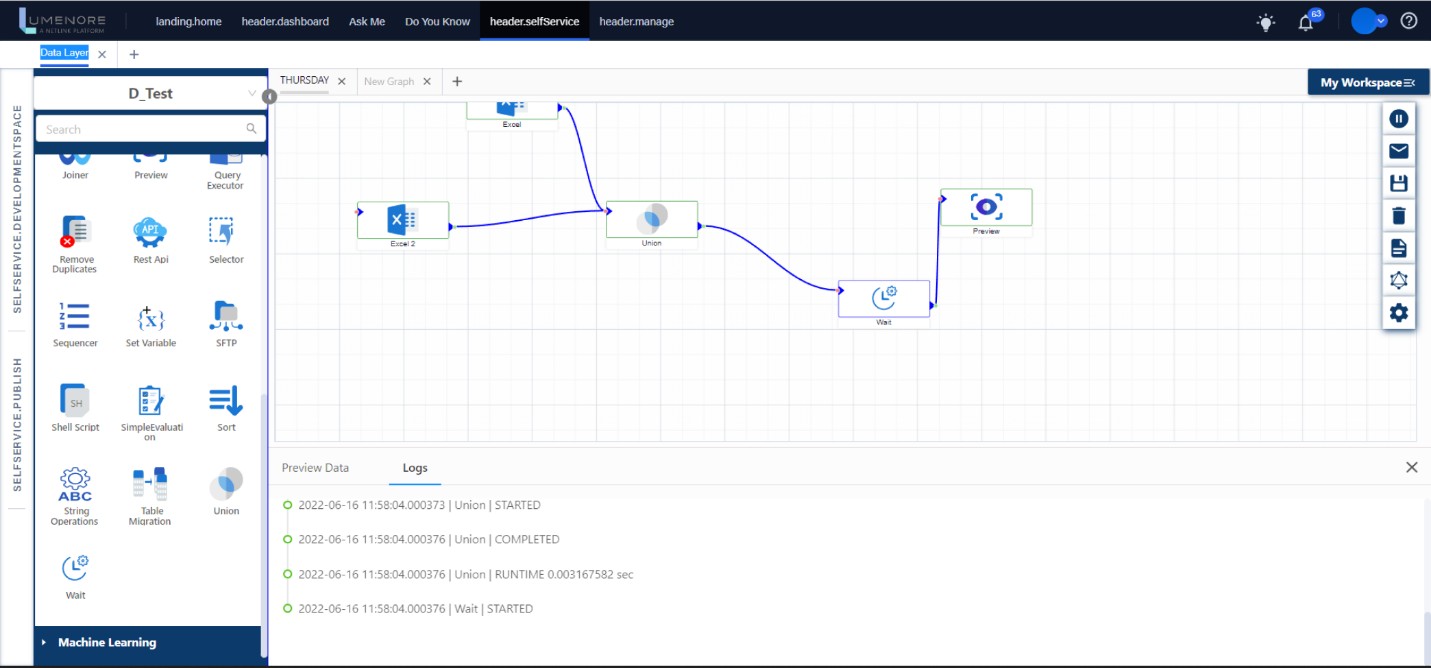


**Wait:** The Wait step allows you to introduce a delay in the execution of the graph. You have the option to specify the time interval and select a time format for this delay.

**How to use Wait?**

* In the "Step Name," provide a name of your choice.
* For "Wait Time," you can specify the desired duration according to your preference.
* Choose the time format that suits your needs.

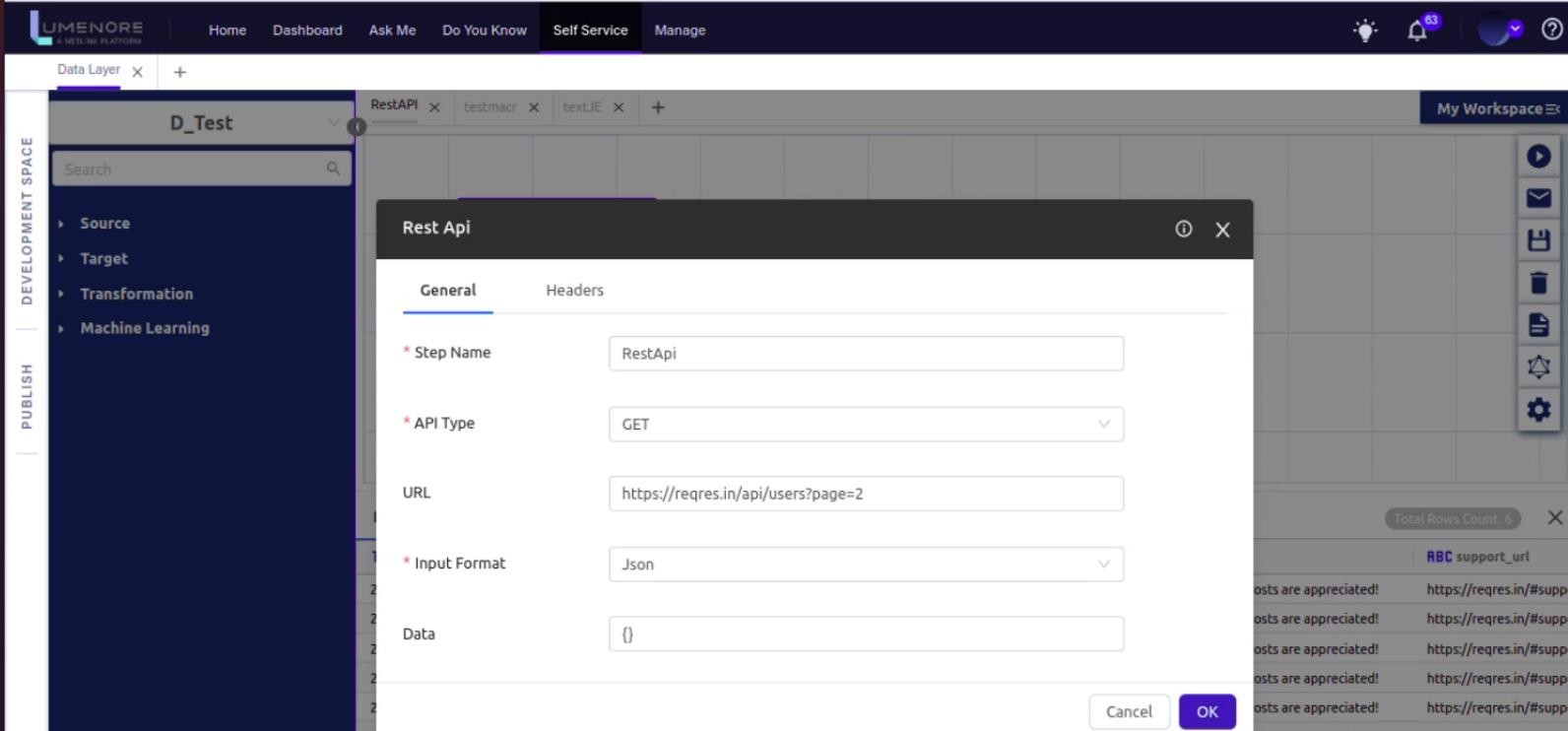


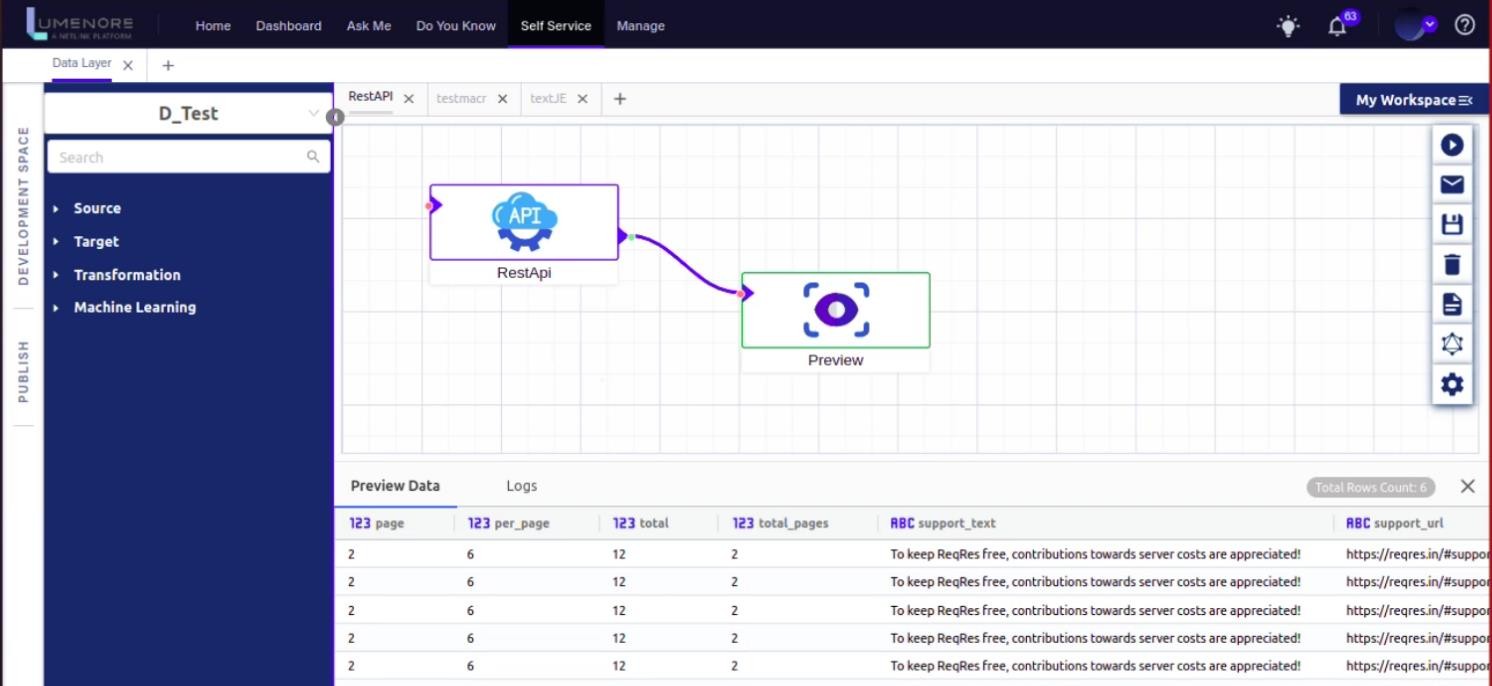


**Rest API:** This step enables you to call a REST API by providing the API URL, API type, headers, and body data. Here's how it works:

* Start by specifying the API URL, its type, and any required parameters in the REST API form.
* Once configured, you can make the API call, and in return, you will receive a response from the called API.







**JavaScript Executer**: The JavaScript Executor is a crucial component in data transformation processes. It enables users to apply custom JavaScript code to manipulate and transform data. This feature is valuable for performing complex calculations, implementing unique business logic, and handling data in ways that standard transformation functions may not support. JavaScript Executor provides the flexibility needed to customize data transformations to meet specific requirements, making it a powerful tool in data processing pipelines.

A screenshot of a computer

Description automatically generated

**Graph:** A graph is a series of connected steps, forming a data pipeline. It's designed to gather data from various sources, apply transformations as per specific needs, offer a preview, and load data into the database when necessary. The snapshot below provides an overview of the entire graph.

A screenshot of a computer

Description automatically generated

**Job:** Jobs are employed when it's necessary to merge multiple graphs for processing data from various sources. What you see here are various graphs combined, and we refer to them as "Jobs."

A screenshot of a computer

Description automatically generated

**Machine Learning**Machine learning is a subfield of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computer systems to improve their performance on a specific task through learning from data, without being explicitly programmed. It is a way for computers to automatically learn and make predictions or decisions based on patterns and insights gleaned from historical data.

Machine learning algorithms can be classified into several categories, including supervised learning, unsupervised learning, and reinforcement learning.

It involves the following key steps:

* **Data Collection:** Gather relevant data from various sources.
* **Data Preprocessing:** Clean, transform, and prepare data for analysis.
* **Feature Engineering:** Create meaningful features for the model.
* **Model Selection:** Choose an appropriate algorithm or model.
* **Model Training:** Teach the model patterns in the data.
* **Evaluation:** Assess the model's performance using metrics.
* **Hyperparameter Tuning:** Optimize model settings for better results.
* **Deployment:** Use the trained model for predictions in real-world applications.

Let us explore the functions offered by Machine learning in the Data Integration Layer:

**Pre-processor:** A pre-processor plays a crucial role in preparing and optimizing data before it is fed into a machine learning model. A pre-processor in machine learning is responsible for preparing and optimizing the dataset to improve the performance and robustness of machine learning models. It ensures that the data is clean, well-structured, and suitable for the chosen algorithm, finally leading to more accurate predictions and better model generalization.

A screenshot of a computer

Description automatically generated

**How to use Preprocessor?**

* In the "Step Name," provide a name.
* Then select the preprocessor type you want to choose.
* Next, provide the Input Column and Output Column.
* Press “OK” to proceed.

**Linear Regression:** Linear regression is a way to identify a relationship between two or more variables. We use this relationship to predict the values for one variable for a given set of value(s) of the other variable(s). The variable, which is used in prediction is termed an independent/explanatory/regressor variable whereas the predicted variable is termed a dependent/target/response/regress variable.

A screenshot of a computer

Description automatically generated

**How to use Linear Regression?**

* In the "Step Name," provide a name.
* Then select the feature you want to choose from "Feature " option.
* Next, specify the label.
* Select “Save model” if you want to save your choices.
* Press “OK” to proceed.

**Predictor**: In machine learning, a predictor is a crucial component of a predictive model. Its primary role is to take input data and make predictions or estimates about specific outcomes or target variables. Here is a brief explanation of the role of a predictor in machine learning:

* **Input Data:** Predictors take input data, which includes features or attributes that describe the characteristics of the problem or domain you are trying to model.
* **Learning Patterns:** During the training phase, predictors analyze historical data to learn patterns, relationships, and dependencies within the input features and the target variable.
* **Prediction:** Once the model has learned from the training data, it can make predictions on new, unseen data. The predictor applies the learned patterns to the input features of new data to estimate or predict the target variable.
* **Output:** The predictor's output is the predicted value or class label associated with the input data. For regression tasks, the output is typically a numerical value, while for classification tasks, it's a class label.
* **Evaluation:** Predictors' performance is assessed using various evaluation metrics, such as mean squared error (MSE) for regression or accuracy, precision, recall, and F1-score for classification. These metrics measure how well the predictor's predictions align with the actual outcomes.
* **Model Improvement:** Based on evaluation results, predictors can be fine-tuned, and hyperparameters can be adjusted to enhance prediction accuracy or generalization to new data.

In summary, a predictor is a core component of a machine learning model responsible for making predictions based on patterns learned from historical data.

A screenshot of a computer

Description automatically generated

**How to use Predictor?**

* In the "Step Name," provide a name.
* Then select the feature you want to choose from "Feature " option.
* Next, specify the label.
* Give a name to your Model.
* Press “OK” to proceed.

Top of Form