PELICAN USER GUIDE

Version no : 1.1.7.0

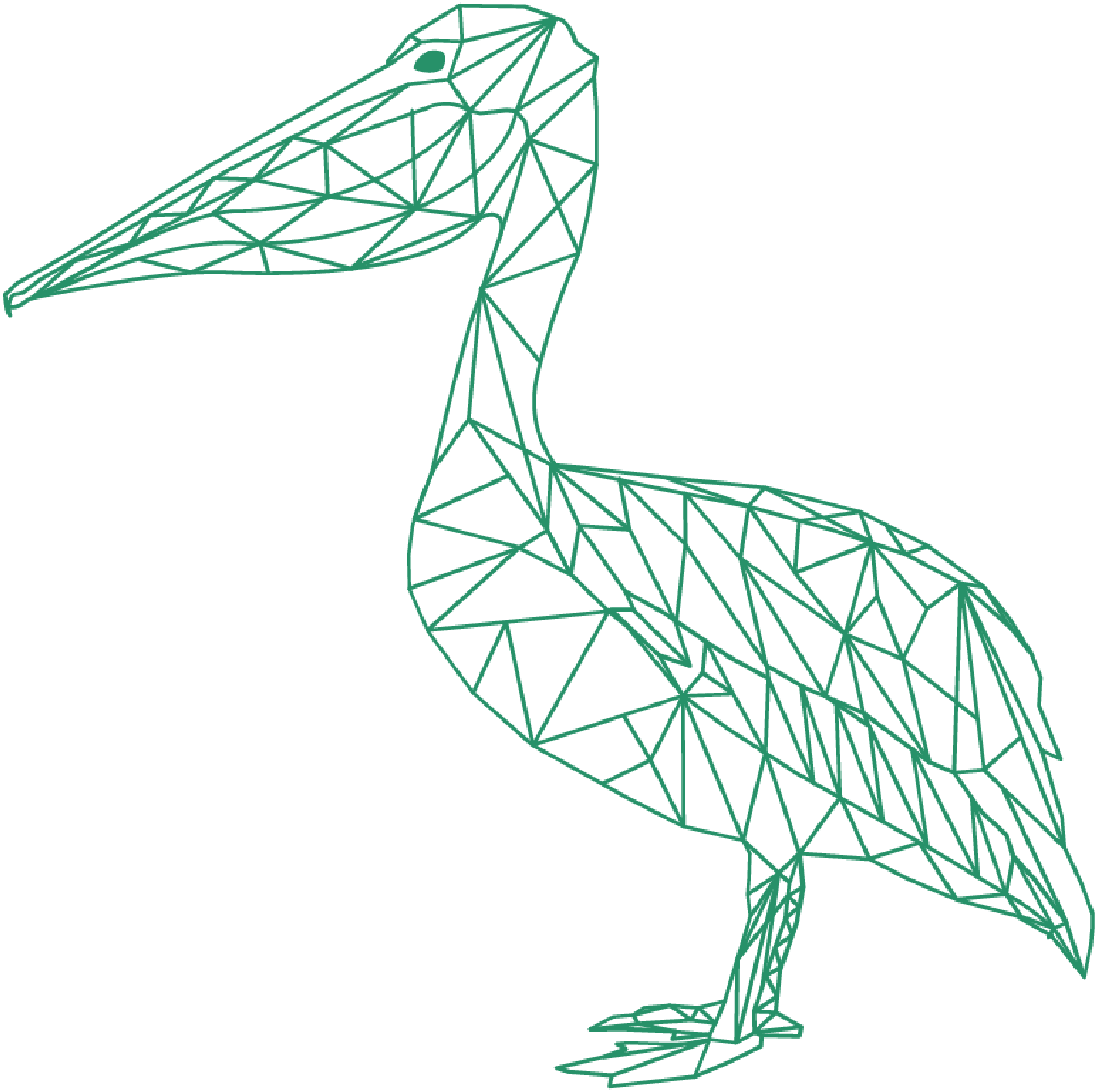


Table of Contents

1. Preface

The Pelican user guide is written for developers and software engineers who are responsible for comparing and validating the data across various data stores. The Pelican user guide assumes that you have an understanding of your operating systems, relational database concepts, database engines, flat files, and how the mainframe system works in your environment. This guide also assumes that you are familiar with the interface requirements for your supporting applications.

2. Pelican Overview

Pelican is an innovative enterprise application that ensures businesses that the data is migrated from reference to destination data store accurately and reliably by performing validation over the migrated data. Pelican helps the user to compare and validate the data across different data stores quickly.

Pelican helps the user to validate large data sets without any data transfer from source to target. Pelican allows the user to copy a large dataset (historical and real time data) to a destination data store for validation. It uses an innovative approach to compare the data without moving it across data stores. It also displays the data records having differences.

The user can automate the comparison using schedulers and API. Pelican uses the phonetic matching and approximate matching algorithms to search the best possible combination of tables from the target datastore. It also facilitates configuring and executing various Schedulers as per the business requirement. These Schedulers are created to compare and validate table dataset periodically.

You can configure an email notification for schedulers that sends an email automatically to various users when the scheduler executes. Pelican provides lineage support, which enables the user to track the movement of data across various nodes. This information is retrieved from the job history server of various data stores configured in the Pelican. Once the validation is done, the application generates statistics reports

3. Data Store Pair Support

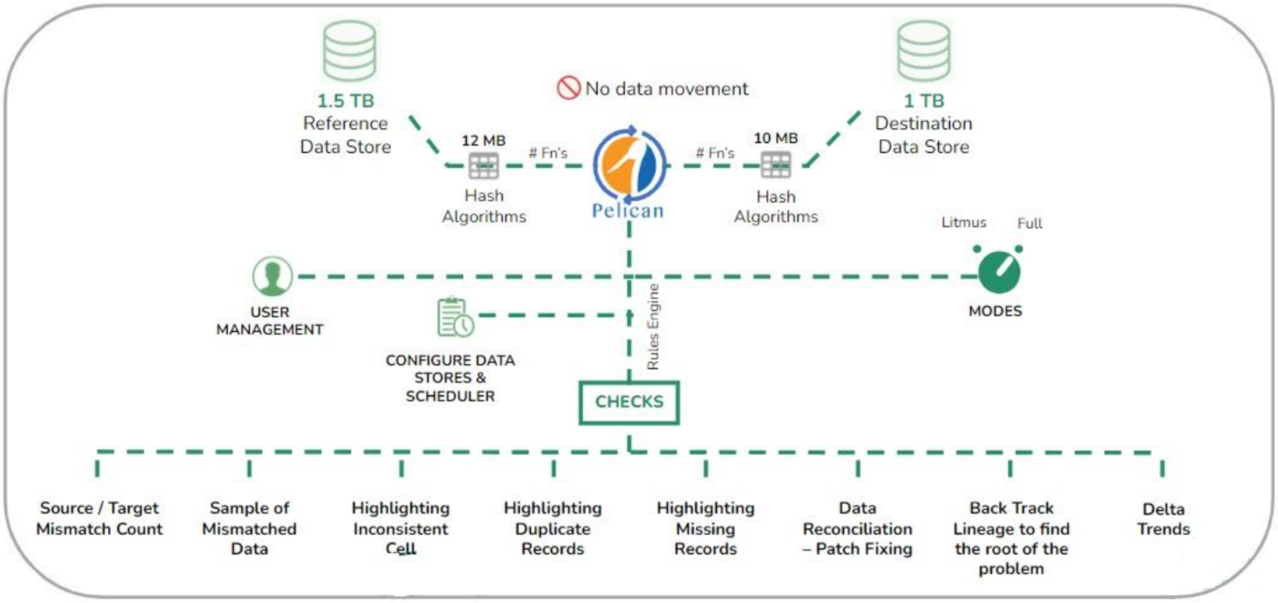
Pelican supports comparison between the following data stores using an innovative approach:

Sr. No.	Source Data store	Destination Data store
1	**Teradata**	**Big Query**
2	**Teradata**	**Snowflake**
3	**Teradata**	**Synapse**
4	**Teradata**	**Hive**
5	**Teradata**	**Delta Lake**
6	**Netezza**	**Big Query**
7	**Netezza**	**Hive**
8	**Netezza**	**Snowflake**
9	**Netezza**	**Synapse**
10	**Hive**	**Big Query**
11	**Hive**	**Hive**
12	**Hive**	**Delta Lake**
13	**Hive**	**Synapse**
14	**Oracle**	**Big Query**
15	**Oracle**	**Hive**
16	**Oracle**	**PostgreSQL**
17	**Oracle**	**PostgreSQL (Google Cloud)**
18	**Big Query**	**Big Query**
19	**Big Query**	**Hive**

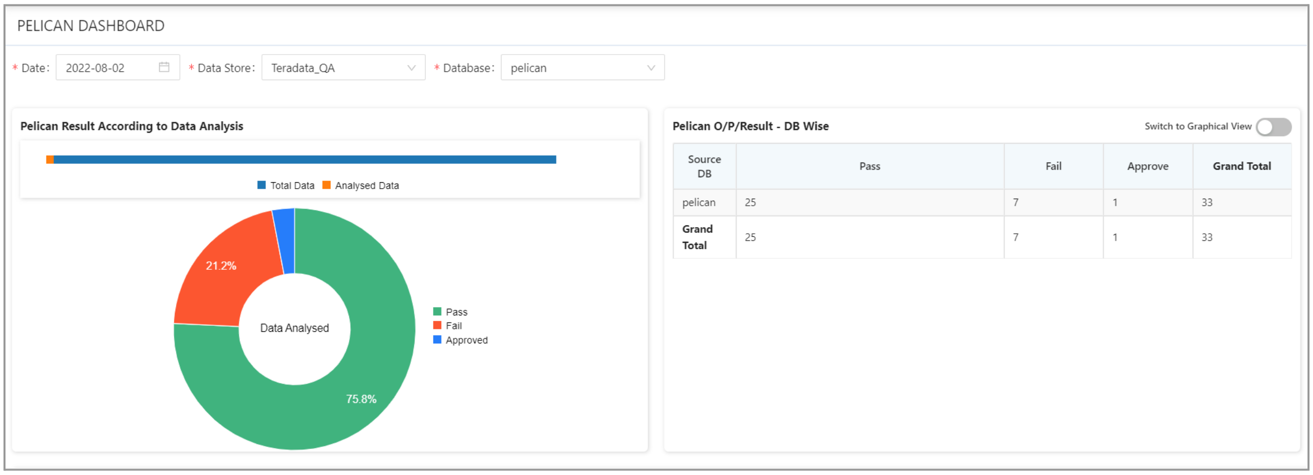
20	**MS SQL Server**	**Hive**
21	**MS SQL Server**	**Big Query**
22	**MS SQL Server**	**MS SQL Server**
23	**MS SQL Server ((Hosted on CloudSQL))**	**Snowflake (AWS hosted)**
24	**MS SQL Server (On-Prem)**	**Snowflake (AWS hosted)**
25	**Impala**	**Big Query**
26	**SAP HANA**	**Big Query**
27	**DB2**	**Big Query**
28	**Greenplum**	**Redshift**
29	**Snowflake**	**Big Query**
30	**Redshift**	**Big Query**
31	**Vertica**	**Big Query**

4. Architecture

Here is the complete architecture of how Pelican works.



5. Dashboard



PELICAN DASHBOARD

* Date: 2022-08-02 * Data Store: Teradata_QA * Database: pelican

Pelican Result According to Data Analysis

Pelican O/P/Result - DB Wise

Source DB	Pass	Fail	Approve	Grand Total
pelican	25	7	1	33
Grand Total	25	7	1	33

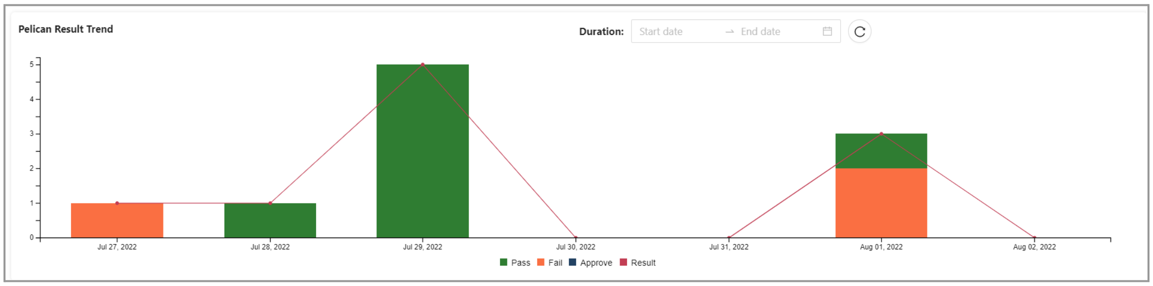
Dashboard screen displays result, trends, graph of the scheduled mappings based on datastore, database, and date. When the dashboard screen is loaded for the first time it will display the dashboard screen for the default datastore. Users can select datastore and databases to see the corresponding graphs.

The Pelican dashboard consists of the following components:

1. **Pelican Pass/Fail/Approve Percentage**

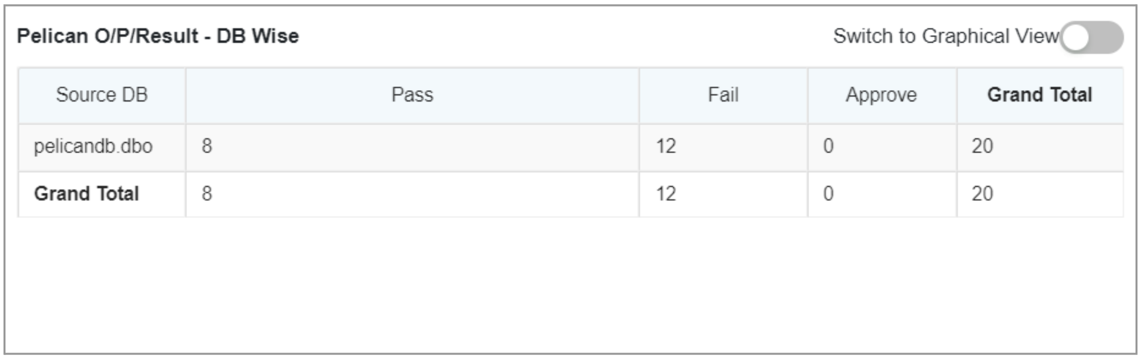
   It represents the total pass/fail/approved percentage of the total scheduled mappings for the particular datastore and databases.

2. **Pelican Result trend**



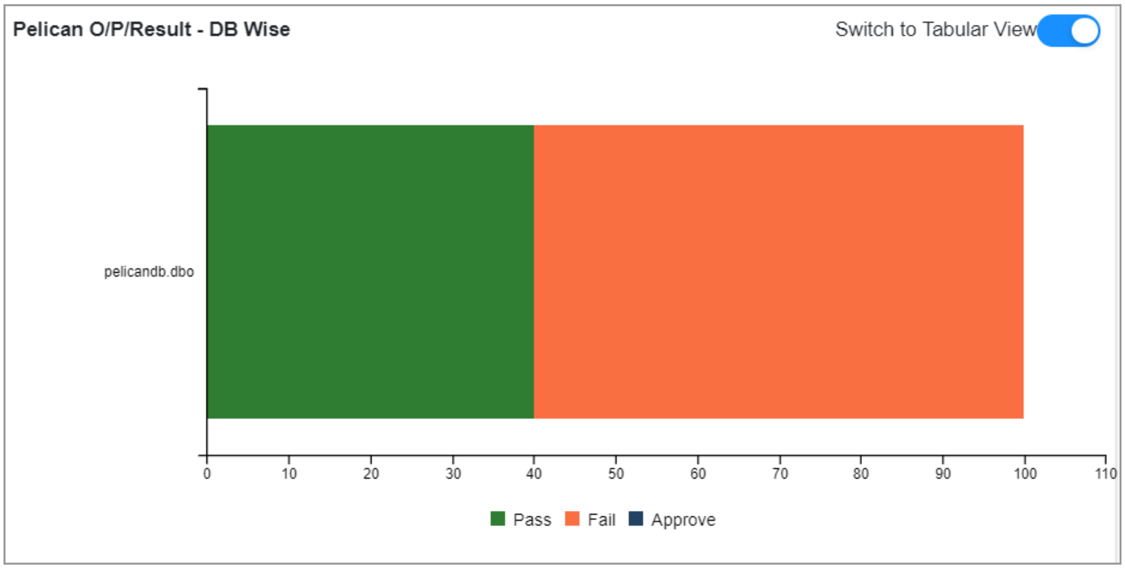
   It represents the total pass/ fail/ approve counts date wise for the selected duration, databases, and tags for particular datastore. By default, duration is seven days from the first time the mapping containing the datastore has been executed. Green represents total pass count; Red represents total fail count and Blue represents total approve count.

3. Pelican Output Result DB wise



Source DB	Pass	Fail	Approve	**Grand Total**
pelicandb.dbo	8	12	0	20
Grand Total	8	12	0	20

It displays the total pass/ fail/ approve count of the scheduled mappings for the selected database.



This info can be seen in two formats shown below, users can switch between graphical and tabular format using the toggle option present on the upper right corner.In graphical format Green represents total pass count, Red represents total fail count and Blue represents total approve count.

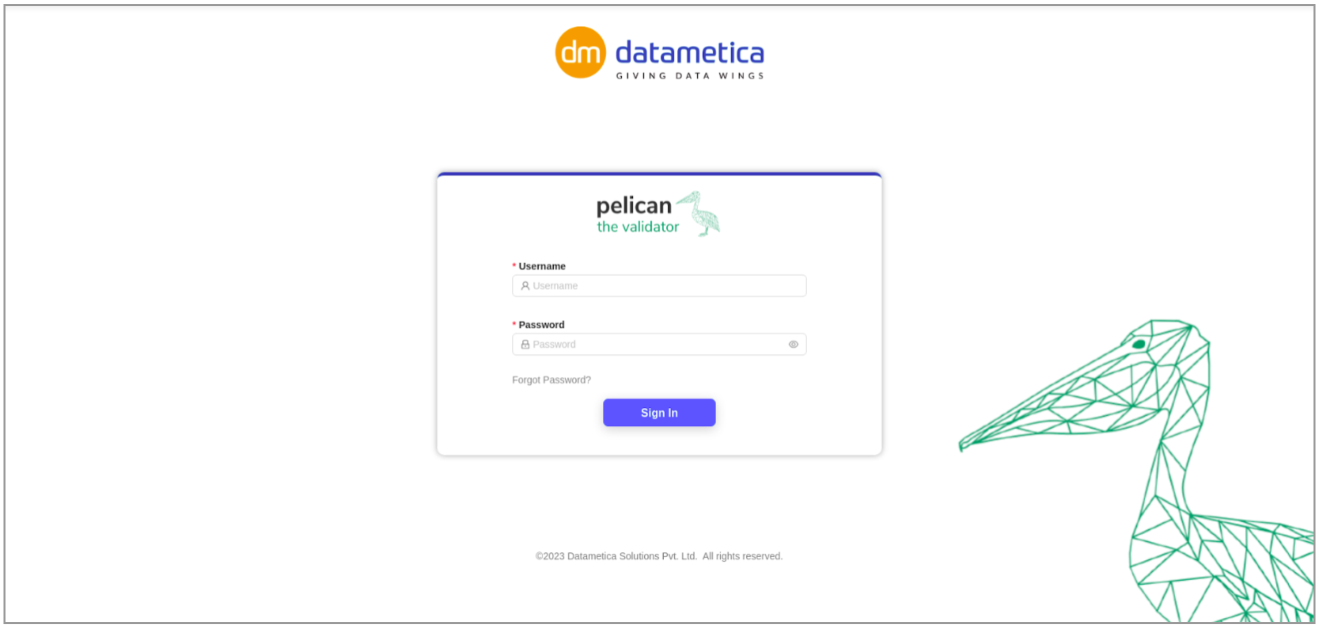
6. User Management

The User Management section enables secured authorization to the users. Each user is assigned a unique identity to authenticate the application.

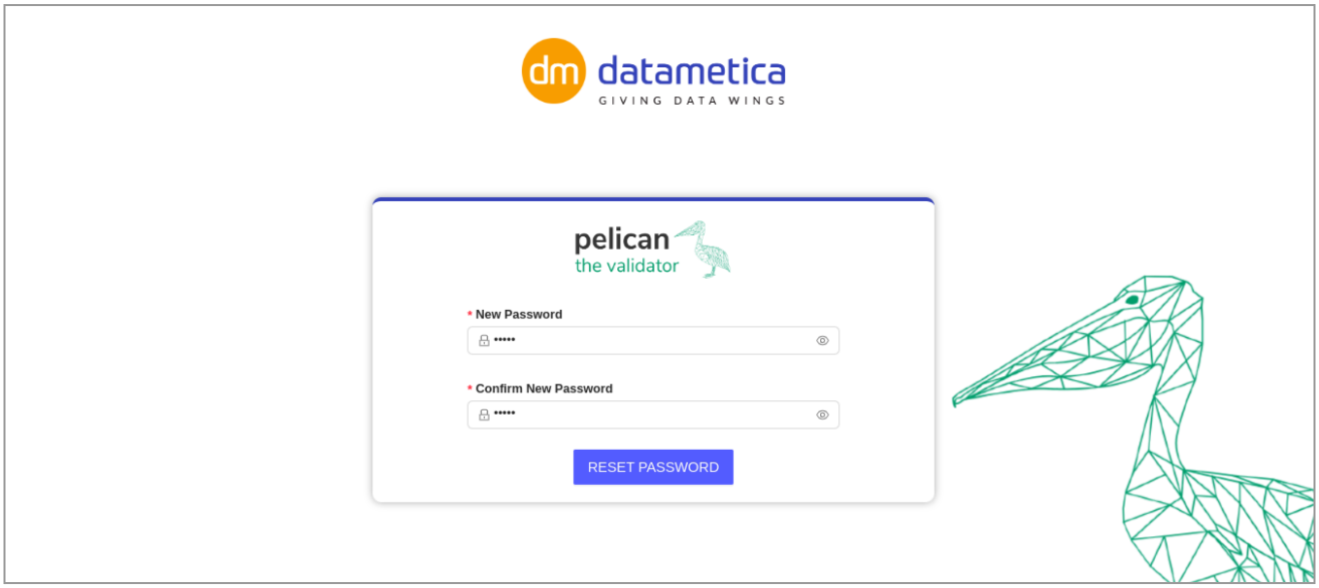
6.1. Login

Pelican needs the following details for logging in:

- User's Email Id
- Password



First time users will be requested to reset the default password.



After completing the steps the user will navigate to the **Validation Result** page.

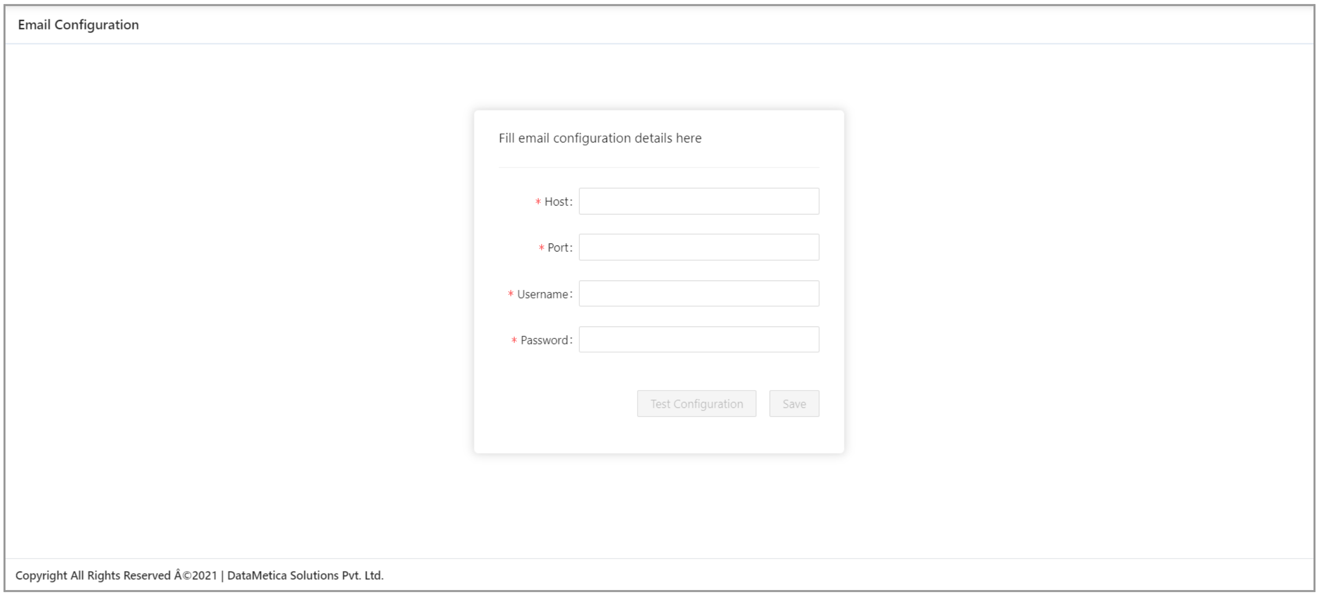
6.2. Configure SMTP server

The SMTP server configuration allows the Administrator to configure the SMTP server which can be used for sending passwords to the newly added user.

Configure SMTP server and Sender Email Address:

To configure the SMTP Server and senders email address:

1. Go to the **Administration** → **Configure** → **Email Configuration**.
2. Fill the below details in the **Email** Configuration window.
   - Enter **Host Name** (Ex. smtp.mail.yahoo.com)
   - Enter **Port Name** (Ex. 587)
   - Enter **User Name**
   - Enter **Password** in the provided field.



Email Configuration

Fill email configuration details here

* Host:

* Port:

* Username:

* Password:

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

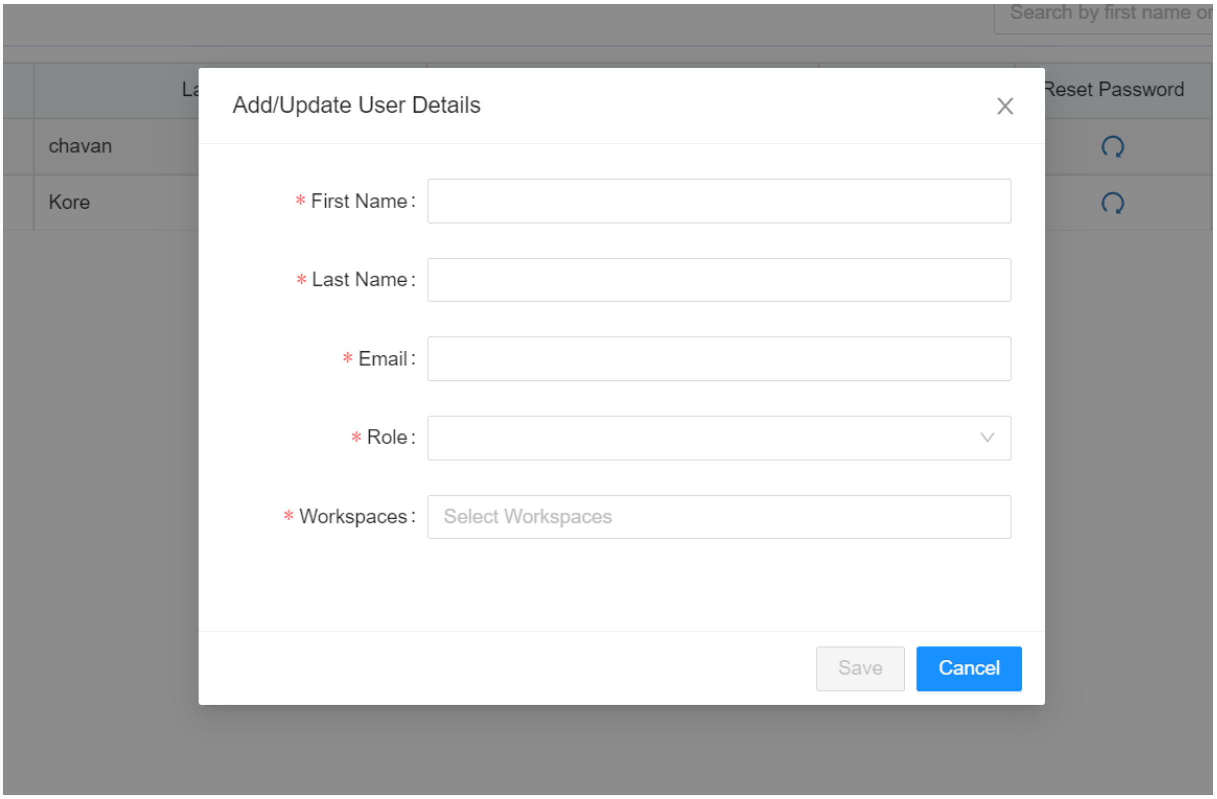
3. Click on **Test Connection**.
4. Click on the **Save** button to save the configuration.

6.3. SMTP User creation

The **User** tab enables creating a new user with the role of Validator or Admin. You can edit or delete an existing user as well. This option will be available if LDAP configuration is not set up. (SMTP)

The steps to add a new SMTP user are as follows:

1. Go to **Administration** → **User Management** → **User.**
2. Click **Add New User.**
3. In **Add / Update User Details** Form, enter First Name, Last Name, Email, and select Role from the drop-down list.



Add/Update User Details

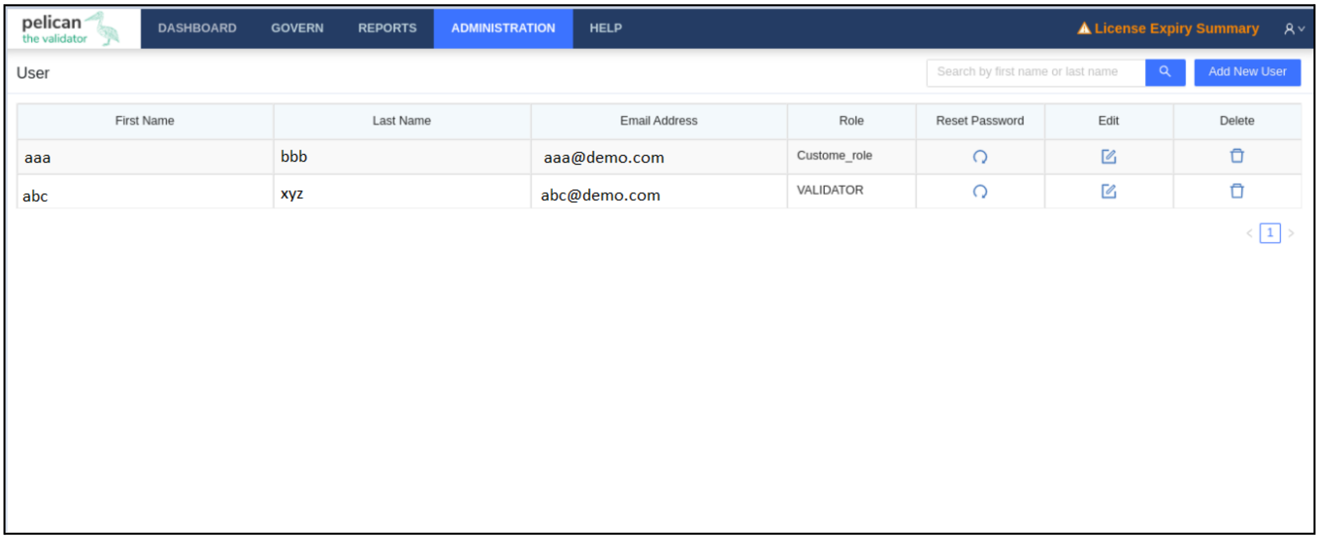
* First Name :

* Last Name :

* Email :

* Role :

* Workspaces : Select Workspaces



User

First Name	Last Name	Email Address	Role	Reset Password	Edit	Delete
aaa	bbb	aaa@demo.com	Custome_role	○	✎	🗑
abc	xyz	abc@demo.com	VALIDATOR	○	✎	🗑

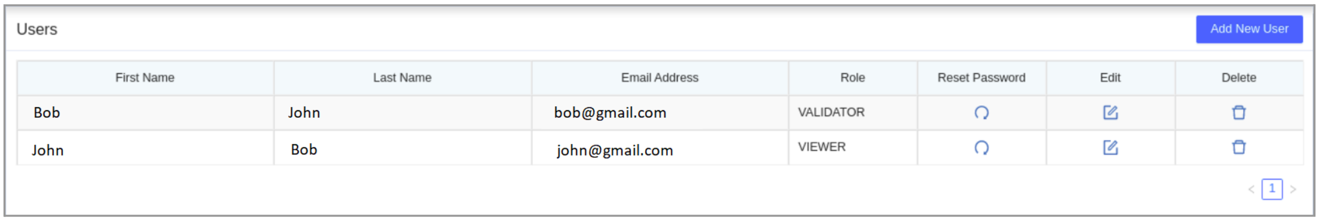
Note:

The roles are Admin, Validator and custom roles. The Admin user can assign all roles.

➢ **Admin:** Users having the role of admin have the same rights as Superuser. They can create new users having roles as Admin or Validator. They can create data stores for validation and configure emails for sending mails. By default, the ‘Pelican’ workspace will be assigned to a user having an Administrator role.

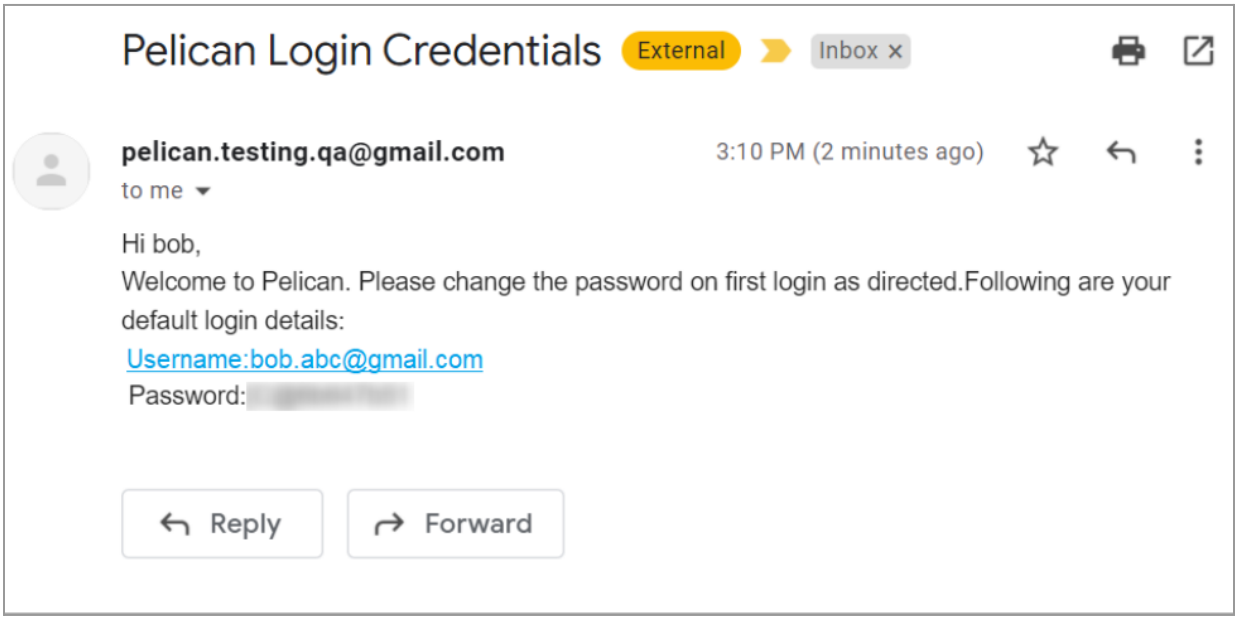
- ➢ **Validator:** The users having a validator role do not have access to create a new datastore and can't create a new user, but they can create mappings, run schedulers, and view reports.
- ➢ **Custom Roles**: Roles that are created by the Administrator as per the requirement.

4. Click **SAVE**.



First Name	Last Name	Email Address	Role	Reset Password	Edit	Delete
Bob	John	bob@gmail.com	VALIDATOR	○	✎	🗑
John	Bob	john@gmail.com	VIEWER	○	✎	🗑

A system generated Pelical Login Credentials email is generated and sent to the first-time login user which includes username and password. The user should login with those credentials and reset the password.



pelican.testing.qa@gmail.com 3:10 PM (2 minutes ago)
to me

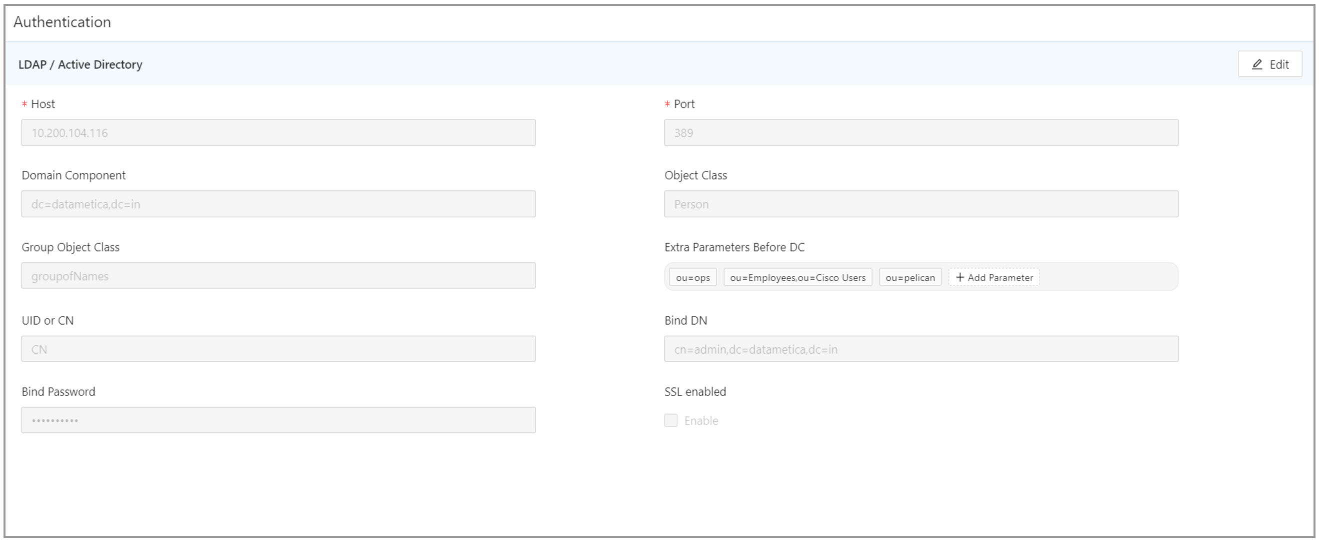
Hi bob,
Welcome to Pelican. Please change the password on first login as directed.Following are your default login details:
Username:bob.abc@gmail.com
Password:[redacted]

6.4. Authentication using LDAP/AD

You can add the Authentication details for the LDAP/ Active Directory.

The steps to add authentication details are as follows:

1. Go to **Administration** -> **User Management** -> **Authentication**.
2. Click **Edit**.



Authentication

LDAP / Active Directory

* Host: 10.200.104.116

* Port: 389

Domain Component: dc=datametica,dc=in

Object Class: Person

Group Object Class: groupofNames

Extra Parameters Before DC: ou=ops | ou=Employees,ou=Cisco Users | ou=pelican

UID or CN: CN

Bind DN: cn=admin,dc=datametica,dc=in

Bind Password: ••••••••••

SSL enabled: □ Enable

3. Enter Host, Port, Domain Component, Object Class, Extra Parameters Before DC, and other fields.
4. Check the checkbox if SSL is enabled.
5. Click **DONE**.

Field Name	Description
Host	Ip or machine name on which Ldap server is running
Port	Port at which connection to the ldap server is to be created
Following fields are used to create the BaseDN, in which user should be searched	
Domain Component	The comma separated values that are put with the attribute “dc”
Object Class	Object class decides the attributes of the entry and is part of the search criteria
Extra Parameters Before DC	Any extra parameters that are present between common name & domain components. We can input multiple tags in this field.
UID or CN	Value of this field will be either uid or cn. This attribute denotes the common name of the user.
Bind DN	This field will hold the qualified value of the user which may be required to connect to ldap
Bind Password	This field will hold the password corresponding to the user mentioned in above field
SSL Enabled	If ldap is secured, check the checkbox. Certificate should also be configured in the pelican jvm

Note:

- Post LDAP/AD successful connection, we will need to restart Pelican tomcat services so that added users can log in to the application.
- After configuring the **Authentication** page the user has to navigate to the tomcat/bin folder where Pelican is installed. Stop the Pelican application using ./shutdown.sh

  Then navigate to /webapps/ROOT/WEB-INF/classes/config.properties file.

  And add the following properties and corresponding values in it:

  1) authenticationByDistinguishedName = false
  2) enterpriseIdAttribute = sAMAccountName
- Once you have added the credentials, super user can only create ldap user and can't create non-ldap user having role as Admin or Validator.
- The user can start the application using ./startup.sh

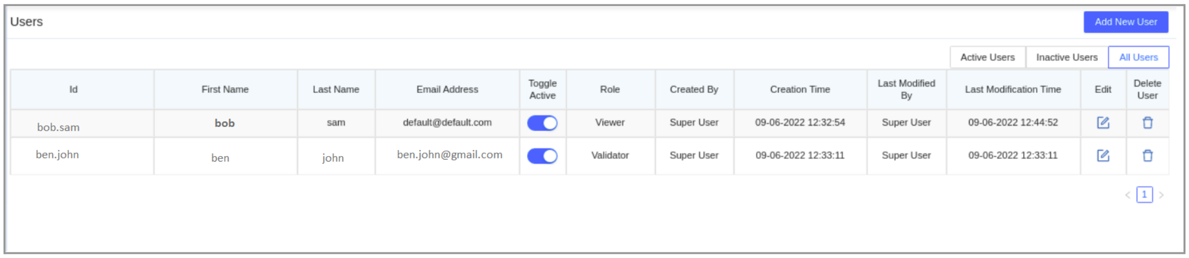
6.5. LDAP/AD User Management

LDAP/AD user authentication is the process of validating a username and password combination with a directory server. Only the superuser and admin has the right to create LDAP/AD users. In the **User Management** module you can perform the following functions:

- Setting LDAP/AD credential
- Adding user
- Accessing control
- Assigning roles to the user
- Administrating Pelican
- Adding roles
- Granting permissions
- Assigning user groups

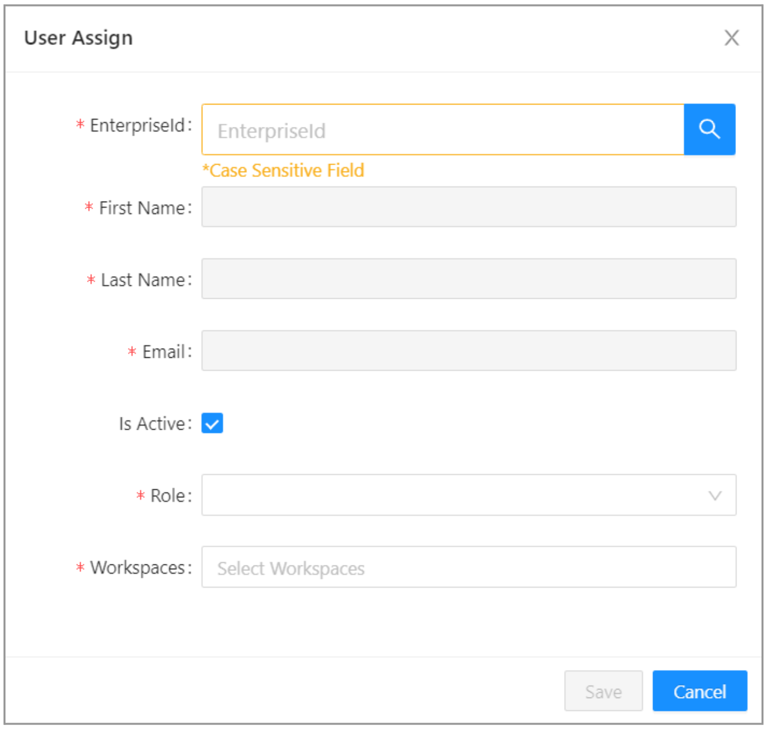
In **User Management** you can add a new LDAP user and view the details on the **User** page.

1. Go to **Administration** -> **User Management** -> **User**.
2. Click the **Add New User**.



Id	First Name	Last Name	Email Address	Toggle Active	Role	Created By	Creation Time	Last Modified By	Last Modification Time	Edit	Delete User
bob.sam	bob	sam	default@default.com	●	Viewer	Super User	09-06-2022 12:32:54	Super User	09-06-2022 12:44:52	✎	🗑
ben.john	ben	john	ben.john@gmail.com	●	Validator	Super User	09-06-2022 12:33:11	Super User	09-06-2022 12:33:11	✎	🗑

3. On the **User Assign** form enter EnterpriseID and all other details are visible automatically in the fields.



User Assign ✕

* EnterpriseId: EnterpriseId 🔍
*Case Sensitive Field

* First Name:

* Last Name:

* Email:

Is Active: ☑

* Role:

* Workspaces: Select Workspaces

The roles are Admin and Validator. You can also assign custom roles according to the requirements. The Admin user can assign all the roles.

Admin: Users having the role of admin have the same rights as Superuser. They can create new users having roles as Admin or Validator. They can create data stores for validation and configure emails for sending mails. By default, the 'Pelican' workspace will be assigned to a user having an Administrator role.

Validator: The users having a validator role do not have access to create a new datastore and can't create a new user, but they can create mappings, run schedulers, and view reports.

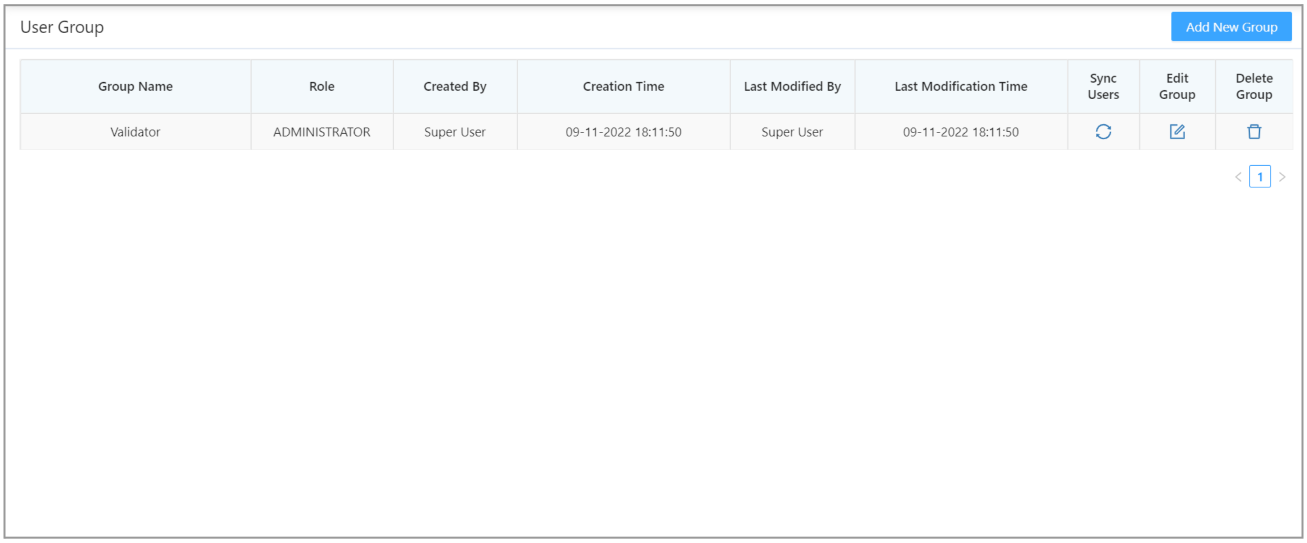
Custom Roles: Roles created according to the requirements by the Administrator.

4. Select the **Is Active** checkbox if required.
5. Click **SUBMIT**.

Note:

- You can apply filters to view all the active and inactive users by clicking Active or Inactive user.
- The LDAP users can be edited and deleted.

6.6. User Group



User Group

Group Name	Role	Created By	Creation Time	Last Modified By	Last Modification Time	Sync Users	Edit Group	Delete Group
Validator	ADMINISTRATOR	Super User	09-11-2022 18:11:50	Super User	09-11-2022 18:11:50	⟳	✎	🗑

The **User Group** feature enables the administrator to give access rights (roles and workspaces) to all the users belonging to the same AD or LDAP group in one instance. This also enables auto-syncing of users from AD/LDAP groups with the Pelican User Groups at the configurable interval.

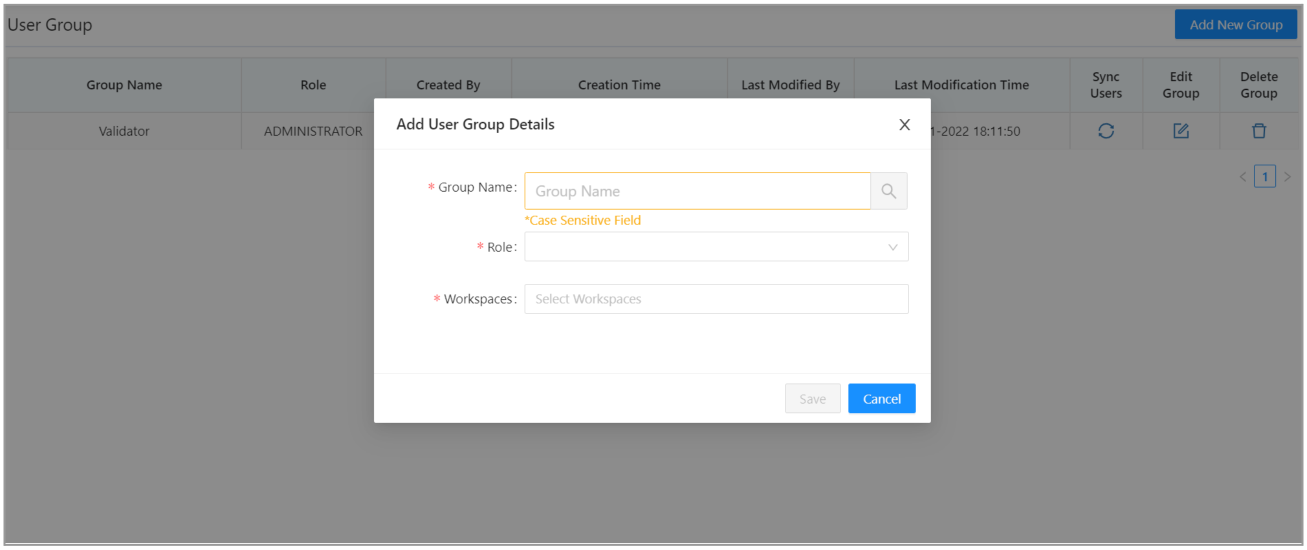
The steps to add a new **User Group** are as follows:

1. Select User Group by the below path.

   Administration -> **User Management** -> **User Group**

2. Click on **Add New Group**.

   The **Add User Group Details** page opens.



Add User Group Details

* Group Name: Group Name
*Case Sensitive Field

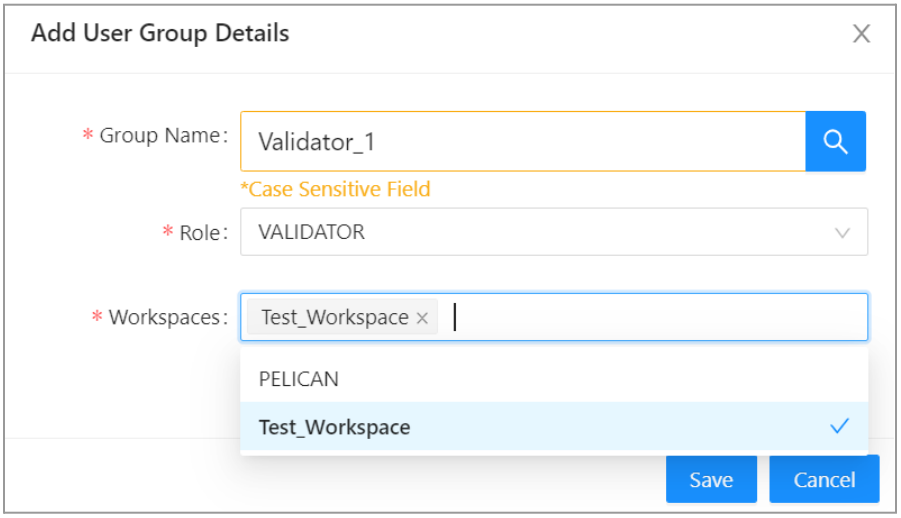
* Role:

* Workspaces: Select Workspaces

3. Enter a group name and click on the Search icon.

   If the Group exists, it will get selected. If the group does not exist, an error message will be displayed.

4. Select **Role** and **Workspace**.



Add User Group Details

* Group Name: Validator_1

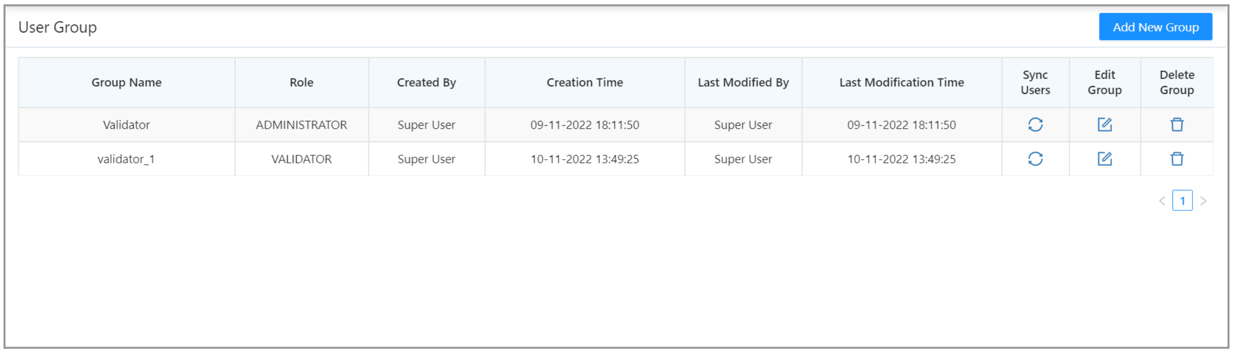
*Case Sensitive Field

* Role: VALIDATOR

* Workspaces: Test_Workspace ×

5. Click **Save**.

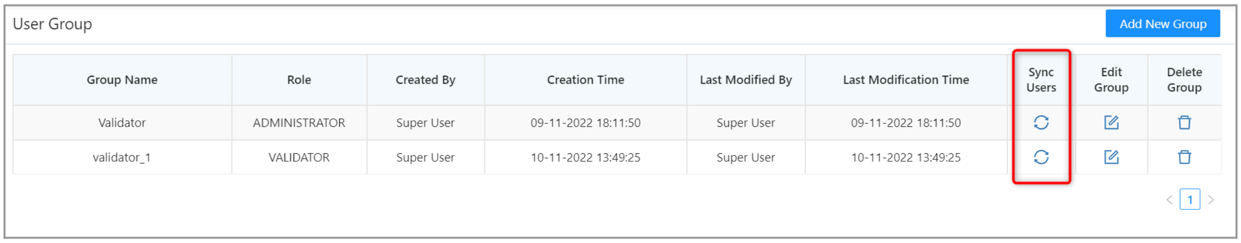
The newly added User Group will be added to the list.



User Group

Group Name	Role	Created By	Creation Time	Last Modified By	Last Modification Time	Sync Users	Edit Group	Delete Group
Validator	ADMINISTRATOR	Super User	09-11-2022 18:11:50	Super User	09-11-2022 18:11:50	↻	✎	🗑
validator_1	VALIDATOR	Super User	10-11-2022 13:49:25	Super User	10-11-2022 13:49:25	↻	✎	🗑

6.6.1. Sync Users



User Group

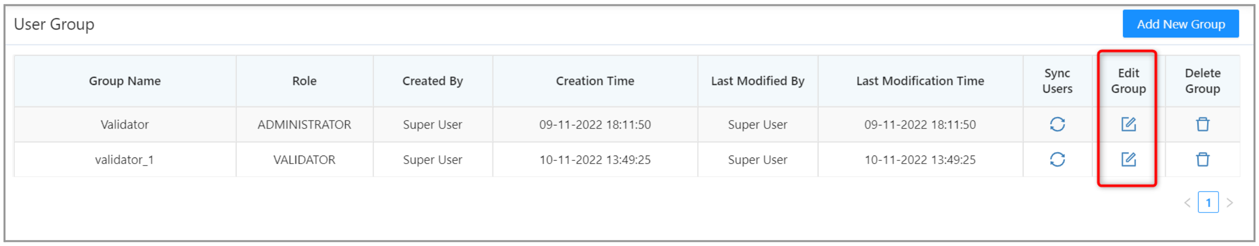
Group Name	Role	Created By	Creation Time	Last Modified By	Last Modification Time	Sync Users	Edit Group	Delete Group
Validator	ADMINISTRATOR	Super User	09-11-2022 18:11:50	Super User	09-11-2022 18:11:50	↻	✎	🗑
validator_1	VALIDATOR	Super User	10-11-2022 13:49:25	Super User	10-11-2022 13:49:25	↻	✎	🗑

This functionality enables the user to check and maintain similar user records in the User group and the client's AD database.

- The records missing from the AD database will be removed from the user group and similarly new records will be added.

- The Administrator can sync the records any time or sync activity can also be scheduled to run after a specific time period (Daily at 3 AM server time).

6.6.2. Edit Group

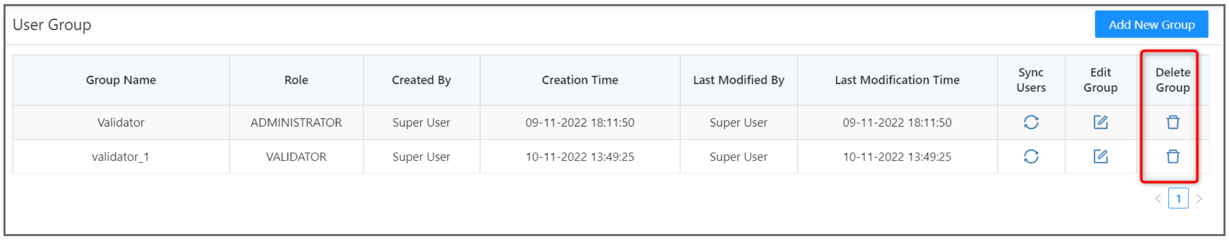


User Group

Group Name	Role	Created By	Creation Time	Last Modified By	Last Modification Time	Sync Users	Edit Group	Delete Group
Validator	ADMINISTRATOR	Super User	09-11-2022 18:11:50	Super User	09-11-2022 18:11:50	⟳	✎	🗑
validator_1	VALIDATOR	Super User	10-11-2022 13:49:25	Super User	10-11-2022 13:49:25	⟳	✎	🗑

A user group can be edited by clicking the edit icon of the corresponding group record.

6.6.3. Delete Group



User Group

Group Name	Role	Created By	Creation Time	Last Modified By	Last Modification Time	Sync Users	Edit Group	Delete Group
Validator	ADMINISTRATOR	Super User	09-11-2022 18:11:50	Super User	09-11-2022 18:11:50	⟳	✎	🗑
validator_1	VALIDATOR	Super User	10-11-2022 13:49:25	Super User	10-11-2022 13:49:25	⟳	✎	🗑

A user group can be deleted by clicking the delete icon of the corresponding group record.

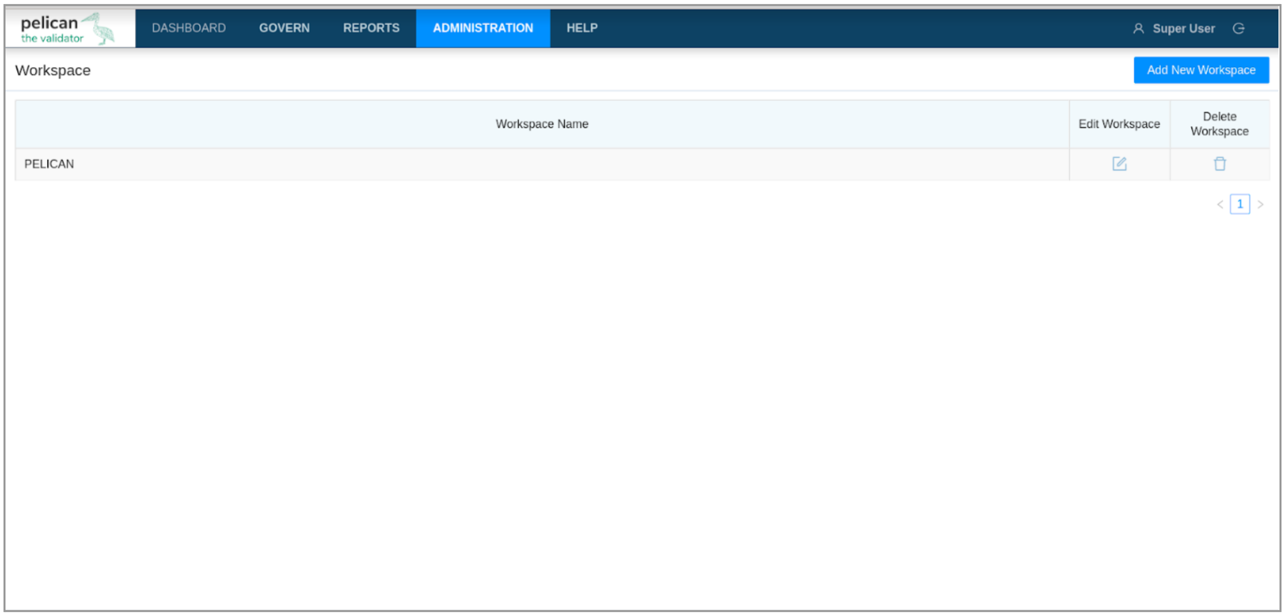
Note: If a User group is deleted, the users belonging to that group also get deleted.

6.7. Workspace

A Workspace can be described as a specific collection of mappings that are associated with a user. A user can have multiple workspaces according to their requirements. A mapping can be a part of multiple workspaces. If no workspaces are assigned to a user, then the default workspace will be assigned to it.

The steps to Add a Workspace in Pelican are as follows:

1. Go to **Administration** -> **User Management** -> **Workspace**.
   The **Workspace** page opens.

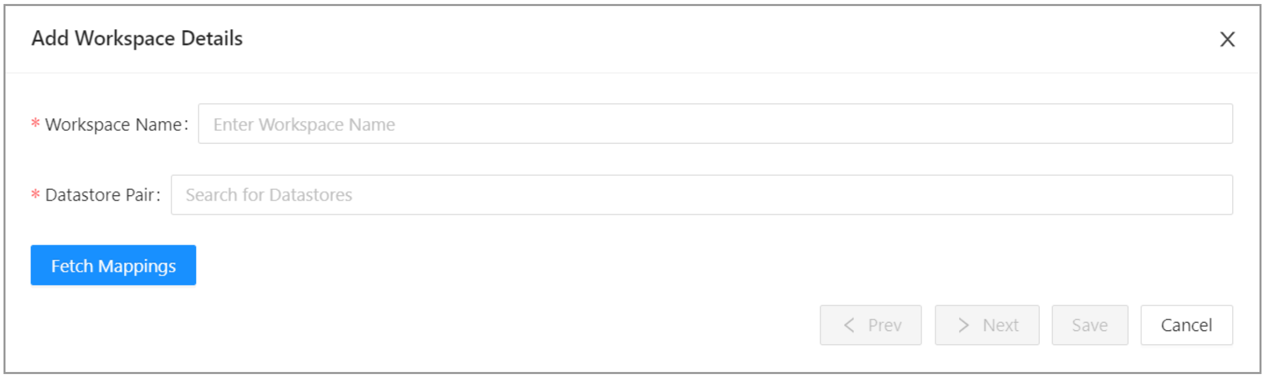


Workspace

Workspace Name	Edit Workspace	Delete Workspace
PELICAN		

2. Click **Add New Workspace** button.

   The **Add Workspace Details** page opens.



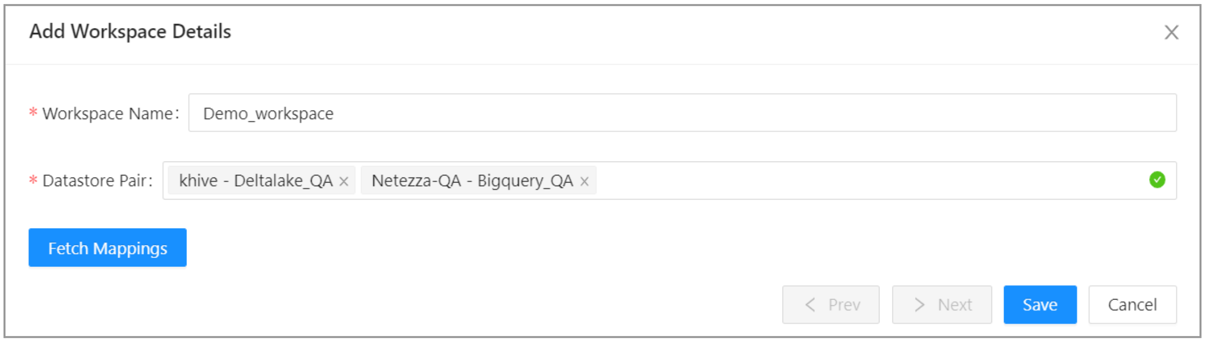
Add Workspace Details

* Workspace Name: Enter Workspace Name

* Datastore Pair: Search for Datastores

3. Enter an appropriate **Workspace** name.
4. Enter the required **Datastore Pair** names.

   Note: The Datastores should be selected while creating workspace to limit the access. Only those datastores which have been assigned to the workspace will be applicable for all the other features in Pelican. In other words, we can restrict the Workspace at the datastore level.


Add Workspace Details

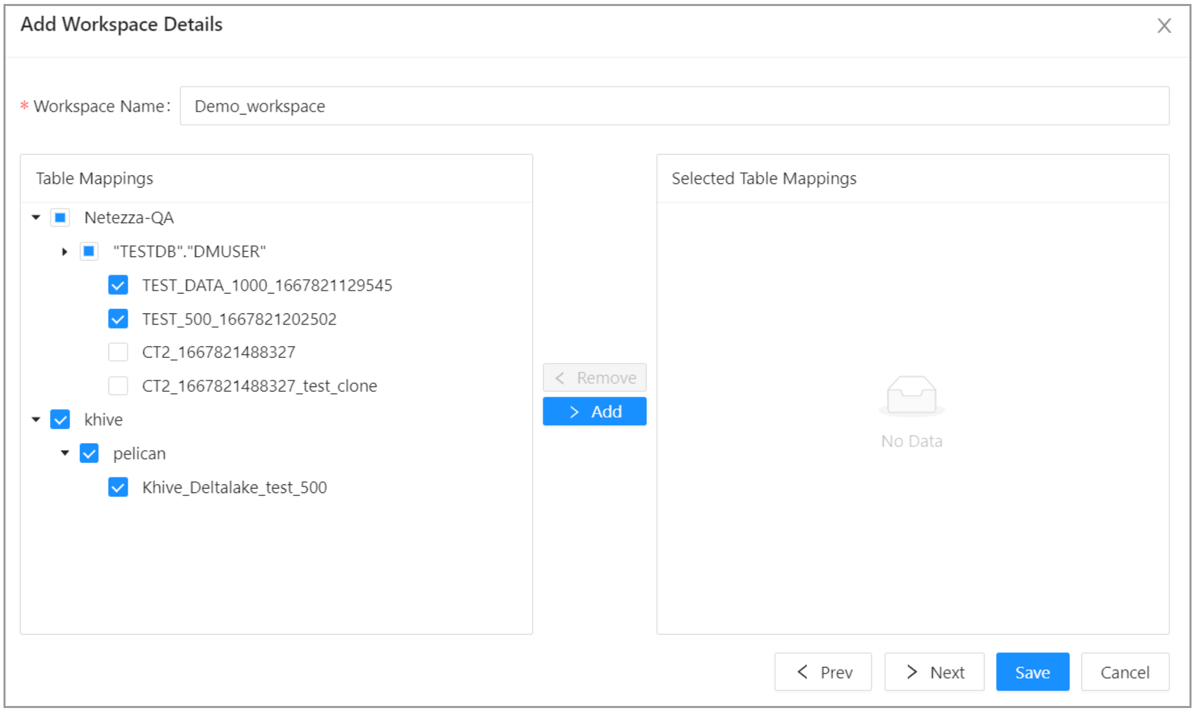
* Workspace Name: Demo_workspace

* Datastore Pair: khive - Deltalake_QA × Netezza-QA - Bigquery_QA ×

5. Click on **Fetch Mapping**.

   The mappings associated with the selected datastore pairs only will be fetched and displayed.

6. Click **Next**.
7. Select the appropriate mappings.


Add Workspace Details

* Workspace Name: Demo_workspace

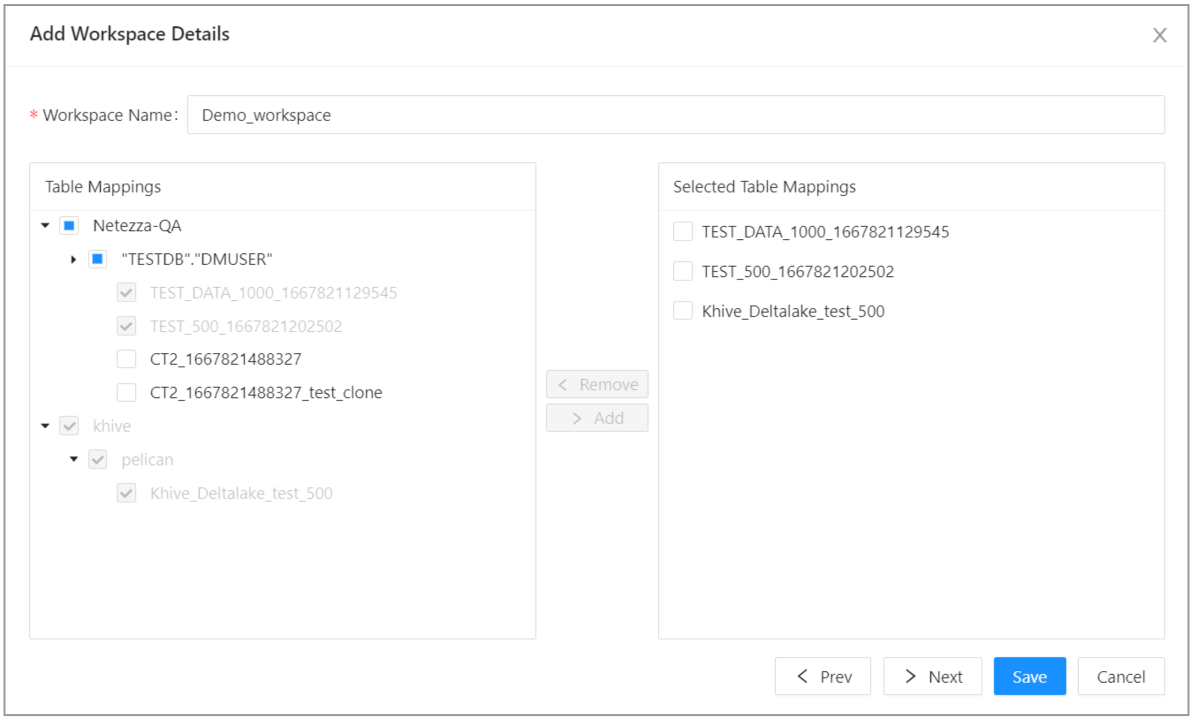
Table Mappings

- ■ Netezza-QA
  - ■ "TESTDB"."DMUSER"
    - ☑ TEST_DATA_1000_1667821129545
    - ☑ TEST_500_1667821202502
    - □ CT2_1667821488327
    - □ CT2_1667821488327_test_clone
- ☑ khive
  - ☑ pelican
    - ☑ Khive_Deltalake_test_500

Selected Table Mappings

No Data

8. Click on **Add** button to add the mappings.



Add Workspace Details

* Workspace Name: Demo_workspace

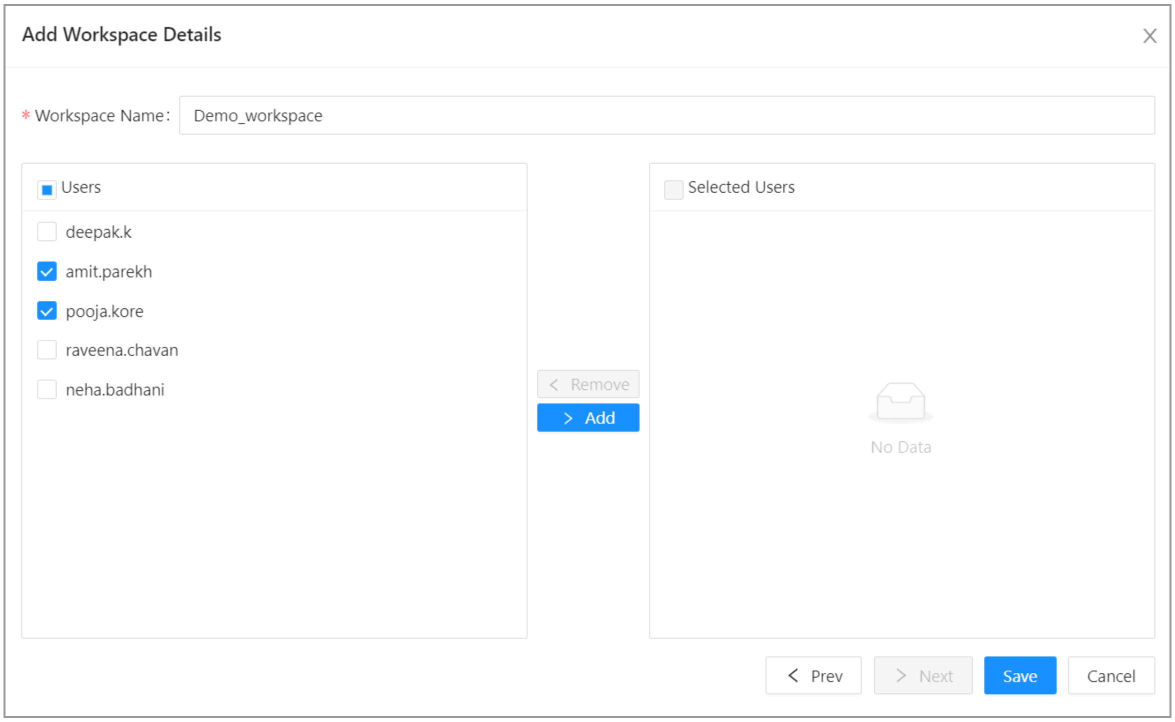
Table Mappings

- ■ Netezza-QA
  - ■ "TESTDB"."DMUSER"
    - ☑ TEST_DATA_1000_1667821129545
    - ☑ TEST_500_1667821202502
    - □ CT2_1667821488327
    - □ CT2_1667821488327_test_clone
- ☑ khive
  - ☑ pelican
    - ☑ Khive_Deltalake_test_500

Selected Table Mappings

- □ TEST_DATA_1000_1667821129545
- □ TEST_500_1667821202502
- □ Khive_Deltalake_test_500

9. Click **Next** to go to the **Users** page.



Add Workspace Details

* Workspace Name: Demo_workspace

■ Users

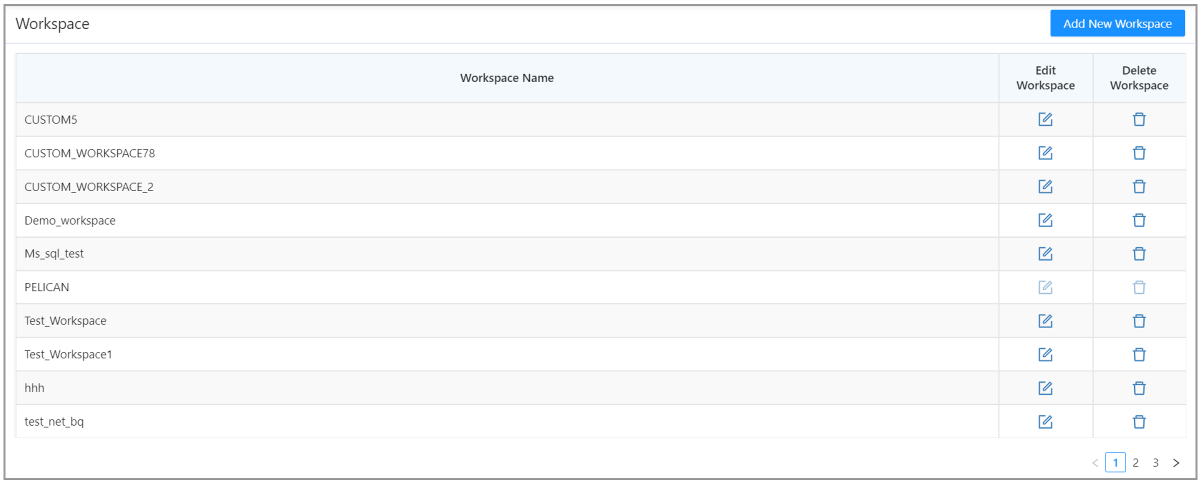
- □ deepak.k
- ☑ amit.parekh
- ☑ pooja.kore
- □ raveena.chavan
- □ neha.badhani

□ Selected Users

No Data

10. Click **Save**.

The newly added workspace will be added to the list of **Workspaces**.



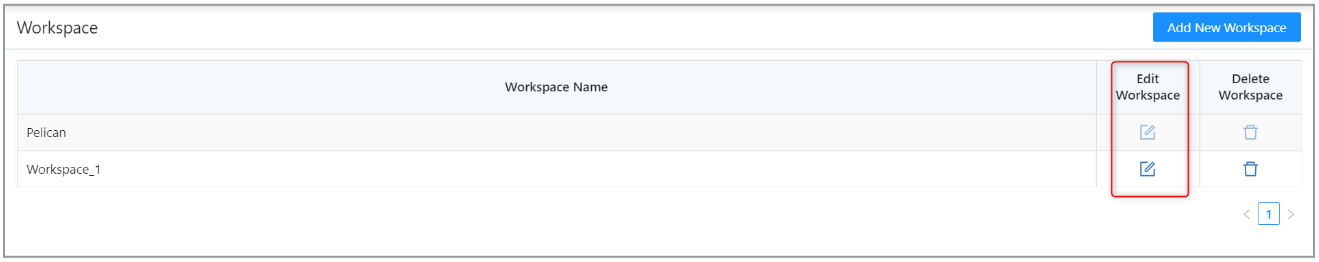
Workspace

Workspace Name	Edit Workspace	Delete Workspace
CUSTOM5	✎	🗑
CUSTOM_WORKSPACE78	✎	🗑
CUSTOM_WORKSPACE_2	✎	🗑
Demo_workspace	✎	🗑
Ms_sql_test	✎	🗑
PELICAN	✎	🗑
Test_Workspace	✎	🗑
Test_Workspace1	✎	🗑
hhh	✎	🗑
test_net_bq	✎	🗑

Notes:

- A User can have multiple workspaces.
- There is a default workspace named ‘Pelican’. It will consist of all mappings in Pelican. By default all users will have access to this workspace. If Admin wants to restrict the access of mappings to any user, he should create a new workspace with selected mappings and assign it to the user, removing the default workspace access.
- If a custom workspace is deleted and any of the users associated with it do not have any other workspace assigned to them, then default “Pelican” workspace will be assigned to such users.
- The default workspace is not editable & deletable.

6.7.1. Editing a Workspace

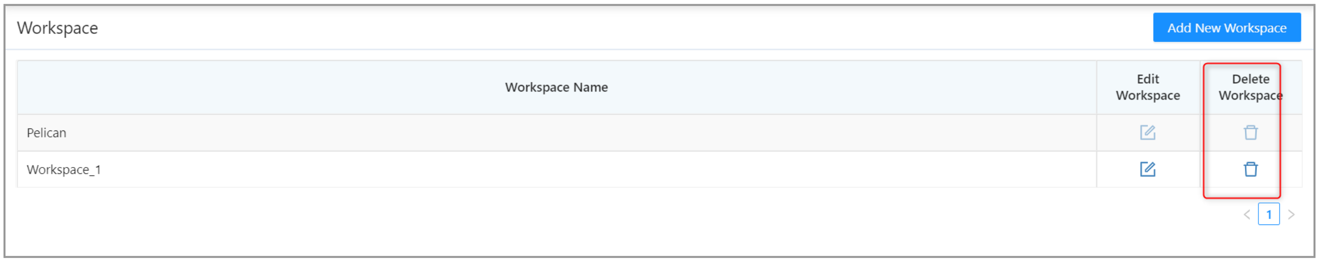


Workspace

Workspace Name	Edit Workspace	Delete Workspace
Pelican	✎	🗑
Workspace_1	✎	🗑

- Only the custom workspace can be edited.
- Click the corresponding edit icon to edit the workspaces.

6.7.2. Deleting a Workspace



Workspace

Add New Workspace

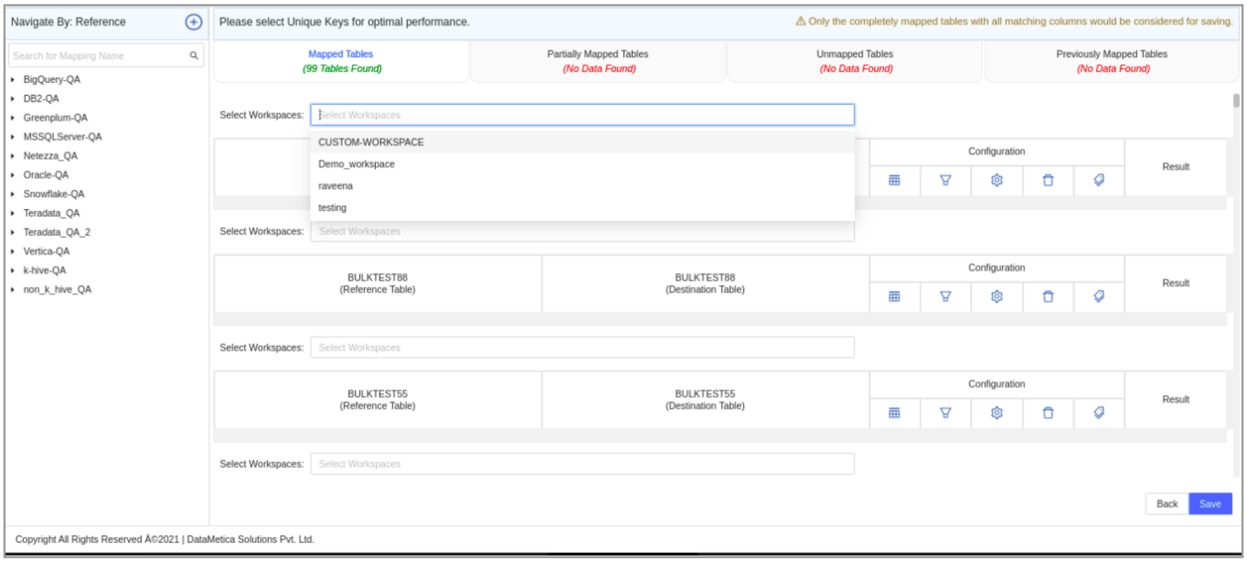
Workspace Name	Edit Workspace	Delete Workspace
Pelican		
Workspace_1		

< 1 >

- Only the custom workspace can be deleted.
- Click the corresponding delete icon to delete the workspace.

6.7.3. Adding mappings to workspaces.

- A user can assign the workspaces while creating a new mapping. By default, a new mapping will be added to the default **Pelican** workspace. Additionally, the user can select the workspaces from the workspaces assigned to him.



Navigate By: Reference

Search for Mapping Name

- BigQuery-QA
- DB2-QA
- Greenplum-QA
- MSSQLServer-QA
- Netezza_QA
- Oracle-QA
- Snowflake-QA
- Teradata_QA
- Teradata_QA_2
- Vertica-QA
- k-hive-QA
- non_k_hive_QA

Please select Unique Keys for optimal performance.

⚠ Only the completely mapped tables with all matching columns would be considered for saving.

Mapped Tables (99 Tables Found) | Partially Mapped Tables (No Data Found) | Unmapped Tables (No Data Found) | Previously Mapped Tables (No Data Found)

Select Workspaces: Select Workspaces

CUSTOM-WORKSPACE
Demo_workspace
raveena
testing

Select Workspaces: Select Workspaces

BULKTEST88 (Reference Table)	BULKTEST88 (Destination Table)	Configuration	Result

Select Workspaces: Select Workspaces

BULKTEST55 (Reference Table)	BULKTEST55 (Destination Table)	Configuration	Result

Select Workspaces: Select Workspaces

Back Save

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

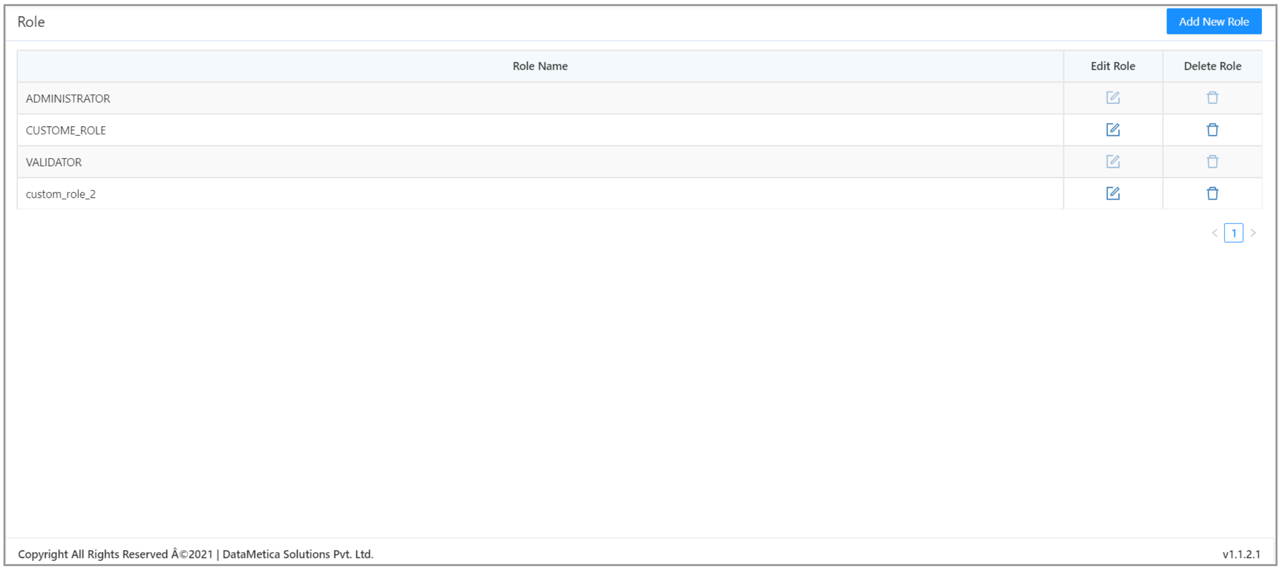
- A user can also add or remove the workspaces assigned to the mapping, using edit mapping functionality. A mapping cannot be removed from the default workspace.
- While cloning a mapping, the user can select the workspaces for it from the workspaces assigned to him. The cloned mapping will also be part of the default workspace.

6.8. Role

A role can be described as a set of permissions that are given to a user to define its access and permission rights in Pelican. A role can be assigned to multiple users, but a user can only have one role. The Steps to Add a Role in Pelican are as follows:

1. Go to **Administration** -> **User Management** -> **Role**.

   The **Role** page opens.

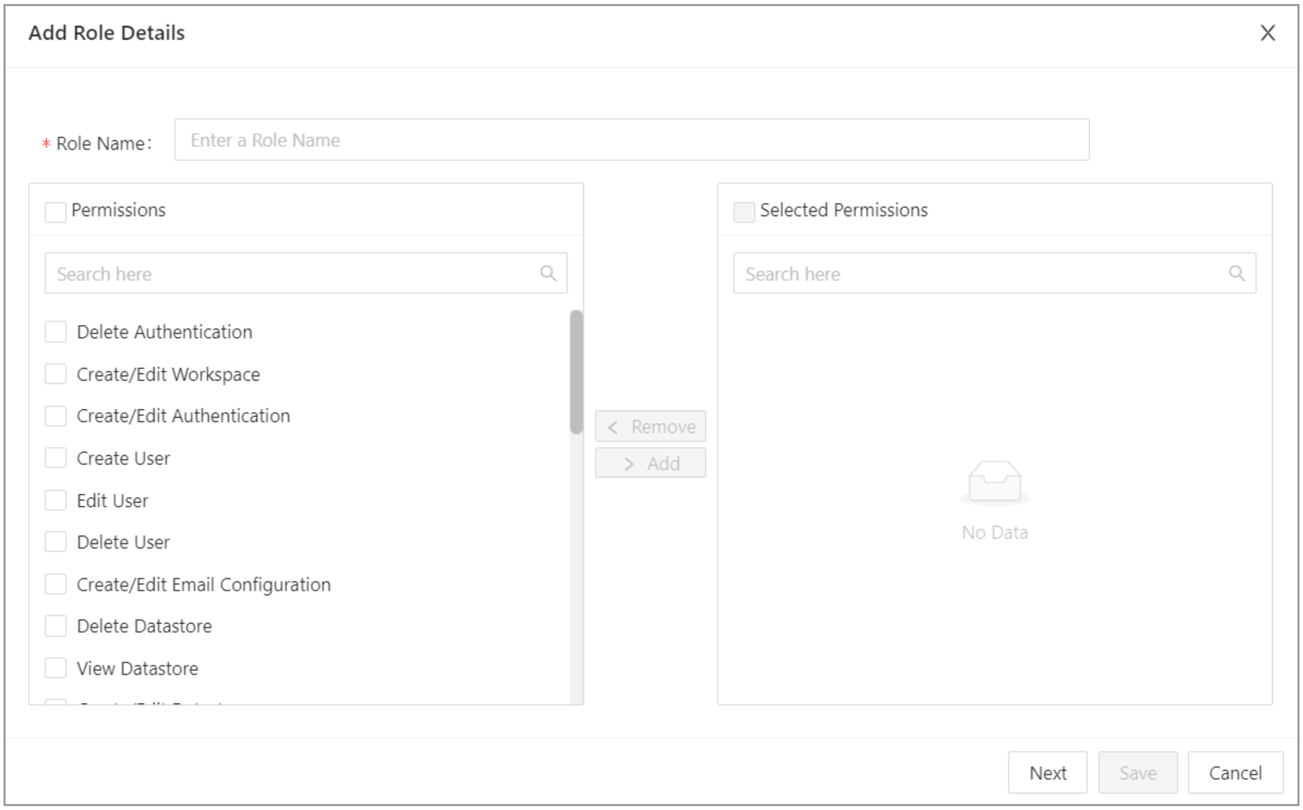


Role Name	Edit Role	Delete Role
ADMINISTRATOR	✎	🗑
CUSTOME_ROLE	✎	🗑
VALIDATOR	✎	🗑
custom_role_2	✎	🗑

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd. v1.1.2.1

2. Click **Add New Role** button.

   The **Add Role Details** page opens.



Add Role Details

* Role Name: Enter a Role Name

□ Permissions

- □ Delete Authentication
- □ Create/Edit Workspace
- □ Create/Edit Authentication
- □ Create User
- □ Edit User
- □ Delete User
- □ Create/Edit Email Configuration
- □ Delete Datastore
- □ View Datastore

□ Selected Permissions

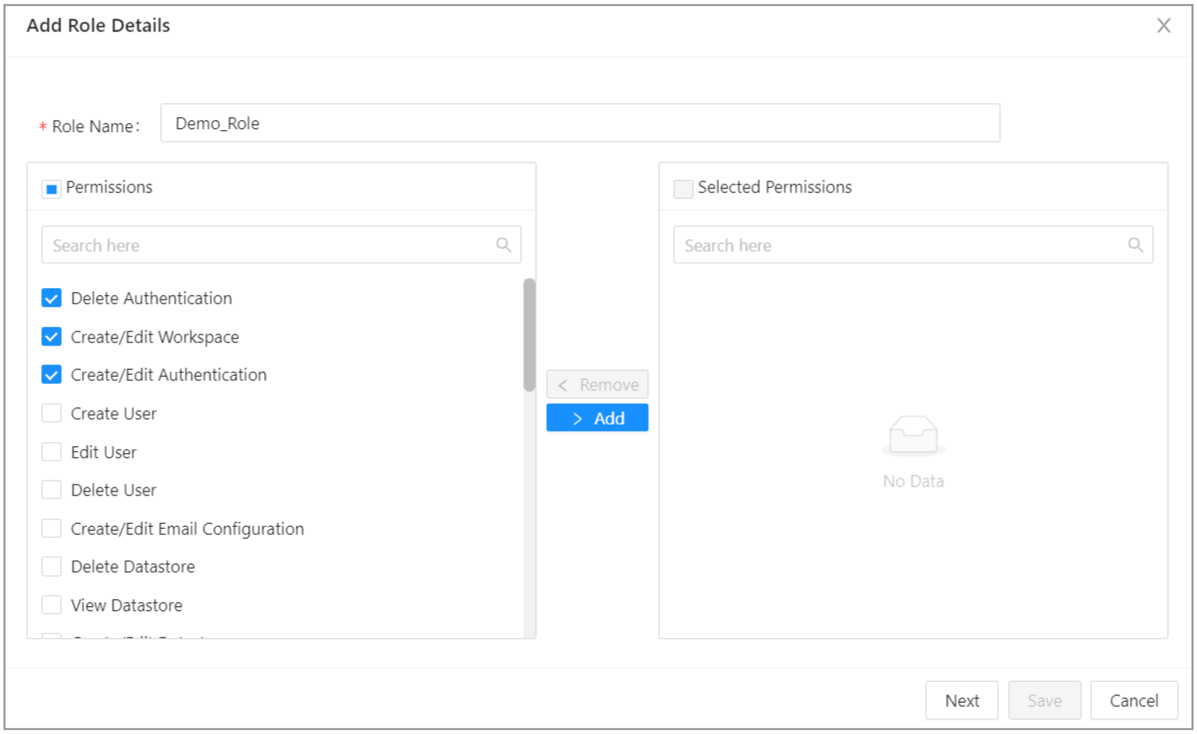
No Data

3. Enter a **Role Name**.

   The role name should be unique in the Pelican.

4. Select the required permissions from the right hand section of the page.

   Please make sure to add view level permissions if you wish to add create, delete permissions.



Add Role Details

* Role Name: Demo_Role

□ Permissions

Search here

☑ Delete Authentication
☑ Create/Edit Workspace
☑ Create/Edit Authentication
□ Create User
□ Edit User
□ Delete User
□ Create/Edit Email Configuration
□ Delete Datastore
□ View Datastore

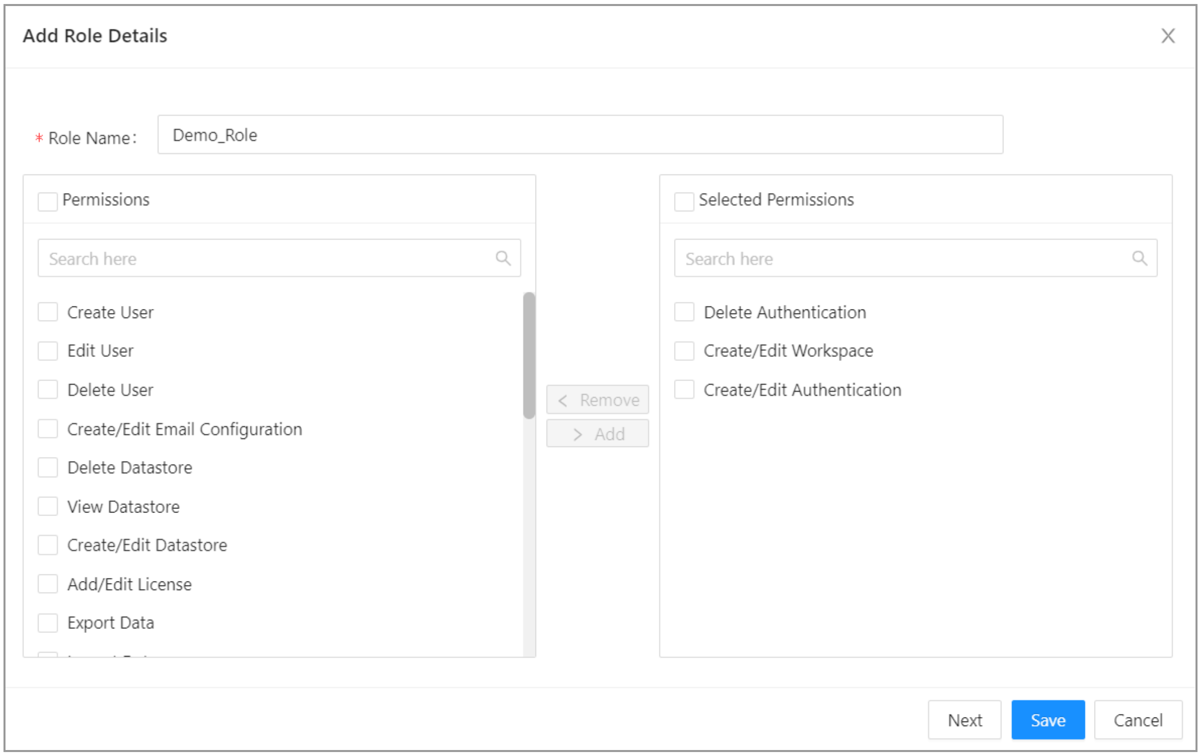
□ Selected Permissions

Search here

No Data

5. Click **Add** to add the permissions.

   Similarly, you can also remove the selected policies by clicking on the **Remove** button.



Add Role Details

* Role Name: Demo_Role

□ Permissions

Search here

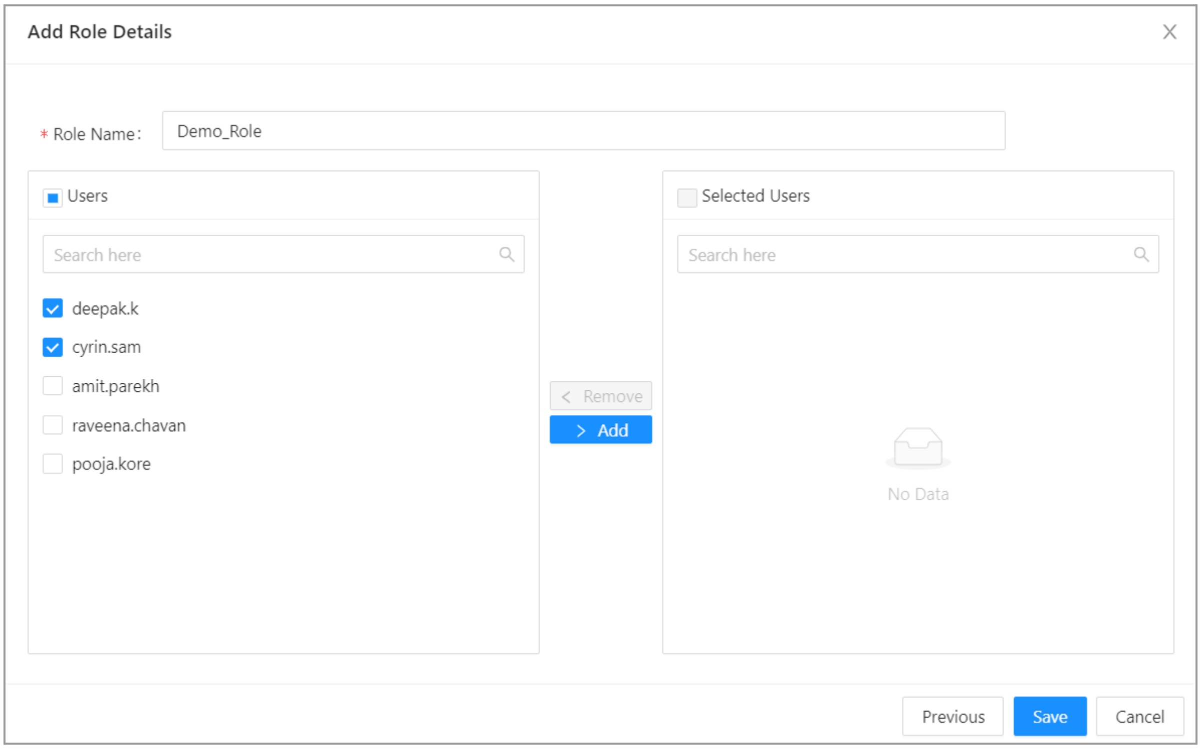
□ Create User
□ Edit User
□ Delete User
□ Create/Edit Email Configuration
□ Delete Datastore
□ View Datastore
□ Create/Edit Datastore
□ Add/Edit License
□ Export Data

□ Selected Permissions

Search here

□ Delete Authentication
□ Create/Edit Workspace
□ Create/Edit Authentication

6. Click **Next** at the bottom of the page to add the user(s).



Add Role Details

* Role Name: Demo_Role

■ Users

☑ deepak.k
☑ cyrin.sam
□ amit.parekh
□ raveena.chavan
□ pooja.kore

□ Selected Users

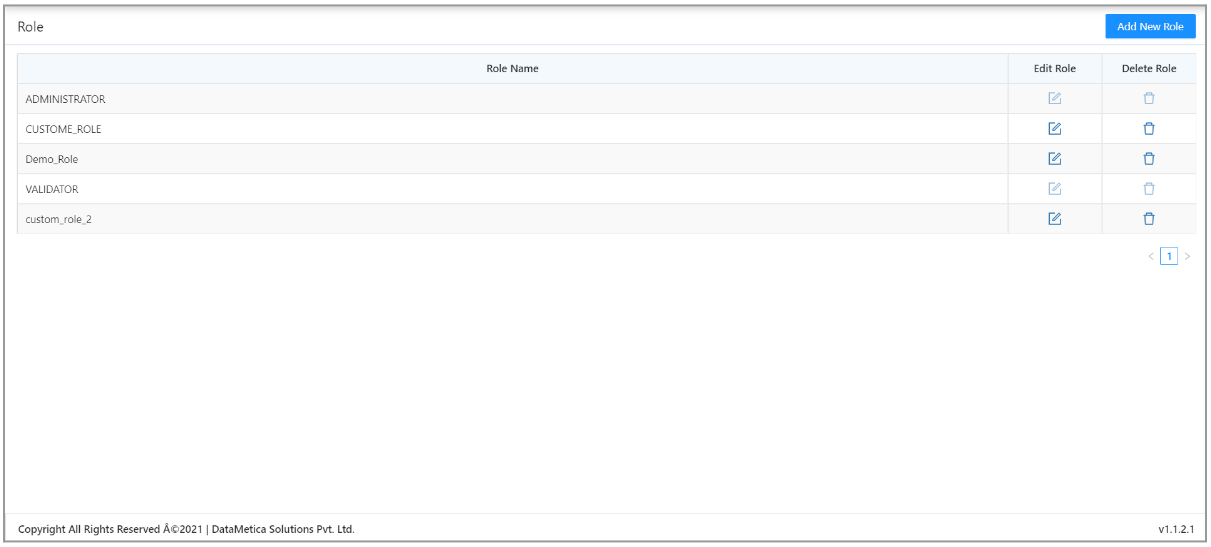
No Data

7. Select the required user(s).

   This step is optional.

8. Click **Save**.

   The newly added role will be added to the list of Roles.



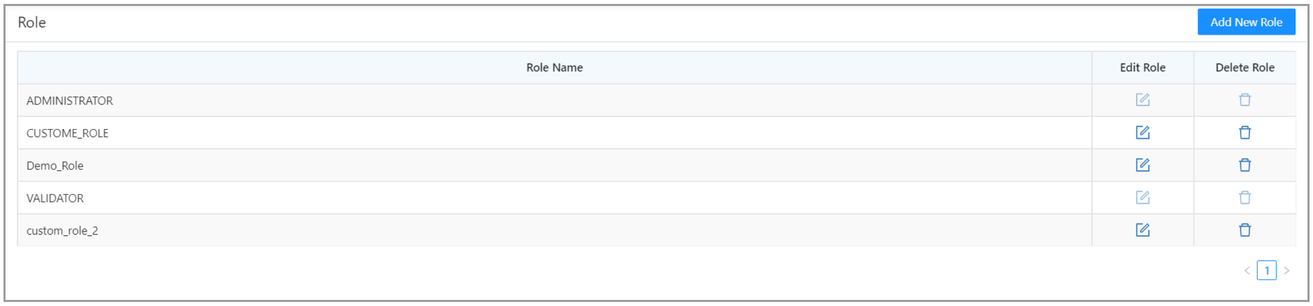
Role

Role Name	Edit Role	Delete Role
ADMINISTRATOR		
CUSTOME_ROLE		
Demo_Role		
VALIDATOR		
custom_role_2		

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

v1.1.2.1

6.8.1. Editing a Role

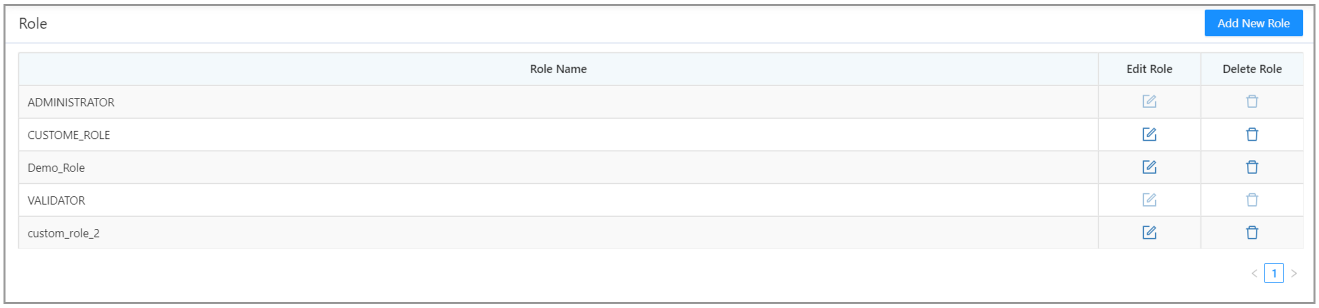


Role

Role Name	Edit Role	Delete Role
ADMINISTRATOR	✎	🗑
CUSTOME_ROLE	✎	🗑
Demo_Role	✎	🗑
VALIDATOR	✎	🗑
custom_role_2	✎	🗑

- Only the custom roles can be edited.
- Click the corresponding edit icon to edit the roles.

6.8.2. Deleting a Role



Role

Role Name	Edit Role	Delete Role
ADMINISTRATOR	✎	🗑
CUSTOME_ROLE	✎	🗑
Demo_Role	✎	🗑
VALIDATOR	✎	🗑
custom_role_2	✎	🗑

- Only the custom roles can be edited.
- Click the corresponding delete icon to delete the role.

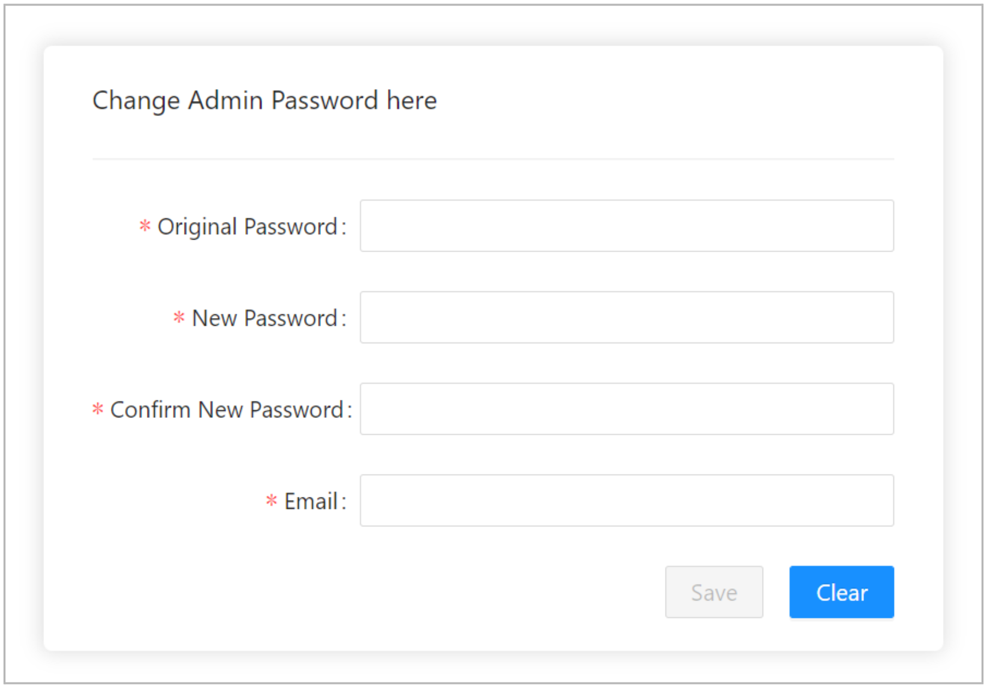
Notes:

- A User can have only one Role.
- If a custom role is deleted, all the users having that role, will be assigned a default Validator role. Administrator should explicitly assign a different custom role if he wants.
- The default Pelican roles **'Administrator'** and **'Validator'** are not editable or deletable.

6.9. Admin Password

In the Admin Password of User Management, the administrator can change the passwords from original to new password.

1. Go to **Administration > User Management > Admin Password**.



Change Admin Password here

* Original Password:

* New Password:

* Confirm New Password:

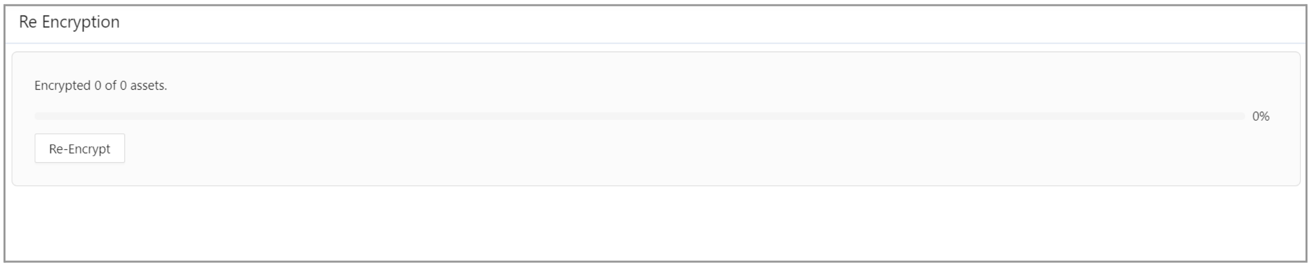
* Email:

2. Enter Original Password, New Password, Confirm New Password, Email.
3. Click SAVE.

6.10. Forgot Password

- The LDAP user will contact their respective Admin to reset the password.
- The SMTP user will contact the Pelican Admin to reset the password.

7. Re-Encryption

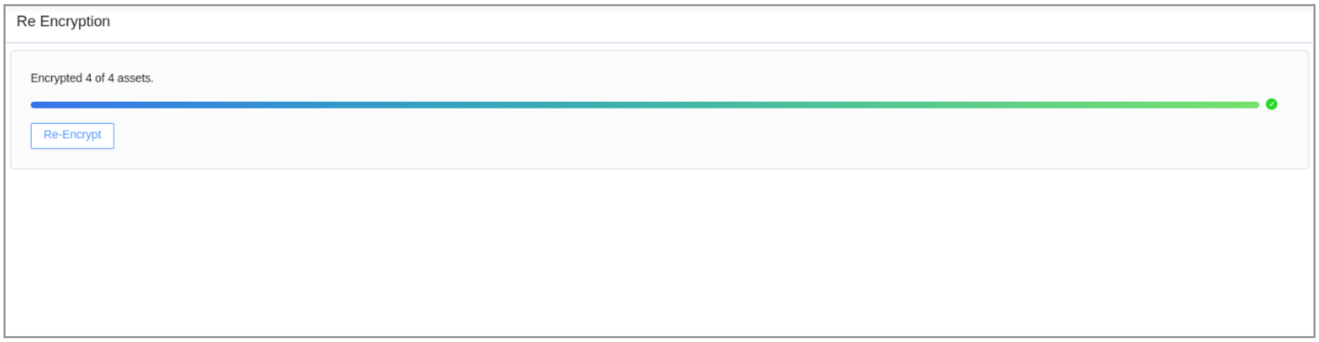


Re Encryption

Encrypted 0 of 0 assets.

0%

- Users using Pelican version 1.1.2.3 or later had samples encrypted with AES-128.
- The latest version of Pelican has an encryption algorithm of AES-256, therefore to fetch all the samples of the previous versions we need to re-encrypt the data.
- The Re Encryption feature enables us to do so by clicking on the **Re-Encrypt** button.



Re Encryption

Encrypted 4 of 4 assets.

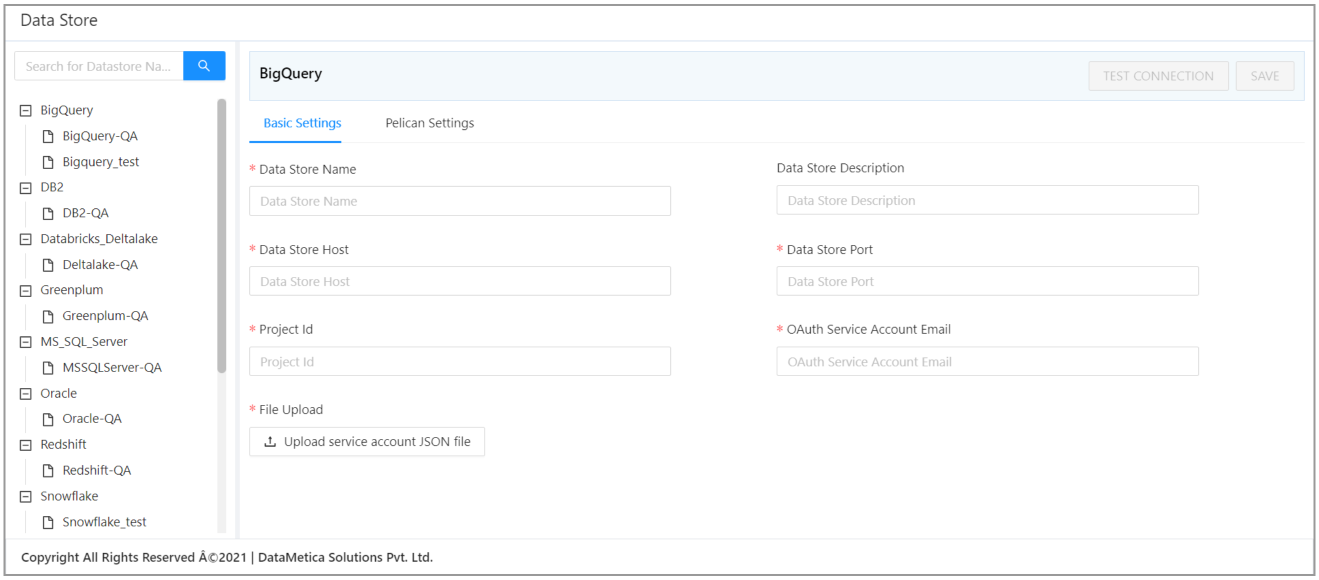
8. Datastore Configuration

To compare and validate data across various data stores, first, you need to configure source and destination datastores in the Pelican. Pelican provides predefined data store types under which you can configure data stores as per the business requirement. The datastore configuration is similar for all the data stores except HIVE which has a few more steps.

Let’s see how to configure the Datastores.

Go to **Administration Configure Data Store**.

This displays the **Data Store screen** as shown below.



8.1. Steps for Adding Database Configuration

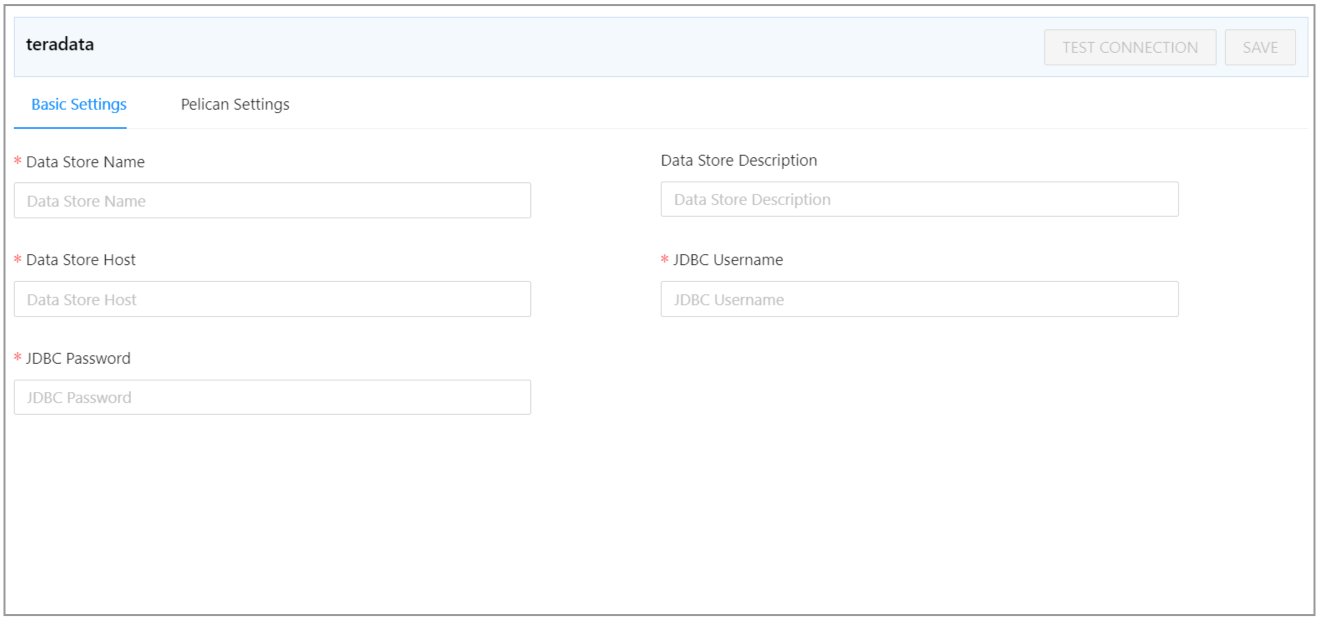
8.1.1. Steps for Teradata Configuration

1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible, at the left-hand side pane.

2. Select **Teradata** in the data store field.

   A screen to enter the corresponding datastore details will be visible.



teradata

* Data Store Name

Data Store Name

Data Store Description

Data Store Description

* Data Store Host

Data Store Host

* JDBC Username

JDBC Username

* JDBC Password

JDBC Password

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

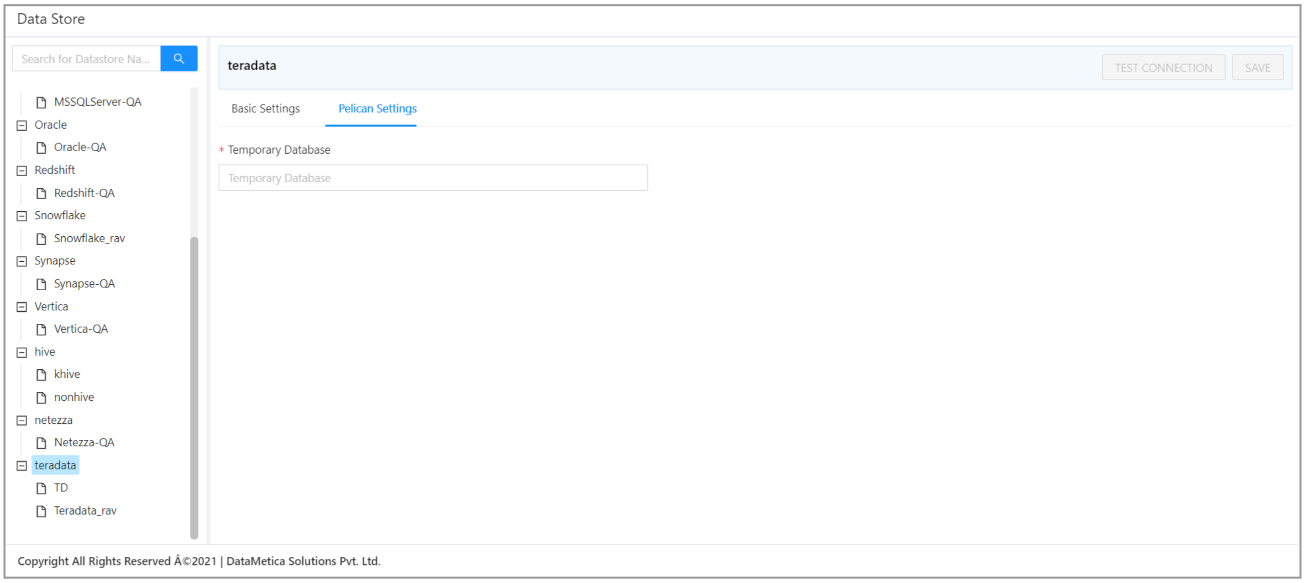
   The newly created datastore will be displayed in the datastore list.

Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Descriptions

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings Description



Data Store

Search for Datastore Na...

- MSSQLServer-QA
- Oracle
  - Oracle-QA
- Redshift
  - Redshift-QA
- Snowflake
  - Snowflake_rav
- Synapse
  - Synapse-QA
- Vertica
  - Vertica-QA
- hive
  - khive
  - nonhive
- netezza
  - Netezza-QA
- teradata
  - TD
  - Teradata_rav

teradata

TEST CONNECTION SAVE

Basic Settings | Pelican Settings

* Temporary Database

Temporary Database

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

Field	Description
Temporary Database	Database used by Pelican for its functioning.

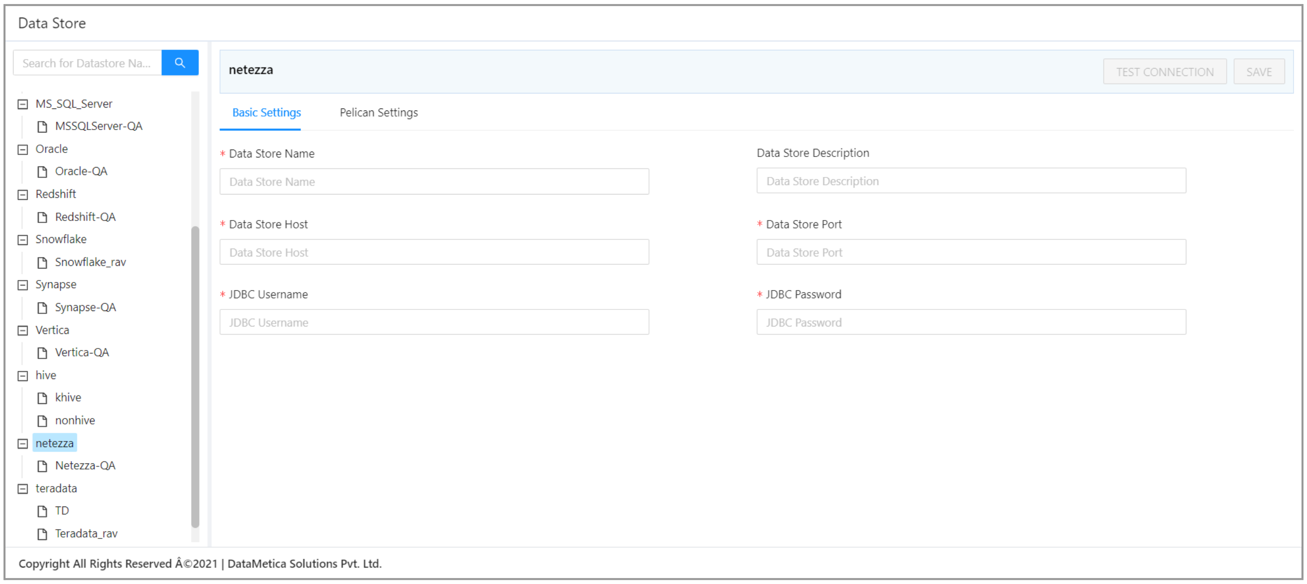
8.1.2. Steps for Netezza Configuration

1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible, at the left-hand side pane.

2. Select the **Netezza** datastore.

   A screen to enter the corresponding datastore details will be visible.



3. Enter information in the **Basic Settings** and **Pelican Settings** to configure the data store.
4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.
5. Click **SAVE**.

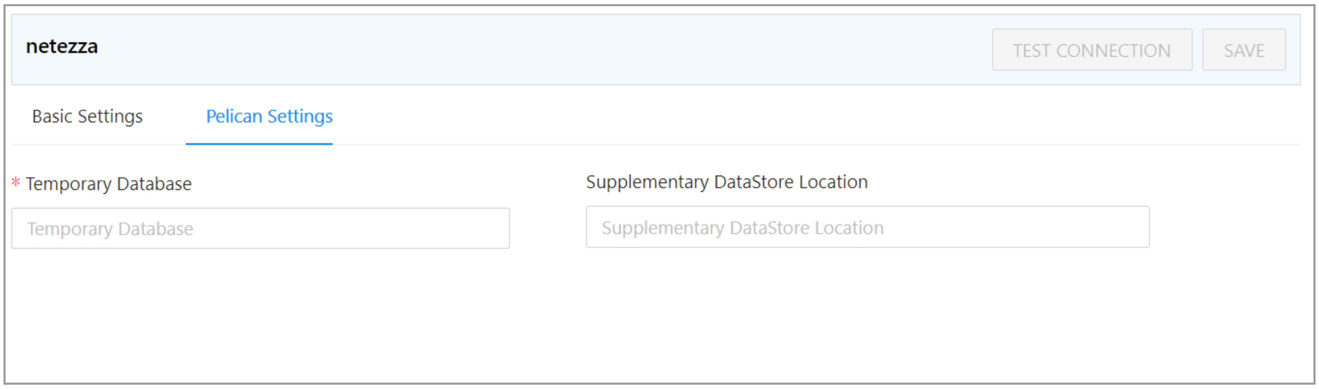
   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Field Descriptions

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings Field Description



netezza

* Temporary Database

Temporary Database

Supplementary DataStore Location

Supplementary DataStore Location

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

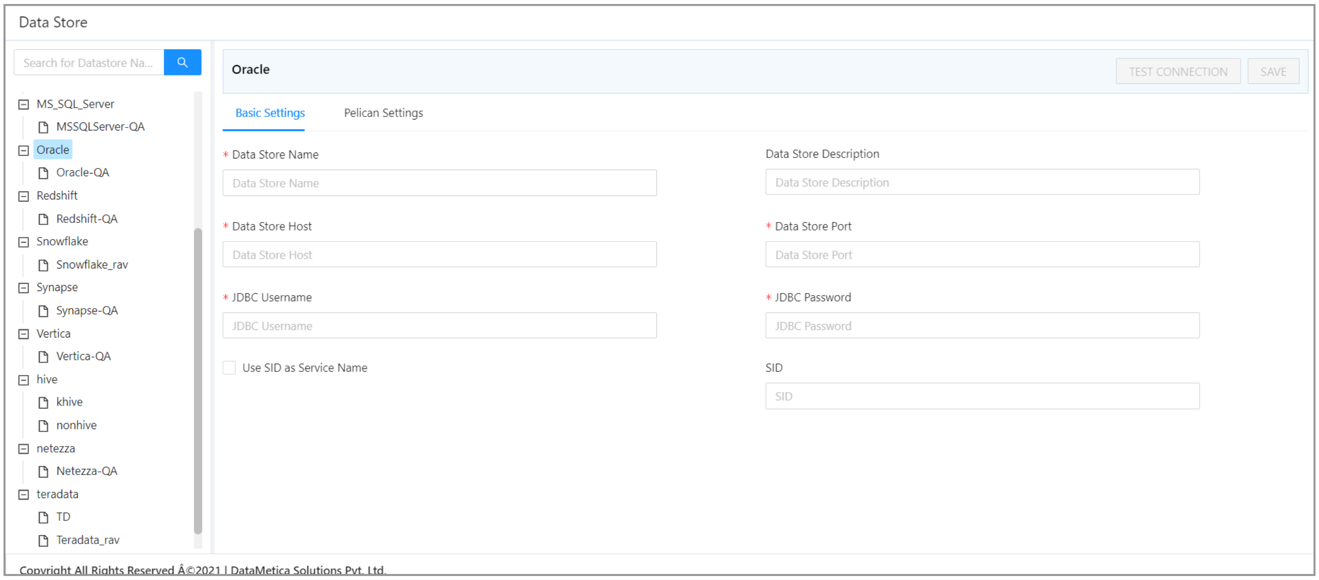
8.1.3. Steps for Oracle Configuration

1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible at the left-hand side pane.

2. Select the **Oracle**.

   A screen to enter the corresponding datastore details will be visible.



Data Store

Search for Datastore Na...

- ⊟ MS_SQL_Server
  - MSSQLServer-QA
- ⊟ Oracle
  - Oracle-QA
- ⊟ Redshift
  - Redshift-QA
- ⊟ Snowflake
  - Snowflake_rav
- ⊟ Synapse
  - Synapse-QA
- ⊟ Vertica
  - Vertica-QA
- ⊟ hive
  - khive
  - nonhive
- ⊟ netezza
  - Netezza-QA
- ⊟ teradata
  - TD
  - Teradata_rav

Oracle

* Data Store Name	Data Store Description
Data Store Name	Data Store Description
* Data Store Host	* Data Store Port
Data Store Host	Data Store Port
* JDBC Username	* JDBC Password
JDBC Username	JDBC Password
□ Use SID as Service Name	SID
	SID

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Descriptions

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.

JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Use SID as Service Name	If the service name is provided for an Oracle instance, then select this checkbox and enter service name value in the SID textbox.
SID	The SID is a site identifier.

Pelican Settings Description

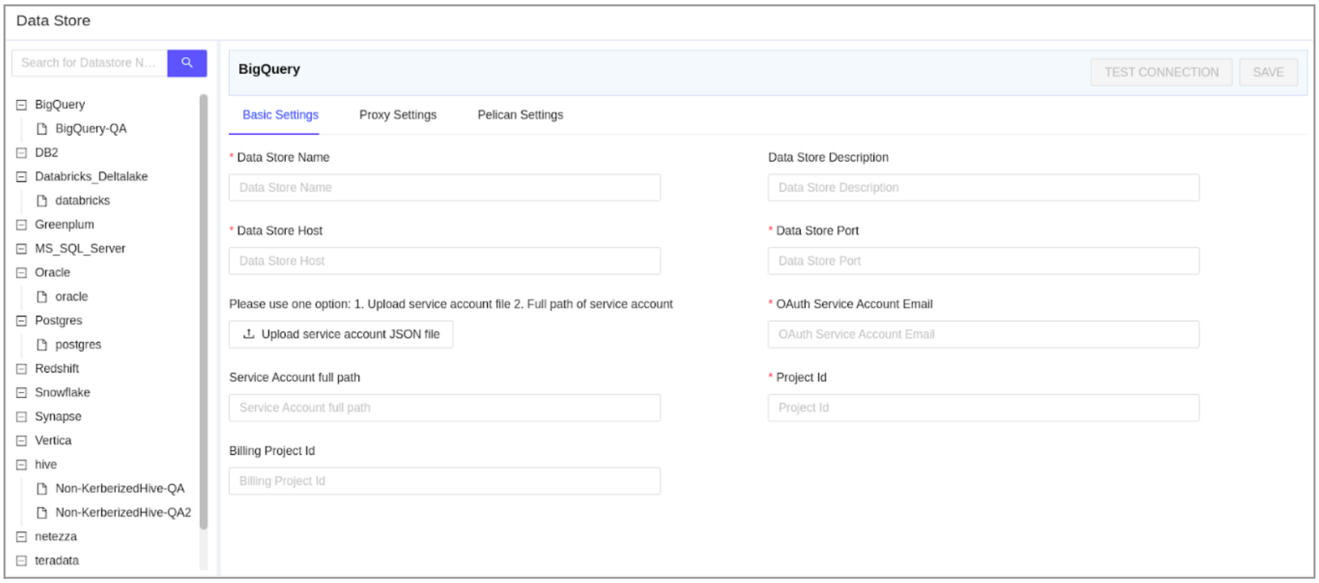
Field	Description
Temporary Database	Database used by Pelican for its functioning.

8.1.4. Steps for BigQuery Configuration

1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible at the left-hand side pane.

2. Select the **BigQuery** datastore.



3. Enter all the details in the **Basic Settings**, **Proxy Settings** and **Pelican Settings** tab fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

   The newly created datastore will be displayed in the datastore list.

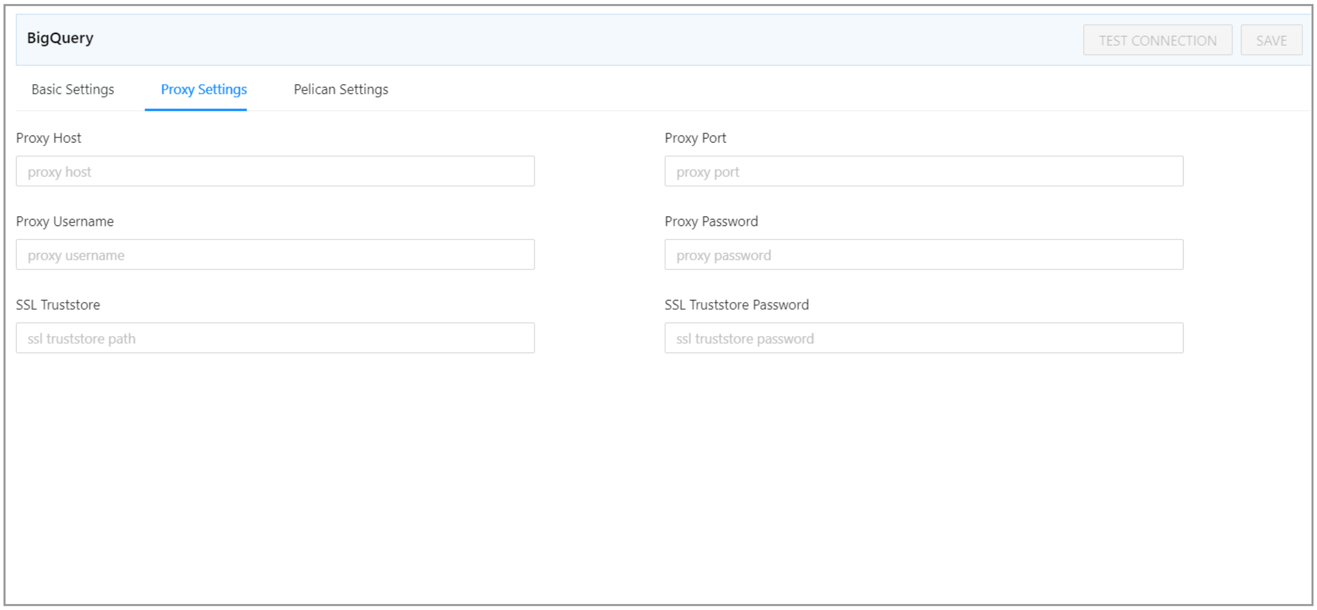
   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Field Descriptions

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	Accepts the hostname.
Data Store Port	JDBC port to connect to the data store.
Project Id	BigQuery Project to which Pelican will connect.
OAuth Service Account Email	Service account email for authentication of the above project ID.
Service Account full path	The path of the service account JSON file. We can either give the full path or upload the file.
Billing Project Id	Id of the project in which all the BigQuery jobs will be created and executed.

Proxy Settings

This setting is required only if there is a Proxy server between the Pelican server and Big Query.



Proxy Host: proxy host
Proxy Port: proxy port
Proxy Username: proxy username
Proxy Password: proxy password
SSL Truststore: ssl truststore path
SSL Truststore Password: ssl truststore password

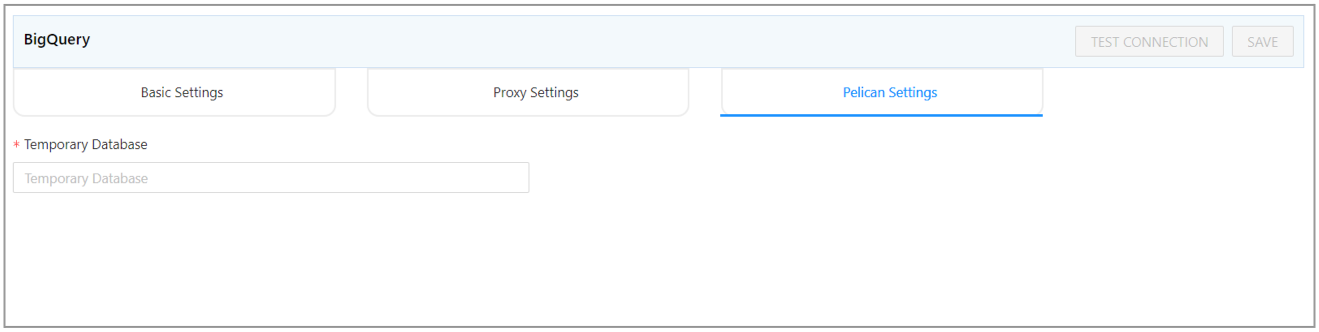
Field	Description
Proxy Host	The host name/IP of the proxy host machine.

Proxy Port	The port of the proxy host machine.
Proxy Username	The username of the proxy host machine. (Not applicable in case of SSL)
Proxy Password	The password of the proxy host machine. (Not applicable in case of SSL)
SSL Truststore	Java Cacerts path.
SSL Truststore Password	Password used to add SSL certificate in Java Cacerts.

Note:

- For SSL based proxy connection, we have to leave the proxy username and proxy password blank in Pelican.
- Pelican supports non SSL based proxy as well, however, in that case the fields - SSL Truststore and SSL Truststore Password, should be left blank.

Pelican Settings Field Description



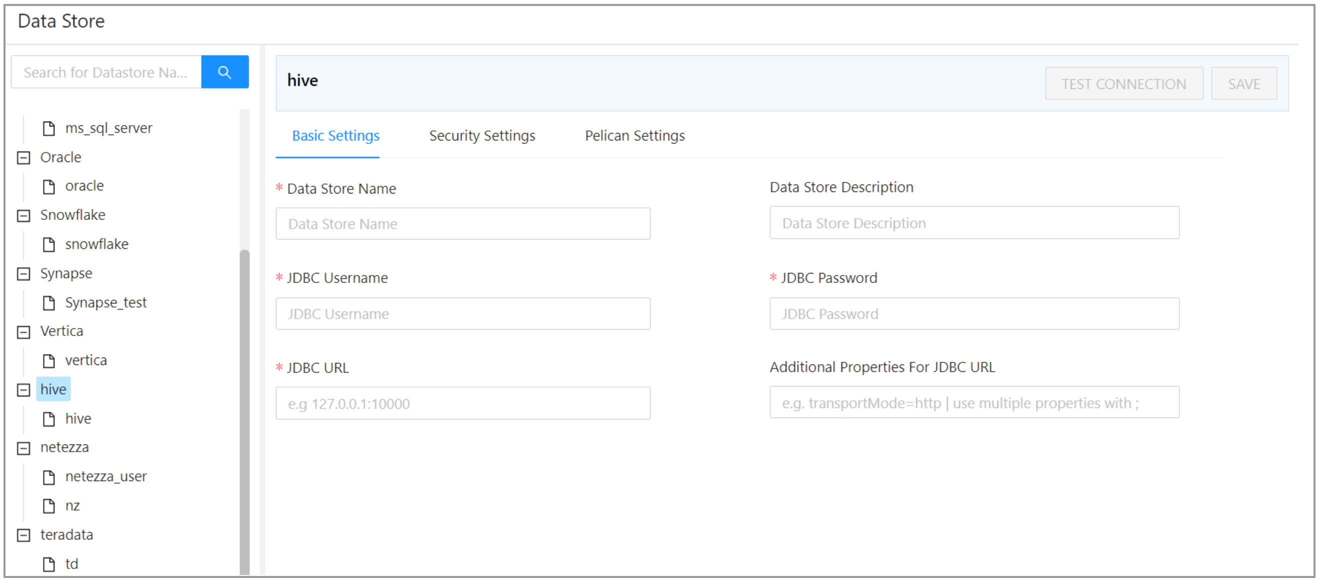
Field	Description
Temporary Database	Database used by Pelican for its functioning.

8.1.5. Steps for HIVE Configuration

1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible at the left-hand side pane.

2. Select **HIVE** data store field. A screen to enter the corresponding datastore details will be visible.



3. Enter all the details in the **Basic Settings, Security Settings** and **Pelican Settings** fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

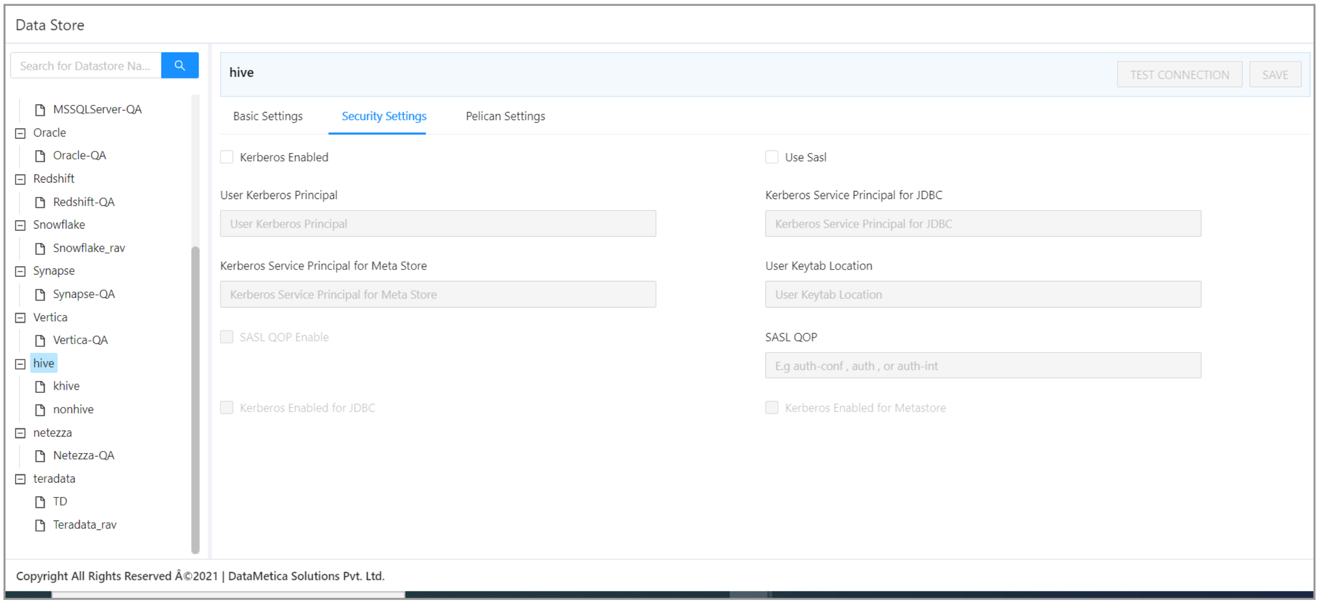
   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Field Descriptions

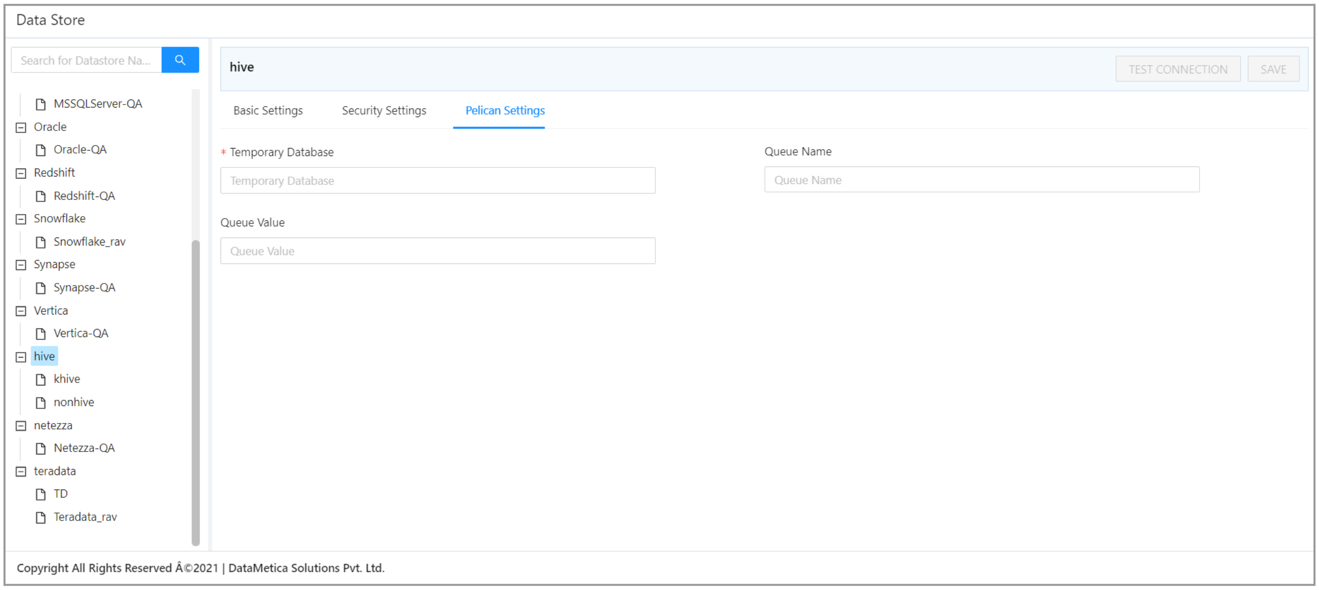
Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
JDBC URL	Hive host URL for JDBC connection.
Additional Properties For JDBC URL	Additional Properties For JDBC URL

Security Settings Field Description



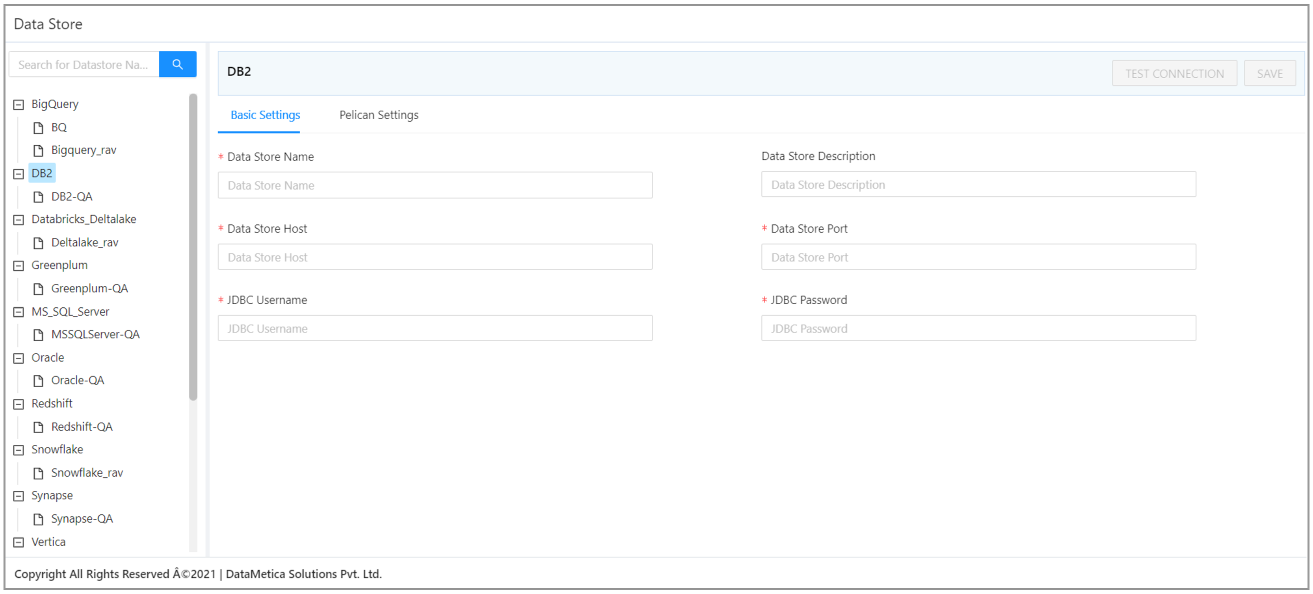
Field	Description
Kerberos Enabled	Check for kerberized Hive.
Use Sasl	Check box based if SASL is required
User Kerberos Principal	Pattern will be like user@organization.com.
Kerberos Service Principal for JDBC	A Kerberos principal is a unique identity to which Kerberos can assign tickets.
Kerberos Service Principal for Meta Store	HDFS meta store path
User Keytab Location	User keytab file location in case of Kerberos Enable.
SASL QOP Enable	Checkbox to enable SASL QOP
SASL QOP	**SASL** Mechanisms
Kerberos Enabled for JDBC	Can be checked if Kerberos Enable is true.
Kerberos Enabled for Metastore	Can be checked if Kerberos Enable is true.

Pelican Settings Field Description



Field	Description
Temporary Database	Database used by Pelican for its functioning.
Queue Name	Describes the name of the queue.
Queue Value	Describes the queue value.

8.1.6. Steps for DB2 Configuration



1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible at the left-hand side pane.
2. Select **DB2** data store field. A screen to enter the corresponding datastore details will be visible.
3. Enter all the details in the **Basic Settings** and **Pelican Settings** tabs.
4. Click **TEST CONNECTION**.

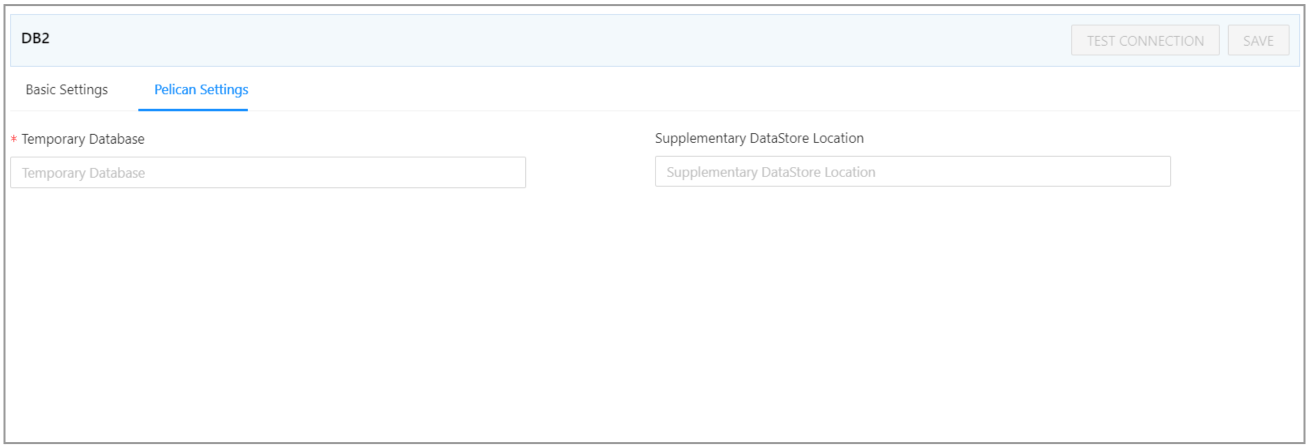
   A success message will be displayed in case of a successful connection.
5. Click **SAVE**.

   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Description

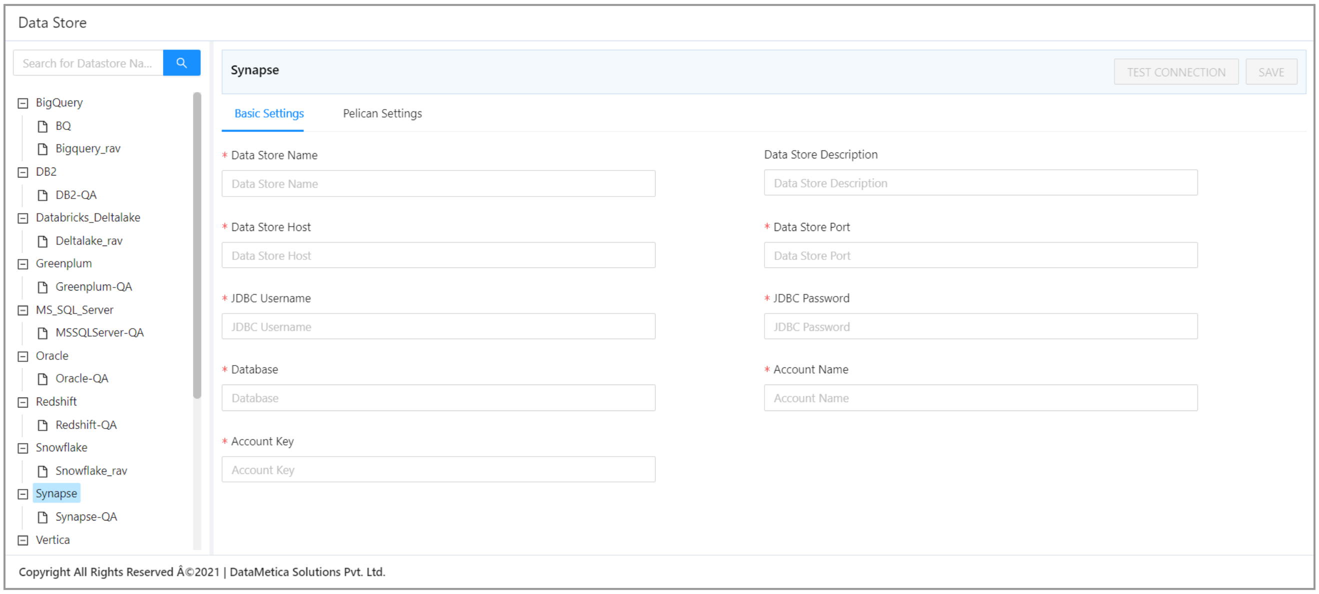
Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings Description



Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

8.1.7. Steps for Synapse Configuration



1. Go to **Administration > Configure > Datastore**.

   At the left-hand side pane, a list of predefined datastores is visible.

2. Select **Synapse** datastore.

   A screen to enter the corresponding datastore details will be visible.

3. Enter all the details in the **Basic Settings** and **Pelican Settings** fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

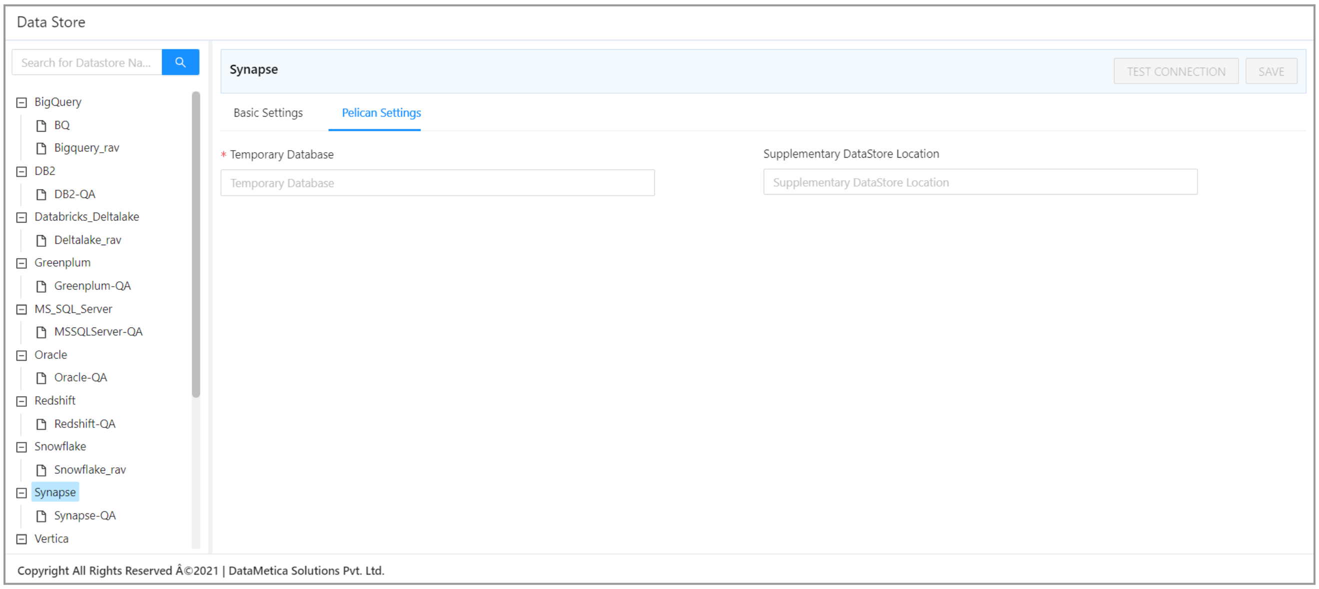
   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Description

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	Database name
Account Name	Name of the account
Account Key	Unique account key

Pelican Settings Field Description



* Temporary Database

Temporary Database

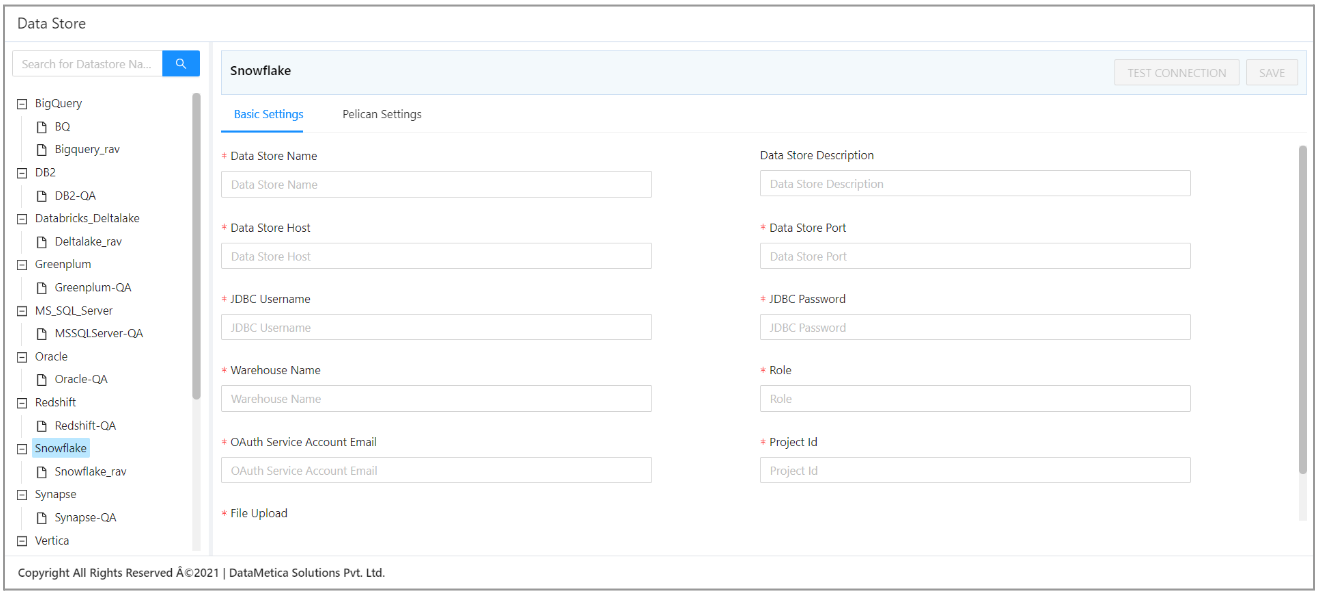
Supplementary DataStore Location

Supplementary DataStore Location

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

8.1.8. Steps for Snowflake Configuration



1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible at the left-hand side pane.

2. Select **Snowflake** datastore from the list. A screen to enter the corresponding datastore details will be visible.

3. Enter all the details in the **Basic Settings** and **Pelican Settings** fields.

Basic Settings Field Description

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Warehouse Name	Name of the warehouse
Role	Add the role

Pelican Settings Field Description

Field	Description
Temporary Database	Database used by Pelican for its functioning.

4. Click **TEST CONNECTION**.

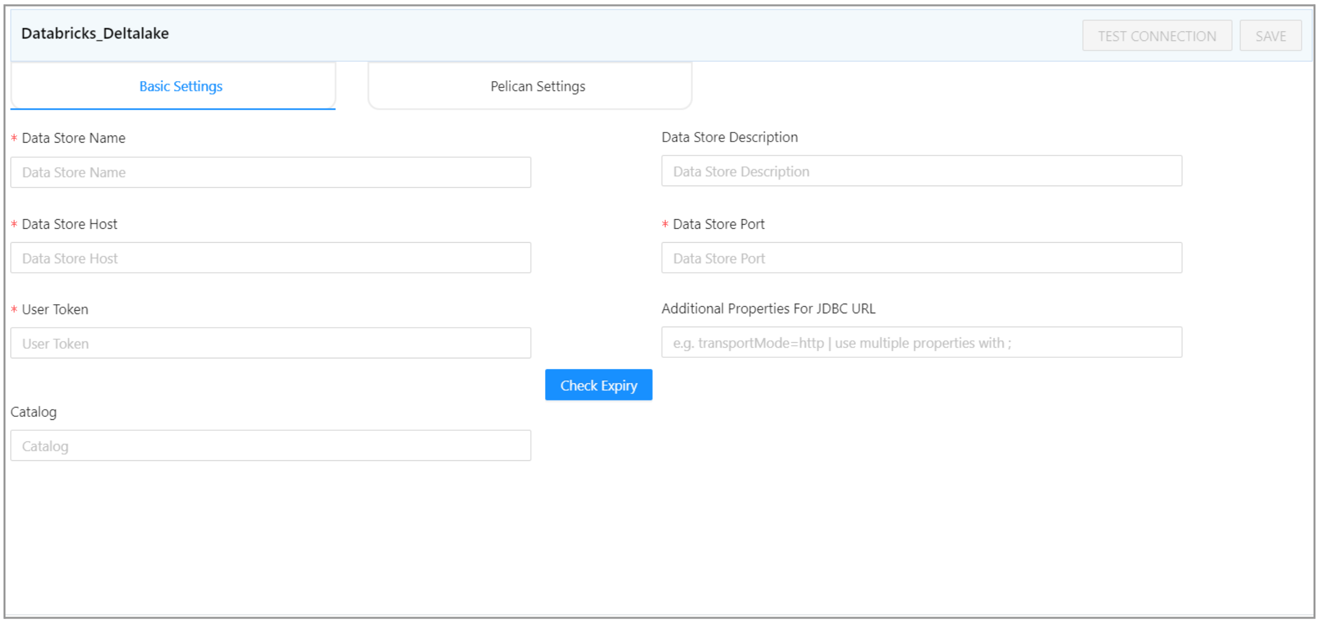
   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

8.1.9. Steps for Deltalake Configuration



The steps for Deltalake configuration are as follows:

1. Go to **Administration** > **Configure** > **Datastore**.

2. Select the Deltalake datastore from the list.

   A screen to enter the corresponding datastore details will be visible.

3. Enter all the details in the: **Basic Settings** and **Pelican Settings** fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

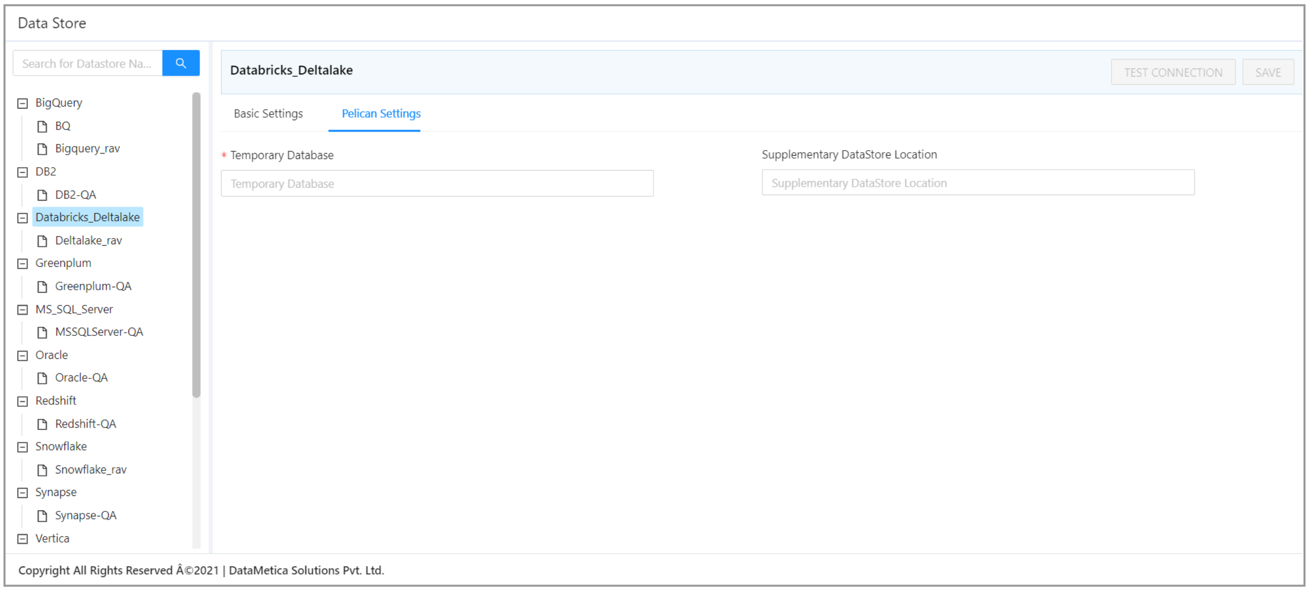
5. Click **SAVE**.

The newly created datastore will be displayed in the datastore list.

Note: The fields marked with an asterisk (*) are mandatory.

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
User Token	A token allows access to the datastore. Expiry date for the token can be set so that the token can no longer be used by an unauthorized person after its use.
Additional Properties For JDBC URL	Additional Properties required for the JDBC connection.
Catalog	Name of the Catalog.

Pelican Settings Field Description



Databricks_Deltalake

* Temporary Database: Temporary Database

Supplementary DataStore Location: Supplementary DataStore Location

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

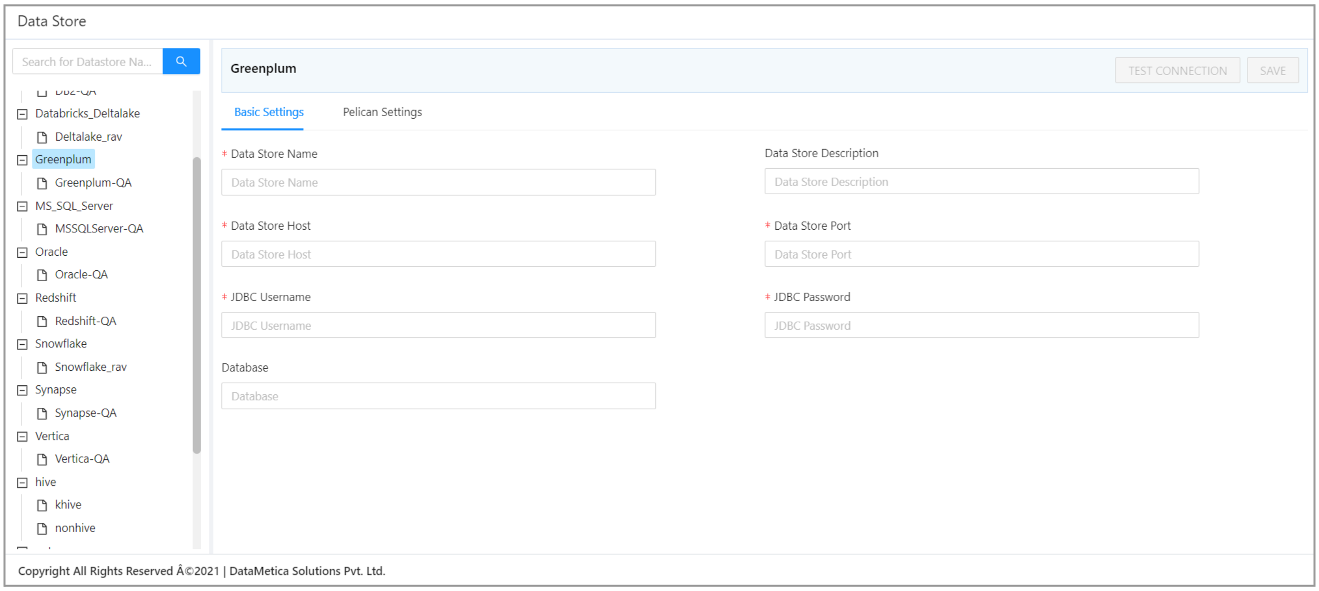
Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

8.1.10. Steps for Greenplum Configuration

1. Go to **Administration > Configure > Datastore**.

2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **Greenplum** datastore field.

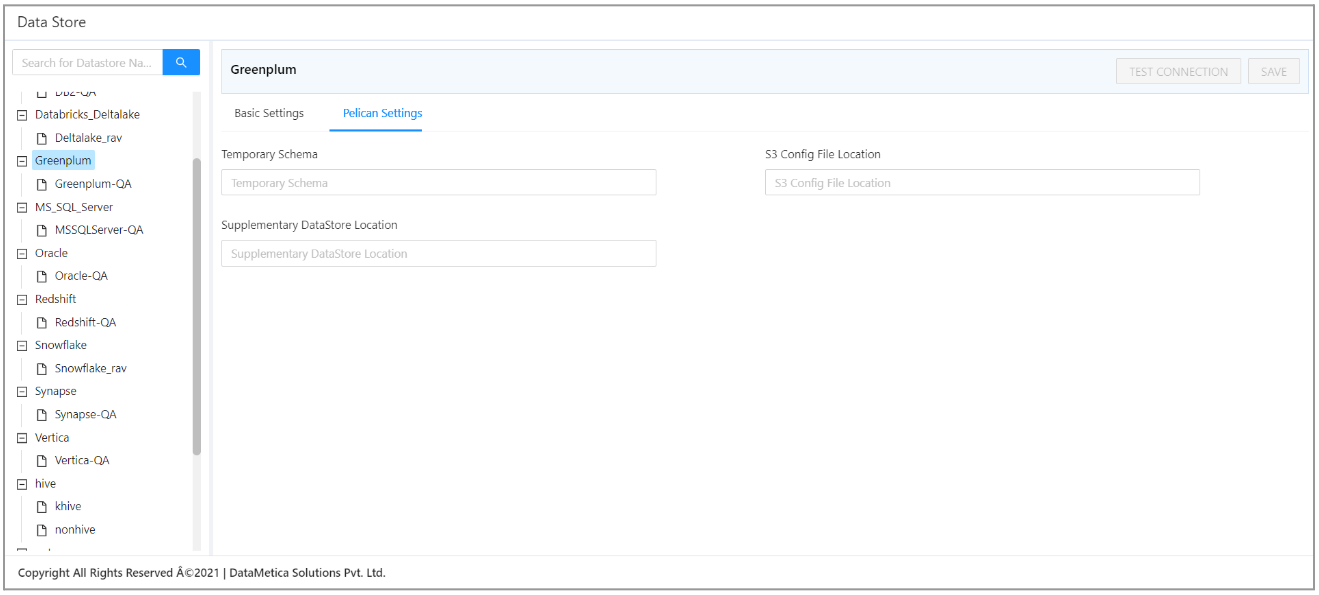
   A screen to enter the corresponding datastore details will be visible.



3. Enter all the details in the **Basic Settings**, and **Pelican Settings** tab fields.

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	A database the user wants to connect.

Pelican Settings Field Description



Field	Description
Temporary Schema	Temporary schema used by Pelican for its functioning.
S3 Config File Location	The location of S3 Config Files.
Supplementary DataStore Location	Supplementary DataStore Location

4. Click **TEST CONNECTION**.

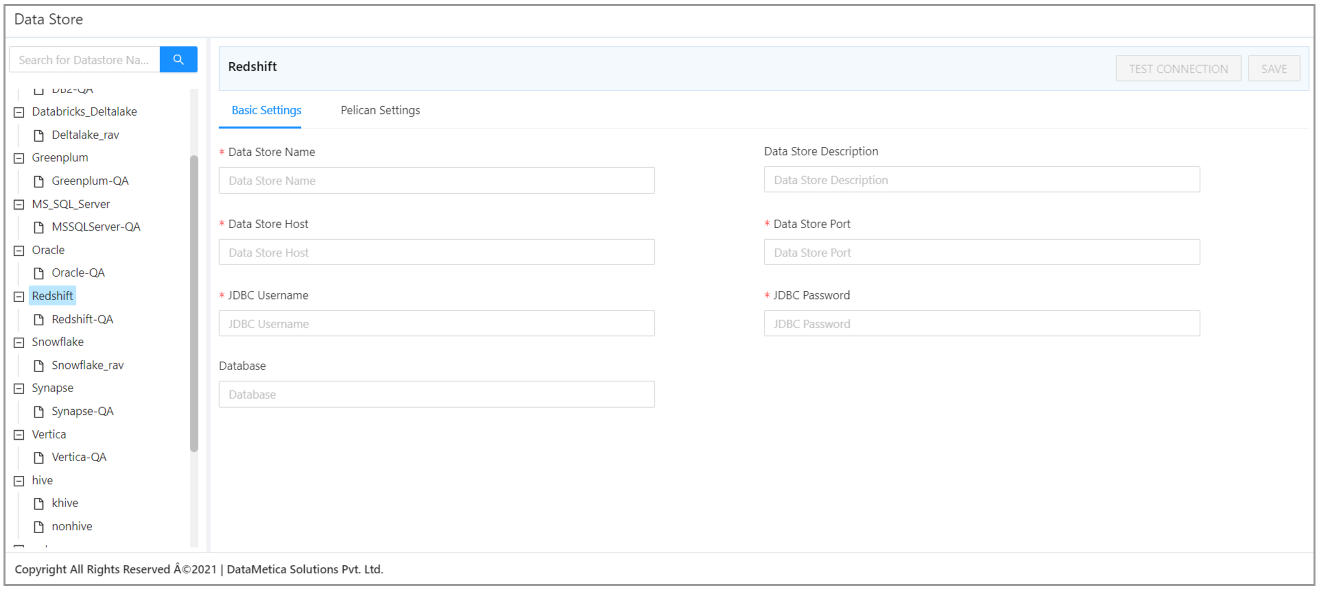
   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

7.1.11. Steps for Redshift Configuration



1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **Redshift datastore** field.

   A screen to enter the corresponding datastore details will be visible.

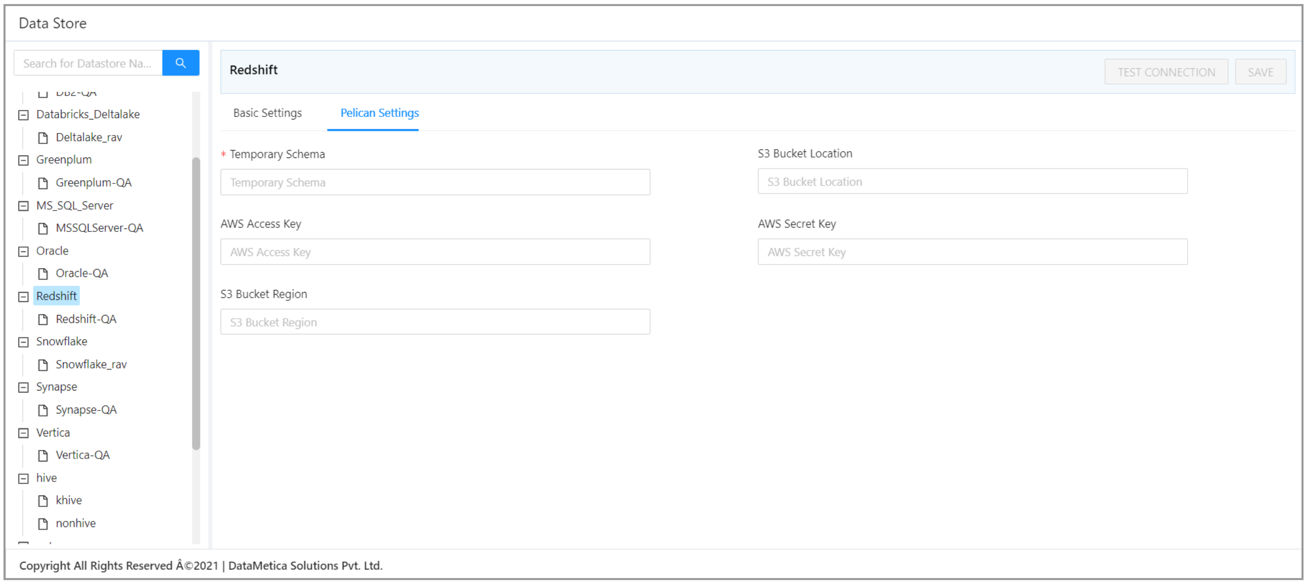
Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	The database the user wants to connect.

3. Enter all the details in the **Basic Settings**, and **Pelican Settings** tab fields.
4. Click **Test Connection**.

   A success message will be displayed in case of a successful connection.
5. Click **Save**.

   The newly created datastore will be displayed in the datastore list.

Note: The fields marked with an asterisk (*) are mandatory.



Data Store

Search for Datastore Na...

- DB2-QA
- Databricks_Deltalake
  - Deltalake_rav
- Greenplum
  - Greenplum-QA
- MS_SQL_Server
  - MSSQLServer-QA
- Oracle
  - Oracle-QA
- Redshift
  - Redshift-QA
- Snowflake
  - Snowflake_rav
- Synapse
  - Synapse-QA
- Vertica
  - Vertica-QA
- hive
  - khive
  - nonhive

Redshift

TEST CONNECTION | SAVE

Basic Settings | Pelican Settings

* Temporary Schema: Temporary Schema

S3 Bucket Location: S3 Bucket Location

AWS Access Key: AWS Access Key

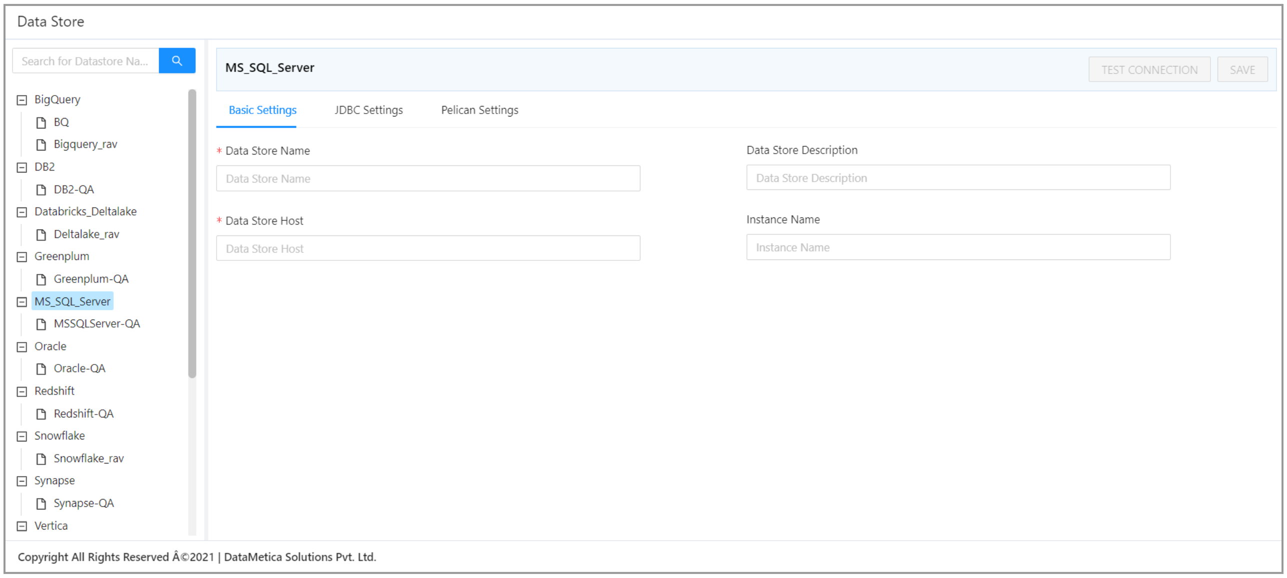
AWS Secret Key: AWS Secret Key

S3 Bucket Region: S3 Bucket Region

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

Field	Description
Temporary Schema	Temporary schema used by Pelican for its functioning.
S3 Bucket Location	Amazon S3 creates buckets in a location that you specify
AWS Access Key	Access keys are long-term credentials for an IAM user or the AWS account root user
AWS Secret key	It is a secret key similar to passwords.
S3 Bucket Region	Amazon S3 creates buckets in a Region that you specify.

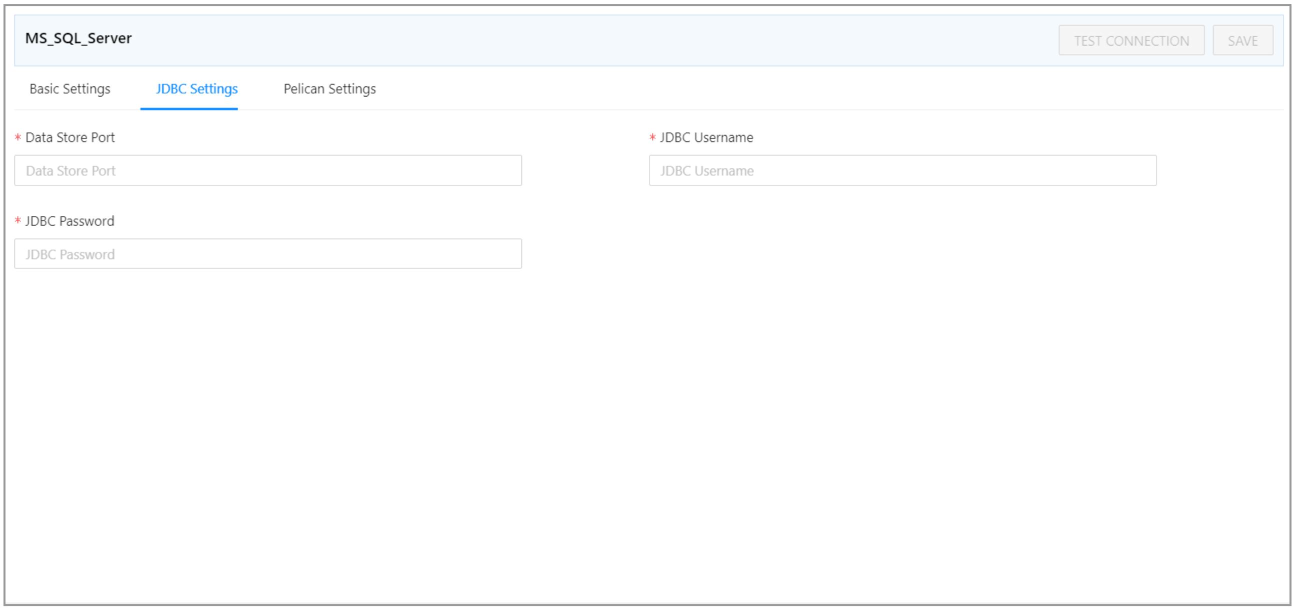
8.1.12. Steps for MSSQL Server Configuration



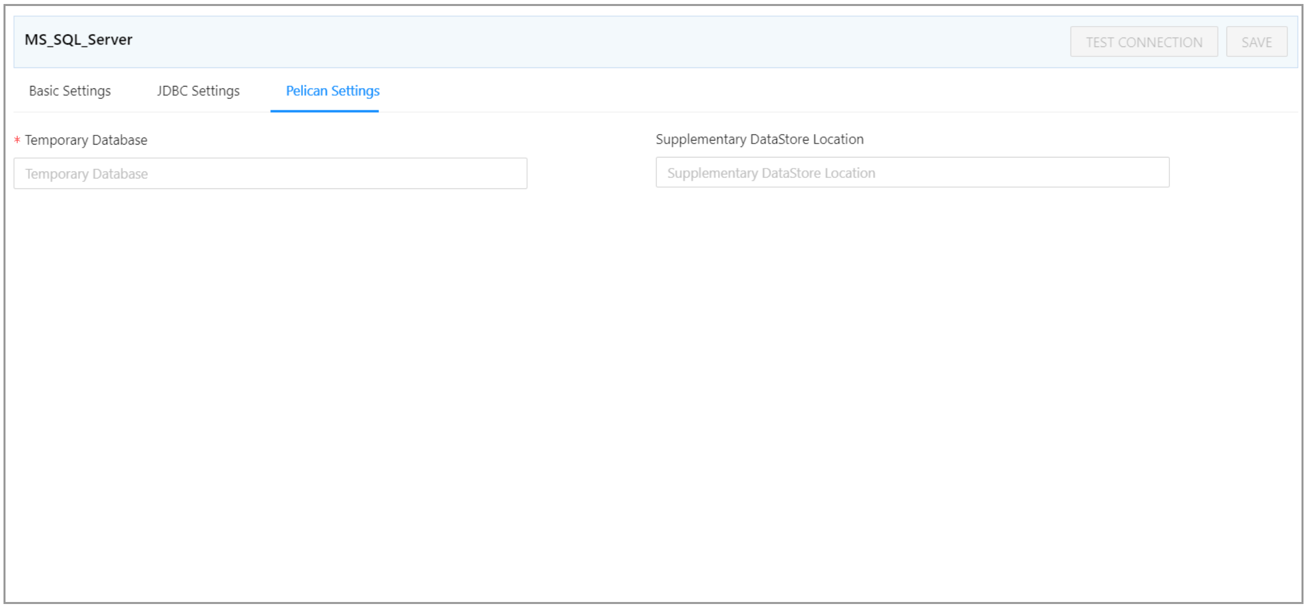
1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **MS SQL Server** datastore field.

   A screen to enter the corresponding datastore details will be visible.
3. Enter all the details in the **Basic Settings**, **JDBC Settings** and **Pelican Settings** tab fields.

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Instance Name	Name of the Instance



Field	Description
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.



Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary DataStore Location	Writable Path on the machine where Pelican is installed.

4. Click **Test Connection**.

   A success message will be displayed in case of a successful connection.

5. Click **Save**.

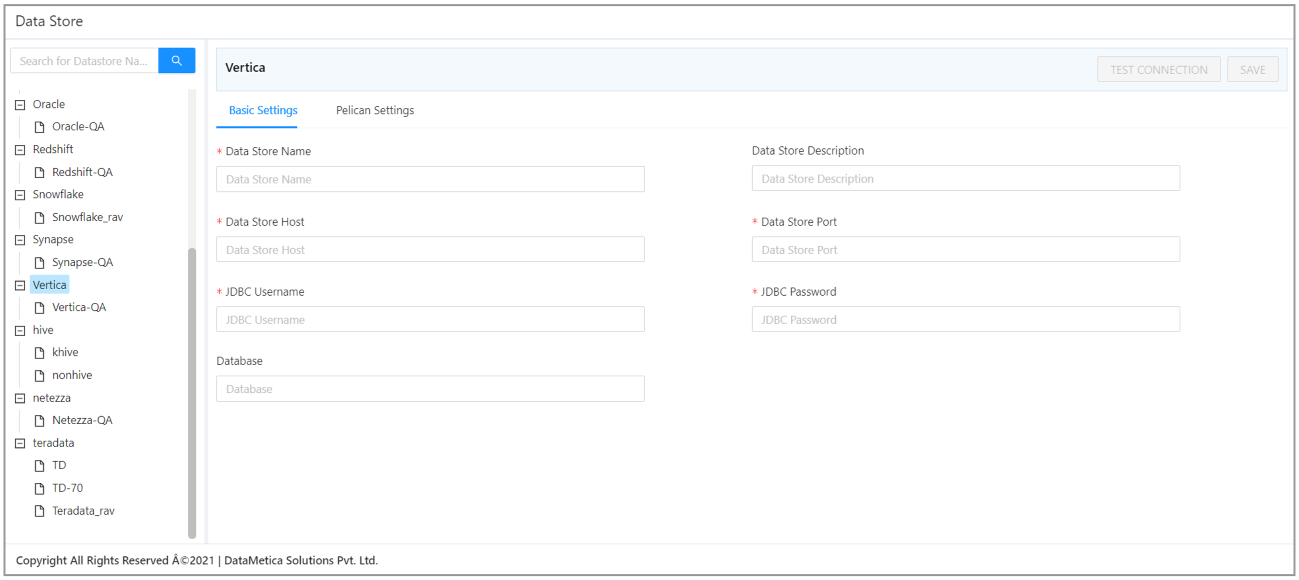
   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

8.1.13. Steps for Vertica Configuration

1. Go to **Administration > Configure > Datastore**.
2. At the left-hand side pane a list of predefined datastore lists is visible. Select the **Vertica** datastore field.

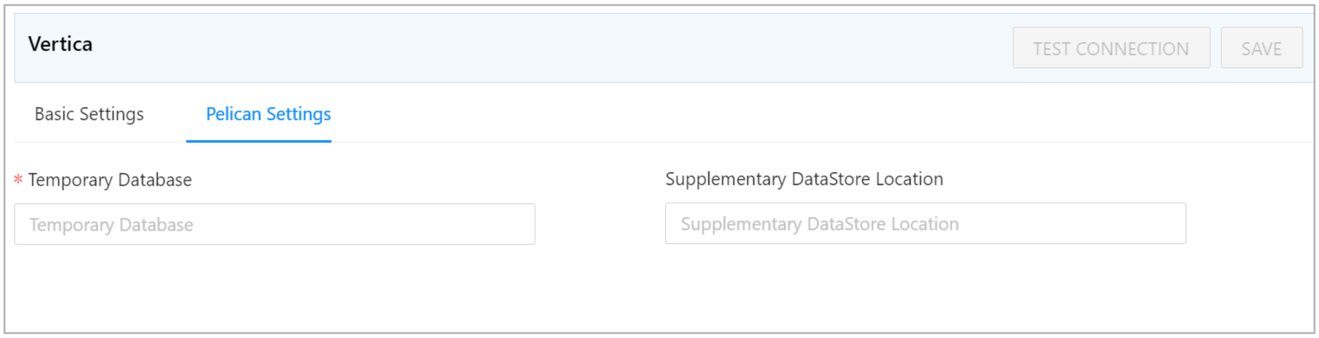
   A screen to enter the corresponding datastore details will be visible.



Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.

Data Store Host	IP address and host name of the data store host machine.
Datastore Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.
Database	Name of the Database

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.



Vertica

* Temporary Database: Temporary Database

Supplementary DataStore Location: Supplementary DataStore Location

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary Datastore Location	Writable Path on the machine where Pelican is installed.

4. Click **Test Connection**.

   A success message will be displayed in case of a successful connection.

5. Click **Save**.

   The newly created datastore will be displayed in the datastore list.

   Note: The fields marked with an asterisk (*) are mandatory.

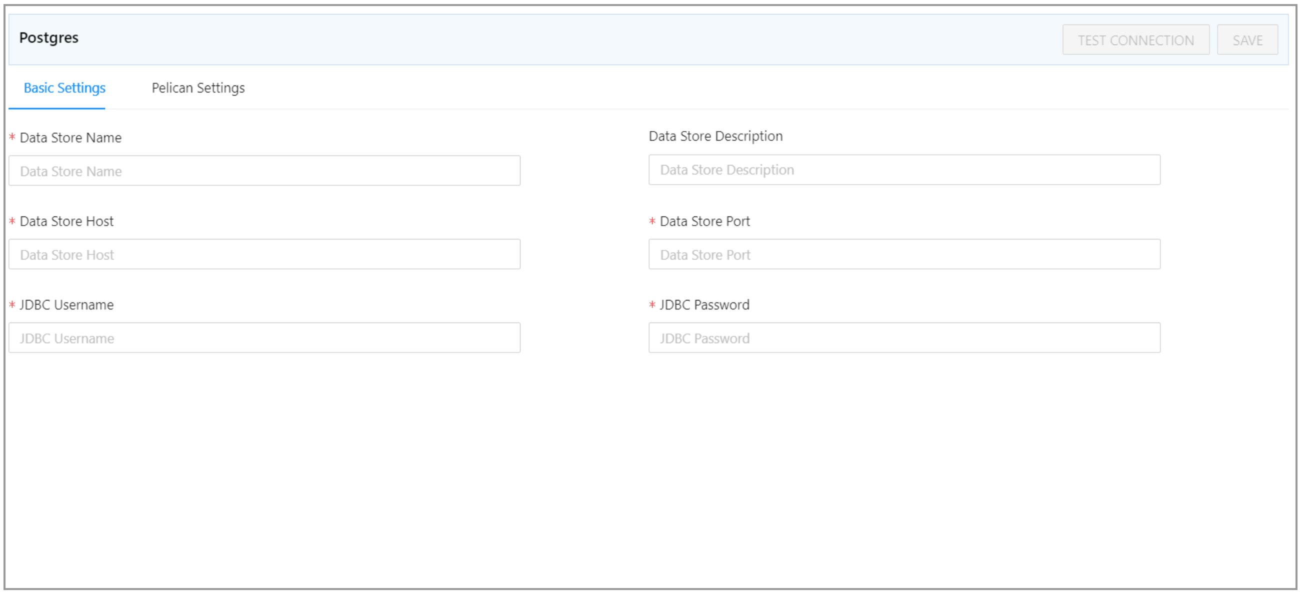
8.1.14. Steps for PostgreSQL Configuration

1. Go to **Administration > Configure > Datastore**.

   At the left-hand side pane a list of predefined datastore lists is visible.

2. Select the **Postgres** datastore field.

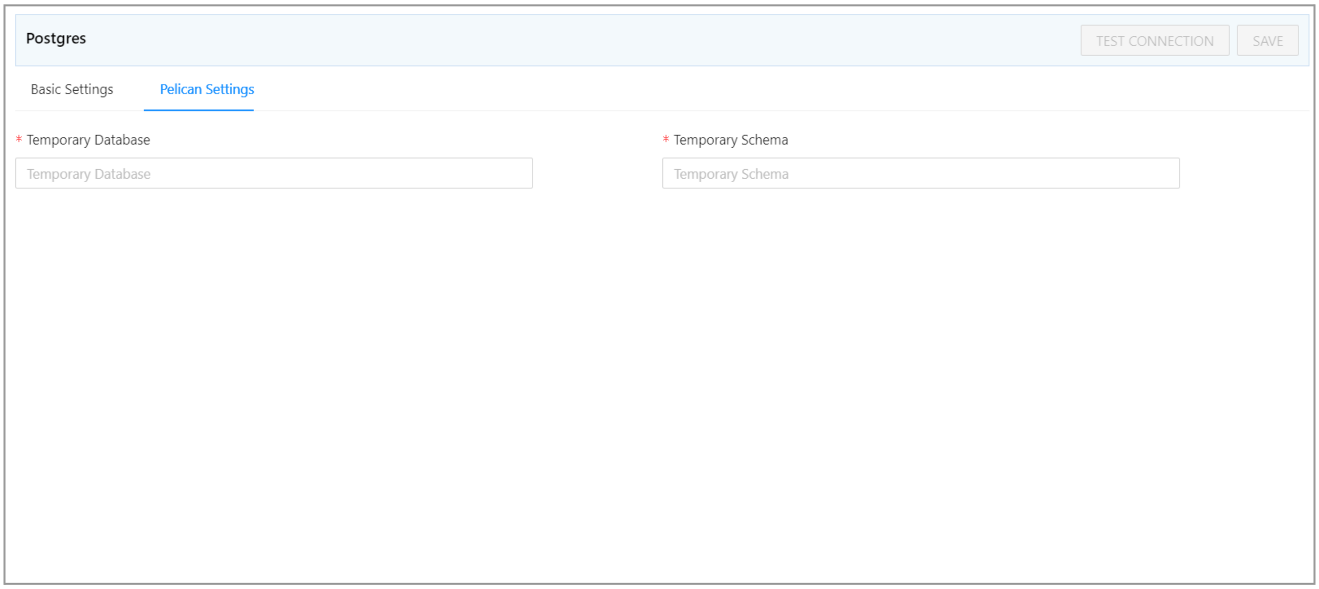
   A screen to enter the corresponding datastore details will be visible.



Postgres

* Data Store Name: Data Store Name
Data Store Description: Data Store Description
* Data Store Host: Data Store Host
* Data Store Port: Data Store Port
* JDBC Username: JDBC Username
* JDBC Password: JDBC Password

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
Data Store Host	IP address and host name of the data store host machine.
Datastore Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.



Postgres

* Temporary Database: Temporary Database
* Temporary Schema: Temporary Schema

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Temporary Schema	Schema used by Pelican for its functioning.

3. Enter all the details in the **Basic Settings** and **Pelican Settings** tab fields.
4. Click **Test Connection**.

   A success message will be displayed in case of a successful connection.
5. Click **Save**.

   The newly created datastore will be displayed in the datastore list.

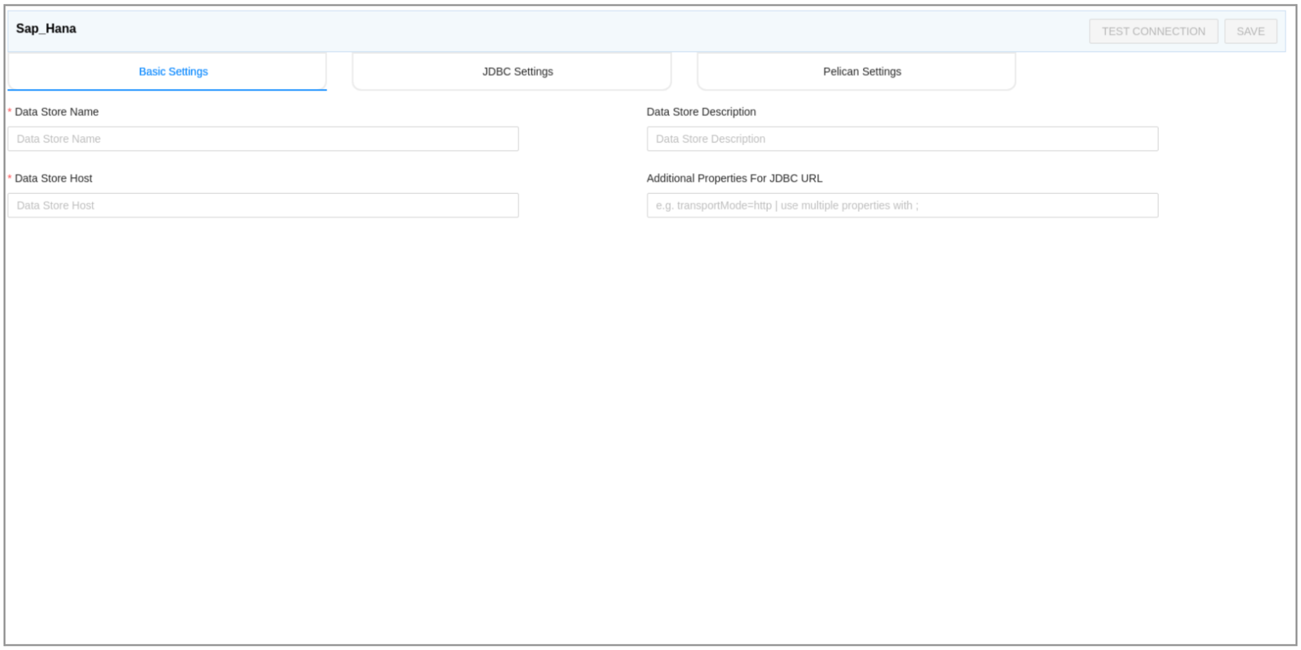
   Note: The fields marked with an asterisk (*) are mandatory.

8.1.15. Steps for SAP HANA Configuration

1. Go to **Administration > Configure > Datastore**.

   At the left-hand side pane a list of predefined datastore lists is visible.
2. Select the **SAP HANA** datastore.

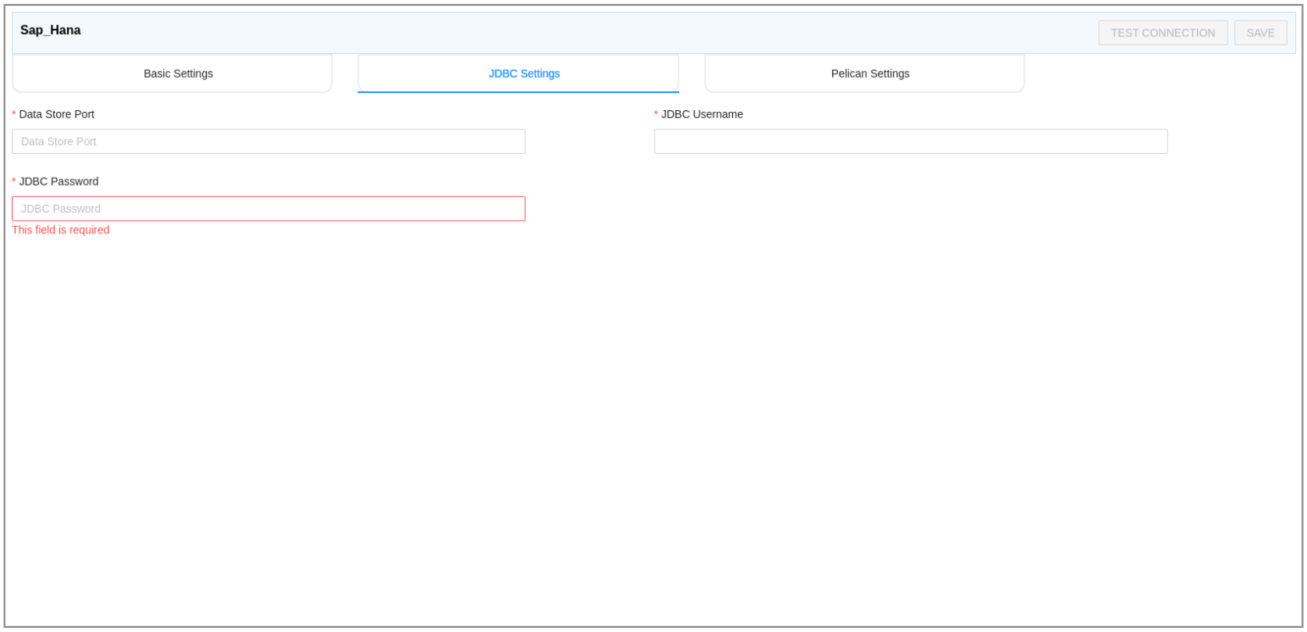
   A screen to enter the corresponding datastore details will be visible.



Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.

Data Store Host	IP address and host name of the data store host machine.
Additional Properties for JDBC URL	Additional Properties required for the JDBC connection.

JDBC Settings



* Data Store Port

Data Store Port

* JDBC Username

* JDBC Password

JDBC Password

This field is required

Field	Description
Data Store Port	JDBC port to connect to the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.
JDBC Password	JDBC password for the provided JDBC user.

Pelican Settings

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Supplementary Datastore Location	Writable Path on the machine where Pelican is installed.

3. Enter all the details in the **Basic Settings, JDBC Settings** and **Pelican Settings** tab fields.
4. Click **Test Connection**.

   A success message will be displayed in case of a successful connection.

5. Click **Save**.

   The newly created datastore will be displayed in the datastore list.

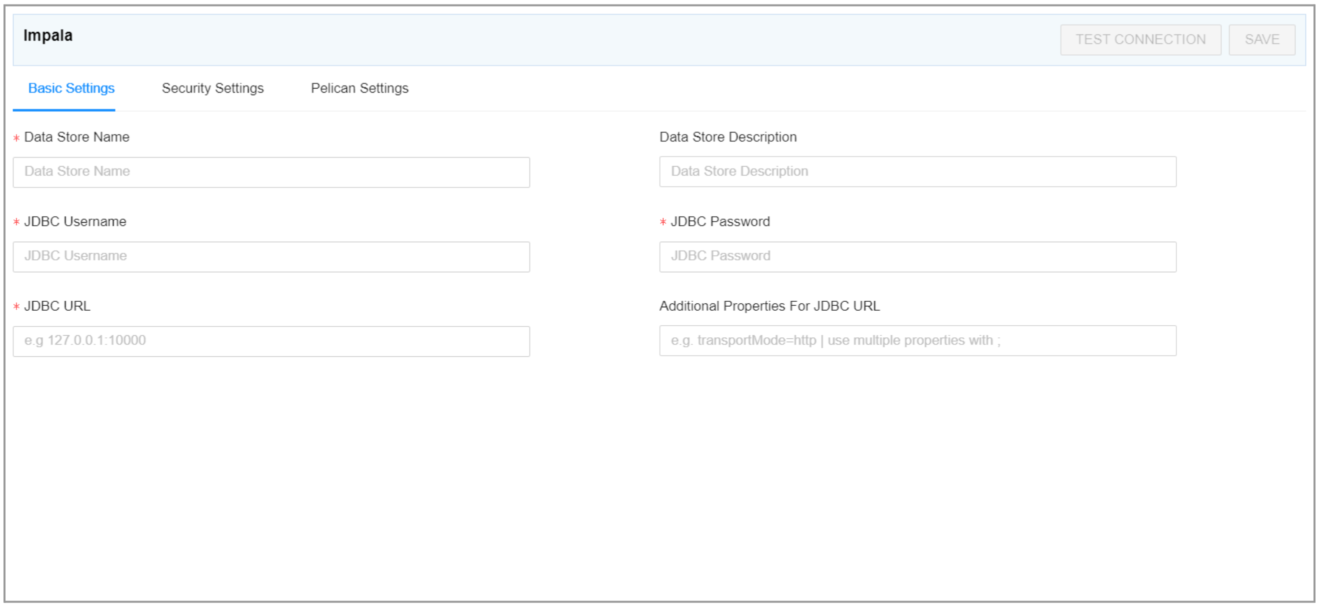
   Note: The fields marked with an asterisk (*) are mandatory.

8.1.16. Steps for Impala Configuration

1. Go to **Administration > Configure > Datastore**.

   A list of predefined datastores is visible at the left-hand side pane.

2. Select **Impala** data store field. A screen to enter the corresponding datastore details will be visible.



* Data Store Name: Data Store Name

Data Store Description: Data Store Description

* JDBC Username: JDBC Username

* JDBC Password: JDBC Password

* JDBC URL: e.g 127.0.0.1:10000

Additional Properties For JDBC URL: e.g. transportMode=http | use multiple properties with ;

3. Enter all the details in the **Basic Settings, Security Settings** and **Pelican Settings** fields.

4. Click **TEST CONNECTION**.

   A success message will be displayed in case of a successful connection.

5. Click **SAVE**.

   The newly created datastore will be displayed in the datastore list.

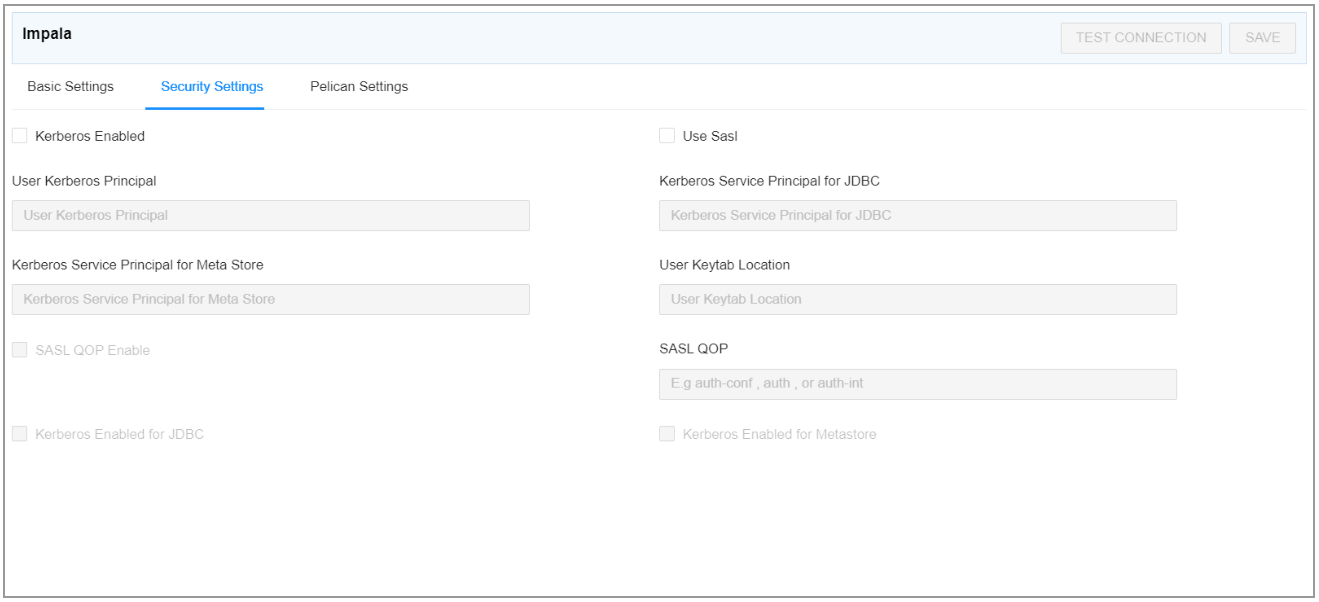
   Note: The fields marked with an asterisk (*) are mandatory.

Basic Settings Field Descriptions

Field	Description
Data Store Name	The name the user wants to give to the data store.
Data Store Description	Description for the data store.
JDBC Username	JDBC username using which Pelican will connect to the data store.

JDBC Password	JDBC password for the provided JDBC user.
JDBC URL	Hive host URL for JDBC connection.
Additional Properties For JDBC URL	Additional Properties For JDBC URL

Security Settings Field Description



□ Kerberos Enabled

□ Use Sasl

User Kerberos Principal: User Kerberos Principal

Kerberos Service Principal for JDBC: Kerberos Service Principal for JDBC

Kerberos Service Principal for Meta Store: Kerberos Service Principal for Meta Store

User Keytab Location: User Keytab Location

□ SASL QOP Enable

SASL QOP: E.g auth-conf , auth , or auth-int

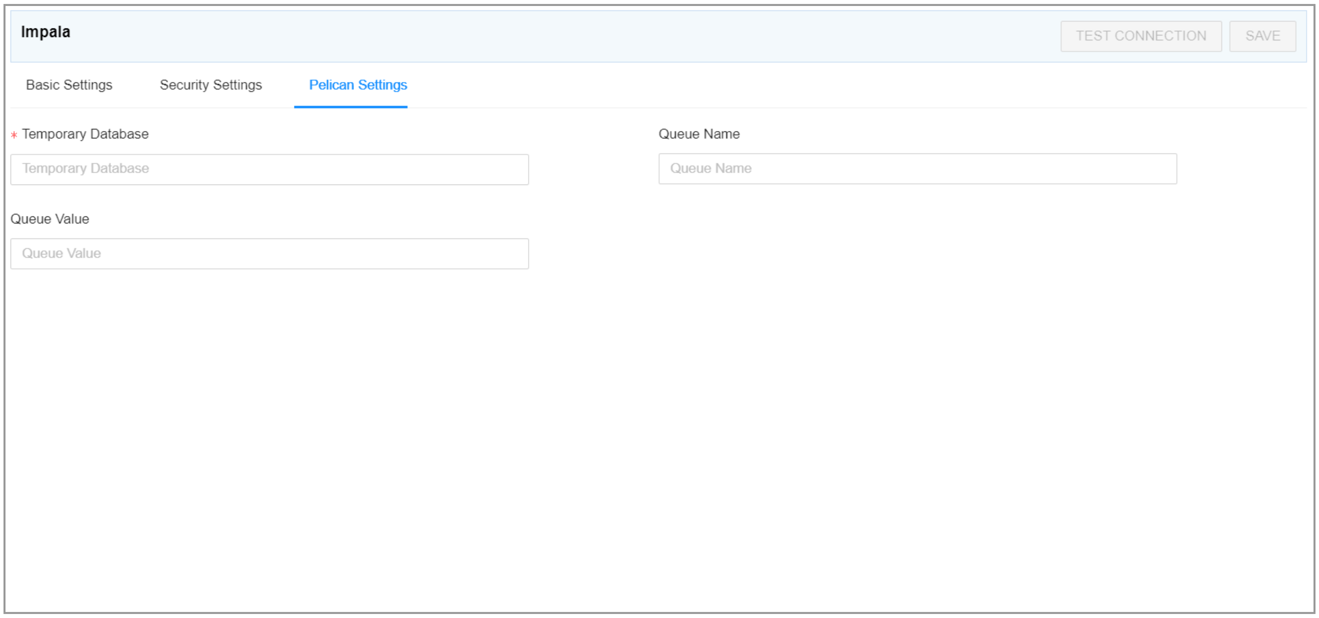
□ Kerberos Enabled for JDBC

□ Kerberos Enabled for Metastore

Field	Description
Kerberos Enabled	Check for kerberized Impala.
Use Sasl	Check box based if SASL is required
User Kerberos Principal	Pattern will be like user@organization.com.
Kerberos Service Principal for JDBC	A Kerberos principal is a unique identity to which Kerberos can assign tickets.
Kerberos Service Principal for Meta Store	HDFS meta store path
User Keytab Location	User keytab file location in case of Kerberos Enable.
SASL QOP Enable	Checkbox to enable SASL QOP
SASL QOP	SASL Mechanisms
Kerberos Enabled for JDBC	Can be checked if Kerberos Enable is true.

Kerberos Enabled for Metastore	Can be checked if Kerberos Enable is true.

Pelican Settings Field Description



Impala

* Temporary Database: Temporary Database

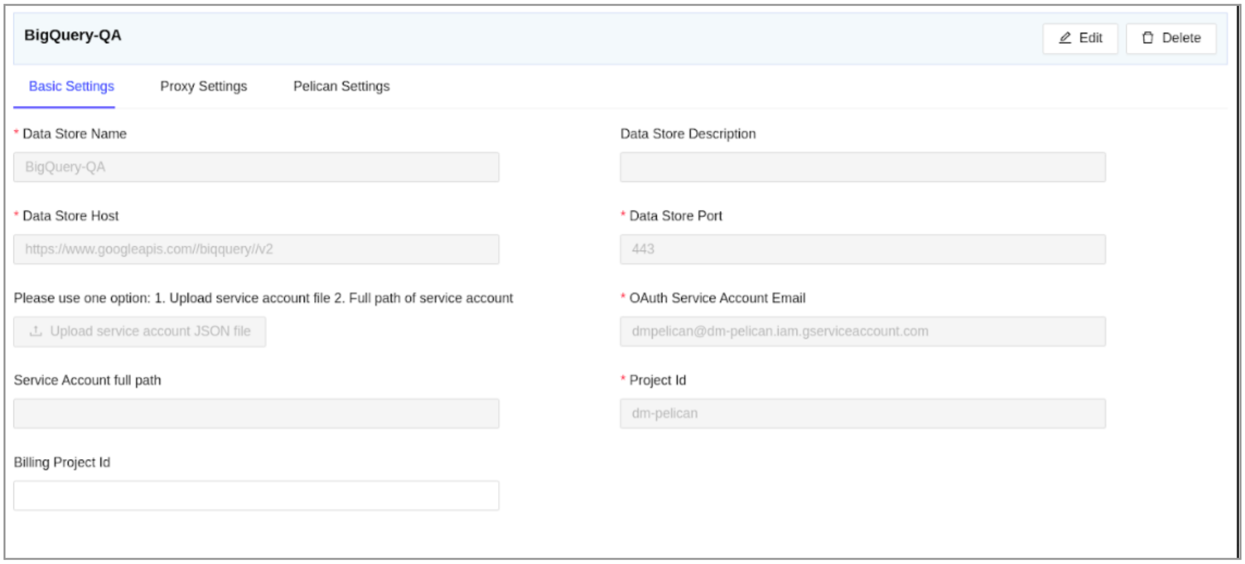
Queue Name: Queue Name

Queue Value: Queue Value

Field	Description
Temporary Database	Database used by Pelican for its functioning.
Queue Name	Describes the name of the queue.
Queue Value	Describes the queue value.

8.2. Editing an existing Datastore

1. Go to **Administration** → **Configure** → **Datastores**.
2. Select the Datastore you want to edit.
3. Click on the **Edit** icon.



BigQuery-QA

* Data Store Name: BigQuery-QA

Data Store Description:

* Data Store Host: https://www.googleapis.com//bigquery//v2

* Data Store Port: 443

Please use one option: 1. Upload service account file 2. Full path of service account

* OAuth Service Account Email: dmpelican@dm-pelican.iam.gserviceaccount.com

Service Account full path:

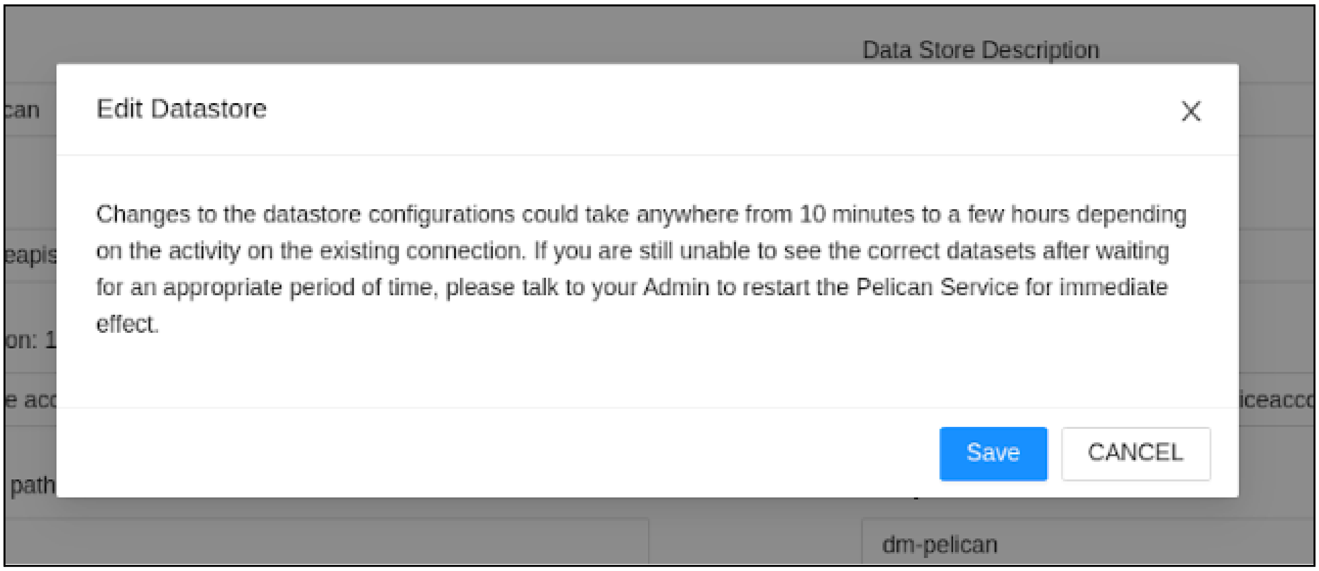
* Project Id: dm-pelican

Billing Project Id:

4. Edit the information of the desired field, upload the service_account.json file.
5. Click on the **Test Connection button**.

   If the connection is correct then a Success message will pop up. Once the connection is verified, the **Save** button will be enabled along with the below pop-up window.

   The pop-up informs the user that the changes made in the datastore configurations can take time ranging from a few minutes to some hours depending on the activity on the existing connection. If the user is unable to see the correct datasets after some time, please contact the Admin to restart the Pelican Service for immediate effect.



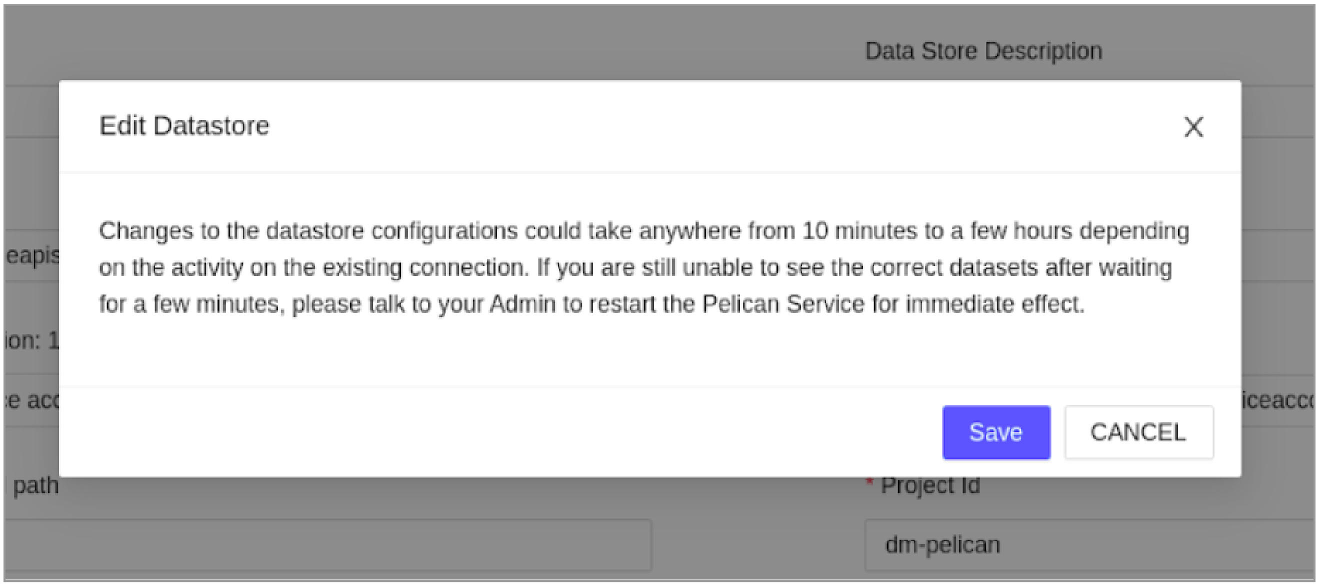
Edit Datastore

Changes to the datastore configurations could take anywhere from 10 minutes to a few hours depending on the activity on the existing connection. If you are still unable to see the correct datasets after waiting for an appropriate period of time, please talk to your Admin to restart the Pelican Service for immediate effect.

6. Click **Save**.

Note:

- The following pop-up window displaying the time period of the Edit datastore process will be displayed while saving the edited datastore.
- Click **Save** to save the datastore.

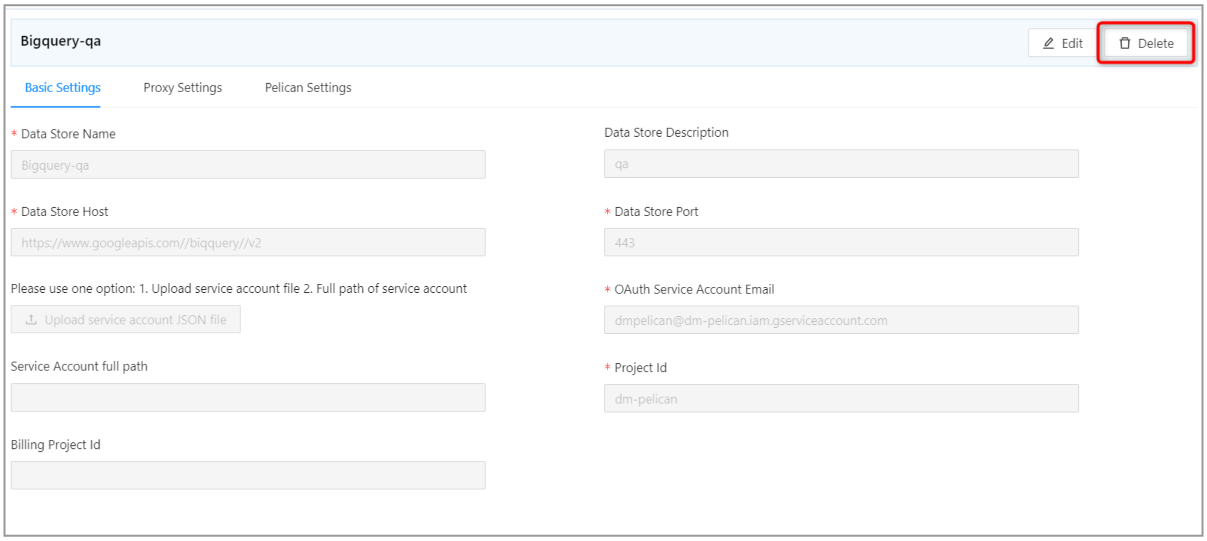


Edit Datastore

Changes to the datastore configurations could take anywhere from 10 minutes to a few hours depending on the activity on the existing connection. If you are still unable to see the correct datasets after waiting for a few minutes, please talk to your Admin to restart the Pelican Service for immediate effect.

8.3. Deleting existing datastore

Select the data store you want to delete and click the Delete Datastore icon. The system displays a confirmation dialog box to ensure the deletion of the data store so click DELETE.



Bigquery-qa

* Data Store Name: Bigquery-qa

Data Store Description: qa

* Data Store Host: https://www.googleapis.com//biqquery//v2

* Data Store Port: 443

Please use one option: 1. Upload service account file 2. Full path of service account

* OAuth Service Account Email: dmpelican@dm-pelican.iam.gserviceaccount.com

Service Account full path

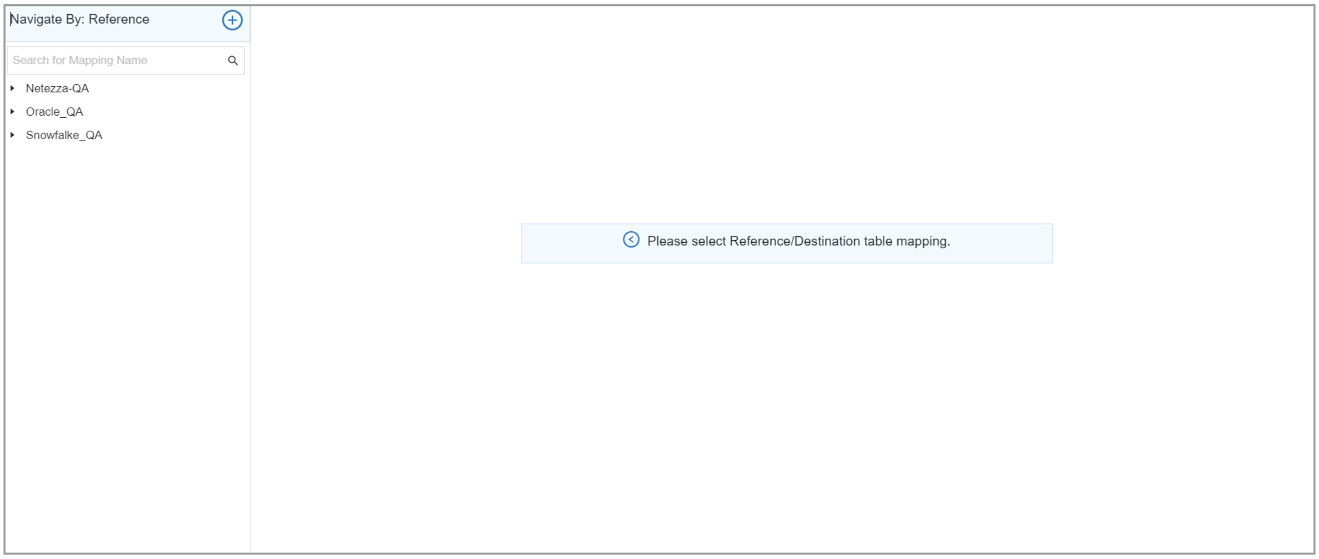
* Project Id: dm-pelican

Billing Project Id

Note:

If a mapping(s) is present under any datastore, then the datastore cannot be deleted until we delete the associated mappings first.

9. Validation Configuration



Navigate By: Reference

Please select Reference/Destination table mapping.

After the source and destination data store is created, the next step is validation configuration. Validation configuration allows you to select tables (source and destination) from the respective data stores and map them.

Validation configuration searches the table with similar column name and data type as in the source Datastore. The application provides various filters to search the desired data store, reference database, destination database, reference schema and destination schema. Additionally, Pelican facilitates the following approximate mapping methods which allows you to map the tables based on their names and patterns:

- **Phonetic Matching**: A phonetic matching is an algorithm for **matching words by their pronounciation**.

  For example, the words Principal and Principle are phonetically matching words. Which means, table names (source and destination) Principal and Principle will be considered for mapping.

- **Approximate Matching**: Approximate matching is based on Levenshtein distance. It is a metric for measuring the difference between two words.

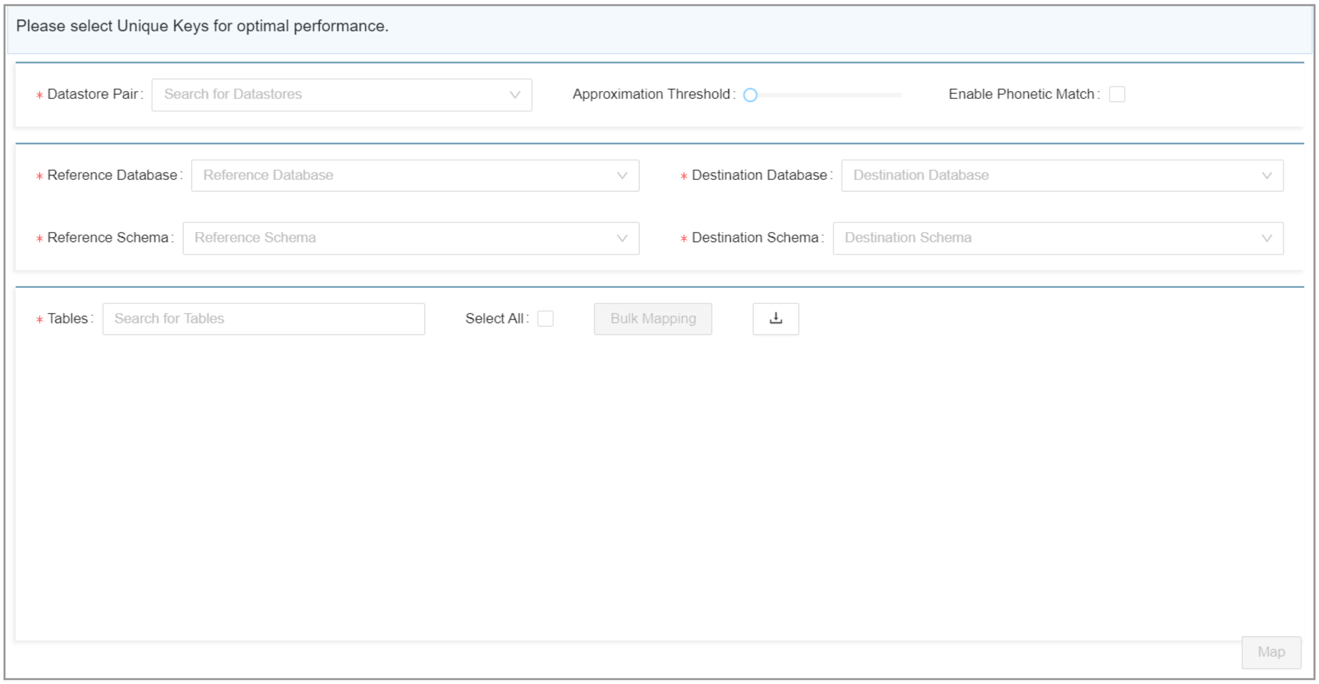
  For Example. In the word ‘Employe’ , insertion of "e" at the end → distance is 1, while in ‘Employee’ (insertion of "ee" at the end) distance is 2.

The above-mentioned table pair (source and destination) is considered for the mapping as the Levenshtein distance between these table pairs is either 1 or 2.

9.1. Create Table Mapping

To validate tables between source and destination data store.

1. Go to **Govern Validation Configuration**.



Please select Unique Keys for optimal performance.

* Datastore Pair: Search for Datastores ⌄ Approximation Threshold: ○ Enable Phonetic Match: □

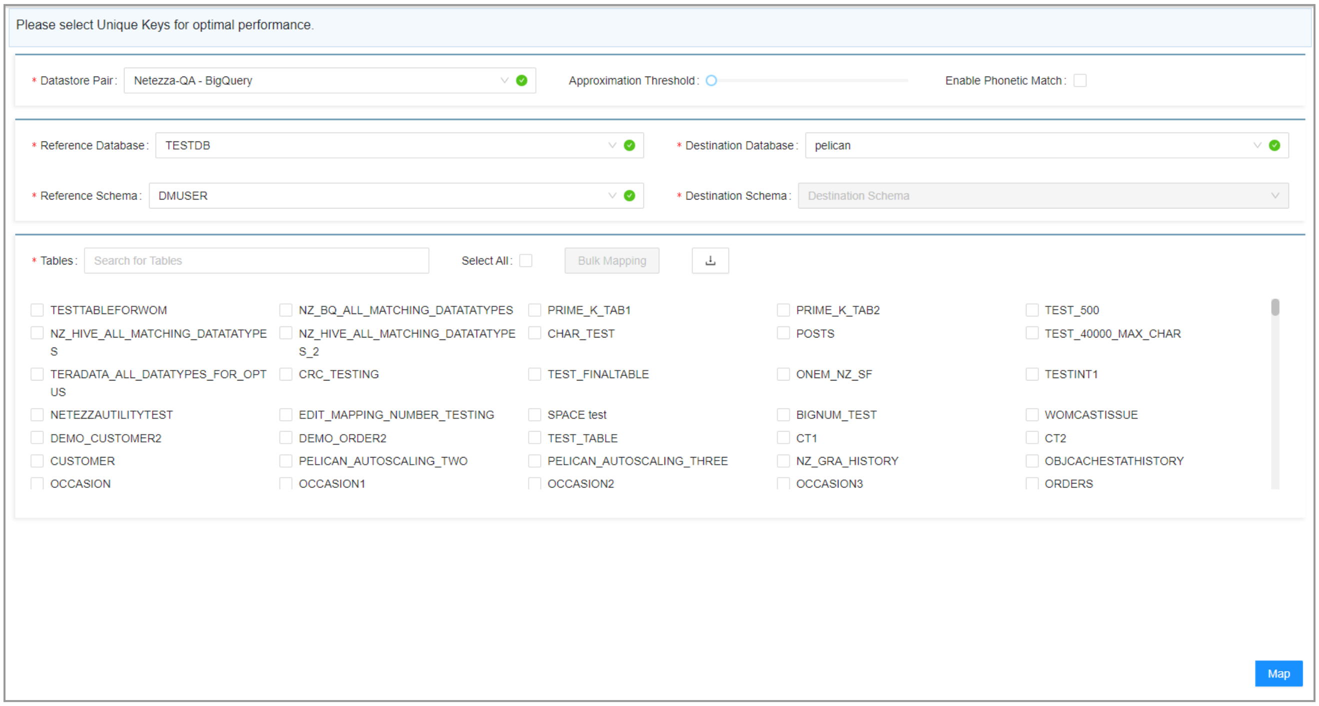
* Reference Database: Reference Database ⌄ * Destination Database: Destination Database ⌄

* Reference Schema: Reference Schema ⌄ * Destination Schema: Destination Schema ⌄

* Tables: Search for Tables Select All: □ Bulk Mapping ⤓

Map

2. Click on the create new mapping icon. ⊕



Please select Unique Keys for optimal performance.

* Datastore Pair : Netezza-QA - BigQuery

Approximation Threshold : ○

Enable Phonetic Match : □

* Reference Database : TESTDB

* Destination Database : pelican

* Reference Schema : DMUSER

* Destination Schema : Destination Schema

* Tables : Search for Tables

Select All : □

□ TESTTABLEFORWOM
□ NZ_HIVE_ALL_MATCHING_DATATYPES
□ TERADATA_ALL_DATATYPES_FOR_OPTUS
□ NETEZZAUTILITYTEST
□ DEMO_CUSTOMER2
□ CUSTOMER
□ OCCASION
□ NZ_BQ_ALL_MATCHING_DATATYPES
□ NZ_HIVE_ALL_MATCHING_DATATYPES_2
□ CRC_TESTING
□ EDIT_MAPPING_NUMBER_TESTING
□ DEMO_ORDER2
□ PELICAN_AUTOSCALING_TWO
□ OCCASION1
□ PRIME_K_TAB1
□ CHAR_TEST
□ TEST_FINALTABLE
□ SPACE test
□ TEST_TABLE
□ PELICAN_AUTOSCALING_THREE
□ OCCASION2
□ PRIME_K_TAB2
□ POSTS
□ ONEM_NZ_SF
□ BIGNUM_TEST
□ CT1
□ NZ_GRA_HISTORY
□ OCCASION3
□ TEST_500
□ TEST_40000_MAX_CHAR
□ TESTINT1
□ WOMCASTISSUE
□ CT2
□ OBJCACHESTATHISTORY
□ ORDERS

3. Select the following information:
   A. **Datastore Pair**: Displays all the combinations of source and target data stores configured in the system.
   B. **Reference Database:** Displays all the databases of the corresponding source data store.
   C. **Reference Schema**: This field will be enabled only if the source datastore supports three level hierarchy. It displays all the schemas present in the database selected in the **Reference Database**.
   D. **Destination Database:** Displays all the databases of the corresponding destination data store.
   E. **Destination Schema**: This field will be enabled only if the target datastore supports a three level hierarchy*. Displays all the schemas present in the database selected in the **Destination Database**.
   F. **Select** tables in the table section.

This section allows the user to select the tables that exist in the selected reference database and schema. While mapping, the application searches the same tables in the destination database and schema. Further it also searches similar column names and data types as in source datastore. You can use the approximation matching parameters to search tables, columns, and data types of similar patterns as in the source data store.

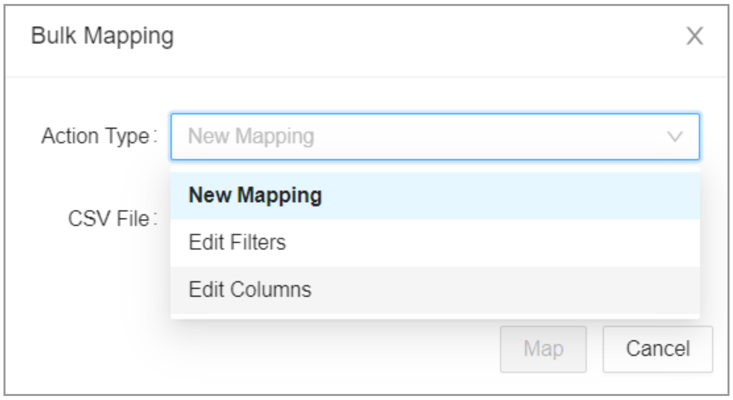
Optional Steps:

A. Select the **Select All** checkbox if required. This functionality enables the user to map all the tables at the same time. You can also upload the table names with the help of a CSV file..

B. **Bulk Mapping** : The Bulk mapping feature enables the user to perform the following operations:

- Create new mappings
- Add/Edit Filter
- Add/Delete/Edit Columns

The user can also apply expressions in the columns with the help of Bulk mapping feature. However, only one operation can be performed at a time.



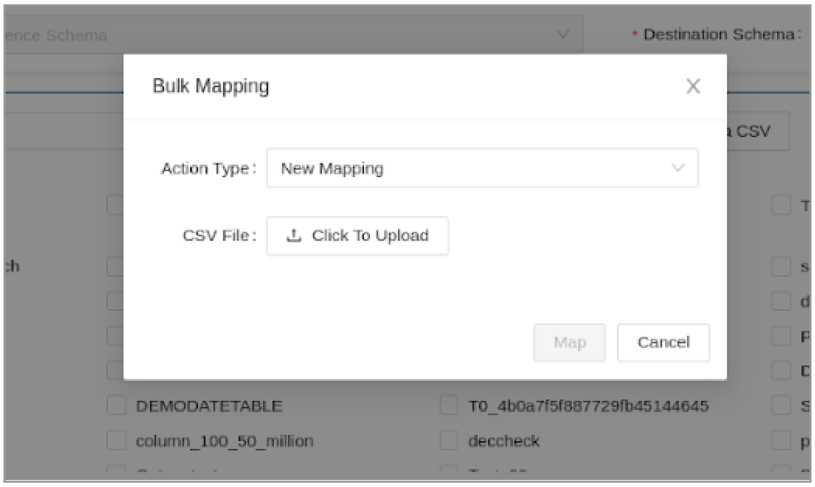
Bulk Mapping

Action Type: New Mapping

CSV File:

Create New Mapping

On selecting this option the following pop-up will be opened.



Bulk Mapping

Action Type: New Mapping

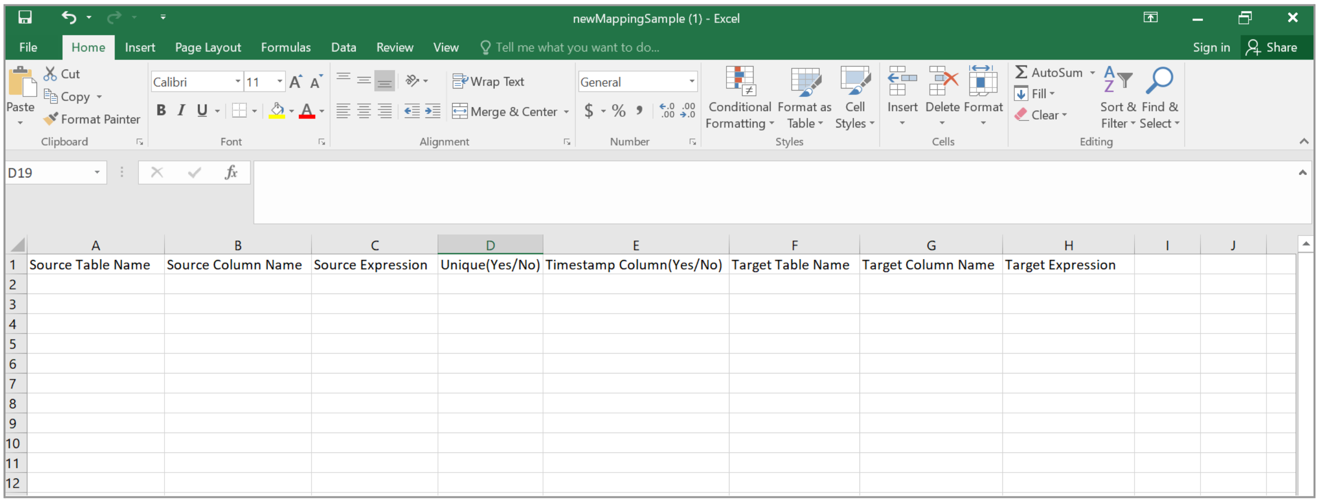
CSV File: Click To Upload

Enter details into the appropriate CSV file and upload it by clicking on the '**Click to upload**' button.

The CSV files can be downloaded by clicking on the Download button.

Note: The order and naming convention should be used as mentioned here. Please ensure that no garbage characters or spaces are present in the header. This applies for all the three types of CSV files.

Field Descriptions:



	A	B	C	D	E	F	G	H
1	Source Table Name	Source Column Name	Source Expression	Unique(Yes/No)	Timestamp Column(Yes/No)	Target Table Name	Target Column Name	Target Expression

Source Table Name: Name of the Source table.

Source Column Name: Name of the Source Column.

Source Expression: If it is not present then Pelican will generate one. If it is present the same one will be used.

Unique(Yes/No): We can assign a column as unique only if it is actually unique in the DB.

Timestamp: Only one column can be a timestamp column.

Target Table Name: Name of the target table.

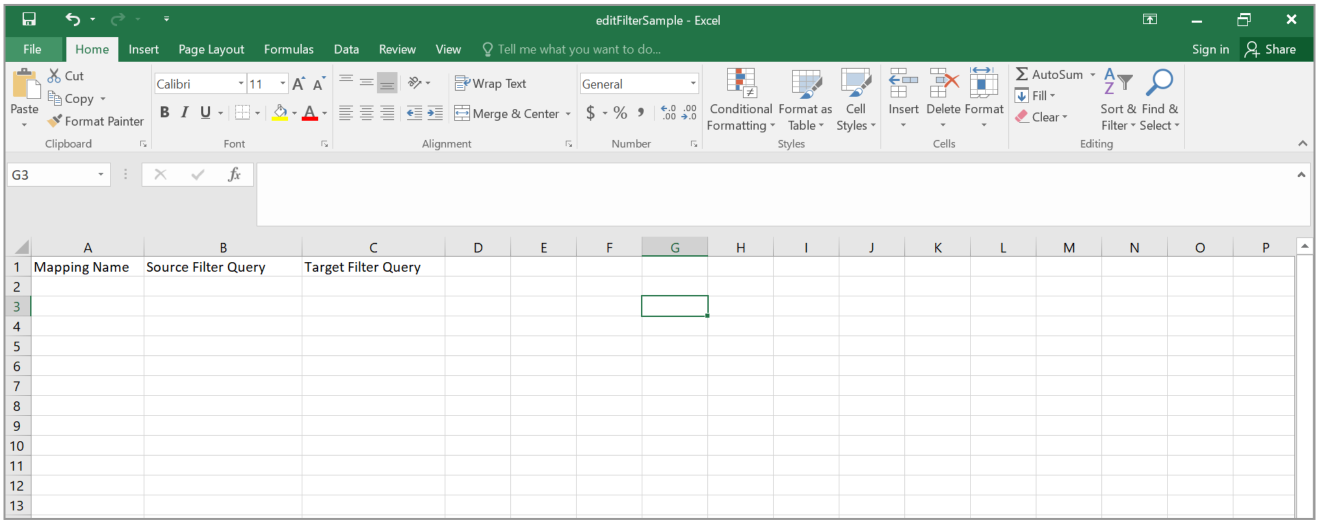
Target Column Name: Name of the target column.

Target Expression: If it is not present then pelican will generate one. If it is present the same one will be used.

Note:

- The **Source Table Name** field is mandatory.
- If the **Target Table Name** and **Target Column Names** are not mentioned, then in that case their values will be considered the same as **Source Table Name** and the **Source Column Names** respectively.
- If a user has entered only a few columns from the table in the CSV file, then in that case only the columns mentioned by the user will get mapped. However, if no column has been mentioned by the user then in that case all the columns will be mapped.
- If a user has given different column names, but the table name is the same in source and target, then in that case, whatever columns are mentioned manually by the user in the CSV file will get mapped. Similarly, upload the following CSV files for **Edit Filter** and **Edit Column** options as well.

Edit Filter



	A	B	C
1	Mapping Name	Source Filter Query	Target Filter Query

Field Descriptions:

Mapping Name: Name of the mapping.

Source Filter Query: Filter to be applied on the Source side.

Target Filter Query: Filter to be applied on the Target side.

Note:

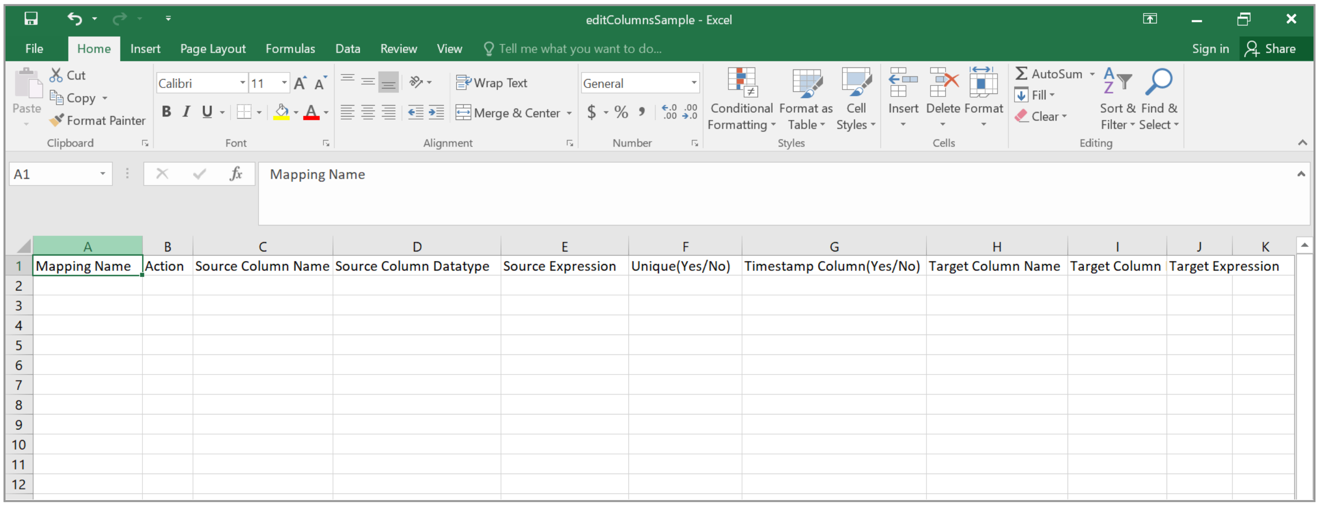
- The **Mapping Name** field is mandatory.
- If the **Source Filter Query** and **Target Filter Query** fields are blank then their filters will be cleared.

Edit Column

The **Edit Column** option enables the user to perform the following modification actions at the same time:

- **Add** - To Add a column, the Source Column Name, Source Column Datatype, Target Column Name and Target Column Datatype fields are mandatory.
- **Edit** - To Edit a column the Source and Target Column Names are mandatory. However, the user can edit only the Source Expression, Target Expression, Uniqueness and Timestamp columns.
- **Delete** - To Delete a column the Source Column Name and Target Column Name fields are mandatory.

Field Descriptions:



	A	B	C	D	E	F	G	H	I	J	K
1	Mapping Name	Action	Source Column Name	Source Column Datatype	Source Expression	Unique(Yes/No)	Timestamp Column(Yes/No)	Target Column Name	Target Column	Target Expression	

Mapping Name: Name of the mapping

Action: Add, Edit or Delete option can be performed

Source Column Name: Name of the source column

Source Column Datatype: Datatype of the source column

Source Expression: Source expression

Unique: Denotes if it is a unique column

Timestamp column: Timestamp column

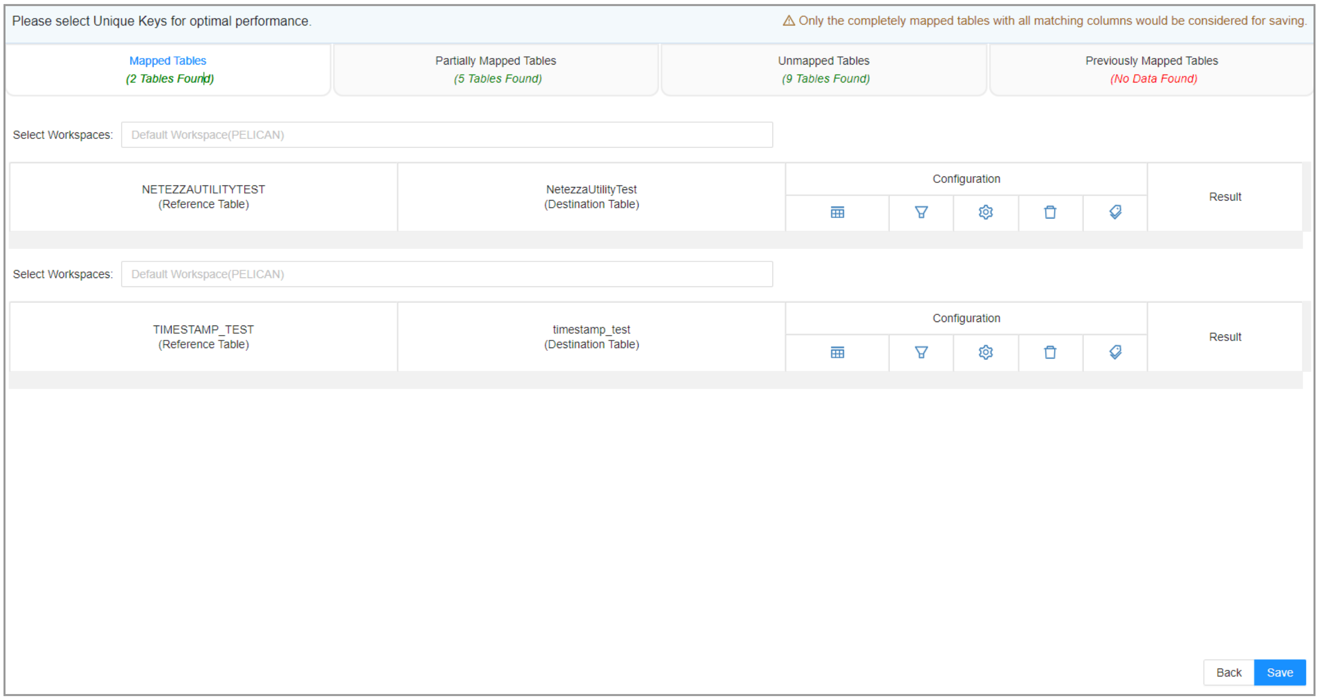
Target Column Name: Name of the target column

Target Column Datatype: Datatype of the target column

Target Expression: Target Expression.

4. Click **Map**.

   After you click on **Map**, the application displays the mapping result in four sections as shown below.



Please select Unique Keys for optimal performance.

⚠ Only the completely mapped tables with all matching columns would be considered for saving.

Select Workspaces: Default Workspace(PELICAN)

NETEZZAUTILITYTEST (Reference Table)	NetezzaUtilityTest (Destination Table)	Configuration	Result

Select Workspaces: Default Workspace(PELICAN)

TIMESTAMP_TEST (Reference Table)	timestamp_test (Destination Table)	Configuration	Result

a. **Mapped Tables**: This tab contains the table where the table name and each column of the source table matches with that of the target table.

b. **Partially Mapped Tables**: This tab contains the table/s where few columns of a source table/s match with the target table/s.

c. **Unmapped Tables**: This tab contains the table/s from the source table which are not present in the target table.

d. **Previously Mapped Tables**: This tab shows the previously mapped tables.

5. Click the respective tab (section) to see the results.

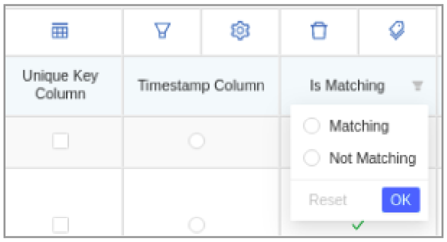
   Note:

   - The tables which are not completely mapped from the **Partially Mapped** and **Unmapped** tab will not be saved.
   - The tables from the **Previously mapped** tab will not be considered for saving again.
   - The user can view only the matching/non-matching columns by setting the **Is Matching** filter accordingly.

6. The user can perform following operations on the result page:
   - Show Column
   - Edit Configuration
     - Edit Column
     - Edit Expression

- Edit Data Type
- Add New Columns

- Is Matching - Displays only the matching/non-matching columns.

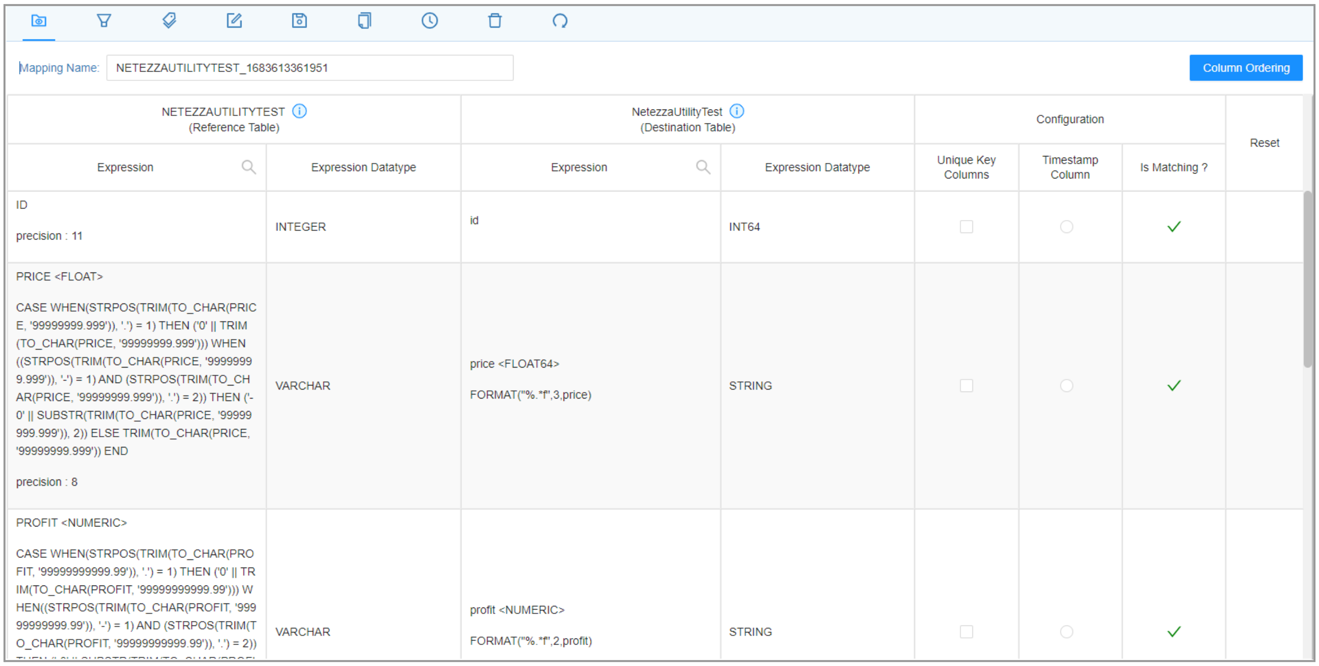


- Delete table and column
- Override Unmapped Columns
- View only the matching or not-matching columns.

Note:

- The column which is a Primary key in the table is selected as a unique key by default. However, if there is no primary key in the table then you will have to select a unique key.
- Only the completely mapped tables with all matching columns would be considered for saving.

7. Click **Save** to save the mapping.

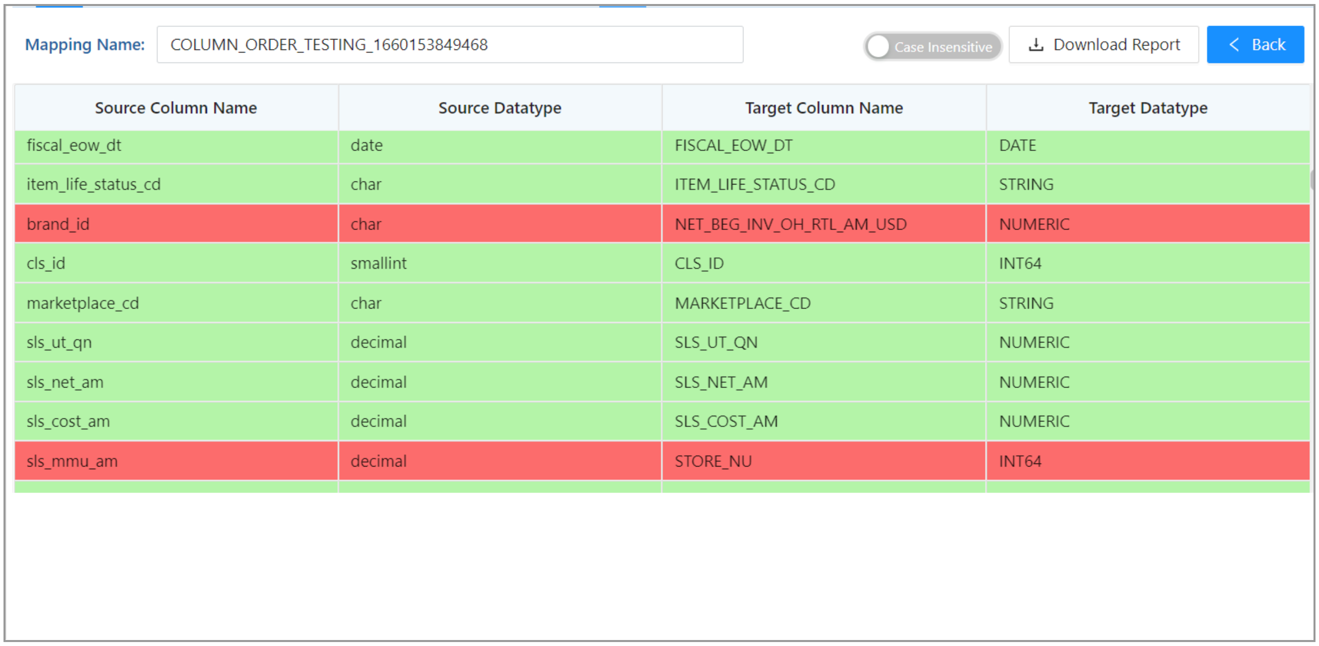


Mapping Name: NETEZZAUTILITYTEST_1683613361951

NETEZZAUTILITYTEST (Reference Table)		NetezzaUtilityTest (Destination Table)		Configuration			Reset
Expression	Expression Datatype	Expression	Expression Datatype	Unique Key Columns	Timestamp Column	Is Matching ?	
ID precision : 11	INTEGER	id	INT64	□	○	✓	
PRICE <FLOAT> CASE WHEN(STRPOS(TRIM(TO_CHAR(PRICE, '99999999.999')), '.') = 1) THEN ('0' || TRIM(TO_CHAR(PRICE, '99999999.999'))) WHEN((STRPOS(TRIM(TO_CHAR(PRICE, '99999999.999')), '-') = 1) AND (STRPOS(TRIM(TO_CHAR(PRICE, '99999999.999')), '.') = 2)) THEN ('-0' || SUBSTR(TRIM(TO_CHAR(PRICE, '99999999.999')), 2)) ELSE TRIM(TO_CHAR(PRICE, '99999999.999')) END precision : 8	VARCHAR	price <FLOAT64> FORMAT("%.*f",3,price)	STRING	□	○	✓	
PROFIT <NUMERIC> CASE WHEN(STRPOS(TRIM(TO_CHAR(PROFIT, '99999999999.99')), '.') = 1) THEN ('0' || TRIM(TO_CHAR(PROFIT, '99999999999.99'))) WHEN((STRPOS(TRIM(TO_CHAR(PROFIT, '99999999999.99')), '-') = 1) AND (STRPOS(TRIM(TO_CHAR(PROFIT, '99999999999.99')), '.') = 2))	VARCHAR	profit <NUMERIC> FORMAT("%.*f",2,profit)	STRING	□	○	✓	

9.2. Column Ordering

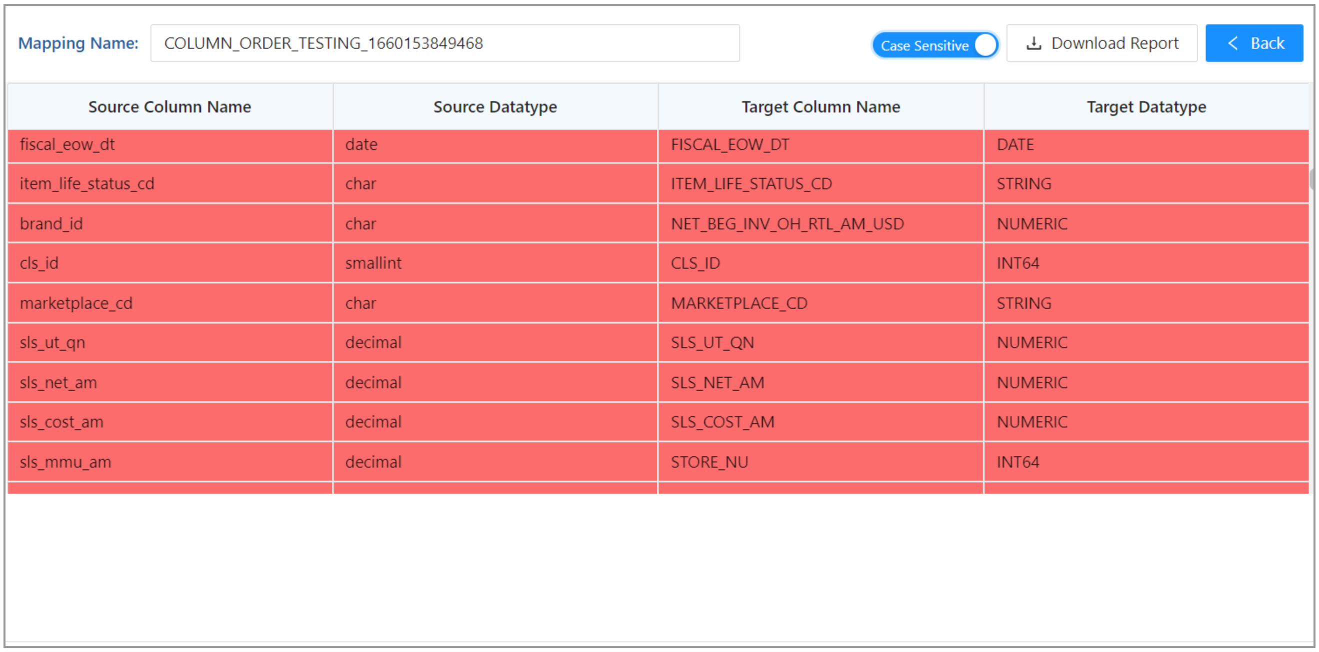
- The **Column Ordering** button displays if the columns from the Source and Target have the same order or not. If the order of the column is similar in both the source and target they will be marked by Green color, while if the columns do not match they will be highlighted in Red.
- **Case Insensitive:** Enabling this toggle button will make the column names case sensitive while matching, while disabling it will match column names irrespective of their case.



Mapping Name: COLUMN_ORDER_TESTING_1660153849468

Source Column Name	Source Datatype	Target Column Name	Target Datatype
fiscal_eow_dt	date	FISCAL_EOW_DT	DATE
item_life_status_cd	char	ITEM_LIFE_STATUS_CD	STRING
brand_id	char	NET_BEG_INV_OH_RTL_AM_USD	NUMERIC
cls_id	smallint	CLS_ID	INT64
marketplace_cd	char	MARKETPLACE_CD	STRING
sls_ut_qn	decimal	SLS_UT_QN	NUMERIC
sls_net_am	decimal	SLS_NET_AM	NUMERIC
sls_cost_am	decimal	SLS_COST_AM	NUMERIC
sls_mmu_am	decimal	STORE_NU	INT64

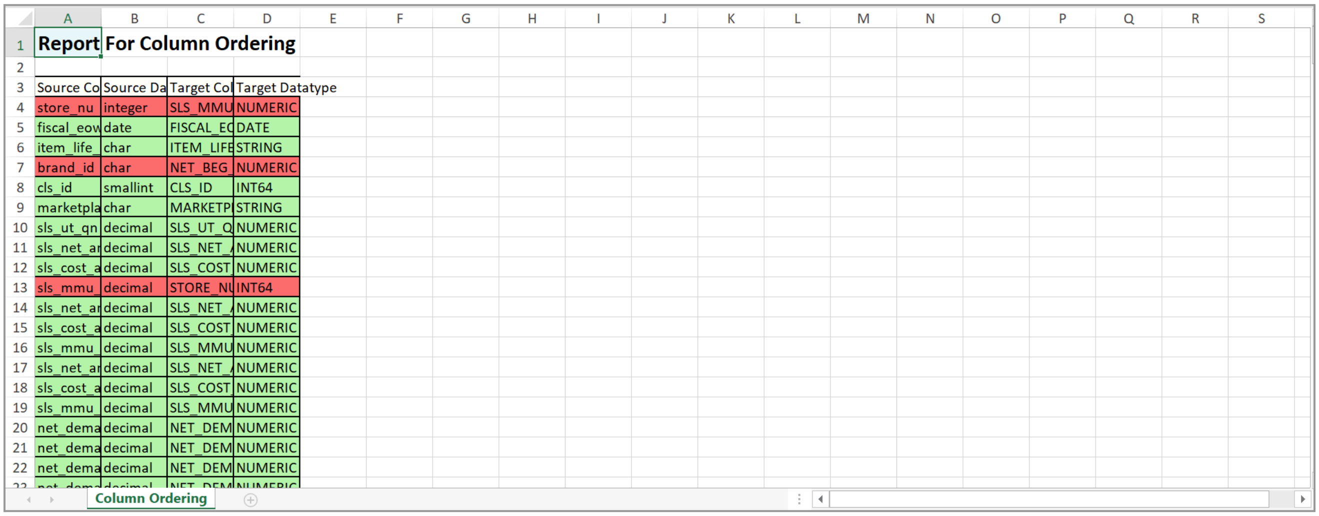
After enabling **Case Sensitive** toggle.



COLUMN_ORDER_TESTING_1660153849468

Source Column Name	Source Datatype	Target Column Name	Target Datatype
fiscal_eow_dt	date	FISCAL_EOW_DT	DATE
item_life_status_cd	char	ITEM_LIFE_STATUS_CD	STRING
brand_id	char	NET_BEG_INV_OH_RTL_AM_USD	NUMERIC
cls_id	smallint	CLS_ID	INT64
marketplace_cd	char	MARKETPLACE_CD	STRING
sls_ut_qn	decimal	SLS_UT_QN	NUMERIC
sls_net_am	decimal	SLS_NET_AM	NUMERIC
sls_cost_am	decimal	SLS_COST_AM	NUMERIC
sls_mmu_am	decimal	STORE_NU	INT64

Report Download: This button downloads the matching result as it is visible on the screen in .xls format.



Report For Column Ordering

Source Co	Source Da	Target Col	Target Datatype
store_nu	integer	SLS_MMU	NUMERIC
fiscal_eow	date	FISCAL_EC	DATE
item_life_	char	ITEM_LIFE	STRING
brand_id	char	NET_BEG_	NUMERIC
cls_id	smallint	CLS_ID	INT64
marketpla	char	MARKETPI	STRING
sls_ut_qn	decimal	SLS_UT_Q	NUMERIC
sls_net_ar	decimal	SLS_NET_	NUMERIC
sls_cost_a	decimal	SLS_COST	NUMERIC
sls_mmu_	decimal	STORE_NU	INT64
sls_net_ar	decimal	SLS_NET_	NUMERIC
sls_cost_a	decimal	SLS_COST	NUMERIC
sls_mmu_	decimal	SLS_MMU	NUMERIC
sls_net_ar	decimal	SLS_NET_	NUMERIC
sls_cost_a	decimal	SLS_COST	NUMERIC
sls_mmu_	decimal	SLS_MMU	NUMERIC
net_dema	decimal	NET_DEM	NUMERIC
net_dema	decimal	NET_DEM	NUMERIC
net_dema	decimal	NET_DEM	NUMERIC

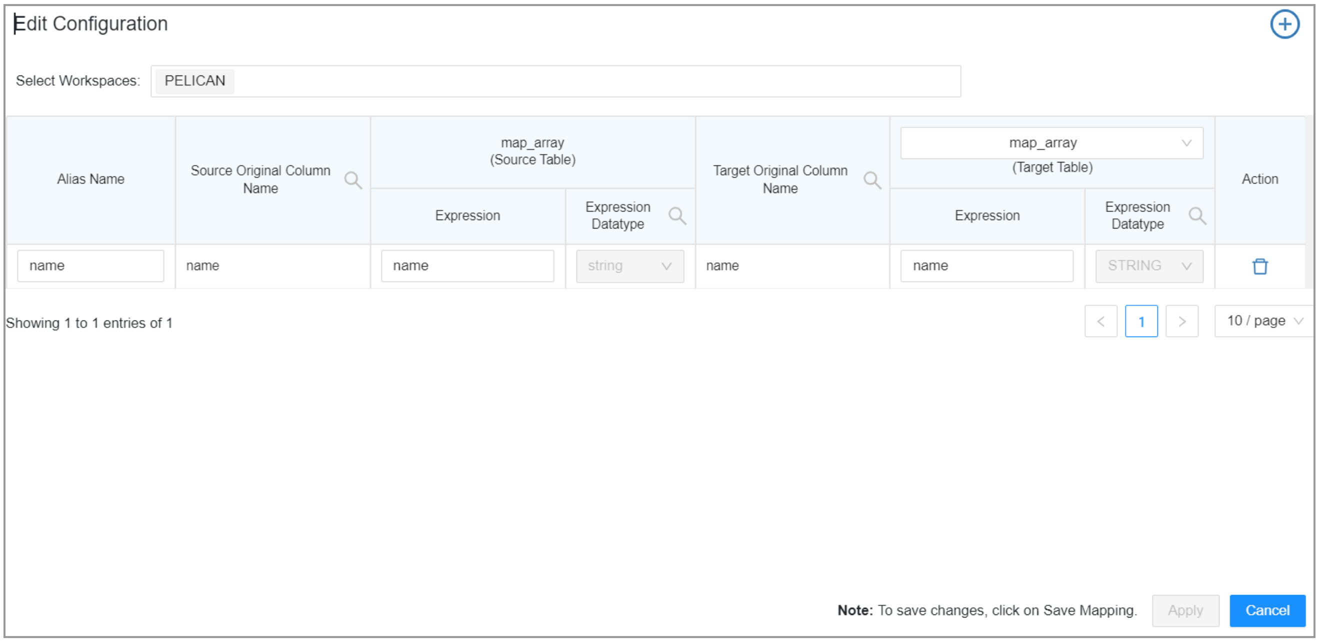
9.3. Edit the existing configuration

This functionality allows you to edit existing columns, expression, data type, and add/delete new columns as per the business requirement.

The steps to edit the existing configuration are as follows:

1. Click the **Edit Mapping** icon on the screen.

   The application displays the **Edit Configuration** window as shown below.



Edit Configuration

Select Workspaces: PELICAN

Alias Name	Source Original Column Name	map_array (Source Table)		Target Original Column Name	map_array (Target Table)		Action
		Expression	Expression Datatype		Expression	Expression Datatype	
name	name	name	string	name	name	STRING	🗑

Showing 1 to 1 entries of 1

Note: To save changes, click on Save Mapping.

In the **Edit Configuration** window, you can edit existing columns, existing expressions, existing data type and add new columns.

- **Edit existing column name**

  Pelican allows the user to edit the existing column name (Alias Name) to make it uniform on both sides (Reference and Destination).

  To edit existing column name:

  - Before saving the mapping enter the required column name in the **Expression** field.
  - If the mapping already exists, the user will have to delete the existing column and add a new one with the required column name with the required source and target column..

  The changes made in the **Edit Configuration** screen will be saved only in Pelican UI and not in the original database.

- **Edit existing expression**

  After mapping, the expressions are automatically created. The expressions are created based on the datatype of the respective column. However, the user can edit these existing expressions as per the requirement.

  For example,

  To edit the existing expression:

  Click in the desired column and edit the expression as per your requirement and then click **OK**.

Edit existing data type

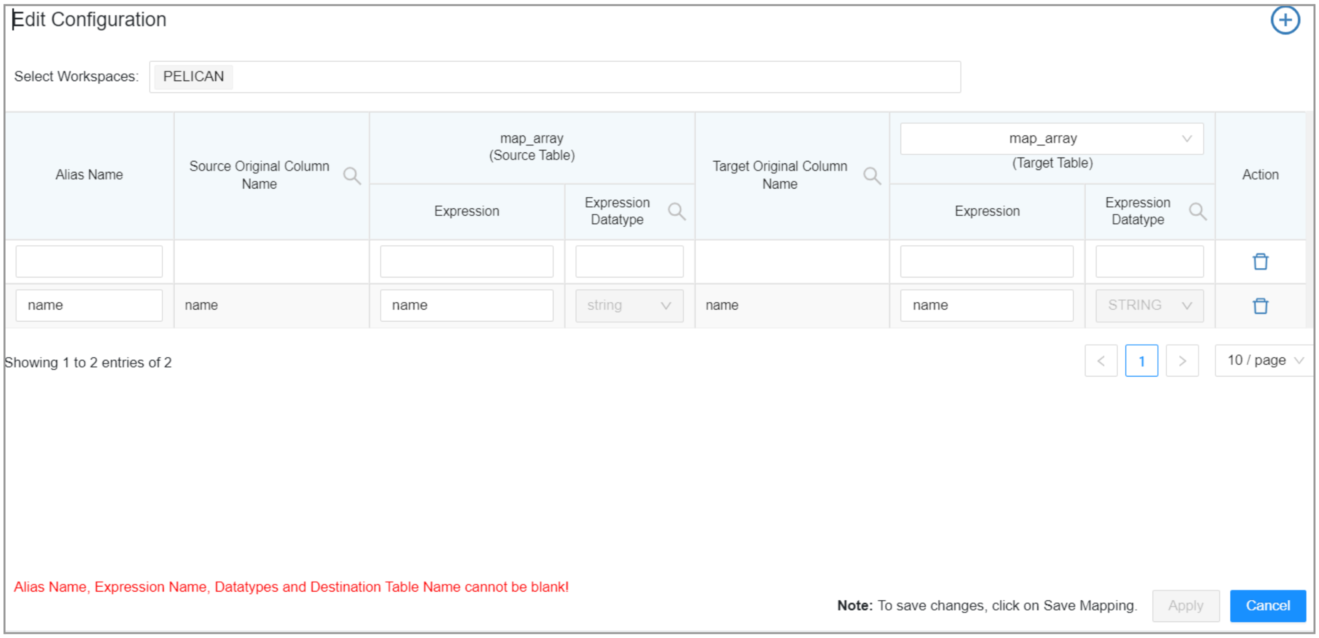
The user can edit the existing data type of any column in the reference and destination table. Suppose there is a column that exists on both the sides (Reference and Destination tables), but the data type of these columns is ***Varchar*** and ***String*** respectively. Logically these two data types are similar in nature. Hence, instead of two data types, the user can change it to either ***String*** or ***Varchar***.

To edit the existing data type:

Select the new data type from the **Datatype** drop-down list and click **OK**.

Add new column

Pelican allows the user to add new columns as per the business requirement, the user can add any number of columns. The application adds the new column as shown below.



Edit Configuration

Select Workspaces: PELICAN

Alias Name	Source Original Column Name	map_array (Source Table)		Target Original Column Name	map_array (Target Table)		Action
		Expression	Expression Datatype		Expression	Expression Datatype	
							🗑
name	name	name	string	name	name	STRING	🗑

Showing 1 to 2 entries of 2

Alias Name, Expression Name, Datatypes and Destination Table Name cannot be blank!

Note: To save changes, click on Save Mapping.

1. Enter column name (Alias Name).
2. Enter expression, data type on both the sides (Reference and Destination tables).
3. Click Save to save the newly added column.
   - **Delete columns**

     The user can delete the existing columns of the reference and destination table if they are not required. Users can delete columns from the mapped, partially mapped, unmapped and previously mapped table sections as well. However, the deleted columns will be removed only from the Pelican UI and not from the original database.

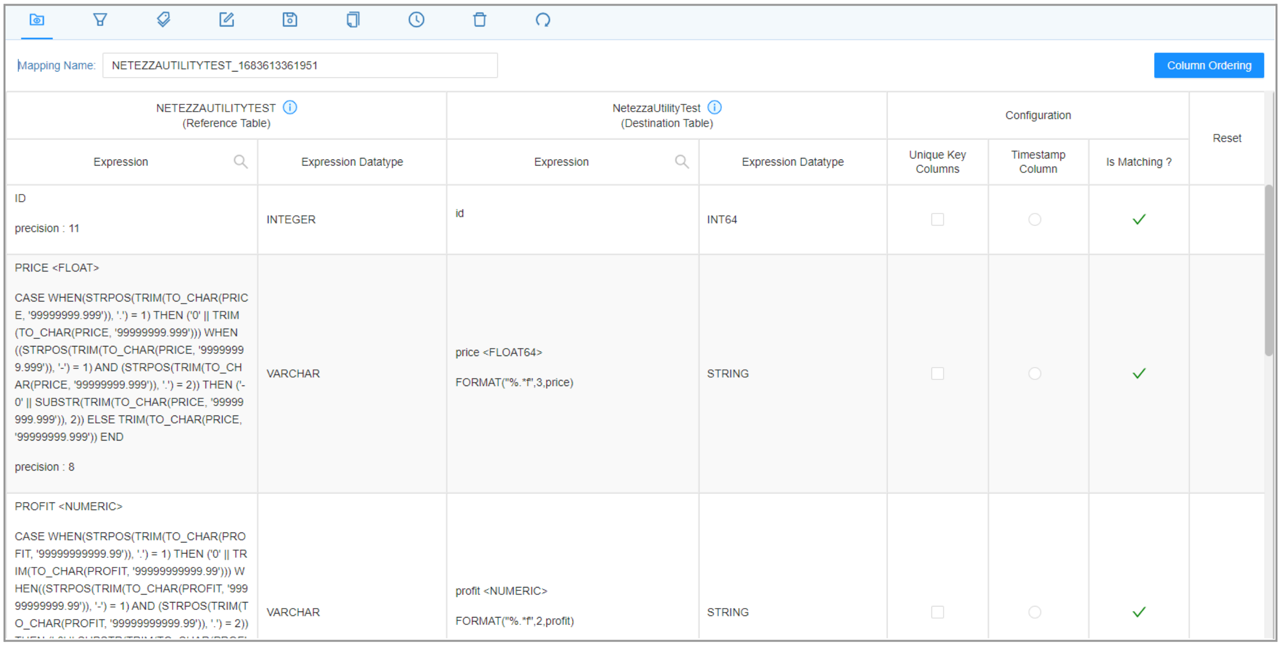
9.4. Delete Mapping

The user can delete mappings from the respective sections if they are not required.

However, the deleted mappings will be removed only from the Pelican UI and not from the original database.

To delete a mapping:

1. Click the Delete Mapping icon .



Mapping Name: NETEZZAUTILITYTEST_1683613361951

NETEZZAUTILITYTEST (Reference Table)		NetezzaUtilityTest (Destination Table)		Configuration			Reset
Expression	Expression Datatype	Expression	Expression Datatype	Unique Key Columns	Timestamp Column	Is Matching ?	
ID precision : 11	INTEGER	id	INT64	□	○	✓	
PRICE <FLOAT> CASE WHEN(STRPOS(TRIM(TO_CHAR(PRICE, '99999999.999')), '.') = 1) THEN ('0' || TRIM (TO_CHAR(PRICE, '99999999.999'))) WHEN ((STRPOS(TRIM(TO_CHAR(PRICE, '99999999.999')), '-') = 1) AND (STRPOS(TRIM(TO_CHAR(PRICE, '99999999.999')), '.') = 2)) THEN ('-0' || SUBSTR(TRIM(TO_CHAR(PRICE, '99999999.999')), 2)) ELSE TRIM(TO_CHAR(PRICE, '99999999.999')) END precision : 8	VARCHAR	price <FLOAT64> FORMAT("%.*f",3,price)	STRING	□	○	✓	
PROFIT <NUMERIC> CASE WHEN(STRPOS(TRIM(TO_CHAR(PROFIT, '99999999999.99')), '.') = 1) THEN ('0' || TRIM(TO_CHAR(PROFIT, '99999999999.99'))) WHEN((STRPOS(TRIM(TO_CHAR(PROFIT, '99999999999.99')), '-') = 1) AND (STRPOS(TRIM(TO_CHAR(PROFIT, '99999999999.99')), '.') = 2))	VARCHAR	profit <NUMERIC> FORMAT("%.*f",2,profit)	STRING	□	○	✓	

9.5. Override unmapped columns

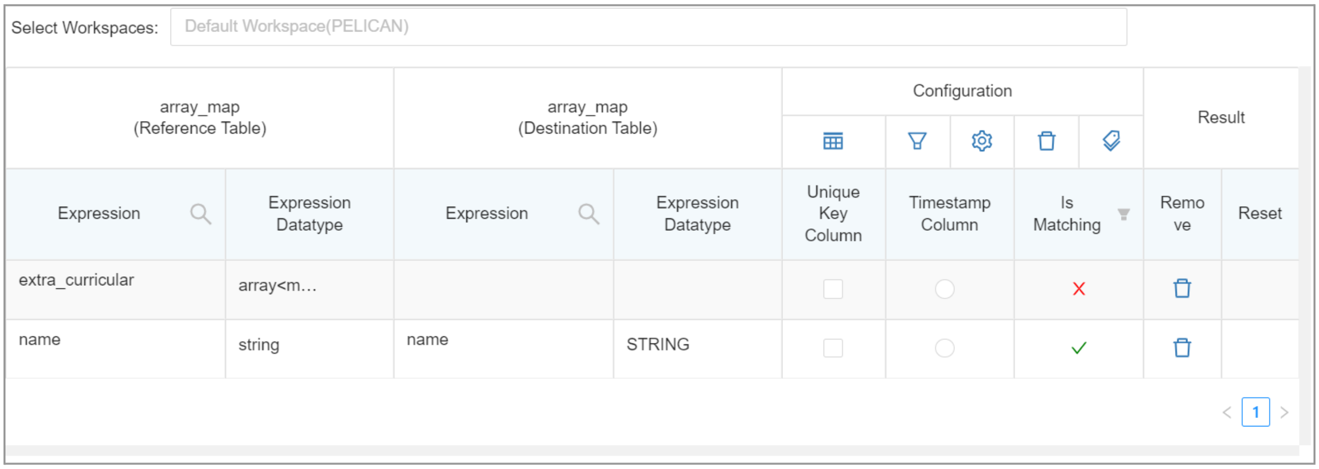
Pelican allows the user to override the unmapped column.

For example,

Suppose column ID_P and id_p exist on both the sides (Reference and Destination tables), logically these two columns are identical as both represent student identification numbers in the respective databases. However, while mapping the application won't map ID_P and id_p due to variation in name and mark then under the unmapped columns section. In such cases, the user can override the unmapped columns into mapped columns.

To override unmapped columns:

1. Locate the unmapped column row on the screen and click the ✖ symbol as shown below.



Select Workspaces: Default Workspace(PELICAN)

array_map (Reference Table)		array_map (Destination Table)		Configuration					Result	
Expression	Expression Datatype	Expression	Expression Datatype	Unique Key Column	Timestamp Column	Is Matching			Remove	Reset
extra_curricular	array<m...			□	○	✗			🗑	
name	string	name	STRING	□	○	✓			🗑	

< 1 >

2. Click the **Save Mapping** button to save the changes.

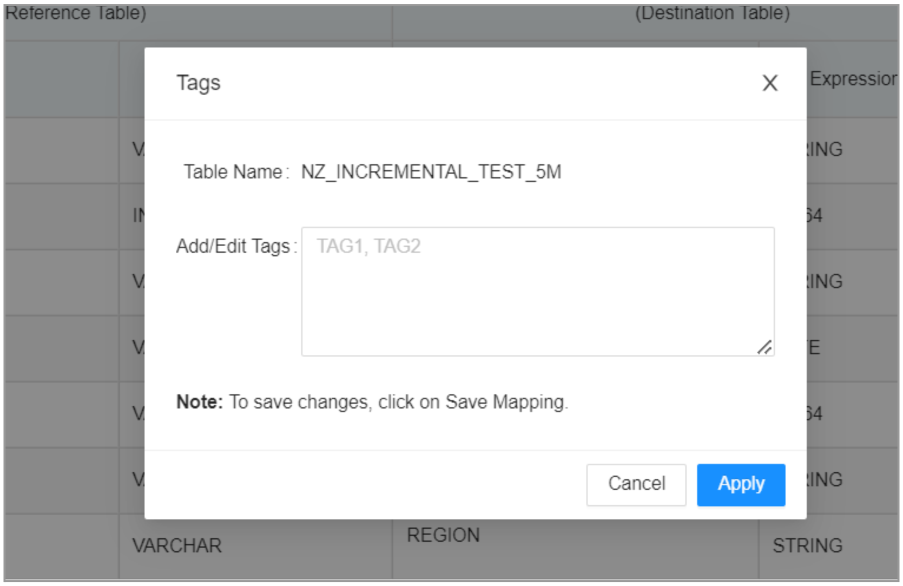
9.6. Add Tags

Tagging is an optional feature associated with the table mappings. It can be used to have additional information for the mappings. user can set one or more tags on a table mapping.
Same tags can be used with multiple mappings. user can change tags any number of times from the Edit Mapping page.

The steps to Add tags are as follows:

1. A user can add a tag to an existing mapping or a new mapping.

2. Add new Tags from the **Mapping page** by clicking the Tag icon 🏷.



Tags

Table Name : NZ_INCREMENTAL_TEST_5M

Add/Edit Tags : TAG1, TAG2

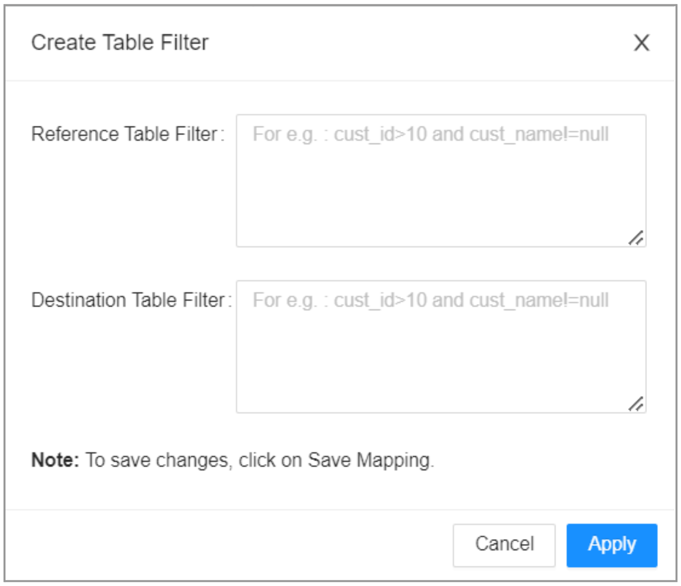
Note: To save changes, click on Save Mapping.

3. Click **Ok**.
4. Click **Save**.

Note: Existing tags can be edited by following similar steps.

9.7. Filter Query

This functionality enables the user to put filters to the Reference and Destination tables.



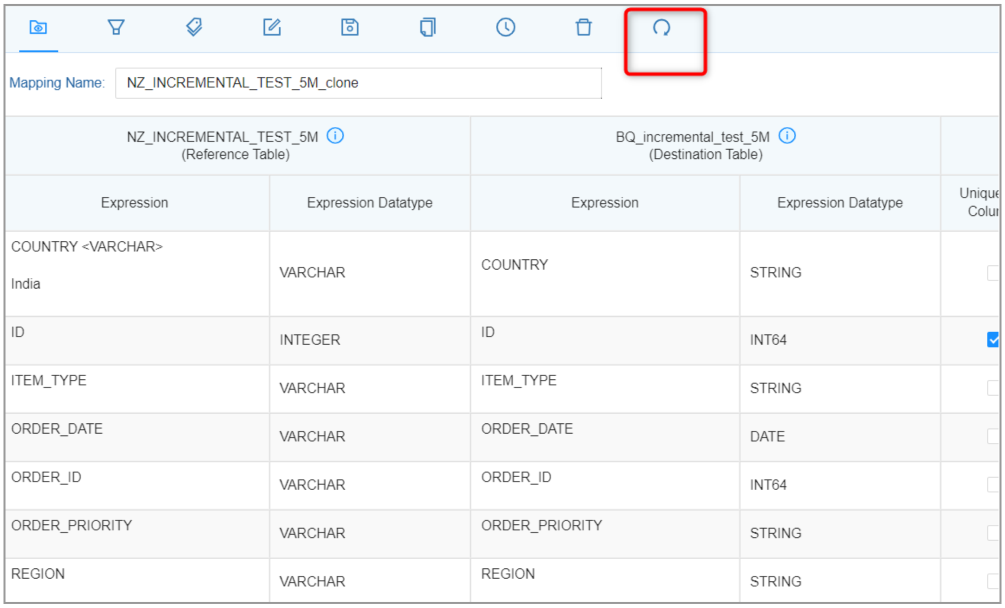
Create Table Filter

Reference Table Filter : For e.g. : cust_id>10 and cust_name!=null

Destination Table Filter : For e.g. : cust_id>10 and cust_name!=null

Note: To save changes, click on Save Mapping.

9.8. Reset Mapping



Mapping Name: NZ_INCREMENTAL_TEST_5M_clone

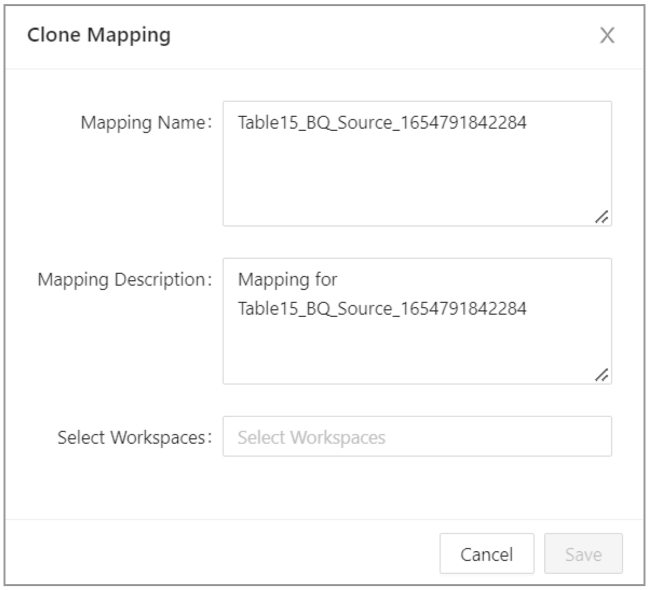
NZ_INCREMENTAL_TEST_5M (Reference Table)		BQ_incremental_test_5M (Destination Table)		
Expression	Expression Datatype	Expression	Expression Datatype	Unique Colu
COUNTRY <VARCHAR> India	VARCHAR	COUNTRY	STRING	□
ID	INTEGER	ID	INT64	☑
ITEM_TYPE	VARCHAR	ITEM_TYPE	STRING	□
ORDER_DATE	VARCHAR	ORDER_DATE	DATE	□
ORDER_ID	VARCHAR	ORDER_ID	INT64	□
ORDER_PRIORITY	VARCHAR	ORDER_PRIORITY	STRING	□
REGION	VARCHAR	REGION	STRING	□

The Reset mapping resets all the parameters of a mapping which were given manually.

Note: The custom expressions are reset to their original format.

9.9. Clone Mapping

This functionality enables the user to create an exact replica of the mapping as per the requirement.



Clone Mapping

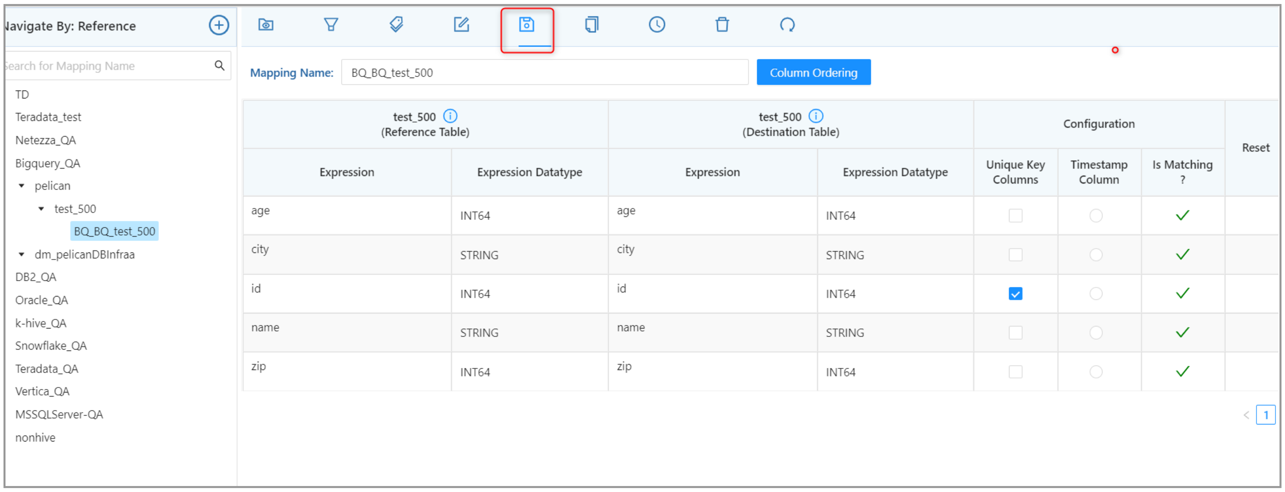
Mapping Name: Table15_BQ_Source_1654791842284

Mapping Description: Mapping for Table15_BQ_Source_1654791842284

Select Workspaces: Select Workspaces

9.10. Saving a Mapping

Click on the **Save Mapping** icon to save the mapping.



Navigate By: Reference

Search for Mapping Name

- TD
- Teradata_test
- Netezza_QA
- Bigquery_QA
- pelican
  - test_500
    - BQ_BQ_test_500
- dm_pelicanDBInfraa
- DB2_QA
- Oracle_QA
- k-hive_QA
- Snowflake_QA
- Teradata_QA
- Vertica_QA
- MSSQLServer-QA
- nonhive

Mapping Name: BQ_BQ_test_500 Column Ordering

test_500 (Reference Table)		test_500 (Destination Table)		Configuration			Reset
Expression	Expression Datatype	Expression	Expression Datatype	Unique Key Columns	Timestamp Column	Is Matching ?	
age	INT64	age	INT64	□	○	✓	
city	STRING	city	STRING	□	○	✓	
id	INT64	id	INT64	☑	○	✓	
name	STRING	name	STRING	□	○	✓	
zip	INT64	zip	INT64	□	○	✓	

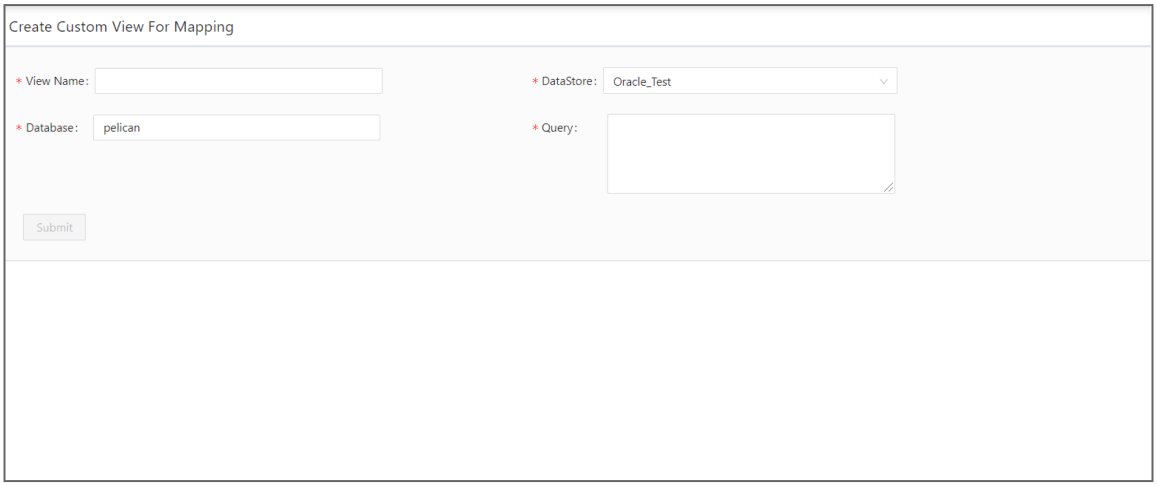
10. Custom View Creation

This functionality provides the capability to create views from the User Interface, i.e. Pelican provides an option from the UI to generate a ‘Create view’ query in which multiple tables to be viewed can be specified. The View Creation feature is only visible to Admin/Super user.

Steps to create custom Views for mapping:

1. Select **View Creation** from the **Govern** menu.

   The **Create Custom Views For Mapping** page will be displayed.



Create Custom View For Mapping

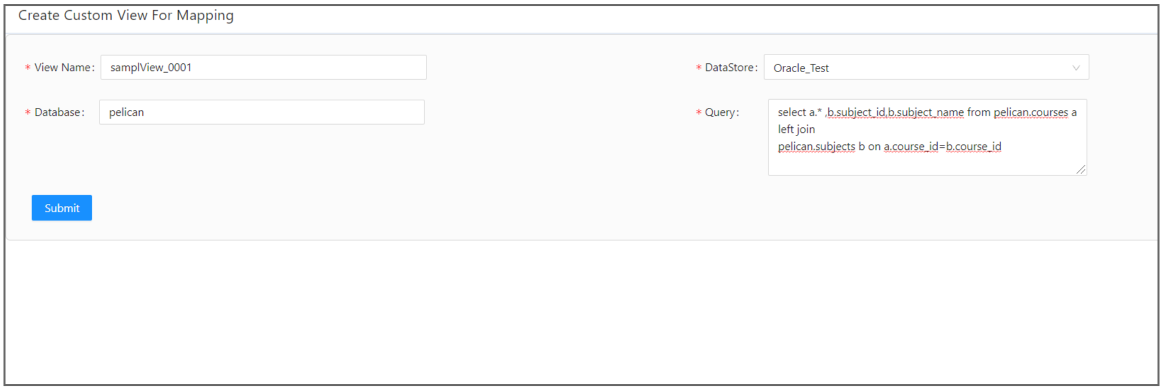
* View Name:

* DataStore: Oracle_Test

* Database: pelican

* Query:

2. Enter a unique **View Name**.
3. Select the **DataStore** and **Database**.
4. Enter an appropriate **Query**.



Create Custom View For Mapping

* View Name: samplView_0001

* DataStore: Oracle_Test

* Database: pelican

* Query:

```
select a.* ,b.subject_id,b.subject_name from pelican.courses a
left join
pelican.subjects b on a.course_id=b.course_id
```

5. Click **Submit**.

   The corresponding view will be generated in the database.

   This newly created view can be used for mapping and validation. It can also be used as an alternative to the surrogate feature.

   Note:

   Rules and Guidelines for SQL Views:

   Certain rules and guidelines apply when you create an SQL view.

   - All tables in the SQL view must exist in the same database.
   - The SQL statement must use valid SQL syntax based on the database of the tables included in the SQL view. Data Validation Option does not validate the SQL statement syntax.

- The number and order of the columns must match in the column definitions and the SQL statement of the SQL view.
- The scale must be zero for datetime, integer, and string data types.
- If you create a column in the SQL view, verify that the datatype, precision, and scale match the PowerCenter data type, precision, and scale shown in the Column Definitions area.
- You can add database functions, any join type, subqueries, and stored procedures in the SQL statement of an SQL view.
- If the SQL view calls a stored procedure, the connection assigned to SQL view must have Execute permission on the stored procedure.
- If you include the SQL view in a table pair, you can create a WHERE clause for the table pair based on a field in the SQL view.
- When you use SQL view as the data source, you can edit the connection in the table pairs, single tables, aggregate views, and join views.

11. Scheduler Configuration

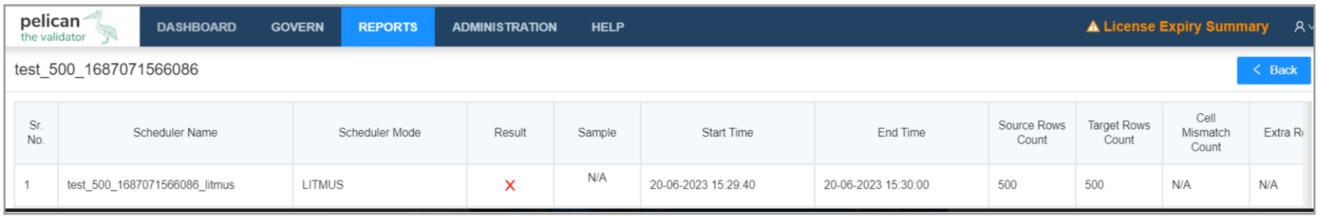
In Pelican, the user can create a scheduler for a saved mapping, so that, after a specific time period the scheduler executes the process and validates the source table with the destination table. Once the user completes the table mapping process the respective schedulers can be configured.

11.1. Modes in Scheduler Configuration

The Scheduler supports two modes of validation - Litmus and Full.

11.1.1. LITMUS mode

The Litmus mode analyzes whether the tables at source and target are matching or not. It does not support cell level differences or samples.

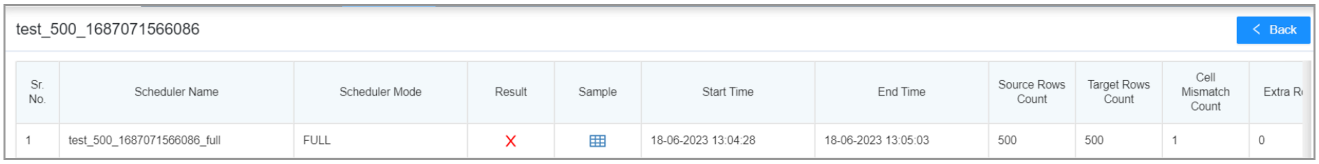


test_500_1687071566086

Sr. No.	Scheduler Name	Scheduler Mode	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra R
1	test_500_1687071566086_litmus	LITMUS	✗	N/A	20-06-2023 15:29:40	20-06-2023 15:30:00	500	500	N/A	N/A

11.1.2. Full mode

Along with data validation as in Litmus mode, the Full mode displays the cell level differences through the sample rows fetched from both source and target tables.

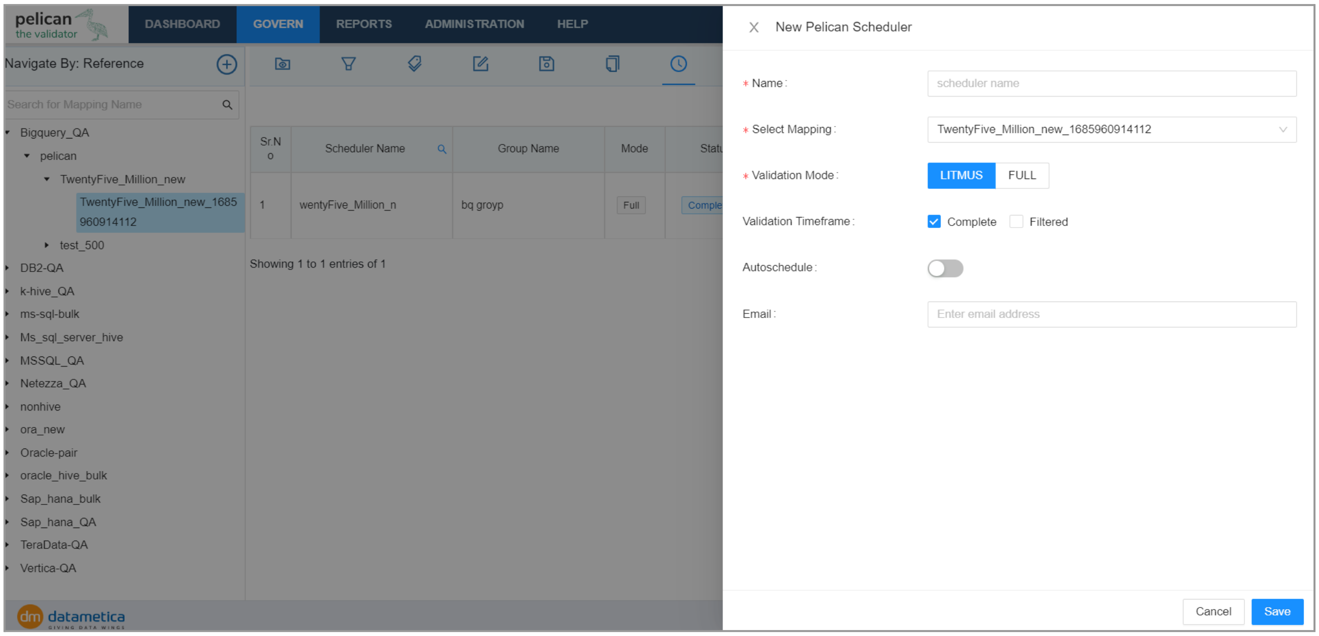


test_500_1687071566086

Sr. No.	Scheduler Name	Scheduler Mode	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra R
1	test_500_1687071566086_full	FULL	✗	▦	18-06-2023 13:04:28	18-06-2023 13:05:03	500	500	1	0

11.2. Steps to configure the scheduler

1. Navigate **Govern** -> **Validation Configuration**.
   (Create **Mappings**-> then go to **View Schedulers** -> **Create New scheduler(+ icon**))



Sr.No	Scheduler Name	Group Name	Mode	Statu
1	wentyFive_Million_n	bq groyp	Full	Comple

Showing 1 to 1 entries of 1

New Pelican Scheduler

* Name : scheduler name

* Select Mapping : TwentyFive_Million_new_1685960914112

* Validation Mode : [✓ LITMUS] FULL

Validation Timeframe : ☑ Complete □ Filtered

Autoschedule : ○

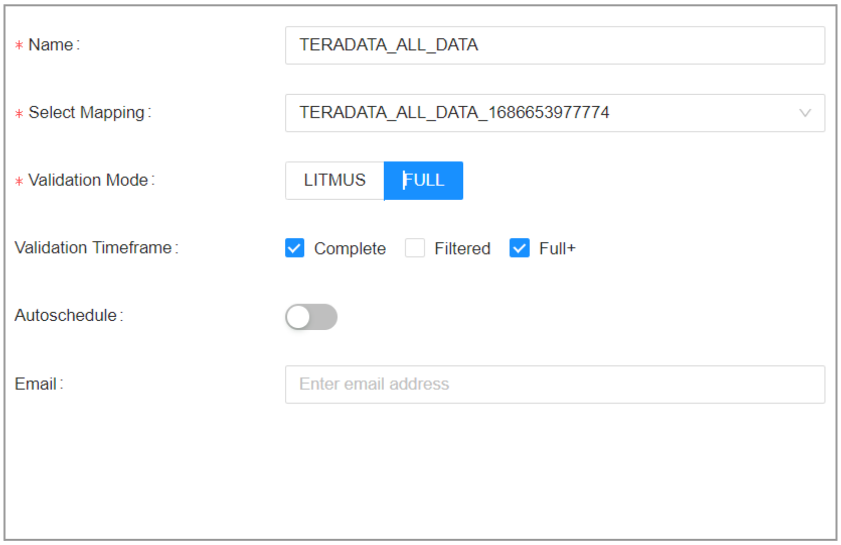
Email : Enter email address

Cancel Save

2. Click the **Add Pelican Scheduler** icon.
3. Enter Scheduler **Name** in the field provided and select table.
4. For validation the user can select either **LITMUS** or **FULL** mode as per requirement.
5. Select the appropriate option from the **Validation Timeframe**.

   Notes:

   - The **FULL+** is an extension to the FULL mode.



* Name : TERADATA_ALL_DATA

* Select Mapping : TERADATA_ALL_DATA_1686653977774

* Validation Mode : LITMUS [✓ FULL]

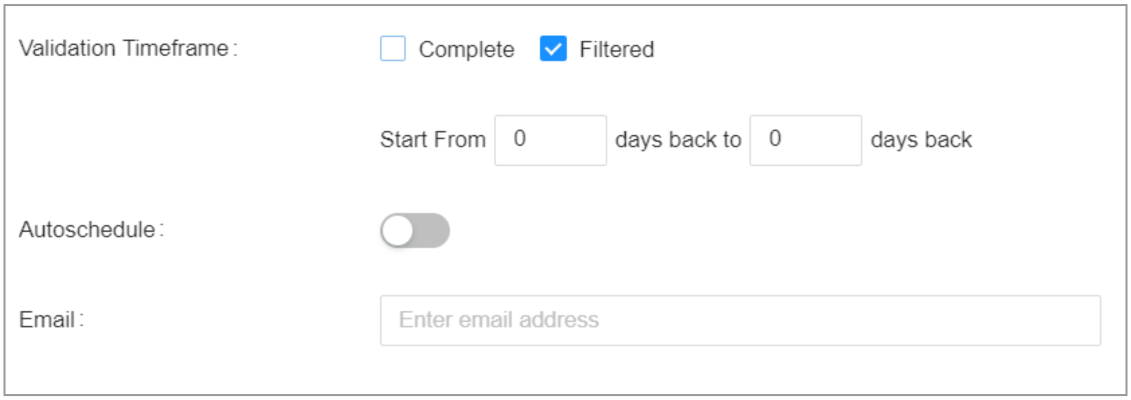
Validation Timeframe : ☑ Complete □ Filtered ☑ Full+

Autoschedule : ○

Email : Enter email address

   - The **'Filtered'** enables incremental validation.

     You can select the validation timeframe for the scheduler execution by setting the filters for the specific time period.



Validation Timeframe : □ Complete ☑ Filtered

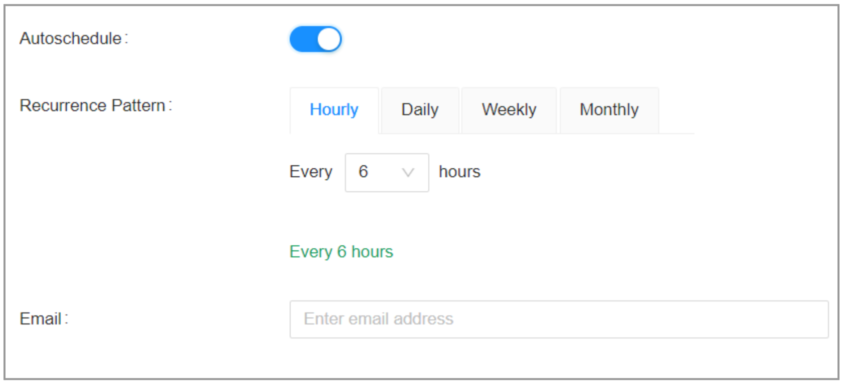
Start From 0 days back to 0 days back

Autoschedule : ○

Email : Enter email address

- To perform this step the appropriate Date column should be selected as a timestamp while creating the table.
- The **Full+** validation provides an advantage of finding the mismatches in configured percentage of no. of total columns in a single execution for tables with more mismatching columns as compared to multiple executions of the Full mode.
- Full+ is applicable only if the FULL mode is selected. Also, while using the Full+ mode either the Complete or Filtered validations should be selected. However, you cannot select Complete and Filtered together.
- Currently the Full+ mode validation is supported only for Teradata-BigQuery, Netezza-BigQuery, Hive-BigQuery and Oracle-BigQuery , MSSQL Server - Snowflake, Snowflake - Big Query.

6. Click the **Autoschedule** toggle button to set recurrence Pattern daily, monthly or weekly.



Autoschedule : ●

Recurrence Pattern : Hourly Daily Weekly Monthly

Every 6 hours

Every 6 hours

Email : Enter email address

7. Select **Email** toggle button and enter email address if user wants reports to be sent for specific scheduler.

   Note: Steps 6, 7 and 8 are optional.

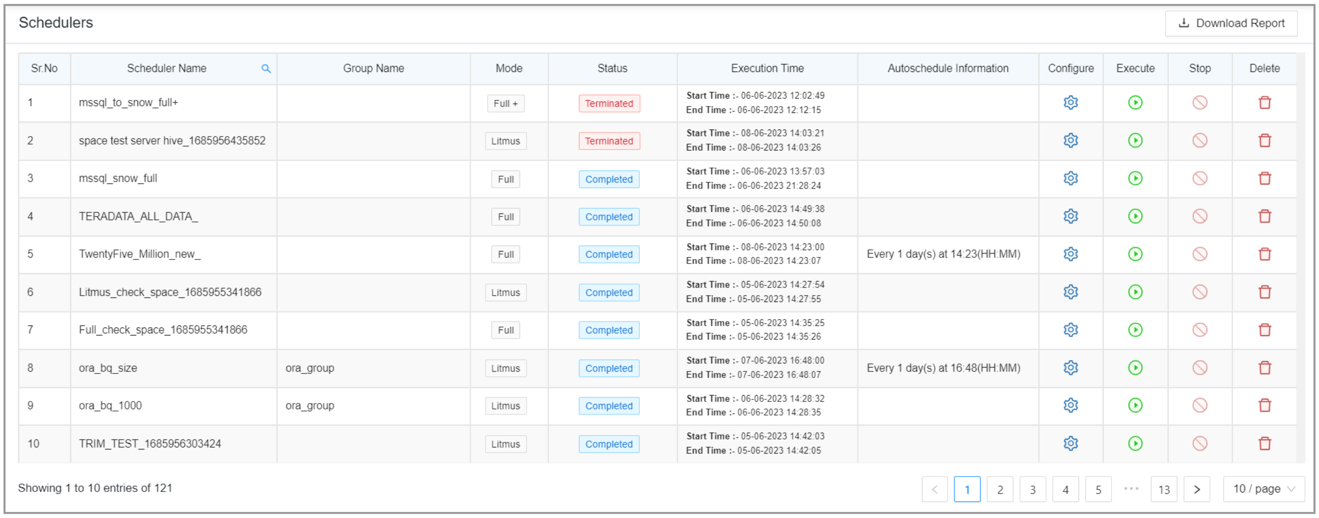
8. Click **SAVE**.

9. Click **Execute Now** for on-demand execution of any of the schedulers.

Note:

- Click **Stop** to stop execution of the Scheduler. The status of the Scheduler will be changed to **Terminated**.
- Click Delete to delete the scheduler.

The newly created scheduler is added to the **Schedulers** screen.



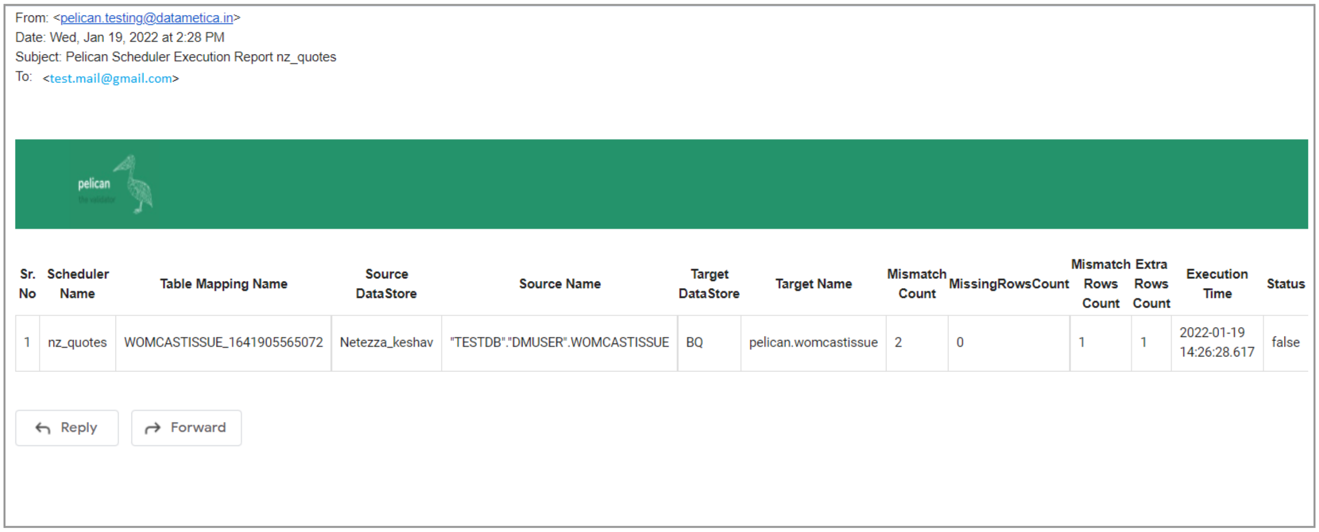
Schedulers

Sr.No	Scheduler Name	Group Name	Mode	Status	Execution Time	Autoschedule Information	Configure	Execute	Stop	Delete
1	mssql_to_snow_full+		Full +	Terminated	**Start Time :-** 06-06-2023 12:02:49 **End Time :-** 06-06-2023 12:12:15		⚙	▶	🚫	🗑
2	space test server hive_1685956435852		Litmus	Terminated	**Start Time :-** 08-06-2023 14:03:21 **End Time :-** 08-06-2023 14:03:26		⚙	▶	🚫	🗑
3	mssql_snow_full		Full	Completed	**Start Time :-** 06-06-2023 13:57:03 **End Time :-** 06-06-2023 21:28:24		⚙	▶	🚫	🗑
4	TERADATA_ALL_DATA_		Full	Completed	**Start Time :-** 06-06-2023 14:49:38 **End Time :-** 06-06-2023 14:50:08		⚙	▶	🚫	🗑
5	TwentyFive_Million_new_		Full	Completed	**Start Time :-** 08-06-2023 14:23:00 **End Time :-** 08-06-2023 14:23:07	Every 1 day(s) at 14:23(HH:MM)	⚙	▶	🚫	🗑
6	Litmus_check_space_1685955341866		Litmus	Completed	**Start Time :-** 05-06-2023 14:27:54 **End Time :-** 05-06-2023 14:27:55		⚙	▶	🚫	🗑
7	Full_check_space_1685955341866		Full	Completed	**Start Time :-** 05-06-2023 14:35:25 **End Time :-** 05-06-2023 14:35:26		⚙	▶	🚫	🗑
8	ora_bq_size	ora_group	Litmus	Completed	**Start Time :-** 07-06-2023 16:48:00 **End Time :-** 07-06-2023 16:48:07	Every 1 day(s) at 16:48(HH:MM)	⚙	▶	🚫	🗑
9	ora_bq_1000	ora_group	Litmus	Completed	**Start Time :-** 06-06-2023 14:28:32 **End Time :-** 06-06-2023 14:28:35		⚙	▶	🚫	🗑
10	TRIM_TEST_1685956303424		Litmus	Completed	**Start Time :-** 05-06-2023 14:42:03 **End Time :-** 05-06-2023 14:42:05		⚙	▶	🚫	🗑

Showing 1 to 10 entries of 121

Note:

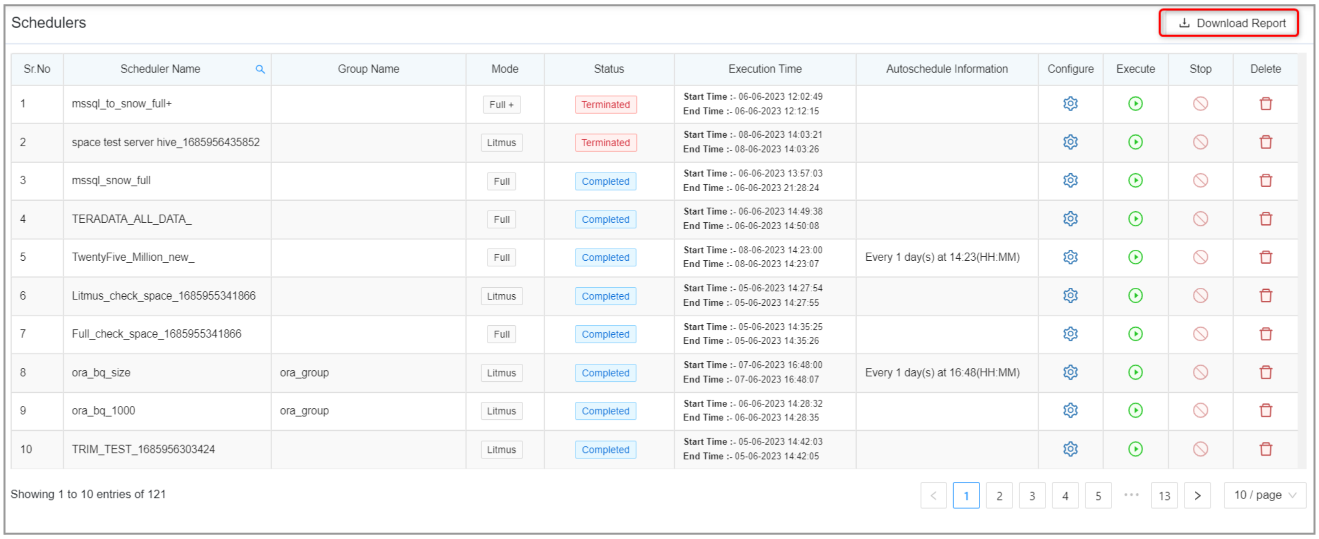
The following scheduler report email will be sent to the email address mentioned while creating the scheduler:



From: <pelican.testing@datametica.in>
Date: Wed, Jan 19, 2022 at 2:28 PM
Subject: Pelican Scheduler Execution Report nz_quotes
To: <test.mail@gmail.com>

Sr. No	Scheduler Name	Table Mapping Name	Source DataStore	Source Name	Target DataStore	Target Name	Mismatch Count	MissingRowsCount	Mismatch Rows Count	Extra Rows Count	Execution Time	Status
1	nz_quotes	WOMCASTISSUE_1641905565072	Netezza_keshav	"TESTDB"."DMUSER".WOMCASTISSUE	BQ	pelican.womcastissue	2	0	1	1	2022-01-19 14:26:28.617	false

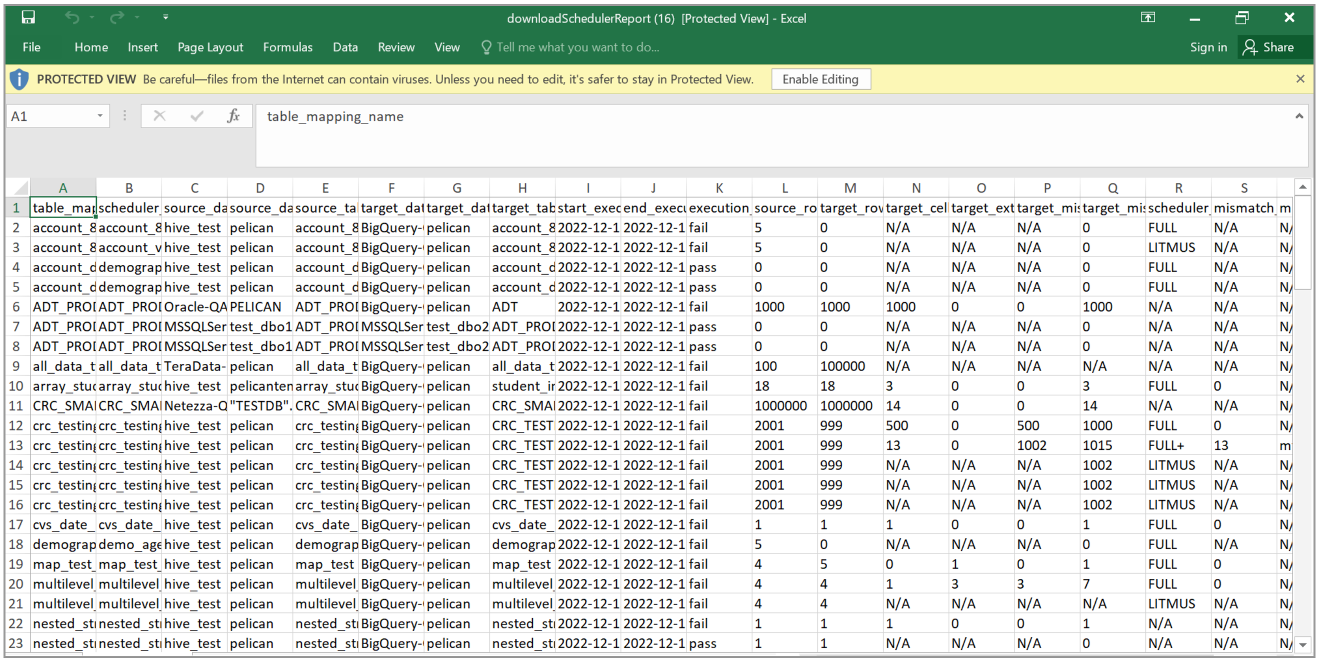
11.3. Scheduler Status Report



Sr.No	Scheduler Name	Group Name	Mode	Status	Execution Time	Autoschedule Information	Configure	Execute	Stop	Delete
1	mssql_to_snow_full+		Full +	Terminated	Start Time :- 06-06-2023 12:02:49 End Time :- 06-06-2023 12:12:15		⚙	▷	⊘	🗑
2	space test server hive_1685956435852		Litmus	Terminated	Start Time :- 08-06-2023 14:03:21 End Time :- 08-06-2023 14:03:26		⚙	▷	⊘	🗑
3	mssql_snow_full		Full	Completed	Start Time :- 06-06-2023 13:57:03 End Time :- 06-06-2023 21:28:24		⚙	▷	⊘	🗑
4	TERADATA_ALL_DATA_		Full	Completed	Start Time :- 06-06-2023 14:49:38 End Time :- 06-06-2023 14:50:08		⚙	▷	⊘	🗑
5	TwentyFive_Million_new_		Full	Completed	Start Time :- 08-06-2023 14:23:00 End Time :- 08-06-2023 14:23:07	Every 1 day(s) at 14:23(HH:MM)	⚙	▷	⊘	🗑
6	Litmus_check_space_1685955341866		Litmus	Completed	Start Time :- 05-06-2023 14:27:54 End Time :- 05-06-2023 14:27:55		⚙	▷	⊘	🗑
7	Full_check_space_1685955341866		Full	Completed	Start Time :- 05-06-2023 14:35:25 End Time :- 05-06-2023 14:35:26		⚙	▷	⊘	🗑
8	ora_bq_size	ora_group	Litmus	Completed	Start Time :- 07-06-2023 16:48:00 End Time :- 07-06-2023 16:48:07	Every 1 day(s) at 16:48(HH:MM)	⚙	▷	⊘	🗑
9	ora_bq_1000	ora_group	Litmus	Completed	Start Time :- 06-06-2023 14:28:32 End Time :- 06-06-2023 14:28:35		⚙	▷	⊘	🗑
10	TRIM_TEST_1685956303424		Litmus	Completed	Start Time :- 05-06-2023 14:42:03 End Time :- 05-06-2023 14:42:05		⚙	▷	⊘	🗑

Showing 1 to 10 entries of 121

- The **Scheduler Status Report** functionality enables the user to view all the iterations of all the schedulers of the given mapping in the Pelican application.
- To View the Scheduler report click on the **Scheduler Status Report** hyperlink located on the top of the Schedulers page. An .xls file will be downloaded at the bottom of the screen.
- This scheduler report file contains the scheduler and mapping details in_ columns such as mapping name, scheduler name, source and destination details, etc.


downloadSchedulerReport (16) [Protected View] - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Tell me what you want to do...
Sign in
Share
Enable Editing
A1
fx

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

table_mapping_name

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	table_ma	scheduler_	source_da	source_da	source_ta	target_dat	target_dat	target_tab	start_exec	end_execu	execution_	source_ro	target_rov	target_cel	target_ext	target_mis	target_mis	scheduler_	mismatch_	m
2	account_8	account_8	hive_test	pelican	account_8	BigQuery-	pelican	account_8	2022-12-1	2022-12-1	fail	5	0	N/A	N/A	N/A	0	FULL	N/A	N/
3	account_8	account_v	hive_test	pelican	account_8	BigQuery-	pelican	account_8	2022-12-1	2022-12-1	fail	5	0	N/A	N/A	N/A	0	LITMUS	N/A	N/
4	account_d	demograp	hive_test	pelican	account_d	BigQuery-	pelican	account_d	2022-12-1	2022-12-1	pass	0	0	N/A	N/A	N/A	0	FULL	N/A	N/
5	account_d	demograp	hive_test	pelican	account_d	BigQuery-	pelican	account_d	2022-12-1	2022-12-1	pass	0	0	N/A	N/A	N/A	0	FULL	N/A	N/
6	ADT_PRO	ADT_PRO	Oracle-QA	PELICAN	ADT_PRO	BigQuery-	pelican	ADT	2022-12-1	2022-12-1	fail	1000	1000	1000	0	0	1000	N/A	N/A	N/
7	ADT_PRO	ADT_PRO	MSSQLSer	test_dbo1	ADT_PRO	MSSQLSer	test_dbo2	ADT_PRO	2022-12-1	2022-12-1	pass	0	0	N/A	N/A	N/A	0	N/A	N/A	N/
8	ADT_PRO	ADT_PRO	MSSQLSer	test_dbo1	ADT_PRO	MSSQLSer	test_dbo2	ADT_PRO	2022-12-1	2022-12-1	pass	0	0	N/A	N/A	N/A	0	N/A	N/A	N/
9	all_data_t	all_data_t	TeraData-	pelican	all_data_t	BigQuery-	pelican	all_data_t	2022-12-1	2022-12-1	fail	100	100000	N/A	N/A	N/A	N/A	N/A	N/A	N/
10	array_stu	array_stu	hive_test	pelicanten	array_stu	BigQuery-	pelican	student_i	2022-12-1	2022-12-1	fail	18	18	3	0	0	3	FULL	0	N/
11	CRC_SMA	CRC_SMA	Netezza-Q	"TESTDB".	CRC_SMA	BigQuery-	pelican	CRC_SMA	2022-12-1	2022-12-1	fail	1000000	1000000	14	0	0	14	N/A	N/A	N/
12	crc_testing	crc_testing	hive_test	pelican	crc_testing	BigQuery-	pelican	CRC_TEST	2022-12-1	2022-12-1	fail	2001	999	500	0	500	1000	FULL	0	N/
13	crc_testing	crc_testing	hive_test	pelican	crc_testing	BigQuery-	pelican	CRC_TEST	2022-12-1	2022-12-1	fail	2001	999	13	0	1002	1015	FULL+	13	m
14	crc_testing	crc_testing	hive_test	pelican	crc_testing	BigQuery-	pelican	CRC_TEST	2022-12-1	2022-12-1	fail	2001	999	N/A	N/A	N/A	1002	LITMUS	N/A	N/
15	crc_testing	crc_testing	hive_test	pelican	crc_testing	BigQuery-	pelican	CRC_TEST	2022-12-1	2022-12-1	fail	2001	999	N/A	N/A	N/A	1002	LITMUS	N/A	N/
16	crc_testing	crc_testing	hive_test	pelican	crc_testing	BigQuery-	pelican	CRC_TEST	2022-12-1	2022-12-1	fail	2001	999	N/A	N/A	N/A	1002	LITMUS	N/A	N/
17	cvs_date_	cvs_date_	hive_test	pelican	cvs_date_	BigQuery-	pelican	cvs_date_	2022-12-1	2022-12-1	fail	1	1	1	0	0	1	FULL	0	N/
18	demograp	demo_age	hive_test	pelican	demograp	BigQuery-	pelican	demograp	2022-12-1	2022-12-1	fail	5	0	N/A	N/A	N/A	0	FULL	N/A	N/
19	map_test_	map_test_	hive_test	pelican	map_test	BigQuery-	pelican	map_test	2022-12-1	2022-12-1	fail	4	5	0	1	0	1	FULL	0	N/
20	multilevel_	multilevel_	hive_test	pelican	multilevel_	BigQuery-	pelican	multilevel_	2022-12-1	2022-12-1	fail	4	4	1	3	3	7	FULL	0	N/
21	multilevel_	multilevel_	hive_test	pelican	multilevel_	BigQuery-	pelican	multilevel_	2022-12-1	2022-12-1	fail	4	4	N/A	N/A	N/A	N/A	LITMUS	N/A	N/
22	nested_st	nested_st	hive_test	pelican	nested_st	BigQuery-	pelican	nested_st	2022-12-1	2022-12-1	fail	1	1	1	0	0	1	N/A	N/A	N/
23	nested_st	nested_st	hive_test	pelican	nested_st	BigQuery-	pelican	nested_st	2022-12-1	2022-12-1	pass	1	1	N/A	N/A	N/A	0	N/A	N/A	N/

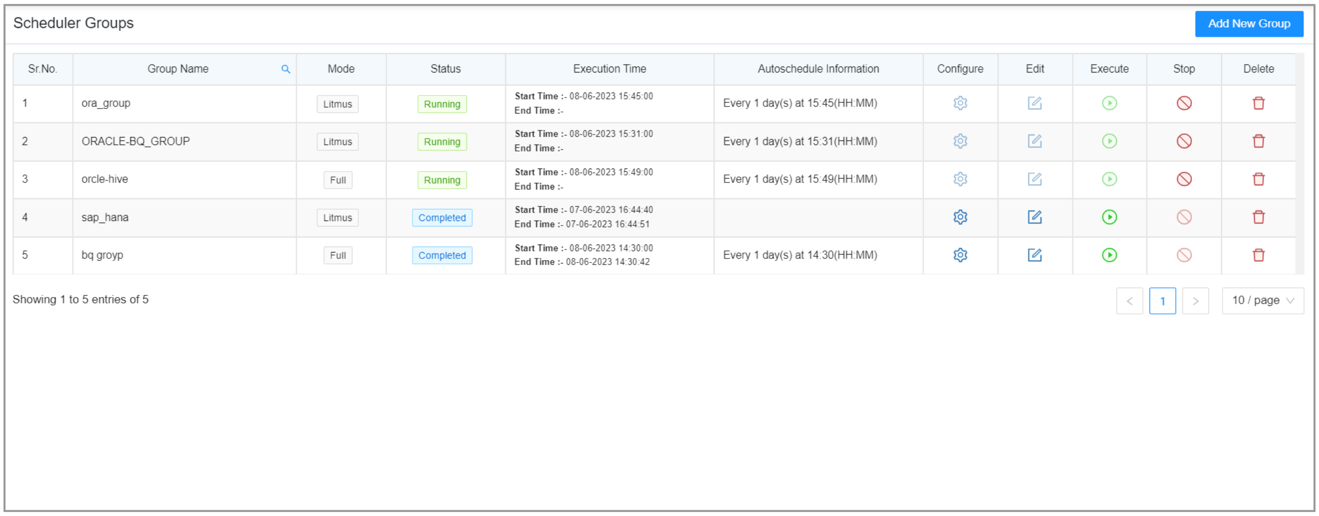
12. Scheduler Group

The Group Scheduling feature enables the user to run multiple schedulers from the same or different datastores at the same time. Only one scheduler can be added for each mapping.

The steps to add a new scheduler group are as follows:

1) Go to **Govern** -> **Scheduler Groups**.

   The **Scheduler** Groups page opens.



Scheduler Groups

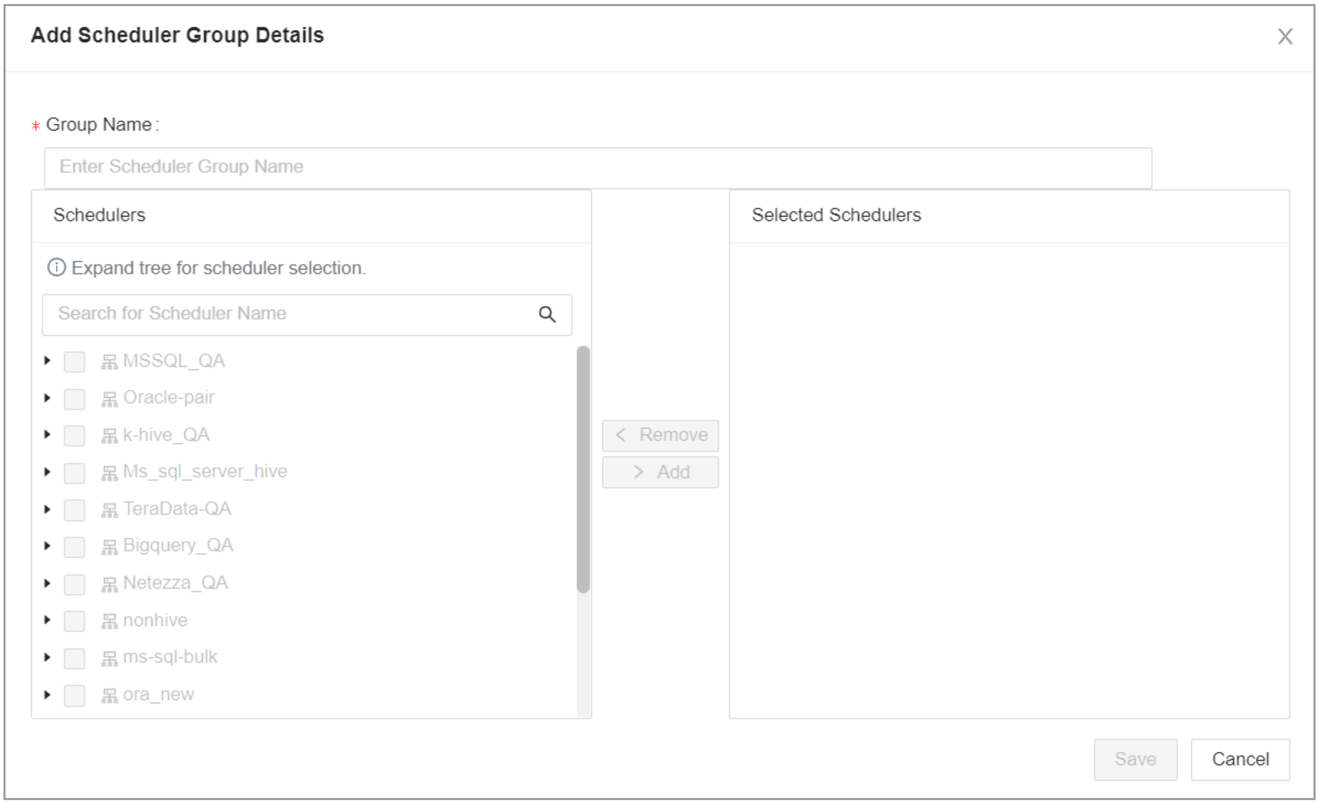
Sr.No.	Group Name	Mode	Status	Execution Time	Autoschedule Information	Configure	Edit	Execute	Stop	Delete
1	ora_group	Litmus	Running	**Start Time** :- 08-06-2023 15:45:00 **End Time** :-	Every 1 day(s) at 15:45(HH:MM)	⚙	✎	▶	⊘	🗑
2	ORACLE-BQ_GROUP	Litmus	Running	**Start Time** :- 08-06-2023 15:31:00 **End Time** :-	Every 1 day(s) at 15:31(HH:MM)	⚙	✎	▶	⊘	🗑
3	orcle-hive	Full	Running	**Start Time** :- 08-06-2023 15:49:00 **End Time** :-	Every 1 day(s) at 15:49(HH:MM)	⚙	✎	▶	⊘	🗑
4	sap_hana	Litmus	Completed	**Start Time** :- 07-06-2023 16:44:40 **End Time** :- 07-06-2023 16:44:51		⚙	✎	▶	⊘	🗑
5	bq groyp	Full	Completed	**Start Time** :- 08-06-2023 14:30:00 **End Time** :- 08-06-2023 14:30:42	Every 1 day(s) at 14:30(HH:MM)	⚙	✎	▶	⊘	🗑

Showing 1 to 5 entries of 5

2) Click on **Add New Group**.

   The **Add Scheduler Group Details** page opens.

   This page contains two sections - **Schedulers** and **Selected Scheduler** Section.



Add Scheduler Group Details

* Group Name :

Enter Scheduler Group Name

Schedulers

ⓘ Expand tree for scheduler selection.

Search for Scheduler Name

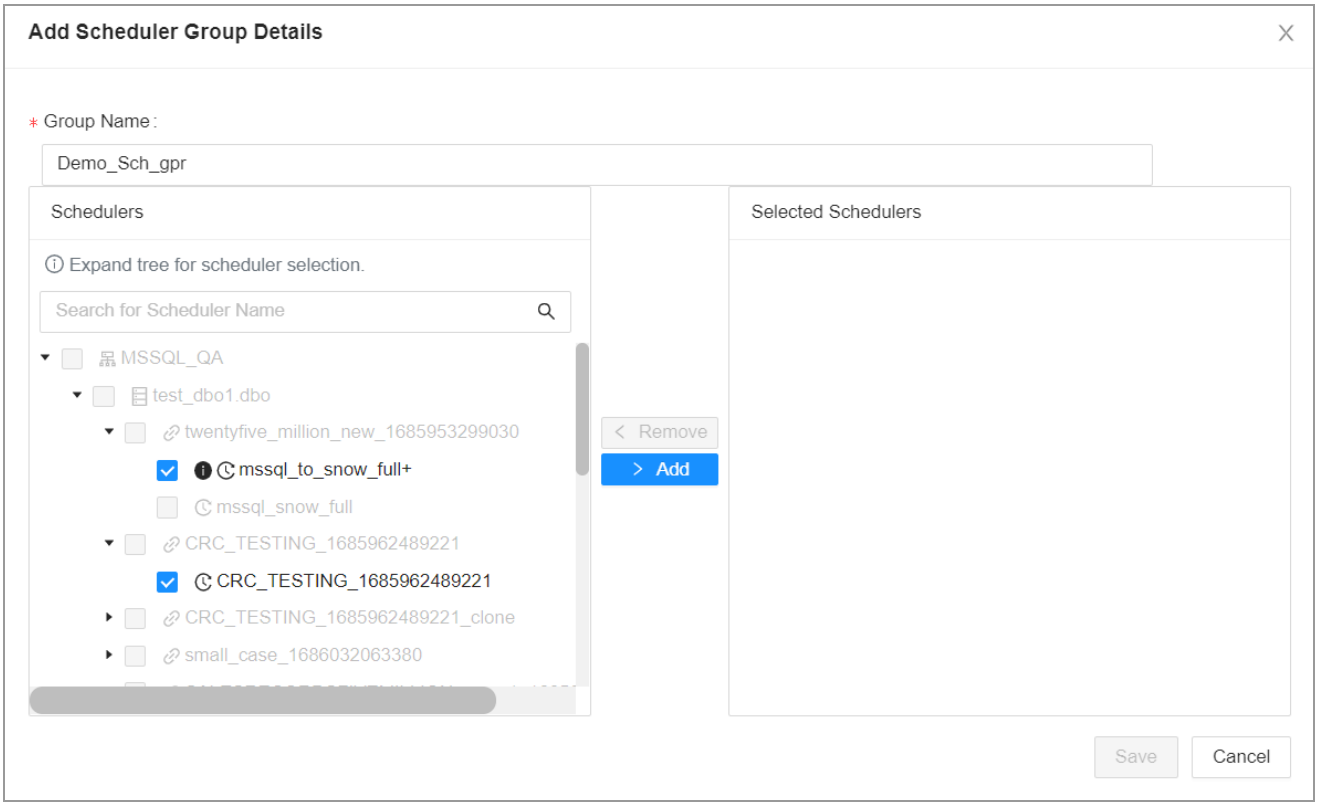
- □ MSSQL_QA
- □ Oracle-pair
- □ k-hive_QA
- □ Ms_sql_server_hive
- □ TeraData-QA
- □ Bigquery_QA
- □ Netezza_QA
- □ nonhive
- □ ms-sql-bulk
- □ ora_new

Selected Schedulers

3) Enter a **Group Name**.

4) Select the Scheduler that you want to add and click on the Add button.

The selected Scheduler will be shifted to the right-side tab.



Add Scheduler Group Details

* Group Name : Demo_Sch_gpr

Schedulers

ⓘ Expand tree for scheduler selection.

Search for Scheduler Name

- ☐ MSSQL_QA
  - ☐ test_dbo1.dbo
    - ☐ twentyfive_million_new_1685953299030
      - ☑ mssql_to_snow_full+
      - ☐ mssql_snow_full
    - ☐ CRC_TESTING_1685962489221
      - ☑ CRC_TESTING_1685962489221
    - ☐ CRC_TESTING_1685962489221_clone
    - ☐ small_case_1686032063380

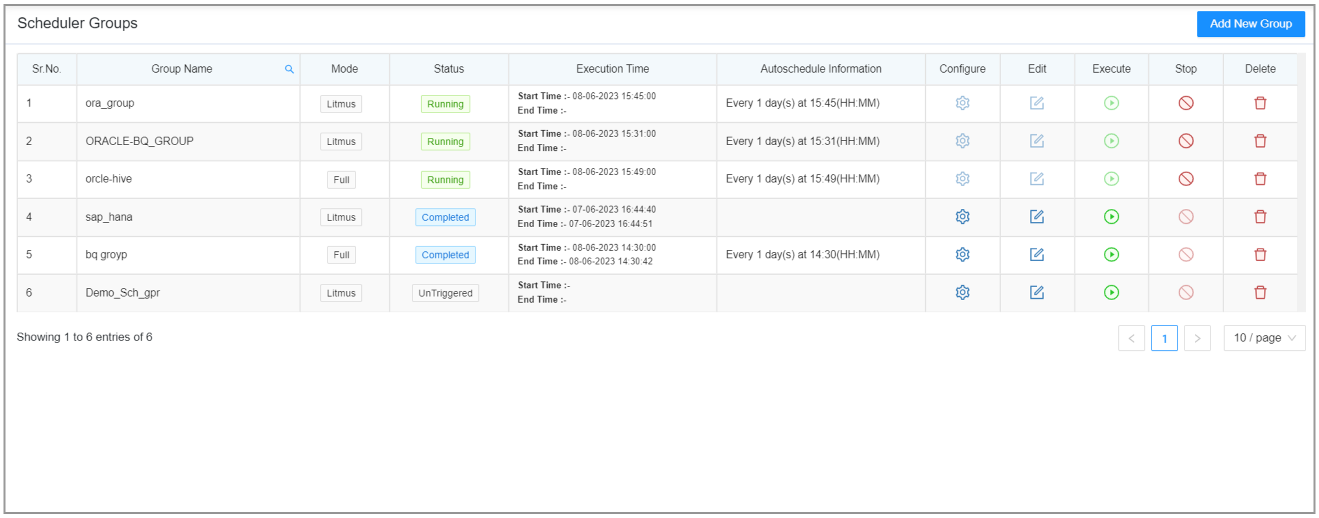
< Remove

> Add

Selected Schedulers

Save Cancel

5) Click **Save** to save the scheduler group.



Scheduler Groups

Add New Group

Sr.No.	Group Name	Mode	Status	Execution Time	Autoschedule Information	Configure	Edit	Execute	Stop	Delete
1	ora_group	Litmus	Running	**Start Time :-** 08-06-2023 15:45:00 **End Time :-**	Every 1 day(s) at 15:45(HH:MM)	⚙	✎	▶	⊘	🗑
2	ORACLE-BQ_GROUP	Litmus	Running	**Start Time :-** 08-06-2023 15:31:00 **End Time :-**	Every 1 day(s) at 15:31(HH:MM)	⚙	✎	▶	⊘	🗑
3	orcle-hive	Full	Running	**Start Time :-** 08-06-2023 15:49:00 **End Time :-**	Every 1 day(s) at 15:49(HH:MM)	⚙	✎	▶	⊘	🗑
4	sap_hana	Litmus	Completed	**Start Time :-** 07-06-2023 16:44:40 **End Time :-** 07-06-2023 16:44:51		⚙	✎	▶	⊘	🗑
5	bq groyp	Full	Completed	**Start Time :-** 08-06-2023 14:30:00 **End Time :-** 08-06-2023 14:30:42	Every 1 day(s) at 14:30(HH:MM)	⚙	✎	▶	⊘	🗑
6	Demo_Sch_gpr	Litmus	UnTriggered	**Start Time :-** **End Time :-**		⚙	✎	▶	⊘	🗑

Showing 1 to 6 entries of 6

< 1 > 10 / page

Now, let us explore each and every column in the **Scheduler Groups** table:

I. **Group Name**: It displays the unique name of the group.

II. **Group Running Time**: It denotes the period of time after which the Group scheduler runs or executes.

III. **Status**: This column displays the status of the scheduler group. For example:

Complete - All the schedulers in the scheduler group have been completed.

Untriggered - Schedulers have not yet run.

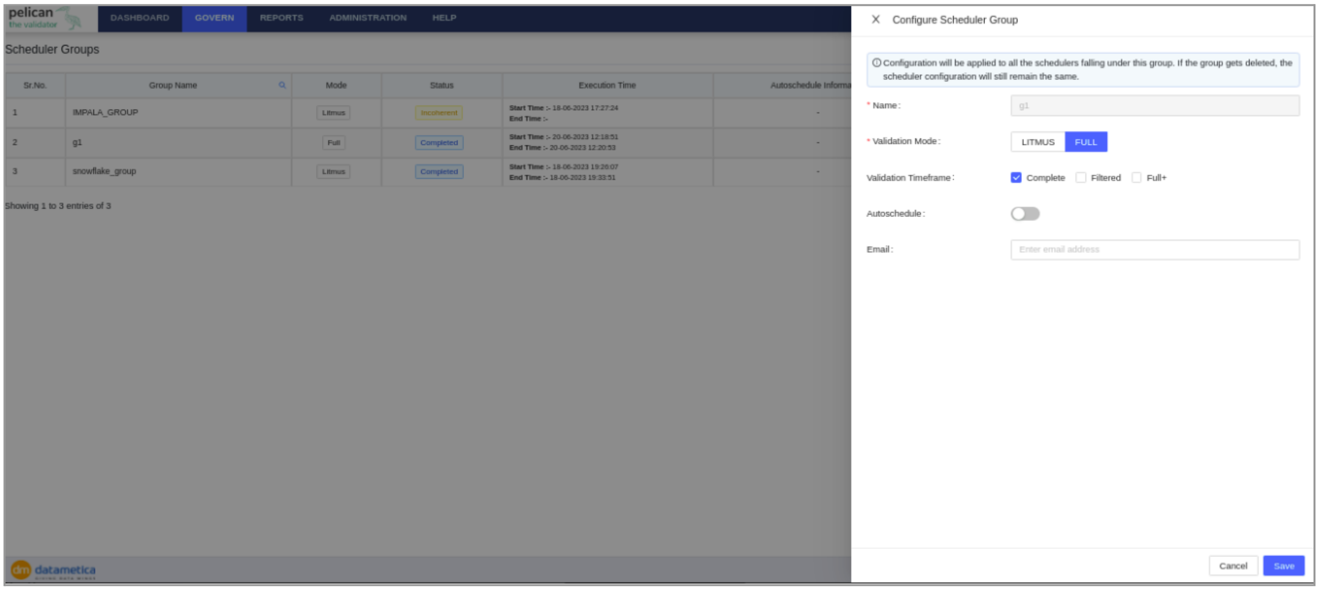
Incoherent - The schedulers have been forcefully stopped while running and some of the schedulers are incomplete.

Terminated - The schedulers have been terminated while running and all the schedulers are incomplete.

Running - The Schedulers are currently running.

IV. **Start Time / End Time**: Denotes the start and end time of the scheduler run.

V. **Configure**:



Scheduler Groups

Sr.No.	Group Name	Mode	Status	Execution Time	Autoschedule Informa
1	IMPALA_GROUP	Litmus	Incoherent	**Start Time :-** 18-06-2023 17:27:24 **End Time :-**	-
2	g1	Full	Completed	**Start Time :-** 20-06-2023 12:18:51 **End Time :-** 20-06-2023 12:20:53	-
3	snowflake_group	Litmus	Completed	**Start Time :-** 18-06-2023 19:26:07 **End Time :-** 18-06-2023 19:33:51	-

Showing 1 to 3 entries of 3

Configure Scheduler Group

Configuration will be applied to all the schedulers falling under this group. If the group gets deleted, the scheduler configuration will still remain the same.

* Name: g1

* Validation Mode: LITMUS [✓ FULL]

Validation Timeframe: ☑ Complete ☐ Filtered ☐ Full+

Autoschedule: ○

Email: Enter email address

This section enables the user to configure the scheduler group. The following entities can be configured and the changes will be applicable to all the schedulers belonging to that group:

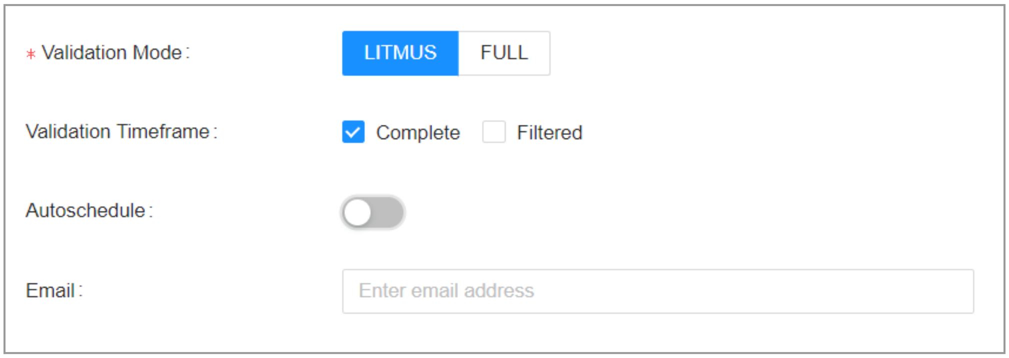
A. **Validation Mode:** The validation mode can be Litmus or Full.

B. **Validation Timeframe:** The Validation timeframe can be Complete/Filtered/Full+.

Note:

- The **'Filtered'** enables incremental validation.

  You can select the validation time frame for the scheduler execution by setting the filters for the specific time period.



* Validation Mode : **LITMUS** FULL

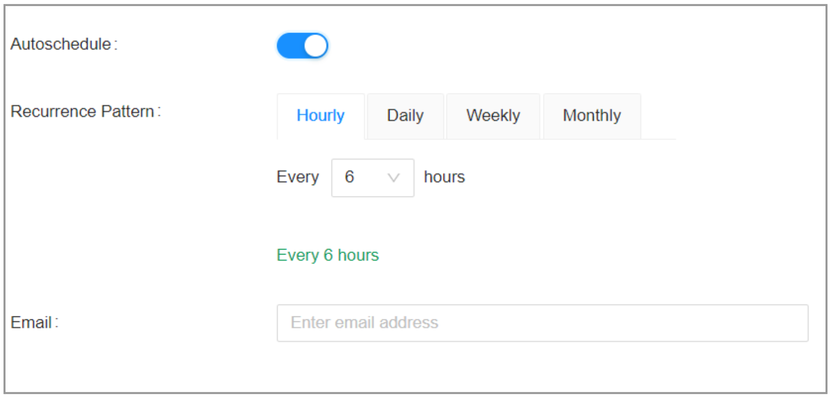
Validation Timeframe : ☑ Complete □ Filtered

Autoschedule : ○

Email : Enter email address

- The **Full+** validation provides an advantage of finding the mismatches in configured percentage of no. of total columns in a single execution for tables with more mismatching columns as compared to multiple executions of the Full mode.
- Full+ is applicable only if the **FULL** mode is selected. Also, while using the **Full+** mode either the **Complete** or **Filtered** validations should be selected. However, you cannot select **Complete** and **Filtered** together.

C. **Autoschedule**:



Autoschedule : ●

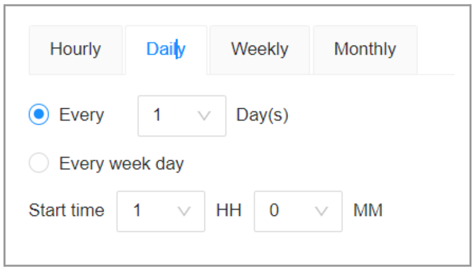
Recurrence Pattern : **Hourly** Daily Weekly Monthly

Every 6 hours

Every 6 hours

Email : Enter email address

The Autoschedule toddle button helps in setting the scheduler group's recurrence pattern. The scheduler can be set to run hourly, daily, weekly or monthly.



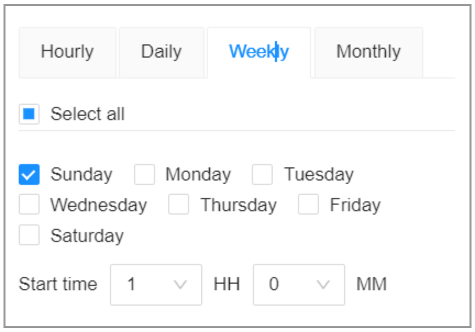
Hourly **Daily** Weekly Monthly

● Every 1 Day(s)

○ Every week day

Start time 1 HH 0 MM

Weekly

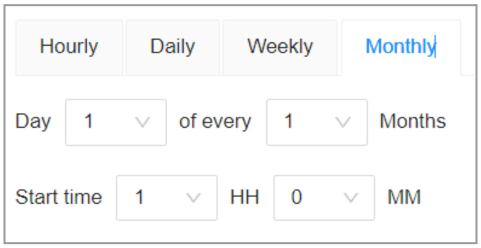


■ Select all

☑ Sunday □ Monday □ Tuesday
□ Wednesday □ Thursday □ Friday
□ Saturday

Start time 1 HH 0 MM

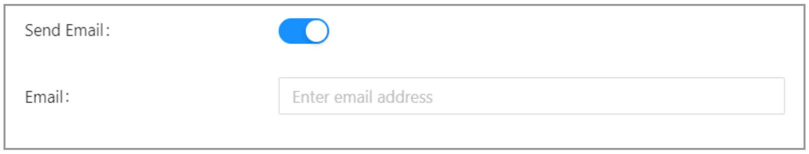
Monthly



Day 1 of every 1 Months

Start time 1 HH 0 MM

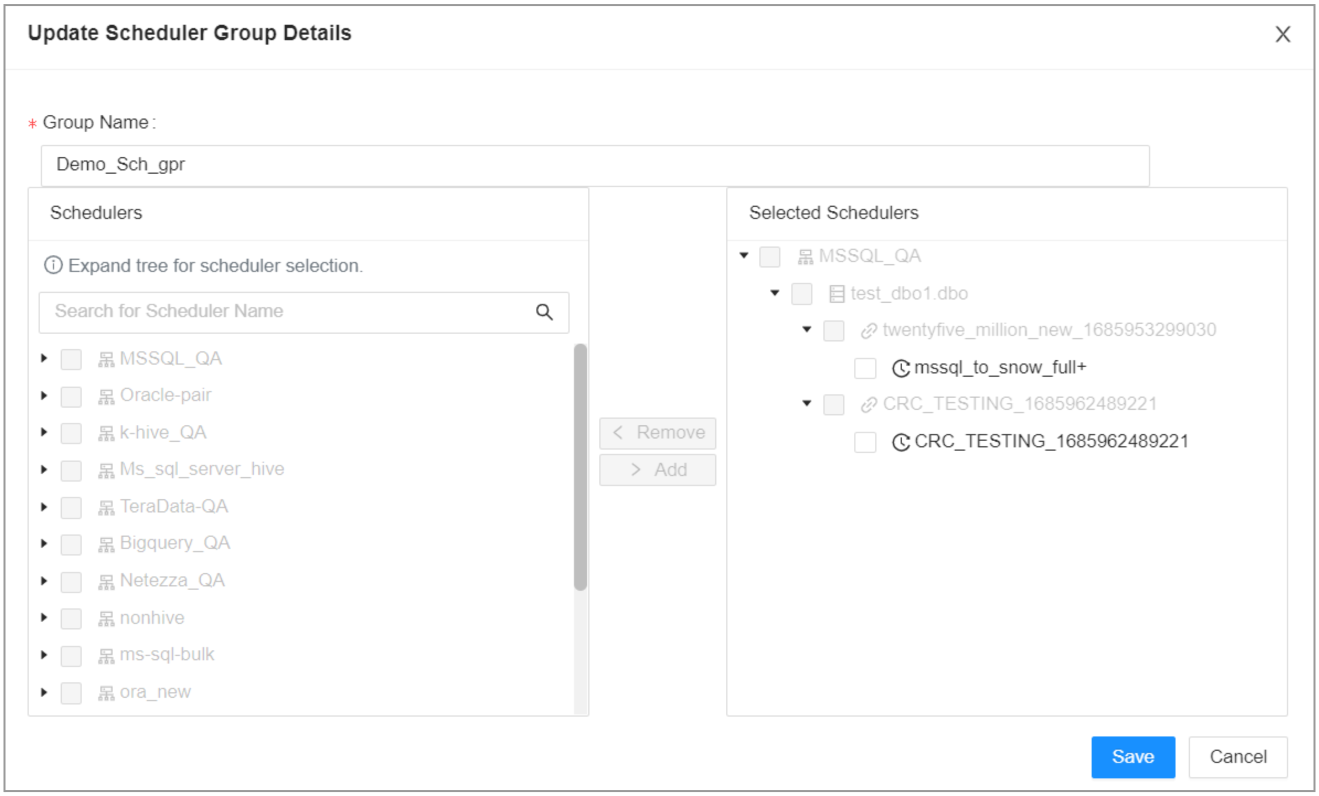
D. Send Email Toggle:


Send Email:

Email: Enter email address

This toggle button allows the user to enter an email address if the user wants reports to be sent for all the schedulers that are part of the group.

E. Edit



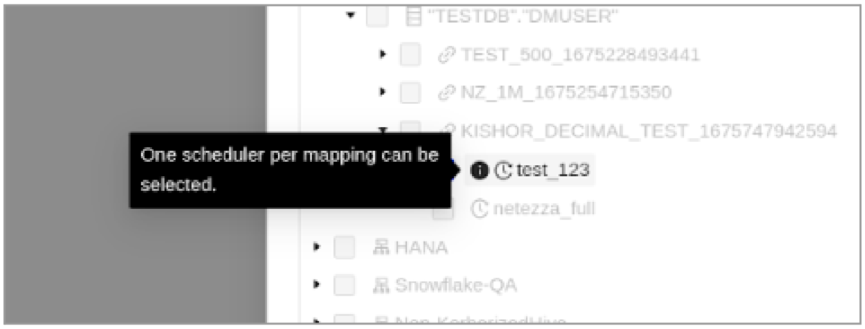
Update Scheduler Group Details

* Group Name : Demo_Sch_gpr

The Edit icon enables the user to update the existing Scheduler Group details.

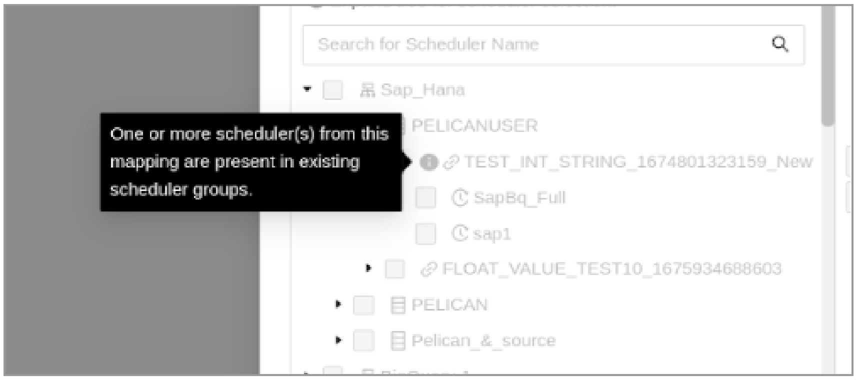
Note:

- Only one scheduler per mapping can be selected while editing the Scheduler Group.



One scheduler per mapping can be selected.

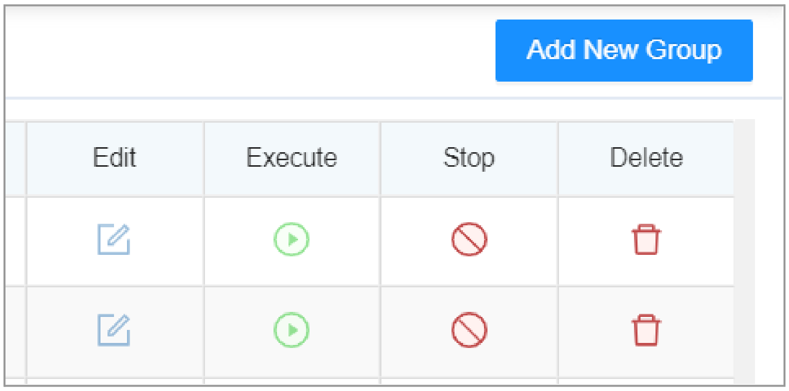
- No sibling scheduler from the existing mapping can be present in any other scheduler group.



One or more scheduler(s) from this mapping are present in existing scheduler groups.

Search for Scheduler Name
Sap_Hana
PELICANUSER
TEST_INT_STRING_1674801323159_New
SapBq_Full
sap1
FLOAT_VALUE_TEST10_1675934688603
PELICAN
Pelican_&_source

VI. **Execute**: The Execute icon enables the user to run the scheduler.



Edit	Execute	Stop	Delete
✎	▶	⊘	🗑
✎	▶	⊘	🗑

VII. **Stop:** This button provides an option to terminate a running scheduler group from the UI. Once the Group Scheduler is stopped all the schedulers which are associated with the group and are in running state should also get terminated except in the below cases:

- Individual scheduler which got completed before stopping group
- Individual scheduler which did not got started before stopping group

VIII. **Delete**: This icon enables the user to delete the scheduler group.

13. Report

13.1. Validation Result

After scheduler execution, the application automatically generates the validation result in the form of a report.

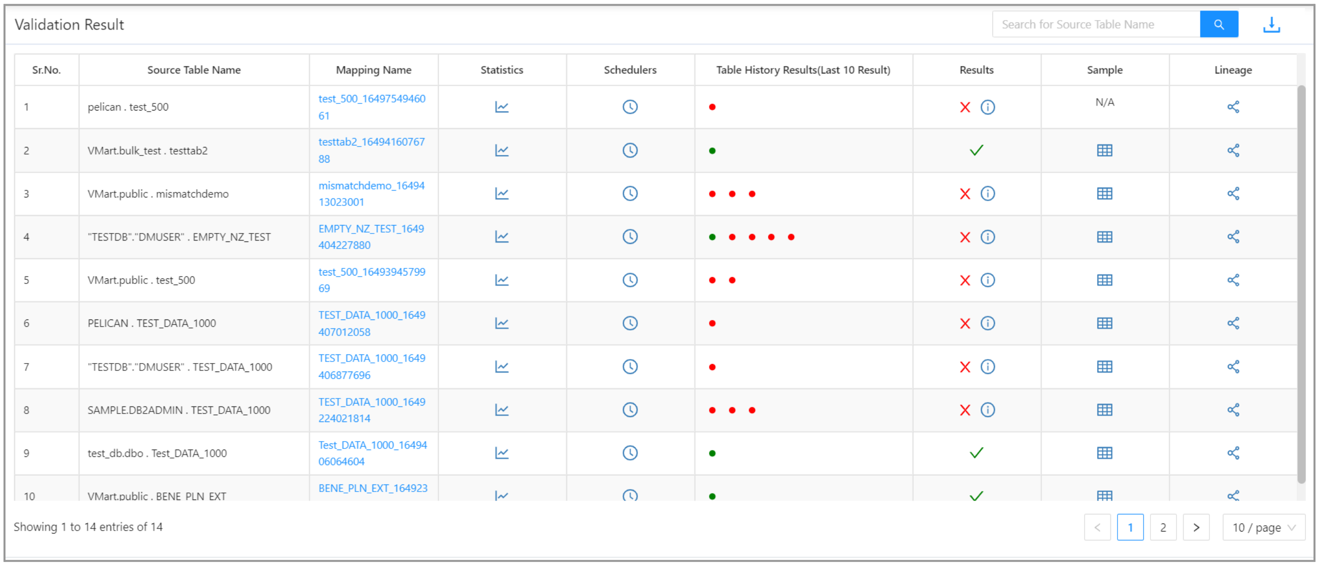
13.1.1. Table/View

13.1.1.1. Results from Report page

The steps to view the validation result:

1. Go to **Reports.**
2. Click on **Validation Result**.
3. Select **Table/View**.

   The application displays the **Validation Result** screen as shown below.



Validation Result

Sr.No.	Source Table Name	Mapping Name	Statistics	Schedulers	Table History Results(Last 10 Result)	Results	Sample	Lineage
1	pelican . test_500	test_500_1649754946061	📈	🕒	●	✕ ⓘ	N/A	⋖
2	VMart.bulk_test . testtab2	testtab2_1649416076788	📈	🕒	●	✓	▦	⋖
3	VMart.public . mismatchdemo	mismatchdemo_1649413023001	📈	🕒	● ● ●	✕ ⓘ	▦	⋖
4	"TESTDB"."DMUSER" . EMPTY_NZ_TEST	EMPTY_NZ_TEST_1649404227880	📈	🕒	● ● ● ● ●	✕ ⓘ	▦	⋖
5	VMart.public . test_500	test_500_1649394579969	📈	🕒	● ●	✕ ⓘ	▦	⋖
6	PELICAN . TEST_DATA_1000	TEST_DATA_1000_1649407012058	📈	🕒	●	✕ ⓘ	▦	⋖
7	"TESTDB"."DMUSER" . TEST_DATA_1000	TEST_DATA_1000_1649406877696	📈	🕒	●	✕ ⓘ	▦	⋖
8	SAMPLE.DB2ADMIN . TEST_DATA_1000	TEST_DATA_1000_1649224021814	📈	🕒	● ● ●	✕ ⓘ	▦	⋖
9	test_db.dbo . Test_DATA_1000	Test_DATA_1000_1649406064604	📈	🕒	●	✓	▦	⋖
10	VMart.public . BENE_PLN_EXT	BENE_PLN_EXT_164923	📈	🕒	●	✓	▦	⋖

Showing 1 to 14 entries of 14

The **Validation Result** screen displays the following information.

- **Source Table Name** : This column displays the list of source tables for which mapping has been executed in the system.
- **Mapping Name**: This column displays the mapping which has been executed for the particular table.
- **Statistics**: This column displays the scheduler history of the respective table mapping.

  Click the **View Mapping Historical Results** 📈 icon to view detailed history. This displays the detailed history of the respective scheduler as shown below.

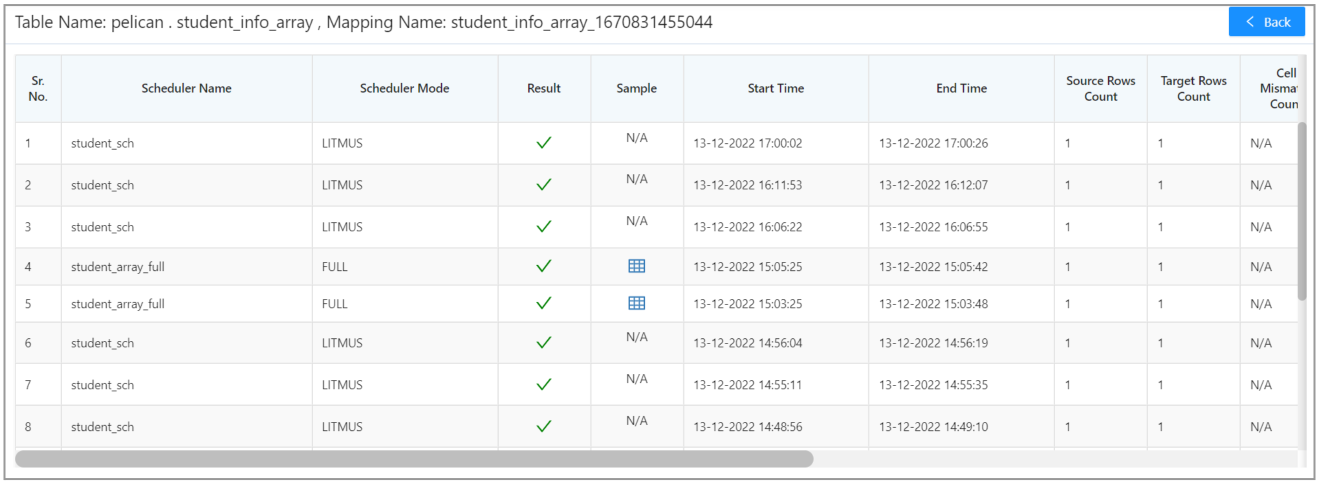


Table Name: pelican . student_info_array , Mapping Name: student_info_array_1670831455044

Sr. No.	Scheduler Name	Scheduler Mode	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Misma Coun
1	student_sch	LITMUS	✓	N/A	13-12-2022 17:00:02	13-12-2022 17:00:26	1	1	N/A
2	student_sch	LITMUS	✓	N/A	13-12-2022 16:11:53	13-12-2022 16:12:07	1	1	N/A
3	student_sch	LITMUS	✓	N/A	13-12-2022 16:06:22	13-12-2022 16:06:55	1	1	N/A
4	student_array_full	FULL	✓	▦	13-12-2022 15:05:25	13-12-2022 15:05:42	1	1	N/A
5	student_array_full	FULL	✓	▦	13-12-2022 15:03:25	13-12-2022 15:03:48	1	1	N/A
6	student_sch	LITMUS	✓	N/A	13-12-2022 14:56:04	13-12-2022 14:56:19	1	1	N/A
7	student_sch	LITMUS	✓	N/A	13-12-2022 14:55:11	13-12-2022 14:55:35	1	1	N/A
8	student_sch	LITMUS	✓	N/A	13-12-2022 14:48:56	13-12-2022 14:49:10	1	1	N/A

The scheduler history page displays the detailed information of the scheduler, such as:

- Scheduler name
- Result
- Sample
- Start time
- End time
- Source Rows Count
- Target Rows Count
- Cell Mismatch Cout
- Extra Rows Count
- Missing Rows Count
- Total Mismatch RowsCount
- Mapped Columns
- Validation Size
- Approve
- Exceptions

Among these following are the some of the major features: -

➢ **Approve**

- It is a feature to approve a mapping if a mapping fails.
- This feature has been added for the cases where the cause of failure is a known issue and can be considered as an exceptional case.
- As shown below we can approve a mapping by clicking on the icon in the approve column, state the reason for approval and click save to approve a mapping.

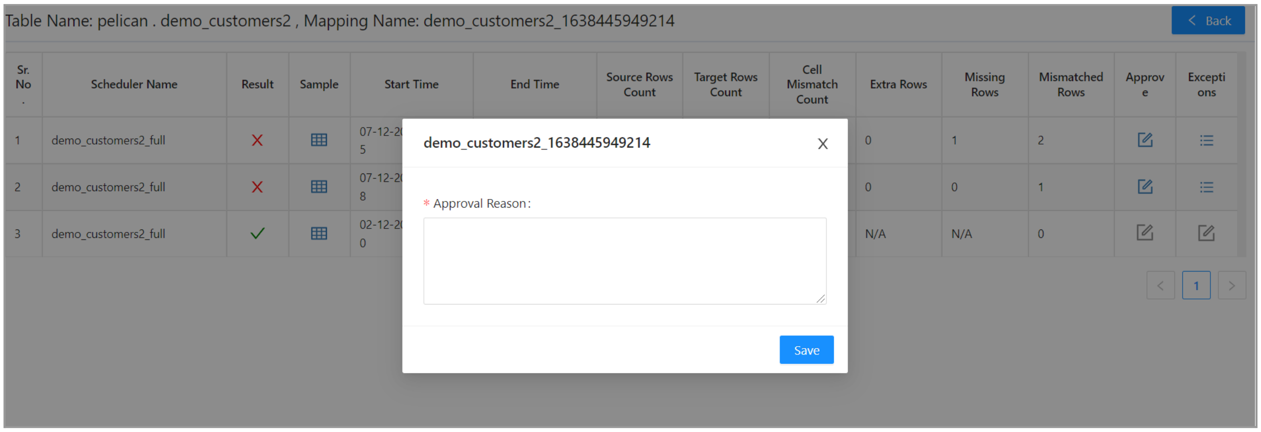


Table Name: pelican . demo_customers2 , Mapping Name: demo_customers2_1638445949214

Sr. No .	Scheduler Name	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra Rows	Missing Rows	Mismatched Rows	Approv e	Excepti ons
1	demo_customers2_full	✗	▦	07-12-20 5					0	1	2	✎	☰
2	demo_customers2_full	✗	▦	07-12-20 8					0	0	1	✎	☰
3	demo_customers2_full	✓	▦	02-12-20 0					N/A	N/A	0	✎	✎

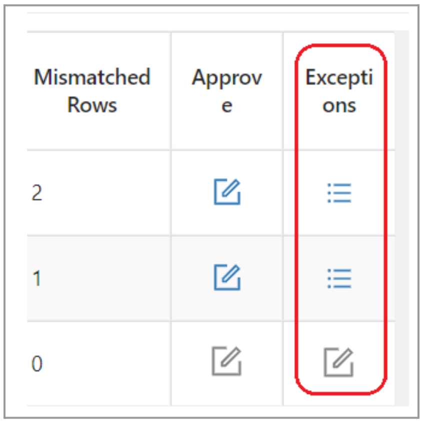
demo_customers2_1638445949214

* Approval Reason :

- If the result is either approved or passed, the icon to open approve a mapping modal will be disabled.

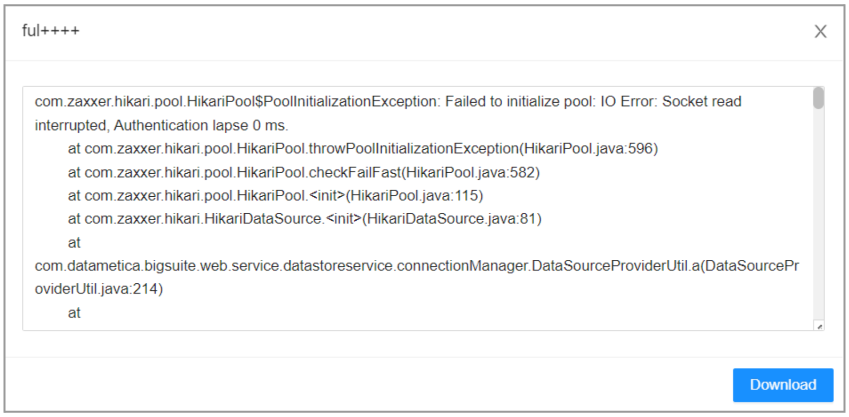
➢ Exceptions

- If a scheduler failed due to an exception, then log will display its exception stack trace. Click on the logs icon to view exception logs.



Mismatched Rows	Approv e	Excepti ons
2	✎	☰
1	✎	☰
0	✎	✎

- Once we click the icon on the logs column, the new pop-up window will open for the logs.

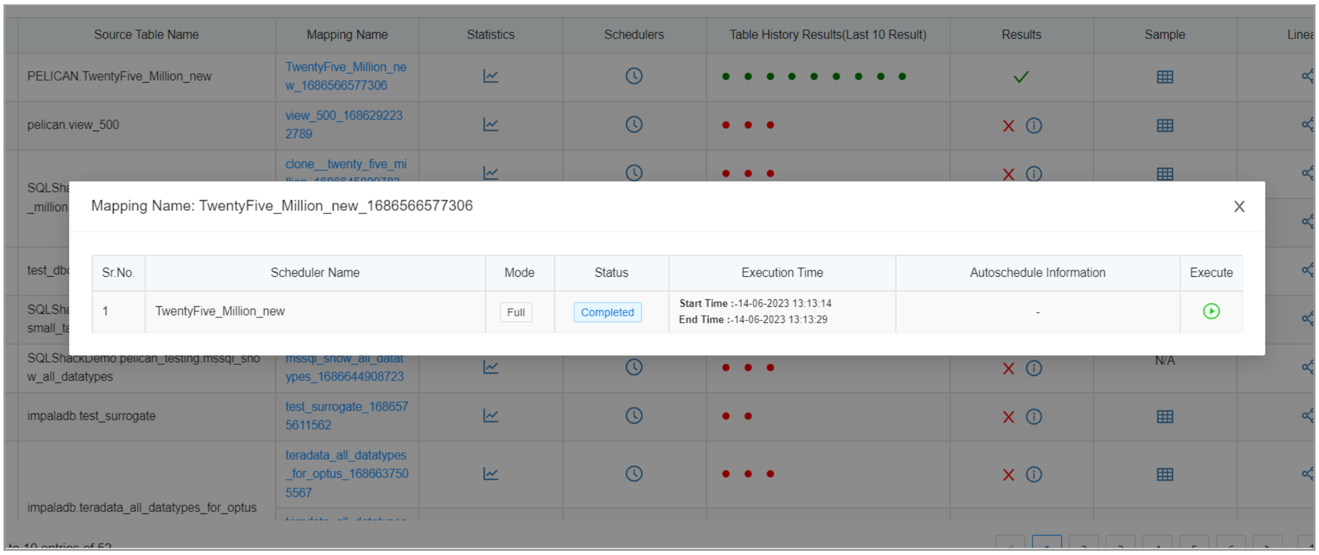


ful++++

```
com.zaxxer.hikari.pool.HikariPool$PoolInitializationException: Failed to initialize pool: IO Error: Socket read
interrupted, Authentication lapse 0 ms.
	at com.zaxxer.hikari.pool.HikariPool.throwPoolInitializationException(HikariPool.java:596)
	at com.zaxxer.hikari.pool.HikariPool.checkFailFast(HikariPool.java:582)
	at com.zaxxer.hikari.pool.HikariPool.<init>(HikariPool.java:115)
	at com.zaxxer.hikari.HikariDataSource.<init>(HikariDataSource.java:81)
	at
com.datametica.bigsuite.web.service.datastoreservice.connectionManager.DataSourceProviderUtil.a(DataSourcePr
oviderUtil.java:214)
	at
```

- If Scheduler runs without Exception, then the Log option will be disabled.

- ➢ **Schedulers:** Click on the icon to view scheduler details.



Source Table Name	Mapping Name	Statistics	Schedulers	Table History Results(Last 10 Result)	Results	Sample	Linea
PELICAN.TwentyFive_Million_new	TwentyFive_Million_new_1686566577306			● ● ● ● ● ● ● ● ●	✓	▦	
pelican.view_500	view_500_1686292232789			● ● ●	✗ ⓘ	▦	
SQLSha… _million	clone__twenty_five_mi…			● ● ●	✗ ⓘ	▦	
test_dbo…							
SQLSha… small_ta…							
SQLShackDemo.pelican_testing.mssql_show_all_datatypes	mssql_show_all_datatypes_1686644908723			● ● ●	✗ ⓘ	N/A	
impaladb.test_surrogate	test_surrogate_1686575611562			● ●	✗ ⓘ	▦	
impaladb.teradata_all_datatypes_for_optus	teradata_all_datatypes_for_optus_1686637505567			● ● ●	✗ ⓘ	▦	

Mapping Name: TwentyFive_Million_new_1686566577306

Sr.No.	Scheduler Name	Mode	Status	Execution Time	Autoschedule Information	Execute
1	TwentyFive_Million_new	Full	Completed	**Start Time** :-14-06-2023 13:13:14 **End Time** :-14-06-2023 13:13:29	-	▶

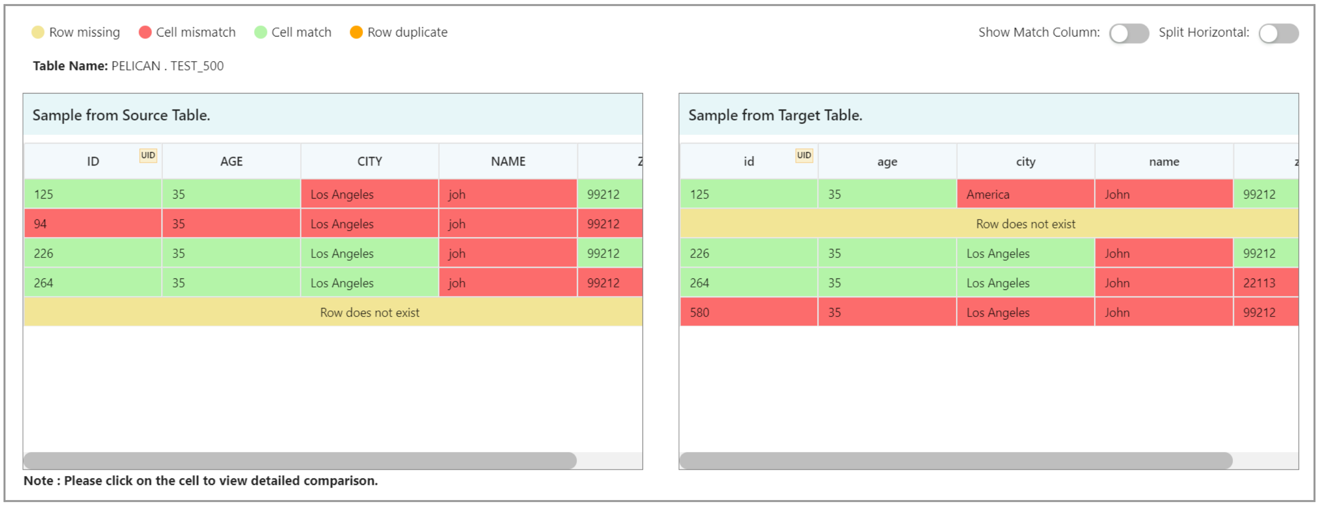
The screen will pop up and will show information and properties of the scheduler such as scheduler name, scheduler running time, on schedule, starting and ending time of last execution of scheduler, execution option of scheduler using Execute Now button and status of the execution.

- ➢ **Table Historical Result**: This column displays the execution history of the schedulers of the respective mapping. It shows the last 10 execution results. This information is represented by **Green**, **Red** and **Blue** circles. The **Green** circle indicates the **Success**, the Red circle indicates the **Failure,** the Blue circle indicates the **Approve**. The user can hover mouse over these circles to view execution start date and end date of the respective scheduler as shown below.
- ➢ **Results:** This column displays the data validation result. The Result is represented by two flags, namely:

  - Success ✓
  - Failure ✗

  Hover mouse over icon to know the reason for failure.

- ➢ **Sample:** This column displays the sample of mismatch data in case the total mismatch count is greater than zero. Click the ▦ symbol to view samples of mismatched data. This displays the **Sample Data** screen as shown below:



● Row missing ● Cell mismatch ● Cell match ● Row duplicate

Show Match Column: Split Horizontal:

Table Name: PELICAN . TEST_500

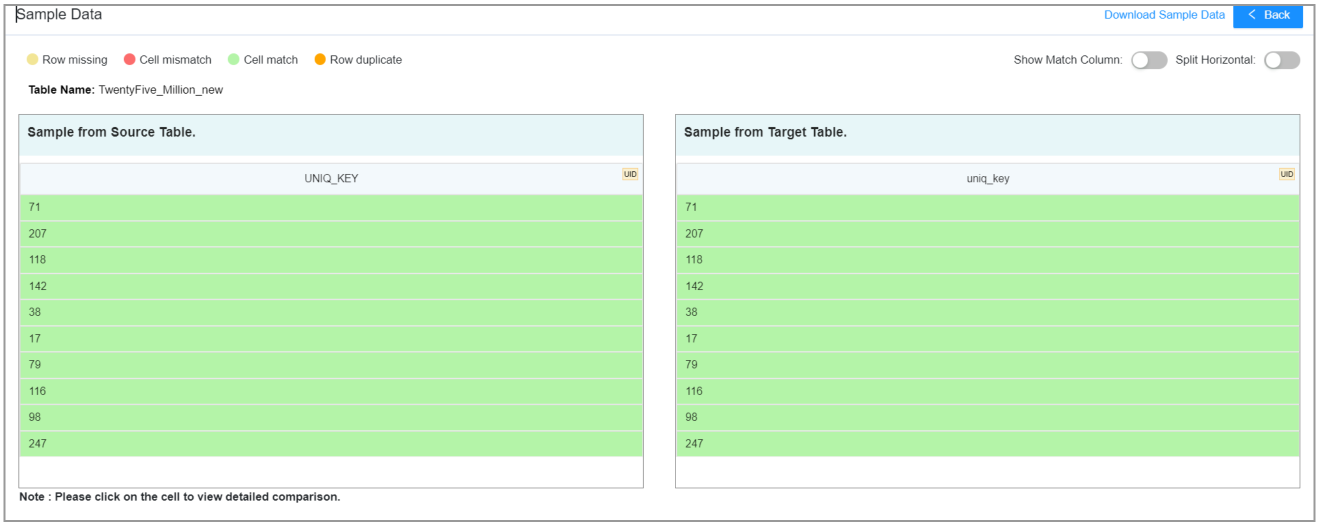
Sample from Source Table.

ID UID	AGE	CITY	NAME	Z
125	35	Los Angeles	joh	99212
94	35	Los Angeles	joh	99212
226	35	Los Angeles	joh	99212
264	35	Los Angeles	joh	99212
Row does not exist				

Sample from Target Table.

id UID	age	city	name	z
125	35	America	John	99212
Row does not exist				
226	35	Los Angeles	John	99212
264	35	Los Angeles	John	22113
580	35	Los Angeles	John	99212

Note : Please click on the cell to view detailed comparison.



Sample Data

● Row missing ● Cell mismatch ● Cell match ● Row duplicate

Show Match Column: Split Horizontal:

Table Name: TwentyFive_Million_new

Sample from Source Table.

UNIQ_KEY UID
71
207
118
142
38
17
79
116
98
247

Sample from Target Table.

uniq_key UID
71
207
118
142
38
17
79
116
98
247

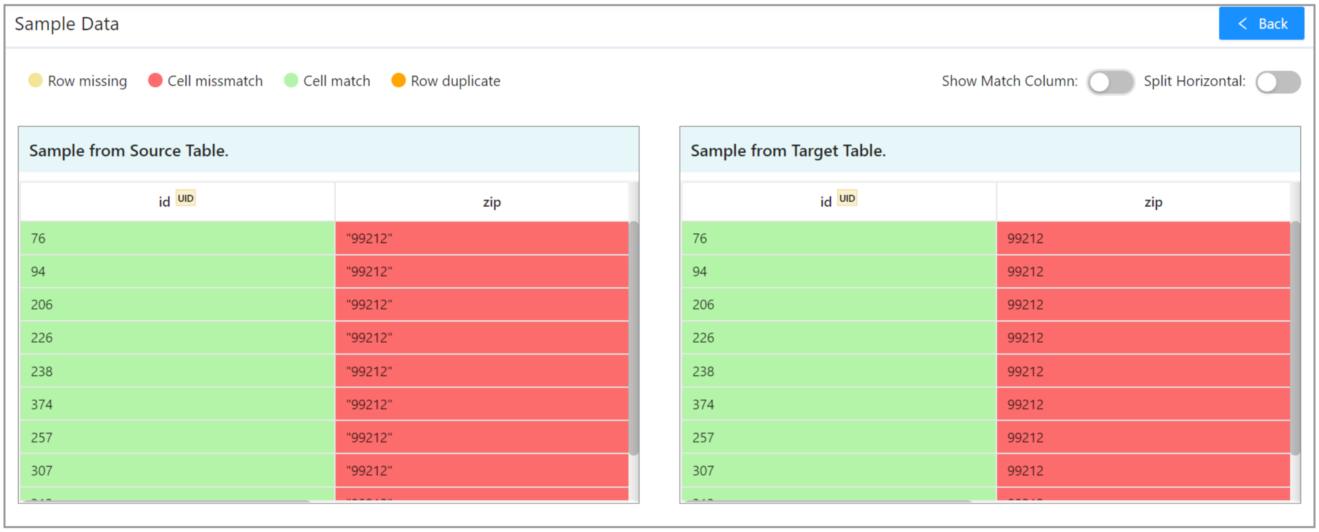
Note : Please click on the cell to view detailed comparison.

The Sample Data report displays sample mismatch data from source and target table with different colors. This report also provides the following filters which helps the user to view the desired information:

- Show Match Column
- Split Horizontal

Show Match Column: The Hide/Show Match Column is a toggle button which shows two States ON and OFF.

When the **Hide/Show Match Column** toggle button is set to **OFF**, then the application displays both matched and unmatched columns as shown below.



Sample Data

< Back

● Row missing ● Cell missmatch ● Cell match ● Row duplicate

Show Match Column: ○ Split Horizontal: ○

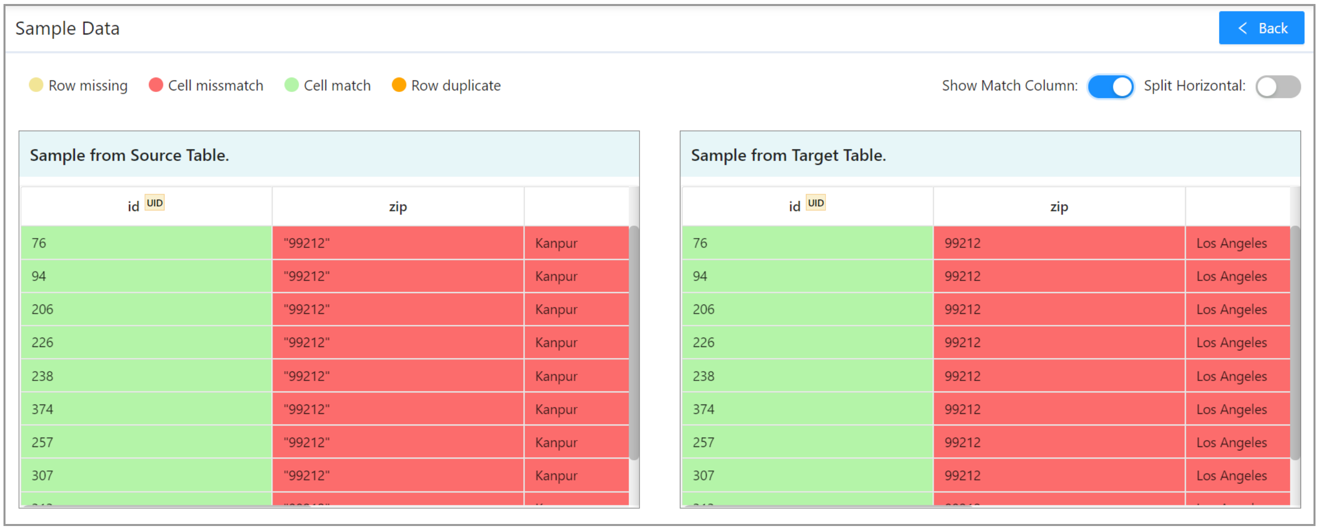
Sample from Source Table.

id UID	zip
76	"99212"
94	"99212"
206	"99212"
226	"99212"
238	"99212"
374	"99212"
257	"99212"
307	"99212"

Sample from Target Table.

id UID	zip
76	99212
94	99212
206	99212
226	99212
238	99212
374	99212
257	99212
307	99212

When the **Hide/Show Match Column** toggle button is set to **ON**, then the application displays only unmatched columns as shown below.



Sample Data

< Back

● Row missing ● Cell missmatch ● Cell match ● Row duplicate

Show Match Column: ● Split Horizontal: ○

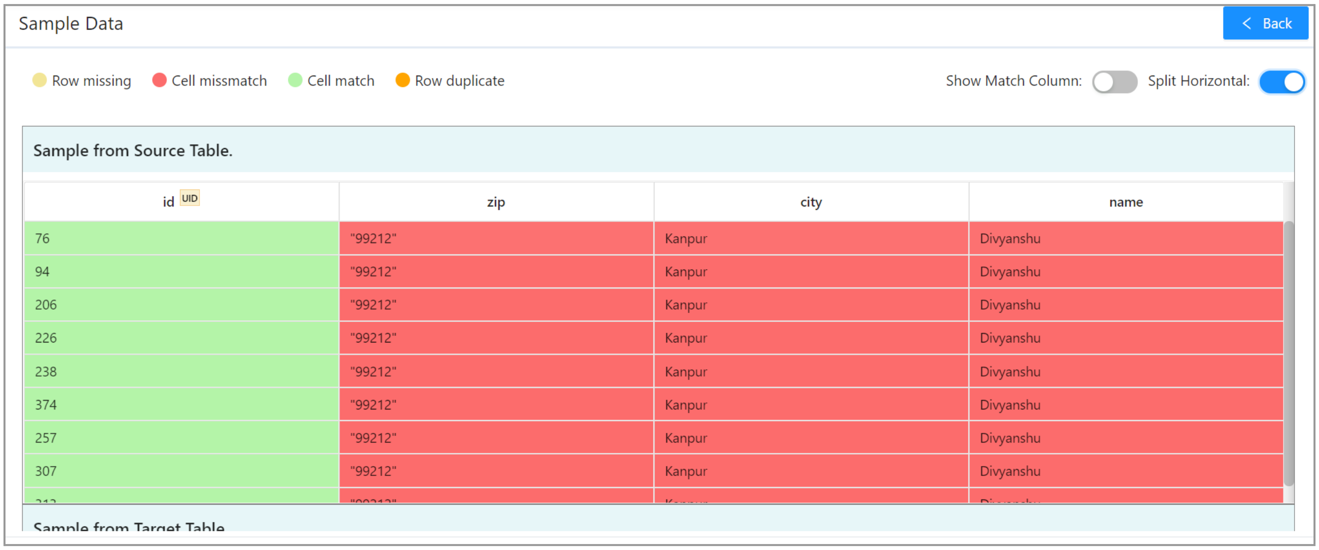
Sample from Source Table.

id UID	zip	
76	"99212"	Kanpur
94	"99212"	Kanpur
206	"99212"	Kanpur
226	"99212"	Kanpur
238	"99212"	Kanpur
374	"99212"	Kanpur
257	"99212"	Kanpur
307	"99212"	Kanpur

Sample from Target Table.

id UID	zip	
76	99212	Los Angeles
94	99212	Los Angeles
206	99212	Los Angeles
226	99212	Los Angeles
238	99212	Los Angeles
374	99212	Los Angeles
257	99212	Los Angeles
307	99212	Los Angeles

Split Horizontal: Set **Split Horizontal** toggle button to **ON** to view mismatched data in the horizontal format, for more information, refer below image.



Sample Data

🟡 Row missing 🔴 Cell missmatch 🟢 Cell match 🟠 Row duplicate

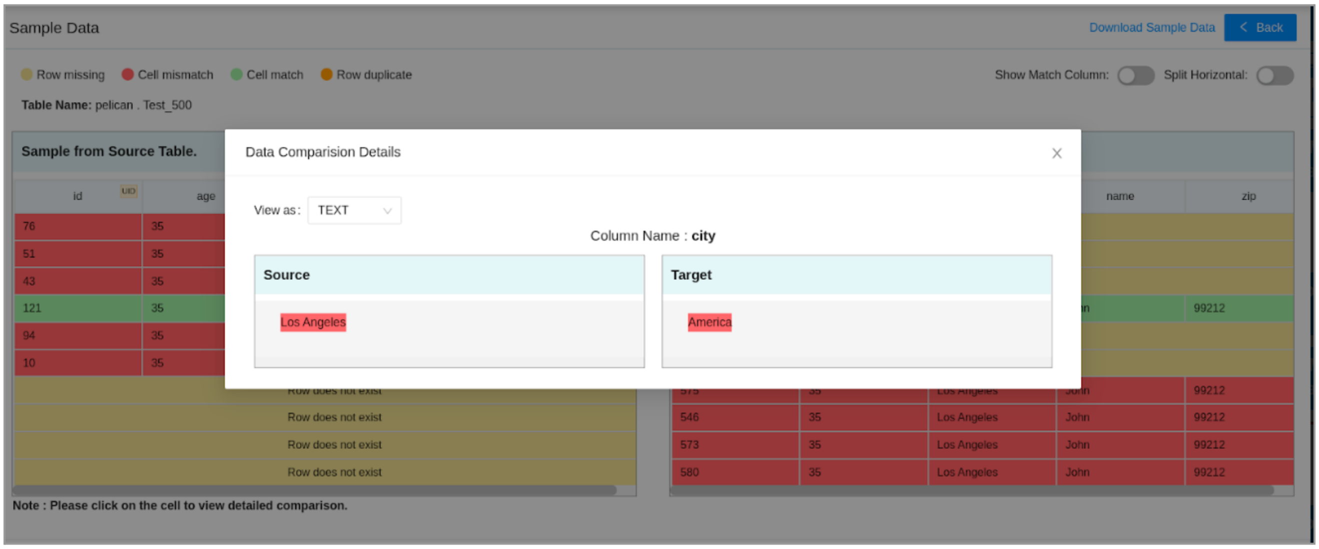
Show Match Column: ○ Split Horizontal: ●

Sample from Source Table.

id UID	zip	city	name
76	"99212"	Kanpur	Divyanshu
94	"99212"	Kanpur	Divyanshu
206	"99212"	Kanpur	Divyanshu
226	"99212"	Kanpur	Divyanshu
238	"99212"	Kanpur	Divyanshu
374	"99212"	Kanpur	Divyanshu
257	"99212"	Kanpur	Divyanshu
307	"99212"	Kanpur	Divyanshu

Sample from Target Table

View Sample Data



Sample Data

🟡 Row missing 🔴 Cell mismatch 🟢 Cell match 🟠 Row duplicate

Show Match Column: ○ Split Horizontal: ○

Table Name: pelican . Test_500

Sample from Source Table.

id UID	age
76	35
51	35
43	35
121	35
94	35
10	35
Row does not exist	
Row does not exist	
Row does not exist	
Row does not exist	

			name	zip
			n	99212
	35	Los Angeles	John	99212
546	35	Los Angeles	John	99212
573	35	Los Angeles	John	99212
580	35	Los Angeles	John	99212

Data Comparision Details ×

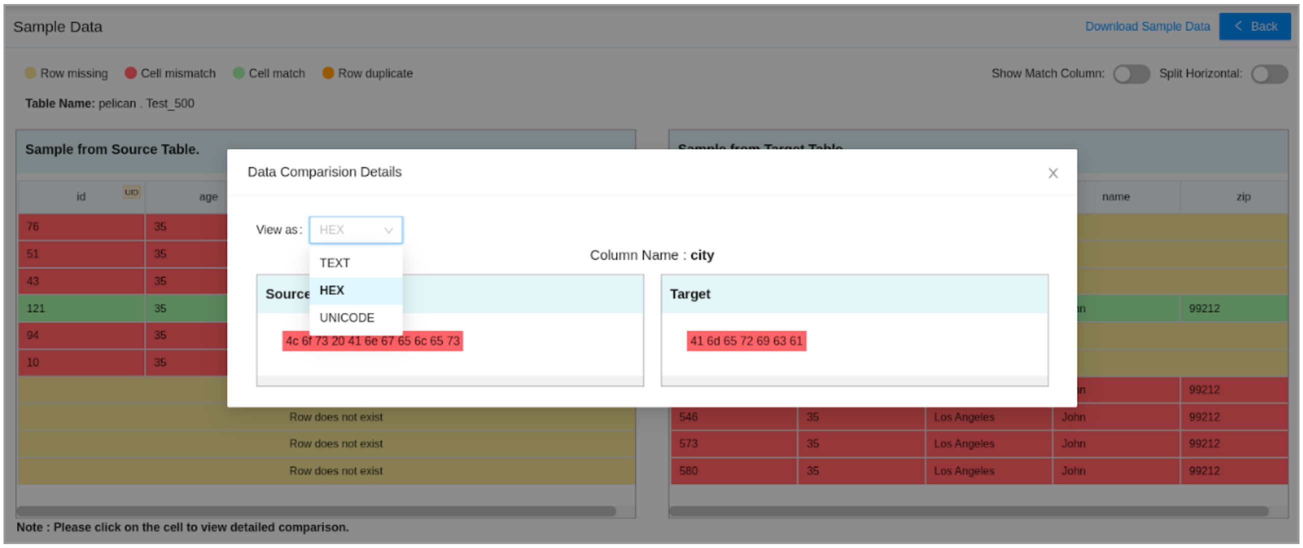
View as : TEXT

Column Name : **city**

Source	Target
Los Angeles	America

Note : Please click on the cell to view detailed comparison.

- This feature enables you to view the actual data of the compared record. Click on any cell to view its corresponding detailed data comparison.
- The data can also be viewed in the form of Text, Hex and Unicode to check for special characters by clicking on the **View as** dropdown.



Sample Data

Download Sample Data | < Back

● Row missing ● Cell mismatch ● Cell match ● Row duplicate

Show Match Column: ○ Split Horizontal: ○

Table Name: pelican . Test_500

Sample from Source Table.

id UID	age
76	35
51	35
43	35
121	35
94	35
10	35
Row does not exist	
Row does not exist	
Row does not exist	

Sample from Target Table.

			name	zip
			n	99212
			n	99212
546	35	Los Angeles	John	99212
573	35	Los Angeles	John	99212
580	35	Los Angeles	John	99212

Data Comparision Details ×

View as : HEX

TEXT
HEX
UNICODE

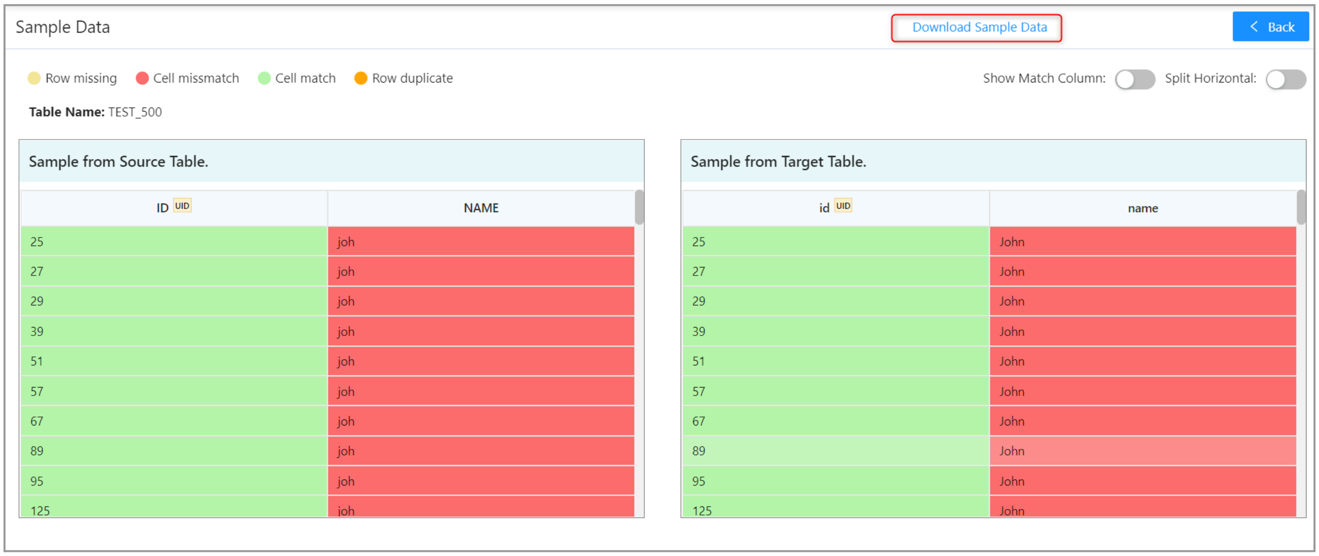
Column Name : **city**

Source	Target
4c 6f 73 20 41 6e 67 65 6c 65 73	41 6d 65 72 69 63 61

Note : Please click on the cell to view detailed comparison.

Download Sample Data

The **Download Sample Data** link enables the user to download the sample data and mapping details of the mismatched samples into an excel sheet. It is recommended to configure up to 100 mismatch samples.



Sample Data

Download Sample Data | < Back

● Row missing ● Cell missmatch ● Cell match ● Row duplicate

Show Match Column: ○ Split Horizontal: ○

Table Name: TEST_500

Sample from Source Table.

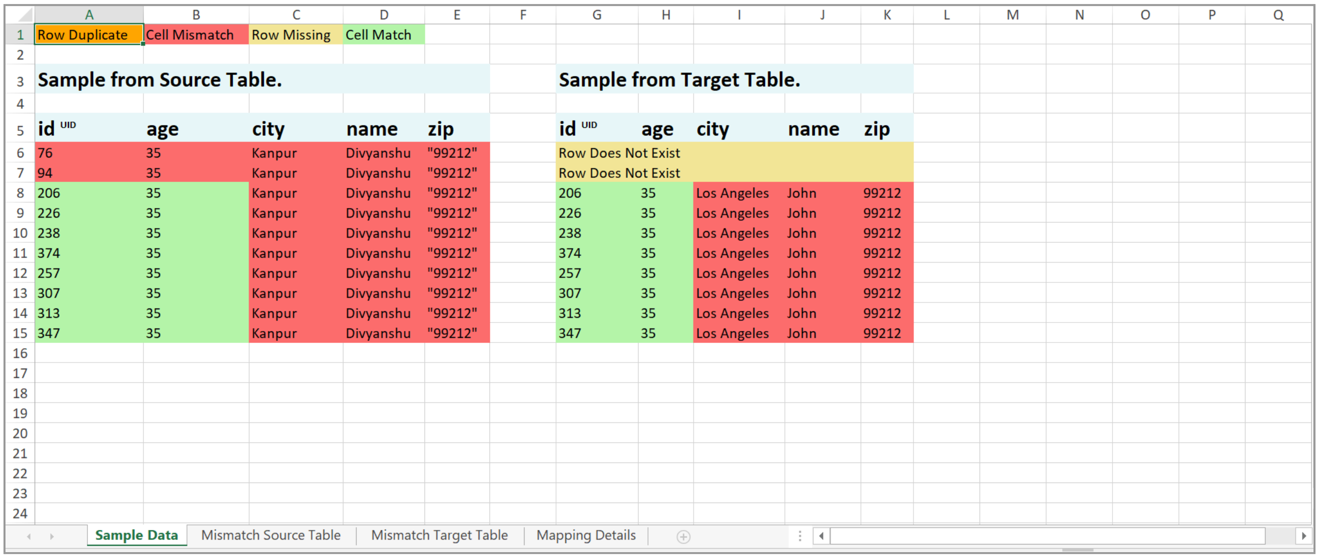
ID UID	NAME
25	joh
27	joh
29	joh
39	joh
51	joh
57	joh
67	joh
89	joh
95	joh
125	joh

Sample from Target Table.

id UID	name
25	John
27	John
29	John
39	John
51	John
57	John
67	John
89	John
95	John
125	John

The downloaded file displays the entire sample data as it is shown on the Pelican screen. It consists of the following tabs to display the sample details:

- **Sample Data**: It displays a sample of the mismatched data found in the Source and Target tables.
- **Mismatch Source Table:** It displays the mismatched data from the Source table.
- **Mismatch Target Table:** It displays the mismatched data from the Target table.
- **Mapping Details:** It displays the mapping details such as mapping name, scheduler name, source database name, target database name, etc.



	A	B	C	D	E	F	G	H	I	J	K
1	Row Duplicate	Cell Mismatch	Row Missing	Cell Match							
2											
3	Sample from Source Table.						Sample from Target Table.				
4											
5	id UID	age	city	name	zip		id UID	age	city	name	zip
6	76	35	Kanpur	Divyanshu	"99212"		Row Does Not Exist				
7	94	35	Kanpur	Divyanshu	"99212"		Row Does Not Exist				
8	206	35	Kanpur	Divyanshu	"99212"		206	35	Los Angeles	John	99212
9	226	35	Kanpur	Divyanshu	"99212"		226	35	Los Angeles	John	99212
10	238	35	Kanpur	Divyanshu	"99212"		238	35	Los Angeles	John	99212
11	374	35	Kanpur	Divyanshu	"99212"		374	35	Los Angeles	John	99212
12	257	35	Kanpur	Divyanshu	"99212"		257	35	Los Angeles	John	99212
13	307	35	Kanpur	Divyanshu	"99212"		307	35	Los Angeles	John	99212
14	313	35	Kanpur	Divyanshu	"99212"		313	35	Los Angeles	John	99212
15	347	35	Kanpur	Divyanshu	"99212"		347	35	Los Angeles	John	99212

- **Lineage**: This column displays the lineage details of the data.

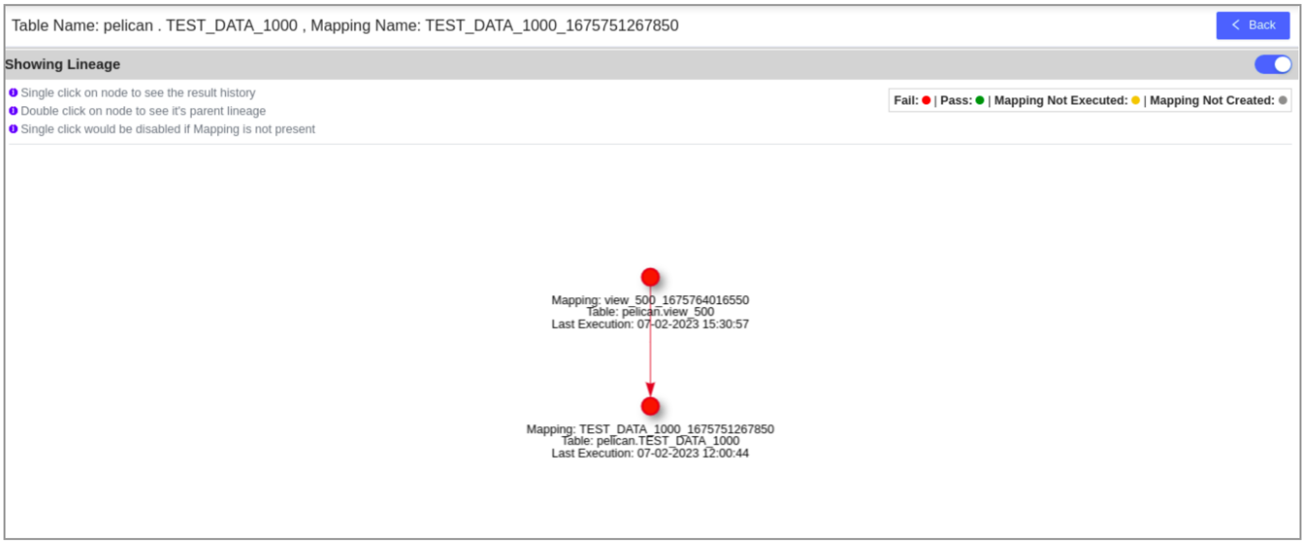


Table Name: pelican . TEST_DATA_1000 , Mapping Name: TEST_DATA_1000_1675751267850

Showing Lineage

- Single click on node to see the result history
- Double click on node to see it's parent lineage
- Single click would be disabled if Mapping is not present

Fail: ● | Pass: ● | Mapping Not Executed: ● | Mapping Not Created: ●

Mapping: view_500_1675764016550
Table: pelican.view_500
Last Execution: 07-02-2023 15:30:57

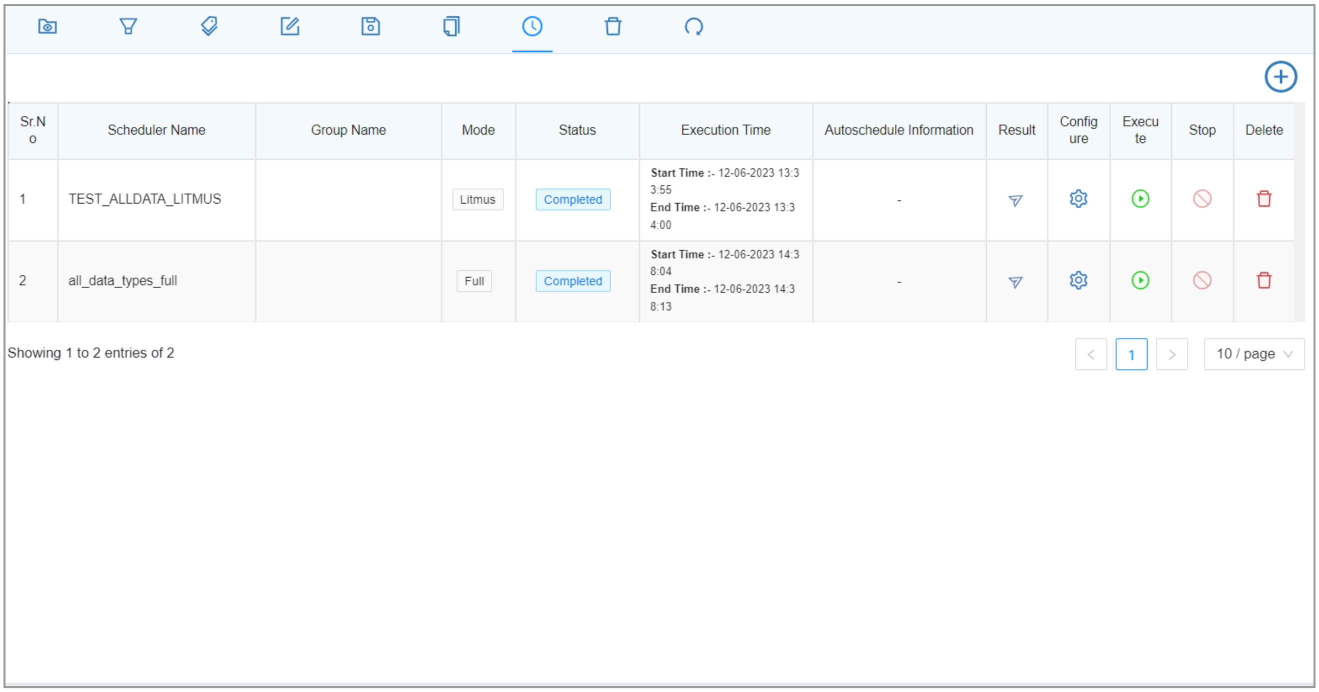
Mapping: TEST_DATA_1000_1675751267850
Table: pelican.TEST_DATA_1000
Last Execution: 07-02-2023 12:00:44

13.1.1.2. Results from Scheduler page

Here, the user can view validation results for the saved mapping.

To view validation result:

1. Go to the **Validation Configuration** page and select the saved mapping for which you want to see the validation result and then click on the **View scheduler** icon ⏱.The following screen will be displayed.



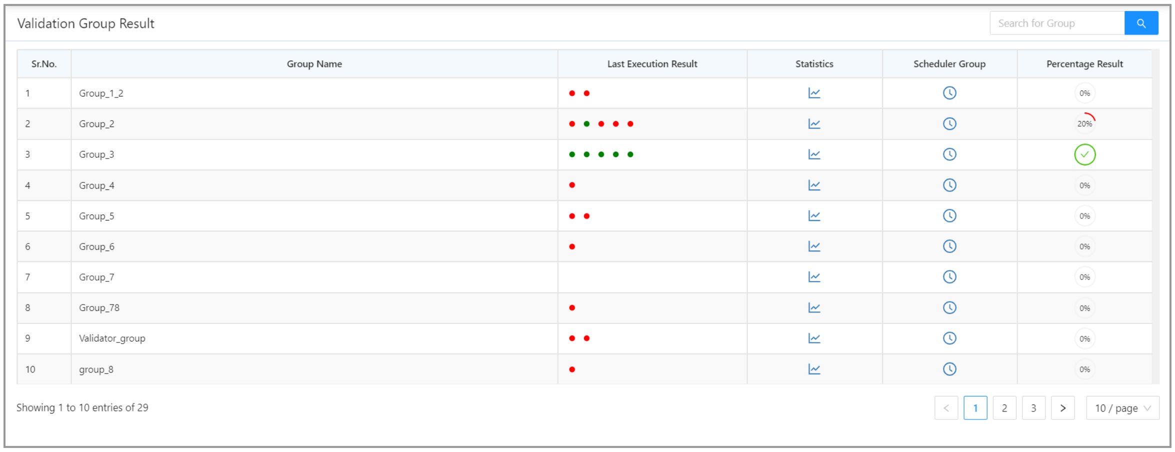
Sr.No	Scheduler Name	Group Name	Mode	Status	Execution Time	Autoschedule Information	Result	Configure	Execute	Stop	Delete
1	TEST_ALLDATA_LITMUS		Litmus	Completed	**Start Time :-** 12-06-2023 13:33:55 **End Time :-** 12-06-2023 13:34:00	-	◁	⚙	▶	⊘	🗑
2	all_data_types_full		Full	Completed	**Start Time :-** 12-06-2023 14:38:04 **End Time :-** 12-06-2023 14:38:13	-	◁	⚙	▶	⊘	🗑

Showing 1 to 2 entries of 2

2. Click on the icon in the **Result** column to see the statistics of the validation result page for the mapping.

   The fields (columns of the validation result page are the same as explained above).

13.1.2. Group (**Validation Group Result**)



Validation Group Result

Sr.No.	Group Name	Last Execution Result	Statistics	Scheduler Group	Percentage Result
1	Group_1_2	🔴 🔴	📈	🕒	0%
2	Group_2	🔴 🟢 🔴 🔴 🔴	📈	🕒	20%
3	Group_3	🟢 🟢 🟢 🟢 🟢	📈	🕒	✓
4	Group_4	🔴	📈	🕒	0%
5	Group_5	🔴 🔴	📈	🕒	0%
6	Group_6	🔴	📈	🕒	0%
7	Group_7		📈	🕒	0%
8	Group_78	🔴	📈	🕒	0%
9	Validator_group	🔴 🔴	📈	🕒	0%
10	group_8	🔴	📈	🕒	0%

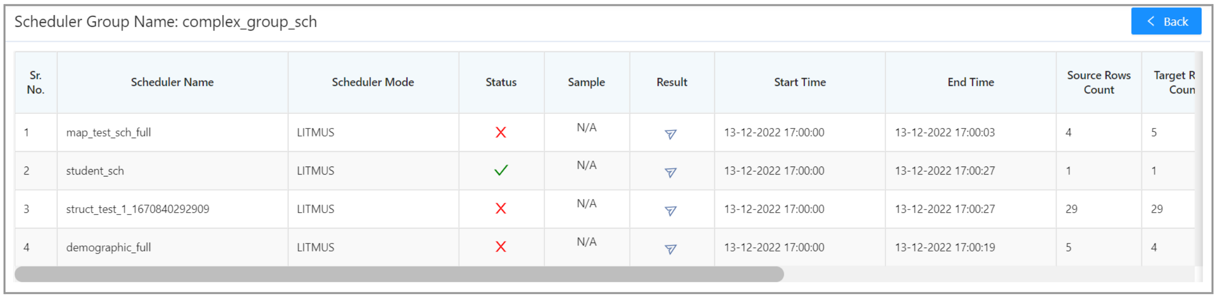
Showing 1 to 10 entries of 29

The **Validation Group Result** page displays the results of the Scheuler groups. It consists of following columns:

I. **Group Name**: Displays the name of the group.

II. **Last Execution Result**: This column displays the last execution results of all the schedulers belonging to that group. The red circle denotes data validation failure and the green circle denotes success. Hovering the mouse over the corresponding circle displays the scheduler name.

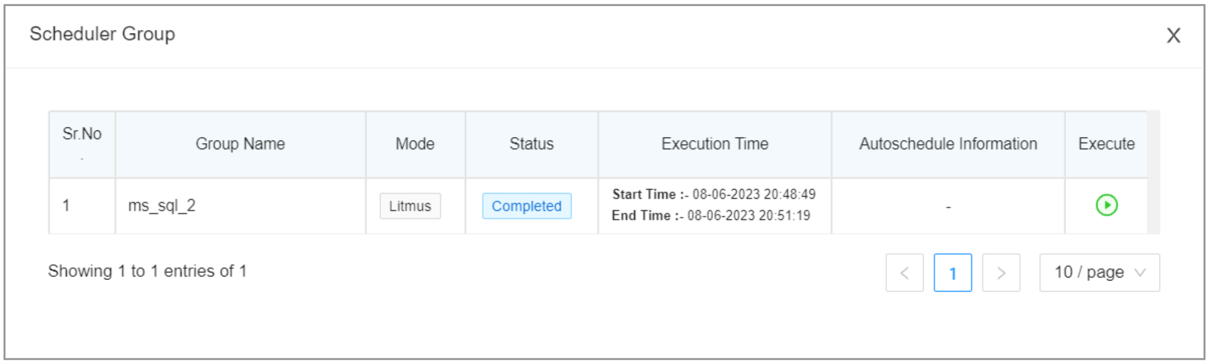
III. **Statistics**: The statistics page displays the last execution summary details of all the schedulers in the group.



Scheduler Group Name: complex_group_sch

Sr. No.	Scheduler Name	Scheduler Mode	Status	Sample	Result	Start Time	End Time	Source Rows Count	Target R Coun
1	map_test_sch_full	LITMUS	✗	N/A	▽	13-12-2022 17:00:00	13-12-2022 17:00:03	4	5
2	student_sch	LITMUS	✓	N/A	▽	13-12-2022 17:00:00	13-12-2022 17:00:27	1	1
3	struct_test_1_1670840292909	LITMUS	✗	N/A	▽	13-12-2022 17:00:00	13-12-2022 17:00:27	29	29
4	demographic_full	LITMUS	✗	N/A	▽	13-12-2022 17:00:00	13-12-2022 17:00:19	5	4

IV. **Scheduler Group**: Click on the Show Scheduler Group icon to view the details of the corresponding Scheduler group.



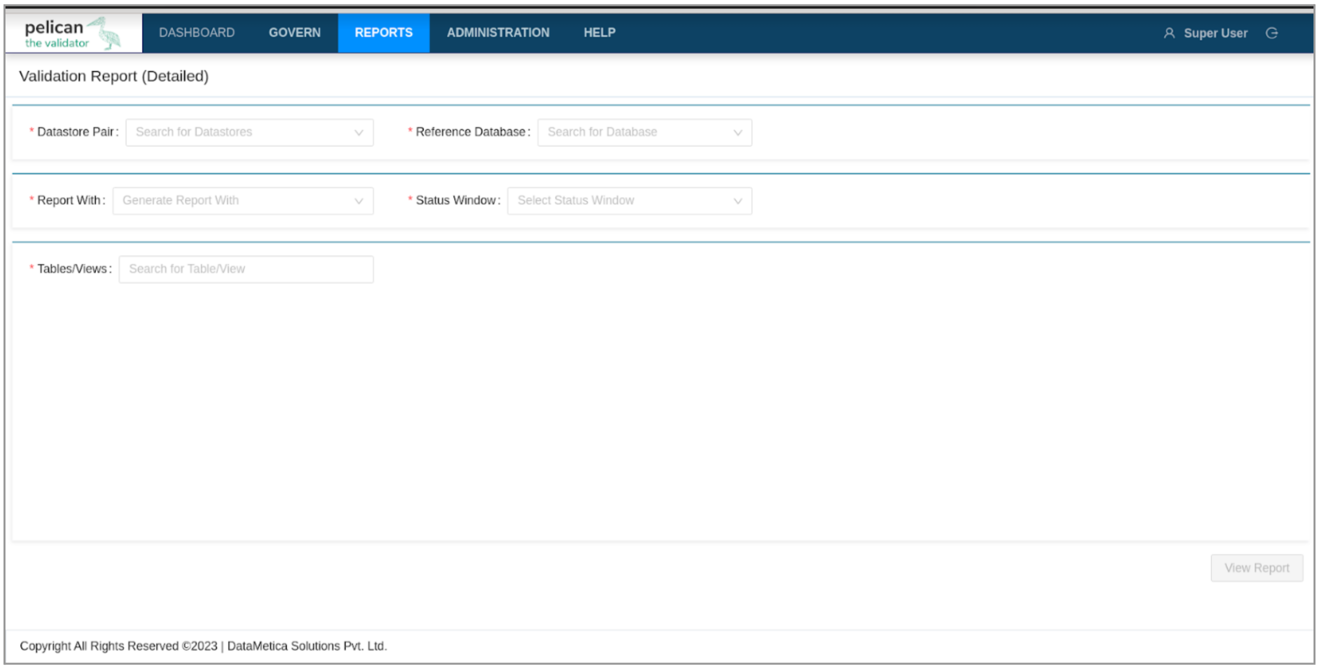
Scheduler Group

Sr.No	Group Name	Mode	Status	Execution Time	Autoschedule Information	Execute
1	ms_sql_2	Litmus	Completed	**Start Time :-** 08-06-2023 20:48:49 **End Time :-** 08-06-2023 20:51:19	-	▶

Showing 1 to 1 entries of 1

V. **Percentage Result**: The Percentage Result shows the percentage of schedulers passed that belong to the group.

13.2. Validation Report (Detailed)



Validation Report (Detailed)

* Datastore Pair: Search for Datastores

* Reference Database: Search for Database

* Report With: Generate Report With

* Status Window: Select Status Window

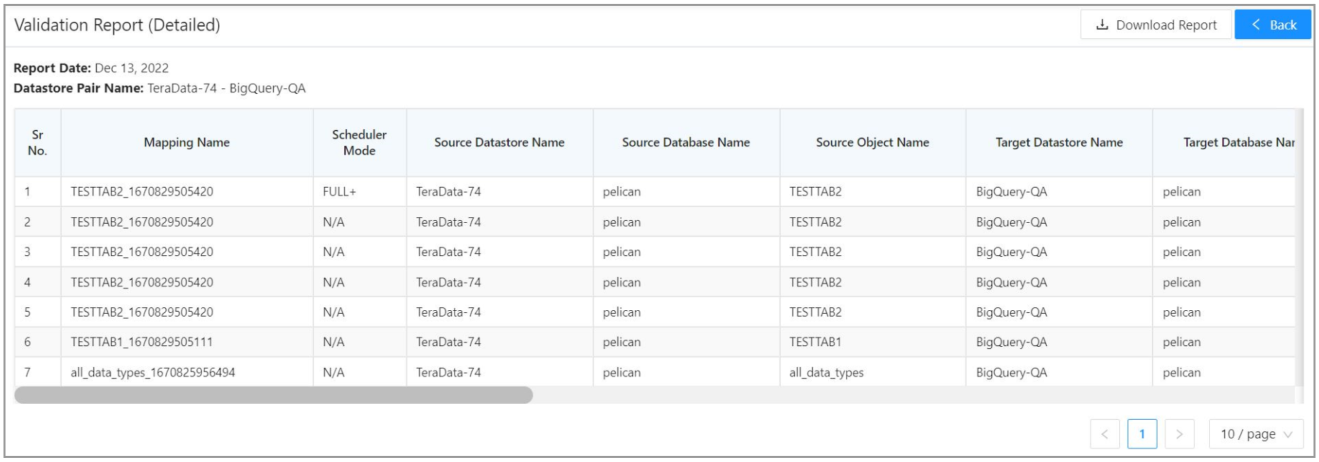
* Tables/Views: Search for Table/View

View Report

Copyright All Rights Reserved ©2023 | DataMetica Solutions Pvt. Ltd.

The Validation Report feature allows the user to create the report as per the requirements. The generated report can be viewed in the form of a table and can also be downloaded in excel format. The **Validation Report (Detailed)** page consists of the following fields:

- **Datastore Pair**: Enables selection of the datastore pair. It displays all the configured datastore pairs in Pelican.
- **Reference Database**: Enables selection of the database from which the values are fetched.
- **Report With**: Enables selection of whether the end report should be based on the Mapping details or Scheduler details. Selecting the Mapping option will display the report based on the Mappings.



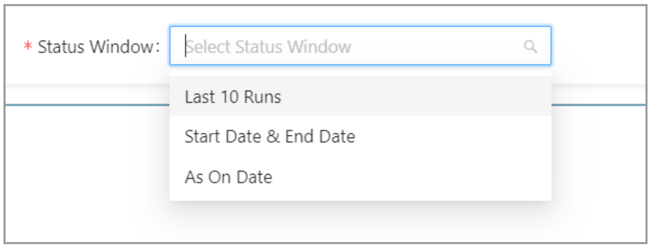
Validation Report (Detailed)

Report Date: Dec 13, 2022
Datastore Pair Name: TeraData-74 - BigQuery-QA

Sr No.	Mapping Name	Scheduler Mode	Source Datastore Name	Source Database Name	Source Object Name	Target Datastore Name	Target Database Nar
1	TESTTAB2_1670829505420	FULL+	TeraData-74	pelican	TESTTAB2	BigQuery-QA	pelican
2	TESTTAB2_1670829505420	N/A	TeraData-74	pelican	TESTTAB2	BigQuery-QA	pelican
3	TESTTAB2_1670829505420	N/A	TeraData-74	pelican	TESTTAB2	BigQuery-QA	pelican
4	TESTTAB2_1670829505420	N/A	TeraData-74	pelican	TESTTAB2	BigQuery-QA	pelican
5	TESTTAB2_1670829505420	N/A	TeraData-74	pelican	TESTTAB2	BigQuery-QA	pelican
6	TESTTAB1_1670829505111	N/A	TeraData-74	pelican	TESTTAB1	BigQuery-QA	pelican
7	all_data_types_1670825956494	N/A	TeraData-74	pelican	all_data_types	BigQuery-QA	pelican

Selecting the Scheduler option will display the result based on the schedulers.

- **Status Window**: Enables selection of the time period during which the report details are to be displayed. This field has the following options:
    - Last 10 Runs: Displays the last 10 days runs of the Scheduler/Mapping execution.
    - Start Date and End Date: Displays the execution details between the start and end date.
    - As on Date: Displays the execution details done on the present day.



* Status Window: Select Status Window

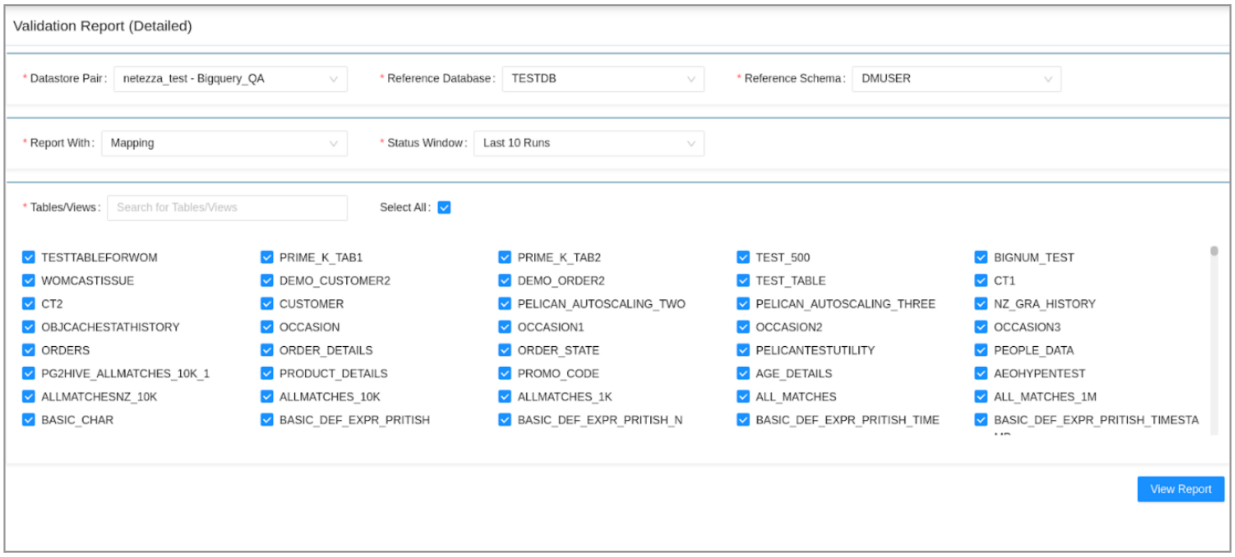
- **Tables/Views**: Enables selection of objects such as tables, views, etc. populated on selecting the schema.

Steps to generate the Detailed Report

1) Go to **Reports** -> **Validation Reports(Detailed)**.
   The **Validation Report(Detailed)** page opens.

2) Select the following details from the drop-downs: **Datastore Pair, Reference Database, Pelican** and **Status Window**.

3) Select the appropriate **Objects**.



Validation Report (Detailed)

* Datastore Pair: netezza_test - Bigquery_QA
* Reference Database: TESTDB
* Reference Schema: DMUSER

* Report With: Mapping
* Status Window: Last 10 Runs

* Tables/Views: Search for Tables/Views
Select All: ☑

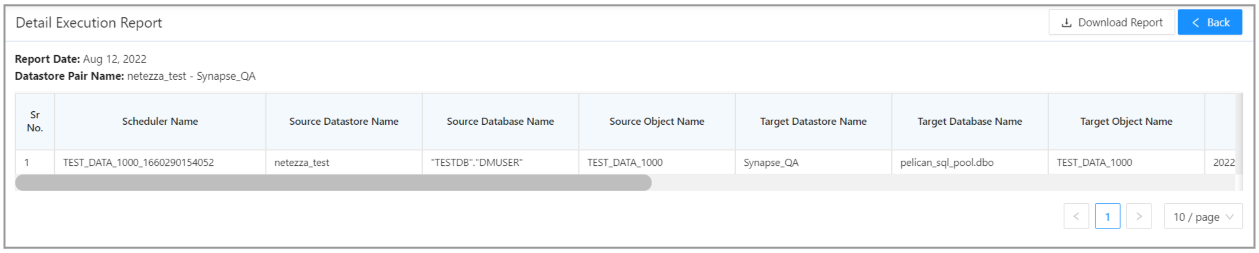
☑ TESTTABLEFORWOM
☑ WOMCASTISSUE
☑ CT2
☑ OBJCACHESTATHISTORY
☑ ORDERS
☑ PG2HIVE_ALLMATCHES_10K_1
☑ ALLMATCHESNZ_10K
☑ BASIC_CHAR
☑ PRIME_K_TAB1
☑ DEMO_CUSTOMER2
☑ CUSTOMER
☑ OCCASION
☑ ORDER_DETAILS
☑ PRODUCT_DETAILS
☑ ALLMATCHES_10K
☑ BASIC_DEF_EXPR_PRITISH
☑ PRIME_K_TAB2
☑ DEMO_ORDER2
☑ PELICAN_AUTOSCALING_TWO
☑ OCCASION1
☑ ORDER_STATE
☑ PROMO_CODE
☑ ALLMATCHES_1K
☑ BASIC_DEF_EXPR_PRITISH_N
☑ TEST_500
☑ TEST_TABLE
☑ PELICAN_AUTOSCALING_THREE
☑ OCCASION2
☑ PELICANTESTUTILITY
☑ AGE_DETAILS
☑ ALL_MATCHES
☑ BASIC_DEF_EXPR_PRITISH_TIME
☑ BIGNUM_TEST
☑ CT1
☑ NZ_GRA_HISTORY
☑ OCCASION3
☑ PEOPLE_DATA
☑ AEOHYPENTEST
☑ ALL_MATCHES_1M
☑ BASIC_DEF_EXPR_PRITISH_TIMESTA

4) Click **View Report.**

   The **Detailed Execution Report** will be displayed.

The report consists of the following columns:

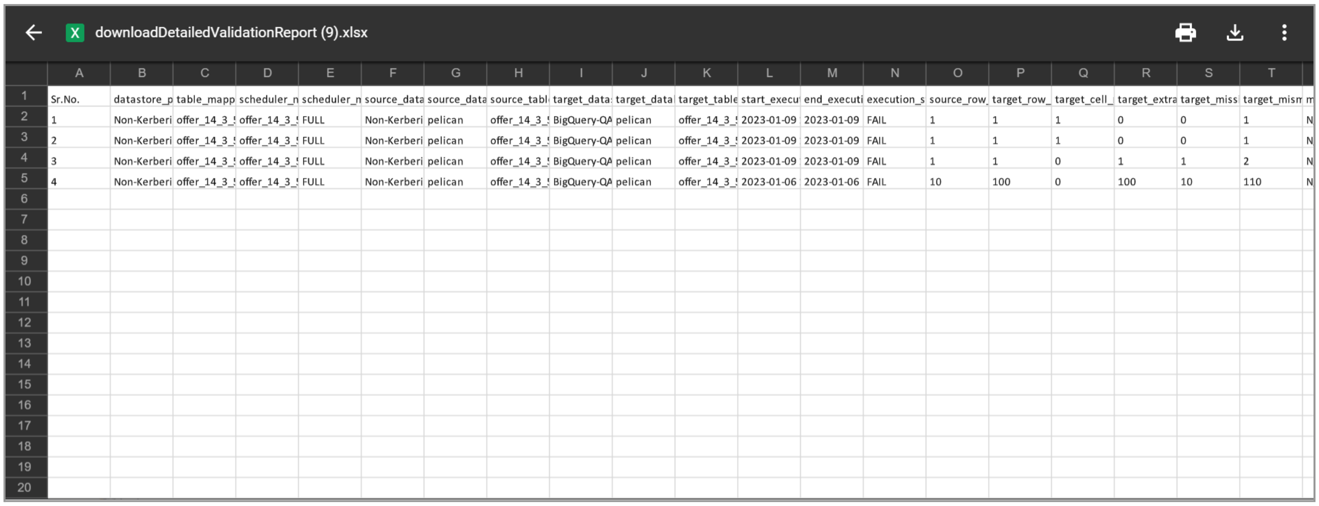
- Mapping Name
- Scheduler Name
- Source Datastore Name
- Source Database Name
- Source Object Name
- Target Datastore Name
- Target Database Name
- Target Object Name
- Start Execution Time
- End Execution Time
- Execution Status
- Source Row Count
- Target Row Count
- Cell Mismatch Row Count
- Extra Row Count
- Missing Row Count
- Mismatched Row Count
- Mismatched Column Count
- Scanned Row Count
- Mapped Columns
- Validation Size
- Remark



Report Date: Aug 12, 2022
Datastore Pair Name: netezza_test - Synapse_QA

Sr No.	Scheduler Name	Source Datastore Name	Source Database Name	Source Object Name	Target Datastore Name	Target Database Name	Target Object Name	
1	TEST_DATA_1000_1660290154052	netezza_test	"TESTDB"."DMUSER"	TEST_DATA_1000	Synapse_QA	pelican_sql_pool.dbo	TEST_DATA_1000	2022

Note: The report can also be downloaded in an .xls format.



downloadDetailedValidationReport (9).xlsx

Sr.No.	datastore_p	table_mapp	scheduler_n	scheduler_n	source_data	source_data	source_tabl	target_data:	target_data	target_table	start_execu	end_executi	execution_s	source_row_	target_row_	target_cell_	target_extra	target_missi	target_mism	m
1	Non-Kerberi	offer_14_3_!	offer_14_3_!	FULL	Non-Kerberi	pelican	offer_14_3_!	BigQuery-QA	pelican	offer_14_3_!	2023-01-09	2023-01-09	FAIL	1	1	1	0	0	1	N
2	Non-Kerberi	offer_14_3_!	offer_14_3_!	FULL	Non-Kerberi	pelican	offer_14_3_!	BigQuery-QA	pelican	offer_14_3_!	2023-01-09	2023-01-09	FAIL	1	1	1	0	0	1	N
3	Non-Kerberi	offer_14_3_!	offer_14_3_!	FULL	Non-Kerberi	pelican	offer_14_3_!	BigQuery-QA	pelican	offer_14_3_!	2023-01-09	2023-01-09	FAIL	1	1	0	1	1	2	N
4	Non-Kerberi	offer_14_3_!	offer_14_3_!	FULL	Non-Kerberi	pelican	offer_14_3_!	BigQuery-QA	pelican	offer_14_3_!	2023-01-06	2023-01-06	FAIL	10	100	0	100	10	110	N

14. License

Pelican has introduced pair restriction defined on the license file, user will not be able to create mapping if the pair limit is exhausted.

There are three tiers of licenses by which the user can access Pelican:

- **Try and Buy**:
  - The number of users accessing the Pelican is restricted and no LDAP/AD or SMTP users are allowed. The Super user will only be allowed to access the application.
  - Only 10 source tables are available for the users to select while creating a mapping.
  - The Import/Export feature is not available for this type of license.
  - By default only 10 schedulers can be run at a time.
- **Standard**:
  - The number of users accessing the application is restricted. By default only 10 schedulers can be run at a time
  - The STANDARD license type does not have any table restriction.
- **Enterprise**:
  - No restrictions on users or number of tables accessible. By default only 30 schedulers can be run at a time.

Note:

- The number of instances is the total number of source datastores that can be created for mapping in Pelican.

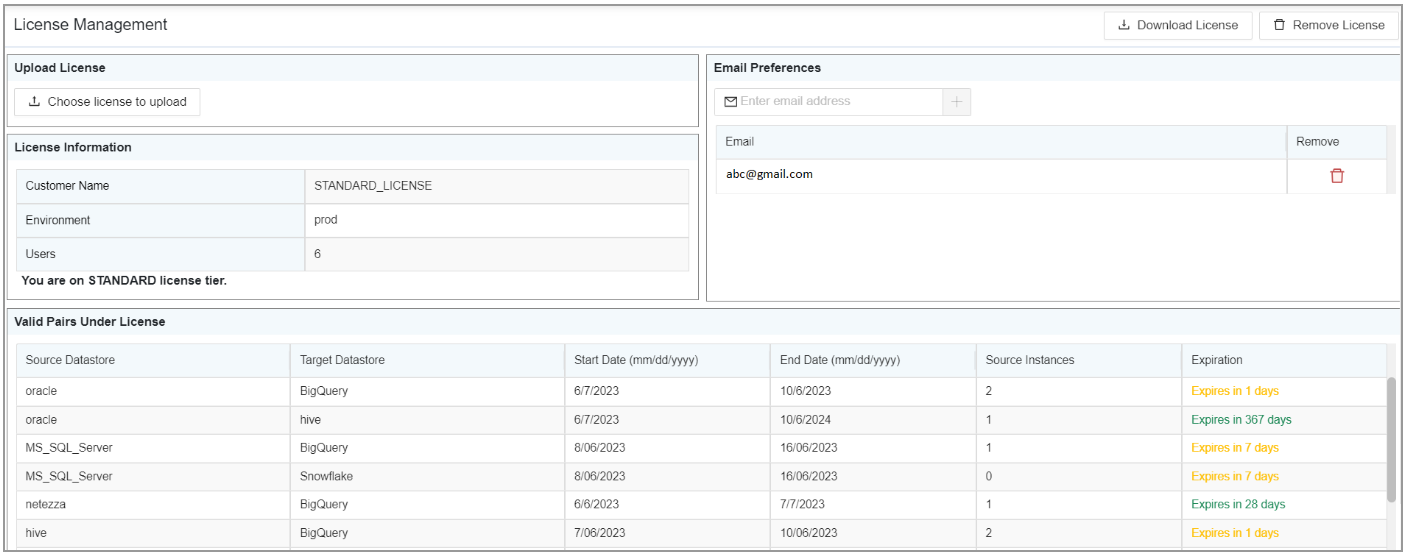
- Can not create source data store as well if source instance limit is exhausted.

To get a new License contact : PelicanCustomerSuccess@datametica.com

14.1. Adding / Updating License

Once you received the pelican license file, i.e. licence.pel follow the following steps:

1. Log in to **Pelican**.
2. Administration -> **License Management**.



License Information

Customer Name	STANDARD_LICENSE
Environment	prod
Users	6

You are on STANDARD license tier.

Email	Remove
abc@gmail.com	🗑

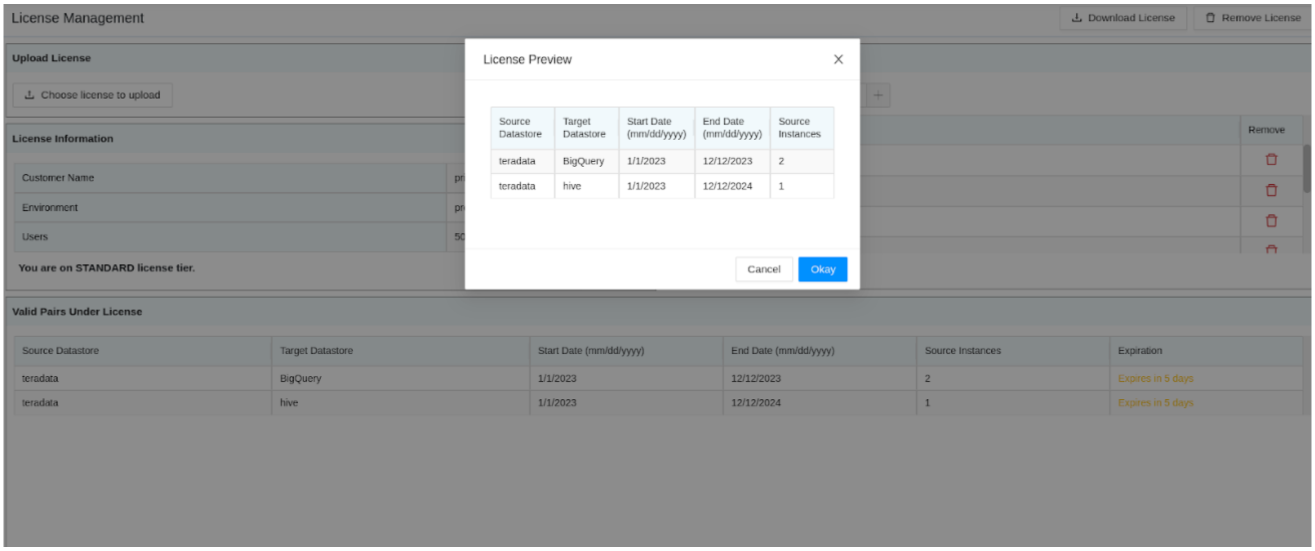
Valid Pairs Under License

Source Datastore	Target Datastore	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Source Instances	Expiration
oracle	BigQuery	6/7/2023	10/6/2023	2	Expires in 1 days
oracle	hive	6/7/2023	10/6/2024	1	Expires in 367 days
MS_SQL_Server	BigQuery	8/06/2023	16/06/2023	1	Expires in 7 days
MS_SQL_Server	Snowflake	8/06/2023	16/06/2023	0	Expires in 7 days
netezza	BigQuery	6/6/2023	7/7/2023	1	Expires in 28 days
hive	BigQuery	7/06/2023	10/06/2023	2	Expires in 1 days

3. Click on the **Choose license to upload** button and select the license from the file system.

   A **License Preview** pop-up will be displayed.

   Note: You can acquire licenses of different datastore pairs with different expiry dates.



Customer Name	pr
Environment	pr
Users	50

You are on STANDARD license tier.

Source Datastore	Target Datastore	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Source Instances
teradata	BigQuery	1/1/2023	12/12/2023	2
teradata	hive	1/1/2023	12/12/2024	1

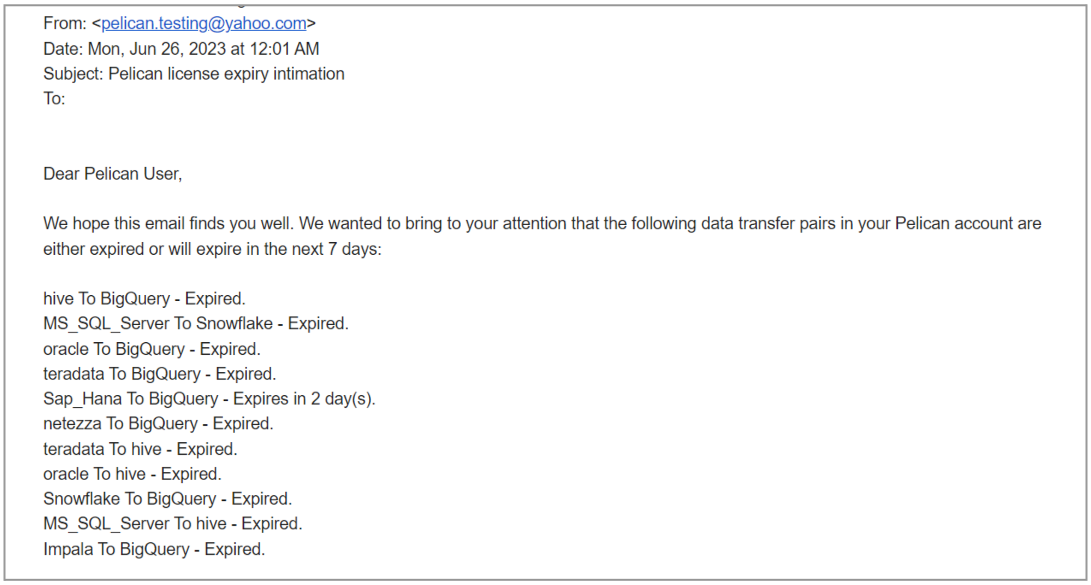
Valid Pairs Under License

Source Datastore	Target Datastore	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Source Instances	Expiration
teradata	BigQuery	1/1/2023	12/12/2023	2	Expires in 5 days
teradata	hive	1/1/2023	12/12/2024	1	Expires in 5 days

4. You will be taken back to the **Login** Page.
   Login back to continue the usage.

Note: You will be notified in the following ways regarding the license renewal:

I. A detailed email is sent to your registered email address prior to 7 days of renewal , mentioning your pair information and renewal process. All the registered email addresses will receive the email of notification. The supported mails (SMTP providers) are Gmail, Microsoft(Office 365) and Yahoo.



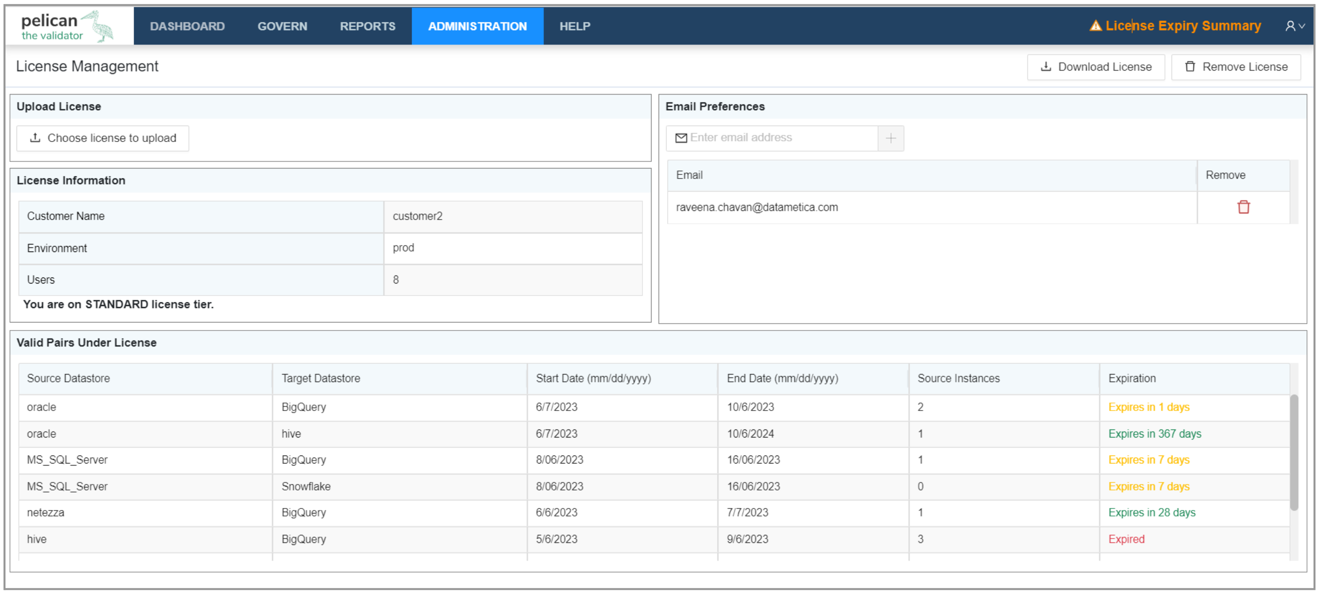
From: <pelican.testing@yahoo.com>
Date: Mon, Jun 26, 2023 at 12:01 AM
Subject: Pelican license expiry intimation
To:

Dear Pelican User,

We hope this email finds you well. We wanted to bring to your attention that the following data transfer pairs in your Pelican account are either expired or will expire in the next 7 days:

hive To BigQuery - Expired.
MS_SQL_Server To Snowflake - Expired.
oracle To BigQuery - Expired.
teradata To BigQuery - Expired.
Sap_Hana To BigQuery - Expires in 2 day(s).
netezza To BigQuery - Expired.
teradata To hive - Expired.
oracle To hive - Expired.
Snowflake To BigQuery - Expired.
MS_SQL_Server To hive - Expired.
Impala To BigQuery - Expired.

II. On the menu bar on the main screen.



License Management

Upload License

License Information

Customer Name	customer2
Environment	prod
Users	8

You are on STANDARD license tier.

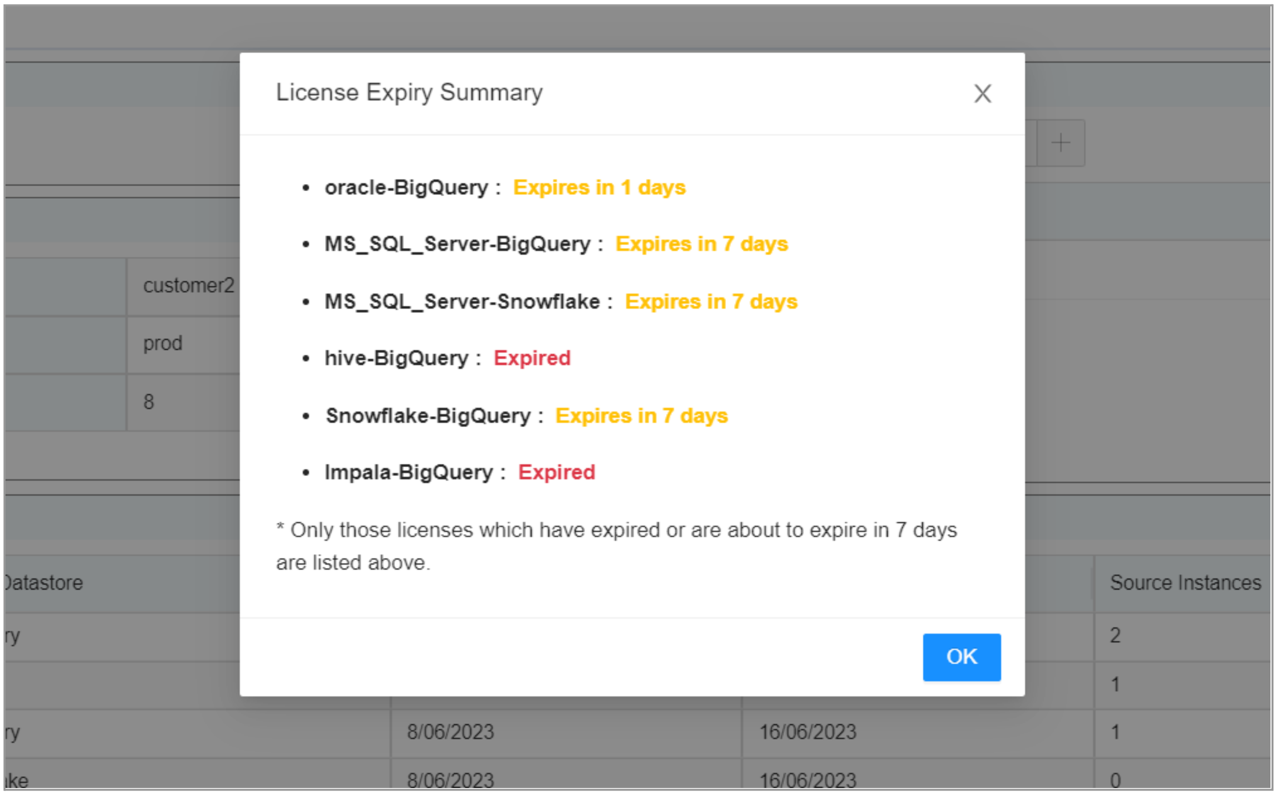
Email Preferences

Email	Remove
raveena.chavan@datametica.com	🗑

Valid Pairs Under License

Source Datastore	Target Datastore	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Source Instances	Expiration
oracle	BigQuery	6/7/2023	10/6/2023	2	Expires in 1 days
oracle	hive	6/7/2023	10/6/2024	1	Expires in 367 days
MS_SQL_Server	BigQuery	8/06/2023	16/06/2023	1	Expires in 7 days
MS_SQL_Server	Snowflake	8/06/2023	16/06/2023	0	Expires in 7 days
netezza	BigQuery	6/6/2023	7/7/2023	1	Expires in 28 days
hive	BigQuery	5/6/2023	9/6/2023	3	Expired

Click on the **License Expiry Summary** link to view the license summary.



License Expiry Summary

- **oracle-BigQuery :** **Expires in 1 days**
- **MS_SQL_Server-BigQuery :** **Expires in 7 days**
- **MS_SQL_Server-Snowflake :** **Expires in 7 days**
- **hive-BigQuery :** **Expired**
- **Snowflake-BigQuery :** **Expires in 7 days**
- **Impala-BigQuery :** **Expired**

* Only those licenses which have expired or are about to expire in 7 days are listed above.

Once your license expires contact Datametica to renew the licenses.

Once you renew the license, all the functionalities will be resumed.

15. Lineage

The Lineage feature enables the user to view the last executions of source tables and display them graphically. Therefore ,it becomes easy for the tester / developer to backtrack and identify the table where the issue started. This results in quicker root cause analysis. The benefits of this functionality is as follows:

- Finding the root-cause of the error.
- Provides metadata information, assisting to fixing issues if any gap is identified in the data processing pipeline.
- Graphic visualization of the data flow.

15.1. Add Lineage

Add Lineage is the process of adding a CSV file, which contains two columns (FromTable, ToTable) with their qualified names in the format.

database_name.table_name or database_name.schema_name.table_name

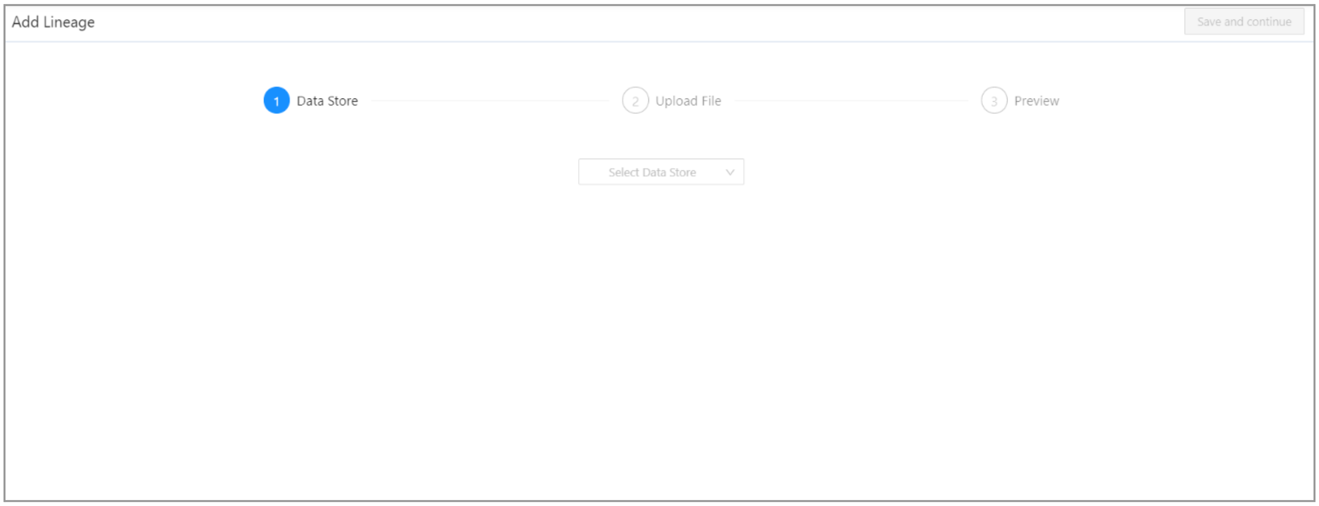
Example:
CSV file contain following details:
FromTable ToTable
Pelican.customer pelican.customer_salesman_order_details
Pelican.salesman pelican.customer_salesman_order_details
Pelican.orders pelican.customer_salesman_order_details

How to Add lineage?

Only Admin or superuser has access to the add Lineage Or upload a CSV file.

To steps to add Lineage are as follows:

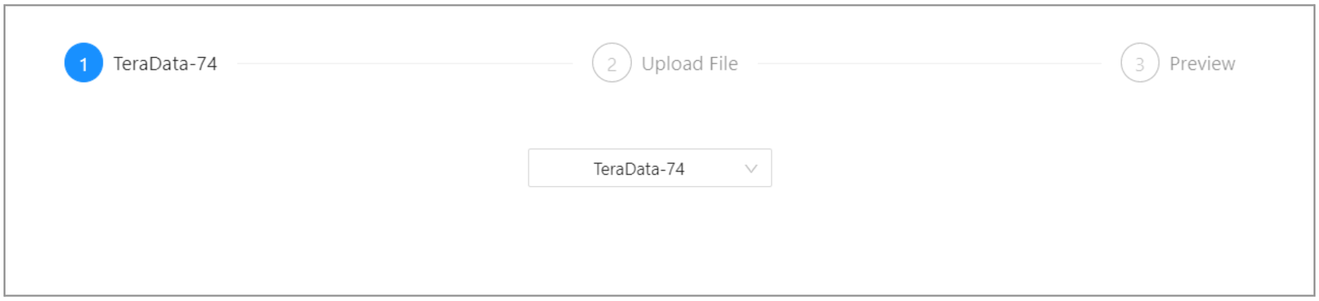
1. Go to **Administration** → **Add Lineage.**



2. Select the **Datastore** and click on **Preview Uploaded File.**

   The **Save and Continue** button will be enabled.

3. Click **Save and Continue.**

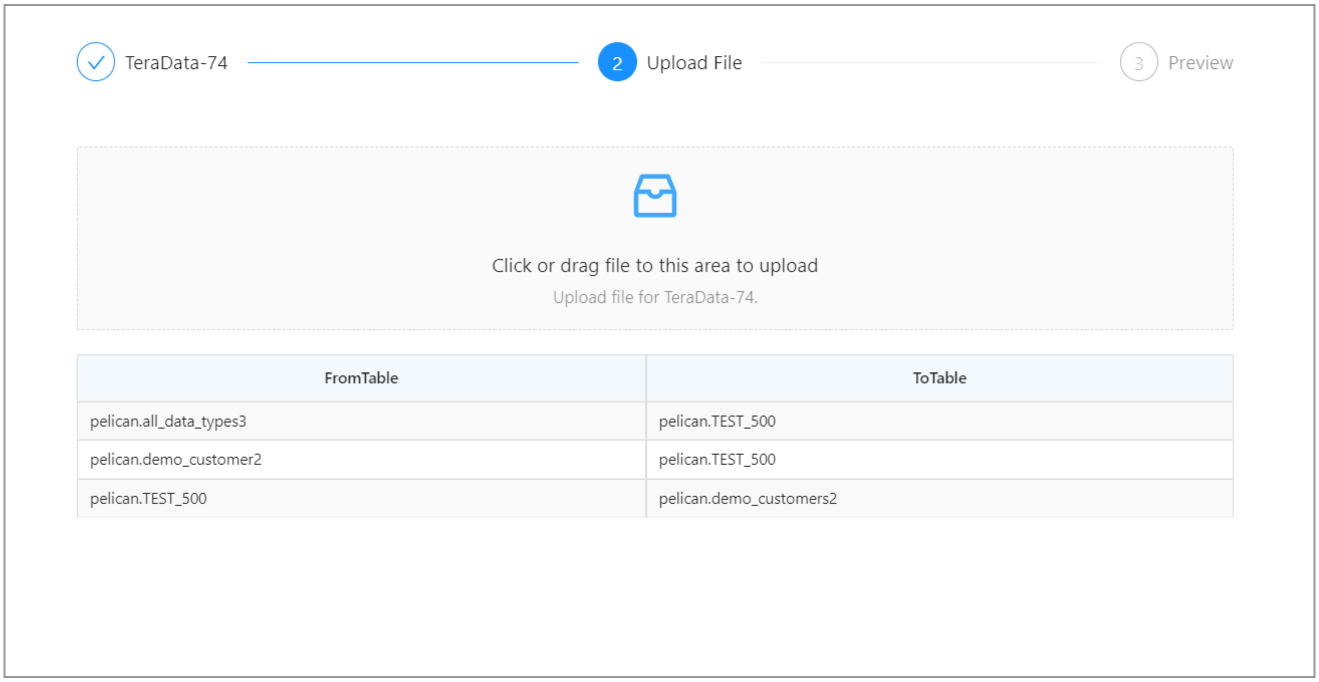


4. A page to upload the CSV file will appear.

   You can **drag and drop** a CSV file or click on **Upload file** link**.**

5. Select a lineage file with .csv extension and click **open.**

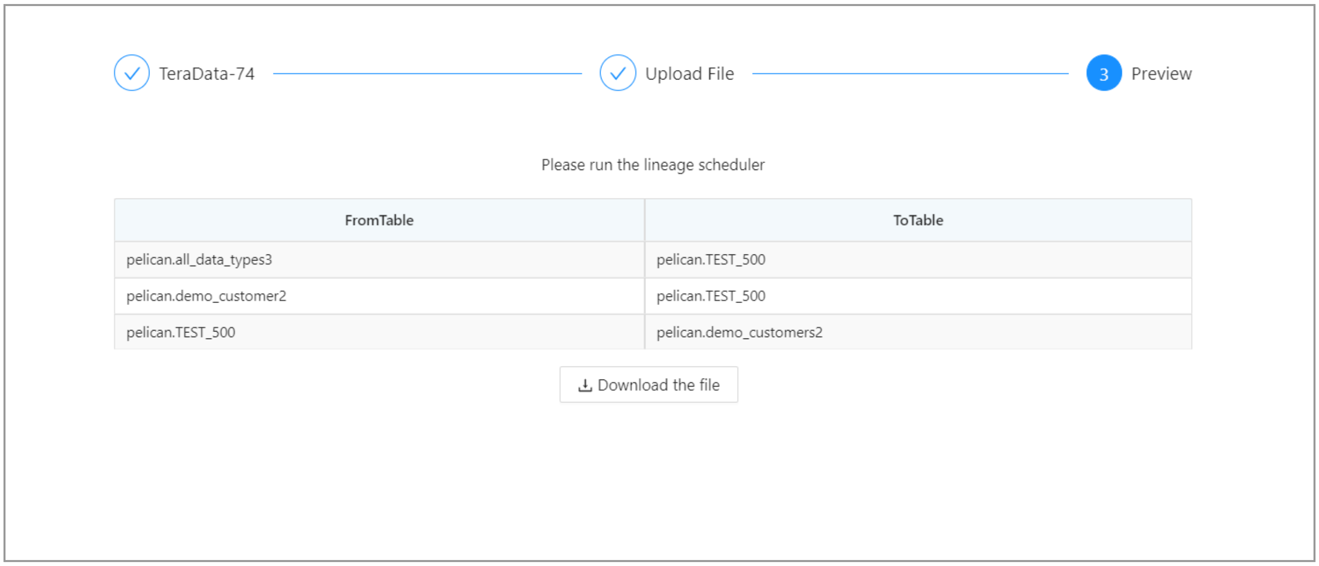
6. The add lineage file preview will be displayed.


TeraData-74
2 Upload File
3 Preview
Click or drag file to this area to upload
Upload file for TeraData-74.

FromTable	ToTable
pelican.all_data_types3	pelican.TEST_500
pelican.demo_customer2	pelican.TEST_500
pelican.TEST_500	pelican.demo_customers2

7. When the file is successfully uploaded, you can see the preview.

   Note: The user can download the uploaded file, by clicking on the **Download the file** button.



Please run the lineage scheduler

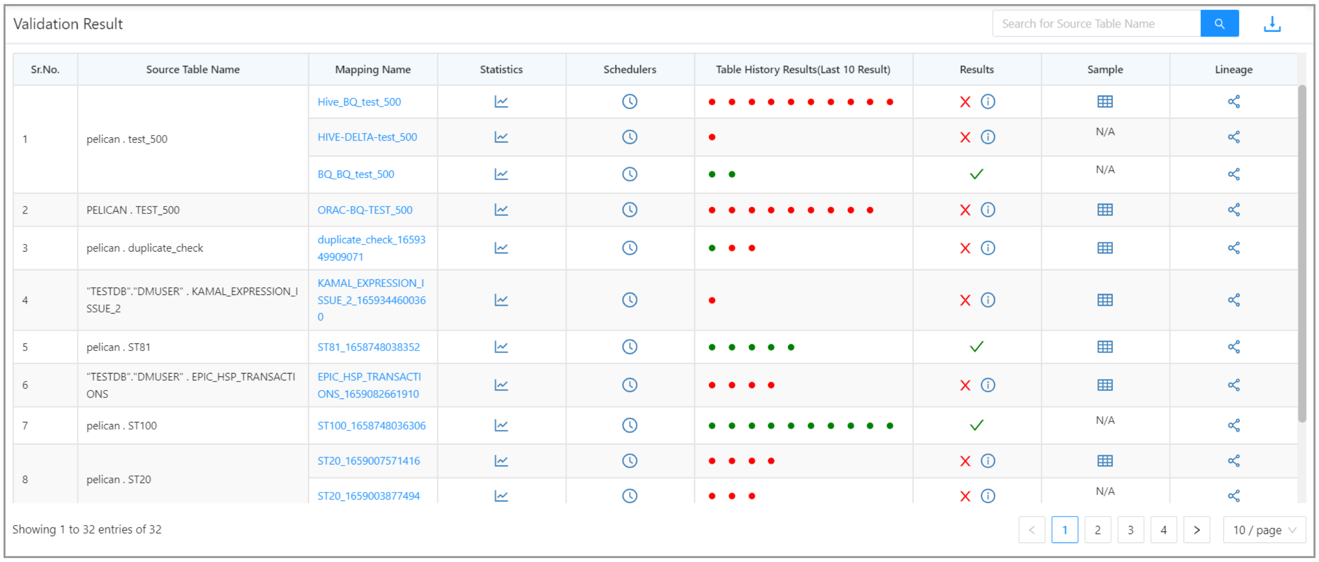
FromTable	ToTable
pelican.all_data_types3	pelican.TEST_500
pelican.demo_customer2	pelican.TEST_500
pelican.TEST_500	pelican.demo_customers2

15.2. How to View Lineage?

To view the lineage status, follow the steps mentioned below

1. Go to the **Reports** → **Validation Result** → **Lineage** (can be viewed on each mapping result).

2. Click on the **Lineage** record.



Validation Result

Search for Source Table Name

Sr.No.	Source Table Name	Mapping Name	Statistics	Schedulers	Table History Results(Last 10 Result)	Results	Sample	Lineage
1	pelican . test_500	Hive_BQ_test_500	📈	🕓	🔴 🔴 🔴 🔴 🔴 🔴 🔴 🔴 🔴 🔴	✗ ⓘ	▦	⋖
		HIVE-DELTA-test_500	📈	🕓	🔴	✗ ⓘ	N/A	⋖
		BQ_BQ_test_500	📈	🕓	🟢 🟢	✓	N/A	⋖
2	PELICAN . TEST_500	ORAC-BQ-TEST_500	📈	🕓	🔴 🔴 🔴 🔴 🔴 🔴 🔴 🔴 🔴	✗ ⓘ	▦	⋖
3	pelican . duplicate_check	duplicate_check_1659349909071	📈	🕓	🟢 🔴 🔴	✗ ⓘ	▦	⋖
4	"TESTDB"."DMUSER" . KAMAL_EXPRESSION_ISSUE_2	KAMAL_EXPRESSION_ISSUE_2_1659344600360	📈	🕓	🔴	✗ ⓘ	▦	⋖
5	pelican . ST81	ST81_1658748038352	📈	🕓	🟢 🟢 🟢 🟢 🟢	✓	▦	⋖
6	"TESTDB"."DMUSER" . EPIC_HSP_TRANSACTIONS	EPIC_HSP_TRANSACTIONS_1659082661910	📈	🕓	🔴 🔴 🔴 🔴	✗ ⓘ	▦	⋖
7	pelican . ST100	ST100_1658748036306	📈	🕓	🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢 🟢	✓	N/A	⋖
8	pelican . ST20	ST20_1659007571416	📈	🕓	🔴 🔴 🔴 🔴	✗ ⓘ	▦	⋖
		ST20_1659003877494	📈	🕓	🔴 🔴 🔴	✗ ⓘ	N/A	⋖

Showing 1 to 32 entries of 32

3. Linagage results will be displayed in the pictorial format.

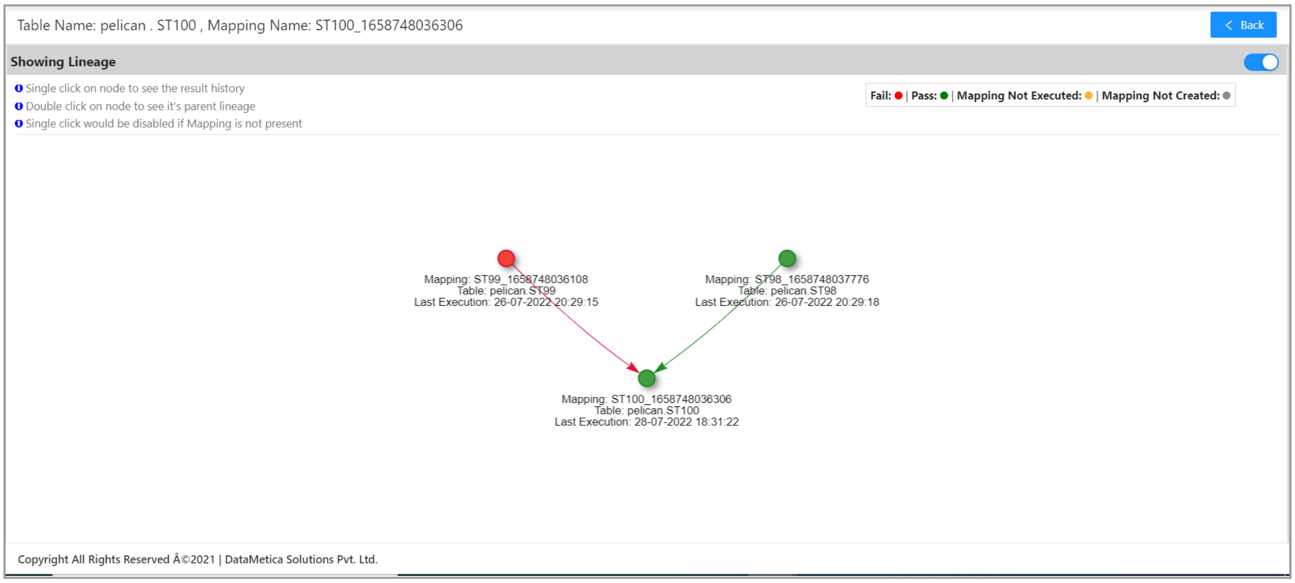


Table Name: pelican . ST100 , Mapping Name: ST100_1658748036306

Showing Lineage

- Single click on node to see the result history
- Double click on node to see it's parent lineage
- Single click would be disabled if Mapping is not present

Fail: 🔴 | **Pass:** 🟢 | **Mapping Not Executed:** 🟡 | **Mapping Not Created:** ⚪

Mapping: ST99_1658748036108
Table: pelican.ST99
Last Execution: 26-07-2022 20:29:15
Mapping: ST98_1658748037776
Table: pelican.ST98
Last Execution: 26-07-2022 20:29:18
Mapping: ST100_1658748036306
Table: pelican.ST100
Last Execution: 28-07-2022 18:31:22

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

Note: Single click on the node to see the result history, while double clicking on the node will take you to its parent lineage.

Single Click Result

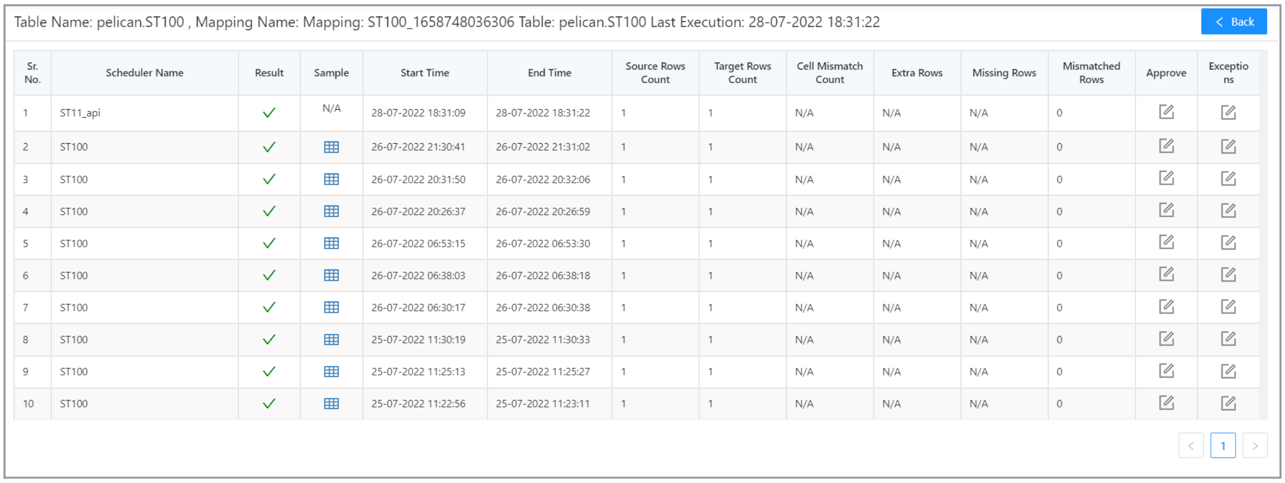


Table Name: pelican.ST100 , Mapping Name: Mapping: ST100_1658748036306 Table: pelican.ST100 Last Execution: 28-07-2022 18:31:22

Sr. No.	Scheduler Name	Result	Sample	Start Time	End Time	Source Rows Count	Target Rows Count	Cell Mismatch Count	Extra Rows	Missing Rows	Mismatched Rows	Approve	Exceptions
1	ST11_api	✓	N/A	28-07-2022 18:31:09	28-07-2022 18:31:22	1	1	N/A	N/A	N/A	0	✎	✎
2	ST100	✓	▦	26-07-2022 21:30:41	26-07-2022 21:31:02	1	1	N/A	N/A	N/A	0	✎	✎
3	ST100	✓	▦	26-07-2022 20:31:50	26-07-2022 20:32:06	1	1	N/A	N/A	N/A	0	✎	✎
4	ST100	✓	▦	26-07-2022 20:26:37	26-07-2022 20:26:59	1	1	N/A	N/A	N/A	0	✎	✎
5	ST100	✓	▦	26-07-2022 06:53:15	26-07-2022 06:53:30	1	1	N/A	N/A	N/A	0	✎	✎
6	ST100	✓	▦	26-07-2022 06:38:03	26-07-2022 06:38:18	1	1	N/A	N/A	N/A	0	✎	✎
7	ST100	✓	▦	26-07-2022 06:30:17	26-07-2022 06:30:38	1	1	N/A	N/A	N/A	0	✎	✎
8	ST100	✓	▦	25-07-2022 11:30:19	25-07-2022 11:30:33	1	1	N/A	N/A	N/A	0	✎	✎
9	ST100	✓	▦	25-07-2022 11:25:13	25-07-2022 11:25:27	1	1	N/A	N/A	N/A	0	✎	✎
10	ST100	✓	▦	25-07-2022 11:22:56	25-07-2022 11:23:11	1	1	N/A	N/A	N/A	0	✎	✎

Double click Result

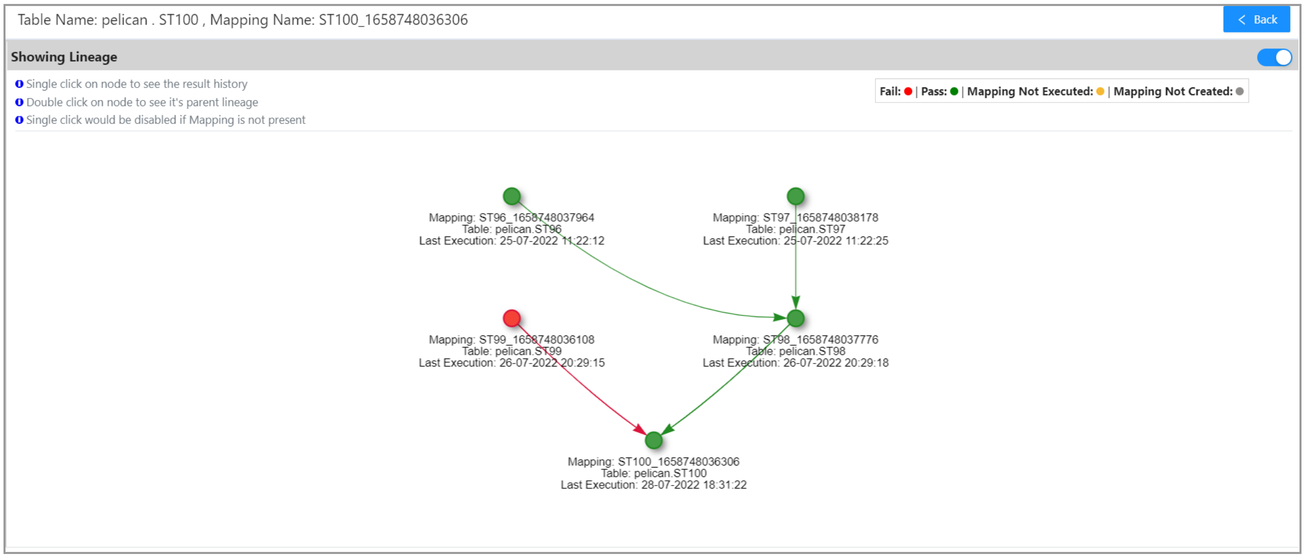


Table Name: pelican . ST100 , Mapping Name: ST100_1658748036306

Showing Lineage

- Single click on node to see the result history
- Double click on node to see it's parent lineage
- Single click would be disabled if Mapping is not present

Fail: 🔴 | **Pass:** 🟢 | **Mapping Not Executed:** 🟡 | **Mapping Not Created:** ●

Mapping: ST96_1658748037964
Table: pelican.ST96
Last Execution: 25-07-2022 11:22:12
Mapping: ST97_1658748038178
Table: pelican.ST97
Last Execution: 25-07-2022 11:22:25
Mapping: ST99_1658748036108
Table: pelican.ST99
Last Execution: 26-07-2022 20:29:15
Mapping: ST98_1658748037776
Table: pelican.ST98
Last Execution: 26-07-2022 20:29:18
Mapping: ST100_1658748036306
Table: pelican.ST100
Last Execution: 28-07-2022 18:31:22

16. Export/Import

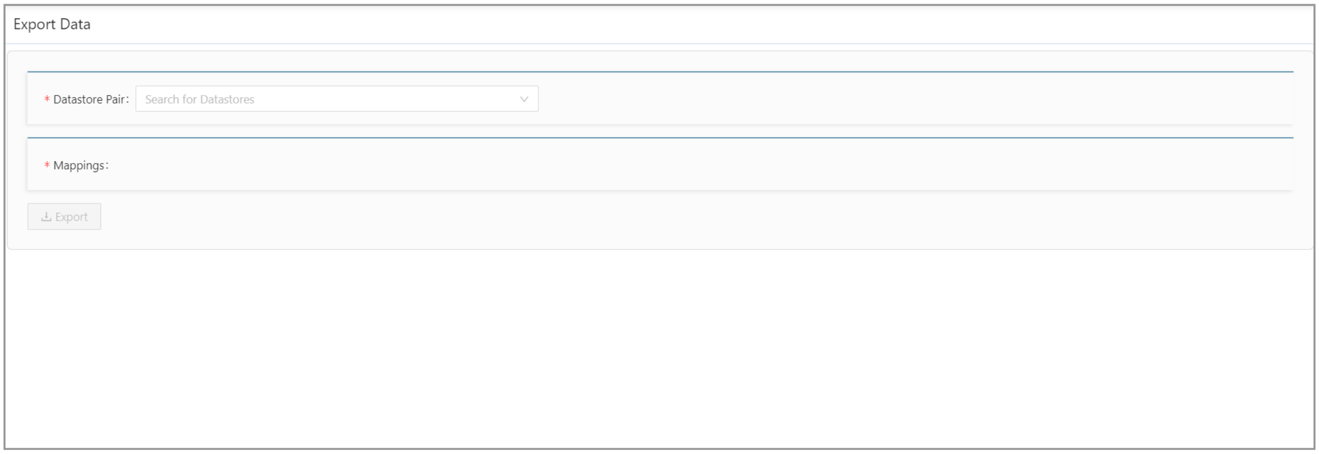
This feature is used to export data from one Pelican instance to another instance of the same version. With this functionality, the user will be able to export and import Pelican metadata viz. table mappings and their scheduler for selected pair of data stores. The export and import can be done across pelican deployments with the same version. It is accessible only to the Administrator. The user needs to repeat the below mentioned activity to export and import the data of a different pair of data stores.

The steps to Export data are as follows:

1) Go to **Administration** -> **Export/Import Data**.

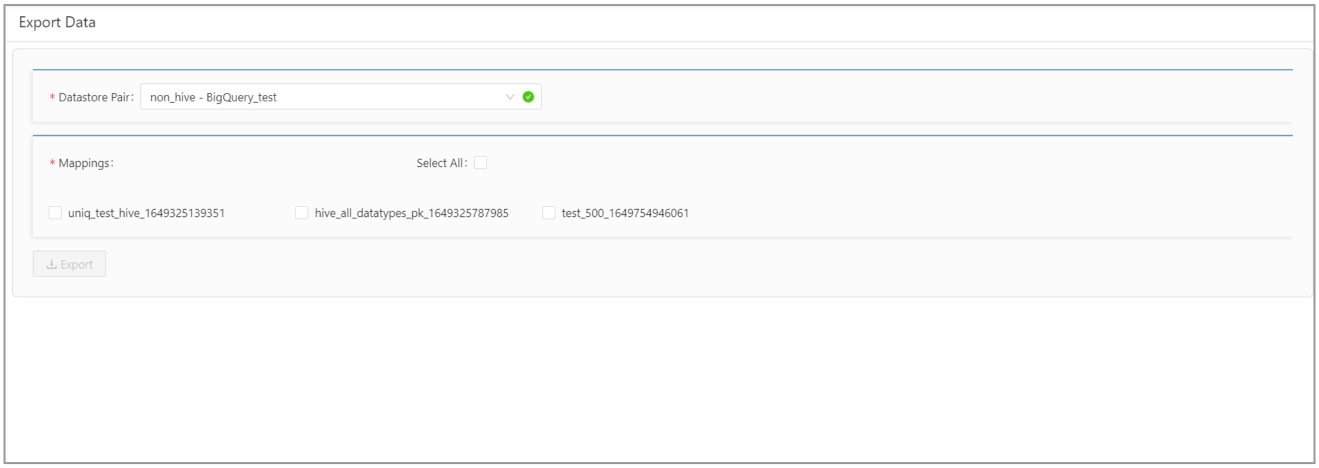
2) Select **Export Data**.

   The **Export Data** page opens.



Export Data

3) Select the **Datastore pair** as per your requirements.
   After selecting the datastore all the mappings associated with the datastore pair will be displayed.


Export Data

* Datastore Pair: non_hive - BigQuery_test

* Mappings: Select All: □

□ uniq_test_hive_1649325139351 □ hive_all_datatypes_pk_1649325787985 □ test_500_1649754946061

The user can select all the mappings by selecting the **Select All** checkbox, or can select the individual mappings for export.

On selecting the mapping(s) the Export button will get enabled.

4) Click **Export**.

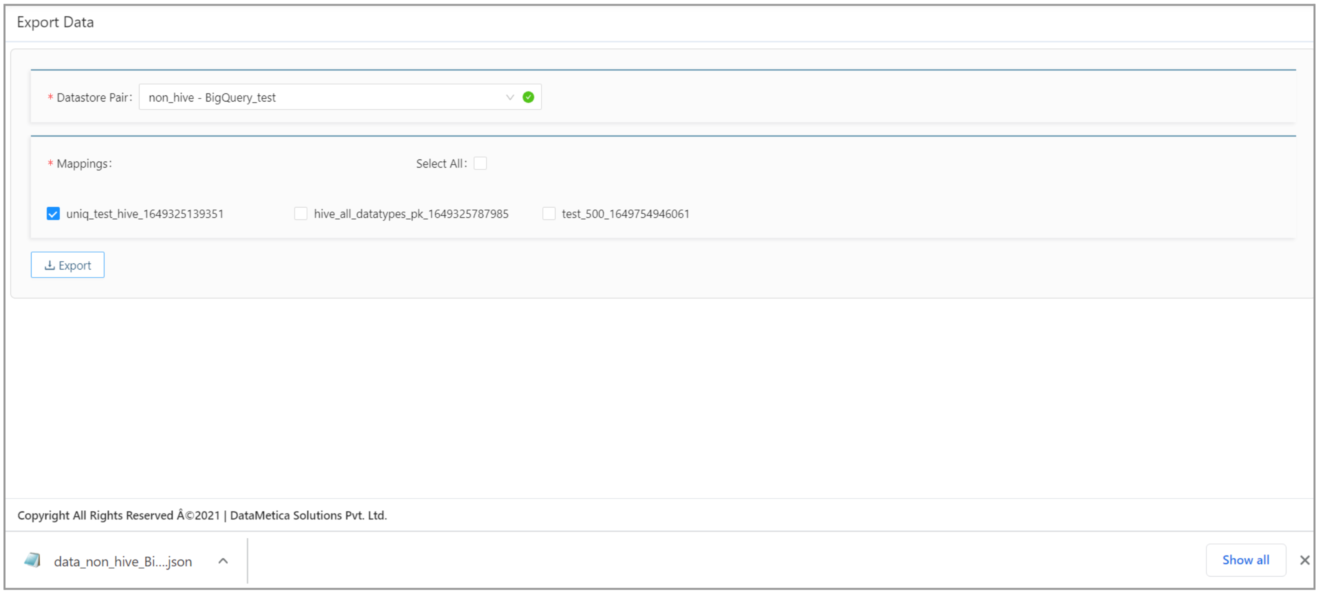
A pop-up to download the JSON file will be displayed.

Note:

- The JSON file name will have the datastore pair and the timestamp of the export.
- Also, the pop-up window will be displayed only if your browser supports it, in some browsers the file will be downloaded directly.

5) Click **Save**.

The JSON file will be downloaded in the **Downloads** folder.

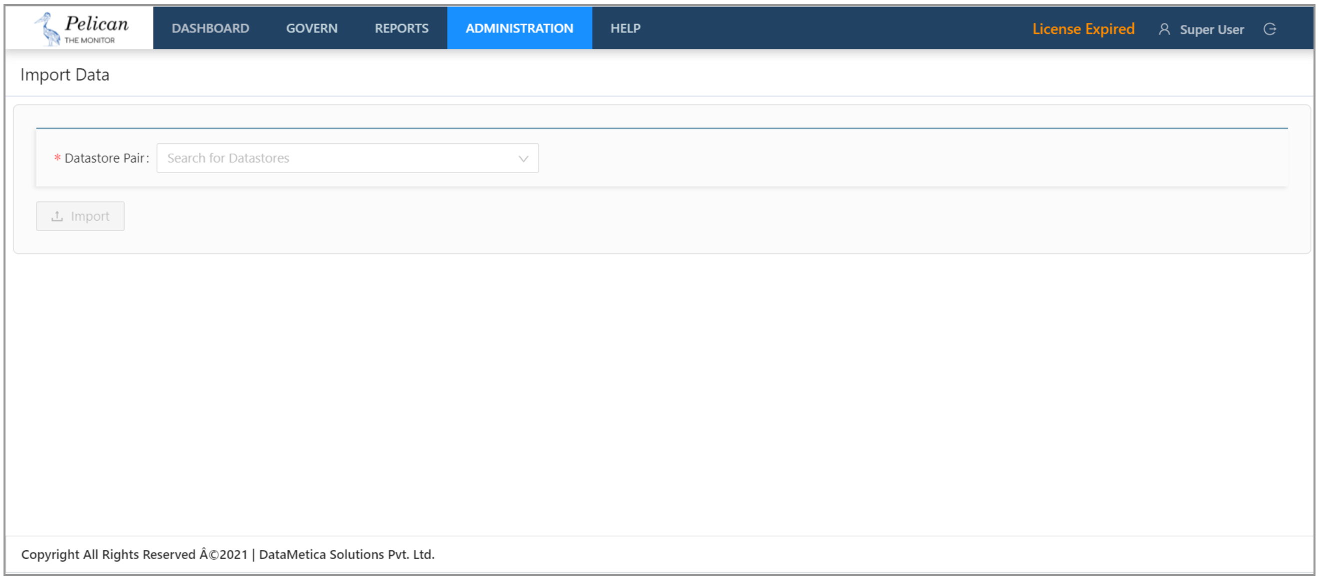

Export Data
* Datastore Pair: non_hive - BigQuery_test
* Mappings: Select All: □
☑ uniq_test_hive_1649325139351 □ hive_all_datatypes_pk_1649325787985 □ test_500_1649754946061
Export
Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.
data_non_hive_Bi....json
Show all

File Import

The steps to import a file are as follows:

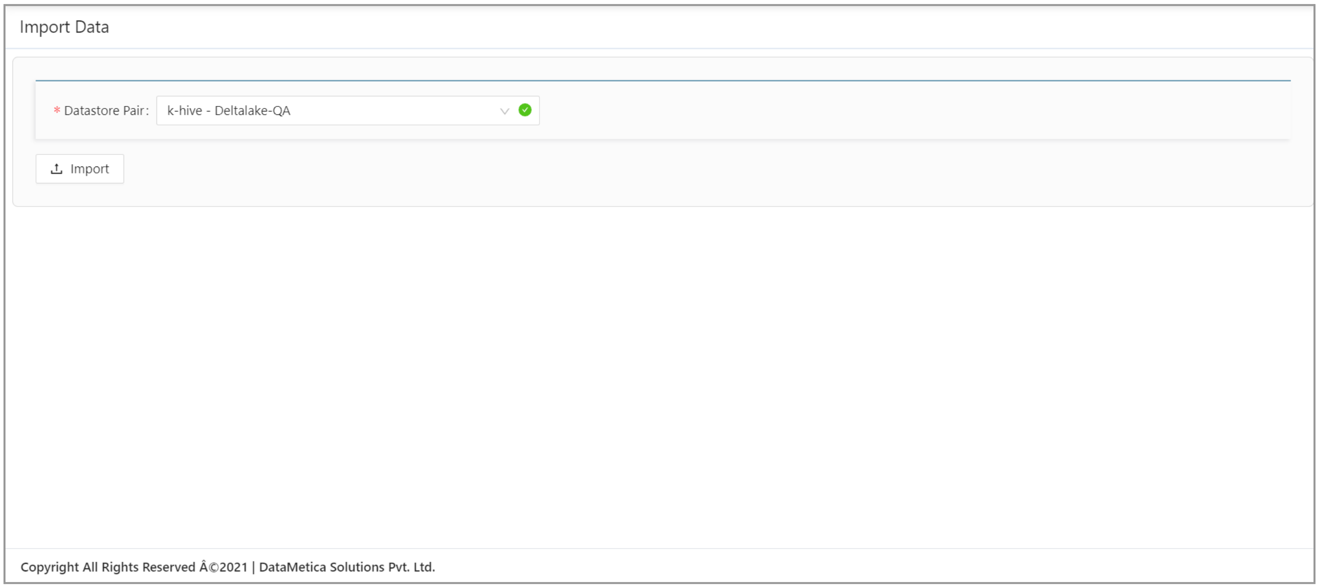
1. Go to **Administration** -> **Export/Import Data**.
2. Select **Import Data**.

   The **Import Data** page opens.


Pelican THE MONITOR
DASHBOARD
GOVERN
REPORTS
ADMINISTRATION
HELP
License Expired
Super User
Import Data
* Datastore Pair: Search for Datastores
Import
Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

3. Select the **Datastore pair** where you want to import the data.

   The **Import** button will be enabled only after selecting the datastore pair.


Import Data

* Datastore Pair: k-hive - Deltalake-QA

Copyright All Rights Reserved Â©2021 | DataMetica Solutions Pvt. Ltd.

4. Click **Import**.

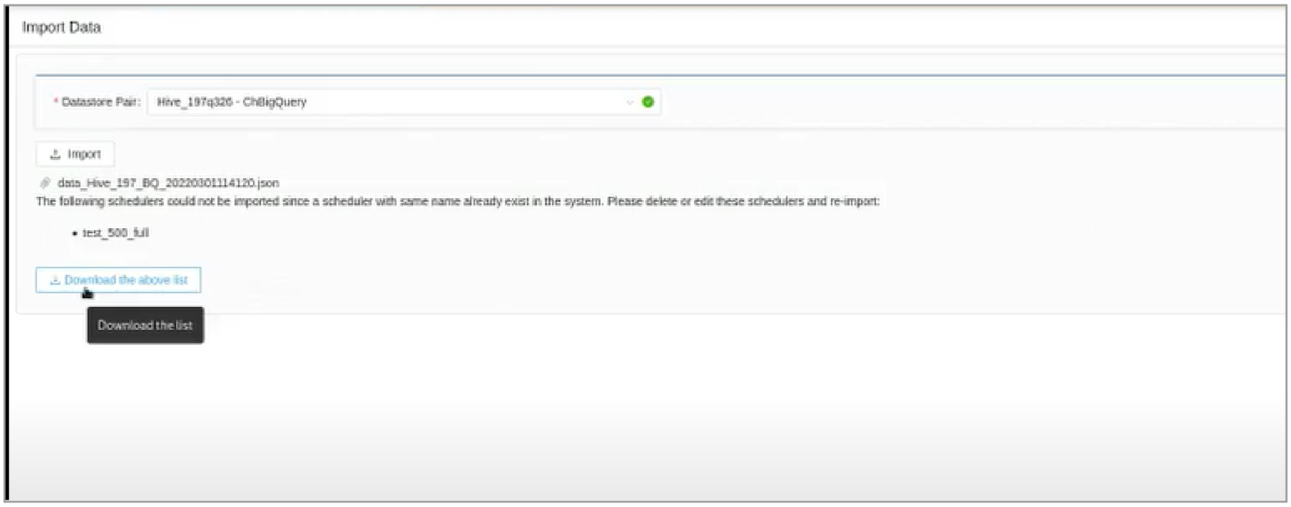
   The File upload pop-up window appears. Go to the previously downloaded location and select the file.

   Note: You can upload a JSON file upto size 5MB.

5. Click **Open**.
   A success message will be displayed on successful import of all the data.

   Note:

   - An error message will be displayed if a scheduler with the same name already exists in the database. The user is advised to delete or edit the scheduler and re-import. Please note that the other schedulers will be imported successfully. This is the case of partial import. The user can download the list of schedulers which were not imported and refer to it to edit or delete them.

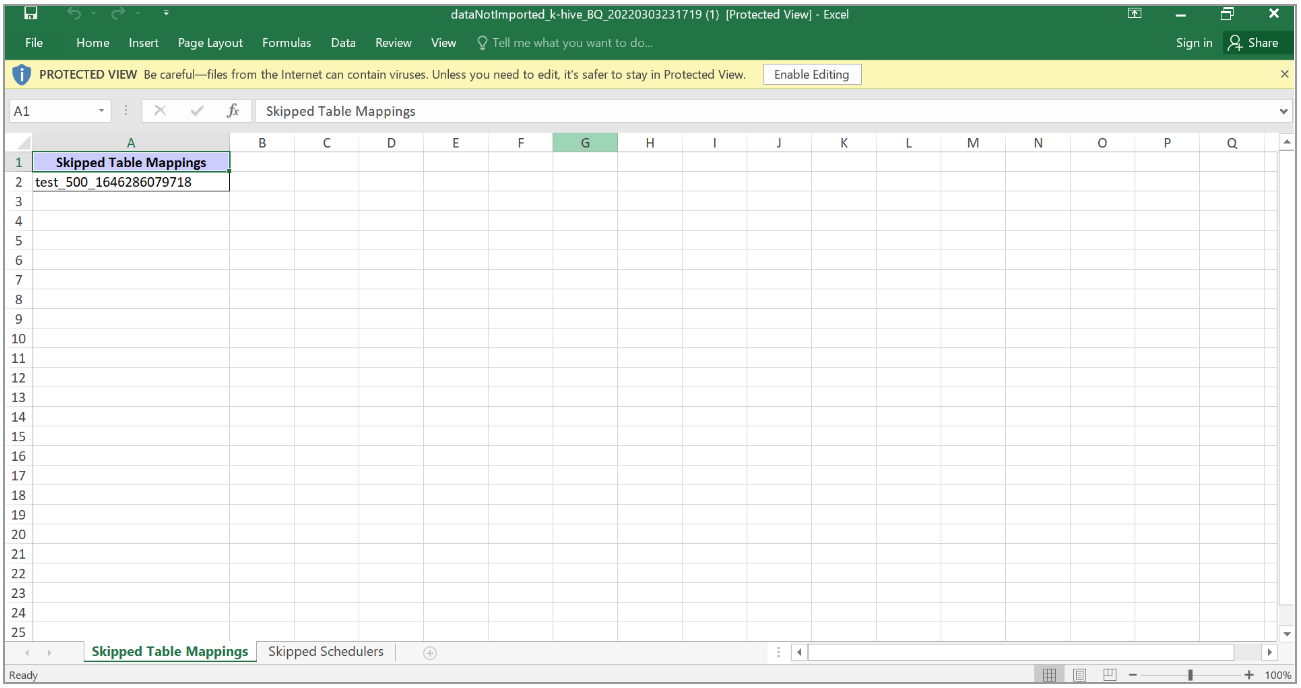

Import Data

* Datastore Pair: Hive_197q326 - ChBigQuery

data_Hive_197_BQ_20220301114120.json

The following schedulers could not be imported since a scheduler with same name already exist in the system. Please delete or edit these schedulers and re-import:

- test_500_full

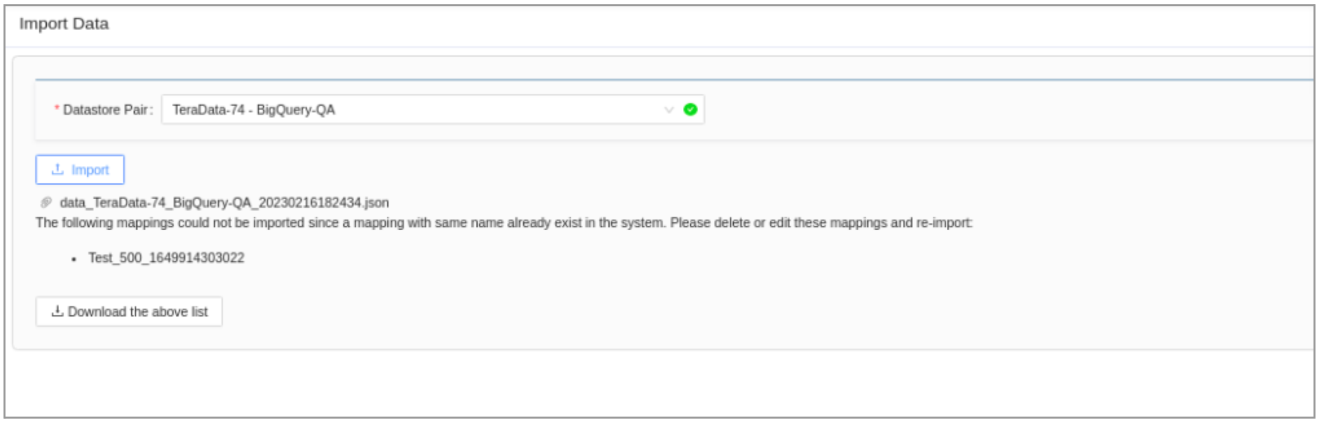

dataNotImported_k-hive_BQ_20220303231719 (1) [Protected View] - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Tell me what you want to do...
Sign in
Share
Enable Editing
A1
Ready
100%

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

	A
1	Skipped Table Mappings
2	test_500_1646286079718

Sheets: Skipped Table Mappings | Skipped Schedulers

- An error message will be displayed if a table mapping with the same name already exists in the database. The user is advised to delete or edit the mapping and re-import. Please note that the other mappings will be imported successfully.



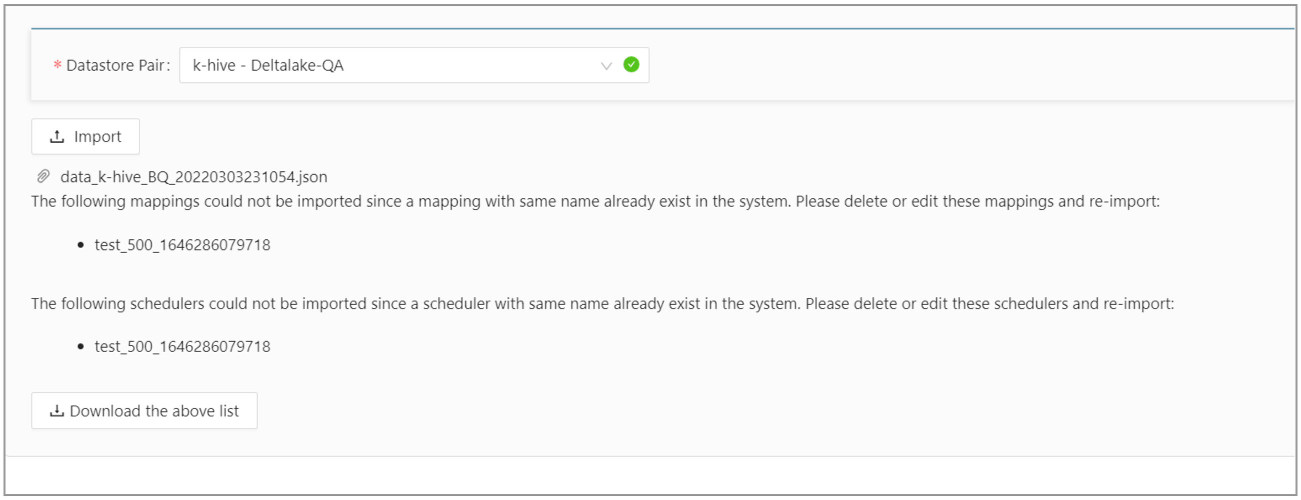
* Datastore Pair: TeraData-74 - BigQuery-QA

data_TeraData-74_BigQuery-QA_20230216182434.json

The following mappings could not be imported since a mapping with same name already exist in the system. Please delete or edit these mappings and re-import:

- Test_500_1649914303022

- If there is an error while processing the uploaded file, the error message will be displayed saying data import failed.



* Datastore Pair : k-hive - Deltalake-QA

Import

data_k-hive_BQ_20220303231054.json

The following mappings could not be imported since a mapping with same name already exist in the system. Please delete or edit these mappings and re-import:

- test_500_1646286079718

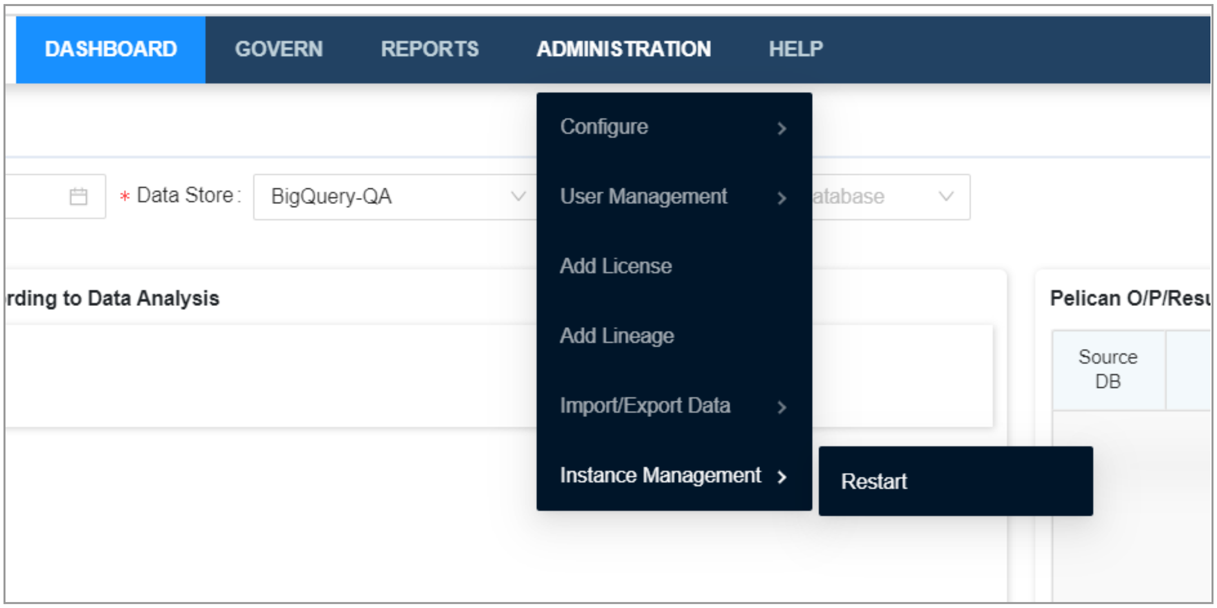
The following schedulers could not be imported since a scheduler with same name already exist in the system. Please delete or edit these schedulers and re-import:

- test_500_1646286079718

Download the above list

17. Instance Management

The Instance Management feature enables the user to restart Pelican (Tomcat server).

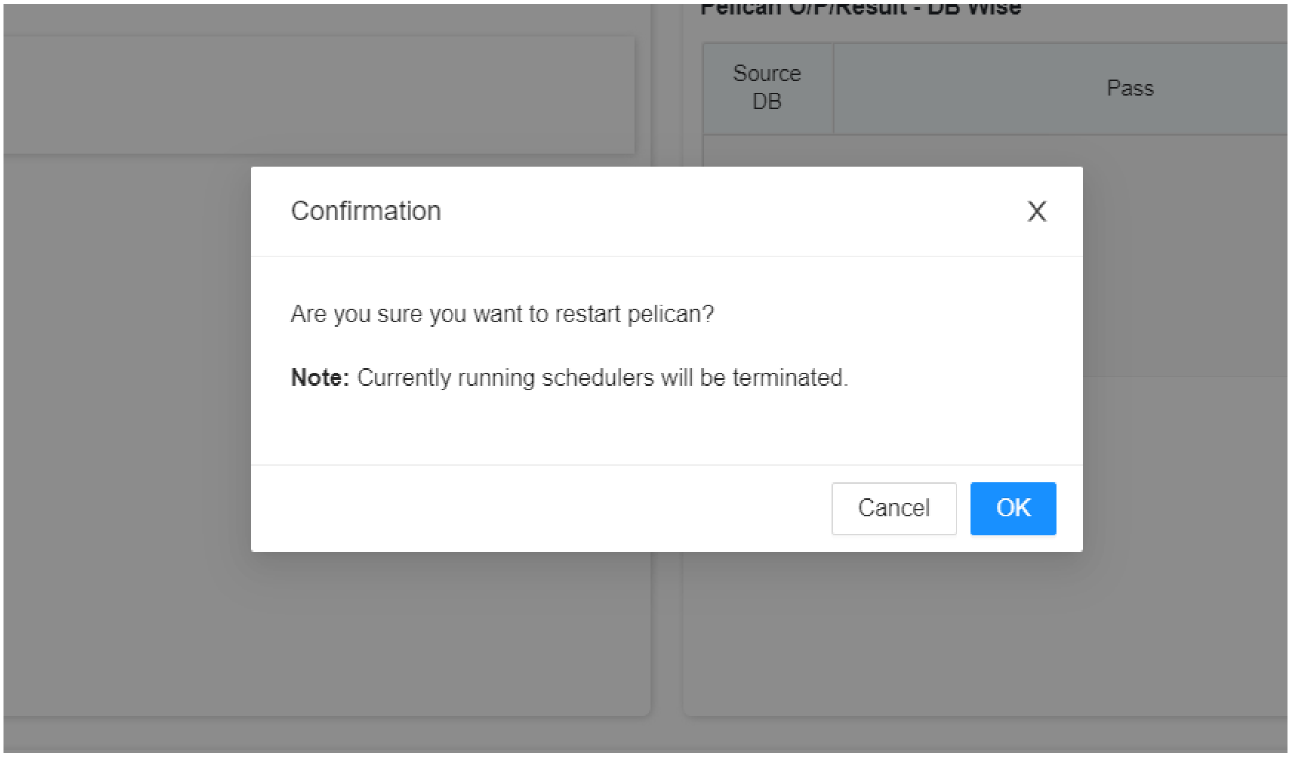


The steps to restart Pelican are as follows:

1. Select **Administration** -> **Instance Management** -> **Restart**.

   A Confirmation pop-up will appear .

   Note: All the Schedulers of the instance will be terminated.



Are you sure you want to restart pelican?

Note: Currently running schedulers will be terminated.

2. Click **Ok**.

18. API

API 1: OAuth API Call

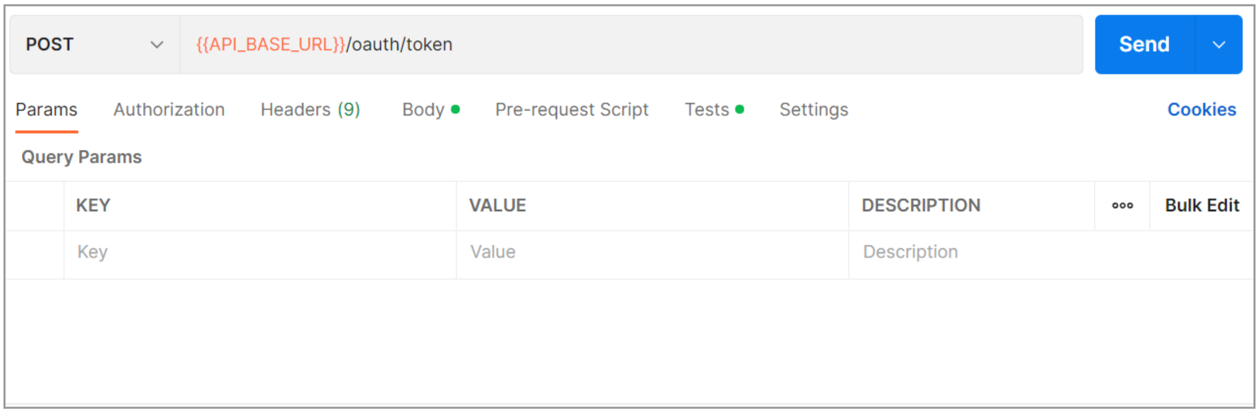
URL: http://<IP of PELICAN Machine>:<port>/oauth/token

Method: POST

Description: This API generates the authentication token for other API's to consume.

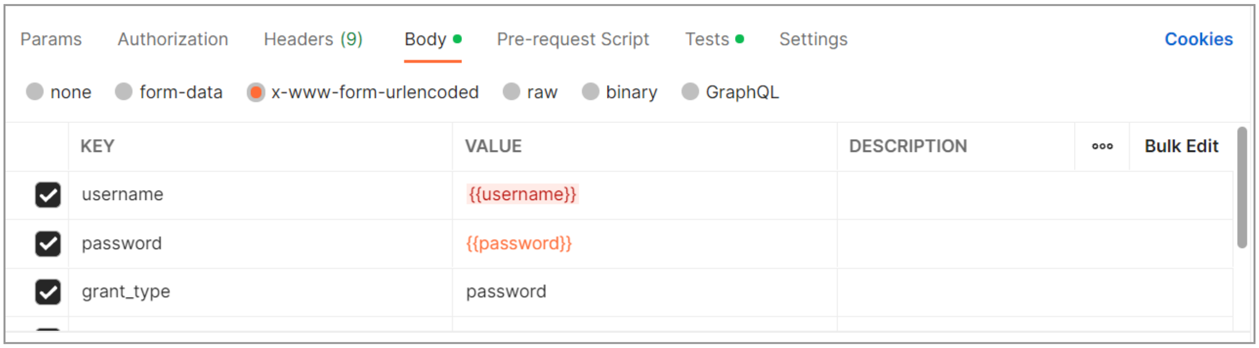
Headers:

Accept :	application/json
Content-Type:	application/x-www-form-urlencoded



Request Body:

username	<username>
password	<password>
grant_type	password
Client_id	ecat
Client_secret	ecat



	KEY	VALUE	DESCRIPTION
☑	username	{{username}}	
☑	password	{{password}}	
☑	grant_type	password	

Response: 200 OK

```
{
   "access_token": "-yxUFR7jEwJX4l-dAQsor-eyNxw",
   "token_type": "bearer",
   "refresh_token": "8u94ZI3t_DYeo-d6MdnH8BAlQpk",
   "expires_in": 19390,
   "scope": "read write"
}
```

API 2: Creates Pelican Table Mapping using Public API Version 3

URL: http://<IP of PELICAN Machine>/api/rest/v3/pelican/mapping

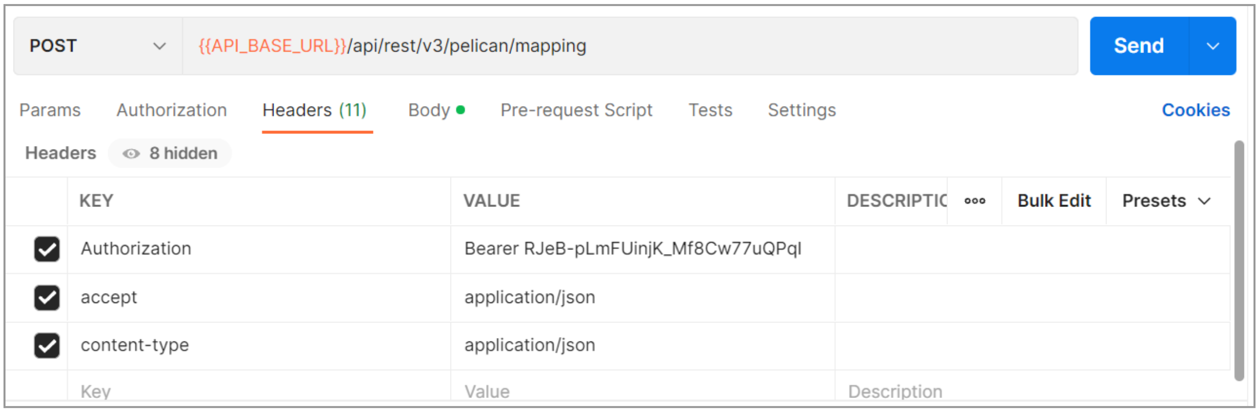
Method: POST

Description: This API creates a new table mapping on the PELICAN instance.

Required Permission: Save Mapping

Headers:

Authorization:	Bearer aKyCgkmoAvK1TBafTVwCp_HYo
accept:	application/json
content-type:	application/json



{{API_BASE_URL}}/api/rest/v3/pelican/mapping

	KEY	VALUE	DESCRIPTIC
☑	Authorization	Bearer RJeB-pLmFUinjK_Mf8Cw77uQPql	
☑	accept	application/json	
☑	content-type	application/json	
	Key	Value	Description

Request Body:

```
{
 "approximationThreshold": 0,
 "enableApproximateMapping": true,
 "enablePhoneticMatch": true,
 "ignoreColumnList": [
          "string"
 ],
 "persistMapping": true,
 "sourceDatabaseName": "pelican",
 "sourceDatastoreName": "TD",
 "sourceTableName": "Test_500",
 "sourceTablePredicate": "string",
 "tableMapUniqueId": "id",
 "targetDatabaseName": "pelican",
 "targetDatastoreName": "BQ",
 "targetTableName": "test_500",
  "targetTablePredicate": "null",

  "timeStampColumnName": "null",

  "uniqueKeyColumn": "id",

  "workspaces": [

          {"id" : 1},

          {"id" : 4}

   ]

}
```

Response: 200 OK

```
{

  "sourceDatastore": "TD",

  "sourceDatabase": "pelican",

  "targetDatastore": "BQ",

  "targetDatabase": "pelican",

  "uniqueTableMappingId": "f850a320f827cdd74c262d205fb34edb",

  "tableMappingResultDTOList": [

          {

          "sourceTablename": "Test_500",

          "targetTablename": "test_500",

          "comparisionDescription": "Table_Mapped",

    "tableMatching": true,

          "sourceTablePredicate": null,

          "targetTablePredicate": null,

          "tableMappingName": "Test_500_1656659253707",

          "columnMappingResultList": [

          {
```

```
   "expressionName": "zip",
   "targetTableColumnName": "zip",
"targetTableColumnDatatype": "INT64",
   "sourceTableColumnName": "zip",
"sourceTableColumnDatatype": "integer",
   "comparisionDescription": null,
   "columnMatching": true,
   "ignore": false,
   "uniqueKeyColumn": false,
   "timeStampColumn": false,
   "columnFilter": null,
   "initialSourceColumnName": "zip",
   "initialTargetColumnName": "zip",
"initialSourceColumnDataType": "integer",
     "initialTargetColumnDataType": "INT64",
"initialSourceColumnLength": 11,
"initialSourceColumnPrecision": null,
"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
   "columnMappingStatus": 1,
   "approximateMatch": false
   },
   {
   "expressionName": "city",
   "targetTableColumnName": "city",
"targetTableColumnDatatype": "STRING",
   "sourceTableColumnName": "city",
    "sourceTableColumnDatatype": "varchar",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
    "uniqueKeyColumn": false,
    "timeStampColumn": false,
    "columnFilter": null,
    "initialSourceColumnName": "city",
    "initialTargetColumnName": "city",
"initialSourceColumnDataType": "varchar",
"initialTargetColumnDataType": "STRING",
"initialSourceColumnLength": 101,
"initialSourceColumnPrecision": null,
"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
    "columnMappingStatus": 1,
    "approximateMatch": false
    },
    {
    "expressionName": "id",
    "targetTableColumnName": "id",
"targetTableColumnDatatype": "INT64",
    "sourceTableColumnName": "id",
"sourceTableColumnDatatype": "integer",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
"uniqueKeyColumn": true,
    "timeStampColumn": false,
    "columnFilter": null,
    "initialSourceColumnName": "id",
    "initialTargetColumnName": "id",
"initialSourceColumnDataType": "integer",
    "initialTargetColumnDataType": "INT64",
"initialSourceColumnLength": 11,
"initialSourceColumnPrecision": null,
"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
    "columnMappingStatus": 1,
"approximateMatch": false
    },
    {
    "expressionName": "name",
    "targetTableColumnName": "name",
"targetTableColumnDatatype": "STRING",
    "sourceTableColumnName": "name",
    "sourceTableColumnDatatype": "varchar",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
    "uniqueKeyColumn": false,
    "timeStampColumn": false,
    "columnFilter": null,
    "initialSourceColumnName": "name",
    "initialTargetColumnName": "name",
"initialSourceColumnDataType": "varchar",
"initialTargetColumnDataType": "STRING",
"initialSourceColumnLength": 101,
"initialSourceColumnPrecision": null,
"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
    "columnMappingStatus": 1,
    "approximateMatch": false
    },
    {
    "expressionName": "age",
    "targetTableColumnName": "age",
"targetTableColumnDatatype": "INT64",
    "sourceTableColumnName": "age",
"sourceTableColumnDatatype": "integer",
    "comparisionDescription": null,
    "columnMatching": true,
    "ignore": false,
        "uniqueKeyColumn": false,
    "timeStampColumn": false,
    "columnFilter": null,
    "initialSourceColumnName": "age",
    "initialTargetColumnName": "age",
"initialSourceColumnDataType": "integer",
    "initialTargetColumnDataType": "INT64",
"initialSourceColumnLength": 11,
        "initialSourceColumnPrecision": null,

        "initialTargetColumnLength": null,

        "initialTargetColumnPrecision": null,

            "columnMappingStatus": 1,

                "approximateMatch": false

            }

            ],

            "approximateMatch": false

            }

 ],

 "workspaces": [

            {"id" : 1},

            {"id" : 4}

  ]

}
```

API 3: Edit Pelican Table Mapping Using Public API Version 3

URL: http://<IP of PELICAN Machine>/api/rest/v3/pelican/mapping/edit

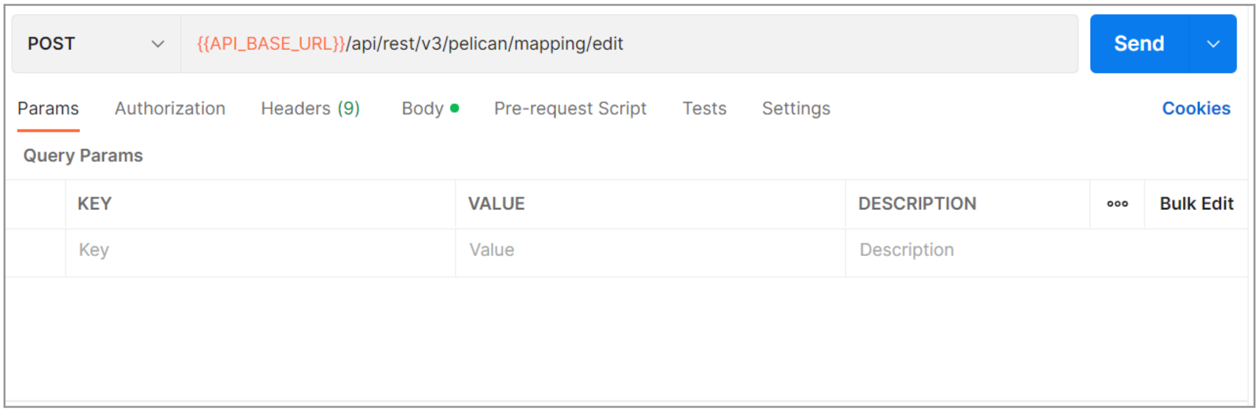
Method: POST

Description: This API edits an existing mapping.

Required Permission: Save Mapping

Headers:

Authorization: Bearer aKyCgkmoAvK81TLBafTVwCp_HYo



POST {{API_BASE_URL}}/api/rest/v3/pelican/mapping/edit

	KEY	VALUE	DESCRIPTION	ooo	Bulk Edit
	Key	Value	Description		

Request Body:

```
{
 "pelicanMappingResult": {
          "sourceDatabase": "pelican",
          "sourceDatastore": "Teradata_QA",
     "sourceTableFilter":"customer_id=300",
   "targetTableFilter":"customer_id=300",
   "tags":["test","test2"],
          "tableMappingResultDTOList": [
          {
          "columnMappingResultList": [
          {
          "columnFilter": null,
          "columnMappingStatus": 0,
          "columnMatching": true,
          "comparisionDescription": "null",
          "expressionName": "name",
          "ignore": true,
        "initialSourceColumnDataType": "string",
"initialSourceColumnLength": 0,

"initialSourceColumnName": "name",

  "initialSourceColumnPrecision": 0,

"initialTargetColumnDataType": "STRING",

"initialTargetColumnLength": 0,

"initialTargetColumnName": "name",

"initialTargetColumnPrecision": 0,

"sourceTableColumnDatatype": "int",

  "sourceTableColumnName": "cast(name as integer)",

"targetTableColumnDatatype": "int64",

  "targetTableColumnName": "cast(name as int64)",

  "timeStampColumn": false,

  "uniqueKeyColumn": false

     }

  ],

  "comparisionDescription": null,

  "sourceTablePredicate": null,

  "sourceTablename": "Test_500",

  "tableMappingName": "Test_500_1663757713597",

  "tableMatching": true,

  "targetTablePredicate": null,

  "targetTablename": "test_500"

  }

  ],

  "targetDatabase": "pelican",

  "targetDatastore": "Bigquery_QA",

  "uniqueTableMappingId": "9c6f89ce882828e2585e4b249bf31d3f",
          "workspaces": [
          {"id" : 1},
          {"id" : 4}
          ]
 },
 "saveAsNewMapping": false
}
```

Response: 200 OK

```
{
 "mappingId": "9c6f89ce882828e2585e4b249bf31d3f",
 "mappingName": "Test_500_1663757713597"
}
```

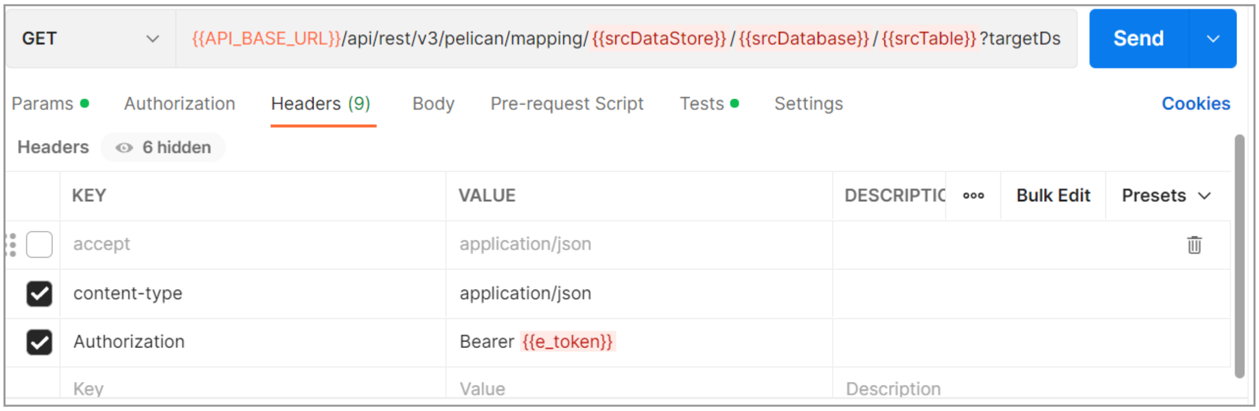
API 4: Fetches All Table Mapping Using Public API Version 3

URL: http:<IP of PELICAN Machine>/api/rest/v3/pelican/mapping/Teradata_Demo/pelican/CT1?targetDsName=BigQuery_Demo&targetDbName=pelican&targetTableName=CT1

Method: GET

Description: This API fetches the mappings that exist for the target table that we pass in the parameters.

Required Permission: View Mapping

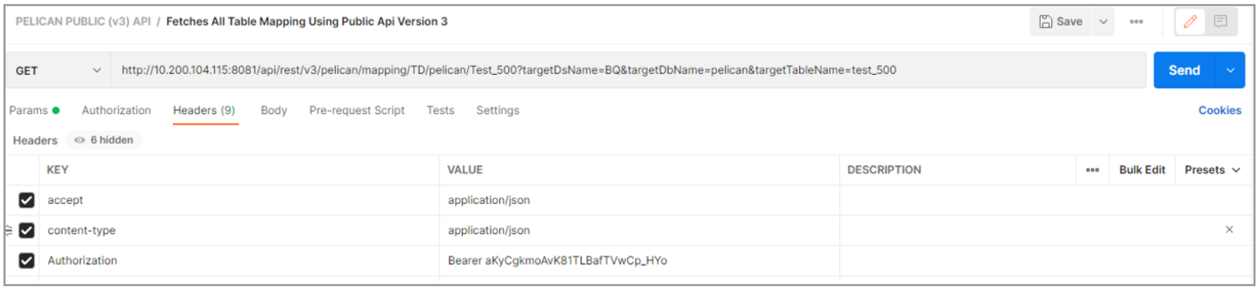


GET {{API_BASE_URL}}/api/rest/v3/pelican/mapping/{{srcDataStore}}/{{srcDatabase}}/{{srcTable}}?targetDs

	KEY	VALUE	DESCRIPTIC
□	accept	application/json	
☑	content-type	application/json	
☑	Authorization	Bearer {{e_token}}	
	Key	Value	Description

Headers:

accept	application/json
content-type	application/json
authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo

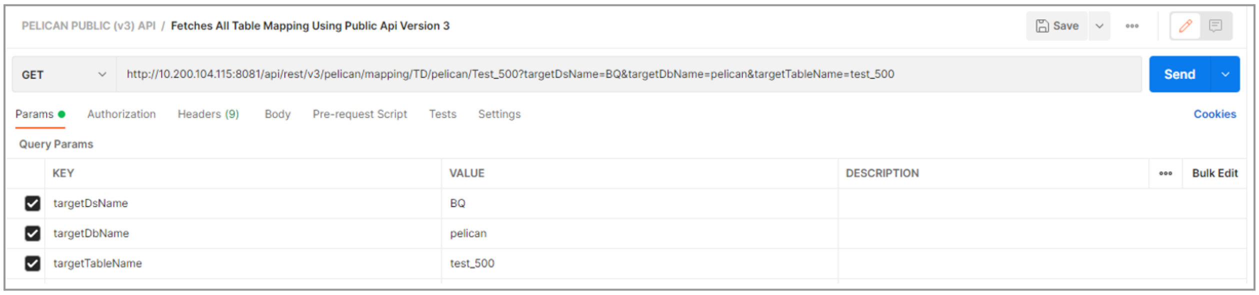


http://10.200.104.115:8081/api/rest/v3/pelican/mapping/TD/pelican/Test_500?targetDsName=BQ&targetDbName=pelican&targetTableName=test_500

	KEY	VALUE	DESCRIPTION
☑	accept	application/json	
☑	content-type	application/json	
☑	Authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo	

Parameters:

targetDsName	BQ
targetDbName	pelican
targetTableName	test_500



http://10.200.104.115:8081/api/rest/v3/pelican/mapping/TD/pelican/Test_500?targetDsName=BQ&targetDbName=pelican&targetTableName=test_500

	KEY	VALUE	DESCRIPTION
☑	targetDsName	BQ	
☑	targetDbName	pelican	
☑	targetTableName	test_500	

Response: 200 OK

```
[
 {
          "mappingId": "f850a320f827cdd74c262d205fb34edb",
          "mappingName": "Test_500_1656659253707"
 },
 {
          "mappingId": "a6b80de67d9375051ff41c85af7a0180",
          "mappingName": "Test_500_1656659253707_clone"
```

}

]

API 5: Fetches Table Mapping Details Using Public API Version 3

URL: http:<IP of PELICAN Machine>/api/rest/v3/pelican/mapping/b102efed8b92729ca1b65b861aeb621d

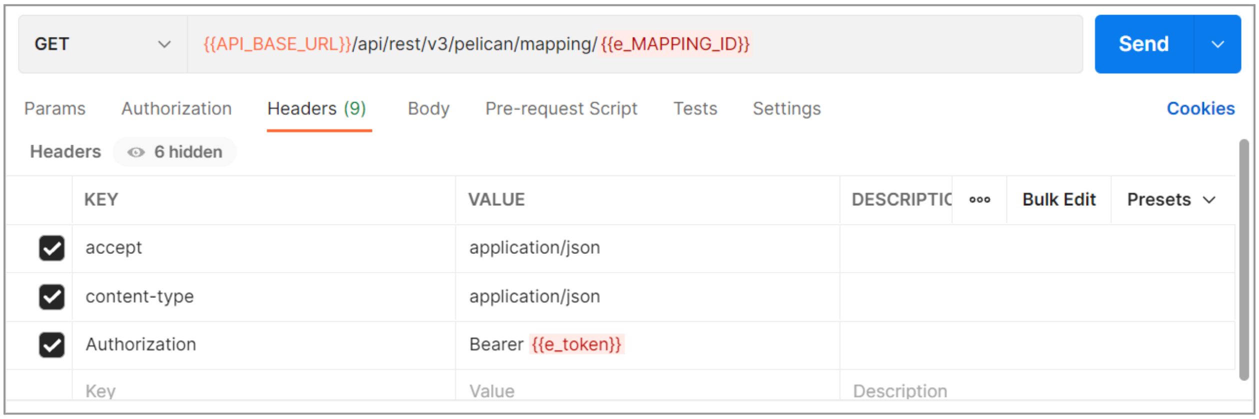
Method: GET

Description: This API fetches the details of a particular mapping for which we pass the mapping ID in the URL.

Required Permission: View Mapping

Headers:

accept	application/json
content-type	application/json
authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo



GET {{API_BASE_URL}}/api/rest/v3/pelican/mapping/{{e_MAPPING_ID}}

	KEY	VALUE	DESCRIPTIC
☑	accept	application/json	
☑	content-type	application/json	
☑	Authorization	Bearer {{e_token}}	
	Key	Value	Description

Response: 200 OK

{

"sourceDatastore": "TD",

"sourceDatabase": "pelican",

"targetDatastore": "BQ",

"targetDatabase": "pelican",

"uniqueTableMappingId": "f850a320f827cdd74c262d205fb34edb",

"tableMappingResultDTOList": [

```
  {
  "sourceTablename": "Test_500",
  "targetTablename": "test_500",
  "comparisionDescription": "Mapping for Test_500_1656659253707",
  "tableMatching": true,
  "sourceTablePredicate": null,
  "targetTablePredicate": null,
  "tableMappingName": "Test_500_1656659253707",
  "columnMappingResultList": [
  {
  "expressionName": "age",
  "targetTableColumnName": "age",
"targetTableColumnDatatype": "INT64",
  "sourceTableColumnName": "age",
"sourceTableColumnDatatype": "integer",
  "comparisionDescription": "age",
  "columnMatching": false,
  "ignore": false,
  "uniqueKeyColumn": false,
  "timeStampColumn": false,
  "columnFilter": null,
  "initialSourceColumnName": "age",
  "initialTargetColumnName": "age",
"initialSourceColumnDataType": "integer",
"initialTargetColumnDataType": "INT64",
"initialSourceColumnLength": 11,

"initialSourceColumnPrecision": null,

"initialTargetColumnLength": null,

"initialTargetColumnPrecision": null,

   "columnMappingStatus": 1

   },

   {

   "expressionName": "id",

   "targetTableColumnName": "id",

"targetTableColumnDatatype": "INT64",

   "sourceTableColumnName": "id",

"sourceTableColumnDatatype": "integer",

   "comparisionDescription": "id",

   "columnMatching": false,

   "ignore": false,

   "uniqueKeyColumn": true,

   "timeStampColumn": false,

   "columnFilter": null,

   "initialSourceColumnName": "id",

   "initialTargetColumnName": "id",

"initialSourceColumnDataType": "integer",

   "initialTargetColumnDataType": "INT64",

"initialSourceColumnLength": 11,

"initialSourceColumnPrecision": null,

"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
    "columnMappingStatus": 1
    },
    {
    "expressionName": "zip",
    "targetTableColumnName": "zip",
"targetTableColumnDatatype": "INT64",
    "sourceTableColumnName": "zip",
"sourceTableColumnDatatype": "integer",
    "comparisionDescription": "zip",
    "columnMatching": false,
    "ignore": false,
    "uniqueKeyColumn": false,
    "timeStampColumn": false,
    "columnFilter": null,
    "initialSourceColumnName": "zip",
    "initialTargetColumnName": "zip",
"initialSourceColumnDataType": "integer",
"initialTargetColumnDataType": "INT64",
"initialSourceColumnLength": 11,
"initialSourceColumnPrecision": null,
"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
    "columnMappingStatus": 1
    },
   {
   "expressionName": "name",
   "targetTableColumnName": "name",
"targetTableColumnDatatype": "STRING",
   "sourceTableColumnName": "name",
"sourceTableColumnDatatype": "varchar",
   "comparisionDescription": "name",
   "columnMatching": false,
   "ignore": false,
   "uniqueKeyColumn": false,
   "timeStampColumn": false,
       "columnFilter": null,
   "initialSourceColumnName": "name",
   "initialTargetColumnName": "name",
"initialSourceColumnDataType": "varchar",
"initialTargetColumnDataType": "STRING",
   "initialSourceColumnLength": 101,
"initialSourceColumnPrecision": null,
"initialTargetColumnLength": null,
"initialTargetColumnPrecision": null,
   "columnMappingStatus": 1
   },
   {
   "expressionName": "city",
   "targetTableColumnName": "city",
        "targetTableColumnDatatype": "STRING",
           "sourceTableColumnName": "city",
        "sourceTableColumnDatatype": "varchar",
           "comparisionDescription": "city",
           "columnMatching": false,
              "ignore": false,
           "uniqueKeyColumn": false,
           "timeStampColumn": false,
           "columnFilter": null,
           "initialSourceColumnName": "city",
           "initialTargetColumnName": "city",
        "initialSourceColumnDataType": "varchar",
        "initialTargetColumnDataType": "STRING",
        "initialSourceColumnLength": 101,
        "initialSourceColumnPrecision": null,
        "initialTargetColumnLength": null,
        "initialTargetColumnPrecision": null,
            "columnMappingStatus": 1
           }
           ]
           }
 ],
"workspaces": [
           {"id" : 1},
           {"id" : 4}
           ]
```

}

API 6: Validates Table Using Public API Version 3

URL: POST

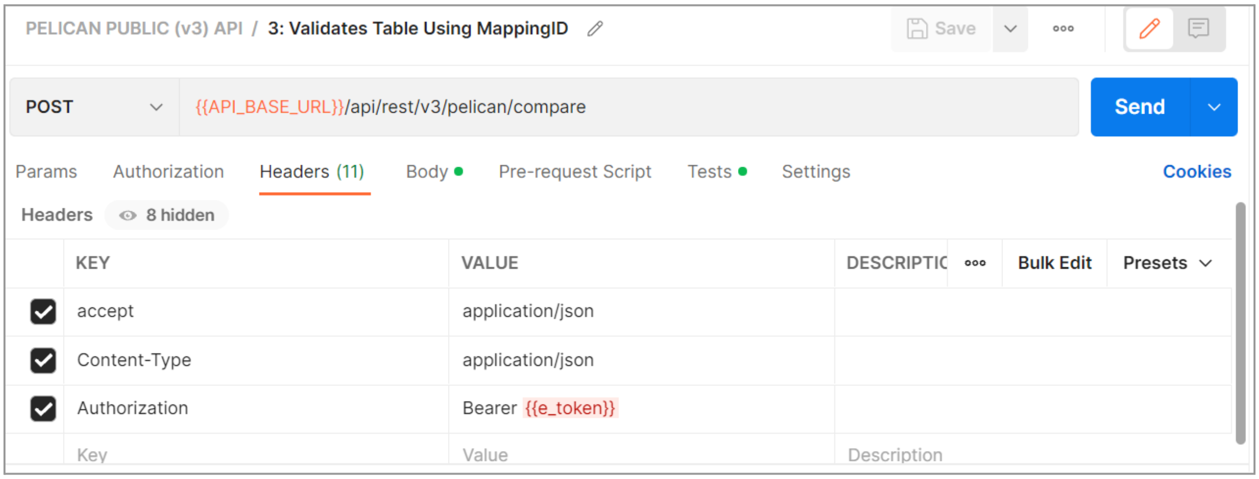
Method: http:<IP of PELICAN Machine>/api/rest/v3/pelican/compare

Description: This API creates a scheduler on the mapping ID we pass in the request body and executes it.

Required Permission: Execute Scheduler

Headers:

Accept	application/json
Content-Type	application/json
Bearer:	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo



POST {{API_BASE_URL}}/api/rest/v3/pelican/compare

	KEY	VALUE	DESCRIPTIC
☑	accept	application/json	
☑	Content-Type	application/json	
☑	Authorization	Bearer {{e_token}}	
	Key	Value	Description

Request Body:

```
{
  "enableScheduling": false,
  "recurrencePattern": "null",
  "recurrencePatternDescription": "null",
  "tableMappingId": "9c6f89ce882828e2585e4b249bf31d3f",
  "validationEndDate": "06/21/2022",
  "validationStartDate": "06/21/2022",
```

"validationpackage": "FULL"

}

Response: 200 OK

```
{
  "requestId": "a19c130d6f40d61196be5abbc9ac37f51663758114731",
  "message": "Scheduler successfully started using V3 API"
}
```

API 7: Fetches All Pelican Execution Details Using Public Api Version 3

URL: http://<IP of PELICAN Machine>/api/rest/v3/pelican/executiondetails?executioncount=3&needsample=true&page=1&size=5

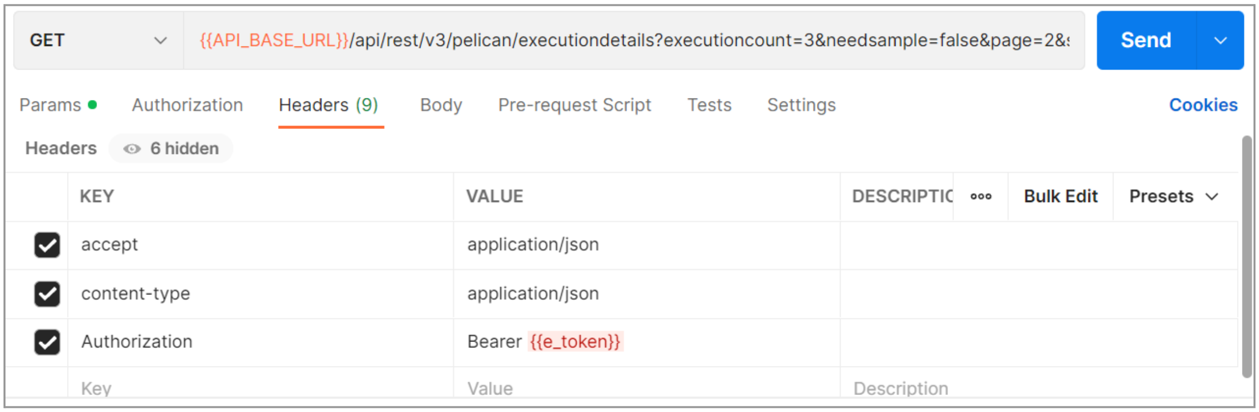
Method: GET

Description: This API fetches all the execution details across pelican instances.

Required Permission: View Results, View Sample (Optional to view samples)

Headers:

Accept	application/json
Content-type	application/json
Authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo



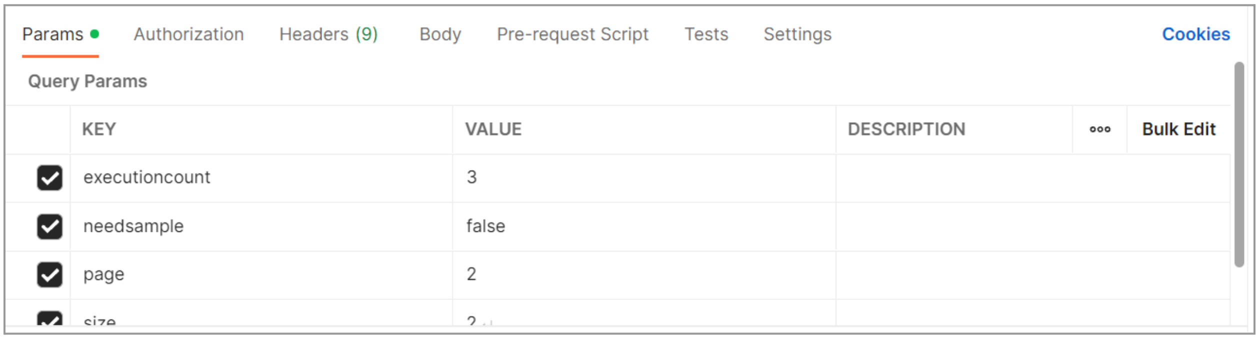
GET {{API_BASE_URL}}/api/rest/v3/pelican/executiondetails?executioncount=3&needsample=false&page=2&

	KEY	VALUE	DESCRIPTIC
☑	accept	application/json	
☑	content-type	application/json	
☑	Authorization	Bearer {{e_token}}	
	Key	Value	Description

Parameters:

executioncount	1
needsample	false

page	2
size	2



	KEY	VALUE	DESCRIPTION
☑	executioncount	3	
☑	needsample	false	
☑	page	2	
☑	size	2	

Response: 200 OK

```
[
 {
  "tokenId": null,
       "count": null,
  "tableName": "test_500",
  "dbName": "pelican",
  "dataStoreHost": null,
  "executionDetails": null
 },
 {
  "tokenId": null,
       "count": null,
  "tableName": "Test_500",
  "dbName": "pelican",
       "dataStoreHost": "10.200.104.66",
  "executionDetails": [
       {
```

```
      "status": false,

      "startTime": "2022-09-21T05:35:50.994+00:00",

      "endTime": "2022-09-21T05:35:52.900+00:00",

      "sourceCount": null,

      "failureType": "DATA_VALIDATION",

      "failureSystemType": "PELICAN",

      "failureCount": 99,

      "description": null,

      "sourceSampleData": null,

      "destinationSampleData": null

          }

          ]

 }

]
```

API 8: Fetches Pelican Execution By Mapping Id Using Public API Version 3

URL: http://<IP of PELICAN Machine>/api/rest/v3/pelican/validationresult/mappingid/b102efed8b92729ca1b65b861aeb621d?executioncount=1&needsample=true

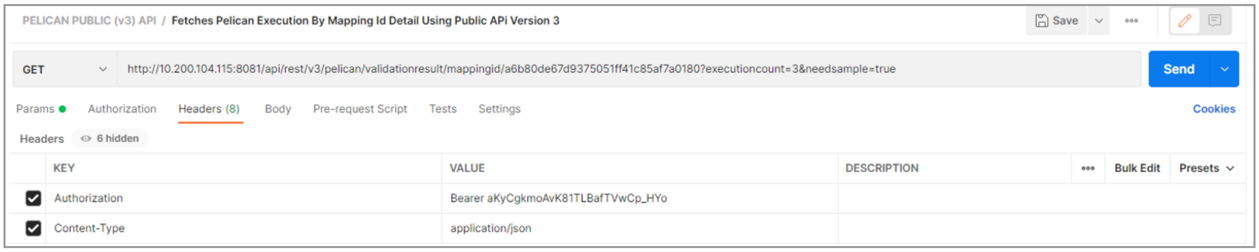
Method: GET

Description: This API fetches the execution details of the given mapping ID passed in the URL.

Required Permission: View Results, View Sample (Optional to view samples)

Headers:

Accept:	application/json
Authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo
Content-Type	application/json



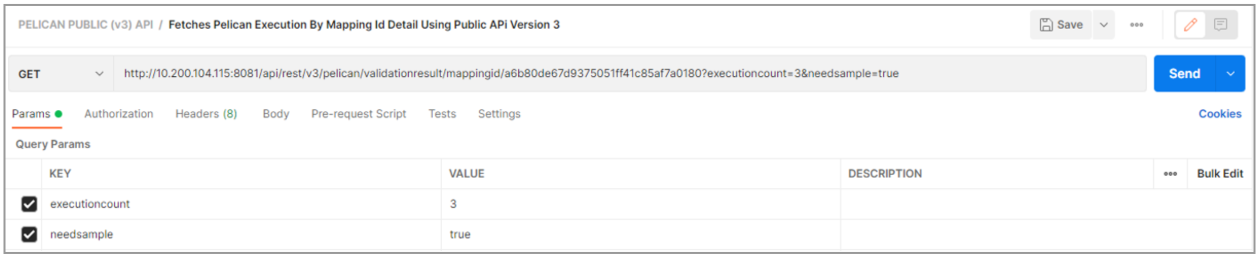
PELICAN PUBLIC (v3) API / Fetches Pelican Execution By Mapping Id Detail Using Public APi Version 3

GET http://10.200.104.115:8081/api/rest/v3/pelican/validationresult/mappingid/a6b80de67d9375051ff41c85af7a0180?executioncount=3&needsample=true

	KEY	VALUE	DESCRIPTION
☑	Authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo	
☑	Content-Type	application/json	

Parameters:

executioncount	**3**
needsample	**true**



PELICAN PUBLIC (v3) API / Fetches Pelican Execution By Mapping Id Detail Using Public APi Version 3

GET http://10.200.104.115:8081/api/rest/v3/pelican/validationresult/mappingid/a6b80de67d9375051ff41c85af7a0180?executioncount=3&needsample=true

	KEY	VALUE	DESCRIPTION
☑	executioncount	3	
☑	needsample	true	

Response: 200 OK

```
{

  "tokenId": null,

  "count": null,

  "tableName": "Test_500",

  "dbName": "pelican",

  "dataStoreHost": "10.200.104.66",

  "executionDetails": [

            {

            "status": false,

            "startTime": "2022-07-01T07:36:25.348+00:00",

            "endTime": "2022-07-01T07:36:28.862+00:00",

            "sourceCount": null,
            "failureType": "DATA_VALIDATION",

            "failureSystemType": "PELICAN",

            "failureCount": null,

            "description": null,

            "sourceSampleData": null,

            "destinationSampleData": null

            },

            {

            "status": false,

            "startTime": "2022-07-01T07:35:47.556+00:00",

            "endTime": "2022-07-01T07:35:50.168+00:00",

            "sourceCount": null,

            "failureType": "DATA_VALIDATION",

            "failureSystemType": "PELICAN",

            "failureCount": 1,

            "description": null,

            "sourceSampleData": null,

            "destinationSampleData": null

            }

  ]

}
```

API 9: Fetches Scheduler Result By request Id Using Public API Version 3

URL: http://<IP of PELICAN Machine>/api/rest/v3/pelican/validationresult/7563b8b2fd2ffe9175a4ef4f7c3491821668164607034

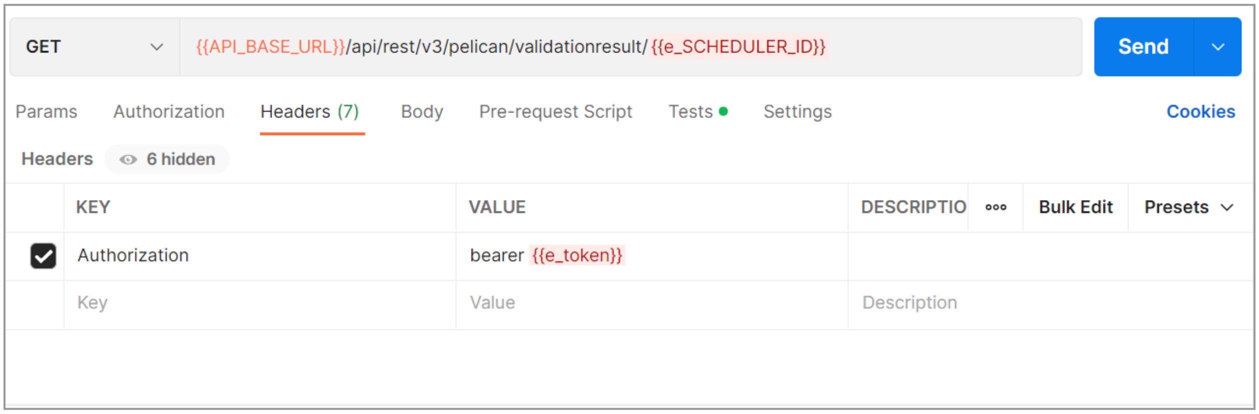
Method: GET

Description: This API fetches the scheduler result by request ID.

Required Permission: View Results

Headers:

Content-Type :	application/json
Accept:	application/json
Authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo



GET {{API_BASE_URL}}/api/rest/v3/pelican/validationresult/{{e_SCHEDULER_ID}}

	KEY	VALUE	DESCRIPTIO
☑	Authorization	bearer {{e_token}}	
	Key	Value	Description

Response: 200 OK

```
{
 "uniqueId": "a19c130d6f40d61196be5abbc9ac37f5",
 "createTimeStamp": 1663758114731,
 "status": "Completed",
 "pelicanExecutionDetails": [
         {
         "startTime": "21 Sep 2022, 04:31:54.884",
         "endTime": "21 Sep 2022, 04:32:59.530",
         "status": "Failed",
   "sourceDatastore": "10.200.104.74",
"sourceDatabaseName": "pelican",

"sourceTableName": "Test_500",

"targetDatastore": "https://www.googleapis.com//biqquery//v2",

"targetDatabaseName": "pelican",

"targetTableName": "test_500",

"pelicanResults": [

    {

  "resultType": "DATA TRANSFER RESULT",

  "status": "Successful",

  "description": "Data transfer successful.",

      "pelicanResultProperties": [],

  "startTime": "21 Sep 2022, 04:32:13.214",

  "endTime": "21 Sep 2022, 04:32:59.087"

    },

    {

  "resultType": "VALIDATION RESULT",

  "status": "Failed",

  "description": "Data validation failed.",

  "pelicanResultProperties": [

    {

    "keyName": "hashValues",

    "referenceValue": "E1A6C1BF5797BE8026C36C133120B43FF4AACA44",

    "destinationValue": "EADEFBD31CACF3EA54957A3B54D67F1F3C6837D1"

    },
{
"keyName": "FullMismatchData",
"referenceValue": "pelican.source_Test_500_761a9f5753ec36722e1f2b22adabcfbc",
"destinationValue": "pelican_temp.target_test_500_761a9f5753ec36722e1f2b22adabcfbc"
},
{
"keyName": "EXTRA",
"referenceValue": "0",
"destinationValue": "120"
},
{
"keyName": "MISSING",
"referenceValue": "0",
"destinationValue": "120"
},
{
"keyName": "MismatchedRowCount",
"referenceValue": "0",
"destinationValue": "240"
},
{
"keyName": "isProbabilistic",
"referenceValue": "true"
},
          {

          "keyName": "RowCount",

          "referenceValue": "500",

          "destinationValue": "500"

          }

          ],

        "startTime": "21 Sep 2022, 04:32:13.214",

        "endTime": "21 Sep 2022, 04:32:59.087"

          }

          ]

          }

 ]

}
```

API 10: Delete Pelican Mapping By Table Mapping Id Using Public API Version 3

URL: http://<IP of PELICAN Machine>/api/rest/v3/pelican/mapping/5f65d9ab18782aadc562137498c8f7e2?force=true

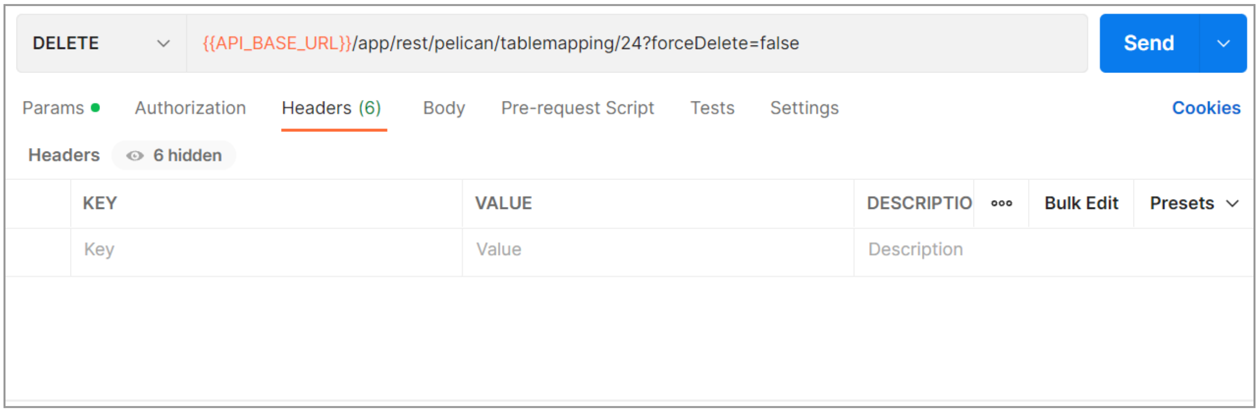
Method: DELETE

Description: This API deletes an existing mapping using mapping ID as an input.

Required Permission: Delete Mapping

Headers:

Authorization	Bearer aKyCgkmoAvK81TLBafTVwCp_HYo



DELETE {{API_BASE_URL}}/app/rest/pelican/tablemapping/24?forceDelete=false

Response: 200 OK

```
{
  "tableMappingId": "f914411dac168f26a193140ac6920e17",
  "deletionStatus": "Success",
  "description": "Deleting Table Mapping : Test_500  as there are no schedulers attached to it."
}
```

19. Automatic Backup

- The frequency of automatic backup is now made configurable. Frequency can be changed using a cron pattern in the application.properties file.
- Only 5 latest backup files are maintained at the designated location, as the oldest file gets replaced by the newest one.
- The backup location is as follows:
  - ➢ For docker and kubernetes:

    /usr/local/apache-tomcat-${TOMCAT_VERSION}/dbbackup

  - ➢ For Jar based installation on VM:

    /(path of tomcat server}/dbbackup

20. Complex Data Type Support

The details of the Data types supported by the following datastores are as follows:

- **Hive**

  We are supporting Struct, Array and Map data types.
- **BQ**

  We are supporting Struct and Array data types.

21. Frequently Asked Questions

1. **How to view the validation result in Pelican?**

   The validation result can be viewed only for those tables whose table mapping is done. So, if you are a new user and then you need to follow the below given steps:

   Step 1: First, you need to configure a data store.

   Step 2: Followed by datastore configuration, you need to map tables between source and destination datastore.

   Step 3: Next, you need to configure the scheduler for the saved mapping

   Step 4: Finally, go to Reports and click Validation Result. For more information, refer to the **Validation Result** topic.

2. **Which are the various data stores the Pelican supports?**

   Pelican supports comparison between various data stores. For more information, refer to the Data Store Support topic.

3. **How to validate tables between source and destination datastore?**

   You can validate tables between source and destination Datastore using Validation Configuration functionality. For more information, refer to the **Validation Mapping** topic.

4. **What is the scheduler and how to configure it?**

   The Scheduler allows the user to execute processes at a regular time interval. In Pelican, the user can create a scheduler for a saved mapping; so that, after a time period the scheduler executes the process and it validates the source table with destination tables. For more information, refer to the Scheduler Configuration topic.

5. **How to configure the email notification?**

   Email notification functionality allows the user to send an email automatically to various users on the execution of the scheduler. For more information, refer to the **Email Notification** topic.

6. **How to update the pelican license?**

Once you received the pelican license file, i.e. licence.pel follow the following steps.

Step 1: Log in to the Pelican.

Step 2: Administration -> Add License.

Step 3: Click on the upload and select the Pelican.pel from the file system.

Step 4: Again login to continue the usage.

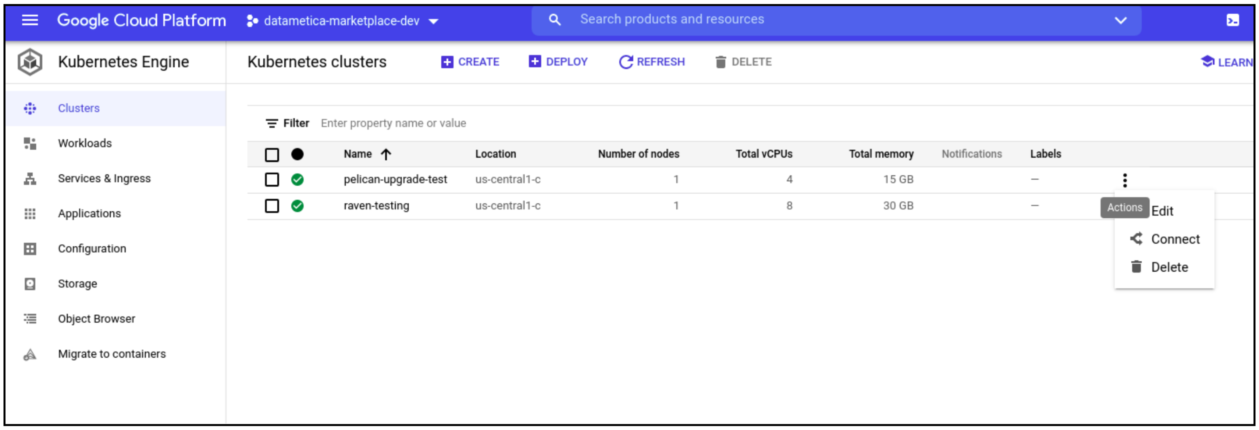
7. **Who can create a LDAP user?**

Only the superuser has the right to create LDAP users. LDAP users can't create any new users. Only users with Admin access can create new LDAP users.

8. **How to get GKE server(Pelican services) start-stop access permissions to QA Lead?**

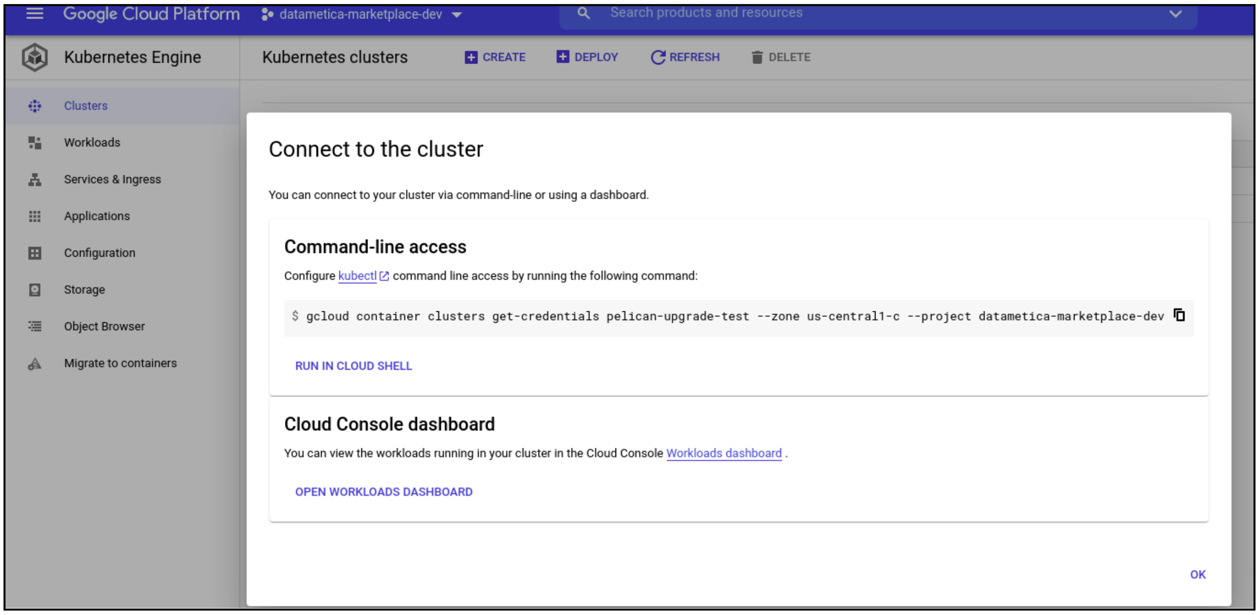
Step 1 - Go to GCP console > Search for GKE.

Step 2 - Click on Connect your cluster.



☐	●	Name ↑	Location	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
☐	✓	pelican-upgrade-test	us-central1-c	1	4	15 GB		—
☐	✓	raven-testing	us-central1-c	1	8	30 GB		—

Step 3 - Click on Run in cloud shell.



Connect to the cluster

You can connect to your cluster via command-line or using a dashboard.

Command-line access

Configure kubectl command line access by running the following command:

```
$ gcloud container clusters get-credentials pelican-upgrade-test --zone us-central1-c --project datametica-marketplace-dev
```

RUN IN CLOUD SHELL

Cloud Console dashboard

You can view the workloads running in your cluster in the Cloud Console Workloads dashboard .

OPEN WORKLOADS DASHBOARD

OK

Step 4 - Run this command after login on cluster - **kubectl get pods**

Step 5 - Copy the web pod name and run this command to restart the service - **kubectl delete pod name.**

Note - Pelican service pod will be automatically started and old one will be deleted.

9. **How to get Application log file url along with access permissions to QA user (catalinaout.log and application.log) ?**

Step 1 - To see the application logs, we need to go inside the pelican pod. Run this command to go inside the pod - **kubectl exec -it podname bash**

Step 2 - Now you can view the logs on this file path - **/usr/local/apache-tomcat-8.5.65/logs/application_logs/log.log**

10. **How to add a lineage file?**

To add a lineage file, login to the system, go to Administration and select Add lineage. The window will be displayed on the screen. For more information, visit the Add Lineage page.

11. **What is a CSV file?**

The CSV file is a text file in which field values are separated by commas and each record is separated by a new line character. First row can optionally be a header for field values.

22. Glossary

Three Level Hierarchy:

The three level DBMS architecture has following three levels:

- External or User view
- Conceptual or Logical
- Internal or Physical

External Level:

It is also called the view level as several users can view their required data by internally fetching it from the database with the help of internal and conceptual level mapping. This level is the top-most level of the three level database architecture.

Conceptual Level:

Also referred to as the logical level it describes the entire database design. The relationship between the data, schema of data etc. are defined at this level by the Database administrator. Security and database constraints are also defined at the conceptual level.

Internal Level:

The Internal level is also known as the Physical level. It describes how the database is actually stored in terms of the record layout of files and type of files (hash, b-tree, flat). The Internal level is the lowest level of database architecture.