



STAR\*M  
SELF SERVICE USER GUIDE  
ENTERPRISE EDITION  
VERSION 1.1

### Contents

1. Introduction .....	2
2. Platform Overview .....	3
3. Organization Management .....	4
Creating an Organization.....	4
Edit or Delete an Organization .....	5
4. Project Management .....	6
Creating a Project.....	6
Edit or Delete a Project .....	7
5. Project Maintenance (Bulk Options).....	8
6. Project Details & File Management .....	9
7. Dashboard .....	10
8. Schema Migration .....	12
Step 1: Schema Migration Setup.....	12
Step 2: Run Schema Migration.....	14
Step 3: Assessment Report .....	20
Step 4: Migration Summary Report .....	31
9. Data Migration .....	35
Data Migration - Database Setup.....	35
Run Data Migration .....	35
Data Validation .....	40
10. SQL Extraction & Remediation .....	43
Import Application Code.....	43
Run Extractor & Remediation .....	44
11. Ad Hoc Transition .....	45
12. License Management.....	47
13. Best Practices .....	48
14. Troubleshooting .....	48
15. Support and Contact Information .....	48

## 1. Introduction

STAR\*M Self-Service is an enterprise-grade database modernization platform developed to accelerate and simplify schema, SQL, and data migration across heterogeneous database environments. The platform enables organizations to independently execute database migration activities within secure, controlled environments while leveraging advanced automation, validation, and reporting. STAR\*M supports migration across major relational database management systems (RDBMS), including Oracle, Microsoft SQL Server, PostgreSQL, MySQL, DB2 LUW, Sybase ASE, Informix, Teradata, Netezza, Ingres Other enterprise and open-source databases.

STAR\*M Self-Service is available through AWS Marketplace, Microsoft Azure Marketplace, Google Cloud Platform (GCP), Oracle Cloud Marketplace, Pre-configured Virtual Machine (VM) deployments, and On-premises customer-hosted environments.

Key capabilities include automated migration assessment and reporting, schema conversion, SQL translation and remediation, embedded SQL extraction, data migration, target data validation, delta automation and version comparison, and Enterprise reporting and analytics. The platform is designed to reduce migration cost, accelerate modernization timelines, improve migration quality, and minimize operational risk.

## 2. Platform Overview

STAR\*M Self-Service follows a structured migration lifecycle consisting of four primary phases:

1. Assessment
2. Translation and Migration
3. Validation and Testing
4. Optimization and Support

The platform architecture is built around the following core modules:

Module	Description
<b>Dashboard</b>	Executive migration analytics and reporting
<b>Organization Management</b>	Organization administration and configuration
<b>Project Management</b>	Project creation and lifecycle management
<b>Schema Migration</b>	Schema conversion and SQL translation
<b>Data Migration</b>	Source-to-target data migration
<b>Validation Framework</b>	Schema and data validation
<b>SQL Extraction</b>	Embedded SQL extraction from applications
<b>Ad Hoc Translation</b>	Interactive SQL translation and validation
<b>Reporting</b>	Migration reporting and analytics
<b>License Manager</b>	License configuration and management

STAR\*M is based on the following hierarchy:

Organization -> Project -> Run Number

An organization can have one or more projects, and each project can have multiple migration runs.

The organization, project and run number hierarchy is established with the organization as the top-level parent, where each organization can have multiple projects. Each project undergoes a run number increment over time. Therefore, each project has many run numbers.

## 3. Organization Management

When a user first logs into the system they must create an organization and then a project before starting the migration process.

### Creating an Organization

To create an organization, the user must click on Organization/Project in the left panel and click on the *Create Organization* button on *Select or create New Organization* page. This will pop up a *Create Organization* Window that enables administrators to create an organizations. This module serves as the top-level administrative layer within the STAR\*M platform.

Administrators can create a new organization by selecting: "Add a New Organization"

Figure 1: Create Organization Window

Required fields include:

Field	Description
Organization Short Name	Short organization identifier
Organization Display Name	Complete organization name
Organization Email Address	A valid mail address of contact person in the organization.

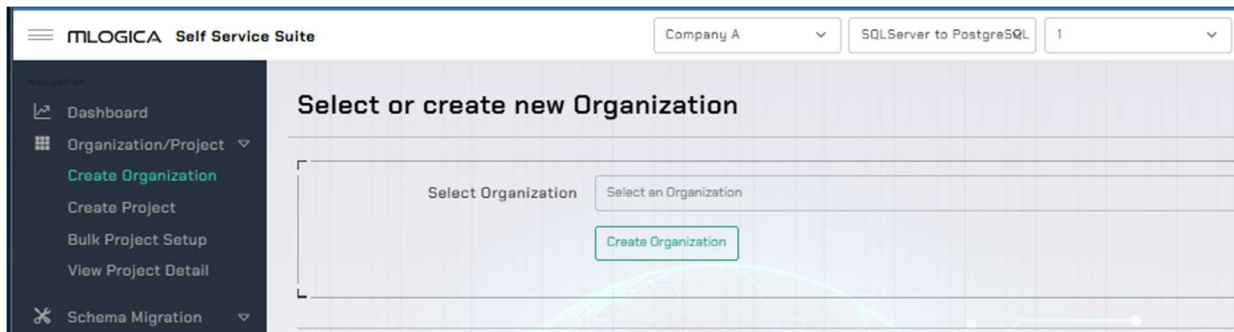
Note:

1. The organization's name and display name have to be unique and can't be repeated.
2. The red star (\*) indicates that the field can't be empty. This field **must** have a value present.

Click on the *Save* button to save the organization information.

Once the organization is created, the user can select the organization by using one of the following two options:

- Selecting the Company Name from the dropdown menu on the top panel
- Clicking on the Organization/Project option in the left panel, clicking on the *Create Organization* and selecting the organization from the drop-down menu.



Select n organization

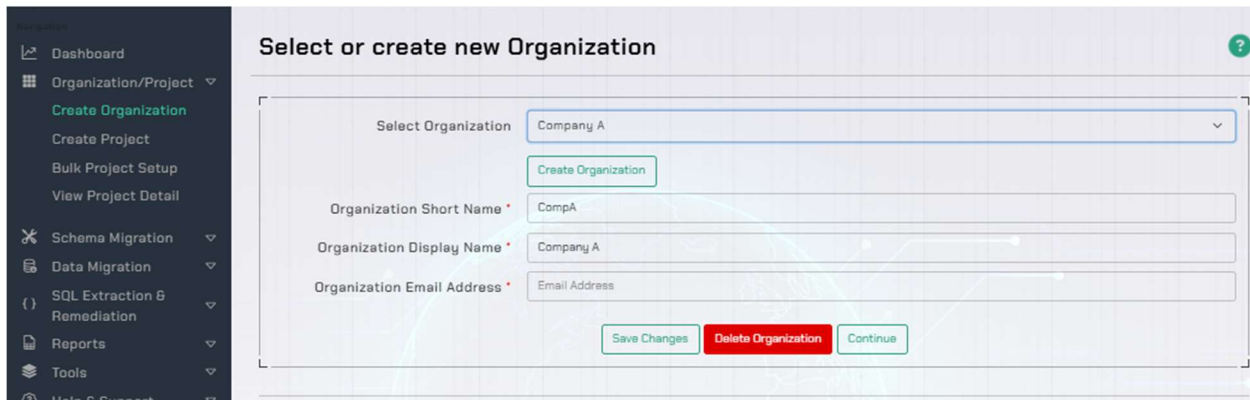
## Edit or Delete an Organization

If the user selects an organization from the dropdown menu on the *Select and Create New Organization* window, the system automatically loads the organization details. The user can modify the content or delete the organization from the system.

To edit the organization information, the user must modify the content and click on the *Save* button after making the changes to the organization information.

To delete the organization from the system, the user must click on the *Delete Organization* button in the *Select or Create New Organization* window. This will remove the organization from the system.

To select the organization for the migration, the user must click on the *continue* button.



Edit existing organization

## 4. Project Management

Once the user has created an organization, they must have at least one project with the source and target database migration selected. Each project is associated with one organization, one source database, one target database and multiple migration runs.

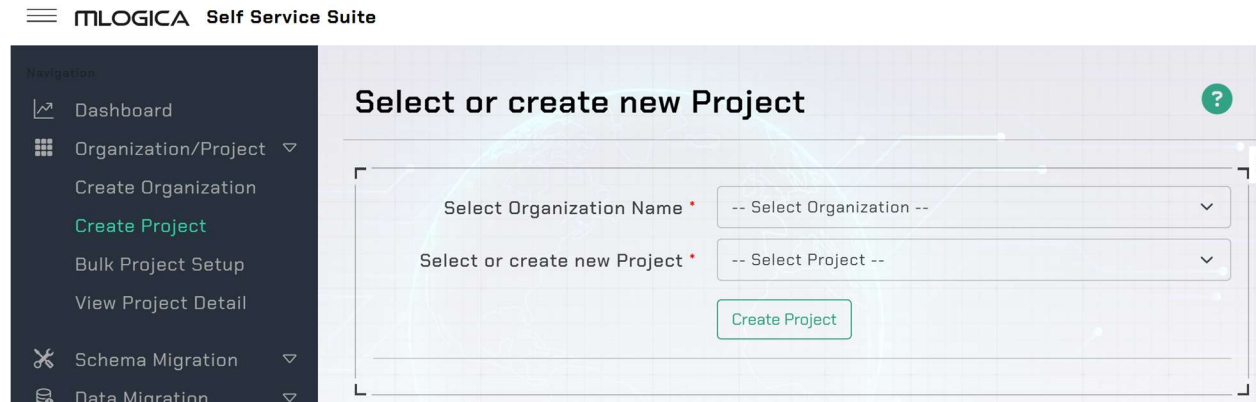


Figure: Select or Create Project Page

### Creating a Project

To create a project, select Organization/Project option in the left panel, and click on the **Create Project** button on the *Select or Create New Project* window. This will pop up a *Create New Project Window* that enables administrators to enter the project information.

*Create Project Window*

Project Configuration Fields include:

Field	Description
<b>Project Name</b>	Migration project name
<b>Estimated Start Date</b>	Planned project start date
<b>Actual Start Date</b>	Actual project start date
<b>Estimated End Date</b>	Planned project completion date
<b>Actual End Date</b>	Actual project completion date

<b>Source Database</b>	Source platform
<b>Target Database</b>	Target platform

Click on the **Save** button to save the project information.

## Edit or Delete a Project

Once the project is created, the user can use one of the following two options to select the project:

- Selecting the Project Name from the drop down menu on the top panel
- Or clicking on the **Organization/Project** option in the left panel, selecting the **Create Project** option and then selecting the organization from the drop-down menu.

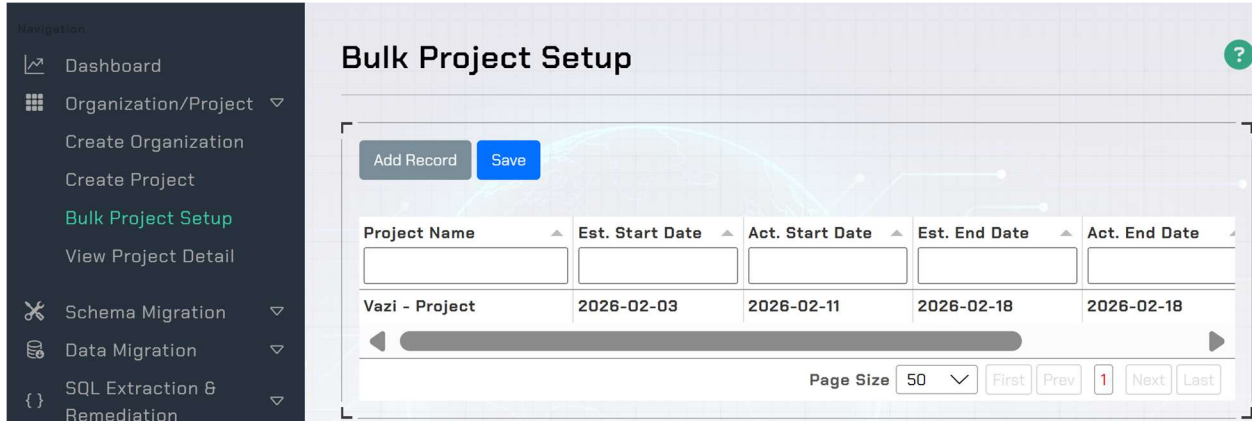
*Prepopulated values for existing project.*

When an existing project is selected:

- An organization can have one or more project
- Existing project details are automatically loaded once the project is selected from the drop down menu.
- Administrators can modify project settings and click on **Update Project** button to save updated project configuration
- Administrators can delete project by clicking on **Delete Project** button to save updated project configuration
- The project's name must unique within the organization
- The red star (\*) indicates that the field can't be empty.

## 5. Project Maintenance (Bulk Options)

Selecting the **Bulk Project Setup** in the left panel allows user to manage multiple project for selected organization. Using **Bult Project Setup** windw allows the user can create projects, update project information, delete projects and perform batch maintenance operations.



Bulk Project Setup action include:

Action	Description
<b>Select Organization</b>	The select organization dropdown option is used to select an organization. The organization name is a mandatory to display projects information associated with the selected organization.
<b>Add Record Button</b>	To inserts new row so that a user can add <i>New Project Name</i> along with other mandatory information related to the project
<b>Delete Row</b>	To delete the selected/checked project during save.
<b>Save Button</b>	Persist all changes made to project for selected organization



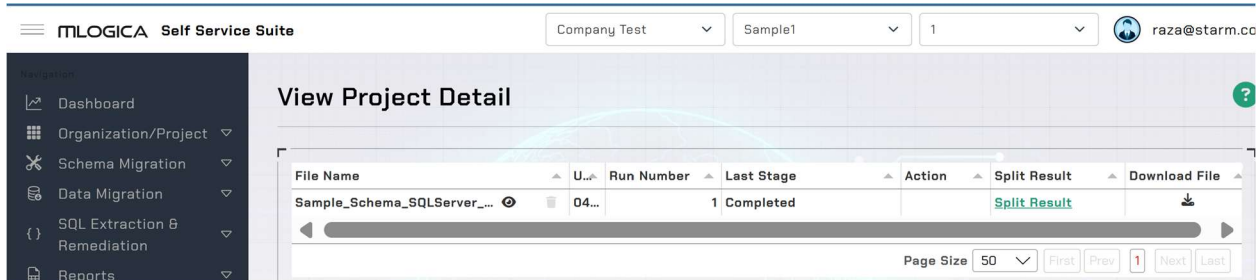
Figure 2: Prepopulated values for existing project. This is to change/update/delete the information.

Bulk Validation Rules include:

- Project names must be unique within the organization
- Mandatory fields (\*) must contain values
- Invalid rows are highlighted for correction

## 6. Project Details & File Management

The **View Project Detail** module provides centralized management of migrated files associated with the selected project. It displays a list of files associate with the selected project and also allows users to add new files either to the existing set (if migration has not been run) or under a new run number. Users can Upload SQL files, download source files, Rename files, Delete files, View migration stages and monitor execution status.



View Project Detail page include supported actions:

Action	Description
<b>Select Organization Name</b>	The <b>Select Organization Name</b> dropdown is used to select an organization. This is mandatory field.
<b>Select Project Name</b>	The <b>Select Project Name</b> dropdown is used to select an existing project associated with selected organization. This is mandatory field.
<b>Last Stage</b>	<b>The Last Stage column</b> displays the migration stage of the selected project and run number.
<b>The Action</b>	<b>The Action column</b> shows the current status—whether the migration is running in the background or completed—and allows the user to stop the migration at the current stage.
<b>Split Result</b>	Users can review SQL splitter results from the interface.
<b>Download File</b>	<b>The Download File column</b> provides a button to download the original uploaded source SQL file.

The list of files associated with the selected project will be displayed in the grid, with options to download, add, delete, or rename existing files.

**File Management** - The system maintains uploaded schema and SQL files for each project and run number.

**Migration Stage Tracking** - The Last Stage column displays the current migration lifecycle stage.

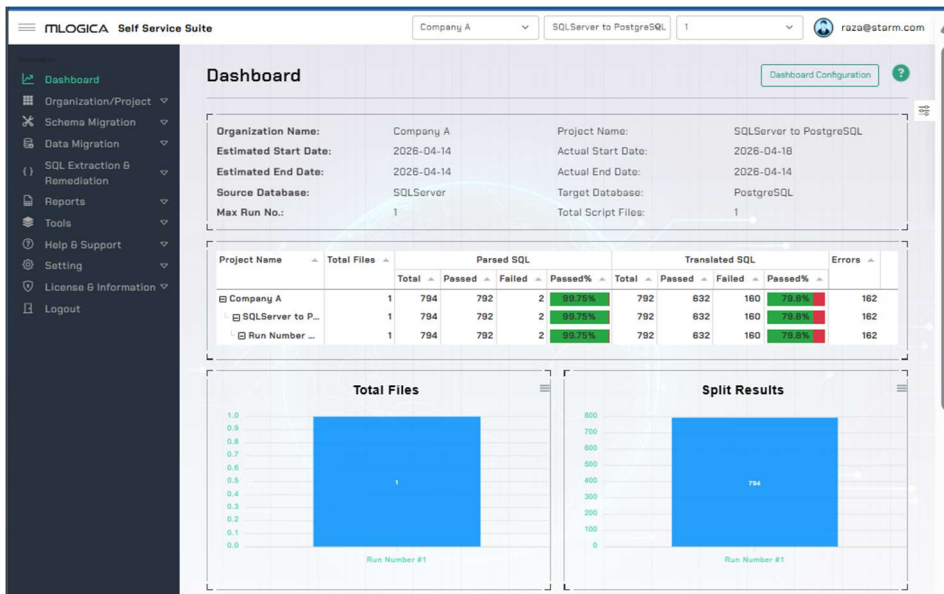
**Migration Status** -The Action column indicates Running, Completed, Failed and Stopped

## 7. Dashboard

Once the user creates the organization and project, the user can select the organization, project, and run number selections from the top navigation bar.



Once the user selects the organization, project and the run number, the application will display the dashboard associated with the project. The Dashboard serves as the primary landing page for STAR\*M Self-Service and provides a centralized analytical overview of migration activities. The dashboard displays project-level metrics, migration statistics, validation results, and operational insights.



Project Level dashboard

The dashboard displays:

Field	Description
<b>Organization Name</b>	Selected organization
<b>Project Name</b>	Selected project
<b>Estimated Start Date</b>	Planned project start date
<b>Actual Start Date</b>	Actual project start date
<b>Estimated End Date</b>	Planned project end date
<b>Actual End Date</b>	Actual project end date
<b>Source Database</b>	Source database platform. Example, <b>SybaseASA</b>
<b>Target Database</b>	Target database platform. Example: <b>PostgreSQL</b> .
<b>Max Run Number</b>	Latest migration run executed
<b>Total Script Files</b>	Number of uploaded source files
<b>Run Number</b>	Selected migration run

Clicking on the **Dashboard Configuration** button, located at the top right corner of the dashboard, pops up the **Set Dashboard Configuration** window that allows user to enable or disable widgets on the dashboard, customize visual layouts, and configure reporting preferences.

**Set Dashboard Configuration** ✕

---

**Summary Charts - Totals**

Select/Unselect ALL

- Total Files
- Split Results
- Delta Automation
- StarM Result
- Target Validation
- Jira Ticket

---

Cancel Save Changes

*Set Dashboard Configuration*

## 8. Schema Migration

Schema Migration module allows users to migrate schema from source database to target database. Please note, that the user must select the organization, project and run number before starting the schema migration project.

The following are the steps to migrate schema from source database to target database:

1. setup the source database and target database.
2. Run Schema Migration
3. Assessment Report

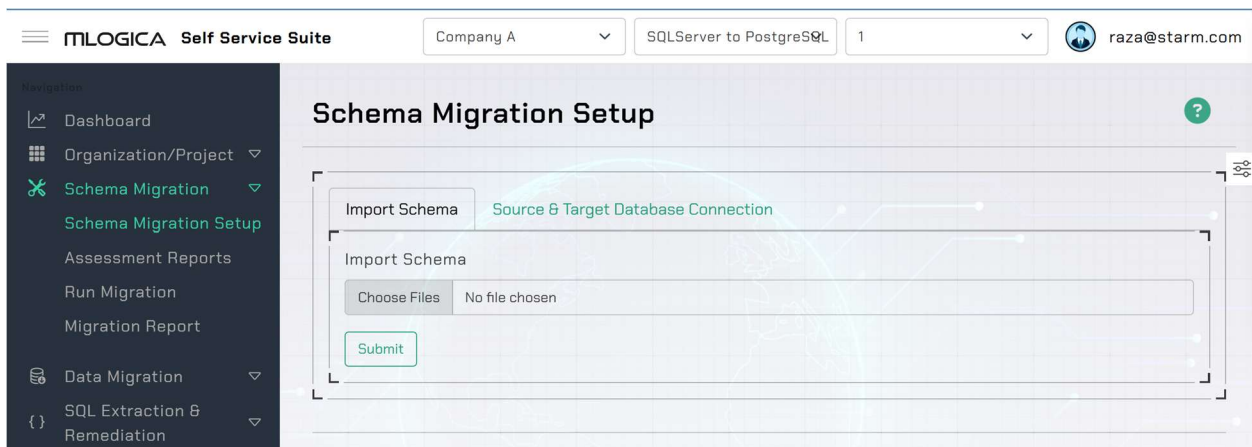
### Step 1: Schema Migration Setup

The **Schema Migration Setup** module, located in the left panel under **Schema Migration**, provides user the option to upload source SQL/schema files or connect directly to source databases for schema extraction. The **Schema Migration** module supports the following schema migration features:

- File-based schema migration
- Direct database connectivity
- Incremental migration runs
- Delta analysis

**File Based Schema Migration:** The Schema Migration Setup page allows administrators to add SQL files to an existing project. When a project is created, it starts with run number 1, and files can be added to this initial run. When the "Run Migration" option is select, the SQL statements in these files are translated to the target database specified during the project's creation.

**Incremental Migration Runs:** As the database evolves to accommodate different schema and data patterns, our system can analyze the differences between snapshots and perform SQL migrations on the modified state. Administrators can add the new state (a new set of SQL files) to run number > 1, and our system will translate these statements accordingly.



Schema Migration Setup

The organization, project and run number hierarchy is established with the organization as the top-level parent, where each organization has multiple projects. Each project undergoes a run number increment over time. Therefore, each project has many run numbers. To add files to a project, the user **must** select the organization, project and run number from the top-level selector as show below:

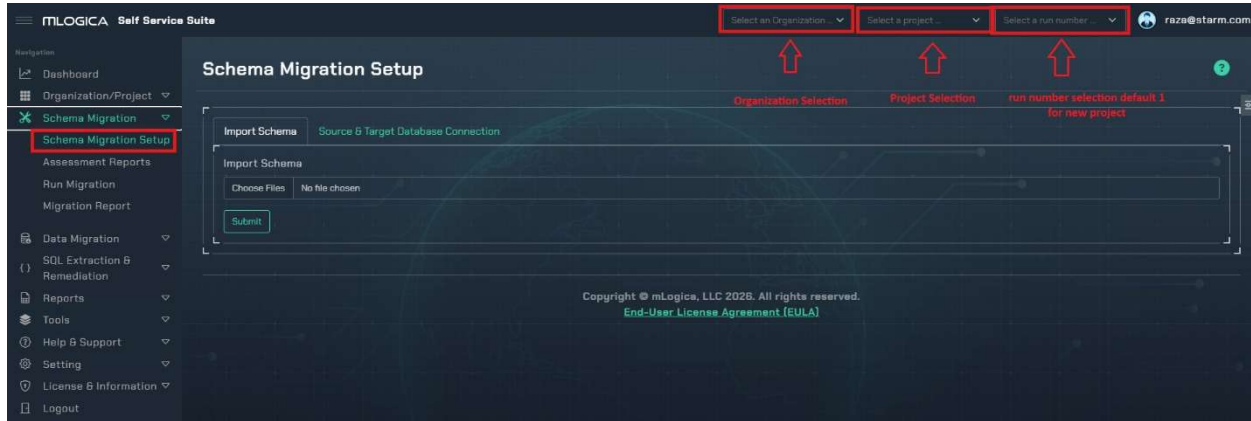


Figure 3: Mandatory selection of the organization, project and run number

- For the selected organization, project and run number, the Import Schema section of the page shows a file selector box, where the user can import schema files from their local storage.



Figure 4: Upload source SQL files from local storage

## Database Connection

Additionally, the user can use a database connection directly to ingest their queries, i.e., directly import the SQL statements from a source database.

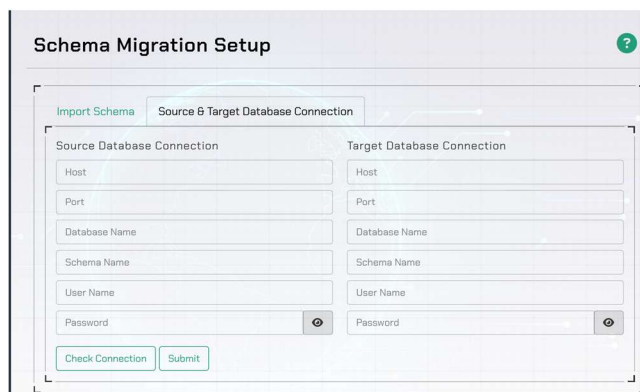


Figure 5: Alternative method to ingest source SQL directly from source database

A user can connect directly to the database using:

- Host: The IP Address of the source/target database
  - Port: The port number if the source/target database
  - Database Name: The name of the source/target database
  - Schema Name: The name of the schema in source/target database
  - User Name: The user's name for the source/target database
  - Password: The password for the source/target database
- For a newly created project, the user will add files to run number 1. To perform the translation task in subsequent iterations, the user can add files to a new run number for each iteration.



Figure 6: For an existing project, user can upload files to a new run number

## Step 2: Run Schema Migration

The **Run Schema Migration** module allows users to execute the StarM self-service suite end-to-end with a single click. The uploaded source SQL files uploaded during Schema Migration Setup will go through the following process:

1. SQL Splitter: The SQL Splitter separates source file into individual executable SQL statements. The captured metadata includes: start line number, end line number, Object category, total lines and total characters.

- Delta Automation: A Delta Automation compares migration snapshots across runs. The supported classification include:

Status	Description
Same	No changes detected
Added	New SQL statement
Modified	Existing SQL upload
Deleted	Existing SQL removed

- SQL Translation: The STAR\*M SQL translation engine translates source database SQL statements into target database SQL statements. Translation statuses include:

Status	Description
SUCCESS	Translation successful
WARNING	Translation completed with limitations
PARSE_ERROR	Unable to parse source SQL
FATAL_ERROR	Critical translation failure

- Target Validation: The target validation execute the translated SSQL against the target platform. Validation statuses include:

Validation Result	Description
SUCCESS	Validation successful
COMPILE_TIME_ERROR	Syntax or compile error
RUNTIME_ERROR	Runtime execution error
UNCATEGORIZED_ERROR	Other validation issue

## Re-Run Feature

Once a schema migration is completed, a translation version is associated with statements. Users have the option to re-run the migration when a new StarM translation version becomes available. When a migration is re-run, the previous migration is marked as obsolete and recorded in history for review, and the updated migration results are reflected in the system.

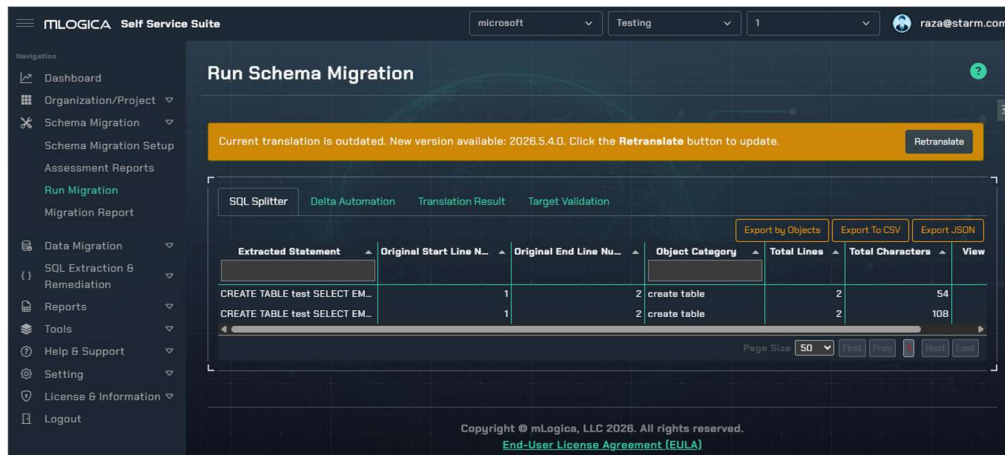


Figure 7: Run Schema Migration Page

- User must select organization, project and run number before running the translation task.

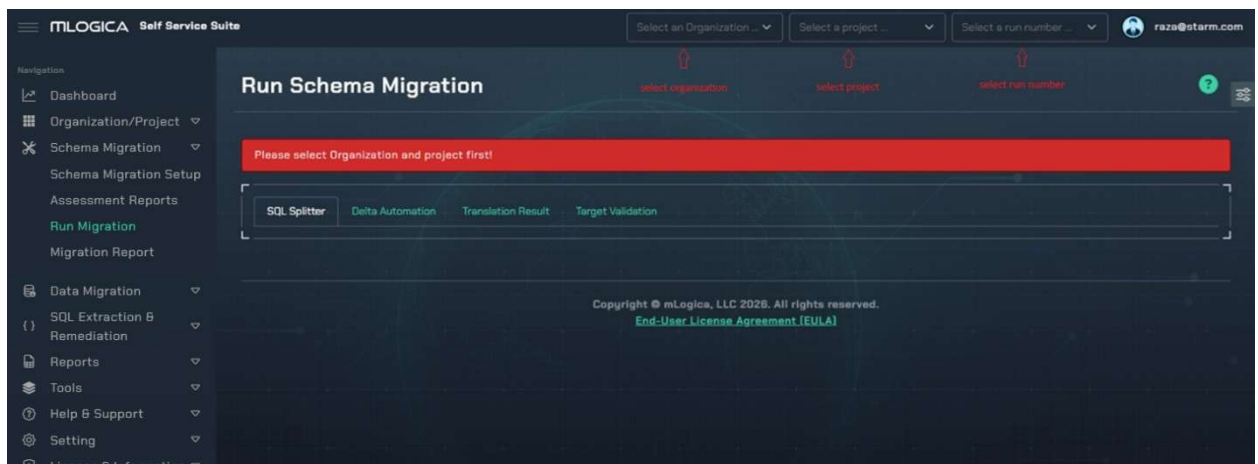


Figure 8: Select organization, project and run number to run the migration

- The user needs to upload source SQL files to a project. Checkout the documentation for Add Files to Project for more information.
- The user can run the automation task by clicking on the Run Scripts button, which performs multiple internal processes and shows the output of each process as a table.

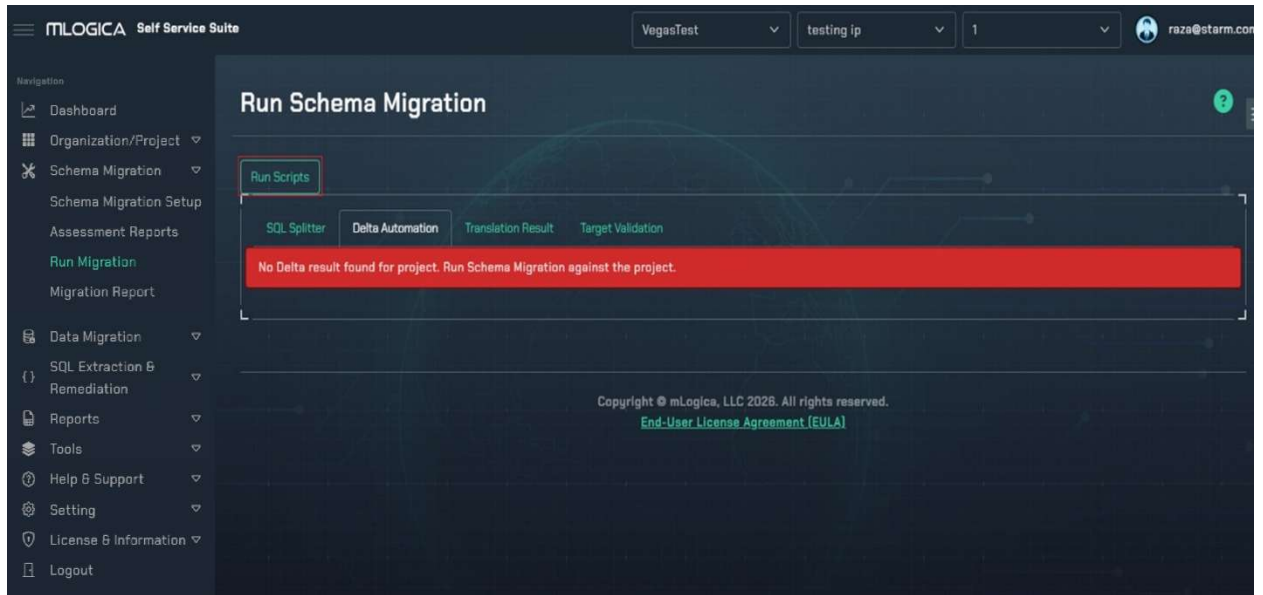


Figure 9: Run Scripts runs an end-to-end migration task for the uploaded SQL files

### Result Overview

- **SQL Splitter:** The files are split into individual SQL statements.

Extracted Statement	Original Start Line N...	Original End Line Nu...	Object Category	Total Lines	Total Characters	View Script
CREATE TABLE dbo.DevLogs ( I...	1	5,844	create table	5,844	128,025	👁

Figure 10: SQL Splitter Result Table

- **Extracted Statement** (string): This is the individual extracted statement.
  - **Original Start Line Number** (integer): The line number on which the statement starts.
  - **Original End Line Number** (integer): The line number on which the statement ends.
  - **Object Category** (string): Object category is an identifier that tells the user the type of the SQL statement. For example, in the figure 4, the first SQL statement belongs to the “create table” category. This category will help visualize the type of statements within the database.
  - **Total Lines:** The total number of lines the SQL statement spans.
  - **Total Characters:** The total number of characters in the SQL statement.
  - **View Script:** The eye icon shows the sql statement fetched from sql script attached.
- **Delta Automation:** This functionality helps the user to investigate the following details within their snapshot of the database:

- **Same:** If the SQL statement in the subsequent snapshot hasn't been modified, then that statement is logged as a same statement.
- **Added:** If a new SQL statement has been added in the subsequent snapshot of the database, then that statement is classified as an **added** statement.
- **Modified:** If a SQL statement is updated or modified in a subsequent snapshot of the database, it is classified as a **modified** statement. Our product tracks these changes, displaying both the old and new SQL statements to the user. Additionally, it highlights the modifications made to the statement for easy identification.
- **Deleted:** If a SQL statement has been deleted in the subsequent snapshot of the database, then that statement is classified as a deleted statement.

Delta Status		Count
Same		0
Added		1
Modified		0
Deleted		0

ID	Object Type	Identifier	Old Statement	Old Statement Lines	Old Statement Chars	New Statement	New
10,360	create table	test				CREATE TABLE test SELE...	

Figure 11: Delta Automation Result Table

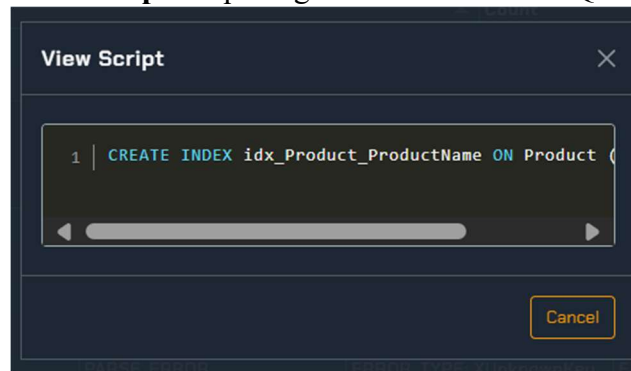
- **Object Type** (string): Same as SQL Splitter Object Category.
- **Identifier** (string): Database object name
- **Old Statement** (string): In the event that the statement is modified, this Old Statement column contains the old SQL statement.
- **Old Statement Lines** (int): The number of lines in the old statement.
- **Old Statement Chars** (int): The number of characters in the old statement.
- **New Statement** (string): In the event that the statement is modified this New Statement column contains the new SQL statement.
- **New Statement Lines** (int): The number of lines in the new statement.
- **New Statement Chars** (int): The number of characters in the new statement.
- **Delta Status** (enum): This can be one of the Delta Automation types namely same, added, modified, or deleted.
- **Partial Diff** (string): A quick glance of the difference between the old and new statement. This shows the modifications at a higher level.
- **Full Diff** (string): A detailed difference between the old and new statement. This shows the modifications at token level, informing the user of the changes at a granular level.

- **Translation Result:** The STARM SQL translation product translates the source SQL statements into target SQL statements.

Object Category					Export By Statement Type	Export To CSV	Export JSON
Object Category	Total	Pass	Fail	Pass%			
create table	5	5	0	100.00			
Grand Total	5	5	0	100.00%			
Error Category					Count		
No data found							
					0		
Warning Category		Warning Message			Count		
No Warnings Found							
Grand Total					0		
Starm Version	Object Category	Source Statement	Status	Translation Output	Error Message	Created At	Conversion Time (h:mm)
2025.10.22.0	create table	CREATE TABLE "DBA"."td...	SUCCESS	CREATE TABLE DBA.td_pr...		11/04/2025 17:31:57	00:0
2025.10.22.0	create table	CREATE TABLE "DBA"."td...	SUCCESS	CREATE TABLE DBA.td_pr...		11/04/2025 17:31:57	00:0
2025.10.22.0	create table	CREATE TABLE "DBA"."td...	SUCCESS	CREATE TABLE DBA.td_pr...		11/04/2025 17:31:19	00:0
2025.10.22.0	create table	CREATE TABLE "DBA"."td...	SUCCESS	CREATE TABLE DBA.td_pr...		11/04/2025 17:31:57	00:0
2025.10.22.0	create table	CREATE TABLE "DBA"."td...	SUCCESS	CREATE TABLE DBA.td_pr...		11/04/2025 17:31:57	00:0

- **Object Category** (string): Object category is an identifier that tells the user the type of the SQL statement. This helps identify issues that may occur within each category.
- **Error Category** (string): STARM detects error groups, allowing users to pinpoint failing cases within the translation job.
- **Warning Category** (string): STARM detects warnings groups, allowing users to pinpoint warning cases within the translation job.
- **Starm Version** (string): The translation version of the STARM that executed the translation job.
- **Source Statement** (string): The source sql statement.
- **Status** (enum): The translation state of the SQL statement.
  - SUCCESS: STARM produced no error while translating the SQL statement.
  - PARSE\_ERROR: STARM was unable to parse the SQL statement.
  - WARNING: STARM parsed the SQL statements with limitation.
  - FATAL\_ERROR: Critical issue occurred while SQL translation.
- **Translation Output**: The STARM translation for the source SQL.
- **Error Message**: Error description if any.
- **Created At**: The time when the source SQL query was translated.
- **Conversion Time**: The time when the source SQL query was translated.

- **View Script:** A quick glance at the source SQL statement.



- **Target Validation:** The translated queries are validated against the target database to ensure the efficiency of the translations.
  - **Object Category:** (string): Object category is an identifier that tells the user the type of the SQL statement. This helps identify issues that may occur within each category.
  - **Source Statement** (string): Source sql statement.
  - **SQL Translated Statement** (string): The STARM translated SQL output
  - **Validation Result** (enum): The validation category
    - SUCCESS: The validation was successful
    - RUNTIME\_ERROR: The category notifies user that the error was raised during runtime
    - COMPILE\_TIME\_ERROR: The category notifies user that the error was raised during compiling
    - UNCATEGORIZED\_ERROR: Other errors raised during validation
  - **Validation Message** (string): The validation error message
  - **View Script:** View the SQL query in formatted text

## Export Options

Users have an option to export the results in “CSV” and “JSON” formats. Users can also filter and select specific objects for export.

## Step 3: Assessment Report

The Assessment Report provides a diagnostic analysis dashboard with high-level snapshot of a system's structure, code volume, and compatibility during schema migration process.

To view the report the user must select organization, project and run number and then click on the Schema Assessment Report.

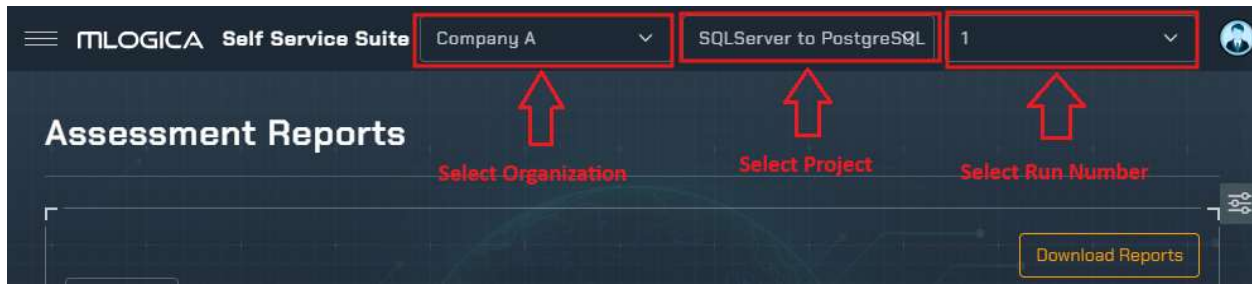
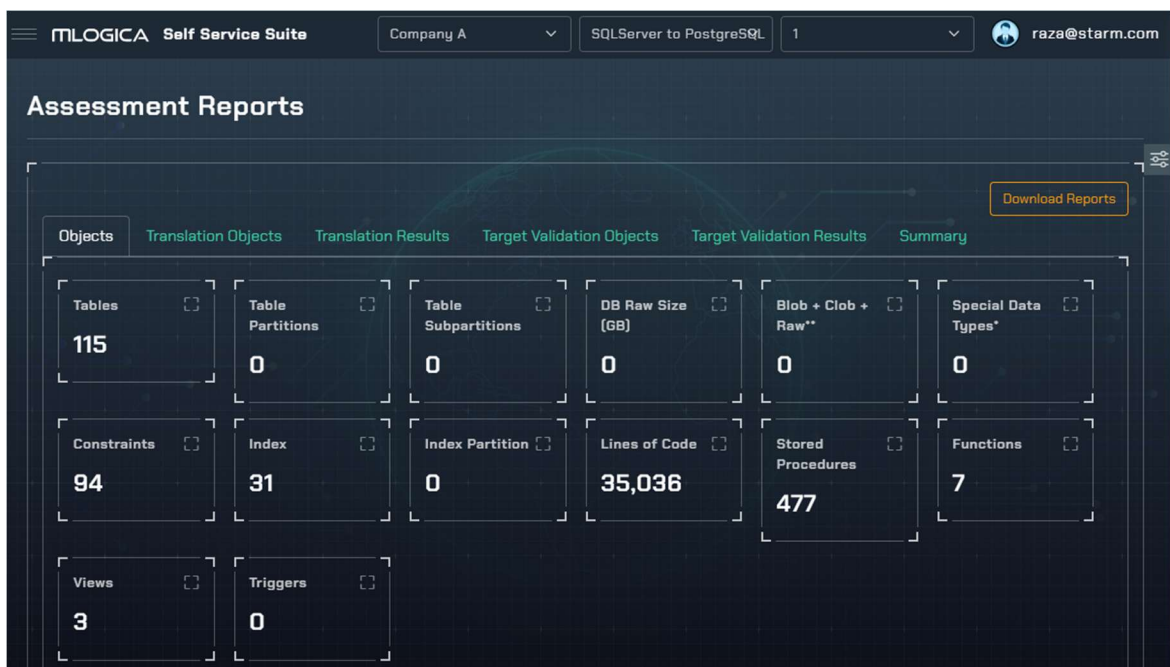


Figure 12: Assessment Report Page

The **Assessment Reports** is divided into three sections: Source Database Objects inventory, SQL Statement Migration Analysis, and Complexity Rating.

### Source Database Object Inventory

The source database object inventory consists on tiles that present the structural components of the source database. The schema inventory includes Tables, Views, Constraints, Indexes, Procedures, Functions, Triggers and Partitions.



Each tile displays a count of the specific object type detected:

- **Core Structures:** Counts for **Tables**, **Views**, **Constraints**, and **Indexes**.
- **Programmability:** Counts for **Stored Procedures**, **Functions**, and **Triggers**.
- **Storage Details:** Includes **Table Partitions**, **DB Raw Size (GB)**, and the presence of **Special Data Types** (e.g., LOBs).

**Scale:** The total **Lines of Code** across all programmable objects is displayed to indicate the volume of logic to be migrated.

## SQL Statement Analysis

The SQL statement analysis section displays the information associated with the parsing and translation of the source code to target code. This include: Parsed SQL statements, Translation success rate, Unsupported syntax and Automation efficiency.



The Parsed SQL Statements tile displays how many statements were parsed successfully. Unsuccessful parsing is typically caused by unsupported syntax or incomplete source files. The **Translated SQL Statements** tile displays the success rate for converting source SQL into the target database SQL format.

- **Passed:** Automated translation was successful.
- **Failed:** These require manual intervention or review.
- **% Passed:** Your primary metric for automation efficiency.

## Complexity Score

Complexity Score is the assessment (Low, Medium, High, Complex) of the migration effort required based on the density of business logic density, inter-dependencies, proprietary syntax usage and procedural complexity used in the source database.

---

## Using the Expand Feature

When you click the Expand Icon on a specific object tile, the dashboard transitions from a summary view to a **Detailed List View**.



- **Table Details:** Clicking expand on the "Tables" tile will display a comprehensive list of all table names.
- **Procedure/Function Details:** Expanding these tiles provides a list of specific object names.

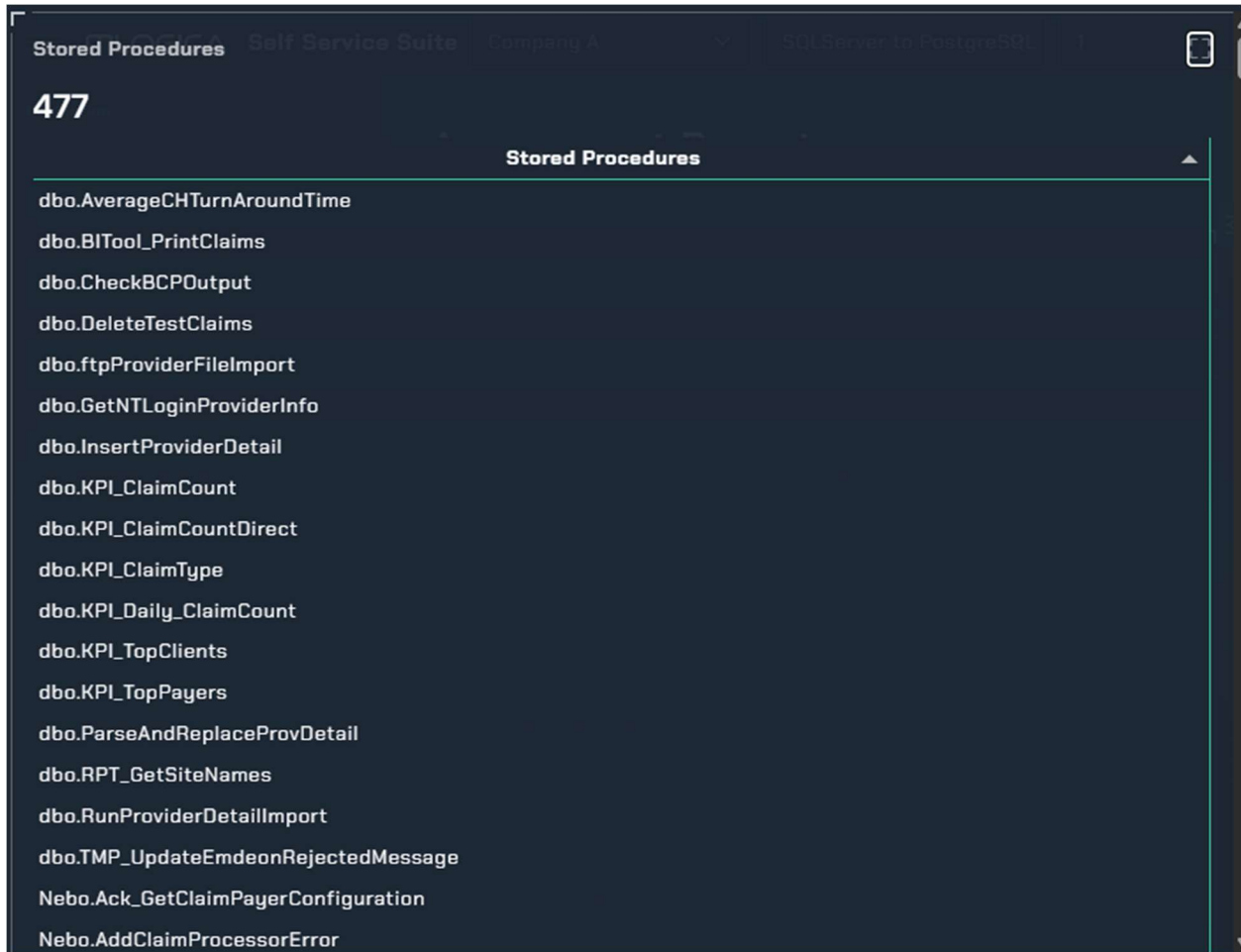


Figure 3: Expand View

### Translation Objects Tab

The **Translation Objects** tab provides a detailed breakdown of the schema conversion process. It allows you to assess the compatibility of your source database objects with the target environment by categorizing them by SQL statement type.

**Assessment Reports**

Objects **Translation Objects** Translation Results Target Validation Objects Target Validation Results Summary Download Reports

Object Category	Total	Pass	Fail	Pass%
alter table	79	78	1	98.73%
create database	1	1	0	100%
create function	7	7	0	100%
create nonclustered index	22	22	0	100%
create partition function	1	1	0	100%
create partition scheme	1	1	0	100%
create proc	4	4	0	100%
create procedure	473	472	1	99.79%
create role	1	1	0	100%
create schema	1	1	0	100%
create synonym	4	4	0	100%
create table	115	115	0	100%
create type	57	57	0	100%
create unique nonclustered index	9	9	0	100%
create user	16	16	0	100%
create view	3	3	0	100%
<b>Grand Total</b>	<b>784</b>	<b>782</b>	<b>2</b>	<b>99.75%</b>

The translation Objects tab features a summary table that displays the success rate of following object categories:

- **Object Category:** Lists the specific types of SQL commands or objects being migrated (e.g., alter table, create procedure, create view).
- **Total:** The total number of objects identified in the source for that specific category.
- **Pass:** The number of objects successfully translated to the target syntax without errors.
- **Fail:** The number of objects that encountered issues during translation and require manual intervention or review.
- **Pass%:** A visual health indicator and percentage representing the success rate for that category.
- **Green:** Indicates a high success rate (typically 100%).
- **Yellow/Red:** Indicates categories where errors occurred, requiring attention.

### Translation Results Tab

The **Translation Results** tab provides a detailed breakdown of SQL statements that failed during the automated conversion process. This view is specifically designed for troubleshooting and debugging syntax errors or unsupported features in the source database/schema. The Translation Results tab provides information related to failed translation analysis, error categorization, unsupported syntax identification and translation output review.

Starm Version	Object Category	Source Statement	Status	Translation Output	Error Message	Created At	Conversion Time (h:mm:ss)	View Script
2026.4.13.0	alter table	ALTER TABLE [Nebol].[Clai...	PARSE_ERROR	ERROR_TYPE: XBadSyntax	ERROR: UNSUPPORTED ...	04/14/2026 17:27:54	00:00:00	
2026.4.13.0	create procedure	CREATE PROCEDURE [Neb...	PARSE_ERROR	ERROR_TYPE: XBadSyntax	ERROR: Expected ] recei...	04/14/2026 17:27:56	00:00:00	

Figure 5: Translation Results Tab

It displays following information:

- **Starm Version:** The specific version of the migration engine used for this assessment.
- **Object Category:** The type of SQL command being processed (e.g., alter table, create procedure).
- **Source Statement:** A snippet of the original code from the source database that triggered the error.
- **Status:** The high-level error state (typically PARSE\_ERROR in this view).
- **Translation Output:** Categorizes the failure type, such as XBadSyntax for syntax-related issues.
- **Error Message:** A detailed technical description of why the translation failed (e.g., ERROR: UNSUPPORTED or ERROR: Expected ]).
- **View Script:** Click the Eye Icon to open a full-screen view of the source script for deeper analysis.

### Target Validation Objects Tab

The Target Validation Object tab provides a detailed breakdown of how your converted SQL code performs when executed against the **target database**. It is the primary dashboard for identifying syntax compatibility issues and structural gaps post-translation.

Assessment Reports

Objects Translation Objects Translation Results **Target Validation Objects** Target Validation Results Summary Download Reports

Object Category	Total	Pass	Fail	Pass%
alter table	78	78	0	100%
create database	1	0	1	0%
create function	7	4	3	57.14%
create nonclustered index	22	22	0	100%
create partition function	1	0	1	0%
create partition scheme	1	0	1	0%
create proc	4	4	0	100%
create procedure	472	339	133	71.82%
create role	1	1	0	100%
create schema	1	1	0	100%
create synonym	4	0	4	0%
create table	115	114	1	98.13%
create type	57	57	0	100%
create unique nonclustered index	9	9	0	100%
create user	16	0	16	0%
create view	3	3	0	100%
Grand Total	792	632	160	79.80%

Figure 6: Target Validation Objects Tab

Target validation reports identify Compile-time issues, Runtime issues, Unsupported features and Database compatibility problems. This tab displays a summary information on the success validation rate of various object categories:

- **Object Category:** Lists the specific types of SQL commands or objects being validated (e.g., alter table, create procedure, create view).
- **Total:** The total number of objects identified in the target for that specific category.
- **Pass:** The number of objects successfully validated to the target syntax without errors.
- **Fail:** The number of objects that encountered issues during validation and require manual intervention or review.
- **Pass%:** A visual health indicator and percentage representing the success rate for that category.
- **Green:** Indicates a high success rate (typically 100%).
- **Yellow/Red:** Indicates categories where errors occurred, requiring attention.

### Target Validation Results Tab

The **Target Validation Results** tab provides a detailed assessment of SQL statements that failed to validate or compile on the target database. This information is used to identify, troubleshoot, and resolve syntax and logic errors introduced during translation.

Assessment Reports

[Download Reports](#)

Objects Translation Objects Translation Results Target Validation Objects **Target Validation Results** Summary

Object Category	Source Statement	Starm Translated State...	Validation Result	Validation Message	View Script
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"taxid" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"taxid" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"taxid" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"taxid" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"vriic" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"vriic" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"vriic" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"vriic" is not a known variable	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	"@claimsToBeBatched" is not...	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	syntax error at end of input	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	syntax error at or near "XML"	
create procedure	CREATE PROCEDURE [Neb...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	syntax error at or near "XML"	
create procedure	Create procedure [Nebo]...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	syntax error at or near "v_Ne...	
create procedure	Create procedure [Nebo]...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	syntax error at or near "v_Ne...	
create procedure	Create procedure [Nebo]...	CREATE PROCEDURE Nebo.B...	COMPILE_TIME_ERROR	syntax error at or near "v_Ne...	

- **Object Category:** The type of SQL object being processed (e.g., create procedure).
- **Source Statement:** The original SQL code from the source system.
- **Translated Statement:** The converted code intended for the target database.
- **Validation Result:** The specific error status (e.g., COMPILE\_TIME\_ERROR).
- **Validation Message:** A descriptive error from the target database engine explaining why the statement failed (e.g., "taxid" is not a known variable or syntax error at or near "XML").
- **View Script:** Click the **eye icon** to open a full-screen view of the translated script for deeper analysis.

## Summary Tab

The **Summary** tab acts as the final report for the automated assessment process. It categorizes findings related to the database inventory, parsing results, translation success rates, and actionable recommendations. The Summary reports include Database inventory, Parsing metrics, Translation metrics, Validation metrics and Migration recommendations

Objects Translation Objects Translation Results Target Validation Objects Target Validation Results **Summary**

**Database Inventory:**

The schema contains **115** tables, **477** stored procedures, **3** Views, **7** functions, and **0** trigger.  
There is a total of **794** SQL statements.

**Key Finding:**

STARM successfully parsed **792** SQL statements (99.75%).  
STARM cannot recognize following statements:

- UNSUPPORTED ALTER TABLE TYPE: SET (1)
- Expected ] received . (1)

Out of **792** SQL statements parsed, it successfully translated **632** SQL statements for the target database (79.8%).  
The following **160** parsed statements cannot be translated to target SQL statements:

- syntax error at or near (83)
- Improper qualified name (too many dotted names): cnsqprclaimmaster.nebo.local.payermaster.dbo.payers (2)
- Improper qualified name (too many dotted names): cnsqprclaimmaster.nebo.local.claimssecurity.dbo.sites (2)
- is not a known variable (51)
- syntax error at end of input (8)
- RETURN cannot have a parameter in a procedure (7)
- Improper qualified name (too many dotted names): cnsqprclaimmaster.nebo.local.payermaster.dbo.route (6)
- too many parameters specified for RAISE (1)

**Recommendations:**

Please fix the above statement(s) before moving forward with migration or contact mLogica for remediation.

To receive a quote for mLogica's migration services, please follow these steps:

1. Download this [Zip file](#)
2. Upload it to the <https://starm-customerportal-uat.mlogica.com/>
3. We will get back to you shortly with further details.

Figure 8: Summary Tab Overview

The report consists of five sections:

### 1. Database Inventory

This section provides a statistical snapshot of the source schema. Use this to verify that the scanner has correctly identified all objects in your scope.

- **Object Counts:** Displays the total number of Tables, Stored Procedures, Views, Functions, and Triggers.
- **Total SQL Statements:** Shows the total volume of SQL code identified for analysis.

## 2. Key Findings

The assessment tool evaluates the "parseability" and "translatability" of the code.

### Parsing Results

- **Success Rate:** Displays the percentage of SQL statements successfully understood by the engine.
- **Unsupported Statements:** Lists specific syntax that the tool cannot recognize (e.g., UNSUPPORTED ALTER TABLE TYPE). These items must be manually reviewed.

## 3. Translation Results

This is the most critical section for migration planning. It lists why certain statements cannot be automatically converted to the target database syntax:

- **Syntax Errors:** Highlights standard code errors or unexpected formatting.
- **Improper Qualified Names:** Identifies objects with too many "dots" in the path (e.g., server.database.schema.table), which often cause issues in target cloud environments.
- **Unknown Variables:** Flags references to variables not defined within the scope of the assessment.
- **Logic Errors:** Points out structural issues, such as RETURN parameters in procedures or incorrect parameter counts for RAISE.

## 4. Recommendations Section

The report concludes with a directive based on the findings.

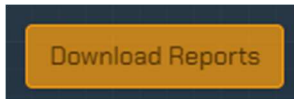
- **Self-Remediation:** Fix the specific SQL errors listed in the "Key Findings" section and re-run the assessment.
- **Professional Remediation:** Contact the support team for complex translation issues that cannot be automated.

## 5. Receiving a Quote for Services

If you require professional migration assistance, follow the three-step workflow at the bottom of the screen:

1. **Download Zip File:** Click the link to download the full diagnostic report packet.
2. **Upload:** Submit this file to the customer portal URL provided.
3. **Review:** A representative will contact you with a tailored migration quote and remediation plan.

The assessment report can be download int Excel/Word zip by clicking on click the *Download Reports* button in the top right corner of the assessment report.



Clicking the *Download Reports* button popup an "*Export Statements by Statement Type*" window that displays all SQL statement by categories. The user has the option to:

- **Select All:** Use the top checkbox to include all object types (Tables, Procedures, Views, etc.).
- **Granular Selection:** Manually check individual boxes (e.g., only create procedure) if you need a targeted report.
- **View List:** You can scroll through the list to verify the specific categories detected in your source code.

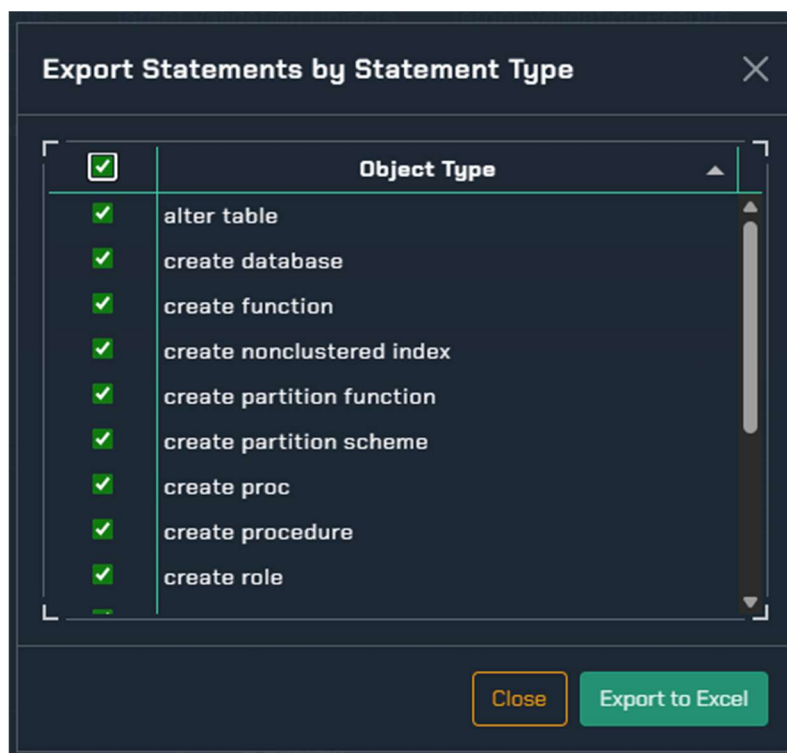


Figure 9: Export Statements

## Generating the Bundle

Once your selection is complete, click the **Export to Excel** button. The system will process your data and generate a **ZIP Bundle** containing two primary files:

## File A: Assessment Data (Excel Doc)

A multi-sheet workbook containing the technical granular data:

- **Translation Objects:** High-level parsing statistics for each category.
- **Translation Results:** Details on any statements that failed the initial translation phase.
- **Target Validation Objects:** Success/Failure rates for the target environment.
- **Target Validation Results:** Specific compiler error messages from the target database.

## File B: Executive Summary (Word Doc)

A formatted document intended for stakeholders, including:

- **Database Inventory:** Total counts of tables, procedures, and views.
- **Key Findings:** Simplified breakdown of the conversion percentage and common syntax errors.

## Step 4: Migration Summary Report

The **Migration Summary Report** page displays the migration summary along with **translation download option**. It displays a consolidated migration health dashboard. User has an option to execute the translated statements on the target database provided through Upload files tab. User can view Error Summary in the last tab of the page.

Below are the features:

Migration Manager Tab

The Migration Manager tab displays following two tabs:

**Download Files:** The *Download StarM Output file* button on the Download file tab allows user to down the translated script for the target database. If the translation is not 100% complete, then the application will display a warning message.

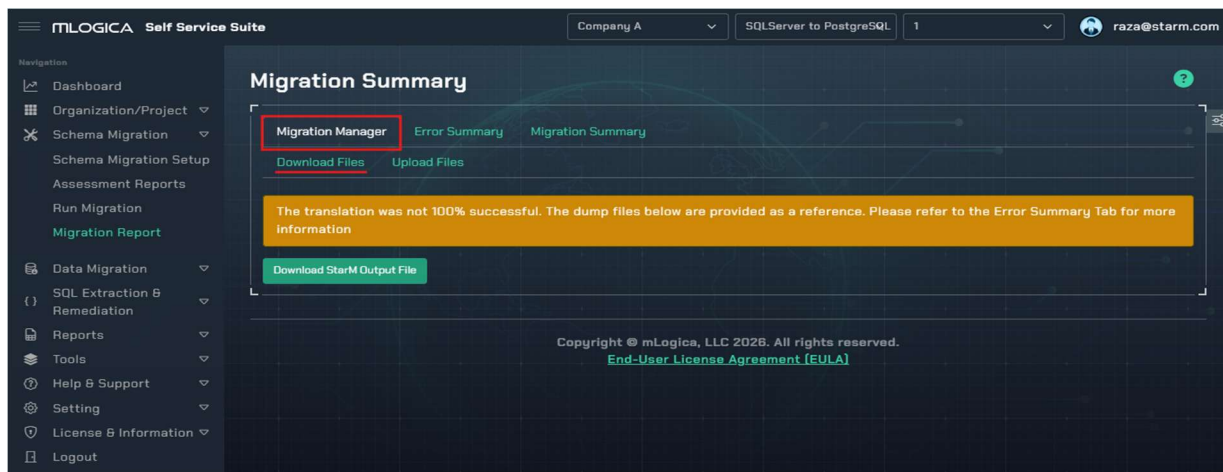


Figure 1: Download Translated Script

- **Upload Files:** The Upload Files allows the user to upload/execute the translated script to run on the target database.

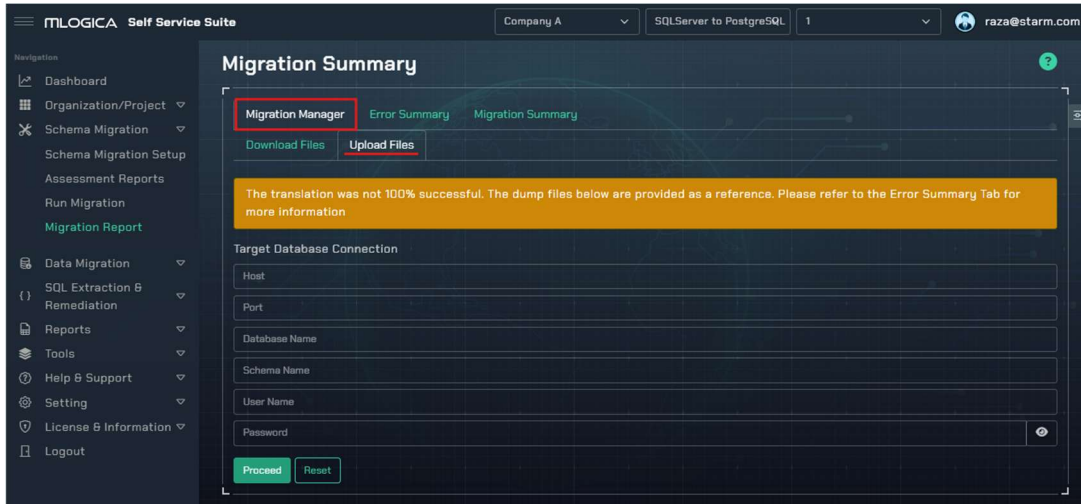


Figure 2: Upload Translated Script

## Error Summary Tab

The **Error Summary** tab provides a categorized breakdown and a detailed drill down view of all validation and parsing errors that occurred during migration.

Error Category	Error Title	Count
TARGET_VALIDATION_ERROR	syntax error at or near	83
TARGET_VALIDATION_ERROR	improper qualified name (too many dotted names): cnsqprclaimmaster.nebo.local.pay...	2
TARGET_VALIDATION_ERROR	Improper qualified name (too many dotted names): cnsqprclaimmaster.nebo.local.clai...	2
TARGET_VALIDATION_ERROR	is not a known variable	51
TARGET_VALIDATION_ERROR	syntax error at end of input	8
TARGET_VALIDATION_ERROR	RETURN cannot have a parameter in a procedure	7
TARGET_VALIDATION_ERROR	improper qualified name (too many dotted names): cnsqprclaimmaster.nebo.local.pay...	6
STARM_PARSE_ERROR	XBadSyntax	2
TARGET_VALIDATION_ERROR	too many parameters specified for RAISE	1
Grand Total		162

The interface is divided into two primary views: a **High-Level Summary** and a **Detailed Error Log**.

### High-Level Summary View:

The high-level summary view provides a bird's-eye view of migration health, allowing user to prioritize bug fixes based on frequency. It consists of the following information:

- **Error Category:** Broad classification of the issue (e.g., TARGET\_VALIDATION\_ERROR for database-specific rules or STARM\_PARSE\_ERROR for syntax reading issues).

- **Error Title:** A concise name for the error (e.g., "syntax error at or near" or "is not a known variable").
- **Count:** The total number of occurrences for that specific error title. Use this to identify widespread systemic issues versus one-off errors.
- **Grand Total:** The sum of all errors across all categories.

### Detailed Error Log View

The detailed error information can be retrieved by drilling down on specific instances of errors to facilitate debugging.

Error Category	Error Title	Error Message	Submitted On	View Script
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "["	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	improper qualified name (too many d...	improper qualified name (too many d...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "]"	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	improper qualified name (too many d...	improper qualified name (too many d...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "NONCLUSTE...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "TRANSACTION...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "v_Nebo.ListO...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "v_Nebo.ListO...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "v_Nebo.ListO...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "v_Nebo.ListO...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	syntax error at or near	syntax error at or near "v_Nebo.ListO...	04/14/2026 17:28:00	
TARGET_VALIDATION_ERROR	is not a known variable	"vriic" is not a known variable	04/14/2026 17:28:00	

It consists of the following information:

- **Error Category:** Broad classification of the issue (e.g., TARGET\_VALIDATION\_ERROR for database-specific rules or STARM\_PARSE\_ERROR for syntax reading issues).
- **Error Title:** A concise name for the error (e.g., "syntax error at or near" or "is not a known variable").
  - **Error Message:** The full system output, often including the specific character or variable name that triggered the failure (e.g., syntax error at or near "[").
  - **Submitted On:** The timestamp when the specific migration attempt or validation check was performed.
  - **View Script (Eye Icon):** Clicking this icon opens the specific code block or SQL script that failed, allowing you to see exactly where the error is located in the source code.

### Migration Summary

The **Migration Summary** tab serves as the primary health dashboard, showing how much of your source code was successfully read and how much was successfully converted to the target dialect.

## Migration Summary

Migration Manager   Error Summary   **Migration Summary**

Total Files ▲	Parsed SQL				Translated SQL				Errors ▲
	Total ▲	Passed ▲	Failed ▲	Passed% ▲	Total ▲	Passed ▲	Failed ▲	Passed% ▲	
1	794	792	2	99.75%	792	632	160	79.8%	162

- Total Files: No. of files attached to the project.
- Parsed SQL: It displays the Total, Passed, failed statements count along with Passed percentage (%).
- Translated SQL: It displays the Total, Passed, failed statement counts along with Passed percentage (%).
- Errors: It shows Error count.

## 9. Data Migration

### Data Migration - Database Setup

The **Database Setup** provides user the ability to manage Source and Target database connections. This page serves as a comprehensive Database setup service. Selection of both **Organization** and **Project** is mandatory.

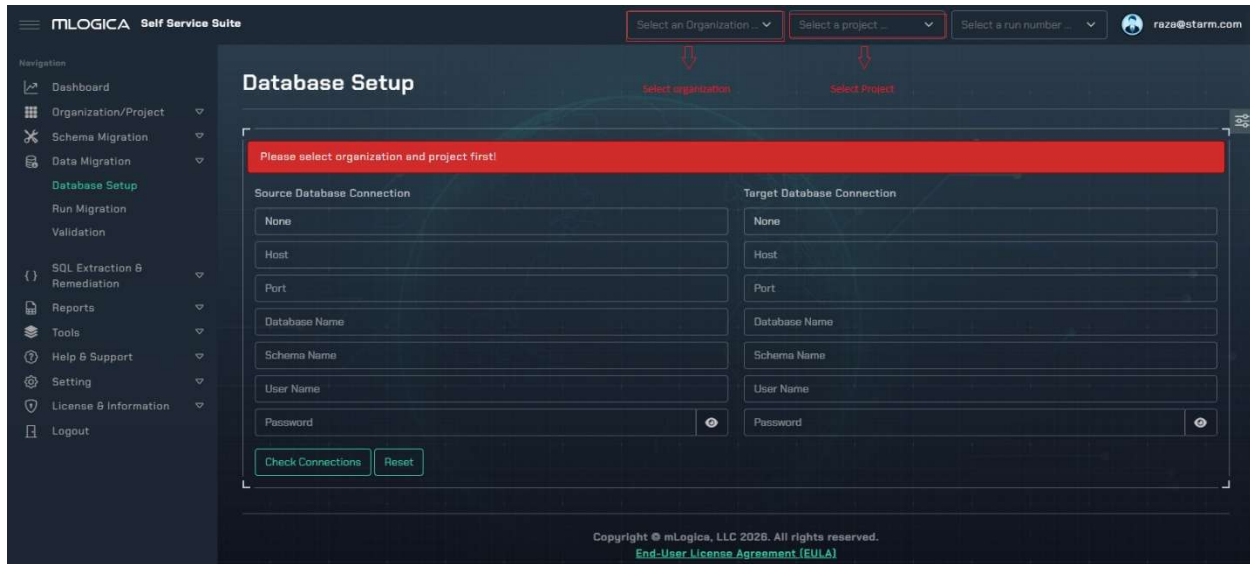


Figure 13: Database Setup Page

The organization and project hierarchy is established with the organization as the top-level parent, where each organization can have multiple projects. It provides following functionalities:

- **Select Organization Name:** This dropdown is used to select an existing organization. Selection of this field is mandatory.
- **Select Project:** This dropdown allows the user to select an existing project under the selected organization.
- **Check Connections:** The system will check the source and target connections, if the connection is successful, give a message to client.
- **Reset:** Allows to reset the form fields.

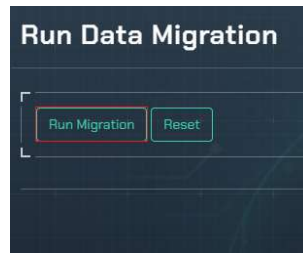
Note:

1. The source and target connection details are saved to upload translated script to client database and data migration setup.
2. User can edit the connection details later under selected organization and project.

### Run Data Migration

The Run **Data Migration** page allows users to migrate data from the source database to the target database. The database setup configuration is used to establish connections to both the source and target databases. When the user clicks the Run Migration button, the data is migrated from the source database to the target database through the following process:

1. The Run Migration button on the Run Data Migration window allows user to start the data migration process.



*Run Data Migration startup Page*

2. Connection Validation: Before starting the migration, the source and target connection parameters will be validated. If the connection is successful, then the process will proceed with the data migration.

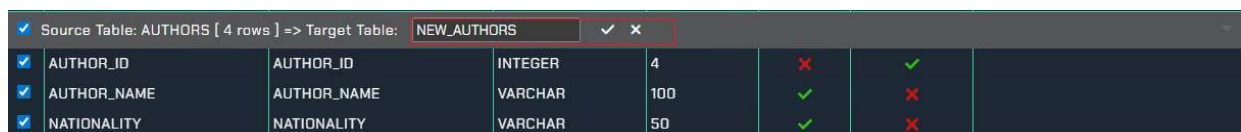


3. Schema Selection: A list of tables with columns will be displayed to allow user to select or deselect tables along with their columns to be part of the data migration process.
4. Each table column detail shows additional information such as data type, size, nullable status, primary key, and any configured default value.

<input type="checkbox"/>	Column Name [Source]	Column Name [Target]	Data Type	Size	Nullable	Primary Key	Default Value
<input checked="" type="checkbox"/>	Source Table: AUTHORS [ 4 rows ] => Target Table: AUTHORS						
<input checked="" type="checkbox"/>	AUTHOR_ID	AUTHOR_ID	INTEGER	4	✗	✓	
<input checked="" type="checkbox"/>	AUTHOR_NAME	AUTHOR_NAME	VARCHAR	100	✓	✗	
<input checked="" type="checkbox"/>	NATIONALITY	NATIONALITY	VARCHAR	50	✓	✗	
<input type="checkbox"/>	Source Table: A_PWORD_MGR [ 0 row ] => Target Table: A_PWORD_MGR						
<input type="checkbox"/>	CONTEXT_ID	CONTEXT_ID	VARCHAR	255	✗	✓	
<input type="checkbox"/>	CONTEXT_NAME	CONTEXT_NAME	VARCHAR	255	✗	✓	
<input type="checkbox"/>	HASH	HASH	VARCHAR	255	✗	✗	
<input checked="" type="checkbox"/>	Source Table: A_PWORD_MGR_TEST [ 0 row ] => Target Table: A_PWORD_MGR_TEST						
<input checked="" type="checkbox"/>	CONTEXT_ID	CONTEXT_ID	VARCHAR	255	✗	✗	
<input checked="" type="checkbox"/>	CONTEXT_NAME	CONTEXT_NAME	VARCHAR	255	✗	✗	
<input type="checkbox"/>	HASH	HASH	VARCHAR	255	✗	✗	
<input checked="" type="checkbox"/>	Source Table: BOOKS [ 7 rows ] => Target Table: BOOKS						
<input checked="" type="checkbox"/>	BOOK_ID	BOOK_ID	INTEGER	4	✗	✓	
<input checked="" type="checkbox"/>	BOOK_TITLE	BOOK_TITLE	VARCHAR	200	✓	✗	
<input checked="" type="checkbox"/>	AUTHOR_ID	AUTHOR_ID	INTEGER	4	✓	✗	

*Figure 3: Table with column selection*

5. User can also rename the table name at target side if required.



6. When the user finalizes the selection, they can click the button labeled “Proceed with Current Selection” to proceed to the next step.

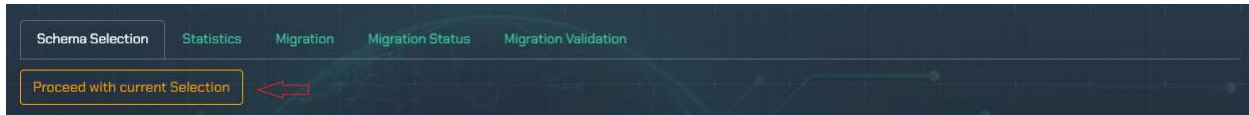


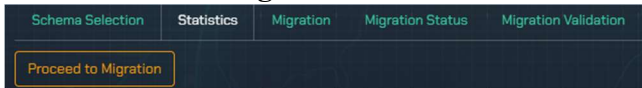
Figure 5: Proceed to next migration step

- Table statistics shown to the user include Table Name along with the number of rows and migration status. When a source table does not have a primary key defined, that table is not included in the migration process.
- User can filter tables based on number of rows in each table.

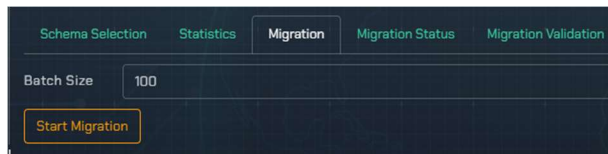
Show tables with 0 rows

Table Name	No. of Rows	Status
A_PWORD_MGR	0	Pending
A_PWORD_MGR_TEST	0	Cannot migrate the table because the primary key is missing
AUTHORS	4	Pending
BOOKS	7	Pending
CATEGORIES	8	Pending
CHANGETIME	0	Cannot migrate the table because the primary key is missing
CUSTOMERS	40	Pending
DEPARTMENT	5	Pending
EMPLOYEES	16	Pending
INVENTORY	20	Pending

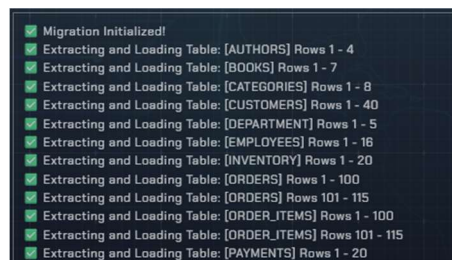
- When the user has reviewed the statistics, they can proceed to the next step by clicking the **“Proceed to Migration”** button.



- Finally, the user can enter the batch size. The default value is a maximum of 100 records in a single batch.
- When the user accepts the batch size, they can proceed with the migration by clicking the button called **“Start Migration”**.



- During Migration process the progress is shown in the list below:



- In the Migration Status tab, the overall status of the migration process includes Source Table, Target Table, Batch No., From Record, To Record, Status, Error Message, Action, and Error.


14. The user has the option to re-run failed batches either individually or all failed batches at once.

Source Table	Target Table	Batch No	From Rec. No.	To Rec No.	Status	Error Message	Action	Error
AUTHORS	AUTHORS	1	1	4	Success		✓	🟢
BOOKS	BOOKS	1	1	7	Success		✓	🟢
CATEGORIES	CATEGORIES	1	1	8	Success		✓	🟢
CUSTOMERS	CUSTOMERS	1	1	40	Success		✓	🟢
DEPARTMENT	DEPARTMENT	1	1	5	Success		✓	🟢
EMPLOYEES	EMPLOYEES	1	1	16	Success		✓	🟢
INVENTORY	INVENTORY	1	1	20	Success		✓	🟢
ORDER_ITEMS	ORDER_ITEMS	1	1	100	Success		✓	🟢
ORDER_ITEMS	ORDER_ITEMS	2	101	115	Success		✓	🟢
ORDERS	ORDERS	1	1	100	Success		✓	🟢
ORDERS	ORDERS	2	101	115	Success		✓	🟢
PAYMENTS	PAYMENTS	1	1	20	Success		✓	🟢
PRODUCTS	PRODUCTS	1	1	48	Success		✓	🟢
SHIPMENTS	SHIPMENTS	1	1	20	Success		✓	🟢
SUPPLIERS	SUPPLIERS	1	1	12	Success		✓	🟢

15. Migration Validation tab shows data validation for each table column.

Table	Column	Batch	Migration Status	Validation Status	View Failed
<b>Table Name: AUTHORS</b>					
Column Name: AUTHOR_ID		1	Success	Success	
Column Name: AUTHOR_NAME		1	Success	Success	
Column Name: NATIONALITY		1	Success	Success	
<b>Table Name: BOOKS</b>					
Column Name: BOOK_ID		1	Success	Success	
Column Name: BOOK_TITLE		1	Success	Success	
Column Name: AUTHOR_ID		1	Success	Success	
Column Name: PUBLICATION_YEAR		1	Success	Success	

16. If data validation fails for a specific column, an eye icon is displayed for that row to show further details.

Table Name: CUSTOMERS				
Column Name: CUSTOMER_ID	1	Success	Success	
Column Name: FIRST_NAME	1	Success	Success	
Column Name: LAST_NAME	1	Success	Success	
Column Name: EMAIL	1	Success	Success	
Column Name: PHONE_NUMBER	1	Success	Success	
Column Name: ADDRESS	1	Success	Success	
Column Name: CITY	1	Success	Success	
Column Name: POSTAL_CODE	1	Success	Success	
Column Name: COUNTRY	1	Success	Success	
Column Name: DATE_CREATED	1	Success	Failed	

17. Below are the details of non-validated columns.

Table Lookup [ CUSTOMERS ]	
SOURCE: DATE CREATED	TARGET: DATE CREATED
2025-03-09T23:09:24.113	2025-03-09T18:09:24.113
2025-03-09T23:12:16.597	2025-03-09T18:12:16.597
2025-03-09T23:12:18.628	2025-03-09T18:12:18.628
2025-03-09T23:12:20.581	2025-03-09T18:12:20.581
2025-03-09T23:12:24.659	2025-03-09T18:12:24.659
2025-03-09T23:12:26.815	2025-03-09T18:12:26.815
2025-03-09T23:12:28.956	2025-03-09T18:12:28.956
2025-03-09T23:12:31.097	2025-03-09T18:12:31.097
2025-03-09T23:12:33.315	2025-03-09T18:12:33.315
2025-03-09T23:12:35.3	2025-03-09T18:12:35.3
2025-03-09T23:12:37.597	2025-03-09T18:12:37.597
2025-03-09T23:12:39.768	2025-03-09T18:12:39.768
2025-03-09T23:12:42.362	2025-03-09T18:12:42.362
2025-03-09T23:12:59.893	2025-03-09T18:12:59.893
2025-03-09T23:13:05.585	2025-03-09T18:13:05.585
2025-03-09T23:13:08.237	2025-03-09T18:13:08.237
2025-03-09T23:13:12.44	2025-03-09T18:13:12.44
2025-03-09T23:13:17.94	2025-03-09T18:13:17.94
2025-03-09T23:13:21.843	2025-03-09T18:13:21.843
2025-03-09T23:13:25.628	2025-03-09T18:13:25.628
2025-03-09T23:13:31.112	2025-03-09T18:13:31.112
2025-03-09T23:13:34.018	2025-03-09T18:13:34.018
2025-03-09T23:13:36.487	2025-03-09T18:13:36.487

## Re-Run Migration Feature

Once the data migration is completed, a migrated status is associated with the selected project and run number. The user can revisit the page to re-run the migration on failed batches when required.

Source Table	Target Table	Batch No.	From Rec. No.	To Rec No.	Status	Error Message	Action	Error
AUTHORS	AUTHORS	1	1	4	Success		↕	🟢
BOOKS	BOOKS	1	1	7	Success		↕	🟢
CATEGORIES	CATEGORIES	1	1	8	Success		↕	🟢
CUSTOMERS	CUSTOMERS	1	1	40	Success		↕	🟢
DEPARTMENT	DEPARTMENT	1	1	5	Success		↕	🟢
EMPLOYEES	EMPLOYEES	1	1	18	Success		↕	🟢
INVENTORY	INVENTORY	1	1	20	Success		↕	🟢
ORDERITEMS	ORDERITEMS	1	1	100	Success		↕	🟢
ORDER_ITEMS	ORDER_ITEMS	2	101	115	Success		↕	🟢
ORDERS	ORDERS	1	1	100	Success		↕	🟢
ORDERS	ORDERS	2	101	115	Success		↕	🟢
PAYMENTS	PAYMENTS	1	1	20	Success		↕	🟢
PRODUCTS	PRODUCTS	1	1	48	Success		↕	🟢
SHIPMENTS	SHIPMENTS	1	1	20	Success		↕	🟢
SUPPLIERS	SUPPLIERS	1	1	12	Success		↕	🟢

Figure 12: Re-run all failed batched feature

The organization, project and run number hierarchy is established with the organization as the top-level parent, where each organization has multiple projects. Each project undergoes a run number increment over time.

## Data Validation

The **Data Validation** page allows users to validate data in the target database. The database setup configuration is used to establish connections to the target databases. When the user clicks the **Run Validation** button, the data is validated through the following process:

1. The system fetches tables from the source and target databases, including the table name, number of columns, and number of rows. The validation result remains **Not Validated** until the validation process is started.



Figure 14: Data Validation startup Page

2. **Connection Validation:** Initially, before starting the validation, the source and target connection parameters are validated. If the connection is successful, the process proceeds with data validation.



Figure 2: Database connections validation

3. **Table Selection:** A list of tables is displayed to the user for selection.
4. The user can select or deselect tables to include in the data validation process.

**Data Validation**

Source Database: Validated  
 Target Database: Validated

Run Validation    Reset

100%

Start Validation

Table	Source Database		Target Database		Validation Result			Sample D...	Summary	Action
	No. of Cols	No. of Rows	No. of Cols	No. of Rows	No. of Cols	No. of Rows				
<input type="checkbox"/> A_PWORD_MGR	3	0	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input type="checkbox"/> A_PWORD_MG...	3	0	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> AUTHORS	3	4	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> BOOKS	4	7	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> CATEGORIES	3	8	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input type="checkbox"/> CHANGETIME	1	0	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> CUSTOMERS	10	40	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> DEPARTMENT	4	5	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> EMPLOYEES	7	16	0	0	Not Matched	Not Matched	Not Matched	Not Validated		

5. Tables that do not contain data are not allowed to be validated.
6. Once the user finalizes the selection, they can click the **“Start Validation”** button to validate the data on both sides.
7. The system will compare the number of rows, number of columns, and sample data from both the source and target databases.
8. If all validations succeed, the validation summary column will show **Validated**.

### Re-Validate Feature

Once the data validation is completed, a validation status is associated with the selected project and run number. The user can revisit the page to re-validate the data when required.

**Data Validation**

Source Database: Validated  
 Target Database: Validated

Run Validation    Reset

100%

Start Validation

Table	Source Database		Target Database		Validation Result			Sample D...	Summary	Action
	No. of Cols	No. of Rows	No. of Cols	No. of Rows	No. of Cols	No. of Rows				
<input type="checkbox"/> A_PWORD_MGR	3	0	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input type="checkbox"/> A_PWORD_MG...	3	0	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> AUTHORS	3	4	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> BOOKS	4	7	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> CATEGORIES	3	8	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input type="checkbox"/> CHANGETIME	1	0	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> CUSTOMERS	10	40	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> DEPARTMENT	4	5	0	0	Not Matched	Not Matched	Not Matched	Not Validated		
<input checked="" type="checkbox"/> EMPLOYEES	7	16	0	0	Not Matched	Not Matched	Not Matched	Not Validated		

The organization, project and run number hierarchy is established with the organization as the top-level parent, where each organization has multiple projects. Each project undergoes a run number increment over time.

The downloaded output package contains the following files:

- **SQL\_Migration\_Report.xlsx**
  - Contains detailed migration and remediation results.
  - Includes the following worksheets:

### **Remediation Sheet**

The **Remediation Sheet** provides detailed information about successfully parsed and processed SQL statements, including:

- Source file name and line number
- Original SQL statement
- SQL parsing and migration status
- Converted target database SQL syntax
- SQL query size
- Java source statement reference
- SQL reference details
- Query pattern type
- Remediation status
- Remediated start and end line numbers

This sheet helps users validate migrated SQL statements and track remediation updates within the Java source code.

### Mitigation Sheet

The **Mitigation Sheet** contains details of SQL statements that could not be fully parsed or automatically remediated. This includes:

- Source file details
- Source line number
- Partial or unparsed SQL statements
- Parsing status
- Parsed message or identified pattern
- Query size information

This sheet assists users in identifying statements that may require manual review or remediation.

- **RemediatedCode.zip**
  - Contains the updated Java source files with remediated SQL statements applied.
  - Users can extract and review the remediated source code before integrating it into the target application environment.

## 10. SQL Extraction & Remediation

The SQL Extraction & Remediation module extracts embedded SQL statements from Java application code. The module supports: Java package analysis, Embedded SQL detection, SQL extraction, SQL translation and Source code remediation. It provides user following two options:

- Import Application Code
- Run Extrator & Remediation

### Import Application Code

This feature allows users to upload and analyze Java application packages to identify and extract embedded SQL statements. The application automatically processes the package by extracting the source code, locating relevant statements within the codebase, and retrieving the SQL queries for migration and remediation. To import application code, the user **must** select the organization, project and run number from the top-level selector as shown below:

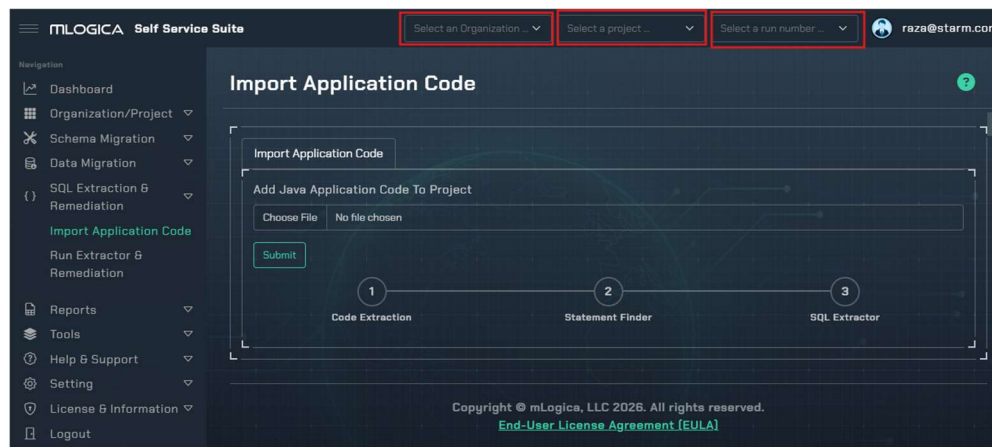
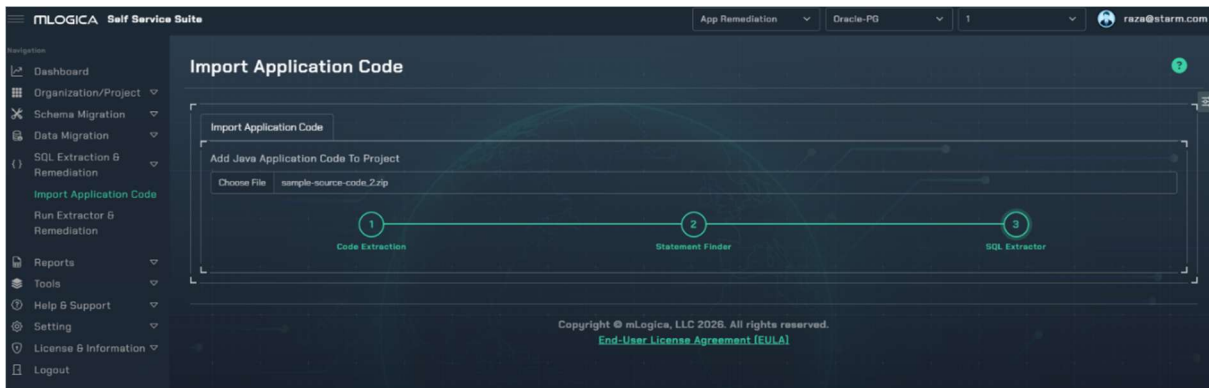


Figure 1: Mandatory selection of the organization, project and run number

For the selected organization, project and run number, the page displays a file selector box, where the user can add Java code as a zip file from their local storage.

The process includes the following steps:

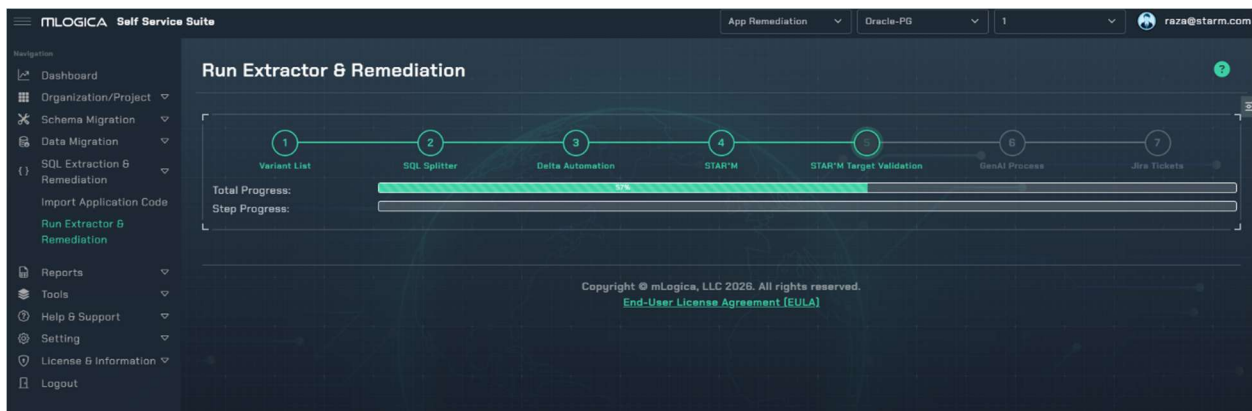
- **Code Extraction:** The application extracts the source code from the provided Java package.
- **Statement Finder:** The application scans the extracted code to identify relevant statements.
- **SQL Extractor:** The application extracts SQL statements detected within the code for further analysis or processing.



Process execution

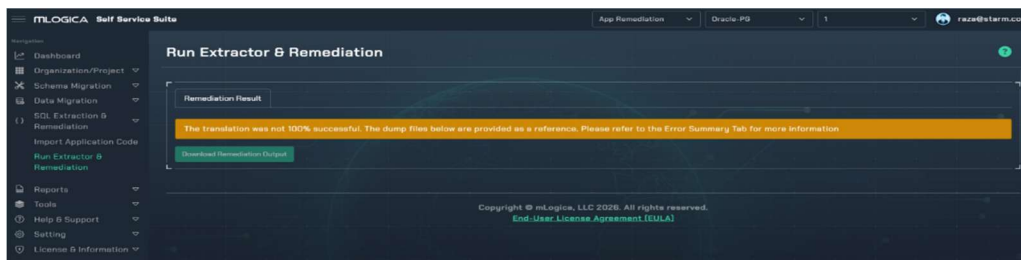
## Run Extractor & Remediation

After the SQL extraction process is completed, the application uses the extracted SQL statements to perform database syntax migration and source code remediation. During this phase, the application converts the extracted SQL statements into the target database syntax and automatically updates the corresponding Java source code at the original extraction locations.



Run Extractor & Remediation

- The remediation process preserves the original source code structure by inserting the translated SQL statements back into the same source file and line reference where the SQL was initially identified. This enables users to review migrated SQL statements directly within the application codebase.
- Once the remediation process is completed, users can download the remediation output package in ZIP format (**application\_output.zip**). The output package contains migration reports and remediated source code artifacts for validation and deployment purposes.



## 11. Ad Hoc Transition

The **Ad Hoc Translation** module allows users to interactively translate and validate the statement from source SQL syntax to target SQL syntax. The source and target databases are fetched from the selected project at the top level. The user also has the option to validate the translated statement during the translation process.



*AdHoc Translation Page*

The Ad Hoc Translation process include the following steps:

- Enter the SQL statement in the input field called **“Source SQL Script.”**
- Select the translation option to translate and validate the given SQL statement on the target database.

The **AdHoc Translation** window provides following Options:

- The **“Translation Only”** button allows the user to translate the source statement and displays the result in the result box called **“Translated SQL.”**

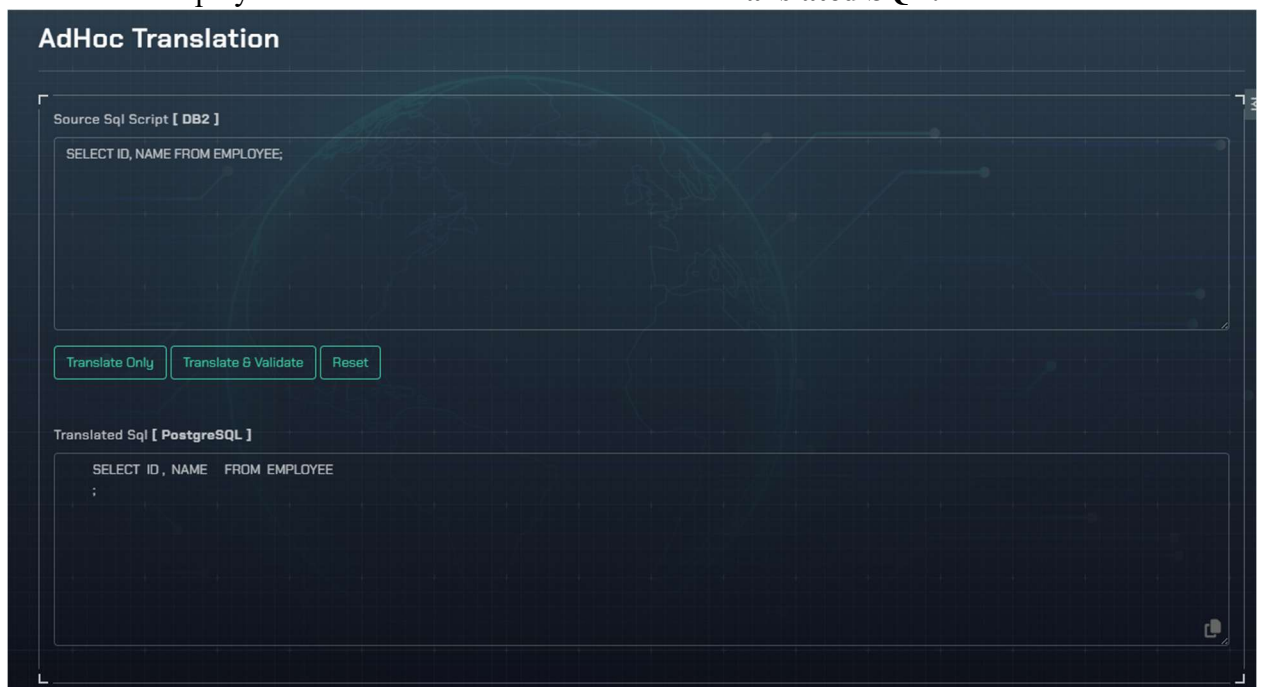


Figure 3: Translation output result

- The **“Translate and Validate”** button allows user to translate and validate the entered statement on the target database. The system executes the entered ad hoc statement on the target database and displays the results to the user in the result box called **“Target Validation Result.”**

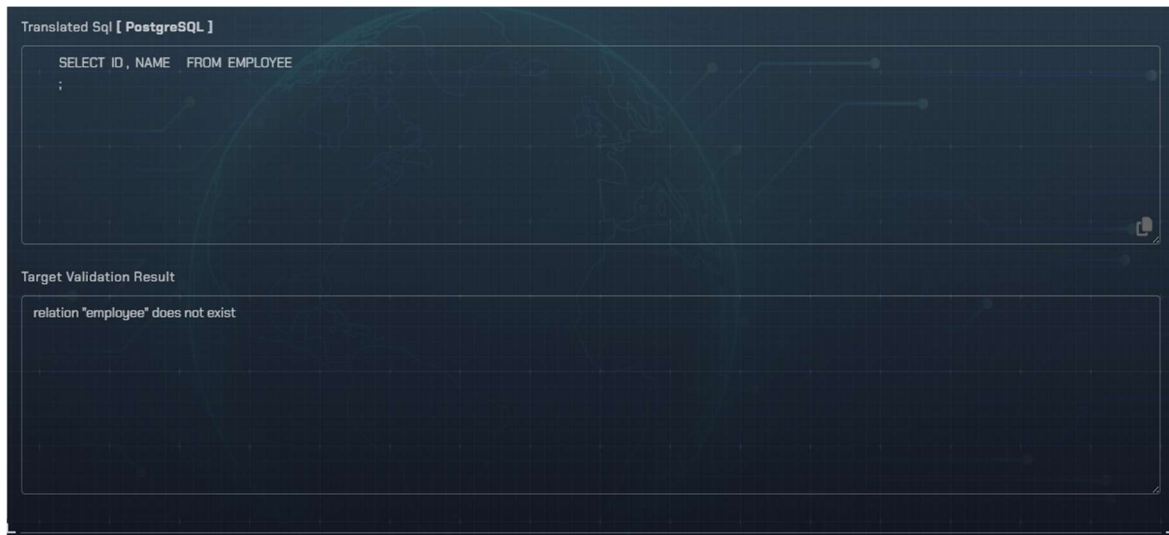
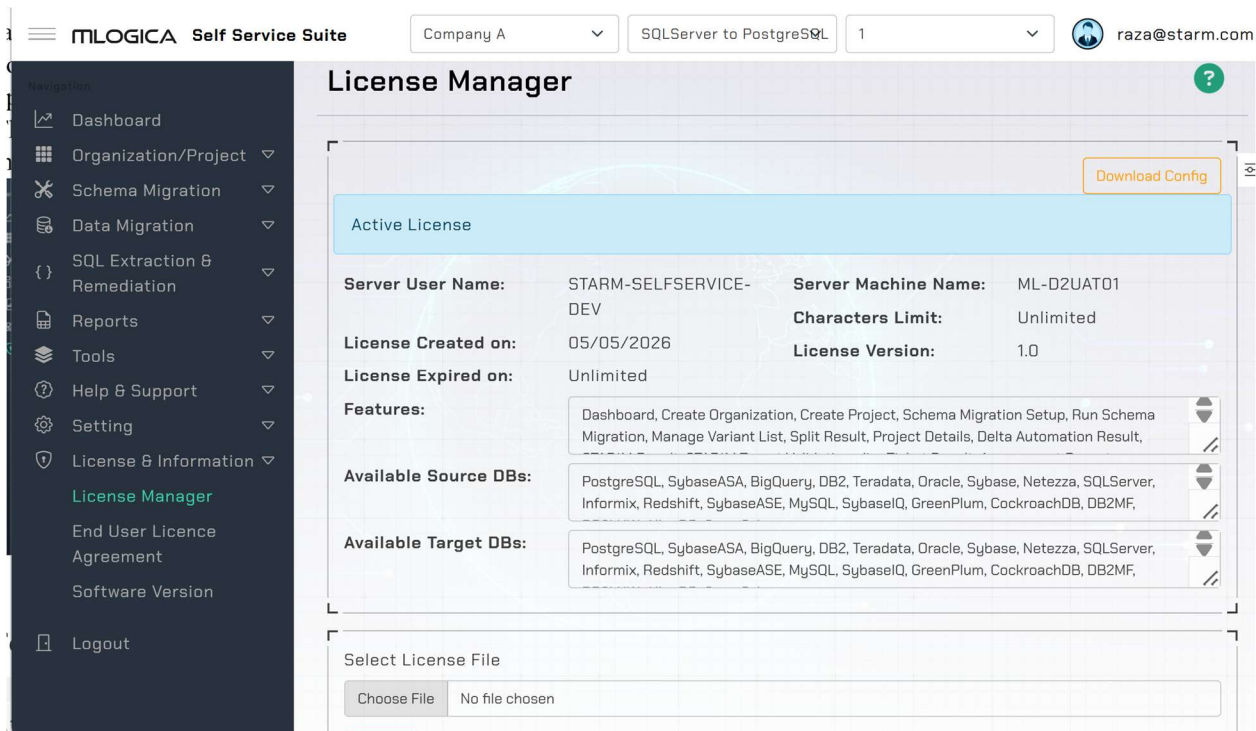


Figure 4: Target validation results

- The **“Reset”** button allows user reset the input and output panels.

## 12. License Management

The **License Manager** module provides centralized license administration. It provides user with access to manage the permissions for using STARM services. It offers detailed information on the active licenses, including the terms and conditions, the scope of permitted use, and any limitations. Additionally, clients can view a full history of their licensing activities, including renewals and expirations. The page also facilitates the downloading of license content, ensuring clients have easy access to their license documentation for compliance and audit purposes. Furthermore, the License Manager includes tools for updating or modifying license details, enabling seamless adjustments to usage rights as needs change. This centralized platform ensures that all licensing information is transparent, accessible, and manageable, enhancing the overall user experience and operational efficiency.



*Licenses Manager Page*

Features include

- Download configuration files
- Upload license files
- Review active licenses
- Review license history
- Manage renewals

License Acquisition Workflow

Download the **config file (License\_Config.txt)** to your device including the Server information, by clicking on the **Download Config** button.

The 2 different ways of receiving the license for the product:

### Option 1 – Email Request

Email the generated License\_Config.txt file to:  
[Starm-Product-Support@mlogica.com](mailto:Starm-Product-Support@mlogica.com)

You will receive the relevant license file from mLogica via email.

### Option 2 – Customer Portal

Login to mLogica Communication Site:

<https://starm-customerportal.mlogica.com/Customer/LicenseRequest>

Upload the **configuration file** (License\_Config.txt) through the License Request Form.

Once mLogica issues the license, you can download the **license.bin** file from the mLogica Communication Site.

## 13. Best Practices

### Recommended Practices

- Validate source schemas before migration
- Review unsupported SQL categories early
- Execute assessment reports prior to production migration
- Use incremental migration runs for large projects
- Perform validation after every migration phase
- Maintain backup copies before cutover
- Review mitigation reports for manual remediation items

## 14. Troubleshooting

Common Issues	Resolution
Connection Failure	Verify host, port, credentials, and firewall access
Translation Failure	Review unsupported syntax and remediation reports
Validation Failure	Review compile-time and runtime logs
Data Validation Errors	Verify source-target consistency
Batch Migration Failure	Re-run failed batches

## 15. Support and Contact Information

mLogica provides:

- 24/7 technical support
- Migration consulting
- Performance optimization assistance
- Custom remediation support
- Training and onboarding services

For support:

- Email: [Starm-Product-Support@mlogica.com](mailto:Starm-Product-Support@mlogica.com)
- Customer Portal: <https://starm-customerportal.mlogica.com>
- Website: <https://www.mlogica.com>

### Copyright Notice

© mLogica. All rights reserved.

This document contains proprietary and confidential information belonging to mLogica. Unauthorized reproduction, distribution, or disclosure is prohibited