

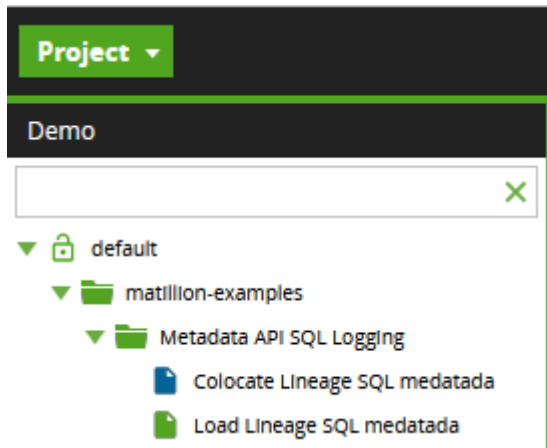
Metadata API SQL Logging

Source : <https://exchange.matillion.com/metl/job/metadata-api-sql-logging/>

Project Files : [https://github.com/anandjha90/MASTER-THE-ART-OF-EXTRACT-TRANSFORM-LOAD-WITH-MATILLION/tree/main/End-To-End%20Project/Matillion API Logger](https://github.com/anandjha90/MASTER-THE-ART-OF-EXTRACT-TRANSFORM-LOAD-WITH-MATILLION/tree/main/End-To-End%20Project/Matillion%20API%20Logger)

A pair of Matillion ETL jobs that maintain a table of the SQL executed by Matillion ETL.

The SQL information can be used to construct data lineage.



The orchestration job uses the [Metadata API](#) to acquire the data. Afterwards it calls the transformation job to copy the newly loaded data into a permanent table.

Features of these jobs include:

- High water mark loading with Matillion ETL
- Demonstration of bronze-to-silver layer data transformation in a medallion data architecture
- How to un-nest and relationalize semi-structured data
- Using an API Extract

Prerequisites

You must first install the [Metadata API](#) Extract connector.

To run the jobs in your own environment you must supply values for the four [Job Variables](#)

Metadata API SQL Logging

Job Variable	Description
prvt_api_username	A Matillion ETL user that has permission to use the API
prvt_api_password	That user's password
prvt_hwm_timestamp	You can leave this blank and the job will still run OK. Set it to 1000000 if you wish to validate the orchestration job
prvt_staging_bucket_name	The name of an S3 bucket to be used for staging. Do not include the <code>s3://</code> prefix nor any path

Steps :

Download the API Profile from the above link and export it successfully. Once done, configure the **Matillion Extract Profile** as shown below by passing matillion below parameters from your **matillion url** :

https://18.232.72.168/#Learn_Matillion/Practice_Matillion/dev_anand/

`http://{url}/rest/v1/group/name/{project_group}/project/name/{project_name}/environment/name/{env}/lineage`

{} -> any value placed in the parameters section will be substituted accordingly

Configure Extract Profile

Endpoints

lineage

New Endpoint

Endpoint Config

Profile Name:

matillion_metadata_api

Endpoint Name:

lineage

Description:

Used for accessing the Matillion ETL "Metadata - Lineage" API from within a Matillion job

URI:

http://{url}/rest/v1/group/name/{project_group}/project/name/{project_name}/environment/name/{env}/lineage

Parameters

Parameter Name	Value	Type	Constant
endTimeStamp		Query	No
env	AWA_AWS	URI	No
project_group	Learn_Matillion	URI	No
project_name	Practice_Matillion	URI	No
startTimestamp	1607954203	Query	No
url	18.232.72.168	URI	No

OK

Cancel

Metadata API SQL Logging

In the Auth section feed your credentials from AWS EC2 page as below where **user -> ec2-user(matillion_username)** and **Password -> instance_id**

Configure Extract Connector : matillion_metadata_api : lineage

1 Source Details 2 Endpoint Configuration 3 Response Configuration 4 Review

Endpoint Configuration

GET Send

URI, Authentication or Parameters changes must be validated by clicking "Send"

Auth Params Body Response Log

☒ ENABLED

Type: Basic Auth

User: Specify the user name

Password: Specify the password

Then click on **Send** and validate the created JSON file from different users working in the same project across different version. **All such details are configured in the form of API which we have imported before (matillion_api_extract profile)**

Configure Extract Connector : matillion_metadata_api : lineage

1 Source Details 2 Endpoint Configuration 3 Response Configuration 4 Review

Endpoint Configuration

GET Send

URI, Authentication or Parameters changes must be validated by clicking "Send"

Auth Params Body Response Log

```

12  },
13  "target": [ {
14    "url": "https://kdgrkf-yc23753.snowflakecomputing.com",
15    "fqm": "\"DEMO_DATABASE\".\"DEMO_SCHEMA\".\"AWS_EXCEL_EMPLOYEE_DATA\""
16  } ],
17  "sourceConnectionUrl": "Not Applicable",
18  "sourceUsername": "Not Applicable",
19  "environment": {
20    "name": "AWA_AWS",
21    "url": "kdgrkf-yc23753",
22    "schema": "DEMO_SCHEMA",
23    "database": "DEMO_DATABASE",
24    "user": "AWA20250209",
25    "connectionOptions": { }
26  },
27  "job": {
28    "name": "excel_query",
29    "type": "ORCHESTRATION"
30  }

```

You can paste a sample response into this screen and validate it to create a profile without a live endpoint. JSON File validated successfully with no errors. Validate

Cancel Back Next

Metadata API SQL Logging

Once **Validated** Click on **Next** and Close it. Disable the Paging

Configure Extract Connector : matillion_metadata_api : lineage

1 Source Details 2 Endpoint Configuration 3 Response Configuration 4 Review

Response Configuration

Select Fields

Repeating Element: /

Each item in the array will correspond to a row in the target table

- STRUCT
 - sql ARRAY
 - target ARRAY
 - source ARRAY
 - environment STRUCT
 - urlPath TEXT
 - job STRUCT
 - componentName TEXT
 - sourceConnectionUrl TEXT
 - componentCanvasName TEXT
 - taskType TEXT
 - startTimestamp LONG
 - sourceUsername TEXT
 - endTimestamp LONG
 - loggedTimestamp LONG

Paging

☐ DISABLED

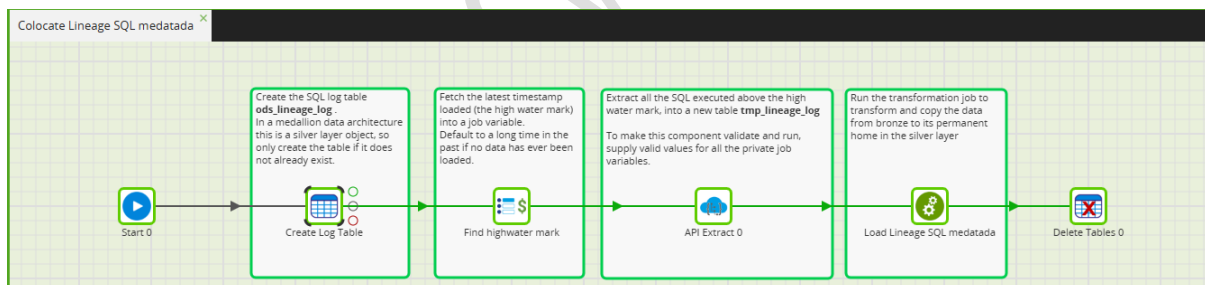
Strategy

Select paging type

Pagination may not occur if the strategy does not match that of the endpoint.

Cancel Back Next

Then export the attached JSON file and configure the parameters as shown below :



Metadata API SQL Logging

Properties | SQL | Export | Help

Create Table OK

Name	Value	Status
Name	Create Log Table	OK
Create/Replace	Replace	OK
Database	[Environment Default]	OK
Schema	[Environment Default]	OK
New Table Name	ods_lineage_log	OK
Table Type	Permanent	OK
Columns	log_logged_timestamp, NUMBER, 38, ...	OK
Default DDL Collation		OK
Primary Keys		OK
Clustering Keys		OK
Data Retention Time ...		OK
Comment		OK

Properties | SQL | Export | Help

```
CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" (
  "log_logged_timestamp" NUMBER(38, 0) NOT NULL,
  "log_start_datetime" TIMESTAMP,
  "log_end_datetime" TIMESTAMP,
  "log_duration_seconds" NUMBER(38, 0),
  "log_component_canvas_name" VARCHAR(64),
  "log_component_name" VARCHAR(64),
  "log_env_name" VARCHAR(32),
  "log_database_name" VARCHAR(32),
  "log_schema_name" VARCHAR(32),
  "log_job_name" VARCHAR(64),
  "log_job_type" VARCHAR(32),
  "log_task_type" VARCHAR(32),
  "log_job_path" VARCHAR(256),
  "log_sql_dialect" VARCHAR(256),
  "log_sql_user" VARCHAR(256),
  "log_sql_endpoint" VARCHAR(2048),
  "log_raw_sql" VARCHAR
)
```

Columns

Column Name	Data Type	Size	Precision	Default Value	Not Null	Unique	Comment
log_logged_timestamp	NUMBER	38	0		Yes	No	
log_start_datetime	TIMESTAMP				No	No	
log_end_datetime	TIMESTAMP				No	No	
log_duration_seconds	NUMBER	38	0		No	No	
log_component_canvas...	VARCHAR	64			No	No	
log_component_name	VARCHAR	64			No	No	
log_env_name	VARCHAR	32			No	No	
log_database_name	VARCHAR	32			No	No	
log_schema_name	VARCHAR	32			No	No	
log_job_name	VARCHAR	64			No	No	
log_job_type	VARCHAR	32			No	No	
log_task_type	VARCHAR	32			No	No	
log_job_path	VARCHAR	256			No	No	
log_sql_dialect	VARCHAR	256			No	No	
log_sql_user	VARCHAR	256			No	No	
log_sql_endpoint	VARCHAR	2048			No	No	
log_raw_sql	VARCHAR				No	No	

+ -

☐ Text Mode

☐ Use Grid Variable

OK Cancel

Properties | Export | Help

Query Result To Scalar OK

Name	Value	Status
Name	Find highwater mark	OK
Basic / Advanced	Advanced	OK
SQL Query	SELECT CASE WHEN MAX("log_logged...	OK
Scalar Variable Mapp...	prvt_hwm_timestamp, log_logged_ti...	OK

In the created job variable as a part of JSON and pass the values accordingly

Manage Job Variables						
Name	Type	Behaviour	Visibility	Value	Description	
prvt_api_password	Text	Shared	Private	i-05cca91fef682088		
prvt_api_username	Text	Shared	Private	ec2-user		
prvt_hwm_timestamp	Numeric	Shared	Private	1000000		
prvt_staging_bucket_na...	Text	Shared	Private	matillionclass		



Metadata API SQL Logging

SQL Query

Metadata Explorer

DEMO_SCHEMA

Table Search...

Name	Type	Size
AGENTS		
AWS_EXCEL_EMPLOYEE_DATA		
BRANCH_COMMIT_RELATION		
COMMIT		
COMMIT_FILE		
COMMIT_PARENT		
CUSTOMERS		
CUSTOMER_CONTACT		

Variables

Name	Default value
prvt_api_password	i-05cca91fef682088
prvt_api_username	ec2-user
prvt_hwm_timestamp	1000000
prvt_staging_bucket_name	matillionclass

Manage Variables

```
1 SELECT CASE WHEN MAX("log_logged_timestamp") IS NULL THEN 1000000
2 ELSE MAX("log_logged_timestamp") + 1
3 END AS "log_logged_timestamp" + 1
4 FROM "ods_lineage_log"
```

Sample

Limit: 10

Row Count

log_logged_timest...
1741177109399

OK Cancel

Properties

Export

Help

API Extract

OK

Name	Value	Status
Name	API Extract 0	OK
Profile	matillion_metadata_api	OK
Data Source	lineage	OK
URI Params	project_name, Practice_Matillion, url, 18.232.72.168, project_group, Learn_Matillion, env, AWA_...	OK
Query Params	startTimestamp, \${prvt_hwm_timestamp}	OK
Header Params		OK
Username	\${prvt_api_username}	OK
Password	*****	OK
Page Limit	1	OK
Location	s3://\${prvt_staging_bucket_name}	OK
Warehouse	[Environment Default]	OK
Database	[Environment Default]	OK
Schema	[Environment Default]	OK
Target Table	tmp_lineage_log_new	OK

Give ec2-user & instance id for user-pwd combination

This is the temporary table name which will be created dynamically so as to keep all the new log files for every

Location

Enter a container path here, or select from the containers you own below.

s3:// \${prvt_staging_bucket_name}

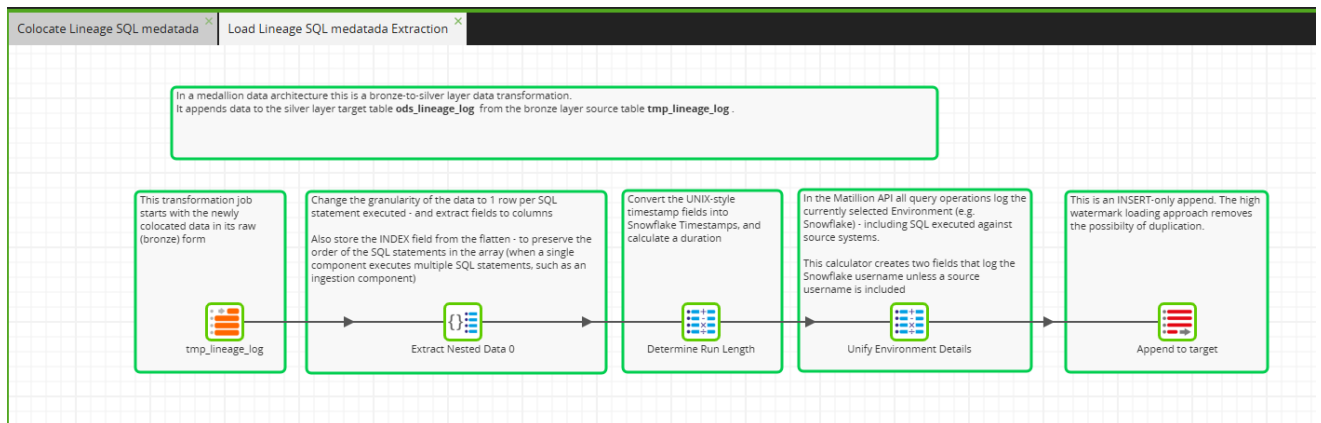
Go

S3 Buckets > \${prvt_staging_bucket_name}

Name	Date Created
matillionclass	2024-12-04 14:30 UTC+5:30

Metadata API SQL Logging

Once orchestration job have been configured, go ahead and configure below transformation job



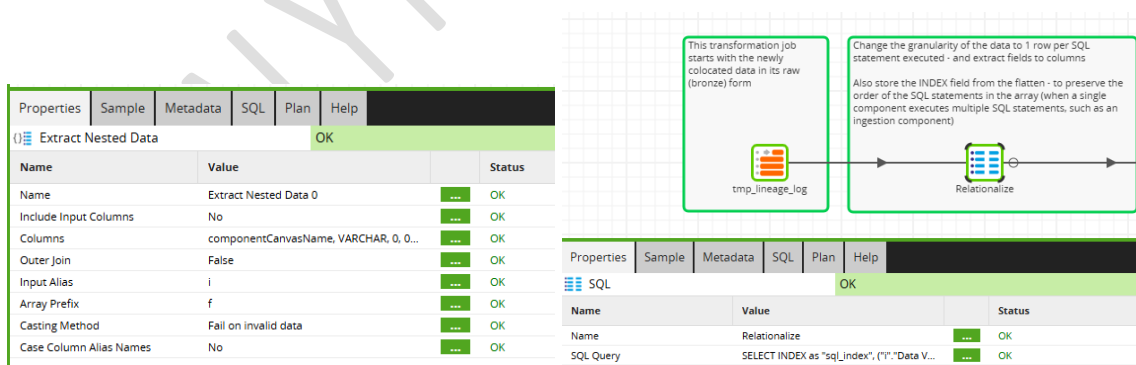
Properties	Sample	Metadata	SQL	Plan	Help
Table Input OK					
Name	Value	Status			
Name	tmp_lineage_log	OK			
Database	[Environment Default]	OK			
Schema	[Environment Default]	OK			
Target Table	tmp_lineage_log_new	OK			
Column Names	Data Value	OK			
Offset		OK			

Properties	Sample	Metadata	SQL	Plan	Help
SELECT "Data Value" FROM "DEMO_DATABASE"."DEMO_SCHEMA"."tmp_lineage_log_new"					

Now we will extract the parameters from the log files generated in the form of JSON through **API Extract Profile** as shown below using **Extract Nested Data** component or one can use **SQL Script** component and can extract the components via **SCRIPT**.

If you are not able to understand the SQL script refer my Youtube Lecture no 39 :

<https://www.youtube.com/live/TbuH6SOD7uw?si=wYRrjSmyCyKGPg4K>



Metadata API SQL Logging

Columns

Define the columns to include in the data structure:

☒ Data Value

☐ componentCanvasName VARCHAR as componentCanvasName
☒ componentName VARCHAR as componentName
☒ endTimestamp NUMBER as endTimestamp
☐ environment VARIANT as environment

☒ database VARCHAR as database
☒ name VARCHAR as environment_name
☒ schema VARCHAR as schema
☒ url VARCHAR as environment_url
☐ connectionOptions VARIANT as connectionOptions
☒ user VARCHAR as user

☐ job VARIANT as job

☒ name VARCHAR as job_name
☒ type VARCHAR as type

☐ source VARIANT(Array) as source

☐ source-element VARIANT as source-element

☒ fqcn VARCHAR as source-element_fqcn
☒ url VARCHAR as source-element_url

☒ sourceUsername VARCHAR as sourceUsername

☐ sql VARIANT(Array) as sql

☐ sql-element VARIANT as sql-element

☒ dialect VARCHAR as dialect
☒ query VARCHAR as query

☒ startTimestamp NUMBER as startTimestamp

☐ target VARIANT(Array) as target

☐ target-element VARIANT as target-element

☒ fqcn VARCHAR as target-element_fqcn
☒ url VARCHAR as target-element_url

☒ urlPath VARCHAR as urlPath
☒ sourceConnectionUrl VARCHAR as sourceConnectionUrl

Autofill Reset Select All Clear

☐ Use Variable

OK Cancel

SQL Query

Variables

Name	Default value
------	---------------

```

1 SELECT INDEX as "sql_index",
2   ("i"."Data Value":"loggedTimestamp")::VARCHAR AS "loggedTimestamp",
3   ("i"."Data Value":"componentCanvasName")::VARCHAR AS "componentCanvasName",
4   ("i"."Data Value":"componentName")::VARCHAR AS "componentName",
5   ("i"."Data Value":"endTimestamp")::NUMBER AS "endTimestamp",
6   ("i"."Data Value":"environment":"name")::VARCHAR AS "env_name",
7   ("i"."Data Value":"environment":"database")::VARCHAR AS "database",
8   ("i"."Data Value":"environment":"schema")::VARCHAR AS "schema",
9   ("i"."Data Value":"environment":"url")::VARCHAR AS "url",
10  ("i"."Data Value":"environment":"user")::VARCHAR AS "user",
11  ("i"."Data Value":"job":"name")::VARCHAR AS "job_name",
12  ("i"."Data Value":"job":"type")::VARCHAR AS "job_type",
13  ("i"."Data Value":"sourceUsername")::VARCHAR AS "sourceUsername",
14  ("fi"."Value":"dialect")::VARCHAR AS "dialect",
15  ("fi"."Value":"query")::VARCHAR AS "query",
16  ("i"."Data Value":"startTimestamp")::NUMBER AS "startTimestamp",
17  ("i"."Data Value":"urlPath")::VARCHAR AS "urlPath",
18  ("i"."Data Value":"sourceConnectionUrl")::VARCHAR AS "sourceConnectionUrl",
19  ("i"."Data Value":"taskType")::VARCHAR AS "taskType"
20 FROM $T(tmp_lineage_log) "i",
21      LATERAL FLATTEN (input => "i"."Data Value":"sql") "fi"
  
```

Run

Press "Run" to test script execution.

Metadata API SQL Logging

Using **calculator component**, extract the below attributes. You can also extract cost associated with the duration based on credit utilized. Try that too

Calculations		
Expressions		start_datetime
start_datetime	→	1 TO_TIMESTAMP_NTZ("startTimestamp" / 1000)
end_datetime	→	
run_duration	→	

Calculations		
Expressions		end_datetime
start_datetime	→	1 TO_TIMESTAMP_NTZ("endTimestamp" / 1000)
end_datetime	→	
run_duration	→	

Calculations		
Expressions		run_duration
start_datetime	→	1 TIMESTAMPDIFF(second, "start_datetime" , "end_datetime")
end_datetime	→	
run_duration	→	
Mat_User	→	

Calculations		
Expressions		Mat_User
start_datetime	→	1 REGEXP_SUBSTR("urlPath", '([~/]+)', 1, 3)
end_datetime	→	
run_duration	→	
Mat_User	→	

Calculations		
Expressions		target_endpoint
target_endpoint	→	1 case when "dialect" = 'Snowflake' then "environment_url" else "sourceConnectionUrl" end
target_user	→	

Calculations		
Expressions		target_user
target_endpoint	→	1 case when "dialect" = 'Snowflake' then "user" else "sourceUsername" end
target_user	→	

Metadata API SQL Logging

Table Output Properties :

Properties

Export

Sample

Metadata

SQL

Plan

Help

Table Output

OK

Name	Value		Status
Name	Append to target	...	OK
Warehouse	[Environment Default]	...	OK
Database	[Environment Default]	...	OK
Schema	[Environment Default]	...	OK
Target Table	ods_lineage_log	...	OK
Fix Data Type Mismatches	No	...	OK
Column Mapping	loggedTimestamp, log_logged_timestam...	...	OK
Order By	log_logged_timestamp, Ascending	...	OK
Truncate	Append	...	OK

This is an INSERT-only append. The high watermark loading approach removes the possibility of duplication.



Append to target

Run the orchestration job in different version by switching into it and along with that other jobs also you can run and and verify the result in snowflake

Final Matillion Output :

Colocate Lineage SQL metadata	Load Lineage SQL metadata Extraction	Load Lineage SQL metadata	Task - Colocate Lineage SQL metadata
Environment: AWA_AWS	Version: dev_anand	Queued: 20:13:57	Duration: 6.9s
Job	Component	Duration	Queued
Colocate Lineage SQL metadata	Start 0	6.9s	20:13:57
Colocate Lineage SQL metadata	Create Log Table	1.0s	20:13:57
Colocate Lineage SQL metadata	Find highwater ...	1.1s	20:13:58
Colocate Lineage SQL metadata	API Extract 0	1.8s	20:13:59
Colocate Lineage SQL metadata	Load Lineage SQL ...	2.9s	20:14:01
Load Lineage SQL metadata	tmp_lineage_log	1.1s	20:14:01
Load Lineage SQL metadata	Extract Nested D...	0.1s	20:14:02
Load Lineage SQL metadata	Determine Run L...	0.1s	20:14:02
Load Lineage SQL metadata	Unify Environme...	0.0s	20:14:02
Load Lineage SQL metadata	Append to target	1.5s	20:14:02
Colocate Lineage SQL metadata	Delete Tables 0	0.1s	20:14:04



Metadata API SQL Logging

Final Snowflake Output :

Web, HTTPC

ods_lineage_log
tmp_lineage_log
tmp_lineage_log_new

```
SELECT * FROM DEMO_DATABASE.DEMO_SCHEMA."ods_lineage_log" ORDER BY "log_logged_timestamp" DESC;
```

Results Chart

#	log_logged_timestamp	log_start_datetime	log_end_datetime	#	log	log_component_Name	log_env_Name	log_database_Name	log_schema_Name	log_job_name
1	1741185838738	2025-03-05 14:43:57.738	2025-03-05 14:43:58.731	1	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage
2	1741185815581	2025-03-05 14:43:34.967	2025-03-05 14:43:35.580	1	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage
3	1741185779173	2025-03-05 14:42:58.199	2025-03-05 14:42:59.171	1	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage
4	1741177285768	2025-03-05 12:21:24.727	2025-03-05 12:21:25.763	1	null	Table Output	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Load Lineage SQL
5	1741177115588	2025-03-05 12:18:33.863	2025-03-05 12:18:35.583	2	null	Table Output	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Load Lineage SQL
6	1741177109398	2025-03-05 12:18:28.575	2025-03-05 12:18:29.397	1	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage
7	1741177053738	2025-03-05 12:17:32.876	2025-03-05 12:17:33.734	1	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage
8	1741176045341	2025-03-05 12:00:43.807	2025-03-05 12:00:45.336	2	null	Table Output	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Load Lineage SQL
9	1741176037870	2025-03-05 12:00:35.985	2025-03-05 12:00:37.884	2	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage
10	1741175411261	2025-03-05 11:50:10.016	2025-03-05 11:50:11.257	1	null	Table Output	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Load Lineage SQL
11	1741175405640	2025-03-05 11:50:04.470	2025-03-05 11:50:05.634	1	null	Create Table	AWA_AWS	DEMO_DATABASE	DEMO_SCHEMA	Colocate Lineage

Web, HTTPC

ods_lineage_log
tmp_lineage_log
tmp_lineage_log_new

```
SELECT * FROM DEMO_DATABASE.DEMO_SCHEMA."ods_lineage_log" ORDER BY "log_logged_timestamp" DESC;
```

Results Chart

#	log_job_type	log_task_type	log_job_path	log_sql_dialect	log_sql_user	log_sql_endpoint	log_raw_sql
1	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB
2	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB
3	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB
4	TRANSFORMATION	Run Job	/Learn_Matillion/Practice_Matillion/dev_suraj/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATAB
5	TRANSFORMATION	Run	/Learn_Matillion/Practice_Matillion/dev_suraj/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATAB
6	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_suraj/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB
7	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_anand/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB
8	TRANSFORMATION	Run	/Learn_Matillion/Practice_Matillion/dev_suraj/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATAB
9	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_suraj/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB
10	TRANSFORMATION	Run	/Learn_Matillion/Practice_Matillion/dev_anand/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATAB
11	ORCHESTRATION	Run	/Learn_Matillion/Practice_Matillion/dev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TAB

ods_lineage_log

tmp_lineage_log

tmp_lineage_log_new

ods_lineage_log38 Rows

log_logged_timestampNUMBER(38,0)

log_start_datetimeTIMESTAMP_NTZ(20)

log_end_datetimeTIMESTAMP_NTZ(20)

log_duration_secondsNUMBER(38,0)

log_component_Canvas_NameVARCHAR(64)

log_component_NameVARCHAR(64)

log_sm_NameVARCHAR(32)

log_database_NameVARCHAR(32)

log_schema_NameVARCHAR(32)

log_job_nameVARCHAR(64)

log_job_typeVARCHAR(32)

log_task_typeVARCHAR(32)

log_job_pathVARCHAR(256)

log_sql_dialectVARCHAR(256)

SELECT * FROM DEMO_DATABASE.DEMO_SCHEMA."ods_lineage_log" ORDER BY "log_logged_timestamp" DESC;

ResultsChart

		log_sql_dialect	log_sql_user	log_sql_endpoint	log_raw_sql	log_mat_us
1	ev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_anand
2	ev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_anand
3	ev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_anand
4	ev_suraj/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("log_logged_timestam	dev_suraj
5	ev_suraj/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("log_logged_timestam	dev_suraj
6	ev_suraj/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_suraj
7	ev_suraj/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_suraj
8	ev_suraj/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("log_logged_timestam	dev_suraj
9	ev_suraj/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_suraj
10	ev_anand/Load Lineage SQL metadata Extraction	Snowflake	AWA20250209	kdgrykf-yc23753	insert into "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("log_logged_timestam	dev_anand
11	ev_anand/Colocate Lineage SQL metadata	Snowflake	AWA20250209	kdgrykf-yc23753	CREATE OR REPLACE TABLE "DEMO_DATABASE"."DEMO_SCHEMA"."ods_lineage_log" ("to	dev_anand

***** THANK YOU *****