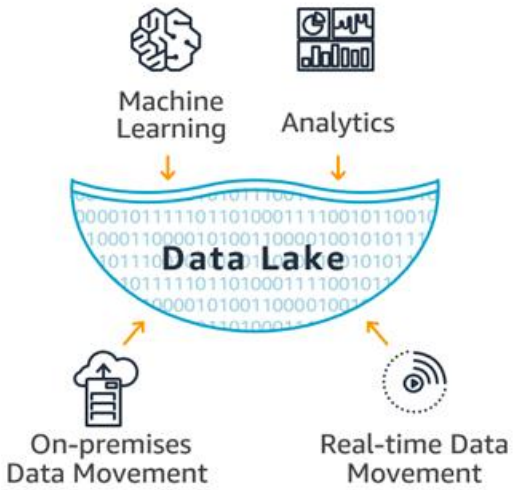

BUILDING A GLUE STREAMING AND TRANSACTIONAL DATA LAKE SOLUTION

ABSTRACT
This solution is a Terraform implementation showcasing Glue streaming, used together with Apache Iceberg Data Lake Framework to build a transactional Data Lake on AWS.
SitadConsulting

Table of Contents

Building a Glue streaming and Transactional data lake solution.

1.0 Introduction

This guide is a Terraform implementation of the AWS workshop https://catalog.us-east-1.prod.workshops.aws/workshops/d261102a-5d6d-4b27-ab96-63d170138db9/en-US

There are 9 key resources that set the stage for proper execution of this solution, these are

i. Glue Job - gsw_datagenerator: responsible for generating streaming data for hydration of Kinesis data stream

ii. Glue Job – gsw_create_iceberg_tables: responsible for the creation of Iceberg tables; leader_board, transactions and transactions_history

iii. Lambda Function – gsw_lambda-sitadinfra-dev: responsible for creating Athena External table – product_data table

iv. Glue Job – gsw_transform: responsible for stream data processing and transformation. Performing record insertion and update to tables in CDC fashion and stream enrichment.

v. Kinesis Data Stream – gsw_stream-dev: responsible for storing the data stream generated by gsw_datagenerator

vi. S3 Bucket – aws-glue-<account-id>-dev: Used as S3 location for the tables, implementation codes, Athena query result output location and Temp location for job runs.

vii. S3 Bucket – aws-glue-assets-<account-id>-<region>: Used as S3 location for Glue custom visual transforms scripts, jobs and Temp location for job runs.

viii. Athena Workgroup – adhoc_query: Provides interactive query execution editor for querying table data, specifies the query result output location.

ix. Iam Role and Policy - AWSLambdaServiceExecutionPolicy-sitadinfra-dev, AWSGlueServiceRole-sitadinfra-dev, AWSLambdaServiceExecutionRole-sitadinfra-dev

2.0 Clone Repository and Initialize Modules and Apply

To begin, please clone the repository to your workspace. Execute the command shown below:

cd /home/dev01 (this workspace may be different in your own case)
git clone https://github.com/sitadconsulting/aws_infra.git
cd aws_infra/projects/proj1 (review the files in this directory and update the "dev.auto.tfvars" file to reflect values from your own environment)

proj1]$ ***terraform init***

Initializing the backend...
Initializing modules...
- athena_query_result_s3_bucket in ../../modules/s3_bucket
- athena_workgroup in ../../modules/athena_workgroup
- glue_developer_iam_group in ../../modules/iam_group
- glue_developer_iam_group_policy_attachment in ../../modules/iam_group_policy_attachment
- glue_service_iam_policy_attachment in ../../modules/iam_role_policy_attachment
- glue_service_iam_role in ../../modules/iam_role
- glue_service_notebook_iam_policy_attachment in ../../modules/iam_role_policy_attachment
- glue_service_notebook_iam_role in ../../modules/iam_role
- gsw_create_iceberg_tables_job in ../../modules/glue_job
- gsw_datagenerator_job in ../../modules/glue_job
- gsw_iceberg_catalog_database in ../../modules/glue_catalog_database
- gsw_lambda_function in ../../modules/lambda_function
- gsw_stream_registry in ../../modules/glue_registry
- gsw_stream_registry_table in ../../modules/glue_catalog_table
- gsw_stream_schema in ../../modules/glue_schema
- gsw_stream_table in ../../modules/glue_catalog_table
- gsw_transform_job in ../../modules/glue_job
- kinesis_stream in ../../modules/kinesis_stream
- lambda_service_iam_policy in ../../modules/iam_policy
- lambda_service_iam_policy_attachment in ../../modules/iam_role_policy_attachment
- lambda_service_iam_role in ../../modules/iam_role
- s3_assets_bucket in ../../modules/s3_bucket
- s3_object in ../../modules/s3_object
- s3_object_file_upload in ../../modules/s3_object
- s3_object_zipfile_upload in ../../modules/s3_object
- sagemaker_notebook_iam_policy_attachment in ../../modules/iam_role_policy_attachment
- sagemaker_notebook_iam_role in ../../modules/iam_role

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.7.0"...

- Finding hashicorp/local versions matching "~> 2.3.0"...
- Finding latest version of hashicorp/template...
- Finding hashicorp/tls versions matching "4.0.4"...
- Finding hashicorp/archive versions matching "2.4.0"...
- Installing hashicorp/template v2.2.0...
- Installed hashicorp/template v2.2.0 (signed by HashiCorp)
- Installing hashicorp/tls v4.0.4...
- Installed hashicorp/tls v4.0.4 (signed by HashiCorp)
- Installing hashicorp/archive v2.4.0...
- Installed hashicorp/archive v2.4.0 (signed by HashiCorp)
- Installing hashicorp/aws v5.7.0...
- Installed hashicorp/aws v5.7.0 (signed by HashiCorp)
- Installing hashicorp/local v2.3.0...
- Installed hashicorp/local v2.3.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.Initializing the backend... Initializing modules...

proj1]$ ***terraform apply***

3.0 Create Iceberg tables

Log on to Glue console, on the left pane, click on ETL Jobs. On the displayed right pane, scroll down to the section Your Jobs and click on the job – gsw_create_iceberg_tables. On the displayed job page, click on the Run button at the top right-hand corner to execute the job. Once the job finishes executing, on the left pane, under Data Catalog, click on Tables, on the displayed right pane, you will see the details of the three iceberg tables created as shown in Fig 3.0 below.

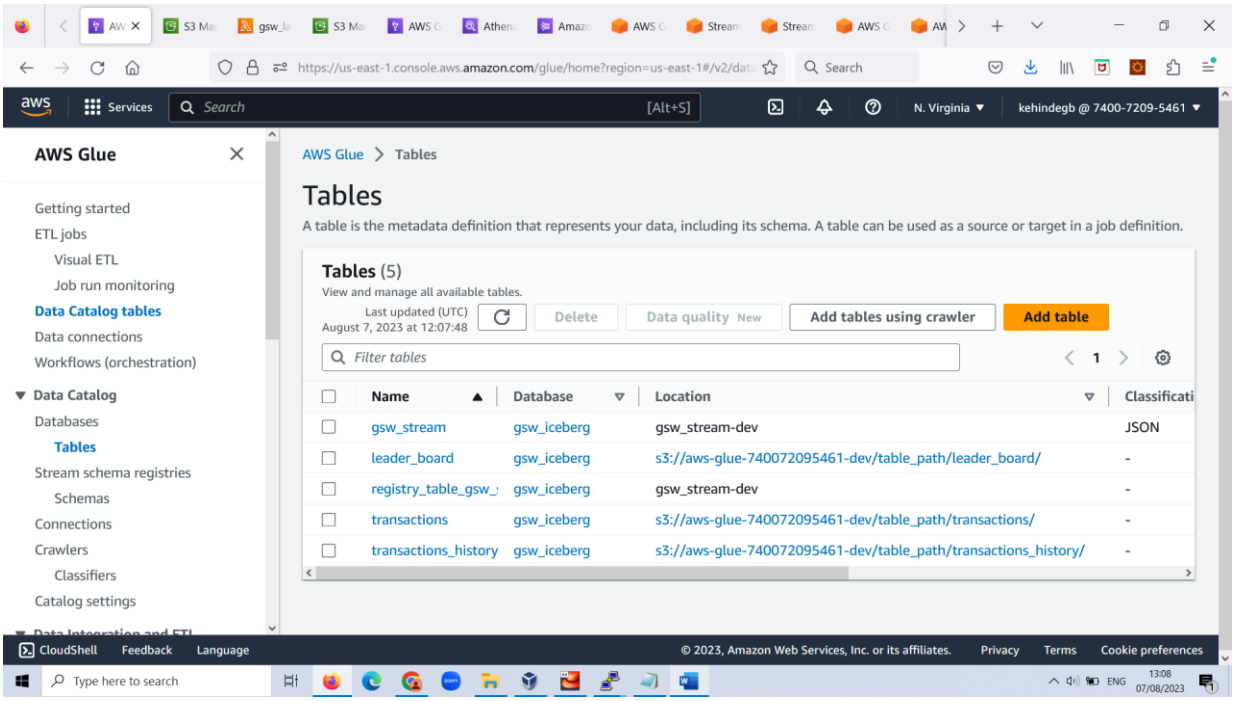

Fig 3.0 Shows the three iceberg tables – Leader_board, transactions and transaction_history and their s3 locations.

3.1 Create Athena External table

Log on to the Lambda console, on the left pane, click on Functions. On the displayed right pane, click on the Function name - gsw_lambda-sitadinfra-dev. On the displayed Lambda Function page, scroll down to the Code Source section, click on the Test menu item to invoke the Lambda Function, on the displayed Lambda invocation page, enter “test” for name and click on the Invoke button. Once the execution of the Lambda Function completes, access Glue console, on the left pane under Data Catalog, click the menu item Tables, on the displayed Tables page on the right pane, you will see a new table “product_data” as shown below:

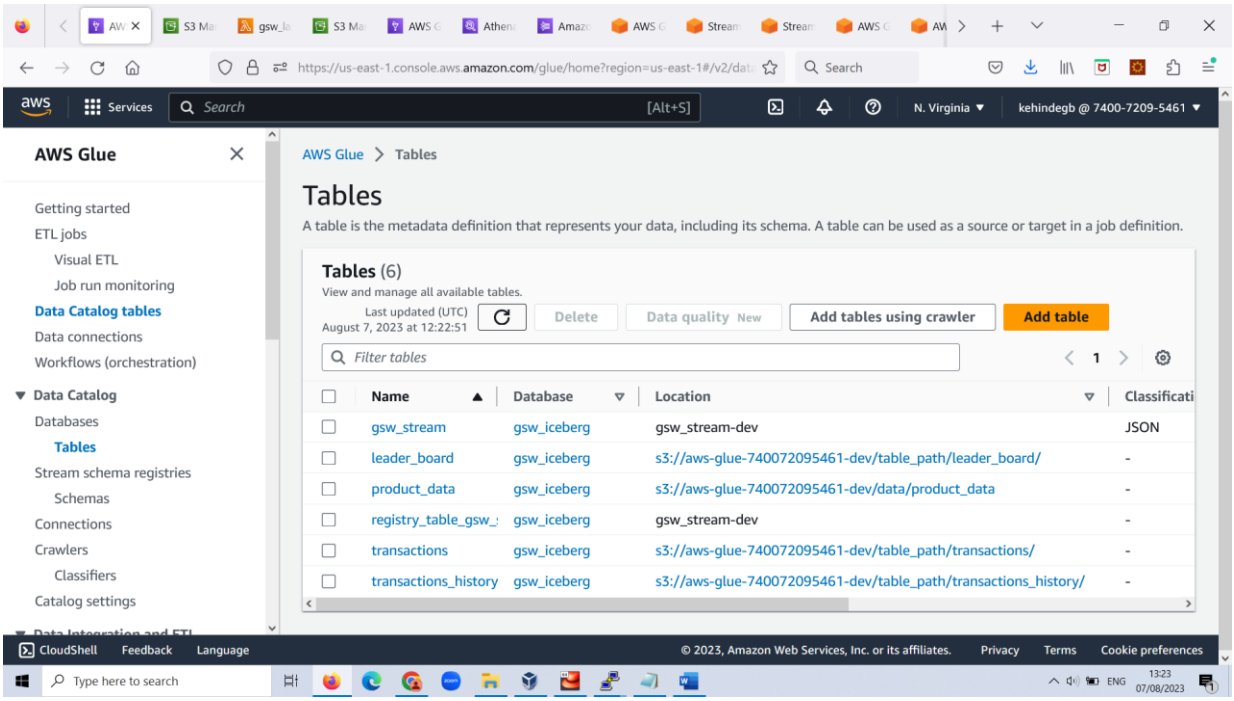

Fig 3.1 Shows the s3 location of the Athena table – product_data

3.2 Generate Data to hydrate Kinesis Data Stream

Access Glue console, on the left pane, click on ETL jobs, on the displayed right pane jobs window, scroll down to the section Your jobs, click on the job – gsw_datagenerator, on the displayed right pane job configuration page, click on Run menu item to execute the job to start the hydration process. Once the job execution is completed, access Kinesis console, on the displayed page, click on Data streams, on the displayed right pane Data streams page, click on the Data stream – gsw_stream_dev, on the displayed page, click on menu item Data viewer, on the displayed page, click on the Shard drop down menu to locate your shard, click on it, on the Starting position drop down menu box, select “Trim horizon”, click on the button Get records, if no records are displayed, click the button Next records until you can see records as shown below:

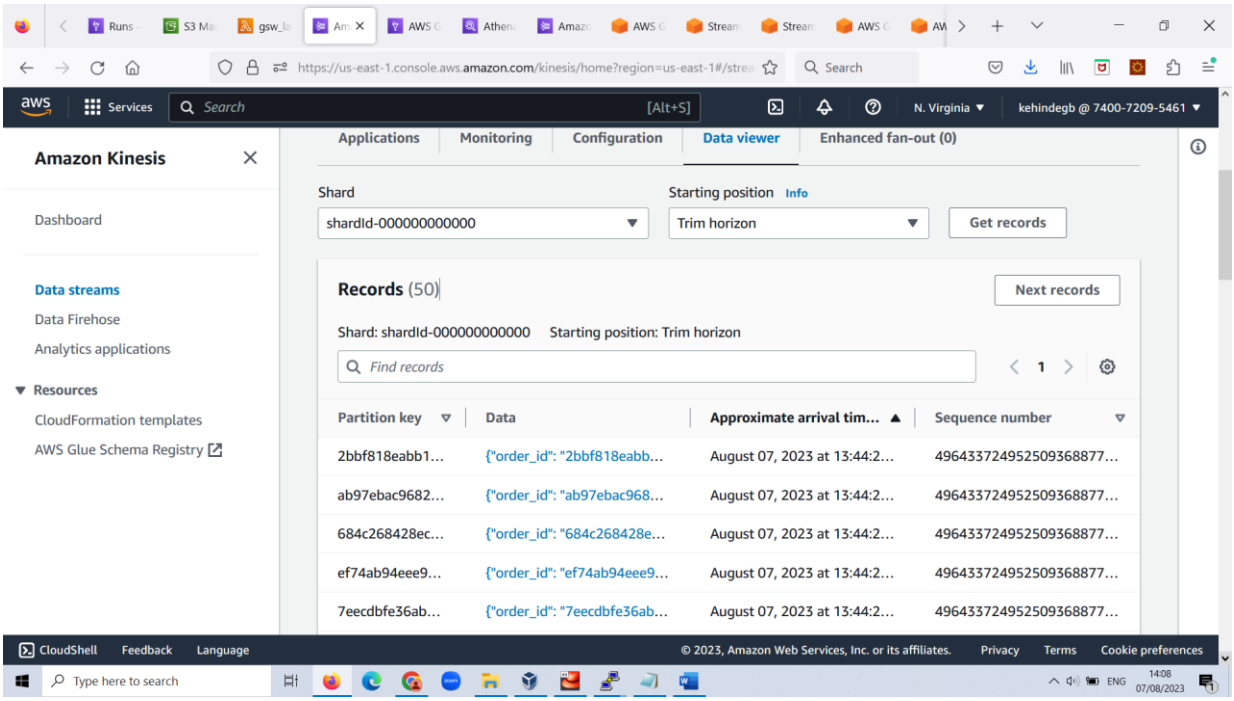

Fig 3.2: Shows Kinesis Data stream records after executing job – gsw_datagenerator

3.3 Perform Athena query on product_data table

Access Athena console and click on Query editor, on the displayed query editor page, in the query window on the right pane, enter this query statement –

SELECT * FROM "AwsDataCatalog"."gsw_iceberg"."product_data" limit 10;

Scroll down and click the Run button to execute the query as specified above. Figure 3.3 below shows the output of the query against the product_data table. This query is being returned from the content of the product_data.csv file store in s3 location of the table - s3://aws-glue-xxxxxxxxxxxx-dev/data/product_data

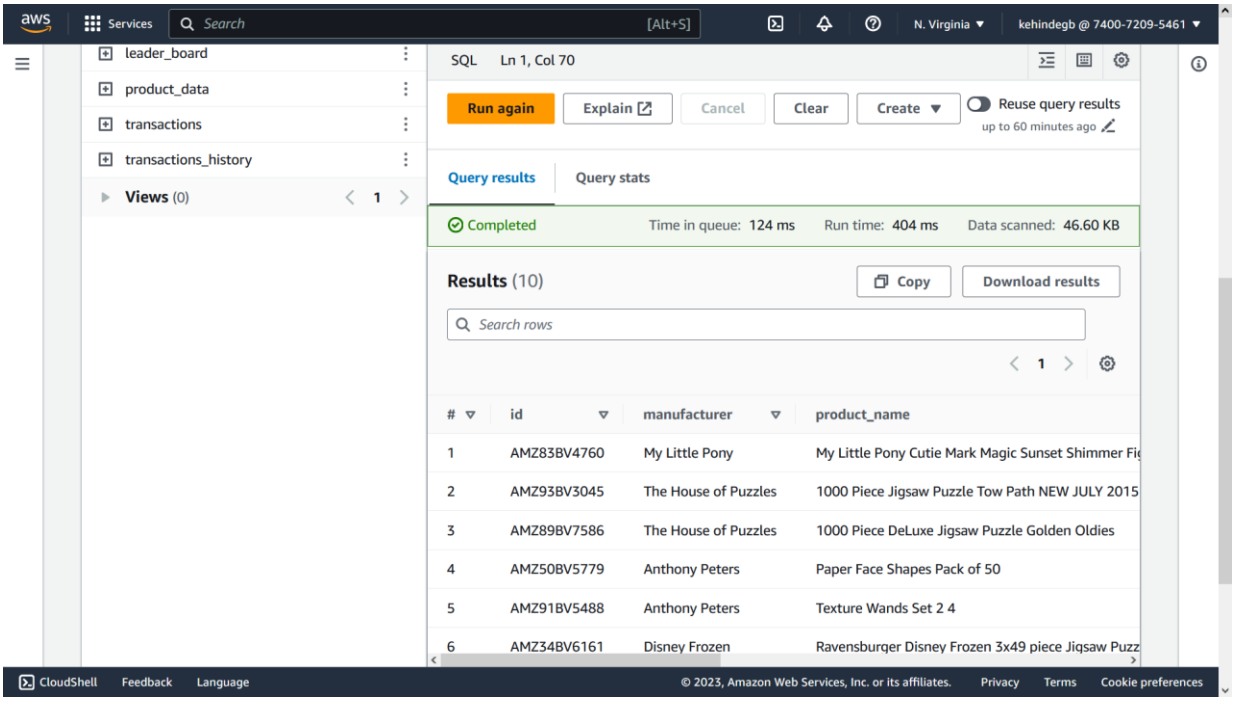

Figure 3.3 Athena query of product_data table

4.0 Initiate Glue streaming transform job

The Glue streaming transform job – gsw_transform, enables the processing and transformation of data that is flowing to Kinesis Data streams – gsw_stream-dev, and data resident in product_data table. This job further processes the stream to perform Change Data Capture for update of transactions and transactions_history tables. The job also processes the stream to update the leader_board table. You will find below the output of Athena query executed against the tables. The steps required to see these outputs are stated below:

i. Execute the gsw_transform job

ii. Whiles the gsw_transform job is running, execute the gsw_datagenerator job

The query outputs of the tables are presented below.

4.1 Perform Athena query on leader_board table

Access Athena console and click on Query editor, on the displayed query editor page, in the query window on the right pane, enter this query statement –

SELECT * FROM "AwsDataCatalog"."gsw_iceberg"."leader_board" limit 10;

Scroll down and click the Run button to execute the query as specified above. Figure 4.1 below shows the output of the query against the leader_board table.

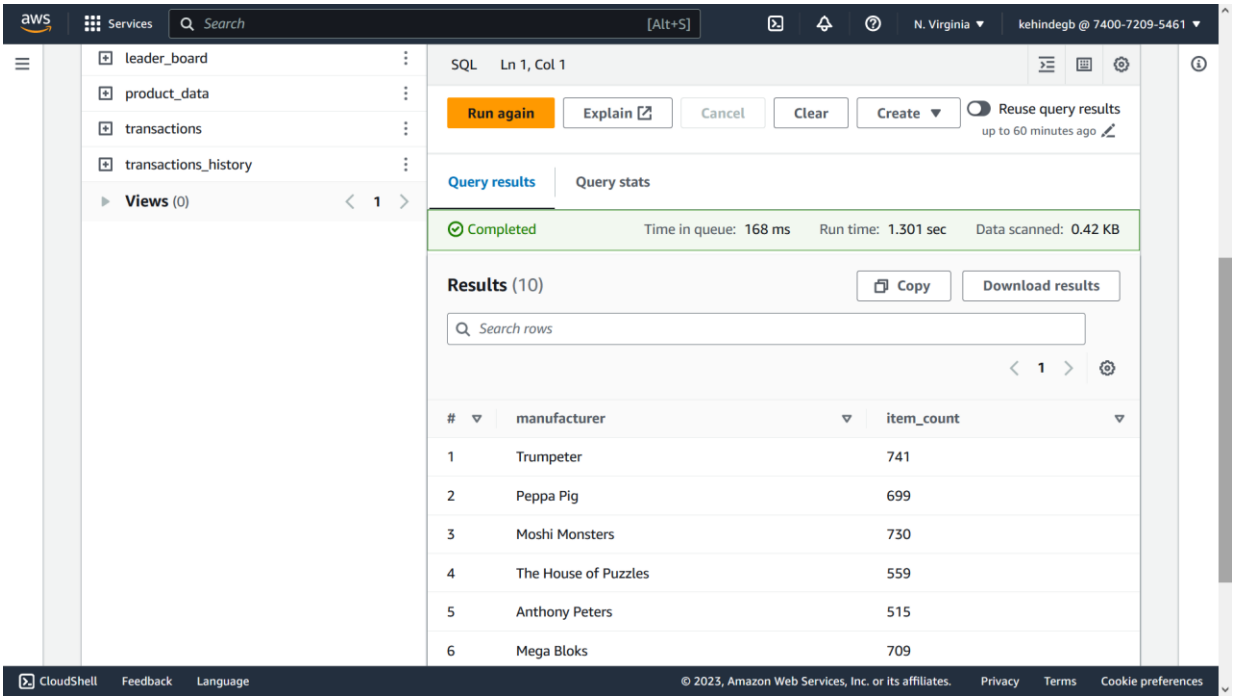

Figure 4.1 Athena query result of leader_board table.

4.2 Perform Athena query on transactions table

Access Athena console and click on Query editor, on the displayed query editor page, in the query window on the right pane, enter this query statement –

SELECT * FROM "AwsDataCatalog"."gsw_iceberg"."transactions" limit 10;

Scroll down and click the Run button to execute the query as specified above. Figure 4.2 below shows the output of the query against the transactions table.

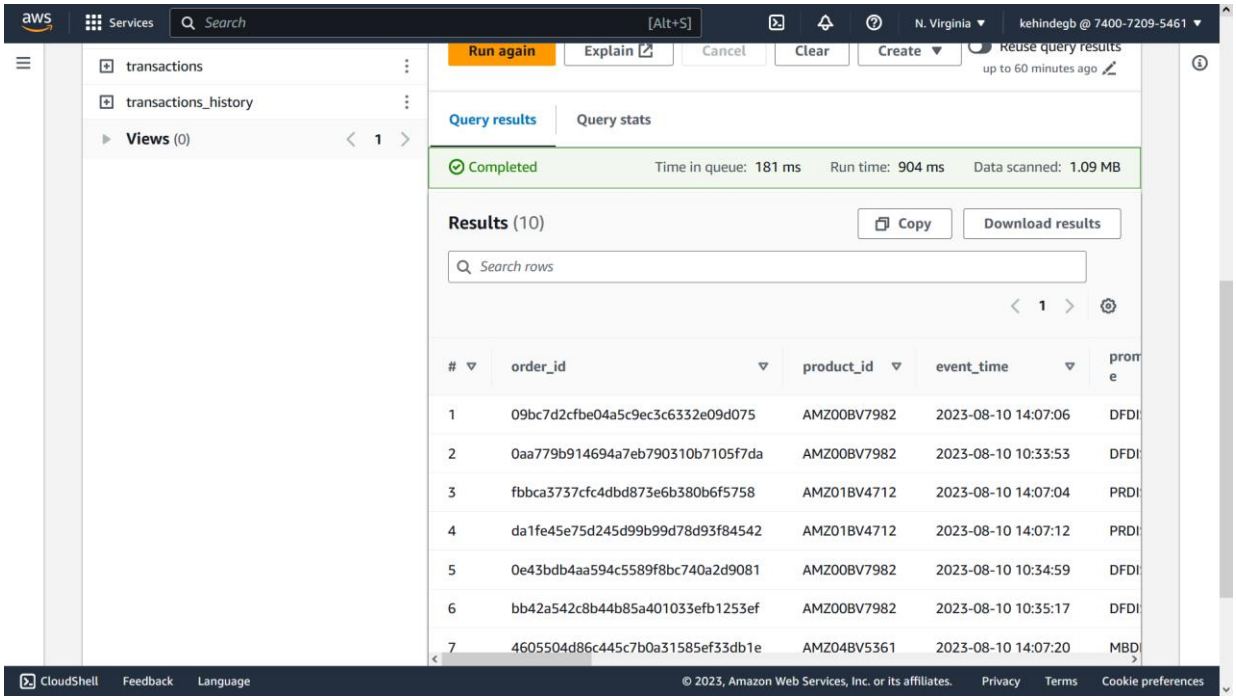

Figure 4.2 Athena query result of transactions table

4.3 Perform Athena query on transactions_history table

Access Athena console and click on Query editor, on the displayed query editor page, in the query window on the right pane, enter this query statement –

SELECT * FROM "AwsDataCatalog"."gsw_iceberg"."transactions_history" limit 10;

Scroll down and click the Run button to execute the query as specified above. Figure 4.3 below shows the output of the query against the transactions table

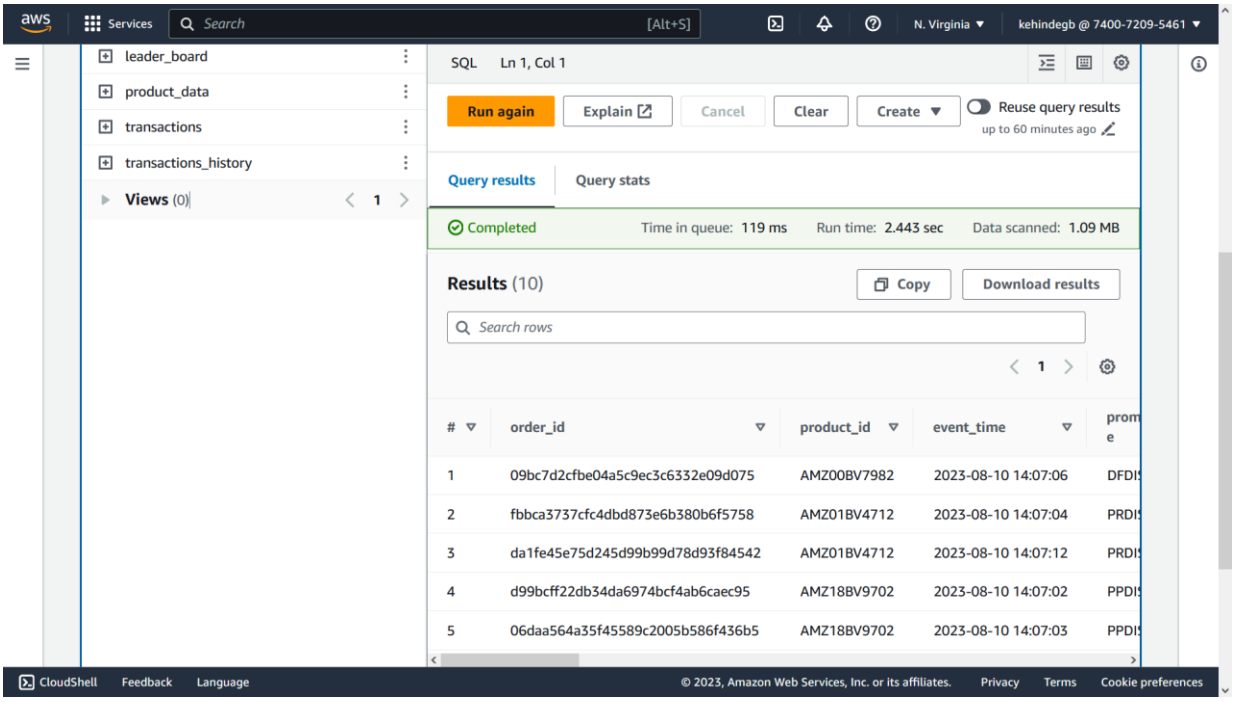

Figure 4.3 Athena query result of transactions_history table