Retail Data Management

MADE BY- Ashish Chamel

COURSE- Extract, Transform, and Load (ETL)

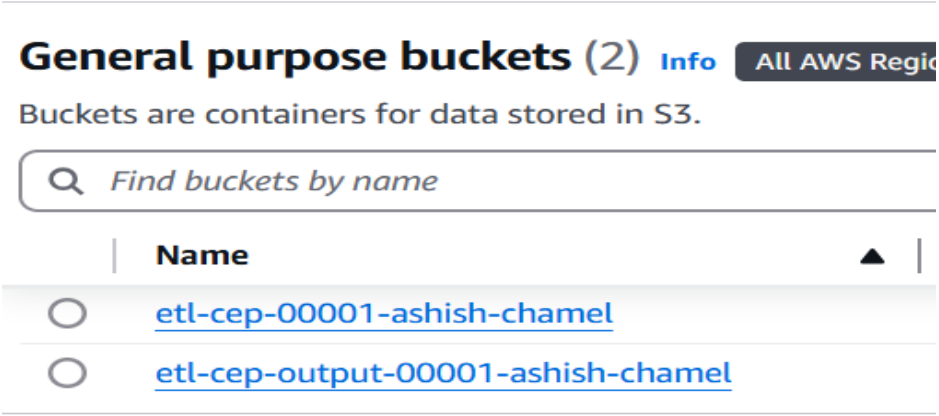
DATE OF SUBMISSON-05/04/2025

Step-1

1. Created buckets in s3-

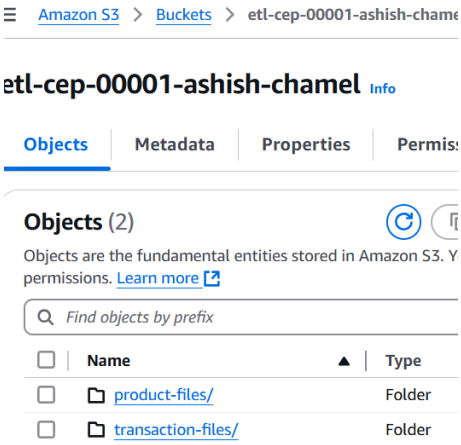
   input bucket- etl-cep-00001-ashish-chamel (input files)

   output bucket- etl-cep-output-00001-ashish-chamel(output from a etl job)

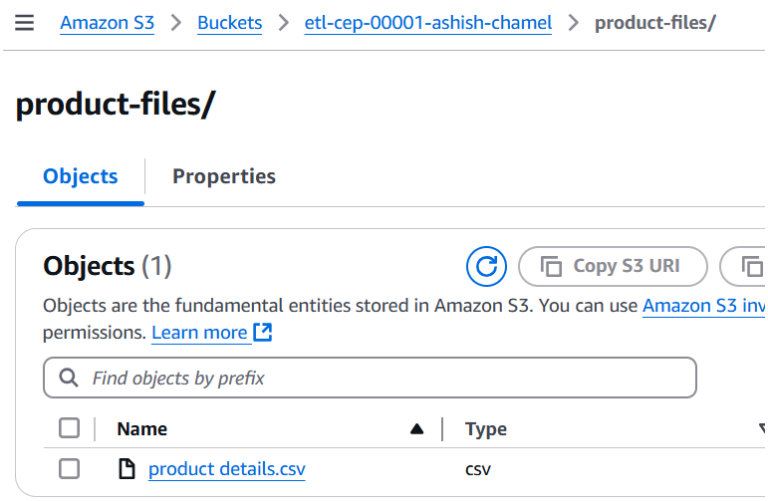


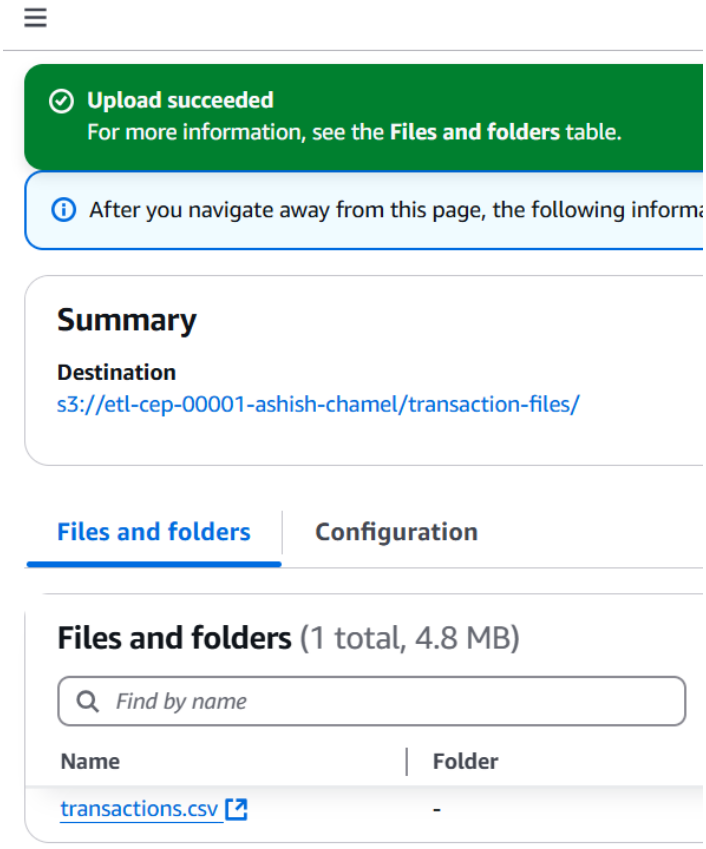
Step-2

1. Created subfolders inside etl-cep-00001-ashish-chamel ie-
   - Product-files
   - transaction-files



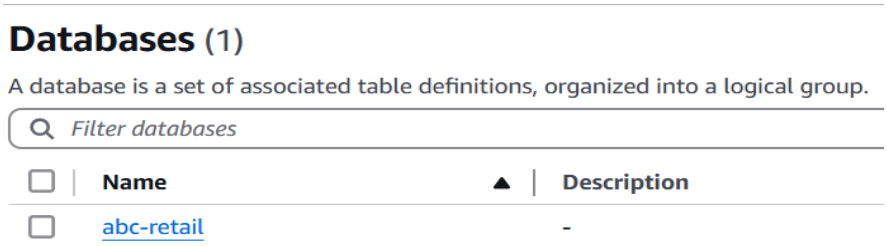
2. Uploaded product details.csv , transaction.csv to corresponding folders ie product files and transaction files





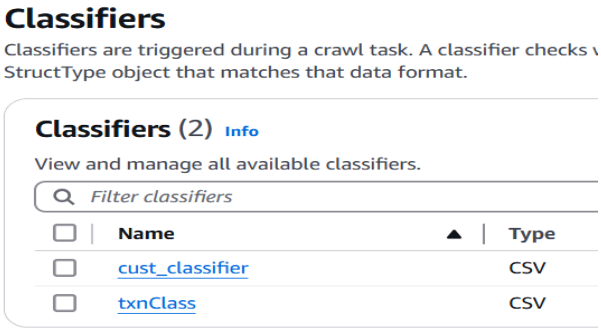
Step-3

1. Created a new database ie. abc-retail

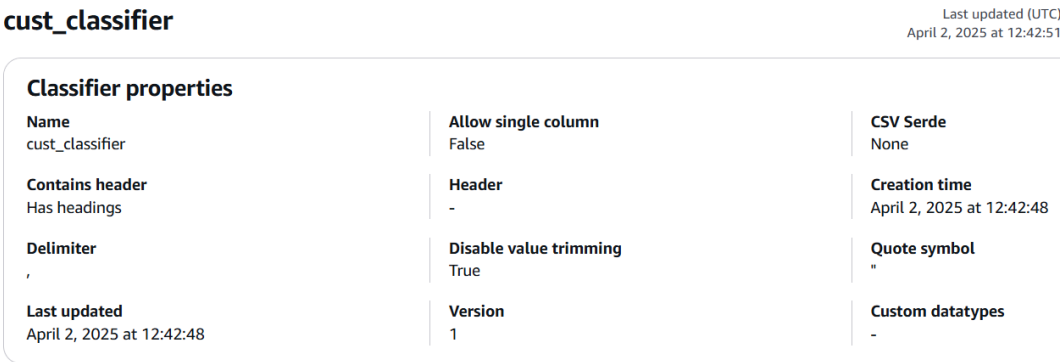


Step-4

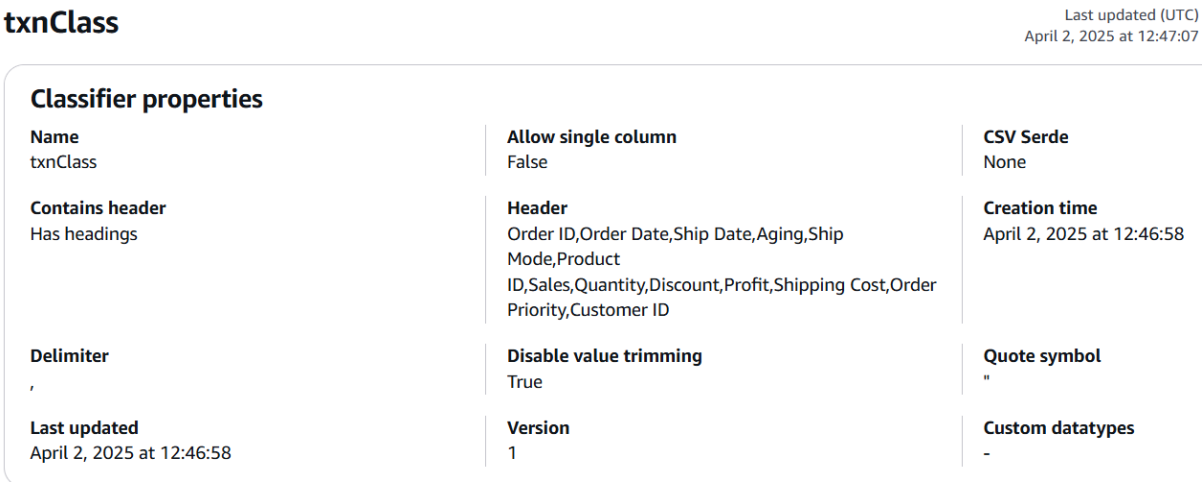
1. Here I setup 2 classifiers ie cust_classifier & txn class,to read transaction data and product data.



2. Created the 1st classifier ie cust_classifier with following settings
   - Classifier type and properties as CSV
   - CSV Serde – optional as None
   - Column delimiter as comma(,)
   - Quote symbol as Double-quote(“)
   - Column headings as Has headings

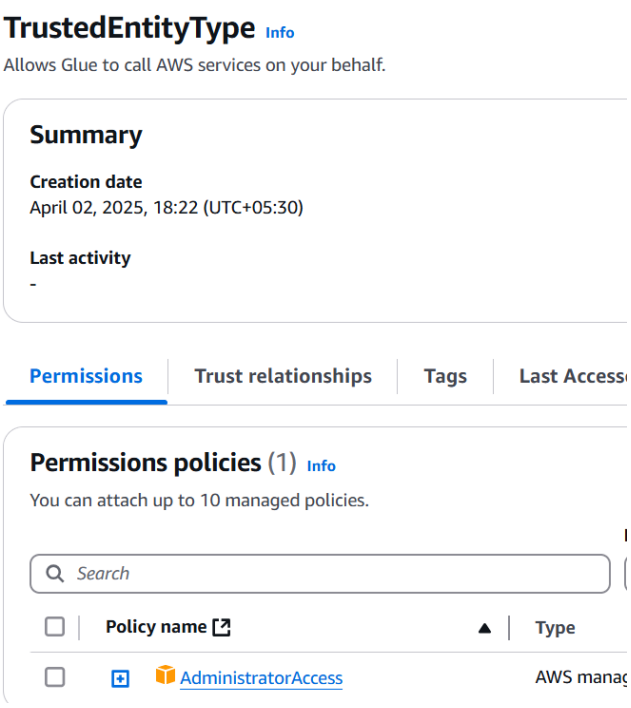


3. Created the 2nd classifier ie txnClass with following settings
    - Classifier type and properties as CSV o CSV Serde – optional as None
    - Column delimiter as comma(,)
    - Quote symbol as Double-quote(“)
    - Column headings as Has headings ie
      Order ID, Order Date, Ship Date, Aging, Ship Mode, Product ID, Sales, Quantity, Discount, Profit, Shipping Cost, Order Priority, Customer ID



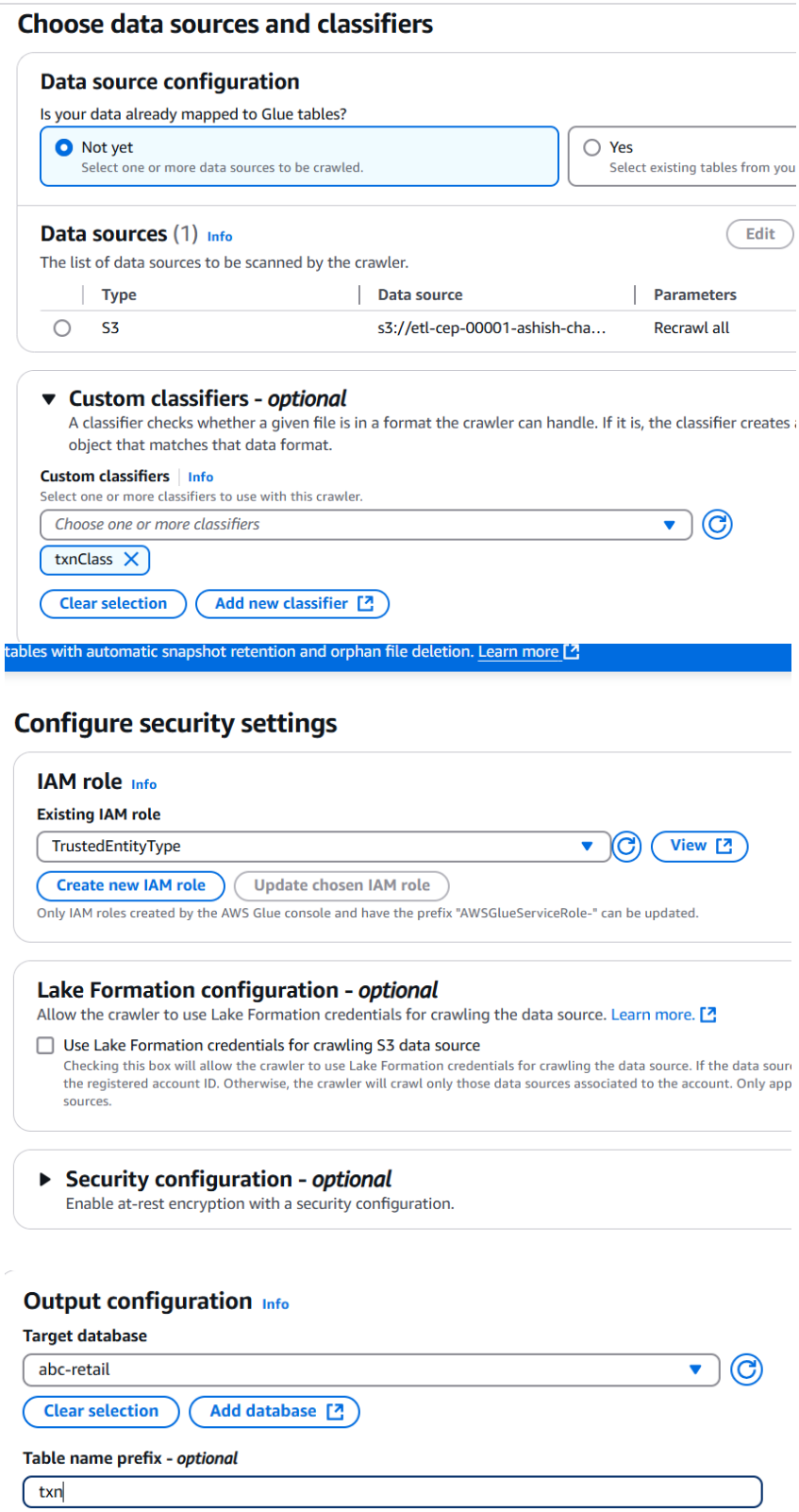
Step-5

1. Created a IAM role with administrator access as TrustedEntityType

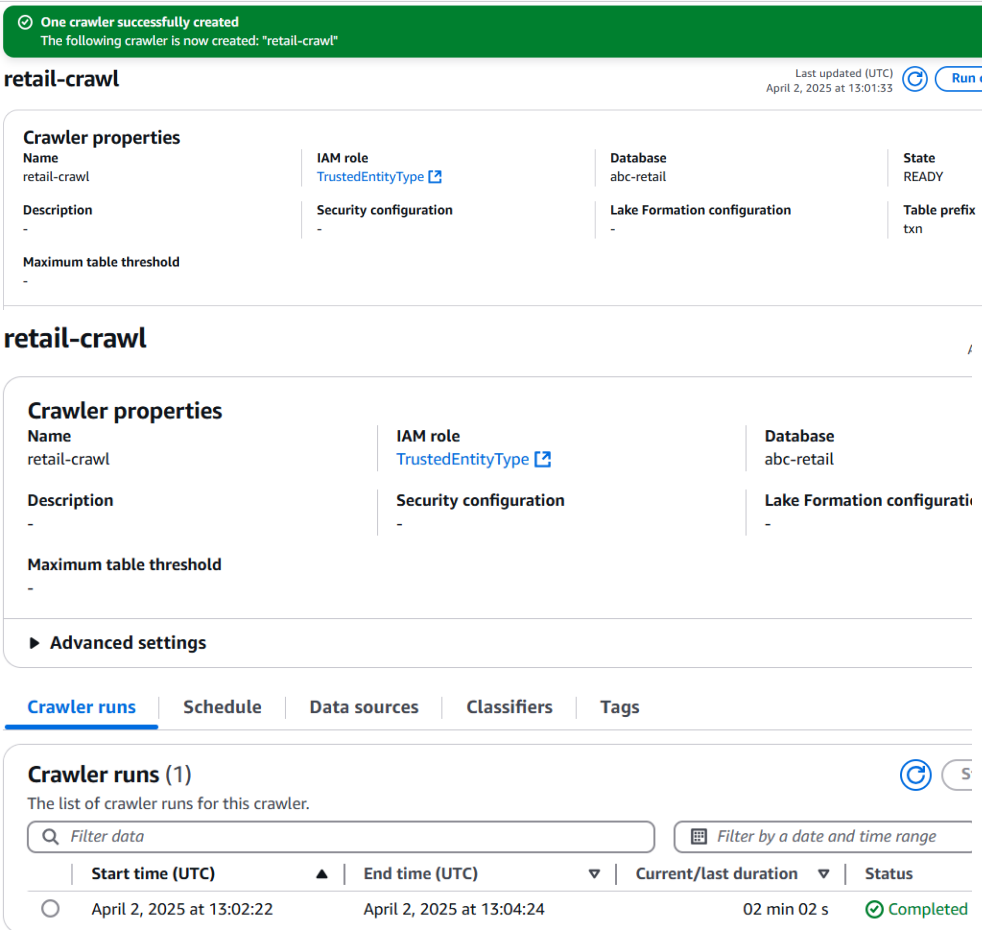


Step-6

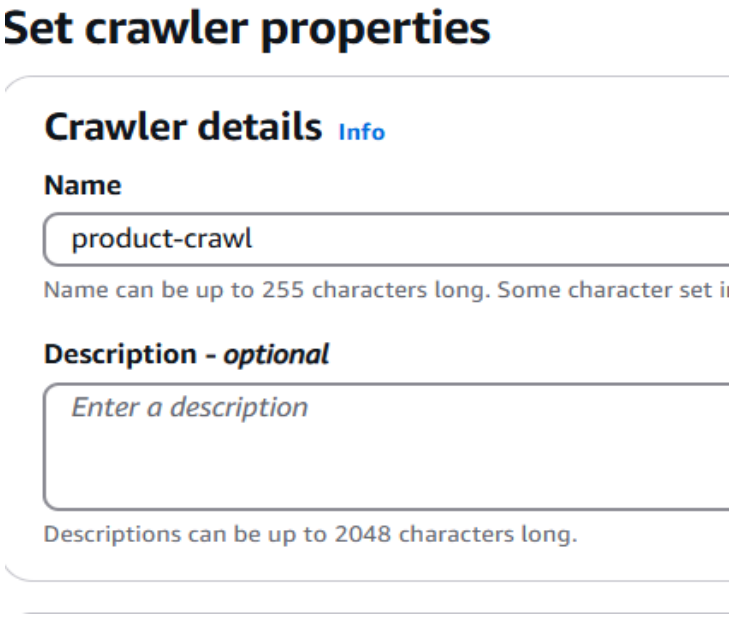
1. Setting up two crawlers ie. retail-crawl and product-crawl for extracting metadata
2. Setup of a retail-crawl



3. Retail-crawl created and ran to extract metadata into database ie. abc-retail



4. Creating a 2nd crawler ie product-crawl to extract the metadata.



Choose data sources and classifiers

Data source configuration

Is your data already mapped to Glue tables?

- (•) Not yet
  Select one or more data sources to be crawled.
- ( ) Yes
  Select existing tables

Data sources (1) Info

The list of data sources to be scanned by the crawler.

	Type	Data source
( )	S3	s3://etl-cep-00001-ashish-chamel/prod...

▼ Custom classifiers - *optional*

A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier object that matches that data format.

Custom classifiers | Info

Select one or more classifiers to use with this crawler.

Choose one or more classifiers ▼ ⟳

cust_classifier ×

Clear selection | Add new classifier

Configure security settings

IAM role Info

Existing IAM role

TrustedEntityType ▼ ⟳ View

Create new IAM role | Update chosen IAM role

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

One crawler successfully created
The following crawler is now created: "product-crawl"

product-crawl

Crawler properties

Name
product-crawl

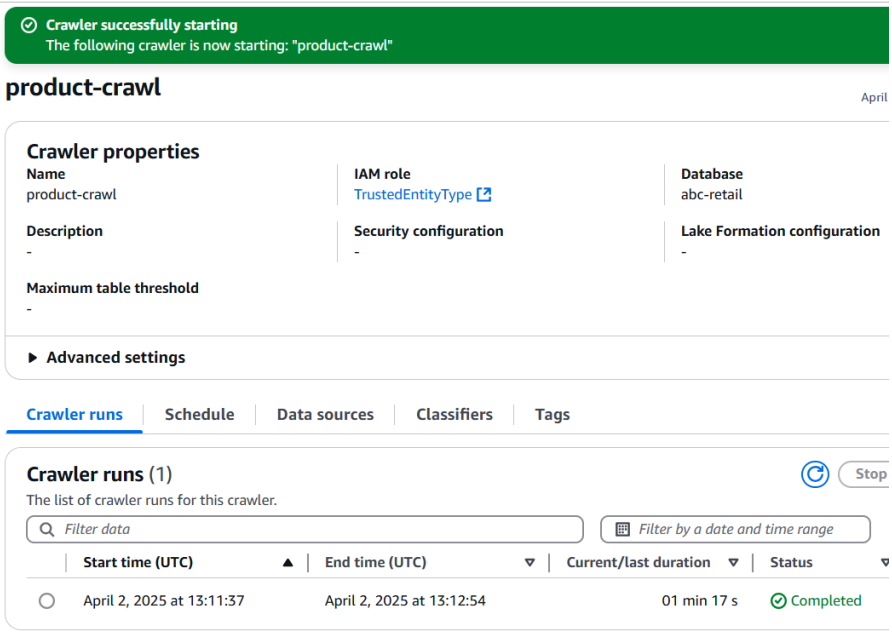
IAM role
TrustedEntityType

Description
-

Security configuration
-

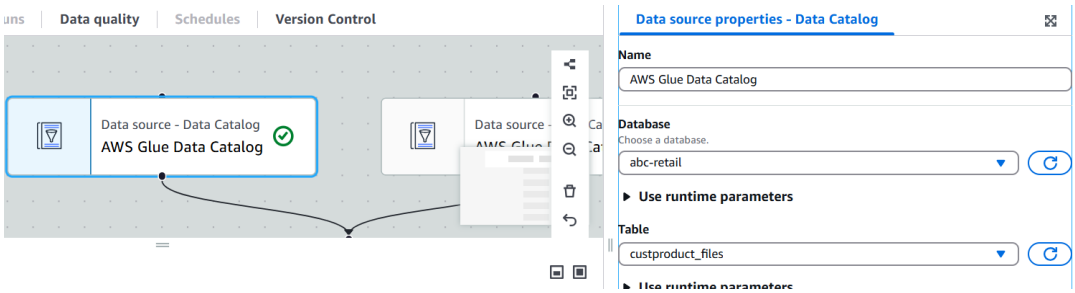
Maximum table threshold
-

5. Product-crawl successfully created and ran to extract metadata in the database abc-retail

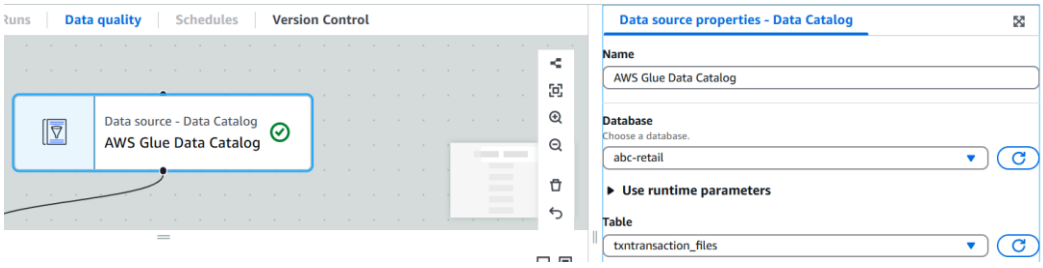


Step-7

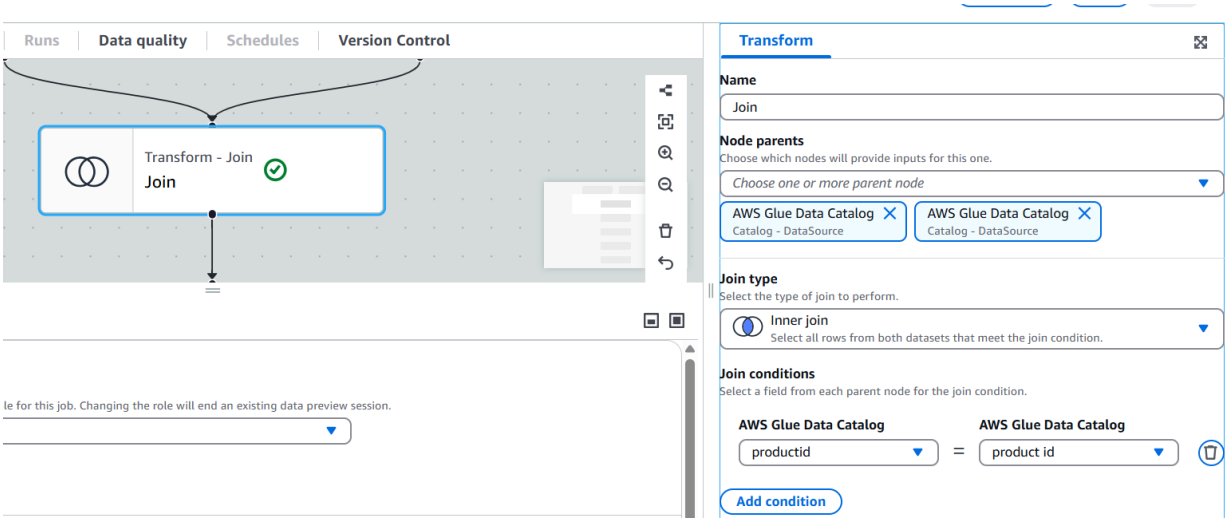
1. Creation of a job in AWS GLUE .
2. Importing AWS Glue Data Catalog from data sources and setting up data catalog 1 where
    - table I choose custproduct_files
    - database as abc-retail
    -



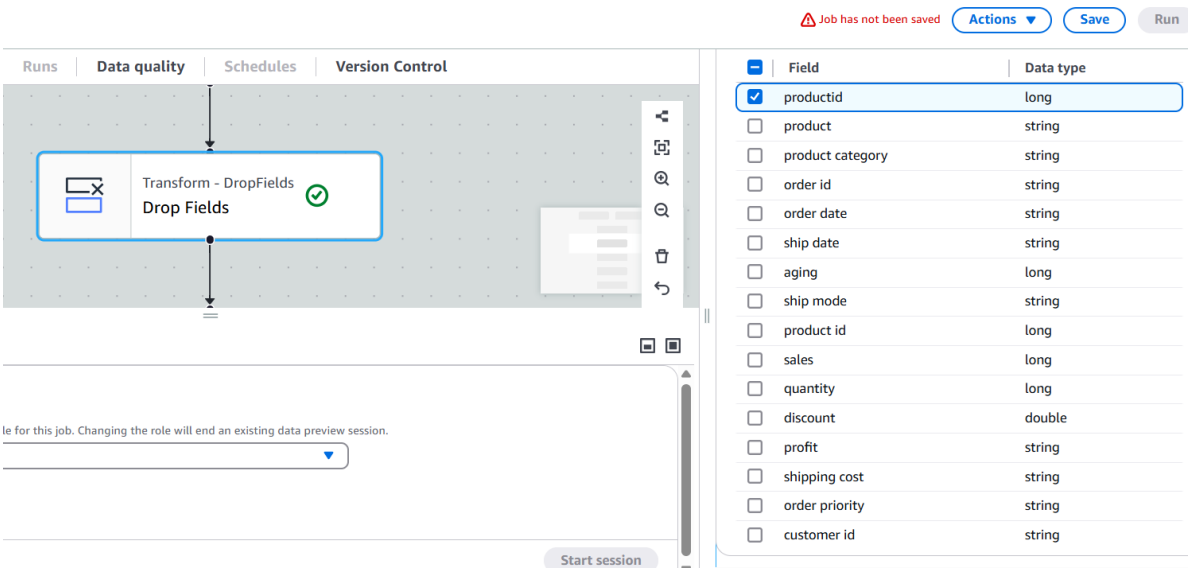
3. Importing AWS Glue Data Catalog from data sources and setting up data catalog 2 where in
    - table I choose txntransaction_files
    - and database as abc-retail



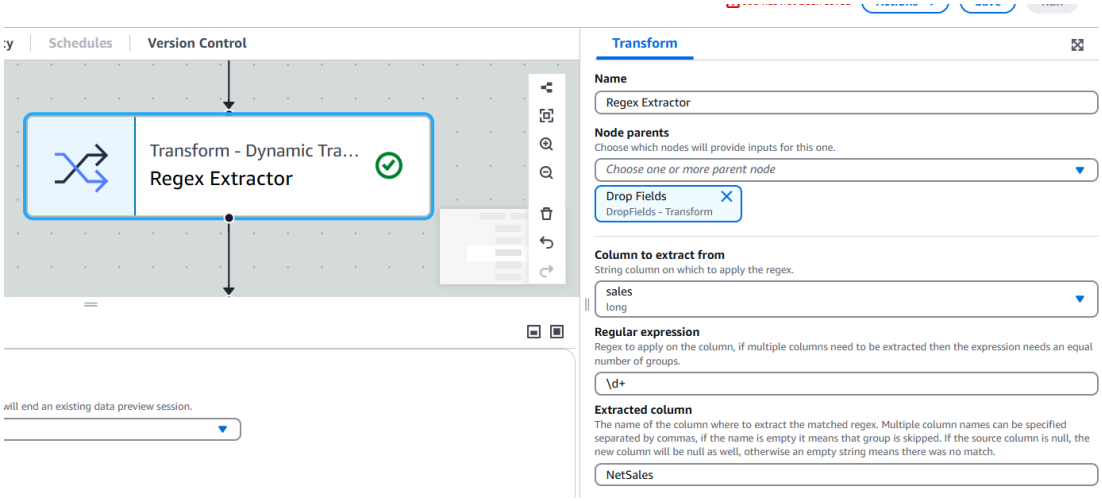
4. Importing & setting up Join from Transforms linking node parents as AWS Glue Data Catalog-1 and AWS Glue Data Catalog-1
    - Join type as inner join
    - join conditions as productid = product id



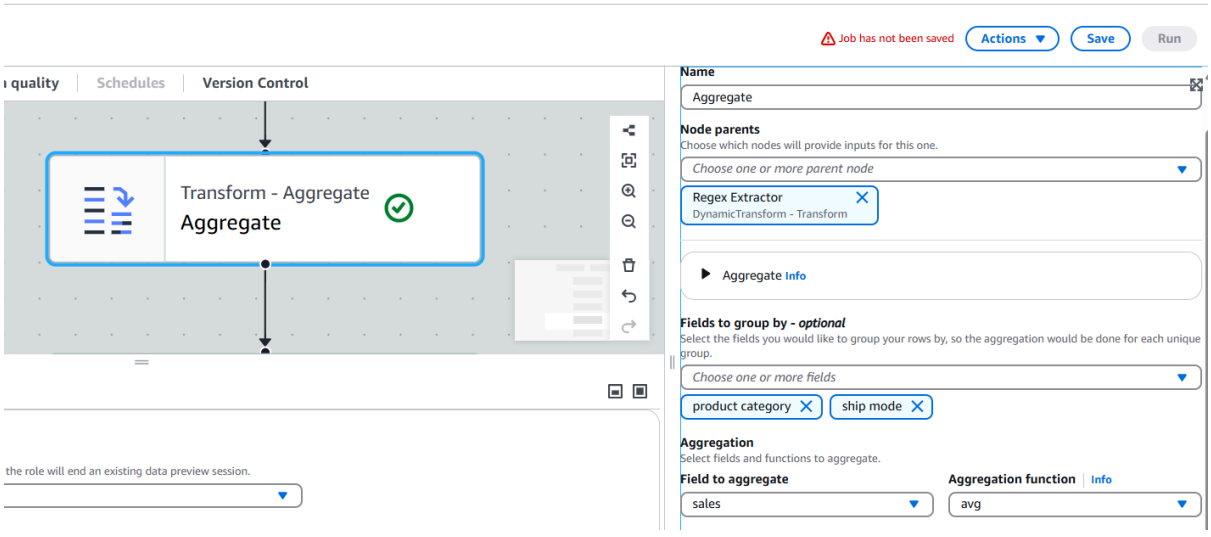
5. Importing & setting up drop fields from Transform Node parent as Join
    - selecting productid.



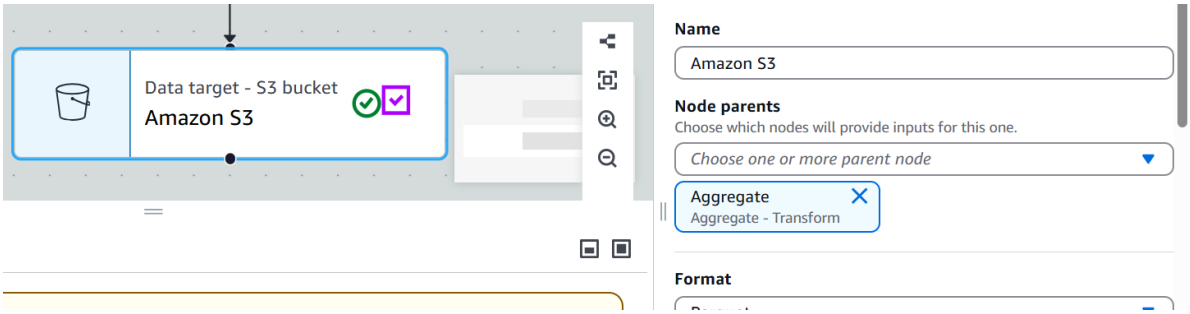
6. Importing & setting up Regex Extractor from transform- choosing node parents as Drop Fields.
    - And Column to extract from as sales
    - Regular expression as \d+
    - Extracted column as NetSales



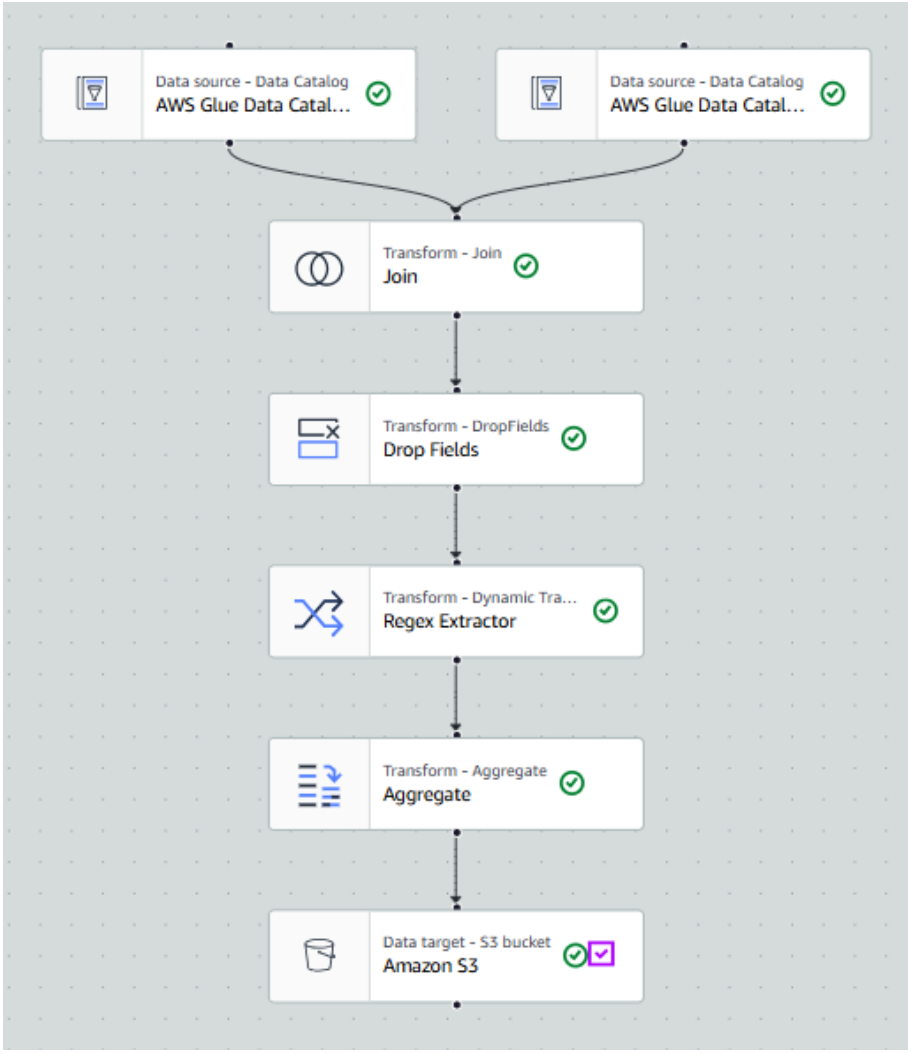
7. Importing & setting up aggregate from transform choosing node as Regex extractor
    - Fields to group by as product category and ship mode
    - In aggregation
    - Field to aggregate as sales and aggregation function as avg



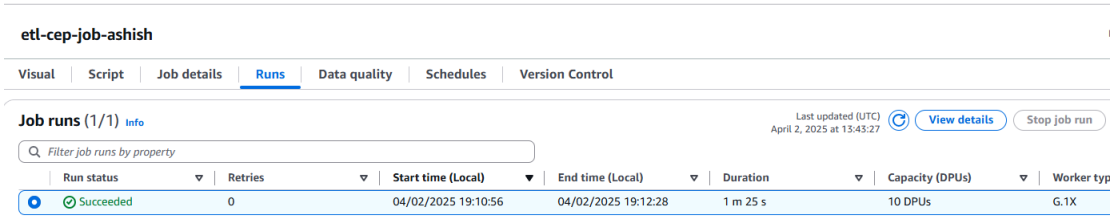
8. Importing & setting up amazon s3 node parents as aggregate
   Location as s3:// etl-cep-output-00001-ashish-chamel



9. Final outlook of model of a etl job

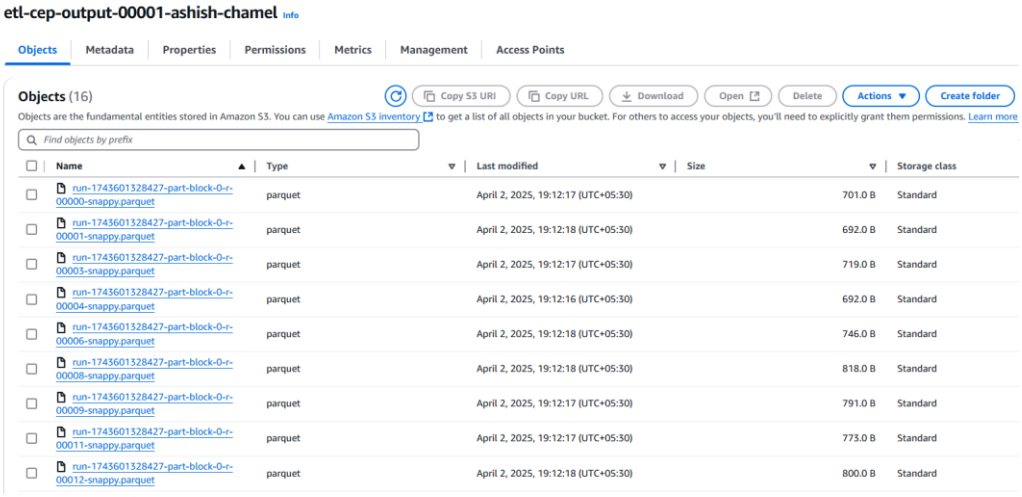


10. Etl-cep-job-ashish (ETL) job ran successfully

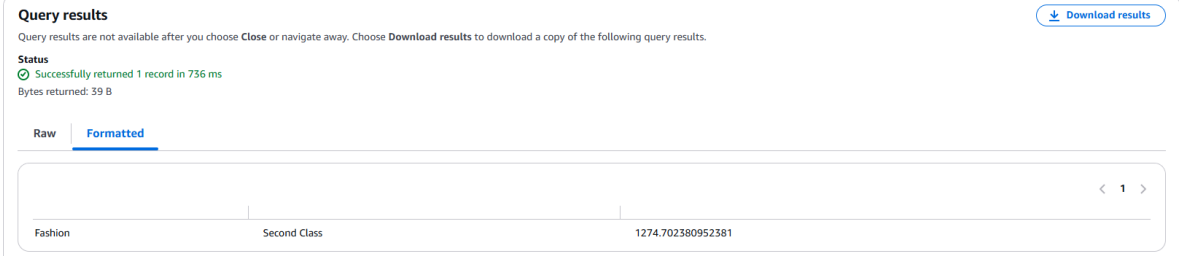


Step-8

1. This is the output after successful completion of ETL job in our output bucket ie etl-cep-output-00001-ashish-chamel



2. Running the sql query to see the results



Product Category	Ship Mode	Average Sales
Fashion	Second Class	1274.702380952381