

# DWDM LAB PROJECT

# ETL using AWS Glue and Redshift

Dataset Used : AMAZON PRODUCT SALES

# Team Members

SERIAL NO.	ROLL NO.	NAME
01.	01	Aboli Patne
02.	08	Shamika Aney
03.	23	Akshay Padia
04.	34	Hariom Nabira

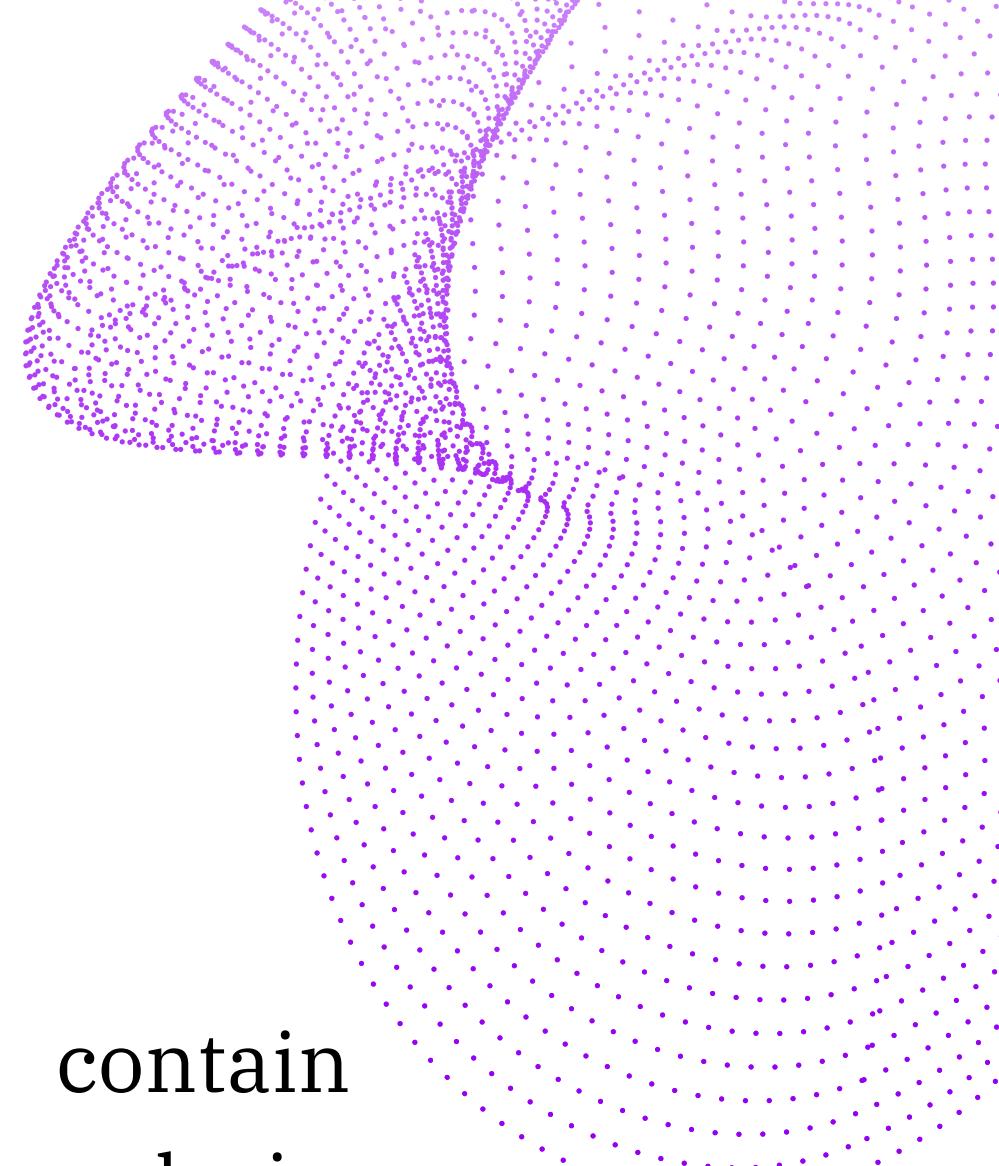
Supervised by :-  
**Prof. Arti Karandikar**

# **Problem Definition**

To efficiently process and clean raw e-commerce product data from Amazon using the ETL pipeline on AWS for improved analytics.

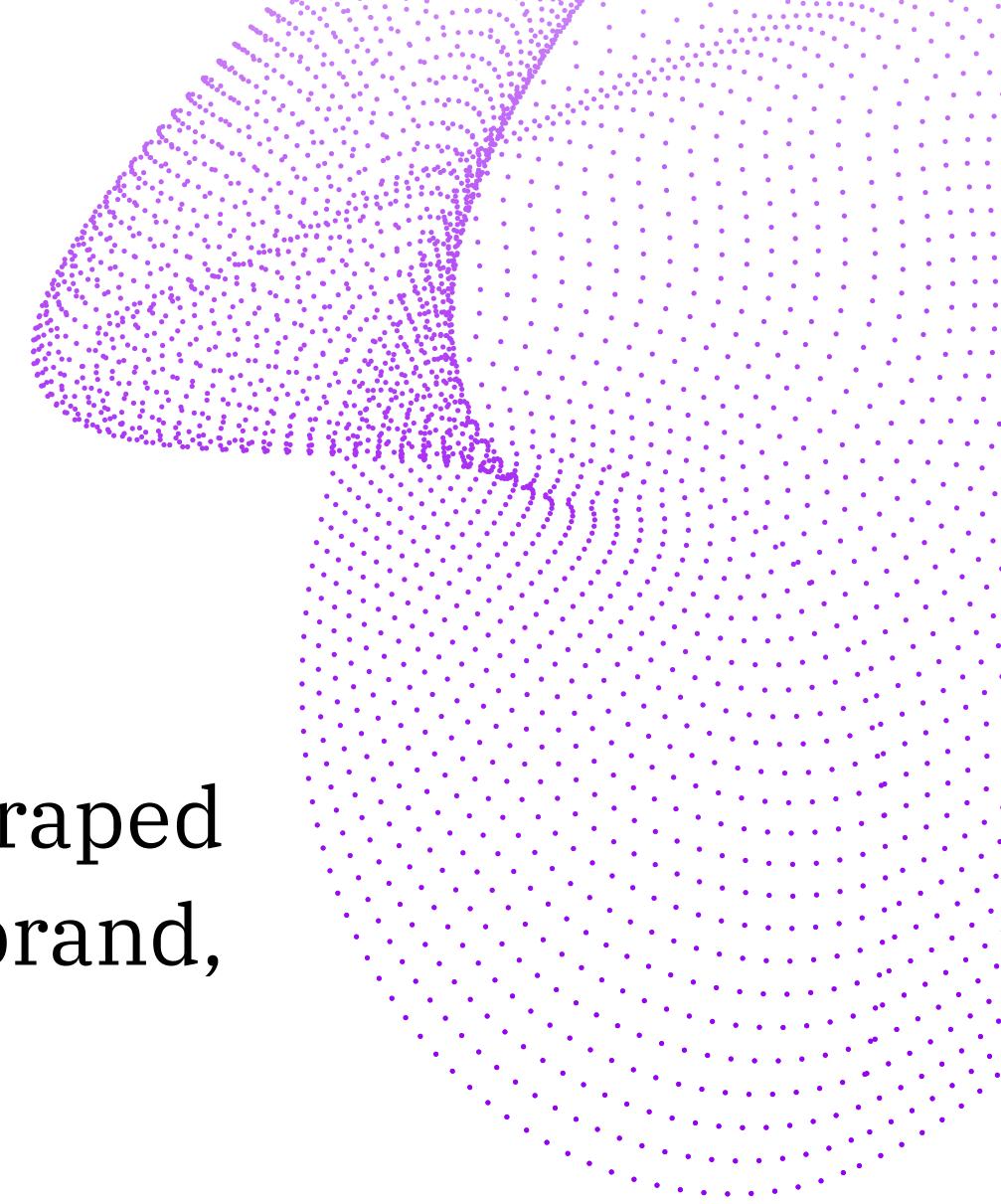
## **Introduction:**

In the age of data-driven decision-making, raw datasets often contain inconsistencies, null values, and non-standard formats that hinder analysis. To address this, we leverage AWS's powerful ETL services specifically AWS Glue and Glue Crawler to extract Amazon product listings, transform them through a series of custom cleaning and formatting rules, and load the structured data into a usable format. This ensures high-quality, standardized product data that supports accurate insights and efficient querying for business intelligence applications.

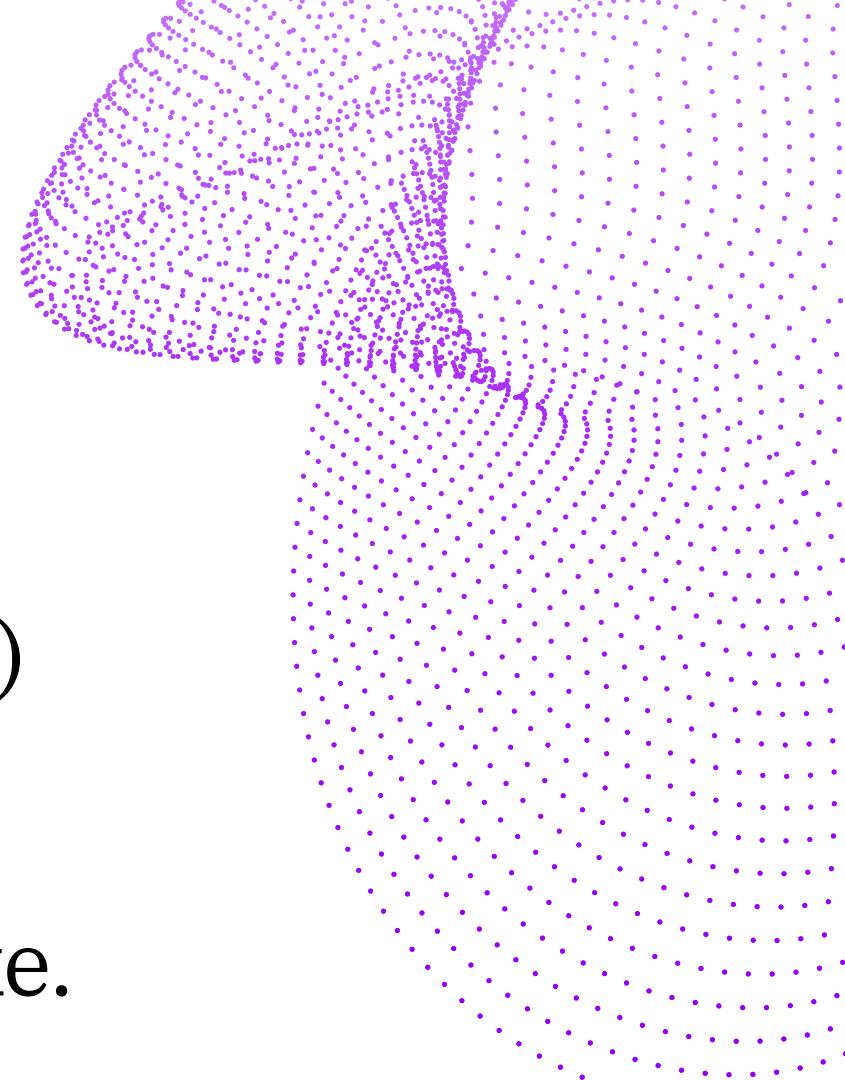


# Dataset

- **Dataset Name:** Amazon Products Dataset
- **Size:** 1000 records × 24 attributes
- **Purpose:** This dataset contains structured product listings scraped from Amazon, including key fields such as product title, brand, price, reviews, ratings, categories, and availability.
- **Key Features:**
  - Product metadata (title, brand, description)
  - Pricing (final\_price, currency)
  - Performance metrics (reviews\_count, rating, answered\_questions)
  - Media info (images\_count, video\_count, image\_url)
  - Additional attributes like asin, seller\_id, manufacturer, availability



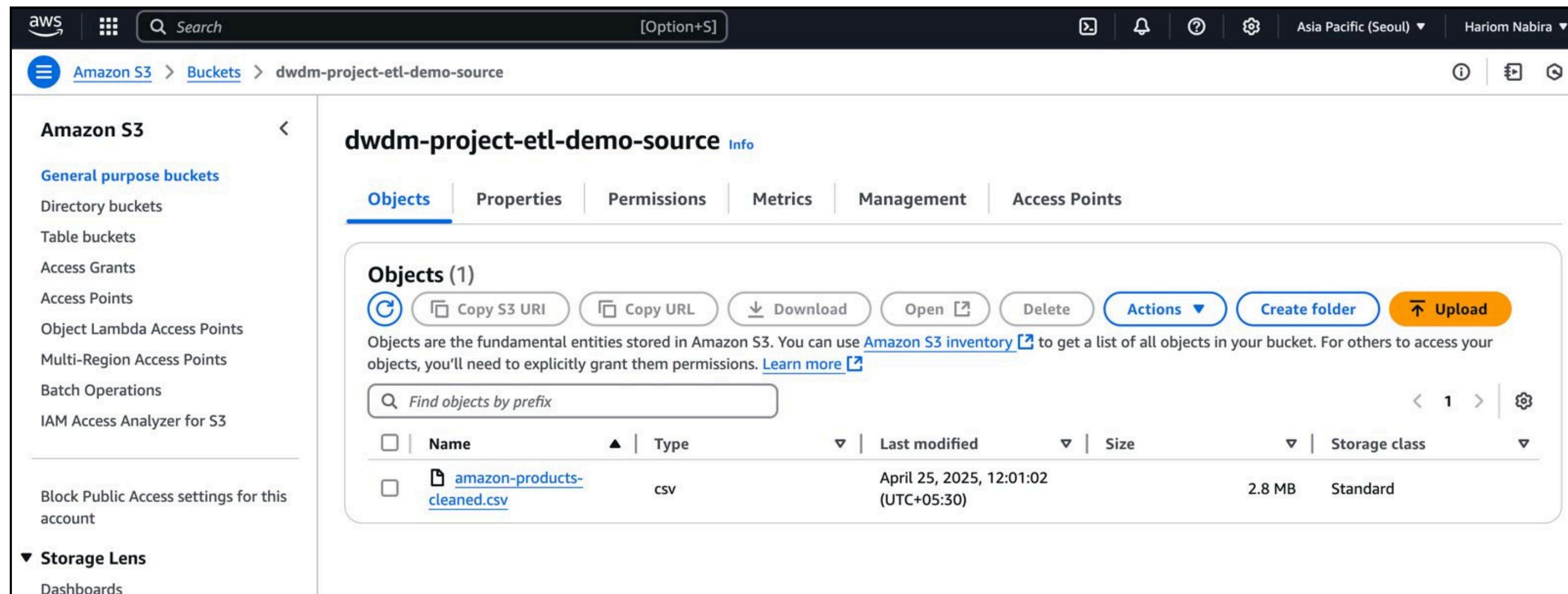
# Extraction



In the extraction phase of the ETL (Extract, Transform, Load) process, Amazon product sales data was first uploaded to an **Amazon S3 bucket**, which served as the centralized data lake. Leveraging **AWS Glue**, a metadata crawler was configured to automatically scan the S3 location and detect the structure of the data. The **crawler** parsed the CSV file to infer the schema identifying data types, column names, and formatting and stored this metadata in the **AWS Glue Data Catalog**.

# Extraction

## Amazon S3 Bucket



The screenshot shows the AWS S3 console interface. The top navigation bar includes the AWS logo, a search bar, and user information for Hariom Nabira. The left sidebar has a 'General purpose buckets' section with options like Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. It also includes a 'Block Public Access settings for this account' link and sections for Storage Lens and Dashboards. The main content area shows a bucket named 'dwdm-project-etl-demo-source'. The 'Objects' tab is selected, displaying one object: 'amazon-products-cleaned.csv' (Type: csv). The object was last modified on April 25, 2025, at 12:01:02 (UTC+05:30), has a size of 2.8 MB, and is stored in the Standard storage class. Action buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload are available above the object list. A search bar at the bottom allows finding objects by prefix.

Name	Type	Last modified	Size	Storage class
amazon-products-cleaned.csv	csv	April 25, 2025, 12:01:02 (UTC+05:30)	2.8 MB	Standard

# Extraction Crawler

The screenshot shows the AWS Glue Crawler properties page for 'dwdm-project-crawler'. The crawler is currently in a READY state. It has an IAM role named 'AWSGlueServiceRole-dwdm-project' and is connected to a database named 'dwdm\_project\_database'. The crawler has run 7 times, all of which are completed. The most recent run was on April 25, 2025, at 13:22:03, taking 1 min 01 s and costing 0.060 DPU hours. The crawler runs are listed in descending order of start time.

Start time (UTC)	End time (UTC)	Duration	Status	DPU hours	Table changes
April 25, 2025 at 13:22:03	April 25, 2025 at 13:23:04	01 min 01 s	Completed	0.060	1 table change, 0 partition changes
April 25, 2025 at 06:32:04	April 25, 2025 at 06:33:15	01 min 11 s	Completed	0.064	3 table changes, 0 partition changes
April 25, 2025 at 05:56:45	April 25, 2025 at 05:57:41	56 s	Completed	0.074	4 table changes, 0 partition changes
April 25, 2025 at 05:50:21	April 25, 2025 at 05:51:14	52 s	Completed	0.082	1 table change, 0 partition changes
April 25, 2025 at 05:28:51	April 25, 2025 at 05:29:47	56 s	Completed	0.050	-
April 25, 2025 at 05:20:13	April 25, 2025 at 05:21:03	50 s	Completed	0.051	1 table change, 0 partition changes
April 25, 2025 at 05:03:22	April 25, 2025 at 05:04:27	01 min 05 s	Completed	0.071	1 table change, 0 partition changes

# Extraction Schema

The screenshot shows the AWS Glue Schema view for a table named "dwdm\_project\_etl\_demo\_source". The left sidebar contains navigation links for AWS Glue, Data Catalog, Data Integration and ETL, and Legacy pages. The main area displays the table schema with 24 columns:

#	Column name	Data type	Partition key	Comment
1	timestamp	string	-	-
2	title	string	-	-
3	brand	string	-	-
4	description	string	-	-
5	final_price	string	-	-
6	currency	string	-	-
7	availability	string	-	-
8	reviews_count	bigint	-	-
9	categories	array	-	-
10	asin	string	-	-
11	number_of_sellers	double	-	-
12	root_bs_rank	double	-	-
13	answered_questions	double	-	-
14	domain	string	-	-
15	images_count	bigint	-	-
16	url	string	-	-
17	video_count	bigint	-	-
18	image_url	string	-	-
19	rating	double	-	-
20	seller_id	string	-	-

# Transformation

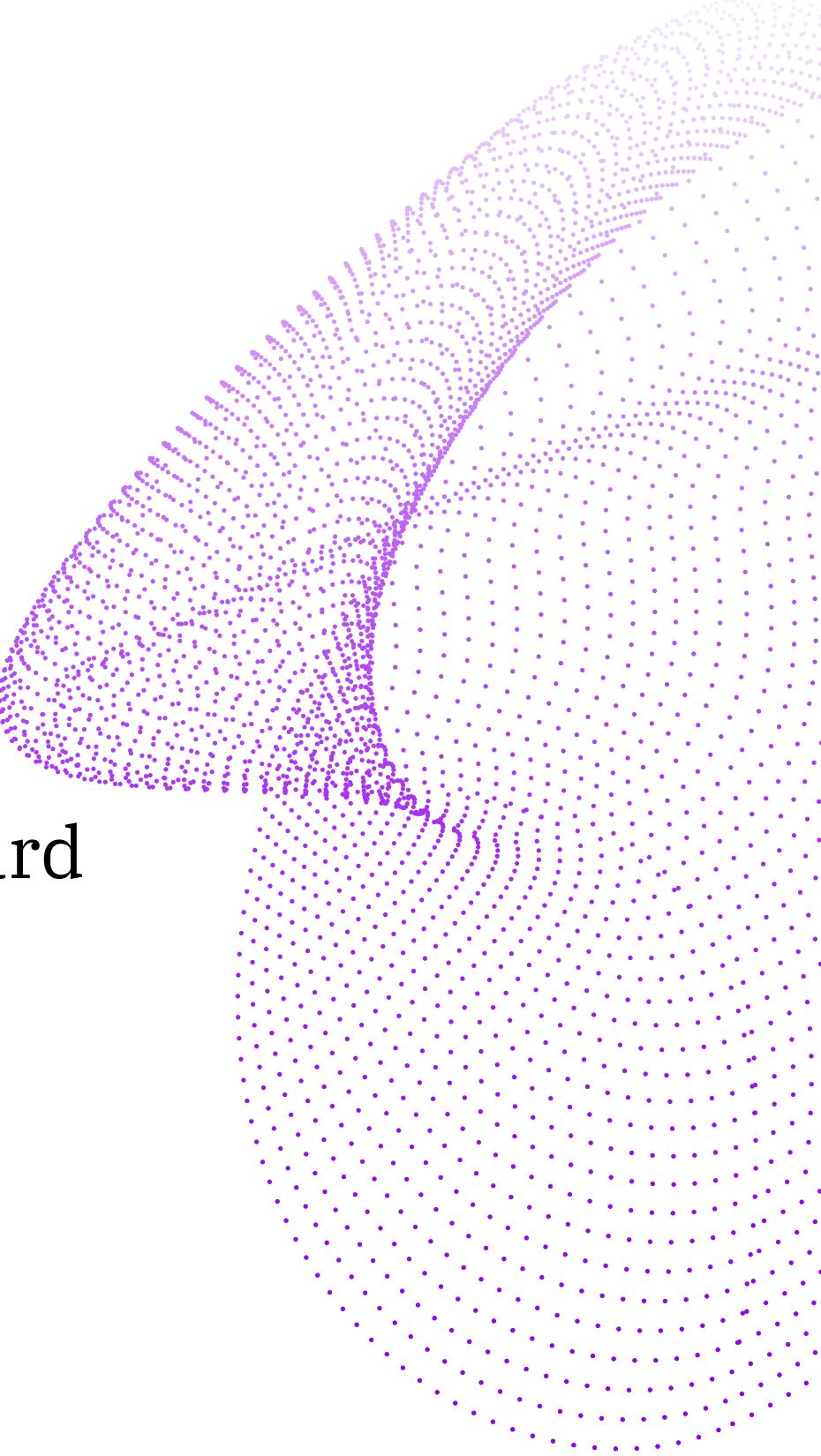
- ETL Job was developed using AWS Glue, where transformation logic was scripted within the Glue job itself.
- Source data was cataloged through an AWS Glue crawler, which automatically classified the schema and made it queryable.
- Custom transformations applied during the process include:
- Availability
  - Values containing the word “left” (e.g., “Only 3 left”) were replaced with "Limited".
  - All variations of “in stock” were standardized to "In Stock".
- Number of Sellers
  - If missing, the value was set to 1.
  - If already present, the original value was retained.

# Transformation

- Domain
  - Converted shortened domain names (e.g., www.amazon.com) to full URLs like "https://www.amazon.com/".
  - If missing, set to default "https://amazon.com/".
- Manufacturer
  - Replaced null or missing values with "unknown".
- Plus Content
  - If the field was null, it was filled with False.
- Top Review
  - If no review was present, replaced with "no review".

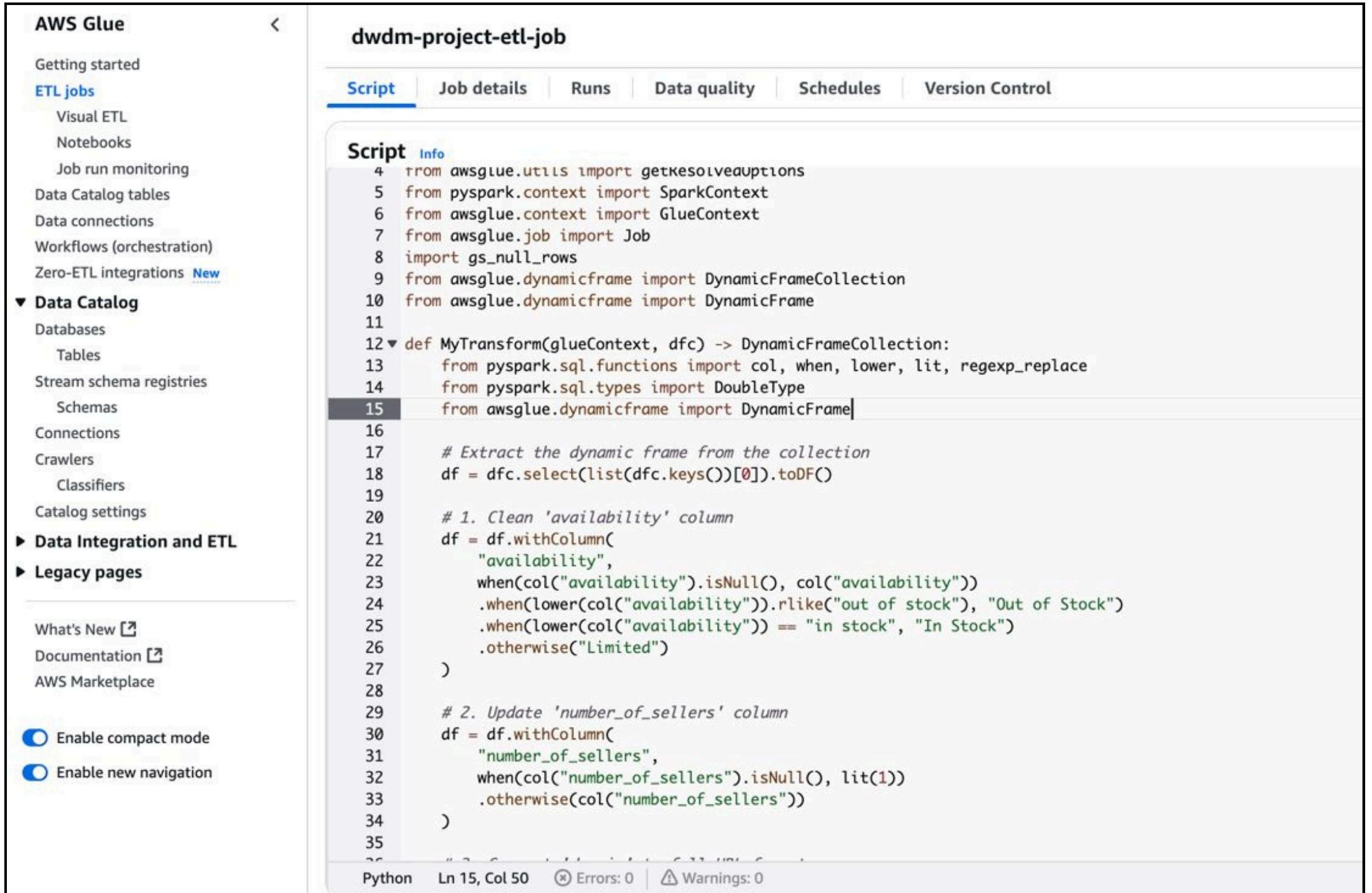
# Transformation

- Null Removal
  - Entire rows or columns with all null values were dropped
- Data Type Conversion
  - Values in string or scientific notation were converted to standard double format.



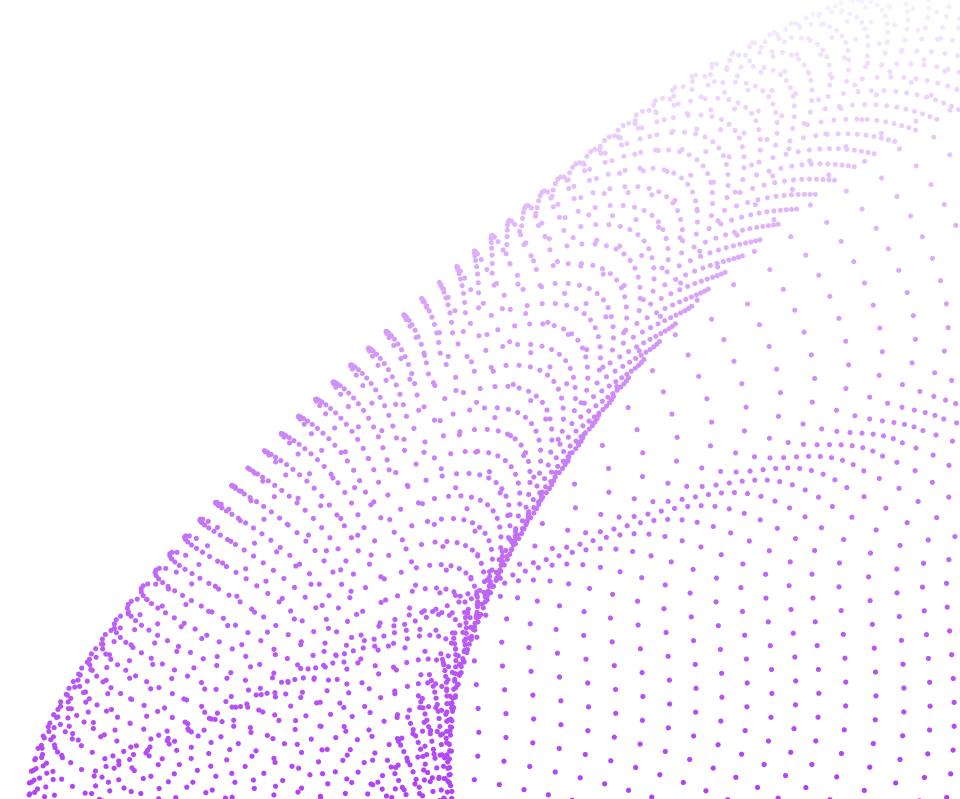
# Transformation

## Transformation script



# Transformation

## Transformation destination S3 bucket



Screenshot of the Amazon S3 console showing the contents of the bucket "dwdm-project-etl-demo-destination".

The left sidebar shows navigation options for Amazon S3, including General purpose buckets, Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, and Storage Lens.

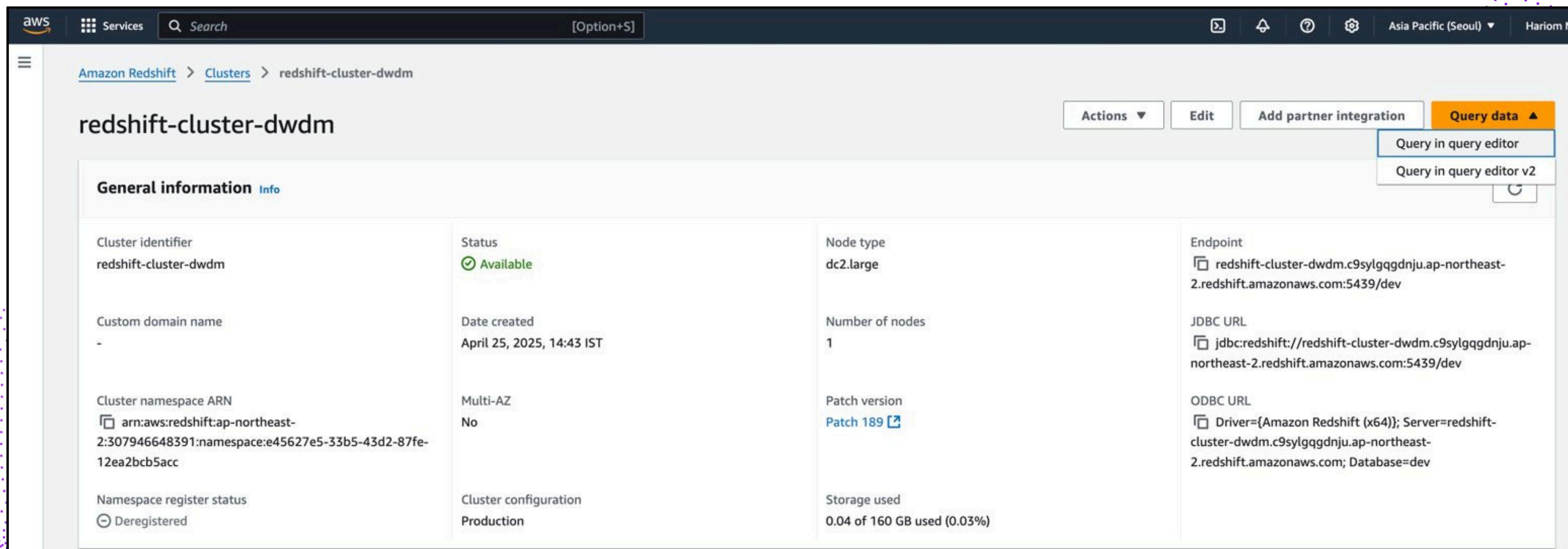
The main content area displays the "Objects" tab for the "dwdm-project-etl-demo-destination" bucket. It shows one object named "run-1745587561625-part-block-0-r-00000-snappy.parquet". The object is a parquet file, last modified on April 25, 2025, at 18:56:06 (UTC+05:30), with a size of 1.5 MB and a storage class of Standard.

Name	Type	Last modified	Size	Storage class
run-1745587561625-part-block-0-r-00000-snappy.parquet	parquet	April 25, 2025, 18:56:06 (UTC+05:30)	1.5 MB	Standard

# Load

In the final Load phase of the ETL pipeline, the transformed data – now stored in the efficient Parquet file format – was loaded into **Amazon Redshift**, a fully managed, petabyte-scale data warehouse solution. A custom **schema was predefined** in Redshift to match the structure of the transformed dataset. This ensured proper column mapping and optimized query performance. The data was loaded from the Parquet files directly stored in **Amazon S3**, with the bucket path specified as the data source during the Redshift COPY operation.

# Load Amazon Redshift Cluster



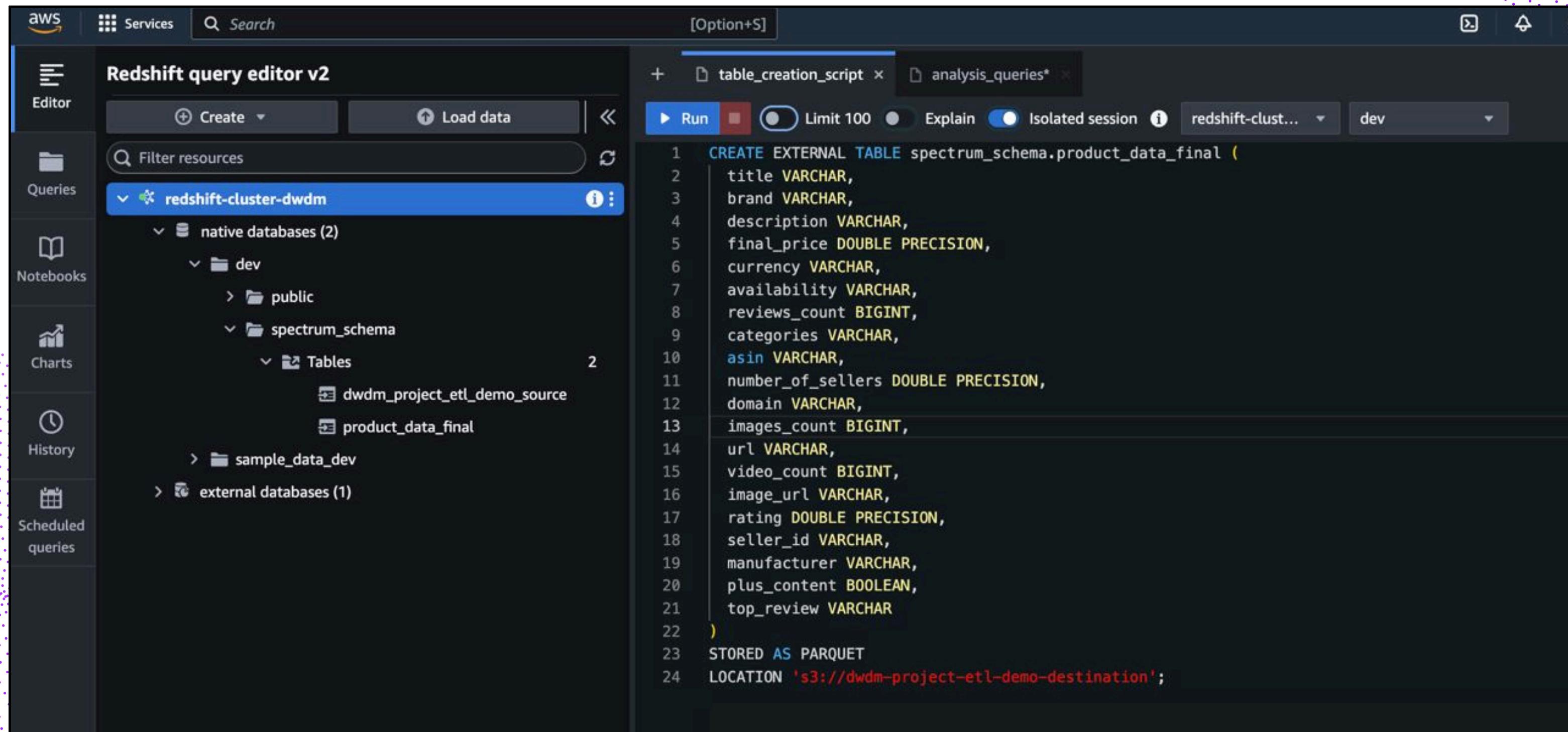
The screenshot shows the AWS Amazon Redshift console with the cluster details for "redshift-cluster-dwdm".

**General information**

Cluster identifier	Status	Node type	Endpoint
redshift-cluster-dwdm	Available	dc2.large	<a href="#">redshift-cluster-dwdm.c9sylgqgdnu.ap-northeast-2.redshift.amazonaws.com:5439/dev</a>
Custom domain name	Date created	Number of nodes	JDBC URL
-	April 25, 2025, 14:43 IST	1	<a href="#">jdbc:redshift://redshift-cluster-dwdm.c9sylgqgdnu.ap-northeast-2.redshift.amazonaws.com:5439/dev</a>
Cluster namespace ARN	Multi-AZ	Patch version	ODBC URL
<a href="#">arn:aws:redshift:ap-northeast-2:307946648391:namespace:e45627e5-33b5-43d2-87fe-12ea2bcb5acc</a>	No	<a href="#">Patch 189</a>	<a href="#">Driver={Amazon Redshift (x64)}; Server=redshift-cluster-dwdm.c9sylgqgdnu.ap-northeast-2.redshift.amazonaws.com; Database=dev</a>
Namespace register status	Cluster configuration	Storage used	
Deregistered	Production	0.04 of 160 GB used (0.03%)	

# Load

# Table Creation Script



The screenshot shows the AWS Redshift Query Editor v2 interface. The left sidebar contains navigation links for Editor, Queries, Notebooks, Charts, History, and Scheduled queries. The main area displays a query editor window titled "Redshift query editor v2". The top bar includes a "Services" menu, a search bar, and a tab labeled "[Option+S]". Below the tabs are buttons for "Run", "Limit 100", "Explain", "Isolated session", and a dropdown for "redshift-clust...". A "dev" tab is selected. The left pane shows a tree view of database structures under "redshift-cluster-dwdm": native databases (2) containing dev (public, spectrum\_schema with Tables: dwdm\_project\_etl\_demo\_source, product\_data\_final) and sample\_data\_dev; and external databases (1). The right pane contains the following SQL code:

```
1 CREATE EXTERNAL TABLE spectrum_schema.product_data_final (
2     title VARCHAR,
3     brand VARCHAR,
4     description VARCHAR,
5     final_price DOUBLE PRECISION,
6     currency VARCHAR,
7     availability VARCHAR,
8     reviews_count BIGINT,
9     categories VARCHAR,
10    asin VARCHAR,
11    number_of_sellers DOUBLE PRECISION,
12    domain VARCHAR,
13    images_count BIGINT,
14    url VARCHAR,
15    video_count BIGINT,
16    image_url VARCHAR,
17    rating DOUBLE PRECISION,
18    seller_id VARCHAR,
19    manufacturer VARCHAR,
20    plus_content BOOLEAN,
21    top_review VARCHAR
22 )
23 STORED AS PARQUET
24 LOCATION 's3://dwdm-project-etl-demo-destination';
```

# SQL Queries Doc Link

[click here](#)

# **THANKYOU**