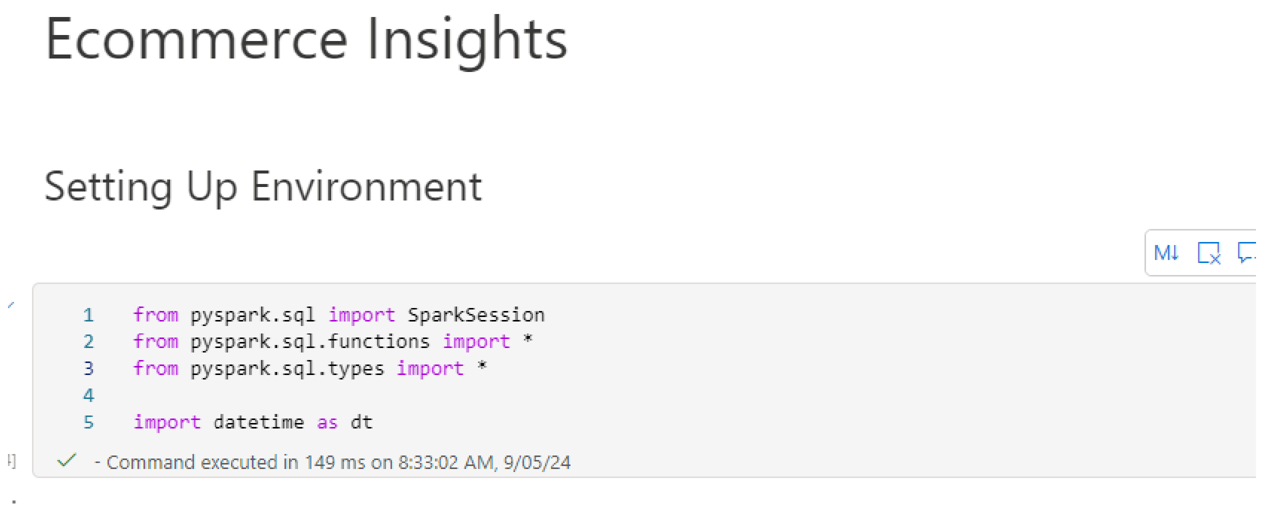
Week 1: Overview and basic configurations

Step 1: Choose a suitable cloud provider and set up a Spark shell environment

Step 2: Configure the necessary dependencies

Step 3: Execute basic Spark commands to make sure Spark is ready



Ecommerce Insights

Setting Up Environment

```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *

import datetime as dt
```

- Command executed in 149 ms on 8:33:02 AM, 9/05/24

Step 4: Use README.md for details, instructions, and commands

---

Week 2: Data ingestion

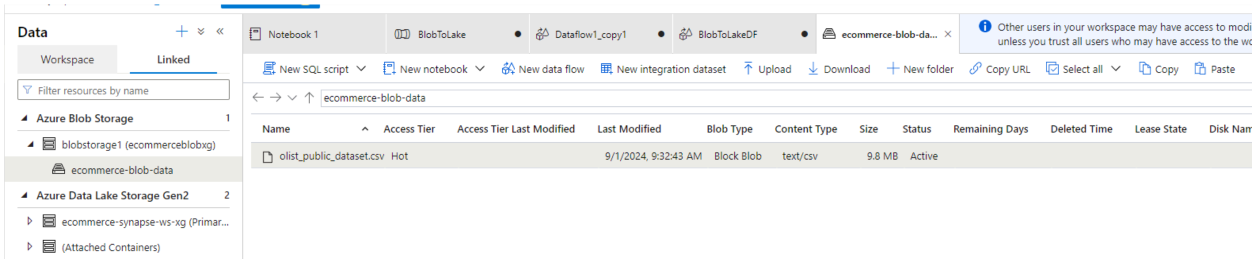
Step 1: Upload the entire data into Hive from CSV using cloud provider cluster setup

1. Log in to PuTTY with the username “hadoop”
2. Enter the command given below:

Command: hive

3. Create a database
4. Create a table with all the relevant details

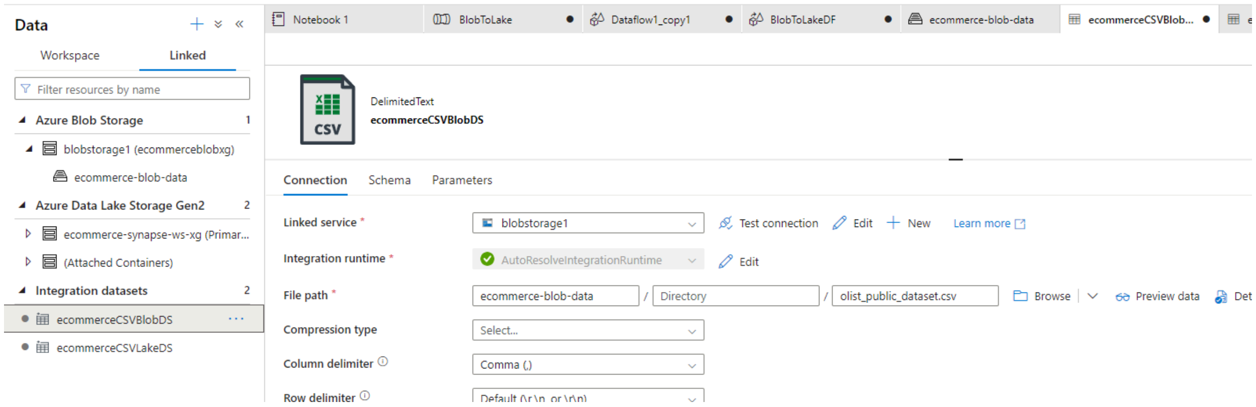
Step 2: Create a bucket (Azure Blob) and upload the csv file



Other users in your workspace may have access to modi
unless you trust all users who may have access to the wo

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days	Deleted Time	Lease State	Disk Nam
olist_public_dataset.csv	Hot		9/1/2024, 9:32:43 AM	Block Blob	text/csv	9.8 MB	Active				

Step 4: Create a new directory in HDFS(Data Lake) and copy the data from Hive into HDFS


Data

Workspace | Linked

Filter resources by name

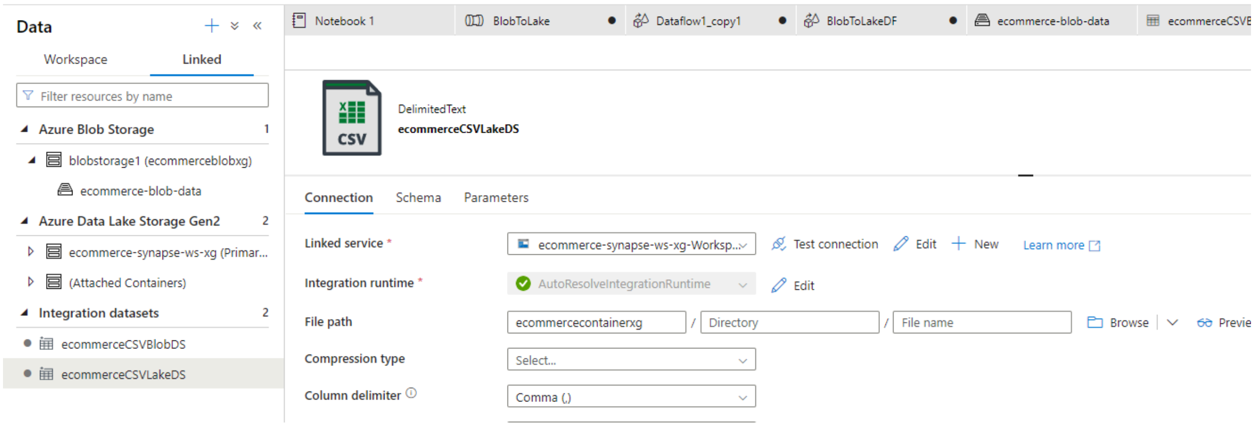
- Azure Blob Storage 1
  - blobstorage1 (ecommerceblobxg)
    - ecommerce-blob-data
- Azure Data Lake Storage Gen2 2
  - ecommerce-synapse-ws-xg (Primar...
  - (Attached Containers)
- Integration datasets 2
  - ecommerceCSVBlobDS
  - ecommerceCSVLakeDS

Notebook 1 | BlobToLake | Dataflow1_copy1 | BlobToLakeDF | ecommerce-blob-data | ecommerceCSVBlob...

DelimitedText
ecommerceCSVBlobDS

Connection | Schema | Parameters

Linked service *	blobstorage1
Integration runtime *	AutoResolveIntegrationRuntime
File path *	ecommerce-blob-data / Directory / olist_public_dataset.csv
Compression type	Select...
Column delimiter	Comma (,)
Row delimiter	Default (\r,\n, or \r\n)


Data

Workspace | Linked

Filter resources by name

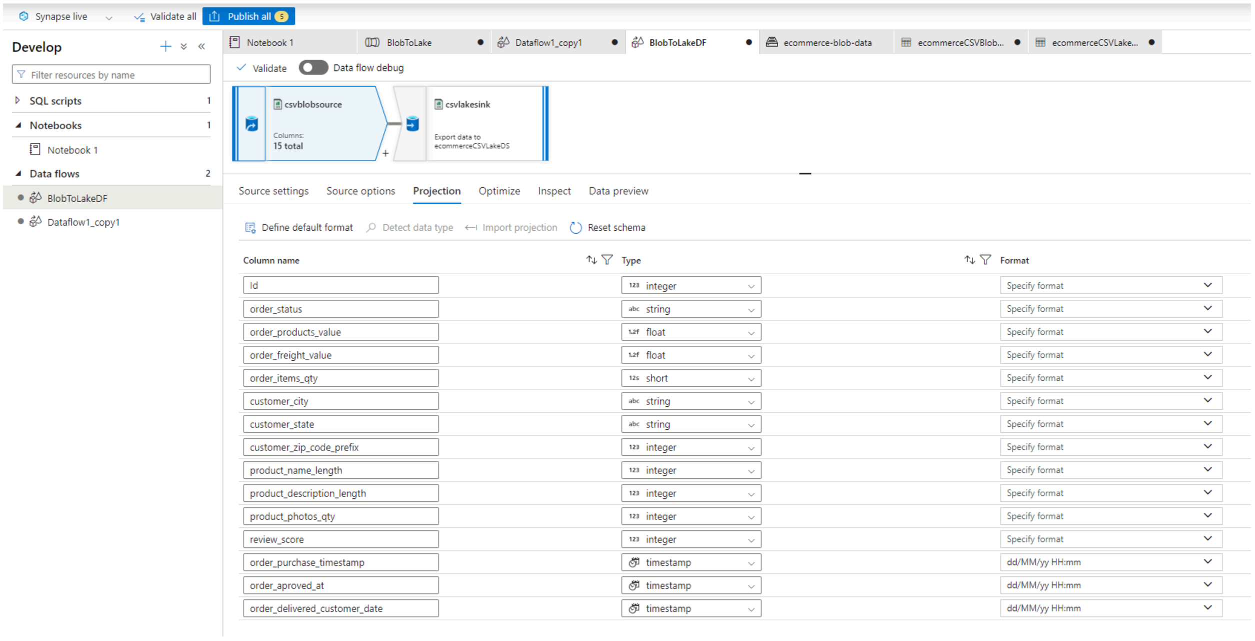
- Azure Blob Storage 1
  - blobstorage1 (ecommerceblobxg)
    - ecommerce-blob-data
- Azure Data Lake Storage Gen2 2
  - ecommerce-synapse-ws-xg (Primar...
  - (Attached Containers)
- Integration datasets 2
  - ecommerceCSVBlobDS
  - ecommerceCSVLakeDS

Notebook 1 | BlobToLake | Dataflow1_copy1 | BlobToLakeDF | ecommerce-blob-data | ecommerceCSVE

DelimitedText
ecommerceCSVLakeDS

Connection | Schema | Parameters

Linked service *	ecommerce-synapse-ws-xg-Worksp...
Integration runtime *	AutoResolveIntegrationRuntime
File path	ecommercecontainerxg / Directory / File name
Compression type	Select...
Column delimiter	Comma (,)



Develop

Filter resources by name

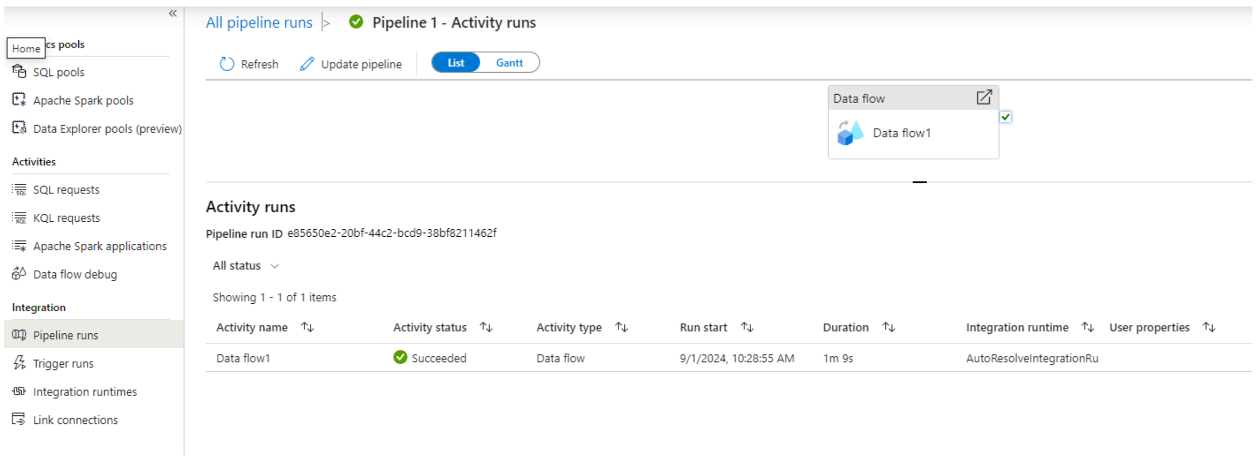
- SQL scripts 1
- Notebooks 1
  - Notebook 1
- Data flows 2
  - BlobToLakeDF
  - Dataflow1_copy1

Notebook 1 | BlobToLake | Dataflow1_copy1 | BlobToLakeDF | ecommerce-blob-data | ecommerceCSVBlob... | ecommerceCSVLake...

Source settings | Source options | Projection | Optimize | Inspect | Data preview

Define default format | Detect data type | Import projection | Reset schema

Column name	Type	Format
Id	integer	Specify format
order_status	string	Specify format
order_products_value	float	Specify format
order_freight_value	float	Specify format
order_items_qty	short	Specify format
customer_city	string	Specify format
customer_state	string	Specify format
customer_zip_code_prefix	integer	Specify format
product_name_length	integer	Specify format
product_description_length	integer	Specify format
product_photos_qty	integer	Specify format
review_score	integer	Specify format
order_purchase_timestamp	timestamp	dd/MM/yy HH:mm
order_aproved_at	timestamp	dd/MM/yy HH:mm
order_delivered_customer_date	timestamp	dd/MM/yy HH:mm



Activity runs

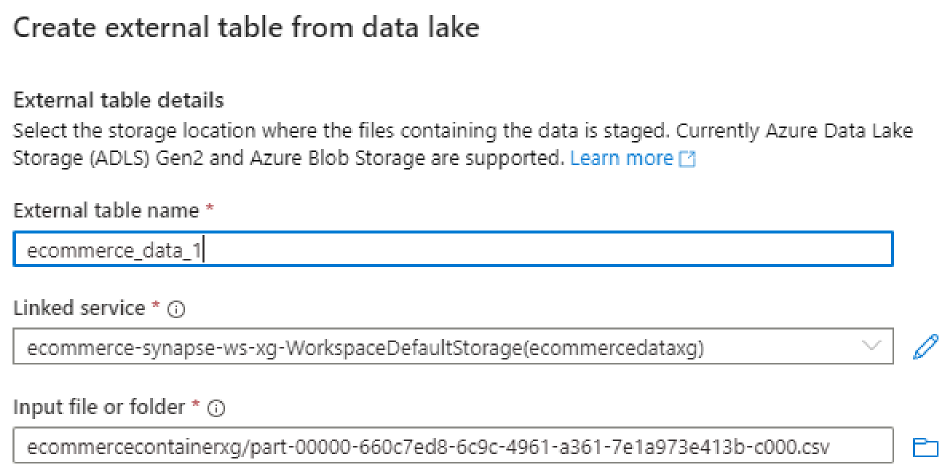
Pipeline run ID e85650e2-20bf-44c2-bcd9-38bf8211462f

All status

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties
Data flow1	✓ Succeeded	Data flow	9/1/2024, 10:28:55 AM	1m 9s	AutoResolveIntegrationRu	

Step 3: Load the data from the bucket into the Hive table(Lake Database)



Create external table from data lake

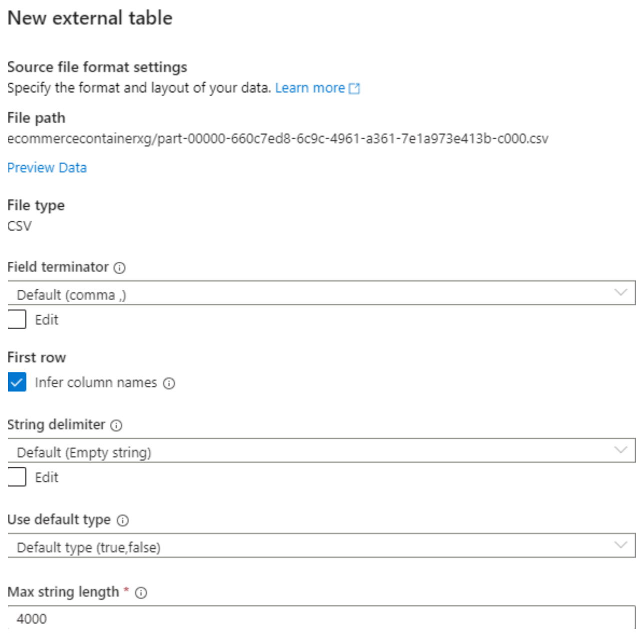
External table details

Select the storage location where the files containing the data is staged. Currently Azure Data Lake Storage (ADLS) Gen2 and Azure Blob Storage are supported. Learn more

External table name *
ecommerce_data_1

Linked service *
ecommerce-synapse-ws-xg-WorkspaceDefaultStorage(ecommercedataxg)

Input file or folder *
ecommercecontainerxg/part-00000-660c7ed8-6c9c-4961-a361-7e1a973e413b-c000.csv



New external table

Source file format settings

Specify the format and layout of your data. Learn more

File path
ecommercecontainerxg/part-00000-660c7ed8-6c9c-4961-a361-7e1a973e413b-c000.csv

Preview Data

File type
CSV

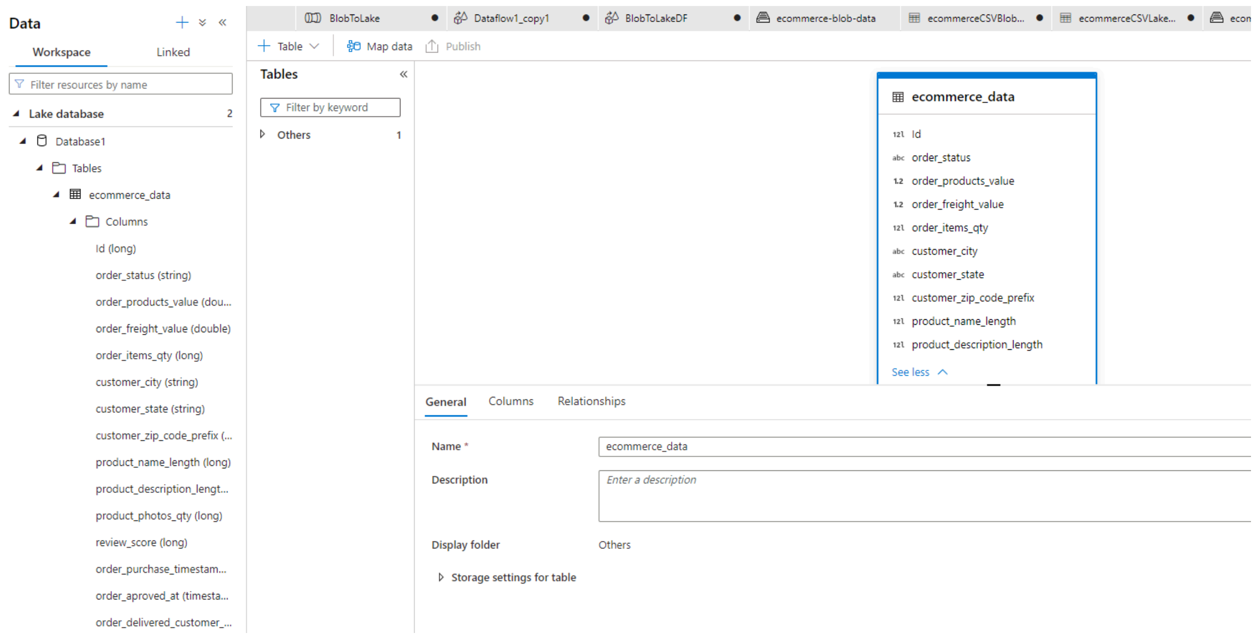
Field terminator
Default (comma ,)
☐ Edit

First row
☑ Infer column names

String delimiter
Default (Empty string)
☐ Edit

Use default type
Default type (true,false)

Max string length *
4000


Data

Workspace | Linked

- Lake database 2
  - Database1
    - Tables
      - ecommerce_data
        - Columns
          - Id (long)
          - order_status (string)
          - order_products_value (dou...
          - order_freight_value (double)
          - order_items_qty (long)
          - customer_city (string)
          - customer_state (string)
          - customer_zip_code_prefix (...
          - product_name_length (long)
          - product_description_lengt...
          - product_photos_qty (long)
          - review_score (long)
          - order_purchase_timestam...
          - order_aproved_at (timesta...
          - order_delivered_customer_...

Tables: Others 1

ecommerce_data
- Id
- order_status
- order_products_value
- order_freight_value
- order_items_qty
- customer_city
- customer_state
- customer_zip_code_prefix
- product_name_length
- product_description_length

General | Columns | Relationships

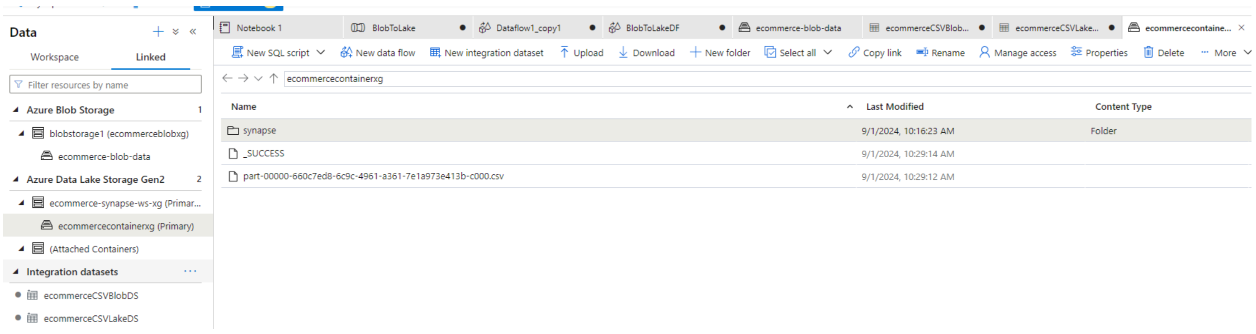
Name * ecommerce_data

Description: Enter a description

Display folder: Others

Storage settings for table

Step 5: Check if the data has been successfully loaded in the HDFS path


Data

Workspace | Linked

- Azure Blob Storage 1
  - blobstorage1 (ecommerceblobxg)
    - ecommerce-blob-data
- Azure Data Lake Storage Gen2 2
  - ecommerce-synapse-ws-xg (Primar...
    - ecommercecontainerxg (Primary)
  - (Attached Containers)
- Integration datasets
  - ecommerceCSVBlobDS
  - ecommerceCSVLakeDS

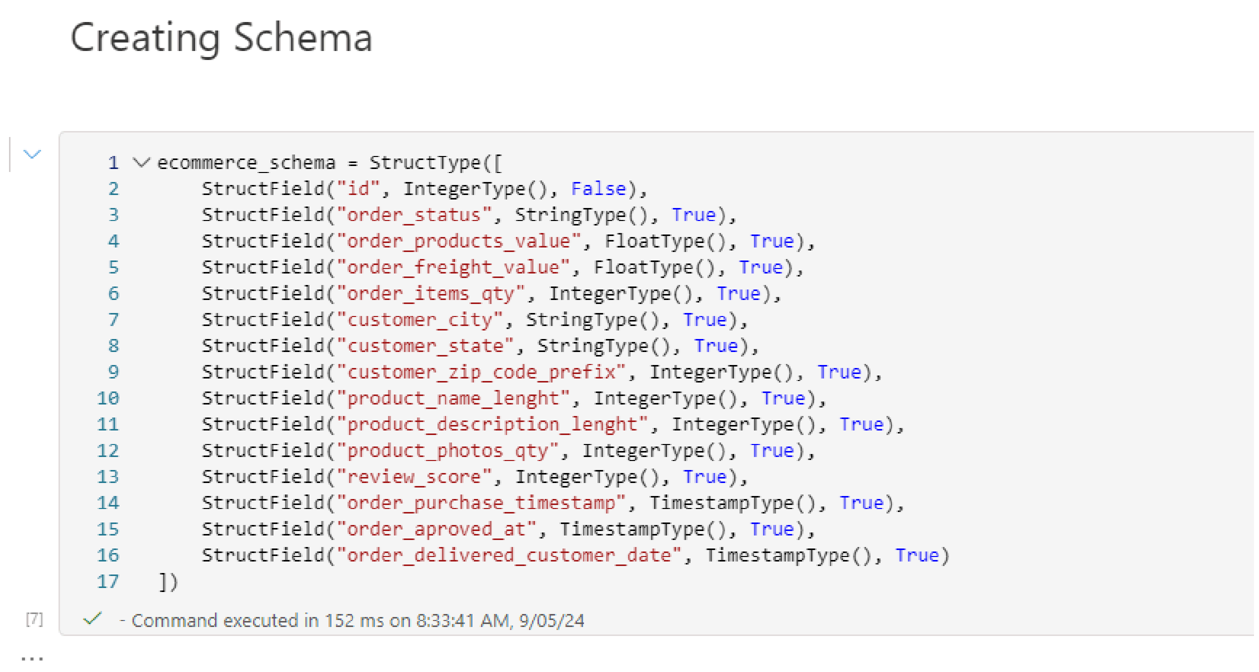
ecommercecontainerxg

Name	Last Modified	Content Type
synapse	9/1/2024, 10:16:23 AM	Folder
_SUCCESS	9/1/2024, 10:29:14 AM	
part-00000-660c7ed8-6c9c-4961-a361-7e1a973e413b-c000.csv	9/1/2024, 10:29:12 AM	

Week 3: Data streaming

Step 1: Connect to Spark shell with all the dependencies (Hive, Hadoop,and HDFS).

1. Create Schema of the CSV files

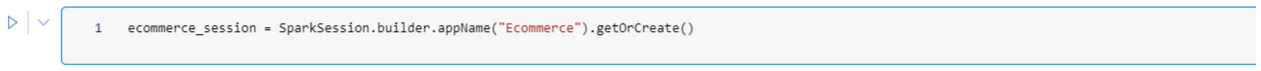


Creating Schema

```python
ecommerce_schema = StructType([
      StructField("id", IntegerType(), False),
      StructField("order_status", StringType(), True),
      StructField("order_products_value", FloatType(), True),
      StructField("order_freight_value", FloatType(), True),
      StructField("order_items_qty", IntegerType(), True),
      StructField("customer_city", StringType(), True),
      StructField("customer_state", StringType(), True),
      StructField("customer_zip_code_prefix", IntegerType(), True),
      StructField("product_name_lenght", IntegerType(), True),
      StructField("product_description_lenght", IntegerType(), True),
      StructField("product_photos_qty", IntegerType(), True),
      StructField("review_score", IntegerType(), True),
      StructField("order_purchase_timestamp", TimestampType(), True),
      StructField("order_aproved_at", TimestampType(), True),
      StructField("order_delivered_customer_date", TimestampType(), True)
])
```

[7] ✓ - Command executed in 152 ms on 8:33:41 AM, 9/05/24

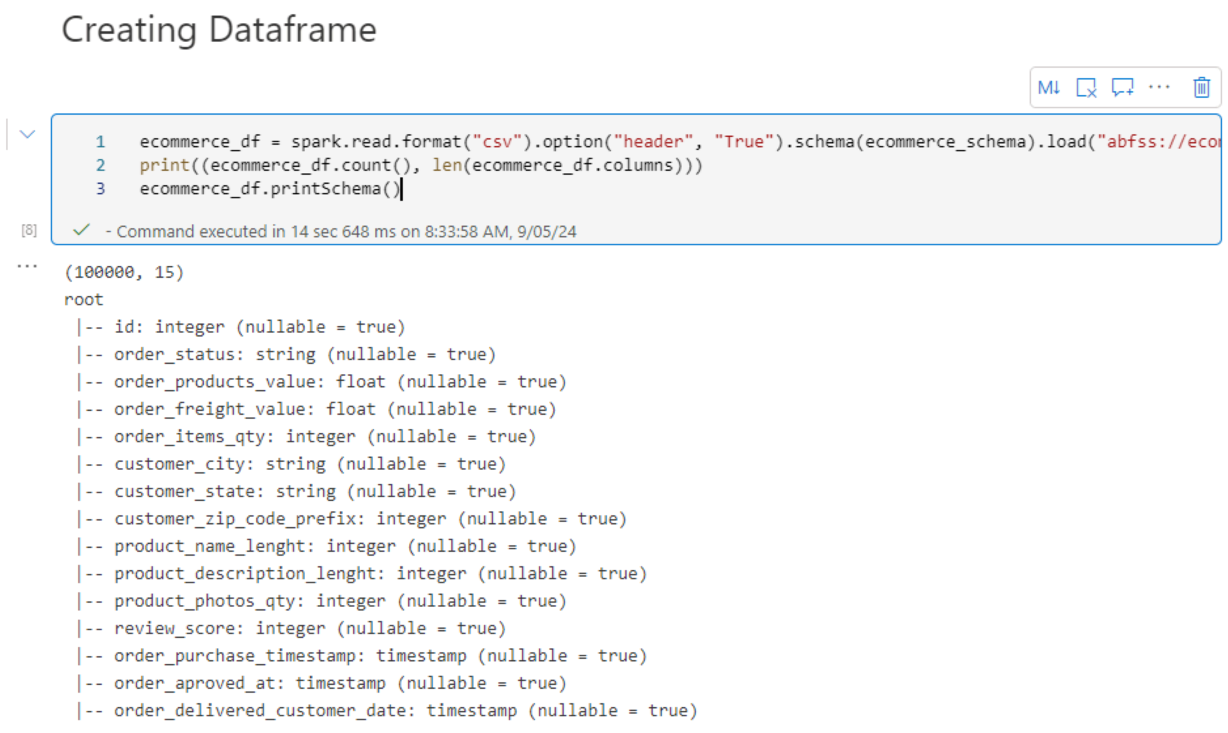
2. Create a Spark session(spark session is already up but this is what it would look like)



```python
ecommerce_session = SparkSession.builder.appName("Ecommerce").getOrCreate()
```

- Add Object Storage Service details as per the Cloud provider
- Add all variables to your environment as they contain sensitive data

Step 2: Read the CSV file and convert the file to a data frame



Creating Dataframe

```
ecommerce_df = spark.read.format("csv").option("header", "True").schema(ecommerce_schema).load("abfss://eco
print((ecommerce_df.count(), len(ecommerce_df.columns)))
ecommerce_df.printSchema()
```

[8] - Command executed in 14 sec 648 ms on 8:33:58 AM, 9/05/24

```
(100000, 15)
root
 |-- id: integer (nullable = true)
 |-- order_status: string (nullable = true)
 |-- order_products_value: float (nullable = true)
 |-- order_freight_value: float (nullable = true)
 |-- order_items_qty: integer (nullable = true)
 |-- customer_city: string (nullable = true)
 |-- customer_state: string (nullable = true)
 |-- customer_zip_code_prefix: integer (nullable = true)
 |-- product_name_lenght: integer (nullable = true)
 |-- product_description_lenght: integer (nullable = true)
 |-- product_photos_qty: integer (nullable = true)
 |-- review_score: integer (nullable = true)
 |-- order_purchase_timestamp: timestamp (nullable = true)
 |-- order_aproved_at: timestamp (nullable = true)
 |-- order_delivered_customer_date: timestamp (nullable = true)
```

Step 3: Convert “order_purchase_timestamp” to week and day using UDF

Setting up Dataframe

Creating Date Columns

```
ecommerce_df = ecommerce_df.withColumn("order_purchase_as_date",to_date(col("order_purchase_timestamp"),'dd/MM/yy'))
ecommerce_df = ecommerce_df.withColumn("order_approved_at_as_date",to_date(col("order_aproved_at"),'dd/MM/yy'))
ecommerce_df = ecommerce_df.withColumn("order_delivery_customer_date_as_date",to_date(col("order_delivered_customer_date"),'dd/MM/yy'))
```

✓ - Command executed in 146 ms on 8:34:00 AM, 9/05/24

```
ecommerce_df.printSchema()
display(ecommerce_df)
```

[10] ✓ - Command executed in 3 sec 990 ms on 8:34:05 AM, 9/05/24

```
root
 |-- id: integer (nullable = true)
 |-- order_status: string (nullable = true)
 |-- order_products_value: float (nullable = true)
 |-- order_freight_value: float (nullable = true)
 |-- order_items_qty: integer (nullable = true)
 |-- customer_city: string (nullable = true)
 |-- customer_state: string (nullable = true)
 |-- customer_zip_code_prefix: integer (nullable = true)
 |-- product_name_lenght: integer (nullable = true)
 |-- product_description_lenght: integer (nullable = true)
 |-- product_photos_qty: integer (nullable = true)
 |-- review_score: integer (nullable = true)
 |-- order_purchase_timestamp: timestamp (nullable = true)
 |-- order_aproved_at: timestamp (nullable = true)
 |-- order_delivered_customer_date: timestamp (nullable = true)
 |-- order_purchase_as_date: date (nullable = true)
 |-- order_approved_at_as_date: date (nullable = true)
 |-- order_delivery_customer_date_as_date: date (nullable = true)
```

Creating a Day of Week Column

```
def dayAsString(day_int):
  day_of_week_string = {
    1: "Monday",
    2: "Tuesday",
    3: "Wednesday",
    4: "Thursday",
    5: "Friday",
    6: "Saturday",
    7: "Sunday"
  }
  return day_of_week_string[day_int]
```

[11] ✓ - Command executed in 157 ms on 8:34:05 AM, 9/05/24

```
day_as_string_udf = udf(lambda x: dayAsString(x), StringType())
```

[12] ✓ - Command executed in 198 ms on 8:34:05 AM, 9/05/24

```
ecommerce_df = ecommerce_df.withColumn("day_of_week",day_as_string_udf(dayofweek(col("order_purchase_as_date"))))
display(ecommerce_df)
```

[13] ✓ - Command executed in 10 sec 676 ms on 8:34:16 AM, 9/05/24

```
ecommerce_df.select("day_of_week").distinct().show()
```

[14] ✓ - Command executed in 1 sec 832 ms on 8:34:18 AM, 9/05/24

```
+-----------+
|day_of_week|
+-----------+
|  Wednesday|
|    Tuesday|
|     Friday|
|   Thursday|
|   Saturday|
|     Monday|
|     Sunday|
+-----------+
```

Creating a Week of Month Column

```
def weekOfMonth(date1):
  month = date1.month
  year = date1.year
  month_start = dt.date(year, month, 1)
  if (dt.date(year, 1, 1) == month_start):
    week_of_year_month_start = 1
  else:
    week_of_year_month_start = month_start.isocalendar()[1]
  week_of_year_date1 = date1.isocalendar()[1]
  return week_of_year_date1 - week_of_year_month_start + 1
```

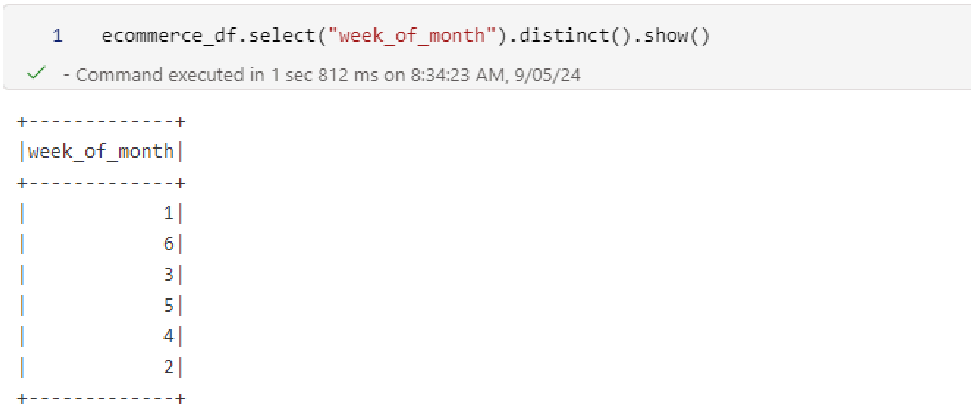
[15] ✓ - Command executed in 152 ms on 8:34:18 AM, 9/05/24

```
weekOfMonth_udf = udf(lambda x: weekOfMonth(x), IntegerType())
```

[16] ✓ - Command executed in 155 ms on 8:34:19 AM, 9/05/24

```
ecommerce_df = ecommerce_df.withColumn("week_of_month",weekOfMonth_udf(col("order_purchase_as_date")))
display(ecommerce_df)
```

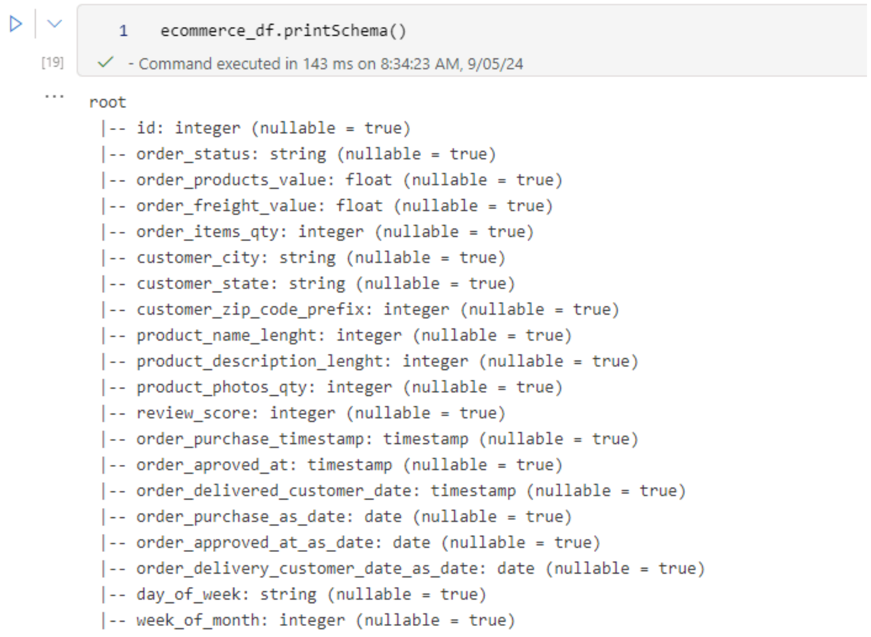
[17] ✓ - Command executed in 1 sec 925 ms on 8:34:21 AM, 9/05/24



```
ecommerce_df.select("week_of_month").distinct().show()
```

✓ - Command executed in 1 sec 812 ms on 8:34:23 AM, 9/05/24

```
+-------------+
|week_of_month|
+-------------+
|            1|
|            6|
|            3|
|            5|
|            4|
|            2|
+-------------+
```



```
ecommerce_df.printSchema()
```

[19] ✓ - Command executed in 143 ms on 8:34:23 AM, 9/05/24

```
root
 |-- id: integer (nullable = true)
 |-- order_status: string (nullable = true)
 |-- order_products_value: float (nullable = true)
 |-- order_freight_value: float (nullable = true)
 |-- order_items_qty: integer (nullable = true)
 |-- customer_city: string (nullable = true)
 |-- customer_state: string (nullable = true)
 |-- customer_zip_code_prefix: integer (nullable = true)
 |-- product_name_lenght: integer (nullable = true)
 |-- product_description_lenght: integer (nullable = true)
 |-- product_photos_qty: integer (nullable = true)
 |-- review_score: integer (nullable = true)
 |-- order_purchase_timestamp: timestamp (nullable = true)
 |-- order_aproved_at: timestamp (nullable = true)
 |-- order_delivered_customer_date: timestamp (nullable = true)
 |-- order_purchase_as_date: date (nullable = true)
 |-- order_approved_at_as_date: date (nullable = true)
 |-- order_delivery_customer_date_as_date: date (nullable = true)
 |-- day_of_week: string (nullable = true)
 |-- week_of_month: integer (nullable = true)
```

Step 4: Calculate the following data:

1. Total sales and order distribution per day and week for each city
2. Total sales and order distribution per day and week for each state
3. Average review score, average freight value, average order approval, and delivery time
4. The freight charges per city and total freight charges

Getting Ecommerce Results

Making Dictionary to Store Results

```
ecommerce_datasets = {}
```

✓ - Command executed in 176 ms on 8:34:23 AM, 9/05/24

Insight on Sales

```
total_sales_df = ecommerce_df.agg(round(sum("order_products_value"),2).alias("total_sales"))
total_sales_df.show()

ecommerce_datasets["total_sales"] = total_sales_df
```

✓ - Command executed in 1 sec 58 ms on 8:34:24 AM, 9/05/24

```
+-------------+
|  total_sales|
+-------------+
|1.284147698E7|
+-------------+
```

Sales by Day

```
total_sales_by_day_df = ecommerce_df.groupBy("day_of_week").agg(round(sum("order_products_value"),2).alias("total_sales")).orderBy("total_sales")
total_sales_by_day_df.show()

ecommerce_datasets["total_sales_by_day"] = total_sales_by_day_df
```

[51] ✓ - Command executed in 1 sec 890 ms on 8:50:44 AM, 9/05/24

```
+-----------+-----------+
|day_of_week|total_sales|
+-----------+-----------+
|     Sunday| 1406302.11|
|     Monday| 1530648.75|
|   Saturday| 1811400.14|
|     Friday| 1922774.44|
|   Thursday| 1989730.72|
|  Wednesday| 2077208.79|
|    Tuesday| 2103412.03|
+-----------+-----------+
```

```
total_sales_by_day_and_city_df = ecommerce_df.groupBy("day_of_week","customer_city").agg(round(sum("order_products_value"),2)\
.alias("total_sales")).orderBy("customer_city")
total_sales_by_day_and_city_df.show()

ecommerce_datasets["total_sales_by_day_and_city"] = total_sales_by_day_and_city_df
```

[52] ✓ - Command executed in 1 sec 845 ms on 8:50:46 AM, 9/05/24

```
+-----------+--------------------+-----------+
|day_of_week|       customer_city|total_sales|
+-----------+--------------------+-----------+
|     Friday|ALMIRANTE TAMANDA...|       49.9|
|    Tuesday|ALMIRANTE TAMANDA...|      99.99|
|     Sunday|ALTA FLORESTA D'O...|     708.99|
|    Tuesday|ALTO ALEGRE DOS P...|      299.0|
|     Sunday|ALTO ALEGRE DOS P...|      299.0|
|     Monday|ALTO ALEGRE DOS P...|     314.99|
|     Friday|     ALVORADA D'OESTE|      328.0|
|    Tuesday|     ALVORADA D'OESTE|     359.98|
|   Saturday| Abadia dos Dourados|       39.9|
|    Tuesday| Abadia dos Dourados|      319.0|
|    Tuesday|           Abadiania|       68.9|
|     Friday|           Abadiania|     949.99|
|   Thursday|              Abaete|     398.79|
|    Tuesday|              Abaete|      449.0|
|  Wednesday|              Abaete|      321.6|
|     Friday|              Abaete|      56.99|
|  Wednesday|          Abaetetuba|     435.41|
|     Monday|          Abaetetuba|     115.99|
|   Saturday|          Abaetetuba|     1574.8|
|     Friday|          Abaetetuba|     164.89|
+-----------+--------------------+-----------+
only showing top 20 rows
```

```python
total_sales_by_day_and_state_df = ecommerce_df.groupBy("day_of_week","customer_state").agg(round(sum("order_products_value"),2)\
.alias("total_sales")).orderBy("customer_state")
total_sales_by_day_and_state_df.show()

ecommerce_datasets["total_sales_by_day_and_state"] = total_sales_by_day_and_state_df
```

```
+-----------+--------------+-----------+
|day_of_week|customer_state|total_sales|
+-----------+--------------+-----------+
|   Thursday|            AC|    2862.77|
|     Monday|            AC|     627.57|
|     Sunday|            AC|     853.49|
|     Friday|            AC|    2542.04|
|   Saturday|            AC|    3987.15|
|  Wednesday|            AC|    3478.97|
|    Tuesday|            AC|    2548.83|
|   Saturday|            AL|    7661.65|
|     Sunday|            AL|    8837.77|
|     Monday|            AL|   13089.56|
|    Tuesday|            AL|   14769.35|
|  Wednesday|            AL|   10002.48|
|     Friday|            AL|   11160.45|
|   Thursday|            AL|   12779.28|
|     Sunday|            AM|    2519.56|
|     Friday|            AM|    2063.87|
|  Wednesday|            AM|    2145.55|
|   Saturday|            AM|    3192.41|
|     Monday|            AM|     3949.5|
|   Thursday|            AM|    2505.38|
+-----------+--------------+-----------+
only showing top 20 rows
```

Sales by Week

```
total_sales_by_week_df = ecommerce_df.groupBy("week_of_month").agg(round(sum("order_products_value"),2).alias("total_sales")).orderBy("week_of_month")
total_sales_by_week_df.show()

ecommerce_datasets["total_sales_by_week"] = total_sales_by_week_df
```

✓ - Command executed in 1 sec 77 ms on 8:50:49 AM, 9/05/24

```
+-------------+-----------+
|week_of_month|total_sales|
+-------------+-----------+
|            1| 1542155.29|
|            2| 3089498.12|
|            3|  2957336.5|
|            4| 3027674.28|
|            5| 2058842.95|
|            6|  165969.84|
+-------------+-----------+
```

```
total_sales_by_week_and_city_df = ecommerce_df.groupBy("week_of_month","customer_city").agg(round(sum("order_products_value"),2)\
.alias("total_sales")).orderBy("customer_city")
total_sales_by_week_and_city_df.show()

ecommerce_datasets["total_sales_by_week_and_city"] = total_sales_by_week_and_city_df
```

✓ - Command executed in 1 sec 782 ms on 8:50:51 AM, 9/05/24

```
week_of_month|       customer_city|total_sales|
-------------+--------------------+-----------+
            2|ALMIRANTE TAMANDA...|      99.99|
            5|ALMIRANTE TAMANDA...|       49.9|
            4|ALTA FLORESTA D'O...|     349.99|
            1|ALTA FLORESTA D'O...|      359.0|
            3|ALTO ALEGRE DOS P...|      299.0|
            4|ALTO ALEGRE DOS P...|     613.99|
            1|    ALVORADA D'OESTE|      328.0|
            5|    ALVORADA D'OESTE|     359.98|
            3|  Abadia dos Dourados|      358.9|
            5|           Abadiania|       68.9|
            1|           Abadiania|     949.99|
            5|              Abaete|      515.7|
            4|              Abaete|     176.89|
            2|              Abaete|      254.9|
            1|              Abaete|      69.99|
            3|              Abaete|      208.9|
            3|          Abaetetuba|    2410.96|
            1|          Abaetetuba|     134.99|
            2|          Abaetetuba|     305.51|
            4|          Abaetetuba|       63.8|
-------------+--------------------+-----------+
nly showing top 20 rows
```

```
total_sales_by_week_and_state_df = ecommerce_df.groupBy("week_of_month","customer_state").agg(round(sum("order_products_value"),2)\
.alias("total_sales")).orderBy("customer_state")
total_sales_by_week_and_state_df.show()

ecommerce_datasets["total_sales_by_week_and_state"] = total_sales_by_week_and_state_df
```

```
+-------------+--------------+-----------+
|week_of_month|customer_state|total_sales|
+-------------+--------------+-----------+
|            1|            AC|     3787.2|
|            4|            AC|    2092.61|
|            5|            AC|    4023.55|
|            3|            AC|    4034.82|
|            6|            AC|      699.0|
|            2|            AC|    2263.64|
|            6|            AL|    1057.29|
|            4|            AL|   18700.29|
|            1|            AL|    7343.29|
|            3|            AL|   17451.02|
|            2|            AL|   19961.02|
|            5|            AL|   13787.63|
|            4|            AM|    5775.42|
|            3|            AM|    5308.23|
|            5|            AM|     1554.8|
|            2|            AM|    6492.69|
|            6|            AM|      255.4|
|            1|            AM|    2789.03|
|            2|            AP|    5580.18|
|            5|            AP|    2929.67|
+-------------+--------------+-----------+
only showing top 20 rows
```

Insights on Orders

```python
total_order_df = ecommerce_df.agg(count("id").alias("total_orders"))
total_order_df.show()

ecommerce_datasets["total_orders"] = total_order_df
```

✓ - Command executed in 512 ms on 8:50:52 AM, 9/05/24

```
+------------+
|total_orders|
+------------+
|      100000|
+------------+
```

Orders by Day

```python
total_order_by_day_df = ecommerce_df.groupBy("day_of_week").agg(count("id").alias("total_orders")).orderBy("total_orders")
total_order_by_day_df.show()

ecommerce_datasets["total_orders_by_day"] = total_order_by_day_df
```

```
+-----------+------------+
|day_of_week|total_orders|
+-----------+------------+
|     Sunday|       10944|
|     Monday|       12034|
|   Saturday|       14199|
|     Friday|       14857|
|   Thursday|       15634|
|  Wednesday|       16045|
|    Tuesday|       16287|
+-----------+------------+
```

```python
total_order_by_day_and_city_df = ecommerce_df.groupBy("day_of_week","customer_city").agg(count("id").alias("total_orders")).orderBy("customer_city")
total_order_by_day_and_city_df.show()

ecommerce_datasets["total_orders_by_day_and_city"] = total_order_by_day_and_city_df
```

✓ - Command executed in 1 sec 163 ms on 8:50:55 AM, 9/05/24

```
+-----------+--------------------+------------+
|day_of_week|       customer_city|total_orders|
+-----------+--------------------+------------+
|     Friday|ALMIRANTE TAMANDA...|           1|
|    Tuesday|ALMIRANTE TAMANDA...|           1|
|     Sunday|ALTA FLORESTA D'O...|           2|
|    Tuesday|ALTO ALEGRE DOS P...|           1|
|     Sunday|ALTO ALEGRE DOS P...|           1|
|     Monday|ALTO ALEGRE DOS P...|           1|
|     Friday|     ALVORADA D'OESTE|           1|
|    Tuesday|     ALVORADA D'OESTE|           1|
|   Saturday| Abadia dos Dourados|           1|
|    Tuesday| Abadia dos Dourados|           2|
|    Tuesday|            Abadiania|           1|
|     Friday|            Abadiania|           1|
|   Thursday|               Abaete|           3|
|    Tuesday|               Abaete|           1|
|  Wednesday|               Abaete|           3|
|     Friday|               Abaete|           1|
|  Wednesday|           Abaetetuba|           3|
|     Monday|           Abaetetuba|           2|
|   Saturday|           Abaetetuba|           3|
|     Friday|           Abaetetuba|           2|
+-----------+--------------------+------------+
only showing top 20 rows
```

```
total_order_by_day_and_state_df = ecommerce_df.groupBy("day_of_week","customer_state").agg(count("id").alias("total_orders")).orderBy("customer_state")
total_order_by_day_and_state_df.show()

ecommerce_datasets["total_orders_by_day_and_state"] = total_order_by_day_and_state_df
```

```
+-----------+--------------+------------+
|day_of_week|customer_state|total_orders|
+-----------+--------------+------------+
|   Thursday|            AC|          10|
|     Monday|            AC|           7|
|     Sunday|            AC|          11|
|     Friday|            AC|          15|
|   Saturday|            AC|          16|
|  Wednesday|            AC|          13|
|    Tuesday|            AC|          12|
|   Saturday|            AL|          53|
|     Sunday|            AL|          54|
|     Monday|            AL|          59|
|    Tuesday|            AL|          80|
|  Wednesday|            AL|          52|
|     Friday|            AL|          71|
|   Thursday|            AL|          65|
|     Sunday|            AM|          21|
|     Friday|            AM|          12|
|  Wednesday|            AM|          22|
|   Saturday|            AM|          21|
|     Monday|            AM|          20|
|   Thursday|            AM|          28|
+-----------+--------------+------------+
only showing top 20 rows
```

Orders by Week

```python
total_order_by_week_df = ecommerce_df.groupBy("week_of_month").agg(count("id").alias("total_orders")).orderBy("week_of_month")
total_order_by_week_df.show()

ecommerce_datasets["total_orders_by_week"] = total_order_by_week_df
```

✓ - Command executed in 1 sec 104 ms on 8:50:57 AM, 9/05/24

```
+-------------+------------+
|week_of_month|total_orders|
+-------------+------------+
|            1|       12202|
|            2|       23676|
|            3|       23255|
|            4|       23392|
|            5|       16160|
|            6|        1315|
+-------------+------------+
```

```python
total_order_by_week_and_city_df = ecommerce_df.groupBy("week_of_month","customer_city").agg(count("id").alias("total_orders")).orderBy("customer_city")
total_order_by_week_and_city_df.show()

ecommerce_datasets["total_orders_by_week_and_city"] = total_order_by_week_and_city_df
```

✓ - Command executed in 1 sec 85 ms on 8:50:59 AM, 9/05/24

```
|week_of_month|       customer_city|total_orders|
+-------------+--------------------+------------+
|            2|ALMIRANTE TAMANDA...|           1|
|            5|ALMIRANTE TAMANDA...|           1|
|            4|ALTA FLORESTA D'O...|           1|
|            1|ALTA FLORESTA D'O...|           1|
|            3|ALTO ALEGRE DOS P...|           1|
|            4|ALTO ALEGRE DOS P...|           2|
|            1|    ALVORADA D'OESTE|           1|
|            5|    ALVORADA D'OESTE|           1|
|            3| Abadia dos Dourados|           3|
|            5|           Abadiania|           1|
|            1|           Abadiania|           1|
|            5|              Abaete|           3|
|            4|              Abaete|           2|
|            2|              Abaete|           1|
|            1|              Abaete|           1|
|            3|              Abaete|           1|
|            3|          Abaetetuba|           7|
|            1|          Abaetetuba|           1|
|            2|          Abaetetuba|           2|
|            4|          Abaetetuba|           2|
+-------------+--------------------+------------+
only showing top 20 rows
```

```python
total_order_by_week_and_state_df = ecommerce_df.groupBy("week_of_month","customer_state").agg(count("id").alias("total_orders")).orderBy("customer_state")
total_order_by_week_and_state_df.show()

ecommerce_datasets["total_orders_by_week_and_state"] = total_order_by_week_and_state_df
```

```
+-------------+--------------+------------+
|week_of_month|customer_state|total_orders|
+-------------+--------------+------------+
|            1|            AC|          15|
|            4|            AC|          20|
|            5|            AC|          15|
|            3|            AC|          20|
|            6|            AC|           1|
|            2|            AC|          13|
|            6|            AL|           4|
|            4|            AL|         104|
|            1|            AL|          52|
|            3|            AL|         113|
|            2|            AL|          89|
|            5|            AL|          72|
|            4|            AM|          41|
|            3|            AM|          30|
|            5|            AM|          18|
|            2|            AM|          41|
|            6|            AM|           2|
|            1|            AM|          22|
|            2|            AP|          25|
|            5|            AP|          14|
+-------------+--------------+------------+
only showing top 20 rows
```

Average of Misc. Columns

By Day

```python
average_review_score_by_day_df = ecommerce_df.groupBy("day_of_week").agg(round(avg("review_score"),2).alias("average_review_score")).orderBy("average_review_score")
average_review_score_by_day_df.show()

ecommerce_datasets["average_review_score_by_day"] = average_review_score_by_day_df
```

✓ - Command executed in 1 sec 69 ms on 8:51:01 AM, 9/05/24

```
+-----------+--------------------+
|day_of_week|average_review_score|
+-----------+--------------------+
|    Tuesday|                4.04|
| |   Friday|                4.04|
|  Wednesday|                4.05|
|   Saturday|                4.05|
|     Sunday|                4.05|
|   Thursday|                4.06|
|     Monday|                4.06|
+-----------+--------------------+
```

```python
average_freight_value_by_day_df = ecommerce_df.groupBy("day_of_week").agg(round(avg("order_freight_value"),2).alias("average_freight_value"))\
.orderBy("average_freight_value")
average_freight_value_by_day_df.show()

ecommerce_datasets["average_freight_value_by_day"] = average_freight_value_by_day_df
```

```
+-----------+---------------------+
|day_of_week|average_freight_value|
+-----------+---------------------+
|     Monday|                21.49|
|    Tuesday|                 21.5|
|  Wednesday|                 21.7|
|   Saturday|                21.79|
|   Thursday|                21.85|
|     Friday|                21.92|
|     Sunday|                21.98|
+-----------+---------------------+
```

```python
ecommerce_df = ecommerce_df.withColumn("time_to_approve_order",col("order_aproved_at").cast("long") - col('order_purchase_timestamp').cast("long"))
ecommerce_df = ecommerce_df.withColumn("time_to_deliver_order",col("order_delivered_customer_date").cast("long") - col('order_purchase_timestamp').cast("long"))
display(ecommerce_df)
```

time_to_approve_order	time_to_deliver_order
660	728940
110580	1190760
1020	811680
1020	1141200
3720	248340
780	1429200
176580	176580
720	863100
116460	848340
600	1574340
540	1093020
720	497160
148020	1043940

```
average_time_to_approve_order_by_day_df = ecommerce_df.groupBy("day_of_week").agg(round(avg("time_to_approve_order"),2)\
.alias("average_time_to_approve_order")).orderBy("average_time_to_approve_order")
average_time_to_approve_order_by_day_df.show()

ecommerce_datasets["average_time_to_approve_order_by_day"] = average_time_to_approve_order_by_day_df
```

[67] ✓ - Command executed in 1 sec 183 ms on 8:51:05 AM, 9/05/24

```
+-----------+-----------------------------+
|day_of_week|average_time_to_approve_order|
+-----------+-----------------------------+
|   Thursday|                     31410.68|
|     Friday|                     31577.61|
|  Wednesday|                     32383.93|
|    Tuesday|                     33473.55|
|     Monday|                     38871.57|
|   Saturday|                     48351.09|
|     Sunday|                     52226.07|
+-----------+-----------------------------+
```

```
average_time_to_deliver_order_by_day_df = ecommerce_df.groupBy("day_of_week").agg(round(avg("time_to_deliver_order"),2)\
.alias("average_time_to_approve_order")).orderBy("average_time_to_approve_order")
average_time_to_deliver_order_by_day_df.show()

ecommerce_datasets["average_time_to_deliver_order_by_day"] = average_time_to_deliver_order_by_day_df
```

```
+-----------+-----------------------------+
|day_of_week|average_time_to_approve_order|
+-----------+-----------------------------+
|     Monday|                   1033518.33|
|    Tuesday|                   1036488.73|
|  Wednesday|                   1043121.45|
|   Thursday|                   1075211.61|
|     Friday|                   1101193.49|
|     Sunday|                   1154172.92|
|   Saturday|                   1172419.75|
+-----------+-----------------------------+
```

By Week

```
average_review_score_by_week_df = ecommerce_df.groupBy("week_of_month").agg(round(avg("review_score"),2).alias("average_review_score")).orderBy("average_review_score")
average_review_score_by_week_df.show()

ecommerce_datasets["average_review_score_by_week"] = average_review_score_by_week_df
```

```
+-------------+-------------------+
|week_of_month|average_review_score|
+-------------+-------------------+
|            3|               4.04|
|            5|               4.04|
|            4|               4.05|
|            2|               4.06|
|            1|               4.07|
|            6|               4.07|
+-------------+-------------------+
```

```
average_freight_value_by_week_df = ecommerce_df.groupBy("week_of_month").agg(round(avg("order_freight_value"),2)\
.alias("average_freight_value")).orderBy("average_freight_value")
average_freight_value_by_week_df.show()

ecommerce_datasets["average_freight_value_by_week"] = average_freight_value_by_week_df
```

[70] ✓ - Command executed in 1 sec 59 ms on 8:51:09 AM, 9/05/24

```
+-------------+---------------------+
|week_of_month|average_freight_value|
+-------------+---------------------+
|            6|                21.43|
|            3|                21.62|
|            5|                21.71|
|            2|                21.78|
|            1|                21.82|
|            4|                21.82|
+-------------+---------------------+
```

```
average_time_to_approve_order_by_week_df = ecommerce_df.groupBy("week_of_month").agg(round(avg("time_to_approve_order"),2)\
.alias("average_time_to_approve_order")).orderBy("average_time_to_approve_order")
average_time_to_approve_order_by_week_df.show()

ecommerce_datasets["average_time_to_approve_order_by_week"] = average_time_to_approve_order_by_week_df
```

[71] ✓ - Command executed in 1 sec 34 ms on 8:51:10 AM, 9/05/24

```
+-------------+-----------------------------+
|week_of_month|average_time_to_approve_order|
+-------------+-----------------------------+
|            3|                     34514.17|
|            2|                     37290.22|
|            5|                     37842.18|
|            6|                     37842.89|
|            4|                     39152.78|
|            1|                      40009.4|
+-------------+-----------------------------+
```

```
average_time_to_deliver_order_by_week_df = ecommerce_df.groupBy("week_of_month").agg(round(avg("time_to_deliver_order"),2)\
.alias("average_time_to_deliver_order")).orderBy("average_time_to_deliver_order")
average_time_to_deliver_order_by_week_df.show()

ecommerce_datasets["average_time_to_deliver_order_by_week"] = average_time_to_deliver_order_by_week_df
```

```
+-------------+----------------------------+
|week_of_month|average_time_to_deliver_order|
+-------------+----------------------------+
|            6|                   812527.32|
|            5|                  1073336.96|
|            3|                  1082696.05|
|            2|                  1089760.94|
|            4|                  1095327.44|
|            1|                  1105146.08|
+-------------+----------------------------+
```

Extra Look at Freight Costs

```
total_freight_charges_df = ecommerce_df.agg(round(sum("order_freight_value"),2).alias("total_freight_charges"))
total_freight_charges_df.show()

ecommerce_datasets["total_freight_charges"] = total_freight_charges_df
```

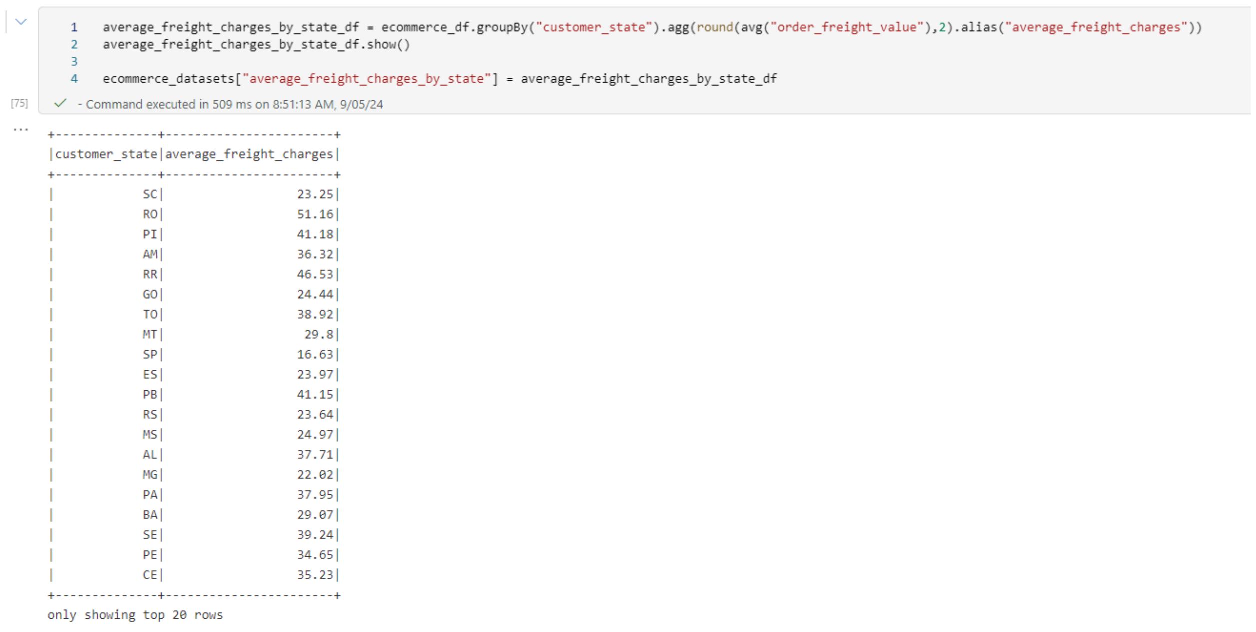
[73] ✓ - Command executed in 507 ms on 8:51:12 AM, 9/05/24

```
+---------------------+
|total_freight_charges|
+---------------------+
|           2174127.92|
+---------------------+
```

```
average_freight_charges_by_city_df = ecommerce_df.groupBy("customer_city").agg(round(avg("order_freight_value"),2).alias("average_freight_charges"))
average_freight_charges_by_city_df.show()

ecommerce_datasets["average_freight_charges_by_city"] = average_freight_charges_by_city_df
```

[74] ✓ - Command executed in 524 ms on 8:51:12 AM, 9/05/24

```
+---------------+-----------------------+
|  customer_city|average_freight_charges|
+---------------+-----------------------+
|       Araruama|                   24.3|
|       Guidoval|                  17.61|
|     Piranguinho|                  16.18|
|     Tres Pontas|                  22.49|
|Senador Guiomard|                  68.55|
|       Rio Novo|                  15.56|
|      Carrancas|                  16.09|
|      Fronteira|                  20.64|
|         Utinga|                  26.07|
|    Assis Brasil|                  24.84|
|         Pianco|                  65.21|
|       Macaubas|                  45.09|
|     Livramento|                  25.47|
|    Cristalandia|                  48.81|
|       Machados|                  42.51|
|    Rio do Campo|                  25.14|
|     Purilandia|                  27.49|
|       Alambari|                  18.78|
|   Guajará-Mirim|                  38.67|
|          Tapes|                  21.69|
+---------------+-----------------------+
```



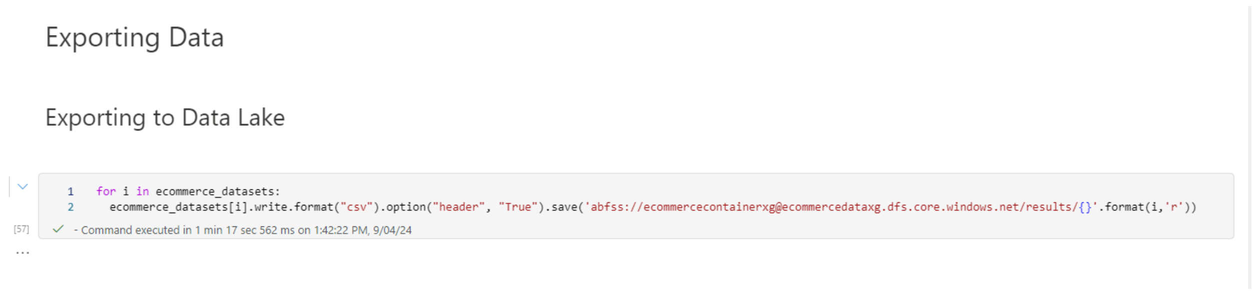
```
average_freight_charges_by_state_df = ecommerce_df.groupBy("customer_state").agg(round(avg("order_freight_value"),2).alias("average_freight_charges"))
average_freight_charges_by_state_df.show()

ecommerce_datasets["average_freight_charges_by_state"] = average_freight_charges_by_state_df
```

[75] ✓ - Command executed in 509 ms on 8:51:13 AM, 9/05/24

```
+--------------+-----------------------+
|customer_state|average_freight_charges|
+--------------+-----------------------+
|            SC|                  23.25|
|            RO|                  51.16|
|            PI|                  41.18|
|            AM|                  36.32|
|            RR|                  46.53|
|            GO|                  24.44|
|            TO|                  38.92|
|            MT|                   29.8|
|            SP|                  16.63|
|            ES|                  23.97|
|            PB|                  41.15|
|            RS|                  23.64|
|            MS|                  24.97|
|            AL|                  37.71|
|            MG|                  22.02|
|            PA|                  37.95|
|            BA|                  29.07|
|            SE|                  39.24|
|            PE|                  34.65|
|            CE|                  35.23|
+--------------+-----------------------+
only showing top 20 rows
```

Week 4: Data analysis and visualization

Step 1: Write the results into HDFS

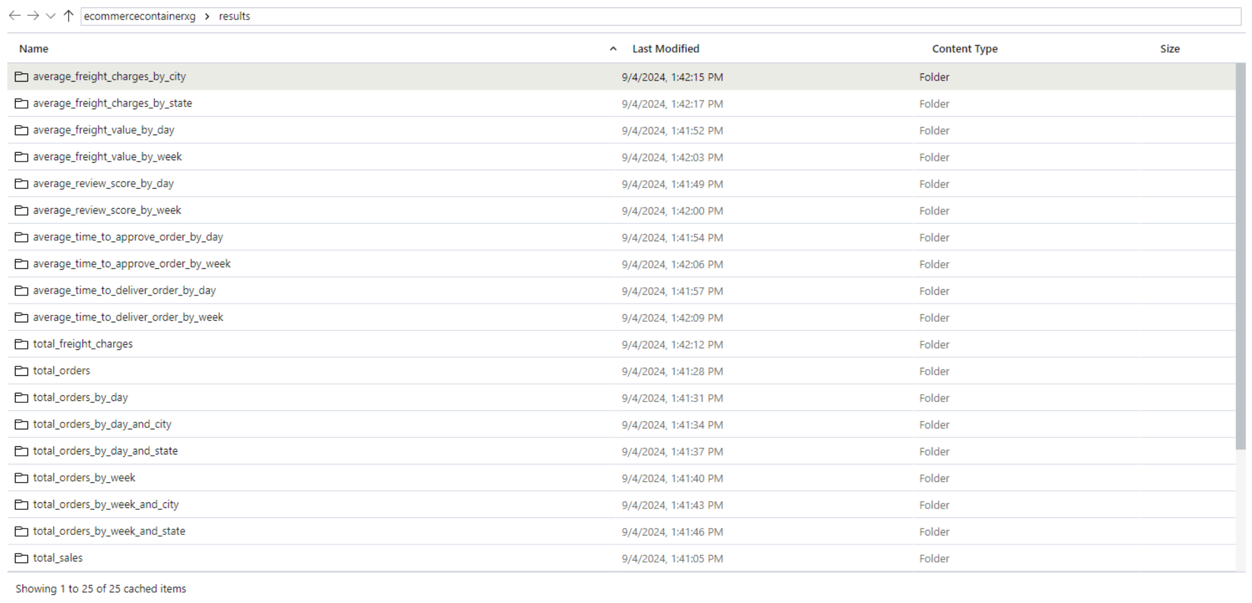


Exporting Data

Exporting to Data Lake

```
for i in ecommerce_datasets:
  ecommerce_datasets[i].write.format("csv").option("header", "True").save('abfss://ecommercecontainerxg@ecommercedataxg.dfs.core.windows.net/results/{}'.format(i,'r'))
```

[57] ✓ - Command executed in 1 min 17 sec 562 ms on 1:42:22 PM, 9/04/24

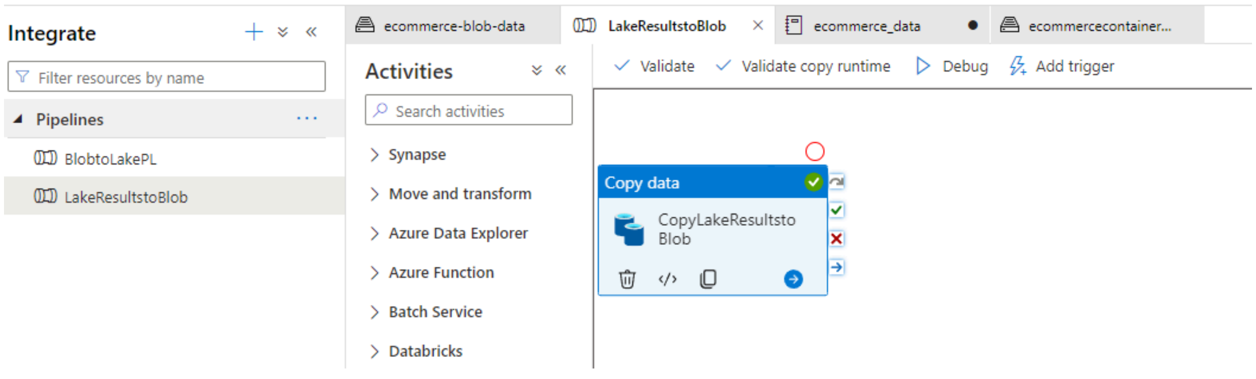


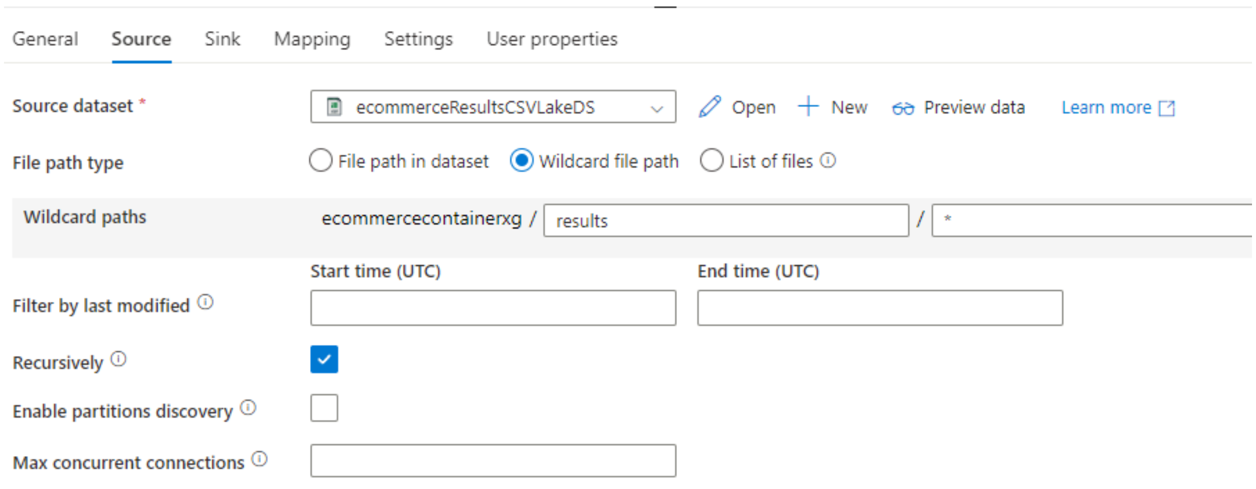
ecommercecontainerxg › results

Name	Last Modified	Content Type	Size
average_freight_charges_by_city	9/4/2024, 1:42:15 PM	Folder	
average_freight_charges_by_state	9/4/2024, 1:42:17 PM	Folder	
average_freight_value_by_day	9/4/2024, 1:41:52 PM	Folder	
average_freight_value_by_week	9/4/2024, 1:42:03 PM	Folder	
average_review_score_by_day	9/4/2024, 1:41:49 PM	Folder	
average_review_score_by_week	9/4/2024, 1:42:00 PM	Folder	
average_time_to_approve_order_by_day	9/4/2024, 1:41:54 PM	Folder	
average_time_to_approve_order_by_week	9/4/2024, 1:42:06 PM	Folder	
average_time_to_deliver_order_by_day	9/4/2024, 1:41:57 PM	Folder	
average_time_to_deliver_order_by_week	9/4/2024, 1:42:09 PM	Folder	
total_freight_charges	9/4/2024, 1:42:12 PM	Folder	
total_orders	9/4/2024, 1:41:28 PM	Folder	
total_orders_by_day	9/4/2024, 1:41:31 PM	Folder	
total_orders_by_day_and_city	9/4/2024, 1:41:34 PM	Folder	
total_orders_by_day_and_state	9/4/2024, 1:41:37 PM	Folder	
total_orders_by_week	9/4/2024, 1:41:40 PM	Folder	
total_orders_by_week_and_city	9/4/2024, 1:41:43 PM	Folder	
total_orders_by_week_and_state	9/4/2024, 1:41:46 PM	Folder	
total_sales	9/4/2024, 1:41:05 PM	Folder	

Showing 1 to 25 of 25 cached items

Step 2: Save the final dataset into object storage service per the cloud platform





Source dataset * ecommerceResultsCSVLakeDS

File path type: File path in dataset [✓ Wildcard file path] List of files

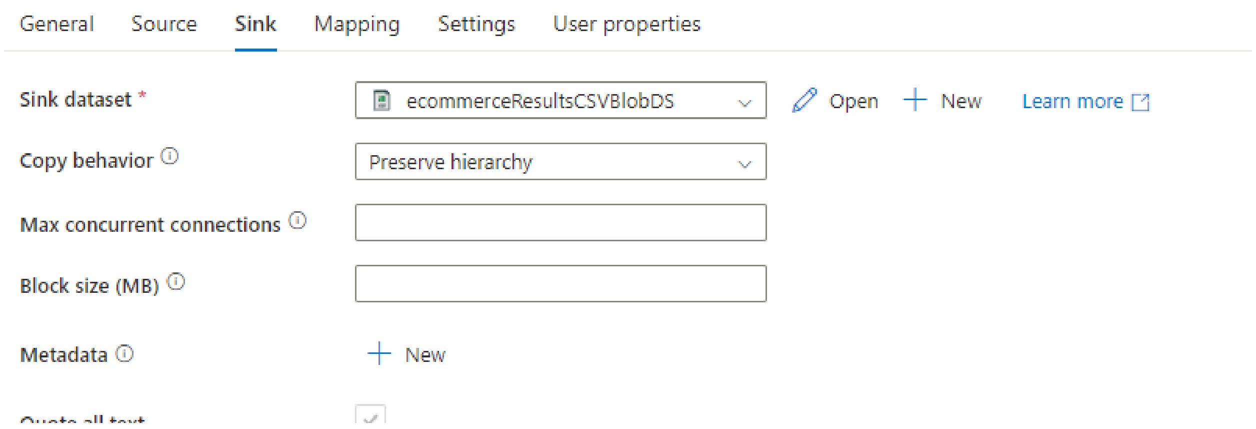
Wildcard paths: ecommercecontainerxg / results / *

	Start time (UTC)	End time (UTC)
Filter by last modified		

Recursively ✓

Enable partitions discovery

Max concurrent connections



Sink dataset * ecommerceResultsCSVBlobDS

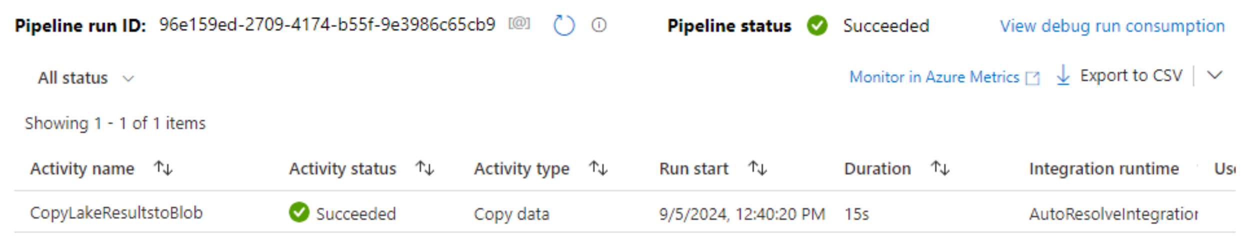
Copy behavior: Preserve hierarchy

Max concurrent connections

Block size (MB)

Metadata + New

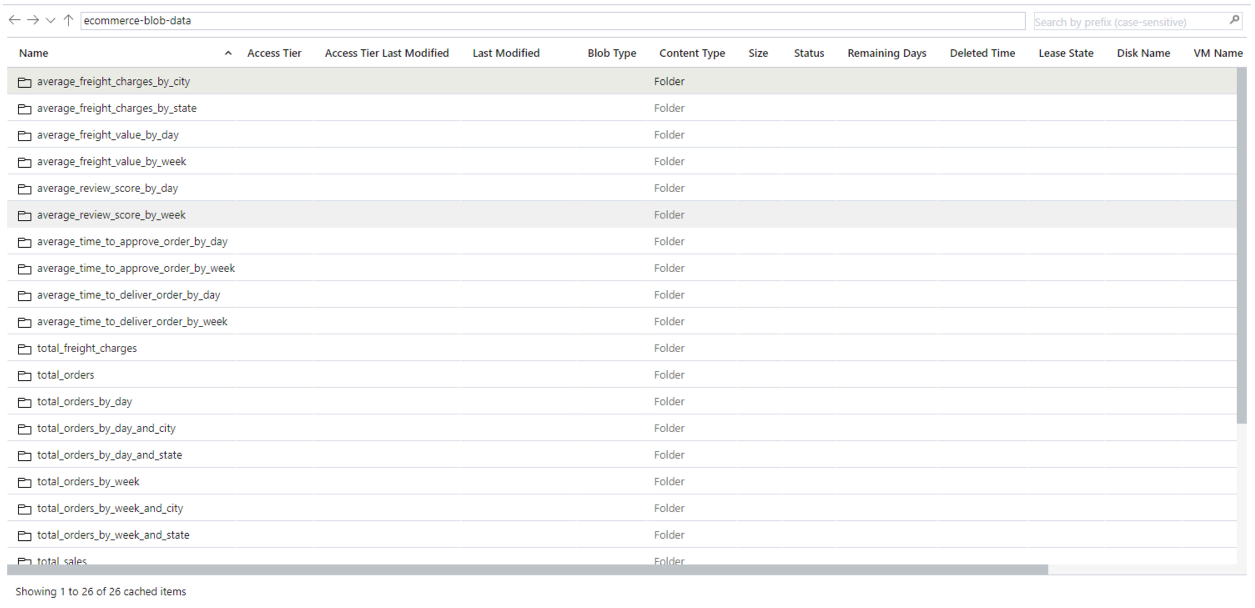
Quote all text



Pipeline run ID: 96e159ed-2709-4174-b55f-9e3986c65cb9 Pipeline status ✓ Succeeded

Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	Us
CopyLakeResultstoBlob	✓ Succeeded	Copy data	9/5/2024, 12:40:20 PM	15s	AutoResolveIntegratior	



ecommerce-blob-data

Search by prefix (case-sensitive)

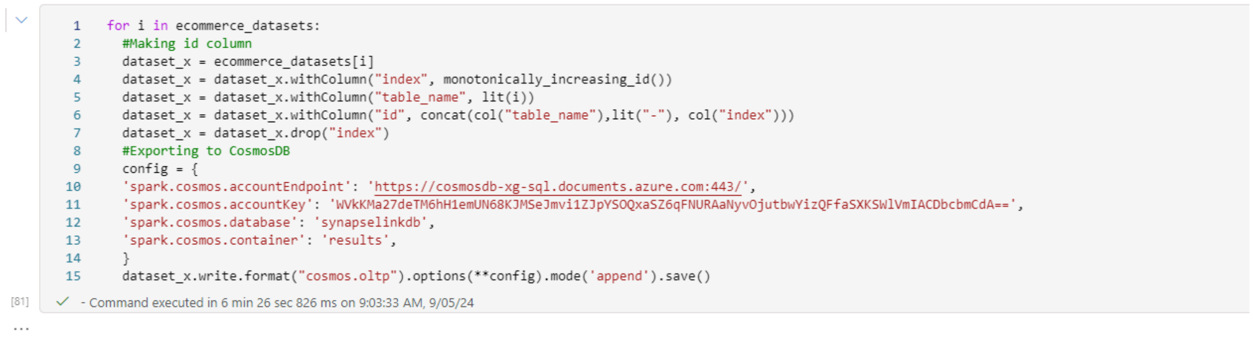
Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days	Deleted Time	Lease State	Disk Name	VM Name
average_freight_charges_by_city					Folder							
average_freight_charges_by_state					Folder							
average_freight_value_by_day					Folder							
average_freight_value_by_week					Folder							
average_review_score_by_day					Folder							
average_review_score_by_week					Folder							
average_time_to_approve_order_by_day					Folder							
average_time_to_approve_order_by_week					Folder							
average_time_to_deliver_order_by_day					Folder							
average_time_to_deliver_order_by_week					Folder							
total_freight_charges					Folder							
total_orders					Folder							
total_orders_by_day					Folder							
total_orders_by_day_and_city					Folder							
total_orders_by_day_and_state					Folder							
total_orders_by_week					Folder							
total_orders_by_week_and_city					Folder							
total_orders_by_week_and_state					Folder							
total_sales					Folder							

Showing 1 to 26 of 26 cached items

Step 3: Create a DB cluster that is also a NoSQL using the relevant service on the cloud platform

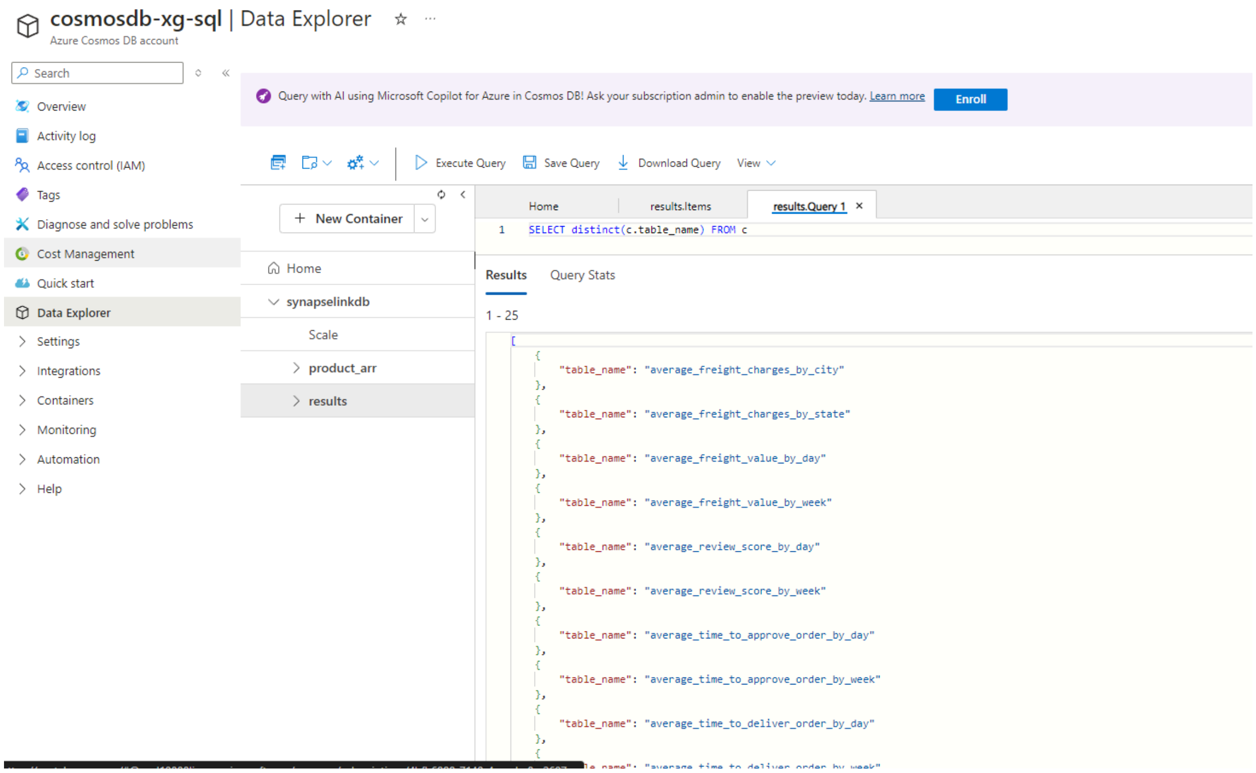
Step 4: Save insights in the NoSQL DB mentioned in the previous step

Exporting to CosmosDB



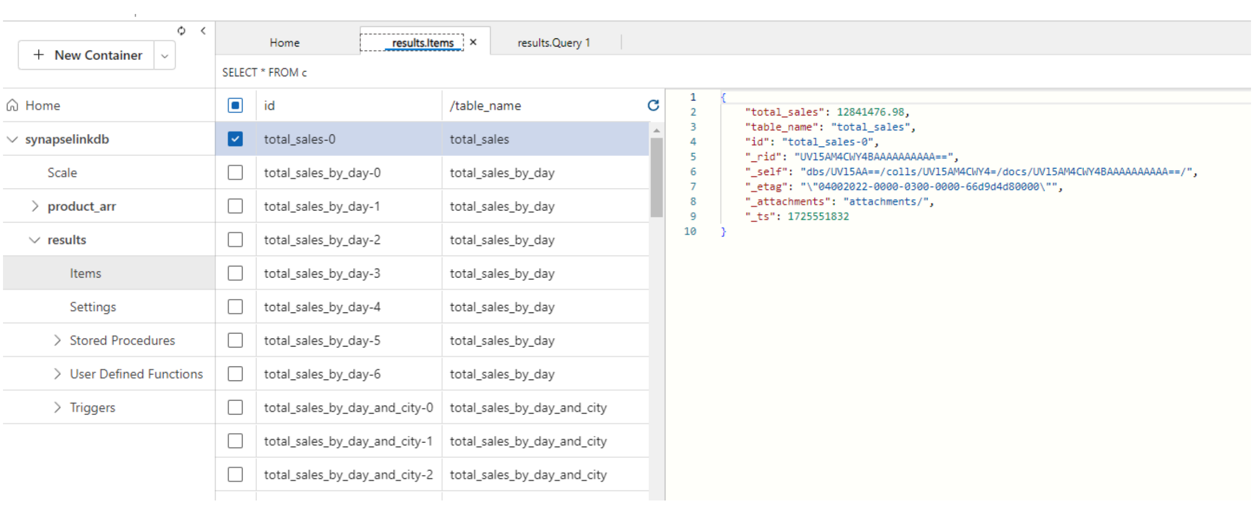
```
for i in ecommerce_datasets:
  #Making id column
  dataset_x = ecommerce_datasets[i]
  dataset_x = dataset_x.withColumn("index", monotonically_increasing_id())
  dataset_x = dataset_x.withColumn("table_name", lit(i))
  dataset_x = dataset_x.withColumn("id", concat(col("table_name"),lit("-"), col("index")))
  dataset_x = dataset_x.drop("index")
  #Exporting to CosmosDB
  config = {
  'spark.cosmos.accountEndpoint': 'https://cosmosdb-xg-sql.documents.azure.com:443/',
  'spark.cosmos.accountKey': 'WVkKMa27deTM6hH1emUN68KJMSeJmvi1ZJpYSOQxaSZ6qFNURAaNyvOjutbwYizQFfaSXKSWlVmIACDbcbmCdA==',
  'spark.cosmos.database': 'synapselinkdb',
  'spark.cosmos.container': 'results',
  }
  dataset_x.write.format("cosmos.oltp").options(**config).mode('append').save()
```

[81] ✓ - Command executed in 6 min 26 sec 826 ms on 9:03:33 AM, 9/05/24



```
SELECT distinct(c.table_name) FROM c
```

```
[
    {
        "table_name": "average_freight_charges_by_city"
    },
    {
        "table_name": "average_freight_charges_by_state"
    },
    {
        "table_name": "average_freight_value_by_day"
    },
    {
        "table_name": "average_freight_value_by_week"
    },
    {
        "table_name": "average_review_score_by_day"
    },
    {
        "table_name": "average_review_score_by_week"
    },
    {
        "table_name": "average_time_to_approve_order_by_day"
    },
    {
        "table_name": "average_time_to_approve_order_by_week"
    },
    {
        "table_name": "average_time_to_deliver_order_by_day"
    },
    {
        "table_name": "average_time_to_deliver_order_by_week"
```



SELECT * FROM c

	id	/table_name
✓	total_sales-0	total_sales
	total_sales_by_day-0	total_sales_by_day
	total_sales_by_day-1	total_sales_by_day
	total_sales_by_day-2	total_sales_by_day
	total_sales_by_day-3	total_sales_by_day
	total_sales_by_day-4	total_sales_by_day
	total_sales_by_day-5	total_sales_by_day
	total_sales_by_day-6	total_sales_by_day
	total_sales_by_day_and_city-0	total_sales_by_day_and_city
	total_sales_by_day_and_city-1	total_sales_by_day_and_city
	total_sales_by_day_and_city-2	total_sales_by_day_and_city

```
{
    "total_sales": 12841476.98,
    "table_name": "total_sales",
    "id": "total_sales-0",
    "_rid": "UV15AM4CWY4BAAAAAAAAAA==",
    "_self": "dbs/UV15AA==/colls/UV15AM4CWY4=/docs/UV15AM4CWY4BAAAAAAAAAA==/",
    "_etag": "\"04002022-0000-0300-0000-66d9d4d80000\"",
    "_attachments": "attachments/",
    "_ts": 1725551832
}
```