DATA ENGINEERING PROJECT

SUPERSTORE

GROUP 2

- ALISHA OKERE
- GETRUDE SHABIHA
- JACQUI KIRUKI

INTRODUCTION

- A company called Superstore would like our team of data engineers to prepare the raw data stored in their server so that a data analyst can be able to analyze the large data set with ease.
- The data needs to go through preparation, cleaning, and some analysis before it is handed over to the client.
- Various questions asked by the client serve as the client's needs and requirements.
- The steps implemented below are geared towards achieving the best results to meet the client's needs and requirements.

DATA PREPARATIONS

We identified the Azure Datalake Storage Gen 2 (ADLS2) as the storage which we recommend to meet the client's storage requirements for their large amounts of data (big data). With that in mind, we proceeded to create the following resources using Microsoft Azure Portal:

- A **Resource Group** called **Superstore**
- A **Storage Account** called **storagesuperstore2** in the Resource group.
- A **container** called **superstoresales** in the Storage Account
- A **synapse workspace** called **superstore-workspace** using the Azure Synapse Analytics

We ingested the data with the URL link provided. Fixed the following error which was not uploading data into the datalake (ADLS2) i.e. the container **superstoresales.**

ERROR ENCOUNTERED

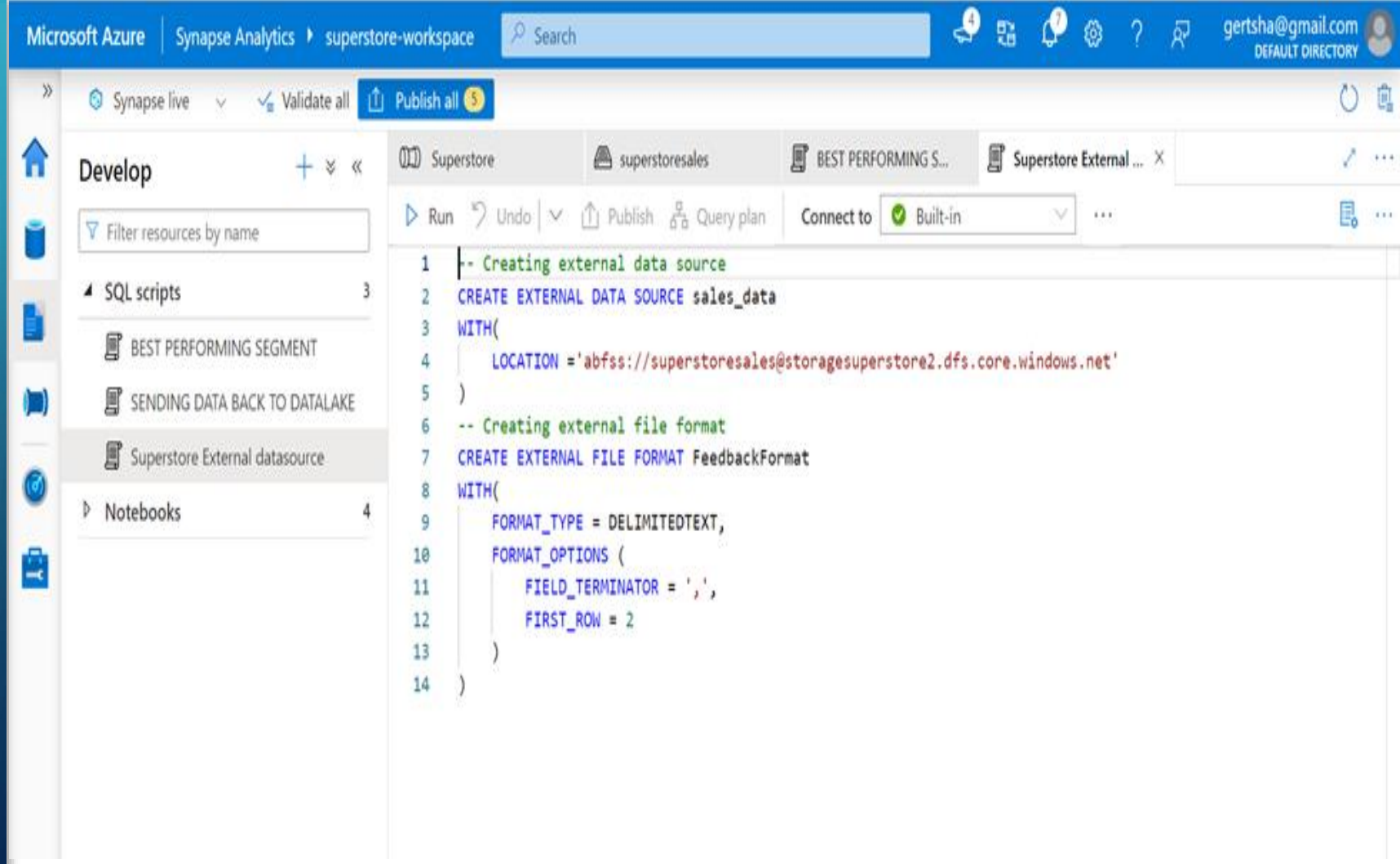
Operation on target Copy_t58 failed: ErrorCode=DelimitedTextMoreColumnsThanDefined,'Type=Microsoft.DataTransfer.Common.Shared.HybridDeliveryException,Message=Error found when processing 'Csv/Tsv Format Text' source 'a28288a9-9fcb-406d-a9c3-2da00fd09254' with row number 34: found more columns than expected column count 21.,Source=Microsoft.DataTransfer.Common,'

SOLUTION

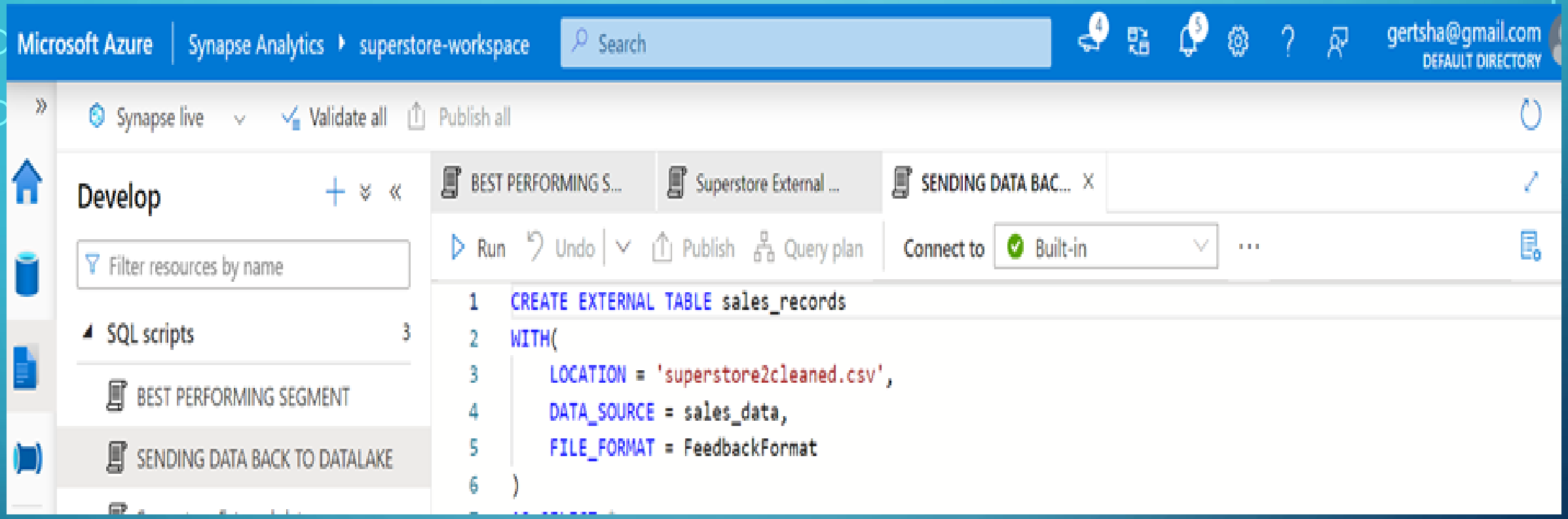
- *Error resolved by setting the Escape Character for both the Source and Sink to Double Quote(") before debugging/Triggering again.*

DATA CLEANING USING SQL

- Used **CETAS** to view and manipulate the data since it is a requirement that we load the cleaned and transformed data back into the data lake. The following CETAS steps were implemented:
  - External database created => **superstore**
  - External Data Source created => **sales_data**
  - External File Format created => **FeedbackFormat**
  - External Table created => **sales_records.** (Completed successfully and querying the table is possible too.



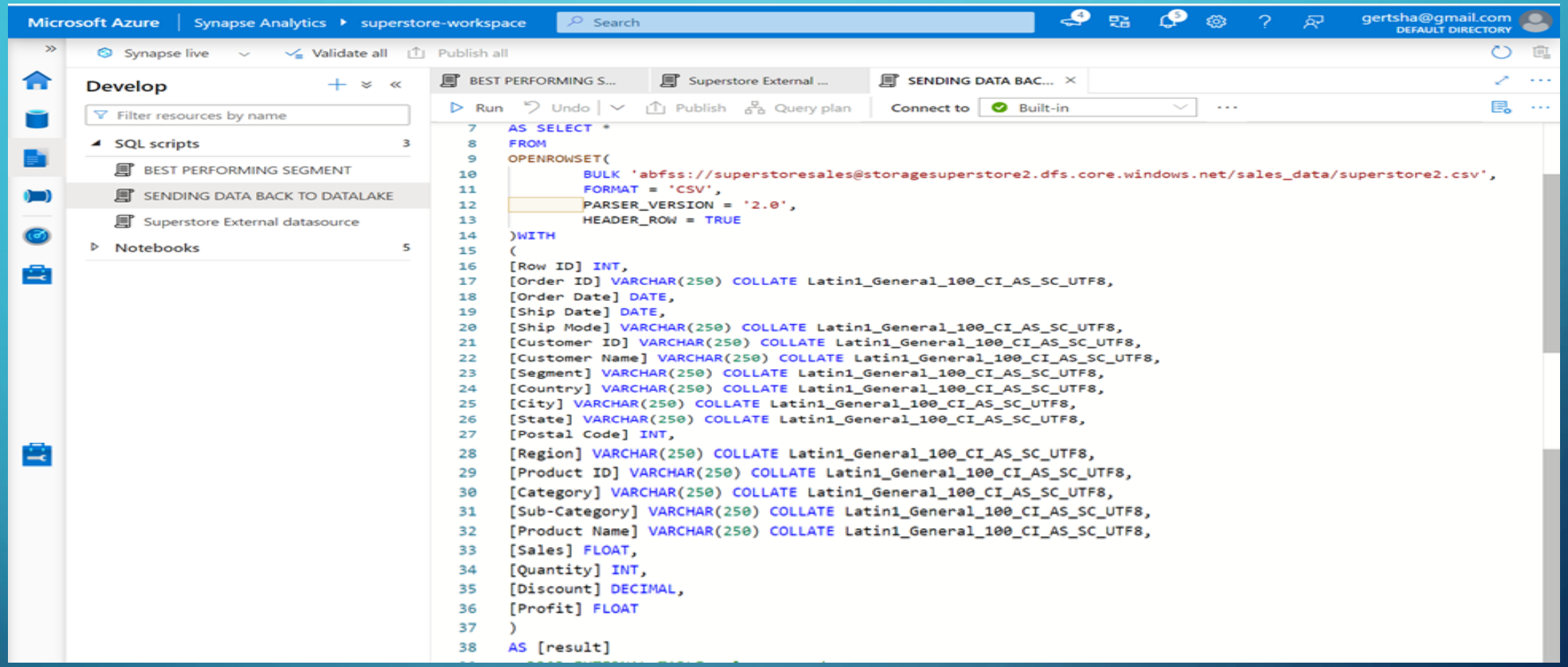
DATA CLEANING USING SQL



Assigned the following datatypes to the various fields while specifying the structure and schema of the external table

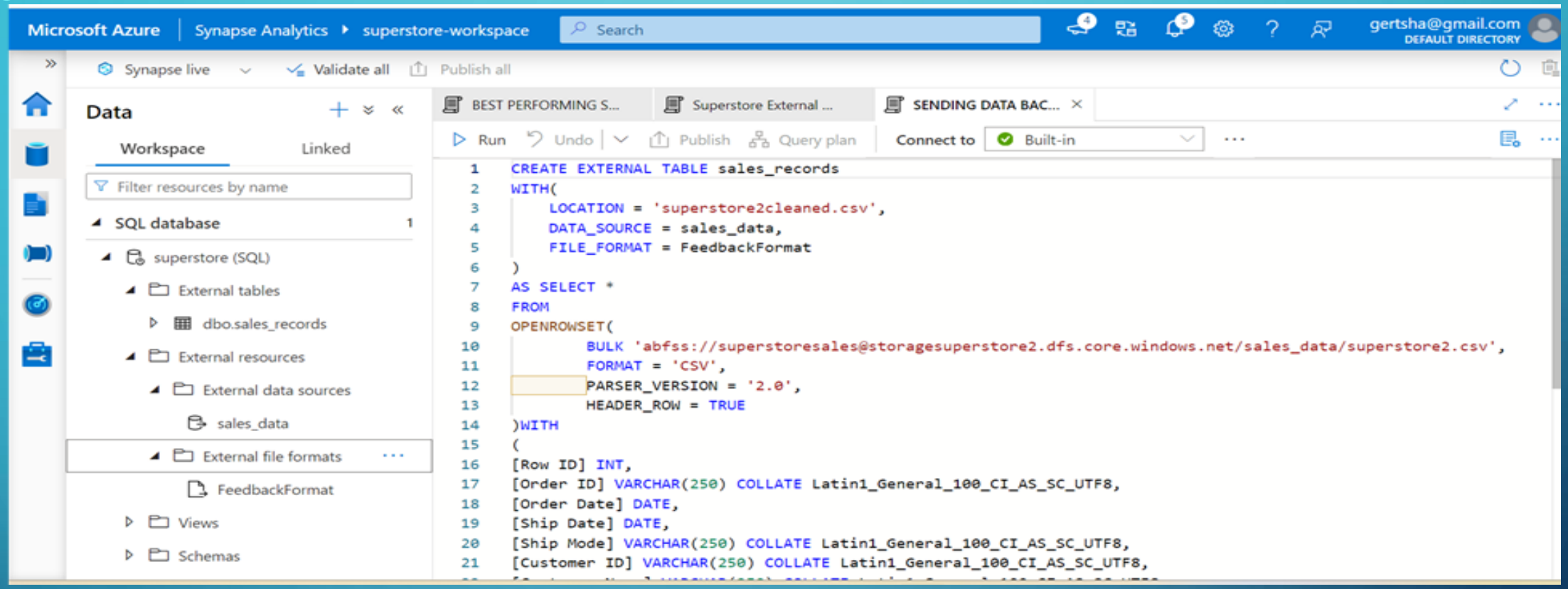
- [Row ID], [Postal Code], [Quantity] => **INT**
- [Order Date], [Ship Date] => **DATE**
- [Sales], [Profit] => **FLOAT**
- [Discount] => **DECIMAL**
- All other fields => VARCHAR()

DATA CLEANING USING SQL



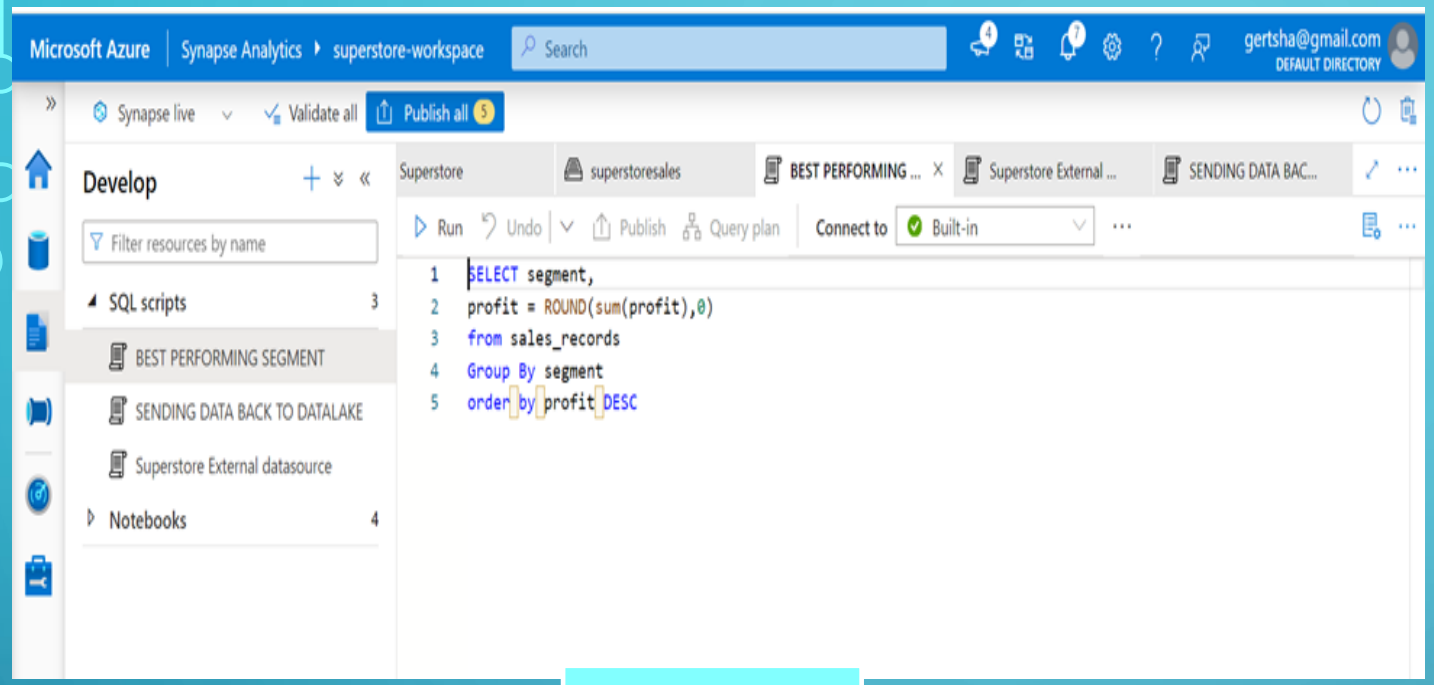
- After Cleaning we would like to now use the newly cleaned data for analysis, therefore load the new data into the data lake for future reference.

DATA CLEANING USING SQL



- The result of our script can be seen on the left panel of the screen ;
- The cleaned data is successfully loaded into the External Table in the Data lake. Querying the data using SQL is possible and it is available for future reference.

DATA CLEANING USING SQL

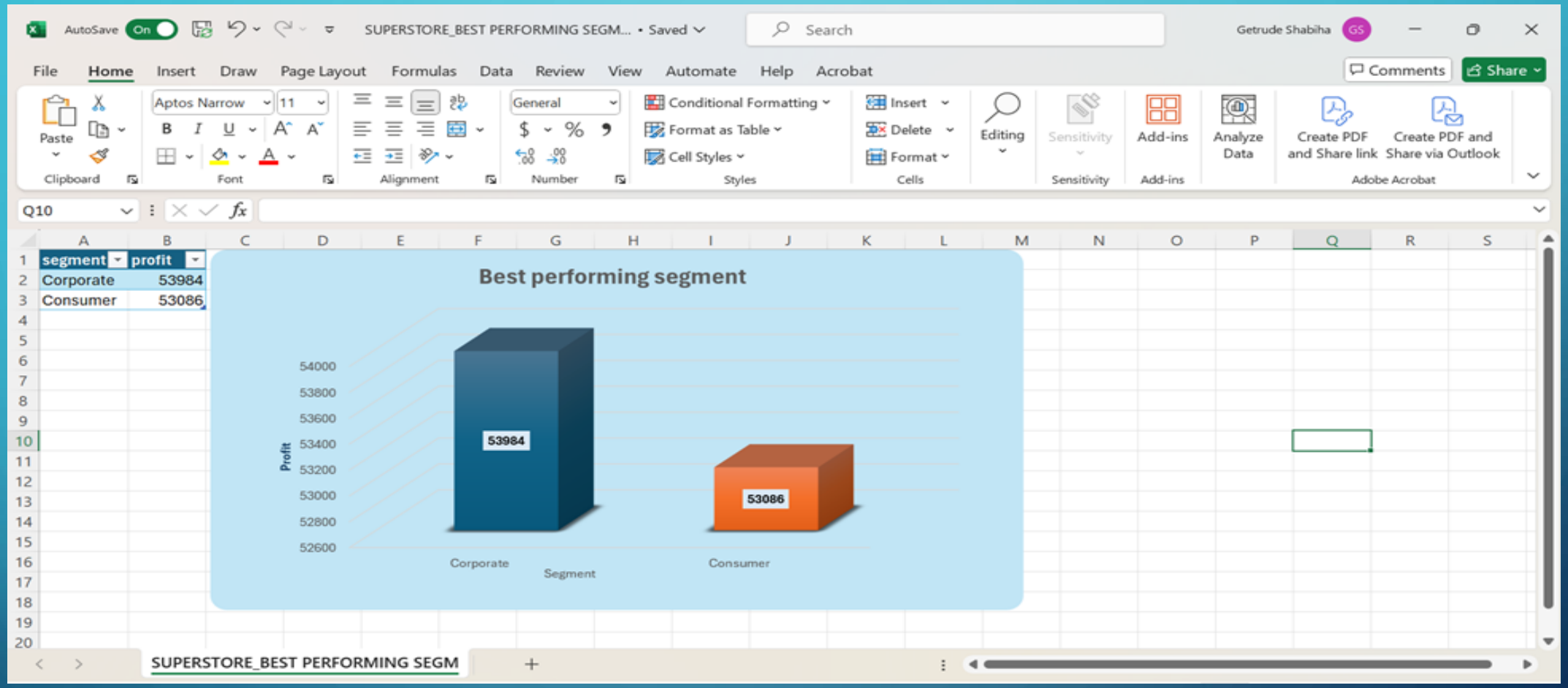


Result

Segment	Total Profit
Corporate	53984
Consumer	53086

- The client wants a quick glance at which segment is performing best, by using the serverless SQL show which segment is performing best (No need to save it back to the data lake).
- To showcase the best performing segment to the client using the serverless SQL, the SQL script shown above was used.

DATA CLEANING USING SQL



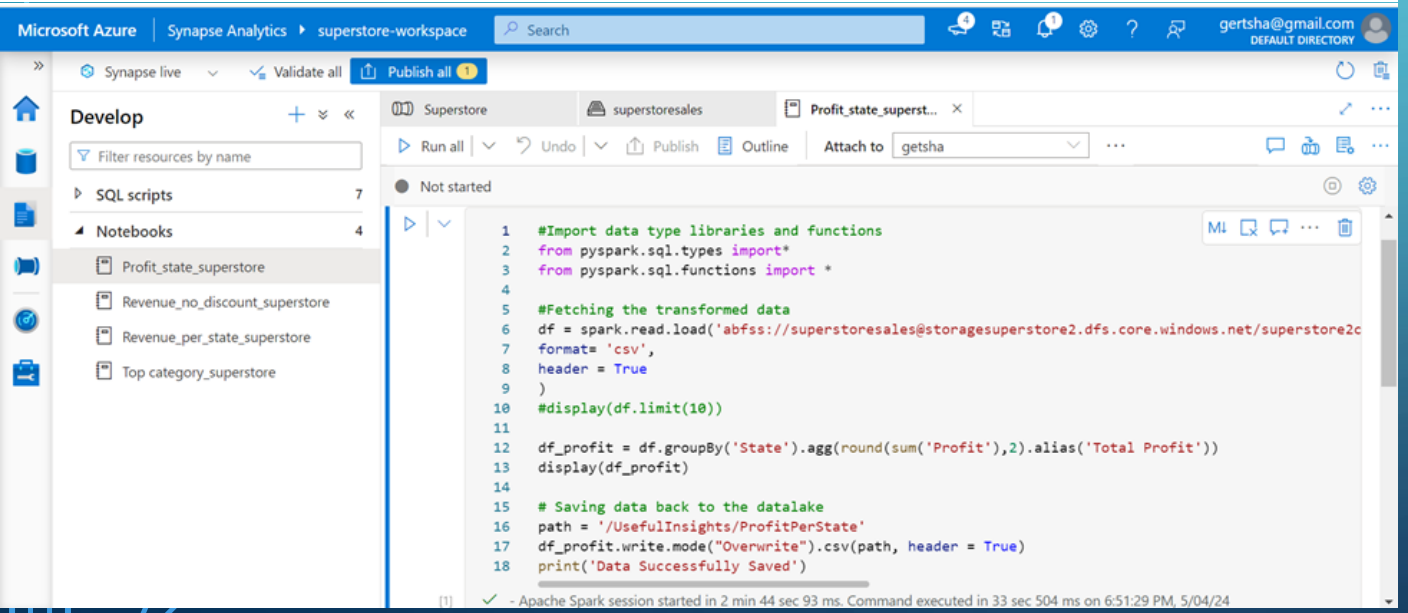
- We exported our result as a csv file and created a table and a pivot chart on excel to display the output in a simple manner that the client would interpret as below:

DATA TRANSFORMATION (PYTHON)

Sales	Quantity	Discount	Profit
3.5	4	0.52	6.72

Unit Price	Quantity	Discount	Profit	Revenue	Cost
3.5	4	0.52	6.72	12.18	5.46

```
Revenue = (Unit Price * Quantity) - (Unit Price * Discount)
        = (3.5 * 4) - (3.5 * 0.52)
        = 14 - 1.82
        = 12.18

Profit = Revenue - Cost
Cost = Revenue - Profit
```



- We then created a spark pool from which our notebooks would run and did a notebook to answer each question now using python.
- Now that the data has been cleaned and prepared, the client would like to see some useful insights from the data. The client would like to know the following: -
- How much profit he is making across all states.
- We did a bit of manipulation of the data with the respective columns/fields to calculate revenue as below;

DATA TRANSFORMATION (PYTHON)

Microsoft Azure | Synapse Analytics ▸ superstore-workspace

Search

gertsha@gmail.com
DEFAULT DIRECTORY

Synapse live | Validate all | Publish all 1

Develop

Filter resources by name

SQL scripts 7

Notebooks 4

- Profit_state_superstore
- Revenue_no_discount_superstore
- Revenue_per_state_superstore
- Top category_superstore

Superstore | superstoresales | Revenue_per_state_s... | Profit_state_superstore

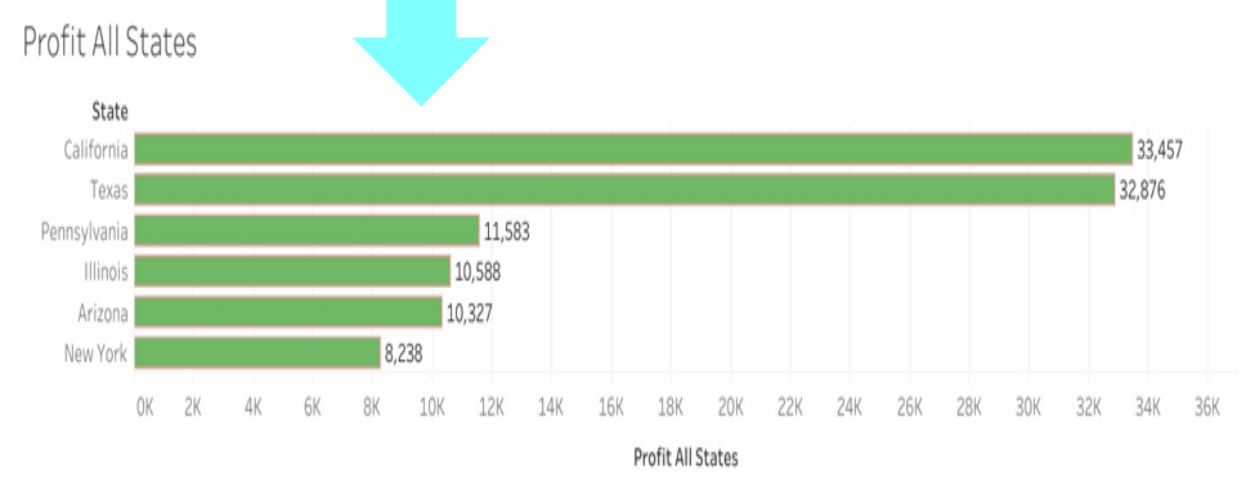
Run all | Undo | Publish | Outline | Attach to getsha

Not started

```
#Revenue across all states
#Import data type libraries and functions
from pyspark.sql.types import*
from pyspark.sql.functions import *

df = spark.read.load('abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/superstore2c
format= 'csv',
header = True
)

df_sales = df.groupBy('State').agg(round(sum((df['Sales'] * df['Quantity'] - df['Sales'] * df['Dis
display(df_sales)

# Saving data back to the datalake
path = '/UsefulInsights/RevenuePerState'
df_sales.write.mode("Overwrite").csv(path, header = True)
print('Data Successfully Saved')
```

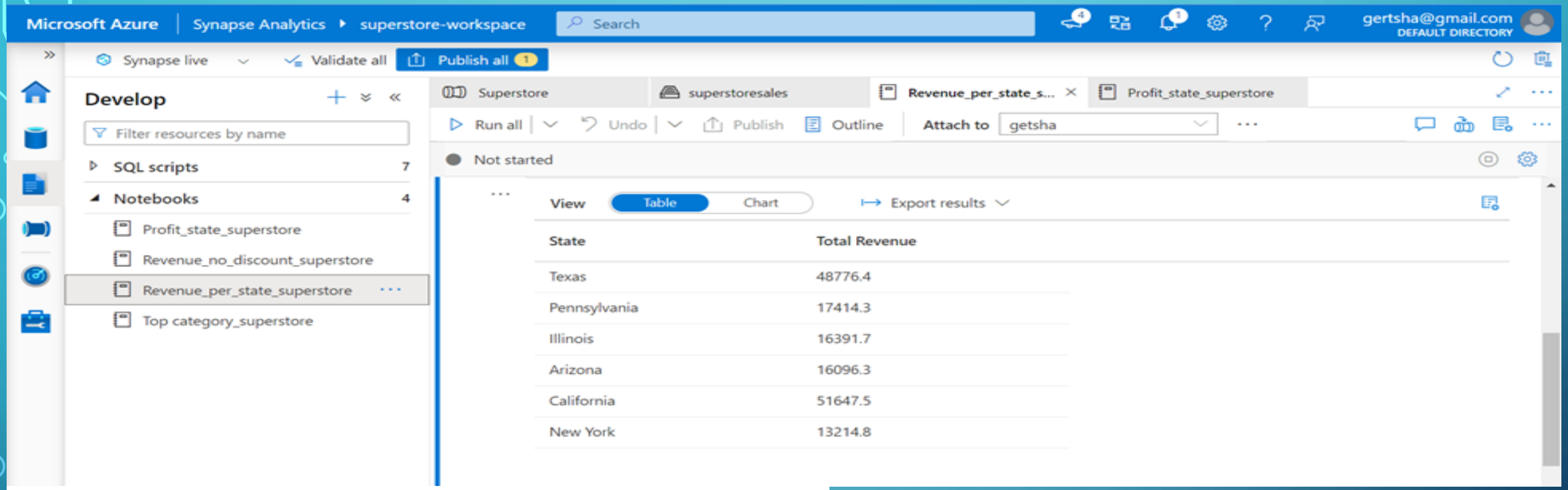
[3] - Command executed in 1 sec 69 ms on 3:52:48 PM, 5/03/24

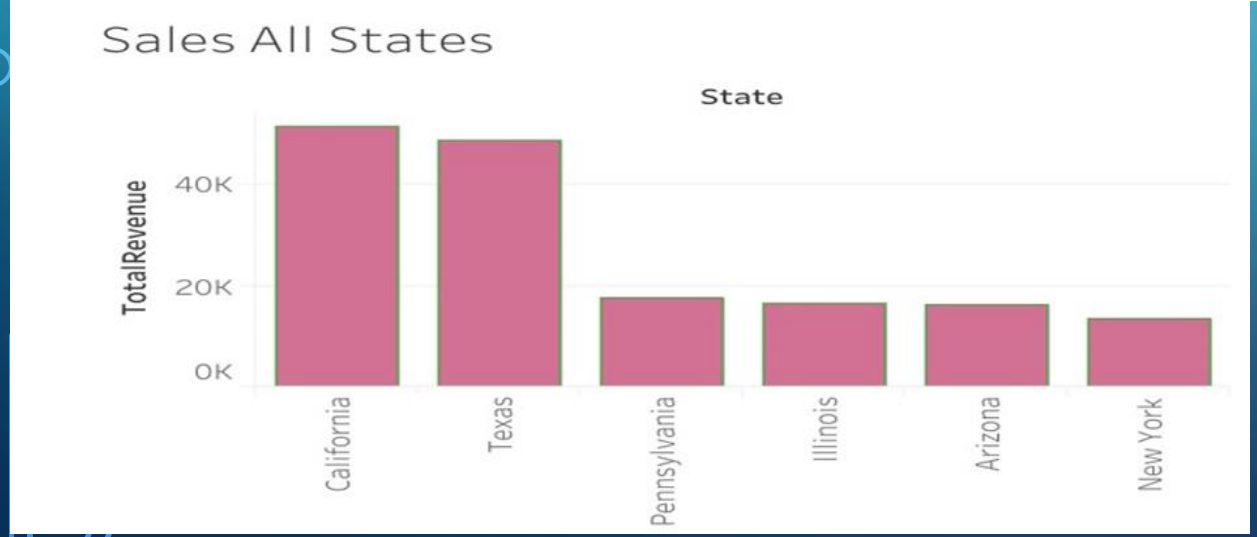
Result



- The state of California displays the highest profit.
- The state of New York shows the lowest overall profit.

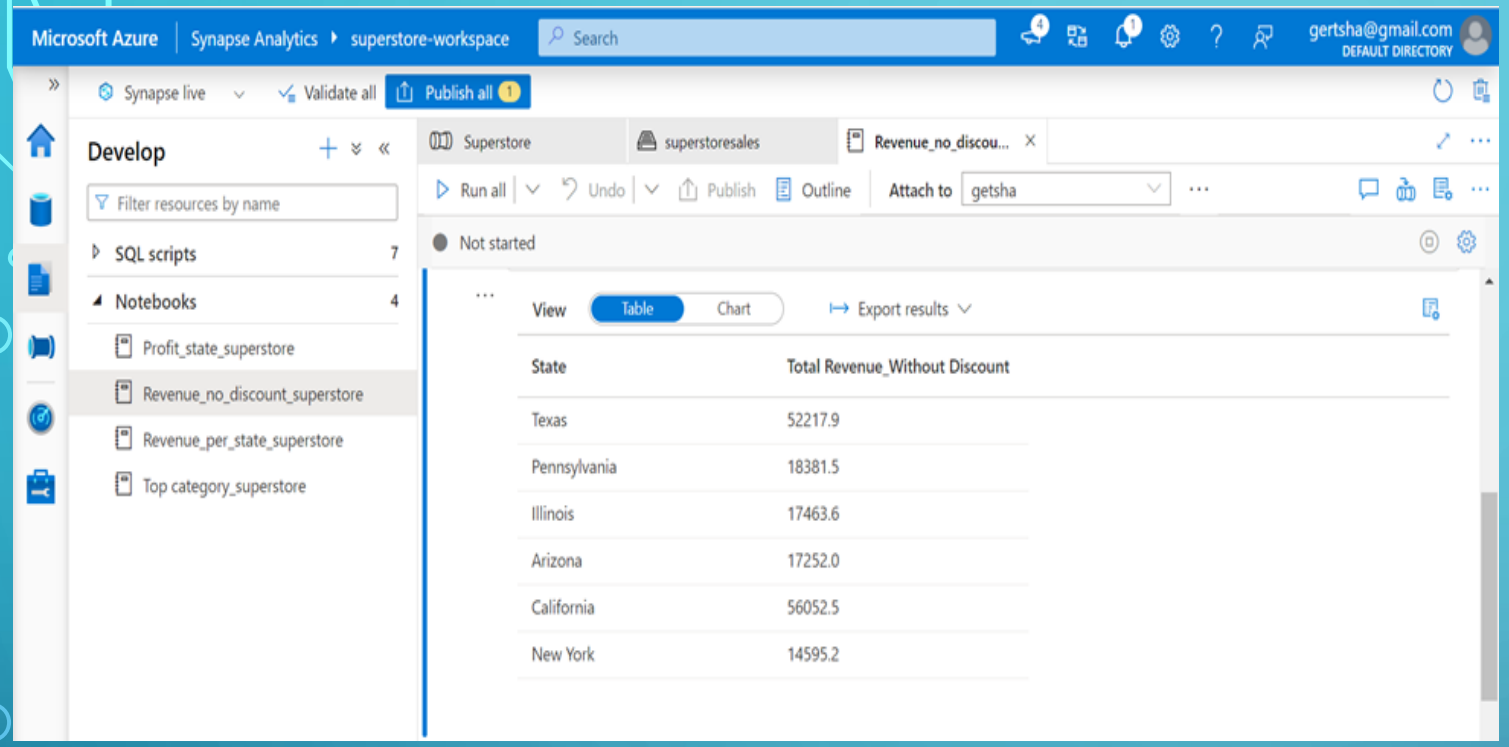
DATA TRANSFORMATION (PYTHON)





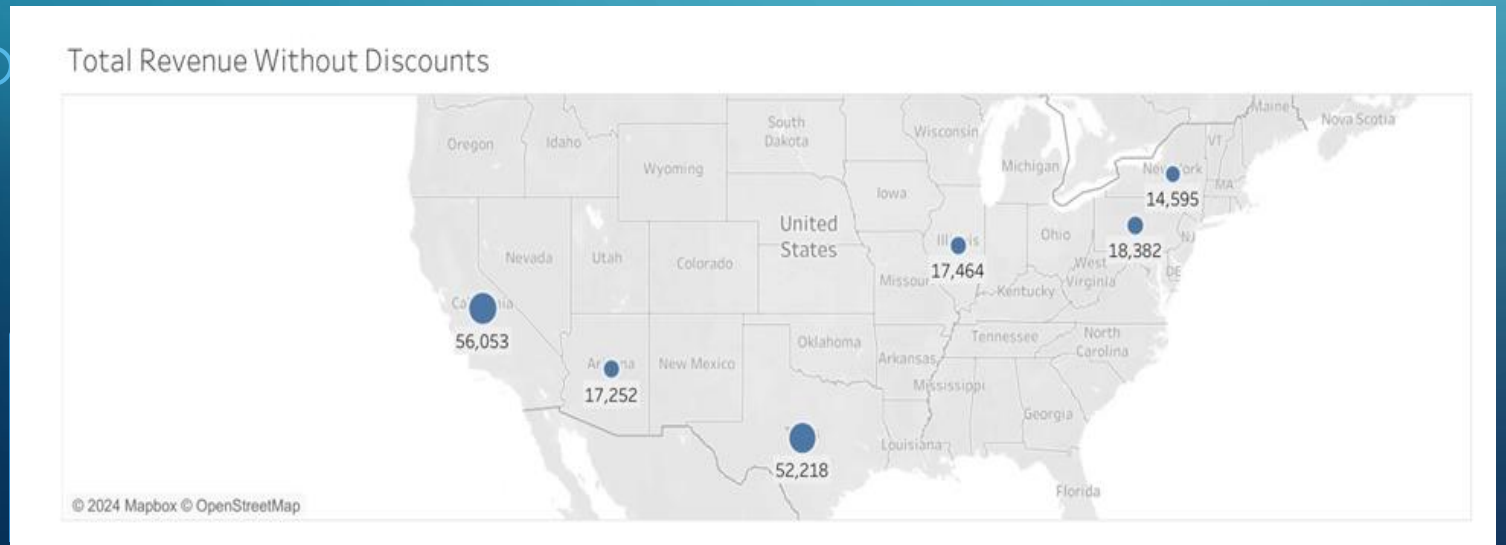
- Results still maintain California as the leading profitable state and New York as the least profitable state.
- Profit across all states as visualized on tableau

DATA TRANSFORMATION (PYTHON)



State	Total Revenue_Without Discount
Texas	52217.9
Pennsylvania	18381.5
Illinois	17463.6
Arizona	17252.0
California	56052.5
New York	14595.2

- In the absence of discounts, the state of Texas shows the highest profit while the state of New York shows the lowest profit.



- Profit if he did not give any discounts across all states as visualized on tableau:

DATA TRANSFORMATION (PYTHON)

Microsoft Azure | Synapse Analytics ▸ superstore-workspace

Search

gertsha@gmail.com
DEFAULT DIRECTORY

Synapse live | Validate all | Publish all 2

Develop

Filter resources by name

SQL scripts 7

Notebooks 4

- Profit_state_superstore
- Revenue_no_discount_superstore
- Revenue_per_state_superstore
- Top category_superstore

Superstore | superstoresales | Top category_super...

Run all | Undo | Publish | Outline | Attach to getsha

Not started

```
#Top Category
#Import data type libraries and functions
from pyspark.sql.types import*
from pyspark.sql.functions import *
df = spark.read.load('abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/superstore2c
format= 'csv',
header = True
)
df_topcategory = df.groupBy('Category').agg(round(sum((df['Sales'] * df['Quantity']) - (df['Sales'
display(df_topcategory.limit(1))
#OR Top Category
#from pyspark.sql.functions import sum, round, desc
#df_topcategory = df.groupBy('Category').agg(round(sum((df['Sales'] * df['Quantity']) - (df['Sales
#display(df_topcategory.limit(1))

# Save data back to the datalake (only the top category)
top_category = df_topcategory.limit(1)
path = '/UsefulInsights/TopCategory'
top_category.write.mode("Overwrite").csv(path, header=True)
print('Data Successfully Saved')
```

Microsoft Azure | Synapse Analytics ▸ superstore-workspace

Search

gertsha@gmail.com
DEFAULT DIRECTORY

Synapse live | Validate all | Publish all 2

Develop

Filter resources by name

SQL scripts 7

Notebooks 5

- Notebook 1
- Profit_state_superstore
- Revenue_no_discount_superstore
- Revenue_per_state_superstore
- Top category_superstore

Superstore | superstoresales | Notebook 1

Run all | Undo | Publish | Outline | Attach to getsha

Ready

[1] 3 min 41 seapache Spark session started in 3 min 20 sec 573 ms. Command executed in 20 sec 859 ms by gertsha on 11:39:39 PM, 5/07.

Job execution Succeeded Spark 2 executors 8 cores View in monitoring Open Spark UI

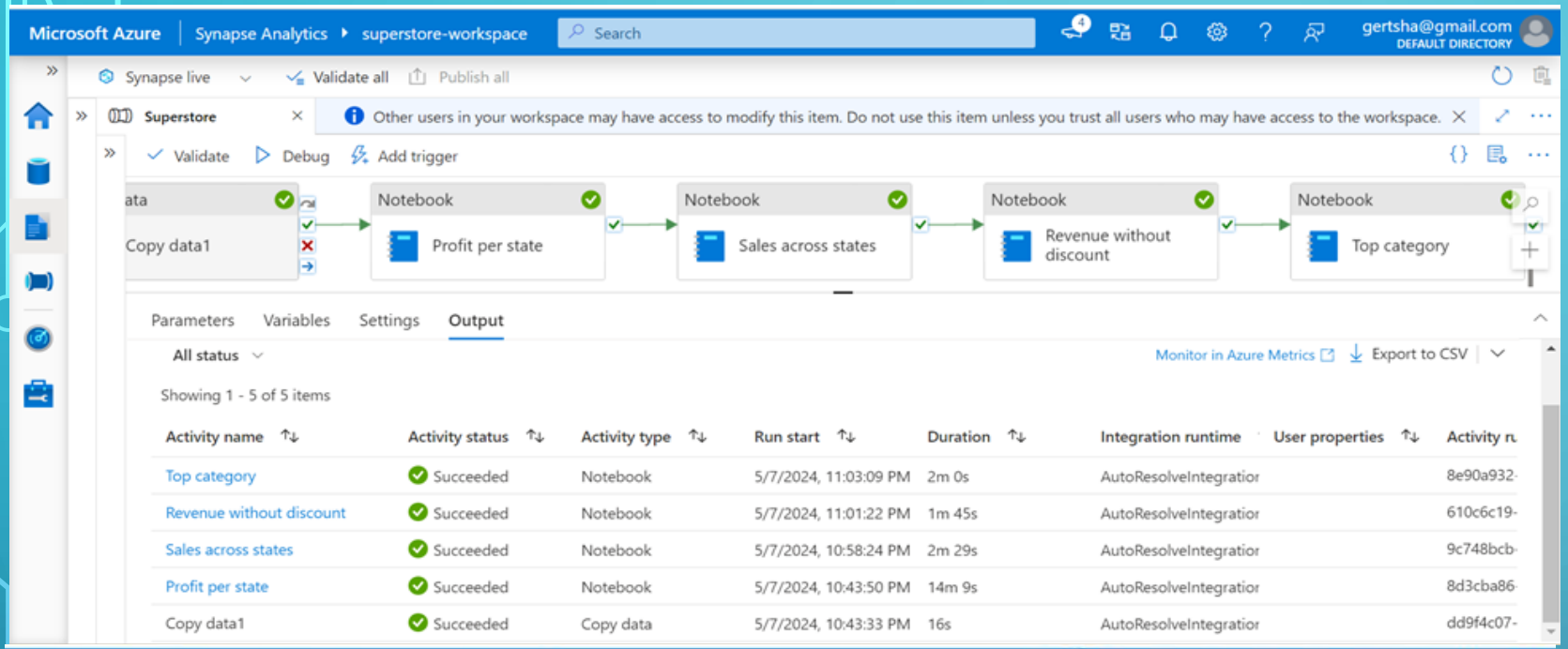
View Table Chart Export results

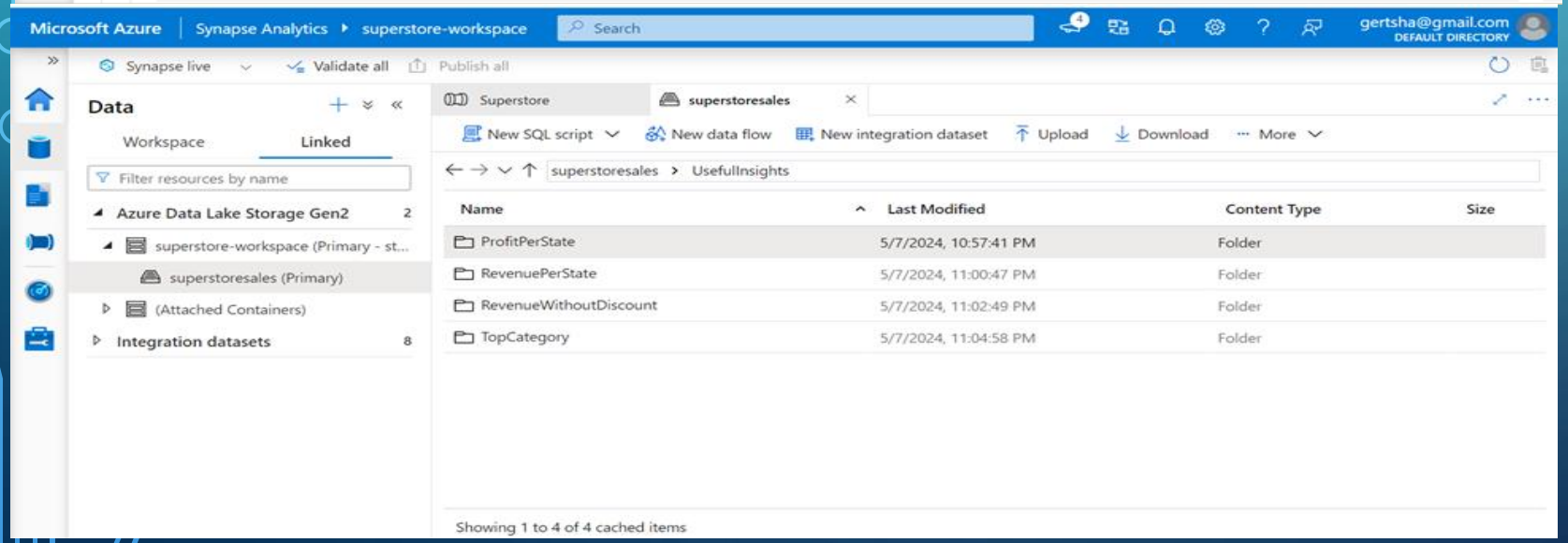
Category	Top Category
Office Supplies	88095.6

Which is the best performing category based off the sales?

- Office suppliers outperforms other categories as the best performing category.

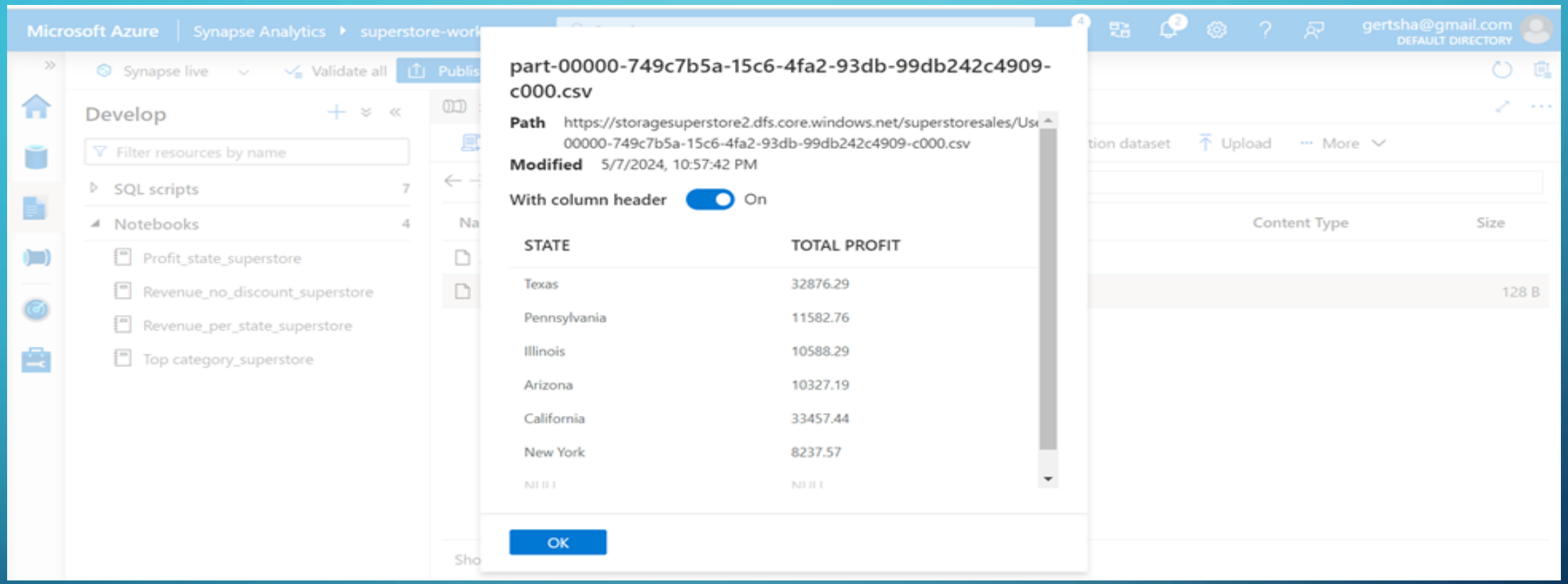
DATA AUTOMATION





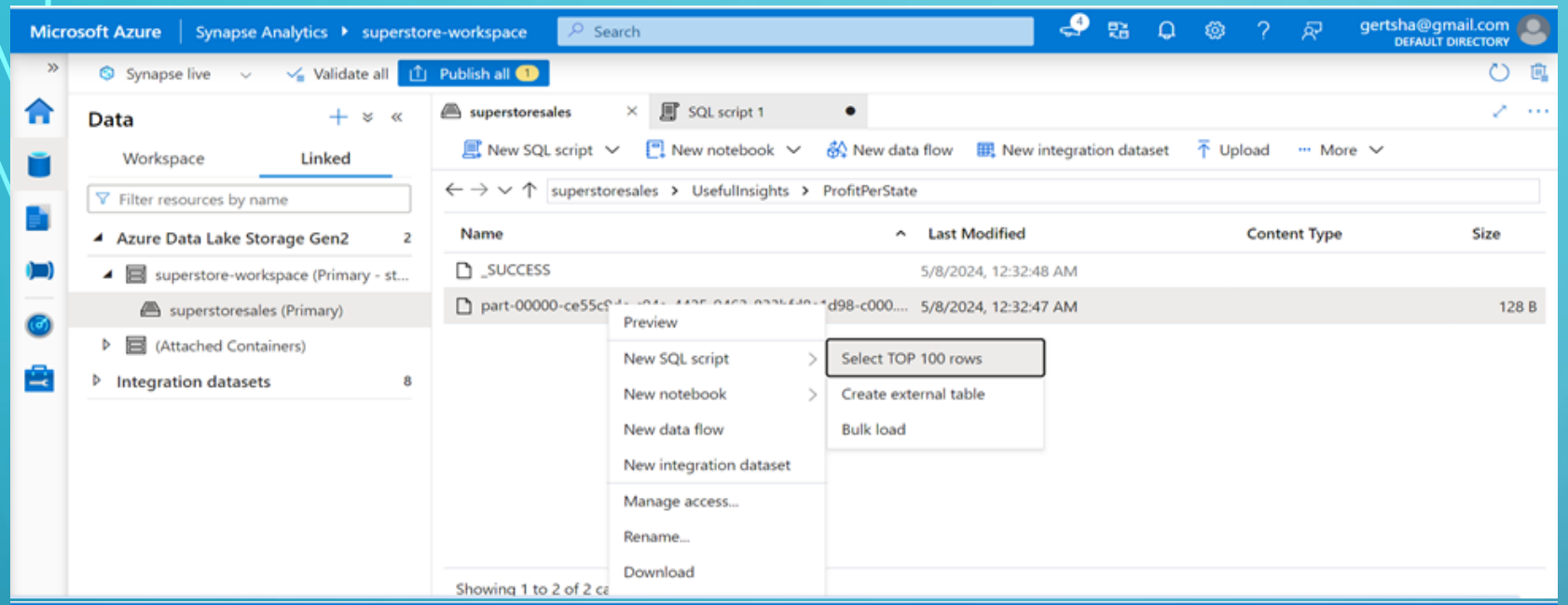
- We implemented a data pipeline to run the above transformation and ensure that the above files are saved in the data lake.
- We ran a debug and then triggered the pipeline.

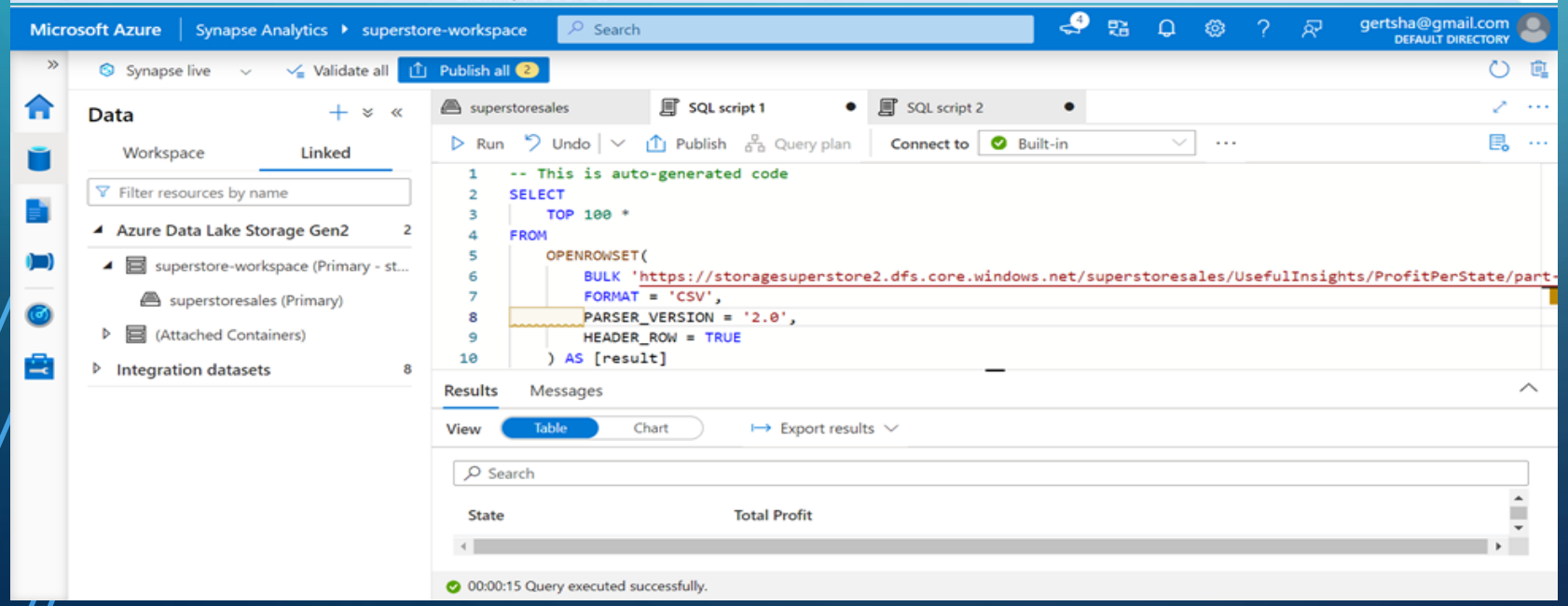
DATA AUTOMATION



- As shown on the diagram on the left, data preview on saved data is possible

DATA AUTOMATION





- You can also run SQL scripts to show output for each result saved back in the datalake as you can see.

CONSTRAINTS

- We encountered some challenges along the way like the below error which implied overuse of resources. We tried to resolve it in various ways, but in vain:

AVAILABLE_WORKSPACE_CAPACITY_EXCEEDED: Livy session has failed. Session state: Error. Error code: AVAILABLE_WORKSPACE_CAPACITY_EXCEEDED. Your job requested 12 vcores.

- There was a time constraint on all members of the group as we are in 3 different time zones, and this limited our time working on the project.

CONCLUSION

- We were able to work around our challenges and ensured that the project was concluded in time.
- We had several resources available that facilitated conveyance of all our findings.
- We employed tools such as Tableau and Excel to visualize our data.
- For this project, Azure Synapse Analytics and Azure Data Lake Storage (ADLS) offered several benefits like **Scalability, Integration, Performance, Security and Serverless Capabilities**

RECOMMENDATIONS

- **Azure Data Factory (ADF)** is a suitable alternative we would recommend for this project as it also offers flexibility, scalability, and integration capabilities required to successfully implement the data engineering tasks outlined in the project.

- **Azure DevOps** can be implemented for continuous integration and deployment (CI/CD) of data engineering pipelines.

- **Azure Monitor can be used to m**onitor the performance, availability, and usage of data engineering pipelines and services.

THANK YOU