

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:

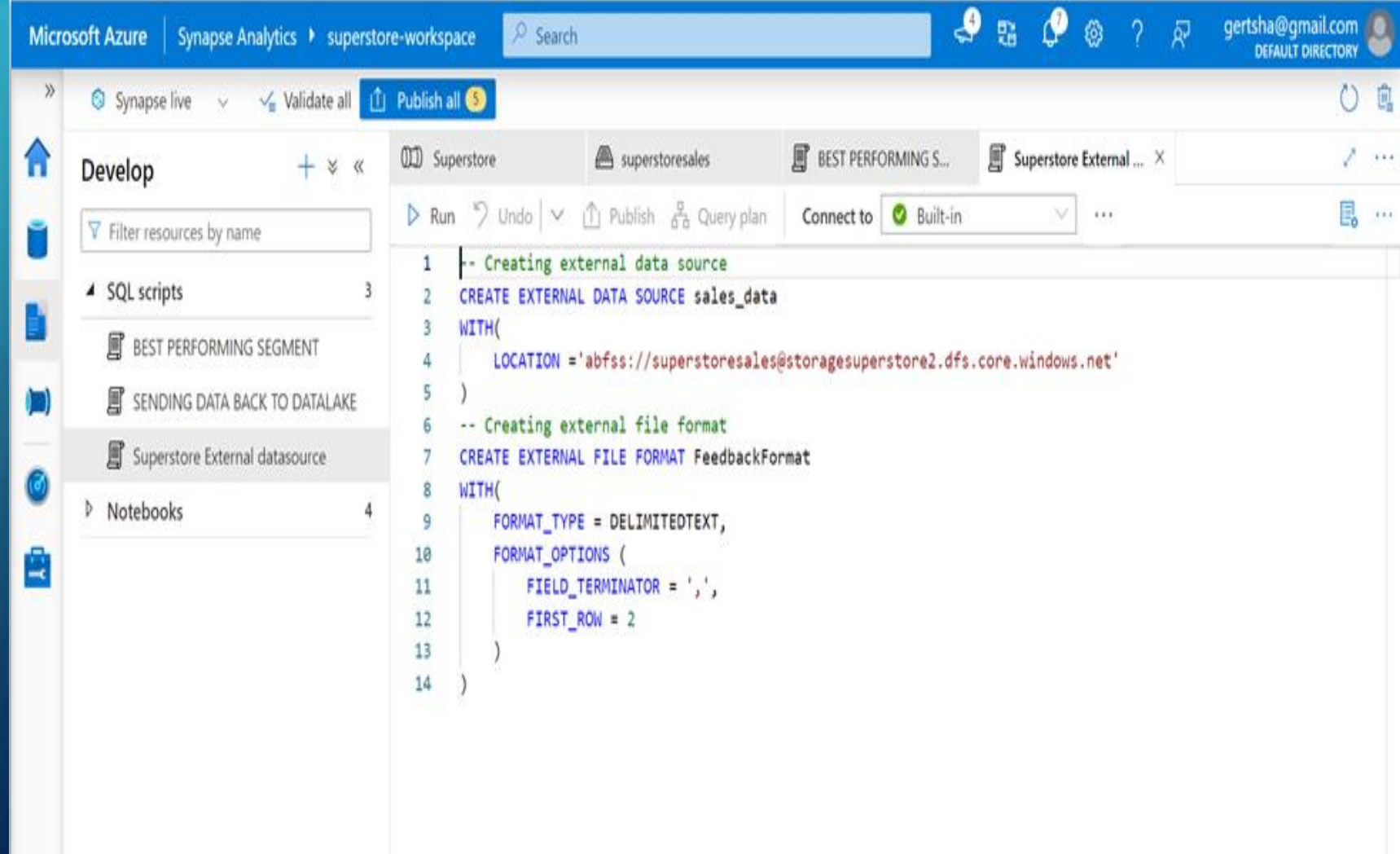
ErrorCode=DelimitedTextMoreColumnsThanDefined,'Type=Microsoft.DataTransfer.Common.Shared.HybridDeliveryException,Message=Error found when processing 'Csv/Tsv Format Text' source 'a28288a9-9fcb-406d-a9c3-2da00fd09254' with row number 34: found more columns than expected column count 21.,Source=Microsoft.DataTransfer.Common,'

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

The screenshot displays the Microsoft Azure Synapse Analytics workspace interface. At the top, the header shows 'Microsoft Azure | Synapse Analytics | superstore-workspace' with a search bar and user information 'gertsha@gmail.com | DEFAULT DIRECTORY'. Below the header, the 'Develop' tab is active, showing a list of SQL scripts on the left: 'BEST PERFORMING SEGMENT', 'SENDING DATA BACK TO DATALAKE', and 'SQL scripts' (3 items). The main editor area shows a SQL script for creating an external table named 'sales_records'. The script is as follows:

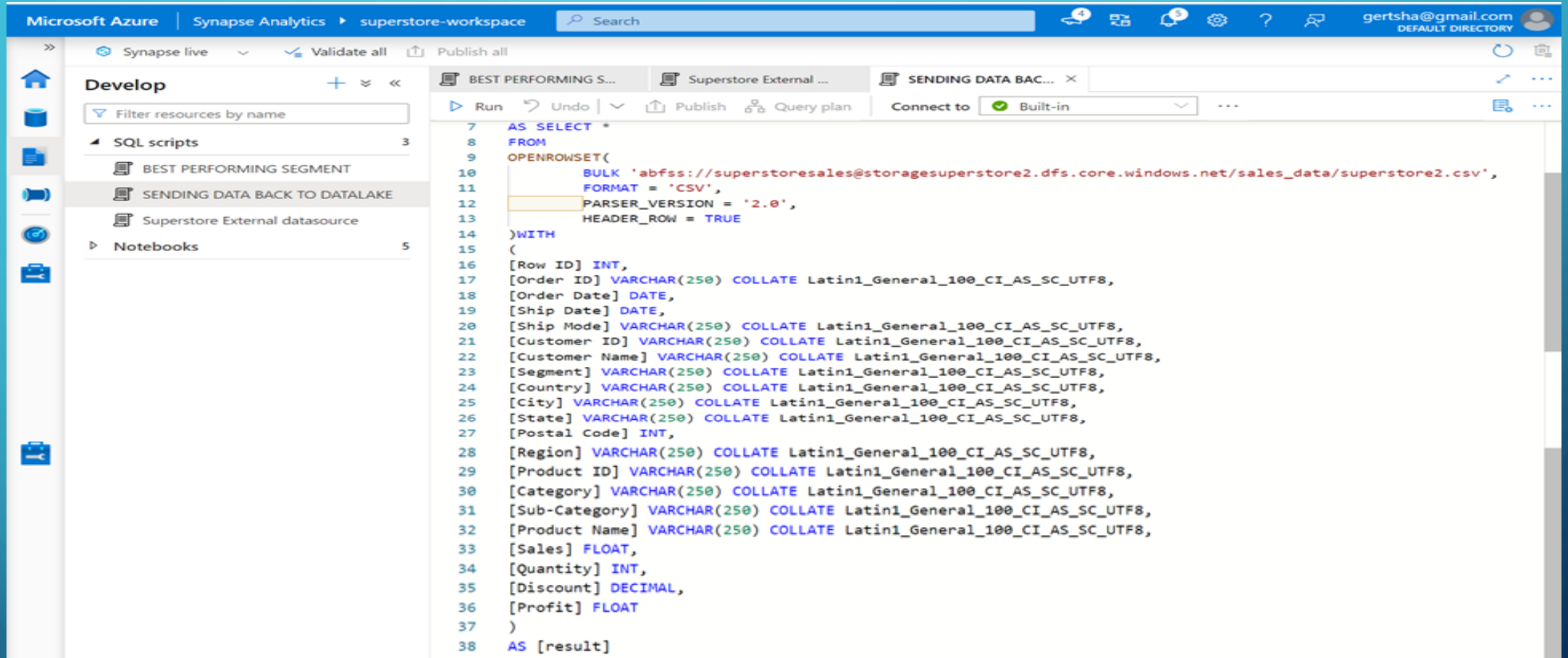
```
1 CREATE EXTERNAL TABLE sales_records
2 WITH(
3     LOCATION = 'superstore2cleaned.csv',
4     DATA_SOURCE = sales_data,
5     FILE_FORMAT = FeedbackFormat
6 )
```

The interface also includes a toolbar with buttons for 'Run', 'Undo', 'Publish', 'Query plan', and 'Connect to' (set to 'Built-in').

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



The screenshot displays the Microsoft Azure Synapse Analytics workspace 'superstore-workspace'. The left sidebar shows the 'Develop' section with a list of resources: 'SQL scripts' (3), 'BEST PERFORMING SEGMENT', 'SENDING DATA BACK TO DATALAKE', 'Superstore External datasource', and 'Notebooks' (5). The main editor area shows a SQL script titled 'BEST PERFORMING S...' with the following content:

```
7 AS SELECT *
8 FROM
9 OPENROWSET(
10     BULK 'abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/sales_data/superstore2.csv',
11     FORMAT = 'CSV',
12     PARSER_VERSION = '2.0',
13     HEADER_ROW = TRUE
14 )WITH
15 (
16     [Row ID] INT,
17     [Order ID] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
18     [Order Date] DATE,
19     [Ship Date] DATE,
20     [Ship Mode] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
21     [Customer ID] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
22     [Customer Name] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
23     [Segment] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
24     [Country] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
25     [City] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
26     [State] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
27     [Postal Code] INT,
28     [Region] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
29     [Product ID] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
30     [Category] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
31     [Sub-Category] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
32     [Product Name] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
33     [Sales] FLOAT,
34     [Quantity] INT,
35     [Discount] DECIMAL,
36     [Profit] FLOAT
37 )
38 AS [result]
```

- 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.

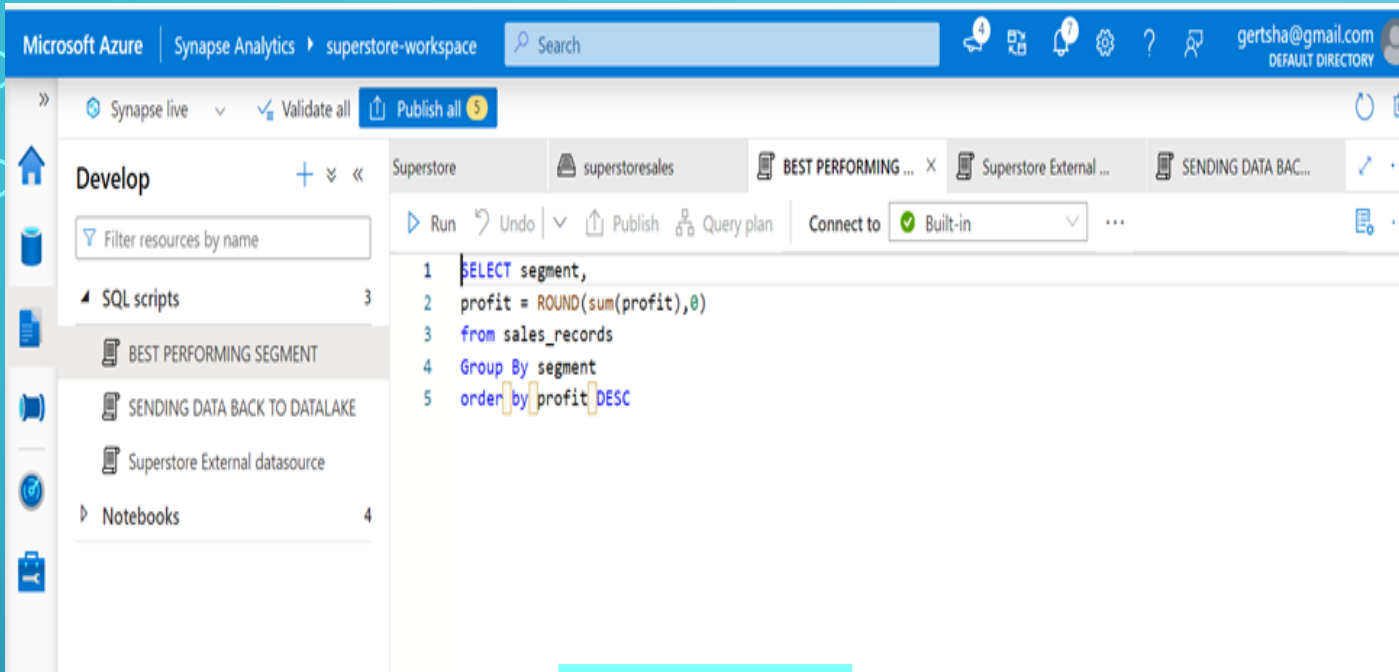
DATA CLEANING USING SQL

The screenshot displays the Microsoft Azure Synapse Analytics interface. The top navigation bar shows 'Microsoft Azure | Synapse Analytics | superstore-workspace'. The left sidebar, titled 'Data', shows a tree view of resources. Under 'SQL database', there is a folder 'superstore (SQL)' which contains 'External tables' (with 'dbo.sales_records' listed), 'External resources', 'External data sources' (with 'sales_data' listed), 'External file formats' (with 'FeedbackFormat' listed), 'Views', and 'Schemas'. The main pane shows a SQL script being executed. The script is as follows:

```
1 CREATE EXTERNAL TABLE sales_records
2 WITH(
3     LOCATION = 'superstore2cleaned.csv',
4     DATA_SOURCE = sales_data,
5     FILE_FORMAT = FeedbackFormat
6 )
7 AS SELECT *
8 FROM
9 OPENROWSET(
10     BULK 'abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/sales_data/superstore2.csv',
11     FORMAT = 'CSV',
12     PARSER_VERSION = '2.0',
13     HEADER_ROW = TRUE
14 )WITH
15 (
16 [Row ID] INT,
17 [Order ID] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
18 [Order Date] DATE,
19 [Ship Date] DATE,
20 [Ship Mode] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
21 [Customer ID] VARCHAR(250) COLLATE Latin1_General_100_CI_AS_SC_UTF8,
```

- 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



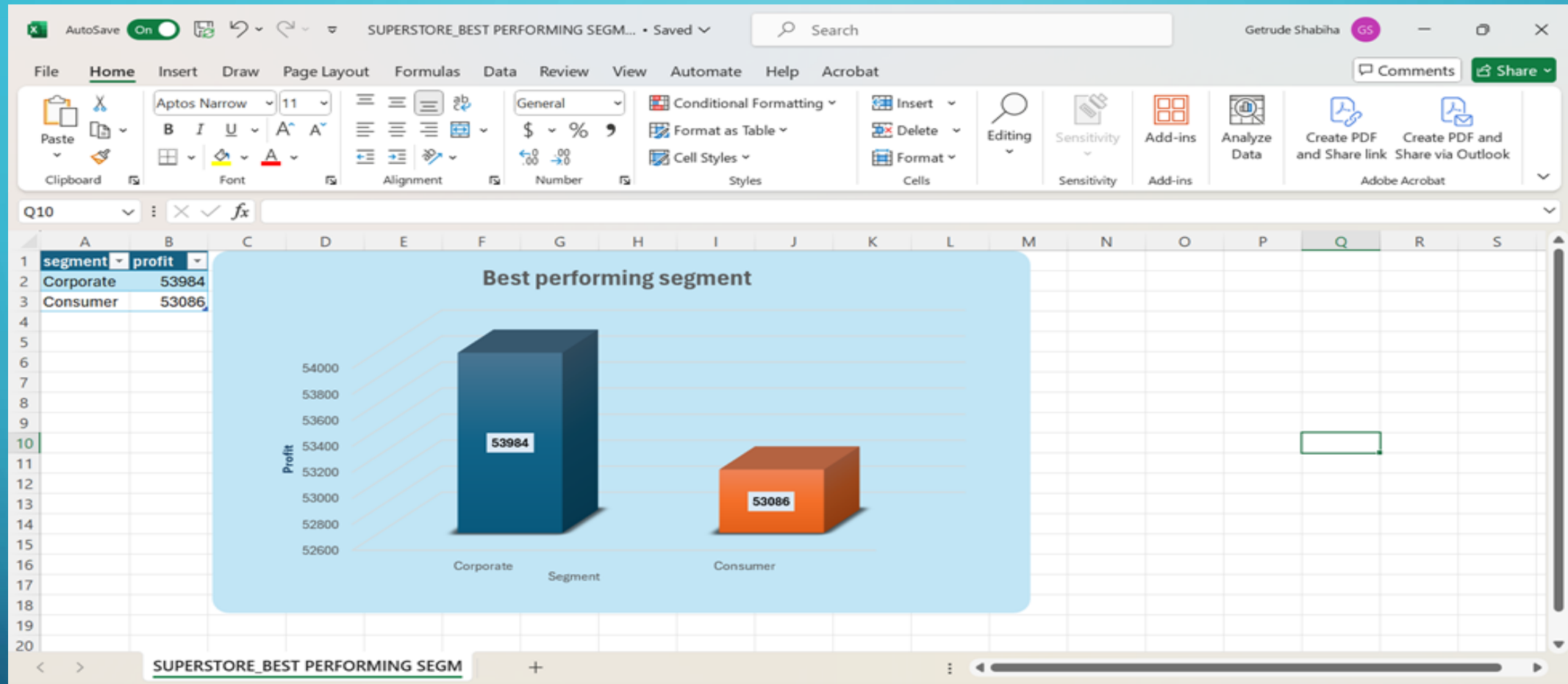
```
1 SELECT segment,
2 profit = ROUND(sum(profit),0)
3 from sales_records
4 Group By segment
5 order by profit DESC
```

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

$$\begin{aligned}\text{Revenue} &= (\text{Unit Price} * \text{Quantity}) - (\text{Unit Price} * \text{Discount}) \\ &= (3.5 * 4) - (3.5 * 0.52) \\ &= 14 - 1.82 \\ &= 12.18\end{aligned}$$

$$\begin{aligned}\text{Profit} &= \text{Revenue} - \text{Cost} \\ \text{Cost} &= \text{Revenue} - \text{Profit}\end{aligned}$$

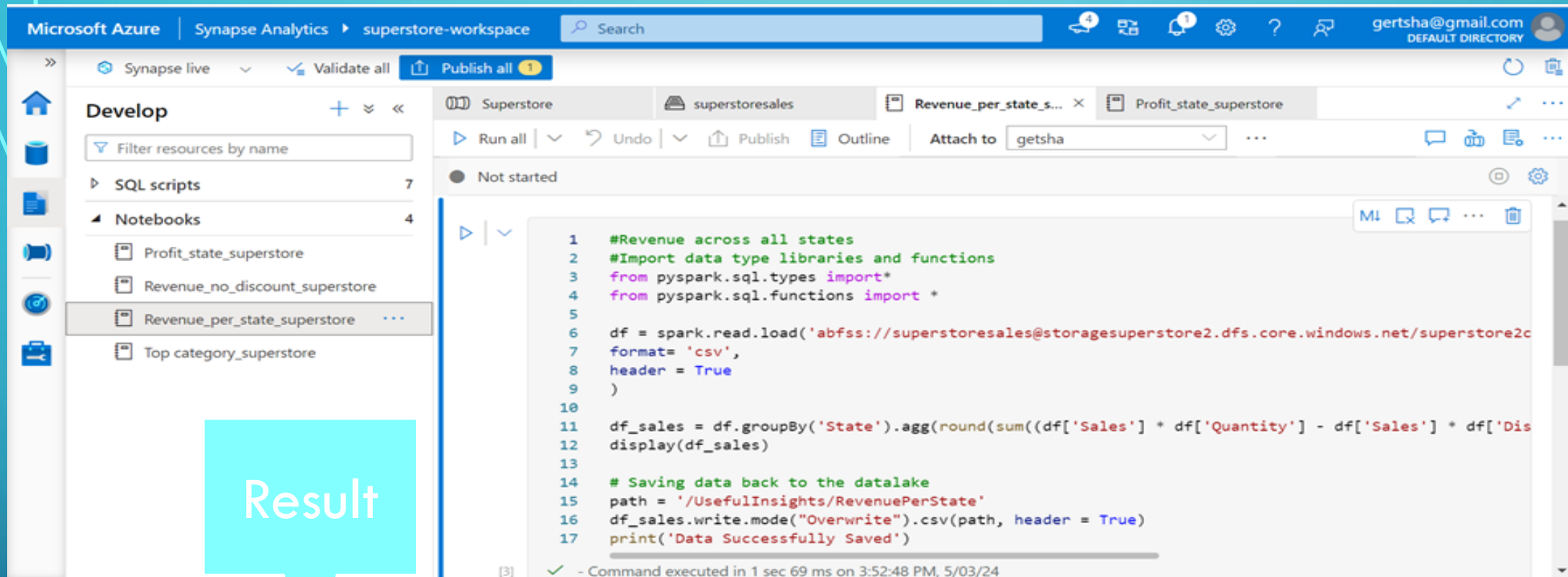
The screenshot shows the Microsoft Azure Synapse Analytics interface. The left sidebar displays the 'Develop' section with a list of notebooks: 'Profit_state_superstore', 'Revenue_no_discount_superstore', 'Revenue_per_state_superstore', and 'Top category_superstore'. The main area shows a notebook titled 'Profit_state_superstore' with the following Python code:

```
1 #Import data type libraries and functions
2 from pyspark.sql.types import *
3 from pyspark.sql.functions import *
4
5 #Fetching the transformed data
6 df = spark.read.load('abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/superstore2c
7 format= 'csv',
8 header = True
9 )
10 #display(df.limit(10))
11
12 df_profit = df.groupBy('State').agg(round(sum('Profit'),2).alias('Total Profit'))
13 display(df_profit)
14
15 # Saving data back to the datalake
16 path = '/UsefulInsights/ProfitPerState'
17 df_profit.write.mode("Overwrite").csv(path, header = True)
18 print('Data Successfully Saved')
```

The status bar at the bottom indicates: '[1] ✓ - Apache Spark session started in 2 min 44 sec 93 ms. Command executed in 33 sec 504 ms on 6:51:29 PM, 5/04/24'.

- 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)



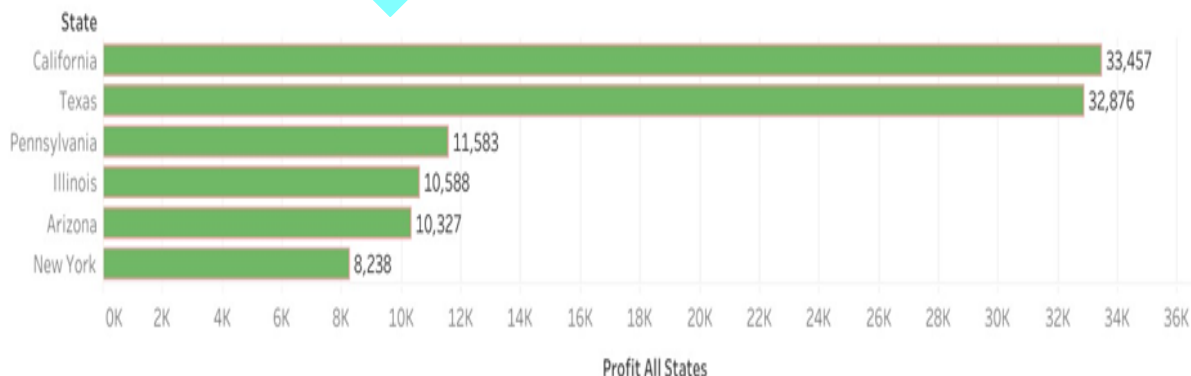
The screenshot shows the Microsoft Azure Synapse Analytics interface. The left sidebar displays the 'Develop' section with a list of notebooks, including 'Profit_state_superstore', 'Revenue_no_discount_superstore', 'Revenue_per_state_superstore', and 'Top category_superstore'. The main area shows a notebook titled 'Revenue_per_state_s...' with a Python script being executed. The script calculates the profit for each state by grouping the data by state and applying a formula: $\text{Profit} = (\text{Sales} \times \text{Quantity}) - \text{Sales}$. The results are saved to a CSV file in the datalake. The status bar indicates the command was executed successfully in 1 second and 69 milliseconds on 5/03/24.

```
1 #Revenue across all states
2 #Import data type libraries and functions
3 from pyspark.sql.types import*
4 from pyspark.sql.functions import *
5
6 df = spark.read.load('abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/superstore2c
7 format= 'csv',
8 header = True
9 )
10
11 df_sales = df.groupBy('State').agg(round(sum((df['Sales'] * df['Quantity'] - df['Sales'] * df['Dis
12 display(df_sales)
13
14 # Saving data back to the datalake
15 path = '/UsefulInsights/RevenuePerState'
16 df_sales.write.mode("Overwrite").csv(path, header = True)
17 print('Data Successfully Saved')
```

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

Result

Profit All States



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

DATA TRANSFORMATION (PYTHON)

Microsoft Azure | Synapse Analytics | superstore-workspace

Search

Synapse live | Validate all | Publish all

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

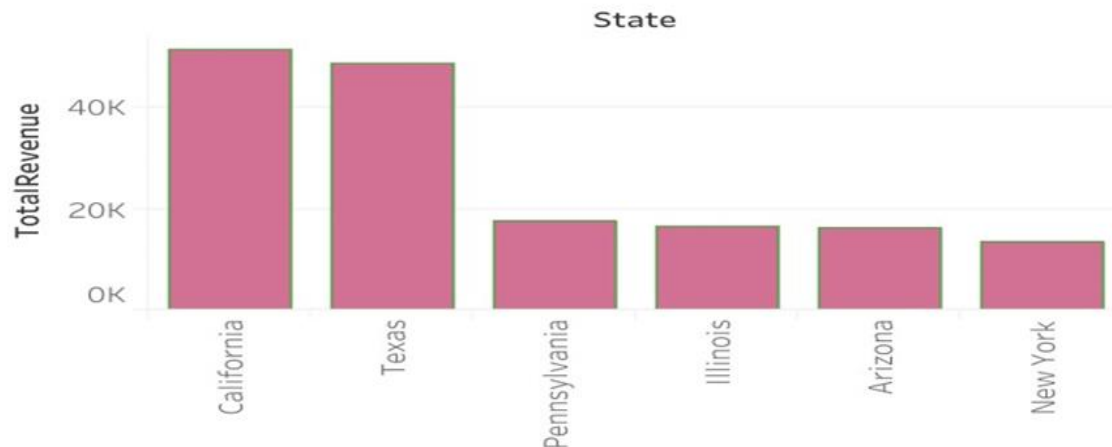
Run all | Undo | Publish | Outline | Attach to: getsha

Not started

View: Table | Chart | Export results

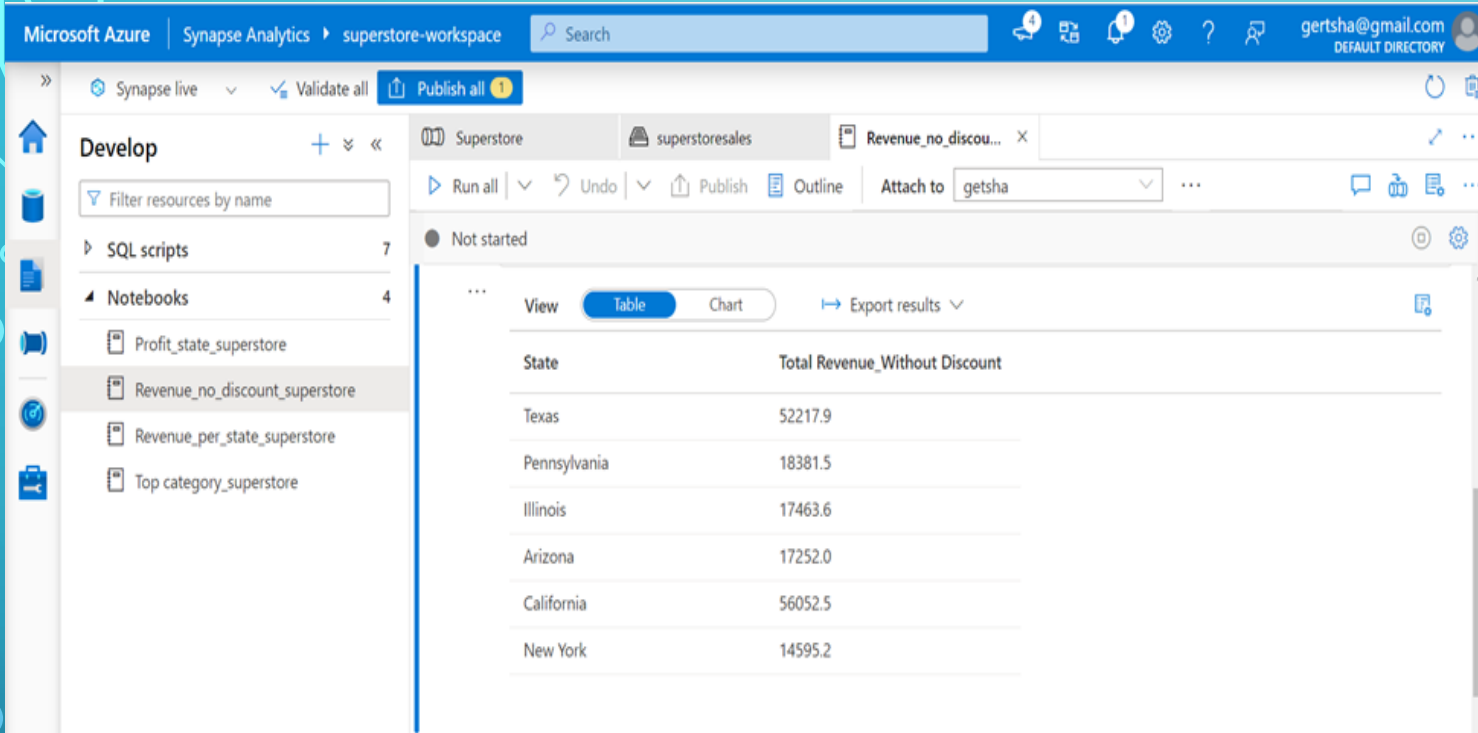
State	Total Revenue
Texas	48776.4
Pennsylvania	17414.3
Illinois	16391.7
Arizona	16096.3
California	51647.5
New York	13214.8

Sales All States



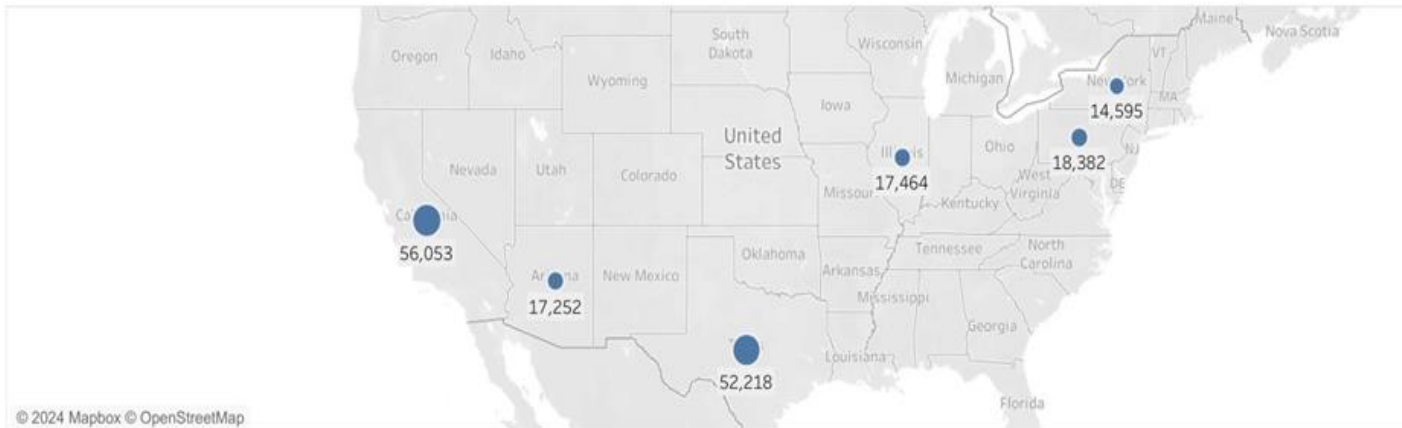
- Results still maintain California as the leading profitable state and New York as the least profitable state.
- Profit across all states as visualized on tableau

DATA TRANSFORMATION (PYTHON)



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

Total Revenue Without Discounts



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

DATA TRANSFORMATION (PYTHON)

Which is the best performing category based off the sales?

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

The screenshot displays the Microsoft Azure Synapse Analytics interface. The top navigation bar shows 'Microsoft Azure | Synapse Analytics | superstore-workspace'. The left sidebar contains a 'Develop' section with a search bar and a list of resources: 'SQL scripts' (7), 'Notebooks' (4), 'Profit_state_superstore', 'Revenue_no_discount_superstore', 'Revenue_per_state_superstore', and 'Top category_superstore'. The main workspace area shows a notebook titled 'Top category_super...' with a 'Run all' button and a 'Publish' button. The notebook content is a Python script that reads data from a storage account, processes it to find the top category by sales, and saves the results back to the datalake. The script is as follows:

```
1 #Top Category
2 #Import data type libraries and functions
3 from pyspark.sql.types import *
4 from pyspark.sql.functions import *
5 df = spark.read.load('abfss://superstoresales@storagesuperstore2.dfs.core.windows.net/superstore2c
6 format= 'csv',
7 header = True
8 )
9 df_topcategory = df.groupBy('Category').agg(round(sum((df['Sales'] * df['Quantity']) - (df['Sales'
10 display(df_topcategory.limit(1))
11 #OR Top Category
12 #from pyspark.sql.functions import sum, round, desc
13 #df_topcategory = df.groupBy('Category').agg(round(sum((df['Sales'] * df['Quantity']) - (df['Sales
14 #display(df_topcategory.limit(1))
15
16 # Save data back to the datalake (only the top category)
17 top_category = df_topcategory.limit(1)
18 path = '/UsefulInsights/TopCategory'
19 top_category.write.mode("Overwrite").csv(path, header=True)
20 print('Data Successfully Saved')
```

The bottom section of the screenshot shows the notebook execution results. The status is 'Ready'. The output indicates that the job execution succeeded, with a Spark session started in 3 min 20 sec 573 ms and 2 executors using 8 cores. The results are displayed in a table:

Category	Top Category
Office Supplies	88095.6

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.

The screenshot displays the Microsoft Azure Synapse Analytics interface for a workspace named 'superstore-workspace'. The top navigation bar includes 'Synapse live', 'Validate all', and 'Publish all' buttons. The main area shows a data pipeline with five activities: 'Copy data1', 'Notebook Profit per state', 'Notebook Sales across states', 'Notebook Revenue without discount', and 'Notebook Top category'. All activities are marked as 'Succeeded' with green checkmarks. Below the pipeline, the 'Output' tab is selected, showing a table of activity details.

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity ID
Top category	✓ Succeeded	Notebook	5/7/2024, 11:03:09 PM	2m 0s	AutoResolveIntegrator		8e90a932-
Revenue without discount	✓ Succeeded	Notebook	5/7/2024, 11:01:22 PM	1m 45s	AutoResolveIntegrator		610c6c19-
Sales across states	✓ Succeeded	Notebook	5/7/2024, 10:58:24 PM	2m 29s	AutoResolveIntegrator		9c748bcb-
Profit per state	✓ Succeeded	Notebook	5/7/2024, 10:43:50 PM	14m 9s	AutoResolveIntegrator		8d3cba86-
Copy data1	✓ Succeeded	Copy data	5/7/2024, 10:43:33 PM	16s	AutoResolveIntegrator		dd9f4c07-

Below the pipeline view, the 'Data' section is shown, displaying a list of resources in the 'superstore-workspace'. The 'Linked' tab is selected, showing a list of folders in the 'superstore-sales' integration dataset. The folders are: 'ProfitPerState', 'RevenuePerState', 'RevenueWithoutDiscount', and 'TopCategory', all created on 5/7/2024.

Name	Last Modified	Content Type	Size
ProfitPerState	5/7/2024, 10:57:41 PM	Folder	
RevenuePerState	5/7/2024, 11:00:47 PM	Folder	
RevenueWithoutDiscount	5/7/2024, 11:02:49 PM	Folder	
TopCategory	5/7/2024, 11:04:58 PM	Folder	

DATA AUTOMATION

The screenshot displays the Microsoft Azure Synapse Analytics web interface. On the left, the 'Develop' section shows a list of notebooks, including 'Profit_state_superstore', 'Revenue_no_discount_superstore', 'Revenue_per_state_superstore', and 'Top_category_superstore'. The main area shows a preview of a CSV file named 'part-00000-749c7b5a-15c6-4fa2-93db-99db242c4909-c000.csv'. The file's path is 'https://storagesuperstore2.dfs.core.windows.net/superstoresales/Us...'. The 'Modified' date is '5/7/2024, 10:57:42 PM'. The 'With column header' toggle is set to 'On'. The preview shows a table with two columns: 'STATE' and 'TOTAL PROFIT'. The data rows are: Texas (32876.29), Pennsylvania (11582.76), Illinois (10588.29), Arizona (10327.19), California (33457.44), and New York (8237.57). The table ends with 'NULL' for both columns. An 'OK' button is at the bottom of the preview window. In the background, another window shows a table with 'Content Type' and 'Size' columns, with a row showing '128 B'.

Microsoft Azure | Synapse Analytics | superstore-work

Synapse live | Validate all | Publish

Develop

Filter resources by name

SQL scripts 7

Notebooks 4

- Profit_state_superstore
- Revenue_no_discount_superstore
- Revenue_per_state_superstore
- Top_category_superstore

part-00000-749c7b5a-15c6-4fa2-93db-99db242c4909-c000.csv

Path https://storagesuperstore2.dfs.core.windows.net/superstoresales/Us...
00000-749c7b5a-15c6-4fa2-93db-99db242c4909-c000.csv

Modified 5/7/2024, 10:57:42 PM

With column header ☒ On

STATE	TOTAL PROFIT
Texas	32876.29
Pennsylvania	11582.76
Illinois	10588.29
Arizona	10327.19
California	33457.44
New York	8237.57
NULL	NULL

OK

Content Type Size

128 B

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

DATA AUTOMATION

The top screenshot shows the Microsoft Azure Synapse Analytics interface. The left sidebar displays the 'Data' section with 'Workspace' and 'Linked' tabs. The 'Linked' tab is active, showing a list of resources including 'Azure Data Lake Storage Gen2', 'superstore-workspace (Primary - st...', 'superstoresales (Primary)', '(Attached Containers)', and 'Integration datasets'. The main pane shows the 'superstoresales' workspace with a file explorer view. A file named 'part-00000-ce55c...' is selected, and a context menu is open, showing options like 'Preview', 'New SQL script', 'New notebook', 'New data flow', 'New integration dataset', 'Manage access...', 'Rename...', and 'Download'. The 'New SQL script' option is highlighted, and a sub-menu is open, showing 'Select TOP 100 rows', 'Create external table', and 'Bulk load'.

The bottom screenshot shows the same interface, but the 'SQL script 1' tab is active. The script content is as follows:

```
1 -- This is auto-generated code
2 SELECT
3   TOP 100 *
4 FROM
5   OPENROWSET(
6     BULK 'https://storagesuperstore2.dfs.core.windows.net/superstoresales/UsefulInsights/ProfitPerState/part-00000-ce55c...',
7     FORMAT = 'CSV',
8     PARSER_VERSION = '2.0',
9     HEADER_ROW = TRUE
10  ) AS [result]
```

The 'Results' tab is active, showing a table with the following columns: 'State' and 'Total Profit'. The table is empty, and the status bar at the bottom indicates '00:00:15 Query executed successfully.'

- 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 monitor the performance, availability, and usage of data engineering pipelines and services.**



THANK YOU