END TO END

DATA ENGINEER PROJECT

ONLINE STORE DATA SALES ANALYSIS

INDEX

ABSTRACT

In this data engineering project, we present an end-to-end solution for processing and analysing online store data. The project begins by acquiring raw data from an online store, which is then processed and transformed through a series of steps. Firstly, utilizing Python scripts, the data is downloaded and loaded from on-premises servers into a SQL Server database. Subsequently, Azure Data factory is employed to seamlessly migrate the data from the SQL Server to Azure Blob Storage, leveraging the power of cloud-based storage solutions.

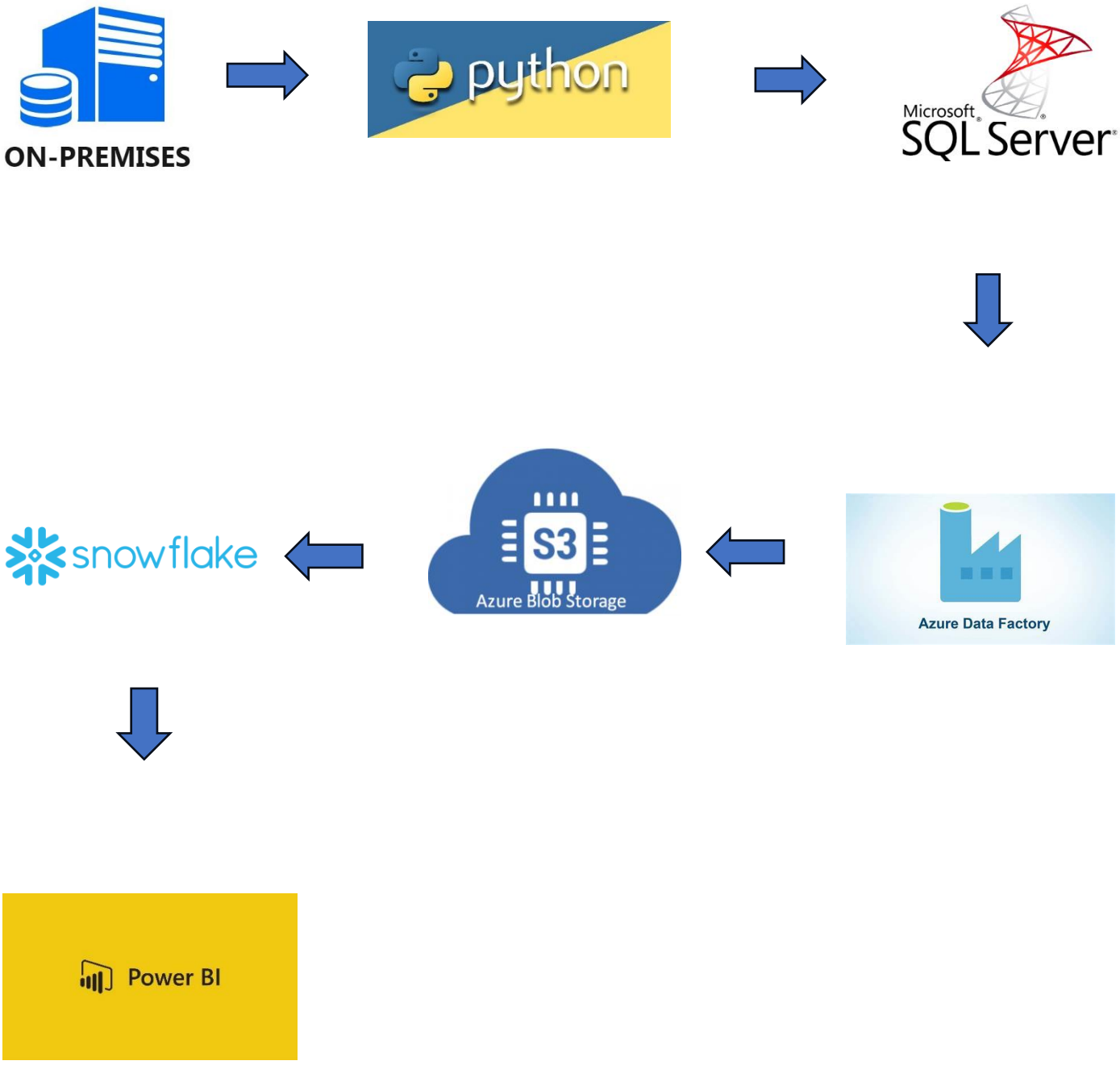
Once the data resides in Azure Blob Storage, it is efficiently loaded into Snowflake, a modern cloud-based data warehousing platform, utilizing external staging capabilities. During this transition, various transformations are applied to enhance the data quality and structure. These transformations are pivotal

for ensuring that the data is optimized for analysis and reporting purposes.

Finally, the processed and transformed data is loaded into Power BI, a robust business analytics service, where it is visually represented and analysed to create insightful reports. These reports offer valuable insights into the online store's performance, enabling data-driven decision-making processes for business stakeholders.

This project exemplifies a comprehensive and efficient data engineering workflow, encompassing data extraction, transformation, and loading (ETL) processes. By seamlessly integrating on-premises, cloud-based, and analytical tools, this solution provides a scalable and streamlined approach for handling and analysing large volumes of online store data, thereby empowering businesses to make informed decisions and gain a competitive edge in the e-commerce landscape.

PROJECT WORK FLOW



TOOLS REQUIRED FOR THE PROJECT

1.Python

Python is a high-level, interpreted programming language known for its simplicity and readability. It offers dynamic typing and dynamic binding, making it an excellent choice for scripting and rapid application development. Python supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural styles. It has a large and active community, extensive standard libraries, and third-party packages, making it a versatile language for various applications, including web development, data analysis, artificial intelligence

Pandas

Pandas is an open-source data analysis and manipulation library for Python. It

provides data structures such as Series and Data Frame, allowing you to efficiently manipulate and analyse large datasets. Pandas is particularly useful for tasks like cleaning, transforming, aggregating, and visualizing data. It integrates seamlessly with other libraries like NumPy and Matplotlib, making it a powerful tool for data analysis and exploration.

2.SQL Server Management Studio

SQL Server Management Studio (SSMS) is a free integrated development environment (IDE) provided by Microsoft for accessing, configuring, managing, and developing SQL Server databases. It provides a graphical interface and a set of tools for working with SQL Server databases, making it easier to create, edit, and manage database objects and data.

3.Azure

Azure, also known as Microsoft Azure, is a comprehensive cloud computing platform and service provided by Microsoft. It offers a wide range of cloud services, including those for computing, analytics, storage, and networking. Azure allows businesses to build, deploy, and manage applications and services through Microsoft-managed data centres.

Azure Storage account

An Azure Storage Account is a Microsoft Azure service that provides highly scalable, secure, and durable cloud storage for data. It's a fundamental building block in Azure, allowing you to store and manage various types of data, including blobs, files, tables, and queues. When you create a storage account, you have the ability to configure different services and access methods according to your specific requirements.

Azure Blob Storage

Azure Blob Storage is a cloud-based object storage service provided by Microsoft Azure. It is part of the Azure Storage service, which also includes Azure Table Storage, Azure Queue Storage, and Azure File Storage. Blob stands for "Binary Large Object," and Azure Blob Storage is optimized for storing massive amounts of unstructured data, such as text or binary data.

Azure Data factory

Azure Data Factory is a cloud-based data integration service that allows you to create, schedule, and manage data pipelines to move and transform data from various sources to different destinations. It enables you to create data-driven workflows for orchestrating and automating data movement and data transformation.

4.Snowflake

Snowflake is a cloud-based data warehousing platform that allows businesses to store and analyse large volumes of data in a scalable and cost-effective manner. It is known for its unique architecture, which separates storage and compute resources, enabling users to scale storage and compute independently. Snowflake is popular among enterprises for its ease of use, performance, and concurrency capabilities.

5.Power BI

Power BI is a business analytics service by Microsoft that provides interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards. It's part of the Microsoft Power Platform, which also includes Power Apps for

app development and Power Automate for workflow automation.

DATASET DESCRIPTION

In this project, I have used the Adventure Works online store dataset. This dataset contains a total of nine tables**.**

1. Adventure Works Customer Table
2. Adventure Works Product Table
3. Adventure Works Product Category Table
4. Adventure Works Product Subcategory Table
5. Adventure Works Return Product Table
6. Adventure Works Sales_2020_Table
7. Adventure Works Sales_2021_Table
8. Adventure Works Sales_2022_Table
9. Adventure Works Territory Table

The information that each table has

Adventure Works Customer Table

1.CustomerKey
2.FirstName
3.LastName
4.BirthDate
5.MaritalStatus
6.Gender
7.EmailAddress
8.AnnualIncome
9.TotalChildren
10.EducationLevel
11.Occupation
12.HomeOwner

Adventure Works Product Table

1.ProductKey
2.ProductSubcategoryKey
3.ProductSKU

4.ProductName

5.ModelName

6.ProductDescription

7.ProductColor

8.ProductSize

9.ProductStyle

10.ProductCost

11.ProductPrice

Adventure Works Product Category Table

1.ProductCategoryKey

2.CategoryName

Adventure Works Product Subcategory Table

1.ProductSubcategoryKey

2.SubcategoryName

3.ProductCategoryKey

Adventure Works Return Product Table

1.ReturnDate
2.TerritoryKey
3.ProductKey
4.ReturnQuantity

Adventure Works Sales_2020_Table

1.OrderDate
2.StockDate
3.OrderNumber
4.ProductKey
5.CustomerKey
6.TerritoryKey
7.OrderLineItem
8.OrderQuantity

Adventure Works Sales_2021_Table

1.OrderDate
2.StockDate
3.OrderNumber
4.ProductKey
5.CustomerKey
6.TerritoryKey
7.OrderLineItem
8.OrderQuantity

Adventure Works Sales_2022_Table

1.OrderDate
2.StockDate
3.OrderNumber
4.ProductKey
5.CustomerKey
6.TerritoryKey
7.OrderLineItem

8.OrderQuantity

Adventure Works Territory Table

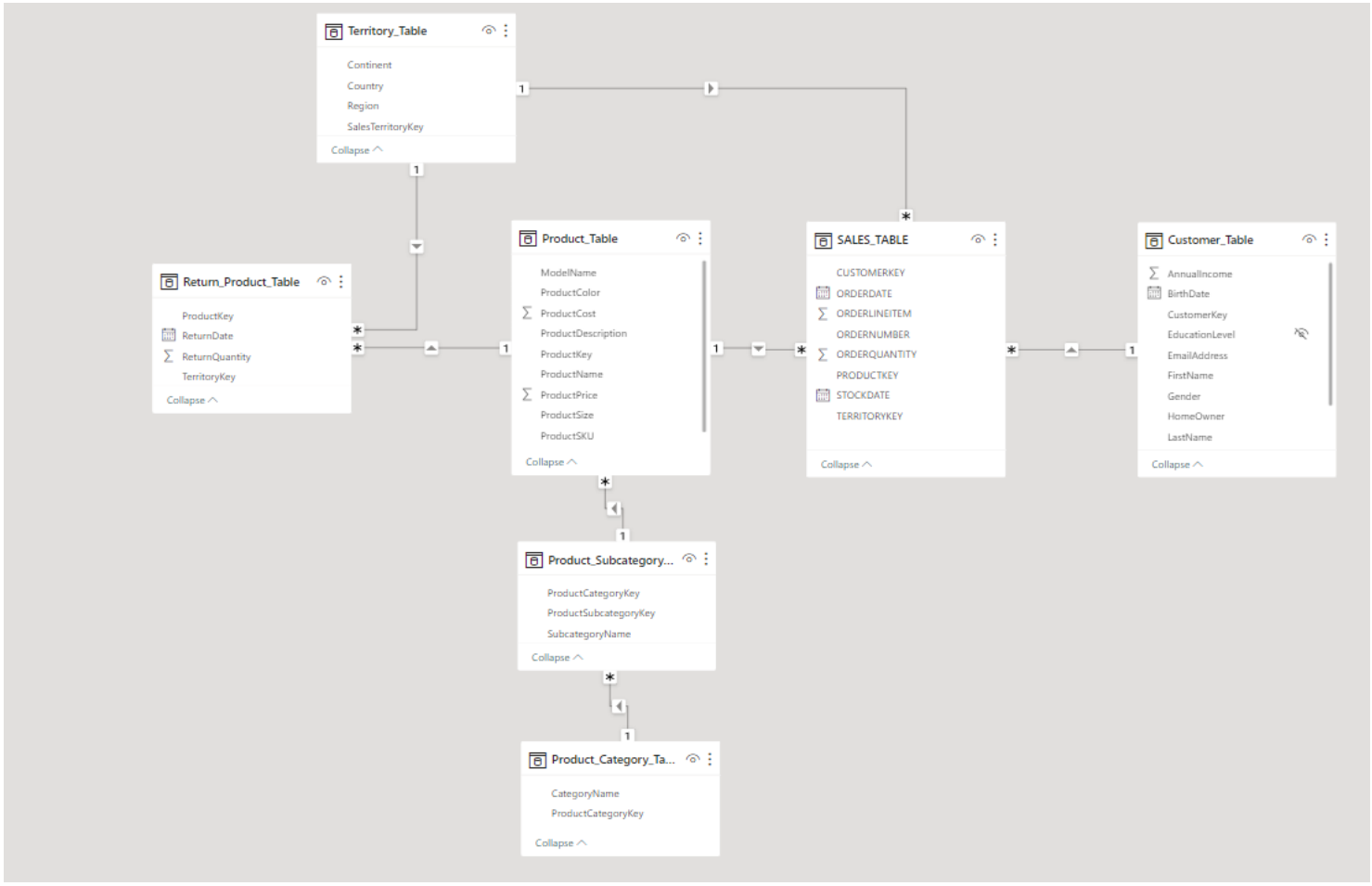
1.SalesTerritoryKey

2.Region

3.Country

4.Continent

DATA MODEL


Territory_Table
Continent
Country
Region
SalesTerritoryKey
Collapse
Return_Product_Table
ProductKey
ReturnDate
ReturnQuantity
TerritoryKey
Collapse
Product_Table
ModelName
ProductColor
ProductCost
ProductDescription
ProductKey
ProductName
ProductPrice
ProductSize
ProductSKU
Collapse
SALES_TABLE
CUSTOMERKEY
ORDERDATE
ORDERLINEITEM
ORDERNUMBER
ORDERQUANTITY
PRODUCTKEY
STOCKDATE
TERRITORYKEY
Collapse
Customer_Table
AnnualIncome
BirthDate
CustomerKey
EducationLevel
EmailAddress
FirstName
Gender
HomeOwner
LastName
Collapse
Product_Subcategory...
ProductCategoryKey
ProductSubcategoryKey
SubcategoryName
Collapse
Product_Category_Ta...
CategoryName
ProductCategoryKey
Collapse

WORKING OF THE PROJECT

1. Introduction

Project Overview:

The Online Store Data Engineering Project presents an end-to-end solution for processing and analysing online store data. This project is designed to tackle the challenges faced by businesses dealing with large volumes of online store data, enabling data-driven decision-making processes for business stakeholders.

Problem Statement:

In the ever-expanding e-commerce landscape, businesses struggle to efficiently handle and analyse vast amounts of data generated by online stores. Extracting meaningful insights from this data is crucial for informed decision-making and gaining a competitive edge.

Objectives:

Optimize Data Processing: Develop efficient methods for acquiring, processing, and transforming raw data.

Enhance Data Quality: Apply transformations to ensure data accuracy and consistency.

Enable Data-Driven Decisions: Visualize and analyse data to create insightful reports for business stakeholders.

Technologies Used:

Data Extraction: Python

Data Storage: SQL Server, Azure Blob Storage, Snowflake

ETL Process: Azure Data Factory

Data Visualization: Power BI

2. Data Acquisition

Data Sources:

The project acquires raw data from online store servers in a CSV format.

Data Extraction (Python Scripts):

Python scripts are employed to download and load data from on-premises servers into a SQL Server database. This initial step ensures the raw data is ready for processing.

Python Scripts:

Installations:

- Python

  Download the latest version of python on python.org

Modules (python libraries):

- Pandas

  command: pip install pandas

- Pyodbc

  command: pip install pyodbc

- OS

  command: pip install os

Source code:

The source code contains the python script for loading a data from on-premises to SQL Database

GitHub link:

https://github.com/ajithkumarece046/Data_Engineer_Sample_Project/blob/main/Data%20Engineering%20Project/Source%20code%20(Internal-SQL).ipynb

3.Data Migration to Azure Blob Storage:

Azure Data Factory is used to seamlessly migrate the loaded data from SQL Server to Azure Blob Storage. This cloud-based storage solution ensures data accessibility and security.

To Create Azure Free Trail: Refer page no 37

Steps to Create an Azure Storage Account:

Navigate to Storage Accounts:

- In the Azure portal, go to "Create a resource" > "Storage" > "Storage account".

Fill in Details:

- Choose your subscription and resource group.
- Enter a unique storage account name.
- Select your preferred location and performance (Standard/ Premium).
- Configure other settings as per your requirements.

Review and Create:

- Review your configurations and click "Create" to create the storage account.

After creating storage account, we need to create container.

Steps to Create Container in Azure Blob Storage:

Access Storage Account:

In the Azure portal, navigate to your storage account.

Create a Container:

- In the left pane, click on "Containers".
- Click on "+ Container" to create a new container.
- Enter a unique name for the container.
- Choose the appropriate public access level (private, blob, container, or anonymous).

Create Container:

- Click "Create" to create the container.

Uploading Files to Azure Blob Storage using Azure Data Factory:

Set Up Azure Data Factory:

Firstly, you need to create an Azure Data Factory, which acts as the orchestrator for your data workflows. In the Azure portal, you can initiate this by providing essential details like your subscription and resource group. Within the Data Factory, create a new pipeline. Think of a pipeline as a workflow that defines the data movement and transformation.

Use Lookup Activity:

In your created pipeline, you can utilize the Lookup activity. This activity allows you to connect to your SQL Server database and retrieve data. Essentially, it's like opening a door to your SQL Server,

enabling Data Factory to peek inside and fetch the required information.

Foreach Activity:

Once you have the data from your SQL Server, you can use the Foreach activity. This activity acts as a loop, iterating through the fetched data one piece at a time. Imagine it as a conveyor belt where each piece of data moves forward for processing.

Inside Foreach Activity:

Within the Foreach loop, use the "Copy Data" activity. This activity is pivotal; it's the hands that move the pieces of your data from your SQL Server to Azure Blob Storage. Here, you configure two vital components: the source dataset (SQL

Server) and the destination dataset (Azure Blob Storage).

Configure Source Dataset (SQL Server):

Specify your SQL Server as the source. Provide connection details, ensuring Data Factory knows where to find the data. Think of this step as setting up a secure tunnel between Data Factory and your SQL Server, allowing the seamless flow of information.

Configure Destination Dataset (Azure Blob Storage):

Similarly, configure Azure Blob Storage as your destination. Provide the necessary details, ensuring Data Factory knows where in the Blob Storage to place your data. This step is akin to specifying the exact shelf on which a product should be placed in a warehouse.

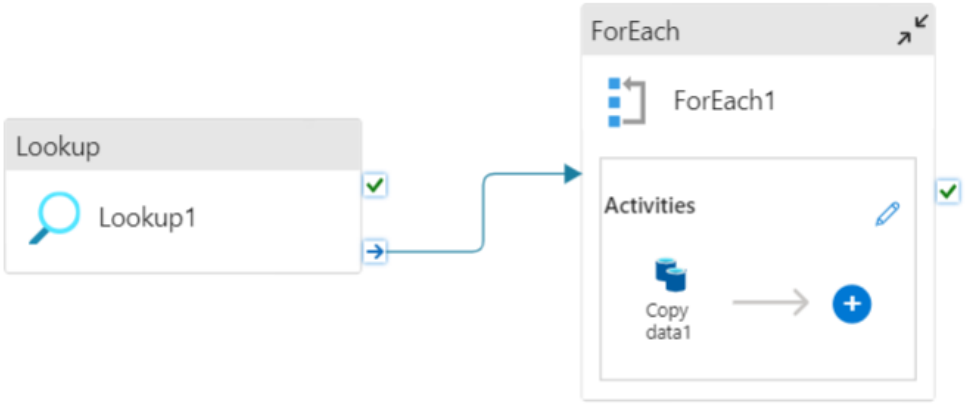
Column Mapping:

The beauty of this process lies in its intelligence. Data Factory understands that data formats might differ between your SQL Server and Azure Blob Storage. So, you map the columns, indicating which piece of data from your SQL Server corresponds to which space in your Azure Blob Storage. It's like ensuring that the shape of the item matches the shape of the space it's going into.

Execute Pipeline:

Finally, you're ready to set your pipeline in motion. Trigger the pipeline to execute. Picture this as pressing the 'start' button on a conveyor belt system. As the pipeline runs, it processes each piece of data through the Lookup, Foreach, and Copy

Data activities, seamlessly moving your CSV data from your SQL Server to Azure Blob Storage.



4. Data Warehousing (Snowflake)

Loading Data into Snowflake:

Data from Azure Blob Storage is efficiently loaded into Snowflake, a modern cloud-based data warehousing platform. External staging capabilities are utilized for seamless integration.

To Create a Snowflake free trail: Refer pgno:39

Once your Snowflake account is established, the initial step involves setting up a robust database infrastructure. Within Snowflake, create a database to house your data and organize it efficiently. A database acts as a container, providing a structured environment for your schemas and tables. Within the database, create schemas, offering a logical grouping for tables and other database objects. These schemas enhance manageability and maintain the integrity of your data.

One of the key features of Snowflake is its seamless integration with external staging, facilitating secure data transfer. Connect Snowflake to Azure Blob Storage, establishing an external stage to serve as a buffer between your Azure storage and

Snowflake. This stage acts as an intermediary where data can be stored temporarily before loading into Snowflake. This connection empowers you to harness the power of cloud storage while seamlessly integrating with Snowflake's analytical capabilities.

To populate your Snowflake tables, employ the COPY INTO command, a versatile tool that allows you to map columns and load data efficiently. By specifying the mapping between your data source (in this case, Azure Blob Storage) and the corresponding columns in your Snowflake table, you ensure that data is accurately transferred. This method ensures data integrity while maintaining the structure of your Snowflake tables.

Source code:

GitHub link:
https://github.com/ajithkumarece046/Data_Engineer_Sample_Project/blob/main/Data%20Engineering%20Project/Snowflake%20external%20stage%20Sql%20code.txt

Transformations in Snowflake:

Additional transformations are performed within Snowflake to further optimize the data for analytical processing. These transformations ensure the data is structured for efficient querying and analysis.

I have made a very small transformation in the data, merging all three sales datasets into one for the efficient analytics report.

Refer the transformation code in the snowflake source code.

5. Data Visualization and Analysis (Power BI)

Data Loading into Power BI:

The processed and transformed data is loaded into Power BI, a robust business analytics service. Power BI facilitates the creation of interactive and visually appealing dashboards.

Using get data we can easily load the data from Snowflake.

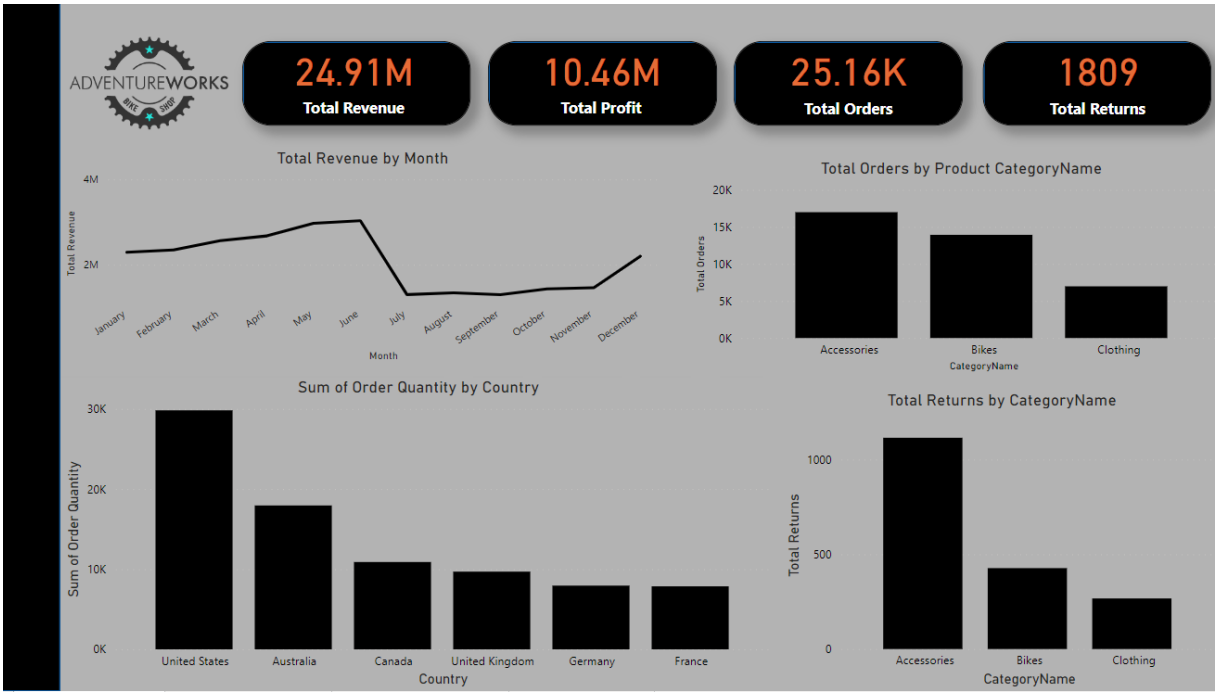
Report Creation:

Insightful reports are created within Power BI, showcasing key performance indicators and trends in the online store data. These reports offer a comprehensive view of the store’s performance metrics.

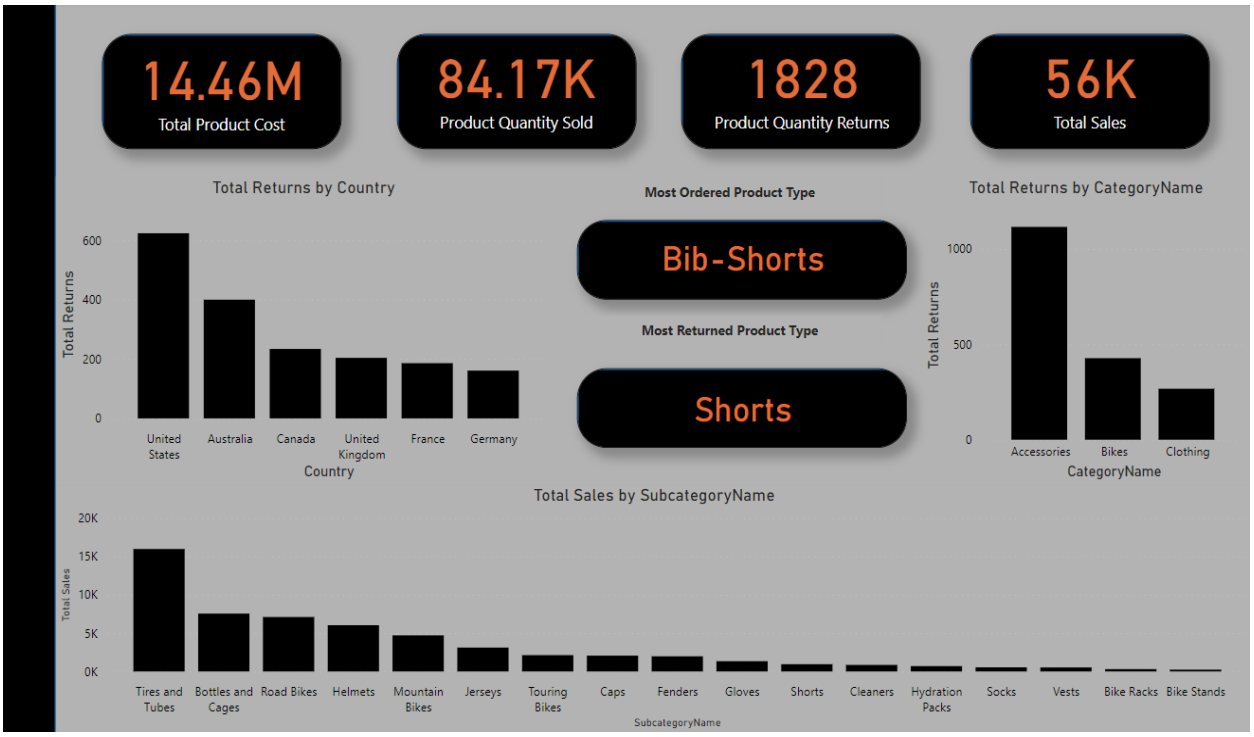
Data Insights:

The analysis in Power BI provides valuable insights into the online store's performance. These insights empower business stakeholders to make data-driven decisions, enhancing overall business strategies.

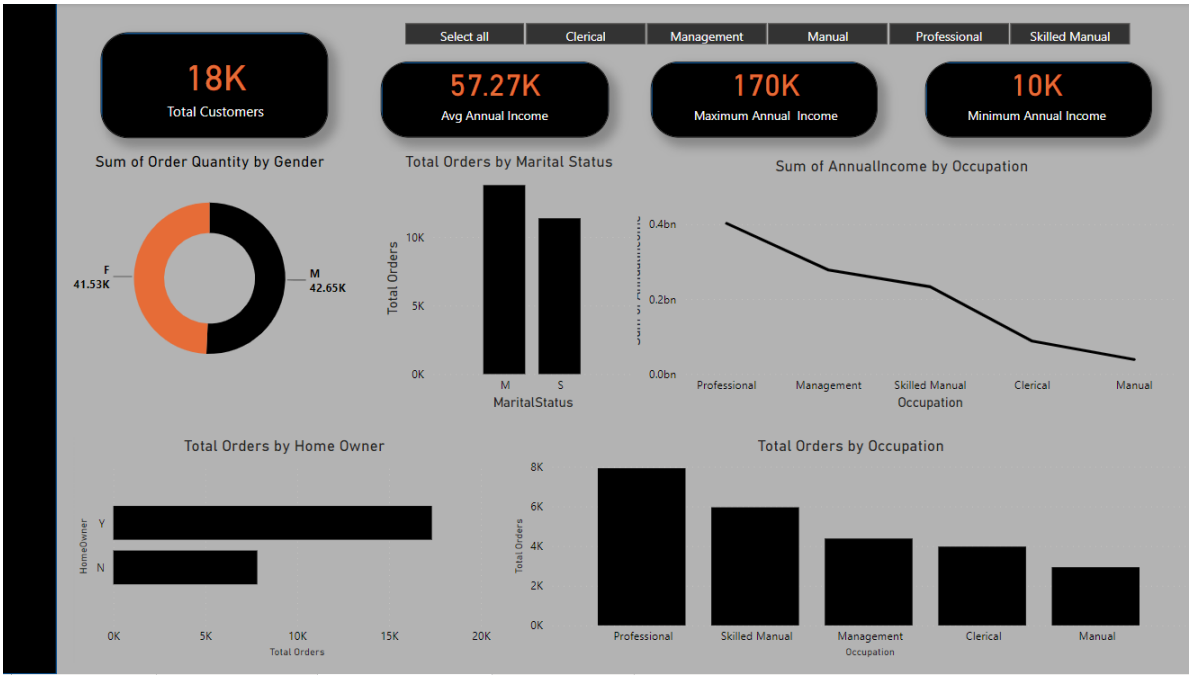
I have created four dashboard pages: the main dashboard, product dashboard, customer dashboard, and a map dashboard. These dashboards empower our business stakeholders to make valuable decisions based on insightful data. Through these visual representations, essential insights are readily accessible, enabling informed and strategic choices for the business.



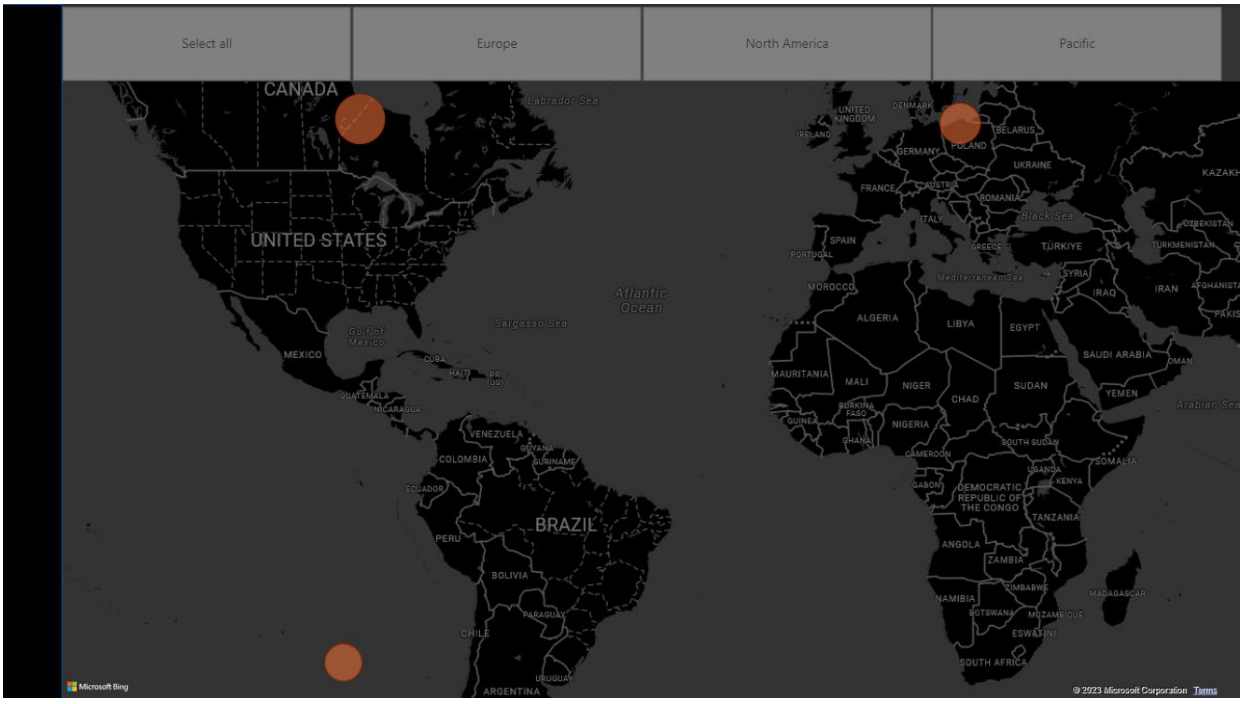
MAIN DASHBOARD



PRODUCT DASHBOARD



CUSTOMER DASHBOARD



MAP DASHBOARD

Source file:

GitHub link:

https://github.com/ajithkumarece046/Data_Engineer_Sample_Project/blob/main/Data%20Engineering%20Project/Result%20Dashboard.pbix

6. Conclusion

Summary of Achievements:

The project successfully implements a comprehensive data engineering workflow, encompassing data extraction, transformation, and loading (ETL) processes. By seamlessly integrating on-premises, cloud-based, and analytical tools, this solution provides a scalable and streamlined approach for handling and analysing large volumes of online store data.

Challenges Faced:

Throughout the project, challenges were met and overcome, ensuring the successful implementation of each stage of the data engineering process. Adaptability and problem-solving skills were key in overcoming these hurdles.

Steps to Create Azure Free Trail

1.Azure Free Trial Account Setup:

To get started with Azure, you can sign up for a free trial account. Here's how you can do it:

2.Visit Azure Portal:

Go to the https://portal.azure.com/ and click on "Start free" to create a new account.

3.Account Information:

Fill out the necessary information, including your email, password, and other required details.

4.Verification:

Verify your account through the email confirmation sent by Azure.

5.Credit Card Verification:

Provide your credit card details for verification purposes. Azure won't charge your card during the free trial period.

6.Access Azure Services:

Once verified, you can access Azure services through your dashboard.

Steps to create Snowflake Free Trail

1. Visit Snowflake's Website:

   Go to Snowflake's official website (https://www.snowflake.com/) and navigate to the "Free Trial" or "Get Started" section.

2. Sign Up for Free Trial:

   Click on the "Free Trial" or "Get Started" button. You'll likely need to provide your email address and create a password to set up your account.

3. Fill Out the Form:

   Fill out the registration form with your personal information, including your name, company name, job title, and other relevant details.

4.Verification:

Verify your email address. Snowflake will send you a verification email with a link to confirm your email address and activate your account.

5.Login to Snowflake:

Once your email is verified, log in to your Snowflake account using your email address and password.

6.Set Up Your Account:

Snowflake might guide you through a setup process, where you'll provide additional information about your use case, industry, and specific requirements.

7.Access Your Snowflake Account:

After the setup, you'll be redirected to your Snowflake account. From here, you can start creating databases, tables, and warehouses to store and analyse your data.

References

SQL Server to Azure Blob Storage:

18. Copy multiple tables in bulk by using Azure Data Factory - YouTube

External Staging from Azure Blob to Snowflake:

CREATE STAGE | Snowflake Documentation

All codes and files:

https://github.com/ajithkumarece046/Data_Engineer_Sample_Project/tree/main/Data%20Engineering%20Project