Kindle Training – Major Project

By Monish NP

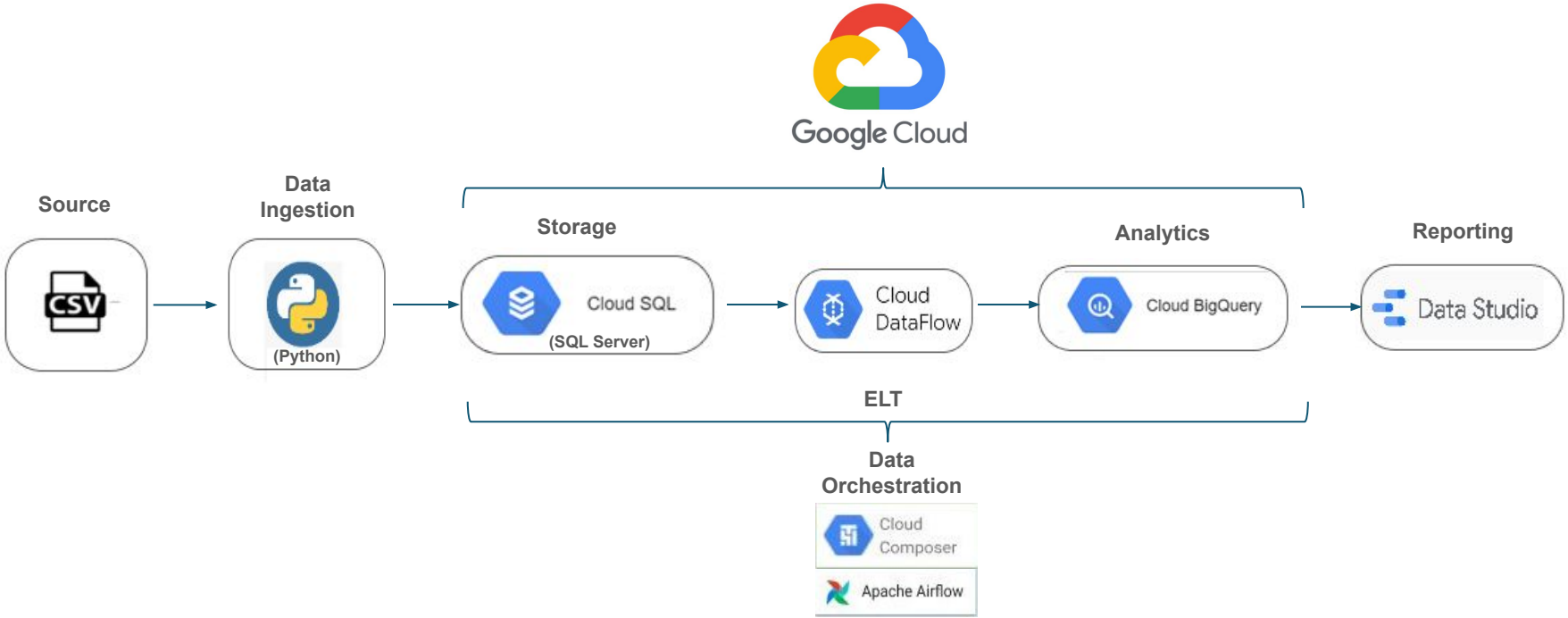
Overview

1. Problem Statement
2. Architecture
3. Solution Approach
4. Data Visualization and Reporting
5. Technology Stack
6. Challenges faced and learnings

Problem Statement

- Customer Segmentation to focus on growth
- Find 20% of customers who drive 80% of the company revenue
- Visualize and report the findings
- Analyze the visualized data for insights

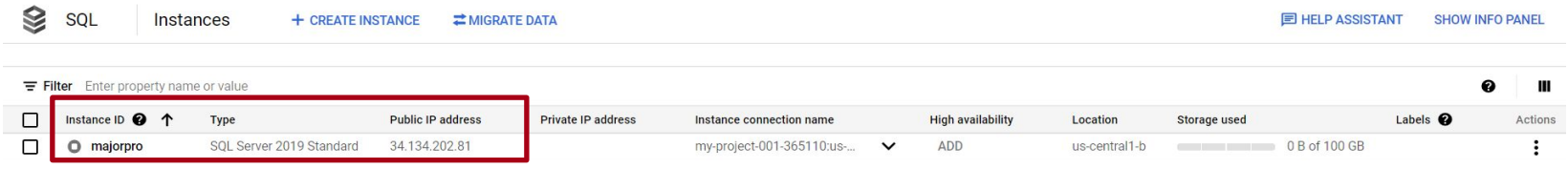
Architecture



Solution Approach

Step -1

- A Cloud SQL Server instance in GCP is created to import data into Cloud SQL.
- Wrote a python code and configured it with server credential to connect to Server instance.
- The below python code will create database and table in it with respective columns.




```
import pyodbc ## Python module for connecting DB from python
import pandas as pd
import numpy as np
```

```
# Connect to SQL Server
conn = pyodbc.connect(Driver='{SQL Server}',
                      Server='34.134.202.81',  # Cloudsqlserver ip
                      Database='MajorProject',
                      UID='sqlserver',
                      PWD='Sandip26@nc',
                      Trusted_Connection='no')
cursor = conn.cursor()
```

```
# Import CSV
data = pd.read_csv("data.csv", encoding='unicode_escape')   ## encoding is used as there are specialchar in the file
df = pd.DataFrame(data)
df.replace({np.inf: np.nan, -np.inf: np.nan}, inplace=True)
df = df.fillna(0)
```

```
df.head(3)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom

```
cursor.execute('''
    CREATE TABLE Sales
(InvoiceNo nvarchar(50),
 StockCode nvarchar(50),
 Description nvarchar(50),
 Quantity int,
 InvoiceDate nvarchar(50),
 UnitPrice nvarchar(50),
 CustomerID float,
 Country nvarchar(50))

   ''')
```

```
#insert records into table

counter=0
#statements = list()
state_list = list()
for row in df.itertuples():
    row_to_insert = [row.InvoiceNo, row.StockCode, row.Description, row.Quantity, row.InvoiceDate, row.UnitPrice,row.CustomerID,r
    rows_normalized = list(map(lambda x: str(x).replace("'", "''") , row_to_insert))
    statement = "INSERT INTO Sales(InvoiceNo, StockCode, Description,Quantity,InvoiceDate,UnitPrice,CustomerID,Country) VALUES (
    state_list.append(statement)
    counter += 1
    if counter == 500:
      statements = ';'.join(state_list)
      cursor.execute(statements)
      conn.commit()
      counter =0
      state_list = []

if not counter == 500:
 statements = ';'.join(state_list)
 cursor.execute(statements)
 conn.commit()
```

Solution Approach

Step -2

- Extract – Designed an Apache beam pipeline and executed it in dataflow to extract data into bigquery table.

```
    p = beam.Pipeline(options=pipeline_options)

    init = p | 'Begin pipeline with initiator' >> beam.Create(['All tables initializer'])
    (init
     # Read the file. This is the source of the pipeline. All further
     # processing starts with lines read from the file. We use the input
     # argument from the command line. We also skip the first line which is a
     # header row.

     | 'Read from a DB' >>  beam.ParDo(ReadSQLTable())

     # This stage of the pipeline translates from a CSV file single row
     # input as a string, to a dictionary object consumable by BigQuery.
     # It refers to a function we have written. This function will
     # be run in parallel on different workers using input from the
     # previous stage of the pipeline.
     | 'Write to BigQuery' >> beam.io.WriteToBigQuery('my-project-001-365110.Major_Project.sqlserver_to_bigquery',
                                        schema='InvoiceNo:STRING,StockCode:STRING,Description:STRING,Quantity:INTEGER,'
'InvoiceDate:STRING,UnitPrice:FLOAT,CustomerID:INTEGER,Country:STRING',
                                        create_disposition=beam.io.BigQueryDisposition.CREATE_IF_NEEDED,
                                        write_disposition=beam.io.BigQueryDisposition.WRITE_TRUNCATE))
    p.run().wait_until_finish()


if __name__ == '__main__':
    logging.getLogger().setLevel(logging.INFO)
    run()
```

Figure : Pipeline Code

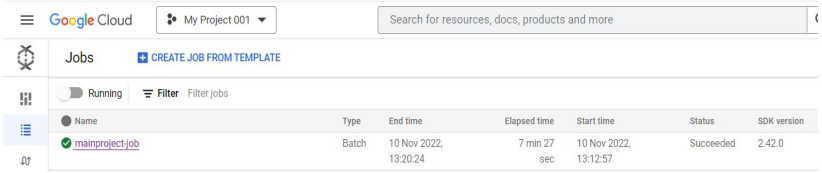


Figure : Pipeline Executed in Dataflow

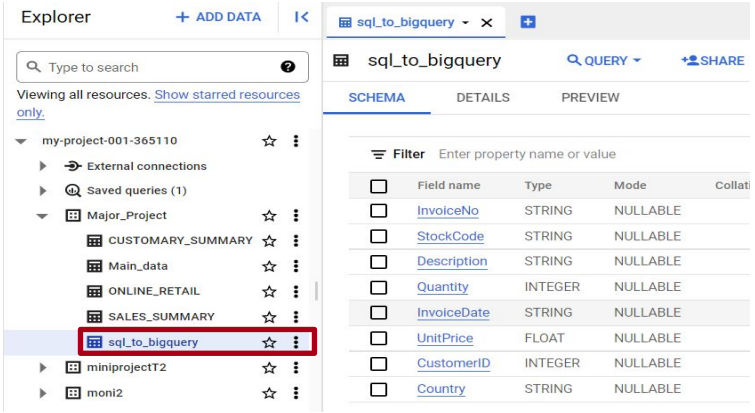


Figure : Table in Bigquery after Pipeline execution

Atos Maven Wave an atos company

Solution Approach

Step -3

- Created an environment in Cloud Composer, uploaded the dag(python file) into the dag folder.
- Transform - Queried raw data table and cleaned it using python Dag concept as per the project requirements.
- Load – After transformation the data was loaded into respective tables i.e. (online_retail. Customer_summary, sales_summary)
- The above ETL process is orchestrated using Cloud Composer

```
cloudsql_to_bq = BeamRunPythonPipelineOperator(
    task_id="cloudsql_to_bq",
    runner="DataflowRunner",
    py_file="gs://us-central1-mainpro-a06ea0ba-bucket/dags/sql  to bq _moni.py",
    pipeline_options={'tempLocation' : 'gs://majorpro/temp/', 'stagingLocation': 'gs://majorpro/temp/'},
    py_options=[],
    py_requirements=['apache-beam[gcp]==2.42.0','cloud-sql-python-connector[pytds]==0.6.1','pyodbc==4.0.34','SQLAlchemy==1.4.41','pymssql==
    py_interpreter='python3',
    py_system_site_packages=False,
    dataflow_config=DataflowConfiguration(job_name='mainproject-job', project_id='my-project-001-365110',
    location="us-central1"),
    dag=dag)


query1 = '''
            SELECT InvoiceNo, StockCode, Description, Quantity, PARSE_DATETIME('%m/%d/%Y %H:%M', InvoiceDate) AS InvoiceDate,
            UnitPrice, CustomerID,Country FROM `my-project-001-365110.Major_Project.sql_to_bigquery`
        '''
```

```
query2 = '''
            delete from `my-project-001-365110.Major_Project.Main_data`
            where Quantity  < 0 or CustomerID = 0;
        '''
```

```
query3 = '''
            delete from `my-project-001-365110.Major_Project.Main_data`
            where  EXTRACT(year FROM InvoiceDate)  = 2011 and EXTRACT(month FROM InvoiceDate) = 12;
        '''
```

Figure : DAG with Transformation

Atos Maven Wave an atos company

Solution Approach

```
query4 = '''
          SELECT InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID,Country,
          Quantity * UnitPrice as ItemTotal
          FROM `my-project-001-365110.Major_Project.Main_data`;
        '''

ONLINE_RETAIL = BigQueryOperator(task_id='ONLINE_RETAIL',
                        destination_dataset_table= "my-project-001-365110.Major_Project.ONLINE_RETAIL",
                        sql=query4,
                        use_legacy_sql=False,
                        create_disposition="CREATE_IF_NEEDED",
                        write_disposition="WRITE_TRUNCATE",
                        dag=dag
                        )

query5 = '''
          select CustomerID, sum(ItemTotal) as TotalSales,count(Quantity) as OrderCount, avg(ItemTotal) as AvgOrderValue
          from `my-project-001-365110.Major_Project.ONLINE_RETAIL`
          group by CustomerID;
        '''

CUSTOMARY_SUMMARY = BigQueryOperator(task_id='CUSTOMARY_SUMMARY',
                        destination_dataset_table= "my-project-001-365110.Major_Project.CUSTOMARY_SUMMARY",
                        sql=query5,
                        use_legacy_sql=False,
                        create_disposition="CREATE_IF_NEEDED",
                        write_disposition="WRITE_TRUNCATE",
                        dag=dag)
```

Figure : Query Creating Tables

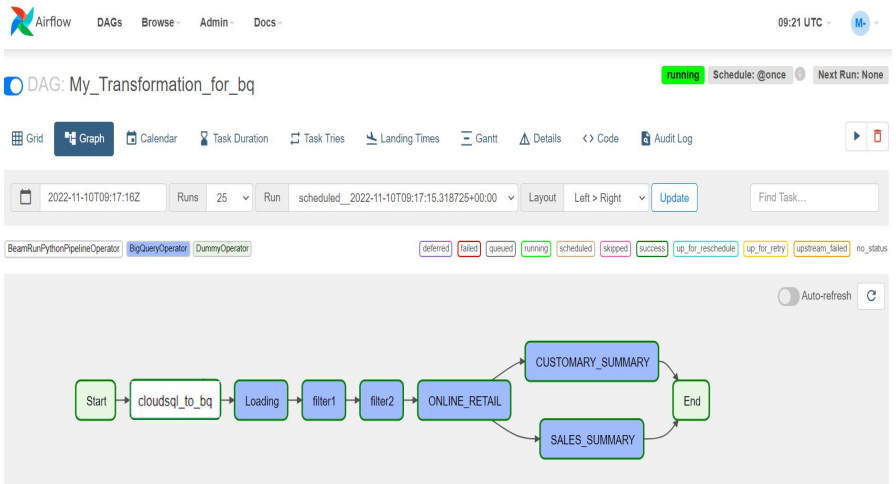
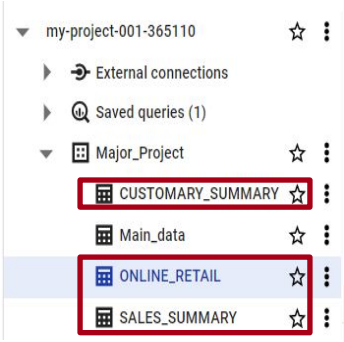
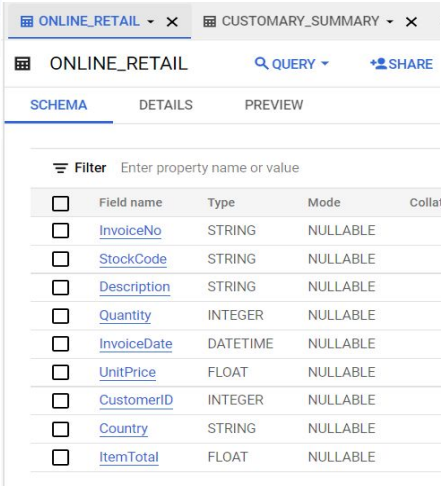


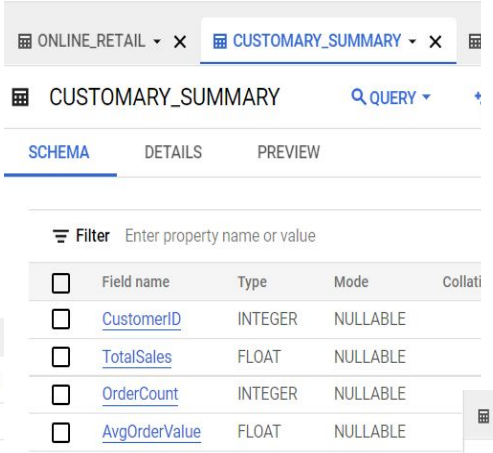
Figure : Air flow UI with Successful Orchestration

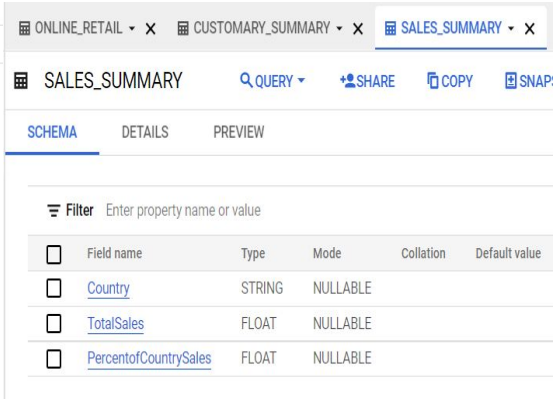
Atos Maven Wave an atos company

Tables Created









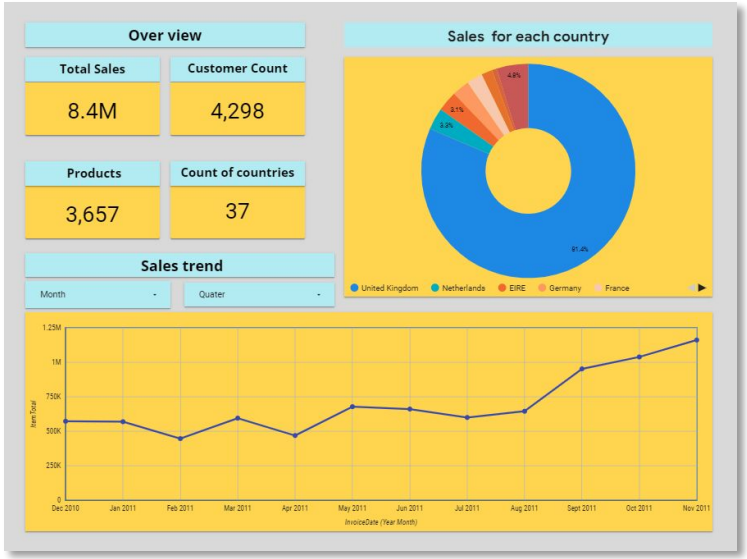
Atos Maven Wave

Data Visualization and Reporting

- Using Bigquery as a data source in data studio we can access the tables stored in Bigquery
- The following questions are answered and a visualized which can be accessed from the link below

- What are the sales figures for each country?
- What is the overall sales trend?
- How many new customers are there each month?
- When do customers make the most purchases?
- Which is the best selling product in each country?
- When were the largest orders made?
- Which customers made the largest orders?

Link - https://datastudio.google.com/s/o7bBBOcgd_s



Atos Maven Wave

Technology Stack

Source	Data Ingestion	Data WareHouse	Orchestrion	Reporting
CSV File Cloud SQL	Python Apache Beam Data Flow	Big Query	Cloud Composer Air Flow	Data Studio

Challenges Faced

- Data ingestion Infinite values, Time consuming during record insertion.
- Running pipeline Startup worker pool in zone Asia failed // replaced zone to us central.
- Run time error - error 403 message forbidden Enabled data flow api , cloud sql api , granted permission for service account
- Orchestration Name error "google" not defined Created a requirement.txt file with dependencies

Learnings

- Alternatively we can implement Federated Queries with MySQL option instead of SQL server
- Practical experience working with GCP services like Cloud shell, Cloud Sql, Data flow, Bigquery.
- Building and running Beam Pipeline.
- Creating Dashboard in Data Studio from BigQuery tables.

Thank you !