Fezekile Gwayi

BI DEV

Solution

* Design a Data Warehouse that will suit the requirements
* Design a pipeline process to load the from files to the database or cloud
* Develop a code in Python to Extract, Transform and Load the data.

**Design Explanation:**

- Create folder **"** **C:/Users/fgwayi/Documents/pargo/data**

**Required Tool Stack**

* Python 3
* MSSQL
* Microsoft SQL Server Management Studio

**Build Process & Design**

Create the database on MSSQL SRVER (database name:pargo\_datalake for raw data from csv files, and pargo\_dm for clean and transformed data)

Create Table 1 => customer in pargo\_datalake DB : Holds imported csv data.

Create Table 2 => pickup\_points: Holds imported csv data.

Create Table 2 => pickup\_points: Holds imported csv data.

In pargo\_dm database we create tables by selecting and transform from pargo\_dm.

Please find the scripts below.

/\*\*\*\*\*\*pickup point validation \*\*\*\*\*\*/

--This table passed the validation

SELECT [Pickup\_Point\_ID],[Suburb],[Province],[Regional]

FROM [pargo\_datalake].[dbo].[Pick\_up\_points]

where [Pickup\_Point\_ID] is null or [Suburb] is null or [Province] is null or [Regional] is null

group by [Pickup\_Point\_ID]

-----create dim\_pickup\_points table in dm database

SELECT [Pickup\_Point\_ID],[Suburb],[Province],[Regional]

into [pargo\_dm].[dbo].[dim\_pickup\_points]

FROM [pargo\_datalake].[dbo].[Pick\_up\_points]

/\*\*\*\*\*\*customer validation \*\*\*\*\*\*/

--checking the primary key

SELECT [Customer\_ID], count([Customer\_ID]) as Dups

FROM [pargo\_datalake].[dbo].[customer]

group by [Customer\_ID]

having count([Customer\_ID]) > 1

---checking the IDs

SELECT [Customer\_ID], count([Customer\_ID]) as counts

FROM [pargo\_datalake].[dbo].[customer]

where len([Customer\_ID]) < 4

group by [Customer\_ID]

--checking names

SELECT [Customer\_Name], count([Customer\_Name]) as Dups

FROM [pargo\_datalake].[dbo].[customer]

where len([Customer\_Name]) < 3

group by [Customer\_Name]

--checking cell numbers

Here we noticed that the number are incosistent

Using the following to get the consistent

SELECT

(case when len([Customer\_Cell]) < 11 then '27'+[Customer\_Cell] else [Customer\_Cell] end) as Customer\_cell

FROM [pargo\_datalake].[dbo].[customer]

--Final table we can have bellow

--by dropping all the invalid rows

SELECT[Customer\_ID],

[Customer\_Name],

(case when len([Customer\_Cell]) < 11 then '27'+[Customer\_Cell] else [Customer\_Cell] end) as Customer\_cell

INTO [pargo\_dm].[dbo].[dim\_customer]

FROM [pargo\_datalake].[dbo].[customer]

where len([Customer\_Name]) > 3 or len([Customer\_ID]) > 4

--Now we have dim\_customer in our dm database

/\*\*\*\*\*\* create fact\_parcels in pargo\_dm \*\*\*\*\*\*/

SELECT [Waybill]

,[Customer\_ID]

,[Order\_Date]

,(case when isnumeric([Parcel\_KG]) = 1 then cast([Parcel\_KG] as decimal(10,2)) else 0.00 end) as Parcel\_KG

,(case when isnumeric([Courier\_Charge]) = 1 then cast([Courier\_Charge] as decimal(10,2)) else 0.00 end) as Courier\_Charge

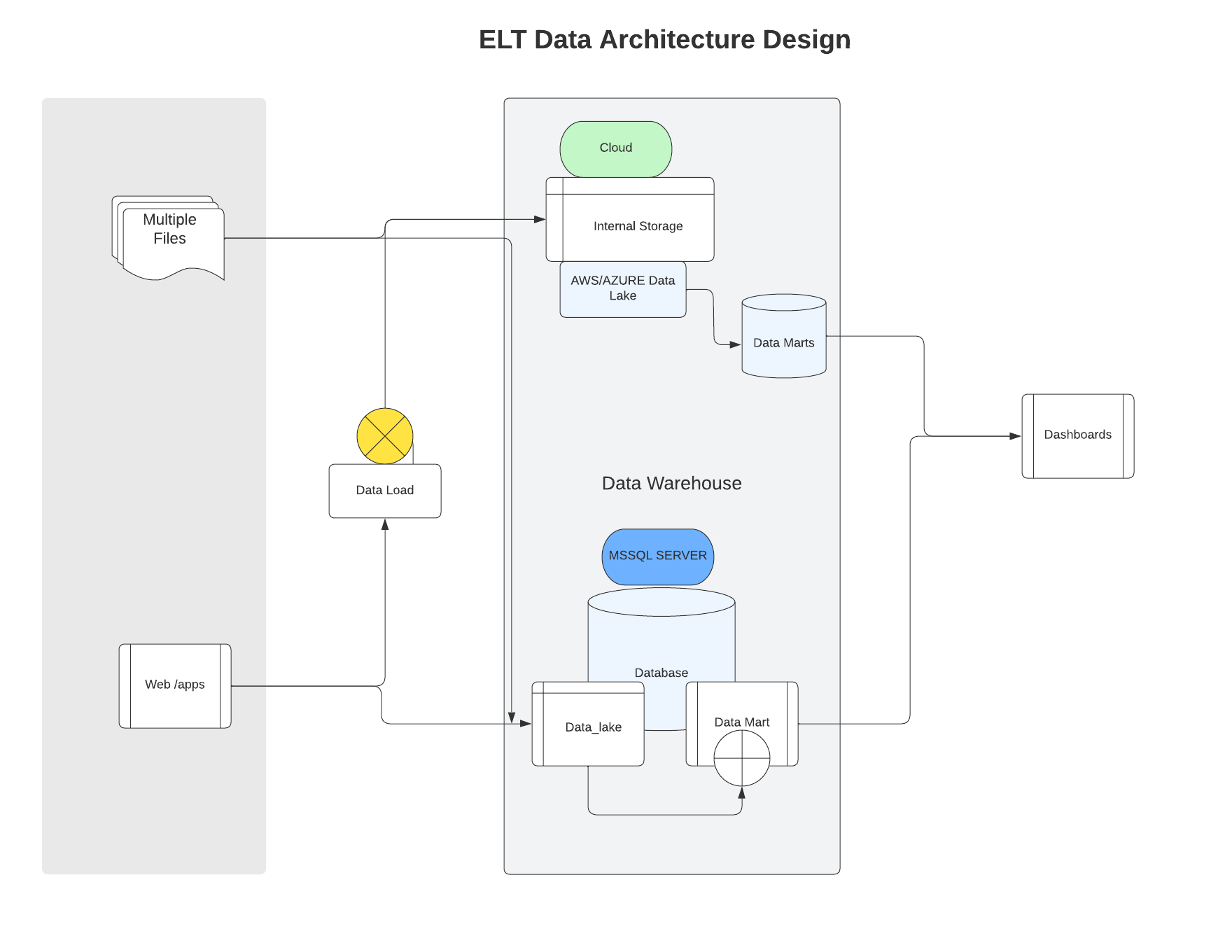
,(case when isnumeric([Sales\_amount]) = 1 then cast([Sales\_amount] as decimal(10,2)) else 0.00 end) as Sales\_amount

,[Pickup\_Point\_ID]

into [pargo\_dm].[dbo].[fact\_parcels]

FROM [pargo\_datalake].[dbo].[parcels]

**Data Pipeline Process**

****

In this case we have daily, weekly or monthly csv files landing in our server. We can have also have data coming from different platforms (e.g. web or app feeds). The csv file are extracted, transformed and loaded to either Azure/AWS s3 Data Lake and read the data using one of the powerful reporting engines (Redshift) or MSSQL Database using python tool/language. In this case we are using MSSQL to store the data extracted from the files and our Business Analyst can connect Tableau, Looker, Power BI, etc. Directly to Redshift or MSSQL Databse(see the image below).

**Dashboard Query**

SELECT

fp.[Pickup\_Point\_ID]

,dp.[Regional]

,dp.[Province]

,dp.[Suburb]

,fp.[Waybill]

,fp.[Customer\_ID]

,fp.[Order\_Date]

,SUM(fp.[Parcel\_KG]) AS total\_weight

,SUM(fp.[Courier\_Charge]) AS total\_courier\_charge

,SUM(fp.[Sales\_amount]) AS total\_Sales\_amount

,COUNT(fp.[Customer\_ID]) AS number\_of\_customers

,COUNT(fp.[Waybill]) AS number\_of\_orders

FROM [pargo\_dm].[dbo].[fact\_parcels] AS fp

INNER JOIN [pargo\_dm].[dbo].[dim\_pickup\_points] AS dp

ON fp.[Pickup\_Point\_ID] = dp.[Pickup\_Point\_ID]

GROUP BY fp.[Pickup\_Point\_ID]

,dp.[Regional]

,dp.[Province]

,dp.[Suburb]

,fp.[Waybill]

,fp.[Customer\_ID]

,fp.[Order\_Date]

**ETL Code in Python**

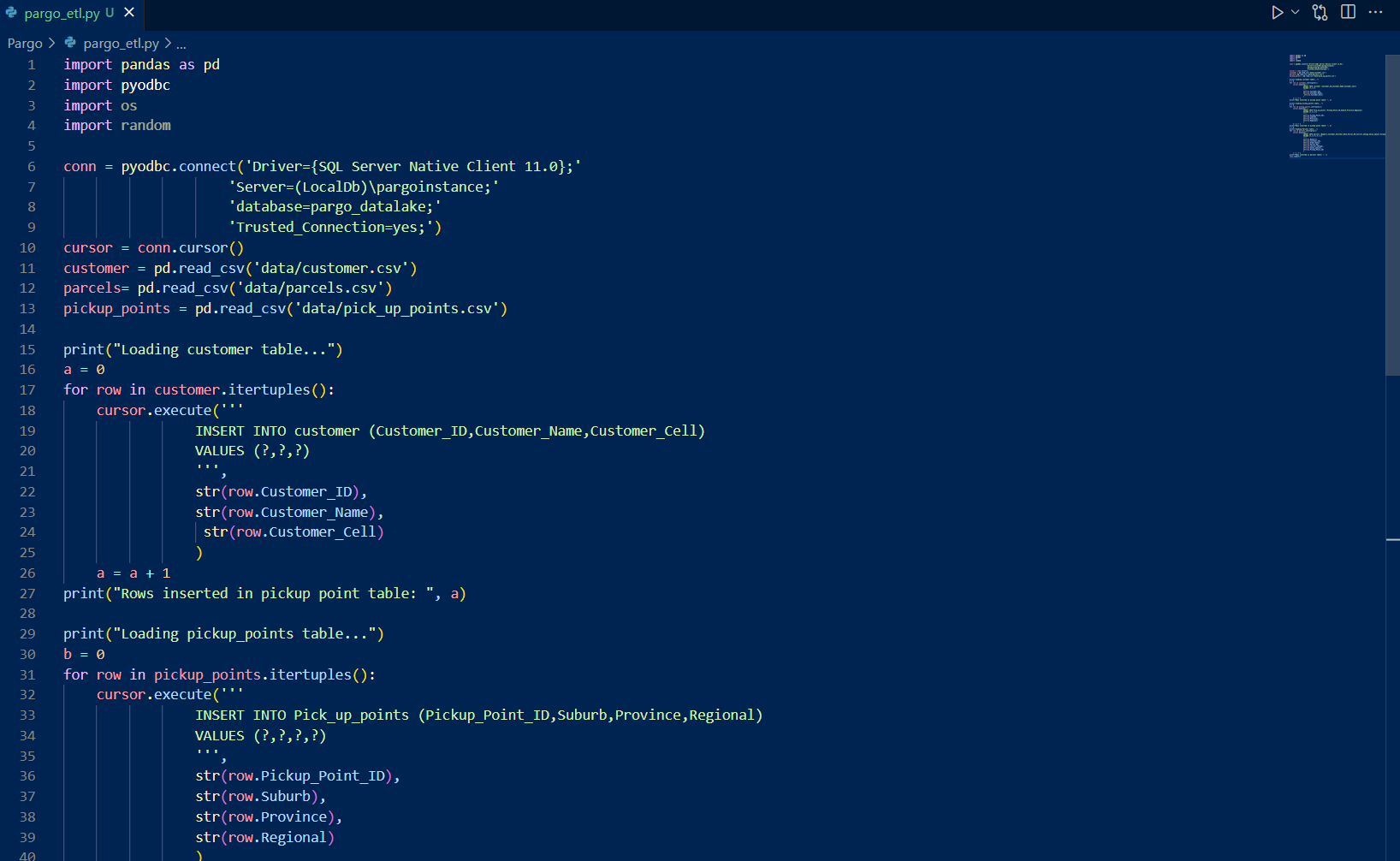
**--Installing the required packeges**

Pip install ‘pandas’

Pip install ‘glob’

Pip install pyodbc

Pip install shutil



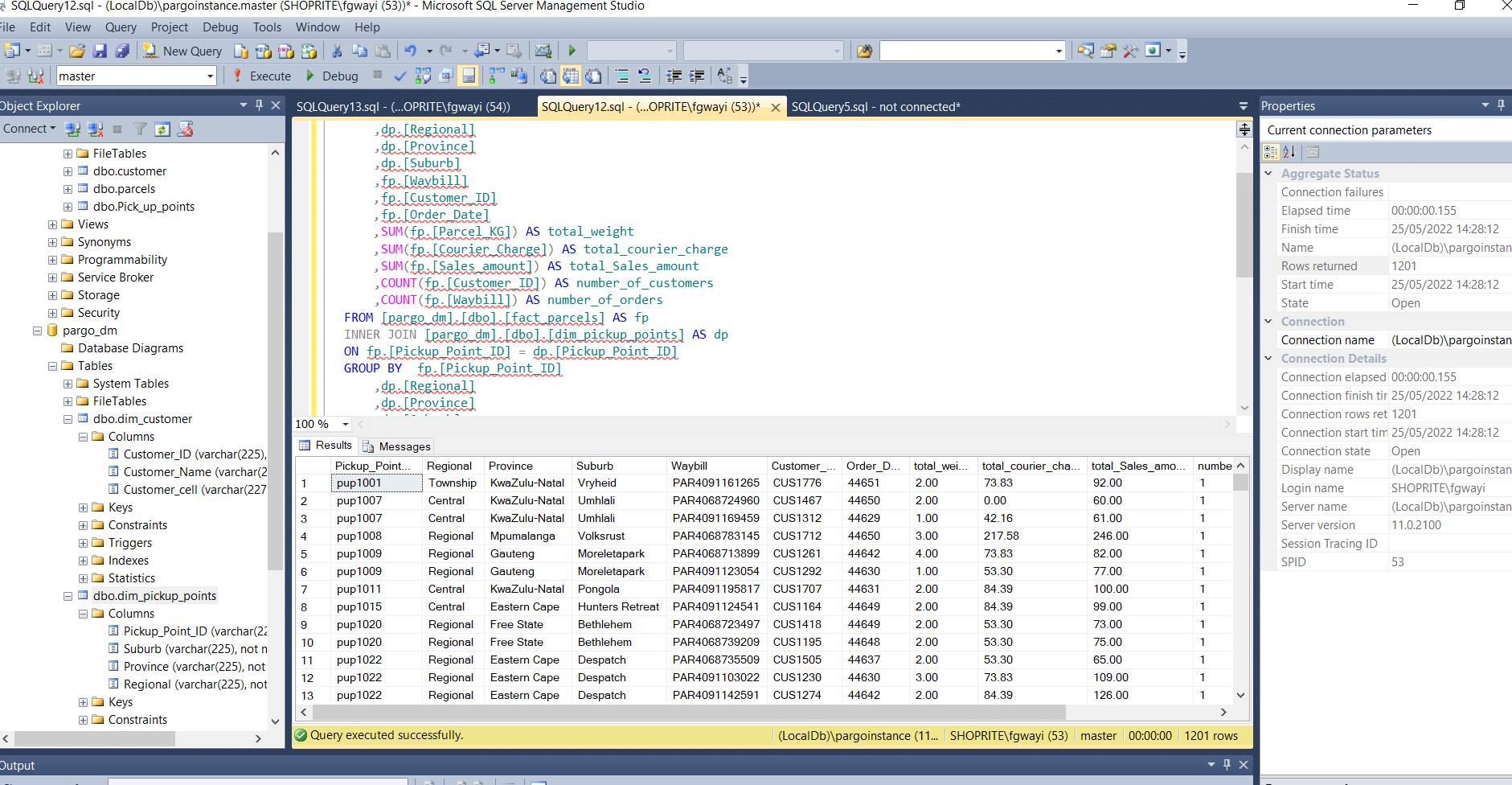
**ETL code execution:**

**Python imprtdata.py then enter**

We can setup an automated task on the local machine that will be scheduled to run/excute our ETL code every day when there is new file in landing dir. (This tool can load dataset with more 2 million rows in less than 2 minutes)

AWS Lambda service can also be used to trigger events to Redshift when there is data loaded in AWS S3 Data Lake.

**Results: Data in MSSQL Database**



**Tableau Dashboard connecting to MSSQL Database/csv for Insights.**

