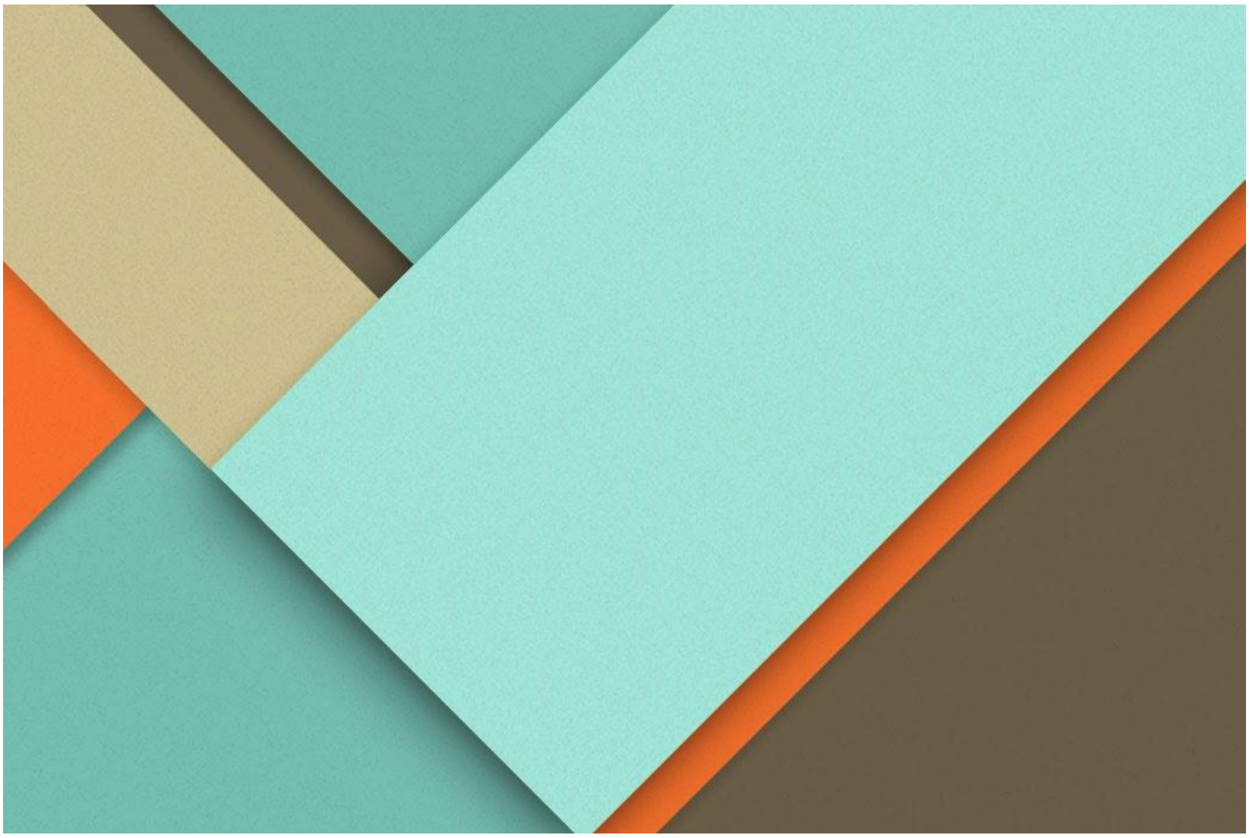


On-Premise ETL

Version: V-1.0.0

14th June, 2020

Prepared By:

Rafiul Ahmed, Anik Mahmud
eGeneration Ltd.
Saimon Center
Gulshan-1, Dhaka – 1212

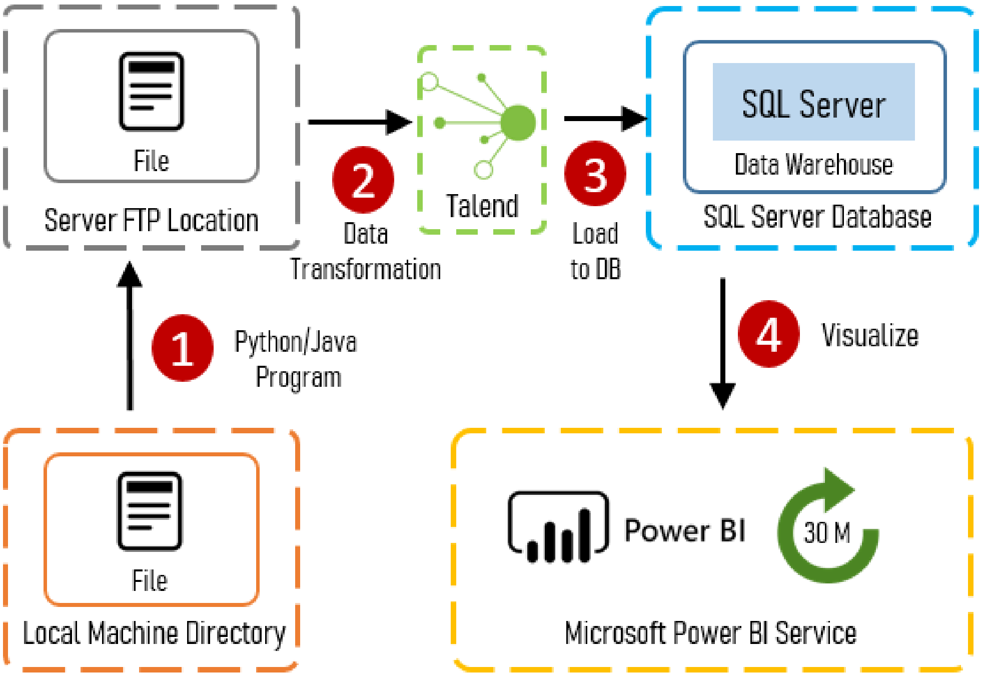
Overview:

ETL is short for extract, transform, load, three database functions that are combined into one tool to pull data out of one database and place it into another database. We want to perform following tasks for this ETL process on-premise.

ETL Steps

- Continuous File Loading: Load Data from Local Machine into an FTP Location Using Python or Java Program.
- Daily Talend Job: Load data from FTP location into talend for Data Transformation
- Load data into SQL Server Database with Talend
- Visualize data in Microsoft Power BI and Schedule Refresh and update Dashboards and Reports on Daily Basis.

Architecture Design:

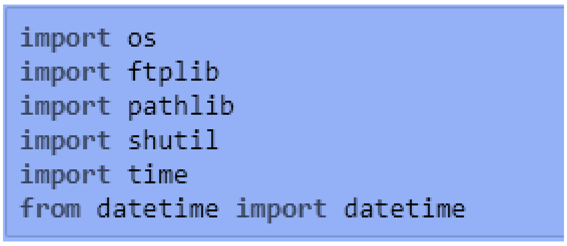


Continuous File Loading into FTP Server:

One of the main features of FTP server is the ability to store and retrieve files. We will use Python for uploading task which will be a continuous process. We will be using Python's built-in ftplib module.

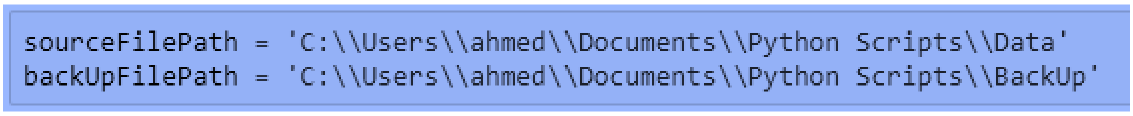
Python code description:

1. **Import necessary libraries.**



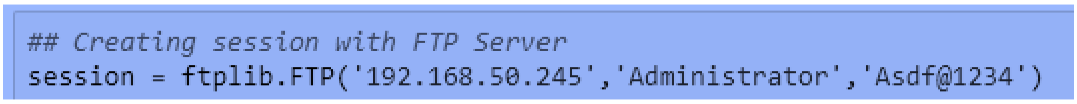
```
import os
import ftplib
import pathlib
import shutil
import time
from datetime import datetime
```

2. **Define Source and Backup directories**



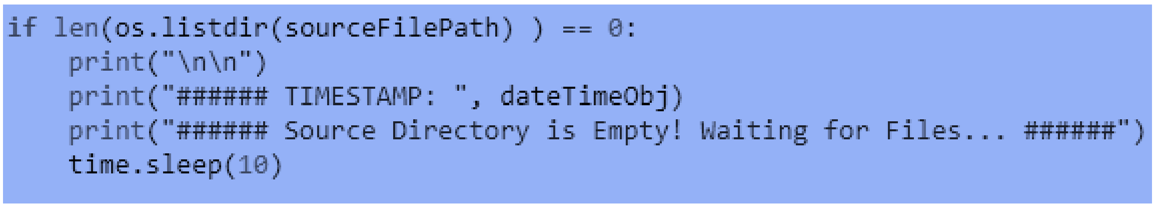
```
sourceFilePath = 'C:\\Users\\ahmed\\Documents\\Python Scripts\\Data'
backUpFilePath = 'C:\\Users\\ahmed\\Documents\\Python Scripts\\BackUp'
```

3. **Creating connection with FTP server**



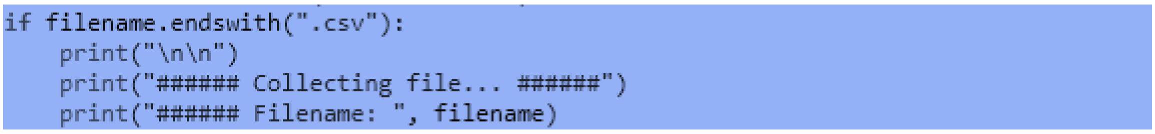
```
## Creating session with FTP Server
session = ftplib.FTP('192.168.50.245','Administrator','Asdf@1234')
```

4. **Check if there is file in the Source Directory. If not, wait for some time and check again**



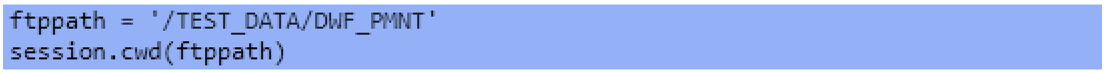
```
if len(os.listdir(sourceFilePath) ) == 0:
    print("\n\n")
    print("###### TIMESTAMP: ", dateTimeObj)
    print("###### Source Directory is Empty! Waiting for Files... ######")
    time.sleep(10)
```

5. **If there are files, take only the files that have .csv extension**



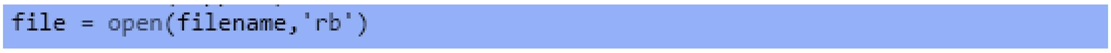
```
if filename.endswith(".csv"):
    print("\n\n")
    print("###### Collecting file... ######")
    print("###### Filename: ", filename)
```

6. **Set the FTP upload directory**



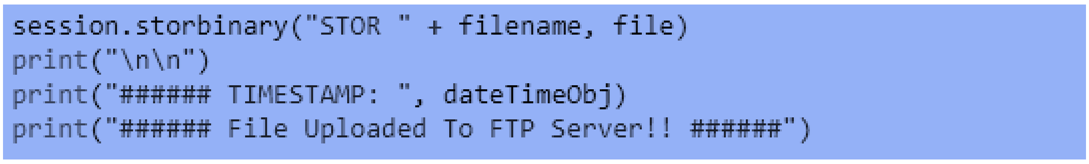
```
ftppath = '/TEST_DATA/DWF_PMNT'
session.cwd(ftppath)
```

7. **Open file in Read mode**



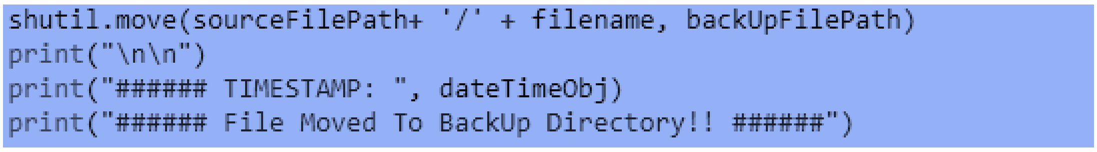
```
file = open(filename,'rb')
```

8. **Upload file in the FTP server location. We will upload Binary files with the storbinary() method.**



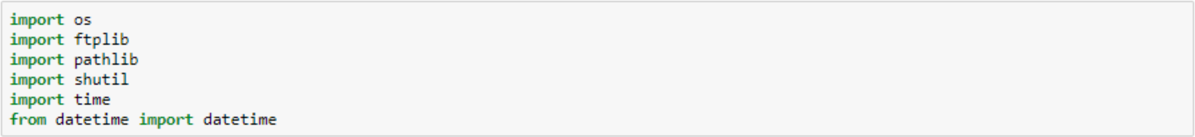
```
session.storbinary("STOR " + filename, file)
print("\n\n")
print("###### TIMESTAMP: ", dateTimeObj)
print("###### File Uploaded To FTP Server!! ######")
```

9. **Move file from Source to Backup Directory**

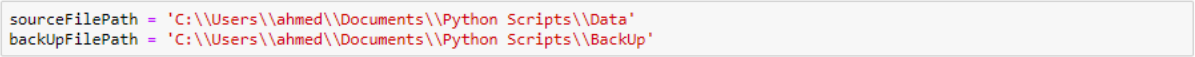


```
shutil.move(sourceFilePath+ '/' + filename, backUpFilePath)
print("\n\n")
print("###### TIMESTAMP: ", dateTimeObj)
print("###### File Moved To BackUp Directory!! ######")
```

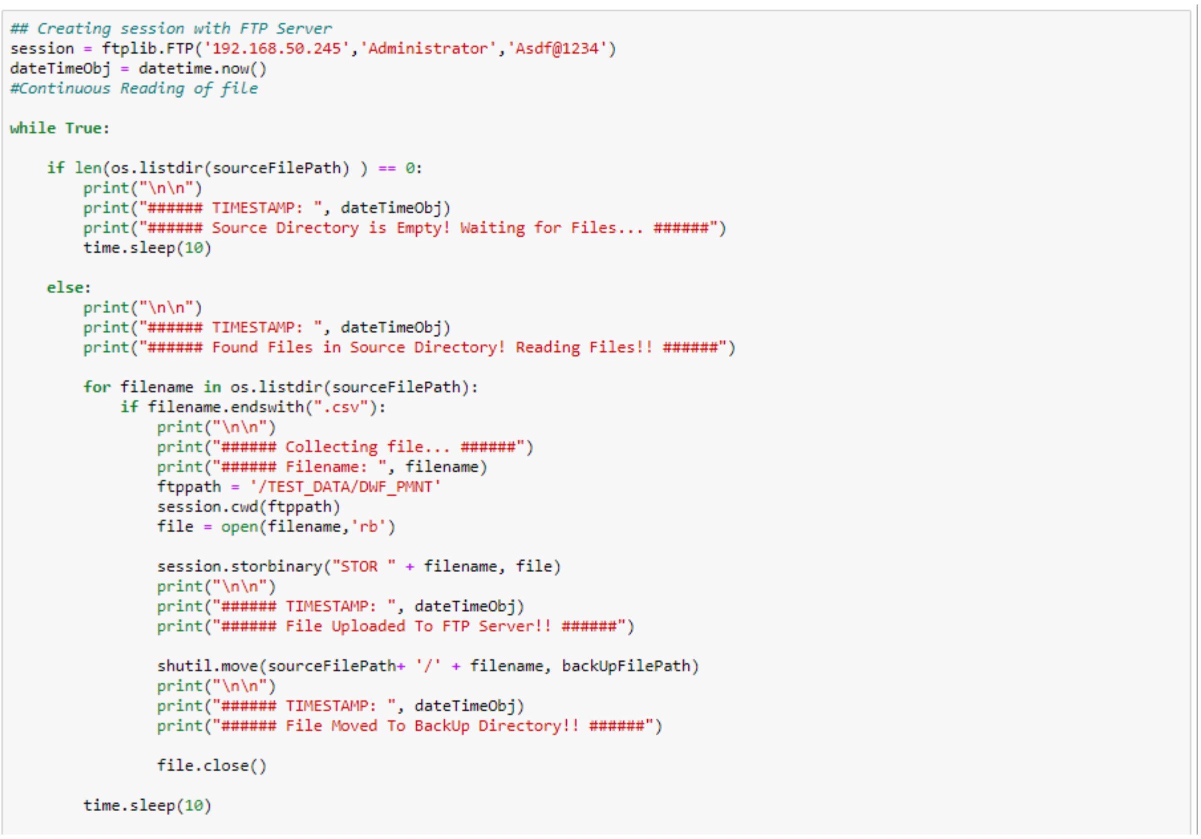
Whole Code Snippet:



```
import os
import ftplib
import pathlib
import shutil
import time
from datetime import datetime
```



```
sourceFilePath = 'C:\\Users\\ahmed\\Documents\\Python Scripts\\Data'
backUpFilePath = 'C:\\Users\\ahmed\\Documents\\Python Scripts\\BackUp'
```



```
## Creating session with FTP Server
session = ftplib.FTP('192.168.50.245','Administrator','Asdf@1234')
dateTimeObj = datetime.now()
#Continuous Reading of file

while True:

    if len(os.listdir(sourceFilePath) ) == 0:
        print("\n\n")
        print("###### TIMESTAMP: ", dateTimeObj)
        print("###### Source Directory is Empty! Waiting for Files... ######")
        time.sleep(10)

    else:
        print("\n\n")
        print("###### TIMESTAMP: ", dateTimeObj)
        print("###### Found Files in Source Directory! Reading Files!! ######")

        for filename in os.listdir(sourceFilePath):
            if filename.endswith(".csv"):
                print("\n\n")
                print("###### Collecting file... ######")
                print("###### Filename: ", filename)
                ftppath = '/TEST_DATA/DWF_PMNT'
                session.cwd(ftppath)
                file = open(filename,'rb')

                session.storbinary("STOR " + filename, file)
                print("\n\n")
                print("###### TIMESTAMP: ", dateTimeObj)
                print("###### File Uploaded To FTP Server!! ######")

                shutil.move(sourceFilePath+ '/' + filename, backUpFilePath)
                print("\n\n")
                print("###### TIMESTAMP: ", dateTimeObj)
                print("###### File Moved To BackUp Directory!! ######")

                file.close()

        time.sleep(10)
```

Code can also be found at:

https://egenerationbangladesh-my.sharepoint.com/:u:/g/personal/rafiul_ahmed_egeneration_co/EWUeq5qr631An1Wv9myQRAEBgbaC818meZKsOBHpFV1hBw?e=iS8oeE

Now Data is in FTP server location in FTP_ROOT/TEST_DATA/DWF_PMNT folder. We will use Talend, an open source data integration tool to Transform and Load into Data Warehouse.

Before we can start working on Talend, we need to prepare our database for loading operation.

Database Preparation:

We need to create all the required tables and data modeling. We will use SQL Server Management Studio (SSMS) for database operations.

1. **Let's start by creating a database.**
   We are using SQL Server 2019 Database. Lunch SSMS and connect to SQL Server using credentials. Right click on the "Database" on the left pane. Select create new. Complete the steps as default. We will name it DW_HOSPITAL.

2. **Now we will run the following SQL scripts to create necessary tables.**

   Whole Database script:
   https://egenerationbangladesh-my.sharepoint.com/:u:/g/personal/rafiul_ahmed_egeneration_co/EaO_FM_uATpAoE6wrBpNbtsB6FUaz58gOyMUVs6tHBLIUw?e=FaAnJp

   Dimension Tables:
   https://egenerationbangladesh-my.sharepoint.com/:t:/g/personal/rafiul_ahmed_egeneration_co/EUc41pKCEJdBn-e9a_BS_JYBLX9FQ8zAnhzdyw7c4HNMTQ?e=KOEGxw

   Fact Tables:
   https://egenerationbangladesh-my.sharepoint.com/:t:/g/personal/rafiul_ahmed_egeneration_co/EZUhEa1qt_dFky4Wa7itnfAB5ifCds1gw1bBjXZzuqIBtg?e=xVRk4R

   Look Up Tables:
   https://egenerationbangladesh-my.sharepoint.com/:t:/g/personal/rafiul_ahmed_egeneration_co/EV-6WgaGRnNLsjiQOeEI1BcBIZNOLSzztFpaWY2RQ5bPlQ?e=EuqSgS

3. **We will insert some data into the tables we just created. Run the following script for data insertion.**
   Dimension Table Data Insertion:
   https://egenerationbangladesh-my.sharepoint.com/:u:/g/personal/rafiul_ahmed_egeneration_co/Ea0GHZtOwT1Nl7aAvumi4qQBXq8f6D64IqQ9GPudZgq7vw?e=VoVFqp

   Fact Table Data Insertion:
   https://egenerationbangladesh-my.sharepoint.com/:u:/g/personal/rafiul_ahmed_egeneration_co/EYoTsFvLrx1Dh4hURPm-bBwBgLLHwdXXfoMJrM8HVQm5gg?e=jkLqbI

4. **Then we will create a data model.**
   Right click on "Database Diagrams" and select create new database diagram. It will open a prompt window where you can select your database and tables you want to create diagram with. Complete the process and you will get a page with all the tables you selected. Now drag and drop the primary keys of tables into the referenced tables to create foreign key relationships.

After completing all the steps you will get something like this:

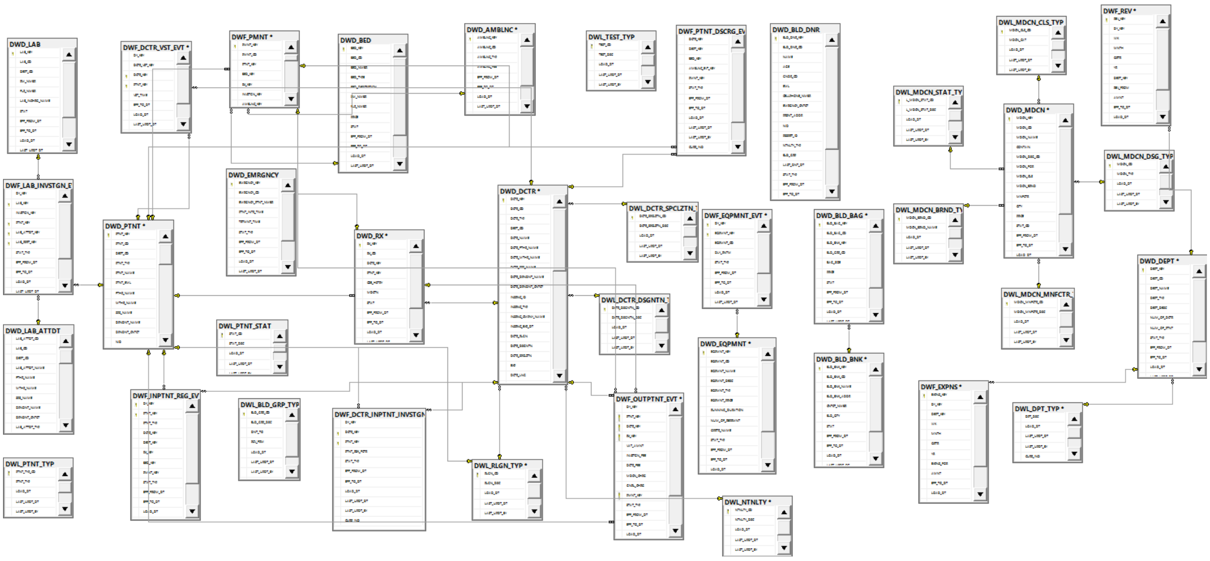


Figure: Data Model in SQL Server

Now that we have completed all pre-requisites of creating a talend job, we will proceed to work on Talend.

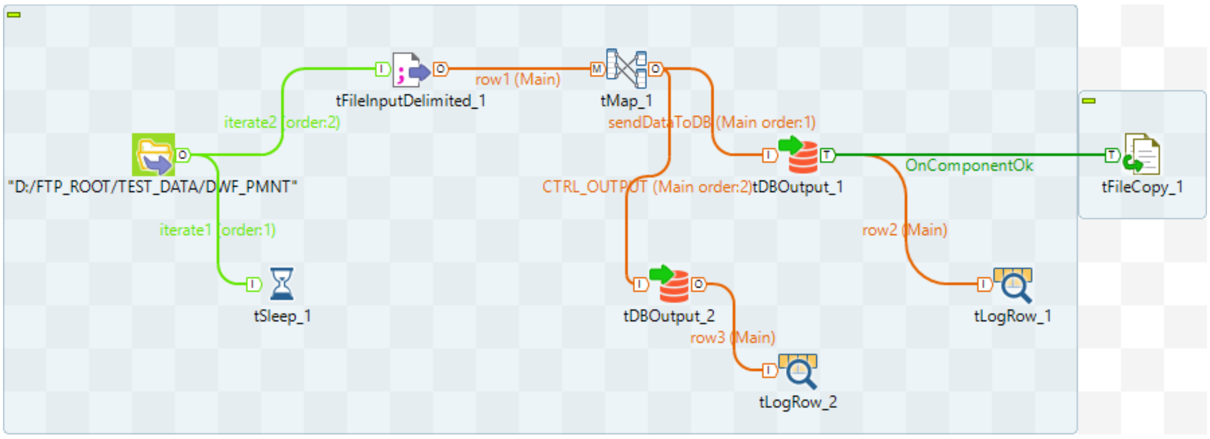
Talend Data Integration:

Talend offers an expansive portfolio of data integration and data management tools. The company's flagship tool, Open Studio for Data Integration, is available via a free open-source license. Talend Integration Cloud is offered in three separate editions (SaaS, hybrid, elastic), and provides broad connectivity, built-in data quality, and native code generation to support big data technologies. Big data components and connectors include Hadoop, NoSQL, MapReduce, Spark, machine leaning, and IoT.

We will design a talend job that will collect data from specified ftp location, transform data according to business needs and finally load into a DW. We will use Talend Open Studio for Big Data (TOSBD 7.3).

1. **Lunch TOSBD 7.3 and create a new job.**
2. **We will use the following components for our job:**
    a. tFileList
    b. tSleep
    c. tFileInputDelimitted
    d. tMap
    e. tDBOutput
    f. tLogRow
    g. tFileCopy
3. **Configure the components according to the following document generated by Talend Open Studio.** https://egenerationbangladesh-my.sharepoint.com/:u:/g/personal/rafiul_ahmed_egeneration_co/EULGyx92I3VKrJ_QdCrX6H4BFoUe3NZiiXgglZ5DUiZLUg?e=zVCxKD

4. **Talend job architecture:**



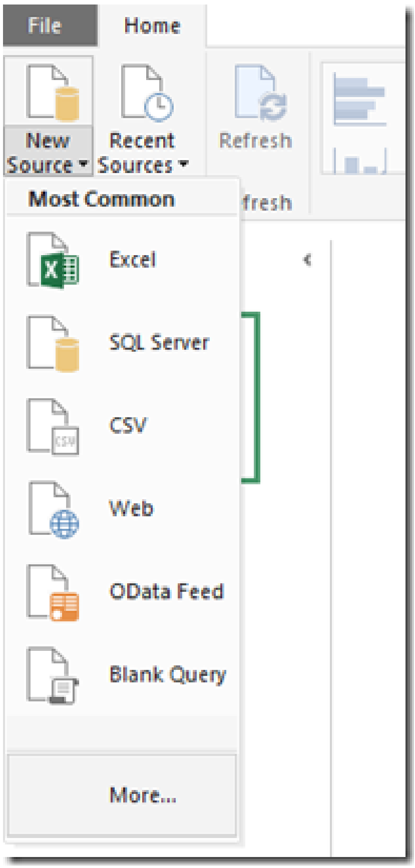
Visualize Data with Power BI:

Transforming your data into rich visuals, Power BI is a customizable data visualization toolset that gives you a complete view of your business. Collaborate and share reports inside and outside your organization, spot trends as they happen, and stay focused on what matters most.

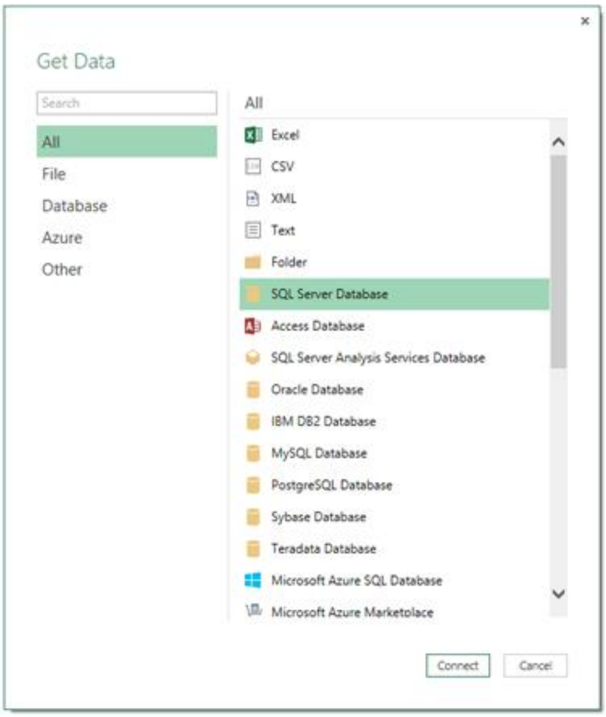
1. **Connect Power BI with SQL Server**

   a. First we need to download and install the Power BI Desktop on your machine and launch it.

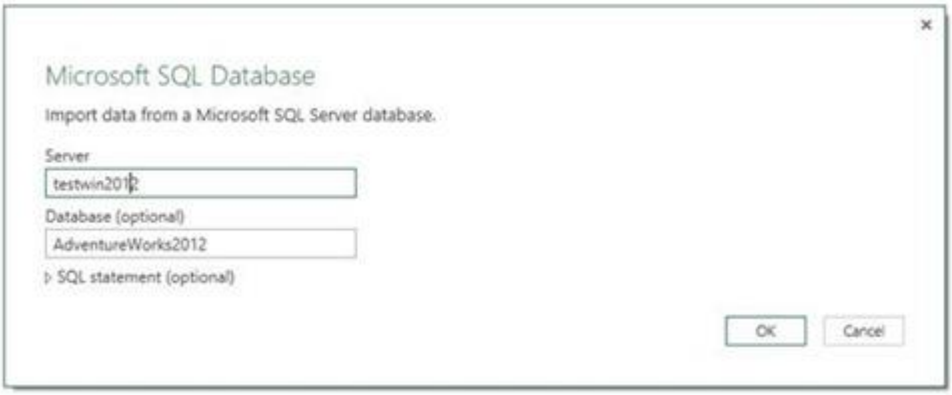
   b. When it first launches, you will see a dialog that will have **Get Data** on it.



   c. Alternatively, you can also click on New Source, which will give you a drop down. You can click on SQL Server, or click on More… to see the full lists of data sources.



d. You will then see a screen to enter the SQL Server name and the Database name. This would be where the data is going to come from.



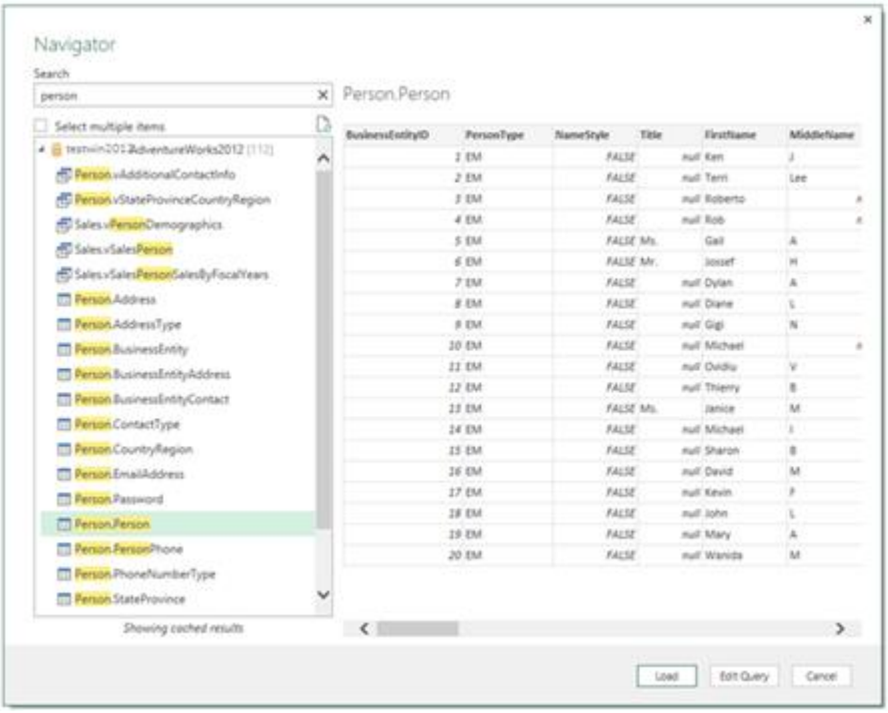
Microsoft SQL Database

Import data from a Microsoft SQL Server database.

Server: testwin2012

Database (optional): AdventureWorks2012

e. Once you are done with entering the information you need, click OK. If you left the SQL Statement field blank, the Navigator window will come up and show a list of tables, views and functions listed for that database. You can select the item from which the data needs to come from. The search option can also be used to filter for a specific item.



person

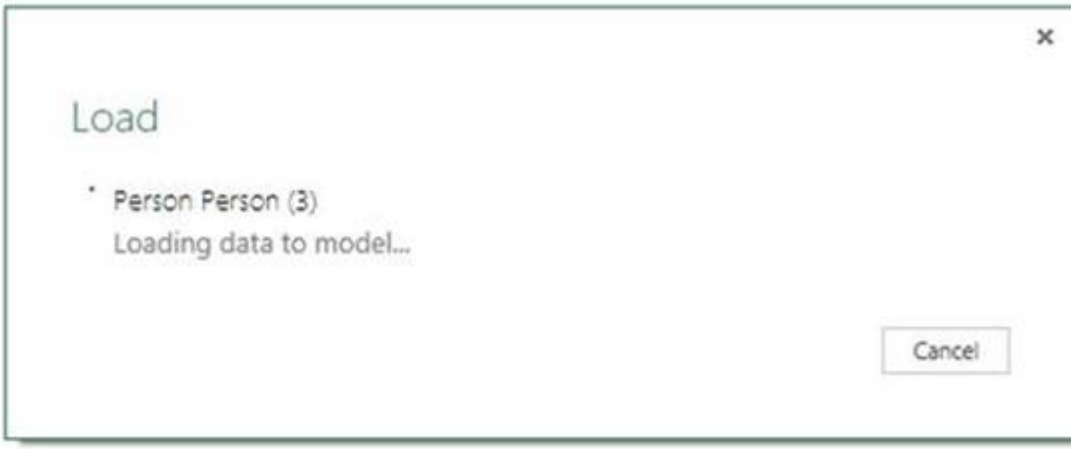
testwin2012 AdventureWorks2012 [112]

- Person.vAdditionalContactInfo
- Person.vStateProvinceCountryRegion
- Sales.vPersonDemographics
- Sales.vSalesPerson
- Sales.vSalesPersonSalesByFiscalYears
- Person.Address
- Person.AddressType
- Person.BusinessEntity
- Person.BusinessEntityAddress
- Person.BusinessEntityContact
- Person.ContactType
- Person.CountryRegion
- Person.EmailAddress
- Person.Password
- Person.Person
- Person.PersonPhone
- Person.PhoneNumberType
- Person.StateProvince

Person.Person

BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName
1	EM	FALSE	null	Ken	J
2	EM	FALSE	null	Terri	Lee
3	EM	FALSE	null	Roberto	
4	EM	FALSE	null	Rob	
5	EM	FALSE	Ms.	Gail	A
6	EM	FALSE	Mr.	Jossef	H
7	EM	FALSE	null	Dylan	A
8	EM	FALSE	null	Diane	L
9	EM	FALSE	null	Gigi	N
10	EM	FALSE	null	Michael	
11	EM	FALSE	null	Ovidiu	V
12	EM	FALSE	null	Thierry	B
13	EM	FALSE	Ms.	Janice	M
14	EM	FALSE	null	Michael	I
15	EM	FALSE	null	Sharon	B
16	EM	FALSE	null	David	M
17	EM	FALSE	null	Kevin	F
18	EM	FALSE	null	John	L
19	EM	FALSE	null	Mary	A
20	EM	FALSE	null	Wanida	M

f. Once you have selected the item, click Load. This will pull the data into the Data Model.



Load

- Person Person (3)

Loading data to model...

2. **Develop the necessary Reports and Dashboards.**
3. **Schedule Refresh:**

   Follow the bellow link to configure schedule refresh

   https://docs.microsoft.com/en-us/power-bi/connect-data/refresh-scheduled-refresh