Lab Work 4: Extraction of Excel Data in Local SQL Server in Monthly Basis

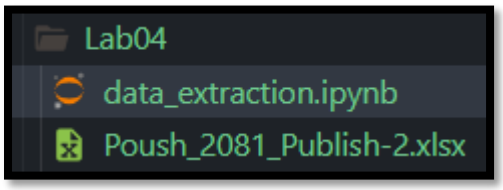
OBJECTIVE:

- To apply data integration using Python to extract, clean, and load Excel data into SQL Server.
- To automate monthly imports with scripting and scheduling tools while building ETL skills for structured data processing.

SOURCE CODE

1. Downloading the Excel File

Downloaded the excel from https://www.nrb.org.np/category/monthly-statistics/?department=bfr



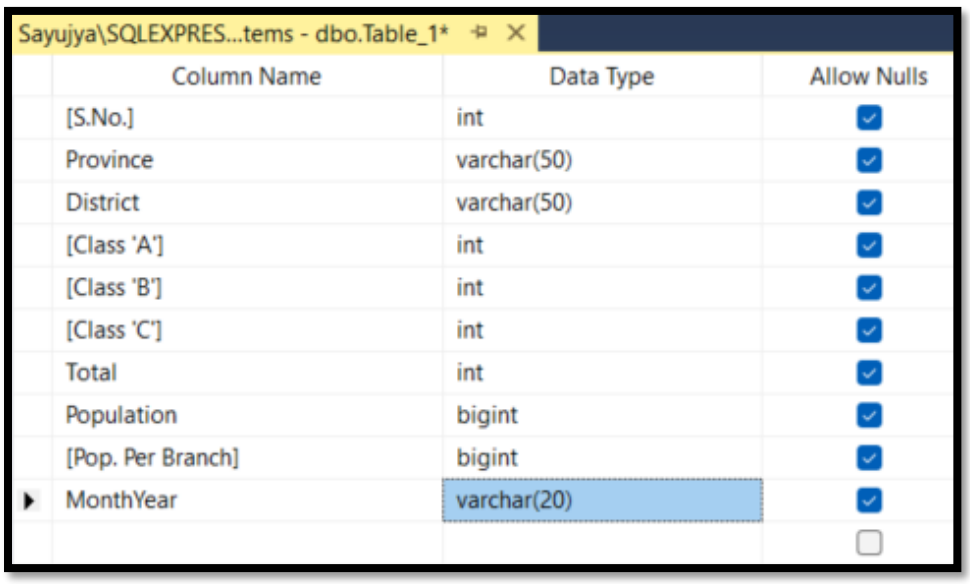
2. Installing Required Python Libraries

We installed python libraries: pandas, sqlalchemy, pyodbc, and openpyxl

```
sayuj@Sayujya MINGW64 ~/OneDrive/Desktop/BIM054/00_BIM_Repo (main)
$ pip install pandas sqlalchemy pyodbc openpyxl
```

3. Setting up the database

A table in SQL named MonthlyStatistics was created and ensured the schema matched the structure of the Excel file and a column named was included as the name of the month and year same as in name of the excel file.



Column Name	Data Type	Allow Nulls
[S.No.]	int	☑
Province	varchar(50)	☑
District	varchar(50)	☑
[Class 'A']	int	☑
[Class 'B']	int	☑
[Class 'C']	int	☑
Total	int	☑
Population	bigint	☑
[Pop. Per Branch]	bigint	☑
MonthYear	varchar(20)	☑
		☐

4. Excel Sheet Processing

The C14 sheet from the excel data of "District wise List of Banks and Financial Institutions Branches", was used.

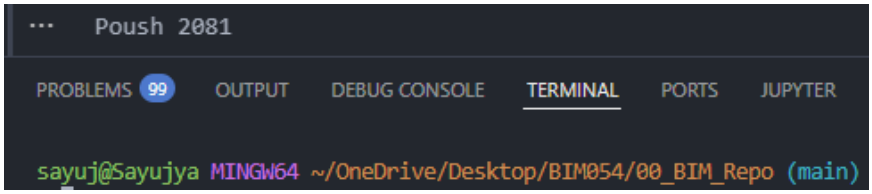
The data was pre-processed using the pandas library

4.1. Set up Python Environment

```
# Setting up the Python Environment
import pandas as pd
import pyodbc
from sqlalchemy import create_engine
import os
import re
```

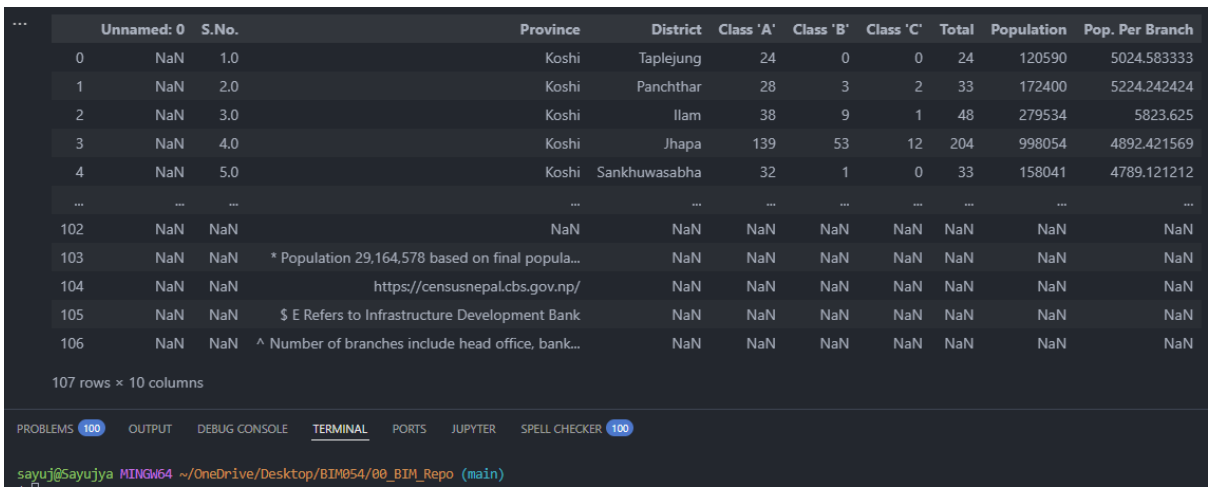
4.2. Extracting the month and year from the file name

```
# Extracting the month and year
file_name = os.path.basename("./Poush_2081_Publish-2.xlsx")
match = re.search(r"([A-Za-z]+)_(\d{4})", file_name)
month_year = f"{match.group(1)} {match.group(2)}"
print(month_year)
```



4.3. Loading and preparing the data

```
# Reading the Data
data = pd.read_excel("./Poush_2081_Publish-2.xlsx", sheet_name="C14",
skiprows=2)
display(data)
```



	Unnamed: 0	S.No.	Province	District	Class 'A'	Class 'B'	Class 'C'	Total	Population	Pop. Per Branch
0	NaN	1.0	Koshi	Taplejung	24	0	0	24	120590	5024.583333
1	NaN	2.0	Koshi	Panchthar	28	3	2	33	172400	5224.242424
2	NaN	3.0	Koshi	Ilam	38	9	1	48	279534	5823.625
3	NaN	4.0	Koshi	Jhapa	139	53	12	204	998054	4892.421569
4	NaN	5.0	Koshi	Sankhuwasabha	32	1	0	33	158041	4789.121212
...	...	...	...	...	...	...	...	...	...	...
102	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
103	NaN	NaN	* Population 29,164,578 based on final popula...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
104	NaN	NaN	https://censusnepal.cbs.gov.np/	NaN	NaN	NaN	NaN	NaN	NaN	NaN
105	NaN	NaN	$ E Refers to Infrastructure Development Bank	NaN	NaN	NaN	NaN	NaN	NaN	NaN
106	NaN	NaN	^ Number of branches include head office, bank...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

107 rows × 10 columns

```
# Data cleaning
# Drop the index, or unnamed column
data.drop(columns=['Unnamed: 0'], inplace=True)

# Remove any of the na values
data = data.dropna()

#As pop. per branch is float, convert it to int
data["Pop. Per Branch"] = data["Pop. Per Branch"].round().astype(int)

# Add new column for MonthYear
data["MonthYear"] = month_year

display(data)
```

	S.No.	Province	District	Class 'A'	Class 'B'	Class 'C'	Total	Population	Pop. Per Branch	MonthYear
0	1.0	Koshi	Taplejung	24	0	0	24	120590	5024	Poush 2081
1	2.0	Koshi	Panchthar	28	3	2	33	172400	5224	Poush 2081
2	3.0	Koshi	Ilam	38	9	1	48	279534	5823	Poush 2081
3	4.0	Koshi	Jhapa	139	53	12	204	998054	4892	Poush 2081
4	5.0	Koshi	Sankhuwasabha	32	1	0	33	158041	4789	Poush 2081
...	...	...	...	...	...	...	...	...	...	...
78	73.0	Sudur Paschim	Achham	18	3	0	21	228852	10897	Poush 2081
79	74.0	Sudur Paschim	Kanchanpur	61	11	1	73	513757	7037	Poush 2081
80	75.0	Sudur Paschim	Darchula	21	2	0	23	133310	5796	Poush 2081
81	76.0	Sudur Paschim	Baitadi	22	2	0	24	242157	10089	Poush 2081
82	77.0	Sudur Paschim	Dadeldhura	24	1	0	25	139602	5584	Poush 2081

77 rows × 10 columns

PROBLEMS 99 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER SPELL CHECKER 99

sayuj@Sayujya MINGW64 ~/OneDrive/Desktop/BIM054/00_BIM_Repo (main)

4.4. Inserting the processed data into SQL server

```
# Connect to SQL Server
# DB Connection
server = r"SAYUJYA\SQLEXPRESS"
database = "BusinessInformationSystems"

# Establish Connection
conn_str =
f"mssql+pyodbc://{server}/{database}?driver=ODBC+Driver+17+for+SQL+Server"
engine = create_engine(conn_str)

# Insert Data into SQL Server
data.to_sql("MonthlyStatistics", engine, if_exists="append", index=False)
print("Data inserted into SQL Server successfully")
```

```
Data inserted into SQL Server successfully
```

PROBLEMS 100 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

```
sayuj@Sayujya MINGW64 ~/OneDrive/Desktop/BIM054/00_BIM_Repo (main)
```

SQLQuery1.sql - SA...AYUJYA\sayuj (81))

```
SELECT TOP (1000) [S.No.]
      ,[Province]
      ,[District]
      ,[Class 'A']
      ,[Class 'B']
      ,[Class 'C']
      ,[Total]
      ,[Population]
      ,[Pop. Per Branch]
      ,[MonthYear]
  FROM [BusinessInformationSystems].[dbo].[MonthlyStatistics]
```

150 %

Results Messages

	S.No.	Province	District	Class 'A'	Class 'B'	Class 'C'	Total	Population	Pop. Per Branch	MonthYear
1	1	Koshi	Taplejung	24	0	0	24	120590	5024	Poush 2081
2	2	Koshi	Panchthar	28	3	2	33	172400	5224	Poush 2081
3	3	Koshi	Ilam	38	9	1	48	279534	5823	Poush 2081
4	4	Koshi	Jhapa	139	53	12	204	998054	4892	Poush 2081
5	5	Koshi	Sankhuwasabha	32	1	0	33	158041	4789	Poush 2081
6	6	Koshi	Bhojpur	24	0	1	25	157923	6316	Poush 2081
7	7	Koshi	Terhathum	21	3	0	24	88731	3697	Poush 2081
8	8	Koshi	Dhankuta	28	9	1	38	150599	3963	Poush 2081
9	9	Koshi	Morang	181	61	5	247	1148156	4648	Poush 2081
10	10	Koshi	Sunsari	134	39	13	186	926962	4983	Poush 2081
11	11	Koshi	Solukhumbu	28	0	0	28	104851	3744	Poush 2081
12	12	Koshi	Khotang	19	6	0	25	175298	7011	Poush 2081
13	13	Koshi	Okhaldhunga	23	4	0	27	139552	5168	Poush 2081
14	14	Koshi	Udayapur	43	7	1	51	340721	6680	Poush 2081
15	15	Madhesh	Saptari	56	7	2	65	706255	10865	Poush 2081
16	16	Madhesh	Siraha	83	12	3	98	739953	7550	Poush 2081
17	17	Madhesh	Dhanusa	87	8	13	108	867747	8034	Poush 2081
18	18	Madhesh	Mahottari	64	7	10	81	706994	8728	Poush 2081
19	19	Madhesh	Sarlahi	82	14	9	105	862470	8214	Poush 2081
20	20	Madhesh	Rautahat	58	9	3	70	813573	11622	Poush 2081
21	21	Madhesh	Bara	81	13	6	100	763137	7631	Poush 2081
22	22	Madhesh	Parsa	70	14	7	91	654471	7191	Poush 2081
23	23	Bagmati	Dolakha	31	3	0	34	172767	5081	Poush 2081
24	24	Bagmati	Ramechhap	22	2	0	24	170302	7095	Poush 2081
25	25	Bagmati	Sindhuli	34	2	2	38	300026	7895	Poush 2081
26	26	Bagmati	Rasuwa	28	1	1	30	46689	1556	Poush 2081
27	27	Bagmati	Dhading	62	17	0	79	325710	4122	Poush 2081
28	28	Bagmati	Nuwakot	40	15	1	56	263391	4703	Poush 2081
29	29	Bagmati	Sindhupalchok	36	11	0	47	262624	5587	Poush 2081
30	30	Bagmati	Kavrepalanchok	70	14	8	92	364039	3956	Poush 2081