REPORTING DATA PIPELINE

PRANAV BEEJMOHUN

GOALS AND OBJECTIVES

Objective n° 1

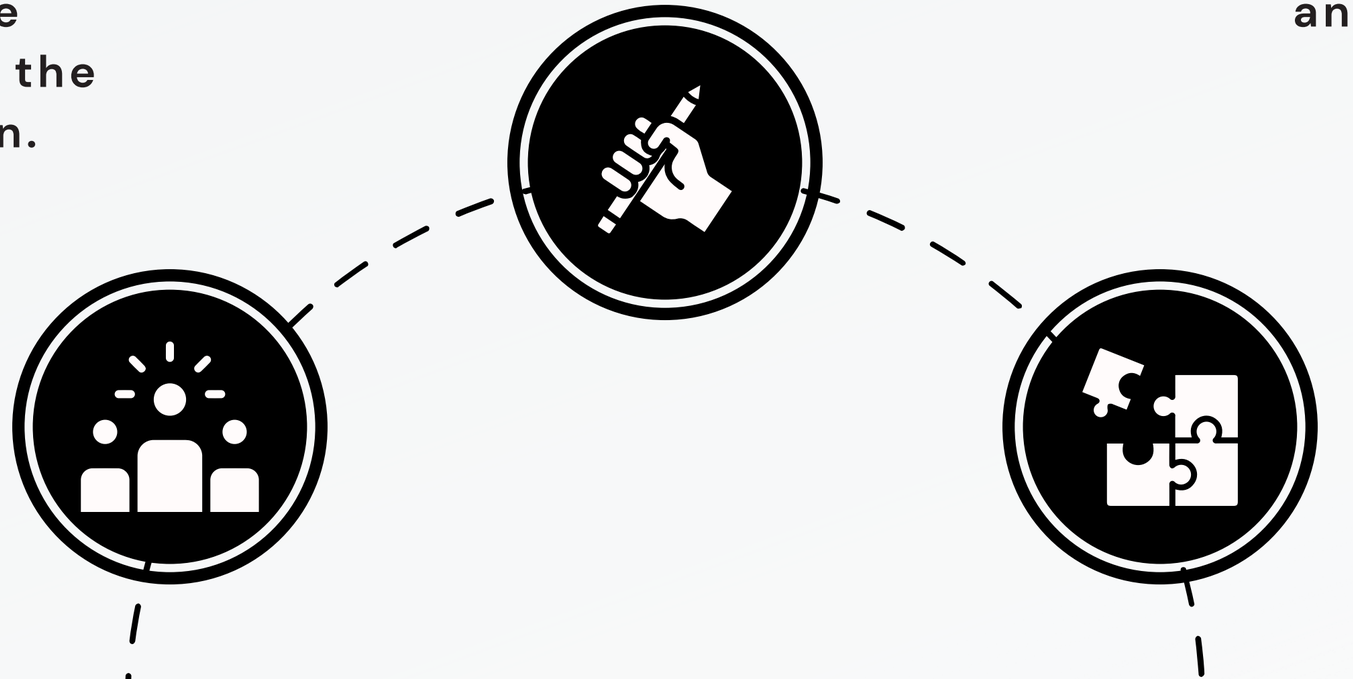
Help to guide the top management in formulating sustainable strategies for the organization.

Objective n° 2

create a data pipeline that can cater for all the reporting requirement.

Objective n° 3

Generate the country macroeconomic analysis report.



TOOLS USED

Jupyter Notebook

Python coding environment for the ETL process

PostgreSQL

Database environment for relational and non-relational database

Power BI

Data Visualisation tool for Report generation

DATA SOURCES

1.Corruption Perception Index Data Set
·Provided in excel format (.xlsx)

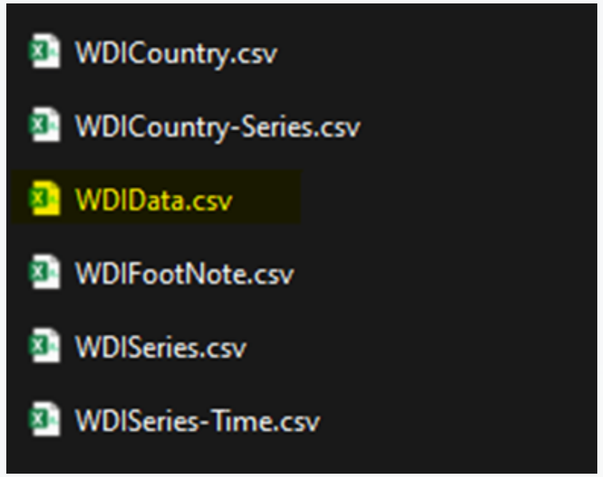
2.World Development Indicators
·Format: CSV files

Note: The data sources are updated every year.
Assumption: The naming and structure of the files remain the same
When updated, a column is added with data of the previous year.

DATA SOURCES

The chosen file from the World Development Indicators CSV files was the ‘WDIData.csv’ file since it contained all necessary data to generate reports for the indicators that follow:

i. Total fisheries production (metric tons)
ii. Agricultural land (sq. km)
iii. Time required to start a business (days)
iv. New businesses registered (number)
v. Employment in agriculture (% of total employment) (modelled ILO estimate)
vi. Self-employed, total (% of total employment) (modelled ILO estimate)



ETL PROCESSES

CORRPUTION PERCEPTION INDEX

CODE: CPI.IPYNB

CORRUPTION PERCEPTION INDEX DATA SET

Extraction

File format: Excel (.xlsx)

The excel file is read using the pandas library

```
import os
import numpy as np
import pandas as pd
import psycopg2

#Import the CSV file into a pandas df
df = pd.read_excel('Corruption Perception Index Data Sets.xlsx', sheet_name="CPI Timeseries 2012 - 2020")
```

CORRUPTION PERCEPTION INDEX DATA SET

Transformation - Headers

The first 2 rows are banners so they were removed and the third row was promoted to headers

1	Corruption Perceptions Index 2020: Score timeseries since 2012									
2										
3	Country	ISO3	Region	CPI score 2020	Rank 2020	Sources 2020	Standard error 2020	CPI score 2019	Rank 2019	Sources 2019

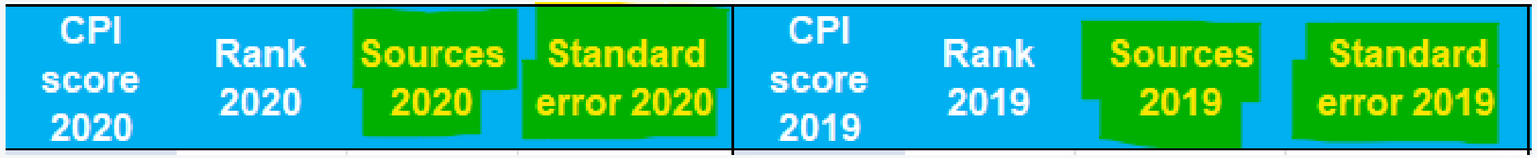
```
# Promote the second row to headers
new_headers = df.iloc[1]
df.columns = new_headers

# Drop the first two rows
df = df.iloc[2:].reset_index(drop=True)
```

CORRUPTION PERCEPTION INDEX DATA SET

Transformation - Remove redundant columns

The columns representing the number of sources and the standard error are independent features and are therefore removed.




```
columns_to_drop = [col for col in df.columns if col.startswith('Standard') or col.startswith('Sources')]
df = df.drop(columns=columns_to_drop)
```

CORRUPTION PERCEPTION INDEX DATA SET

Transformation - Prepare Loading Phase

The creation of the table in the database should be dynamic since the file is updated every year.

To generate the SQL table creation script dynamically:

- Headers are put in lower characters
- Spaces are replaced by underscore
- The first 3 features have data types varchar and they others are converted to int

```
#Clean the colum headers and remove all extra symbols, spaces and capital letter
df.columns = [x.lower().replace(" ", "_") for x in df.columns]

df.fillna(0, inplace=True)

df = df.apply(lambda x: x.astype(int) if x.name not in ['country', 'iso3', 'region'] else x)



replacements = {
    'object' : 'varchar',
    'float64' : 'float',
    'int32' : 'int',
    'datetime64' : 'timestamp'
}

col_str = ", ".join("{} {}".format(n, d) for (n, d) in zip(df.columns, df.dtypes.replace(replacements)))
```

CORRUPTION PERCEPTION INDEX DATA SET

Loading

A connection is set to a PostgreSQL database and the Corruption Perception Index data is loaded using the dynamic SQL statement seen above.

```
# conn_string = 'host="localhost", port="5432", dbname="WDI", user="postgres", password="admin"'
conn = psycopg2.connect(host="localhost", port="5432", dbname="WDI", user="postgres", password="admin")
cursor = conn.cursor()
# print('Opened databse successfully')

# drop table with same name
cursor.execute("drop table if exists corruption_perception_index_data_sets;")
# create table
cursor.execute("create table corruption_perception_index_data_sets(%s)" %(col_str))
# print('success')
```

```
# save df to csv
df.to_csv('corruption_perception_index_data_sets.csv', header=df.columns, index=False, encoding='utf-8')

#open the csv file

my_file = open('corruption_perception_index_data_sets.csv')
# print('file opened in memory')

SQL_STATEMENT = """
COPY corruption_perception_index_data_sets FROM STDIN WITH
    CSV
    HEADER
    DELIMITER AS ','
"""
```

ETL PROCESSES

WORLD DEVELOPMENT INDICATORS

CODE: WDI.IPYNB

WORLD DEVELOPMENT INDICATORS

Extraction

File format: CSV

The csv file is read using the pandas library

```
import os
import numpy as np
import pandas as pd
import psycopg2

# Read the csv file
df = pd.read_csv('WDIData.csv')
```

WORLD DEVELOPMENT INDICATORS

Transformation - Headers & Prepare loading phase

- ‘yr’ was added in front of every year header to dynamically create SQL query later while respecting naming convention in SQL queries.
- Spaces are replaced by underscore in headers.
- Any unnamed column is removed from the dataset.
- The first 4 features have data types varchar and they others are converted to float

```
df.columns = ['yr' + col if col not in df.columns[:4] and not col.startswith('yr') else col for col in df.columns]
df.columns = [x.lower().replace(" ", "_") for x in df.columns]
df = df.filter(regex='^(?!yrunnamed)')
df.fillna(0, inplace=True)

replacements = {
    'object' : 'varchar',
    'float64' : 'float',
    'int32' : 'int',
    'datetime64' : 'timestamp'
}

col_str = ", ".join("{} {}".format(n, d) for (n, d) in zip(df.columns, df.dtypes.replace(replacements)))
```

WORLD DEVELOPMENT INDICATORS

Loading

A connection is set to a PostgreSQL database and the WDI dataset is loaded using the dynamic SQL statement seen above.

```
# conn_string = 'host="localhost", port="5432", dbname="WDI", user="postgres", password="admin"'
conn = psycopg2.connect(host="localhost", port="5432", dbname="WDI", user="postgres", password="admin")
cursor = conn.cursor()
print('Opened databse successfully')

# drop table with same name
cursor.execute("drop table if exists WDIData;")

cursor.execute("create table WDIData(%s)" % (col_str))

# save df to csv
df.to_csv('wdidata.csv', header=df.columns, index=False, encoding='utf-8')
```

```
my_file = open('wdidata.csv')
print('file opened in memory')

SQL_STATEMENT = """
COPY wdidata FROM STDIN WITH
    CSV
    HEADER
    DELIMITER AS ','
"""

cursor.copy_expert(sql=SQL_STATEMENT, file=my_file)
print('file copied to db')

cursor.execute("grant select on WDIData to public")
conn.commit()
cursor.close()
print('table WDIData import to db completed')
```

WORLD DEVELOPMENT INDICATORS

Loading

Note:

- All indicators were added to the Database and the selection of indictors to display was done in Power Query
- The features representing the years between 1960 and 2022 are not pivoted in the ETL process. They are pivoted in Power Query to make selection and filters.

REPORT - CPI

Country	CPI Score 2020	CPI Score selected Year	CPI Score Change
Afghanistan	19	8	11
Albania	36	33	3
Algeria	36	34	2
Angola	27	22	5
Argentina	42	35	7
Armenia	49	34	15
Australia	77	85	-8
Austria	76	69	7
Azerbaijan	30	27	3
Bahamas	63	71	-8
Bahrain	42	51	-9
Bangladesh	26	26	0
Barbados	64	76	-12
Belarus	47	31	16
Belgium	76	75	1
Benin	41	36	5
Bhutan	68	63	5
Bolivia	31	34	-3
Bosnia and Herzegovina	35	42	-7
Botswana	60	65	-5
Brazil	38	43	-5
Brunei Darussalam	60	55	5
Bulgaria	44	41	3
Burkina Faso	40	38	2
Burundi	19	19	0

Country

All

Select Year

2012

The Year can be chosen from the slicer and the score of 2020 is compared to the score of the year selected

Note: The year 2020 is taken as reference since there was no data for the current year

REPORT - WDI

Country	Country Code	Score Current Year	Score Last Year	Score Change	Rank Current Year	Rank Last Year	Change in Rank
Afghanistan	AFG	379,140.00	379150	-10.00	77	77	0
Africa Eastern and Southern	AFE	6,789,159.83	6747105	42,054.95	26	26	0
Africa Western and Central	AFW	3,529,813.27	3514584	15,229.46	39	39	0
Albania	ALB	12,013.00	12010	3.00	185	183	2
Algeria	DZA	413,981.90	413880	101.90	70	69	1
American Samoa	ASM	26.30	26	0.30	249	248	1
Andorra	AND	187.60	198	-10.10	227	226	1
Angola	AGO	453,030.00	453160	-130.00	65	65	0
Antigua and Barbuda	ATG	90.00	90	0.00	235	235	0
Arab World	ARB	5,212,816.33	5467540	-254,723.44	34	33	1
Argentina	ARG	1,247,419.00	1253184	-5,765.00	51	52	-1
Armenia	ARM	16,830.00	16981	-151.00	176	174	2
Aruba	ABW	20.00	20	0.00	250	249	1
Australia	AUS	3,870,760.00	3961210	-90,450.00	37	37	0
Austria	AUT	27,352.00	27573	-221.00	152	150	2
Azerbaijan	AZE	47,683.00	47687	-4.00	136	135	1
Bahamas, The	BHS	130.00	130	0.00	230	230	0
Bahrain	BHR	86.40	87	-1.00	236	236	0
Bangladesh	BGD	91,230.00	91280	-50.00	120	119	1
Barbados	BRB	130.00	140	-10.00	230	228	2
Belarus	BLR	87,960.00	88750	-790.00	122	121	1
Belgium	BEL	13,320.00	13357	-37.00	179	178	1
Belize	BLZ	1,600.00	1570	30.00	207	206	1

Indicator_Name

Agricultural land (sq. km)

Country_Name

All

YEAR

2012

The year, Indicator and Country Names can be selected from the slicers.

The user can make multi-selection on the Country Name slicer

A selection of indicators were added to the slicer as requested in the question

PROJECT LINK

PranavBeejmohun/
WDI_Reporting

1 Contributor | 0 Issues | 0 Stars | 0 Forks

PranavBeejmohun/WDI_Reporting

Contribute to PranavBeejmohun/WDI_Reporting development by creating an account on GitHub.

GitHub

REPORT LINK

POWER BI DASHBOARD ZIP

https://drive.google.com/file/d/1xUPfpZMv7w7cMob1qJjT6IPEMv7le-ll/view?usp=sharing