UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

BSD2343 DATA WAREHOUSING

2023/2024 SEMESTER II

(SDG 8 : DECENT WORK AND ECONOMIC GROWTH)

FACTOR AFFECT ECONOMIC GROWTH IN COUNTRY

PREPARED FOR :

DR AZUANA BINTI RAMLI

GROUP NAME : DICING

MATRIC ID	NAME	SECTION
SD22049	THAM ZHI YUN	01G
SD22021	AMIRAH YASMIN BINTI ZAILANEE	02G
SD22052	NURUL HIDAYAH BINTI ROSLAN	02G
SD22009	HASNIZA BINTI MOHD SOMAN	02G
SD22046	MUHAMMAD ZAMIL SYAFIQ BIN ZANZALI	02G

ABSTRACT

Under the framework of Sustainable Development Goal (SDG) 8, Economic growth and employment are two critical indicators that drive a nation's development and overall well-being. The objectives of this project are to determine the relationship of the rich level of a country with GDP per capita in different countries, to identify factors influencing the rich level of a country, and to find out the pattern of growth of economics in Malaysia. In this study, two datasets will be used to achieve the objective. To extract, load and transform the data, ETL process will be handled by using Jupyter Notebook ang pgAdmin, and the architecture of the project will be discussed. Plus, by determining the relationship between the table, it is found this is a star schema build for this project. Lastly, the data will be analysed by using OLAP and visualised by using PowerBI. These results of the project may lead to a meaningful insight to the economy across the world as well as give effective suggestions to improve the richest level of Malaysia. In conclusion, the economy of Malaysia can grow rapidly as well as Malaysians are able to have various job opportunities.

TABLE OF CONTENT

1.0 BACKGROUND

1.1 Project Background

Economic growth and employment are two critical indicators that drive a nation's development and overall well-being. Under the framework of Sustainable Development Goal (SDG) 8, which aims for decent work and economic growth, it is vital to understand how these two variables interact and influence each other. This project aims to explore the relationship between economic growth, measured by GDP per capita and other economic factors, and employment rates across various nations. By analysing diverse datasets, including those related to GDP, tourism, cost of living, and unemployment rates, this study seeks to provide a comprehensive understanding of how economic expansions or contractions impact employment levels (Wignaraja G & Jinjarak Y,2015).

In the current global landscape, where economic uncertainty has been increased by events such as the COVID-19 pandemic, understanding the dynamics between economic growth and employment is more crucial than ever. This project is significant as it provides insights that can help policymakers design better strategies to foster economic resilience and ensure sustainable employment opportunities. By identifying the key economic drivers that are most closely associated with employment rates, this study will contribute to efforts aimed at achieving SDG 8, which not only promotes sustained, inclusive, and sustainable economic growth but also full and productive employment for all (Koirala J,2020).

Utilising a cross-sectional dataset from various countries, this study will employ statistical analysis methods to examine the relationships between different economic indicators and employment rates. The datasets include measures of GDP per capita, tourism revenue, cost of living indexes, and unemployment rates. These variables will be analysed to determine how variations in economic growth relate to changes in employment levels across different regional and economic contexts. The methodology will focus on correlation analysis and regression models to ascertain the strength and nature of these relationships (Uysal P & Warren M,2020).

The project is expected to reveal significant patterns and trends that link economic growth with employment rates. For instance, higher GDP per capita might correlate with lower unemployment rates, but this relationship could vary significantly depending on the region and economic sector. Additionally, the study will explore how external factors like tourism and cost of living impact economic growth and employment, providing a more shaded understanding of the economic ecosystem. These findings will not only add to the academic literature but also offer practical insights for economic and employment policy.

Upon completion, this study will provide valuable data-driven insights that can influence policy decisions aimed at promoting economic stability and job creation. The findings will be particularly useful for governments and international organisations as they develop strategies to achieve SDG 8. Furthermore, the research could pave the way for future studies exploring other related aspects, such as the impact of technology and innovation on economic growth and employment. Ultimately, this project aims to contribute to a sustainable economic future where growth leads to enhanced employment opportunities for all.

1.2 Description of Data

Two datasets are used in this project to determine the world economy by corruption, tourism, unemployment and cost of living all around the world as well as to determine the Gross Domestic Product (GDP) value of countries around the world in the year 2022. The descriptions of each variable in each data table used are listed below.

Corruption Table

VARIABLE	DATA TYPE	DESCRIPTION
anual_income	Integer	Average annual income in USD for each country
corruption_index	Integer	Corruption perception index value of each country
country	String	Name of country

Table 1.1 Description Table for Corruption

Cost of Living Table

VARIABLE	DATA TYPE	DESCRIPTION
cost_index	Float	Cost of living index value
country	String	Name of country
monthy_income	Integer	Average monthly income in USD
purchasing_power_index	Float	Purchasing power index value

Table 1.2 Description Table for Cost of Living

Richest Ranking Table

VARIABLE	DATA TYPE	DESCRIPTION
country	String	Name of country
gdp_per_capita	Integer	Gross Domestic Product per Capita of each country in USD

Table 1.3 Description Table for Richest Ranking

Tourism Table

VARIABLE	DATA TYPE	DESCRIPTION
country	String	Name of country
tourists_in_millions	Float	Total number of tourists in millions
receipts_in_billions	Float	Total tourism receipts in billions
receipts_per_tourist	Integer	Average receipts per tourist
percentage_of_gdp	Float	Tourism receipts as a percentage of GDP

Table 1.4 Description Table for Tourism

Unemployment Table

VARIABLE	DATA TYPE	DESCRIPTION
country	String	Name of country
unemployment_rate	Float	Unemployment rate as a percentage

Table 1.5 Description Table for Unemployment

GDP per Capita Penn World Table

VARIABLE	DATA TYPE	DESCRIPTION
Entity	String	Name of country
Code	String	A unique name for each country
Year	Integer	Year of GDP
GDP per capita (output, multiple price benchmarks)	Float	Gross Domestic Product per Capita of each country in USD, adjusted for multiple price benchmarks

Table 1.6 Description Table for GDP per Capita Penn World

GDP per Capita Maddison Table

VARIABLE	DATA TYPE	DESCRIPTION
Entity	String	Name of country
Code	String	A unique name for each country
Year	Integer	Year of GDP
417485-annotations	String	Typically provide additional information or metadata about the data points
GDP per capita	Float	The total output of goods and services generated by all residents of an economy.

Table 1.7 Description Table for GDP per capita Maddison

GDP per Capita World Bank Table

VARIABLE	DATA TYPE	DESCRIPTION
Entity	String	Name of country
Code	String	A unique name for each country
Year	Integer	Year of GDP
GDP per capita, PPP (constant 2017 international $)	Float	GDP per capita or total GDP for each country, measured in constant 2017 international dollars using PPP.

Table 1.8 Description Table for GDP per Capita World Bank

National GDP WB Table

VARIABLE	DATA TYPE	DESCRIPTION
Entity	String	Name of country
Code	String	A unique name for country
Year	Integer	Year of GDP
GDP, PPP (constant 2017 international $)	Integer	Refers to the GDP per person in a country, measured in constant 2017 international dollars using PPP

Table 1.9 Description Table for National GDP WB

National GDP Penn World Table

VARIABLE	DATA TYPE	DESCRIPTION
Entity	String	Name of country
Code	String	A unique name for country
Year	Integer	Year of GDP
GDP (output, multiple price benchmarks)	Integer	GDP of the country in a given year, calculated using multiple price benchmarks

Table 1.10 Description Table for National GDP Penn World

1.3 Problem to be Solved

Countries must maintain a high rate of economic growth while identifying and resolving differences in GDP and various manufacturing sectors as the global economy continues to change. Given how much a nation's economy depends on the diversity and balance of its industries, this is an important topic that has to be carefully considered.

The key problem that emerges is how nations may manage industrial diversity and economic growth while making sure that the advantages of growth are distributed fairly among various industries and geographical areas.

1. How do factors such as GDP influence a country's economic performance?
2. What are the key factors that contribute to a country's rich level of economic growth?
3. What is the pattern of growth of economics in Malaysia?

1.4 Objectives

1. To determine the relationship of the rich level of a country with GDP per capita in different countries.
2. To identify factors influencing the rich level of a country.
3. To find out the pattern of growth of economics in Malaysia.

1.5 Data Schema

A data schema is an assemblage of database elements, such as synonyms, views, tables, and indexes. Schema objects in data warehousing-specific schema models can be arranged in several ways. The ten tables that make up the list dataset are corruption, cost of living, richest ranking, tourism, unemployment, GDP per capita Penn World, GDP per capita Maddison, GDP per capita World Bank, National GDP WB, and National GDP Penn World.

In this step, the Jupyter Notebook is used to identify the type of each data by showing the data schema. (**Refer to Appendix**)

```
corruption=pd.read_csv("corruption.csv")
```

```
corruption.dtypes
```

```
country              object
annual_income         int64
corruption_index      int64
dtype: object
```

```
corruption.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110 entries, 0 to 109
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   country           110 non-null    object
 1   annual_income     110 non-null    int64 
 2   corruption_index  110 non-null    int64 
dtypes: int64(2), object(1)
memory usage: 2.7+ KB
```

Figure 1.1 Data Schema for Corruption

Figure 1.1 shows the data schema for the corruption table which consists of 3 columns. The data frame consists of only two data types which are string and integer. Only one column has string data type while the rest are integer data types.

```
cost_living=pd.read_csv("cost_of_living.csv")
```

```
cost_living.dtypes
```

```
country                     object
cost_index                 float64
monthly_income               int64
purchasing_power_index     float64
dtype: object
```

```
cost_living.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107 entries, 0 to 106
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   country                 107 non-null    object 
 1   cost_index              107 non-null    float64
 2   monthly_income          107 non-null    int64  
 3   purchasing_power_index  107 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 3.5+ KB
```

Figure 1.2 Data Schema for Cost of Living

Figure 1.2 shows the data schema for the cost of living table which consists of 4 columns. The data frame consists of three data types which are string, float and integer. Only one column has string data type, one column has integer data type, and two columns are float data types.

```
richest_ranking=pd.read_csv("richest_countries.csv")
```

```
richest_ranking.dtypes
```

```
country           object
gdp_per_capita     int64
dtype: object
```

```
richest_ranking.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   country         50 non-null     object
 1   gdp_per_capita  50 non-null     int64 
dtypes: int64(1), object(1)
memory usage: 932.0+ bytes
```

Figure 1.3 Data Schema for Richest Ranking

Figure 1.3 shows the data schema for the richest ranking table which consists of only 2 columns. The data frame consists of two data types which are string and integer, one column has string data type, and another one column has integer data type.

```
tourism=pd.read_csv("tourism.csv")
```

```
tourism.dtypes
```

```
country                  object
tourists_in_millions    float64
receipts_in_billions    float64
receipts_per_tourist      int64
percentage_of_gdp       float64
dtype: object
```

```
tourism.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41 entries, 0 to 40
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   country               41 non-null     object 
 1   tourists_in_millions  41 non-null     float64
 2   receipts_in_billions  41 non-null     float64
 3   receipts_per_tourist  41 non-null     int64  
 4   percentage_of_gdp     41 non-null     float64
dtypes: float64(3), int64(1), object(1)
memory usage: 1.7+ KB
```

Figure 1.4 Data Schema for Tourism

Figure 1.4 shows the data schema for the tourism table which consists of 5 columns. The data frame consists of three data types which are string, float and integer. Only one column has string data type, one column has integer data type, and the rest are float data types.

```
unemployment=pd.read_csv("unemployment.csv")
```

```
unemployment.dtypes
```

```
country               object
unemployment_rate    float64
dtype: object
```

```
unemployment.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64 entries, 0 to 63
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   country            64 non-null     object
 1   unemployment_rate  64 non-null     float64
dtypes: float64(1), object(1)
memory usage: 1.1+ KB
```

Figure 1.5 Data Schema for Unemployment

Figure 1.5 shows the data schema for the unemployment table which consists of only 2 columns. The data frame consists of two data types which are string and float, one column has string data type, and another one column has float data type.

```
gdp_per_capita_penn_world=pd.read_csv("1- gdp-per-capita-penn-world-table.csv")
```

```
gdp_per_capita_penn_world.dtypes
```

```
Entity                                                object
Code                                                  object
Year                                                   int64
GDP per capita (output, multiple price benchmarks)   float64
dtype: object
```

```
gdp_per_capita_penn_world.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10399 entries, 0 to 10398
Data columns (total 4 columns):
 #   Column                                              Non-Null Count  Dtype
---  ------                                              --------------  -----
 0   Entity                                              10399 non-null  object
 1   Code                                                10399 non-null  object
 2   Year                                                10399 non-null  int64
 3   GDP per capita (output, multiple price benchmarks)  10399 non-null  float64
dtypes: float64(1), int64(1), object(2)
memory usage: 325.1+ KB
```

Figure 1.6 Data Schema for GDP per Capita Penn World

Figure 1.6 shows the data schema for the GDP per Capita Penn World table which consists of 4 columns. The data frame consists of three data types which are string, float and integer. Only one column has integer data type, one column has float data type, and the rest columns are string data types.

```
gdp_per_capita_maddison=pd.read_csv("2- gdp-per-capita-maddison.csv")

gdp_per_capita_maddison.dtypes

Entity              object
Code                object
Year                 int64
GDP per capita     float64
Unnamed: 4          object
dtype: object

gdp_per_capita_maddison.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19876 entries, 0 to 19875
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Entity          19876 non-null  object 
 1   Code            19651 non-null  object 
 2   Year            19876 non-null  int64  
 3   GDP per capita  19876 non-null  float64
 4   Unnamed: 4      21 non-null     object 
dtypes: float64(1), int64(1), object(3)
memory usage: 776.5+ KB
```

Figure 1.7 Data Schema for GDP per Capita Maddison

Figure 1.7 shows the data schema for the GDP per Capita Maddison table which consists of 5 columns. The data frame consists of three data types which are string, float and integer. Only one column has integer data type, one column has float data type, and the rest columns are string data types.

```
gdp_per_capita_world_bank=pd.read_csv("3- gdp-per-capita-worldbank.csv")

gdp_per_capita_world_bank.dtypes

Entity                                                 object
Code                                                   object
Year                                                    int64
GDP per capita, PPP (constant 2017 international $)   float64
dtype: object

gdp_per_capita_world_bank.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6346 entries, 0 to 6345
Data columns (total 4 columns):
 #   Column                                               Non-Null Count  Dtype  
---  ------                                               --------------  -----  
 0   Entity                                               6346 non-null   object 
 1   Code                                                 5902 non-null   object 
 2   Year                                                 6346 non-null   int64  
 3   GDP per capita, PPP (constant 2017 international $)  6346 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 198.4+ KB
```

Figure 1.8 Data Schema for GDP per Capita World Bank

Figure 1.8 shows the data schema for the GDP per Capita World Bank table which consists of 4 columns. The data frame consists of three data types which are string, float and integer. Only one column has integer data type, one column has float data type, and the rest columns are string data types.

```
national_gdp_wb=pd.read_csv("4- national-gdp-wb.csv")
```

```
national_gdp_wb.dtypes
```

```
Entity                                      object
Code                                        object
Year                                         int64
GDP, PPP (constant 2017 international $)     int64
dtype: object
```

```
national_gdp_wb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6346 entries, 0 to 6345
Data columns (total 4 columns):
 #   Column                                    Non-Null Count  Dtype 
---  ------                                    --------------  ----- 
 0   Entity                                    6346 non-null   object
 1   Code                                      5902 non-null   object
 2   Year                                      6346 non-null   int64 
 3   GDP, PPP (constant 2017 international $)  6346 non-null   int64 
dtypes: int64(2), object(2)
memory usage: 198.4+ KB
```

Figure 1.9 Data Schema for National GDP WB

Figure 1.9 shows the data schema for the National GDP WB table which consists of 4 columns. The data frame consists of two data types which are string and integer. Two columns have string data types, and two columns are integer data types.

```
national_gdp_penn_world=pd.read_csv("5- national-gdp-penn-world-table.csv")
```

```
national_gdp_penn_world.dtypes
```

```
Entity                                        object
Code                                          object
Year                                           int64
GDP (output, multiple price benchmarks)        int64
dtype: object
```

```
national_gdp_penn_world.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10399 entries, 0 to 10398
Data columns (total 4 columns):
 #   Column                                   Non-Null Count  Dtype
---  ------                                   --------------  -----
 0   Entity                                   10399 non-null  object
 1   Code                                     10399 non-null  object
 2   Year                                     10399 non-null  int64
 3   GDP (output, multiple price benchmarks)  10399 non-null  int64
dtypes: int64(2), object(2)
memory usage: 325.1+ KB
```

Figure 1.10 Data Schema for National GDP Penn World

Figure 1.10 shows the data schema for the National GDP Penn World table which consists of 4 columns. The data frame consists of two data types which are string and integer. Two columns have string data types, and two columns are integer data types

2.0 ARCHITECTURE

2.1 Pipeline Structure

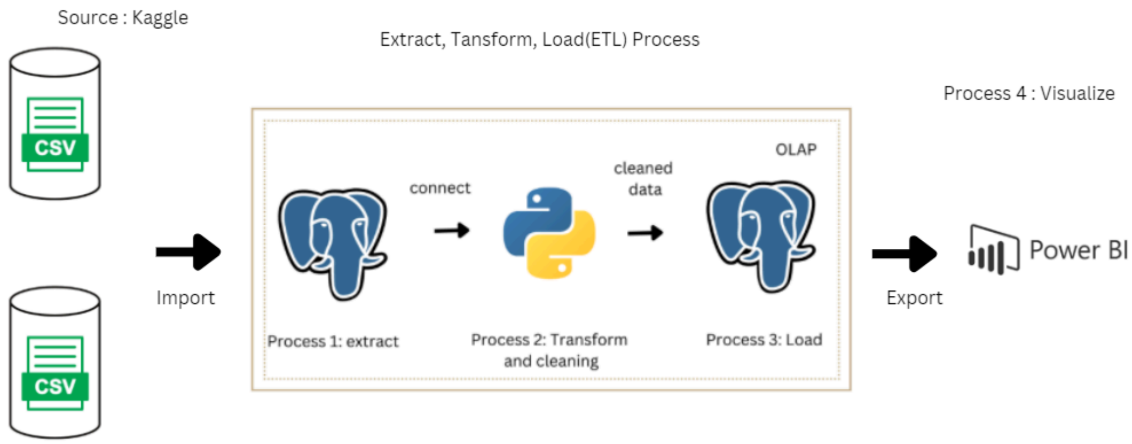


Figure 2.1 Architecture of This Project

Based on **Figure 2.1**, the list data set was acquired in Kaggle.com. Two datasets are used in this project, one of the datasets is World Economics Dataset which contains tables corruption, table tourism, table unemployment and table cost of living. Another dataset is about GDP values which contains table GDP per capita Penn World, table GDP per capita Maddison, table GDP per capita World Bank, table National GDP WB, and table National GDP Penn World. These tables were then imported into PostgreSQL. After successfully importing all the tables into PostgreSQL, the process continues as we connect to python in Jupyter Notebook for further operations. Some necessary libraries such as sqlalchemy, ipython-sql and numpy need to be installed first to extract data from PostgreSQL. After installing the libraries needed, the dataset was then cleaned and transformed where NULL values were identified and removed. After the cleaning and transformation process was done, the data integration process took place. Data integration process is where the tables are combined and provide a unified, single view of data. The OLAP operations like slicing, roll up and dicing were performed after the cleaned data load into PostgreSQL again in order to provide a better view and make it easier for analysis steps that are going to take place. Lastly, after OLAP operations are done, the results are then imported into powerBI for the visualisation part. As in power BI there are various analyses that can be performed in order to observe the outcomes needed in the project.

2.2 Process Flow

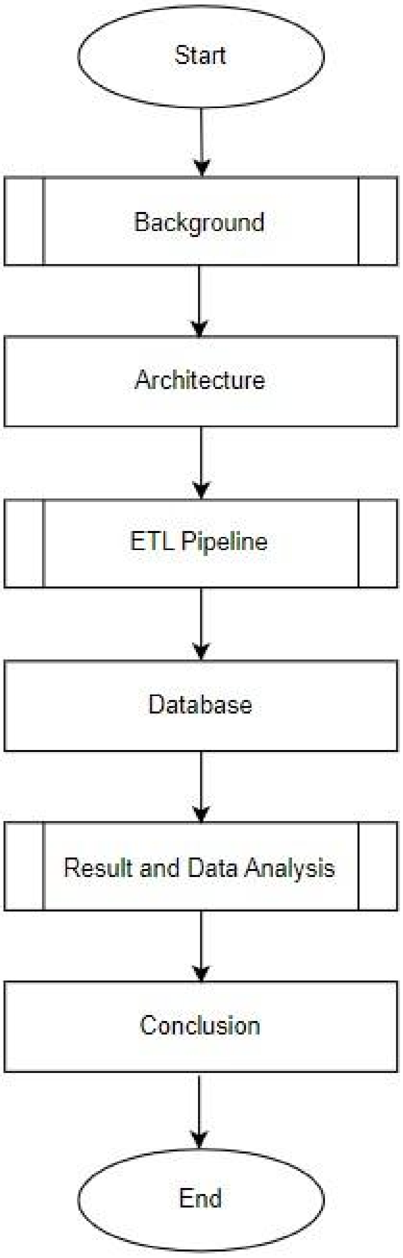


Figure 2.2 Flow of the project

Figure 2.2 illustrates the overall process of the project, which will be completed in six stages.

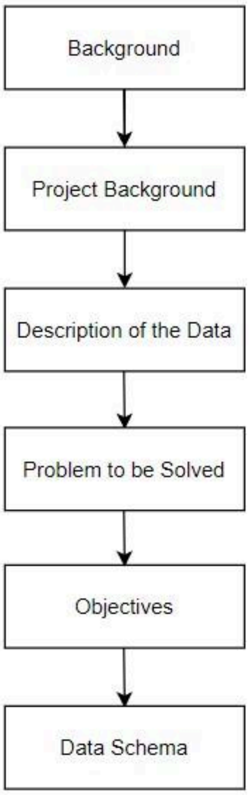


Figure 2.3 Process of the background

Figure 2.3 illustrates the background process, encompassing the project's background, the data description, the problem to be addressed, the project's objectives and the data schema. The information used in this study was sourced from Kaggle. Subsequently, the architecture was designed to ensure the project proceeds smoothly and according to plan.

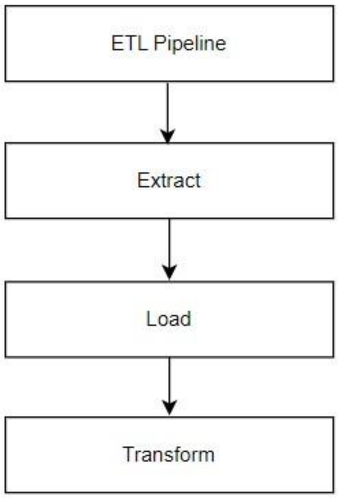


Figure 2.4 Process of ETL Pipeline

Figure 2.4 illustrates the Extract, Load, and Transform (ETL) pipeline process. The ETL process begins by extracting and transforming raw data from PgAdmin to Jupyter Notebook for data cleaning and merging. After extraction, the cleaned data is loaded into pgAdmin again. The final step is transforming the data within pgAdmin for the OLAP process.

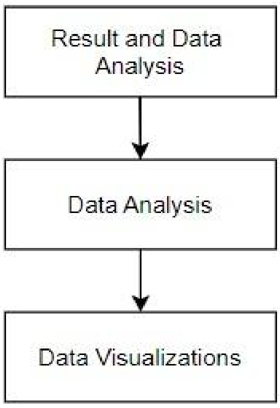


Figure 2.5 Process for the results and data analysis

Figure 2.5 presents the findings and data analysis, which includes step data visualisation. PgAdmin will be used to perform data analysis tasks such as roll-up and slicing. Following that, we will employ Microsoft Power BI to finalise the data visualisation. Lastly, conclusions will be drawn based on the results of the data analysis and visualisation.

3.0 ETL PIPELINE

3.1 ETL Pipeline

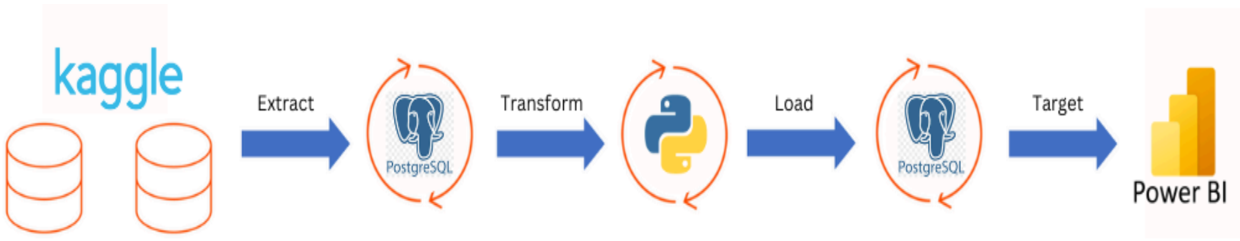


Figure 3.1 ETL Pipeline

Figure 3.1 illustrates the process of extracting data from Kaggle datasets (World Economic Data and World GDP), transforming it using Python, and loading it into a PostgreSQL database for final analysis and visualisation in Power BI. Initially, the datasets are extracted and loaded into PostgreSQL, where Python scripts are used to clean, merge, and preprocess the data. The transformed data is then reloaded into PostgreSQL, making it ready for Power BI to create visualisations and reports, thereby enabling insights and data-driven decision-making.

3.2 ETL PROCESS

3.2.1 Extract

Extracting data is the first step in the ETL process. The datasets needed to be saved in the PostgreSQL database before starting the ETL process. Thus, a new database named "employee" will be created in PostgreSQL as shown in **Figure 3.2** at the first. **Figure 3.3** shows all the tables created in the database. Then, the data will be copied from each csv file and retrieved to each table builded in PostgreSQL. Lastly, all the tables built will be shown in **Table 3.1**.

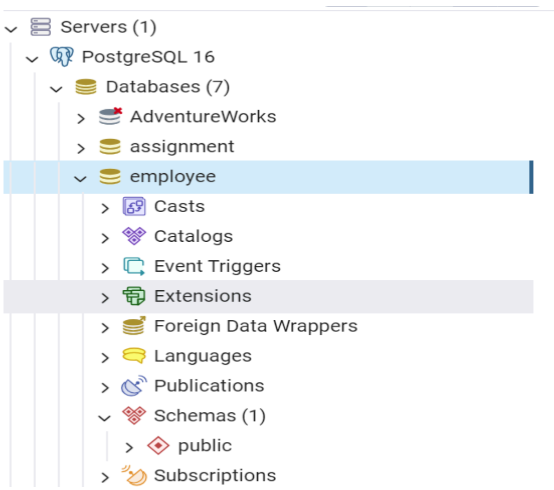


Figure 3.2 Database in PostgreSQL

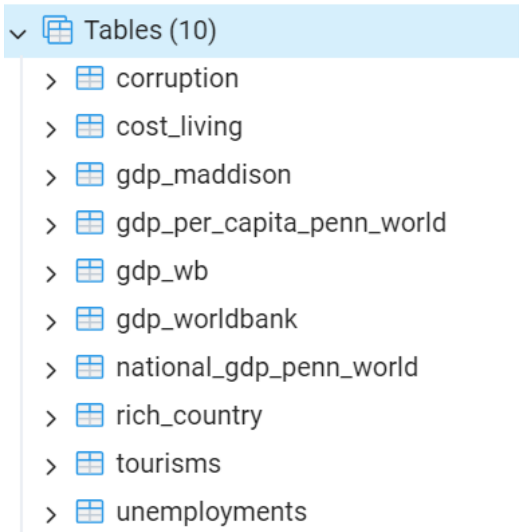


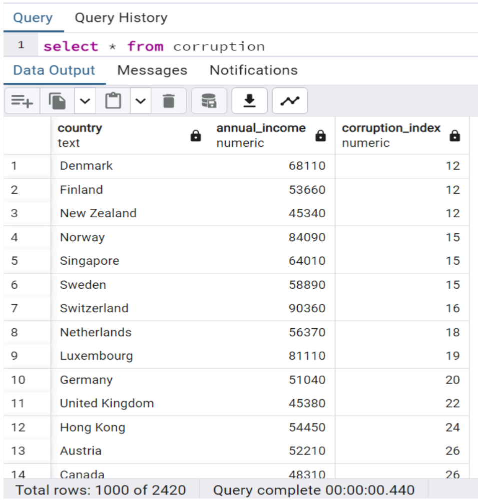
Figure 3..3 Table Created in Database

QUERY	OUTPUT

QUERY

```
CREATE TABLE corruption
(
    country text,
    annual_income numeric,
    corruption_index numeric
);

SELECT * FROM corruption
```

OUTPUT



Query | Query History

```
1 select * from corruption
```

Data Output | Messages | Notifications

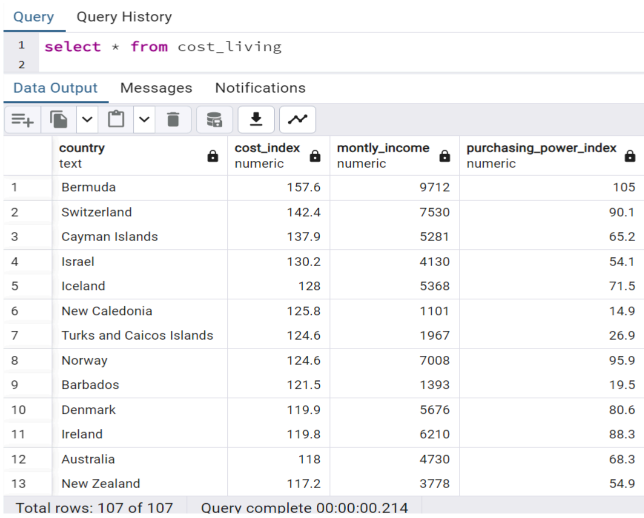
	country text	annual_income numeric	corruption_index numeric
1	Denmark	68110	12
2	Finland	53660	12
3	New Zealand	45340	12
4	Norway	84090	15
5	Singapore	64010	15
6	Sweden	58890	15
7	Switzerland	90360	16
8	Netherlands	56370	18
9	Luxembourg	81110	19
10	Germany	51040	20
11	United Kingdom	45380	22
12	Hong Kong	54450	24
13	Austria	52210	26
14	Canada	48310	26

Total rows: 1000 of 2420 Query complete 00:00:00.440

QUERY

```
CREATE TABLE cost_living
(
    country text,
    cost_index numeric,
    montly_income numeric,
        purchasing_power_index numeric
);

SELECT * FROM cost_living
```

OUTPUT



Query | Query History

```
1 select * from cost_living
2
```

Data Output | Messages | Notifications

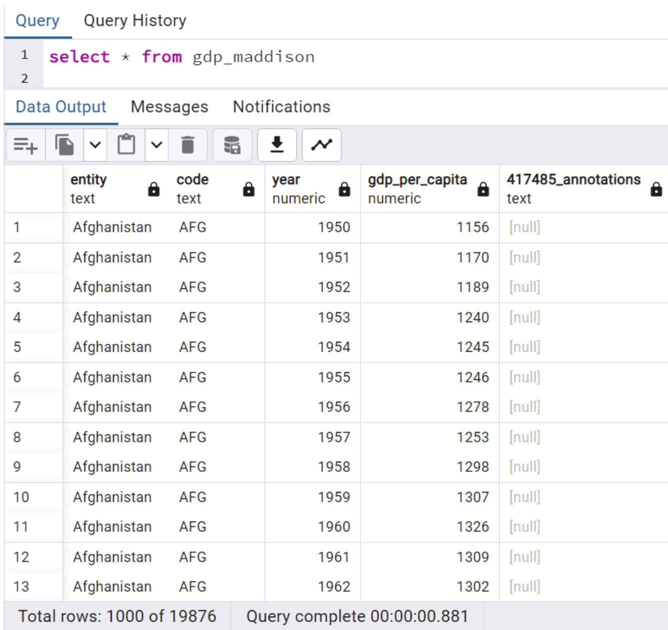
	country text	cost_index numeric	montly_income numeric	purchasing_power_index numeric
1	Bermuda	157.6	9712	105
2	Switzerland	142.4	7530	90.1
3	Cayman Islands	137.9	5281	65.2
4	Israel	130.2	4130	54.1
5	Iceland	128	5368	71.5
6	New Caledonia	125.8	1101	14.9
7	Turks and Caicos Islands	124.6	1967	26.9
8	Norway	124.6	7008	95.9
9	Barbados	121.5	1393	19.5
10	Denmark	119.9	5676	80.6
11	Ireland	119.8	6210	88.3
12	Australia	118	4730	68.3
13	New Zealand	117.2	3778	54.9

Total rows: 107 of 107 Query complete 00:00:00.214

QUERY

```
CREATE TABLE gdp_maddison
(
    entity text,
    code text,
    year numeric,
    gdp_per_capita numeric,
    “417485_annotations” text
);

SELECT * FROM gdp_maddison
```

OUTPUT



Query | Query History

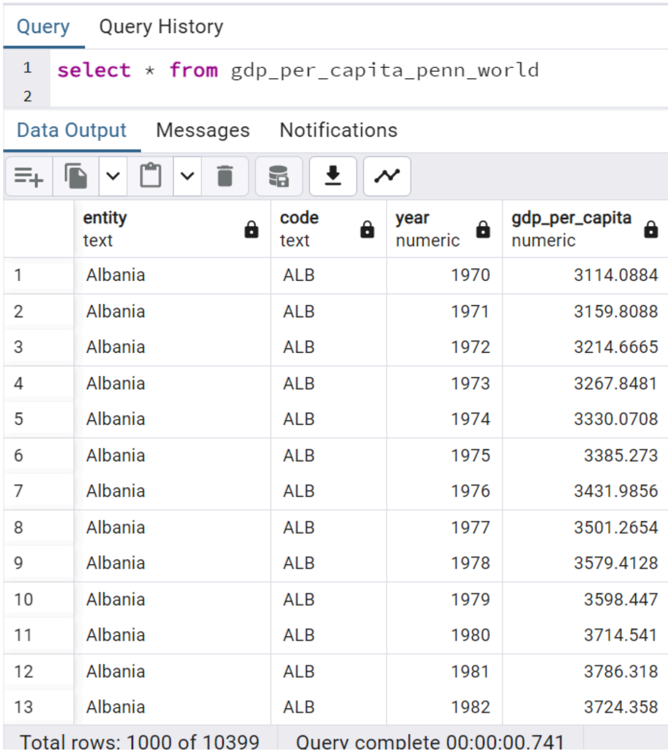
```
1 select * from gdp_maddison
2
```

Data Output | Messages | Notifications

	entity text	code text	year numeric	gdp_per_capita numeric	417485_annotations text
1	Afghanistan	AFG	1950	1156	[null]
2	Afghanistan	AFG	1951	1170	[null]
3	Afghanistan	AFG	1952	1189	[null]
4	Afghanistan	AFG	1953	1240	[null]
5	Afghanistan	AFG	1954	1245	[null]
6	Afghanistan	AFG	1955	1246	[null]
7	Afghanistan	AFG	1956	1278	[null]
8	Afghanistan	AFG	1957	1253	[null]
9	Afghanistan	AFG	1958	1298	[null]
10	Afghanistan	AFG	1959	1307	[null]
11	Afghanistan	AFG	1960	1326	[null]
12	Afghanistan	AFG	1961	1309	[null]
13	Afghanistan	AFG	1962	1302	[null]

Total rows: 1000 of 19876 Query complete 00:00:00.881

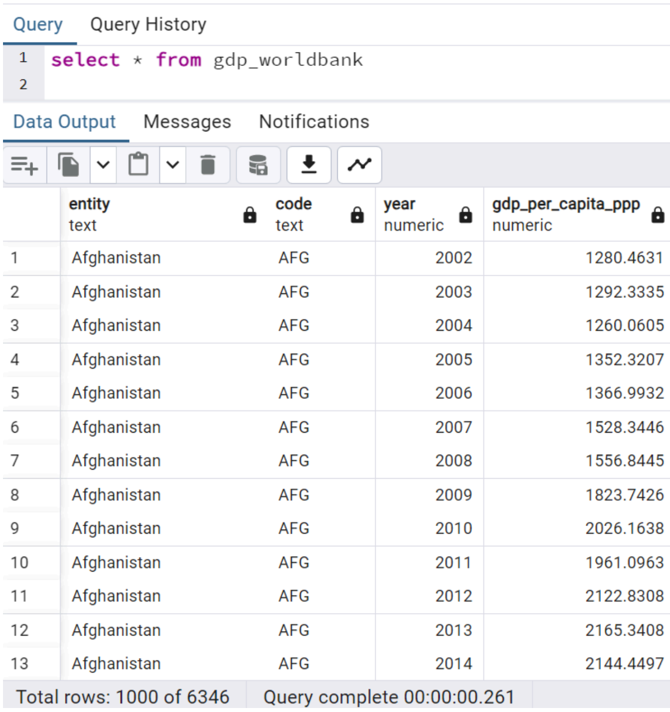
```
CREATE TABLE
gdp_per_capita_penn_world
(
entity text,
code text,
year numeric,
gdp_per_capita numeric
);

SELECT * FROM
gdp_per_capita_penn_world
```



	entity text	code text	year numeric	gdp_per_capita numeric
1	Albania	ALB	1970	3114.0884
2	Albania	ALB	1971	3159.8088
3	Albania	ALB	1972	3214.6665
4	Albania	ALB	1973	3267.8481
5	Albania	ALB	1974	3330.0708
6	Albania	ALB	1975	3385.273
7	Albania	ALB	1976	3431.9856
8	Albania	ALB	1977	3501.2654
9	Albania	ALB	1978	3579.4128
10	Albania	ALB	1979	3598.447
11	Albania	ALB	1980	3714.541
12	Albania	ALB	1981	3786.318
13	Albania	ALB	1982	3724.358

Total rows: 1000 of 10399 Query complete 00:00:00.741

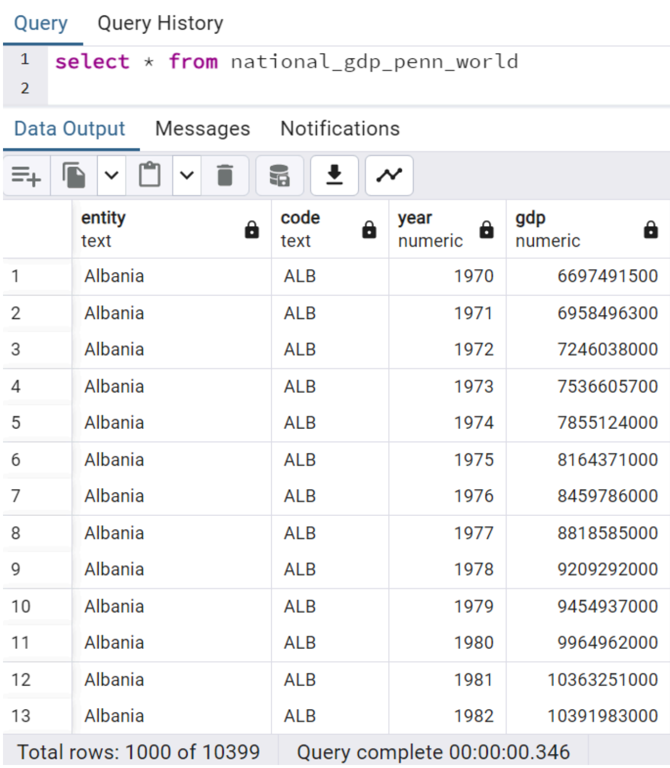
```
CREATE TABLE gdp_worldbank
(
    entity text,
    code text,
    year numeric,
    gdp_per_capita_ppp numeric
);

SELECT * FROM gdp_worldbank
```



	entity text	code text	year numeric	gdp_per_capita_ppp numeric
1	Afghanistan	AFG	2002	1280.4631
2	Afghanistan	AFG	2003	1292.3335
3	Afghanistan	AFG	2004	1260.0605
4	Afghanistan	AFG	2005	1352.3207
5	Afghanistan	AFG	2006	1366.9932
6	Afghanistan	AFG	2007	1528.3446
7	Afghanistan	AFG	2008	1556.8445
8	Afghanistan	AFG	2009	1823.7426
9	Afghanistan	AFG	2010	2026.1638
10	Afghanistan	AFG	2011	1961.0963
11	Afghanistan	AFG	2012	2122.8308
12	Afghanistan	AFG	2013	2165.3408
13	Afghanistan	AFG	2014	2144.4497

Total rows: 1000 of 6346 Query complete 00:00:00.261

```
CREATE TABLE
national_gdp_penn_world
(
    entity text,
    code text,
    year numeric,
    gdp numeric
);

SELECT * FROM
national_gdp_penn_world
```



Query   Query History

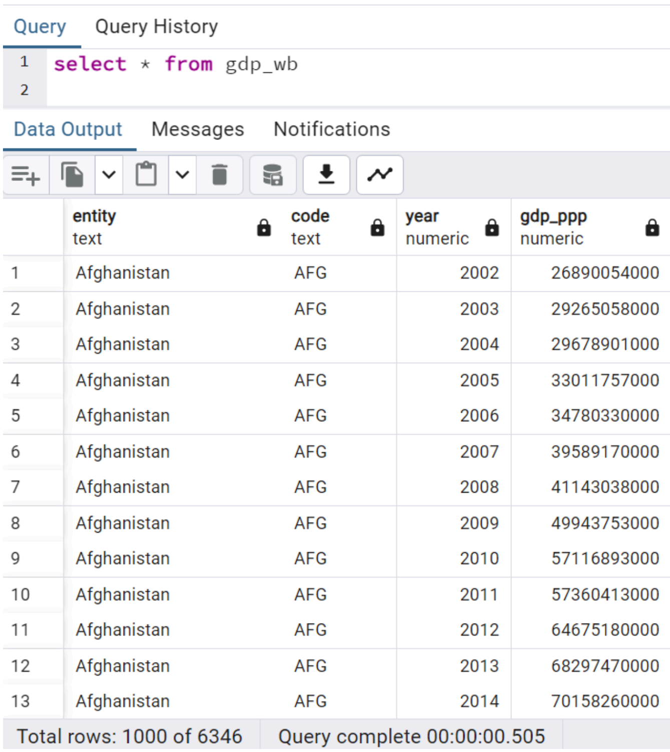
```
1 select * from national_gdp_penn_world
2
```

Data Output   Messages   Notifications

	entity text	code text	year numeric	gdp numeric
1	Albania	ALB	1970	6697491500
2	Albania	ALB	1971	6958496300
3	Albania	ALB	1972	7246038000
4	Albania	ALB	1973	7536605700
5	Albania	ALB	1974	7855124000
6	Albania	ALB	1975	8164371000
7	Albania	ALB	1976	8459786000
8	Albania	ALB	1977	8818585000
9	Albania	ALB	1978	9209292000
10	Albania	ALB	1979	9454937000
11	Albania	ALB	1980	9964962000
12	Albania	ALB	1981	10363251000
13	Albania	ALB	1982	10391983000

Total rows: 1000 of 10399   Query complete 00:00:00.346

```
CREATE TABLE gdp_wb
(
    entity text,
    code text,
    year numeric,
    gdp_ppp numeric
);

SELECT * FROM gdp_wb
```



Query   Query History

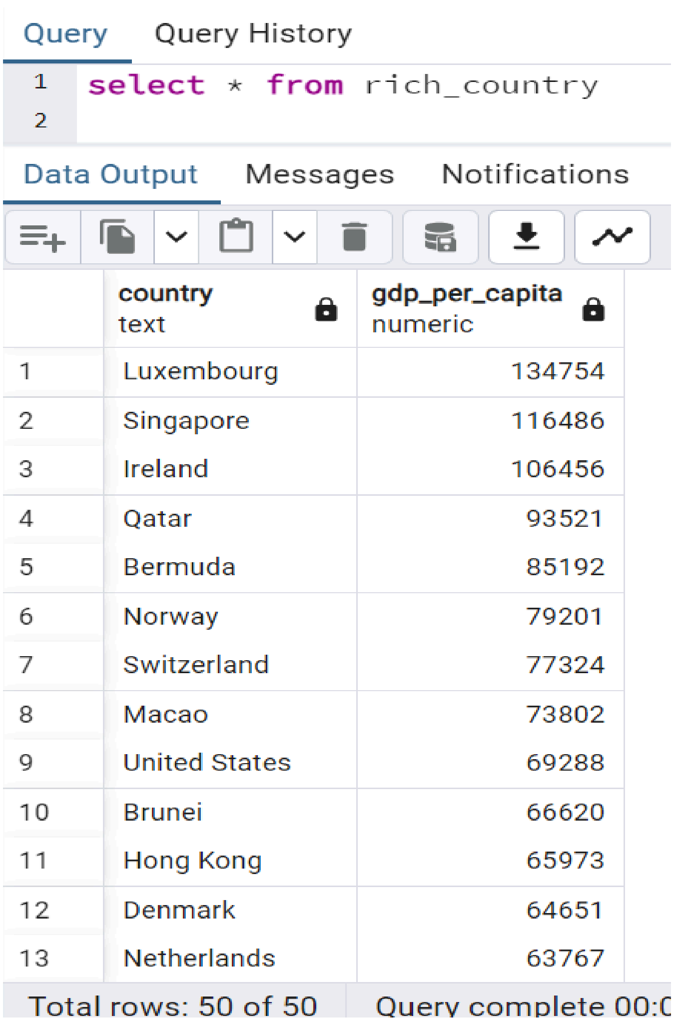
```
1 select * from gdp_wb
2
```

Data Output   Messages   Notifications

	entity text	code text	year numeric	gdp_ppp numeric
1	Afghanistan	AFG	2002	26890054000
2	Afghanistan	AFG	2003	29265058000
3	Afghanistan	AFG	2004	29678901000
4	Afghanistan	AFG	2005	33011757000
5	Afghanistan	AFG	2006	34780330000
6	Afghanistan	AFG	2007	39589170000
7	Afghanistan	AFG	2008	41143038000
8	Afghanistan	AFG	2009	49943753000
9	Afghanistan	AFG	2010	57116893000
10	Afghanistan	AFG	2011	57360413000
11	Afghanistan	AFG	2012	64675180000
12	Afghanistan	AFG	2013	68297470000
13	Afghanistan	AFG	2014	70158260000

Total rows: 1000 of 6346   Query complete 00:00:00.505

```
CREATE TABLE rich_country
(
    country text,
    gdp_per_capita numeric
);

SELECT * FROM rich_country
```



Query Query History

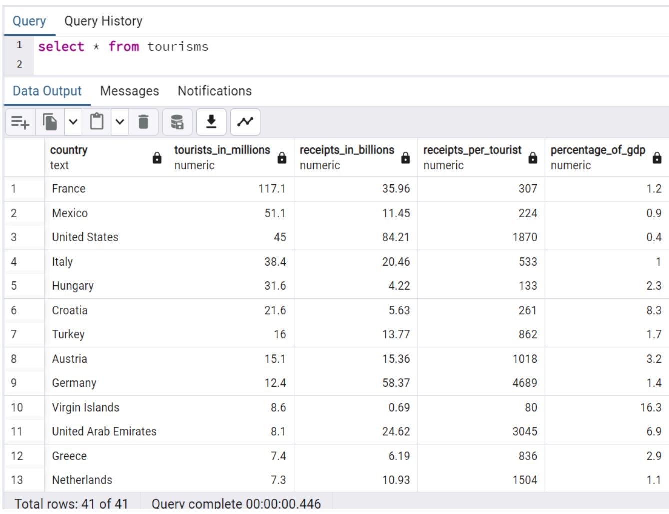
```
1 select * from rich_country
2
```

Data Output Messages Notifications

	country text	gdp_per_capita numeric
1	Luxembourg	134754
2	Singapore	116486
3	Ireland	106456
4	Qatar	93521
5	Bermuda	85192
6	Norway	79201
7	Switzerland	77324
8	Macao	73802
9	United States	69288
10	Brunei	66620
11	Hong Kong	65973
12	Denmark	64651
13	Netherlands	63767

Total rows: 50 of 50 Query complete 00:0

```
CREATE TABLE tourisms
(
    country text,
    tourists_in_millions  numeric,
    receipts_in_billions numeric,
    receipts_per_tourists numeric,
    percentage_of_gdp numeric
);

SELECT * FROM tourisms
```



Query Query History

```
1 select * from tourisms
2
```

Data Output Messages Notifications

	country text	tourists_in_millions numeric	receipts_in_billions numeric	receipts_per_tourist numeric	percentage_of_gdp numeric
1	France	117.1	35.96	307	1.2
2	Mexico	51.1	11.45	224	0.9
3	United States	45	84.21	1870	0.4
4	Italy	38.4	20.46	533	1
5	Hungary	31.6	4.22	133	2.3
6	Croatia	21.6	5.63	261	8.3
7	Turkey	16	13.77	862	1.7
8	Austria	15.1	15.36	1018	3.2
9	Germany	12.4	58.37	4689	1.4
10	Virgin Islands	8.6	0.69	80	16.3
11	United Arab Emirates	8.1	24.62	3045	6.9
12	Greece	7.4	6.19	836	2.9
13	Netherlands	7.3	10.93	1504	1.1

Total rows: 41 of 41 Query complete 00:00:00.446

```
CREATE TABLE unemployments
(
    country text,
    unemployment_rate numeric
);

SELECT * FROM unemployments
```

	country text	unemployment_rate numeric
1	Marshall Islands	36.0
2	South Africa	33.6
3	Kiribati	30.6
4	Kosovo	30.5
5	American Samoa	29.8
6	Djibouti	28.4
7	Eswatini	25.8
8	Greece	14.8
9	Spain	14.7
10	Brazil	14.4
11	Turkey	13.4
12	Italy	9.8
13	Nigeria	9.8

Table 3.1 Table Shows All Tables Created in Database

After the raw data has been extracted into PgAdmin, now we connect our PgAdmin with the Jupyter Notebook to proceed to the next step which transforms the data. Before starting the process, we are required to install a few packages as shown in **Figure 3.4.**

```
In [1]: ! pip install ipython-sql
        ! pip install sqlalchemy
        ! pip install psycopg2
        ! pip install python-sql
        ! pip install pandas-sql
        ! pip install sql-queries
```

Figure 3.4 Package installed

Then, the ipython-sql will be loaded and the engine function will be created to connect the PgAdmin to the Jupyter Notebook**. (Refer to Appendix)** Lastly, the data will be extracted from PgAdmin to Jupyter Notebook to continue the following steps.

3.2.2 Transform

After the connecting process, the data needs to be cleaned as the data cleaning process is a crucial part in data processing. Firstly, the data will be stored in DataFrame as shown in **Figure 3.5**. Then, the missing values will be checked and cleaned as shown in **Figure 3.6.** It's important for us to look for primary key duplication after checking for null data. After the dataset has been cleaned, we must make sure that its primary key remains unique in order to use the software to connect the dataset and create a relational model. Proceed with each data frame in turn. In addition to looking for null values, the cleaning procedure required us to examine and analyse the data so that we could remove or remove any extraneous columns or data from the dataset as shown in **Figure 3.7.** Lastly, repeat all the steps for all the tables transformed into Jupyter Notebook.. **(Refer to Appendix)**

```
In [53]: df = pd.read_sql_query(sql,engine)
         df.head()
```

Out[53]:

	entity	code	year	gdp_per_capita	417485_annotations
0	Afghanistan	AFG	1950.0	1156.0	None
1	Afghanistan	AFG	1951.0	1170.0	None
2	Afghanistan	AFG	1952.0	1189.0	None
3	Afghanistan	AFG	1953.0	1240.0	None
4	Afghanistan	AFG	1954.0	1245.0	None

Figure 3.5 Data Stored into DataFrame

```
In [54]: df.isnull().sum()

Out[54]: entity                   0
         code                   225
         year                     0
         gdp_per_capita           0
         417485_annotations   19855
         dtype: int64
```

Figure 3.6(a) Missing Values Handling

```
In [55]: df.duplicated(subset="entity").sum()

Out[55]: 19698

In [56]: new_gdp_maddison=df.dropna(axis=0,how="any")
         new_gdp_maddison.isna().sum()

Out[56]: entity                0
         code                  0
         year                  0
         gdp_per_capita        0
         417485_annotations    0
         dtype: int64
```

Figure 3.6(b) Missing Values Handling

```
In [57]: df.drop(columns=['417485_annotations'], inplace=True)
         print(df)

               entity code    year  gdp_per_capita
0         Afghanistan  AFG  1950.0       1156.0000
1         Afghanistan  AFG  1951.0       1170.0000
2         Afghanistan  AFG  1952.0       1189.0000
3         Afghanistan  AFG  1953.0       1240.0000
4         Afghanistan  AFG  1954.0       1245.0000
...               ...  ...     ...             ...
19871        Zimbabwe  ZWE  2014.0       1594.0000
19872        Zimbabwe  ZWE  2015.0       1560.0000
19873        Zimbabwe  ZWE  2016.0       1534.0000
19874        Zimbabwe  ZWE  2017.0       1582.3662
19875        Zimbabwe  ZWE  2018.0       1611.4052

[19876 rows x 4 columns]
```

Figure 3.7 Action to Drop Columns

3.2.3 Load

After cleaning the data, the data are then loaded into PostgreSQL again. After creating the database and table in pgAdmin, we can directly import the data from the Jupyter Notebook into PostgreSQL **(Refer to Appendix). Figure 3.8** shows that data successfully loaded into PostgreSQL and **Figure 3.9** shows the new database built for loading data. Lastly, **Table 3.2** shows the loading of data from Jupyter to PGAdmin.

```
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
the dataframe succesfully inserted
```

Figure 3.8 Result which Shown Successfully Loading

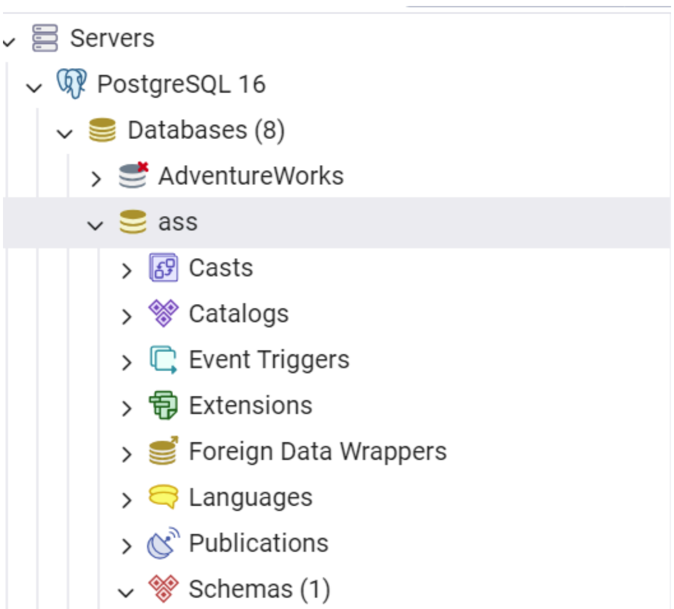
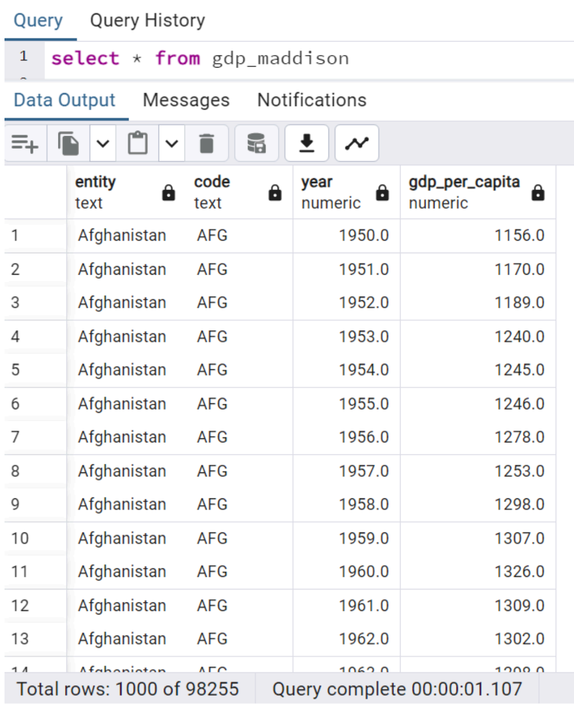
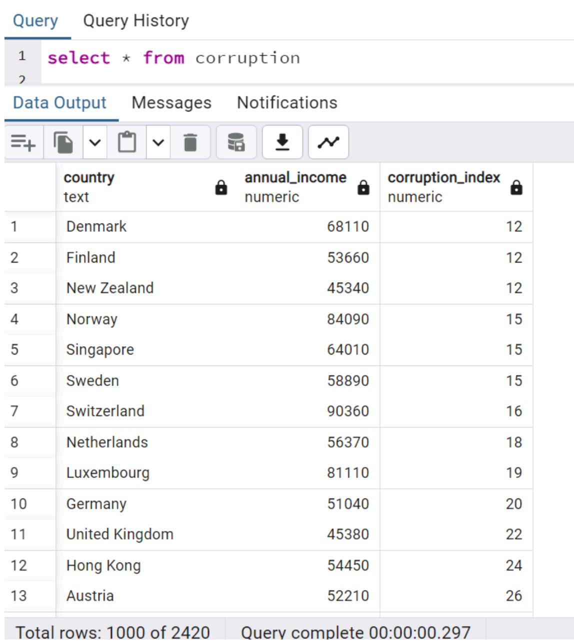


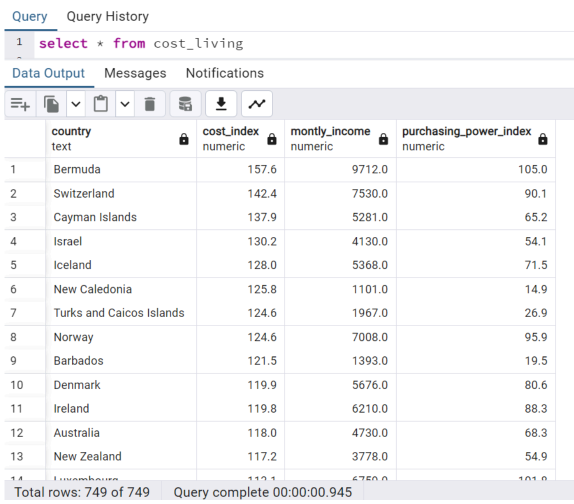
Figure 3.9 Database Built for Loading Data

<table>
<tr><th>QUERY</th><th>OUTPUT</th></tr>
<tr><td>CREATE TABLE gdp_maddison
(
entity text,
code text,
year numeric,
gdp_per_capita numeric
);

SELECT * FROM gdp_maddison</td><td>
Query Query History
1 select * from gdp_maddison
Data Output Messages Notifications
<table>
<tr><th></th><th>entity text</th><th>code text</th><th>year numeric</th><th>gdp_per_capita numeric</th></tr>
<tr><td>1</td><td>Afghanistan</td><td>AFG</td><td>1950.0</td><td>1156.0</td></tr>
<tr><td>2</td><td>Afghanistan</td><td>AFG</td><td>1951.0</td><td>1170.0</td></tr>
<tr><td>3</td><td>Afghanistan</td><td>AFG</td><td>1952.0</td><td>1189.0</td></tr>
<tr><td>4</td><td>Afghanistan</td><td>AFG</td><td>1953.0</td><td>1240.0</td></tr>
<tr><td>5</td><td>Afghanistan</td><td>AFG</td><td>1954.0</td><td>1245.0</td></tr>
<tr><td>6</td><td>Afghanistan</td><td>AFG</td><td>1955.0</td><td>1246.0</td></tr>
<tr><td>7</td><td>Afghanistan</td><td>AFG</td><td>1956.0</td><td>1278.0</td></tr>
<tr><td>8</td><td>Afghanistan</td><td>AFG</td><td>1957.0</td><td>1253.0</td></tr>
<tr><td>9</td><td>Afghanistan</td><td>AFG</td><td>1958.0</td><td>1298.0</td></tr>
<tr><td>10</td><td>Afghanistan</td><td>AFG</td><td>1959.0</td><td>1307.0</td></tr>
<tr><td>11</td><td>Afghanistan</td><td>AFG</td><td>1960.0</td><td>1326.0</td></tr>
<tr><td>12</td><td>Afghanistan</td><td>AFG</td><td>1961.0</td><td>1309.0</td></tr>
<tr><td>13</td><td>Afghanistan</td><td>AFG</td><td>1962.0</td><td>1302.0</td></tr>
</table>Total rows: 1000 of 98255 Query complete 00:00:01.107</td></tr>
<tr><td>CREATE TABLE corruption
(
country text,
annual_income numeric,
corruption_index numeric
);

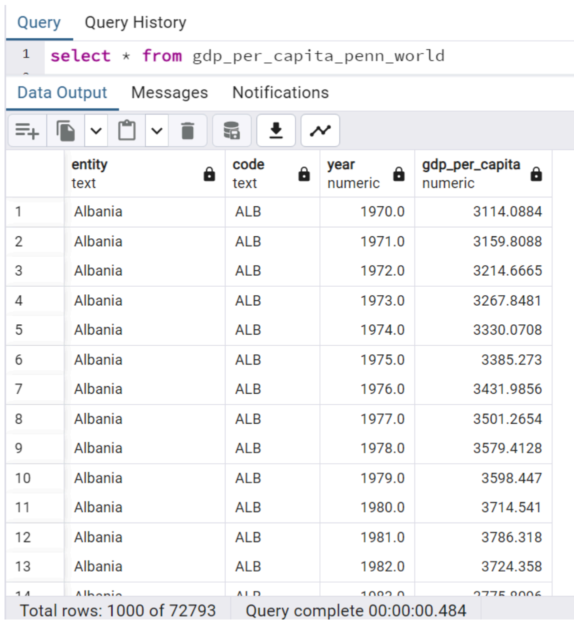
SELECT * FROM corruption</td><td>
Query Query History
1 select * from corruption
Data Output Messages Notifications
<table>
<tr><th></th><th>country text</th><th>annual_income numeric</th><th>corruption_index numeric</th></tr>
<tr><td>1</td><td>Denmark</td><td>68110</td><td>12</td></tr>
<tr><td>2</td><td>Finland</td><td>53660</td><td>12</td></tr>
<tr><td>3</td><td>New Zealand</td><td>45340</td><td>12</td></tr>
<tr><td>4</td><td>Norway</td><td>84090</td><td>15</td></tr>
<tr><td>5</td><td>Singapore</td><td>64010</td><td>15</td></tr>
<tr><td>6</td><td>Sweden</td><td>58890</td><td>15</td></tr>
<tr><td>7</td><td>Switzerland</td><td>90360</td><td>16</td></tr>
<tr><td>8</td><td>Netherlands</td><td>56370</td><td>18</td></tr>
<tr><td>9</td><td>Luxembourg</td><td>81110</td><td>19</td></tr>
<tr><td>10</td><td>Germany</td><td>51040</td><td>20</td></tr>
<tr><td>11</td><td>United Kingdom</td><td>45380</td><td>22</td></tr>
<tr><td>12</td><td>Hong Kong</td><td>54450</td><td>24</td></tr>
<tr><td>13</td><td>Austria</td><td>52210</td><td>26</td></tr>
</table>Total rows: 1000 of 2420 Query complete 00:00:00.297</td></tr>
</table>

```
CREATE TABLE cost_living
(
    country text,
    cost_index numeric,
    montly_income numeric,
    purchasing_power_index numeric
);

SELECT * FROM cost_living
```



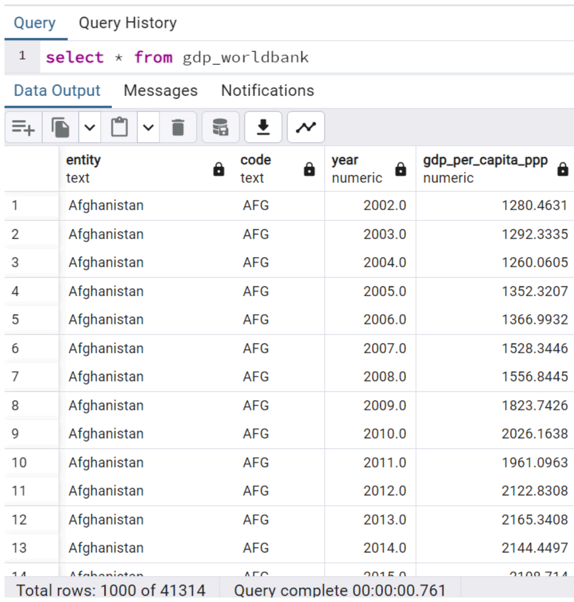
	country text	cost_index numeric	montly_income numeric	purchasing_power_index numeric
1	Bermuda	157.6	9712.0	105.0
2	Switzerland	142.4	7530.0	90.1
3	Cayman Islands	137.9	5281.0	65.2
4	Israel	130.2	4130.0	54.1
5	Iceland	128.0	5368.0	71.5
6	New Caledonia	125.8	1101.0	14.9
7	Turks and Caicos Islands	124.6	1967.0	26.9
8	Norway	124.6	7008.0	95.9
9	Barbados	121.5	1393.0	19.5
10	Denmark	119.9	5676.0	80.6
11	Ireland	119.8	6210.0	88.3
12	Australia	118.0	4730.0	68.3
13	New Zealand	117.2	3778.0	54.9

```
CREATE TABLE
gdp_per_capita_penn_world
(
entity text,
code text,
year numeric,
gdp_per_capita numeric
);

SELECT * FROM
gdp_per_capita_penn_world
```



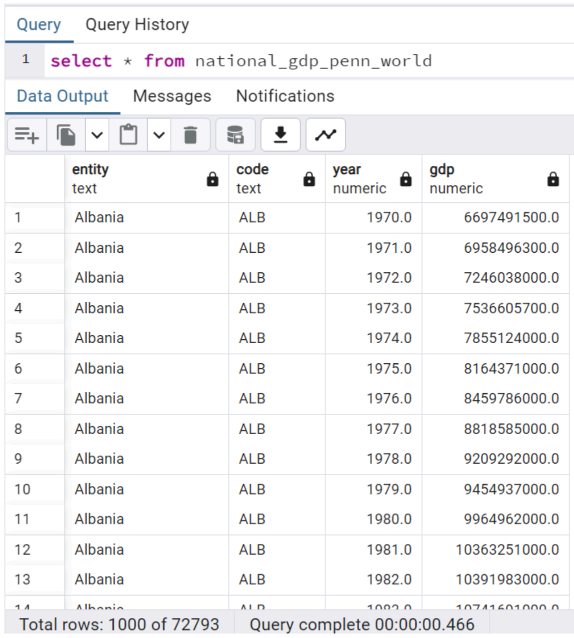
	entity text	code text	year numeric	gdp_per_capita numeric
1	Albania	ALB	1970.0	3114.0884
2	Albania	ALB	1971.0	3159.8088
3	Albania	ALB	1972.0	3214.6665
4	Albania	ALB	1973.0	3267.8481
5	Albania	ALB	1974.0	3330.0708
6	Albania	ALB	1975.0	3385.273
7	Albania	ALB	1976.0	3431.9856
8	Albania	ALB	1977.0	3501.2654
9	Albania	ALB	1978.0	3579.4128
10	Albania	ALB	1979.0	3598.447
11	Albania	ALB	1980.0	3714.541
12	Albania	ALB	1981.0	3786.318
13	Albania	ALB	1982.0	3724.358

```
CREATE TABLE gdp_worldbank
(
    entity text,
    code text,
    year numeric,
    gdp_per_capita_ppp numeric
);

SELECT * FROM gdp_worldbank
```



	entity text	code text	year numeric	gdp_per_capita_ppp numeric
1	Afghanistan	AFG	2002.0	1280.4631
2	Afghanistan	AFG	2003.0	1292.3335
3	Afghanistan	AFG	2004.0	1260.0605
4	Afghanistan	AFG	2005.0	1352.3207
5	Afghanistan	AFG	2006.0	1366.9932
6	Afghanistan	AFG	2007.0	1528.3446
7	Afghanistan	AFG	2008.0	1556.8445
8	Afghanistan	AFG	2009.0	1823.7426
9	Afghanistan	AFG	2010.0	2026.1638
10	Afghanistan	AFG	2011.0	1961.0963
11	Afghanistan	AFG	2012.0	2122.8308
12	Afghanistan	AFG	2013.0	2165.3408
13	Afghanistan	AFG	2014.0	2144.4497

```
CREATE TABLE
national_gdp_penn_world
(
    entity text,
    code text,
    year numeric,
    gdp numeric
);

SELECT * FROM
national_gdp_penn_world
```



Query | Query History

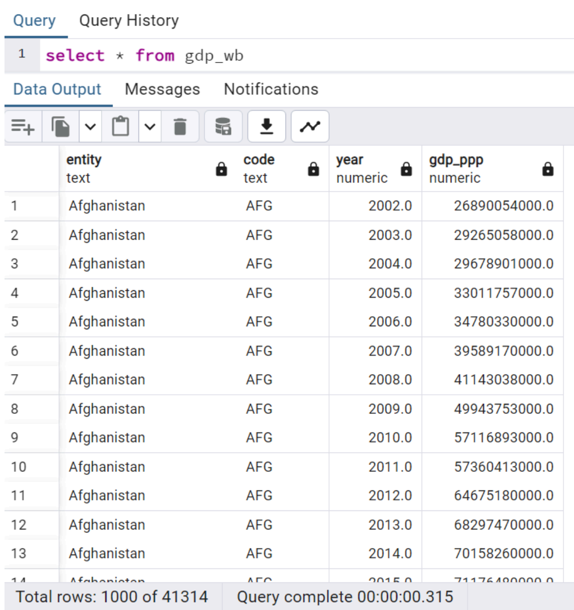
```
1 select * from national_gdp_penn_world
```

Data Output | Messages | Notifications

	entity text	code text	year numeric	gdp numeric
1	Albania	ALB	1970.0	6697491500.0
2	Albania	ALB	1971.0	6958496300.0
3	Albania	ALB	1972.0	7246038000.0
4	Albania	ALB	1973.0	7536605700.0
5	Albania	ALB	1974.0	7855124000.0
6	Albania	ALB	1975.0	8164371000.0
7	Albania	ALB	1976.0	8459786000.0
8	Albania	ALB	1977.0	8818585000.0
9	Albania	ALB	1978.0	9209292000.0
10	Albania	ALB	1979.0	9454937000.0
11	Albania	ALB	1980.0	9964962000.0
12	Albania	ALB	1981.0	10363251000.0
13	Albania	ALB	1982.0	10391983000.0

Total rows: 1000 of 72793 Query complete 00:00:00.466

```
CREATE TABLE gdp_wb
(
    entity text,
    code text,
    year numeric,
    gdp_ppp numeric
);

SELECT * FROM gdp_wb
```



Query | Query History

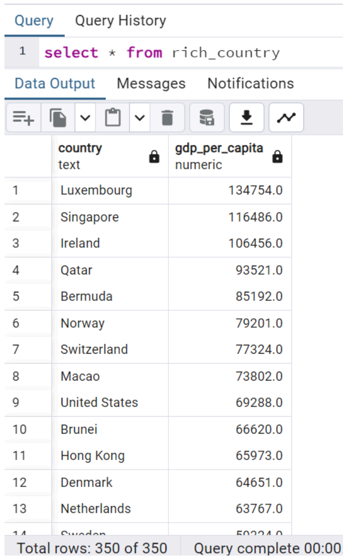
```
1 select * from gdp_wb
```

Data Output | Messages | Notifications

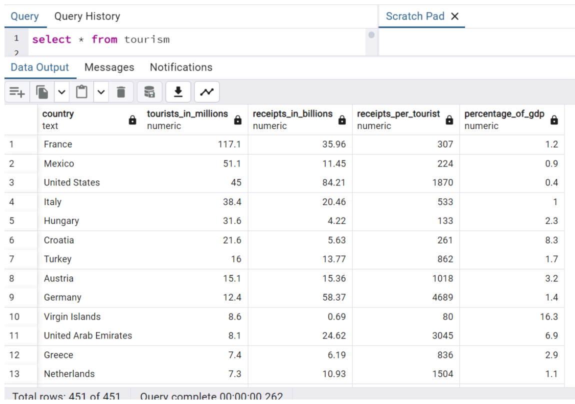
	entity text	code text	year numeric	gdp_ppp numeric
1	Afghanistan	AFG	2002.0	26890054000.0
2	Afghanistan	AFG	2003.0	29265058000.0
3	Afghanistan	AFG	2004.0	29678901000.0
4	Afghanistan	AFG	2005.0	33011757000.0
5	Afghanistan	AFG	2006.0	34780330000.0
6	Afghanistan	AFG	2007.0	39589170000.0
7	Afghanistan	AFG	2008.0	41143038000.0
8	Afghanistan	AFG	2009.0	49943753000.0
9	Afghanistan	AFG	2010.0	57116893000.0
10	Afghanistan	AFG	2011.0	57360413000.0
11	Afghanistan	AFG	2012.0	64675180000.0
12	Afghanistan	AFG	2013.0	68297470000.0
13	Afghanistan	AFG	2014.0	70158260000.0

Total rows: 1000 of 41314 Query complete 00:00:00.315

```
CREATE TABLE rich_country
(
    country text,
    gdp_per_capita numeric
);

SELECT * FROM rich_country
```




```
CREATE TABLE tourisms
(   country text,
    tourists_in_millions  numeric,
    receipts_in_billions numeric,
    receipts_per_tourists numeric,
    percentage_of_gdp numeric
);
SELECT * FROM tourisms
```




```
CREATE TABLE unemployments
(
    country text,
    unemployment_rate numeric
);

SELECT * FROM unemployments
```

Table 3.2 Table Shows All Loading Data in SQL

4.0 DATABASE

4.1 Relational Model

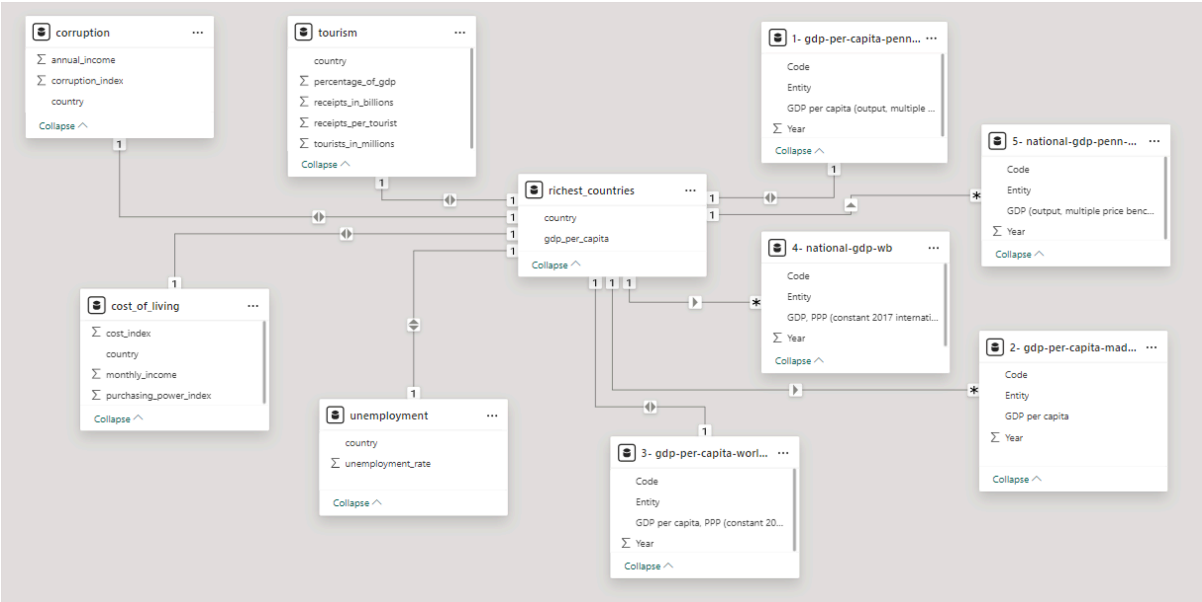


Figure 4.1 Model View in Power BI

Figure 4.1 shows the relational model built by using Power BI. It shows the relationship between the data used in this project.

4.2 Relationship between Data

Table 4.1 shows the relationship between each data in the dataset used.

Data	Relationship
richest ranking -> corruption	one to one
richest ranking -> cost of living	one to one
richest ranking -> tourism	one to one
richest ranking -> unemployment	one to one
richest ranking -> GDP per Capita Penn World	one to one
richest ranking -> GDP per Capita Maddison	one to one

richest ranking -> GDP per capita World Bank	one to one
richest ranking -> National GDP WB	one to one
richest ranking -> National GDP Penn World	one to one

Table 4.1 Relationship between Data.

4.3 Identification of Data Warehouse Schema

As shown in **Figure 4.1** above, the data warehouse schema used in this project is Star Schema. This is because it contains only one fact table, and has a total of nine dimensions tables connected to the fact table. The fact table is the richest ranking table, whereas the dimensional tables are corruption, cost of living, tourism, unemployment, GDP per Capita Penn World, GDP per Capita Maddison, GDP per Capita World Bank, National GDP WB, and National GDP Penn World tables.

5.0 RESULT AND DATA ANALYSIS

5.1 OLAP

5.1.1 Top 5 Richest Country in the World

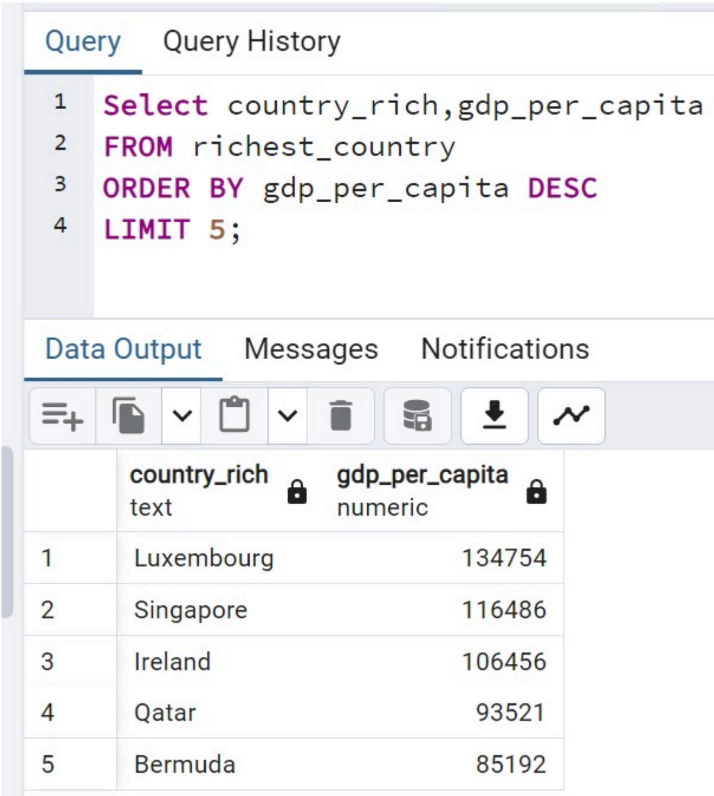


Figure 5.1 Drill-down Operation to Identify the Richest Country in the World

The OLAP visualisation above shows the drill-down operation conducted to identify the richest country in the world. We can see the top 5 richest countries in the world. Luxembourg ranked as number one with the highest number of gdp per capita which is 134,754 followed by Singapore,Ireland,Qatar and Bermuda. By analysing these richest countries, we can gain valuable insights into economic prosperity and tailor strategies accordingly. For the investors, this data will lead them on where they should allocate the resources for the optimal returns. Furthermore, the policymakers can leverage this knowledge to design policies that foster economic growth and stability. Lastly, this information helps the businesses to target their markets more precisely.In conclusion, understanding these rankings is crucial as it provides insights into economic prosperity and potential investment opportunities.

5.1.2 Relationship Between GDP per Capita and Tourism Percentage by Country

Query Query History

```sql
SELECT gdp_per_capita_maddison.gdp_per_capita, tourism.country, tourism.percentage_of_gdp
FROM gdp_per_capita_maddison, tourism
ORDER BY gdp_per_capita_maddison.gdp_per_capita DESC
LIMIT 10;
```

Data Output Messages Notifications

	gdp_per_capita numeric	country text	percentage_of_gdp numeric
1	156299.0	France	1.2
2	156299.0	Mexico	0.9
3	156299.0	United States	0.4
4	156299.0	Italy	1
5	156299.0	Hungary	2.3
6	156299.0	Croatia	8.3
7	156299.0	Turkey	1.7
8	156299.0	Austria	3.2
9	156299.0	Germany	1.4
10	156299.0	Virgin Islands	16.3

Total rows: 10 of 10 Query complete 00:00:00.403 Ln 4

Figure 5.2 : Drill-down Operation to Identify Relationship Between GDP per capita and Tourism

Based on the OLAP operation above, it shows the relationship between GDP per capita and the percentage of GDP per capita of tourism by country. We can see that the Virgin Islands has the highest percentage of GDP per capita in tourism, which is 16.3%. Contributions to GDP per capita were recorded as the highest, which is 156,299. It shows that the Virgin Islands, with stronger economies, may have more resources to invest in tourism infrastructure, marketing, and development, thus attracting more tourists and generating a higher proportion of GDP from tourism activities. On the other hand, the United States is recorded to have the lowest contribution to GDP per capita in tourism, which is only 0.4%, whereas GDP per capita remains higher in the Virgin Islands, which is 156,299. This may have fewer resources available for tourism development and may rely less on tourism as a significant economic contributor. Lastly, we can observe that the country having the same highest GDP value but different in percentage of tourism in GDP, hence we may conclude that tourism is not a key factor which affects the richest level of a country.

5.1.3 Factors Influencing the Rich Level of a Country

Query   Query History

```sql
SELECT rc.country AS country_rich,rc.gdp_per_capita,c.corruption_index,u.unemployment_rate
FROM richest_countries rc
JOIN Corruption c ON rc.country = c.country
JOIN Unemployment u ON rc.country = u.country
ORDER BY rc.gdp_per_capita DESC
LIMIT 10;
```

Data Output   Messages   Notifications

	country_rich text	gdp_per_capita numeric	corruption_index numeric	unemployment_rate numeric
1	Luxembourg	134754.0	19.0	5.2
2	Luxembourg	134754.0	19.0	5.2
3	Luxembourg	134754.0	19.0	5.2
4	Luxembourg	134754.0	19.0	5.2
5	Luxembourg	134754.0	19.0	5.2
6	Luxembourg	134754.0	19.0	5.2
7	Luxembourg	134754.0	19.0	5.2
8	Luxembourg	134754.0	19.0	5.2
9	Luxembourg	134754.0	19.0	5.2
10	Luxembourg	134754.0	19.0	5.2

Total rows: 10 of 10   Query complete 00:00:00.215

Figure 5.3 Roll-up Operation to Identify Factors Influencing the Rich Level of a Country

Based on the OLAP above, it actually shows a focus on the richest country according to the richest country ranking based on data 2022, Luxembourg. Luxembourg is a country which has the highest value in GDP per capita, 134,754 dollars. This may be because Luxembourg is known around the world as a business-friendly country, with low corporate taxes, a stable workforce, and government incentives with respect to investment. Also, with a small population, it will result in an unusually high GDP per capita. Next, the corruption perception index value of Luxembourg is 19.0. A nation with a low score on the Corruption Perception Index is seen to have a high amount of corruption. This may be the result of several things, including bribery, the misuse of governmental authority for personal benefit, the incompetence of public servants, etc. It is important to remember that corruption may seriously harm a nation's social welfare, political stability, and economics. As a result, initiatives to lessen corruption are crucial for prosperity and sustainable development. Lastly, the unemployment rate of Luxembourg is 5.2. As the world unemployment rate for 2022 was 5.27%, hence the unemployment rate of 5.2 is around the average value. When a nation's working-age population is highly employed, it means that a sizable share of it is employed. This might also indicate that the nation has a robust economy, successful job-creation strategies, and a highly educated and proficient workforce.

5.1.4 Top 10 Tourism Value and GDP per capita of Country

Query | Query History

```sql
Select country, country_rich, tourists_in_millions, gdp_per_capita
From tourism, richest_country
Where country=country_rich
Order By gdp_per_capita DESC
LIMIT 10;
```

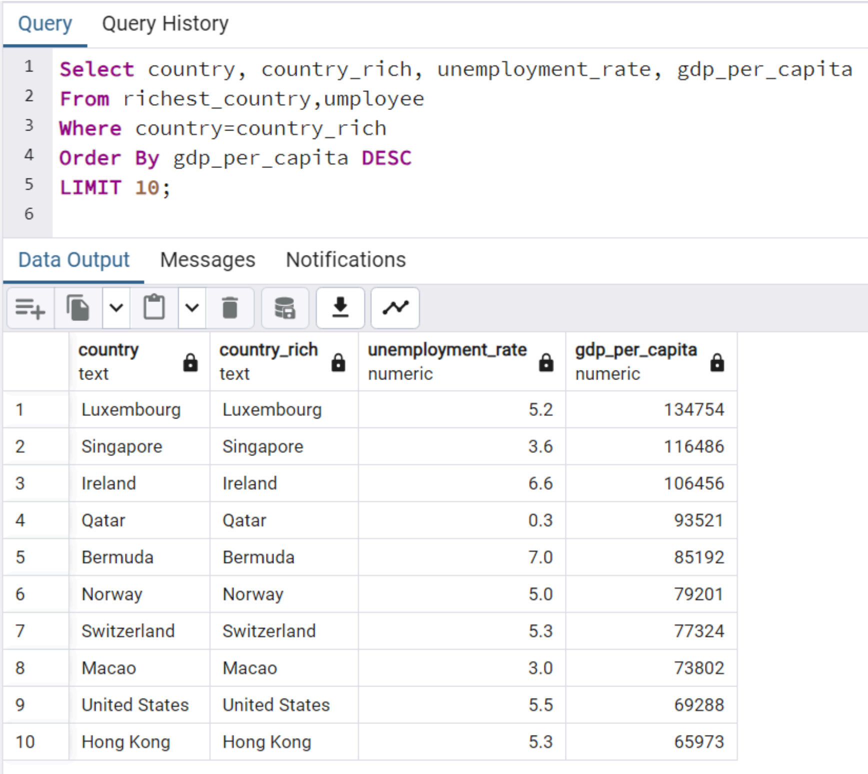
Data Output | Messages | Notifications

	country text	country_rich text	tourists_in_millions numeric	gdp_per_capita numeric
1	Luxembourg	Luxembourg	0.5	134754
2	Qatar	Qatar	0.6	93521
3	Bermuda	Bermuda	0.1	85192
4	Norway	Norway	1.4	79201
5	Macao	Macao	5.9	73802
6	United States	United States	45.0	69288
7	Hong Kong	Hong Kong	3.6	65973
8	Netherlands	Netherlands	7.3	63767
9	Belgium	Belgium	2.6	58931
10	Austria	Austria	15.1	58427

Figure 5.4 Roll-up Operation to Identify Tourism Value per Capita of Country

Based on the OLAP above, we can observe that the top 10 countries ranked by their tourism value are GDP per capita.. Luxembourg ranked as the first country in this analysis, securing the first position with a highest GDP per capita of 134,754 dollars and relatively low total number of tourists in 0.5 millions. On the other hand, Austria occupies the 10th position in this ranking, displaying the lowest GDP per capita at 58,427 dollars and a slightly higher total number of tourists at 15.1 millions. The highest number of tourism is recorded by the United States with values of 45 millions. This is because the U.S. government's initiatives to market the country as a top travel destination and to support an industry that boosts economic growth, excellent jobs, sustainability, and conservation may be the cause of this. In short, this analysis underscores the varying economic landscapes and the performance in tourism across different countries.

5.1.5 Top 10 Unemployee rate and GDP per capita of Country



Query | Query History

```sql
Select country, country_rich, unemployment_rate, gdp_per_capita
From richest_country,umployee
Where country=country_rich
Order By gdp_per_capita DESC
LIMIT 10;
```

Data Output | Messages | Notifications

	country text	country_rich text	unemployment_rate numeric	gdp_per_capita numeric
1	Luxembourg	Luxembourg	5.2	134754
2	Singapore	Singapore	3.6	116486
3	Ireland	Ireland	6.6	106456
4	Qatar	Qatar	0.3	93521
5	Bermuda	Bermuda	7.0	85192
6	Norway	Norway	5.0	79201
7	Switzerland	Switzerland	5.3	77324
8	Macao	Macao	3.0	73802
9	United States	United States	5.5	69288
10	Hong Kong	Hong Kong	5.3	65973

Figure 5.5 Drill-down Operation to Identify Unemployee Rate and GDP per capita

From the OLAP visualisation above, we can observe the top 10 countries ranked by their unemployment rates and GDP per capita. Luxembourg ranked as the first country in this analysis, securing the first position with a highest GDP per capita of 134,754 and a relatively low unemployment rate of 5.2%. This insight highlights the strong economic performance of Luxembourg, indicating a powerful economy with high income levels and a relatively stable job market. On the other hand, Hong Kong occupies the 10th position in this ranking, displaying the lowest GDP per capita at 65,973 and a slightly higher unemployment rate of 5.3%. This contrast with Luxembourg's position underscores the diversity in economic conditions among different countries. Despite its position at the bottom of the list, Hong Kong still maintains a relatively low unemployment rate, indicating a certain level of economic stability despite its lower GDP per capita compared to the top-ranking countries.Lastly, Bermuda recorded as the highest unemployment rate at 7%, indicating potential challenges in its labour market.This could be due to various factors such as economic fluctuations, structural issues within the labour market or specific policy measures impacting employment dynamics in Bermuda. In short, this analysis underscores the varying economic landscapes and employment dynamics across different countries.

5.2 Visualisation

5.2.1 Sum of Receipts and Average of Receipts per Tourism by Country in Asia

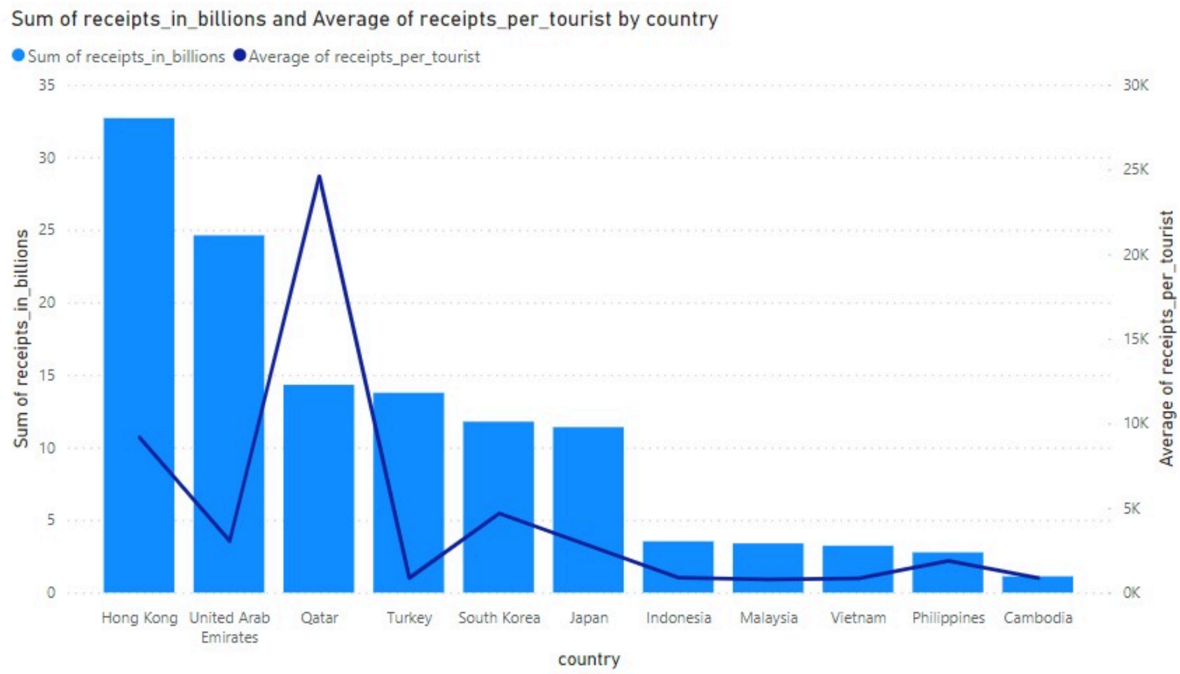


Figure 5.6 Bar Chart and Line Chart which show the Sum of Receipts and Average of Receipts per Tourism by Country

This visualisation shows the total tourism receipts in billions together with the average receipts per visitor for various countries. Qatar has the highest average money spent per tourist, indicating fewer but high-spending tourists. This may be because The 2022 FIFA World Cup, which Qatar hosted, revolutionised the travel and tourism sector in the nation. Football fans and viewers from all over the world were drawn to it. Following the World Cup, Qatar made the most of its increased international exposure by drawing more travellers looking for a luxury, cultural, and sporting experience. However, Hong Kong leads with the highest overall receipts, showing a big amount of tourists. The low total receipts and average spending of the UAE, Turkey, South Korea, and Japan indicate balanced tourism activity. As a result of either fewer tourists or lower average spending each visit, Indonesia, Malaysia, Vietnam, the Philippines, and Cambodia have lower overall receipts and average spending. These observations get attention to the different facets of tourism in these countries, referring to chances for focused advertising and financial support to improve tourist experiences and expenditures. Hence, to improve the richest level of Malaysia, the government of Malaysia may improve the promotions toward Malaysia culture, to enhance tourism performance.

5.2.2 Relationship between Average of Corruption Index and Average of GDP per Capita by Country

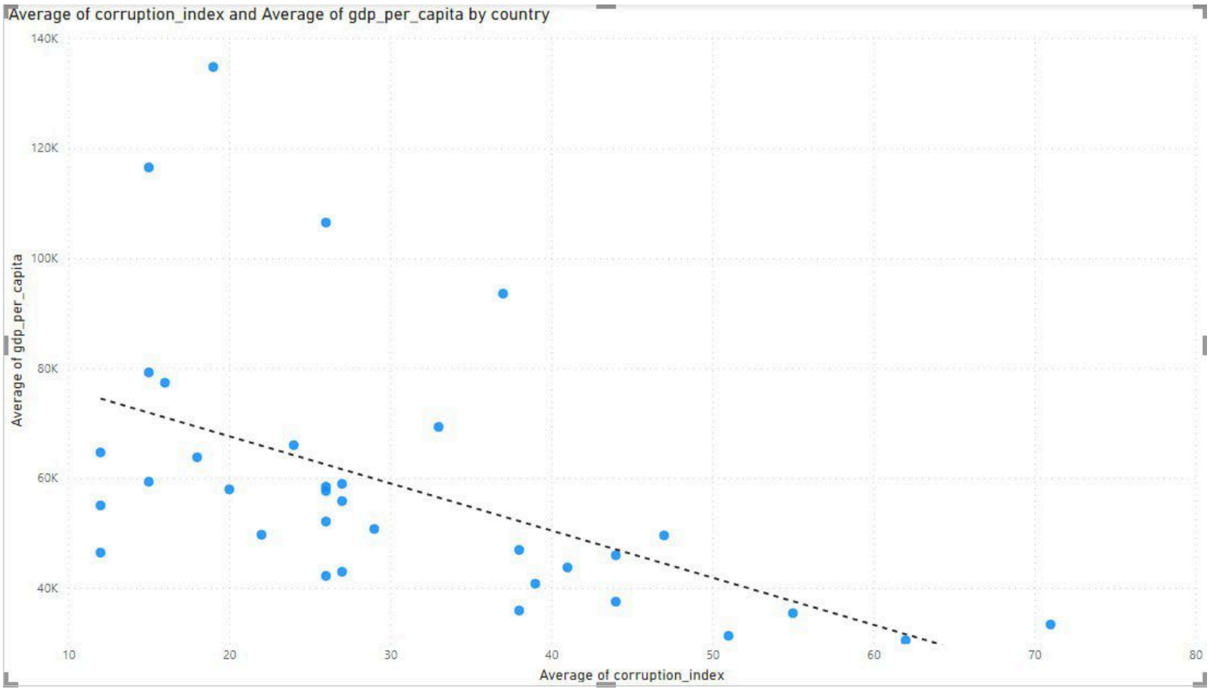


Figure 5.7 Scatter plot of Avg of corruption index and GDP per capita by country

This visualisation above shows the relationship between average Corruption perceptions index (CPI) and GDP per capita, a fascinating pattern starts to take shape. This plot with a trend has clearly revealed negative correlation between two variables. The trend line underscores the significant impact that countries that have lower CPI scores, which indicate less corruption, have greater GDP per capita. Conversely, countries with higher CPI ratings, which indicate greater corruption, tend to have lower GDP per capita. The relationship raises the possibility that corruption might hinder economic growth. Reduced levels of corruption can result in improved governance, a more welcoming business climate, and more effective resource allocation, all of which promote economic growth and raise GDP per capita. Conversely, increased levels of corruption can hinder economic growth by causing inefficiencies, discouraging investment, and misallocating resources, all of which can lead to a decrease in GDP per capita. To improve the richest level of Malaysia, the problem of corruption must be focused, and be decreased, and hence the GDP of Malaysia can be increased.

5.2.3 Grow of GDP in PPP in Malaysia from Year 1990 until 2019

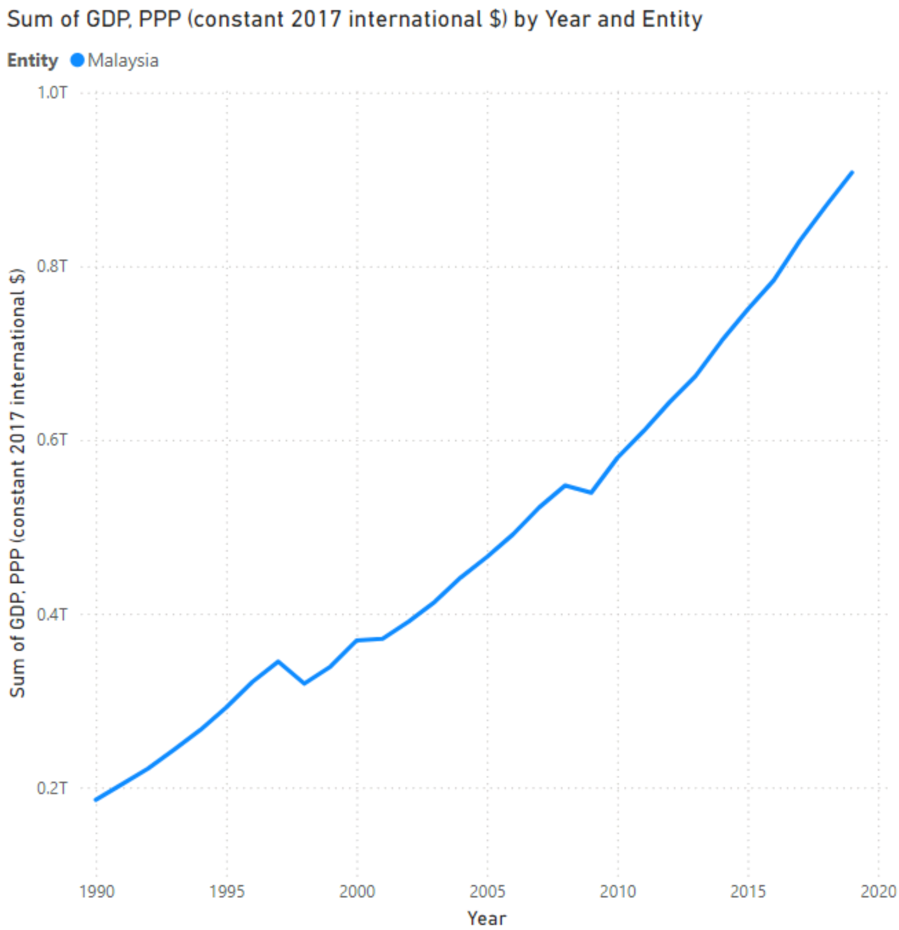


Figure 5.8 Line Chart of GDP in PPP in Malaysia from Year 1990 until 2019

The growth of purchasing power parity (PPP) in Malaysia from 1990 to 2019 is depicted in the line chart. Initially, there was a gradual increase in PPP value from 1990 to 1997, followed by a significant decrease between 1997 and 1998. Subsequently, there was a slow increase from 1998 to 2008, a slight decrease from 2008 to 2009, and a substantial increase from 2009 to 2019. Over the span of 30 years, Malaysia's PPP value rose from 185,816,880,000 to 907,831,600,000. PPP is a vital tool in macroeconomic analysis, enabling economists to compare living standards and economic performance across different countries. The steady increase in PPP reflects the gradual growth in living standards and economic productivity in Malaysia. This progress can be attributed to the implementation of various economic plans by the Malaysian government, including Malaysia Plans (RMK), Outline Perspective Plans (OPP), New Economic Policy (NEP), and Vision 2020. It is anticipated that with the continued efforts of the Malaysian government and the support of its citizens, the purchasing power parity in Malaysia will continue to increase steadily.

5.2.4 Grow of Output of GDP in Malaysia from Year 1990 until 2019

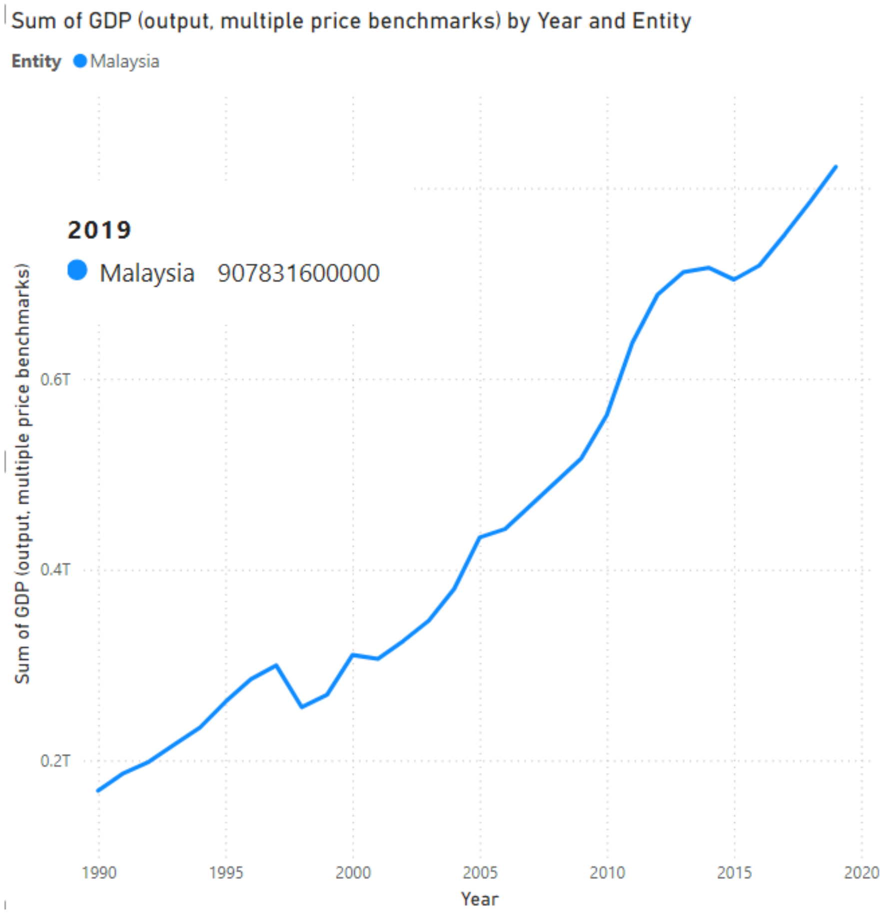


Figure 5.9 Line Chart of Output of GDP in Malaysia from Year 1990 until 2019

The line chart illustrates Malaysia's GDP growth from 1990 to 2019 showing a steady upward trend.GDP in Malaysia grew steadily after beginning at about 0.2 trillion in 1990, as a result of economic changes and industrialization. The decrease in GDP values of Malaysia happened in 1998, 2001, 2006, and 2015. The decrease in GDP values is due to the Asian Financial Crisis. Many Asian nations, including Malaysia, experienced a severe economic downturn as a result of the crisis. In 1998, the Malaysian GDP shrank by 7.4%. Also, The Malaysian economy expanded by 5% in 2015, a lower rate than the 6% growth in 2014. Numerous factors contributed to this growth slowdown, including a decline in exports and a reduction in private consumption5. Lastly, the GDP values increased to roughly 0.9 trillion by 2019 as a result of infrastructure spending, policy changes, growth due to exports, and the development of people. The graph highlights Malaysia's consistent economic expansion, crisis-resilience, and smooth adjustment to a more developed and service-oriented economy.

6.0 CONCLUSION

In conclusion, it is clearly shown that the strong positive relationship between the rich level of a country and the GDP per capita of the country. The higher the GDP level in overall and PPP, the higher ranking in richest rank among the world. Also, It is clearly shown that the main factor which influences the rich level of a country is the unemployment rate as well as the growing pattern of economics in Malaysia is slowly increasing. When a nation's working-age population is highly employed, it means that a sizable share of it is employed. This might also indicate that the nation has a robust economy, successful job-creation strategies, and a highly educated and proficient workforce. Plus, it is clearly shown that the richest country in the world is Luxembourg with a total GDP per capita of $ 134,754. Why can it become the richest country? By focusing on the factor influencing the rich level of the country, we may observe that Luxembourg has a total unemployment rate of 5.2 and a corruption index of 19.0Hence, it is important for the government to focus on the employment rate and the unemployment rate in a country to increase the GDP as well as enhance the rich level of a country. Upon completion, this study will provide valuable data-driven insights that can influence policy decisions aimed at promoting economic stability and job creation. The findings will be particularly useful for governments and international organisations as they develop strategies to achieve SDG 8. Furthermore, the research could pave the way for future studies exploring other related aspects, such as the impact of technology and innovation on economic growth and employment. Ultimately, this project is able to contribute to a sustainable economic future where growth leads to enhanced employment opportunities for all. Lastly, with the project conducted, more employment opportunities will be provided to the Malaysian as well as the economy of Malaysia will be increasing more rapidly with the growth in GDP values.This project required several obstacles to be overcome in order to be completed. Choosing the most pertinent datasets to meet the project's goals was one of the early difficulties encountered. The selected datasets were thoroughly examined and evaluated to make sure they matched the objectives and scope of the project. Selecting the right tools for the Extract, Transform, and Load (ETL) process was another difficulty. PostgreSQL was chosen as the database for storing and managing the data in the galaxy schema after considerable deliberation and teamwork. This choice made it possible to organise data well and to query and analyse it effectively. In the process of analysis, the datasets yielded significant insights into a range of economic factors.

Reference

Adeneye, Y. B., & Chu, E. Y. (2020). Managerial Aversion and Capital Structure: Evidence from Southeast Asia. Asian Academy of Management Journal of Accounting and Finance, 16(1), 155–183. https://doi.org/10.21315/aamjaf2020.16.1.8

FACT SHEET: 2022 National Travel and Tourism Strategy. (2022, June 6). U.S. Department of Commerce. https://www.commerce.gov/news/fact-sheets/2022/06/fact-sheet-2022-national-travel-and-tourism-strategy

Karabulut, Gokhan. (2010). The Relationship between GDP and Unemployment: Evidence from MENA Countries. Zagreb International Review of Economics and Business. 13. 17-28.

Most corrupt countries 2024. (n.d.). https://worldpopulationreview.com/country-rankings/most-corrupt-countries

RAPID ASSESSMENT OF COVID-19 RELATED POLICY AUDIT IN NEPAL Government of Nepal Nepal Health Research Council Ramshahpath, Kathmandu. (2020).

Staff, I., & Staff, I. (2019, March 29). Luxembourg: The Country with the Highest GDP Per Capita. The Santa Barbara Independent. https://www.independent.com/2015/01/11/luxembourg-country-highest-gdp-per-capita

Team, I. (2024, May 3). What is purchasing power parity (PPP), and how is it calculated? Investopedia. https://www.investopedia.com/updates/purchasing-power-parity-ppp/

Topic: Tourism industry in Qatar. (2023, December 21). Statista. https://www.statista.com/topics/11645/tourism-industry-in-qata

Transparency International. (2024, February 20). 2023 Corruption Perceptions Index: Explore the results. Transparency.org. https://www.transparency.org/en/cpi/2023

World Employment and Social Outlook: Trends 2021. (2024, February 1). International Labour Organization. https://www.ilo.org/publications/world-employment-and-social-outlook-trends-2021

World Unemployment rate 1960-2024. (n.d.). MacroTrends. https://www.macrotrends.net/global-metrics/countries/WLD/world/unemployment-rate

Zakaria, N. M., & Basah, M. Y. A. (2021). FINANCIAL DEVELOPMENT AND ECONOMIC GROWTH IN MALAYSIA FROM 1990 TO 2019: VECM APPROACH. Advanced International Journal of Banking, Accounting and Finance, 3(8), 37–54. https://doi.org/10.35631/aijbaf.38004

Appendix