BSD2343 DATA WAREHOUSING GROUP PROJECT

GROUP GOAL GETTERS

TITLE: DESIGN AND IMPLEMENTATION OF A DATA WAREHOUSE FOR CARDIOVASCULAR DISEASE RISK ANALYSIS

SDG 3: GOOD HEALTH AND WELL-BEING

PREPARED FOR: DR NOR AZUANA BINTI RAMLI

STUDENT ID	NAME	SECTION
SD23050	AIDA KAMILA FILZA BINTI ABDUL MANAF	02G
SD23068	NUR FARAHIEMAH BINTI AB RAHIN	01G
SD23021	NURFARAH ASYIQIN BINTI MD ADIM	02G
SD23054	ALIA AYUNNI BINTI MOHD SHUKRI	02G
SD23031	NUR ALIYYATUL QUBRA BINTI SHARMAN	01G

TABLE OF CONTENTS

1.0 BACKGROUND

1.1 Project Description

In a futuristic world, millions of people worldwide, despite the advanced technologies, are still impacted by disease and one of them is cardiovascular disease (CDV). This disease continues to grow to be one of the leading causes of fatalities globally. The rise of this disease has been observed and some factors have been concluded to become the main factors of it, including poor access to healthcare in some areas, unhealthy lifestyle and genetic factors. The risk of this disease is significantly influenced by lifestyle factors like lack of physical exercise and smoking, cholesterol, diabetes, obesity and hypertension.

It is absolutely vital to have a system that can effectively evaluate and predict the risks of this condition in order to manage the rising prevalence. Conventional risk assessment methods often use various sources of information that can leave out crucial knowledge on the contributing factor, for example medical histories, lifestyle habits and socioeconomic status. A broad perspective is needed to handle the humongous dataset, however the limitations of it prevents early detection and effective prevention efforts.

The aim of this project is to model and implement a data warehouse for cardiovascular disease risk analysis. Health metrics like cholesterol and blood pressure readings, demographic details, lifestyle information and clinical records will be integrated into the database. Medical professionals will get tremendous help from this risk analysis as they can be more equipped to identify the trends and correlations related to cardiovascular health and deliver data-driven insights for better patient care and prevention strategies,

The project will not only improve the ability to predict the risk of heart disease but also yield valuable information on how particular risk factors contribute to cardiovascular disease in varied populations and geographical areas. Through high-end data processing and analysis, the data warehouse will culminate in an improved understanding of the trends and underlying causes of cardiovascular diseases, which will reduce mortality rates and improve patient outcomes.

1.2 Problem to be Solved

Cardiovascular diseases continue to become a leading major global health concern with still a quite number of people oblivious to their own risks until they realize it is too late. Despite multiple solutions and efforts to enhance early detection, restricted analytic power, lack of integration and fragmented data continue to be obstacles in a lot of healthcare systems. A comprehensive risk analysis is significantly limited due to the inability to combine various variables such as lab results, health histories, lifestyle and demographics into a single and easily accessible system.

Traditional ways to assess heart attack risks are likely to ignore the entire spectrum of risk indicators and pay full attention to a small factor. For instance, mental health, family history, physical activity and nutrition are not considered as a possible factor as the medical professionals only observe cholesterol levels and blood pressure. It is also crucial to look into that, but it does not mean they should dismiss other possible indicators. As a result from that, they missed the chance to detect individuals who are at risk which is jeopardising the efficacy of prevention and intervention strategies.

In addition, the lack of health system integration has a tendency to increase this issue. Patient data are generally stored in distinct systems across various hospitals, clinics, and other medical centers, and therefore tend to be unavailable and unprocessed as a set. Unless these informations are brought under a centralized platform that performs an aggregation and a sorting of the data, healthcare providers are limited to performing advanced analytics, risk patterns identification, or even prediction of future cardiovascular health with accuracy.

The goal of this project is to counter these challenges through the establishment and deployment of a data warehouse that unifies disparate datasets across various health care systems with a single point of cardiovascular disease risk assessment. By the aggregation of demographics, health metrics, medical history, and lifestyle information, the data warehouse will enable physicians to provide more accurate risk assessments and formulate targeted treatment plans. Lastly, the project will lead to an improved, more proactive system for prevention and control of cardiovascular disease by bringing in and combining all relevant data in an effective manner.

1.3 Objectives

This project outlines several objectives to address the problem statement which are:

- To construct a data warehouse for heart risk analysis that includes the data of lifestyle, clinical and demographic
- To visualize an interactive dashboard to display the risk indicators
- To identify the population group that has a high risk for cardiovascular disease and give solutions for healthcare decision-making

1.4 Data Schema

A data schema is the blueprint or design that specifies how information is arranged, saved, and retrieved in a database management system (DBMS). The tables, fields, relationships, and data types that facilitate business intelligence and analytical queries are described. Furthermore, it is very important for keeping data in check and to make sure that queries run smoothly. For this project, we have 6 datasets that we use which are health metrics, lifestyle, medical history, patients, risk assessment and socioeconomic status as shown below:

No	Table Name	Column Name	Data Type	Description
1	healthmetrics	patient_id	String	Unique identifier for each patient
		cholestrol	Numeric	Cholesterol level
		blood_pressure	String	Blood pressure reading
		heart_rate	Numeric	Resting heart rate in beats (bpm)
		triglycerides	Numeric	Triglyceride level

		bmi	Numeric	3ody mass inde
2	lifestyle	patient_id	String	Jnique identifier for each patient
		smoking	Boolean	Го indicate if the patient mokes or not
		obesity	Boolean	Го indicate if the patient has besity or not
		lcohol_consumption	Boolean	Го indicate if the patient onsumes alcohol or not
		diet	String	Dietary classification healthy, average, etc.)
		physical_activity_days_per_week	Numeric	Number of days per week he patient does exercises
		sleep_hours_per_day	Numeric	Number of hours the patient leeps a day
		sedentary_hours_per_day	Numeric	Number of hours the patient as little to no physical novement
		exercise_hours_per_week	Numeric	Total hours of exercise per veek for patients
3	medicalhistory	patient_id	String	Jnique identifier for each patient
		diabetes	Boolean	Го indicate if the patient has liabetes or not
		previous_heart_problem	Boolean	Го indicate if the patient had ny previous heart problem
		medication_use	Boolean	Го indicate if the patient is urrently taking any nedication
		stress_level	Numeric	The patient's reported stress evel
4	patients	patient_id	String	Jnique identifier for each patient
		age	Numeric	Patient's age

		sex	String	Patient's gender
		family_history	Boolean	To indicate if the patient has a family history of heart disease
		country	String	Country of residence
		continent	String	Continent of residence
		hemisphere	String	Position of residence on the hemisphere (northern, southern etc.)
5	riskassessment	patient_id	String	Unique identifier for each patient
		heart_attack_risk	Boolean	To indicate if the patient is at risk of a heart attack
6	socioeconomicstatus	patient_id	String	Unique identifier for each patient
		income	Numeric	Patient's income

Check for data type:

```
[2] import pandas as pd
    healthmetrics = pd.read_csv("/content/HealthMetrics.csv")
    lifestyle = pd.read_csv("/content/Lifestyle.csv")
    medicalhistory = pd.read_csv("/content/MedicalHistory.csv")
    patients = pd.read_csv("/content/Patients.csv")
    risk = pd.read_csv("/content/RiskAssessment.csv")
    status = pd.read_csv("/content/SocioeconomicStatus.csv")
```

Figure 1.4.1 shows the library that was used to find the data schema

```
patients.dtypes
patients.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8763 entries, 0 to 8762
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Patient ID      8763 non-null   object
 1   Age             8762 non-null   float64
 2   Sex             8762 non-null   object
 3   Family History  8763 non-null   int64
 4   Country         8762 non-null   object
 5   Continent       8763 non-null   object
 6   Hemisphere      8763 non-null   object
dtypes: float64(1), int64(1), object(5)
memory usage: 479.4+ KB
```

Figure 1.4.2 patients table

Based on figure 1.4.2, the patients' raw dataset is displayed above, and it contains the patients' personal information like age, gender and origin. Each row represents each patient with a unique identifier as a primary key which is patient ID. There are two numeric columns and the others are strings.

```
healthmetrics.dtypes
healthmetrics.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8763 entries, 0 to 8762
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Patient ID      8763 non-null   object
 1   Cholesterol     8762 non-null   float64
 2   Blood Pressure  8762 non-null   object
 3   Heart Rate      8763 non-null   int64
 4   Triglycerides   8763 non-null   int64
 5   BMI             8762 non-null   float64
dtypes: float64(2), int64(2), object(2)
memory usage: 410.9+ KB
```

Figure 1.4.3 healthmetrics table

Based on the figure above, it shows the healthmetric data schema and it consists of 6 columns. This dataset tells about the patients' current health condition and it contains 4 numeric columns and 2 strings.

```
lifestyle.dtypes
lifestyle.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8763 entries, 0 to 8762
Data columns (total 9 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Patient ID                      8763 non-null   object 
 1   Smoking                         8763 non-null   int64  
 2   Obesity                         8763 non-null   int64  
 3   Alcohol Consumption             8762 non-null   float64
 4   Diet                            8762 non-null   object 
 5   Physical Activity Days Per Week 8762 non-null   float64
 6   Sleep Hours Per Day             8762 non-null   float64
 7   Sedentary Hours Per Day         8763 non-null   float64
 8   Exercise Hours Per Week         8762 non-null   float64
dtypes: float64(5), int64(2), object(2)
memory usage: 616.3+ KB
```

Figure 1.4.4 lifestyle table

The data schema that is displayed above is about the patient's lifestyle. This table has a total of 9 columns and there are 2 string columns, 3 Boolean columns and the rest are numeric columns. This table has detailed data on how patients manage their daily habits, routines, and behaviours that may influence their mental well-being.

```
medicalhistory.dtypes
medicalhistory.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8763 entries, 0 to 8762
Data columns (total 5 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Patient ID               8763 non-null   object 
 1   Diabetes                 8763 non-null   int64  
 2   Previous Heart Problems  8762 non-null   float64
 3   Medication Use           8763 non-null   int64  
 4   Stress Level             8762 non-null   float64
dtypes: float64(2), int64(2), object(1)
memory usage: 342.4+ KB
```

Figure 1.4.5 medicalhistory table

Based on figure 1.4.5, the data schema above can be seen there are 5 columns. Most of the columns are numeric values except for patient ID. This data schema tells about the patient's medical history to observe the indicators that cause heart risk.

```
risk.dtypes
risk.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8763 entries, 0 to 8762
Data columns (total 2 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Patient ID         8763 non-null   object
 1   Heart Attack Risk  8762 non-null   float64
dtypes: float64(1), object(1)
memory usage: 137.1+ KB
```

Figure 1.4.6 risk assessment table

The data schema above shows the heart attack risk for patients and it contains two columns. Patient ID is a string and the heart attack risk is numeric.

```
status.dtypes
status.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8763 entries, 0 to 8762
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Patient ID  8763 non-null   object
 1   Income      8762 non-null   float64
dtypes: float64(1), object(1)
memory usage: 137.1+ KB
```

Figure 1.4.7 socioeconomic status table

Lastly, the socioeconomic table consists of two columns which are patient ID (string) and income (numeric) . This table shows the income of the patients.

2.0 ARCHITECTURE

2.1 Architecture Structure

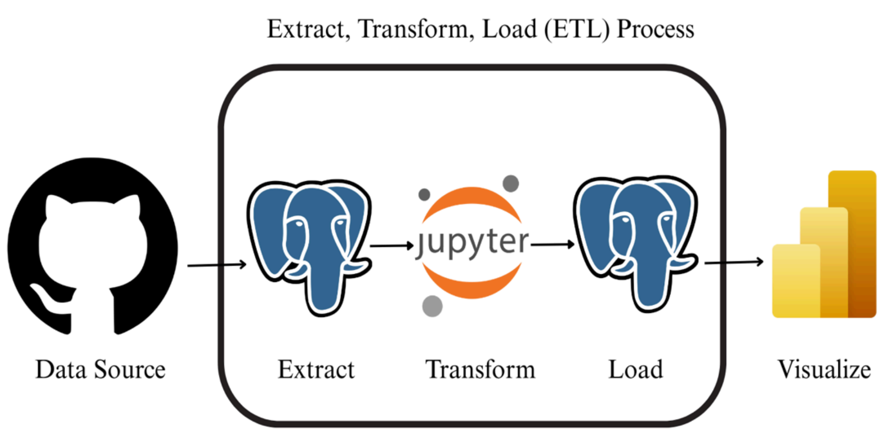


Figure 2.1.1

Based on Figure 2.1, the HeartriskDB dataset was obtained in the Github open-source platform. The dataset contains 6 tables which are health metrics, lifestyle, medical history, patients, risk assessments and socioeconomic status. Through a thorough investigation, our group has chosen Kimball's approach to build and design our project entitled, “Design and Implementation of a Data Warehouse for Cardiovascular Disease Risk Analysis”. This approach has a strong emphasis on creating data marts that are both performance-optimized and user-friendly. This method employs a more easy data integration and allows for an effective data analysis of heart attack risk factors.

We started by extracting the data into PostgreSQL for the initial step and building databases referring to our tables. Then, we imported the data to Jupyter Notebook for the transformation process. We started with installing important and necessary libraries that are needed. The data went through data cleaning and transformation processes where null values are identified and removed. The data was loaded back into PostgreSQL and was ready for data integration processes like the OLAP operations. The operations were executed for a better analysis and clearer view for an in-depth visualisation.

Finally, we used Power BI to visualize our findings. This is a platform that helps to create interactive visualizations, and the results are interpreted for meaningful insights and better understanding towards the data.

2.2 ETL Pipeline

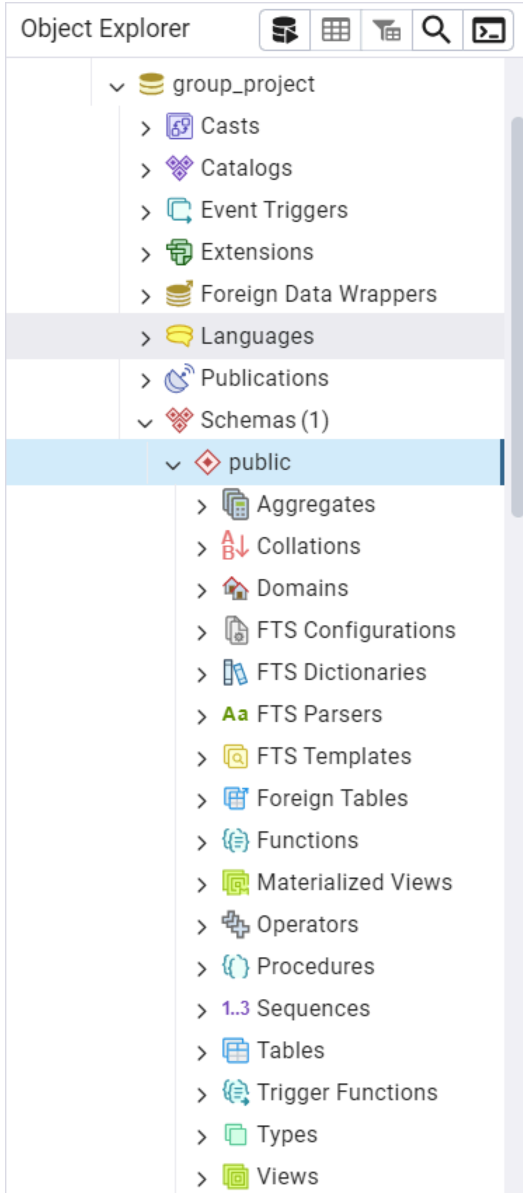
Figure 2.2.1

Figure 2.2.1 displays the ETL pipeline for the dataset. The process involves extracting data from a source, transforming it, and loading as Data Warehouse System. For this project, in details, we use PostgreSQL to extract data from CSV file, transformed it using Python in Jupyter Notebook connected to PostgreSQL, loaded the clean data back into PostgreSQL, and finally visualized the data using OLAP and Power BI. In total we have 6 tables in a database, so the ETL process is repeated to those 6 tables, and the data is ready to be visualized and analysed.

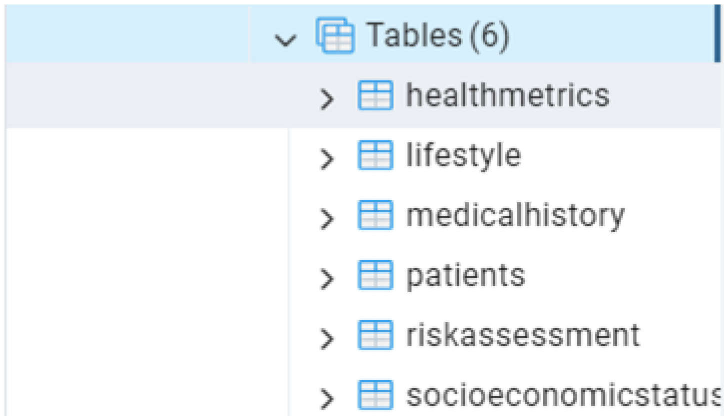
2.3 ETL Process

Extract:

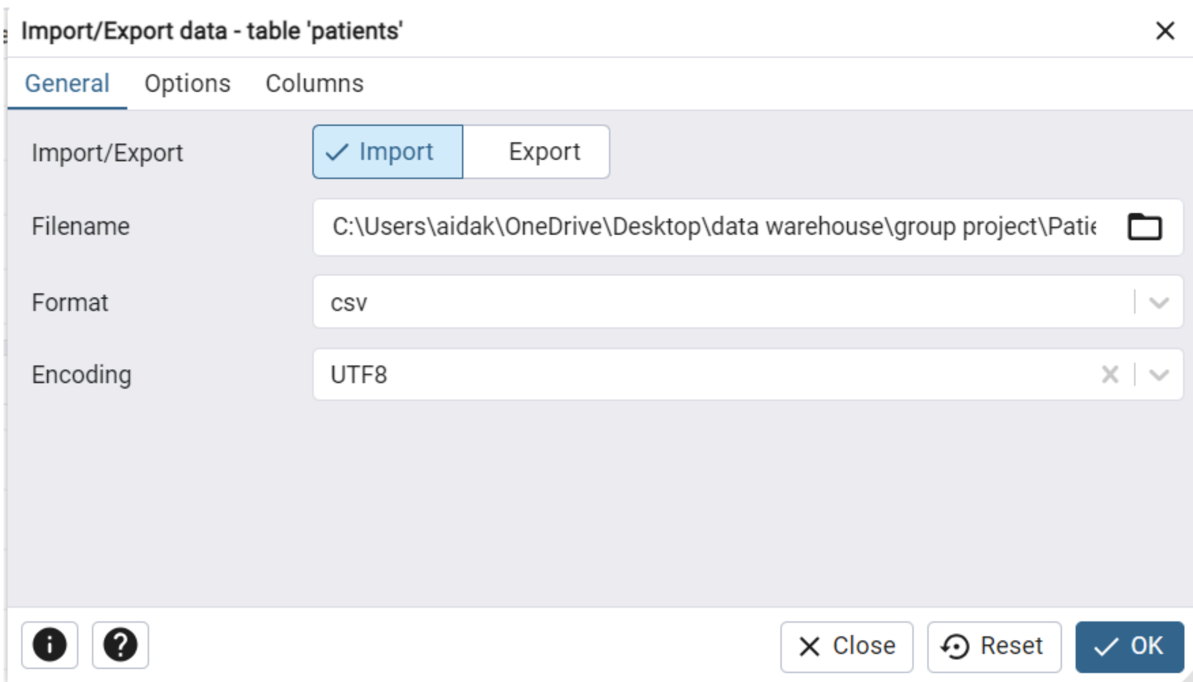
The datasets must be saved in a PostgreSQL database before the ETL procedure can begin. To begin with, make a new database and utilize database connectors to pull pertinent information from every table.



In PostgreSQL, we have successfully created a database called 'group_project'



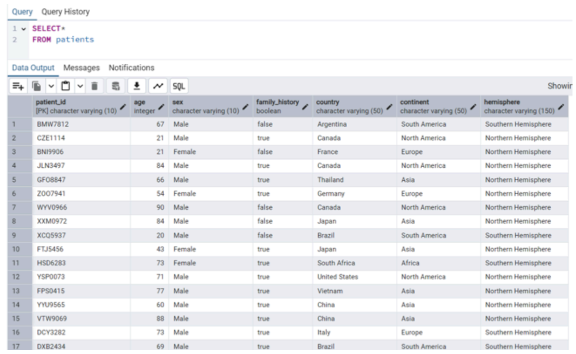
Tables created



Import CSV file into table, repeat to other 5 CSV file

Create table	Output

```sql
CREATE TABLE Patients (
    patient_id VARCHAR(10) PRIMARY KEY,
    Age INT,
    Sex VARCHAR(10),
    Family_History BOOLEAN,
    Country VARCHAR (50),
    Continent VARCHAR (50),
    Hemisphere VARCHAR (150)
);
```



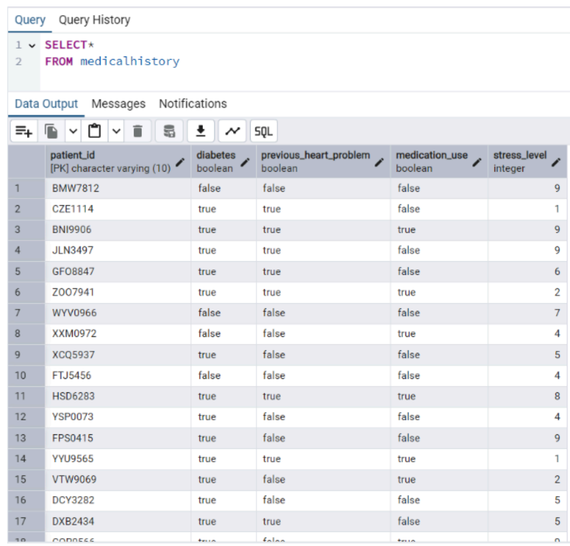
Query Query History

```sql
SELECT*
FROM patients
```

Data Output Messages Notifications

	patient_id [PK] character varying (10)	age integer	sex character varying (10)	family_history boolean	country character varying (50)	continent character varying (50)	hemisphere character varying (150)
1	BMW7812	67	Male	false	Argentina	South America	Southern Hemisphere
2	CZE1114	21	Male	true	Canada	North America	Northern Hemisphere
3	BNI9906	21	Female	false	France	Europe	Northern Hemisphere
4	JLN3497	84	Male	true	Canada	North America	Northern Hemisphere
5	GFO8847	66	Male	true	Thailand	Asia	Northern Hemisphere
6	ZOO7941	54	Female	true	Germany	Europe	Northern Hemisphere
7	WYV0966	90	Male	false	Canada	North America	Northern Hemisphere
8	XXM0972	84	Male	false	Japan	Asia	Northern Hemisphere
9	XCQ5937	20	Male	false	Brazil	South America	Southern Hemisphere
10	FTJ5456	43	Female	true	Japan	Asia	Northern Hemisphere
11	HSD6283	73	Female	true	South Africa	Africa	Southern Hemisphere
12	YSP0073	71	Male	true	United States	North America	Northern Hemisphere
13	FPS0415	77	Male	true	Vietnam	Asia	Northern Hemisphere
14	YYU9565	60	Male	true	China	Asia	Northern Hemisphere
15	VTW9069	88	Male	true	China	Asia	Northern Hemisphere
16	DCY3282	73	Male	true	Italy	Europe	Southern Hemisphere
17	DXB2434	69	Male	true	Brazil	South America	Southern Hemisphere

```sql
CREATE TABLE MedicalHistory (
    patient_id VARCHAR(10),
    diabetes BOOLEAN,
    previous_heart_problem BOOLEAN,
    medication_use BOOLEAN,
    stress_level INT,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES Patients (patient_id)
);
```



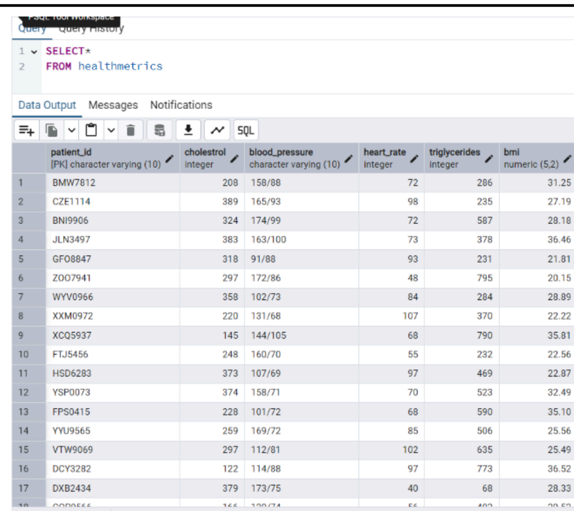
Query Query History

```sql
SELECT*
FROM medicalhistory
```

Data Output Messages Notifications

	patient_id [PK] character varying (10)	diabetes boolean	previous_heart_problem boolean	medication_use boolean	stress_level integer
1	BMW7812	false	false	false	9
2	CZE1114	true	true	false	1
3	BNI9906	true	true	true	9
4	JLN3497	true	true	false	9
5	GFO8847	true	true	false	6
6	ZOO7941	true	true	true	2
7	WYV0966	false	false	false	7
8	XXM0972	false	false	true	4
9	XCQ5937	true	false	false	5
10	FTJ5456	false	false	false	4
11	HSD6283	true	true	true	8
12	YSP0073	true	false	false	4
13	FPS0415	true	false	false	9
14	YYU9565	true	true	true	1
15	VTW9069	true	false	true	2
16	DCY3282	true	false	false	5
17	DXB2434	true	true	false	5

```sql
CREATE TABLE HealthMetrics(
    patient_id VARCHAR (10),
    Cholestrol INT,
    Blood_Pressure VARCHAR (10),
    Heart_Rate INT,
    Triglycerides INT,
    BMI DECIMAL (5,2),
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES Patients (patient_id)
);
```



Query Query History

```sql
SELECT*
FROM healthmetrics
```

Data Output Messages Notifications

	patient_id [PK] character varying (10)	cholestrol integer	blood_pressure character varying (10)	heart_rate integer	triglycerides integer	bmi numeric (5,2)
1	BMW7812	208	158/88	72	286	31.25
2	CZE1114	389	165/93	98	235	27.19
3	BNI9906	324	174/99	72	587	28.18
4	JLN3497	383	163/100	73	378	36.46
5	GFO8847	318	91/88	93	231	21.81
6	ZOO7941	297	172/86	48	795	20.15
7	WYV0966	358	102/73	84	284	28.89
8	XXM0972	220	131/68	107	370	22.22
9	XCQ5937	145	144/105	68	790	35.81
10	FTJ5456	248	160/70	55	232	22.56
11	HSD6283	373	107/69	97	469	22.87
12	YSP0073	374	158/71	70	523	32.49
13	FPS0415	228	101/72	68	590	35.10
14	YYU9565	259	169/72	85	506	25.56
15	VTW9069	297	112/81	102	635	25.49
16	DCY3282	122	114/88	97	773	36.52
17	DXB2434	379	173/75	40	68	28.33

```sql
CREATE TABLE SocioeconomicStatus(
    patient_id VARCHAR(10),
    Income INT,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES Patients (patient_id)

);
```

Query Query History

```sql
SELECT*
FROM socioeconomicstatus
```

Data Output Messages Notifications

	patient_id [PK] character varying (10)	income integer
1	BMW7812	261404
2	CZE1114	285768
3	BNI9906	235282
4	JLN3497	125640
5	GFO8847	160555
6	ZOO7941	241339
7	WYV0966	190450
8	XXM0972	122093
9	XCQ5937	25086
10	FTJ5456	209703
11	HSD6283	50030
12	YSP0073	163066
13	FPS0415	29886
14	YYU9565	292173
15	VTW9069	165300
16	DCY3282	265839

```sql
CREATE TABLE Lifestyle(
    patient_id VARCHAR (10),
    Smoking BOOLEAN,
    Obesity BOOLEAN,
    Alcohol_Consumption BOOLEAN,
    Diet VARCHAR (50),
    Physical_Activity_Days_Per_Week INT,
    Sleep_Hours_Per_Day INT,
    Sedentary_Hours_Per_Day FLOAT,
    Exercise_Hours_Per_Week FLOAT,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES Patients (patient_id)
);
```

Query Query History Scratch Pad

```sql
SELECT*
FROM lifestyle
```

Data Output Messages Notifications

Showing rows: 1 to 1000 Page No: 1 of 9

	patient_id [PK] character varying (10)	smoking boolean	obesity boolean	alcohol_consumption boolean	diet character varying (50)	physical_activity_days_per_week integer	sleep_hours_per_day integer	sedentary_hours_per_day double precision	exercise_hours_per_week double precision
1	BMW7812	true	false	false	Average	0	6	6.615001453	4.168188835
2	CZE1114	true	true	true	Unhealthy	1	7	4.96345884	1.813241618
3	BNI9906	false	false	false	Healthy	4	4	9.463425838	2.078352986
4	JLN3497	true	false	true	Average	3	4	7.648980824	9.828129593
5	GFO8847	true	true	false	Unhealthy	1	5	1.514820926	5.80429882
6	ZOO7941	true	false	true	Unhealthy	5	10	7.798752409	0.625008024
7	WYV0966	true	false	true	Healthy	4	10	0.627356001	4.098177091
8	XXM0972	true	true	true	Average	6	7	10.54378024	3.427928754
9	XCQ5937	true	true	false	Average	7	4	11.34878687	16.86830224
10	FTJ5456	true	true	true	Unhealthy	7	7	4.055114782	0.194515061
11	HSD6283	true	false	true	Average	0	4	8.919879241	16.84198759
12	YSP0073	true	true	true	Average	4	8	7.227338314	8.251995072
13	FPS0415	true	true	true	Unhealthy	7	6	10.91752425	19.63326816
14	YYU9565	true	false	true	Healthy	1	4	8.727417212	17.03737418
15	VTW9069	true	false	true	Unhealthy	3	6	10.42548958	15.38760463
16	DCY3282	true	false	true	Average	5	8	10.08647897	14.5596645
17	DXB2434	true	true	true	Average	3	6	9.060508568	4.184648046

```sql
CREATE TABLE RiskAssessment(
    patient_id VARCHAR (10),
    Heart_Attack_Risk BOOLEAN,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES Patients (patient_id)
);
```

Query Query History

```sql
SELECT*
FROM riskassessment
```

Data Output Messages Notifications

	patient_id [PK] character varying (10)	heart_attack_risk boolean
1	BMW7812	false
2	CZE1114	false
3	BNI9906	false
4	JLN3497	false
5	GFO8847	false
6	ZOO7941	true
7	WYV0966	true
8	XXM0972	true
9	XCQ5937	false
10	FTJ5456	false
11	HSD6283	false
12	YSP0073	false
13	FPS0415	true
14	YYU9565	true
15	VTW9069	false
16	DCY3282	true

This table shows the query to create the tables and the outputs for each table

Transform:

After the raw data has been extracted into pgAdmin, we connect our pgAdmin with the Jupyter Notebook to proceed to the next step which transforms the data.

```
[*]: ! pip install ipython-sql
     ! pip install sqlalchemy
     ! pip install psycopg2
     ! pip install python-sql
     ! pip install pandas-sql
     ! pip install sql-queries
     !pip install missingno
```

This figure shows the packages that we installed

After the packages have been installed, we load ipython-sql with the following command:

```
: %reload_ext sql
```

Calling the create engine function:

```
from sqlalchemy import create_engine
```

Importing necessary libraries for ETL process:

```
import pandas as pd
import psycopg2 as ps
import pandas.io.sql as sqlio
import missingno as msno
```

Connect from PgAdmin into Jupyter Notebook:

```
connectpg = ps.connect(dbname="group_project",
                       user="postgres", password ="1234", host="localhost",
                       port="5432")
```

After connecting the PgAdmin with the Jupyter Notebook the dataset needs to be cleaned as it is important to do the data cleaning process. Some connectors are installed to ensure the data can be transferred from PostgreSQL to Python. The data needs to be stored in a data frame by the pandas library to ease the data cleaning process.

Selecting the first table patients and checking if there are any null or missing values.

```
sql = "SELECT * FROM pg_catalog.pg_tables"
```

```
sql = "SELECT * FROM patients"
```

```python
patients =sqlio.read_sql_query(sql,connectpg)
print(patients)
```

```
     patient_id   age     sex  family_history         country      continent  \
0       BMW7812  67.0    Male           False       Argentina  South America   
1       CZE1114  21.0    Male            True          Canada  North America   
2       BNI9906  21.0  Female           False          France         Europe   
3       JLN3497  84.0    Male            True          Canada  North America   
4       GFO8847  66.0    Male            True        Thailand           Asia   
...         ...   ...     ...             ...             ...            ...   
8758    MSV9918  60.0    Male            True        Thailand           Asia   
8759    QSV6764  28.0  Female           False          Canada  North America   
8760    XKA5925  47.0    Male            True          Brazil  South America   
8761    EPE6801  36.0    Male           False          Brazil  South America   
8762    ZWN9666  25.0  Female            True  United Kingdom         Europe   

               hemisphere  
0     Southern Hemisphere  
1     Northern Hemisphere  
2     Northern Hemisphere  
3     Northern Hemisphere  
4     Northern Hemisphere  
...                   ...  
8758  Northern Hemisphere  
8759  Northern Hemisphere  
8760  Southern Hemisphere  
8761  Southern Hemisphere  
8762  Northern Hemisphere  
```

```
patients.isnull().sum()
```

```
patient_id        0
age               1
sex               1
family_history    0
country           1
continent         0
hemisphere        0
dtype: int64
```

Checking null for patients

```
7]: patients_new=patients.dropna()
    patients_new
```

7]:

	patient_id	age	sex	family_history	country	continent	hemisphere
0	BMW7812	67.0	Male	False	Argentina	South America	Southern Hemisphere
1	CZE1114	21.0	Male	True	Canada	North America	Northern Hemisphere
2	BNI9906	21.0	Female	False	France	Europe	Northern Hemisphere
3	JLN3497	84.0	Male	True	Canada	North America	Northern Hemisphere
4	GFO8847	66.0	Male	True	Thailand	Asia	Northern Hemisphere
...	...	...	...	...	...	...	...
8758	MSV9918	60.0	Male	True	Thailand	Asia	Northern Hemisphere
8759	QSV6764	28.0	Female	False	Canada	North America	Northern Hemisphere
8760	XKA5925	47.0	Male	True	Brazil	South America	Southern Hemisphere
8761	EPE6801	36.0	Male	False	Brazil	South America	Southern Hemisphere
8762	ZWN9666	25.0	Female	True	United Kingdom	Europe	Northern Hemisphere

Drop null values

```
patients_new.isna().sum()
```

```
patient_id        0
age               0
sex               0
family_history    0
country           0
continent         0
hemisphere        0
dtype: int64
```

Recheck the null values

Select the second table medicalhistory and check if there are any null or missing values.

```
sql = "SELECT * FROM medicalhistory"
```

```
medicalhistory =sqlio.read_sql_query(sql,connectpg)
print(medicalhistory)
```

```
     patient_id  diabetes  previous_heart_problem  medication_use  stress_level
0       BMW7812     False                   False           False           9.0
1       CZE1114      True                    True           False           1.0
2       BNI9906      True                    True            True           9.0
3       JLN3497      True                    True           False           9.0
4       GFO8847      True                    True           False           6.0
...         ...       ...                     ...             ...           ...
8758    MSV9918      True                    True            True           8.0
8759    QSV6764      True                   False           False           8.0
8760    XKA5925     False                    True           False           5.0
8761    EPE6801      True                    True            True           5.0
8762    ZWN9666      True                   False           False           8.0

[8763 rows x 5 columns]
```

```
medicalhistory.isna().sum()
```

```
patient_id               0
diabetes                 0
previous_heart_problem   1
medication_use           0
stress_level             1
dtype: int64
```

Checking null values

```
medicalhistory_new=medicalhistory.dropna()
medicalhistory_new
```

	patient_id	diabetes	previous_heart_problem	medication_use	stress_level
0	BMW7812	False	False	False	9.0
1	CZE1114	True	True	False	1.0
2	BNI9906	True	True	True	9.0
3	JLN3497	True	True	False	9.0
4	GFO8847	True	True	False	6.0
...	...	...	...	...	...
8758	MSV9918	True	True	True	8.0
8759	QSV6764	True	False	False	8.0
8760	XKA5925	False	True	False	5.0
8761	EPE6801	True	True	True	5.0
8762	ZWN9666	True	False	False	8.0

```
medicalhistory_new.isna().sum()
```

```
patient_id                0
diabetes                  0
previous_heart_problem    0
medication_use            0
stress_level              0
dtype: int64
```

Drop and recheck the null values

Selecting the third table healthmetrics and checking if there are any null and missing values.

```
sql = "SELECT * FROM healthmetrics"
```

```python
healthmetrics =sqlio.read_sql_query(sql,connectpg)
print(healthmetrics)
```

```
     patient_id  cholestrol blood_pressure  heart_rate  triglycerides    bmi
0       BMW7812       208.0         158/88          72            286  31.25
1       CZE1114       389.0         165/93          98            235  27.19
2       BNI9906       324.0         174/99          72            587  28.18
3       JLN3497       383.0        163/100          73            378  36.46
4       GFO8847       318.0          91/88          93            231  21.81
...         ...         ...            ...         ...            ...    ...
8758    MSV9918       121.0          94/76          61             67  19.66
8759    QSV6764       120.0        157/102          73            617  23.99
8760    XKA5925       250.0         161/75         105            527  35.41
8761    EPE6801       178.0         119/67          60            114  27.29
8762    ZWN9666       356.0         138/67          75            180  32.91

[8763 rows x 6 columns]
```

```python
healthmetrics.isna().sum()
```

```
patient_id        0
cholestrol        1
blood_pressure    1
heart_rate        0
triglycerides     0
bmi               1
dtype: int64
```

Checking null values

```
healthmetrics_new=healthmetrics.dropna()
healthmetrics_new
```

	patient_id	cholestrol	blood_pressure	heart_rate	triglycerides	bmi
0	BMW7812	208.0	158/88	72	286	31.25
1	CZE1114	389.0	165/93	98	235	27.19
2	BNI9906	324.0	174/99	72	587	28.18
3	JLN3497	383.0	163/100	73	378	36.46
4	GFO8847	318.0	91/88	93	231	21.81
...	...	...	...	...	...	...
8758	MSV9918	121.0	94/76	61	67	19.66
8759	QSV6764	120.0	157/102	73	617	23.99
8760	XKA5925	250.0	161/75	105	527	35.41
8761	EPE6801	178.0	119/67	60	114	27.29
8762	ZWN9666	356.0	138/67	75	180	32.91

```
healthmetrics_new.isna().sum()
```

```
patient_id        0
cholestrol        0
blood_pressure    0
heart_rate        0
triglycerides     0
bmi               0
dtype: int64
```

Drop and recheck the null values

Selecting the fourth table socioeconomicstatus and checking if there are any null and missing values.

```
sql = "SELECT * FROM socioeconomicstatus"
```

```
socioeconomicstatus=sqlio.read_sql_query(sql,connectpg)
print(socioeconomicstatus)
```

```
     patient_id    income
0       BMW7812  261404.0
1       CZE1114  285768.0
2       BNI9906  235282.0
3       JLN3497  125640.0
4       GFO8847  160555.0
...         ...       ...
8758    MSV9918  235420.0
8759    QSV6764  217881.0
8760    XKA5925   36998.0
8761    EPE6801  209943.0
8762    ZWN9666  247338.0
```

```
socioeconomicstatus.isnull().sum()
```

```
patient_id    0
income        1
dtype: int64
```

Checking null values

```
]: socioeconomicstatus_new=socioeconomicstatus.dropna()
   socioeconomicstatus_new
```

]:

	patient_id	income
0	BMW7812	261404.0
1	CZE1114	285768.0
2	BNI9906	235282.0
3	JLN3497	125640.0
4	GFO8847	160555.0
...	...	...
8758	MSV9918	235420.0
8759	QSV6764	217881.0
8760	XKA5925	36998.0
8761	EPE6801	209943.0
8762	ZWN9666	247338.0

```
socioeconomicstatus_new.isna().sum()
```

```
patient_id    0
income        0
dtype: int64
```

Drop and recheck the null values

Selecting the fifth table lifestyle and checking if there are any null and missing values.

```
sql = "SELECT * FROM lifestyle"
```

```
]: lifestyle =sqlio.read_sql_query(sql,connectpg)
print(lifestyle)
```

```
     patient_id  smoking  obesity alcohol_consumption       diet  \
0       BMW7812     True    False               False    Average
1       CZE1114     True     True                True  Unhealthy
2       BNI9906    False    False               False    Healthy
3       JLN3497     True    False                True    Average
4       GFO8847     True     True               False  Unhealthy
...         ...      ...      ...                 ...        ...
8758    MSV9918     True    False                True    Healthy
8759    QSV6764    False     True               False    Healthy
8760    XKA5925     True     True                True    Average
8761    EPE6801     True    False               False  Unhealthy
8762    ZWN9666    False    False                True    Healthy

      physical_activity_days_per_week  sleep_hours_per_day  \
0                                 0.0                  6.0
1                                 1.0                  7.0
2                                 4.0                  4.0
3                                 3.0                  4.0
4                                 1.0                  5.0
...                               ...                  ...
8758                              7.0                  7.0
8759                              4.0                  9.0
8760                              4.0                  4.0
8761                              2.0                  8.0
8762                              7.0                  4.0

      sedentary_hours_per_day  exercise_hours_per_week
0                    6.615001                 4.168189
1                    4.963459                 1.813242
2                    9.463426                 2.078353
3                    7.648981                 9.828130
4                    1.514821                 5.804299
...                       ...                      ...
```

```python
lifestyle.isnull().sum()
```

```
patient_id                      0
smoking                         0
obesity                         0
alcohol_consumption             1
diet                            1
physical_activity_days_per_week 1
sleep_hours_per_day             1
sedentary_hours_per_day         0
exercise_hours_per_week         1
dtype: int64
```

Checking null values

```python
lifestyle_new=lifestyle.dropna()
lifestyle_new
```

	patient_id	smoking	obesity	alcohol_consumption	diet	physical_activity_days_per_week	sleep_hours_per_day	sedentary_hours_per_day	exercise_hours_per_wee
0	BMW7812	True	False	False	Average	0.0	6.0	6.615001	4.16818
1	CZE1114	True	True	True	Unhealthy	1.0	7.0	4.963459	1.81324
2	BNI9906	False	False	False	Healthy	4.0	4.0	9.463426	2.07835
3	JLN3497	True	False	True	Average	3.0	4.0	7.648981	9.82813
4	GFO8847	True	True	False	Unhealthy	1.0	5.0	1.514821	5.80429
...	...	...	...	...	...	...	...	...	.
758	MSV9918	True	False	True	Healthy	7.0	7.0	10.806373	7.91734
759	QSV6764	False	True	False	Healthy	4.0	9.0	3.833038	16.55842
760	XKA5925	True	True	True	Average	4.0	4.0	2.375214	3.14843
761	EPE6801	True	False	False	Unhealthy	2.0	8.0	0.029104	3.78995
762	ZWN9666	False	False	True	Healthy	7.0	4.0	9.005234	18.08174

62 rows × 9 columns

```
lifestyle_new.isna().sum()
```

```
patient_id                        0
smoking                           0
obesity                           0
alcohol_consumption               0
diet                              0
physical_activity_days_per_week   0
sleep_hours_per_day               0
sedentary_hours_per_day           0
exercise_hours_per_week           0
dtype: int64
```

Drop and recheck the null values

Selecting the sixth table riskassesment and checking if there are any null and missing values.

```
sql = "SELECT * FROM riskassessment"
```

```
riskassessment =sqlio.read_sql_query(sql,connectpg)
print(riskassessment)
```

```
     patient_id heart_attack_risk
0       BMW7812             False
1       CZE1114             False
2       BNI9906             False
3       JLN3497             False
4       GFO8847             False
...         ...               ...
8758    MSV9918             False
8759    QSV6764             False
8760    XKA5925              True
8761    EPE6801             False
8762    ZWN9666              True
```

```
riskassessment.isnull().sum()
```

```
patient_id           0
heart_attack_risk    1
dtype: int64
```

Checking null values

```
riskassessment_new=riskassessment.dropna()
riskassessment_new
```

	patient_id	heart_attack_risk
0	BMW7812	False
1	CZE1114	False
2	BNI9906	False
3	JLN3497	False
4	GFO8847	False
...	...	...
8758	MSV9918	False
8759	QSV6764	False
8760	XKA5925	True
8761	EPE6801	False
8762	ZWN9666	True

```
riskassessment_new.isna().sum()
```

```
patient_id           0
heart_attack_risk    0
dtype: int64
```

Drop and recheck the null values

Load:

Once we have finished cleaning our data, the next step is to transfer it into PostgreSQL. We can achieve this by creating a database and table in PostgreSQL. By using the following code, we can get cleaned dataset imported effortlessly to our desktop, and then it is our job to import the cleaned csv file in the database for each tables:

```
import os
output_directory=(r"C:\Users\aidak\OneDrive\Desktop\data warehouse\group project")
os.makedirs(output_directory,exist_ok=True)
altered_tablenames= ['patients_cleaned','medical_history','health_metrics','status','life_style','risk_assessment']
dfdict={
    'patients_cleaned':patients_new,
    'medical_history':medicalhistory_new,
    'health_metrics':healthmetrics_new,
    'status':socioeconomicstatus_new,
    'life_style':lifestyle_new,
    'risk_assessment':riskassessment_new
}

for table_name in altered_tablenames:
    csv_filepath=os.path.join(output_directory,f"{table_name}.csv")
    dfdict[table_name].to_csv(csv_filepath,index=False)
```

Data loaded into desktop

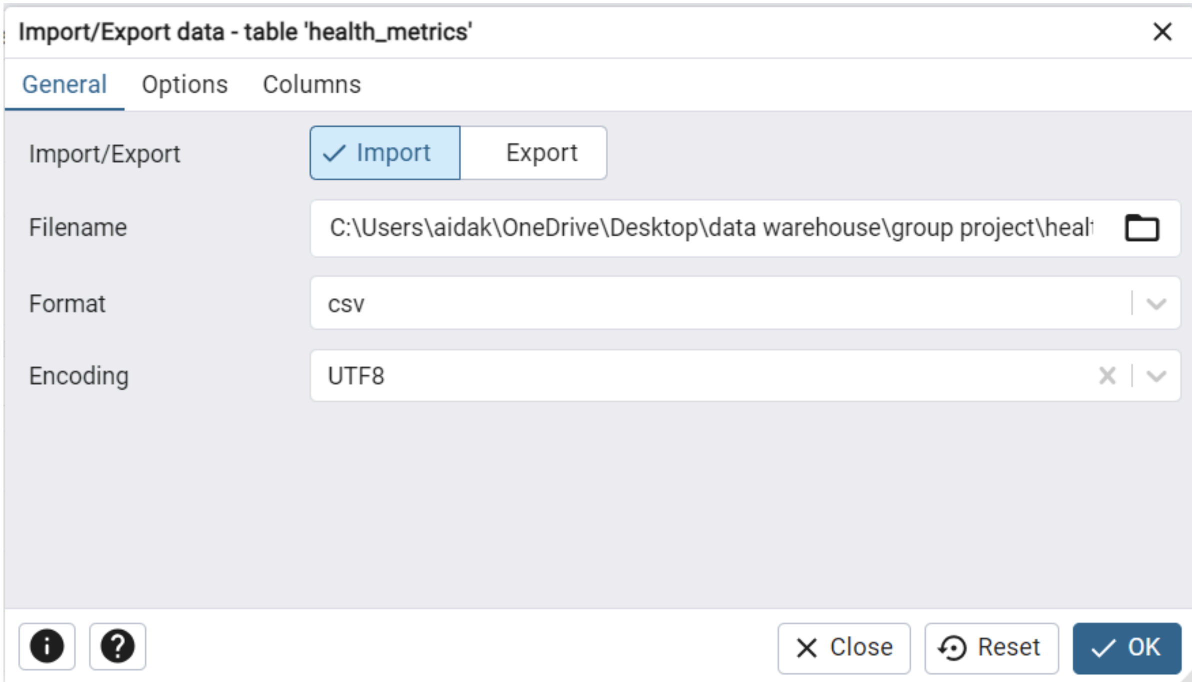
New query for cleaned tables:

Query Query History

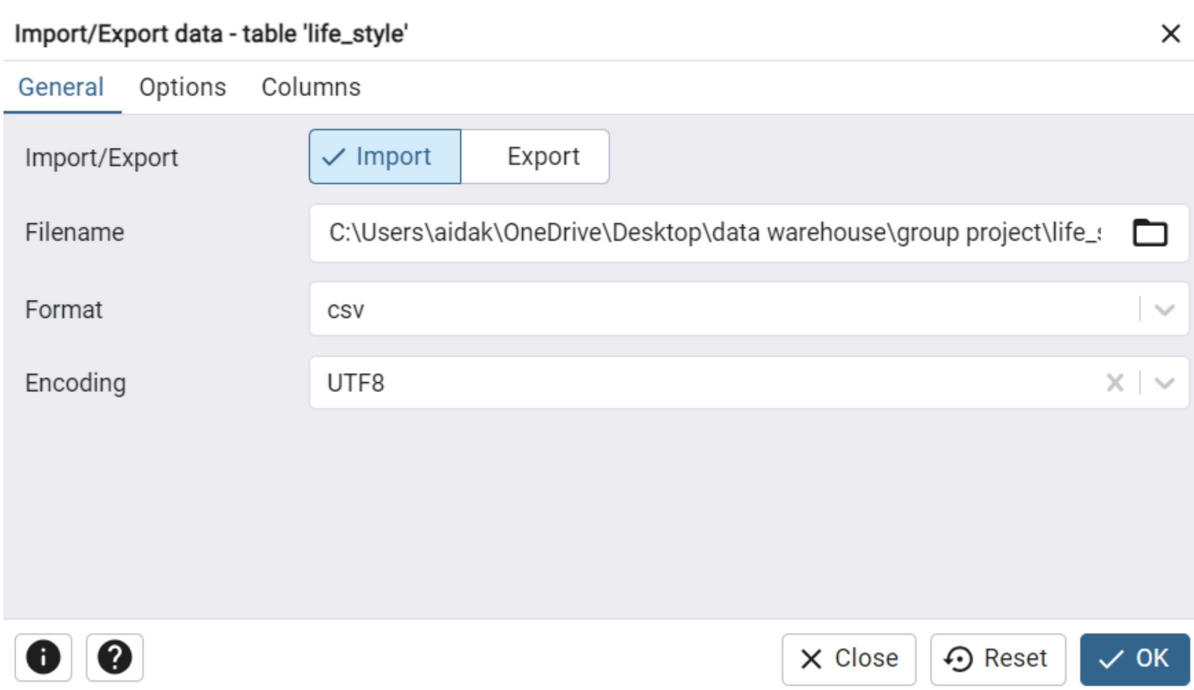
```
CREATE TABLE patients_cleaned (
    patient_id VARCHAR(10) PRIMARY KEY,
    Age INT,
    Sex VARCHAR(10),
    Family_History BOOLEAN,
    Country VARCHAR (50),
    Continent VARCHAR (50),
    Hemisphere VARCHAR (150)
);

CREATE TABLE medical_history (
    patient_id VARCHAR(10),
    diabetes BOOLEAN,
    previous_heart_problem BOOLEAN,
    medication_use BOOLEAN,
    stress_level INT,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES patients_cleaned (patient_id)
);

CREATE TABLE health_metrics(
    patient_id VARCHAR (10),
    Cholestrol INT,
    Blood_Pressure VARCHAR (10),
    Heart_Rate INT,
    Triglycerides INT,
    BMI DECIMAL (5,2),
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES patients_cleaned (patient_id)
);
```

```sql
CREATE TABLE status(
    patient_id VARCHAR(10),
    Income INT,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES patients_cleaned (patient_id)

);

CREATE TABLE life_style(
    patient_id VARCHAR (10),
    Smoking BOOLEAN,
    Obesity BOOLEAN,
    Alcohol_Consumption BOOLEAN,
    Diet VARCHAR (50),
    Physical_Activity_Days_Per_Week FLOAT,
    Sleep_Hours_Per_Day FLOAT,
    Sedentary_Hours_Per_Day FLOAT,
    Exercise_Hours_Per_Week FLOAT,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES patients_cleaned (patient_id)
);

CREATE TABLE risk_assessment(
    patient_id VARCHAR (10),
    Heart_Attack_Risk BOOLEAN,
    PRIMARY KEY (patient_id),
    FOREIGN KEY (patient_id) REFERENCES patients_cleaned (patient_id)
);
```

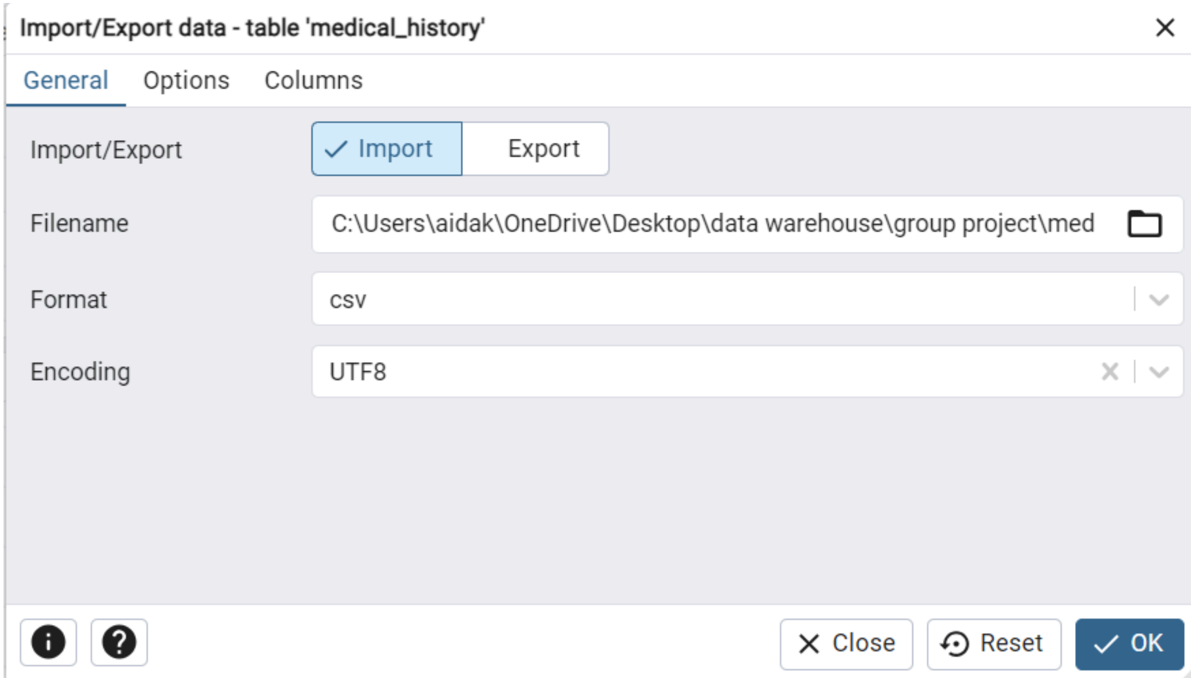
Import clean data:



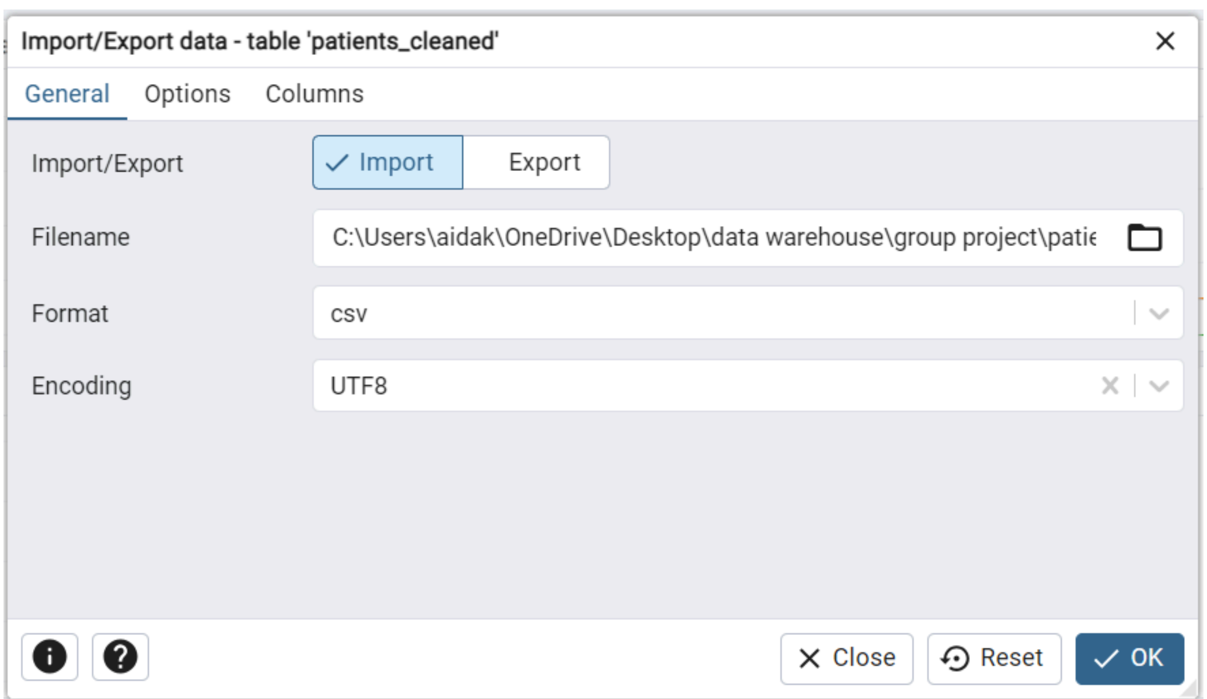
Import new csv file into table of health_metrics



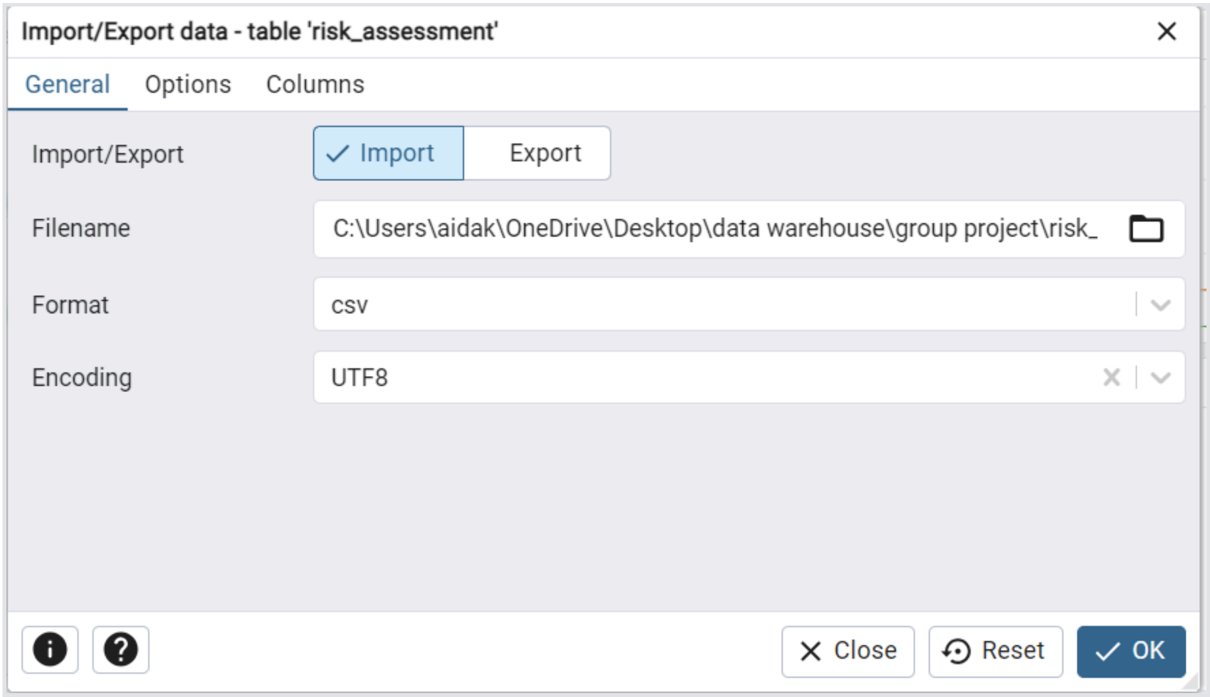
Import new csv file into table of life_style



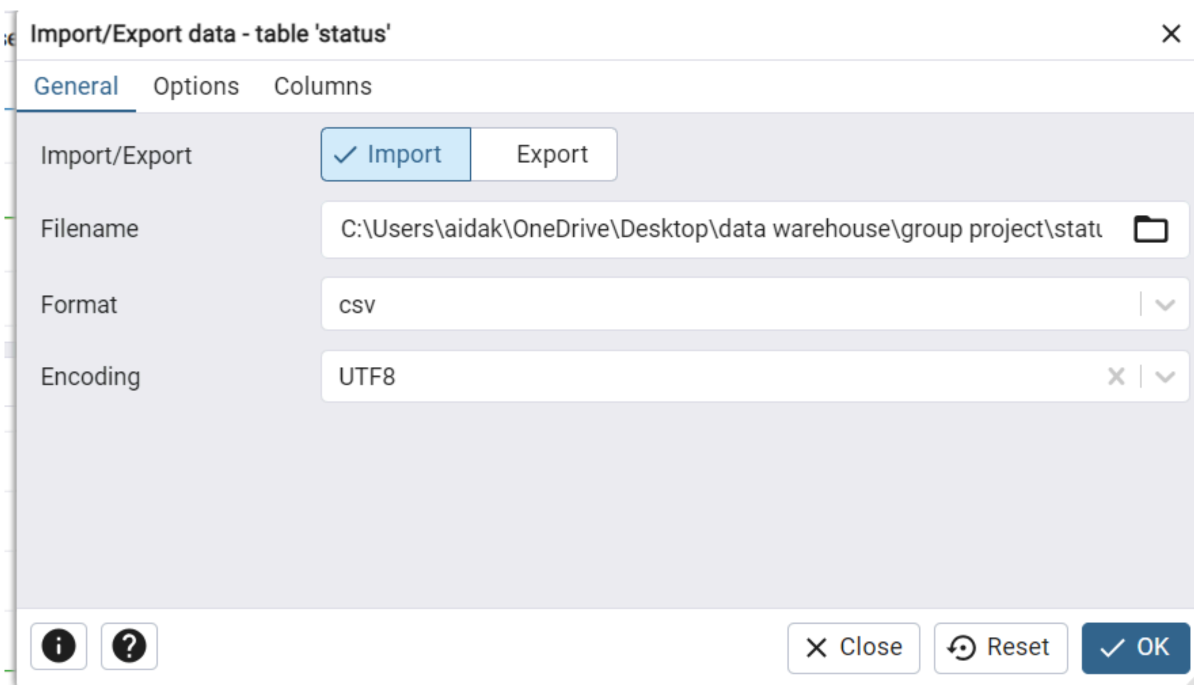
Import new csv file into table of medical_history



Import new csv file into table of patients_cleaned



Import new csv file into table of risk_assessment



Import new csv file into table of status

3.0 DATABASE

3.1 Relational Model and Relationship between Data

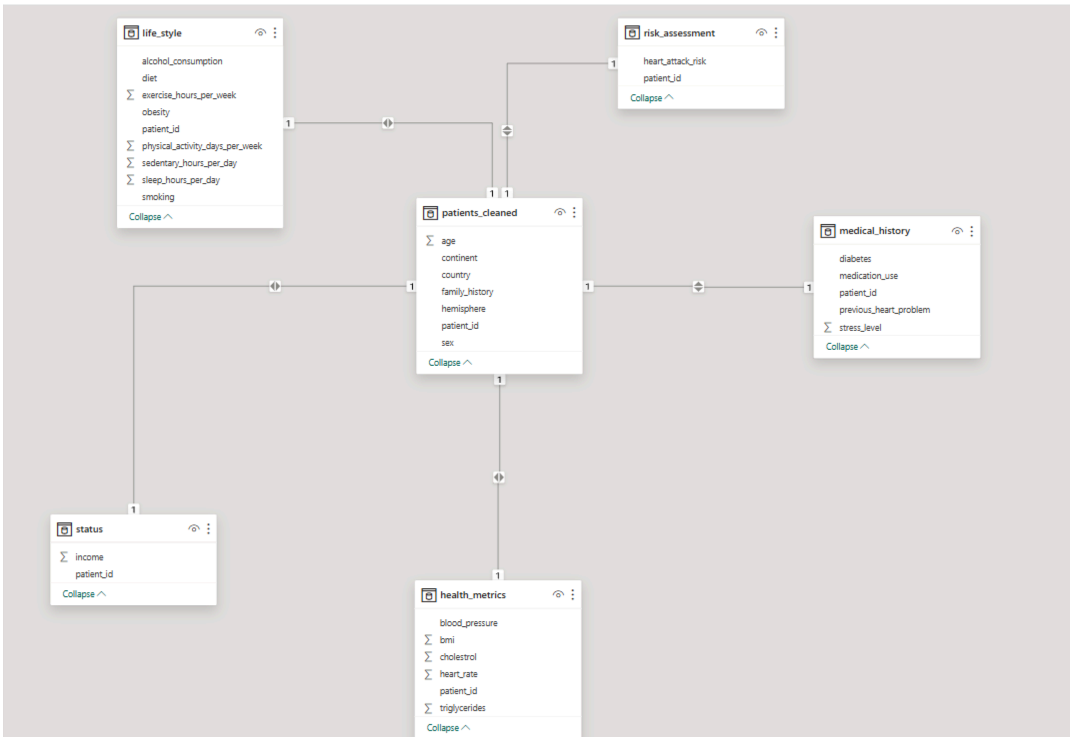


Figure 3.1 Relational Model using Power BI

3.2 Relationship between Data

DATA	RELATIONSHIP
Patients_cleaned -> risk_assessment	One to one
Patients_cleaned -> medical_history	One to one
Patients_cleaned -> health_metrics	One to one
Patients_cleaned -> life_style	One to one
Patients_cleaned -> status	One to one

3.3 Identification of Data Warehouse Schema

The data warehouse schema for these datasets is star schema, as seen in Figure 3.1 above. A star schema is characterized by a central fact table that is directly connected to multidimensional tables through primary foreign key relationships. There is a one-to-one relationship for all tables shown in Figure 3.1. But after the discussion from our group, we think that the relationship between Patients_cleaned with risk_assessment should be one-to-many relationships. But the relational model on Power BI shows the opposite. So, we decided to ignore this relationship.

4.0 RESULTS AND DATA ANALYSIS

4.1 OLAP Coding

1) Slicing Operator

```
SELECT
    p.Sex AS gender,
    AVG(h.Cholestrol) AS avg_cholesterol,
    AVG(h.BMI) AS avg_bmi
FROM
    patients_cleaned p
JOIN
    health_metrics h ON p.patient_id = h.patient_id
GROUP BY
    p.Sex;
```

Result:

Data Output Messages Notifications

	gender character varying (10)	avg_cholesterol numeric	avg_bmi numeric
1	Female	258.9426847662141780	28.9204562594268477
2	Male	260.2862520458265139	28.8792013093289689

Interpretation:

The slicing OLAP operation aggregates the average cholesterol levels and BMI for male and female patients in the dataset. For average cholesterol levels, the results show that females have an average of 258.94, while males have a slightly higher average of 260.29. The difference in cholesterol levels between genders is minimal, indicating that cholesterol may not be significantly influenced by gender in this dataset. This could suggest that cholesterol management strategies may not need to be gender-specific, but further research could confirm if other factors play a more significant role in these levels.

When considering average BMI, females have an average BMI of 28.92, while males have a similar average of 28.88. Both genders have BMI values that are above the healthy range of 18.5-24.9, classifying them as overweight. The near-identical BMI values across genders suggest that weight management and interventions to prevent obesity may be equally relevant for both males and females. Given the association between high BMI and cardiovascular disease, both genders would benefit from programs focusing on weight management, physical activity, and healthier eating habits.

This OLAP slicing operation helps provide a clear comparison of gender differences in key health metrics like cholesterol and BMI. The findings imply that while the averages are similar, there is a need for continued focus on both cholesterol and weight management across both genders to reduce the risk of cardiovascular diseases.

2) Dicing Operator

```
SELECT
    p.patient_id,
    p.family_history,
    m.diabetes,
     m.previous_heart_problem,
    m.stress_level,
    r.Heart_Attack_Risk
FROM patients_cleaned p
JOIN medical_history m ON p.patient_id = m.patient_id
JOIN risk_assessment r ON p.patient_id = r.patient_id;
```

Result:

	patient_id character varying (10)	family_history boolean	diabetes boolean	previous_heart_problem boolean	stress_level integer	heart_attack_risk boolean
1	BMW7812	false	false	false	9	false
2	CZE1114	true	true	true	1	false
3	BNI9906	false	true	true	9	false
4	JLN3497	true	true	true	9	false
5	GFO8847	true	true	true	6	false
6	ZOO7941	true	true	true	2	true
7	WYV0966	false	false	false	7	true
8	XXM0972	false	false	false	4	true
9	XCQ5937	false	true	false	5	false
10	FTJ5456	true	false	false	4	false
11	HSD6283	true	true	true	8	false
12	YSP0073	true	true	false	4	false
13	FPS0415	true	true	false	9	true
14	YYU9565	true	true	true	1	true
15	VTW9069	true	true	false	2	false
16	DCY3282	true	true	false	5	true

Interpretation:

The dicing operator includes 16 patients with information on family history, diabetes, previous heart problems, stress levels, and heart attack risk status. Out of these, five patients are marked as being at risk of a heart attack, including some like ZOO7941 who have all three major medical risk factors but low stress, and others like WYV0966 and XXM0972 who have no medical history but moderate stress levels. However, patients JLN3497, GFO8847, and HSD6283 have high stress and don't have heart attack risk. Based on the results, these patients are suggested to look at other factors such as cholesterol or lifestyle that may influence the heart attack since they had heart problems previously. In addition, high stress does not indicate the heart attack risk, as seen with BMW7812, who has a stress level of 9 but no medical conditions. Overall, the data indicates that heart attack risk is not based on previous heart problems or stress but likely involves a more complex check-up.

3) Roll Up Operator

```
SELECT
        p.Continent,
        p.Country,
        p.Sex,
        ROUND(AVG(h.BMI)::numeric, 2) AS Avg_BMI,
        ROUND(AVG(h.Heart_Rate)::numeric, 2) AS Avg_Heart_Rate
        FROM patients_cleaned p
        JOIN health_metrics h ON p.patient_id = h.patient_id
        GROUP BY
                ROLLUP (p.Continent, p.Country, p.Sex);
```

Result:

	continent character varying (50)	country character varying (50)	sex character varying (10)	avg_bmi numeric	avg_heart_rate numeric
1	[null]	[null]	[null]	28.89	75.03
2	South America	Colombia	Female	29.25	76.08
3	South America	Argentina	Male	28.50	74.91
4	Asia	Thailand	Male	29.02	74.80
5	Africa	South Africa	Female	28.85	75.22
6	Australia	New Zealand	Female	29.23	77.88
7	Europe	Germany	Female	29.08	79.11
8	Europe	Spain	Male	28.72	73.10
9	South America	Argentina	Female	29.38	75.70
10	North America	Canada	Male	28.88	75.33
11	South America	Brazil	Female	29.01	74.29
12	Asia	Vietnam	Female	28.33	74.84
13	Asia	South Korea	Male	29.46	73.65
14	Asia	Thailand	Female	28.36	74.52
15	Asia	China	Male	28.84	75.43
16	Africa	South Africa	Male	29.17	77.73
17	Australia	New Zealand	Male	28.59	75.23
18	Australia	Australia	Female	29.07	74.77
19	Asia	Japan	Female	28.41	72.95
20	Asia	India	Male	29.13	74.46

Interpretation:

By using the rollup OLAP operator, the output reveals that the average BMI across the continent be it female or male has an average of 28 to 29 BMI and an average of 73 to 79 heart rate. This operator helps to compare health metrics across different regions and demographics like gender. For instance, we can see which country has the highest or lowest average BMI or heart rate. This can help to observe and investigate areas that need more health attention and care.

4) Drill Down

```
SELECT
    p.Country,
    p.Sex,
    r.Heart_Attack_Risk AS risk_level,
```

```
    COUNT(*) AS PatientCount
FROM
    patients_cleaned p
JOIN
    risk_assessment r ON p.patient_id = r.patient_id
GROUP BY
    p.Country, p.Sex, r.Heart_Attack_Risk
ORDER BY
    p.Country, r.Heart_Attack_Risk, PatientCount DESC;
```

Output:

	country character varying (50)	sex character varying (10)	risk_level boolean	patientcount bigint
1	Argentina	Male	false	194
2	Argentina	Female	false	103
3	Argentina	Male	true	119
4	Argentina	Female	true	55
5	Australia	Male	false	204
6	Australia	Female	false	77
7	Australia	Male	true	112
8	Australia	Female	true	56
9	Brazil	Male	false	209
10	Brazil	Female	false	90
11	Brazil	Male	true	118
12	Brazil	Female	true	45
13	Canada	Male	false	197
14	Canada	Female	false	85
15	Canada	Male	true	108
16	Canada	Female	true	50

Interpretation:

By using the drill down OLAP operator, the output represents 16 rows of the count of patient heart attack risk across the country, sex and risk level. This table shows that the patient that has the highest risk level comes from Argentina. We can see that male patients consistently have a higher risk level which is 119 patients compared to female patients which is 45 patients . Followed by Brazil also being the second highest which is 118 individuals of male that have heart attack risk. For instance, we can see which countries have the highest heart attack risk.

5) Pivot

```
CREATE EXTENSION IF NOT EXISTS tablefunc;

SELECT * FROM crosstab(
  $$
  SELECT p.Country,
         CASE WHEN r.Heart_Attack_Risk THEN 'At Risk' ELSE 'Not at Risk' END AS
risk_status,
         COUNT(*)
  FROM patients_cleaned p
  JOIN risk_assessment r ON p.patient_id = r.patient_id
  GROUP BY p.Country, risk_status
  ORDER BY p.Country, risk_status
  $$,
  $$
  SELECT unnest(ARRAY['At Risk', 'Not at Risk'])
  $$
) AS pivot_table(
  Country VARCHAR,
  "At Risk" INT,
  "Not at Risk" INT
);
```

	country character varying	At Risk integer	Not at Risk integer
1	Argentina	174	297
2	Australia	168	281
3	Brazil	163	299
4	Canada	158	282
5	China	155	281
6	Colombia	162	267
7	France	157	289
8	Germany	171	305
9	India	129	283
10	Italy	136	295
11	Japan	144	289
12	New Zealand	151	284
13	Nigeria	178	270
14	South Africa	144	281
15	South Korea	163	246
16	Spain	150	280
17	Thailand	161	267
18	United Kingdom	160	297
19	United States	166	254
20	Vietnam	148	277

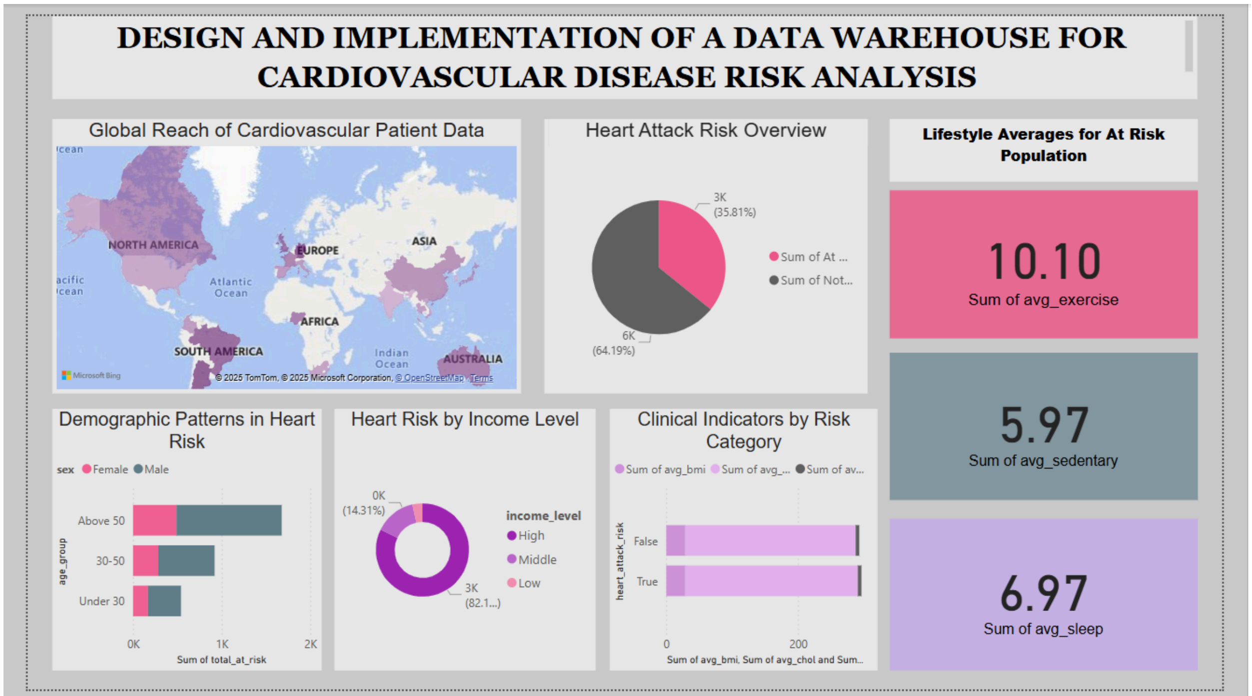
Figure 3.5 Pivot

Interpretation:

This table shows 20 various countries and is divided into two categories which is "At Risk" and "Not at Risk" . Nigeria has the most persons at risk (178), followed by Argentina (174) and Germany (171). However, India has the fewest people at risk, which means that Indians have better living conditions or fewer issues that have been recorded. In terms of the number of people who are not in health attack risk, South Korea has the highest heart attack risk with 246, while Germany has the lowest heart attack risk which is 305. It's very important to find out that Nigeria stands out as the highest number of people at risk of heart attack, due to a high population in that country of those at risk and low population in the country that is not at risk, which may indicate that its citizens confront more difficulties. On the other hand, nations such as Germany and the United Kingdom displayed high counts in both categories, which may simply be the result of more extensive data collecting or larger populations. All things considered, this table provides us with an overview of how various nations are faring in terms of danger and aids in identifying areas that may require additional focus or assistance.

4.2 DATA VISUALIZATION

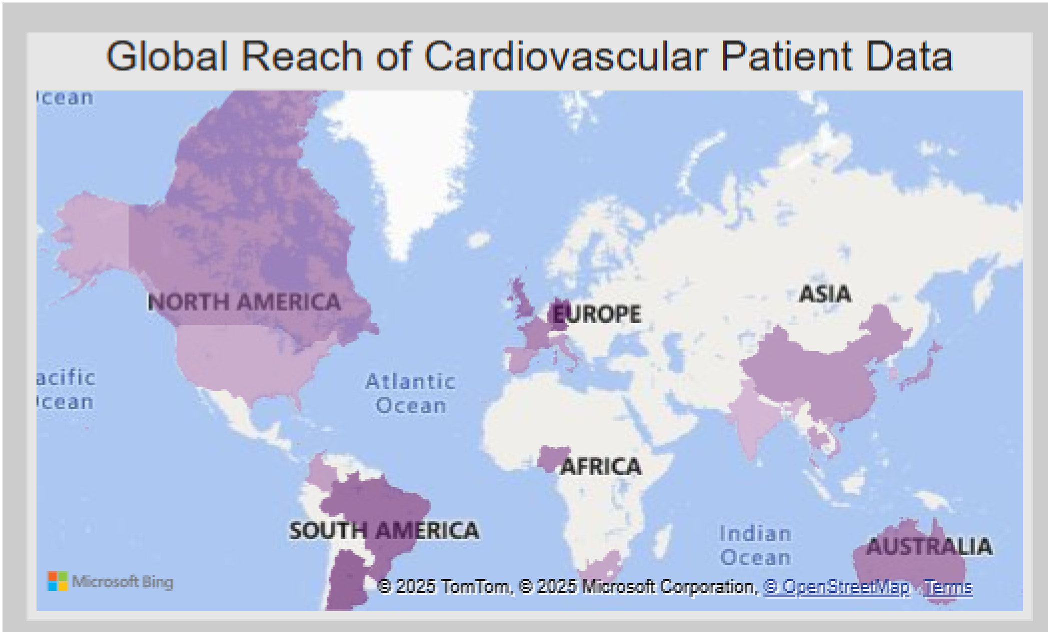
4.2.1 Dashboard



The dashboard above shows the cardiovascular risk analysis with multiple visualizations combined together indicating the key demographic, lifestyle and socioeconomic factors. It highlights the pattern in the heart attack risk based on the factors mentioned, like gender, income level, age, geographical areas and health behaviours. The visualisations in the dashboard will be interpreted in depth below.

4.2.2 Visualisation

1) Global Reach of Cardiovascular Patient Data



Interpretation:

The map provides insights into how different regions are impacted by cardiovascular diseases (CVDs) by visualising the global distribution of CVD patient data. It shows that North America, Europe, and some regions of Asia have a greater concentration of cardiovascular patients. The greater number of data points in these areas points to a higher prevalence of cardiovascular problems, which are probably caused by lifestyle choices like stress, smoking, and diet.

In North America, rising obesity, high-calorie diets, and relentless schedules feed problems such as high blood pressure and diabetes. Doctors report these cases widely, so the region appears brighter on the map than it might be if records were patchy.

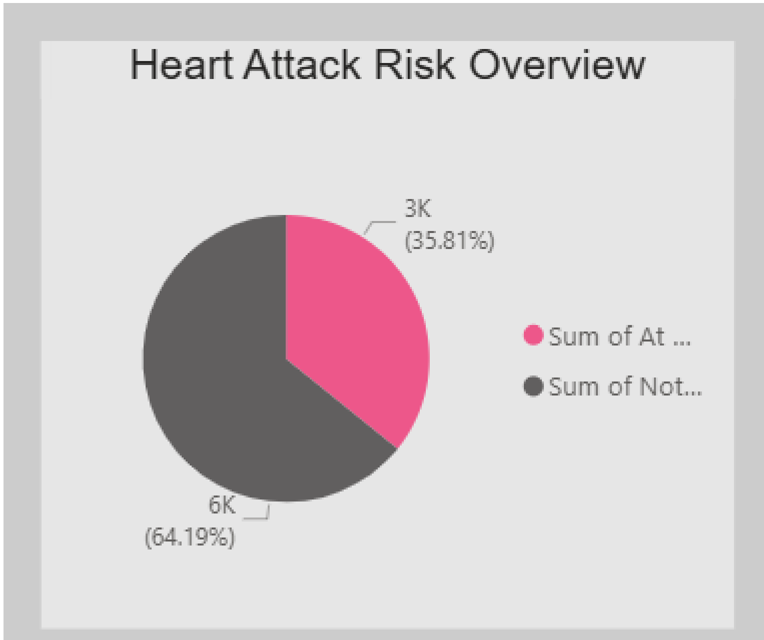
Europe paints a similar picture, though causes vary by country. Many people still smoke or skip exercise, but strong public health systems mean symptoms are tracked and treated early. That thorough record-keeping, along with full clinics, pushes patient numbers up on the continent even as prevention efforts slowly curb growth.

Asia is quickly becoming a major focus for heart-health experts watching where cardiovascular risks are climbing fastest. Once mostly protected by home-cooked meals, many cities now lean on packaged, West-influenced foods that pile on extra fat and sugar. As scales tip upward and cholesterol scores creep higher, country after country has begun reporting troubling upward trends, pushing researchers and planners to act.

South America and Australia appear quieter on the patient front, yet statistics remind us they still deserve a seat at the table. In South America, limited clinics and uneven transport may mask the true number of heart cases, leaving deaths uncounted and risk hidden. Australia, already healthier than North America, shows moderate pressures-mostly in big-city dwellers who drive, work, and relax without moving much.

In conclusion, understanding the global distribution of cardiovascular disease and its risk factors across regions highlights the importance of targeted health interventions and policies that address the unique needs of each region, ultimately contributing to improved cardiovascular health outcomes worldwide.

2) Heart Attack Risk Overview



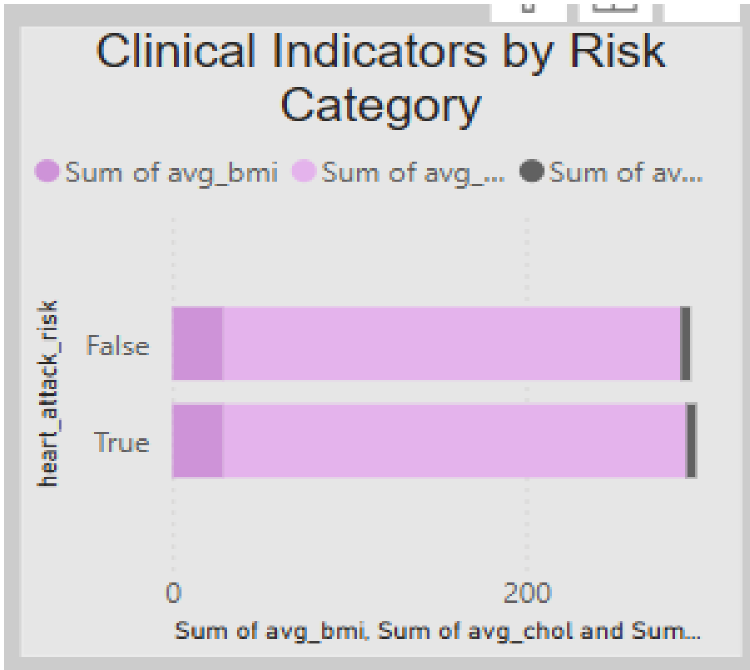
Interpretation:

This pie chart displayed a graphic visualization of how people are grouped according to their risk of having a heart attack. This graphic shows that 3,000 people, or around 35.81%, are having risk for heart attack, meanwhile the remaining 6,000 people, or 64.19%, are shown as not at risk. According to this visualization, more than one third of the pie chart, which means 3,000 people are facing health issues that lead to cardiovascular disease. The risk that has been faced for a large population is significant and may prove hidden problems including unhealthy lifestyle choices, lack of education, or restricted access to medical care.

This figure highlights how important it is in focusing initiatives to give public health awareness to the population. It might be necessary for people that have heart attack risk to establish a regular check-up that supports their healthy lifestyle choices and early detection for the cardiovascular. These initiatives will help to lower the potential for heart attack in that population by promoting regular cardiovascular screenings, healthy diets, and physical activity.

Overall, the significant population of people in the risk phase should not be ignored, even though the majority of those in the current figures are low-risk, since this suggests the importance of taking preventative initiative to reduce the potential rise in heart-related illnesses.

3) Clinical Indicators by Risk Category



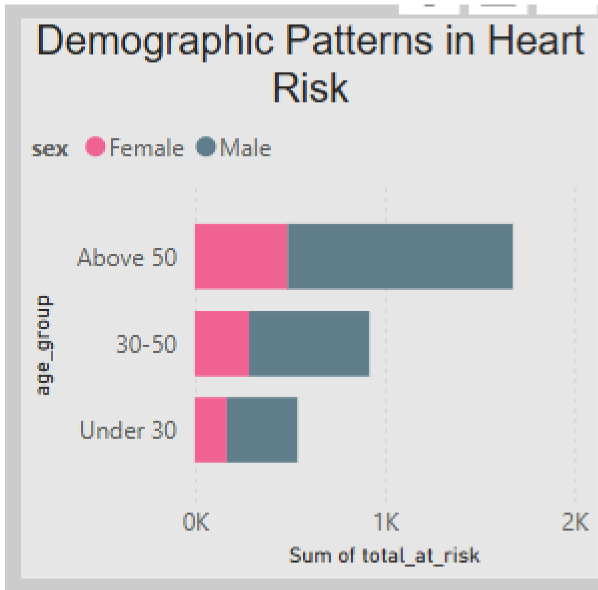
Interpretation:

This "Clinical Indicators by Risk Category" graphic shows the sum of the values of three important clinical variables which is average BMI, average cholesterol, and average stress level. These values are grouped according to whether a person has a heart attack risk (False) or not (True) which is represented by each horizontal bar and the variables divided into three color segments that represent orange for stress, dark blue for cholesterol, and light blue for BMI.

The graphic shows that the overall sums of the clinical variables for the two heart attack risk groups are nearly comparable. However, the most to the overall value is cholesterol and the bar chart shows people who have heart attack risk are the one that have the higher levels of total cholesterol than people who are not at risk. These results displayed the possibility between high cholesterol and the risk of having a heart attack are high. On the other hand, average BMI and stress levels are relatively stable between the two groups and have less effect in this chart. The orange portion, which stands for stress, indicates particularly low and stable with the stress levels.

In conclusion, cholesterol is the most important factor that can lead to heart attack in this case, even if all three clinical variables are present. Average of stress level and BMI appear to have less of an effect because of their relatively small and distributed equally that effects in both groups.

4) Demographic Patterns in Heart Risk



The demographic patterns in heart risk represents the heart attack risk from a demographic perspective, focusing on two main factors which is age and sex. Divided into 3 age groups which are under 30, 30 to 50, and 50 and above, and then split by gender - male and female.

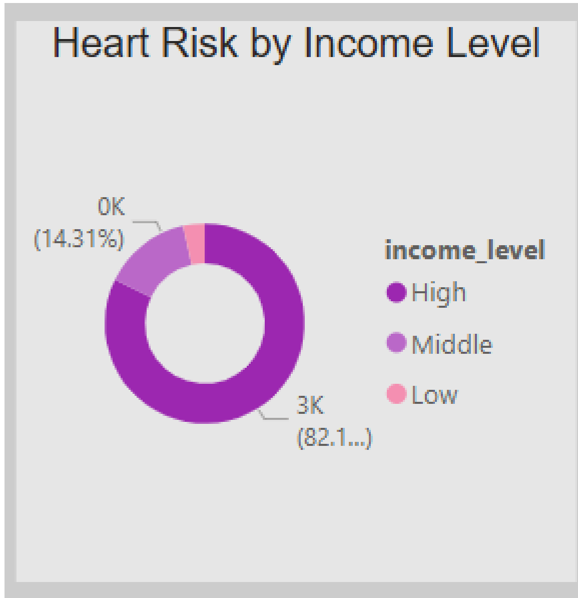
The most leading insight is the risk of heart attack increases significantly with age. The age group of 50 and above shows the highest number of individuals at risk, this shows that aging is a strong risk factor for cardiovascular disease. As people age, the likelihood of chronic diseases such as diabetes, hypertension and high cholesterol all contribute to heart attacks.

Additionally, the chart represents a consistent gender gap across all age groups, with males showing the highest heart risk than females for each category. This may be due to lifestyle differences. For example, men may be more likely to engage in high-risk behaviors such as smoking, heavy drinking, or an unbalanced diet. In addition, women are less likely than men to have it because women, mostly because of perimenopause, tend to have natural protection against heart disease due to estrogen.

The age group 30-50 shows a moderate level of risk, acting as a warning zone where interventions such as routine health check-ups, dietary control and stress management can be very effective. The under 30 age group, although showing a small number, still shows the possibility of having a heart risk. This shows that young people are not immune. This may be due to lifestyle issues, genetic predisposition or extreme stress levels.

In summary, the chart emphasizes the importance of targeted health education and prevention programs, especially for men aged 30 and over, with increased screening for those over 50. It supports that age and gender are critical predictors in the fight against heart disease and should guide both public health messaging and personal lifestyle choices.

5) Heart Risk by Income Level

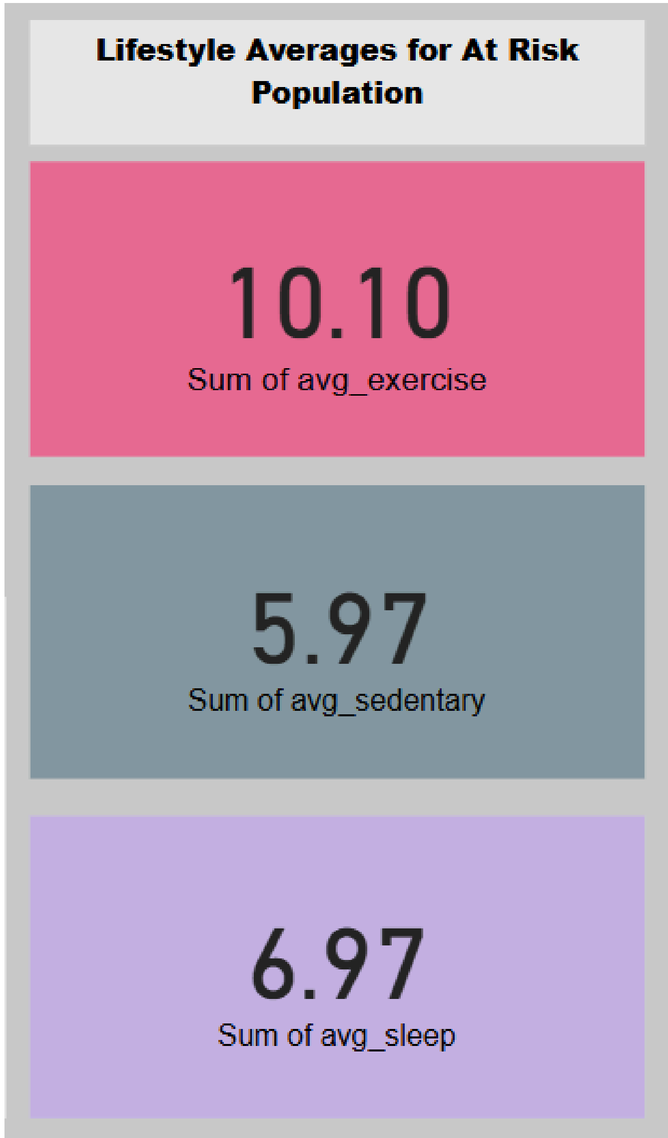


Interpretation:

The Heart Risk by Income Level doughnut chart illustrates that there are people with a spread of three types of earning which are Low, Middle and High are facing heart attack risks. As shown above, there are 3.51% of at-risk people who belong to the Low-income class, 14.31% to the Middle-income class and 82.19% to the High-income class.

At first glance, the graph appears to show that most people flagged as being at risk for heart attacks belong to the upper-income bracket. That observation might lead someone to argue that having more money does not automatically guarantee good heart health, since wealthier individuals still face stress, desk-bound careers, and questionable eating habits. Yet readers should tread carefully, the image presents only raw head counts for each income tier, omitting the crucial detail of what fraction of each bracket is actually at risk. If the high-income cohort is simply much larger, the high-risk tally could mirror the population distribution rather than true elevated danger. For that reason, a deeper inquiry should calculate and compare the percentage of at-risk persons within each income group. Only then can researchers judge whether income itself drives heart attack risk, or whether the apparent link is nothing more than a numbers game.

6) Lifestyle Averages for At Risk Population



Interpretation:

The at-risk population's average lifestyle reveals both healthy habits and some areas for concern. With an average exercise level of 10.10, people in this group are exhibiting a comparatively high degree of physical activity. Frequent exercise is essential for preserving cardiovascular health since it lowers the risk of diseases including high blood pressure and high cholesterol, improves heart function, and aids in weight management. Although this number indicates good habits, knowing the precise measurement such as the number of hours per week would help to clarify whether exercise is sufficient.

Nonetheless, the group's average sedentary behaviour is 5.97, suggesting that a sizable amount of time is still spent idle. One well-known risk factor for cardiovascular illnesses is prolonged sedentary behaviour, such as sitting for lengthy periods of time. This emphasises the necessity of promoting greater exercise throughout the day in order to decrease sedentary time, which can help lower the risk of heart disease.

The average sleep length of 6.97 hours is little less than the 7-9 hours of sleep that people are advised to have each night. Although this indicates that the majority of people are receiving enough sleep, the somewhat lower average sleep suggests that there may be room for improvement. A number of health problems, such as elevated blood pressure and stress levels, which might raise cardiovascular risk, are associated with inadequate sleep. Enhancing the length and quality of sleep may aid in lowering these risks even more.

In conclusion, moderate sedentary behaviour and inadequate sleep still pose serious dangers, even though the at-risk group exhibits healthy behaviours like regular physical activity. Improved sleep habits and increased physical activity could significantly improve cardiovascular health and lower the population's overall risk of heart disease.

5.0 CONCLUSION

According to Cardiovascular Disease Risk data, we know that certain populations are unbalanced affected by heart attack health issues. In this study, we can see that the older adults have a higher prevalence of heart attack risk. In contrast, the lowest heart risk level comes from the younger and female patients. For individuals with higher cholesterol, BMI and stress levels will be categorized as high risk levels to get a heart attack. Not only that, clinical indicators also will be combined with unhealthy lifestyle habits such as low physical activity and poor sleep that contribute to heart disease risk to individuals. We see that the populations with lower income will be face burned out so that socioeconomic factors must play a crucial role in these health outcomes. This study needs to be emphasized to achieve the targeted levels of health, by increasing awareness and equitable access to preventative care to reduce cardiovascular risk and promote the overall well-being of the population.

6.0 REFERENCES

Cleveland Clinic. (2022, September 1). Cardiovascular disease. Cleveland Clinic. Retrieved June 16, 2025, from https://my.clevelandclinic.org/health/diseases/21493-cardiovascular-disease

W3Schools.com. (n.d.). https://www.w3schools.com/postgresql/postgresql_pgadmin4.php

Byron Neji. (2023, June 22). *Connecting PostgreSQL with Jupyter Notebook* [Video]. YouTube. https://www.youtube.com/watch?v=le_GyH6kZTo

Website, N. (2025, April 22). *Cardiovascular disease*. nhs.uk. https://www.nhs.uk/conditions/cardiovascular-disease/#:~:text=Cardiovascular%20disease%20(CVD)%20is%20a,increased%20risk%20of%20blood%20clots.

GeeksforGeeks. (2025, March 27). *ETL process in data warehouse*. GeeksforGeeks. https://www.geeksforgeeks.org/dbms/etl-process-in-data-warehouse/

Neji, B. (2023, June 22). *Connecting PostgreSQL with Jupyter Notebook* [Video]. YouTube. https://www.youtube.com/watch?v=le_GyH6kZTo