PHASE 4 DOCUMENTATION

Date	26-10-2023
Team ID	4162
Project Name	Data Warehousing with IBM Cloud Db2 Warehouse

Table of Contents

1. Introduction

This document provides an overview of the Extract, Transform, Load (ETL) process for the Diabetes dataset. The objective is to extract data from a CSV file, transform it, and load it into a MySQL database for further analysis. Diabetes is a prevalent chronic health condition, and managing and analyzing the data related to it can be instrumental in healthcare research, treatment, and decision-making. This project facilitates the conversion of raw diabetes data into a structured, query able format, thus enabling healthcare professionals and data analysts to gain valuable insights from the data.

2. Problem Statement

The project seeks to address the critical problem of disorganized, unstructured diabetes data, hindering effective healthcare and research efforts. Inconsistent data sources, poor data quality, and a lack of centralized storage make it challenging for healthcare professionals and researchers to access, analyze, and derive insights from diabetes-related data. The project's goal is to create an ETL pipeline to extract, transform, and load this data into a MySQL database, thus providing a reliable, organized, and accessible repository for diabetes information. By doing so, we aim to enable informed decision-making, advanced research, and improved diabetes management and public health outcomes.

3. ETL Process:

The ETL (Extract, Transform, Load) process is a fundamental data integration process used to collect, clean, transform, and load data from various sources into a target data repository, such as a database, data warehouse, or data lake.

3.1 Extract:

Data extraction is the first step in the ETL (Extract, Transform, Load) process, where data is collected or retrieved from one or more source systems for further processing. In your specific project of loading a diabetes CSV file into a MySQL database, data extraction involves obtaining the diabetes data from the CSV file and loading it into a Python environment for further transformation and loading into the database.

```
import pandas as pd
df = pd.read_csv('diabetes_dataset.csv')
```

This code reads the CSV file and stores the data in the df DataFrame.

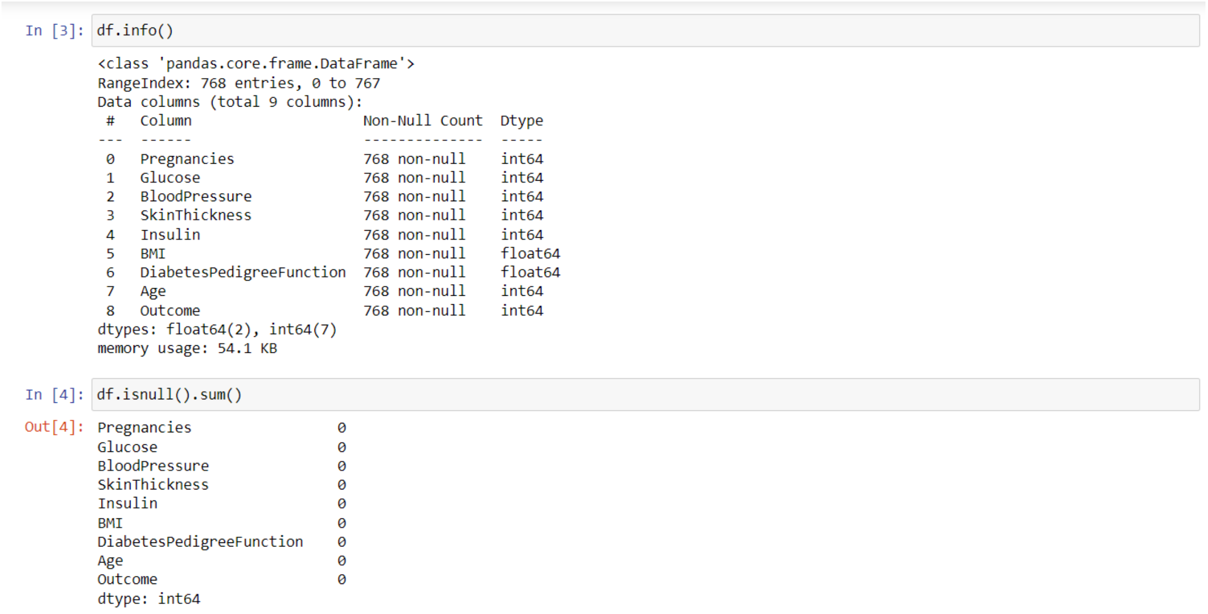
To use Python's pandas library to load the data from the CSV file. The pd.read_csv() function is a common method for reading data from CSV files into a Pandas DataFrame, which is a tabular data structure. The CSV file is typically located in your local directory or at a specified file path.

3.2 Transform:

The data transformation step in the ETL (Extract, Transform, Load) process is crucial for preparing the raw data extracted from the source (in this case, a diabetes CSV file) for loading into a MySQL database. Transformation involves cleaning, structuring, and enriching the data to ensure it is in the right format and quality for its intended use.

- Dropping duplicate values
- Checking NULL values
- Checking for 0 value and replacing it :- It isn't medically possible for some data record to have 0 value such as Blood Pressure or Glucose levels. Hence we replace them with the mean value of that particular column.

```
df.info()
df.isnull().sum()
```



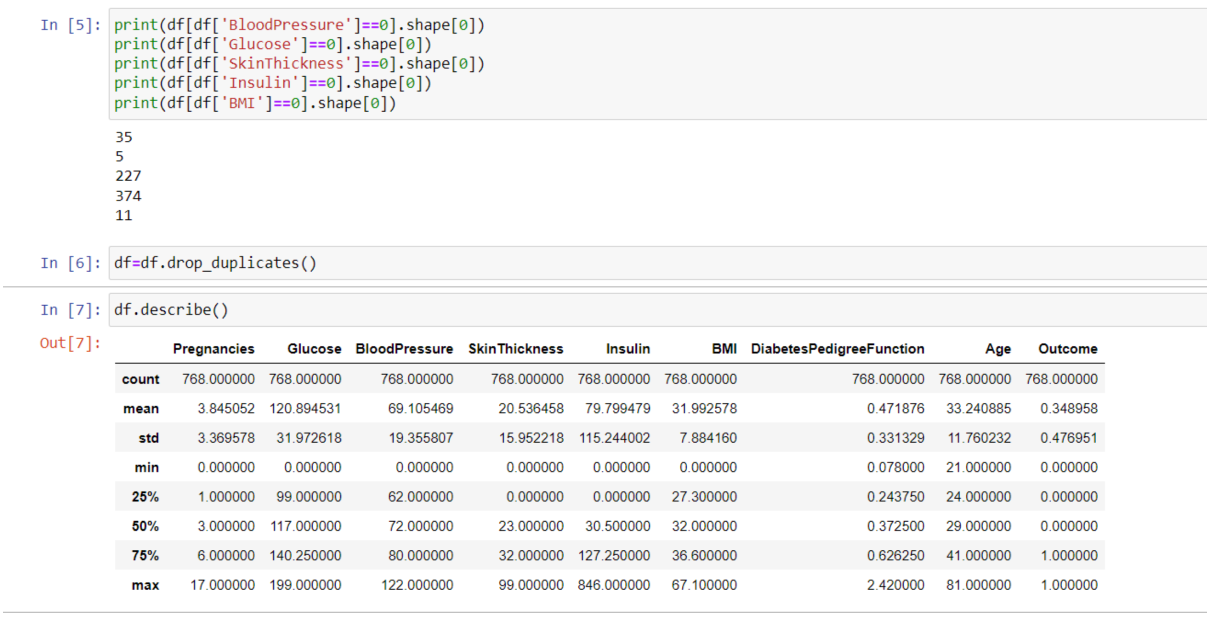
```
In [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB

In [4]: df.isnull().sum()

Out[4]: Pregnancies                 0
        Glucose                     0
        BloodPressure               0
        SkinThickness               0
        Insulin                     0
        BMI                         0
        DiabetesPedigreeFunction    0
        Age                         0
        Outcome                     0
        dtype: int64
```

```
print(df[df['BloodPressure']==0].shape[0])
print(df[df['Glucose']==0].shape[0])
print(df[df['SkinThickness']==0].shape[0])
print(df[df['Insulin']==0].shape[0])
print(df[df['BMI']==0].shape[0])
df=df.drop_duplicates()
df.describe()
```



```
In [5]: print(df[df['BloodPressure']==0].shape[0])
        print(df[df['Glucose']==0].shape[0])
        print(df[df['SkinThickness']==0].shape[0])
        print(df[df['Insulin']==0].shape[0])
        print(df[df['BMI']==0].shape[0])

        35
        5
        227
        374
        11

In [6]: df=df.drop_duplicates()

In [7]: df.describe()

Out[7]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

3.3 Load:

In the ETL (Extract, Transform, Load) process, the "Load" step is the final phase where the transformed data is loaded into the target destination, which is typically a database. In the project of extracting and transforming diabetes data from a CSV file, this step involves loading the cleaned and structured data into a MySQL database. Here's a detailed explanation of the data loading step:

To load data into a MySQL database, First need to establish a connection to the MySQL server. To require necessary credentials to access the database server.

```
import mysql.connector
conn = mysql.connector.connect(
    host='local',
    user='root',
    password='pass_word',
    database='diabetes_data'
)

cursor = conn.cursor()
```

To define the structure of the table in the MySQL database where the data will be stored. This structure should match the schema of the transformed data.

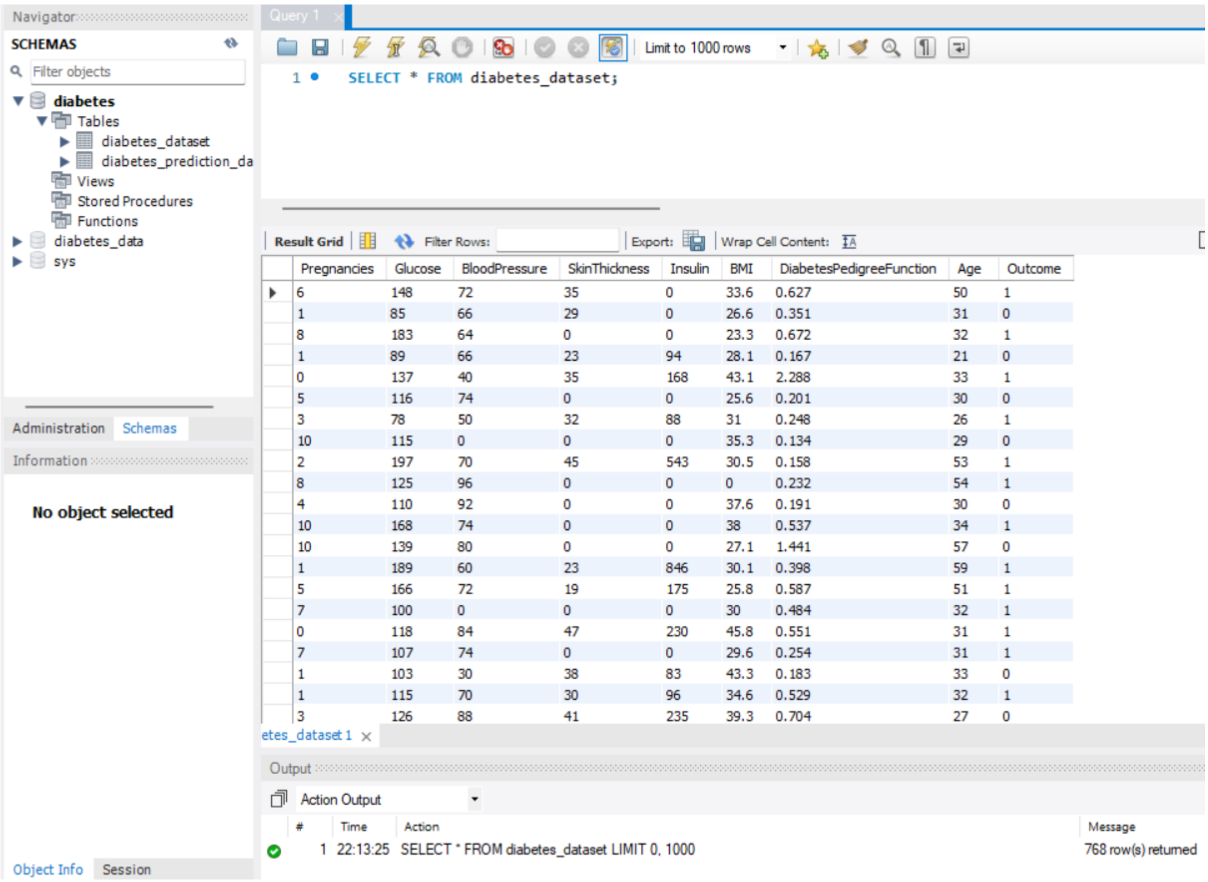
```
create_table_query = "CREATE TABLE diabetes_pred (Pregnancies INT, Glucose
INT, BloodPressure INT, SkinThickness INT, Insulin INT, BMI float,
DiabetesPedigreeFunction float, Age INT, Outcome BINARY);"

cursor.execute(create_table_query)

for index, row in df.iterrows():
    cursor.execute("INSERT INTO diabetes_pred (Pregnancies, Glucose,
BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age,
Outcome) VALUES (%s, %s, %s, %s,%s, %s,%s, %s,%s);"
    ,(row['Pregnancies'], row['Glucose'], row['BloodPressure'],
row['SkinThickness'], row['Insulin'], row['BMI'],
row['DiabetesPedigreeFunction'], row['Age'], row['Outcome']))

conn.commit()
conn.close()
```

The "Load" step completes the ETL process by moving the transformed data from your Python environment into a MySQL database, making it accessible for querying and analysis within the database system. This step ensures that the data is structured, organized, and stored in a way that allows for efficient retrieval and analysis.



```
SELECT * FROM diabetes_dataset;
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0

#	Time	Action	Message
1	22:13:25	SELECT * FROM diabetes_dataset LIMIT 0, 1000	768 row(s) returned

4.CONCLUSION:

In conclusion, this ETL project successfully extracted, transformed, and loaded diabetes data from a CSV file into a MySQL database. The extraction process retrieved the raw data from the CSV file, while the transformation step cleaned, enriched, and restructured the data for analysis. Finally, the loading step facilitated the insertion of the transformed data into a MySQL database. This project ensures that the data is now efficiently stored in a structured format, ready for further analysis, reporting, and decision-making. It showcases the power of ETL processes in preparing data for meaningful insights and demonstrates the importance of data quality and consistency for accurate analysis in the context of diabetes research or healthcare analytics.