# ***How to read a dataset and write in a SQL Database: Using MySql And Python (Glass Dataset)***

Working with a database is easy but when you need to deal with the clients, you might need to know about the insertion of data that you will get from you clients and handle the database.

Here I am trying to read Glass Dataset & implement as follows:

1. Connecting the database with MySQL and Python

2. Creating A Database

3. Creating new table to store the data

4. Taking the dataset from the local machine and write it down inside a database

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

**1. Connecting the database with MySQL and python:**

· First, we need to install ““mysql-connector-python”” package so that we can create connection with Mysql.

**pip install mysql-connector-python**

·  After installing the package, we have to establish the connection.

**import mysql.connector**

If the above code runs without any errors, then you have successfully installed mysql.connector, and it is ready to use.

· Now, connect to the database using your username and password.

**import pandas as pandas**

**import csv**

**mydb = connection.connect(host="localhost", user="root", passwd="password",use\_pure=True)**

**#check if the connection is established**

**print(mydb.is\_connected())**

To check if the connection is established, we can use print (mydb.is\_connected). It will return **TRUE** if the connection is established else **FALSE**.

### 2. Creating a Database:

·Now, we will create a database with the name **Glassdata**.

·To create a database in MySQL, we use CREATE DATABASEdatabase\_name statement.

**#create a new database**

**query = "Create database GlassData;"**

**cursor = mydb.cursor() #create a cursor to execute queries**

**cursor.execute(query)**

**print("Database Created!!")**

**mydb.close() #close the connection**

·For executing any queries in database, we have to create a cursor and execute it using the query.

·If the database already exists you will get an error. Make sure that the database does not exist.

·To see all the databases we use **SHOW DATABASES** statement.

### 3.  Creating a new table to store the data:

·Creating tables in the database to store the information. Before creating tables, we have to select a database first.

·Run the following code, to select Glassdata database which we have created already.

**#Establish a new connection to the database created above**

**mydb = connection.connect(host="localhost",**

**database = 'GlassData',user="root", passwd="mysql", use\_pure=True)**

·Use the CREATE TABLE table\_name to create a table in the selected database.

**#create a new table to store glass data**

**query = "CREATE TABLE IF NOT EXISTS GlassData (Index\_Number INT(10)," \**

**"RI float(10,5), Na float(10,5), Mg float(10,5),Al float(10,5),**

**" Si float(10,5), K float(10,5), Ca float(10,5)," \**

**"Ba float(10,5), Fe float(10,5), Class INT(5))"**

**cursor = mydb.cursor() # create a cursor to execute queries**

**cursor.execute(query)**

**print("Table Created!!")**

·You will see “Table Created”. If every steps followed successfully.

**4. Selecting a dataset from local machine and write it down inside a database:**

·To select a dataset, we have to write the following code,”

**with open('glass.data', "r") as f:”.**

·Then to read the file we will use **csv.reader.**

****

It will return a reader object which will iterate over lines in the given csvfile. Csvfile can be any object which supports the iterator protocol.

·After this we have to use **enumerate()** function in order to iterate the data. A lot of times when dealing with iterators, we also get a need to keep a count of iterations.

Python eases the programmers’ task by providing a built-in function enumerate() for this task.Enumerate () method adds a counter to an iterable and returns it in a form of enumerate object. This enumerate object can then be used directly in for loops or be converted into a list of tuples using list() method.

·In order to insert the value, we will write down the aboce that will create a loop and insert one by one the whole dataset into the database.

**“cursor.execute('INSERT INTO GlassData values ({values})'.format(values=(list\_)))”**

·And finally we have to commit the database and close the cursor.

·Full code for the reading dataset and write in a database are given below

**#read from the file**

**with open('glass.data', "r") as f:**

**next(f)**

**glass\_data = csv.reader(f, delimiter="\n")**

**for line in enumerate(glass\_data):**

**for list\_ in (line[1]):**

**cursor.execute('INSERT INTO GlassData values ({values})'.format(values=(list\_)))**

**print("Values inserted!!")**

**mydb.commit()**

**cursor.close()**

**mydb.close()**

This is how we can easily store any data from any dataset and update the Database.