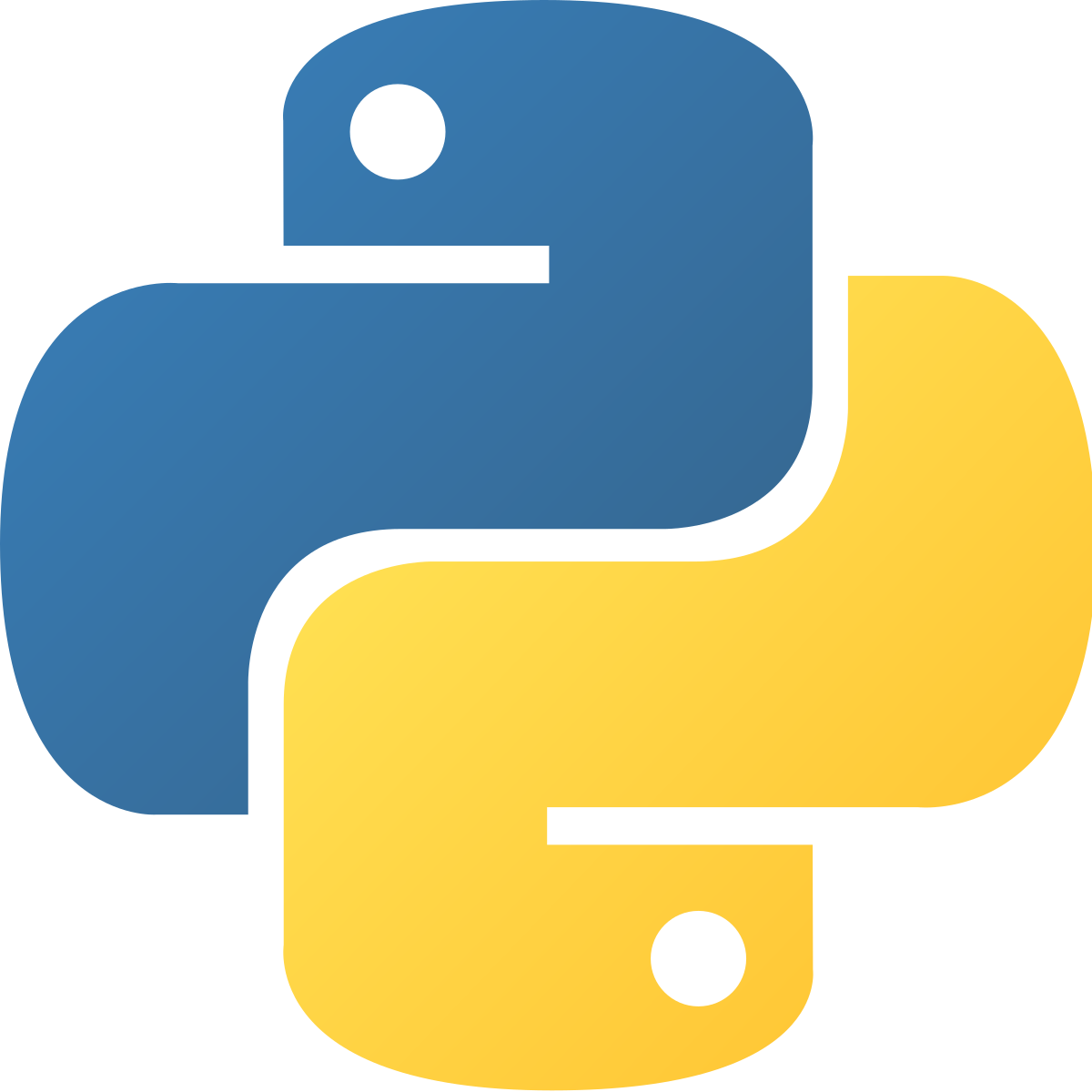
**A PROJECT REPORT ON**

**THE ULTIMATE MANAGER**

**Atharv Agarwal**

| Index | | | |
| --- | --- | --- | --- |
| **S.NO** | **Topic** | **Page No.** | **Teacher’s Signature** |
|  | **Introduction to Python** | 4 |  |
|  | **Features of Python** | 4 – 5 |  |
|  | **Introduction to MySQL** | 6 |  |
|  | **Introduction to MySQL-Python Connectivity** | 6 |  |
|  | **Establishing Connectivity** | 6 |  |
|  | **About the Project** | 7 |  |
|  | **Explanation** | 8-9 |  |
|  | **Coding** | 10-15 |  |
|  | **Output and Execution** | 16-21 |  |
|  | **Technology Used** | 22 |  |
|  | **Bibliography** | 23 |  |

**Introduction to Python**

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* Web Development (server-side),
* Software development
* Mathematics
* System scripting

**Features Of Python**

Python has several features. Some of most popular and powerful features are listed below:

1. **Easy:**
2. **Easy to Code**

Python is very **easy to code**as compared to other popular languages like Java and C++.  
  
 **B) Easy to Read**

Being a high-level language, Python code is quite like English. Looking at it, you can tell what the code is supposed to do. Also, since it is **dynamically-typed**, it mandates indentation. This aids readability

1. **Expressive:**

Python provides us with a myriad of constructs that help us focus on the solution rather than on the syntax.

1. **Free & Open source:**

Firstly, Python is **freely available**. You can download it from the [Python Official Website](https://www.python.org/downloads/).

Secondly, it is **open-source**. This means that its source code is available to the public. You can download it, change it, use it, and distribute it.

1. **High Level:**

Python is a high-level language. This means that as programmers, we don’t need to remember the system architecture. Also, we need not manage memory. This makes it more **programmer-friendly** and is one of the key python features.

1. **Potable:**

You can take one code and run it on any machine. This makes Python a **portable language**. However, you must avoid any system-dependent features in this case.

**Introduction to MySQL**

MySQL is an open-source, fast reliable, and flexible relational database management system.

* **Features of MySQL:**

1. MySQL server design is multi-layered with independent modules.
2. MySQL provides transactional and non-transactional storage engines.
3. MySQL has a high-speed thread based-memory allocation system.

**Introduction to MySQL-Python Connectivity:**

Python MySQL connector is a python driver that helps integrate Python and MySQL. This Python MySQL library allows the conversation between Python and SQL language. MySQL connector is implemented using pure python and doesn’t require any third-party library.

**Establishing Connectivity:**

Python needs a MySQL driver to access the SQL database.

To access the MySQL database from python follow the following steps:

1. Open the Command Prompt (cmd prompt).
2. Change the directory to the scripts folder of the python installation.
3. Type the Command:

pip install mysql-connector-python

**The Ultimate Manager**

* **About The Project:**

It's too expensive to hire someone to create and maintain a management system. So, we are here with an affordable solution. **The** **Ultimate Manager** is a software which provides functionality to create a management system with all the basic functions any organization needs.

The project aims towards providing an affordable solution to anyone who is looking for affordable ways to manage their organisational or individual data. It asks for certain information about the management system user needs and then use this information to create a "Database Management System" which can be used by the user to manage and process their data effectively.

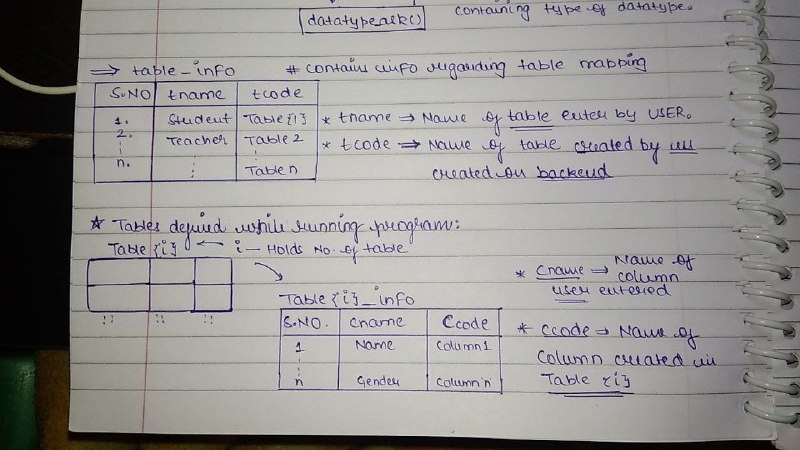
The software uses MySQL database as the backend language and so the user data is secured and encrypted by a password.

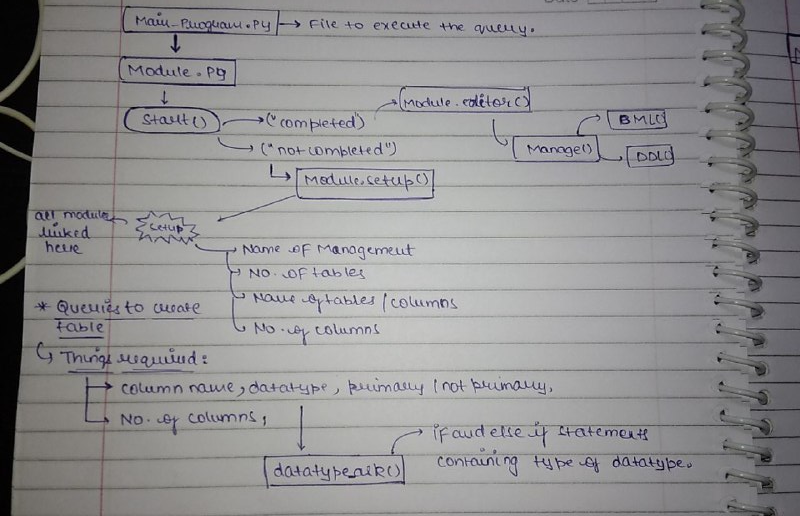
Python is being used as the frontend language which makes the readability of code and bug fixing easy even for the beginners.

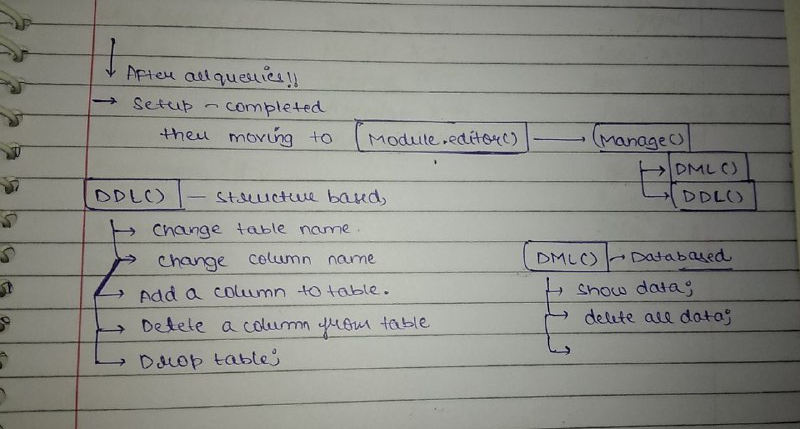
The Ultimate Manager helps you to create a management system which can help you with following: -

* Add data
* View data
* Remove/Modify data
* Create data file

**Explanation**







**Coding**

**Main\_Program.py**import Module as mo  
import mysql.connector as con

path=con.connect(host=”localhost”,user=”root”, passwd=”12345”,database=”user1”)  
  
cur=path.cursor()

cur.execute(“drop database user1”)  
cur.execute(“create database user1”)

**mo.start()**

**Module.py**

import Module  
import mysql.connector as con

path=con.connect(host=”localhost”, user=”root”, passwd=”12345”,database=”user1”)

cur=path.cursor()

global status  
global manager

def start():

status=””:  
 if status == “”:  
 print(“Hello User, Welcome to DBMS”)  
 Module.setup()

**def Editor():**

print(“\nHello User, Welcome to manager”)  
print(“!! Setup Already completed...”,”\t Procedding to Management... !! “)

Module.Manage()

**def setup():**

status=””  
manager=input(“Please Enter the Name of the Management : “)

tno=int(input(“\n\nPlease Enter the Number of Tables : “))

cur.execute(“create table Table\_info(Table\_Name   
 varchar(30),Table\_id varchar(30))”)

for i in range(1,tno+1):

name=input(f”Input the name of table{i} : “)  
cur.execute(f”insert into table\_info values(‘{name}’,’table{i}’)”)

cur.execute(f”create table table{i}\_info(Sno int,column\_name  
varchar(30),column\_id varchar(30),datatype varchar(30))”)

cno= int(input(“\nHow many columns you want : “))

for j in range(1,cno + 1):  
 c\_name=input(f”\nEnter the name of Column{j} : “)  
 m=Module.datatype\_ask()

rec=j,c\_name,f”cname{j}”,m

z=f”cname{j}”

cur.execute(f”insert into table{i}\_info   
 values({j},’{c\_name}’,’{z}’,’{m}’)”)

print(“\nSetup Completed !! Congo !!”)

status=”completed”  
 Module.Editor()

**def datatype\_ask():**

print(“What Type of Data will it Hold:-“)

print(“1. Character (Recomended for Alpha Numerical Values)”,”2.   
 Integer”)  
 print(“3. Fractional (Decimal)”,”4. Date 5. Time”)

n=int(input(‘Enter the orresponding number : ‘))  
 dtype=””

if n==1:  
 dtype=’varchar(100)’

elif n==2:  
 dtype=’int’

elif n==3:  
 dtype=’float’

elif n==4:  
 dtype=’date’

elif n==5:  
 dtype=’time’

else:  
 print(“Invalid Query”)  
return dtype

**def table\_no():**

cur.execute(“select Table\_Name from Table\_info”)  
num=0

for i in cur:  
 num+=1

for j in i:  
 print(j,end=”\t”)  
print(num)

table\_no=int(input(“Please Enter the Corresponding No. to access a   
table : “))

return table\_no

**def Manage():**

print(“\n Choose the type of orrespondi : “)  
print(“1. DDL Options 2.DML options”)  
query = int(input(“Enter the orresponding number : “))

if query==1:  
 Module.DDL()

elif query==2:  
 Module.DML()

else:  
 print(“Invalid Query”)

**def DDL():**

print(“\n What Option do you want to execute ?? “)  
print(“1. Change Table Name”,”2. Drop table”)  
print(“3. Reset Software”,”\n”)

n=int(input(‘Enter the orresponding number : ‘))

if n==1:

table\_no=Module.table\_no()  
 new\_name=input(“Enter the New Name of table : “)

cur.execute(f”update table table\_info set table\_name={new\_name}   
where table\_id=’table{table\_no}’”)  
cur.commit()

print(“Table Name changed Successfully !! “)

elif n==2:

table\_no=Module.table\_no()  
cur.execute(“drop table table{table\_no}\_info”)

cur.execute(“delete from table\_info where   
table\_id=table{table\_no}”)

elif n==3:

cur.execute(“drop database user1”)  
cur.execute(“create database user1”)  
print(“Software Reset !! \t Please run the Setup Again !!”)

else:  
 print(“Invalid Query”)

**def DML():**

print(“What Option do you want to execute ?? “,”\n”)  
print(“1. View Data”,”2. Delete all Data”,”\n”)  
n=int(input(‘Enter the orresponding number : ‘))

if n==1:

table\_no=Module.table\_no()  
 option=cur.execute(f”select \* from table{table\_no}\_info”)  
 print(option)

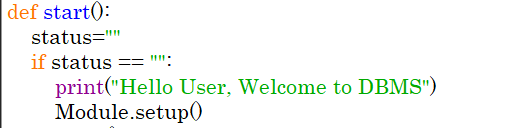
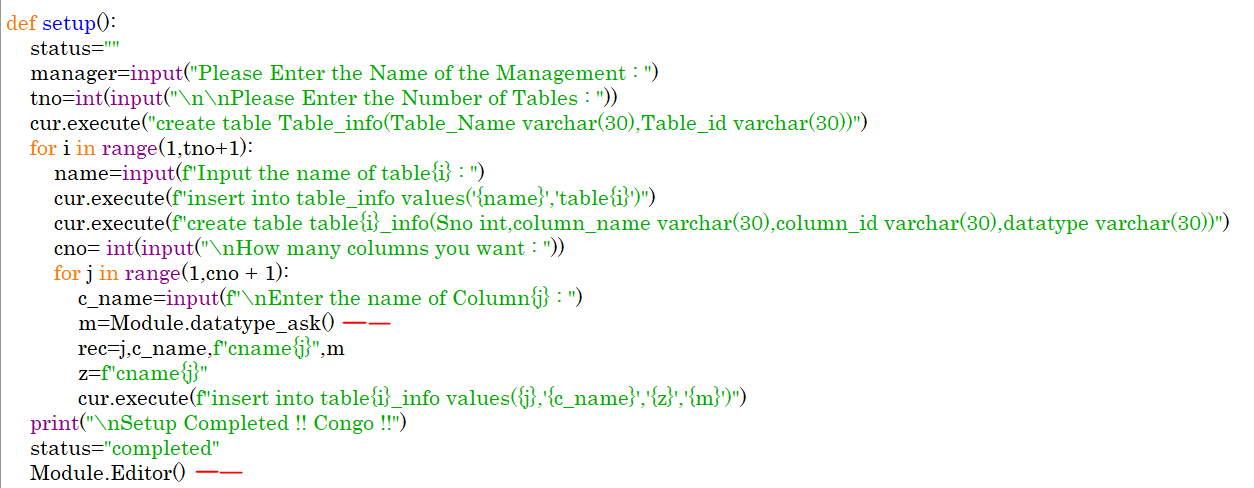
elif n==2:

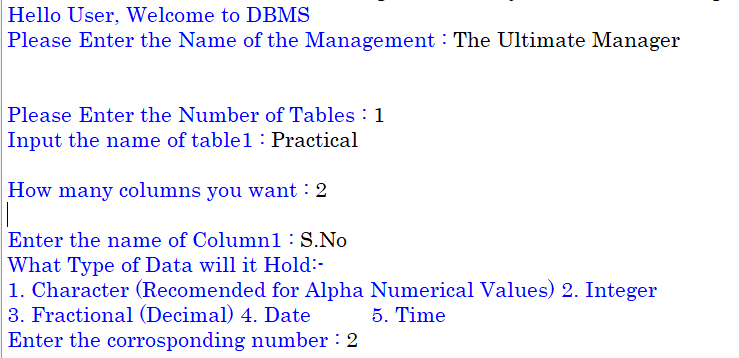
table\_no=Module.table\_no()  
 cur.execute(f”delete from table{table\_no}\_info”)  
 print(“Query Executed !! “)

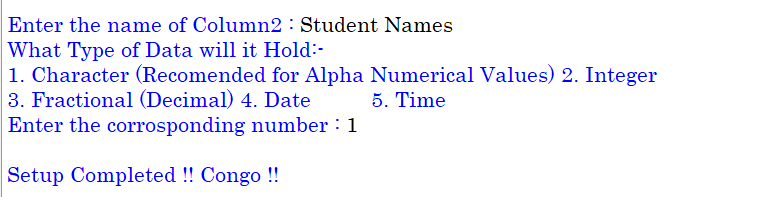
else:  
 print(“Invalid Query”)

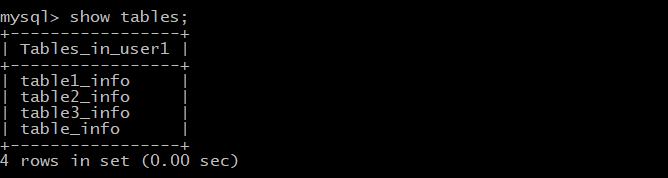
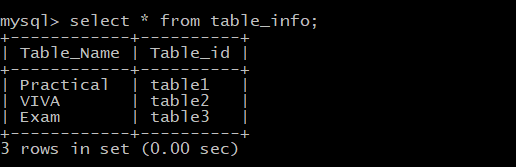
**Execution and Output**

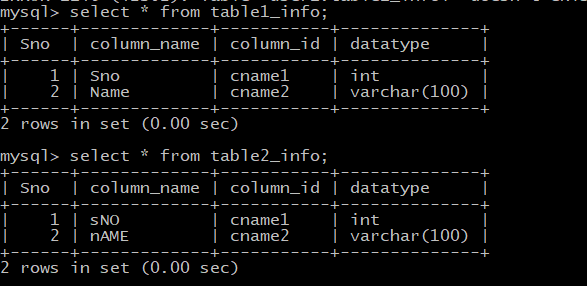
**start(), setup(), datatype\_ask()**

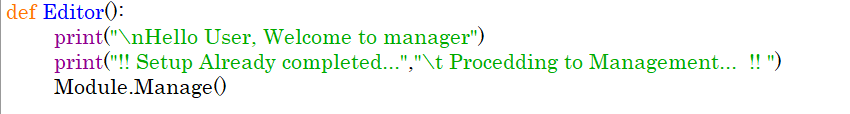
1. 
2. 

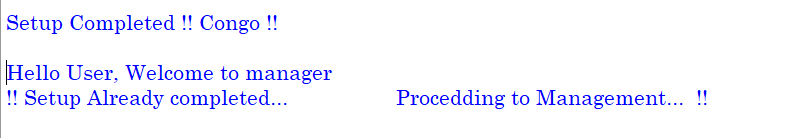


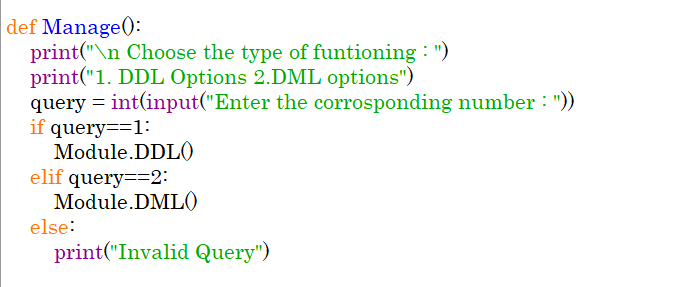


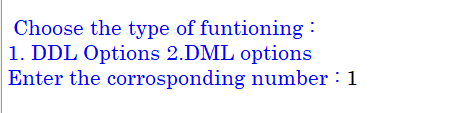
 

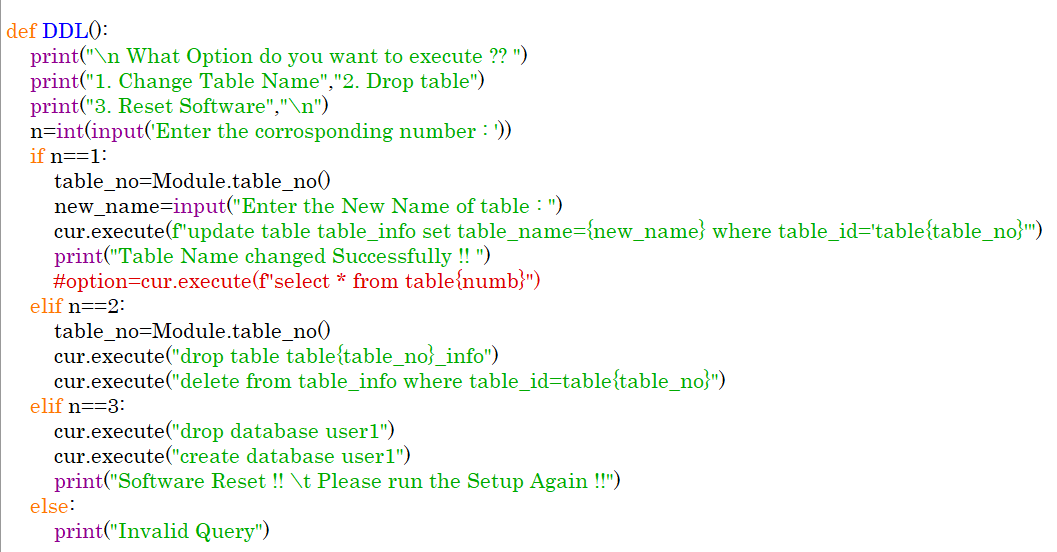


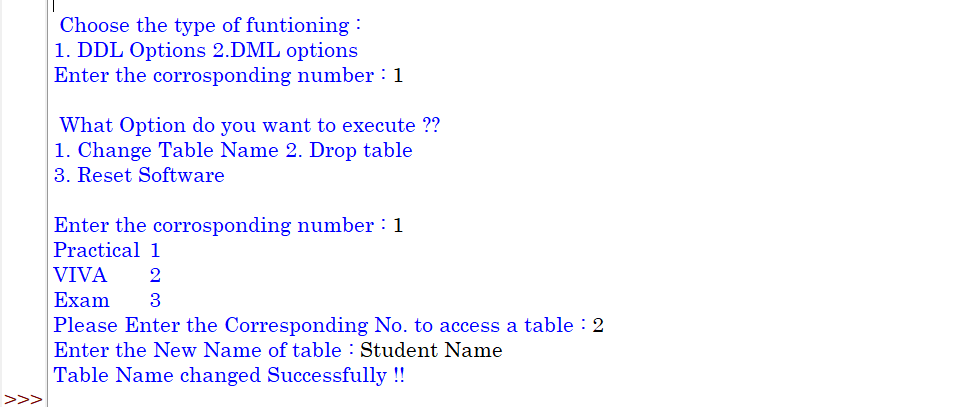
**3)**

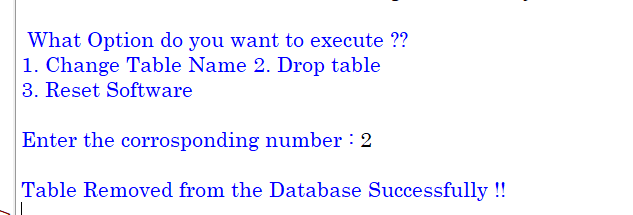


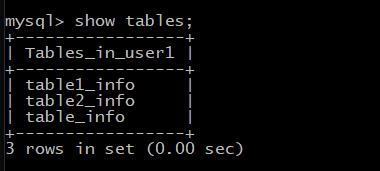
**4)** 

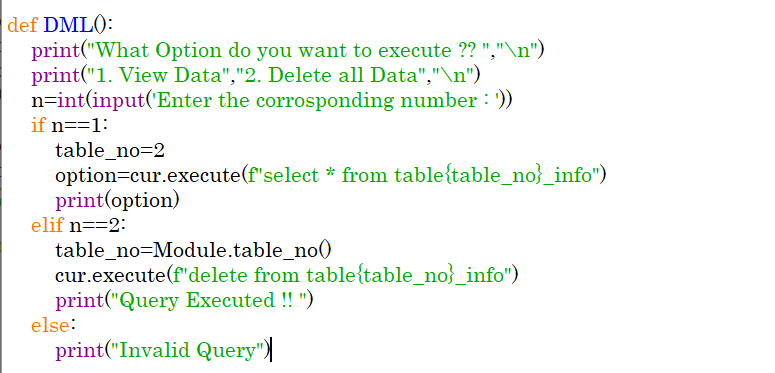


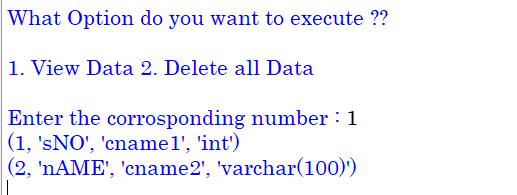
**5)** 

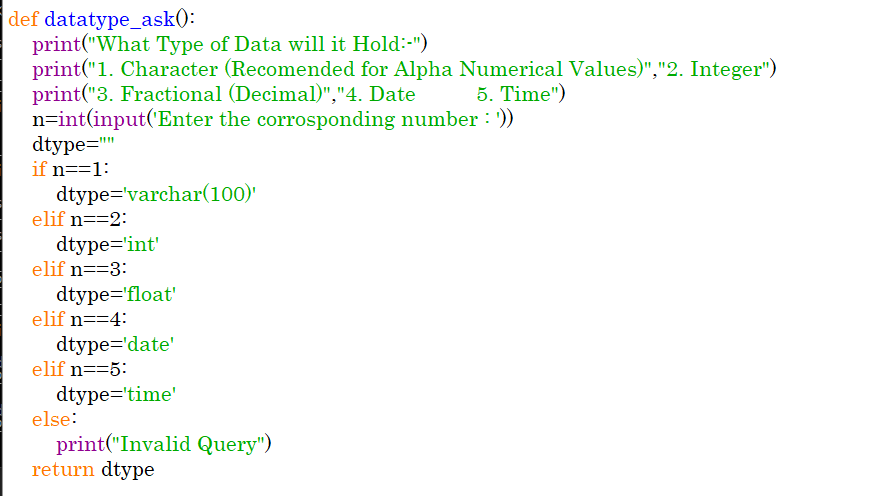


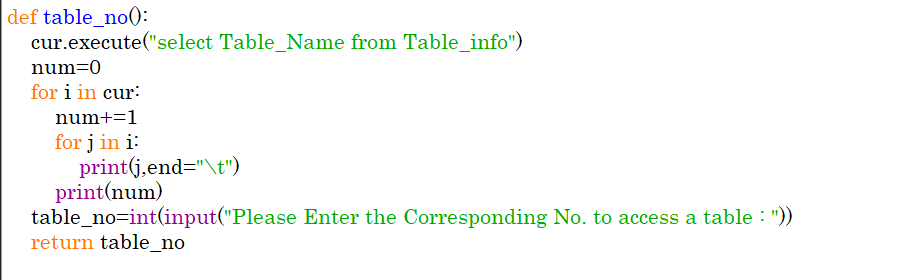




**6)**







**Hardware Requirements**

|  | OPERATING SYSTEM | Windows 7 and Above |
| --- | --- | --- |
|  | PROCESSOR | Pentium(Any) or AMD  Athalon(3800+ - 4200+ DUAL Core) |
|  | MOTHERBOARD | 1.845 or 915,995 for Pentium or MSI |
|  | RAM | 512MB+ |
|  | HARD DISK | SATA 40 GB or Above |

**Software Requirements**

1. Windows OS
2. Python
3. MySQL

**BIBLOGRAPHY**

* Sumita Arora/Preeti Arora  
  (Dhanpat Rai Publications)
* [www.google.com](http://www.google.com), [www.w3schools.com](http://www.w3schools.com)
* [www.python.org.in](http://www.python.org.in), [www.mysql.org](http://www.mysql.org)