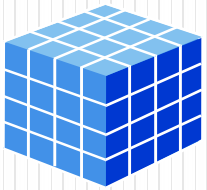
Scripting Language

UNIT-V



K.RaviKanth, Ast.Prof., Dept. of CSE, IIIT- RGUKT- Basara, T.S, IND.

- DB Programming using python
- Creating Database and Tables
- Inserting data
- SelectingData
- Deleting Data
- Updating Data

- The Python standard for database interfaces is the Python DB-API. Most Python database interfaces adhere to this standard.
- You can choose the right database for your application. Python Database API supports a wide range of database servers such as −
- GadFly
- mSQL
- MySQL
- PostgreSQL
- Microsoft SQL Server 2000
- Informix
- Interbase
- Oracle
- Sybase

- Here is the list of available Python database interfaces: Python Database Interfaces and APIs .You must download a separate DB API module for each database you need to access. For example, if you need to access an Oracle database as well as a MySQL database, you must download both the Oracle and the MySQL database modules.
- The DB API provides a minimal standard for working with databases using Python structures and syntax wherever possible. This API includes the following:
  - Importing the API module.
  - Acquiring a connection with the database.
  - Issuing SQL statements and stored procedures.
  - Closing the connection

- We would learn all the concepts using MySQL, so let us talk about MySQLdb module.

- **What is MySQLdb?**
- MySQLdb is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2.0 and is built on top of the MySQL C API.

- **How do I Install MySQLdb?**
- Before proceeding, you make sure you have MySQLdb installed on your machine. Just type the following in your Python script and execute it:

- STEP-1: Firstly choose application mysql-essential-5.0.41-win32.msi
- STEP-2: Run the file
- STEP-3: Next
- STEP-4: Choose complete and click on next
- STEP-5: Click Install
- STEP-6: Click Next and Next
- STEP-7: Close the file

  And click on windows button

  Type mysql choose second one

  Open (Mysql server instance config wizard)
- STEP-8: Cilck next
- STEP-9: Click next

  Upto Modify security settings and set password for your mysql database and click next
- STEP-10: Click on execute

- #If does not exexute all configuration go to (STEP-7)
- Then another file name Is mysql connectivity
- Click to install

  Then click next

  Then next

  Then install
- Ok enjoy with mysql server

Procedure to work with Python Connection with MySQL Database

- Check whether Database is ready on to your system[Install MySQL]
- Do remember the Username & Password and Database name
- Create Database by User Defined Name
- Create Table & Insert the values and Update & Delete the values
- Check the Data by Select command
- Hope now your ready with your Database

- Now open Python IDLE and check for MySQL Module by typing
- **import mysql**
- If not existing do install MySQL Module by following the Procedure:
- Go to mysqlconnector.exe file and run it and install it on to your pc
- Then check your IDLE by typing **import mysql**
- If it shows any error , you can repeat the installation procedure

- **#WAP to connect python application with mysql database**
- **# STEP-1**
- import mysql.connector # it helps us to conenct with the DB
- **# STEP-2 # Open the Database Connection**
- db=mysql.connector.connect(host="localhost",user="root",password="root",database ="anyname")
- # Used to connect the DB. where username and password will be as per system database)
- **# STEP-3** # prepare a cursor object using cursor() m ethod
- cursor=db.cursor() # cursor is an object which is pointing to database
- **#STEP-4** # execute SQL query using execute() m ethod.
- cursor.execute("select version()") # Used to execute the SQL query
- # Fetch a single row using fetchone() m ethod.
- data = cursor.fetchone()
- #data=cursor.fetchall() # which will collect all the records from the DB Table
- print(data) # Print the data form the DB
- **# STEP-5**
- db.close() # close the Database connectivity

- While running this script, it is producing the following result in my Windows machine.
- Database version : 5.0.45
- If a connection is established with the datasource, then a Connection Object is returned and saved into db for further use, otherwise db is set to None. Next, db object is used to create a cursor object, which in turn is used to execute SQL queries. Finally, before coming out, it ensures that database connection is closed and resources are released.
- **Creating Database Table**
- Once a database connection is established, we are ready to create tables or records into the database tables using execute method of the created cursor.

- **Example**
- **#Let us create Database table EMPLOYEE:**
- im port MySQLdb
- # Open database connection
- db = MySQLdb.connect("localhost","testuser","test123","TESTDB")
- # prepare a cursor object using cursor() m ethod
- cursor = db.cursor()
- # Drop table if it already exist using execute() m ethod.
- cursor.execute("DROP TABLE IF EXISTS EMPLOYEE")
- # Create table as per requirem ent
- sql = """CREATE TABLE EMPLOYEE (FIRST_NAME CHAR(20) NOT NULL, LAST_NAME CHAR(20), AGE INT, SEX CHAR(1), INCOME FLOAT)"""
- cursor.execute(sql)
- db.commit()
- # disconnect from server
- db.close()

- **INSERT Operation**
- It is required when you want to create your records into a database table.
- **Example**
- The following example, executes SQL INSERT statement to create a record into EMPLOYEE table
- import MySQLdb
- db = MySQLdb.connect("localhost","testuser","test123","TESTDB")
- # prepare a cursor object using cursor() m ethod
- cursor = db.cursor()
- # Prepare SQL query to INSERT a record into the database.
- sql = """INSERT INTO EMPLOYEE(FIRST_NAME, LAST_NAME, AGE, SEX, INCOME) VALUES ('Mac', 'Mohan', 20, 'M', 2000)"""
- try:
- # Execute the SQL com m and
- cursor.execute(sql)
- # Com m it your changes in the database
- db.com m it()
- except:
- # Rollback in case there is any error
- db.rollback( )
- # disconnect from server
- db.close( )

- **READ Operation**
- READ Operation on any database means to fetch some useful information from the database.
- Once our database connection is established, you are ready to make a query into this database.You can use either **fetchone** method to fetch single record or **fetchall** method to fetch multiple values from a database table.
- **fetchone:** It fetches the next row of a query result set. A result set is an object that is returned when a cursor object is used to query a table.
- **fetchall:** It fetches all the rows in a result set. If some rows have already been extracted from the result set, then it retrieves the remaining rows from the result set.
- **rowcount:** This is a read-only attribute and returns the number of rows that were affectedby an execute method.

Case Study
&
Academic Project

For Details Contact Me @ :

9247448766

ravikanth27787@gmail.com

python

Programming Language