Python Database Communication (PDBC)

Every application or project required to store or save data permanently.
This is done using two systems.

1. File System
2. Database System

Limitations of file system

1. Files are not secured
2. Files cannot hold large amount of data
3. File System does not provide any Query Language

Database Applications or software's

1. Oracle
2. MySQL
3. SQLServer
4. DB2
5. MS-Excel
6. PostgreSQL

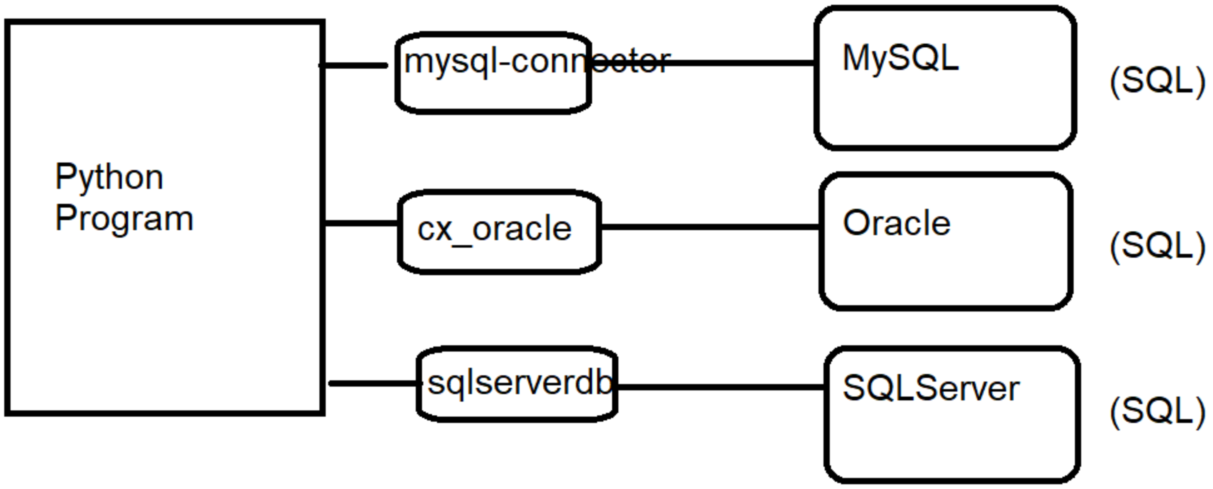
Every Database understands only one language called SQL.
SQL stands for structured query language. This language provides set commands to perform operations on database.
SQL command are classified into different categories

1. DDL (Data Definition Language)
    a. Create
    b. Alter
    c. Drop
2. DML (Data Manipulation Language)
    a. Insert
    b. Update
    c. Delete
3. DRL (Data Read Language)
    a. Select
4. TCL (Transaction Control Language)
    a. Commit
    b. rollback
5. DCL (Data Control Language)
    a. Grant

b. Revoke

In database, data is represented as tables. A table is collection rows and columns.

Python program send these SQL commands to database to perform operations.



Python program communicate with various database applications using libraries. These libraries are provides by database vendors.

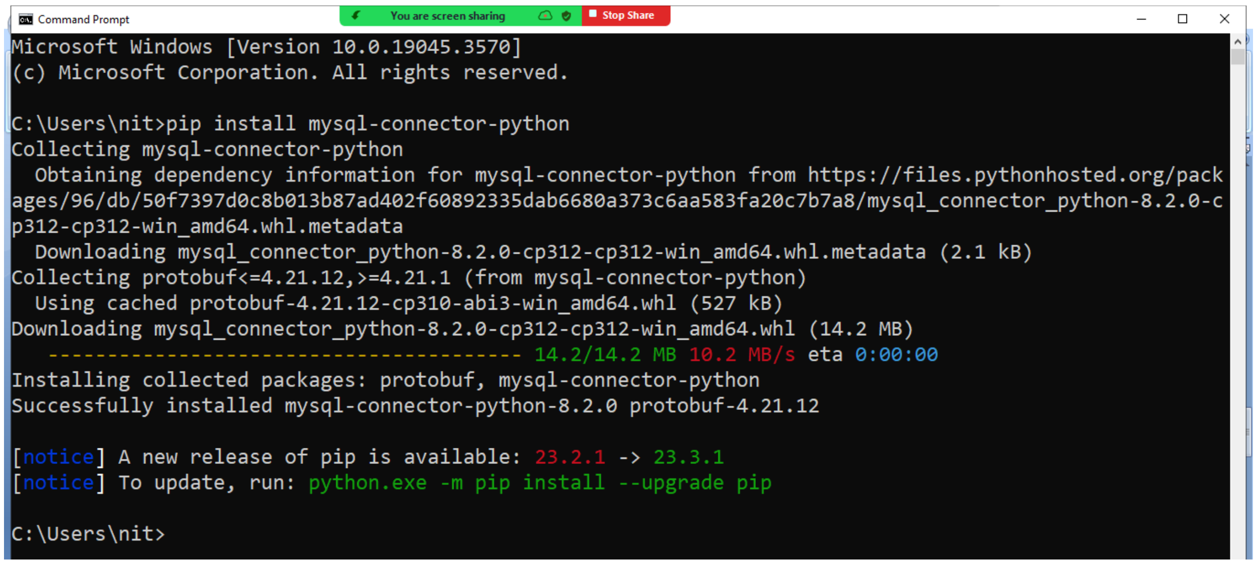
Python-Mysql communication

1. Install mysql database software
2. Install mysql-connector library

Link to download and install mysql database software

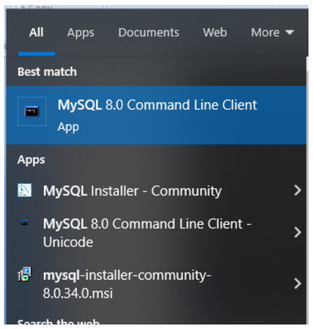
https://dev.mysql.com/downloads/installer/

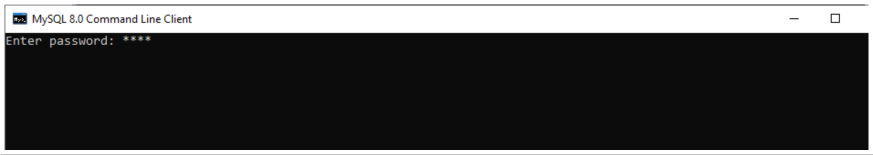
install mysql library (mysql-connector-python)



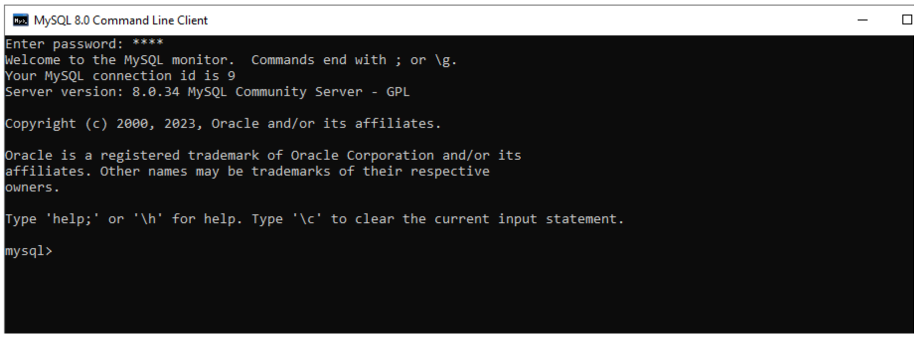
```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nit>pip install mysql-connector-python
Collecting mysql-connector-python
  Obtaining dependency information for mysql-connector-python from https://files.pythonhosted.org/packages/96/db/50f7397d0c8b013b87ad402f60892335dab6680a373c6aa583fa20c7b7a8/mysql_connector_python-8.2.0-cp312-cp312-win_amd64.whl.metadata
  Downloading mysql_connector_python-8.2.0-cp312-cp312-win_amd64.whl.metadata (2.1 kB)
Collecting protobuf<=4.21.12,>=4.21.1 (from mysql-connector-python)
  Using cached protobuf-4.21.12-cp310-abi3-win_amd64.whl (527 kB)
Downloading mysql_connector_python-8.2.0-cp312-cp312-win_amd64.whl (14.2 MB)
   ---------------------------------------- 14.2/14.2 MB 10.2 MB/s eta 0:00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.2.0 protobuf-4.21.12

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\nit>
```

How to work with Mysql Database?


All
Apps
Documents
Web
More
Best match
MySQL 8.0 Command Line Client
App
Apps
MySQL Installer - Community
MySQL 8.0 Command Line Client - Unicode
mysql-installer-community-8.0.34.0.msi
Search the web

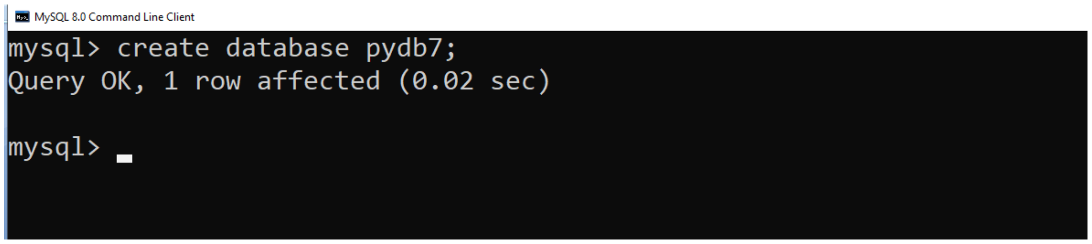


```
Enter password: ****
```



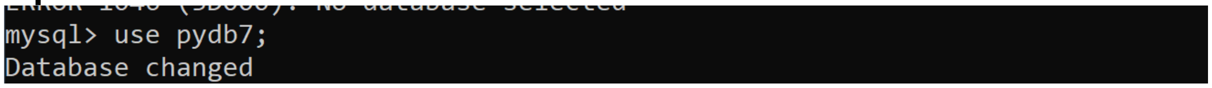
```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Create database in mysql



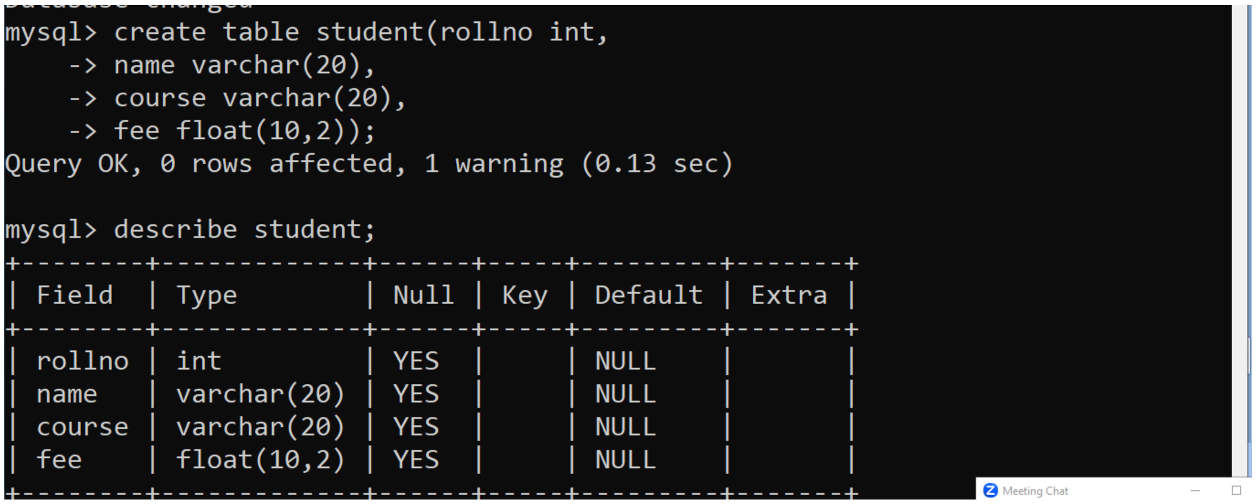
```
mysql> create database pydb7;
Query OK, 1 row affected (0.02 sec)

mysql>
```

Open Database or activate database



```
mysql> use pydb7;
Database changed
```

Create a table



```
mysql> create table student(rollno int,
    -> name varchar(20),
    -> course varchar(20),
    -> fee float(10,2));
Query OK, 0 rows affected, 1 warning (0.13 sec)

mysql> describe student;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| rollno | int         | YES  |     | NULL    |       |
| name   | varchar(20) | YES  |     | NULL    |       |
| course | varchar(20) | YES  |     | NULL    |       |
| fee    | float(10,2) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

Basic steps to communicate with mysql database

1. Import library
2. Establish connection to database
3. Create cursor for sending SQL statements
4. Read result stored in cursor
5. Close connection

Step1:

Import mysql.connector

Step2:

Establishing connection to database

In order send SQL statements, python program (client) should establish connection to database (server).

mysql.connector.connect()
connect() is a predefined function, which establish connection to database and return connection object.

Syntax: connect(database,user,password)

Example:
```
# Program to establish connection to database

import mysql.connector

cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
print("connection established")
```

Output:
```
connection established
```

Step3:
Create cursor for sending SQL statements

Syntax: connection-object.cursor()

Cursor() function return cursor object, using this cursor object python program send SQL statements to database.

Cursor object provides the following methods for sending SQL statements.

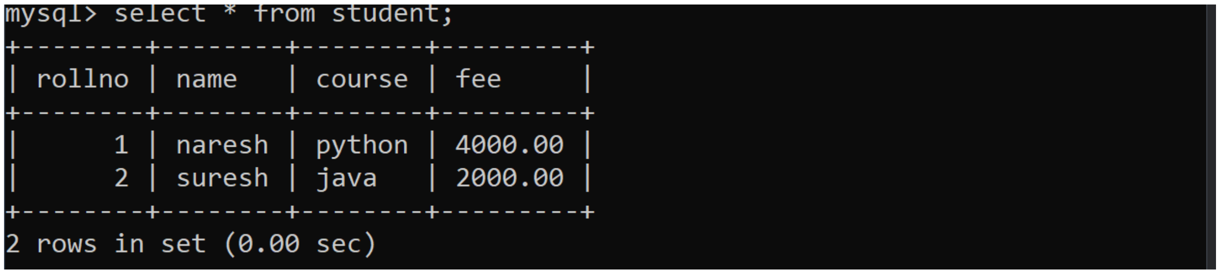
1. execute(SQL)
2. executemany(SQL)

Example:
# Program to insert data into student table (Static SQL statement → SQL statement with values)

import mysql.connector

cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
c=cn.cursor()
c.execute("insert into student values(1,'naresh','python',4000.0)")
c.execute("insert into student values(2,'suresh','java',2000.0)")
print("data is inserted...")
cn.commit()
cn.close()

Output:
data is inserted...



```
mysql> select * from student;
+--------+--------+--------+---------+
| rollno | name   | course | fee     |
+--------+--------+--------+---------+
|      1 | naresh | python | 4000.00 |
|      2 | suresh | java   | 2000.00 |
+--------+--------+--------+---------+
2 rows in set (0.00 sec)
```

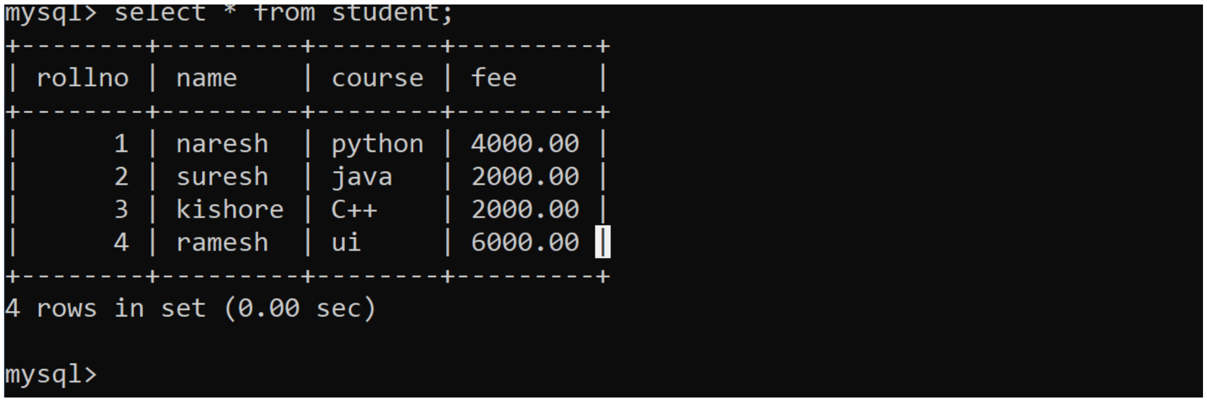
Example:
# Python program to insert data into student table (Dynamic SQL statement → SQL statement with replacement fields)

import mysql.connector

```
cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
c=cn.cursor()
while True:
    rollno=int(input("Enter Rollno "))
    name=input("Enter Name ")
    course=input("Enter Course ")
    fee=float(input("Enter Fees "))
    c.execute("insert into student values(%s,%s,%s,%s)"
,params=[rollno,name,course,fee])
    ans=input("Add another student?")
    if ans=="no":
        cn.commit() # Save changes to database
        break
cn.close()
```

Output:

```
Enter Rollno 3
Enter Name kishore
Enter Course C++
Enter Fees 2000
Add another student?yes
Enter Rollno 4
Enter Name ramesh
Enter Course ui
Enter Fees 6000
Add another student?no
```



```
mysql> select * from student;
+--------+---------+--------+---------+
| rollno | name    | course | fee     |
+--------+---------+--------+---------+
|      1 | naresh  | python | 4000.00 |
|      2 | suresh  | java   | 2000.00 |
|      3 | kishore | C++    | 2000.00 |
|      4 | ramesh  | ui     | 6000.00 |
+--------+---------+--------+---------+
4 rows in set (0.00 sec)

mysql>
```

Reading data from database table

Reading data from database table is done by sending “SELECT” command.

Cursor execute method send “SELECT” query to database and database read data and store inside cursor object.
Cursor object provides the following method to read data.

1. fetchone()
2. fetchmany()
3. fetchall()

fetchone() method returns one row or fetch one row from cursor

```
# Python program to read data from database table

import mysql.connector


cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
c=cn.cursor()
c.execute("select * from student")
s1=c.fetchone()
s2=c.fetchone()
s3=c.fetchone()
print(s1)
print(s2)
print(s3)
cn.close()
```

Output:

```
(1, 'naresh', 'python', 4000.0)
(2, 'suresh', 'java', 2000.0)
(3, 'kishore', 'C++', 2000.0)
```

fetchmany(n)

this method fetch n rows from cursor object.

```
# Python program to read data from database table

import mysql.connector


cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
c=cn.cursor()
c.execute("select * from student")
rows=c.fetchmany(3)
print(rows)
cn.close()
```

Output:
```
[(1, 'naresh', 'python', 4000.0), (2, 'suresh', 'java', 2000.0), (3, 'kishore',
'C++', 2000.0)]
```

fetchall()
fetchall() method of cursor fetch/read all rows from cursor object.
It returns all row in list of tuple.

```
# Python program to read data from database table

import mysql.connector


cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
c=cn.cursor()
c.execute("select * from student")
rows=c.fetchall()
print(rows)
cn.close()
```

Output:
```
[(1, 'naresh', 'python', 4000.0), (2, 'suresh', 'java', 2000.0), (3, 'kishore',
'C++', 2000.0), (4, 'ramesh', 'ui', 6000.0)]
```

Example:
```
# Python program to read data from database table

import mysql.connector


cn=mysql.connector.connect(database="pydb7",user="root",password="roo
t")
c=cn.cursor()
c.execute("select * from student")
rows=c.fetchall()
tot=0
for row in rows:
    print(row)
    tot=tot+row[3]

print(f"Total Fee Collected {tot:.2f}")
cn.close()
```

Output:

```
(1, 'naresh', 'python', 4000.0)
(2, 'suresh', 'java', 2000.0)
(3, 'kishore', 'C++', 2000.0)
(4, 'ramesh', 'ui', 6000.0)
Total Fee Collected 14000.00
```

Updating data or replacing data of table

In order to update or replace values of database table SQL provide “UPDATE” command.

Example:

```
# Write a program to update fee of a given student

import mysql.connector as mysql


with mysql.connect(database="pydb7",user="root",password="root") as cn:
    c=cn.cursor()
    rollno=int(input("Enter Rollno "))
    fee=float(input("Enter Fee "))
    c.execute("update student set fee=%s where
rollno=%s",params=[fee,rollno])
    a=c.rowcount
    if a>0:
        print("student fee is updated...")
        cn.commit()
    else:
        print("Invalid Rollno ")
```

Output:

```
Enter Rollno 1
Enter Fee 9000
student fee is updated...
```

```
Enter Rollno 10
Enter Fee 2000
Invalid Rollno
```

Deleting rows from database table
For deleting rows from database table, SQL provides a command called "DELETE".