**Que.1-Introduction to sqlite3 and pymysql for database connectivity.**

**Ans.**

* **Sqlite3**

**~>**SQLite3 is a lightweight , file based database engine.

~>SQLite3 is embedded within your application ,no separate server required.

~>It’s a Ideal for small to medium-sized applications, local data storage , and prototyping.

Ex.,

Import sqlite3

Conn=sqlite3.connect(‘mydatabase.db’)

Cursor=conn.cursor()

Cursor.execute(‘’’CREATE TABLE customers id INTEGER PRIMARY KEY,name TEXT,email TEXT)’’’)

Cursor.execute(“INSERT INTO customers (name,email) values (‘John Doe’,’john.doe@example.com’)”)

Conn.commit()

Conn.close()

* Pymysql

~>pymysql is a pure-python mysql client library,which enables python applications to connect to mysql databases.

Install pymysql:~

Pip install pymysql

Example.,

Import pymysql

Conn=pymysql.connect(

Host=’localhost’

User=’root’

Password=’ ‘

Db=’sampledb’

)

Cursor=conn.cursor()

Cursor.execute(‘ ‘ ‘ CREATE TABLE IF NOT EXIST Users(id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(255))’ ‘ ‘ )

Cursor.execute(‘SELECT \* FROM users’)

Print(cursor.fetchall())

Conn.close()

**Que.2-creating and executing sql queries from python using these connecters.**

**Ans.**

* Followings are creating and executing sql queries with sqllite3
* Importing sqlite3:--

Import sqlite3

* Connecting to the database:--

conn=sqlite3.connect(‘example.db’)

c=conn.cursor()

* Creating a table:--

c.execute(‘ ‘ ‘ CREATE TABLE IF NOT EXISTS users(id INTEGER PRIMARY KEY AUTOINCREMENT , name TEXT NOT NULL , age INTEGER) ‘ ‘ ‘ )

* Inserting Data:--

c.execute(“INSERT INTO users (name,age) VALUES (? , ?)” , (‘Alice’,30))

c.execute(“INSERT INTO users

(name,age) VALUES (? , ?)” , (‘Bob’,25))

Conn.commit()

* Fetching Data:--

c.execute(‘SELECT \* FROM users’)

Rows=c.fetchall()

For row in rows:

Print(row)

* Updating Data:--

c.execute(“update users set age=? Where name = ? “,(35,’Akshay’))

conn.commit()

* Deleting Data:--

c.execute(“DELETE FROM users WHERE name=?”,(‘Bob’,))

conn.commit()

* Closing the connection:--

conn.close()

* Folllowings are creating and executing sql queries with pymysql
* Installing and importing pymysql:--

Pip install pymysql

Import pymysql

* Connecting tot the database:--

conn = pymysql.connect(

host='localhost',

user='yourusername',

password='yourpassword',

db='yourdatabase'

)

cursor = conn.cursor()

* Creating a table:--

cursor.execute('''

CREATE TABLE IF NOT EXISTS users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

age INT

)

''')

* Inserting Data:--

cursor.execute("INSERT INTO users (name, age) VALUES (%s, %s)", ('Alice', 30))

cursor.execute("INSERT INTO users (name, age) VALUES (%s, %s)", ('Bob', 25))

conn.commit()

* Fetching Data:--

cursor.execute('SELECT \* FROM users')

rows = cursor.fetchall()

for row in rows:

print(row)

* Updating Data:--

cursor.execute("UPDATE users SET age = %s WHERE name = %s", (35, 'Alice'))

conn.commit()

* Deleting Data:--

cursor.execute("DELETE FROM users WHERE name = %s", ('Bob',))

conn.commit()

* Closing the connection:--

conn.close()