Ch-15 Interface of Python with SQL Database

Interface of Python with SQL Database

Python can interact with SQL databases using the **mysql.connector** module. This enables Python programs to:

- Connect to a MySQL database
- Perform SQL queries: INSERT, UPDATE, DELETE, SELECT
- Fetch and display data

1. Connecting Python with MySQL Database

Step-by-Step Setup:

```
import mysql.connector

# Establish connection
con = mysql.connector.connect(
    host="localhost",
    user="root",
    password="your_password",
    database="School"
)

# Create cursor object
cursor = con.cursor()
```

2. SQL Queries Using Python

Inserting Data

```
query = "INSERT INTO Student (RollNo, Name, Marks) VALUES (%s, %s, %s)"
values = (101, 'Ravi', 92)

cursor.execute(query, values)
con.commit()
print("Record inserted successfully.")
```

Updating Data

```
query = "UPDATE Student SET Marks = %s WHERE RollNo = %s"
values = (95, 101)

cursor.execute(query, values)
con.commit()
print("Record updated.")
```

Deleting Data

```
query = "DELETE FROM Student WHERE RollNo = %s"
values = (101,)

cursor.execute(query, values)
con.commit()
print("Record deleted.")
```

Fetching Records (Display)

```
cursor.execute("SELECT * FROM Student")
```

```
# Fetch all rows
rows = cursor.fetchall()

for row in rows:
    print(row)
```

3. Important Functions and Methods

Function	Description
connect()	Connects Python to the MySQL server
cursor()	Creates cursor to execute SQL queries
execute()	Executes an SQL query
commit()	Saves changes made to the database
fetchone()	Fetches one row from result set
fetchall()	Fetches all rows from result set
rowcount	Returns the number of affected rows

Example:

```
cursor.execute("SELECT * FROM Student")
print("Total Records:", cursor.rowcount)
```

4. Using %s Format Specifier and format()

Both are used to safely insert variables into SQL queries to avoid SQL injection.

Using %s (Recommended)

```
query = "INSERT INTO Student (RollNo, Name, Marks) VALUES (%s, %s, %s)"
values = (102, 'Neha', 88)
cursor.execute(query, values)
```

Using .format() (Not safe for user input)

```
roll = 103
name = "Amit"
marks = 75

query = "INSERT INTO Student (RollNo, Name, Marks) VALUES ({}, '{}', {})".format(roll, name, marks)
cursor.execute(query)
```

Note: %s is preferred for security and proper formatting.

5. Sample Python-SQL App (Mini Example)

```
import mysql.connector

def add_student():
    con = mysql.connector.connect(host="localhost", user="root", password="123", database="School")
    cur = con.cursor()
    name = input("Enter name: ")
    roll = int(input("Enter roll no: "))
    marks = float(input("Enter marks: "))
    cur.execute("INSERT INTO Student VALUES (%s, %s, %s)", (roll, name, marks))
    con.commit()
    print("Record inserted.")

add_student()
```

Summary

Task	Method
Connect DB	mysql.connector.connect()
Create Cursor	cursor()
Run Query	execute()
Commit Changes	commit()
Fetch Data	fetchone(), fetchall()
Row Count	rowcount