



جلسه سیزدهم : ارتباط با Python

پایتون و MySQL ارتباط نزدیکی دارند و توسعه‌دهندگان اغلب از این دو ابزار در کنار هم برای ساخت برنامه‌های کاربردی استفاده می‌کنند. پایتون با استفاده از کتابخانه‌هایی مانند **mysql-connector****، یا **SQLAlchemy** امکان اتصال به پایگاه داده MySQL را فراهم می‌کند. این ابزارها به شما اجازه می‌دهند تا دستورات SQL مانند ایجاد جداول، وارد کردن داده‌ها، به‌روزرسانی، حذف و بازیابی اطلاعات را از طریق کد پایتون اجرا کنید. ترکیب قدرت پایتون در پردازش داده‌ها و انعطاف‌پذیری MySQL در مدیریت داده‌های ساختاریافته، آن را به انتخابی ایده‌آل برای توسعه برنامه‌های وب، تحلیل داده و سایر پروژه‌های داده‌محور تبدیل کرده است.

دستور کار :

پایگاه داده ای به نام **serajdb** و جدولی بنام **student** در آن ایجاد نمایید:

```
MariaDB [(none)]> use serajdb;
Database changed
MariaDB [serajdb]> describe student;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    |       |
| name  | varchar(100)  | YES  |     | NULL    |       |
| city  | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.052 sec)
```

برای ارتباط پایتون با mysql لازم است کتابخانه ای بنام **mysql-connector** را به سیستم اضافه نماییم:

خواهیم نوشت :

```
C:\>pip install mysql-connector-python
Requirement already satisfied: mysql-connector-python in c:\users\microsoft\appdata\local\programs\python\python312\l
b\site-packages (9.1.0)
```

توضیح : در سیستم ما قبلا این فایل نصب شده پس خروجی سیستم شما متفاوت خواهد بود.





در اولین بخش برقراری ارتباط با پایگاه داده را کنترل می کنیم :

```
import mysql.connector
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="serajdb" )
if connection.is_connected():
    print("Connected")
```

Connected

خروجی Connected نشان می دهد که ارتباط با موفقیت برقرار شده است.

```
cursor = connection.cursor()
cursor.execute("SELECT * FROM STUDENT")
results = cursor.fetchall()
for row in results:
    print(row)

(11, 'hadi', 'tabriz')
(22, 'reza', 'Tehran')
(33, 'farhad', 'Tabriz')
(44, 'ali', 'Tehran')
```

توضیح : ارسال مستقیم دستورات sql به پایگاه داده روش امنی نمی باشد و چهارچوب هایی مانند Django, Laravel از معماری سه لایه برای این کار استفاده می کنند. مثلا در Django، ارتباط با پایگاه داده به صورت خودکار و از طریق تنظیمات فایل settings.py انجام می شود. ابتدا نوع پایگاه داده (مانند MySQL، PostgreSQL یا SQLite) و اطلاعات اتصال شامل نام پایگاه داده، کاربر، رمز عبور، و آدرس سرور را در بخش DATABASES تنظیم می کنید. به عنوان مثال، برای MySQL از موتور django.db.backends.mysql استفاده می شود. پس از تنظیم، Django به طور خودکار اتصال را مدیریت می کند.





در ادامه تابعی بنام `addStudent` خواهیم نوشت که شماره دانشجویی، نام و شهر دانشجویی را بعنوان پارامتر گرفته و داده ها را به جدول دانشجو اضافه می کند:

```
import mysql.connector
def addstudent(student_id, name, city):
    connection = mysql.connector.connect(
        host="localhost", user="root", password="", database="serajdb")
    cursor = connection.cursor()
    sql = "INSERT INTO STUDENT(id,name,city) VALUES (%s, %s, %s)"
    values = (student_id, name, city)
    cursor.execute(sql, values)
    connection.commit()
    cursor.close()
    connection.close()
addstudent(100, "Farhad", "Tehran")
```

این تابع در ساده ترین حالت ممکن نوشته شده و اصولا باید از بلوک های `try` , `except` استفاده شود تا از ایجاد خطا جلوگیری شود، داده ها را هم از نظر نوع و خالی نبودن قبل از ارسال به پایگاه باید کنترل کرد. بعد از اجرای کد :

```
MariaDB [serajdb]> select * from student;
+-----+-----+-----+
| id  | name  | city  |
+-----+-----+-----+
| 11  | hadi  | tabriz |
| 22  | reza  | Tehran |
| 33  | farhad | Tabriz |
| 44  | ali   | Tehran |
| 100 | Farhad | Tehran |
+-----+-----+-----+
5 rows in set (0.002 sec)
```

تمرین : در زبان پایتون تابعی به نام `get_name_by_id` بنویسید که شماره دانشجویی را بعنوان پارامتر گرفته و نام او را بازگرداند.

