به نام خدا

Sintaxs Mysql in Python

نصب و اتصال

- **نصب کانکتور :**برای استفاده از Mysql در پایتون ابتدا باید آن را نصب کنید.

pip install mysql-connector-python

این دستور را برای نصب Mysql در CMD وارد میکنیم.

.اتصال به دیتابیس

- **هدف :**ایجاد ارتباط بین پایتون و **MySQL**.
- **جزئیات :**باید نام هاست (معمولاً (**`localhost`**، نام کاربری، رمز عبور و دیتابیس رو مشخص کنی.
- **نکته :**اگر دیتابیس رو مشخص نکنی، فقط اتصال برقرار می‌شه و باید بعداً با دستور **`USE`** **`database_name`**دیتابیس رو انتخاب کنی.

۲ .ایجاد دیتابیس و جدول

- **CREATE DATABASE:**برای ساخت دیتابیس جدید استفاده می‌شه.

```
mycursor.execute("CREATE DATABASE mydatabase")
```

- **CREATE TABLE:**برای تعریف ساختار جدول.

- ستون‌ها رو با نوع داده مشخص می‌کنی (**`INT, VARCHAR, DATE`**) و...).
- می‌تونی کلید اصلی (**`PRIMARY KEY`**) و ویژگی‌های خاص مثل **`AUTO_INCREMENT`** بذاری.

```
mycursor.execute("CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), age INT)")
```

۳ .درج داده(INSERT INTO)

- **هدف :**اضافه کردن رکورد جدید.
- **جزئیات :**از **placeholder (%s)** استفاده می‌کنیم تا جلوی تزریق کد (**SQL Injection**) گرفته بشه.
- **نکته :**بعد از درج باید `commit()` بزنی تا تغییرات ذخیره بشه.

```
sql = "INSERT INTO students (name, age) VALUES (%s, %s)" val = ("Ali", 21) mycursor.execute(sql, val) mydb.commit()
```

۴ .خواندن داده(SELECT)

- **هدف :**گرفتن اطلاعات از جدول.
- **روش‌ها:**
  - `fetchall()` همه رکوردها رو میاره.
  - `fetchone()` فقط یک رکورد میاره.
- **مثال:**

```
mycursor.execute("SELECT * FROM students WHERE age > 20 ORDER BY name") result = mycursor.fetchall() for row in result: print(row)
```

۵ .بروزرسانی داده(UPDATE)

- **هدف :**تغییر مقدار ستون‌ها.
- **نکته :**همیشه شرط (`WHERE`) بذار تا همه رکوردها اشتباهی تغییر نکنن.

```
sql = "UPDATE students SET age = %s WHERE name = %s" val = (22, "Ali") mycursor.execute(sql, val) mydb.commit()
```

۶ .حذف داده(DELETE)

- **هدف :**پاک کردن رکوردها.
- **نکته :**اگر شرط نذاری، کل جدول پاک می‌شه.

```
sql = "DELETE FROM students WHERE name = %s" val = ("Ali",) mycursor.execute(sql, val) mydb.commit()
```

۷ .مدیریت خطاها

- **هدف :**جلوگیری از کرش کردن برنامه در صورت خطا.

- **مثال:**

```
try: mycursor.execute("SELECT * FROM students") except mysql.connector.Error as err: print("Error:", err)
```

۸. بستن اتصال

- **هدف:** آزاد کردن منابع و جلوگیری از مشکلات.

```
mycursor.close() mydb.close()
```

۹. دستورات پیشرفته‌تر

- **LIMIT:** محدود کردن تعداد رکوردها.

```
mycursor.execute("SELECT * FROM students LIMIT 5")
```

- **JOIN:** ترکیب داده‌ها از چند جدول.

```
mycursor.execute("SELECT students.name, courses.title FROM students JOIN courses ON students.course_id = courses.id")
```

- **GROUP BY و توابع تجمعی:** برای دسته‌بندی داده‌ها.

```
mycursor.execute("SELECT age, COUNT(*) FROM students GROUP BY age")
```