

# اتصال به پایگاه داده MySQL/PostgreSQL در پایتون

در این بخش یاد می‌گیریم که چگونه به پایگاه داده‌های MySQL و PostgreSQL متصل شویم، اطلاعات اتصال را ارائه دهیم و خطاهای احتمالی را مدیریت کنیم.

## ۱. استفاده از `connect()` برای اتصال به پایگاه داده

برای اتصال به پایگاه داده از تابع `connect()` که در کتابخانه‌های مربوطه وجود دارد، استفاده می‌کنیم.

### 📌 اتصال به MySQL با `mysql-connector-python`

```
import mysql.connector

try:
    # اتصال به پایگاه داده
    conn = mysql.connector.connect(
        host="localhost", # آدرس سرور
        user="root",      # نام کاربری
        password="password", # رمز عبور
        database="test_db" # نام پایگاه داده
    )

    if conn.is_connected():
        print("✅ MySQL اتصال به")

    # بستن اتصال
    conn.close()

except mysql.connector.Error as e:
    print(f"❌ MySQL خطا در اتصال به {e}")
```

📌 نکات: ✅

- از `conn.is_connected()` برای بررسی موفقیت‌آمیز بودن اتصال استفاده شده است.
- در صورت بروز خطا، آن را در `except` مدیریت می‌کنیم.

### 📌 اتصال به MySQL با `PyMySQL`

```
import pymysql

try:
    conn = pymysql.connect(
        host="localhost",
        user="root",
        password="password",
        database="test_db"
    )
```

```
print("✅ برقرار شد PyMySQL با MySQL اتصال به")
```

```
conn.close()
```

```
except pymysql.MySQLError as e:
```

```
print(f"❌ MySQL خطا در اتصال به {e}")
```

✅ تفاوت با `mysql-connector-python`

• `PyMySQL` برای پروژه‌های سبک‌تر گزینه مناسبی است.

## 📌 اتصال به PostgreSQL با `psycopg2`

```
import psycopg2
```

```
try:
```

```
    # اتصال PostgreSQL به
```

```
    conn = psycopg2.connect(
```

```
        host="localhost", # آدرس سرور
```

```
        user="postgres", # نام کاربری PostgreSQL (پیش‌فرض)
```

```
        password="password", # رمز عبور
```

```
        database="test_db" # نام پایگاه داده
```

```
    )
```

```
    print("✅ برقرار شد PostgreSQL اتصال به")
```

```
    conn.close()
```

```
except psycopg2.Error as e:
```

```
    print(f"❌ PostgreSQL خطا در اتصال به {e}")
```

✅ نکات:

- نام کاربری پیش‌فرض PostgreSQL `postgres` است.
- در صورت بروز خطا، پیام خطا نمایش داده می‌شود.

## ۲. مدیریت خطاهای اتصال

یکی از مهم‌ترین بخش‌های اتصال به پایگاه داده، مدیریت خطاها است.

### 📌 انواع خطاهای رایج در اتصال به MySQL و PostgreSQL

خطا	دلیل
<code>Access denied for user</code>	نام کاربری یا رمز عبور اشتباه است.
<code>Unknown database</code>	پایگاه داده موردنظر وجود ندارد.
<code>Can't connect to MySQL server</code>	MySQL اجرا نشده یا فایروال دسترسی را مسدود کرده است.

خطا	دلیل
could not connect to server	PostgreSQL در حال اجرا نیست یا پورت صحیح تنظیم نشده است.

## 📌 مدیریت پیشرفته خطاها

```
import mysql.connector

try:
    conn = mysql.connector.connect(
        host="localhost",
        user="wrong_user", # نام کاربری اشتباه
        password="wrong_password",
        database="test_db"
    )
    print("✅ اتصال موفقیت‌آمیز بود")
    conn.close()

except mysql.connector.Error as e:
    if e.errno == 1045: # Access denied
        print("❌ خطا: نام کاربری یا رمز عبور اشتباه است")
    elif e.errno == 1049: # Unknown database
        print("❌ خطا: پایگاه داده یافت نشد")
    else:
        print(f"❌ خطای دیگر: {e}")
```

✅ مدیریت دقیق‌تر خطاها باعث می‌شود مشکلات اتصال را سریع‌تر شناسایی کنیم.

## جمع‌بندی

- ✅ MySQL: استفاده از `mysql-connector-python` یا `PyMySQL` برای اتصال
- ✅ PostgreSQL: استفاده از `psycopg2` برای اتصال
- ✅ ارائه جزئیات اتصال مانند `host`, `user`, `password`, `database`
- ✅ مدیریت خطاها با استفاده از `try-except` برای تشخیص مشکلات متداول

🚀 در ادامه: انجام عملیات CRUD (ایجاد، خواندن، به‌روزرسانی و حذف) در MySQL و PostgreSQL!