

عملیات CRUD در SQLite با پایتون

CRUD مخفف چهار عملیات اصلی در پایگاه داده است:

- Create (ایجاد): ساخت جداول و افزودن داده‌ها
- Read (خواندن): بازیابی داده‌ها از پایگاه داده
- Update (به‌روزرسانی): تغییر مقادیر موجود در جداول
- Delete (حذف): حذف داده‌های خاص از جداول

SQLite از SQL استاندارد برای این عملیات پشتیبانی می‌کند. در ادامه، هر یک از این عملیات را همراه با مثال توضیح می‌دهیم.

۱. Create (ایجاد جداول و افزودن داده)

برای ایجاد یک جدول جدید از `CREATE TABLE` و برای افزودن داده‌ها از `INSERT INTO` استفاده می‌کنیم.

ایجاد جدول

```
import sqlite3

conn = sqlite3.connect("test.db")
cursor = conn.cursor()

# ایجاد جدول کاربران (اگر وجود ندارد)
cursor.execute("""
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    age INTEGER,
    email TEXT UNIQUE
)
""")

conn.commit()
conn.close()
```

افزودن داده‌ها به جدول

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

# افزودن یک کاربر جدید
cursor.execute("""
INSERT INTO users (name, age, email) VALUES (?, ?, ?)
""", ("محمدجواد", 25, "mj@example.com"))

# افزودن چندین کاربر
users = [
    ("علی", 30, "ali@example.com"),
    ("سارا", 28, "sara@example.com")
]
```

```
]
cursor.executemany("INSERT INTO users (name, age, email) VALUES (?, ?, ?)", users)

conn.commit()
conn.close()
```

۲. Read (خواندن داده‌ها از پایگاه داده)

از `SELECT` برای دریافت داده‌ها استفاده می‌کنیم.

دریافت همه کاربران

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("SELECT * FROM users")
users = cursor.fetchall() # دریافت تمام سطرها

for user in users:
    print(user)

conn.close()
```

📌 توضیح: `fetchall()` تمام نتایج را در قالب لیستی از تاپل‌ها برمی‌گرداند.

دریافت یک کاربر خاص

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("SELECT * FROM users WHERE name = ?", ("محمدجواد",))
user = cursor.fetchone() # دریافت فقط یک سطر

print(user)

conn.close()
```

📌 توضیح: `fetchone()` فقط یک سطر از نتیجه را برمی‌گرداند.

دریافت کاربران با فیلتر خاص

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("SELECT name, age FROM users WHERE age > ?", (25,))
users = cursor.fetchall()

for user in users:
    print(user) # نمایش نام و سن کاربران بالای ۲۵ سال

conn.close()
```

۳. Update (به‌روزرسانی داده‌ها)

برای تغییر داده‌های موجود از `UPDATE` همراه با `WHERE` استفاده می‌شود.

به‌روزرسانی سن یک کاربر

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("""
UPDATE users
SET age = ?
WHERE name = ?
""", (26, "محمدجواد"))

conn.commit()
conn.close()
```

📌 توضیح: مقدار `age` برای کاربری با نام "محمدجواد" به ۲۶ تغییر می‌کند.

به‌روزرسانی ایمیل کاربر خاص

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("""
UPDATE users
SET email = ?
WHERE id = ?
""", ("mj_new@example.com", 1))

conn.commit()
conn.close()
```

۴. Delete (حذف داده‌ها)

برای حذف داده از `DELETE FROM` همراه با `WHERE` استفاده می‌شود.

حذف یک کاربر خاص

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("DELETE FROM users WHERE name = ?", ("علی",))

conn.commit()
conn.close()
```

📌 توضیح: این دستور تمام رکوردهایی که نام آن‌ها "علی" است را حذف می‌کند.

حذف تمامی کاربران بالای ۲۹ سال

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("DELETE FROM users WHERE age > ?", (29,))

conn.commit()
conn.close()
```

حذف تمامی داده‌ها از جدول (ریست کردن جدول)

```
conn = sqlite3.connect("test.db")
cursor = conn.cursor()

cursor.execute("DELETE FROM users")

conn.commit()
conn.close()
```

📌 توضیح: این کد تمامی داده‌ها را حذف می‌کند اما خود جدول باقی می‌ماند.

جمع‌بندی

- ✓ Create (ایجاد): ایجاد جدول با `CREATE TABLE` و درج داده با `INSERT INTO`
- ✓ Read (خواندن): دریافت داده‌ها با `SELECT` و متدهای `fetchone()` و `fetchall()`
- ✓ Update (به‌روزرسانی): تغییر داده‌ها با `UPDATE` و تعیین شرط با `WHERE`
- ✓ Delete (حذف): حذف داده‌ها با `DELETE FROM` و اعمال فیلتر مناسب

با این روش‌ها، شما می‌توانید عملیات CRUD را به‌طور کامل روی پایگاه داده SQLite انجام دهید. اگر سوالی داشتی بگو! 🚀