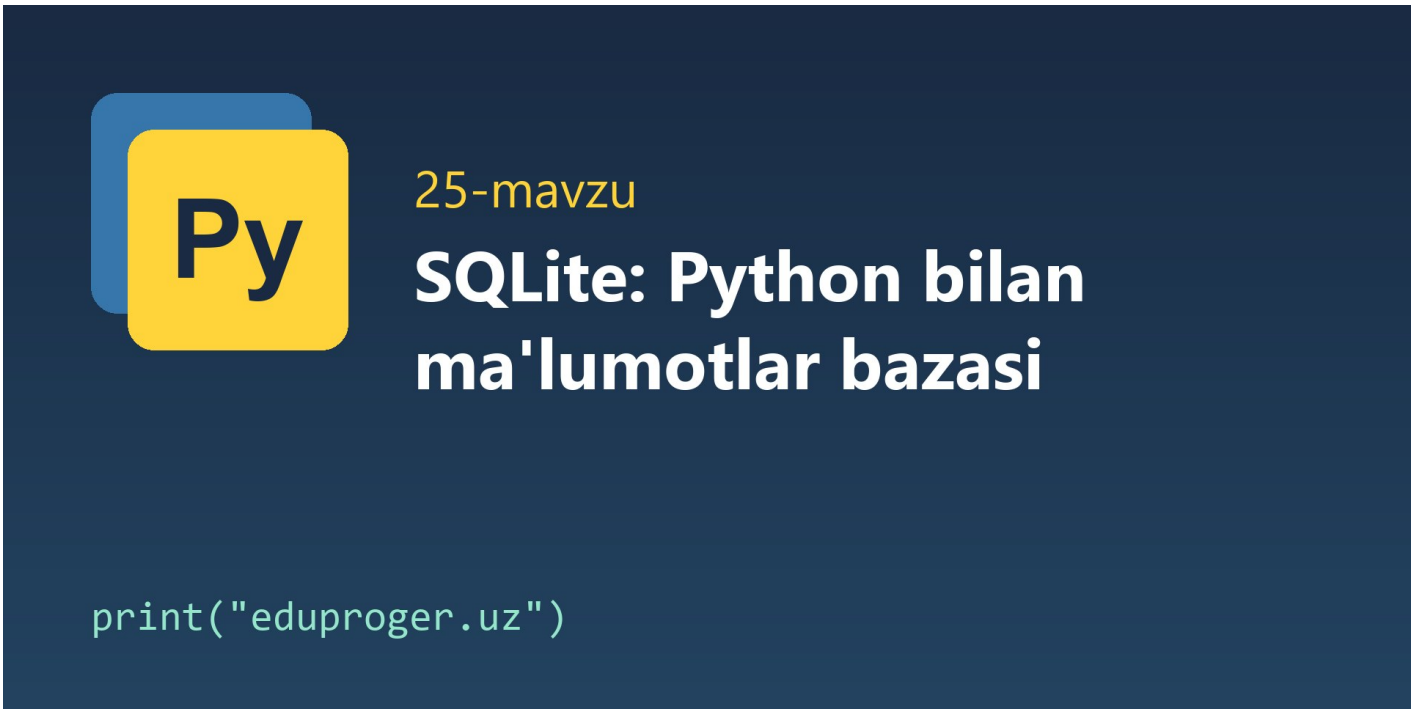
25. SQLite: Python bilan ma'lumotlar bazasi



Ma'lumot ko'payganda uni fayllarda saqlash noqulay bo'ladi: qidirish, saralash, bir vaqtda yozish qiyinlashadi. Buning uchun **ma'lumotlar bazasi** ishlatiladi. **SQLite** — bitta fayldan iborat yengil baza, Python bilan birga keladi (`sqlite3` moduli), o'rnatish shart emas. Bazaga so'rovlar **SQL** tilida yoziladi.

> **Bog'liq**
>
> Saytdagi «SQL» bosqichi masalalarida SQL'ni alohida mashq qilish mumkin. Bu darsda SQL'ni Python dasturi ICHIDAN ishlatamiz.

1. Ulanish va jadval yaratish

`sqlite3.connect("maktab.db")` fayldagi bazaga ulanadi (fayl bo'lmasa yaratadi). `":memory:"` — xotiradagi vaqtinchalik baza (misollar uchun qulay). So'rovlar `cursor` orqali yoki to'g'ridan-to'g'ri `conn.execute()` bilan bajariladi:

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.execute("""
    CREATE TABLE talabalar (
        id INTEGER PRIMARY KEY,
        ism TEXT NOT NULL,
        sinf INTEGER,
```

```
        ball INTEGER
    )
""")
print("Jadval yaratildi")
```

Natija:

```
Jadval yaratildi
```

SQLite turi	Python turi
INTEGER	int
REAL	float
TEXT	str
NULL	None

2. Ma'lumot qo'shish: INSERT

Qiymatlar so'rov matniga TO'G'RIDAN-TO'G'RI yopishtirilmaydi — `?` belgilari va alohida kortej orqali beriladi (**parametrli so'rov**). Bir nechta qatorni `executemany` bilan qo'shamiz. O'zgarishlar `commit()` bilan saqlanadi:

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.execute("CREATE TABLE talabalar (id INTEGER PRIMARY KEY, ism TEXT, sinf I
conn.execute("INSERT INTO talabalar (ism, sinf, ball) VALUES (?, ?, ?)", ("Ali
royxat = [("Malika", 9, 95), ("Sardor", 10, 88), ("Zarina", 10, 64), ("Bekzod"
conn.executemany("INSERT INTO talabalar (ism, sinf, ball) VALUES (?, ?, ?)", r
conn.commit()
print(conn.execute("SELECT COUNT(*) FROM talabalar").fetchone()[0], "ta talaba
```

Natija:

```
5 ta talaba
```

3. O'qish: SELECT

Natijani `fetchall()` (hammasi — kortejlar ro'yxati), `fetchone()` (bittasi) yoki to'g'ridan-to'g'ri `for` bilan olamiz:

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.execute("CREATE TABLE t (ism TEXT, sinf INTEGER, ball INTEGER)")
conn.executemany("INSERT INTO t VALUES (?, ?, ?)", [("Ali", 9, 78), ("Malika",

print(conn.execute("SELECT ism, ball FROM t WHERE ball >= 70 ORDER BY ball DES
for ism, sinf in conn.execute("SELECT ism, sinf FROM t WHERE sinf = ? ORDER BY
    print(ism, sinf)
print(conn.execute("SELECT ism FROM t WHERE ism LIKE '%a'").fetchall())
```

Natija:

```
[('Malika', 95), ('Sardor', 88), ('Ali', 78)]
Sardor 10
Zarina 10
[('Malika',), ('Zarina',)]
```

4. Agregat funksiyalar va GROUP BY

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.execute("CREATE TABLE t (ism TEXT, sinf INTEGER, ball INTEGER)")
conn.executemany("INSERT INTO t VALUES (?, ?, ?)", [("Ali", 9, 78), ("Malika",

print(conn.execute("SELECT COUNT(*), AVG(ball), MAX(ball), MIN(ball) FROM t").
for sinf, soni, ortacha in conn.execute("SELECT sinf, COUNT(*), ROUND(AVG(ball
    print(f"{sinf}-sinf: {soni} ta, o'rtacha {ortacha}")
```

Natija:

```
(5, 75.4, 95, 52)
9-sinf: 3 ta, o'rtacha 75.0
10-sinf: 2 ta, o'rtacha 76.0
```

5. O'zgartirish va o'chirish

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.execute("CREATE TABLE t (ism TEXT, ball INTEGER)")
conn.executemany("INSERT INTO t VALUES (?, ?)", [("Ali", 78), ("Vali", 40), ("
conn.execute("UPDATE t SET ball = ball + 5 WHERE ball < 60")
kur = conn.execute("DELETE FROM t WHERE ball < 50")
print("o'chirildi:", kur.rowcount)
conn.commit()
print(conn.execute("SELECT * FROM t").fetchall())
```

Natija:

```
o'chirildi: 1
[('Ali', 78), ('Gani', 60)]
```

> **Diqqat**
>
> `UPDATE` va `DELETE` da `WHERE` ni unutmang — aks holda JADVALDAGI HAMMA qator o'zgaradi yoki o'chadi!

6. SQL injection: nega parametrli so'rov SHART

Foydalanuvchi kiritgan matnni so'rovga f-string bilan qo'shish — eng xavfli xato.
Yomon niyatli foydalanuvchi maxsus matn bilan so'rovni o'zgartirib yuborishi mumkin:

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.execute("CREATE TABLE users (login TEXT, parol TEXT)")
```

```
conn.execute("INSERT INTO users VALUES ('admin', '12345')")

login, parol = "admin", "' OR '1'='1"
xavfli = f"SELECT * FROM users WHERE login = '{login}' AND parol = '{parol}'"
print("f-string:", conn.execute(xavfli).fetchall())     # parolsiz kirib oldi!
xavfsiz = "SELECT * FROM users WHERE login = ? AND parol = ?"
print("parametr:", conn.execute(xavfsiz, (login, parol)).fetchall())
```

Natija:

```
f-string: [('admin', '12345')]
parametr: []
```

7. Ikki jadval: JOIN

```
import sqlite3
conn = sqlite3.connect(":memory:")
conn.executescript("""
    CREATE TABLE sinflar (id INTEGER PRIMARY KEY, nom TEXT);
    CREATE TABLE talabalar (ism TEXT, sinf_id INTEGER);
    INSERT INTO sinflar VALUES (1, '9-A'), (2, '10-B');
    INSERT INTO talabalar VALUES ('Ali', 1), ('Malika', 2), ('Sardor', 1);
""")
for ism, sinf in conn.execute("SELECT t.ism, s.nom FROM talabalar t JOIN sinfl
    print(ism, "—", sinf)
```

Natija:

```
Malika — 10-B
Ali — 9-A
Sardor — 9-A
```

8. with va fayldagi baza

Haqiqiy dasturda baza faylda saqlanadi va `with` bloki tranzaksiyani boshqaradi (xatosiz tugasa — commit, xato bo'lsa — rollback):

```
import sqlite3
conn = sqlite3.connect("maktab.db")
with conn:
    conn.execute("CREATE TABLE IF NOT EXISTS izohlar (matn TEXT)")
    conn.execute("INSERT INTO izohlar VALUES (?)", ("Birinchi yozuv",))
print(conn.execute("SELECT COUNT(*) FROM izohlar").fetchone()[0])
conn.close()
```

Natija:

```
1
```

9. Xulosa

- `sqlite3.connect(fayl)` → `execute(sql, params)` → `fetchall/fetchone` → `commit()`.

- Qiymatlar FAQAT ? parametrlari orqali — SQL injection'dan himoya.
- SELECT: WHERE, ORDER BY, LIKE, COUNT/AVG/MAX, GROUP BY, JOIN.
- UPDATE/DELETE — WHERE bilan; `with conn:` — avtomatik commit/rollback.