

Amaliy ish № 5

SQL bilan ishlash

Ishning maqsadi:

Python orqali SQL so'rovlari yordamida relyatsion ma'lumotlar bazalaridan ma'lumotlarni ajratib olishni o'rganish.

Nazariy qism:

1. SQL nima?

SQL (Structured Query Language) - relyatsion ma'lumotlar bazalari bilan ishlash uchun mo'ljallangan dasturlash tili. Bu sizga ma'lumotlarni o'qish, yozish, yangilash va o'chirish operatsiyalarini bajarish imkonini beradi.

SQL quyidagilar uchun ishlatiladi:

- Jadvallar va ma'lumotlar bazalarini yaratish.
- So'rovlar yordamida ma'lumotlarni olish (`SELECT`).
- Ma'lumotlarni filtrlash (`WHERE`).
- Ma'lumotlarni guruhlash va yig'ish (`GROUP BY`, `COUNT`, `SUM` va boshqalar).

2. SQLite nima?

- SQLite - bu alohida serverni talab qilmaydigan engil o'rnatilgan DBMS (ma'lumotlar bazasini boshqarish tizimi). Bu o'qitish va kichik loyihalar uchun ideal.
- SQLite ma'lumotlar bazasini bitta faylda saqlaydi va undan foydalanishni osonlashtiradi.

3. Python orqali SQL bilan qanday ishlash mumkin?

- Pythonda SQL bilan ishlash uchun `SQLAlchemy` yoki `sqlite3` kutubxonalaridan foydalaniladi. Hozir `SQLAlchemy`, chunki u universalroq va turli ma'lumotlar bazalarini qo'llab-quvvatlaydi.

Amaliy qism:

1. Kerakli vositalarni o'rnatish:

- Python (3.8+ versiyasi) va SQLAlchemy kutubxonasi o'rnatilganligiga ishonch hosil qiling. Agar ular o'rnatilmagan bo'lsa, buyruqni bajaring:

```
pip install sqlalchemy pandas
```

2. Ma'lumotlarni tayyorlash:

- Oldingi ishlarda yuklab olgan Titanic (`train.csv`) ma'lumotlar to'plamidan foydalaning.

3. Ma'lumotlar bilan ishlash:

- Yangi Python faylini yarating (masalan, `amaliy_4.py`) va quyidagi amallarni bajaring:

```
import pandas as pd
from sqlalchemy import create_engine

# Ma'lumotlar yuklash
data = pd.read_csv('train.csv')

# SQLite ma'lumotlar bazasini yaratish
engine = create_engine('sqlite:///titanic.db') # "titanic.db"
# Ma'lumotlar bazasi faylini yaratish
data.to_sql('passengers', engine, index=False,
if_exists='replace') # Ma'lumotlarni "passengers" jadvaliga
yuklash

# 1. SQL so'rovi: 30 yoshdan oshgan barcha yo'lovchilarni
tanlash
query = "SELECT * FROM passengers WHERE Age > 30"
result = pd.read_sql(query, engine)
print("30 yoshdan oshgan yo'lovchilar:")
print(result.head())

# 2. SQL so'rovi: omon qolgan ayollar sonini hisoblash
query = "SELECT COUNT(*) AS Survived_Females FROM passengers
WHERE Sex = 'female' AND Survived = 1"
```

```
result = pd.read_sql(query, engine)
print("\nTirik qolgan ayollar soni:")
print(result)
```

```
# 3. SQL so'rovi: ma'lumotlarni chipta narxi bo'yicha saralash
query = "SELECT * FROM passengers ORDER BY Fare DESC"
result = pd.read_sql(query, engine)
print ("\nChipta narxi bo'yicha saralangan n ma'lumotlar
(pasayish bo'yicha):")
print(result.head())
```

4. Dasturni ishga tushirish:

Faylni saqlash va uni terminal orqali ishga tushirish:

```
python amaliy_4.py
```

Natijada ko'rasiz:

- 30 yoshdan oshgan yo'lovchilar.
- Tirik qolgan ayollar soni.
- Chipta narxi bo'yicha saralangan ma'lumotlar.

Qo'shimcha materiallar:

- Rasmiy SQLAlchemy hujjatlari: [SQLAlchemy Documentation](#).
- SQL qo'llanmasi: [SQL Tutorial](#).
- Python da SQLite dan foydalanishga misollar: [SQLite with Python](#).

Amaliy vazifa:

Vazifa: № 4

Titanic (`train.csv`) ma'lumotlar to'plamidan foydalanish va SQLite ma'lumotlar bazasi uchun quyidagi amallarni bajarish:

1. Omon qolgan barcha erkaklarni tanlash uchun SQL so'rovini yozing.
2. Har bir kabina sinfidagi yo'lovchilarning o'rtacha yoshini hisoblash uchun SQL so'rovini yozing (`Pclass`).
3. Maksimal chipta narxini topish uchun SQL so'rovini yozing (`Fare`).