

Órai feladat

2025.09.17.

Időjárás Adatbázis Létrehozása és Python Kapcsolat

Időtartam: 75 perc

Cél:

A feladat során a következőket sajátítod el:

- **SQL alapok:** adatbázis létrehozása, táblák definiálása, adatok beszúrása és lekérdezése.
 - **Python és PostgreSQL kapcsolat:** A psycopg2 modul használatával kapcsolatot létesítesz az adatbázissal.
 - **OOP és tervezési minták alkalmazása:** SOLID, DRY, KISS, YAGNI elvek alapján felépített kód, melyben szerepel a Singleton, Abstract Factory és Strategy minta.
 - **Többféle SQL lekérdezés megvalósítása:** Legalább 10 különböző lekérdezést írsz meg Pythonban.
-

1. SQL Rész (PostgreSQL):

a. Adatbázis létrehozása

A következő parancs létrehozza az "idojaras_db" nevű adatbázist:

```
CREATE DATABASE idojaras_db;
```

b. Táblák létrehozása

Hozzuk létre az "idojaras" nevű táblát, amely az alábbi oszlopokat tartalmazza:

- id: egyedi azonosító (automatikusan növekvő érték)
- varos: a város neve
- homerseklet: a mért hőmérséklet érték
- paratartalom: a páratartalom százalékos értéke
- szelsebesseg: a szélsébség
- datum: a mérés dátuma, amely alapértelmezettként az aktuális időpontot veszi fel

```
CREATE TABLE idojaras (  
    id SERIAL PRIMARY KEY,  
    varos VARCHAR(50),  
    homerseklet FLOAT,  
    paratartalom INT,  
    szelsebesseg FLOAT,  
    datum TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

c. Dummy adatok beszúrása

Beszúrunk néhány mintaadatot az "idojaras" táblába:

```
INSERT INTO idojaras (varos, homerseklet, paratartalom, szelsebesseg)  
VALUES  
( 'Budapest', 15.2, 60, 10.5),  
( 'Debrecen', 17.5, 55, 8.2),  
( 'Szeged', 18.0, 50, 12.1),  
( 'Győr', 14.0, 65, 9.3),  
( 'Pécs', 16.7, 70, 7.5);
```

2. Python Rész:

A Python program feladata, hogy csatlakozzon az "idojaras_db" adatbázishoz, és különféle SQL műveleteket hajtson végre OOP elvek és tervezési minták segítségével.

Tervezési minták és OOP elvek:

- **Singleton minta:** Biztosítja, hogy csak egyetlen adatbázis kapcsolat létezzen az alkalmazásban.
 - **Abstract Factory/Absztrakt osztály:** Meghatározza az alapvető adatbázis műveleteket, melyeket a konkrét osztályok örökölnek.
 - **Strategy minta:** Különböző keresési stratégiákat valósít meg, így például a hőmérséklet vagy a páratartalom alapján történő keresés egyszerűen kicserélhető.
-

SQL rész:

1. Adatbázis létrehozása

A `CREATE DATABASE idojaras_db;` parancs létrehozza az adatbázist, ahol az időjárási adatokat fogjuk tárolni.

2. Táblák létrehozása

A `CREATE TABLE idojaras (...)` parancs segítségével egy olyan táblát hozunk létre, amely a város nevét, hőmérsékletét, páratartalmát, szélsébségét és a mérés dátumát tartalmazza.

3. Dummy adatok beszúrása

Az `INSERT INTO` parancsokkal feltöltjük a táblát mintaadatokkal, amelyekkel a Python kódot tesztelhetjük.

Python rész:

1. Singleton minta a kapcsolatnál

A `DatabaseConnection` osztály garantálja, hogy az alkalmazás futása során csak egy adatbázis kapcsolat jön létre. Ez segít az erőforrások hatékony kezelésében.

2. AbstractDatabase és IdojarasDatabase osztályok

Az `AbstractDatabase` absztrakt osztály meghatározza a közös adatbázis műveleteket, melyeket az `IdojarasDatabase` implementál, így biztosítva a DRY és SOLID elveket.

3. CRUD műveletek

Az `IdojarasDatabase` osztály metódusai segítségével elvégezheted az adatok beszúrását, lekérdezését, frissítését és törlését.

4. Strategy minta

A `SearchStrategy` absztrakt osztály, valamint a `SearchByTemperature` és `SearchByHumidity` osztályok lehetővé teszik, hogy különböző keresési módokat alkalmazz anélkül, hogy módosítanod kellene az alapvető kódot. Így rugalmasan bővíthető a keresési logika.

5. Lekérdezések

A Python kódban többféle lekérdezést hajtunk végre:

- Összes rekord lekérdezése
 - Lekérdezés város alapján
 - Hőmérséklet frissítése
 - Adatok lekérdezése hőmérséklet tartományban
 - Lekérdezés szélsőérték alapján
 - Adatok rendezése dátum szerint
 - Rekordok számolása, átlag hőmérséklet, maximum páratartalom kiszámítása
 - Stratégiai keresési módok alkalmazása
 - Rekord törlése
-