

Házi Feladat

Beadási határidő:

2025.09.28 23:59:59

A házifeladat megoldásait külön-külön fájlban mentsétek el, zippelve az alábbi mail címre küldhetitek el: vince.dulicz@gmail.com

Könyvesbolt adatbázis

Táblák

1. Authors (Szerzők)

- `author_id` (int, PK) – Szerző azonosító
- `first_name` (varchar) – Keresztnév
- `last_name` (varchar) – Vezetéknév
- `birth_year` (int) – Születési év
- `country` (varchar) – Ország

2. Books (Könyvek)

- `book_id` (int, PK) – Könyv azonosító
- `title` (varchar) – Könyv címe
- `author_id` (int, FK) – Kapcsolat az Authors táblával
- `genre` (varchar) – Műfaj
- `price` (decimal) – Ár

3. Sales (Eladások)

- `sale_id` (int, PK) – Eladás azonosító
- `book_id` (int, FK) – Kapcsolat a Books táblával
- `sale_date` (date) – Eladás dátuma
- `quantity` (int) – Eladott példányszám
- `total_amount` (decimal) – Teljes bevétel

Feladat: SQL lekérdezések

1. **Egyszerű lekérdezés:** Listázd az összes szerzőt.
2. **Szűrés:** Listázd azokat a könyveket, amelyek ára 3000 Ft felett van.
3. **Rendezés:** Listázd a könyveket ár szerint csökkenő sorrendben.
4. **Összegzés:** Számold meg, hány könyvet adott el a bolt.
5. **Csoportosítás:** Összesítsd a bevételt könyvenként.
6. **Összekapcsolás (JOIN):** Listázd a könyvek címét és a szerző nevét.
7. **Bonyolultabb JOIN + feltétel:** Listázd a szerzők nevét, akiknek a könyvei összesen több mint 50000 Ft bevételt hoztak.
8. **Allekérdezés:** Listázd azokat a könyveket, amelyek ára magasabb, mint az összes könyv átlagára.
9. **Top eladók:** Listázd az 5 legtöbbet eladott könyvet példányszám szerint.
10. **Drága könyvek szerzői:** Mutasd meg a szerzők nevét és a könyv címét, ahol a könyv ára 4000 Ft felett van.
11. **Év szerinti összesítés:** Összesítsd a bevételt könyvenként, de csoportosítva az eladás évére.
12. **Műfaj kedveltség:** Számold meg, hány könyvet adtak el műfajonként, és listázd a legtöbbet eladott műfajt legfelül.

Feladat: Könyvesbolt lekérdezések Pythonban

Készíts egy Python programot, ami:

- OOP elveket használ (osztályok, öröklődés, interfészek)
 - SOLID, KISS és DRY szabályokat követ
 - Singleton mintát alkalmaz az adatbázis kapcsolat kezelésére
 - Strategy mintát használ a különböző SQL lekérdezések végrehajtására
 - A lekérdezéseket YAML fájlban tárolja, onnan betöltve hajtja végre
-

1. Követelmények

1. **DatabaseConnection** – Singleton osztály az adatbázis kapcsolat kezelésére.
2. **QueryStrategy** – Absztrakt osztály/interface a lekérdezések végrehajtására.
3. **ConcreteQuery** – Több Strategy osztály, mindegyik egy SQL lekérdezést reprezentál (pl. ListAuthorsQuery, TopSellingBooksQuery stb.).
4. **QueryExecutor** – Egy osztály, ami betölti a lekérdezéseket YAML-ból, kiválasztja a megfelelő Strategy-t és végrehajtja.

- Tárold a SQL lekérdezéseket kulcs-érték párokként. Például:

```
list_authors: "SELECT * FROM Authors;"
```

- Legalább 8 SQL parancs legyen YAML-ban, az előző feladatok alapján.
 - Betöltéskor a Python Strategy mintát használva futtasd le a lekérdezéseket.
 - **SOLID:** Egy lekérdezés végrehajtása ne függjön másik lekérdezéstől (Single Responsibility)
 - **DRY:** Ne ismételd a kódot – közös logika legyen például a Strategy-ben
 - **KISS:** Ne legyen túl bonyolult az osztály struktúra
 - **Singleton:** Egyetlen adatbázis kapcsolat legyen az egész programban
 - **Strategy:** Különböző lekérdezések külön Strategy osztályokban legyenek
-

3. Működés

1. Betöltöd a YAML fájlt.
2. A QueryExecutor kiválaszt egy lekérdezést (pl. list_authors).
3. A Strategy végrehajtja az SQL parancsot a Singleton adatbázis kapcsolaton keresztül.
4. Az eredményt visszaadja, kiírja a konzolra vagy feldolgozza.

Dummy adatok

Authors

```
INSERT INTO Authors (author_id, first_name, last_name, birth_year, country) VALUES
(1, 'John', 'Smith', 1975, 'USA'),
(2, 'Anna', 'Kovács', 1980, 'Hungary'),
(3, 'Maria', 'Rossi', 1968, 'Italy'),
(4, 'Li', 'Wei', 1990, 'China'),
(5, 'Ahmed', 'Hassan', 1985, 'Egypt'),
(6, 'Sophie', 'Dubois', 1978, 'France'),
(7, 'Peter', 'Müller', 1982, 'Germany'),
(8, 'Yuki', 'Tanaka', 1995, 'Japan'),
(9, 'Carlos', 'Silva', 1970, 'Brazil'),
(10, 'Elena', 'Petrova', 1988, 'Russia'),
(11, 'Liam', 'Johnson', 1992, 'UK');
```

Books

```
INSERT INTO Books (book_id, title, author_id, genre, price) VALUES
(1, 'The Silent Forest', 1, 'Mystery', 1900.99),
(2, 'Love in Budapest', 2, 'Romance', 14000.50),
(3, 'Italian Recipes', 3, 'Cooking', 3400.00),
(4, 'Chinese Philosophy', 4, 'Philosophy', 2900.99),
(5, 'Desert Adventures', 5, 'Adventure', 7100.50),
(6, 'Paris Nights', 6, 'Romance', 4800.75),
(7, 'German History', 7, 'History', 1200.00),
(8, 'Sakura Dreams', 8, 'Fiction', 3600.80),
(9, 'Amazon Secrets', 9, 'Adventure', 5500.00),
(10, 'Russian Tales', 10, 'Fantasy', 2000.00),
(11, 'London Fog', 11, 'Mystery', 6700.50);
```

Sales

```
INSERT INTO Sales (sale_id, book_id, sale_date, quantity, total_amount) VALUES
(1, 1, '2025-01-05', 3, 59.97),
(2, 2, '2025-02-10', 5, 72.50),
(3, 3, '2025-03-12', 2, 48.00),
(4, 4, '2025-01-20', 1, 29.99),
(5, 5, '2025-04-15', 4, 86.00),
(6, 6, '2025-05-01', 6, 112.50),
(7, 7, '2025-03-30', 2, 44.00),
(8, 8, '2025-06-10', 3, 50.40),
(9, 9, '2025-07-08', 5, 125.00),
(10, 10, '2025-08-21', 2, 40.00),
(11, 11, '2025-09-15', 1, 17.50);
```