PDU 2022/2023

Praca domowa nr 5 (max. = 20 p.)

Prace domowe należy przesłać za pośrednictwem platformy `Moodle` przygotowany zgodnie z szablonem[1] plik z rozwiązanianimi. **Prowadzący grupę laboratoryjną określa czy rozwiązania mają być przesłane jako moduł języka Python `.py` albo notatnik Jupyter `.ipynb`.**

1 Zbiory danych

Ponownie będziemy pracować na uproszczonym zrzucie zanonimizowanych danych z serwisu https://travel.stackexchange.com/, który składa się z następujących ramek danych:

- Posts.csv.gz
- Users.csv.gz
- Comments.csv.gz

Uwaga: wykorzystujemy ramki danych z pracy domowej nr 3.

Przed przystąpieniem do rozwiązywania zadań przypomnij sobie znaczenie poszczególnych kolumn we wspomnianych ramkach danych, zob. https://ia600107.us.archive.org/27/items/stackexchange/readme.txt

Przykładowe wywołanie — ładowanie zbioru `Tags`:

```
import pandas as pd
import numpy as np

Posts = pd.read_csv("travel_stackexchange_com/Posts.csv.gz",
                    compression = 'gzip')
Posts.head()
```

Każdą z ramek danych należy wyeksportować do bazy danych SQLite przy użyciu wywołania metody `to_sql()` w klasie `pandas.DataFrame`. Dokładniej, pracę z bazą danych możemy przeprowadzić w następujący sposób.

```
import os, os.path
import sqlite3

baza = 'przyklad.db'               # sciezka dostępu do bazy danych:

conn = sqlite3.connect(baza)       # połączenie do bazy danych

Comments.to_sql("Comments", conn) # importujemy ramkę danych do bazy danych
Posts.to_sql("Posts", conn)
Users.to_sql("Users", conn)

#
pd.read_sql_query("""
```

[1] Szablony dostępne są na 'Moodle'.

```
                  Zaptanie SQL
                  """, conn)
# ...
# rozwiązania zadan
# po skończonej pracy zamykamy połączenie
#
conn.close()
```

W szczególności należy zagwarantować, że w każdym przypadku wynik jest klasy `DataFrame`, a nie `Series`.

Uwaga: Nazwy ramek danych po wczytaniu zbiorów powinny wyglądać następująco: `Badges`, `Comments`, `Tags`, `Posts`, `Users`, `Votes`, `PostLinks`.

2 Informacje ogólne

Rozwiąż poniższe zadania przy użyciu wywołań funkcji i metod z pakietu `pandas`. Każdemu z **5** poleceń SQL powinny odpowiadać dwa równoważne sposoby ich implementacji, kolejno:

1. wywołanie `pandas.read_sql_query("""zapytanie SQL""")`;
2. wywołanie ciągu „zwykłych” metod i funkcji z pakietu `pandas`.

Upewnij się, że zwracane wyniki są ze sobą tożsame (ewentualnie z dokładnością do permutacji wierszy i kolumn wynikowych ramek danych), por. np. metodę `.equals()` z pakietu `pandas`.

3 Zadania do rozwiązania

```
--- 1)
SELECT Location, SUM(UpVotes) as TotalUpVotes
FROM Users
WHERE Location != ''
GROUP BY Location
ORDER BY TotalUpVotes DESC
LIMIT 10
```

```
--- 2)
SELECT STRFTIME('%Y', CreationDate) AS Year, STRFTIME('%m', CreationDate) AS Month,
       COUNT(*) AS PostsNumber, MAX(Score) AS MaxScore
FROM Posts
WHERE PostTypeId IN (1, 2)
GROUP BY Year, Month
HAVING PostsNumber > 1000
```

```
--- 3)
SELECT Id, DisplayName, TotalViews
FROM (
        SELECT OwnerUserId, SUM(ViewCount) as TotalViews
        FROM Posts
        WHERE PostTypeId = 1
        GROUP BY OwnerUserId
     ) AS Questions
JOIN Users
ON Users.Id = Questions.OwnerUserId
ORDER BY TotalViews DESC
LIMIT 10
```

```sql
--- 4)
SELECT DisplayName, QuestionsNumber, AnswersNumber, Location, Reputation, UpVotes, DownVotes
FROM (
        SELECT *
        FROM (
                SELECT COUNT(*) as AnswersNumber, OwnerUserId
                FROM Posts
                WHERE PostTypeId = 2
                GROUP BY OwnerUserId
             ) AS Answers
        JOIN
             (
                SELECT COUNT(*) as QuestionsNumber, OwnerUserId
                FROM Posts
                WHERE PostTypeId = 1
                GROUP BY OwnerUserId
              ) AS Questions
        ON Answers.OwnerUserId = Questions.OwnerUserId
        WHERE AnswersNumber > QuestionsNumber
        ORDER BY AnswersNumber DESC
        LIMIT 5
    ) AS PostsCounts
JOIN Users
ON PostsCounts.OwnerUserId = Users.Id
```

```sql
--- 5)
SELECT Title, CommentCount, ViewCount, CommentsTotalScore, DisplayName, Reputation, Location
FROM (
        SELECT Posts.OwnerUserId, Posts.Title, Posts.CommentCount, Posts.ViewCount,
               CmtTotScr.CommentsTotalScore
        FROM (
                SELECT PostId, SUM(Score) AS CommentsTotalScore
                FROM Comments
                GROUP BY PostId
             ) AS CmtTotScr
        JOIN Posts ON Posts.Id = CmtTotScr.PostId
        WHERE Posts.PostTypeId=1
    ) AS PostsBestComments
JOIN Users ON PostsBestComments.OwnerUserId = Users.Id
ORDER BY CommentsTotalScore DESC
LIMIT 10
```