Анализ базы данных книг (SQL запросы)

выпускной проект студента когорты DA61
Алиева Рустама

Задача проекта: проанализировать базу данных книг с целью получения важной информации о книгах, издательствах, авторах, пользовательских обзорах книг для формирования предложения по созданию нового продукта.

Описание данных

Таблица books (содержит данные о книгах):

- book_id — идентификатор книги;
- author_id — идентификатор автора;
- title — название книги;
- num_pages — количество страниц;
- publication_date — дата публикации книги;
- publisher_id — идентификатор издателя.

Таблица authors (содержит данные об авторах):

- author_id — идентификатор автора;
- author — имя автора.

Таблица publishers (содержит данные об издательствах):

- publisher_id — идентификатор издательства;
- publisher — название издательства;

Таблица ratings (содержит данные о пользовательских оценках книг):

- rating_id — идентификатор оценки;
- book_id — идентификатор книги;
- username — имя пользователя, оставившего оценку;
- rating — оценка книги.

Таблица reviews (содержит данные о пользовательских обзорах):

- review_id — идентификатор обзора;
- book_id — идентификатор книги;
- username — имя автора обзора;
- text — текст обзора.

Ввод [1]:
```
# импортируем библиотеки
import pandas as pd
from sqlalchemy import text, create_engine
```

Ввод [2]:
```
# устанавливаем параметры
db_config = {'user': 'praktikum_student', # имя пользователя
'pwd': '***********', # пароль скрыт
'host': 'rc1b-wcoijxj3yxfsf3fs.mdb.yandexcloud.net',
'port': 6432, # порт подключения
'db': 'data-analyst-final-project-db'} # название базы данных
connection_string = 'postgresql://{user}:{pwd}@{host}:{port}/{db}'.format(**db_config

# сохраняем коннектор
engine = create_engine(connection_string, connect_args={'sslmode':'require'})
```

1 Загрузка и знакомство с данными

Ввод [3]:
```
# используем микрофункцию для загрузки и первичного знакомства с данными
def hello_dolly(data):
    query = 'SELECT * FROM '+data
    con=engine.connect()
    df = pd.io.sql.read_sql(sql=text(query), con = con)
    display(df.info())
    print('Количество дубликатов:', df.duplicated().sum())
    display(df.head(5))
```

Ввод [4]:
```
# books
hello_dolly('books')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   book_id           1000 non-null   int64 
 1   author_id         1000 non-null   int64 
 2   title             1000 non-null   object
 3   num_pages         1000 non-null   int64 
 4   publication_date  1000 non-null   object
 5   publisher_id      1000 non-null   int64 
dtypes: int64(4), object(2)
memory usage: 47.0+ KB

None

Количество дубликатов: 0
```

	book_id	author_id	title	num_pages	publication_date	publisher_id
0	1	546	'Salem's Lot	594	2005-11-01	93
1	2	465	1 000 Places to See Before You Die	992	2003-05-22	336
2	3	407	13 Little Blue Envelopes (Little Blue Envelope...	322	2010-12-21	135
3	4	82	1491: New Revelations of the Americas Before C...	541	2006-10-10	309
4	5	125	1776	386	2006-07-04	268

Таблица содержит информацию о 1000 книг и из данных, пропусков нет, дубликатов тоже

Ввод [5]:
```
# authors
hello_dolly('authors')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 636 entries, 0 to 635
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   author_id  636 non-null    int64 
 1   author     636 non-null    object
dtypes: int64(1), object(1)
memory usage: 10.1+ KB

None

Количество дубликатов: 0
```

	author_id	author
0	1	A.S. Byatt
1	2	Aesop/Laura Harris/Laura Gibbs
2	3	Agatha Christie

Как и обещано, тут информация об авторах - всего 636 замечательных людей (коллективов), пропусков нет, дублей нет.

Ввод [6]:
```
# ratings
hello_dolly('ratings')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6456 entries, 0 to 6455
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   rating_id  6456 non-null   int64 
 1   book_id    6456 non-null   int64 
 2   username   6456 non-null   object
 3   rating     6456 non-null   int64 
dtypes: int64(3), object(1)
memory usage: 201.9+ KB

None

Количество дубликатов: 0
```

	rating_id	book_id	username	rating
0	1	1	ryanfranco	4

Читатели оценили книги почти 6,5тыс. раз, датасет также без пропусков и дублей.

Ввод [7]:
```
# reviews
hello_dolly('reviews')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2793 entries, 0 to 2792
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   review_id  2793 non-null   int64 
 1   book_id    2793 non-null   int64 
 2   username   2793 non-null   object
 3   text       2793 non-null   object
dtypes: int64(2), object(2)
memory usage: 87.4+ KB

None

Количество дубликатов: 0
```

	review_id	book_id	username	text
0	1	1	brandtandrea	Mention society tell send professor analysis. ...

Эти же читатели накатали отзывов почти 2,8 тыс. штук. Ещё и на английском - какие молодцы. Пропусков нет, дублей нет.

Ввод [8]:
```
# publishers
hello_dolly('publishers')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 340 entries, 0 to 339
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   publisher_id  340 non-null    int64 
 1   publisher     340 non-null    object
dtypes: int64(1), object(1)
memory usage: 5.4+ KB

None

Количество дубликатов: 0
```

	publisher_id	publisher
0	1	Ace
1	2	Ace Book
2	3	Ace Books

Датасет по издателям содержит информацию о 340 замечательных фирмах, явных дублей нет, пропусков тоже.

2 Анализ данных

2.1 Задание 1: определить количество книг, которые вышли после 1 января 2000 года

Ввод [9]:
```
query = '''SELECT COUNT (book_id)
           FROM books
           WHERE publication_date > '2000-01-01'
       ;'''

con=engine.connect()

books_count_after_2000_01_01 = pd.io.sql.read_sql(sql=text(query), con = con)
```

Ввод [10]:
```
books_count_after_2000_01_01
```

Out[10]:

	count
0	819

В данных есть информация о 819 книгах, вышедших после начала 2000 года.

2.2 Задание 2: сделать расчет количества обзоров и средней оценки для каждой книги

Ввод [11]:
```
query = '''WITH avg_rat as(SELECT book_id
                                  ,ROUND(AVG(rating),3) as average_rating
                           FROM ratings rt
                           GROUP BY(rt.book_id)
                          ),
               rev_cnt as(SELECT book_id
                                  ,COUNT(review_id) as review_count
                           FROM reviews rv
                           GROUP BY(rv.book_id)
                          )
           SELECT title
                 ,average_rating
                 ,review_count
           FROM books b
           LEFT JOIN avg_rat ON b.book_id=avg_rat.book_id
           LEFT JOIN rev_cnt ON b.book_id=rev_cnt.book_id
           ORDER BY (review_count, average_rating) DESC
       ;'''

con=engine.connect()

review_count_av_rating_per_book = pd.io.sql.read_sql(sql=text(query), con = con)
```

Ввод [12]:
```
review_count_av_rating_per_book.head(10)
```

Out[12]:

	title	average_rating	review_count
0	The Cat in the Hat and Other Dr. Seuss Favorites	5.000	NaN
1	Disney's Beauty and the Beast (A Little Golden...	4.000	NaN
2	Leonardo's Notebooks	4.000	NaN
3	Essential Tales and Poems	4.000	NaN
4	Anne Rice's The Vampire Lestat: A Graphic Novel	3.667	NaN
5	The Natural Way to Draw	3.000	NaN
6	Twilight (Twilight #1)	3.663	7.0
7	Harry Potter and the Prisoner of Azkaban (Harr...	4.415	6.0
8	Harry Potter and the Chamber of Secrets (Harry...	4.288	6.0
9	The Book Thief	4.264	6.0

Если считать именно для каждой из 1000 книг, то максимальные средние рейтинги у книг без обзоров...

2.3 Задание 3: определить издательство, которое выпустило наибольшее число книг толще 50 страниц (исключаем брошюры)

Ввод [13]:
```
query = '''SELECT publisher
                  ,COUNT(b.book_id) books_count
           FROM publishers p
           RIGHT JOIN books b ON p.publisher_id=b.publisher_id
           WHERE b.num_pages > 50
           GROUP BY(publisher)
           ORDER BY(COUNT(b.book_id)) DESC
           LIMIT 1
       ;'''

con=engine.connect()

leader_publisher = pd.io.sql.read_sql(sql=text(query), con = con)
```

Ввод [14]:
```
leader_publisher
```

Out[14]:

	publisher	books_count
0	Penguin Books	42

Издательство-лидер - Penguin Books! Это издательство выпустило 42 книги толще 50 страниц.

2.4 Задание 4: Определить автора с самой высокой средней оценкой книг (книги с 50 и более оценками)

Ввод [16]:
```
query = '''SELECT author
                ,AVG(rt.rating) avg_rating
           FROM authors a
           JOIN books b ON a.author_id=b.author_id
           JOIN ratings rt ON b.book_id=rt.book_id
           WHERE rt.book_id IN (SELECT book_id
                                FROM ratings rt
                                GROUP BY(book_id)
                                HAVING(COUNT(rating_id)>=50)
                               )
           GROUP BY(author)
           ORDER BY(avg_rating) DESC
           LIMIT 1
       ;'''

con=engine.connect()

leader_author = pd.io.sql.read_sql(sql=text(query), con = con)
```

Ввод [17]:
```
leader_author
```

Out[17]:

	author	avg_rating
0	J.K. Rowling/Mary GrandPré	4.287097

Ожидамый лидер, автор бестселлеров Дж.Роулинг... Её книги - всегда нарасхват!

2.5 Задание 5: Посчитать среднее количество обзоров от пользователей, которые поставили больше 48 оценок

Ввод [18]:
```
query = '''WITH user_48 as (SELECT COUNT(review_id) cnt
                            FROM reviews rv
                            WHERE rv.username IN (SELECT rt.username
                                                  FROM ratings rt
                                                  GROUP by(rt.username)
                                                  HAVING COUNT(rt.rating_id) > 48
                                                  )
                            GROUP BY rv.username
                            )
           SELECT ROUND(AVG(cnt), 0)
           FROM user_48
      ; '''

con=engine.connect()

number_review_48 = pd.io.sql.read_sql(sql=text(query), con = con)
number_review_48
```

Out[18]:

	round
0	24.0

Вот молодцы! И читать успевают много и оценок поставили больше 48, ещё и в среднем 24 обзора выложили - настоящие ценители!

3 Выводы

Целью проекта было получение ценной информации из базы данных по книгам.
В ходе проекта были проведены работы:

- знакомство с таблицами БД, поиск в них ошибок
- непосредственно анализ:
  - посчитано количество книг, выпущенных после 01.01.2000
  - расчитано количество обзоров и средняя оценка для всех книг
  - определено издательство, которое выпустило наибольшее число книг
  - найден автор с самой высокой средней оценкой книг
  - вычислено среднее количество обзоров от пользователей, которые поставили больше 48 оценок.