Анализ базы данных для сервиса чтения книг по подписке.

Содержание

1 Цель исследования

Проанализировать базу данных крупного сервиса для чтения книг по подписке. В базе данных содержится информация о книгах, издательствах, авторах, а также пользовательские обзоры книг. Эти данные помогут сформулировать ценностное предложение для нового продукта.

1.1 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.

1.2 План исследования:

1. **Исследование данных - проверка на пропуски, дубликаты.**

2. **Ответы на вопросы исследования:**

   Посчитайте, сколько книг вышло после 1 января 2000 года;
   Для каждой книги посчитайте количество обзоров и среднюю оценку;
   Определите издательство, которое выпустило наибольшее число книг толще 50 страниц — так вы исключите из анализа брошюры;
   Определите автора с самой высокой средней оценкой книг — учитывайте только книги с 50 и более оценками;
   Посчитайте среднее количество обзоров от пользователей, которые поставили больше 50 оценок.

3. **Общие выводы и рекомендации.**

2 Исследование данных - проверка на пропуски, дубликаты

к содержанию

Импортируем нужные библиотеки и создадим подключение к базе данных

In [1]:
```
# импортируем библиотеки
import pandas as pd
from sqlalchemy import create_engine
```

In [2]:
```
# устанавливаем параметры
db_config = {'user': 'praktikum_student', # имя пользователя
'pwd': '******', # пароль
'host': '******',
'port': ****, # порт подключения
'db': '******'} # название базы данных
connection_string = 'postgresql://{}:{}@{}:{}/{}'.format(db_config['user'],
 db_config['pwd'],
 db_config['host'],
 db_config['port'],
 db_config['db'])
```

In [3]:
```
# сохраняем коннектор
engine = create_engine(connection_string, connect_args={'sslmode':'require'})
```

Выведем на экран и исследуем датасет с книгами

In [4]:
```
# формируем sql-запрос.
query = ''' SELECT *
            FROM books
        '''
```

In [5]:
```
books = pd.io.sql.read_sql(query, con = engine)
books.head()
```

Out[5]:

	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

In [6]:
```
books.info()
```

```
<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
```

In [7]:
```
type(books['publication_date'])
```

Out[7]:
```
pandas.core.series.Series
```

Пропусков в датасете нет, наименования и типы данных корректные. Проверим есть ли явные и неявные дубликаты.

In [8]:
```
books.duplicated().sum()
```

Out[8]:
```
0
```

In [9]:
```
books[['author_id', 'title', 'num_pages', 'publication_date', 'publisher_id']].duplicate
```

Out[9]:
```
0
```

Дубликатов нет. Посмотрим на статистики датасета, чтобы увидеть максимальные и минимальные цифры количества книг и страниц, идентификаторов авторов и издательств, и сравнить их в дальнейшем с соответствующими таблицами

In [10]:
```
books.describe()
```

Out[10]:

	book_id	author_id	num_pages	publisher_id
count	1000.000000	1000.000000	1000.00000	1000.000000
mean	500.500000	320.417000	389.11100	171.270000
std	288.819436	181.620172	229.39014	99.082685
min	1.000000	1.000000	14.00000	1.000000
25%	250.750000	162.750000	249.00000	83.000000
50%	500.500000	316.500000	352.00000	177.500000
75%	750.250000	481.000000	453.00000	258.000000
max	1000.000000	636.000000	2690.00000	340.000000

Итак, у нас 1000 книг, 636 авторов, 340 издательств. Книги содержат от 14 до 2690 страниц - есть брошюры и объемные тома.

Посмотрим на датасет с авторами.

In [11]:
```
query = '''SELECT *
           FROM authors
        '''
```

In [12]:
```
authors = pd.io.sql.read_sql(query, con = engine)
authors.head()
```

Out[12]:

	author_id	author
0	1	A.S. Byatt
1	2	Aesop/Laura Harris/Laura Gibbs
2	3	Agatha Christie
3	4	Alan Brennert
4	5	Alan Moore/David Lloyd

In [13]:
```
authors.info()
```

```
<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
```

In [14]:
```
authors['author'].duplicated().sum()
```

Out[14]:
```
0
```

Датасет без пропусков и дубликатов, наименования и типы данных корректные, содержит 636 авторов, как и

датасет с книгами.

Посмотрим на датасет с издательствами.

In [15]:
```
query = '''SELECT *
           FROM publishers
        '''
```

In [16]:
```
publishers = pd.io.sql.read_sql(query, con = engine)
publishers.head()
```

Out[16]:

	publisher_id	publisher
0	1	Ace
1	2	Ace Book
2	3	Ace Books
3	4	Ace Hardcover
4	5	Addison Wesley Publishing Company

In [17]:
```
publishers.info()
```

```
<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
```

In [18]:
```
publishers['publisher'].duplicated().sum()
```

Out[18]: 0

Датасет без пропусков и дубликатов, наименования и типы данных корректные, содержит 340 издательств, как и датасет с книгами.

Посмотрим на датасет с оценками.

In [19]:
```
query = '''SELECT *
           FROM ratings
        '''
```

In [20]:
```
ratings = pd.io.sql.read_sql(query, con = engine)
ratings.head()
```

Out[20]:

	rating_id	book_id	username	rating
0	1	1	ryanfranco	4
1	2	1	grantpatricia	2
2	3	1	brandtandrea	5
3	4	2	lorichen	3
4	5	2	mariokeller	2

In [21]:
```
ratings.info()
```

```
<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
```

In [22]:
```
ratings.duplicated().sum()
```

Out[22]:
```
0
```

In [23]:
```
ratings[['book_id', 'username', 'rating']].duplicated().sum()
```

Out[23]:
```
0
```

Датасет без пропусков и дубликатов, наименования и типы данных корректные, содержит 6456 оценок.

Посмотрим на датасет с отзывами.

In [24]:
```
query = '''SELECT *
           FROM reviews
        '''
```

In [25]:
```
reviews = pd.io.sql.read_sql(query, con = engine)
reviews.head()
```

Out[25]:

	review_id	book_id	username	text
0	1	1	brandtandrea	Mention society tell send professor analysis. ...
1	2	1	ryanfranco	Foot glass pretty audience hit themselves. Amo...
2	3	2	lorichen	Listen treat keep worry. Miss husband tax but ...
3	4	3	johnsonamanda	Finally month interesting blue could nature cu...
4	5	3	scotttamara	Nation purpose heavy give wait song will. List...

In [26]:
```
reviews.info()
```

```
<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
```

In [27]:
```
reviews.duplicated().sum()
```

Out[27]: 0

In [28]:
```
reviews[['book_id', 'username', 'text']].duplicated().sum()
```

Out[28]: 0

Датасет без пропусков и дубликатов, наименования и типы данных корректные, содержит 2793 отзыва.

Вывод: Все датасеты корректные, можно переходить к анализу данных.

3 Ответы на вопросы исследования

к содержанию

3.1 Задача 1:

Посчитайте, сколько книг вышло после 1 января 2000 года

In [29]:
```
query = '''
        SELECT COUNT(book_id) AS books_after_2000
        FROM books
        WHERE publication_date > '2000-01-01'
        '''
pd.io.sql.read_sql(query, con=engine)
```

Out[29]:

	books_after_2000
0	819

Вывод: После 1 января 2000 года вышло 819 книг из 1000. Подборка в нашей библиотеке достаточно свежая.

3.2 Задача 2:

Для каждой книги посчитайте количество обзоров и среднюю оценку

In [30]:
```
# отсортируем книги по количеству обзоров
query = '''
        SELECT b.title AS book_title,
               COUNT(DISTINCT rw.review_id) AS review_count,
               AVG(rt.rating) AS average_rating
        FROM books AS b
        LEFT JOIN ratings AS rt ON rt.book_id=b.book_id
        LEFT JOIN reviews AS rw ON rw.book_id=b.book_id
        GROUP BY b.book_id
        ORDER BY review_count DESC,
                 average_rating DESC
        '''
pd.io.sql.read_sql(query, con=engine)
```

Out[30]:

	book_title	review_count	average_rating
0	Twilight (Twilight #1)	7	3.662500
1	Harry Potter and the Prisoner of Azkaban (Harr...	6	4.414634
2	Harry Potter and the Chamber of Secrets (Harry...	6	4.287500
3	The Book Thief	6	4.264151
4	The Glass Castle	6	4.206897
...	...	...	...
995	Disney's Beauty and the Beast (A Little Golden...	0	4.000000
996	Leonardo's Notebooks	0	4.000000
997	Essential Tales and Poems	0	4.000000
998	Anne Rice's The Vampire Lestat: A Graphic Novel	0	3.666667
999	The Natural Way to Draw	0	3.000000

1000 rows × 3 columns

In [35]:
```
# отсортируем книги по средней оценке
query = '''
        SELECT b.title AS book_title,
               COUNT(DISTINCT rw.review_id) AS review_count,
               AVG(rt.rating) AS average_rating
        FROM books AS b
        LEFT JOIN ratings AS rt ON rt.book_id=b.book_id
        LEFT JOIN reviews AS rw ON rw.book_id=b.book_id
        GROUP BY b.book_id
        ORDER BY average_rating DESC,
                 review_count DESC
        '''
pd.io.sql.read_sql(query, con=engine)
```

Out[35]:

	book_title	review_count	average_rating
0	A Dirty Job (Grim Reaper #1)	4	5.00
1	School's Out—Forever (Maximum Ride #2)	3	5.00
2	Moneyball: The Art of Winning an Unfair Game	3	5.00
3	Arrows of the Queen (Heralds of Valdemar #1)	2	5.00
4	Wherever You Go There You Are: Mindfulness Me...	2	5.00
...	...	...	...
995	The World Is Flat: A Brief History of the Twen...	3	2.25
996	Drowning Ruth	3	2.00
997	His Excellency: George Washington	2	2.00
998	Junky	2	2.00
999	Harvesting the Heart	2	1.50

1000 rows × 3 columns

Вывод: Мы можем видеть книги, отсортированные по количеству обзоров и по средней оценке, и выбирать наиболее или наименее популярные. Это может помочь нашим читателям выбирать книги.

3.3 Задача 3:

Определите издательство, которое выпустило наибольшее число книг толще 50 страниц — так вы исключите из анализа брошюры

In [32]:
```
query = '''
        SELECT pb.publisher,
               COUNT(b.book_id) AS books_count
        FROM books AS b
        LEFT JOIN publishers AS pb ON pb.publisher_id=b.publisher_id
        WHERE b.num_pages > 50
        GROUP BY pb.publisher
        ORDER BY books_count DESC
        LIMIT 1
        '''
pd.io.sql.read_sql(query, con=engine)
```

Out[32]:

	publisher	books_count
0	Penguin Books	42

Вывод: Издательство, которое выпустило наибольшее число книг толще 50 страниц - Penguin Books, оно выпустило 42 таких книги. Возможно, за новинками следует прежде всего обращаться к ним.

3.4 Задача 4:

Определите автора с самой высокой средней оценкой книг — учитывайте только книги с 50 и более оценками

In [33]:
```
query = '''
        WITH
        books_50 AS (SELECT b.book_id,
                            b.author_id,
                           COUNT(rt.book_id)
                    FROM books AS b
                    LEFT JOIN ratings AS rt ON rt.book_id=b.book_id
                    GROUP BY b.book_id
                    HAVING COUNT(rt.book_id)>=50)

        SELECT a.author,
               AVG(rt.rating) AS average_rating
        FROM books_50 AS b_50
        LEFT JOIN authors AS a ON a.author_id=b_50.author_id
        LEFT JOIN ratings AS rt ON rt.book_id=b_50.book_id
        GROUP BY a.author
        ORDER BY average_rating DESC
        LIMIT 1

        '''
pd.io.sql.read_sql(query, con=engine)
```

Out[33]:

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

Вывод: Автор с самой высокой средней оценкой книг (учитывая только книги с 50 и более оценками) - J.K. Rowling/Mary GrandPré

3.5 Задача5:

Посчитайте среднее количество обзоров от пользователей, которые поставили больше 50 оценок

In [34]:
```
query = '''
        WITH
        users AS (SELECT username,
                         COUNT(rating)
                  FROM ratings
                  GROUP BY username
                  HAVING COUNT(rating)>50)

        SELECT ROUND(AVG(count)) AS average_rewiews
        FROM (SELECT u.username,
                     COUNT(rw.review_id)
              FROM users AS u
              LEFT JOIN reviews AS rw ON rw.username=u.username
              GROUP BY u.username) AS rw_count
        '''
pd.io.sql.read_sql(query, con=engine)
```

Out[34]:

	average_rewiews
0	24.0

Вывод: В базе есть очень активные пользователи - они поставили больше 50 оценок и при этом написали более 24 обзоров.

4 Общие выводы и рекомендации

к содержанию

Мы исследовали базу данных крупного сервиса для чтения книг по подписке. Проверили данные на пропуски и дубликаты, а потом ответили на вопросы исследования и получили следующие результаты:

1. После 1 января 2000 года было издано 819 книг из 1000. Подборка в нашей библиотеке достаточно свежая. Но, возможно, читатели захотят увидеть и классические произведения в нашей библиотеке. Можно расширять ассортимент в эту сторону.

2. Мы можем сортировать книги по количеству обзоров и по средней оценке, и выбирать наиболее или наименее популярные. Это может помочь нашим читателям выбирать книги, а сервису - составлять рекомендации.

3. Издательство, которое выпустило наибольшее число книг толще 50 страниц - Penguin Books, оно выпустило 42 таких книги. Возможно, за новинками следует прежде всего обращаться к ним.

4. Автор с самой высокой средней оценкой книг (учитывая только книги с 50 и более оценками) - J.K. Rowling. Ее средняя оценка 4,3. Похоже, истории про Гарри Поттера и Фантастических тварей мало кого оставляют равнодушным. А учитывая отличную экранизацию - тем более. Это можно учесть в рекомендациях и рекламных акциях.

5. В базе есть очень активные пользователи - они поставили больше 50 оценок и при этом написали более 24 обзоров. Можно разработать систему поощрений для таких пользователей, чтобы стимулировать написание обзоров и выставление оценок.