Проект выполнил: Алексей Становой, в рамках учебного курса "Аналитик данных" Yandex.Practicum Москва, Февраль 2021г. +7 (962) 985-51-77 Телефон - WhatsApp - Telegram - e-mail: a.v.stanovoy@yandex.ru

Проект

Сервис для чтения книг.

Цель:

Проанализировать информацию из базы данных сервиса для чтения книг, ответив на поставленные вопросы:

```
- Сколько книг вышло после 1 января 2000 года?
- Какое количество обзоров и какая средняя оценка для каждой книги?
- Какое издательство выпустило наибольшее число книг толще 50 страниц?
- Кто из авторов имеет самую высокую среднюю оценку книг, учитывая только книги с 50 и более оценками?
- Какое количество обзоров пишут пользователи, поставившие больше 50 оценок?
```

Содержание:

1. Описание данных

Наверх

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 на книги:

- `review_id` — идентификатор обзора;
- `book_id` — идентификатор книги;
- `username` — имя пользователя, написавшего обзор;
- `text` — текст обзора.

2. Загрузка данных.

Наверх

Подключение к базе данных.

In [1]:
```
# импорт библиотек
import pandas as pd
from sqlalchemy import create_engine
from IPython.display import display

# устанавка параметров
db_config = {'user': 'praktikum_student', # имя пользователя
             'pwd': 'Sdf4$2;d-d30pp', # пароль
             'host': 'rc1b-wcoijxj3yxfsf3fs.mdb.yandexcloud.net',
             'port': 6432, # порт подключения
```

```
            'db': 'data-analyst-final-project-db'} # название базы данных
connection_string = 'postgresql://{}:{}@{}:{}/{}'.format(db_config['user'],
                                                         db_config['pwd'],
                                                         db_config['host'],
                                                         db_config['port'],
                                                         db_config['db'])
# сохранение коннектора
engine = create_engine(connection_string, connect_args={'sslmode':'require'})
```

Таблица `books`

In [2]:
```
# Извлечение таблицы
query='''
    SELECT * FROM books
    '''
# Вывод 5 первых строк
display(pd.io.sql.read_sql(query, con = engine).head())
# Вывод общей информации о таблице
display(pd.io.sql.read_sql(query, con = engine).info())
```

	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

```
<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
```

Таблица `authors`

In [3]:
```
# Извлечение таблицы
query='''
    SELECT * FROM authors
    '''
# Вывод 5 первых строк
display(pd.io.sql.read_sql(query, con = engine).head())
```

```
# Вывод общей информации о таблице
display(pd.io.sql.read_sql(query, con = engine).info())
```

	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

```
<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
```

Таблица `publishers`

In [4]:
```
# Извлечение таблицы
query='''
    SELECT * FROM publishers
    '''
# Вывод 5 первых строк
display(pd.io.sql.read_sql(query, con = engine).head())
# Вывод общей информации о таблице
display(pd.io.sql.read_sql(query, con = engine).info())
```

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

```
<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
```

Таблица `ratings`

In [5]:
```
# Извлечение таблицы
query='''
    SELECT * FROM ratings
    '''
# Вывод 5 первых строк
display(pd.io.sql.read_sql(query, con = engine).head())
# Вывод общей информации о таблице
display(pd.io.sql.read_sql(query, con = engine).info())
```

	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

```
<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
```

Таблица `reviews`

In [6]:
```
# Извлечение таблицы
query='''
    SELECT * FROM reviews
    '''
# Вывод 5 первых строк
display(pd.io.sql.read_sql(query, con = engine).head())
# Вывод общей информации о таблице
display(pd.io.sql.read_sql(query, con = engine).info())
```

	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...

	review_id	book_id	username	text
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...

```
<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
```

3. Анализ данных

Наверх

3.1. Количество книг вышедших после 1 января 2000 года.

In [7]:
```
# Подсчет количества уникальных названий книг с условием по полю "publication_date"
query='''
    SELECT  COUNT(DISTINCT book_id) AS cnt_books
    FROM books
    WHERE CAST(publication_date AS date) > '2000-01-01';
    '''
# Вывод результата
pd.io.sql.read_sql(query, con = engine)
```

Out[7]:

	cnt_books
0	819

ВЫВОД:

База данных содержит информацию о 819 книгах, вышедших в свет после первого января 2000 года.

3.2. Подсчет количества обзоров и средней оценки для каждой книги.

In [8]:
```
# Для решения задачи добавляются 2 подзапроса, в которых идет подсчет количества обзоров и средней оценки.
query='''
    SELECT
        books.book_id AS book_id,
```

```
        books.title AS books_title,
        Sub.cnt_reviews AS cnt_reviews,
        Sub1.average_rating AS average_rating
    FROM
       books
    LEFT JOIN (SELECT
                    reviews.book_id,
                    COUNT(reviews.review_id) AS cnt_reviews
                FROM
                    reviews
                GROUP BY
                    reviews.book_id) AS Sub ON Sub.book_id = books.book_id
    LEFT JOIN  (SELECT
                    ratings.book_id,
                    ROUND(AVG(ratings.rating),1) AS average_rating
                FROM
                    ratings
                GROUP BY
                    ratings.book_id) AS Sub1 ON Sub1.book_id = books.book_id
    GROUP BY
        books.book_id,
        books.title,
        Sub.cnt_reviews,
        Sub1.average_rating
    ORDER BY
        book_id;

    '''
# Вывод пяти первых строк результата
display(pd.io.sql.read_sql(query, con = engine).head())
```

	book_id	books_title	cnt_reviews	average_rating
0	1	'Salem's Lot	2.0	3.7
1	2	1 000 Places to See Before You Die	1.0	2.5
2	3	13 Little Blue Envelopes (Little Blue Envelope...	3.0	4.7
3	4	1491: New Revelations of the Americas Before C...	2.0	4.5
4	5	1776	4.0	4.0

ВЫВОД: Книги различаются по количеству отзывов и средней оценке.

3.3. **Издательство, которое выпустило наибольшее число книг толще 50 страниц (т.е. без учета брошюр).**

In [9]:
```
# Необходимая информация из таблицы "publishers" добавляется методом JOIN.
# Сортировка ORDER BY по убыванию покажет лидеров в Топе.
query='''
    SELECT
        publishers.publisher AS publisher,
```

```
        COUNT(books.book_id) AS cnt_books_more50
    FROM
        books
    LEFT JOIN publishers ON publishers.publisher_id = books.publisher_id
    WHERE
        books.num_pages > 50
    GROUP BY
        publishers.publisher
    ORDER BY
        cnt_books_more50 DESC
    LIMIT 1;
    '''
# Вывод результата
pd.io.sql.read_sql(query, con = engine)
```

Out[9]:

	publisher	cnt_books_more50
0	Penguin Books	42

ВЫВОД:

Издательством, выпустившим наибольшее количество книг толще 50 страниц, является Penguin Books с 42 книгами.

3.4. **Автор с самой высокой средней оценкой книг — учитываются только книги с 50 и более оценками.**

In [10]:

```
# В подзапросе нужно сделать подсчет количества оценок для каждой книги и определить среднюю оценку.
# А в основном запросе нужно сгруппировать по автору и подсчитать средние занчения количества оценок книг и средних оценок.
query='''
    SELECT
        authors.author AS author,
        ROUND(AVG(Sub.average_rating),1) AS average_rating_books,
        ROUND(AVG(Sub.cnt_rating),1) AS average_cnt_rating_books
    FROM
       books
    LEFT JOIN  (SELECT
                    ratings.book_id,
                    ROUND(AVG(ratings.rating),1) AS average_rating,
                    COUNT(ratings.rating_id) AS cnt_rating
                FROM
                    ratings
                GROUP BY
                    ratings.book_id) AS Sub ON Sub.book_id = books.book_id
    LEFT JOIN authors ON authors.author_id = books.author_id
    WHERE
        Sub.cnt_rating >= 50
    GROUP BY
        authors.author
    ORDER BY
        average_rating_books DESC;
    '''
# Вывод результата
display(pd.io.sql.read_sql(query, con = engine).head(5))
```

	author	average_rating_books	average_cnt_rating_books
0	J.K. Rowling/Mary GrandPré	4.3	77.5
1	Markus Zusak/Cao Xuân Việt Khương	4.3	53.0
2	J.R.R. Tolkien	4.3	81.0
3	Louisa May Alcott	4.2	52.0
4	Rick Riordan	4.1	62.0

ВЫВОД:

В число авторов с самой высокой средней оценкой книг, попало трое:

- J.K. Rowling/Mary GrandPré.
- Markus Zusak/Cao Xuân Việt Khương
- J.R.R. Tolkien

3.5. Среднее количество обзоров от пользователей, которые поставили больше 50 оценок.

In [11]:
```
# Нужно объеденить два подзапроса, сгруппированных по "username", один с подсчетом количества оценок,
# сделав срез по количеству оценок в соответствии с заданием, другой с подсчетом количества озывов.
query='''
    SELECT
        ROUND(AVG(Sub1.cnt_reviews),1) AS avg_cnt_reviews
    FROM
        (SELECT
            ratings.username,
            COUNT(ratings.rating) AS cnt_ratings
        FROM
            ratings
        GROUP BY
            ratings.username
        HAVING
            COUNT(ratings.rating) > 50) AS Sub
    LEFT JOIN (SELECT
                    reviews.username,
                    COUNT(reviews.review_id) AS cnt_reviews
                FROM
                    reviews
                GROUP BY
                    reviews.username) AS Sub1 ON Sub1.username = Sub.username;

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

Out[11]:

	avg_cnt_reviews

	avg_cnt_reviews
0	24.3

ВЫВОД:

Пользователи, оставляющие более 50 оценок, пишут в среднем 24,3 отзыва.

4. Выводы.

Наверх

- Для анализа были загружены 5 таблиц из базы данных.
- Выявлено следующее:
  - База содержит данные о 819 книгах, выпущенных после первого января 2000 года.
  - Для книг подобраны отзывы и оценки пользователей.
  - Самое широко представленное в базе издательство Penguin Books с 42 книгами (учитывались только книги толще 50 страниц).
  - ТОП-3 авторов с самой высокой оценкой книг:
    - J.K. Rowling/Mary GrandPré.
    - Markus Zusak/Cao Xuân Việt Khương.
    - J.R.R. Tolkien.
  - Активные пользователи сервиса, поставив оценку книге, примерно в 50% таких случаев пишут также отзыв о ней.

In [ ]: