SQL - проект

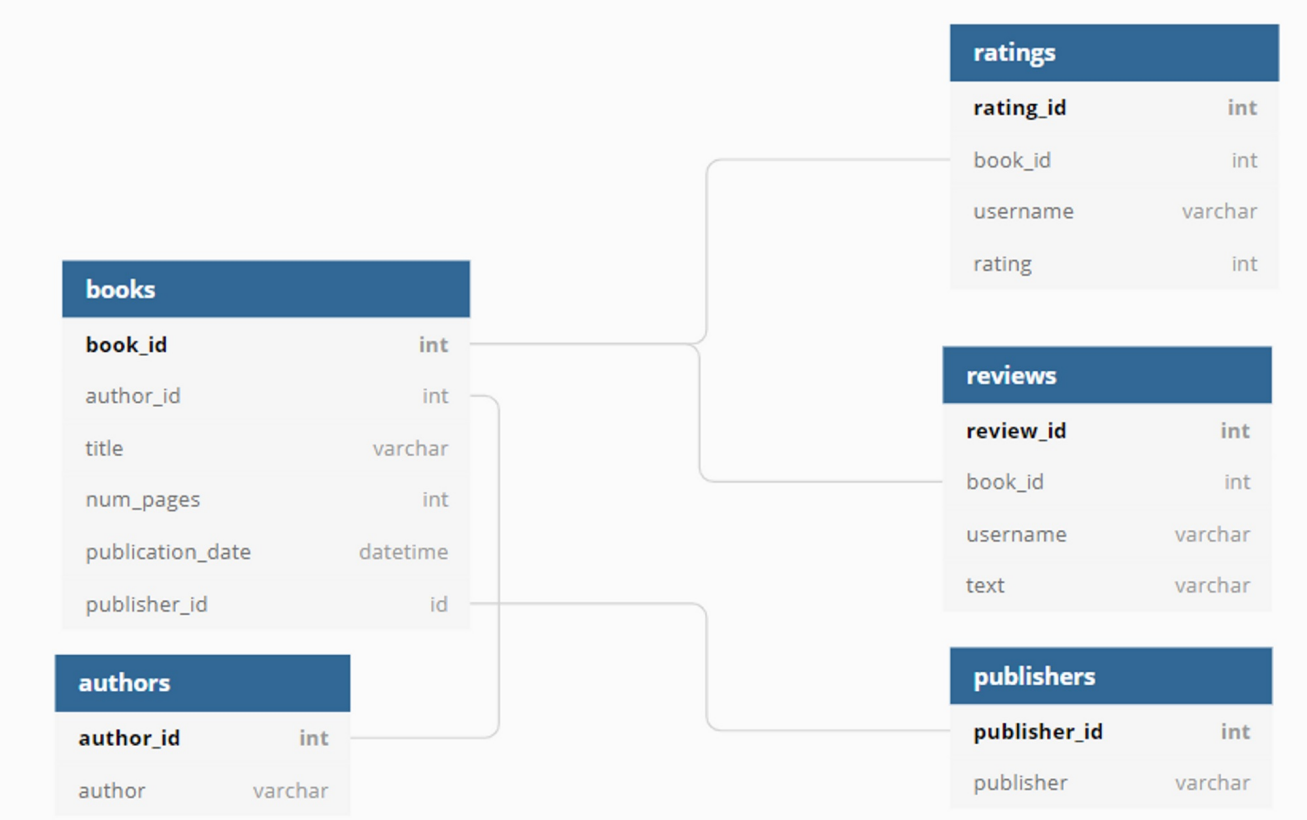
Постановка задачи

Цель проекта: проанализировать базу данных.

В ней — информация о книгах, издательствах, авторах, а также пользовательские обзоры книг. Эти данные помогут сформулировать ценностное предложение для нового продукта.

Задания проекта

- 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.



Подготовка к анализу

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

In [2]:
```
# устанавливаем параметры
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'])
```

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

Исследование данных

In [3]:
```
#запрос таблицы
query = '''
SELECT *
FROM books
'''

books = pd.io.sql.read_sql(query, con = engine)
display(books.duplicated().sum())
display(books.isna().sum())
books
```

```
0
book_id             0
author_id           0
title               0
num_pages           0
publication_date    0
publisher_id        0
dtype: int64
```

Out[3]:

	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
...	...	...	...	...	...	...
995	996	571	Wyrd Sisters (Discworld #6; Witches #2)	265	2001-02-06	147
996	997	454	Xenocide (Ender's Saga #3)	592	1996-07-15	297
997	998	201	Year of Wonders	358	2002-04-30	212
998	999	94	You Suck (A Love Story #2)	328	2007-01-16	331
999	1000	509	Zen and the Art of Motorcycle Maintenance: An ...	540	2006-04-25	143

1000 rows × 6 columns

Как мы видим в таблице books 1000 записей о книгах. Пропуски и дубликаты отсутствуют.

In [4]:
```
#запрос таблицы
query = '''
SELECT *
FROM authors
'''

authors = pd.io.sql.read_sql(query, con = engine)
display(authors.duplicated().sum())
display(authors.isna().sum())
authors
```

```
0
author_id    0
author       0
dtype: int64
```

Out[4]:

	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
...	...	...
631	632	William Strunk Jr./E.B. White
632	633	Zadie Smith
633	634	Zilpha Keatley Snyder
634	635	Zora Neale Hurston
635	636	Åsne Seierstad/Ingrid Christopherson

636 rows × 2 columns

В таблице authors 636 записей об авторах. Пропусков и дубликатов не обнаружено.

In [5]:
```
#запрос таблицы
query = '''
SELECT *
FROM ratings
'''

ratings = pd.io.sql.read_sql(query, con = engine)
display(ratings.duplicated().sum())
display(ratings.isna().sum())
ratings
```

```
0
rating_id    0
book_id      0
username     0
rating       0
dtype: int64
```

Out[5]:

	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
...	...	...	...	...
6451	6452	1000	carolrodriguez	4
6452	6453	1000	wendy18	4
6453	6454	1000	jarvispaul	5
6454	6455	1000	zross	2
6455	6456	1000	fharris	5

6456 rows × 4 columns

В таблице ratings 6456 записей с рейтингами пользователей. Пропуски и дубликаты отсутствуют.

In [6]:
```
#запрос таблицы
query = '''
SELECT *
FROM reviews
'''

reviews = pd.io.sql.read_sql(query, con = engine)
display(reviews.duplicated().sum())
display(reviews.isna().sum())
reviews
```

```
0
review_id    0
book_id      0
username     0
text         0
dtype: int64
```

Out[6]:

	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...
...	...	...	...	...
2788	2789	999	martinadam	Later hospital turn easy community. Fact same ...
2789	2790	1000	wknight	Change lose answer close pressure. Spend so now.
2790	2791	1000	carolrodriguez	Authority go who television entire hair guy po...
2791	2792	1000	wendy18	Or western offer wonder ask. More hear phone f...
2792	2793	1000	jarvispaul	Republican staff bit eat material measure plan...

2793 rows × 4 columns

В таблице reviews 2793 записей с отзывами. Пропусков и дубликатов нет.

In [7]:
```
#запрос таблицы
query = '''
SELECT *
FROM publishers
'''

publishers = pd.io.sql.read_sql(query, con = engine)
display(publishers.duplicated().sum())
display(publishers.isna().sum())
publishers
```

```
0
publisher_id    0
publisher       0
dtype: int64
```

Out[7]:

	publisher_id	publisher
0	1	Ace
1	2	Ace Book
2	3	Ace Books
3	4	Ace Hardcover
4	5	Addison Wesley Publishing Company
...	...	...
335	336	Workman Publishing Company
336	337	Wyatt Book
337	338	Yale University Press
338	339	Yearling
339	340	Yearling Books

340 rows × 2 columns

В таблице publishers 340 записей об издателях. Пропусков и дубликатов не обнаружено.

Данные чисты и готовы к анализу.

Задание №1

Посчитайте, сколько книг вышло после 1 января 2000 года;

In [8]:
```
#запрос таблицы
query = '''
SELECT COUNT(book_id)
FROM books
WHERE DATE_TRUNC('day',publication_date::date) > '2000-01-01'
'''

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

Out[8]:

	count
0	819

После 1 января 2000 года вышло 819 книг.

Задание №2

Для каждой книги посчитайте количество обзоров и среднюю оценку;

In [9]:
```
#запрос таблицы
query = '''
SELECT b.book_id,
b.title,
COUNT(DISTINCT(r.review_id)),
AVG(ra.rating)
FROM books AS b
FULL JOIN reviews AS r ON b.book_id = r.book_id
FULL JOIN ratings AS ra ON ra.book_id = b.book_id
GROUP BY b.book_id
'''

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

Out[9]:

	book_id	title	count	avg
0	1	'Salem's Lot	2	3.666667
1	2	1 000 Places to See Before You Die	1	2.500000
2	3	13 Little Blue Envelopes (Little Blue Envelope...	3	4.666667
3	4	1491: New Revelations of the Americas Before C...	2	4.500000
4	5	1776	4	4.000000
...	...	...	...	...
995	996	Wyrd Sisters (Discworld #6; Witches #2)	3	3.666667
996	997	Xenocide (Ender's Saga #3)	3	3.400000
997	998	Year of Wonders	4	3.200000
998	999	You Suck (A Love Story #2)	2	4.500000
999	1000	Zen and the Art of Motorcycle Maintenance: An ...	4	3.833333

1000 rows × 4 columns

Получаем таблицу с количеством обзоров и средней оценкой каждой книги.

Задание №3

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

In [10]:
```
#запрос таблицы
query = '''
SELECT p.publisher_id,
p.publisher,
COUNT(b.book_id) AS count_of_books
FROM books AS b
LEFT JOIN publishers AS p ON b.publisher_id = p.publisher_id
WHERE b.num_pages > 50
GROUP BY p.publisher_id
ORDER BY count_of_books DESC
LIMIT 1
'''

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

Out[10]:

	publisher_id	publisher	count_of_books
0	212	Penguin Books	42

Издатель Penguin Books выпустил наибольшее количество книг - 42 издания, которые толще 50 страниц.

Задание №4

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

In [11]:
```
#запрос таблицы
query = '''
SELECT t.author,
AVG(avg)
FROM
(SELECT b.book_id,
a.author_id,
a.author,
COUNT(ra.rating),
AVG(ra.rating)
FROM books AS b
LEFT JOIN authors AS a ON b.author_id = a.author_id
LEFT JOIN ratings AS ra ON b.book_id = ra.book_id
GROUP BY b.book_id,a.author_id
HAVING COUNT(ra.rating) >= 50) AS t
GROUP BY t.author
ORDER BY avg DESC
'''
output = pd.io.sql.read_sql(query, con = engine)
output
```

Out[11]:

	author	avg
0	J.K. Rowling/Mary GrandPré	4.283844
1	Markus Zusak/Cao Xuân Việt Khương	4.264151
2	J.R.R. Tolkien	4.258446
3	Louisa May Alcott	4.192308
4	Rick Riordan	4.080645
5	William Golding	3.901408
6	J.D. Salinger	3.825581
7	Paulo Coelho/Alan R. Clarke/Özdemir İnce	3.789474
8	William Shakespeare/Paul Werstine/Barbara A. M...	3.787879
9	Dan Brown	3.754540
10	Lois Lowry	3.750000
11	George Orwell/Boris Grabnar/Peter Škerl	3.729730
12	Stephenie Meyer	3.662500
13	John Steinbeck	3.622951

Как мы видим автором с самой высокой средней оценкой является Джоан Роулинг с оценкой в 4.3 балла.

Задание №5

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

In [12]:
```
#запрос таблицы
query = '''
SELECT AVG(count)
FROM(SELECT COUNT(review_id)
FROM reviews
WHERE username in (SELECT username
FROM ratings
GROUP BY username
HAVING COUNT(rating) > 50)
GROUP BY username) AS t
'''
output = pd.io.sql.read_sql(query, con = engine)
output
```

Out[12]:

	avg
0	24.333333

Как мы видим пользователи, поставившие больше 50 оценок в среднем пишут 24 обзора.

Выводы

Был проведен анализ, для каждой из задач написан запрос. Подытожим результаты: 1) После 1 января 2000 года вышло 819 книг.

2) Построена таблица с количеством обзоров и средней оценкой каждой книги.

3) Издатель Penguin Books выпустил наибольшее количество книг - 42 издания, которые толще 50 страниц.

4) Джоан Роулинг имеет наибольший средний рейтинг среди авторов, ее оценка 4.3 балла.

5) Пользователи, которые поставили больше 50 оценок в среднем пишут 24 обзора.