Иследование данных сервиса для чтения книг.

Нужна проанализировать базу данных крупный сервис для чтения книг по подписке что бы сформулировать ценностное предложение для нового продукта.

- Первый этап - изучение общий информации.
- Второй этап - исследовательский анализ.
- Третий этап - выводы.

Среди данных у нас есть информация о книгах, издательствах, авторах, а также пользовательские обзоры книг.

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.

Изучение общий информации.

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'})
```

In [4]:
```
query = '''
            SELECT * FROM books
        '''
books = pd.io.sql.read_sql(query, con = engine)
books.head()
```

Out[4]:

	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 [5]:
```
query = '''
            SELECT * FROM authors
        '''
authors = pd.io.sql.read_sql(query, con = engine)
authors.head()
```

Out[5]:

	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 [6]:
```
query = '''
            SELECT * FROM publishers
        '''
publishers = pd.io.sql.read_sql(query, con = engine)
publishers.head()
```

Out[6]:

	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 [7]:
```
query = '''
            SELECT * FROM ratings
        '''
ratings = pd.io.sql.read_sql(query, con = engine)
ratings.head()
```

Out[7]:

	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 [8]:
```
query = '''
            SELECT * FROM reviews
        '''
reviews = pd.io.sql.read_sql(query, con = engine)
reviews.head()
```

Out[8]:

	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...

Исследовательский анализ

- Посчитаем, сколько книг вышло после 1 января 2000 года.

In [9]:
```
query = '''
            SELECT COUNT(book_id)
            FROM books
            WHERE EXTRACT(YEAR FROM CAST(publication_date AS date)) >= 2000

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

Out[9]:

	count
0	821

- Для каждой книги посчитаем количество обзоров и среднюю оценку.

In [10]:
```
query = '''
            SELECT r.book_id,
```

```
                    title,
                    COUNT(DISTINCT review_id) AS review_count,
                    AVG(rating) AS rating_avg
            FROM ratings AS r
            LEFT JOIN reviews AS o ON o.book_id=r.book_id
            LEFT JOIN books AS b ON r.book_id=b.book_id
            GROUP BY r.book_id, title
            ORDER BY COUNT(review_id) DESC
        '''
data_1 = pd.io.sql.read_sql(query, con = engine)
data_1.head(10)
```

Out[10]:

	book_id	title	review_count	rating_avg
0	948	Twilight (Twilight #1)	7	3.662500
1	750	The Hobbit or There and Back Again	6	4.125000
2	673	The Catcher in the Rye	6	3.825581
3	302	Harry Potter and the Prisoner of Azkaban (Harr...	6	4.414634
4	299	Harry Potter and the Chamber of Secrets (Harry...	6	4.287500
5	75	Angels & Demons (Robert Langdon #1)	5	3.678571
6	301	Harry Potter and the Order of the Phoenix (Har...	5	4.186667
7	779	The Lightning Thief (Percy Jackson and the Oly...	6	4.080645
8	722	The Fellowship of the Ring (The Lord of the Ri...	5	4.391892
9	79	Animal Farm	5	3.729730

- Определим издательство, которое выпустило наибольшее число книг толще 50 страниц.

In [11]:
```
query = '''
            SELECT p.publisher,
                    COUNT(DISTINCT b.book_id)
            FROM books AS b
            LEFT JOIN publishers AS p ON b.publisher_id=p.publisher_id
            LEFT JOIN ratings AS r ON b.book_id=r.book_id
            WHERE num_pages > 50
            GROUP BY p.publisher
            ORDER BY COUNT(b.book_id) DESC
        '''
data_2 = pd.io.sql.read_sql(query, con = engine)
data_2.head(1)
```

Out[11]:

	publisher	count
0	Penguin Books	42

- Определим автора с самой высокой средней оценкой книг.

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

In [12]:
```
query = '''
            WITH
            avg_r AS
            (SELECT a.author_id,
```

```
                b.book_id,
                COUNT(DISTINCT rating_id)
            FROM ratings AS r
            LEFT JOIN books AS b ON r.book_id=b.book_id
            LEFT JOIN authors AS a ON b.author_id=a.author_id
            GROUP BY a.author_id, b.book_id
            HAVING COUNT(rating_id) >= 50
            ORDER BY COUNT(rating_id) DESC)

            SELECT a.author,
                   AVG(rating)
            FROM avg_r AS r
            LEFT JOIN authors AS a ON r.author_id=a.author_id
            LEFT JOIN books AS b ON r.book_id=b.book_id
            LEFT JOIN ratings AS ra ON b.book_id=ra.book_id
            GROUP BY a.author
            ORDER BY AVG(rating) DESC

        '''
data_3 = pd.io.sql.read_sql(query, con = engine)
data_3.head(1)
```

Out[12]:

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

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

In [13]:
```
query = '''
            WITH
            avg_r AS
            (SELECT r.username,
                   COUNT(rating)
            FROM ratings AS r
            GROUP BY r.username
            HAVING COUNT(rating) > 50),

            avg_c AS
            (SELECT r.username,
                   COUNT(review_id)
            FROM avg_r AS r
            LEFT JOIN reviews AS o ON r.username=o.username
            GROUP BY r.username)

            SELECT ROUND(AVG(count))
            FROM avg_c AS c
        '''
data_4 = pd.io.sql.read_sql(query, con = engine)
data_4
```

Out[13]:

	round
0	24.0

Выводы.

Итоги иследования:

- 821 книга вышло после 1 января 2000 года;

- В топ три по количеству обзоров вошли Twilight, The Hobbit or There and Back Again и The Catcher in the Rye, их средняя оценка не привышает четырех;
- Penguin Books издательство, которое выпустило наибольшее число книг;
- J.K. Rowling/Mary GrandPré автор с самой высокой средней оценкой книг среди 50 и более оценками;
- В среднем количество 24 обзора от пользователей, которые поставили больше 50 оценок.