Проект SQL

Описание проекта

Коронавирус застал мир врасплох, изменив привычный порядок вещей. В свободное время жители городов больше не выходят на улицу, не посещают кафе и торговые центры. Зато стало больше времени для книг. Это заметили стартаперы — и бросились создавать приложения для тех, кто любит читать.

Ваша компания решила быть на волне и купила крупный сервис для чтения книг по подписке. Ваша первая задача как аналитика — проанализировать базу данных. В ней — информация о книгах, издательствах, авторах, а также пользовательские обзоры книг. Эти данные помогут сформулировать ценностное предложение для нового продукта.

Описание данных

Структура таблицы **books**:

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

Структура таблицы **authors**:

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

Структура таблицы **publishers**:

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

Структура таблицы **ratings**:

- `rating_id` — идентификатор оценки;
- `book_id` — идентификатор книги;
- `username` — имя пользователя, оставившего оценку;

- `rating` — оценка книги.

Структура таблицы **reviews**:

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

Цели исследования

- проанализировать базу данных сервиса для чтения книг по подписке
- подготовить выводы по анализу

Задачи:

1. Посчитать, сколько книг вышло после 1 января 2000 года;
2. Для каждой книги посчитать количество обзоров и среднюю оценку;
3. Определить издательство, которое выпустило наибольшее число книг толще 50 страниц, исключив тем самым из анализа брошюры;
4. Определить автора с самой высокой средней оценкой книг — учитывать только книги с 50 и более оценками;
5. Посчитать среднее количество обзоров от пользователей, которые поставили больше 48 оценок.

Получим доступ к базе данных:

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

# 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://{user}:{pwd}@{host}:{port}/{db}'.format(**db_config)

# сохраняем коннектор
engine = sa.create_engine(connection_string, connect_args={'sslmode':'require'})
# чтобы выполнить SQL-запрос, пишем функцию с использованием Pandas

def get_sql_data(query:str, engine:sa.engine.base.Engine=engine) -> pd.DataFrame:
    '''Открываем соединение, получаем данные из sql, закрываем соединение'''
    with engine.connect() as con:
        return pd.read_sql(sql=sa.text(query), con = con)
```

Исследуем первые строки таблиц в базе данных:

In [2]:
```
# запишем функцию для вывода первых 5 строк датафрейма
def data(data):
    print('Первые пять строк датафрейма', data)
    query = '''SELECT * FROM ''' + data + ''' LIMIT 5'''
    display(get_sql_data(query))
    query = '''SELECT COUNT(*) FROM ''' + data + ''' LIMIT 5'''
    print('Всего строк в датафрейме:', int(get_sql_data(query).iloc[0]))
    print()
    print()
```

In [3]:
```
for i in ['books', 'authors', 'publishers', 'ratings', 'reviews']:
    data(i)
```

Первые пять строк датафрейма books

	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

Всего строк в датафрейме: 1000

Первые пять строк датафрейма authors

	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

Всего строк в датафрейме: 636

Первые пять строк датафрейма publishers

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

```
Всего строк в датафрейме: 340
```

```
Первые пять строк датафрейма ratings
```

	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

```
Всего строк в датафрейме: 6456
```

```
Первые пять строк датафрейма reviews
```

	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...

```
Всего строк в датафрейме: 2793
```

Выведены первые 5 строк каждого датафрейма. Проверим типы столбцов и наличие пропусков в датафреймах:

In [4]:
```
# запишем функцию для вывода основных характеристик датафрейма
def data_char(data):
    print('Характеристики датафрейма', data)
    print()
    query = '''SELECT * FROM ''' + data
    display(get_sql_data(query).info())
    print()
```

In [5]:
```
for i in ['books', 'authors', 'publishers', 'ratings', 'reviews']:
    data_char(i)
```

```
Характеристики датафрейма 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


Характеристики датафрейма 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
```

```
Характеристики датафрейма 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


Характеристики датафрейма 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


Характеристики датафрейма 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
```

Пропуски не обнаружены, типы столбцов соответствуют значениям, которые в них хранятся.

- **Задание 1**. Посчитать, сколько книг вышло после 1 января 2000 года;

In [6]:
```
query = '''SELECT COUNT(book_id) FROM books
           WHERE publication_date > '2000-01-01'
         '''
print('Количество книг, опубликованное после 1 января 2000 года -', int(get_sql_data(query).iloc[0]))
```

```
Количество книг, опубликованное после 1 января 2000 года - 819
```

- **Задание 2**. Для каждой книги посчитать количество обзоров и среднюю оценку;

In [7]:
```
query = '''SELECT books.book_id, COUNT(DISTINCT reviews.review_id), AVG(ratings.rating) FROM books
           LEFT JOIN reviews ON reviews.book_id = books.book_id
           LEFT JOIN ratings ON ratings.book_id = books.book_id
           GROUP BY books.book_id
           ORDER BY AVG(ratings.rating) DESC,
                    COUNT(DISTINCT reviews.review_id) DESC
        '''
print('Количество обзоров и средняя оценка для каждой книги')
display(get_sql_data(query))
print('Всего книг с обзорами и оценками:',
      len(get_sql_data(query)))
#get_sql_data(query)
```

```
Количество обзоров и средняя оценка для каждой книги
```

	book_id	count	avg
0	17	4	5.00
1	553	3	5.00
2	444	3	5.00
3	86	2	5.00
4	972	2	5.00
...	...	...	...
995	915	3	2.25
996	202	3	2.00
997	316	2	2.00
998	371	2	2.00
999	303	2	1.50

1000 rows × 3 columns

```
Всего книг с обзорами и оценками: 1000
```

- **Задание 3**. Определить издательство, которое выпустило наибольшее число книг толще 50 страниц, исключив тем самым из анализа брошюры.

In [8]:
```
query = '''SELECT publishers.publisher, COUNT(DISTINCT books.book_id) FROM books
           LEFT JOIN publishers ON publishers.publisher_id = books.publisher_id
           WHERE books.num_pages > 50
           GROUP BY publishers.publisher
           ORDER BY COUNT(DISTINCT books.book_id) DESC
           LIMIT 1
        '''
display(get_sql_data(query))
print('Издательство, которое выпустило наибольшее число книг толще 50 страниц:',
     get_sql_data(query)['publisher'].iloc[0], '.',
     'Оно выпустило', get_sql_data(query)['count'].iloc[0], 'таких книг.')
```

	publisher	count
0	Penguin Books	42

```
Издательство, которое выпустило наибольшее число книг толще 50 страниц: Penguin Books . Оно выпустило 42 таких книг.
```

- **Задание 4**. Определить автора с самой высокой средней оценкой книг — учитывать только книги с 50 и более оценками

In [9]:
```
query = '''
          SELECT authors.author, one.author_id, one.avg
          FROM (SELECT books.author_id, AVG(rating) FROM books
                LEFT JOIN ratings ON ratings.book_id = books.book_id
                WHERE books.book_id IN (
                                        SELECT ratings.book_id FROM ratings
                                        GROUP BY ratings.book_id
                                        HAVING COUNT(ratings.rating) > 50
                                        )
               GROUP BY books.author_id
               ORDER BY AVG(rating) DESC) AS one
          JOIN authors ON authors.author_id = one.author_id
          ORDER BY avg DESC
        '''
display(get_sql_data(query))
```

	author	author_id	avg
0	J.K. Rowling/Mary GrandPré	236	4.287097
1	Markus Zusak/Cao Xuân Việt Khương	402	4.264151
2	J.R.R. Tolkien	240	4.246914
3	Louisa May Alcott	376	4.192308
4	Rick Riordan	498	4.080645
5	William Golding	621	3.901408
6	J.D. Salinger	235	3.825581
7	Paulo Coelho/Alan R. Clarke/Özdemir İnce	469	3.789474
8	William Shakespeare/Paul Werstine/Barbara A. M...	630	3.787879
9	Lois Lowry	372	3.750000
10	Dan Brown	106	3.741259
11	George Orwell/Boris Grabnar/Peter Škerl	195	3.729730
12	Stephenie Meyer	554	3.662500
13	John Steinbeck	311	3.622951

Автор с самой высокой средней оценкой книг- `J.K. Rowling/Mary GrandPré` со средней оценкой 4.287.

- **Задание 5**. Посчитайте среднее количество обзоров от пользователей, которые поставили больше 48 оценок.

In [10]:
```
query = '''
          SELECT AVG(one.count) FROM (
                                     (SELECT COUNT(DISTINCT review_id) AS count FROM reviews
                                      WHERE username IN (
                                                         SELECT ratings.username FROM ratings
                                                         GROUP BY ratings.username
                                                         HAVING COUNT(DISTINCT ratings.rating_id) > 48
                                                         )
                                      GROUP BY username)) AS one
        '''
print('Среднее количество обзоров от пользователей, которые поставили больше 48 оценок:',
      float(get_sql_data(query).iloc[0]))
```

```
Среднее количество обзоров от пользователей, которые поставили больше 48 оценок: 24.0
```

Общий вывод:

- Пропуски не обнаружены, типы столбцов соответствуют значениям, которые в них хранятся.
- Количество книг, опубликованное после 1 января 2000 года - `819` .
- Всего в базе `1000` книг с определенным количеством обзоров и оценок.
- Издательство, которое выпустило наибольшее число книг толще 50 страниц - `Penguin Books` , оно выпустило 42 таких книг.
- Автор с самой высокой средней оценкой книг- `J.K. Rowling/Mary GrandPré` со средней оценкой `4.287` .
- Среднее количество обзоров от пользователей, которые поставили больше 48 оценок: `24.0`