Сервис чтения книг

1 Цель исследования

- проанализировать базу данных

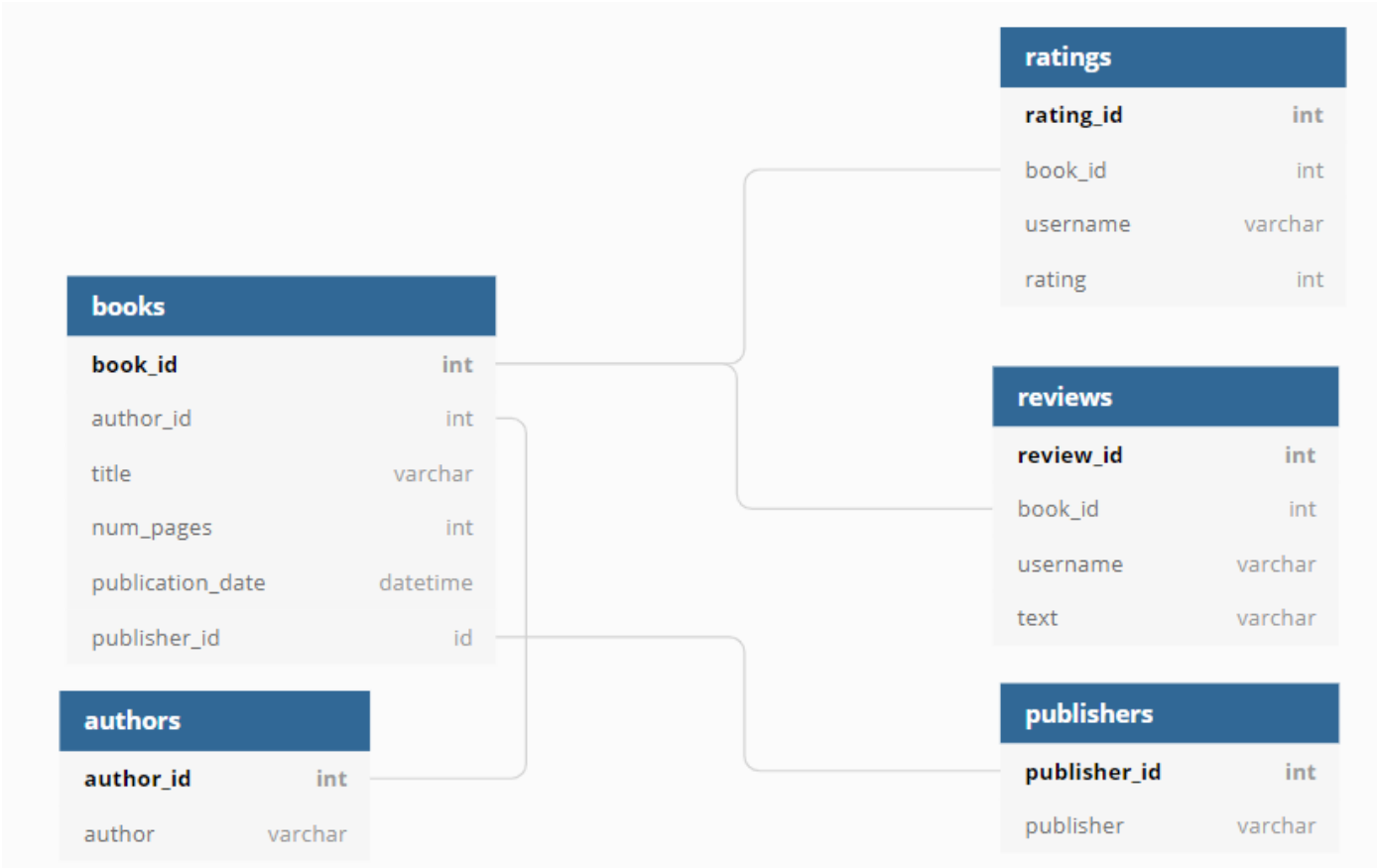
2 Задачи

- прочесть все таблички
- сколько книг вышло после 1 января 2000 года
- для каждой книги посчитать количество обзоров и среднюю оценку
- определить издательство, которое выпустило наибольшее число книг толще 50 страниц
- - для исключения из анализа брошюр
- определить автора с самой высокой средней оценкой книг — учитываем только книги с 50 и более оценками
- посчитать среднее количество обзоров от пользователей, которые поставили больше 50 оценок

In [10]:

```
# импорт библиотек
import pandas as pd
from sqlalchemy import create_engine
```

2.1 Схема данных



In [11]:

```
# подключаем базу данных
# устанавливаем параметры
...
```

2.2 прочитаем все таблички

2.2.1 books

In [12]:

```
query = '''
            SELECT *
            FROM books
        '''
books = pd.io.sql.read_sql(query, con = engine, index_col='book_id')

```

In [13]:

```
books.head(5)
```

Out[13]:

book_id	author_id	title	num_pages	publication_date	publisher_id
1	546	'Salem's Lot	594	2005-11-01	93
2	465	1 000 Places to See Before You Die	992	2003-05-22	336
3	407	13 Little Blue Envelopes (Little Blue Envelope...	322	2010-12-21	135
4	82	1491: New Revelations of the Americas Before C...	541	2006-10-10	309
5	125	1776	386	2006-07-04	268

2.2.2 authors

In [14]:

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

```

Out[14]:

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

2.2.3 publishers

In [15]:

```
query = '''
            SELECT *
            FROM publishers
        '''
publishers = pd.io.sql.read_sql(query, con = engine, index_col='publisher_id')
publishers.head(5)

```

Out[15]:

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

2.2.4 ratings

In [16]:

```
query = '''
            SELECT *
            FROM ratings
        '''
ratings = pd.io.sql.read_sql(query, con = engine, index_col='rating_id')
ratings.head(5)

```

Out[16]:

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

2.2.5 reviews

In [17]:

```
query = '''
            SELECT *
            FROM reviews
        '''
reviews = pd.io.sql.read_sql(query, con = engine, index_col='review_id')
reviews.head(5)

```

Out[17]:

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

2.3 сколько книг вышло после 1 января 2000 года

- табл books
- поле publication_date Условие publication_date > 1 января 2000 года
- publication_date в формат даты

In [18]:

```
query = '''
            SELECT

            COUNT(book_id)
            FROM books
            WHERE CAST(publication_date AS timestamp) > '2000-01-01'
        '''
books_after_01_01_2000 = pd.io.sql.read_sql(query, con = engine)
books_after_01_01_2000

```

Out[18]:

	count
0	819

2.4 для каждой книги посчитать количество обзоров и среднюю оценку

- количество обзоров в связке таблиц
- - books
- - reviews по полю book_id, подсчитываем количество строк
- оценка в связке таблиц
- - books
- - ratings по полю book_id, вычисляем среднее по полю rating

In [19]:

```
query = '''
            SELECT
            b.title,

            COUNT(rev.text) AS count_reviews,
            ROUND(AVG(rat.rating), 2)  AS avg_rating

            FROM books AS b
            LEFT JOIN ratings AS rat ON b.book_id = rat.book_id
            LEFT JOIN reviews AS rev ON b.book_id = rev.book_id

            GROUP BY b.book_id

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

```

Out[19]:

	title	count_reviews	avg_rating
0	The Body in the Library (Miss Marple #3)	4	4.50
1	Galápagos	4	4.50
2	A Tree Grows in Brooklyn	60	4.25
3	Undaunted Courage: The Pioneering First Missio...	4	4.00
4	The Prophet	28	4.29
...	...	...	...
995	Alice in Wonderland	52	4.23
996	A Woman of Substance (Emma Harte Saga #1)	4	5.00
997	Christine	21	3.43
998	The Magicians' Guild (Black Magician Trilogy #1)	4	3.50
999	The Plot Against America	4	3.00

1000 rows × 3 columns

2.5 определить издательство, которое выпустило наибольшее число книг толще 50 страниц

- издательство в табл publishers - publisher
- количество страниц в books - num_pages Соеденим publishers с books по полю publisher_id
- фильтр num_pages > 50
- МАКС для publisher по количеству книг

In [20]:

```
query = '''
            SELECT
            publisher,
            COUNT(book_id) AS count_book

            FROM publishers
            LEFT JOIN books ON publishers.publisher_id = books.publisher_id
            WHERE num_pages > 50
            GROUP BY publisher
            ORDER BY count_book DESC
            LIMIT 1


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

```

Out[20]:

	publisher	count_book
0	Penguin Books	42

2.6 определить автора с самой высокой средней оценкой книг — учитываем только книги с 50 и более оценками

- автор в authors в поле author - связь с books по author_id
- оценка в ratings в поле rating - связь с books по book_id

In [21]:

```
query = '''
            SELECT
            author,

            AVG(rating)

            FROM books
            LEFT JOIN authors ON books.author_id = authors.author_id
            LEFT JOIN ratings ON books.book_id = ratings.book_id

            GROUP BY author
            HAVING COUNT(rating) >= 50
            ORDER BY avg DESC
            LIMIT 1


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

```

Out[21]:

	author	avg
0	Diana Gabaldon	4.3

2.7 посчитать среднее количество обзоров от пользователей, которые поставили больше 50 оценок

- обзоры в reviews
- оценки в ratings Таблицы свяжем по поле username.
- при связывании таблиц обозначим username каждой таблицы через синонимы

In [22]:

```
query = '''
            SELECT
             AVG(users_list.number_of_reviews) as averadge_number_of_reviews
             FROM
                 (SELECT
                     username,
                     COUNT(review_id) AS number_of_reviews
                 FROM
                     reviews
                 WHERE
                     username IN (
                             SELECT
                             username
                             FROM
                             ratings
                         GROUP BY
                         username
                         HAVING
                         COUNT(rating_id) > 50
                         )
             GROUP BY
                 username
              ) AS users_list


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

```

Out[22]:

	averadge_number_of_reviews
0	24.333333

3 Итог

- сколько книг вышло после 1 января 2000 года
- - 819
- для каждой книги посчитать количество обзоров и среднюю оценку
- - books_reviews_ratings
- определить издательство, которое выпустило наибольшее число книг толще 50 страниц
- - Penguin Books
- определить автора с самой высокой средней оценкой книг — учитываем только книги с 50 и более оценками
- - Diana Gabaldon
- посчитать среднее количество обзоров от пользователей, которые поставили больше 50 оценок
- - 24.33