SQL

Оглавление :

Шаг 1. Загрузка данных и создание подключения

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

In [2]:
```
db_config = {'user': 'praktikum_student',
    'pwd': 'Sdf4$2;d-d30pp',
    'host': 'rc1b-wcoijxj3yxfsf3fs.mdb.yandexcloud.net',
    'port': 6432,
    'db': 'data-analyst-final-project-db'}
```

In [3]:
```
connection_string = 'postgresql://{}:{}@{}:{}/{}'.format(db_config['user'],
                                                         db_config['pwd'],
                                                         db_config['host'],
                                                         db_config['port'],
                                                         db_config['db'])
engine = create_engine(connection_string, connect_args={'sslmode':'require'})
```

In [4]:
```
query = ''' SELECT *
            FROM books
        '''
```

In [5]:
```
books = pd.io.sql.read_sql(query, con = engine)
books.head()
```

Out[5]:

	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 [6]:
```
query = ''' SELECT *
            FROM authors
        '''
```

In [7]:
```
authors = pd.io.sql.read_sql(query, con = engine)
authors.head()
```

Out[7]:

	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 [8]:
```
query = ''' SELECT *
            FROM publishers
        '''
```

In [9]:
```
publishers = pd.io.sql.read_sql(query, con = engine)
publishers.head()
```

Out[9]:

	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 [10]:
```
query = ''' SELECT *
            FROM ratings
        '''
```

In [11]:
```
ratings = pd.io.sql.read_sql(query, con = engine)
ratings.head()
```

Out[11]:

	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 [12]:
```
query = ''' SELECT *
            FROM reviews
        '''
```

In [13]:
```
reviews = pd.io.sql.read_sql(query, con = engine)
reviews.head()
```

Out[13]:

	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...

image

Шаг 2. Задачи

Посчитайте, сколько книг вышло после 1 января 2000 года

In [14]:
```
query = ''' SELECT COUNT(book_id)
            FROM books
            WHERE publication_date > '2000-01-01'
        '''
```

In [15]:
```
Books_after_2005 = pd.io.sql.read_sql(query, con = engine)
Books_after_2005
```

Out[15]:

	count
0	819

Для каждой книги посчитайте количество обзоров и среднюю оценку

In [16]:
```
query = ''' SELECT books.book_id, books.title, COUNT(DISTINCT reviews.review_id), AVG(ratings.rating)
            FROM books
            JOIN ratings ON books.book_id = ratings.book_id
            JOIN reviews ON books.book_id = reviews.book_id
            GROUP BY books.book_id

        '''
```

In [17]:
```
review_and_rating = pd.io.sql.read_sql(query, con = engine)
review_and_rating
```

Out[17]:

	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
...	...	...	...	...
989	996	Wyrd Sisters (Discworld #6; Witches #2)	3	3.666667
990	997	Xenocide (Ender's Saga #3)	3	3.400000
991	998	Year of Wonders	4	3.200000
992	999	You Suck (A Love Story #2)	2	4.500000
993	1000	Zen and the Art of Motorcycle Maintenance: An ...	4	3.833333

994 rows × 4 columns

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

In [18]:
```
query = ''' SELECT publisher
            FROM publishers
            WHERE publisher_id IN ( SELECT publisher_id
                                    FROM books
                                    WHERE num_pages > 50
                                    GROUP BY publisher_id
                                    ORDER BY COUNT(title) DESC
                                    LIMIT 1)
        '''
```

In [19]:
```
thick_book_publisher = pd.io.sql.read_sql(query, con = engine)
thick_book_publisher
```

Out[19]:

	publisher
0	Penguin Books

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

In [20]:
```
query = ''' SELECT DISTINCT author, AVG(rating)
            FROM books
            JOIN authors ON books.author_id = authors.author_id
            JOIN ratings ON books.book_id = ratings.book_id
            WHERE books.book_id IN (SELECT book_id
                                    FROM ratings
                                    GROUP BY book_id
                                    HAVING COUNT(rating) >= 50)
            GROUP BY author
            ORDER BY AVG(rating) DESC
        '''
```

In [21]:
```
highest_rated_author = pd.io.sql.read_sql(query, con = engine)
highest_rated_author
```

Out[21]:

	author	avg
0	J.K. Rowling/Mary GrandPré	4.287097
1	Markus Zusak/Cao Xuân Việt Khương	4.264151
2	J.R.R. Tolkien	4.246914
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	Lois Lowry	3.750000
10	Dan Brown	3.741259
11	George Orwell/Boris Grabnar/Peter Škerl	3.729730
12	Stephenie Meyer	3.662500
13	John Steinbeck	3.622951

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

In [22]:
```
query = ''' SELECT AVG(count)
            FROM (  SELECT COUNT(text)
                    FROM reviews
                    WHERE username IN ( SELECT username
                                        FROM ratings
                                        GROUP BY username
                                        HAVING COUNT(rating) > 50)
                    GROUP BY username) AS reviews
        '''
```

In [23]:
```
pd.io.sql.read_sql(query, con = engine)
```

Out[23]:

	avg
0	24.333333

In [ ]:
```

```