Анализ базы данных с помощью SQL запросов.

Задача: Проанализировать базу данных, сформулировать ценностное предложение для нового продукта.

Данные:

Информация о книгах, издательствах, авторах, ользовательские обзоры книг.

Импорт библиотек:

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

Ввод [2]:
```
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'])
```

Загрузка данных books:

Ввод [3]:
```
# пять строк таблицы books
query = '''
SELECT
*
FROM
books
LIMIT 5;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

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

Ввод [4]:
```
display(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

В таблице с названием книг находятся данные с идентификационным номером книги, идентификационным номером автора, датой выхода книги и идентификационным номером издательства, опубликовавшего книгу.

Загрузка данных authors:

Ввод [5]:
```
# пять строк таблицы authors
query1 = '''
SELECT
*
FROM
authors
LIMIT 5;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

authors = pd.io.sql.read_sql(query1, con=engine)
```

Ввод [6]:
```
display(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

В таблице с именем автора, содержится также его идентификационный номер.

Загрузка данных publishers:

Ввод [7]:
```
# пять строк таблицы authors
query2 = '''
SELECT
*
FROM
publishers
LIMIT 5;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

publishers = pd.io.sql.read_sql(query2, con=engine)
```

Ввод [8]:
```
display(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

В таблице с информацией об издательстве хранится название и идентификационный номер издательства.

Загрузка данных ratings:

Ввод [9]:
```
# пять строк таблицы authors
query3 = '''
SELECT
*
FROM
ratings
LIMIT 5;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

ratings = pd.io.sql.read_sql(query3, con=engine)
```

Ввод [10]:
```
display(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

В таблице с информацией о рейтингах книг хранится имя читателя, его оценка книги, идентификационный номер оценки, сама оценка и идентификационный номер книги, которая была оценена.

Загрузка данных reviews:

Ввод [11]:
```
# пять строк таблицы authors
query4 = '''
SELECT
*
FROM
reviews
--LIMIT 5;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

reviews = pd.io.sql.read_sql(query4, con=engine)
```

Ввод [12]:
```
display(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...
...	...	...	...	...
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 с информацией об отзывах читателей хранится имя читателя, его отзыв в текстовом формате, идентификационный номер отзыва и идентификационный номер книги, о которой был отзыв.

Количество книг, которое вышло после 1 января 2000 года:

Ввод [13]:
```
# количество книг с первого января включительно
query5 = '''
SELECT
COUNT(book_id)
FROM
books
WHERE
CAST(books.publication_date as date) >= '2000-01-01';
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

quest1 = pd.io.sql.read_sql(query5, con=engine)
```

Ввод [14]:
```
print(quest1)
```

```
   count
0    821
```

С 01 января 2000 года была опубликована 821 книга.

Количество обзоров и средняя оценка для каждой книги:

Ввод [15]:
```
# количество обзоров и средняя оценка
query5 = '''
SELECT
books.book_id as book_number,
books.title as name_book,
COUNT(DISTINCT reviews.text) as review,
AVG(ratings.rating) as averagerating
FROM
books
LEFT JOIN reviews ON reviews.book_id = books.book_id
LEFT JOIN ratings ON ratings.book_id = reviews.book_id
WHERE
CAST(books.publication_date as date) >= '2000-01-01'
GROUP BY
book_number,
name_book
ORDER BY
review DESC;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

quest2 = pd.io.sql.read_sql(query5, con=engine)
```

Ввод [16]:
```
display(quest2.head(10))
```

	book_number	name_book	review	averagerating
0	948	Twilight (Twilight #1)	7	3.662500
1	302	Harry Potter and the Prisoner of Azkaban (Harr...	6	4.414634
2	207	Eat Pray Love	6	3.395833
3	779	The Lightning Thief (Percy Jackson and the Oly...	6	4.080645
4	854	The Road	6	3.772727
5	963	Water for Elephants	6	3.977273
6	497	Outlander (Outlander #1)	6	4.125000
7	750	The Hobbit or There and Back Again	6	4.125000
8	656	The Book Thief	6	4.264151
9	696	The Da Vinci Code (Robert Langdon #2)	6	3.830508

Лидирующую позицию занимает "Сумерки" по количеству отзывов по среднему рейтингу книга - средняя 3,7 из 5-ти, у остальных произведеней количество отзывов в два раза меньше, количесвто отзывов свидетельствует о популярности книги. Также в 10-ке книг три книги о Гарри Поттере, как подтверждение несомненной популярности серии книг о нём с рейтингом выше 4-х.

Издательство, которое выпустило наибольшее число книг толще 50 страниц

Ввод [17]:
```
# количество книг по издательствам и средняя оценка
query6 = '''
SELECT
publishers.publisher as publisher,
COUNT(DISTINCT books.title) as cnt
FROM
books
LEFT JOIN publishers ON publishers.publisher_id = books.publisher_id

GROUP BY
publisher
ORDER BY
cnt DESC;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

quest3 = pd.io.sql.read_sql(query6, con=engine)
```

Ввод [18]:
```
display(quest3.head(10))
```

	publisher	cnt
0	Penguin Books	42
1	Vintage	31
2	Grand Central Publishing	25
3	Penguin Classics	24
4	Ballantine Books	19
5	Bantam	19
6	Berkley	17
7	St. Martin's Press	14
8	Berkley Books	14
9	William Morrow Paperbacks	13

Наибольшее количество книг выпущено Penguin Books, между ними и 10-м издательством трехразовое превосходство, судя по информации для анализа предоставлены данные британских издательств и британских предпочтений.

Автор с самой высокой средней оценкой книг с 50 и более оценками:

Ввод [19]:
```
# автор с самой высмокой оценкой книг с 50-тью и более
query7 = '''
SELECT
authors.author as author,
AVG(ratings.rating) as AverageRating
FROM
authors
RIGHT JOIN books ON books.author_id = authors.author_id
LEFT JOIN ratings ON ratings.book_id = books.book_id

GROUP BY
author

HAVING
COUNT(ratings.book_id) > 50

ORDER BY
AverageRating DESC;
'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

quest4 = pd.io.sql.read_sql(query7, con=engine)
```

Ввод [20]:
```
display(quest4)
```

	author	averagerating
0	J.K. Rowling/Mary GrandPré	4.288462
1	Agatha Christie	4.283019
2	Markus Zusak/Cao Xuân Việt Khương	4.264151
3	J.R.R. Tolkien	4.240964
4	Roald Dahl/Quentin Blake	4.209677
5	Louisa May Alcott	4.203704
6	Rick Riordan	4.130952
7	Arthur Golden	4.107143
8	Stephen King	4.009434
9	John Grisham	3.971429
10	William Golding	3.901408
11	Nicholas Sparks	3.882883
12	Jodi Picoult	3.881579
13	Sophie Kinsella	3.877193
14	James Patterson	3.859375
15	J.D. Salinger	3.846939
16	Paulo Coelho/Alan R. Clarke/Özdemir İnce	3.789474
17	William Shakespeare/Paul Werstine/Barbara A. M...	3.787879
18	Dan Brown	3.741259
19	Lois Lowry	3.738462
20	George Orwell/Boris Grabnar/Peter Škerl	3.729730
21	Stephenie Meyer	3.662500
22	John Steinbeck	3.643836

Наивысший рейтинг имеют кники Джоан Роулинг с иллюстрациями Мари Гранпре, действительно, все герои книг о Гарри Поттере представляются в образах, созданных Мари Гранпре. Второй автор с высоким рейтингом Агата Кристи, вероятно, она до сих пор популярна.

Среднее количество обзоров от пользователей, которые поставили больше 50 оценок.

Ввод [21]:
```
# автор с самой высокой оценкой книг с 50-тью и более
query8 = '''
SELECT
AVG(SUB1.reviews) AS averagereviews

FROM
(
SELECT
username,
COUNT(rating) AS ratings
FROM
ratings

GROUP BY
username

HAVING
COUNT(rating) > 50) AS SUB

JOIN
(
SELECT
username,
COUNT(text) AS reviews

FROM
reviews

GROUP BY
username
) AS SUB1 ON SUB.username = SUB1.username;

'''
engine = create_engine(connection_string,connect_args={'sslmode':'require'})

quest5 = pd.io.sql.read_sql(query8, con=engine)
```

Ввод [22]:
```
display(quest5)
```

	averagereviews
0	24.333333

Среднее количество отзывов на одного пользователя, который ставит более 50 оценок - 24,33!