Анализ базы данных книг для читательского сервиса по подписке

# Описание проекта.

Компания-заказчик решила быть на волне и купила крупный сервис для чтения книг по подписке. Наша первая задача — проанализировать базу данных. В ней — информация о книгах, издательствах, авторах, а также пользовательские обзоры книг. Эти данные могут помчь сформулировать ценностное предложение для нового продукта.

# Описание данных.

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.

Ввод [1]:

*# импортируем библиотеки*

**import** pandas **as** pd

**from** sqlalchemy **import** text, create\_engine

Ввод [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://{user}:{pwd}@{host}:{port}/{db}'.format(**\*\***db\_config)

*# сохраняем коннектор*

engine **=** create\_engine(connection\_string, connect\_args**=**{'sslmode':'require'})

## 1  Исследование таблиц.

Ввод [3]:

query **=** '''SELECT \* FROM books LIMIT 5'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[3]:

|  | **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 |

Ввод [4]:

query **=** '''SELECT \* FROM authors LIMIT 5'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[4]:

|  | **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 |

Ввод [5]:

query **=** '''SELECT \* FROM publishers LIMIT 5'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[5]:

|  | **publisher\_id** | **publisher** |
| --- | --- | --- |
| **0** | 1 | Ace |
| **1** | 2 | Ace Book |
| **2** | 3 | Ace Books |
| **3** | 4 | Ace Hardcover |
| **4** | 5 | Addison Wesley Publishing Company |

Ввод [6]:

query **=** '''SELECT \* FROM ratings LIMIT 5'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[6]:

|  | **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 |

Ввод [7]:

query **=** '''SELECT \* FROM reviews LIMIT 5'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[7]:

|  | **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... |

С данными из всех 5 таблиц ознакомились.

**SELECT** **\***, **COUNT**(**\***) OVER() **AS** table\_size

**FROM** books

**LIMIT** 5

## 2  Задания.

1. Посчитайте, сколько книг вышло после 1 января 2000 года.

Ввод [8]:

query **=** '''

​

SELECT COUNT(DISTINCT book\_id)

FROM books

WHERE publication\_date > '2000-01-01';

​

'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[8]:

|  | **count** |
| --- | --- |
| **0** | 819 |

После 1 января 2000 года вышло 819 книг.

1. Для каждой книги посчитайте количество обзоров и среднюю оценку.

Ввод [9]:

query **=** '''

​

SELECT b.title,

COUNT(DISTINCT rv.review\_id) AS cnt\_reviews,

ROUND(AVG(r.rating),1) AS avg\_rating

FROM books b

​

LEFT JOIN reviews rv

ON b.book\_id = rv.book\_id

​

LEFT JOIN ratings r

ON b.book\_id = r.book\_id

​

GROUP BY b.book\_id, b.title

ORDER BY cnt\_reviews DESC

​

'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[9]:

|  | **title** | **cnt\_reviews** | **avg\_rating** |
| --- | --- | --- | --- |
| **0** | Twilight (Twilight #1) | 7 | 3.7 |
| **1** | Water for Elephants | 6 | 4.0 |
| **2** | The Glass Castle | 6 | 4.2 |
| **3** | Harry Potter and the Prisoner of Azkaban (Harr... | 6 | 4.4 |
| **4** | The Curious Incident of the Dog in the Night-Time | 6 | 4.1 |
| **...** | ... | ... | ... |
| **995** | Anne Rice's The Vampire Lestat: A Graphic Novel | 0 | 3.7 |
| **996** | The Natural Way to Draw | 0 | 3.0 |
| **997** | The Cat in the Hat and Other Dr. Seuss Favorites | 0 | 5.0 |
| **998** | Essential Tales and Poems | 0 | 4.0 |
| **999** | Disney's Beauty and the Beast (A Little Golden... | 0 | 4.0 |

1000 rows × 3 columns

Всего в сервисе 1000 книг со средними оценками от 3.6 до 4, максимальное каличество обзоров - 7. Высокий рейтинг есть как у книг с обзорами, так и без - можно предположить, что эти параметры никак не влияют друг на друга.

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

Ввод [10]:

query **=** '''

​

SELECT p.publisher\_id, p.publisher, COUNT(b.book\_id) AS book\_count

FROM publishers p

LEFT JOIN books b ON p.publisher\_id=b.publisher\_id

WHERE b.num\_pages > 50

GROUP BY p.publisher\_id

ORDER BY book\_count DESC

LIMIT 1

​

'''

con**=**engine.connect()

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[10]:

|  | **publisher\_id** | **publisher** | **book\_count** |
| --- | --- | --- | --- |
| **0** | 212 | Penguin Books | 42 |

Рекордсменом по выпуску книг на более чем 50 станиц становится издательство Penguin Books!

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

Ввод [11]:

query **=** '''

WITH q AS (

SELECT b.book\_id, b.author\_id, COUNT(rating)

FROM authors a JOIN books b ON a.author\_id = b.author\_id JOIN ratings r ON b.book\_id = r.book\_id

GROUP BY b.book\_id

HAVING COUNT(rating) >= 50)

​

SELECT a.author\_id, a.author, AVG(r.rating)

FROM authors a JOIN q ON a.author\_id = q.author\_id JOIN ratings r ON q.book\_id = r.book\_id

GROUP BY a.author\_id

ORDER BY avg DESC

LIMIT 1

​

'''

​

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[11]:

|  | **author\_id** | **author** | **avg** |
| --- | --- | --- | --- |
| **0** | 236 | J.K. Rowling/Mary GrandPré | 4.287097 |

Ввод [12]:

*# ячейка ревьюера:*

query **=** '''

WITH q AS (

SELECT b.book\_id, b.author\_id, AVG(rating)

FROM authors a

JOIN books b ON a.author\_id = b.author\_id

JOIN ratings r ON b.book\_id = r.book\_id

GROUP BY b.author\_id, b.book\_id

HAVING COUNT(rating) >= 50)

​

SELECT a.author, AVG(q.avg)

FROM authors a

JOIN q ON a.author\_id = q.author\_id

GROUP BY q.author\_id, a.author

ORDER BY avg DESC

LIMIT 1

​

'''

​

pd.io.sql.read\_sql(sql**=**text(query), con **=** con)

Out[12]:

|  | **author** | **avg** |
| --- | --- | --- |
| **0** | J.K. Rowling/Mary GrandPré | 4.283844 |

Автор с самой высокой средней оценкой книг - это Джоан Роулинг, чьи романы прочитали тысячи взрослых и детей.

1. Посчитайте среднее количество обзоров от пользователей, которые поставили больше 48 оценок.

Ввод [13]:

query **=** '''

​

SELECT COUNT(text)/ COUNT(DISTINCT username) AS avg\_rv\_count

FROM reviews

WHERE username IN (

SELECT username

FROM ratings

GROUP BY username

HAVING COUNT(rating\_id) > 48

)

​

'''

​

pd.io.sql.read\_sql(query, con **=** engine)

Out[13]:

|  | **avg\_rv\_count** |
| --- | --- |
| **0** | 24 |

Среднее количество обзоров от пользователей, которые поставили больше 48 оценок составляет 24.

## 3  Вывод.

В ходе анализа данных для сервиса чтения книг по подписке нам удалось получить к ним доступ и ознакомиться, было получено 5 таблиц: books, authors, publishers, ratings и reviews с информацией о книгах/авторах/издательствах/рейтингах и отзывах. В ходе сотавления 5 sql-запросов было выявлено:

* после 1 января 2000 года вышло 819 книг.
* всего в сервисе 1000 книг со средними оценками от 3.6 до 4, максимальное каличество обзоров - 7, высокий рейтинг есть как у книг с обзорами, так и без - можно предположить, что эти параметры никак не влияют друг на друга;
* рекордсменом по выпуску книг на более чем 50 станиц становится издательство Penguin Books;
* автор с самой высокой средней оценкой книг - это Джоан Роулинг, чьи романы прочитали тысячи взрослых и детей;
* и среднее количество обзоров от пользователей, которые поставили больше 48 оценок составляет 24.