

**Final Project**

**GRAVITY BOOKS**

**Course:** Applied Data Structure & DB

CIS-525-19000

**Submitted By,**

Risha Alam

ID: 2327100

**Submitted to,**

Dr. Elizabeth Henley

Associate Professor of Computer Information Systems

Co-Chair of the Business Administration and Management Department

Southern New Hampshire University

**Date of Submission: April 11, 2023**

**Table of Contents**

[Question 1 1](#_Toc132062953)

[Question 2 4](#_Toc132062954)

[Question 3 7](#_Toc132062955)

[Question 4 10](#_Toc132062956)

[Additional Insight 12](#_Toc132062957)

[Appendix 15](#_Toc132062958)

# Question 1

Which language should they stock more books in?

**Answer**

**SQL Statement:**

SELECT book\_language.language\_name, COUNT(book.book\_id) AS book\_count FROM book JOIN book\_language ON book.language\_id = book\_language.language\_id GROUP BY language\_name ORDER BY COUNT(book.book\_id) DESC LIMIT 1;

**Explanation of the SQL Statement:**

The statement is selecting two columns from two tables. The first column is the language\_name column from the book\_language table, and the second column is the count of book\_id values from the book table. The AS book\_count part is giving an alias to this count column, so that it can be referred to later in the query by this name. Then the statement is specifying the tables that the query is selecting from, and also how they are joined together with ‘FROM book JOIN book\_language ON book.language\_id = book\_language.language\_id’. The GROUP BY part is grouping the results of the query by the language\_name column. This means that for each unique value of language\_name, the query will return a single row with the count of books in that language. After that, the ORDER BY is sorting the results of the query by the book\_count column in descending order. This will show the language with the highest count of books first. Finally, LIMIT 1 is limiting the number of results returned by the query to just one row, which is the language with the highest count of books.

**Why this Statement:**

This SQL statement is selecting the language with the highest count of books from Gravity books and returning the name of the language and the count of books in that language. This statement will indicate which languages book is the most famous in Gravity books and thus help the upper board to stock more books of that language.

**Result:**

|  |  |
| --- | --- |
| **language\_name** | **book\_count** |
| English | 8911 |

**Explanation of the Result:**

The result is showing that Gravity books has 8911 books in English Language which means that English is the most popular language in books there.

**Business Decision:**

The Upper Management of Gravity books should stock more books in English language as this is the most famous and popular book language in Gravity books.

# Question 2

What author should they discontinue stocking?

**Answer**

**SQL Statement:**

SELECT author.author\_name, COUNT(order\_line.order\_id) AS order\_count FROM author JOIN book\_author ON author.author\_id = book\_author.author\_id JOIN book ON book\_author.book\_id = book.book\_id JOIN order\_line ON book.book\_id = order\_line.book\_id GROUP BY author\_name ORDER BY COUNT(order\_line.order\_id) ASC LIMIT 10;

**Explanation of the SQL Statement:**

The statement line is selecting two columns from two tables. The first column is the author\_name column from the author table, and the second column is the count of order\_id values from the order\_line table. The AS order\_count part is giving an alias to this count column, so that it can be referred to later in the query by this name. The ‘FROM author JOIN book\_author ON author.author\_id = book\_author.author\_id’ part is specifying the tables that the query is selecting from, and also how they are joined together. Again, the query is joining two more tables named book and order\_line with JOIN clause to figure out the order\_count. Then, the GROUP BY is grouping the results of the query by the author\_name column. This means that for each unique value of author\_name, the query will return a single row with the count of books written by that author. And, the ‘ORDER BY COUNT(order\_line.order\_id) ASC’ is sorting the results of the query by the order\_count column in ascending order. This will show the authors with the lowest count of orders first. Finally, ‘LIMIT 10’ is limiting the number of results returned by the query to 10 rows.

**Why this Statement:**

This SQL statement is selecting the top 10 authors with the lowest count of orders from Gravity books and returning the name of the author and the count of orders written by that author. This query will help the upper board to decide which author they should discontinue stocking. By selecting the result of 10 rows, the upper board is getting more than one option to select from and taking the final decision.

**Result:**

|  |  |
| --- | --- |
| **author\_name** | **order\_count** |
| Michael Hague | 1 |
| Brian Bolland | 1 |
| Jorma-Veikko Sappinen | 1 |
| Lloyd P. Gerson | 1 |
| Rebecca Brown | 1 |
| George J. Becker | 1 |
| Margot Norris | 1 |
| James Thomas | 1 |
| Narendra Jadhav | 1 |
| Stephen Biesty | 1 |

**Explanation of the Result:**

The result is showing that these 10 authors (Michael Hague, Brian Bolland, Jorma-Veikko Sappinen, Lloyd P. Gerson, Rebecca Brown, George J. Becker, Margot Norris, James Thomas, Narendra Jadhav, and Stephen Biesty) have the least number of orders for their books in Gravity books which is 1. That means their books are not very popular with the customers and customers are not preferring to order more books by these writers.

**Business Decision:**

The Upper Management of Gravity books should discontinue stocking these 10 authors which will help them to have more room for the popular author’s books. This will lead them to get more orders, which will increase their profit.

# Question 3

Who are the 10 best customers?

**Answer**

**SQL Statement:**

SELECT customer.customer\_id, customer.first\_name, customer.last\_name, COUNT(cust\_order.order\_id) AS order\_count FROM customer JOIN cust\_order ON customer.customer\_id = cust\_order.customer\_id GROUP BY customer.first\_name ORDER BY COUNT(cust\_order.order\_id) DESC LIMIT 10;

**Explanation of the SQL Statement:**

The statement is selecting four columns from two tables. The first column is the customer\_id column from the customer table, the second column is the first\_name column from the customer table, the third column is the last\_name column from the customer table, and the fourth column is the count of order\_id values from the cust\_order table. The AS order\_count part is giving an alias to this count column, so that it can be referred to later in the query by this name. The ‘FROM customer JOIN cust\_order ON customer.customer\_id = cust\_order.customer\_id’ part is specifying the tables that the query is selecting from, and also how they are joined together. The GROUP BY part is grouping the results of the query by the first\_name column of the customer table. This means that for each unique value of first\_name, the query will return a single row with the count of orders placed by that customer. Then, the ‘ORDER BY COUNT(cust\_order.order\_id) DESC’ is sorting the results of the query by the order\_count column in descending order. This will show the customers with the highest count of orders first. Finally, ‘LIMIT 10’ is limiting the number of results returned by the query to just 10 rows.

**Why this Statement:**

This SQL statement is selecting the top 10 customers with the highest count of orders from Gravity books, and returning the customer ID, first name, last name, and the count of orders placed by each customer. To answer the question of the upper board, this query shows which 10 customers has the highest orders which will make them the best customers.

**Result:**

|  |  |  |  |
| --- | --- | --- | --- |
| **customer\_id** | **first\_name** | **last\_name** | **order\_count** |
| 31 | Tripp | Halsworth | 36 |
| 95 | Ardra | Tripony | 35 |
| 59 | Emylee | Hubbert | 32 |
| 89 | Bennie | Mepsted | 30 |
| 116 | Shae | Leser | 28 |
| 96 | Lannie | Michurin | 28 |
| 118 | Jocelyn | Budding | 28 |
| 353 | Lorita | Underdown | 28 |
| 100 | Dud | Pusill | 27 |
| 6 | Kandy | Adamec | 27 |

**Explanation of the Result:**

The result is showing that These 10 customers with the customer id of 31, 95, 59, 89, 116, 96,118, 353, 100, and 6 have the greatest number of orders. These are the customers who mostly order books and thus they are the 10 best customers.

**Business Decision:**

The upper board can arrange special offers for the 10 best customers or arrange priority package for them which will motivate them to purchase more books.

# Question 4

Gravity Books is looking to add a warehouse to help optimize shipping. Where should they locate that warehouse?

**Answer**

**SQL Statement:**

SELECT country.country\_name, COUNT(cust\_order.order\_id) AS order\_count FROM country JOIN address ON country.country\_id = address.country\_id JOIN cust\_order ON address.address\_id = cust\_order.dest\_address\_id GROUP BY country\_name ORDER BY COUNT(cust\_order.order\_id) DESC LIMIT 1;

**Explanation of the SQL Statement:**

The statement is selecting two columns from three tables. The first column is the country\_name column from the country table, and the second column is the count of order\_id values from the cust\_order table. The AS order\_count part is giving an alias to this count column, so that it can be referred to later in the query by this name. The ‘FROM country JOIN address ON country.country\_id = address.country\_id JOIN cust\_order ON address.address\_id = cust\_order.dest\_address\_id’ part is specifying the tables that the query is selecting from, and also how they are joined together. The JOIN keyword is used to join the country, address, and cust\_order tables. The country\_id column is used to join the country table with the address table, and the address\_id column is used to join the address table with the cust\_order table. The GROUP BY part is grouping the results of the query by the country\_name column of the country table. This means that for each unique value of country\_name, the query will return a single row with the count of orders that were shipped to that country. And, the ‘ORDER BY COUNT(cust\_order.order\_id) DESC’ part of the statement is sorting the results of the query by the order\_count column in descending order. This will show the country with the highest count of orders first. Finally, LIMIT 1 is limiting the number of results returned by the query to just 1 row.

**Why this Statement:**

This SQL statement is selecting the name of the country that has received the highest count of orders from gravity books, and returning the name of the country and the count of orders that were shipped to that country. This query will show to which country Gravity Books are shipping most of their orders. Thus, they will be able to select that country to add a warehouse that will help them to optimize shipping.

**Result:**

|  |  |
| --- | --- |
| **country\_name** | **order\_count** |
| China | 1266 |

**Explanation of the Result:**

The result is showing that Gravity books make the highest number of shipments in China than other countries which indicates a warehouse in China can be very profitable for them.

**Business Decision:**

The upper management can select the country China as their next warehouse location. This will help them to optimize shipping.

# Additional Insight

From evaluating the database, I found two additional insights that can be beneficial for the company Gravity books in the long run. Following are the two insights:

* **Revenue Calculation**

According to me, the order\_line table in the database needs some modification which will be helpful for the company. The order\_line table has line\_id, order\_id, book\_id and price columns. An additional column with the quantity of books ordered could be very informative. The column can be named as ‘order\_quantity’ and the data type of this column can be ‘INT(11)’. From this column, Gravity books can easily figure out how many numbers of books are in each order\_id.

Also, it will be easier for them to calculate the revenue generated by each order as they will get the price and quantity together. Other than that, any other query regarding revenue, order and price will be more specific with this additional information.

* **Publisher**

Top 5 publishers whose books are mostly ordered in Gravity books.

**SQL Statement:**

SELECT publisher.publisher\_name, COUNT(order\_line.order\_id) AS order\_count FROM publisher JOIN book ON publisher.publisher\_id = book.publisher\_id JOIN order\_line ON book.book\_id = order\_line.book\_id GROUP BY publisher.publisher\_name ORDER BY COUNT(order\_line.order\_id) DESC LIMIT 5;

**Explanation of the SQL Statement:**

The statement is selecting two columns from three tables. The first column is the publisher\_name column from the publisher table, and the second column is the count of order\_id values from the order\_line table. The AS order\_count part is giving an alias to this count column, so that it can be referred to later in the query by this name. The ‘FROM publisher JOIN book ON publisher.publisher\_id = book.publisher\_id JOIN order\_line ON book.book\_id = order\_line.book\_id’ part is specifying the tables that the query is selecting from, and also how they are joined together. The GROUP BY part is grouping the results of the query by the publisher\_name column of the publisher table. This means that for each unique value of publisher\_name, the query will return a single row with the count of orders that has books published by the publishers. And, the ‘ORDER BY COUNT(order\_line.order\_id) DESC’ part of the statement is sorting the results of the query by the order\_count column in descending order. This will show the country with the highest count of orders first. Finally, LIMIT 5 limits the number of results returned by the query to 5 rows.

**Result:**

|  |  |
| --- | --- |
| **publisher\_name** | **order\_count** |
| Vintage | 209 |
| Penguin Books | 165 |
| Penguin Classics | 127 |
| Mariner Books | 104 |
| Ballantine Books | 103 |

**Explanation of the Result:**

The result shows that Gravity books get the highest number of orders of the books published by Vintage, Penguin Books, Penguin Classics, Mariner Books, and Ballantine Books which is consecutively 209, 165, 127, 104 and 103.

**Business Decision:**

Gravity books can build stronger relationships with these 5 top publishers, which can lead them to better pricing, more favorable payment terms, and other benefits. This will help them to reduce costs and improve profitability.

# Appendix

**Entity Relationship Diagram**

Diagram

Description automatically generated