



Rockbuster Stealth Data Analysis

Ally Fornino

Background

This project was a part of the CareerFoundry Data Analytics course and was centered around a fictional international movie rental company, Rockbuster Stealth.

The company used to have physical locations, but due to competition from streaming services they are now planning to use existing licenses to launch a new online rental service. Their business intelligence (BI) department is in the process of creating a launch strategy, but needs more information about their customer base.

Data and Tools

The dataset used in this project is a PostgreSQL database that includes transaction, customer, inventory, and film data.

pgAdmin 4 (an open-source management tool for PostgreSQL) was used to clean and summarize the data.

Tableau was used to create visualizations of the data.

Timeline

This project was completed in 3 weeks



KEY QUESTIONS

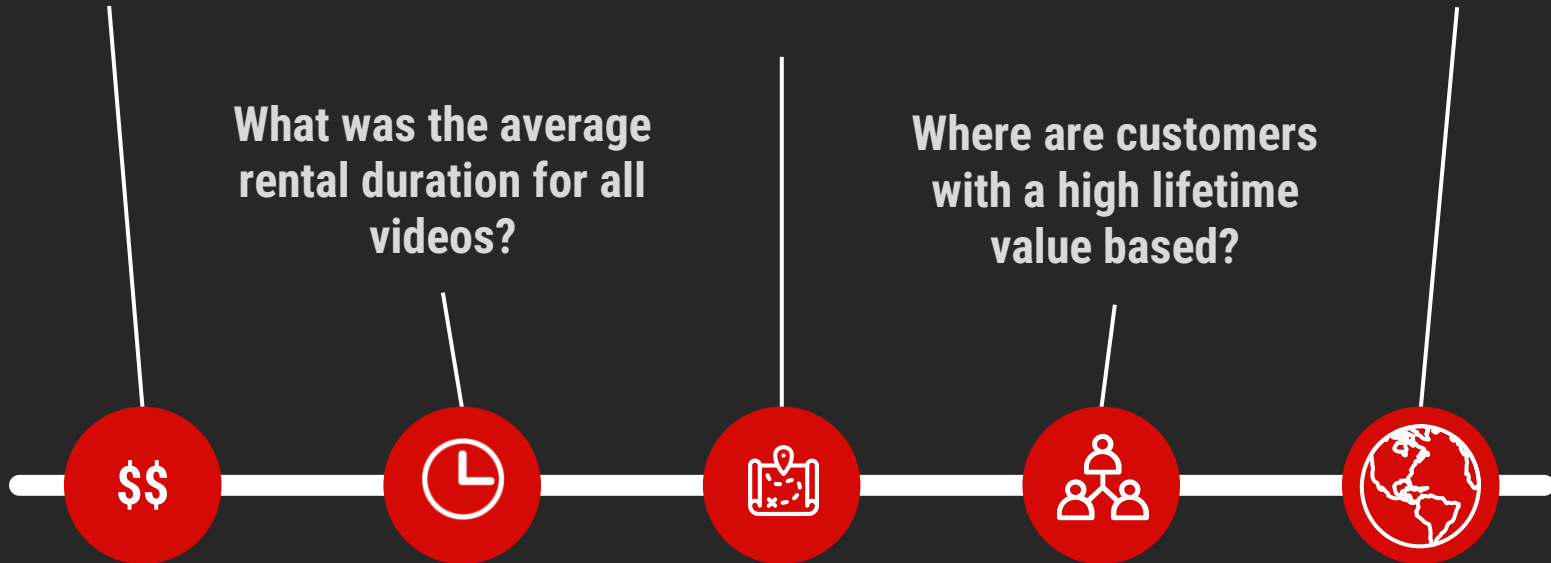
Which movies contributed the most/least to revenue gain?

Which countries are customers based in?

Do sales figures vary between geographic regions?

What was the average rental duration for all videos?

Where are customers with a high lifetime value based?



Goal

- Provide data-driven answers to the BI department's key business questions by cleaning and summarizing the provided data with SQL queries/subqueries.
- Create deliverables to present results to BI department using Tableau.

Understanding the Data

This project was my first time using a relational database as the source data; therefore, the first step was to understand how the data was stored and create references to use throughout the project (in this case, a data dictionary and an entity relationship diagram).

An entity relationship diagram (ERD), shown on the right, is a schematic that displays the structure of a relational database by showing the links between tables. This allows viewers to quickly see what information is held by each table and how the tables are connected.

A data dictionary is a reference document that contains the database's metadata (table names, what information is stored in those tables, and how the tables are linked). It is used to find out which tables and columns contain the information needed when conducting an analysis. The data dictionary created for this analysis can be seen [here](#).



Figure 1: The ERD for the Rockbuster Stealth database. An enlarged version is available in Page 3 of the Data Dictionary.

Cleaning and Summarizing the Data

The next step was to prepare the data for analysis by cleaning and summarizing the database.

Data Cleaning

To clean the data, the database was checked for duplicate, non-uniform, and missing data.

○ Duplicate Data:

For duplicate data, a view* can be created to show the duplicates.

If there were duplicates present, then a view that only contains distinct data would be created or the UNIQUE keyword would be used when fetching data.

There were not any duplicates in the database so no action was needed.

A view is a temporary, table-like structure that shows data in real time and saves storage space

○ Non-Uniform Data:

If there were any inconsistencies in how values were entered into the database, they would be identified by using the GROUP BY clause. Then, the UPDATE command would be used to change any variations into one value.

There were not any non-uniform values in the database.

○ Missing Data:

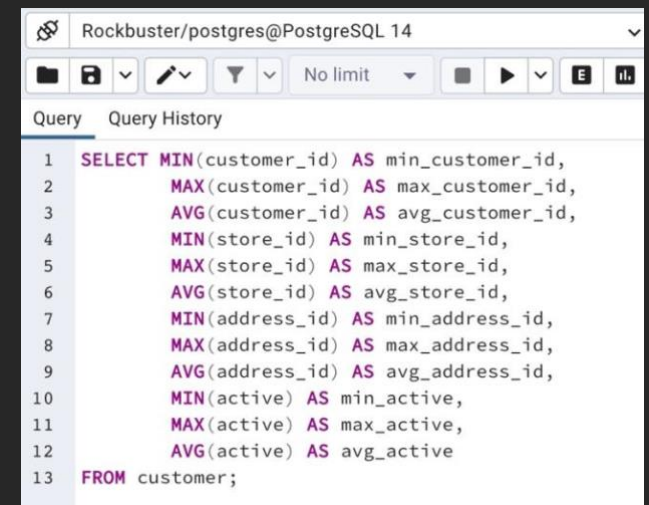
There were not any missing values in the database.

If a particular column in the database had data missing, the following actions could have been taken:

- Significant amount of data missing → omit column in SELECT statement
- Small percentage of data missing → values can be imputed using statistical methods, such as replacing it with the average of the present data

Summarizing the Data

To better understand the data before conducting the analysis as well as creating an overview of the data for non-analysts, a summary of each variable in the film and customer tables were created. For numerical values, this included the minimum, maximum, and average and for the non-numerical values, the mode for each variable was found. The query written to summarize the numerical values from the customer table is shown on the right. The full overviews can be found in the first two tabs of [this Excel workbook](#).

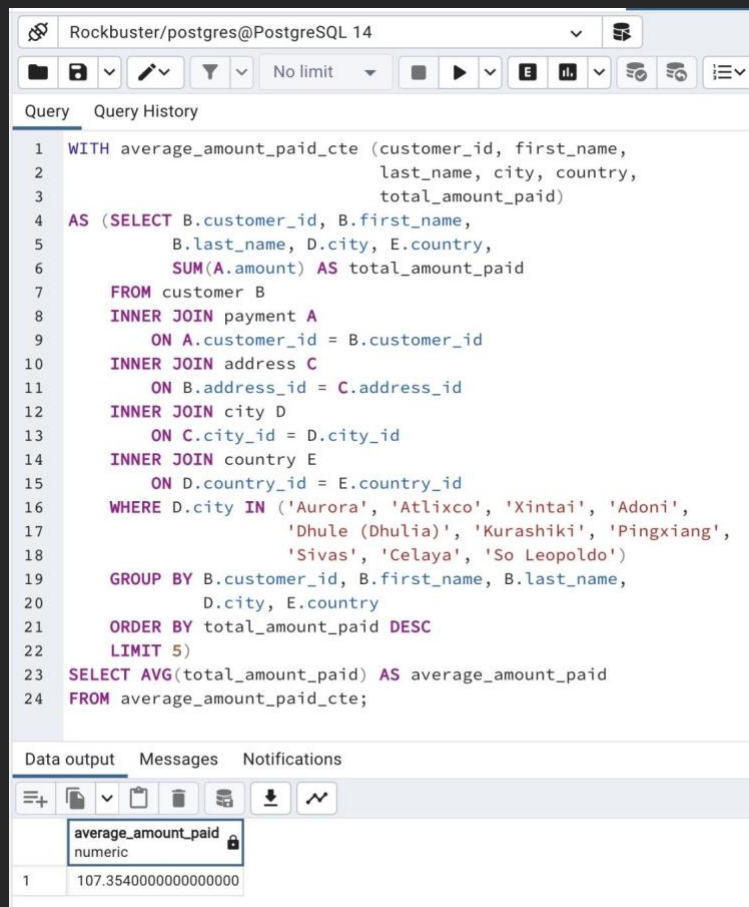


```
Rockbuster/postgres@PostgreSQL 14
Query History
1 SELECT MIN(customer_id) AS min_customer_id,
2      MAX(customer_id) AS max_customer_id,
3      AVG(customer_id) AS avg_customer_id,
4      MIN(store_id) AS min_store_id,
5      MAX(store_id) AS max_store_id,
6      AVG(store_id) AS avg_store_id,
7      MIN(address_id) AS min_address_id,
8      MAX(address_id) AS max_address_id,
9      AVG(address_id) AS avg_address_id,
10     MIN(active) AS min_active,
11     MAX(active) AS max_active,
12     AVG(active) AS avg_active
13 FROM customer;
```

Figure 2: The query for summarizing numerical values in the customer table. The full summaries for the film and customer tables can be found in the Excel workbook.

Analyzing the Data

Below is an outline of how a CTE was used to find the average amount paid by the top 5 customers in the top 10 cities:



```
1 WITH average_amount_paid_cte (customer_id, first_name,
2                               last_name, city, country,
3                               total_amount_paid)
4 AS (SELECT B.customer_id, B.first_name,
5           B.last_name, D.city, E.country,
6           SUM(A.amount) AS total_amount_paid
7       FROM customer B
8       INNER JOIN payment A
9           ON A.customer_id = B.customer_id
10      INNER JOIN address C
11           ON B.address_id = C.address_id
12      INNER JOIN city D
13           ON C.city_id = D.city_id
14      INNER JOIN country E
15           ON D.country_id = E.country_id
16      WHERE D.city IN ('Aurora', 'Atlixco', 'Xintai', 'Adoni',
17                     'Dhule (Dhulia)', 'Kurashiki', 'Pingxiang',
18                     'Sivas', 'Celaya', 'So Leopoldo')
19      GROUP BY B.customer_id, B.first_name, B.last_name,
20              D.city, E.country
21      ORDER BY total_amount_paid DESC
22      LIMIT 5)
23 SELECT AVG(total_amount_paid) AS average_amount_paid
24 FROM average_amount_paid_cte;
```

The screenshot shows the query editor interface with tabs for Query, Query History, Data output, Messages, and Notifications. The Data output tab is active, showing a table with one row and one column: 'average_amount_paid' with a numeric value of 107.35400000000000.

Figure 3: The query written to find the average amount paid by the top 5 customers in the top 10 cities. This was achieved by creating a CTE.

In order to answer the business questions outlined by the BI department, the following methods were used to explore and manipulate the data:

- **JOIN:** used to combine tables when information held by multiple ones is needed
- **Subqueries with SELECT clauses:** used to create a new column in data output
- **Common Table Expressions (CTEs):** used to create a temporary table that can be referenced in the main query

Step 1: Locate the data needed in the ERD (or Data Dictionary)

- Determine which data is necessary to write this query, including the tables (payment, customer, address, city, and country) and how each are linked (primary/foreign keys).

Step 2a (Lines 1-3): Define the CTE by using a WITH clause

- In this case, the CTE's name is average_amount_paid_cte and it will contain the columns listed in the parentheses.

Step 2b (Lines 4-22): Finish creating the CTE with the AS keyword

Lines 4-6 : The SELECT statement started in line 4 outlines what the CTE will contain: the customer's id, first and last names, city and country, and the sum of all amounts paid by this customer which will appear as the column 'total_amount_paid'.

Lines 7-18 : This information is being retrieved from the following tables that have been combined using INNER JOINS: customer, payment, address, city, and country.

For the city table, however, we only want to consider customers that are in the top 10 cities for total sales, so a WHERE clause is used to only include the cities listed in the parentheses (lines 16-18) which had been found in a previous task.

Lines 19-22 : To complete the CTE, the GROUP BY clause is used to organize the records so that each row is one customer. Then the top 5 customers are found by using the ORDER BY clause with the records being listed by the total amount paid value in descending order. Because we only want the top 5 customers, the LIMIT keyword is used at the end of the statement.

Step 3 (Lines 23-24): Writing the main query

- The final step is to write the main query. In this case, we want to determine the average amount paid by the top 5 customers.
- Because we already have the top 5 customers' information stored in the CTE, all we have to do is write a SELECT statement that fetches the average total amount paid value from the CTE
- The output is this average value under the column name 'average_amount_paid'

Visualizing the Data

The next step was to visualize the data with Tableau. Below is an outline of how I answered one of the key business questions with a visualization as well as how I expanded upon it to further explore spending habits of the top customers.

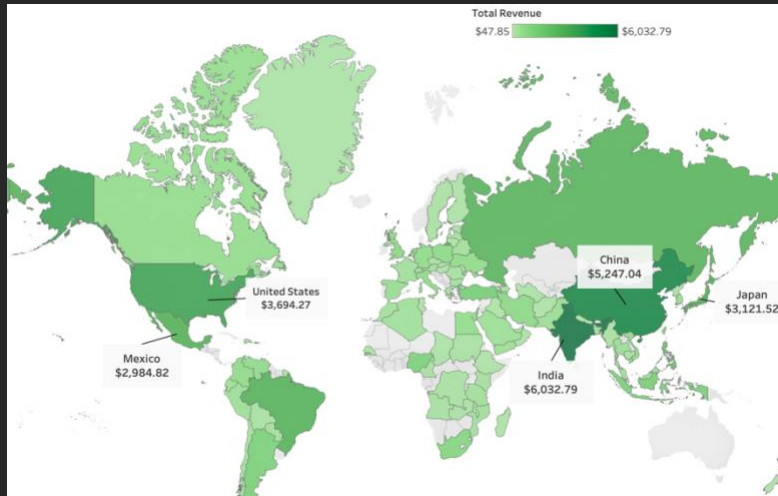


Figure 4: The choropleth map created to display each country's total revenue.

In order to answer the key question of 'Do sales figures vary between geographic regions?', I originally created a choropleth map (shown on the left) that displayed the total sales for each country for easy comparison.

However, when it came to comparing spending habits in the top customers, I struggled on how I could display them in a way that was easy to understand and showed their individual locations while still being able to compare them to the overall revenue trends.

I decided to create a combination map that still showed the revenue for each country, but also had each top customer shown as a circle at their location. The size and color of the circle would represent their total payment history and their average order amount, respectively.

This was effective in presenting all of the information, but was not the most readable as the full choropleth map underneath the callouts was a little distracting. In order to remedy this, I limited the choropleth to only include the top 10 countries. This made the visualization easier to understand while also highlighting that only 2 of the top 5 customers were from countries that earned the highest revenues, displaying that the company's customers with a high lifetime value were not necessarily based in the highest earning countries.

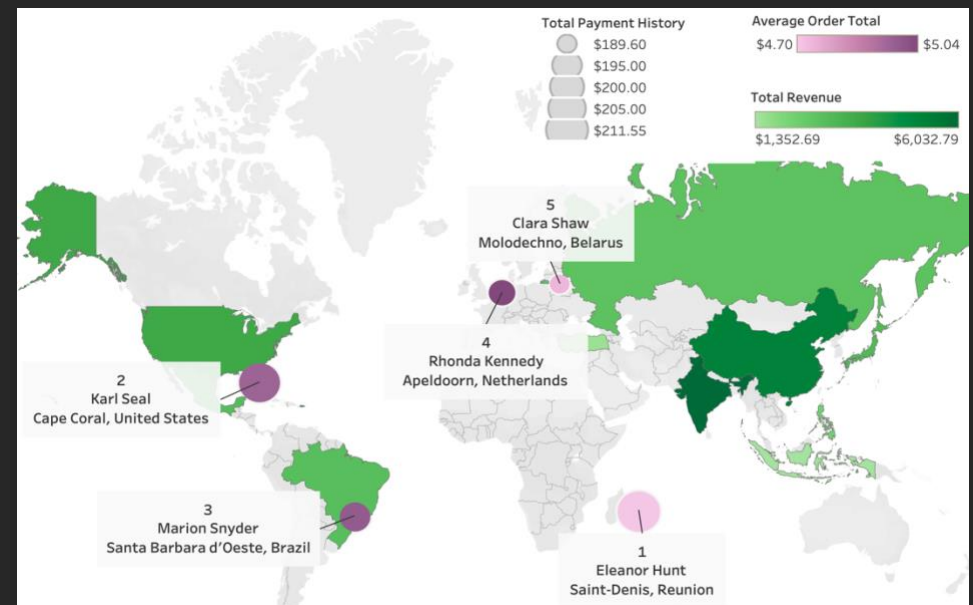


Figure 5: The final combination map used to compare the top customers' spending habits with each other as well as their geographic regions.

Presenting the Results

The final step was to present the results of the analysis in a clear, easily digestible format.

A presentation was created that outlined the project, including the key questions and answers along with a company overview, visualizations for each answer, and my data-driven recommendations for the company in designing their launch strategy.

The presentation can be viewed [here](#).

An Excel workbook, as referenced earlier, was also created to display the SQL queries written throughout the project and their data outputs. It also holds the summaries for each table in the database.

The workbook can be viewed [here](#).

The Data Dictionary, also referenced earlier, has the ERD and metadata of the database.

The data dictionary can be viewed [here](#).

All of the Tableau visualizations were compiled into a dashboard.

The Tableau dashboard can be viewed [here](#).

All project materials can also be viewed through a GitHub repository.

The repository can be viewed [here](#).

Thank you for reading!