Project 9: Visualization of Box Office Data of Top Grossing Movies

Team 3: David Henningsen, Alexander Hintz, Chamal Kayssar, Ziru Wang

Introduction

The goal of Project 9 was to find an interesting public dataset to transform into a relational model onto the database and then analyze the data in an interesting way. For this project, we decided to focus on movie box office data for top grossing movies that have premiered in the US and internationally. The goal of the project was to create a GUI that could allow a user to visualize box office data in charts and graphs such as comparing the effect of budget data to profit. Because visualization through graphs is a meaningful and significant way to analyze data in general, we wanted to make the GUI as adaptable to other datasets as possible. Although we specifically looked at movie data for this project, the GUI can be expanded to other datasets as well.

The Data

We chose to do our project with movie data, specifically Movie Budget vs. Worldwide Gross Income and specific country Gross Income. With current Hollywood/Movie industry trends, where certain movies can generate billions of dollars in sales, we were interested to see if there is a correlation between movies with a big budget and the revenue generated. Additionally, since box office trends tend to differ from country to country, we wanted to be able to look at data in each country separately, but also compare them as a whole. Therefore, we wanted a dataset that had total sales data and sales data broken down by country.

The goal of the project was to analyze box office trends, so we chose to focus only on the top 300 grossing movies of all time. The specific countries we focused on were the USA, Australia, the UK, and China, which are generally countries that take up a large share of a movie's total income because of their large movie markets. These are also countries that we observed to have the most completeness in box office data; many other countries only had sales data available for a few movies, whereas these countries had data available for most movies. Since we wanted to look at trends, we needed consistency in the data available and this was best met by the four countries chosen. However, the dataset can be expanded to analyze other countries' box office trends if more information becomes available.

The data imported into the database is from IMDB (International Movie DataBase) and its affiliated site BoxOfficeMojo, which records the box office earnings of movies by USD. The movie titles, release year, and budget data was parsed from IMDB's plain text database. Total and individual country box office data was compiled from each movie's page on BoxOfficeMojo.com. These two datasets were then converted to a csv file and transferred to the database using the *Copy* command in Postgres.

We chose data sets from IMDB and BoxOfficeMojo for three main reasons:

1. Reliability and completeness of data due to IMDB being a highly credible source for movie information.
2. The size of the dataset is large enough to allow expandability of our project into different comparison fields.
3. Ease of parsing file data and transferring into the database as IMDB provided text files available for downloading .

While other databases may contain the the needed information for one part of our project, one main goal of this project is to implement more comparisons than simply between budget and gross values; while we may not have time for further implementation beyond those two value sets in this project, having the available data to expand if time allows is never a bad thing. The abundance of movie data on IMDB and BoxOfficeMojo allows us to easily expand the dataset.

License Restrictions

As to the licensing of the data, IMDB allows personal use of their plain text data available through *http://www.imdb.com/interfaces* as is stated on their website here *http://www.imdb.com/help/show_leaf?usedatasoftware*. Similarly, BoxOfficeMojo also allows non-commercial use of their data. Licensing information is available at *http://www.boxofficemojo.com/about/termsofuse.htm*. All data used is legally available assuming this project will not be distributed and the database tables loaded with the data will not be used beyond this project. Also, because this project is for educational purposes, use of the data falls under fair use in these sites' policies.

Tables

Two tables adhering to the relational model were created to import the data collected into the database. The first table is titled *Total_Box_Office* and contains the title of each movie, an ID associated with the movie, the year it was released , the total box office gross as a *bigint* data type, and the budget of the movie as a *bigint* data type.

A second relation is called *International_Box_Office* which we created under the assumption that a movie has sales that generate profit. This table contains box office gross for individual countries. Each tuple in the data has a movie ID as a foreign key from *Total_Box_Office* that allows us to relate back to that table, a country column that indicates the country the data was taken from as a text entry, and the gross box office

income associated with the country as a *bigint* data type. The UML diagram the design of the tables is based upon is shown in Figure 1 below.

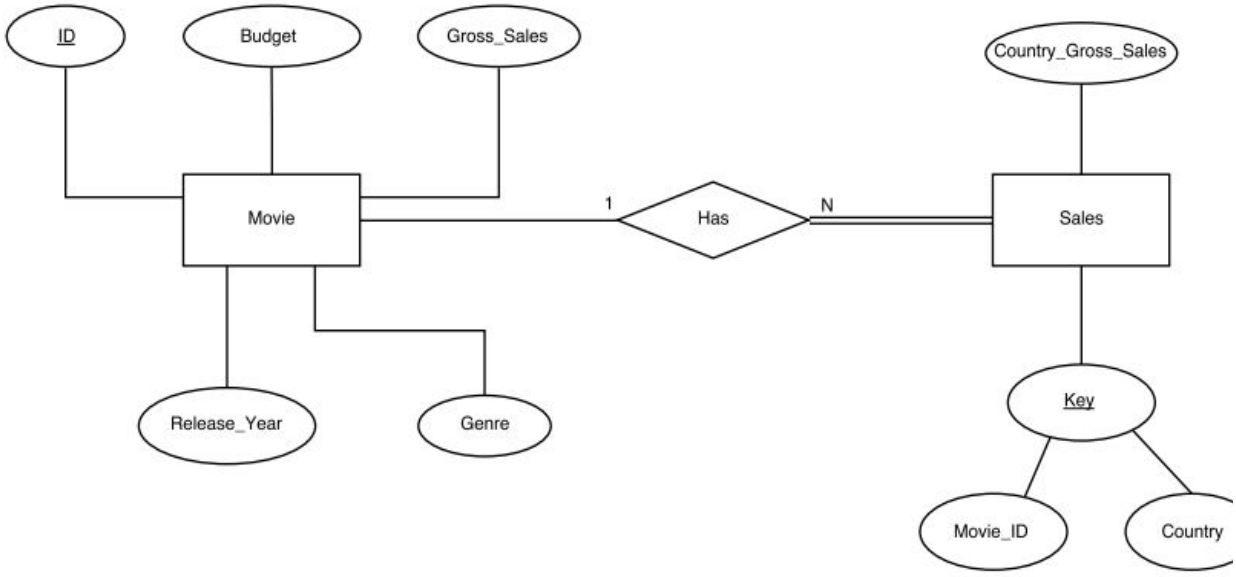

Figure 1: Rough UML diagram of Movie tables

We believed that this was the best way to set up the tables for our project because it would allow us to look at data from each country individually, but also allow us to compare the data as a whole. The setup of our data in this way also allows us to easily import additional budget and country data in a few queries.

Visualization of Data

For this project, we decided that a meaningful way to process the data is to create a GUI in which the user can visualize box office trends in each country. We programmed in Java and used the JFreeChart graphing library to graph the data. Specifically, we wanted to see a graphs showing the market share different countries have on a movie and the trend between budget and profit a movie yields.

The user is able to choose the data he/she would like to see presented in a table via the drop down menu at the top. After choosing the attributes that the user would like to see plotted, he/she also has the option to filter out the data to look at specific movies or specific attribute values, such as all movies that contain ‘Harry Potter’ in the title.The search button runs a query with the requirements chosen by the user and lists the

resulting relation with values for each attribute. This functionality is displayed in Figure 2.

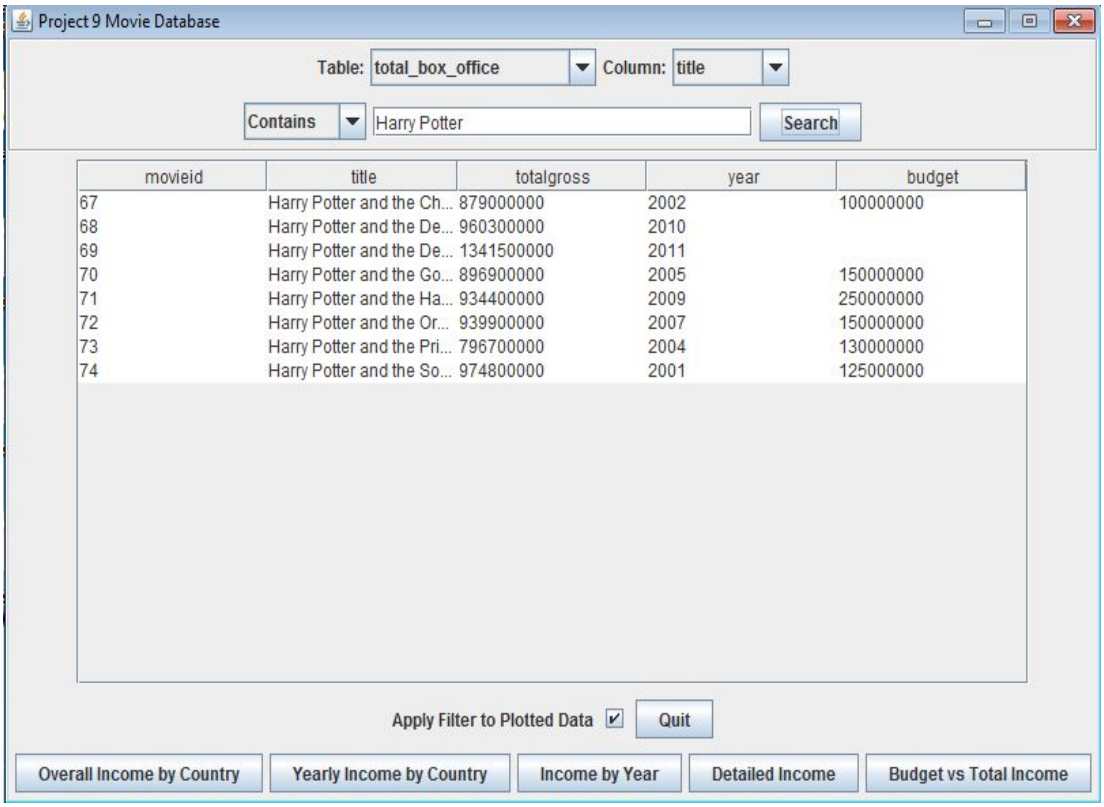

Figure 2: Plotting Menu of Movie Data

For quick access, the GUI has five default graphs the user may want to see. With a click of each button, the user can see generated charts such as a pie chart showing the breakdown of income for each country (Figure 3), a bar graph separating movie income by country (Figure 4), box office income separated by year of release (Figure 5), breakdown of total movie income by country (Figure 6), and Budget vs. Income (Figure 7) to observe trends. These charts take into account of any filter applied at the top of the window, but can be disabled if desired.

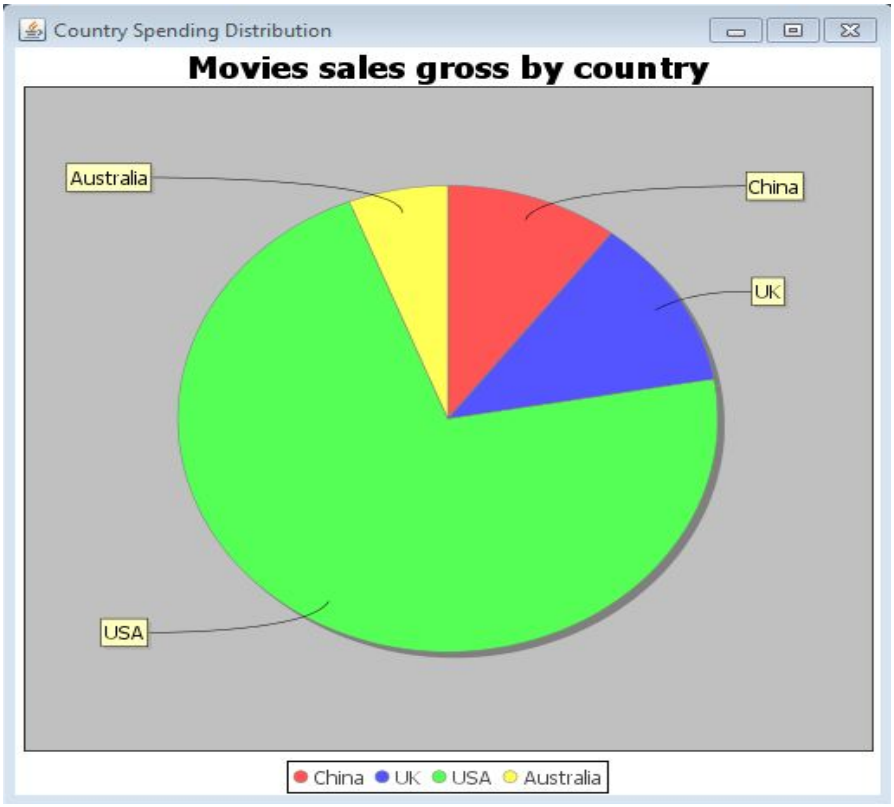

Figure 3: Pie Chart of Income by Country

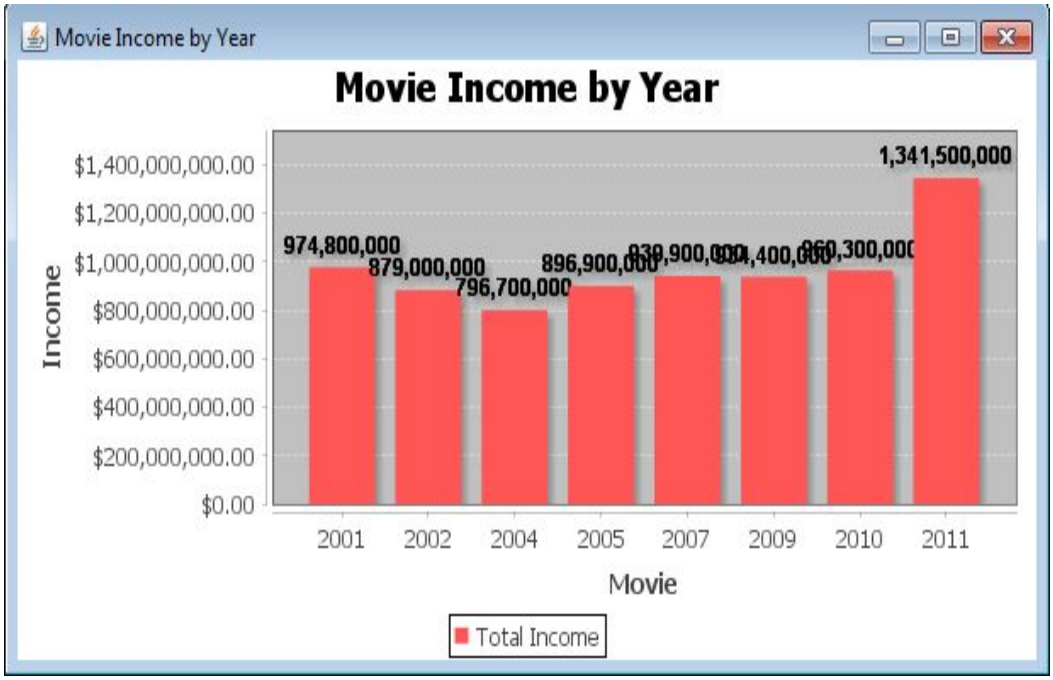

Figure 4: Movie Income by Year

.

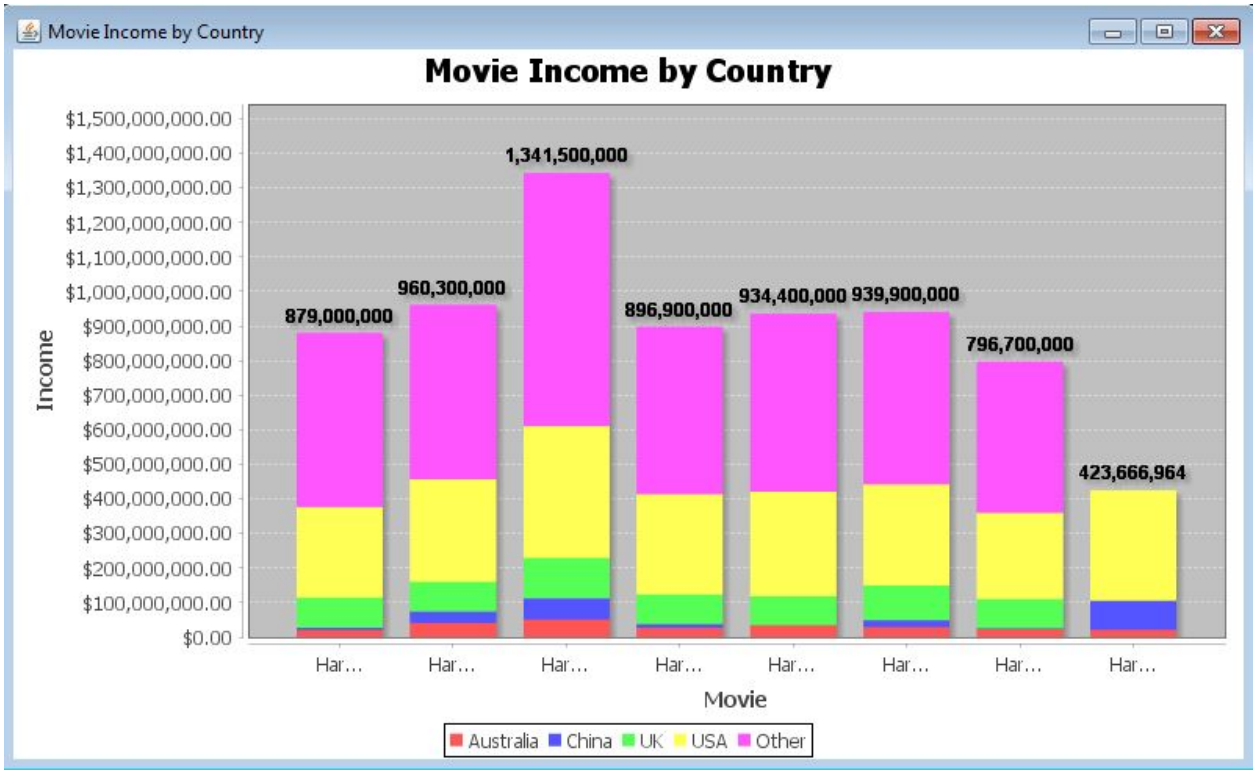

Figure 6: Movie Income by Country

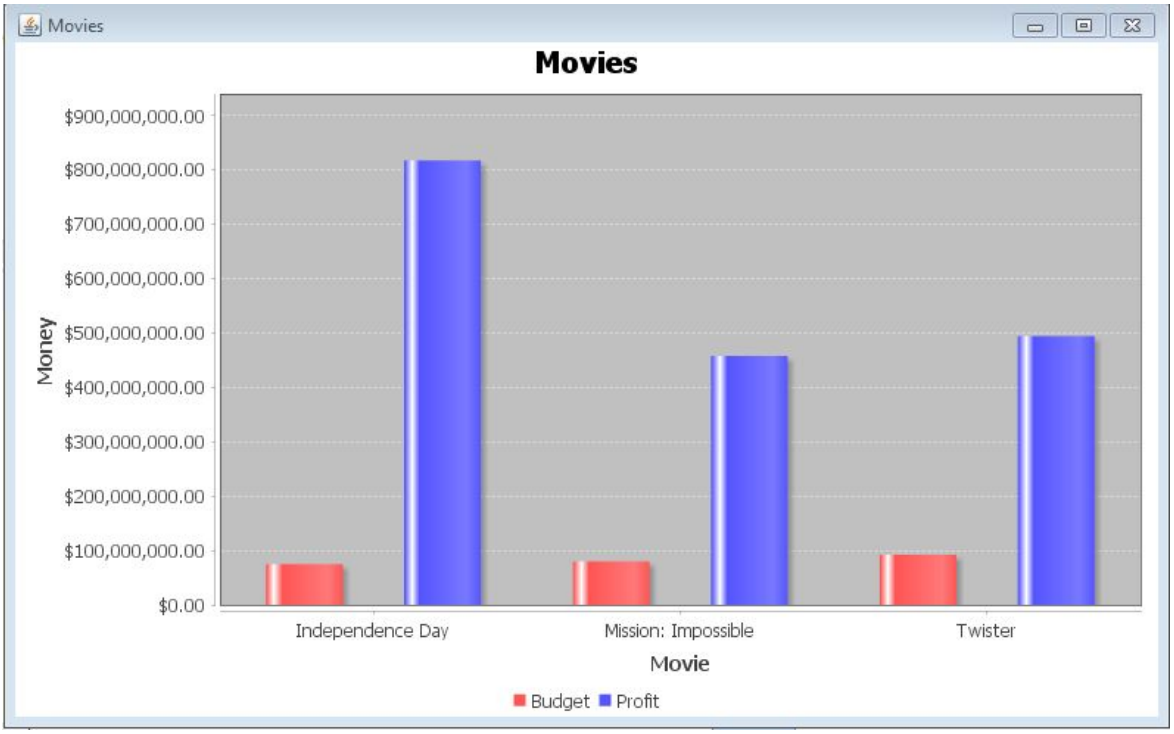

Figure 7: Budget v. Profit for movies

The way that the GUI is set up allows for analysis of datasets other than box office data. The gui can be further adapted to look at trends between two other sets of data; it is not limited simply budget vs. gross income. For example, for the movie data tables, new data such as advertising investment could easily be added to the tables and will appear as a drop down option in the GUI and will be available to filter. As an extension of this project, one may also want to pursue completely different datasets to look at the correlation between two other attributes. While the buttons for default graphs use prepared queries, the code contains the option to create custom queries in a secure manner, so that any data can be plotted, even if a new table is added.

Technical Challenges

The main technical challenges in processing the data was compiling the dataset into one file. IMDB and BoxOfficeMojo provided different data organized different ways, so we had to manually organize the data so that it would fit how we designed the tables. For example, IMDB provided budget data for it's top 1500 movies in 2016, while BoxOfficeMojo provided data for all movies, including currently showing movies. This caused a discrepancy in how the data was organized, so we had to manually go through and remove movies that were in the BoxOfficeMojo dataset but not in the IMDB dataset. Because of this reason, we chose to focus on the top 300 grossing movies on BoxOfficeMojo. The list and values for these movies stayed fairly consistent between the two data sources.

Another challenge in processing the data was that the box office data for each country was listed under each individual movie and not available as an individual dataset. A lot of manual work was required to look through the movies and compile the country data into one. As a result of this, we chose only to focus on countries that take a large market share. The box office revenue that all other countries make up can be calculated with a simple query that adds up the individual country data and subtracts it from the total gross data found in the *Total_Box_Office* relation. Additional analysis could be done off of this, such as calculating the percent market share a country takes up. However, based on the design of tables, it is a possibility to add in additional country data as additional tuples with a foreign key relating back to its associated movie. The design of the model for the movie data should be adaptable to add new relations and data as needed.

After the data was compiled, loading the data was fairly simple. The files with the data were converted to a CSV file. Data was imported into the Postgres database via the COPY command and adhering to the relational model.

A final problem involved the queries for the database. This included making sure that a table or column existed, which involved dynamically looking at the database

metadata. Also, creating the query in such a way that all tables were mentioned where needed without repeating was a problem, which is why most of the queries have some sort of pre-defined start. Finally, parsing the return data sets in such a way that it could be plotted proved to be a semi-difficult endeavor. While the JFree plotting library made this easy for simple data, most of the predefined plots required post processing that are specific to the data being returned, and therefore could not easily be expanded to other datasets, despite our intentions of making this project as versatile as possible.

Conclusion

Through this project we were able to visualize the trends between box office budget data and Gross revenue for individual countries. Based off this data, we were unable to determine whether or not a movie's budget was correlated to its success, because no definite trend was observed. For example, looking at Figure 7, Independence Day, which has the smallest budget, actually has the largest income. However, this project was a very interesting way to look at movie box office data and the market share each certain countries take up. Also, we believe that the GUI created in this project can be useful to visualize other datasets and can be modified to determine other trends in data, making it a very practical program.