CALIFORNIA STATE UNIVERSITY, FRESNO

CSCI 126

YOUTUBE DATABASE REPORT

Jose Rodolfo Nieblas, Emmy Tran

May 10th, 2023

TABLE OF CONTENTS

PHASE 1

Formalizing The Problem

In this project, we wanted to ask the following questions.

- What trending channel had the most views in the year (2020-2021)?
- Which videos have the highest number of views in each category?
- Is there a correlation between the day of the week a video is published and its success?
- How does the length of a video's title impact its engagement on YouTube?

We hoped to answer these questions through the datasets we gathered. More specifically, to answer these questions we retrieved YouTube data from trending videos and data about each country's population given at any year. We found that this data was sufficient enough to meet our needs in answering these questions.

Data Collection

Our data sets were derived from the following weblinks.

- https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset
- https://data.worldbank.org/indicator/SP.POP.TOTL
- https://data.worldbank.org/indicator/SP.POP.TOTL.MA.IN
- https://data.worldbank.org/indicator/SP.POP.TOTL.FE.IN

Some attributes were renamed for easier readability, but the important changes can be found in attributes that we decided to remove or add. For the YouTube trending video dataset, we removed: Thumbnail_link, Tags, and Description. The only addition to this dataset was the attribute Country_name. As for the population dataset, we only took three tables. A table holds the population of each country per year. These three tables are distinctive in the way that one shows the total population, and the others show the total male and female population accordingly. We only removed the attribute "indicator code" as it's only purpose was to indicate the table's code, meaning that this code tells us what specific table we are looking at. Furthermore, we had to rearrange the columns for each table because the attribute names were initially numbers such as: 1970, 1971, 1972, etc. After discovering that attribute names can not be purely numbers in MySQL, that is when we decided to make the change. This enlarged the records in our final table because we created the attribute "year" and stacked all the years into one column with its corresponding population in another column for each specific country. The population attribute was also created due to this. Finally, I should also mention that the rest of the original attributes were renamed, not far from their original title, only made for better readability.

PHASE 2

Entity Relationship Diagram

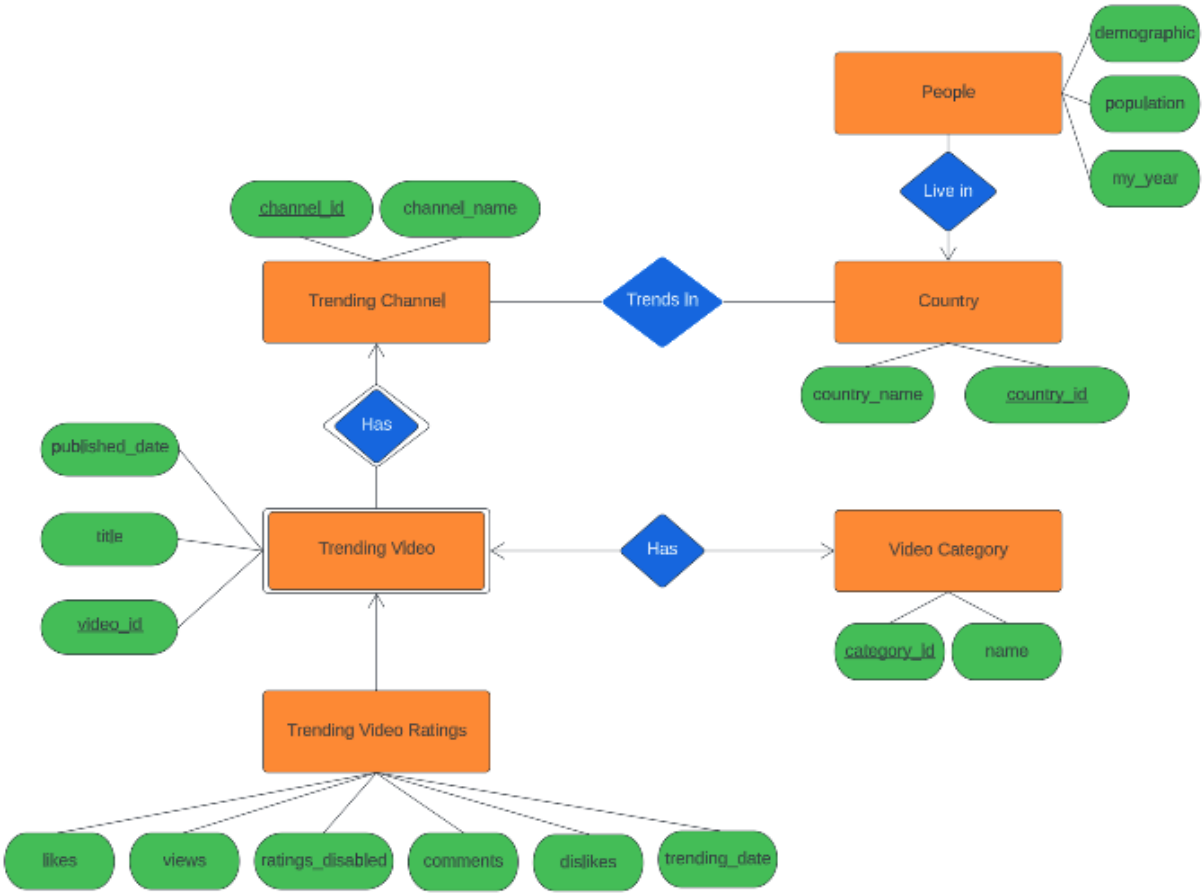
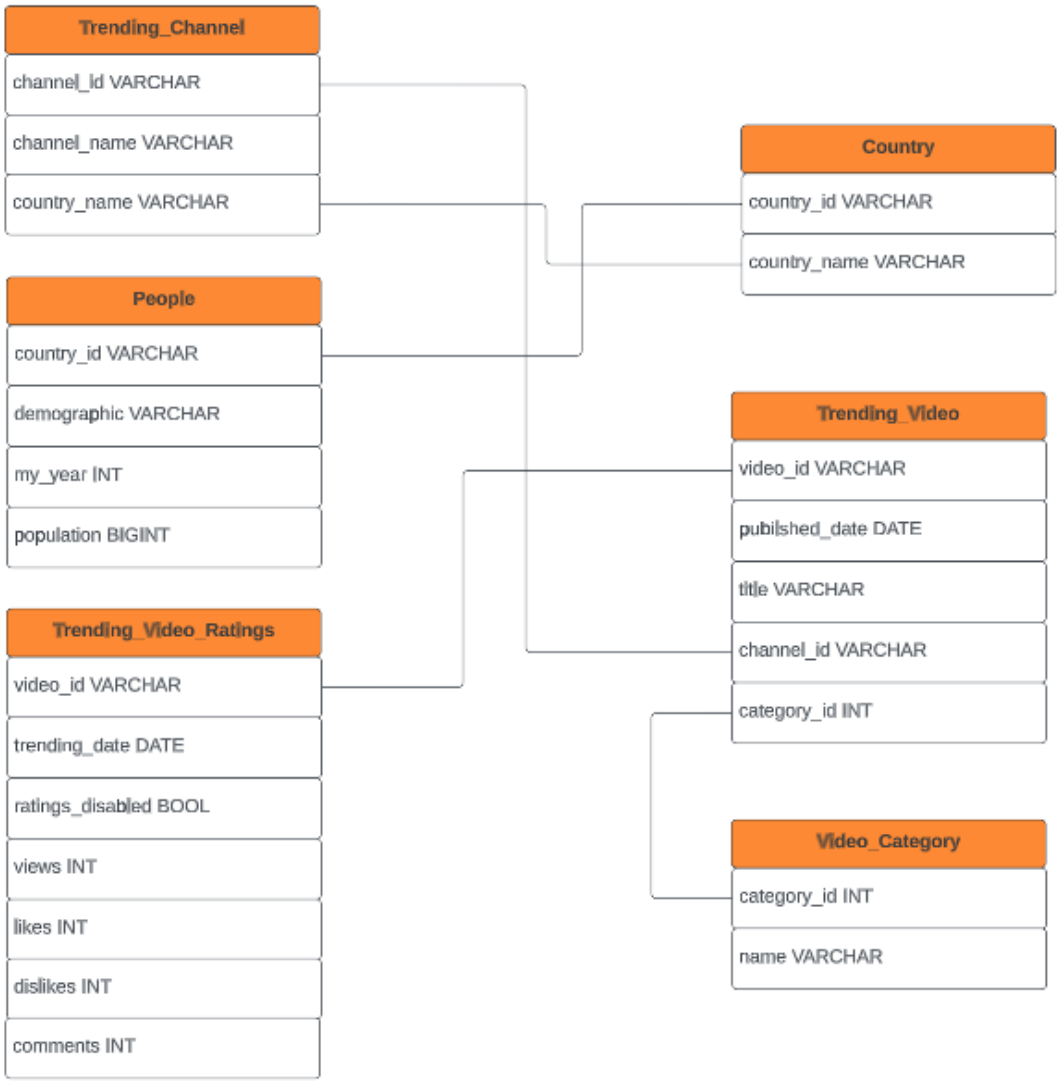


Diagram Exposition

The “Trending Channel” entity has a many-to-one relationship with the “Trending Video” weak entity because one YouTube channel is allowed to post many videos. This supporting relationship is due to the fact that a trending video can only be identified through the YouTube channel that posted it. The “Trending Video” entity has a one-to-one relationship with the “Video Category” entity because every video must belong to a category. The “Trending Video” entity also holds a many-to-one relationship with the “Trending Video Ratings” entity due to the fact that a trending YouTube video can trend on YouTube on many different occasions. The “Trending Channel” entity has a many-to-many relationship with the “Country” entity because a YouTube channel, that is one who owns trending videos, can trend in many different countries. Finally, the “Country” entity holds a many-to-one relationship with the “People” entity as each country contains a population of people from many different years.

PHASE 3

Database Schema Graph



This graph provides a visual image of how our database is interconnected while showcasing the schema. The orange bar provides the name of the table, while the white bars provide its corresponding attributes and its type. The lines that connect specific attributes together provide the referential integrity constraints.

Database Schema Code

```
CREATE TABLE Country (
    country_id VARCHAR(255) PRIMARY KEY,
    country_name VARCHAR(255)
);
```

```
CREATE TABLE People (
    country_id VARCHAR(255),
    demographic VARCHAR(255),
    my_year INTEGER,
    population BIGINT,
    PRIMARY KEY (country_id, demographic, my_year),
    FOREIGN KEY (country_id) REFERENCES Country(country_id)
);
```

```
CREATE TABLE Trending_Channel (
    channel_id VARCHAR(255) PRIMARY KEY,
    channel_name VARCHAR(255),
    country_name VARCHAR(255),
);
```

```
CREATE TABLE Video_Category (
    category_id INTEGER PRIMARY KEY,
    name VARCHAR(255)
);
```

```
CREATE TABLE Trending_Video (
    video_id VARCHAR(255) PRIMARY KEY,
    published_date DATE,
    title VARCHAR(1000),
    channel_id VARCHAR(255),
    category_id INTEGER,
    FOREIGN KEY (channel_id) REFERENCES Trending_Channel(channel_id),
    FOREIGN KEY (category_id) REFERENCES Video_Category(category_id)
);
```

```
CREATE TABLE Trending_Video_Ratings (
    video_id VARCHAR(255),
    trending_date DATE,
    ratings_disabled BOOLEAN,
```

```
    views INTEGER,
    likes INTEGER,
    dislikes INTEGER,
    comments INTEGER,
    PRIMARY KEY (video_id, trending_date),
    FOREIGN KEY (video_id) REFERENCES Trending_Video(video_id)
);
```

Inserting Records

After filtering through our data and splitting it into it's according tables, we where going to use MySQL's import wizard to insert the records from the .csv files. We immediately noticed that this process was extremely time consuming and could potentially take days or weeks to insert over a hundred thousand records. Because of this issue, we resolved it by using the LOAD DATA INFILE statement as this inserted the records in seconds.

```
LOAD DATA LOCAL INFILE 'home/folder/file.csv' INTO TABLE People
FIELDS TERMINATED BY ','
IGNORE 1 LINES;
```

This snippet reads a local .csv file where we want to ignore the first row as those are attribute names and not the records themselves, where all attributes are separated by a comma.

PHASE 4

Functional Dependencies

The only functional dependencies, per table, that we found to exist are:

- Trending_Channel
  - channel_id → channel_name
- Trending_Video
  - video_id → published_date, title, channel_id
- Trending_Video_Ratings
  - video_id → likes, views, ratings_disabled, comments, dislikes
- Video_Category
  - category_id → name
- Country
  - country_id → country_name
- People
  - country_id, demographic, my_year → population

Normal Forms

Country table

Column Name	Data Type	Key	Constraints
country_id	VARCHAR	Primary Key	Not Null
country_name	VARCHAR	-	Not Null

- 3rd Normal Form: No violation since all non-key attributes depend solely on the primary key.
- Boyce-Codd Normal Form: No violation since the table only has a single candidate key which is also the primary key.
- 4th Normal Form: No violation since there are no multi-valued dependencies.

People table

Column Name	Data Type	Key	Constraints
country_id	VARCHAR	Foreign Key	Not Null
demographic	VARCHAR	-	Not Null
my_year	INTEGER	-	Not Null
population	BIGINT	-	Not Null

- 3rd Normal Form: No violation since all non-key attributes depend solely on the primary key.
- Boyce-Codd Normal Form: No violation since the table only has a single candidate key which is also the primary key.
- 4th Normal Form: No violation since there are no multi-valued dependencies.

Trending_Channel table

Column Name	Data Type	Key	Constraints
channel_id	VARCHAR	Primary Key	Not Null
channel_name	VARCHAR	-	Not Null
country_name	VARCHAR	-	Not Null

- 3rd Normal Form: No violation since all non-key attributes depend solely on the primary key.
- Boyce-Codd Normal Form: No violation since the table only has a single candidate key which is also the primary key.

- 4th Normal Form: No violation since there are no multi-valued dependencies.

Video_Category table

Column Name	Data Type	Key	Constraints
category_id	INTEGER	Primary Key	Not Null
name	VARCHAR	-	Not Null

- 3rd Normal Form: No violation since all non-key attributes depend solely on the primary key.
- Boyce-Codd Normal Form: No violation since the table only has a single candidate key which is also the primary key.
- 4th Normal Form: No violation since there are no multi-valued dependencies.

Trending_Video table

Column Name	Data Type	Key	Constraints
video_id	VARCHAR	Primary Key	Not Null
published_date	DATE	-	Not Null
title	VARCHAR	-	Not Null
channel_id	VARCHAR	Foreign Key	Not Null
category_id	INTEGER	Foreign Key	Not Null

- 3rd Normal Form: No violation since all non-key attributes depend solely on the primary key.
- Boyce-Codd Normal Form: No violation since the table only has a single candidate key which is also the primary key.
- 4th Normal Form: No violation since there are no multi-valued dependencies.

Trending_Video_Ratings table

Column Name	Data Type	Key	Constraints
video_id	VARCHAR	Foreign Key	Not Null
trending_date	DATE	-	Not Null
ratings_disabled	BOOLEAN	-	Not Null
views	INTEGER	-	Not Null
likes	INTEGER	-	Not Null
dislikes	INTEGER	-	Not Null
comments	INTEGER	-	Not Null

- 3rd Normal Form: No violation since each non-key attribute is dependent only on the primary key and there are no transitive dependencies.
- Boyce-Codd Normal Form: No violation since the table only has a single candidate key which is also the primary key.
- 4th Normal Form: No violation since there are no multi-valued dependencies.

PHASE 5

Test Queries

Overview

You will find some queries ready for execution in our database. If you would like to test these out, here you will find a quick guide to access our database. You must have MySQL Workbench installed on your machine.

After installation, find the plus icon that will allow for a new connection. Then fill each field accordingly.

- **Hostname:** db126.cmocladgarwr.us-west-1.rds.amazonaws.com
- **Port:** 3306
- **Username:** admin
- **Password:** Xr7G9001s

Subquery

```
SELECT name
FROM Video_Category
WHERE category_id IN
  (SELECT category_id
  FROM Trending_Video);
```

- This shows all the categories that all the current YouTube trending videos belong to. This is a subset of all the video categories available, as the resulting table will show when compared to all the records in the Video_Category table.

Aggregation

```
SELECT AVG(views)
FROM Trending_Video_Ratings
WHERE ratings_disabled = 0;
```

- From all the videos that have trended, this query gathers the average amount of views for those videos that do not have their ratings disabled.

Insert Query

```
INSERT INTO Video_Category
VALUES (45, 'Keyboards');
```

- For our category table, we did not expect to add new categories so when inserting a record the key will not auto increment. Although when inserting manually any value over 44 should be fine.

Update Query

```
UPDATE Video_Category
SET name = 'Keyboards and Mouses'
WHERE category_id = 45;
```

- Updates the attribute name for the record previously inserted in the Video_Category table issued the number 45 as its key.

Drop Query

```
DELETE FROM Video_Category
WHERE category_id = 45;
```

- This query is provided for quick removal of the previously inserted record.

PHASE 6

Querying The Database

1. **What trending channel had the most views in the year (2020-2021)?**

```
SELECT Trending_Channel.channel_name, SUM(Trending_Video_Ratings.views) as
total_views
FROM Trending_Channel
JOIN Trending_Video ON Trending_Channel.channel_id = Trending_Video.channel_id
JOIN Trending_Video_Ratings ON Trending_Video.video_id =
Trending_Video_Ratings.video_id
WHERE YEAR(Trending_Video_Ratings.trending_date) BETWEEN 2020 AND 2021
GROUP BY Trending_Channel.channel_name
ORDER BY total_views DESC
LIMIT 1;
```

- This query retrieves the name of the trending channel that had the highest total views for videos posted in the years 2020 and 2021. It does this by joining the `Trending_Channel`, `Trending_Video`, and `Trending_Video_Ratings` tables on their respective IDs, and then filtering the results based on the `trending_date` field being within the years 2020 and 2021. The `SUM()` function is used to calculate the total number of views for each channel, and the results are grouped by channel name. The `ORDER BY` clause sorts the results in descending order of total views, and the `LIMIT` clause restricts the output to only the top result.

Resulting Table

channel_name	total_views
MrBeast	9699720274

2. **Which videos have the highest number of views in each category?**

```
SELECT c.name AS category_name, v.title AS video_title, MAX(tr.views) AS max_views
FROM Trending_Video_Ratings tr
INNER JOIN Trending_Video v ON tr.video_id = v.video_id
INNER JOIN Video_Category c ON v.category_id = c.category_id
GROUP BY c.name
ORDER BY c.name ASC, max_views DESC
```

- This query joins three tables: Trending_Video_Ratings, Trending_Video, and Video_Category. It selects the category name, video title, and maximum number of views for each video in each category. The tables are joined using INNER JOIN, which means that only records that have matching values in all three tables are returned. The

result set is grouped by category name and sorted in ascending order by category name and descending order by the maximum number of views. This means that videos within each category are sorted by their maximum number of views, with the most viewed video at the top. The query uses the AS keyword to give aliases to the selected columns, making it easier to read the results.

Resulting Table

category_name	video_title	max_views
Music	[쇼! 음악중심] 블랙핑크 -Lovesick Girls (BLACKPINK -Lovesick Girls) 20201010	264407389
Entertainment	Binging with Babish: Chef's Choice Platter from Monster Hunter: World	206202284
Sports	Shaq & Chuck Roasting Zion Williamson on Inside the NBA🤔	103564168
Howto & Style	Top Mexican Makeup Artists Do My Makeup	89075984
Comedy	THIS YOUTUBER CHALLENGE UNCLE ROGER (Joshua Weissman)	87284105
Film & Animation	The Discovery 🔍 I Pokémon Evolutions: Episode 8	86415224
People & Blogs	The Merchant...	84063330
Science & Technology	Should you upgrade to the iPhone 12 and iPhone 12 Pro?	77745621
Gaming	A New Journey on a Mysterious Island Begins! - ARK Lost Island [DLC Episode 1]	73728043
Education	The Battle of SHARKS!	55299186
News & Politics	U.S. Senate Impeachment Trial of Former President Trump (Day 3)	46246802
Autos & Vehicles	GINTANI 1500 HP BUILT 720 GTR MOTOR UPDATE... *$100000 ENGINE*	35708883
Travel & Events	TRYING COSTCO Instant Noodles ASIAN FOOD I COSTCO Food Tour!	22912715
Pets & Animals	BITTEN - by a GIANT CATFISH!	9094409
Nonprofits & Activism	I Was Hacked. But Now I'm BACK!	4745668

3. **Is there a correlation between the day of the week a video is published and its success?**

```
SELECT DAYOFWEEK(published_date) AS day_of_week, AVG(views) AS avg_views
FROM Trending_Video
JOIN Trending_Video_Ratings ON Trending_Video.video_id =
Trending_Video_Ratings.video_id
GROUP BY day_of_week
ORDER BY avg_views DESC;
```

- This query first joins the `Trending_Video` and `Trending_Video_Ratings` tables using the video_id column. It then uses the `DAYOFWEEK()` function to extract the day of the week from the `published_date` column, and calculates the average number of views for each day of the week using the `AVG()` function. Finally, the results are sorted in descending order by the average number of views. You can then analyze the results to see if there is a significant difference in average views depending on the day of the week a video is published.

Resulting Table

day_of_week	avg_views
6	3908077.4353
7	3194336.0336
5	2596460.5628
3	2586726.7211
2	2466616.7557
4	2434368.0967
1	2209968.4880

4. **How does the length of a video's title impact its engagement on YouTube?**

```
SELECT
  CASE
    WHEN CHAR_LENGTH(title) <= 20 THEN '0-20'
    WHEN CHAR_LENGTH(title) <= 40 THEN '21-40'
    WHEN CHAR_LENGTH(title) <= 60 THEN '41-60'
    ELSE '61+'
  END AS title_length_range,
  CAST(AVG(views) AS UNSIGNED) AS avg_views,
  CAST(AVG(likes) AS UNSIGNED) AS avg_likes,
  CAST(AVG(dislikes) AS UNSIGNED) AS avg_dislikes,
  CAST(AVG(comments) AS UNSIGNED) AS avg_comments
FROM
  Trending_Video
  JOIN Trending_Video_Ratings ON Trending_Video.video_id =
Trending_Video_Ratings.video_id
GROUP BY
  title_length_range;
```

- This query calculates the average number of views, likes, dislikes, and comments for videos in different title length ranges. The results are grouped into four categories based on the length of the video title (0-20, 21-40, 41-60, and 61+). The query uses a CASE statement to categorize the video titles into different length ranges based on the character length of the title. It then joins the Trending_Video and Trending_Video_Ratings tables to get the necessary data for calculating the average views, likes, dislikes, and comments. Finally, the results are grouped by the title length range and the average values for each metric are calculated and displayed in the output.

Resulting Table

title_length_range	avg_views	avg_likes	avg_dislikes	avg_comments
41-60	2569193	143001	2907	11406
61+	2202239	96487	2264	8141
0-20	2618525	183101	3054	19756
21-40	3616571	209797	3712	19440

PHASE 7

Data Analysis

After analyzing the resulting tables per query, this is what the data tells us about each question accordingly. When mentioning a collective of people, it is referred to the country of the United States of America as our data reflects this.

1. **What trending channel had the most views in the year (2020-2021)?**
    a. The resulting table tells us that the YouTube channel titled “Mr.Beast” holds the number one spot with a total of over nine billion views. As an avid YouTube viewer myself, this is no surprise as his popularity has been well established in the top ranks of YouTube.
2. **Which videos have the highest number of views in each category?**
    a. The categories with the most popular, in terms of views, videos were music, entertainment, and sports. The bottom three categories were travel and events, pets and animals, and nonprofits and activism. This data also tells us that we, the United States, are more likely to consume content in the top three categories rather than in the bottom three categories. Although this is mere speculation, other factors can contribute as to what trends, such as corporations paying money or partnering with YouTube.
3. **Is there a correlation between the day of the week a video is published and its success?**
    a. Success was measured in terms of views. On average, from the best to worst day to publish a video was: Saturday, Sunday, Friday, Wednesday, Tuesday, Thursday, and Monday. At a first glance, this makes sense as the top three days are when most of us have the most free time to spend watching YouTube videos. Although we can not say for certain as this could just mean that most videos are usually published on the top three days, contributing to a higher average of views.
4. **How does the length of a video’s title impact its engagement on YouTube?**
    a. Overall, the result suggests that the length of a video's title does have some impact on its engagement on YouTube, with videos with titles between 21-40 characters generally performing better than videos with shorter or longer titles. However, it's important to note that this analysis only considers one aspect of a video's metadata, and other factors such as the quality of the content, the timing of the upload, and the promotion of the video can also play a role in its success.

Concluding Statement

Project Discrepancies

It should be noted that we lacked a query involving data from the “People” and “Country” tables. Thus making this data redundant to our project. The lack of usage was due to the difficulties that were encountered. Out of all countries, we could only use one of them, being the United States. Because of this, we could not create a significant query involving more than one country, as we only had that singular one. This was due to the arduous effort of reformatting the data through excel, which would require days if not weeks worth of copying and pasting just to create .csv files that could be uploaded to a table in our database.To give a clearer image of this issue, we had to split one .csv file per country into three separate files, while heavily editing the data within it. One example of something that took a long time was reformatting the date as the original date was not up to the standards of MySQL. Excel does not support regex so we had much difficulty for this one task alone.