**Analysis Of Crowdfunding**

**Team:**

Carlos Ruiz

George Stalker

Joy Weishan

**Introduction:**

This project is to demonstrate building an ETL pipeline. This will include extracting data from crowdfunding files, data transformation, creating a database and table schema with keys and indexes. Post transformation and database creation, the team will load the database via importing transformed data. The team will then extract the data via SQL and/or ORM for data analysis and convert the data into data visualizations as well as include analysis of the data.

**Project Scope:**

This project will produce a crowdfunding database schema, SQL, database image, 4 transformed data files in CSV format. Post transformation the data will be loaded to the database via Python using an application layer separated from the SQL layer part of the application. The application will include the use of flask as well. Post data imports, the data will be extracted via SQL and/or ORM for analysis which will include visualizations.

**Color Palette:**

There will be a color pallet for consistency across the graphs.



**Dataset Cleaning Steps:**

Campaign:

Renamed columns for clarity, such as rename of “blurb” to “description” and “cf\_id” to “funding\_id”. Renamed columns for consistency across the database. Converted goal and pledged datatypes to floats for better data analysis. Converted date columns to date time for importing to the database as a timestamp and proper data analysis.

Category/Subcategory:

The Category and Subcategory columns were originally in one column and had to be split into two columns and added the prefix “cat” and “subcat” to each so that the join to the campaign table would work. Also, provided column naming consistency across the database. Also insured uniqueness in both the category and subcategory tables. Merged the campaign with category and subcategory to get the ID for further joins on the database during analysis. Dropped category and subcategory columns after the merge as database will be normalized with category subcategory tables.

Contact:

JSON is effective in creating a data frame with the input provided. This project also included the use of regular expressions to find the proper data within a column to extract and assign to the proper columns. Therefore, this project demonstrated JSON and regular expressions in the code. The JSON data was used to load to the database used during analysis. Either file could have been used as the end result is the same formatted data file. The name column was split into first name and last name so that proper data analysis and data retrieval can be done as needed. The full name column was dropped as not needed. The order of the columns was modified to match the database table schema prior to import.

**Crowdfunding dataset Visualizations:**

How many Country projects reached their goal and if they didn’t, how close were they to reaching their goal. Analysis can be done by Country with this application. A sub select was used to calculate the pledge success and then summed in the select by country so that analysis can be done by country.

Example: Canada with Country=”CA”

22 out of 44 reached their goal

3 reached 90% of goal

2 reached 80% of goal

2 reached 70% of goal

3 reached 60% of goal

3 reached 50% of goal

This means in Canada 35 out of 44 (79.5%) made over 50% of goal.

Example: Canada with Country=”US”

436 out of 763 reached their goal

26 reached 90% of goal

33 reached 80% of goal

27 reached 70% of goal

41 reached 60% of goal

34 reached 50% of goal

In the United States 597 out of 763 (78.2%) made over 50% of goal.

AU is 86%

DK is 77.4%

CH is 69.65

IT is 75%

GB is 75%

Australia has the highest percentage of achieving 50% of goal

SELECT

SUM (perfect + ninety +eighty + seventy + sixty + fifty)as above\_fifty,

COUNT(country) AS grand\_total,

SUM(perfect) AS goalreached,

SUM(ninety) AS ninetypercent\_goalreached,

SUM(eighty) AS eightypercent\_goalreached,

SUM(seventy) AS seventypercent\_goalreached,

SUM(sixty) AS sixtypercent\_goalreached,

SUM(fifty) AS fiftypercent\_goalreached,

SUM(fail) AS lessthanfiftypercent\_goalreached

FROM

(

SELECT

country,

CASE WHEN pledged >= goal then 1 else 0 END AS perfect,

CASE WHEN pledged >= goal \* 0.9 AND pledged < goal then 1 else 0 END AS ninety,

CASE WHEN pledged >= goal \* 0.8 AND pledged < goal \* 0.9 then 1 else 0 END AS eighty,

CASE WHEN pledged >= goal \* 0.7 AND pledged < goal \* 0.8 then 1 else 0 END AS seventy,

CASE WHEN pledged >= goal \* 0.6 AND pledged < goal \* 0.7 then 1 else 0 END AS sixty,

CASE WHEN pledged >= goal \* 0.5 AND pledged < goal \* 0.6 then 1 else 0 END AS fifty,

CASE WHEN pledged < goal \* 0.5 then 1 else 0 END AS fail

FROM

campaign

)

GROUP BY

country

HAVING

country = 'GB';

<INSERT 2nd ANALYSIS HERE>

<INSERT 2nd ANALYSIS HERE>

**Limitations/Biases:**

**Future work:**

Add more ways to slice and dice the data for pledge to goal for each country. Use percentage as input from the application so that various SQL could be run to determine things like what if I live in the US, I want to know what are the chances I’ll reach 75% of my goal or 80% of my goal.

**:**

**Sources:**

**"Coolors."** *Coolors.co,* n.d.,

https://coolors.co/visualizer/bdc667-4bd61d-626d58-544343-4f282c-653c40-734e51-737

171-8c6d6f-7885e7

**"Microsoft Template Designs"** *Microsoft*

https://create.microsoft.com/en-us/template/blob-design-62339b2c-5cf2-47ff-90b5-2207f

1f6a677