**ETL Project: Carrie, Godday, Victoria**

**Requirements:**

1. *Extract: your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc).*
   * Original data sources were CSV files:
     + [World Happiness from kaggle.com](https://www.kaggle.com/unsdsn/world-happiness)
     + [Starbucks data from data.world](https://data.world/alice-c/starbucks/workspace/file?filename=Starbucks+World+Stats.csv)
2. *Transform: what data cleaning or transformation was required.*
   * Using pgAdmin:
     + Created two tables: Happiness & Starbucks
     + Used an INNER join to combine the sources by Country
       - Columns included in final database: Country from each; Number of Starbucks; Happiness Rank and Happiness Score
       - A total of 10 columns from original datasets were not included
3. *Load: the final database, tables/collections, and why this was chosen.*
   * Database Name: Starbucks\_Happiness\_DB
   * Final Table Name: StarbucksHappiness\_Final
   * We chose to house our data in a Postgres (relational) database because we were all comfortable working with the toolset provided by PGAdmin. Since we had a small dataset, we did not need a more robust tool for handling a large amount of data. We chose Postgres because allowed us to easily manipulate our datasets and create a database that was readily accessible to a variety of users.

**Schemas:**

|  |  |
| --- | --- |
| CREATE TABLE starbucks (  country VARCHAR,  population INT,  Numer\_of\_Starbucks INT,  Starbucks\_per\_million INT  );  SELECT \* FROM starbucks;  CREATE TABLE happiness (  country VARCHAR,  happiness\_rank INT,  happiness\_score NUMERIC (5,2),  whisker\_high NUMERIC (5,2),  whisker\_low NUMERIC (5,2),  economy\_GDP NUMERIC (5,2),  fam NUMERIC (5,2),  health\_life\_expectancy NUMERIC (5,2),  freedom NUMERIC (5,2),  generosity NUMERIC (5,2),  trust\_gov NUMERIC (5,2),  dystopia NUMERIC (5,2)  );  SELECT \* FROM happiness;  CREATE TABLE StarbucksHappiness\_Final (  country VARCHAR,  population INT,  Numer\_of\_Starbucks INT,  happiness\_rank INT,  happiness\_score NUMERIC (5,2)  ); | select \* from StarbucksHappiness\_Final;  select \* from happiness;  select \* from starbucks;  delete from StarbucksHappiness\_Final;  Insert into StarbucksHappiness\_Final  SELECT happiness.country,  starbucks.population,  starbucks.Numer\_of\_Starbucks,  happiness.happiness\_rank,  happiness.happiness\_score    FROM happiness  INNER JOIN starbucks  ON happiness.country = starbucks.country;    select \* from StarbucksHappiness\_Final; |