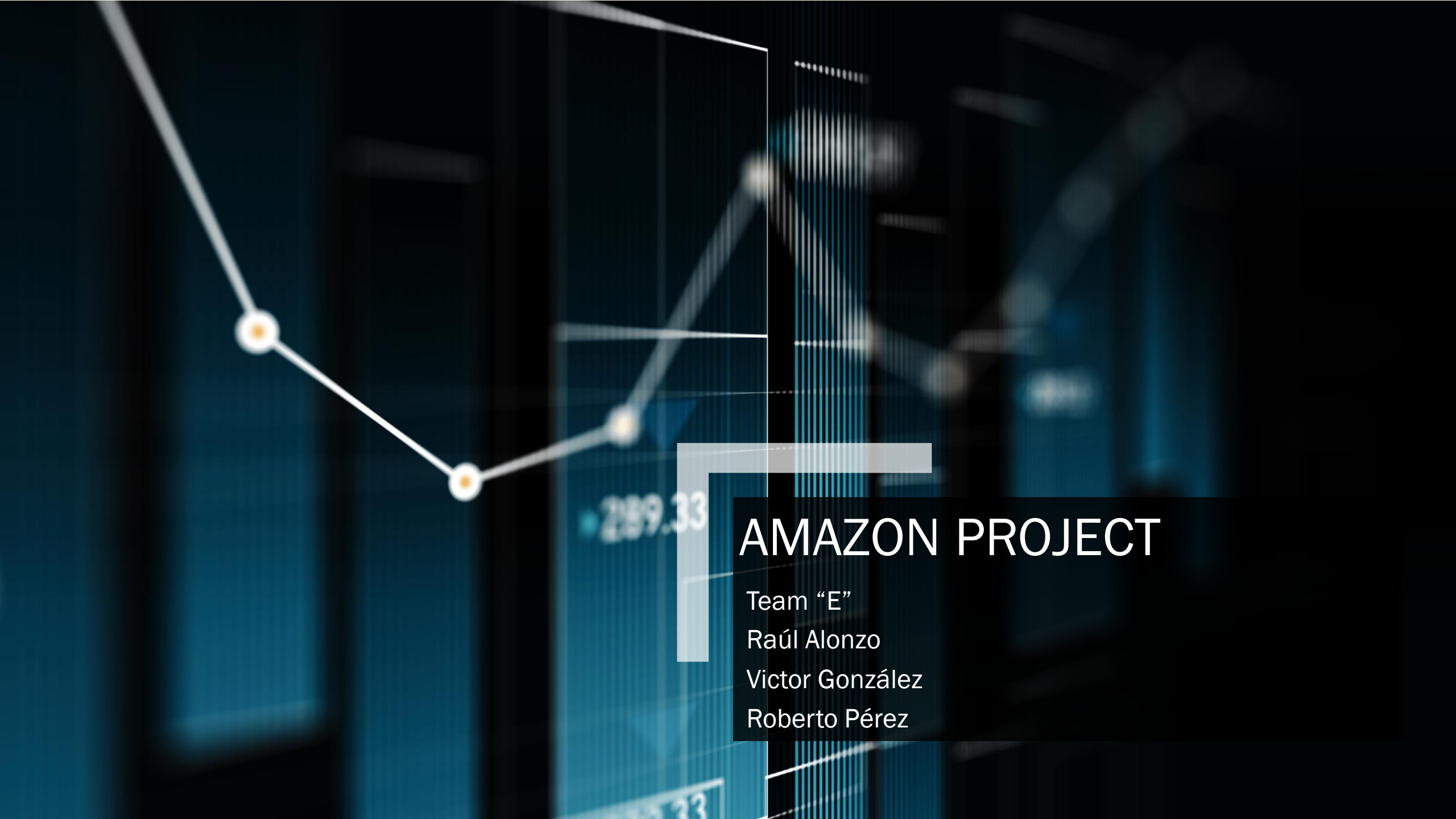


# AMAZON PROJECT

Team “E”
Raúl Alonzo
Victor González
Roberto Pérez

## Background

This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

Data includes:

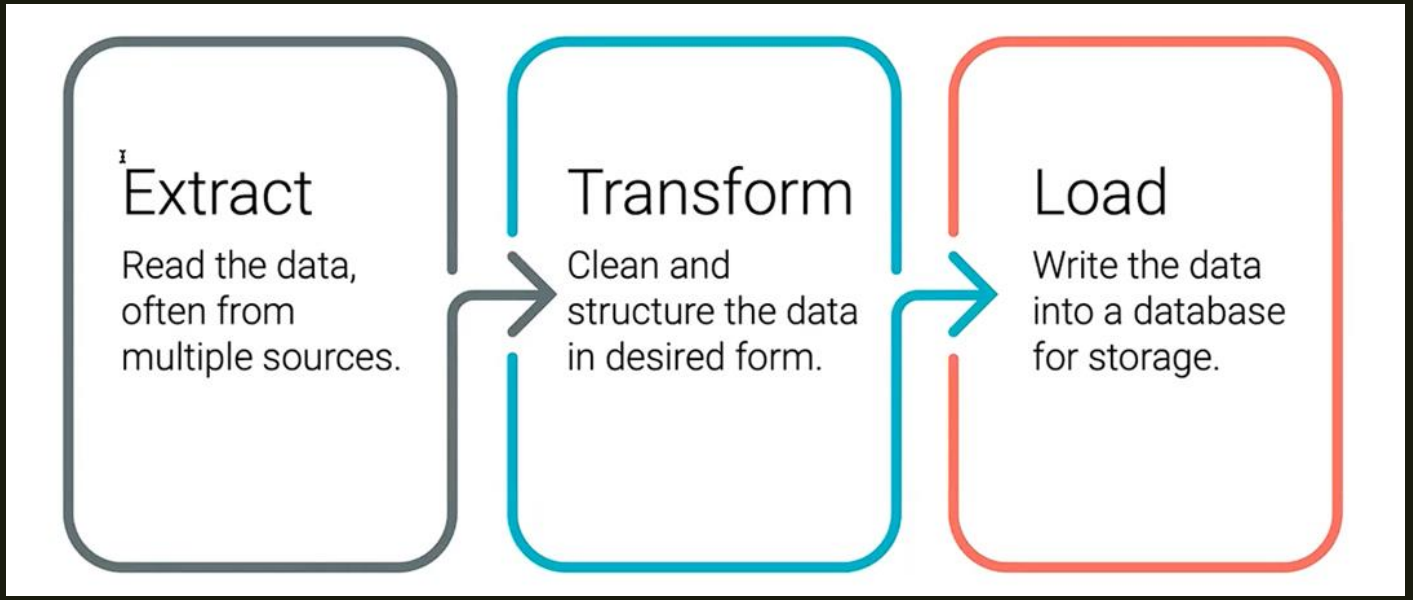
Reviews from Oct 1999 - Oct 2012
568,454 reviews
256,059 users
74,258 products
260 users with > 50 reviews
wordcloud

## Project

Our Project consist on reviewing Amazon´s satisfaction method. Actually Amazon does a posotive/negative qualification and we believe ther must be a more wide range n order to take actions and get a Better feeling about customer service.

.

# PROCESS


Extract
Read the data, often from multiple sources.
Transform
Clean and structure the data in desired form.
Load
Write the data into a database for storage.

# Workflow

**Data Sets Exploration**

- Select Database From Kaggle (**Amazon Fine Food Reviews**)

**Business Question to answer**

- Is it all-right Amazon´s classification model?

**Tools**

- For this Project we are going to work with text. Normalize will be the main challenge so we will use: Python, Visual Studio, PG Admin, Postgres, SQLAlchemy, Pandas, Beautiful Soup & QuickDatabase Diagrams

**Extract**

Extract info from a .csv file

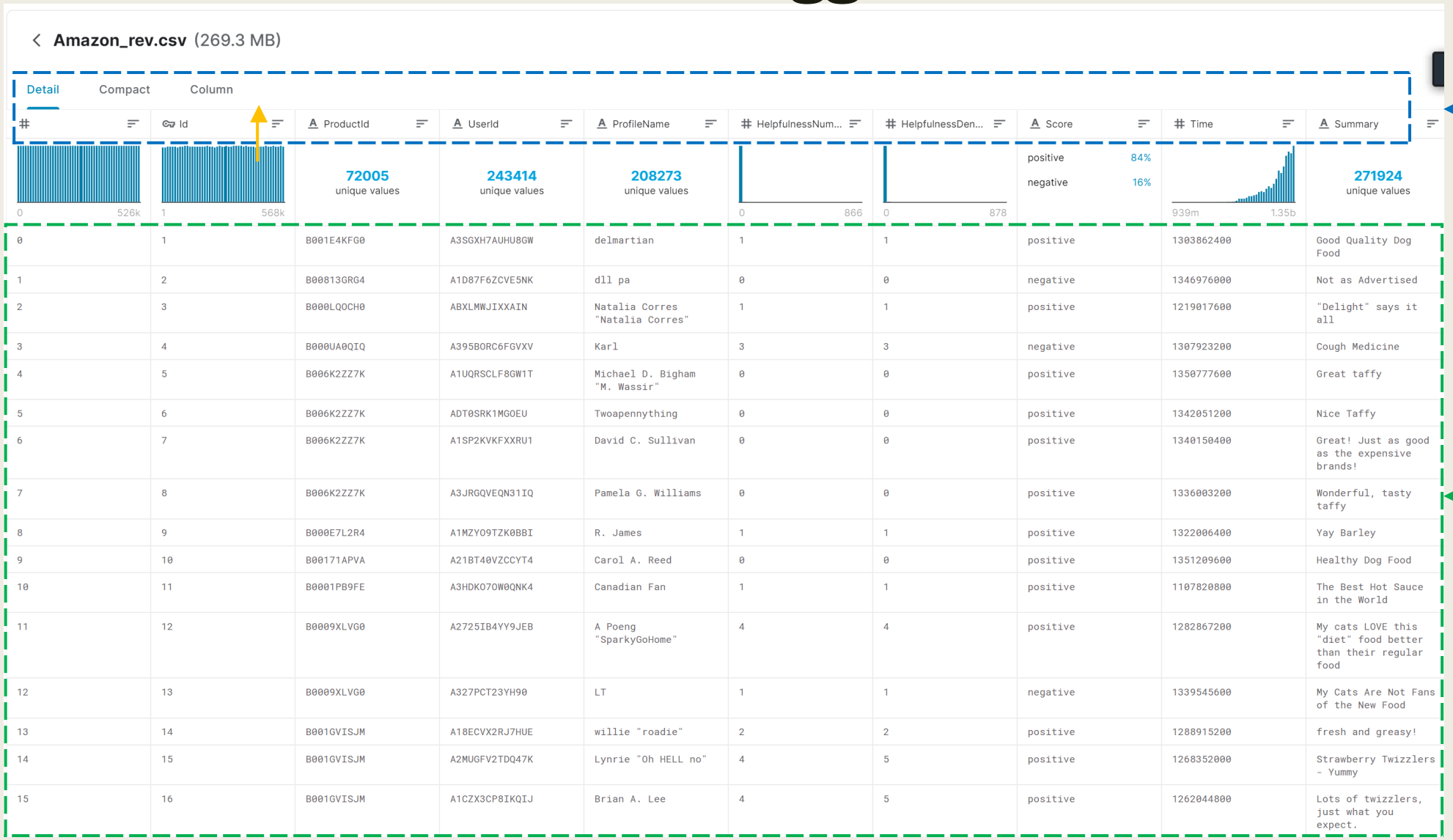
**Transform**

- Normalize: ortography, special characters, convert into lower case

**Load**

- Create several data frames and load into sql in a new Data Base

# Extract data From a Kaggle CSV file




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | B001E4KFG0 | A3SGXH7AUHU8GW | delmartian | 1 | 1 | positive | 1303862400 | Good Quality Dog Food |
| 1 | 2 | B00813GRG4 | A1D87F6ZCVE5NK | dll pa | 0 | 0 | negative | 1346976000 | Not as Advertised |
| 2 | 3 | B000LQOCH0 | ABXLMWJIXXAIN | Natalia Corres "Natalia Corres" | 1 | 1 | positive | 1219017600 | "Delight" says it all |
| 3 | 4 | B000UA0QIQ | A395BORC6FGVXV | Karl | 3 | 3 | negative | 1307923200 | Cough Medicine |
| 4 | 5 | B006K2ZZ7K | A1UQRSCLF8GW1T | Michael D. Bigham "M. Wassir" | 0 | 0 | positive | 1350777600 | Great taffy |
| 5 | 6 | B006K2ZZ7K | ADT0SRK1MGOEU | Twoapennything | 0 | 0 | positive | 1342051200 | Nice Taffy |
| 6 | 7 | B006K2ZZ7K | A1SP2KVKFXXRU1 | David C. Sullivan | 0 | 0 | positive | 1340150400 | Great! Just as good as the expensive brands! |
| 7 | 8 | B006K2ZZ7K | A3JRGQVEQN31IQ | Pamela G. Williams | 0 | 0 | positive | 1336003200 | Wonderful, tasty taffy |
| 8 | 9 | B000E7L2R4 | A1MZYO9TZK0BBI | R. James | 1 | 1 | positive | 1322006400 | Yay Barley |
| 9 | 10 | B00171APVA | A21BT40VZCCYT4 | Carol A. Reed | 0 | 0 | positive | 1351209600 | Healthy Dog Food |
| 10 | 11 | B0001PB9FE | A3HDKO7OW0QNK4 | Canadian Fan | 1 | 1 | positive | 1107820800 | The Best Hot Sauce in the World |
| 11 | 12 | B0009XLVG0 | A2725IB4YY9JEB | A Poeng "SparkyGoHome" | 4 | 4 | positive | 1282867200 | My cats LOVE this "diet" food better than their regular food |
| 12 | 13 | B0009XLVG0 | A327PCT23YH90 | LT | 1 | 1 | negative | 1339545600 | My Cats Are Not Fans of the New Food |
| 13 | 14 | B001GVISJM | A18ECVX2RJ7HUE | willie "roadie" | 2 | 2 | positive | 1288915200 | fresh and greasy! |
| 14 | 15 | B001GVISJM | A2MUGFV2TDQ47K | Lynrie "Oh HELL no" | 4 | 5 | positive | 1268352000 | Strawberry Twizzlers - Yummy |
| 15 | 16 | B001GVISJM | A1CZX3CP8IKQIJ | Brian A. Lee | 4 | 5 | positive | 1262044800 | Lots of twizzlers, just what you expect. |

Data set column titles

Data set detail information

# Transform Data

In [1]:
```
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup as bs
import re
```

In [2]:
```
# List of unpleasant words to check
unpleasant = ['lousy','disappointed','discouraged','ashamed','powerless','diminished','guilty','dissatisfied',
        'miserable','detestable','repugnant','despicable','disgusting','abominable','terrible','in despair',
        'sulky','bad','upset','doubtful','uncertain','indecisive','perplexed','embarassed','hesitant',
        'shy','stupefied','disillusioned','unbelieving','skeptical','distrustful','misgiving','lost',
        'unsure','uneasy','pessimistic','tense''incapable','alone','paralyzed','fatigued','useless',
        'inferior','vulnerable','empty','forced','hesitant','despair','frustrated','distressed','woeful',
        'pathetic','tragic','in a stew','dominated','irritated','enraged','hostile','insulting',
        'annoyed','upset','hateful','offensive','bitter','aggresive','resentful','inflamed','provoked',
        'incensed','infuriated','cross','worked up','boiling','fuming','fearful','terrified','suspicious',
        'anxious','alarmed','panic','nervous','scared','worried','frightened','timid','shaky','restless',
        'doubtful','threatened','cowardly','quaking','wary','crushed','tormented','deprived','pained',
        'tortured','dejected','rejected','injured','offended','afflicted','aching','victimized',
        'heartbroken','agonized','appalled','humiliated','wronged','alienated','tearful','sorrowful',
        'pained','grief','anguish','desolate','desperate','pessimistic','unhappy','lonely','grieved',
        'mournful','dismayed','insensitive','dull','nonchalant','neutral','reserved','weary','bored',
        'preoccupied','cold','disinterested','lifeless', 'never'];
```

In [3]:
```
# List of pleasant words to check
pleasant = ['understanding','confident','reliable','easy','amazed','free','sympathetic','interested','satisfied',
            'receptive','accepting','kind', 'great','joyous','lucky','fortunate','delighted','overjoyed','gleeful','thankful
            'festive','ecstatic','glad','cheerful','elated','jubilant', 'playful','courageous','energetic','liberated','opti
            'impulsive','free','animated','spirited','thrilled','wonderful','calm','peaceful','at ease','comfortable','pleas
            'clever','surprised','content','quiet','certain','relaxed','serene','reassured', 'loving','considerate','affecti
            'tender','devoted','attracted','passionate','admiration','warm','touched','close','comforted','loved','concerned
            'intrigued','absorbed','inquisitive','engrossed','curious','drawn toward','eager','keen','earnest','intent','ins
            'enthusiastic','bold','brave','daring',
            'optimistic','impulsive','free','sure','certain','rebellious','unique','dynamic','tenacious','hardy','secure',
            'confident','challenged', 'love'];
```

In [6]:
```
# Read csv
path = 'Resources/Amazon_rev.csv'
```

In [7]:
```
amazon_foods_df = pd.read_csv(path)
```

In [8]:
```
# Drop first column because it creates junk info
amazon_foods_df = amazon_foods_df.drop(columns=["Unnamed: 0"])
```

In [9]:
```
# Copy of dataframe
amazon_foods_modified_df = amazon_foods_df
```

In [10]:
```
# Add a regular expression to look for all the symbols
regex = "[.\"\',!@#$%^&*()\_\-+=?:;|/!]"
```

In [11]:
```
# Tex replacement for all rules in the regex
text_replace = ''
```

In [12]:
```
#Create first column with text in lower case and no special characters
amazon_foods_modified_df['Lowercase Text'] = ''
```

In [15]:
```
# Split data frame to run demo version
amazon_foods_modified_df = amazon_foods_df.iloc[:100]
```

In [16]:
```
%%time
# Populate new column column
amazon_foods_modified_df['Lowercase Text'] = amazon_foods_df.apply(
    lambda row: (
            re.sub(
                regex,
                text_replace,
                row[9].lower()
            )
    ),
    axis=1)
```

```
Wall time: 17.5 s
```

```
C:\Users\victo\anaconda3\lib\site-packages\ipykernel_launcher.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # Remove the CWD from sys.path while we load stuff.
```

In [17]:
```
#Create second column and clear br html tags
amazon_foods_modified_df['Lowercase Text Clean'] = ''
```

# Transform Data

In [18]:
```
# Define function to extract with beautifull soup br and just return text property
def extract_br_tags(soup):
    for e in soup.findAll('br'):
        e.extract()
    if soup.find('p'):
        return soup.find('p').text
    elif soup.find('span'):
        return soup.find('span').text
    elif soup.find('a'):
        return soup.find('a').text
    return soup.find('body').text
```

In [19]:
```
# Define function to count positive and negative words based on pleasant and unpleasant lists
def add_positive_negative(text, is_positive):
    text_list = text.split()
    count = 0
    for word in text_list:
        if is_positive:
            for pleasant_word in pleasant:
                if word == pleasant_word:
                    count+=1
        else:
            for unpleasant_word in unpleasant:
                if word == unpleasant_word:
                    count+=1
    return count


```

In [20]:
```
%%time
amazon_foods_modified_df['Lowercase Text Clean'] = amazon_foods_modified_df.apply(
    lambda row: (
        extract_br_tags(bs(row[10],'lxml'))
    ),
    axis=1)
```

```
Wall time: 94.9 ms
```

In [21]:
```
# Create column for positive count
amazon_foods_modified_df['Positive Count'] = ''
```

```
C:\Users\victo\anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [22]:
```
%%time
amazon_foods_modified_df['Positive Count'] = amazon_foods_modified_df.apply(
    lambda row: (
        add_positive_negative(row[11], True)
    ),
    axis=1)
```

```
Wall time: 36.6 ms
```

```
C:\Users\victo\anaconda3\lib\site-packages\ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
```

In [23]:
```
# Create column for negative count
amazon_foods_modified_df['Negative Count'] = ''
```

```
C:\Users\victo\anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [24]:
```
%%time
amazon_foods_modified_df['Negative Count'] = amazon_foods_modified_df.apply(
    lambda row: (
        add_positive_negative(row[11], False)
    ),
    axis=1)
```

```
Wall time: 46 ms
```

# Transform Data

```
In [25]: 1 amazon_foods_modified_df
```

Out[25]:

| UserId | ProfileName | HelpfulnessNumerator | HelpfulnessDenominator | Score | Time | Summary | Text | Lowercase Text | Lowercase Text Clean | Positive Count | Negative Count |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AUHU8GW | delmartian | 1 | 1 | positive | 1303862400 | Good Quality Dog Food | I have bought several of the Vitality canned d... | i have bought several of the vitality canned d... | i have bought several of the vitality canned d... | 0 | 0 |
| 5ZCVE5NK | dll pa | 0 | 0 | negative | 1346976000 | Not as Advertised | Product arrived labeled as Jumbo Salted Peanut... | product arrived labeled as jumbo salted peanut... | product arrived labeled as jumbo salted peanut... | 1 | 0 |
| MWJIXXAIN | Natalia Corres "Natalia Corres" | 1 | 1 | positive | 1219017600 | "Delight" says it all | This is a confection that has been around a fe... | this is a confection that has been around a fe... | this is a confection that has been around a fe... | 0 | 0 |
| RC6FGVXV | Karl | 3 | 3 | negative | 1307923200 | Cough Medicine | If you are looking for the secret ingredient i... | if you are looking for the secret ingredient i... | if you are looking for the secret ingredient i... | 0 | 0 |
| CLF8GW1T | Michael D. Bigham "M. Wassir" | 0 | 0 | positive | 1350777600 | Great taffy | Great taffy at a great price. There was a wid... | great taffy at a great price there was a wide... | great taffy at a great price there was a wide... | 2 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| V6MXO9B0 | pionex1796 | 0 | 0 | positive | 1326412800 | Loved these Tartlets | What a nice alternative to an apple pie. Love ... | what a nice alternative to an apple pie love t... | what a nice alternative to an apple pie love t... | 3 | 0 |
| 521O626G | Rachel Westendorf | 0 | 0 | positive | 1308700800 | The best | I like Creme Brulee. I loved that these were s... | i like creme brulee i loved that these were so... | i like creme brulee i loved that these were so... | 3 | 0 |
| UIR6F8KB | Adam E. Smith | 2 | 2 | positive | 1295308800 | Wasting Vinegar on a Cucumber is a Shamel | I first bought pickled asparagus at an Amish m... | i first bought pickled asparagus at an amish m... | i first bought pickled asparagus at an amish m... | 1 | 0 |

```
In [26]: 1 # Drop columns not needed for modified csv
         2 amazon_foods_modified_df = amazon_foods_modified_df.drop(columns=['Lowercase Text'])

In [27]: 1 amazon_foods_modified_df.to_csv('Amazon_rev_modified.csv')

In [ ]: 1
```

# Load Data

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

### CREATE DATABASE CONNECTION

In [129]:
```
connection_string = "postgres:B00tc@mp@localhost:5432/Amazon"
```

In [130]:
```
# confirm tables
engine = create_engine(f"postgresql+psycopg2://{connection_string}")
```

### LOAD DATAFRAME INTO DATABASE

In [131]:
```
data_df.to_sql(name="data", con=engine, if_exists="append", index=False)
```

### CREATE & LOAD 'USER' DATAFRAME INTO DATABASE

In [132]:
```
# Create DF for table 'user'
user_df_table = data_df[['user_id','profile_name']]
```

In [133]:
```
user_df_table
```

In [134]:
```
# Identify Duplicated
user_df_table = user_df_table.assign(repeat=lambda r:r['user_id'].duplicated())
```

In [135]:
```
# Query
user_df_table = user_df_table.query('repeat == False')
```

In [136]:
```
# Drop repeated
user_df_table = user_df_table.drop(columns=['repeat'])
```

In [137]:
```
# Print
user_df_table
```

In [138]:
```
# Send data to SQL
user_df_table.to_sql(name='user', con=engine, if_exists="append", index=False)
```

### CREATE & LOAD 'PRODUCT' DATAFRAME INTO DATABASE

In [139]:
```
# Create DF for table 'product'
product_df_table = data_df[['product_id']]
```

In [140]:
```
# Print
product_df_table
```

In [141]:
```
# Identify Duplicated
product_df_table = product_df_table.assign(repeat=lambda r:r['product_id'].duplicated())
```

In [142]:
```
# Query
product_df_table = product_df_table.query('repeat == False')
```

In [143]:
```
# Drop repeated
product_df_table = product_df_table.drop(columns=['repeat'])
```

In [144]:
```
product_df_table
```

In [145]:
```
# Send data to SQL
product_df_table.to_sql(name='product', con=engine, if_exists="append", index=False)
```

### CREATE & LOAD 'REVIEW' DATAFRAME INTO DATABASE

In [146]:
```
# Create DF for table 'review'
review_df_table = data_df[['review_id','score', 'summary', 'text']]
```

In [147]:
```
# Print
review_df_table
```

In [148]:
```
# Identify Duplicated
review_df_table = review_df_table.assign(repeat=lambda r:r['review_id'].duplicated())
```

In [149]:
```
# Query
review_df_table = review_df_table.query('repeat == False')
```

In [150]:
```
# Drop repeated
review_df_table = review_df_table.drop(columns=['repeat'])
```

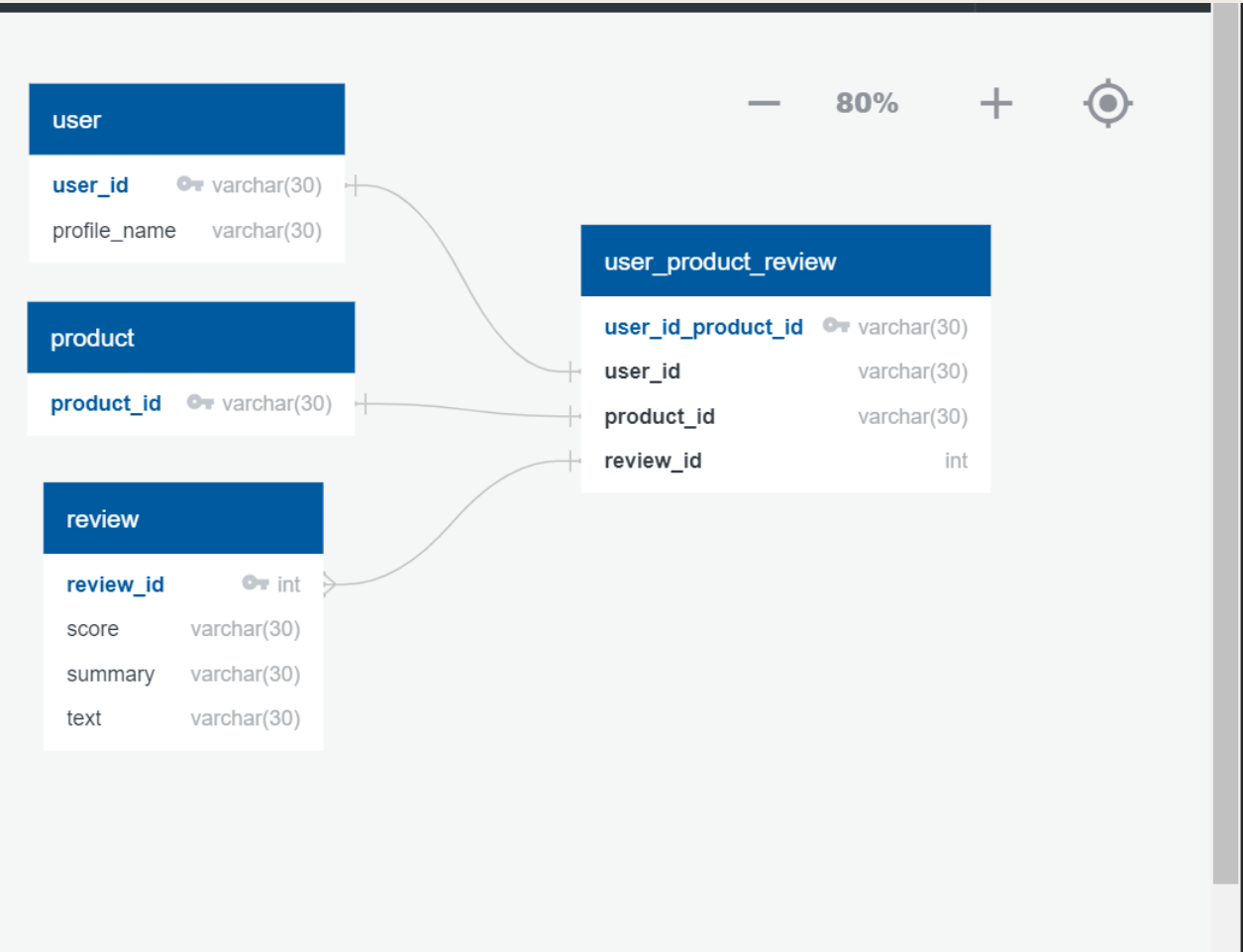
In [151]:
```
# Print
review_df_table
```

In [152]:
```
# Send data to SQL
review_df_table.to_sql(name='review', con=engine, if_exists="append", index=False)
```

# Relational Diagram

```
user
-
user_id varchar(30) pk
profile_name varchar(30)


product
-
product_id varchar(30) pk FK


review
-
review_id int pk
score varchar(30)
summary varchar(30)
text varchar(30)

user_product_review
-
user_id_product_id varchar(30) pk
user_id varchar(30) fk - user.user_id
product_id varchar(30) fk - product.product_id
review_id int fk -< review.review_id

```

No errors found. Good job!




7

8

9

# Amazon new Data Base