**ETL Project Report**

**Shubhangi Vidhate**

**Lixin Tian**

**Reena Patel**

-----------------------------------------------------------------------------------------------------------------------------------

**Project Scope:**

The U.S. has almost 500 students for every guidance counselor. Underserved youth lack the network to find their career role models, making CareerVillage.org the only option for millions of young people in America and around the globe with nowhere else to turn. To date, 25,000 volunteers have created profiles and opted in to receive emails when a career question is a good fit for them. To help students get the advice they need, the team at CareerVillage.org needs to be able to send the right questions to the right volunteers. CareerVillage.org has provided several years of anonymized data and each file comes from a table in their database. Currently data is being saved in several different CSV files so using files from various locations we cannot answer complex business questions that can be answered by ETL.

Goal of the project is to define ETL process to provide clean data for analyze questions like How well does the solution match professionals to the questions they would be motivated to answer? Final dataset will help us to develop a method to recommend relevant questions to the professionals who are most likely to answer them.

**Downloaded CSV files from data source: CareerVillage.org has provided several years of anonymized data and each file comes from a table in their database.**

**Data source Link:** <https://www.kaggle.com/c/data-science-for-good-careervillage/data>

* answer\_scores.csv
* answers.csv
* comments.csv
* emails.csv
* group\_memberships.csv
* groups.csv
* matches.csv
* professionals.csv
* question\_scores.csv
* questions.csv
* school\_memberships.csv
* students.csv
* tag\_questions.csv
* tag\_users.csv
* tags.csv

**Source API(Optional):** **kaggle competitions download -c data-science-for-good-careervillage**

Examples:

kaggle datasets download data-science-for-good-careervillage/data

kaggle datasets download data-science-for-good-careervillage/data -f answers.csv

answers.csv: Answers are what this is all about! Answers get posted in response to questions. Answers can only be posted by users who are registered as Professionals. However, if someone has changed their registration type after joining, they may show up as the author of an Answer even if they are no longer a Professional.

-----------------------------------------------------------------------------------------------------------------------------------

comments.csv: Comments can be made on Answers or Questions. We refer to whichever the comment is posted to as the "parent" of that comment. Comments can be posted by any type of user. Our favorite comments tend to have "Thank you" in them :)

-----------------------------------------------------------------------------------------------------------------------------------

emails.csv: Each email corresponds to one specific email to one specific recipient. The frequency level refers to the type of email template which includes immediate emails sent right after a question is asked, daily digests, and weekly digests.

-----------------------------------------------------------------------------------------------------------------------------------

group\_memberships.csv: Any type of user can join any group. There are only a handful of groups so far.

-----------------------------------------------------------------------------------------------------------------------------------

groups.csv: Each group has a "type". For privacy reasons we have to leave the group names off.

-----------------------------------------------------------------------------------------------------------------------------------

matches.csv: Each row tells you which questions were included in emails. If an email contains only one question, that email's ID will show up here only once. If an email contains 10 questions, that email's ID would show up here 10 times.

-----------------------------------------------------------------------------------------------------------------------------------

professionals.csv: We call our volunteers "Professionals", but we might as well call them Superheroes. They're the grownups who volunteer their time to answer questions on the site.

-----------------------------------------------------------------------------------------------------------------------------------

questions.csv: Questions get posted by students. Sometimes they're very advanced. Sometimes they're just getting started. It's all fair game, as long as it's relevant to the student's future professional success.

-----------------------------------------------------------------------------------------------------------------------------------

school\_memberships.csv: Just like group memberships, but for schools instead.

-----------------------------------------------------------------------------------------------------------------------------------

students.csv: Students are the most important people on CareerVillage.org. They tend to range in age from about 14 to 24. They're all over the world, and they're the reason we exist!

-----------------------------------------------------------------------------------------------------------------------------------

tag\_questions.csv: Every question can be hash tagged. We track the hashtag-to-question pairings, and put them into this file.

-----------------------------------------------------------------------------------------------------------------------------------

tag\_users.csv: Users of any type can follow a hashtag. This shows you which hashtags each user follows.

-----------------------------------------------------------------------------------------------------------------------------------

tags.csv: Each tag gets a name.

-----------------------------------------------------------------------------------------------------------------------------------

question\_scores.csv: "Hearts" scores for each question.

-----------------------------------------------------------------------------------------------------------------------------------

answer\_scores.csv: "Hearts" scores for each answer.

**ETL Process step by step process:**

**Extract:**

* 1. Read CSV file from the source link
  2. Created notebook code to read CSV file

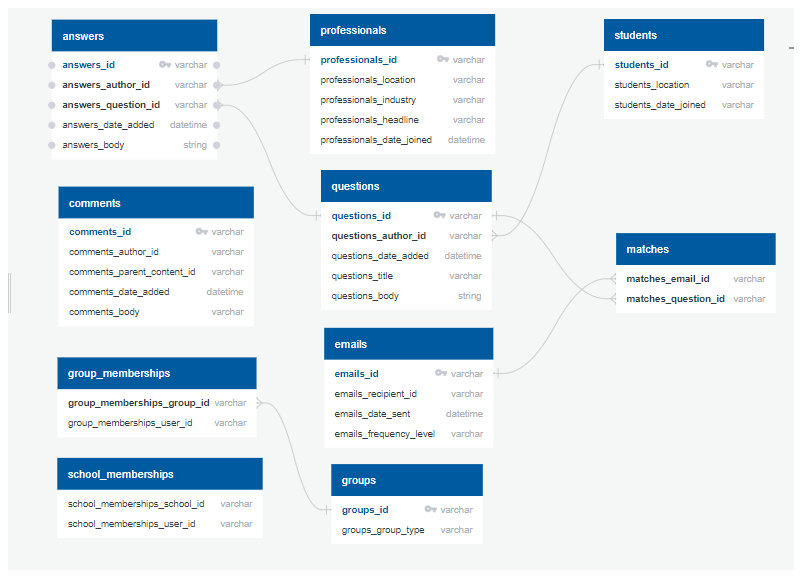
**Transform:**

* 1. Create ERD diagram to build primary key and foreign key relationship.
  2. Create script to build postgres database.
  3. Write python code to read csv from local folder and add index and change date format and save file to postgres table.

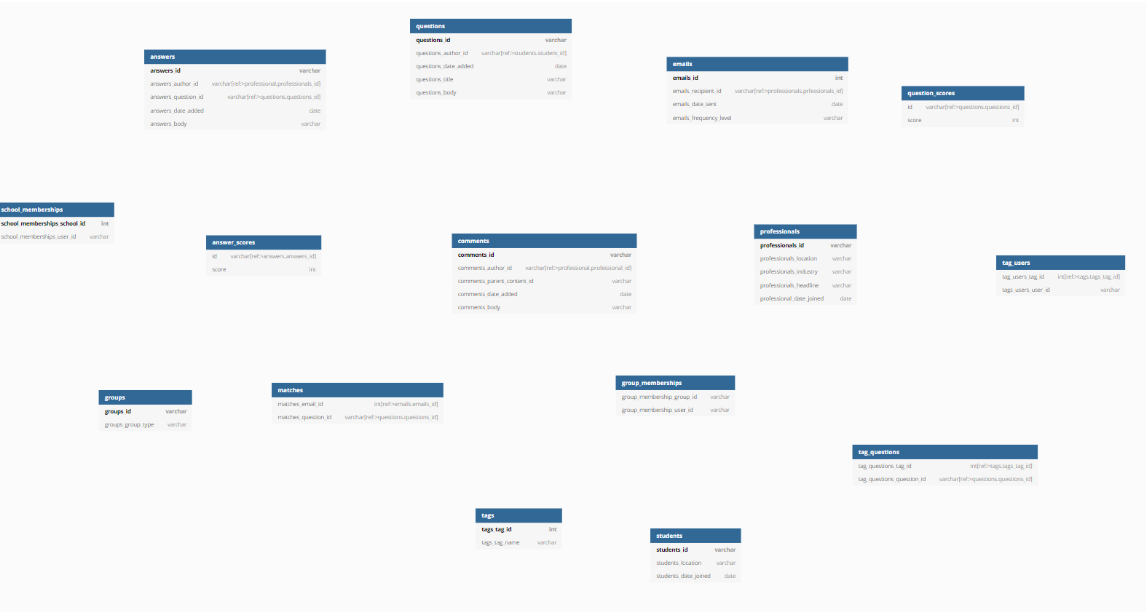
**Load:**

* 1. Import CSV files into postgres database tables.
  2. Write queries to perform analysis and aggregate data

**ERD Diagram 1:**



**ERD Diagram 2:**



**Database Script:** create database careervillage;

-- postgres sql to create tables

-- 1.

-- answer\_scores

-- created on 07/24/19

-- drop table if it exists

drop table answer\_scores;

-- Create a new table answer\_scores

CREATE TABLE answer\_scores (

id VARCHAR(60) PRIMARY KEY,

score INT

);

-- 2.

-- professionals

-- created on 07/15/19

-- drop table if it exists

drop table professionals;

-- Create a new table professionals

CREATE TABLE professionals (

professionals\_id VARCHAR(60) PRIMARY KEY,

professionals\_location VARCHAR(200),

professionals\_industry VARCHAR(200),

professionals\_headline TEXT,

professionals\_date\_joined Date

);

-- 3.

-- students

-- created on 07/25/19

--drop table if it exists

drop table students;

-- Create a new table students

CREATE TABLE students (

students\_id VARCHAR(60) PRIMARY KEY,

students\_location VARCHAR(200),

students\_date\_joined Date

);

-- 4.

-- emails

-- drop table if it exists

drop table emails;

-- Create a new table emails

CREATE TABLE emails (

emails\_id INT PRIMARY KEY,

emails\_recipient\_id VARCHAR(60),

emails\_date\_sent Date,

emails\_frequency\_level VARCHAR(60)

);

-- 5.

-- tags

-- created on 07/25/19

-- drop table if it exists

drop table tags;

-- Create a new table answer\_scores

CREATE TABLE tags (

tags\_tag\_id INT PRIMARY KEY,

tags\_tag\_name TEXT

);

-- 6.

-- groups

-- created on 07/25/19

-- drop table if it exists

drop table groups;

-- Create a new table groups

CREATE TABLE groups (

groups\_id VARCHAR(60) PRIMARY KEY,

groups\_group\_type VARCHAR(60)

);

--7.

--school\_memberships

--created on 07/25/19

-- drop table if it exists

drop table school\_memberships;

-- Create a new table school\_memberships

CREATE TABLE school\_memberships (

school\_memberships\_school\_id INT PRIMARY KEY,

school\_memberships\_user\_id VARCHAR(60)

);

-- 8.

-- matches

-- created on 07/25/19

--drop table if it exists

drop table matches;

-- Create a new table matches

CREATE TABLE matches (

matches\_email\_id INT,

matches\_question\_id VARCHAR(60),

PRIMARY KEY (matches\_email\_id, matches\_question\_id)

);

--sql to create tables

-- 9.

-- answers

-- created on 07/24/19

-- drop table if it exists

drop table answers;

-- Create a new table answers

CREATE TABLE answers (

answers\_id VARCHAR(60) PRIMARY KEY,

answers\_author\_id VARCHAR(60),

answers\_question\_id VARCHAR(60),

answers\_date\_added date,

answers\_body TEXT

);

-- 10.

-- tag\_users

-- created on 07/25/19

-- drop table if it exists

drop table tag\_users;

-- Create a new table tag\_users

CREATE TABLE tag\_users(

tag\_users\_tag\_id INT PRIMARY KEY,

tag\_users\_user\_id VARCHAR(60)

);

-- 11.

-- tag\_questions

-- created on 07/25/19

-- drop table if it exists

drop table tag\_questions;

-- Create a new table tag\_questions

CREATE TABLE tag\_questions (

tag\_questions\_tag\_id INT PRIMARY KEY,

tag\_questions\_question\_id VARCHAR(60)

);

-- 12.

-- questions

-- created on 07/24/19

-- drop table if it exists

drop table questions;

-- Create a new table questions

CREATE TABLE questions (

questions\_id VARCHAR(60),

questions\_author\_id VARCHAR(60),

questions\_date\_added date,

questions\_title TEXT,

questions\_body TEXT

);

-- 13.

-- group\_memberships

-- created on 07/25/19

-- drop table if it exists

drop table group\_memberships;

-- Create a new table group\_memberships

CREATE TABLE group\_memberships (

group\_memberships\_group\_id VARCHAR(60) PRIMARY KEY,

group\_memberships\_user\_id VARCHAR(60)

);

-- 14.

-- comments

-- created on 07/24/19

CREATE DATABASE careervillage;

--drop table if it exists

drop table comments;

-- Create a new table comments

CREATE TABLE comments (

comments\_id VARCHAR(60),

comments\_author\_id VARCHAR(60),

comments\_parent\_content\_id VARCHAR(60),

comments\_date\_added DATE,

comments\_body TEXT

);

-- 15.

-- questions\_score

-- created on 07/24/19

--drop table if it exists

drop table question\_scores;

-- Create a new table question\_scores

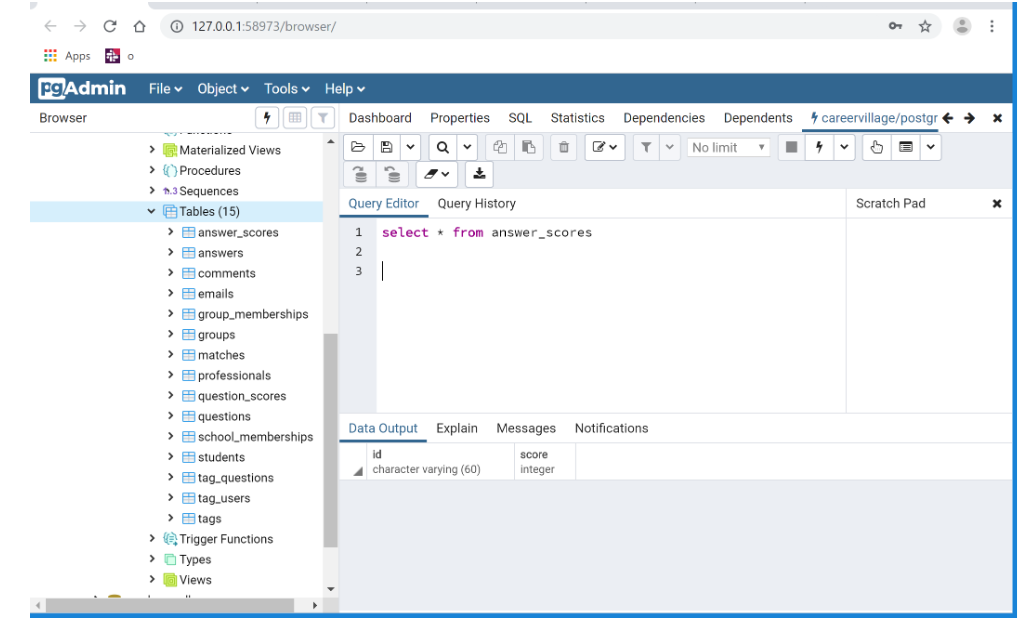
CREATE TABLE question\_scores(

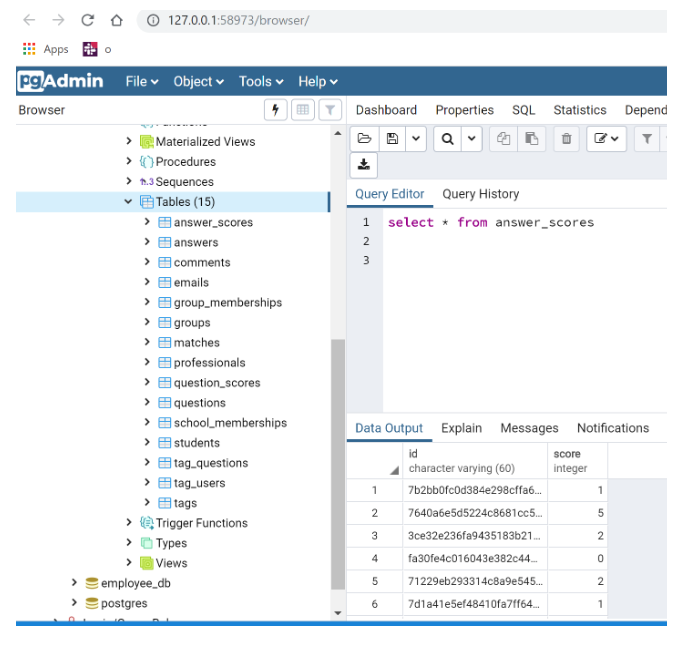
id VARCHAR(60),

score INT

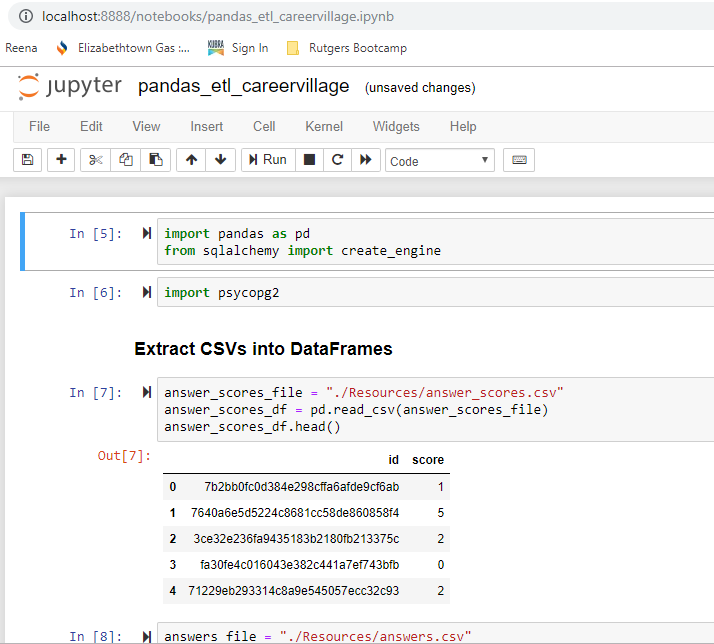
)

**PostgreSQL Database snapshot:**





**Notebook Script snapshot:** .ipyb file is present in Lisa’s folder



**Should be able to write queries from notebook, see below snapshot.**

