DAILY DOSE OF DOPAMINE

A Database Experiment

Kendra Danforth Osburn

IST 659 Database Administration Concepts

and Database Management

12.08.18

INTRODUCTION

Our historical moment is in many ways characterized by negativity and conflict. Between climate change, mass shootings, fake news, racial tensions, and an increasingly contentious political and cultural landscape, people are more stressed and divided than ever. Clearly, we're suffering from a shortage of dopamine—and while technology has undoubtedly magnified the problem, it may also be the solution.

For this assignment, I decided to reprise one of my very first online projects: Daily Dose of Dog. Numerous studies have shown that household pets have a powerful impact on humans' wellbeing. Thanks to the internet, my mother and I can enjoy the emotional benefits of pet ownership by trading cute photos of dogs. But in 2012, before Instagram, we had no easy way of sharing those photos, other than via email (a tool my mother still didn't fully understand). To address this issue, I created Daily Dose of Dog, a user-generated stream of cute dog photos uploaded on the Internet.

Today, I've realized that sharing cute dog photos is just one of many ways humans can use technology to hack our brain chemistry, stimulating the production of dopamine and other hormones linked to happiness. From videos, to webcomics, to inspirational stories of all kinds, the internet is bursting with a constant supply of mood-lifting media. Daily Dose of Dopamine will aggregate, sort, and display those media based on user preferences.

EXECUTABLE CODE -- DATABASE CREATION

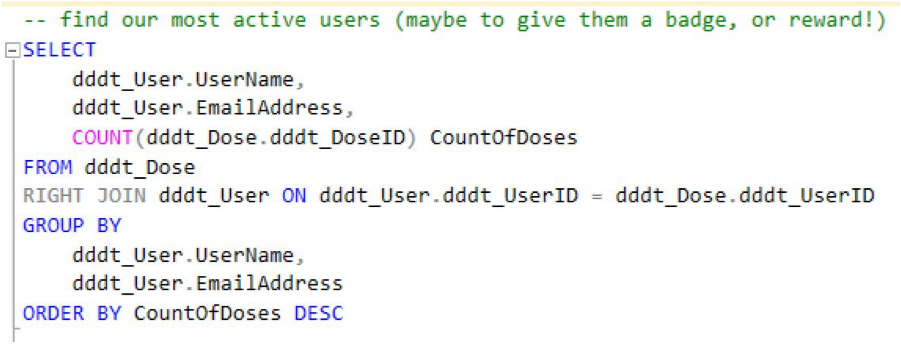
1. https://gist.github.com/yesthisiskendra/5a03556ce03a7f1e04ac51f15a6de0cc

CODE TO ANSWER DATA QUESTIONS

1. https://gist.github.com/yesthisiskendra/1d2a2699342a4c1c96e2cc491ba5ec08

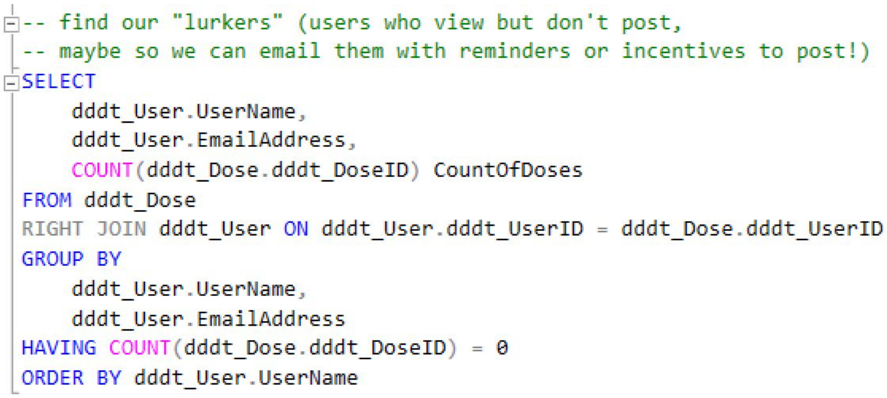
ANSWERING DATA QUESTIONS

1. Which users contribute the most to the community?



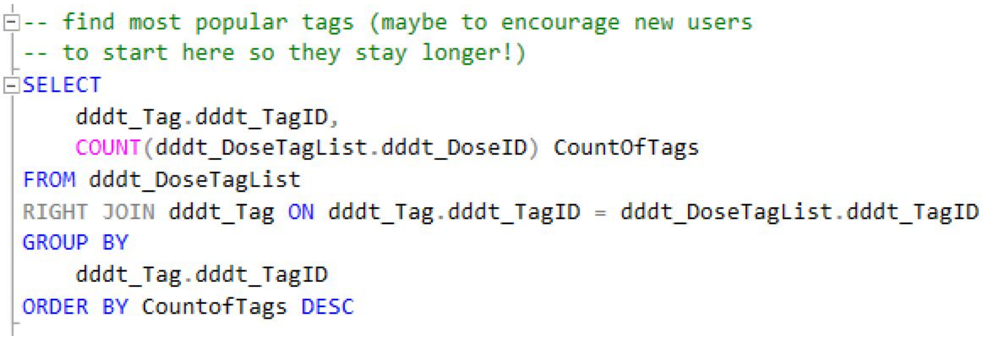
```
-- find our most active users (maybe to give them a badge, or reward!)
SELECT
    dddt_User.UserName,
    dddt_User.EmailAddress,
    COUNT(dddt_Dose.dddt_DoseID) CountOfDoses
FROM dddt_Dose
RIGHT JOIN dddt_User ON dddt_User.dddt_UserID = dddt_Dose.dddt_UserID
GROUP BY
    dddt_User.UserName,
    dddt_User.EmailAddress
ORDER BY CountOfDoses DESC
```

2. Which users are part of, but do not contribute to, the community?



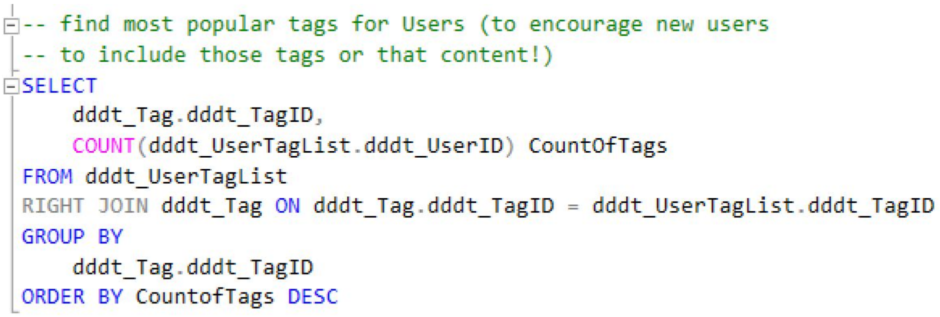
```
-- find our "lurkers" (users who view but don't post,
-- maybe so we can email them with reminders or incentives to post!)
SELECT
    dddt_User.UserName,
    dddt_User.EmailAddress,
    COUNT(dddt_Dose.dddt_DoseID) CountOfDoses
FROM dddt_Dose
RIGHT JOIN dddt_User ON dddt_User.dddt_UserID = dddt_Dose.dddt_UserID
GROUP BY
    dddt_User.UserName,
    dddt_User.EmailAddress
HAVING COUNT(dddt_Dose.dddt_DoseID) = 0
ORDER BY dddt_User.UserName
```

3. Which tags are used the most frequently?



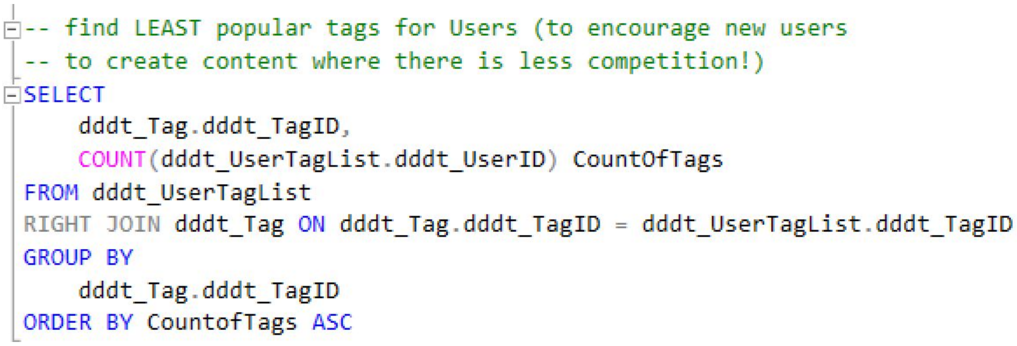
```
-- find most popular tags (maybe to encourage new users
-- to start here so they stay longer!)
SELECT
    dddt_Tag.dddt_TagID,
    COUNT(dddt_DoseTagList.dddt_DoseID) CountOfTags
FROM dddt_DoseTagList
RIGHT JOIN dddt_Tag ON dddt_Tag.dddt_TagID = dddt_DoseTagList.dddt_TagID
GROUP BY
    dddt_Tag.dddt_TagID
ORDER BY CountofTags DESC
```

4. Which tags are most used among users?



```sql
-- find most popular tags for Users (to encourage new users
-- to include those tags or that content!)
SELECT
    dddt_Tag.dddt_TagID,
    COUNT(dddt_UserTagList.dddt_UserID) CountOfTags
FROM dddt_UserTagList
RIGHT JOIN dddt_Tag ON dddt_Tag.dddt_TagID = dddt_UserTagList.dddt_TagID
GROUP BY
    dddt_Tag.dddt_TagID
ORDER BY CountofTags DESC
```

5. Which tags are used least?



```sql
-- find LEAST popular tags for Users (to encourage new users
-- to create content where there is less competition!)
SELECT
    dddt_Tag.dddt_TagID,
    COUNT(dddt_UserTagList.dddt_UserID) CountOfTags
FROM dddt_UserTagList
RIGHT JOIN dddt_Tag ON dddt_Tag.dddt_TagID = dddt_UserTagList.dddt_TagID
GROUP BY
    dddt_Tag.dddt_TagID
ORDER BY CountofTags ASC
```

PHYSICAL DATABASE DESIGN

See Appendix iii

DATA CREATION -- INSERT

See Appendix iv for a snippet

IMPLEMENTATION

See inbetween.fun

REFLECTION & SUMMARY

Well, I did it again and totally overcommitted to a second project so I'm left with two half complete projects and a deadline that is slowly fading in the rear-view. I could pepper you, dear reader, with excuses (all "valid" -- "valid" here meaning "if I asked 100 people in the parking lot of the supermarket, most would agree that this is indeed a appropriate and acceptable excuse for either (1) a belated assignment or (2) an extension" -- and very, very real. However, I didn't ask for an extension because I was too proud and I definitely didn't actually pole 100 people in a parking lot, and, as a budding data scientist, I feel exceptionally obligated to let you know that in addition to this being totally hypothetical and inside my own head, this "study" has almost no academic integrity due to the (1) lack of standards (2) lack of existence and should be taken with a sizable grain of salt (and equal quantity of humor)), the most important being a minor allergic reaction turned into flu-like symptoms, however, the real reason this isn't on time is because instead of focusing on the initial project from Part One -- Fantasy Mixtape League -- I spun up an entirely new project. You see, I got supremely hung up on my "business questions" and found myself in a bit of a hard place -- the projects with the most realistic and "good" business questions were slightly beyond my scope of database implementation. Examples of projects I considered based solely on business questions include:

Car accidents in the united states. Interesting business questions might be (but are definitely not limited to)

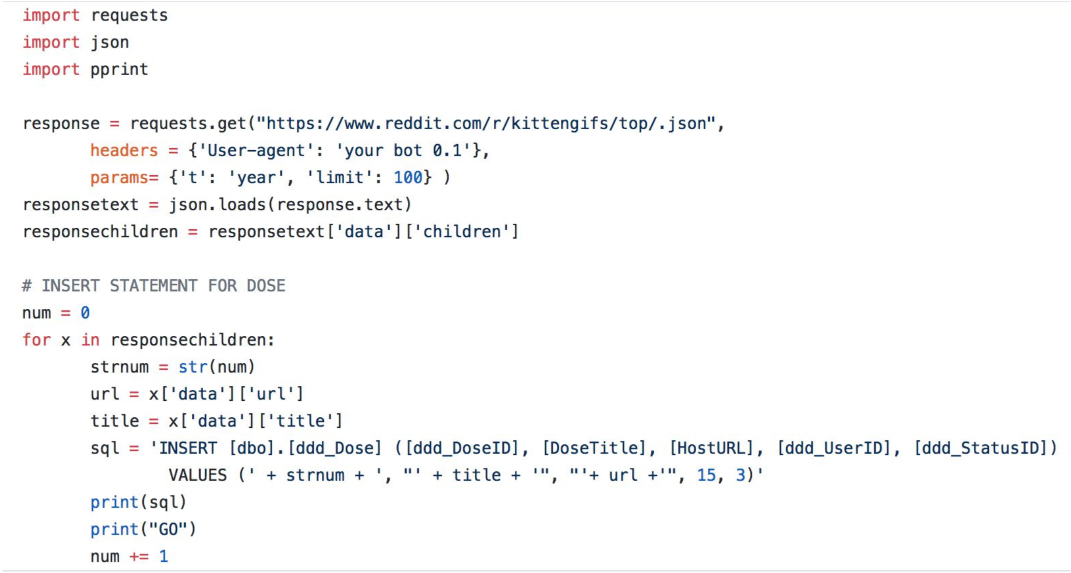
1. What type of car is most likely to be involved in an accident? Make, model, color, year?
2. What type of person is most likely to be involved in an accident? Gender, age, ethnicity?
3. What accident-inducing action is most likely to be fatal? Lane changing, distracted driving, left-turning?
4. What type of day is most "dangerous" for drivers? Dangerous meaning highest concentration of accidents.
5. DREAM QUESTIONS THAT WOULD BE CHALLENGING TO ANSWER CURRENTLY:
    a. What type of cellular smart device, if any, did the driver of the accident-causing car own?
    b. What song, if any, was the driver of the accident-causing car listening to? Potential to cross-reference driver's Spotify or Apple Music account. What BPM, if any, is more likely to be associated with accidents?

See Appendix i titled "Business Questions" for the five other projects I considered.

And (picking up right where we left off in the paragraph above, now you can see why it is so hard for me to stick to a project when I can hardly stick to a sentence or even a single aside) the projects that were within my comfort-level of database creation (limited joins and foreign keys because there is a positive linear relationship between "Number of joins and foreign keys" and "Likelihood Kendra will mess something up beyond repair" and while it's good to practice messing up and accidentally deleting things, a beginner course is enough mental consternation in and of itself to save this level of complexity for later projects.

Instead of diligently answering question (1) from Milestone Two for my initial project, I decided to dive back into my php days and create a whole new web app just for this project. An excellent use of time, yes? In addition to having to refresh myself on the coding ins and outs of a language I haven't used in over a decade, I decided this would be an excellent time to experiment with the Reddit API to populate my databases. So instead of spending time answering the questions asked of me, I wrote neat little python scripts to help me bulk "write" my INSERT statements. While this was thrilling and exciting, I completely forgot that SQL in the wild is different from MSSQLServer so I had to spend time "debugging" that and realizing that MSSQL has some pretty nifty features that ferrel SQL doesn't (IF EXISTS being one of them).

Here is the python script for reference:



```
import requests
import json
import pprint

response = requests.get("https://www.reddit.com/r/kittengifs/top/.json",
       headers = {'User-agent': 'your bot 0.1'},
       params= {'t': 'year', 'limit': 100} )
responsetext = json.loads(response.text)
responsechildren = responsetext['data']['children']

# INSERT STATEMENT FOR DOSE
num = 0
for x in responsechildren:
       strnum = str(num)
       url = x['data']['url']
       title = x['data']['title']
       sql = 'INSERT [dbo].[ddd_Dose] ([ddd_DoseID], [DoseTitle], [HostURL], [ddd_UserID], [ddd_StatusID])
              VALUES (' + strnum + ', "' + title + '", "'+ url +'", 15, 3)'
       print(sql)
       print("GO")
       num += 1
```

APPENDIX i:

BUSINESS QUESTIONS

CODESPARK RETENTION BUSINESS QUESTIONS:

codeSpark is the company I work for. We teach kids to code through games.

1. When do people usually unsubscribe?
2. When was the last time an unsubscriber logged in?
3. How many unsubscribers have had no gameplay time?
4. Who are the people most likely to unsubscribe?
5. Do gift-incentives (such as plush toys) help increase subscriptions?

CODESPARK DASHBOARD BUSINESS QUESTIONS:

There is a teacher component of codeSpark that I'm actively involved in because our program is starting to be used in classrooms across the globe.

1. Which teacher is the most active?
2. Which teacher has the most students?
3. Which classrooms have the highest average?
4. Is there a relationship between minutes played and game success?
5. Where are there optimization opportunities?

OPTI-FRIDGE BUSINESS QUESTIONS:

This is my dream smart fridge where things are scanned, either from a raspberry-pi DIY door scanner attached to the inside door of the fridge (hard mode) or from using OCR on our receipts (easy mode -- I really should be doing this already to track our fridge contents).

1. What single ingredient can be added to most meals?
2. Which protein appears the most often?
3. Which vegetable can be eaten in the most ways?
4. Which meal would take the least amount of time to cook?
5. What is the cheapest menu with the most variety?
6. Which ingredients are least expensive?
7. What ingredients appear in most recipes?
8. What ingredient has the most protein?
9. What ingredient has the best protein-per-cost ratio?
10. Which five ingredients should I always have in my fridge?

CLUELESS CLOSET:

Attempting to answer the age-old question, "What do I wear today?"

1. Which pair of pants is the most versatile (how many outfits is it part of?)
2. Which top is the most versatile (how many outfits is it part of?)
3. Which pair of shoes is the optimal pair for comfort and outfit variability?
4. Which outfit is "the best" outfit (based on quality/name brand)?
5. Which outfit is the most diverse (can be worn in any climate/with modifications)?

APPENDIX ii EXECUTABLE CODE:

My own riff on "VidCasts"

https://gist.github.com/yesthisiskendra/5a03556ce03a7f1e04ac51f15a6de0cc

APPENDIX iii (see separate document, too large for this one)

APPENDIX iv (see separate document, too large for this one)