Data engineering

Project theme: Internet Memes

Group members: Ülle Püttsepp, Allan Mitt, Costa Thomas
Data Engineering course, fall 2021, University of Tartu

I. Choosing a dataset

The first purpose of this project was to choose a dataset among 3 provided. We have decided to work with the dataset kym.JSON because we think this is the dataset containing the most useful data usable for the queries we want to implement after the creation of the entire pipeline.

To understand the dataset more precisely, we have used the software Dadroit JSON Viewer. It is a software allowing us to open a heavy JSON file and offering a nice presentation of JSON objects.

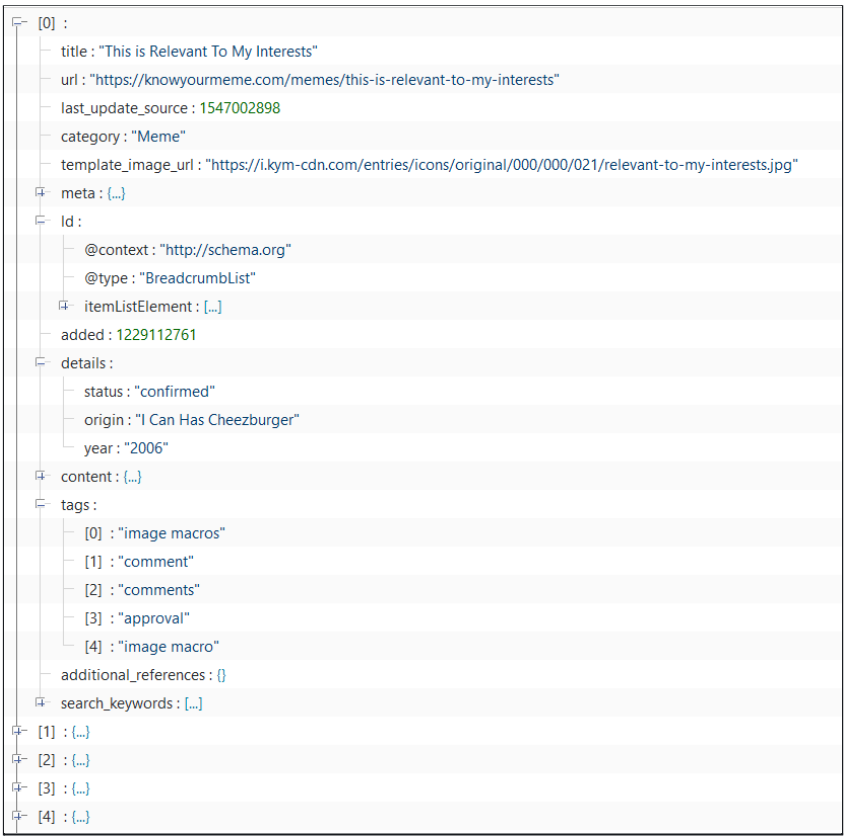

Figure 1. Screenshot of the software Dadroit JSON Viewer

II. Data cleansing

In this first part, we are going to set up a process to fix and remove incorrect, corrupt, duplicate or incomplete data within the dataset. This part is essential because we want to avoid all mistakes in the databases.

After studying in depth the keys of the JSON objects, we can clean the data in the following way:

The first thing we can notice is that they are different possible values for the key *category*:

```
# Check all categories in database
df.category.unique()
```

```
array(['Meme', 'Subculture', 'Event', 'Culture', 'Site', 'Person'],
      dtype=object)
```

Figure 2. Categories

As we can see, there are several categories, and one of them is called *Meme*. Consequently, we can drop all JSON objects with the key category different to *Meme.* Then, the *category* attributes become redundant itself because all the fields contain one and the same value *Meme*, so we can drop this *category*.

Secondly, we notices in the object details the key status composed with the following values:

```
import numpy as np
details_value = []
for index,element in enumerate(df.details):
    details_value.append(element["status"])
print(np.unique(np.array(details_value)))
```

```
['confirmed' 'deadpool' 'submission' 'unlisted']
```

Figure 3. Details

Then, we can check manually the memes that contain the value *deadpool*. According to the website Knowyourmeme.com, it means that the meme is inappropriate:

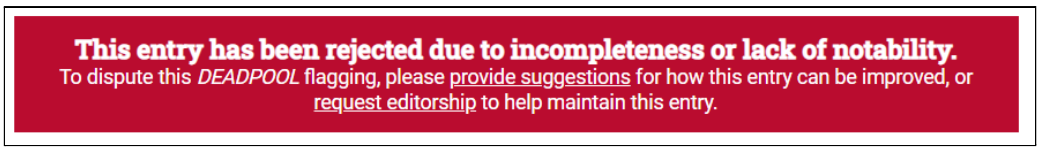

Figure 4. Inappropriate memes

In the case of *submission*, it means the meme is under review and for *unlisted* memes, the information regarding its content is unknown. Consequently, in order to avoid inappropriate memes if they are *unlisted* or in the case of *submission,* we can use the library profanity-check and check the content of the key *og:title* to know if the meme is inappropriate or not. Indeed, this library allows us to check each word of a string and determine if this string contains inappropriate words.

In the third phase, we can display the list of all possible values for the publication date of a same:

```
df.sort_values(['year'],ascending=False).year.unique()
```

```
array(['2916', '2500', '2343', '2107', '2104', '2074', '2069', '2050',
       '2025', '2020', '2019', '2018', '2017', '2016', '2015', '2014',
       '2013', '2012', '2011', '2010', '2009', '2008', '2007', '2006',
       '2005', '2004', '2003', '2002', '2001', '2000', '1999', '1998',
       '1997', '1996', '1995', '1994', '1993', '1992', '1991', '1990',
       '1989', '1988', '1987', '1986', '1985', '1984', '1983', '1982',
       '1981', '1980', '1979', '1978', '1977', '1976', '1975', '1974',
       '1973', '1972', '1971', '1970', '1969', '1968', '1967', '1966',
       '1965', '1964', '1963', '1962', '1961', '1960', '1959', '1958',
       '1957', '1956', '1955', '1954', '1953', '1952', '1951', '1950',
       '1949', '1948', '1947', '1946', '1945', '1944', '1943', '1942',
       '1941', '1940', '1939', '1938', '1937', '1935', '1933', '1932',
       '1931', '1930', '1929', '1928', '1927', '1926', '1925', '1924',
       '1923', '1922', '1921', '1920', '1919', '1918', '1916', '1915',
       '1914', '1913', '1912', '1911', '1910', '1909', '1908', '1907',
       '1906', '1905', '1904', '1903', '1902', '1900', '1897', '1894',
       '1893', '1891', '1890', '1889', '1888', '1880', '1878', '1876',
       '1874', '1869', '1867', '1865', '1863', '1861', '1857', '1848',
       '1846', '1844', '1839', '1837', '1829', '1814', '1809', '1800',
       '1797', '1795', '1792', '1790', '1785', '1780', '1776', '1775',
       '1774', '1753', '1717', '1700', '1621', '1605', '1600', '1596',
       '1590', '1589', '1580', '1564', '1561', '1531', '1503', '1495',
       '1490', '1445', '1415', '1392', '1341', '1336', '1232', '1212',
       '1111', '1100', '1070', '1066', '1000', None], dtype=object)
```

Figure 5. All possible publication dates

According to Wikipedia, the website Know Your Meme was created in 2007. Consequently, all memes with a publication date out of the range [2007;2021] must be removed because they contain a wrong value for their publication date.

Regarding the description and the URL links available, we are facing redundant data. For example, the keys *title, og:title and twitter:title* provided the same information, the purpose is to keep just the *title* of the meme.

Figure 6. Same title in different keys

In the same way, *twitter:description, description* and *og:description* contain the same value. We just have to keep the key *description.* In the case of the URL links, we just keep the key *url.*

Once we have removed all the redundant data, we can clean it. Regarding the description, we have noticed that some strings contain unicode characters because these descriptions have been written in other languages. One thing we can do is to remove these unicode characters.

In addition, a useful and essential tool we can use is to check the validity of the url links, to know if they are broken or not.

Finally, last but not least, we have noticed that some tags contained in the key *tags* are redundant because they appear in singular and plural form, or with a first letter in capital letter and a second time in lower case. Consequently, we must transform all the words in lowercase and singular form.

III. Data Transformation

As we can see, the key *description* is the assembly of multiple sentences. However, this cannot be interpreted by the machine and our goal is to transform it into more readable content for the machine. In this way, we are going to use a library to create a dictionary where each word (key) has an associated occurrence in the description. In one sentence we keep only the key words. In addition, we can add a column containing the number of words in the description.

When we look at the keys *last_update_source* and *added,* we can't interpret these information. We need to transform this format into a Date format. We don't know yet how to convert this data into a date.

In the same way, it is difficult to work with an url link. Consequently, in the case of the key *children* there are multiple links and the idea is to keep only the last part of the string. Especially the part containing the child name of the meme. Regarding the links provided in the key *about,origin,* and *spread,* the idea is to keep only the name of the meme related rather than keep the URL links which is irrelevant information.

IV. Data modeling

Firstly, we can create the SQL relational model using the star schema:

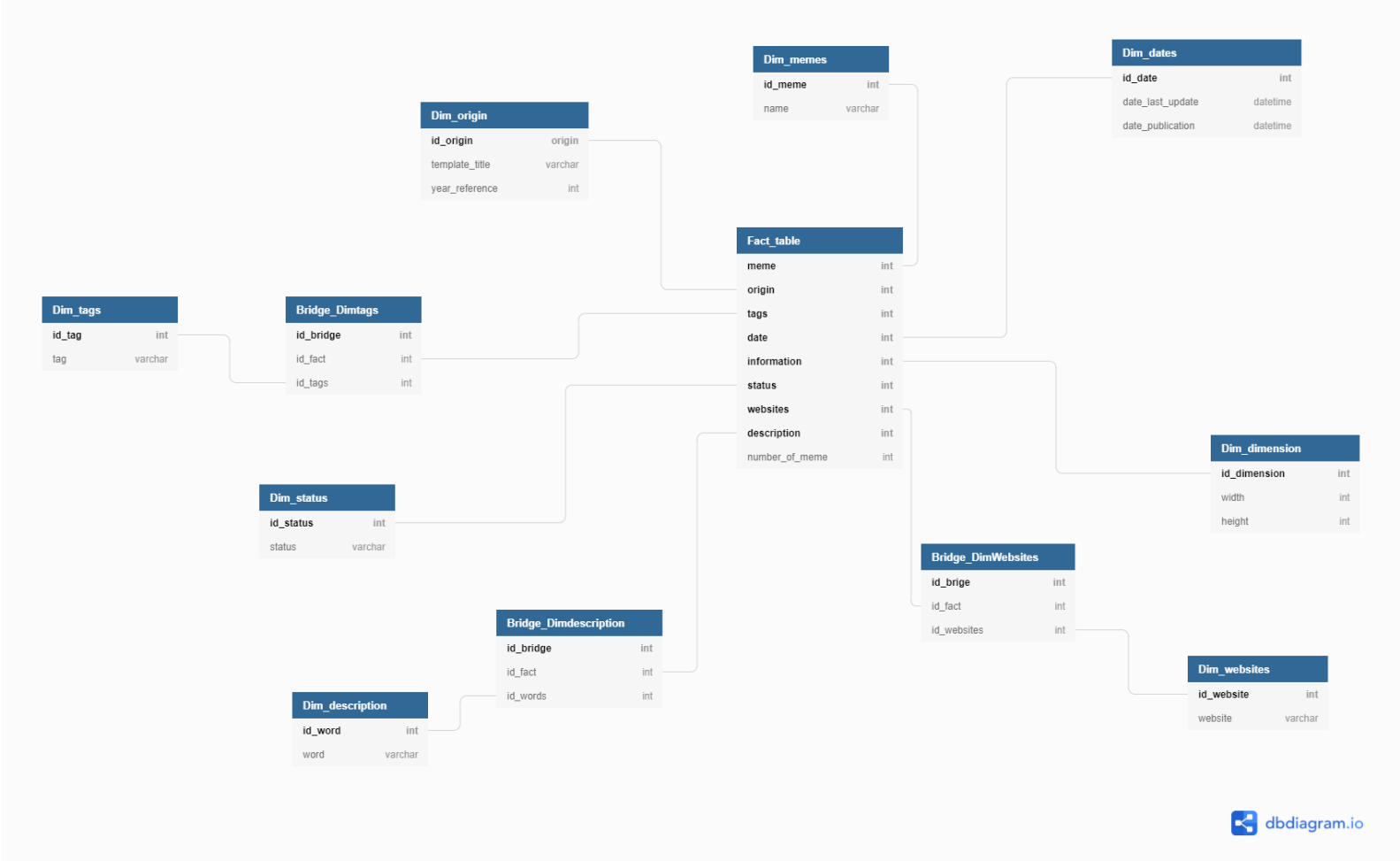

Figure 7. Logical level of the SQL relational model

Fact_table(*meme,origin,tags,date,information,status,*<u>**websites**</u>,<u>**description**</u>,number_of_meme)
Dim_memes(<u>**id_meme**</u>,name)
Dim_dates(<u>**id_date**</u>,date_last_update,date_publication)
Dim_dimension(<u>**id_dimension**</u>,width,height)
Bridge_DimWebsites(<u>**id_brige**</u>,*id_fact*,*id_website*)
Dim_websites(<u>**id_websites**</u>,website)
Bridge_Dimdescription(<u>**id_bridge**</u>,*id_fact,id_words*)
Dim_description(<u>**id_word**</u>,word)
Dim_states(<u>**id_status**</u>,status)
Bridge_DimWebtags(<u>**id_brige**</u>,*id_fact*,*id_tags*)
Dim_tags(<u>**id_tag**</u>,tag)
Dim_origin(<u>**id_origin**</u>,template_title,year_reference)

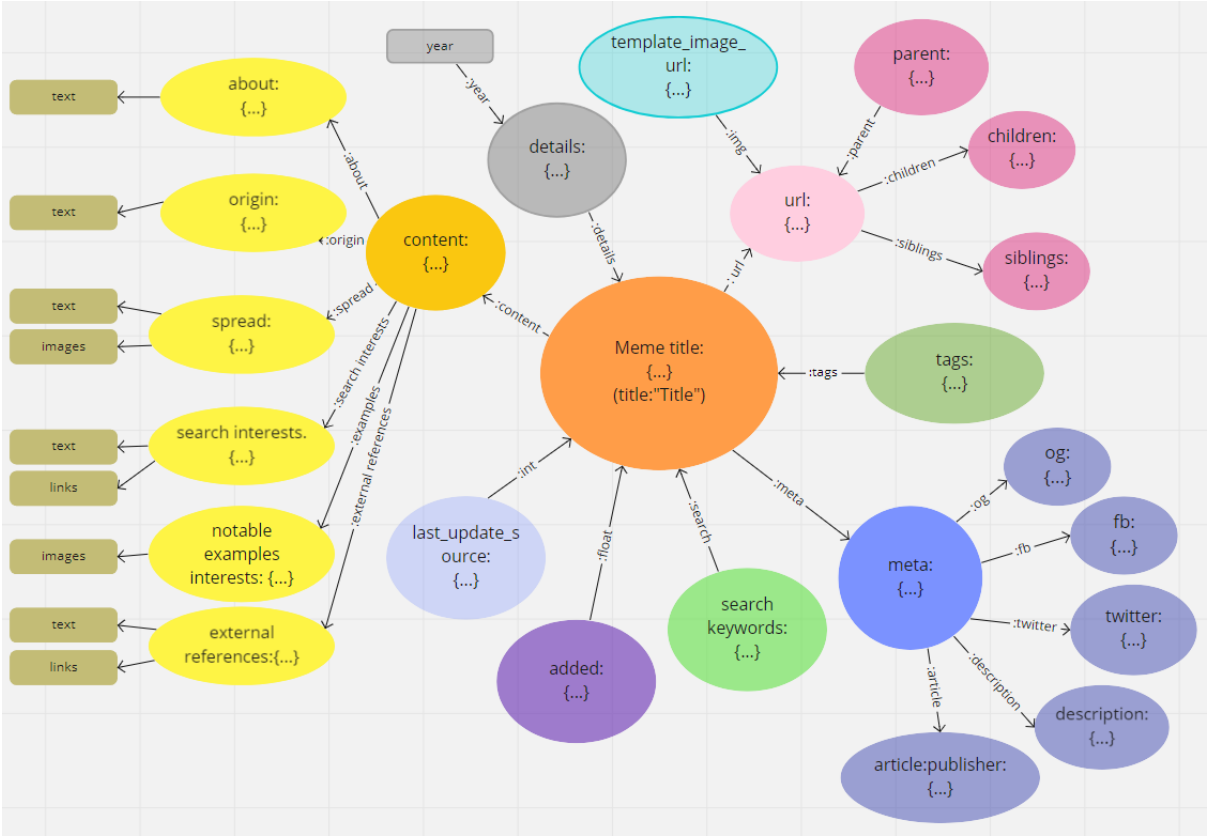

Figure 8. Graph conceptual design

The graph concept follows the idea of a property graph model. As a social media phenomenon, the topic of Mems consists of a lot of relationships, suitable for graph models. Relationships are directed edges between nodes, marked by a shortened label for space sparing purposes in the Fig 8. The nodes have key-value structure.

V. Queries

One of the most interesting groups of queries relates to the popularity of the memes. This can be estimated via reflections in other social media platforms (fb, twitter). In the Content object texts describe the popularity of a particular Meme. A query would assume textual analysis, filtering of relevant words, and based on frequency of occurrence (described in Data Transformations). We could cluster Memes by similarity, based on similar techniques. Connectivity with other Memes can be queried based on parents, children and siblings. Similarly, the Memes not related to others can be counted. The character of the links between Memes should be defined clearly. We can also study the dynamics of Memes through years, to find out trends about most populated periods.

VI. Distribution of work during the first part of the project

Task	Members
Analysis of data/ Choosing a dataset	Thomas, Ülle, Allan
Data cleansing	Thomas
Data enrichment	Allan
Data transformation	Thomas
Data modeling	Thomas, Ülle
Queries	Ülle