

Compte rendu

Objectif : L'objectif du projet est de créer un pipeline streaming avec pour finalité de créer des graphes pour afficher les données recueillies.

Afin de réaliser ce pipeline, nous avons utilisé une stack composé de 4 technologies à savoir Apache Spark (Avec Spark streaming et Spark SQL), Apache Kafka, Une base de données MySQL et Apache superset pour la visualisation.

L'objectif du projet est donc de recueillir des données en temps réel via l'API de twitter, les envoyer sur le topic kafka que l'on va créer afin que notre Spark streaming récupère les données du topic, les traite et les insère dans la base de donnée MySQL pour que Superset puisse les lire et afficher le ou les graphes que l'on créera par la suite.

Nous avons commencé par installer les machines virtuelles une pour Spark, une pour Kafka, une pour exécuter nos scripts python qui produit la donnée et une pour la base de donnée et superset. Nous avons réalisé cela à l'aide de VirtualBox et Vagrant. (Le script utilisé dans le vagrantfile génère la configuration pour nous à savoir la gestion des ports, l'installation de Python, Jupyter, Spark, MySQL, Superset et Java pour la machine spark et Kafka.

Une fois les machines créées nous nous sommes connecté aux machines en utilisant la commande vagrant ssh.

Sur la machine Kafka, le broker et le zookeeper étant déjà instanciés grâce au fichier start.sh dans la configuration du Vagrantfile. Nous avons donc simplement créé le topic qui nous servira à connecter le producer et le consumer puis envoyer quelques données de test.

```
vagrant@vagrant:~/kafka/bin$ ./kafka-topics.sh --create --topic tweets-topic --bootstrap-server 192.168.33.13:9092  
Created topic tweets-topic.
```

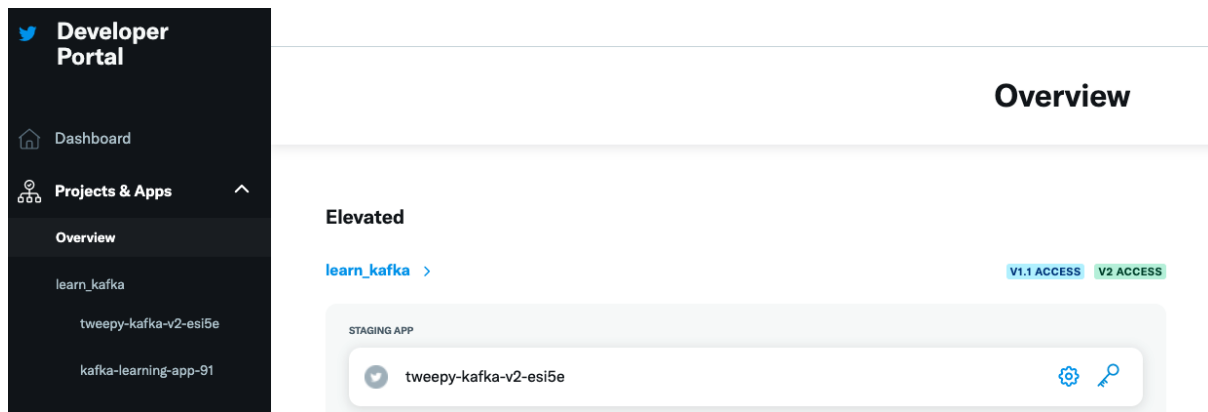
On vérifie qu'il est bien créé

```
vagrant@vagrant:~/kafka/bin$ ./kafka-topics.sh --list --bootstrap-server 192.168.33.13:9092  
tweets-topic
```

Nous avons créé le script permettant de produire de la donnée dans kafka en récupérant les tweets de l'API Twitter

L'idée d'origine était de récupérer la géolocalisation des tweets pour étudier dans quels pays ils étaient le plus envoyés etc...

Pour cela je me suis d'abord intéressé à l'objet "location" du tweet. Mais cet élément peut être inséré par les utilisateurs librement et certains ne l'utilisent pas pour donner leur ville ou leur pays. Ensuite j'ai essayé avec l'objet "place" mais mon compte étant au rang "elevated"



Je n'ai pas le niveau le plus élevé et ne peux pas avoir ces informations.

J'ai donc décidé d'étudier uniquement les hashtags des le nombre de followers des utilisateurs.

Pour ce faire, nous avons installé kafka-python sur la machine puis nous avons écrit ce script.

```
python > vm > producer-streaming-twitter.py > ...
1 from tweepy.streaming import Stream
2 import time
3 from kafka import KafkaProducer
4 import tweepy
5 import json
6
7 API_KEY = "mwqLxJ64vVtWtoy8l08XHnwsj"
8 API_KEY_SECRET = "1B5xRMLMEapdFChbMUMTWZpK17Sp9J5usv4r0bZrjjqXWjIthV"
9 BEARER_TOKEN = "AAAAAAAAAAAAAAAAAAHK4bwEAAAAA3jfVvGdvNiRZjn%2By4H4tgcGUBlg%3Dnu30ECI0qb5Io7CeYTD5DHBuy98l77WkjAqUBA4oGxxrHi7iFd"
10 ACCESS_TOKEN = "898851040451321857-swnGgmKR2BKv6rkivheg4jJjpa4Aws"
11 ACCESS_TOKEN_SECRET = "7V5Dk09GR4Sy80webTtZCgGW1tpDE1UU5fereCu20xu2X"
12
13 producer = KafkaProducer(bootstrap_servers='192.168.33.13:9092')
14
15
16 class Listener(Stream):
17
18     tweetsProceeded = []
19     limit = 1000
20
21     def on_data(self, raw_data):
22         self.process_data(raw_data)
23         return True
24
25     def process_data(self, raw_data):
26         if len(self.tweetsProceeded) == self.limit:
27             print("disconnecting ...")
28             self.disconnect()
29         else:
30             message = json.loads(raw_data)
31             if message['user']['location'] is not None:
32                 print(message)
33                 producer.send('tweets-topic', raw_data)
34                 self.tweetsProceeded.append(raw_data)
35
36     def on_error(self, status_code):
37         if status_code == 420:
38             # returning false in on_data disconnects the stream
39             return False
```

```

41
42 # start the stream
43 if __name__ == "__main__":
44     auth = tweepy.OAuthHandler(API_KEY, API_KEY_SECRET)
45     auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
46
47     api = tweepy.API(auth)
48
49     listener = Listener(API_KEY, API_KEY_SECRET,
50                         ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
51
52     keywords = ["*"]
53
54     listener.filter(track=keywords)
55

```

Comme on peut le voir au moment ou nous lançons le script, les tweets sont affiché sur la console

```

vagrant@vagrant:~/vagrant$ python3 prodtools-streaming-twitter.py
{'created_at': 'Wed May 04 16:28:31 +0000 2022', 'id': '1521889520261627904', 'id_str': '1521889520261627904', 'text': '@Suzanne_Smith From the first testament *',
 'display_text_range': [15, 41], 'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', 'truncated': False, 'in_reply_to_status_id': '1521871374695817216', 'in_reply_to_status_id_str': '1521871374695817216', 'in_reply_to_user_id': 22652933, 'in_reply_to_user_id_str': '22652933', 'in_reply_to_screen_name': 'NDRumark', 'user': {'id': 22652933, 'id_str': '22652933', 'name': 'Doc Tas Denmark', 'screen_name': 'NDRumark', 'location': 'UT: 33.431212, -111.903974', 'url': 'http://culturedconvos.com', 'description': 'ED, AT, CPWR (she/her) Educator #AT #SocialJustice and #Equity Warrior #mother #wife a nd #activelove x u200d9 Belarus #New Jersey #California', 'translator_type': 'none', 'protected': False, 'verified': False, 'followers_count': 698, 'friends_count': 811, 'listed_count': 11, 'favourites_count': 21957, 'statuses_count': 6904, 'created_at': 'Tue Mar 03 18:04:04 +0000 2009', 'utc_offset': None, 'time_zone': None, 'geo_enabled': True, 'lang': 'None', 'contributors_enabled': False, 'is_translator': False, 'profile_background_color': 'BADCFD', 'profile_background_image_url': 'https://pbs.twimg.com/profile_images/1269794625/0J2UHNND4A_normal.jpg', 'profile_image_url': 'https://pbs.twimg.com/profile_images/1269794625/0J2UHNND4A_normal.jpg', 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/22652933/156168171627', 'profile_link_color': '000000', 'profile_sidebar_border_color': 'F2F2F2', 'profile_sidebar_fill_color': 'FFFFFF', 'profile_text_color': '003366', 'profile_use_background_image': True, 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1478436621128794625/0J2UHNND4A_normal.jpg', 'default_profile': False, 'default_profile_image': False, 'following': None, 'follow_request_sent': None, 'notifications': None, 'withheld_in_countries': [], 'geo': None, 'coordinates': None, 'place': None, 'contributors': None, 'quote_status': False, 'quote_count': 0, 'reply_count': 0, 'retweet_count': 0, 'favorite_count': 0, 'entities': {'hashtags': [], 'urls': [], 'user_mentions': [{'screen_name': 'Suzanne_Smith', 'name': 'Suzanne', 'id': 19152377, 'id_str': '19152377', 'indices': [0, 14]}, {'screen_name': 'NDRumark', 'name': 'NDRumark', 'id': 22652933, 'id_str': '22652933', 'indices': [15, 31]}]}, 'symbols': [], 'favorited': False, 'retweeted': False, 'filter_level': 'low', 'lang': 'en', 'timestamp_ms': '156168171627'}
```

une fois que le nombre de message à été produit, le stream se stop :

```

● ● ● vm — vagrant@vagrant:/vagrant — ssh - vagrant ssh — 147x33
$ curl -in -reply to status_id:str:None, 'in_reply_to_status_id':None, 'in_reply_to_screen_name':None, 'user':{'id':1896613713533082017, 'me_str':'id_str':1896613713533082017, 'name':'pous', 'screen_name':'ibesthalo','location':'1989', 'url':'https://ibesthalo.carrd.co', 'description':'me & a z&#x27;y herry', 'translator_type':None, 'protected':False, 'verified':False, 'followers_count':1994, 'friends_count':1252, 'listed_count':14, 'favourites_count':31277, 'statuses_count':26419, 'created_at':'Wed Jan 30 14:12:26 +0000 2019', 'utc_offset':None, 'time_zone':None, 'geo_enabled':False, 'lang':None, 'contributors_enabled':False, 'is_translator':False, 'profile_background_color':'F5F8FA', 'profile_background_image_url':'', 'profile_background_image_url_https':'', 'profile_link_color':'1DA1F2', 'profile_sidebar_border_color':'C0DEED', 'profile_sidebar_fill_color':'DDEEFF', 'profile_text_color':'333333', 'profile_use_background_image':True, 'profile_images':http://pbs.twimg.com/profile_images/147056377864980680/HtSMWzZo_normal.jpg, 'profile_image_url_https':'https://pbs.twimg.com/profile_images/147056377864980680/HtSMWzZo_normal.jpg', 'profile_banner':http://pbs.twimg.com/profile_banners/1896613713533082017/1622418340, 'default_profile':True, 'default_profile_image':False, 'following_request_sent':None, 'notifications':None, 'withheld_in_countries':[], 'geo':None, 'coordinates':None, 'place':None, 'contributors':None, 'quoted_status_id':1524854914127749938, 'quoted_status_str':'1524854914127749938', 'quote_status':{'created_at':'Thu May 12 20:51:56 +0000 2022', 'id':1524854914127749938, 'text':'1524854914127749938', 'text':'tengo q estudiar un montón \nse acuesta s', 'source':{'<a href=http://twitter.com/download/iphone rel=nofollow>Twitter for iPhone</a>', 'truncated':False, 'in_reply_to_status_id':None, 'in_reply_to_user_id':None, 'in_reply_to_status_id':None, 'in_reply_to_screen_name':None, 'user':{'id':1242122123549827075, 'id_str':'1242122123549827075', 'name':'said 2.', 'screen_name':'saidstatiksta', 'location':None, 'url':'https://www.instagram.com/saidstatiksta/', 'description':None, 'translator_type':None, 'protected':False, 'verified':False, 'followers_count':7174, 'friends_count':7, 'listed_count':9, 'favourites_count':344, 'statuses_count':103, 'created_at':'Mon Mar 23 16:13:09 +0000 2020', 'utc_offset':None, 'time_zone':None, 'geo_enabled':False, 'lang':None, 'contributors_enabled':False, 'is_translator':False, 'profile_background_color':'F5F8FA', 'profile_background_image_url':'', 'profile_bg_image_url_https':'', 'profile_link_color':'1DA1F2', 'profile_sidebar_border_color':'C0DEED', 'profile_sidebar_fill_color':'DDEEFF', 'profile_text_color':'333333', 'profile_use_background_image':True, 'profile_image_url':http://pbs.twimg.com/profile_images/1514254126678349826/VSKrThwS_normal.jpg, 'profile_image_url_https':'https://pbs.twimg.com/profile_images/1514254126678349826/VSKrThwS_normal.jpg', 'file_image_path':http://pbs.twimg.com/profile_images/1514254126678349826/VSKrThwS_normal.jpg, 'default_profile':True, 'default_profile_image':False, 'following_request_sent':None, 'notifications':None, 'withheld_in_countries':[], 'geo':None, 'coordinates':None, 'place':None, 'contributors':None, 'is_quote_status':False, 'quote_count':992, 'reply_count':143, 'retweet_count':18571, 'favorite_count':94886, 'entities':{'hashtags':[], 'urls':[], 'user_mentions':[], 'symbols':[], 'favorited':False, 'retweeted':False, 'filter_level':'low', 'lang':'es'}, 'quoted_status_permalink':{'url':'https://t.co/Ylsf9HRVDe', 'expanded':{'https://twitter.com/saidstatik/status/1524854914127749938', 'display':'twitter.com/saidstatik/status'}, 'is_quote_status':True, 'quote_count':0, 'reply_count':0, 'retweet_count':0, 'favorite_count':0, 'entities':{'hashtags':[], 'urls':[], 'user_mentions':[], 'symbols':[], 'favorited':False, 'retweeted':False, 'filter_level':'low', 'lang':'es'}, 'timestamp_ms':'165252972991'}}}
discarding...
Stream connection closed by Twitter
vagrant@vagrant:~$

```

En lançant un kafka-console-consumer nous voyons bien les tweets passer aussi

```
{
  "avatar_url": "https://kafka/bin/j.kafk-console-consumer.sh --bootstrap-server 192.168.33.10:9992 -from-beginning --topic tweets-topic",
  "created_at": "Wed May 04 16:28:31 +0000 2022", "id": 152188952061627904, "id_str": "152188952061627904", "text": "@Suzanne_Smith From the first testament #, \"display_text_range\": [15, 41], \"source\": \"iPhone3c via href=\"http://v/twittter.com/download/viphone\" rel=\"nofollow\"#u003eTwitter for iPhone#u003c/u003e\", \"truncated\": false, \"in_reply_to_status_id\": 1521871374695817216, \"in_reply_to_status_id_str\": \"1521871374695817216\", \"in_reply_to_user_id\": 22652933, \"in_reply_to_user_id_str\": \"22652933\", \"in_reply_to_screen_name\": \"NDRmark\", \"user\": {\"id\": 22652933, \"id_str\": \"22652933\", \"name\": \"Doc Tash Denmark\", \"screen_name\": \"NDRmark\", \"location\": \"\u200d\u200cdct: 33.432122, -111.903974\", \"url\": \"http://culturedconvos.com/\", \"description\": \"EdD, ATC, CPWR (she/her) #educator @AT #SocialJustice and #Equity Warrior U272Uf0ef #mother #wife and #activesoul u0d3 cufdc3\u200du0db3\u200duddfb\u200du200du2648\u200dufe0f Belarus u02a7u\u200fu0ef0f New Jersey u02a7u\u200fu0ef0f California\", \"translator_type\": \"none\", \"protected\": false, \"verified\": false, \"followers_count\": 698, \"friends_count\": 811, \"listed_count\": 11, \"favorites_count\": 21957, \"statuses_count\": 19904, \"created_at\": \"Tu e Mar 03 18:04:04 +0000 2009\", \"utc_offset\": null, \"time_zone\": null, \"geo_enabled\": true, \"lang\": null, \"contributors_enabled\": false, \"is_translator\": false, \"profile_background_color\": \"BADFCD\", \"profile_background_image_url\": \"http://abs.twimg.com/images/themes/theme12/bg.gif\", \"profile_background_image_url_https\": \"https://abs.twimg.com/images/themes/theme12/bg.gif\", \"profile_background_tile\": false, \"profile_link_color\": \"FF0000\", \"profile_sidebar_border_color\": \"F2E195\", \"profile_sidebar_fill_color\": \"FFFFFF\", \"profile_text_color\": \"0C3E53\", \"profile_use_background_image\": true, \"profile_image_url\": \"http://pbs.twimg.com/profile_images/1478436621120794625/JUZnHND4_normal.jpg\", \"profile_image_url_https\": \"https://pbs.twimg.com/profile_images/1478436621120794625/JUZnHND4_normal.jpg\", \"default_profile_image\": false, \"following\": null, \"follow_request_sent\": null, \"notifications\": null, \"withheld_in_countries\": [], \"geo\": null, \"coordinates\": null, \"place\": null, \"contributors\": null, \"is_quote_status\": false, \"quote_count\": 0, \"reply_count\": 0, \"retweet_count\": 0, \"favorite_count\": 0, \"entities\": {\"hashtags\": [], \"urls\": [], \"user_mentions\": [{\"screen_name\": \"Suzanne_Smith\", \"name\": \"Suzanne\", \"id\": 19152377, \"id_str\": \"19152377\", \"indices\": [0, 14]}, {\"symbol\": \"$\": []}], \"favorited\": false, \"retweeted\": false, \"filter_level\": \"low\", \"lang\": \"en\", \"timestamp_ms\": \"1651681711627\"}
```

Puis nous avons créé le job Spark qui va consommer les données du topic

```

1 import string
2 from pyspark.sql import SparkSession, Row
3 from pyspark.streaming import StreamingContext
4 from pyspark.streaming.kafka import KafkaUtils
5 from pyspark.sql.types import FloatType, StringType
6 from pyspark.sql.functions import to_date
7 import json
8 import re
9
10 emoji_pattern = re.compile("[
11     u'\U0001F600-\U0001F64F' # emoticons
12     u'\U0001F300-\U0001F5FF' # symbols & pictographs
13     u'\U0001F680-\U0001F6FF' # transport & map symbols
14     u'\U0001F1E0-\U0001F1FF' # flags (iOS)
15     u'\U00002702-\U000027B0'
16     u'\U000024C2-\U0001F251'
17     u'\U0001F926-\U0001F937'
18     u'\U00010000-\U00010fff'
19     u'\u200d"
20     u"\u2640-\u2642"
21     u"\u2600-\u2B55"
22     u"\u23cf"
23     u"\u23e9"
24     u"\u231a"
25     u"\u3030"
26     u"\ufe0f"
27     "]+", flags=re.UNICODE)
28
29 def getHashtags(rdd_collected):
30     list_hashtags = []
31     for element in rdd_collected:
32         print(element)
33         for hashtag in element:
34             if hashtag["text"] != None:
35                 list_hashtags.append(hashtag["text"])
36     return list_hashtags
37
38 def stripTextAndRemoveEmoji(text):
39     stringWithoutEmoji = emoji_pattern.sub(r'', text)
40     nameIsBlank = re.search('^s*$', stringWithoutEmoji)
41     if not stringWithoutEmoji or nameIsBlank:
42         return "No name"
43     return stringWithoutEmoji.strip()
44
45
46
47 def process(time, rdd):
48     print("===== %s =====" % str(time))
49     if not rdd.isEmpty():
50         print(rdd)
51         locationDF = rdd.map(lambda tweet: Row(username=stripTextAndRemoveEmoji(tweet['user']['name']),
52             nb_friends=tweet['user']['friends_count'])).toDF()
53
54         locationDF = locationDF.withColumn("username", locationDF["username"].cast(StringType())) \
55             .withColumn("nb_friends", locationDF["nb_friends"].cast(IntegerType()))
56
57         locationDF.printSchema()
58
59         locationDF.write.format('jdbc').options(
60             url='jdbc:mysql://192.168.33.10/data',
61             dbtable='users',
62             user='admin',
63             password='admin').mode('append').save()
64
65
66         rdd_collected = rdd.map(lambda tweet: tweet["entities"]["hashtags"]).collect()
67
68         if len(rdd_collected) >= 1:
69             hashtags_list = getHashtags(rdd_collected)
70             rdd_hashtags = sc.parallelize(hashtags_list)
71             if rdd_hashtags.isEmpty() == False:
72                 hashtagsDF = rdd_hashtags.map(lambda hashtag: Row(hashtag=hashtag)).toDF()
73                 hashtagsDF = hashtagsDF.withColumn("hashtag", hashtagsDF["hashtag"].cast(StringType()))
74                 hashtagsDF.printSchema()
75                 hashtagsDF.write.format('jdbc').options(
76                     url='jdbc:mysql://192.168.33.10/data',
77                     dbtable='hashtags',
78                     user='admin',
79                     password='admin').mode('append').save()
80
81
82 spark = SparkSession.builder \
83     .master("local[2]") \
84     .appName("data") \
85     .getOrCreate()
86
87
88
89
90 sc = spark.sparkContext
91 sc.setLogLevel("ERROR")
92 ssc = StreamingContext(sc, 10)
93
94 directKafkaStream = KafkaUtils.createDirectStream(ssc, ["tweets"], {"metadata.broker.list": "192.168.33.13:9092"})
95 rdd = directKafkaStream.map(lambda tweet: json.loads(tweet[1]))
96 rdd.foreachRDD(process)
97
98
99
100 ssc.start()
101 ssc.awaitTermination()

```

ensuite, sur la machine spark, nous avons installé spark-streaming-kafka : [wget https://repo1.maven.org/maven2/org/apache/spark/spark-streaming-kafka-0-8-assembly_2.11/2.4.7/spark-streaming-kafka-0-8-assembly_2.11-2.4.7.jar](https://repo1.maven.org/maven2/org/apache/spark/spark-streaming-kafka-0-8-assembly_2.11/2.4.7/spark-streaming-kafka-0-8-assembly_2.11-2.4.7.jar)

Puis déplacé ce jar dans les jars de spark `sudo mv spark-streaming-kafka-0-8-assembly_2.11-2.4.7.jar /usr/local/spark/jars`

Récupération des données sur le consumer spark en lançant le job avec la commande spark-submit

```
vagrant@vagrant:~$ /usr/local/spark/bin/spark-submit spark-consumer-tweets.py
```

Log du consumer spark : J'ai affiché les hashtags et les schémas des Dataframes pour vérifier que le job fonctionnait correctement.

```
===== 2022-05-14 10:25:30 =====
PythonRDD[2247] at RDD at PythonRDD.scala:53
root
|-- location: string (nullable = true)
|-- username: string (nullable = true)

[{'text': 'David', 'indices': [132, 137]}]
[]
[{'text': 'Davido', 'indices': [91, 98]}]
[]
root
|-- hashtag: string (nullable = true)
```

J'ai ensuite vérifié que les données s'insèrent bien dans la base de données.
Sur la machine MySQL / Superset, en accédant à la base de données avec les commandes

```
"sudo mysql"
"use data;"
"select * from users;"
```

```
1083 rows in set (0.01 sec)
```

et pour les hashtags

```
"select * from hashtags;"
```

```
401 rows in set (0.00 sec)
```

Nous allons maintenant connecter notre base de données sur superset pour pouvoir effectuer des actions sur les données.

On ajoute donc notre base de données :

Edit database

MYSQL

MySQL_data

BASIC

ADVANCED

HOST *

127.0.0.1

PORT *

3306

DATABASE NAME *

data

Copy the name of the database you are trying to connect to.

USERNAME *

admin

PASSWORD

DISPLAY NAME *

MySQL_data

Pick a nickname for this database to display as in Superset.

ADDITIONAL PARAMETERS

e.g. param1=value1¶m2=value2

Add additional custom parameters

☐ SSL

CLOSE

FINISH

Création du dataset à partir de la table de notre base de données

Superset

DashboardsChartsSQL LabData

Settings

Data

Databases

Datasets

Saved queries

Query history

BULK SELECT

DATASET

OWNER

SELECT or type a value

DATABASE

SELECT or type a value

SCHEMA

SELECT or type a value

Name

hashtags

demo

channels

video_game_sales

covid_vaccines

users_channels

messages

users

exported_stats

threads

messages_channels

cleaned_sales_data

channel_members

modified by

Owners

Actions

ADMIN admin

ADMIN admin

ADD dataset

DATABASE

mysqlMySQL_data

SCHEMA

data

SEE TABLE SCHEMA

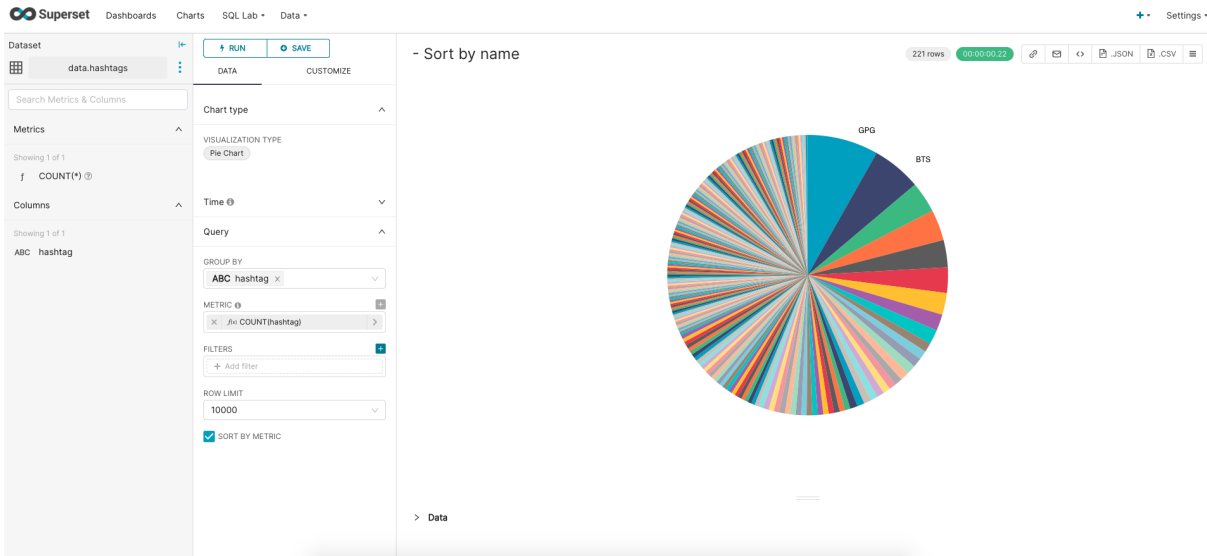
hashtags

CANCEL

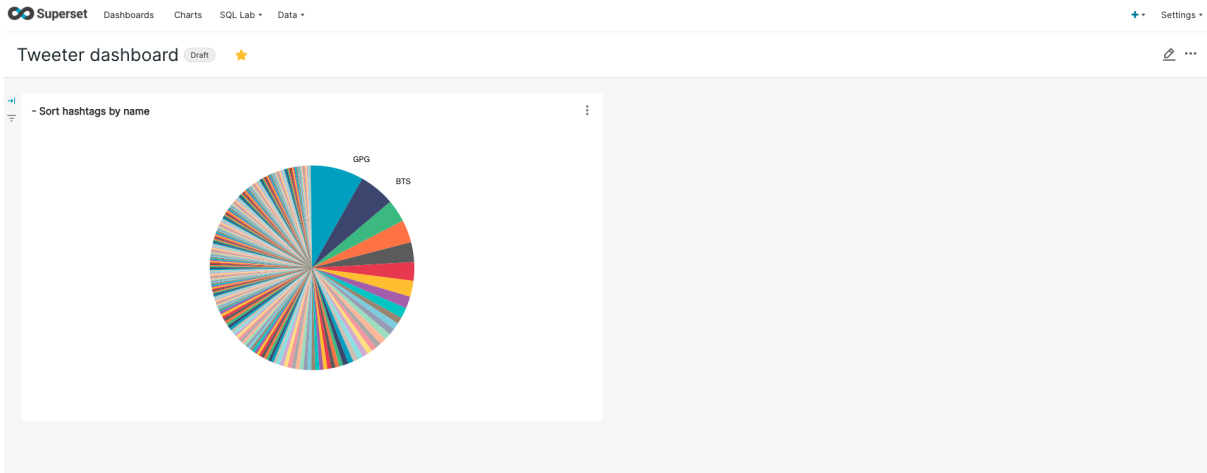
ADD

Création d'un chart en fonction du count des hashtag présents dans la base de données

Loche Rémy ESI5E



Ajout du chart à un nouveau dashboard "Tweeter dashboard"



J'ai donc créé différents graph représentant sous différentes formes les utilisateurs et le nombre d'amis ainsi que les hashtags les plus utilisés au moment où j'ai réalisé l'exercice.

