



Big Data Management

Project 2

Καραπιπεράκης Εμμανουήλ

A.M: 2022201800075

dit18075@go.uop.gr

Καπέλος Γεώργιος

A.M: 2022201800066

dit18066@go.uop.gr

Περιεχόμενα

Προ επεξεργασία Δεδομένων.....	<u>3</u>
Ερώτημα 1(Εισαγωγή δεδομένων στο MongoDB).....	<u>4</u>
Ερώτημα 2.1.....	<u>6</u>
Ερώτημα 2.2.....	<u>8</u>
Ερώτημα 2.3.....	<u>12</u>
Ερώτημα 2.4.....	<u>16</u>
Ερώτημα 2.5.....	<u>19</u>
bonus.....	<u>22</u>

Προ επεξεργασία Δεδομένων

Για την εργασία η ομάδα μας επέλεξε να χρησιμοποιήσει το MongoDB Compass, το GUI για το MongoDB δηλαδή, προκειμένου να διατυπώσει τα ερωτήματα της εκφώνησης. Ανοίγοντας το, συνδεθήκαμε στο localhost και δημιουργήσαμε ένα καινούργιο Database με όνομα **Project2** για την αποθήκευση και τη διαχείριση της πληροφορίας μας. Για την εισαγωγή των δεδομένων είδαμε πως έπρεπε να φτιάξουμε ένα καινούργιο collection και να προσθέσουμε δεδομένα μέσω ενός αρχείου κατάληξης .csv ή .json. Επομένως πρώτο μας βήμα ήταν η προσθήκη του αρχείου Gbvideos.csv το οποίο κατεβάσαμε σύμφωνα με το link της εκφώνησης. Αφού το προσθέσαμε παρατηρήσαμε πως ενδεχομένως να υπήρχαν ζητήματα με το field tags. Για αυτό αποφασίσαμε να φτιάξουμε ένα script αρχείο σε java το οποίο θα λάμβανε ως είσοδο το αρχείο Gbvideos.csv ή το USvideos.csv σε επόμενο ερώτημα και θα μετατρέπαμε το field tag σε array και θα μετατρέπαμε το αρχείο μας από .csv σε .json αλλάζοντας το format του.

Επομένως στο φάκελο **input** βρίσκεται το script μας, το οποίο δέχεται ως είσοδο το Gbvideos.csv και βγάζει ως έξοδο ένα αρχείο με όνομα GBDData.json το οποίο στη συνέχεια θα προσθέσουμε στο MongoDB. Για το USvideos.csv το αντίστοιχο script βρίσκεται στο φάκελο **usa Data**

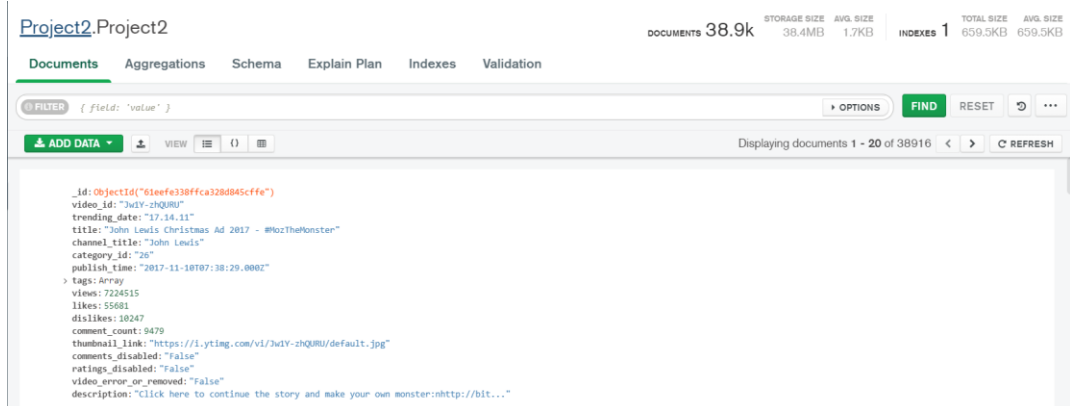
Επεξήγηση κώδικα:

Αρχικά υποθέσαμε πως στο αρχείο εισόδου θα υπάρχουν unicode χαρακτήρες μιας που πρόκειται για δεδομένα αντλούμενα από το youtube και διάφορα πεδία όπως το description ή ο τίτλος ενδεχομένως να περιέχουν ειδικούς χαρακτήρες ή σύμβολα. Οπότε δημιουργήσαμε έναν UTF-8 reader ο οποίος ήταν υπεύθυνος να διαβάζει τα δεδομένα μας από το .csv αρχείο και έναν UTF-8 writer ο οποίος ήταν υπεύθυνος για τη δημιουργία του json αρχείου. Στη συνέχεια διαβάζαμε γραμμή προς γραμμή το αρχείο εισόδου μας και αποθηκεύαμε σε έναν πίνακα string μεγέθους 16 το περιεχόμενο του κάθε πεδίου. Ειδική μεταχείριση κάναμε στα πεδία που αφορούσαν numeric τιμές, καθώς φροντίσαμε να μην προσθέσουμε τα “ ” γύρω από τη λέξη, έτσι ώστε το MongoDB να τις αντιμετωπίσει κατάλληλα. Επίσης αναφορικά με τη στήλη όπου βρισκόταν το πεδίο tags, αντικαταστήσαμε τους χαρακτήρες “|” με “,” και βάλαμε αγκύλες ([]) στην αρχή και στο τέλος του πεδίου έτσι ώστε το MongoDB να αναγνωρίσει αυτό το πεδίο ως πίνακα.

Οι δυσκολίες που αντιμετωπίσαμε στο κομμάτι του κώδικα ήταν ορισμένες περιπτώσεις όπου το περιεχόμενο του τελευταίου πεδίου(description) συνεχιζόταν και σε επόμενες γραμμές, οπότε χρειάστηκε ειδική μεταχείριση.

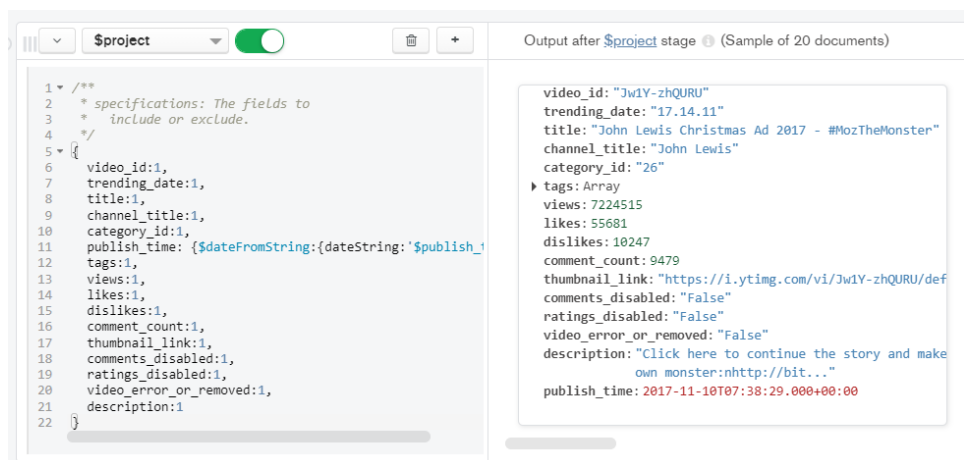
Ερώτημα 1

Αφού κατασκευάσαμε το αρχείο json που προαναφέρθηκε, έπρεπε να το εισάγουμε στη βάση μας. Οπότε συνδεθήκαμε ξανά, πήγαμε στη βάση Project2 και δημιουργήσαμε ένα collection με όνομα Project2 στο οποίο κάναμε Add Data πατώντας το κατάλληλο κουμπί το αρχείο **GBData.json**.



Όπως φαίνεται το field tags αποτελεί πλέον πίνακα και κάνοντας κλικ πάνω του μπορούμε να δούμε το περιεχόμενό του. Στη συνέχεια παρατηρώντας τα δεδομένα μας συνειδητοποιήσαμε πως ορισμένες εγγραφές επαναλαμβάνονται. Υποθέσαμε πως αυτό οφείλεται σε διάφορες εκδόσεις του εκάστοτε video και έτσι καταλήξαμε στο συμπέρασμα πως πρέπει να κρατήσουμε μόνο μια έκδοση του video, την πιο πρόσφατη. Λαμβάνοντας υπόψιν πως σε ένα βίντεο που έχει διάφορες εκδόσεις μέσα σε ένα χρονικό διάστημα, τιμές όπως τα likes, τα views, τα dislikes, ο αριθμός των comments τείνουν να αυξάνονται ανάλογα μέχρι την πιο πρόσφατη έκδοση και έτσι ενδιαφερόμαστε μόνο για τη μεγαλύτερη τιμή που λαμβάνουν. Επομένως, πήγαμε στο aggregation της συλλογής μας και κάναμε χρήση του **\$group** προκειμένου να ομαδοποιήσουμε σύμφωνα με το **video_id** κρατώντας κάθε φορά τη max τιμή των υπόλοιπων πεδίων για τους λόγους που προαναφέραμε.

Επιπλέον παρατηρήσαμε πως το πεδίο **publish_time** το οποίο θα έπρεπε να χρησιμοποιήσουμε στο ερώτημα 2.5 ήταν σε μορφή string. Έτσι στην προηγούμενη διαδικασία προσθέσαμε και την μετατροπή του field publish_time από string σε date με χρήση του **\$dateFromString** και με χρήση του \$out δημιουργήσαμε ένα καινούργιο collection με όνομα GBData όπως φαίνεται παρακάτω



\$group

Output after \$group stage ⓘ (Sample of 20 documents)

```
1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$video_id",
7   trending_date: {$max: "$trending_date"},
8   title: {$max: "$title"},
9   channel_title: {$max: "$channel_title"},
10  category_id: {$max: "$category_id"},
11  publish_time: {$max: "$publish_time"},
12  tags: {$max: "$tags"},
13  views: {$max: "$views"},
14  likes: {$max: "$likes"},
15  dislikes: {$max: "$dislikes"},
16  comment_count: {$max: "$comment_count"},
17  thumbnail_link: {$max: "$thumbnail_link"},
18  comments_disabled: {$max: "$comments_disabled"},
19  ratings_disabled: {$max: "$ratings_disabled"},
20  video_error_or_removed: {$max: "$video_error_or_removed"},
21  description: {$max: "$description"}
22 }
```

```
_id: "KTbOAMemfI"
trending_date: "18.13.02"
title: "Sean 'Love' Combs Makes a Fashionably Late Entrance"
channel_title: "TheEllenShow"
category_id: "24"
publish_time: 2018-02-08T14:00:11.000+00:00
tags: Array
views: 425784
likes: 5351
dislikes: 179
comment_count: 295
thumbnail_link: "https://i.ytimg.com/vi/KTbOAMemfI/default.jpg"
comments_disabled: "False"
ratings_disabled: "False"
video_error_or_removed: "False"
```

```
_id: "pVgx8pYpwf"
trending_date: "18.13.02"
title: "Camila Cabello - Consequence of Love (Official Music Video)"
channel_title: "CamilaCabelloVEVO"
category_id: "10"
publish_time: 2018-02-08T14:00:11.000+00:00
tags: Array
views: 254444
likes: 7119
dislikes: 195
comment_count: 195
thumbnail_link: "https://i.ytimg.com/vi/pVgx8pYpwf/default.jpg"
comments_disabled: "False"
ratings_disabled: "False"
video_error_or_removed: "False"
description: "Camila Cabello - Consequence of Love (Official Music Video) http://smarturl.it/ConsequenceOfLove"
```

\$out

Documents will be saved to the collection: 'GBData'

The \$out operator will cause the pipeline to persist the results to the specified location. If the collection already exists, its contents will be replaced. Please confirm to proceed.

SAVE DOCUMENTS

Με την ίδια λογική προσθέσαμε και το αρχείο UsaData.json σε ένα collection με όνομα **USADData** και εφαρμόζοντας τα παραπάνω βήματα μεταφέραμε τα δεδομένα μας απαλλαγμένα από τις διπλοεγγραφές στη συλλογή **UsaDataUPDATED**
Πλέον στη συλλογή GBData έχουμε 3272 εγγραφές από τις 38916 που είχαμε προηγουμένως και το πεδίο publish_time αποτελεί date value φέροντας και το αντίστοιχο format

Project2.GBData

DOCUMENTS 3.3k STORAGE SIZE 3.1MB AVG. SIZE 1.7KB INDEXES 1 TOTAL SIZE 86.0KB AVG. SIZE 86.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

OPTIONS FIND RESET ↺ ...

ADD DATA VIEW {}

Displaying documents 1 - 20 of 3272 < > C REFRESH

```
_id: "HKIIgYFhQIE"
trending_date: "17.19.12"
title: "G-Eazy - Sober (Audio) ft. Charlie Puth"
channel_title: "GEazyMusicVEVO"
category_id: "10"
publish_time: 2017-12-08T05:00:01.000+00:00
tags: Array
views: 3905540
likes: 130790
dislikes: 1197
comment_count: 4230
thumbnail_link: "https://i.ytimg.com/vi/HKIIgYFhQIE/default.jpg"
comments_disabled: "False"
ratings_disabled: "False"
video_error_or_removed: "False"
description: "New Album 'The Beautiful & Damned' Available Everywhere http://smartur..."
```

Ερώτημα 2.1

Για το ερώτημα 2.1 αρχικά πήγαμε στο aggregation της συλλογής GBData από όπου και τρέξαμε τα ερωτήματά μας ως εξής:

Αρχικά έπρεπε να κάνουμε ένα ταίριασμα σύμφωνα με τη λέξη “Saturday Night Lives” στο πεδίο channel_title το οποίο πραγματοποιήθηκε με χρήση της εντολής **\$match**. Στη συνέχεια μέσω της εντολής **\$project** κρατήσαμε μόνο τα πεδία που μας ενδιέφεραν σύμφωνα με την εκφώνηση. Τέλος μέσω του **\$out** εξαγάγαμε τα αποτελέσματα σε ένα καινούργιο collection results2.1 για περαιτέρω μελέτη.

The image displays three screenshots of the MongoDB aggregation pipeline interface, showing the stages \$match, \$project, and \$sort.

\$match stage: The query is `{ channel_title: "Saturday Night Live" }`. The output shows a sample of 20 documents, including one with `_id: "h5Xf6ozQEwo"`, `trending_date: "17.19.11"`, `title: "Leslie Jones Had a Dream Just Like This - SNL"`, `channel_title: "Saturday Night Live"`, `category_id: "24"`, `publish_time: 2017-11-17T16:35:09.000+00:00`, `tags: Array`, and `views: 281064`.

\$project stage: The specifications are `{ views:1, title:1, likes:1, dislikes:1 }`. The output shows a sample of 20 documents, including one with `_id: "h5Xf6ozQEwo"`, `title: "Leslie Jones Had a Dream Just Like This - SNL"`, `views: 281064`, `likes: 5230`, and `dislikes: 145`.

\$sort stage: The sort criteria are `{ views: -1 }`. The output shows a sample of 20 documents, including one with `_id: "Qcj15vHJTtk"`, `title: "Royal Wedding - SNL"`, `views: 8607264`, `likes: 66559`, and `dislikes: 14179`.

Ακολουθούν ενδεικτικά οι 20 πρώτες εγγραφές των αποτελεσμάτων σύμφωνα με την εκφώνηση:

```
_id,dislikes,likes,title,views
Qcj15vHJTtk,14179,66559,Royal Wedding - SNL,8607264
q2sI-T69sjs,5101,110621,A Kanye Place - SNL,5547578
QS8bma7LRX4,3195,79865,Natalie's Rap 2 - SNL,5156609
lpkRFHSpvGI,6752,55076,George W. Bush Returns Cold Open - SNL,5147621
1l26UFQ06eQ,8997,60641>Welcome to Hell - SNL,4649310
kXIF2FQpgjM,4666,40718,Meet the Parents Cold Open - SNL,4548677
AiyZ92_JZxA,3658,31716,Morning Joe Michael Wolff Cold Open - SNL,4162540
6IL4sMC_XBE,3603,31791,Presidential Address Cold Open - SNL,3763816
ZJkc_C5-Cd8,10550,35137,What Even Matters Anymore - SNL,3362428
A8HLnDP6uRM,4642,28777,Visit with Santa Cold Open - SNL,2935898
hOulcmOHCIQ,5428,27542,White House Tree Trimming Cold Open - SNL,2701442
_H1vcldAsP0,3044,36494,"Eminem: Walk on Water, Stan, Love the Way You Lie (ft. Skylar
```

Grey) (Live) - SNL",2200854
LAQRtjvtQRo,761,23694,"Weekend Update: Stefon on St. Patrick's Day - SNL",2189861
j7WlcSwe8Ws,1741,18641,"Weekend Update on Donald Trump's Asia Trip - SNL",1907912
54BVvP7zvDQ,1483,32619,"Queer Eye's Tan France Takes Pete Davidson Shopping -
SNL",1856381
3s1rvMFUweQ,3214,35741,Taylor Swift: ...Ready for It? (Live) - SNL,1776127
pkJtiVCMXVg,857,16328,Family Feud: Oscars Edition - SNL,1352472
RUCXD3_wW2w,2437,9870,New Hulu Show - SNL,1250529
ibI5-k5sin8,871,10478,Cut for Time: New Year's Kiss - SNL,830143
12HlSkv5T-A,259,9515,"SNL Host Donald Glover Is Not Here for Beck Bennett's
Tribute",411490

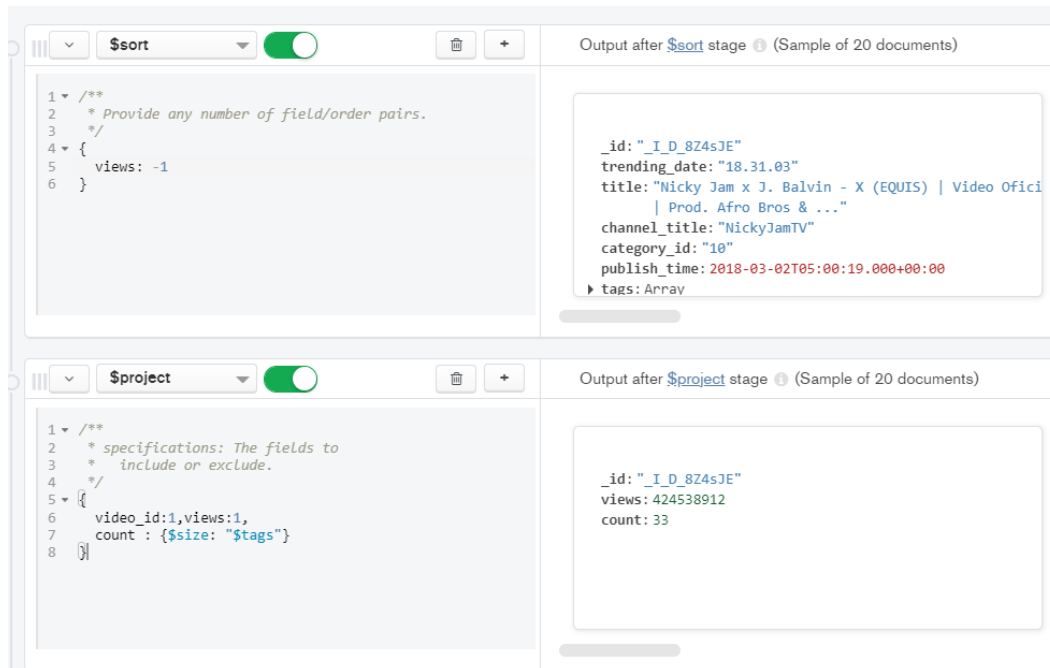
Τα πλήρη αποτελέσματα(results2.1.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.1

Συμπεράσματα

Παρατηρώντας τα αποτελέσματα μας, καταλήγουμε στο συμπέρασμα πως πρόκειται για ένα δημοφιλές κανάλι που συγκεντρώνει υψηλά ποσοστά τηλεθέασης στα βίντεο του. Διαβάζοντας τους τίτλους, καταλαβαίνουμε πως πρόκειται για ένα κανάλι με τοποθεσία κατά πάσα πιθανότητα στις Η.Π.Α, το οποίο φαίνεται να ασχολείται με την εγχώρια show biz και όχι μόνο. Οι τίτλοι συχνά φέρουν ονόματα διάσημων προσώπων αλλά και λιγότερο γνωστά, ίσως καλεσμένων της εκπομπής, οι οποίοι φαίνεται να είναι πρόσωπα, σχετιζόμενα με την αμερικάνικη επικαιρότητα, όχι τόσο οικεία δηλαδή στον υπόλοιπο κόσμο. Στα περισσότερα βίντεο η απόκλιση των likes με τα dislikes είναι αρκετά μεγάλη, γεγονός που σημαίνει πως το περιεχόμενο του καναλιού είναι ποιοτικό και δεν υπάρχουν αρκετοί δυσανεστημένοι χρήστες.

Ερώτημα 2.2

Για το ερώτημα 2.2 όπως και πριν, πήγαμε στο aggregation για το collection GBData. 1^ο μας βήμα ήταν η ταξινόμηση σύμφωνα με τις προβολές σε φθίνουσα σειρά όπως μας ζητάει η εκφώνηση. Άρα κάναμε χρήση της εντολής **\$sort**. Ακολούθως με χρήση της εντολής **\$project**, κρατήσαμε μόνο όσα πεδία μας απασχολούν στο ερώτημά μας, δηλαδή το video_id, τα views και το μέγεθος του κάθε πίνακα tags. Το μέγεθος του πίνακα, ουσιαστικά μας δείχνει πόσες ετικέτες χρησιμοποιεί το κάθε βίντεο, το οποίο είναι και το ζητούμενο. Τέλος μέσω του **\$out** εξαγάγαμε τα αποτελέσματα σε ένα καινούργιο collection results2.2 για περαιτέρω μελέτη

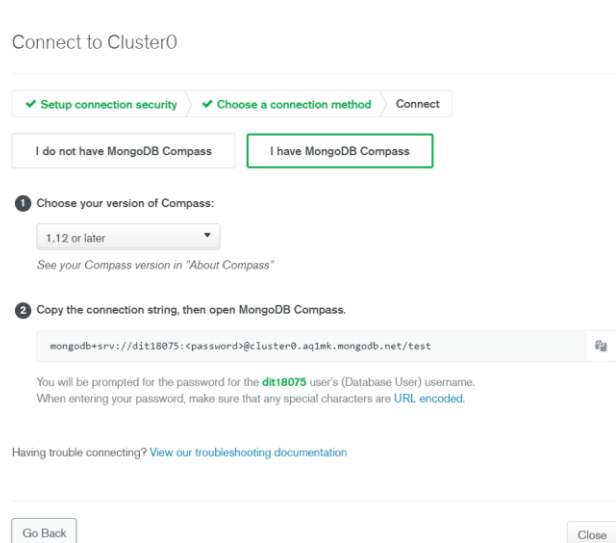


Ακολουθούν ενδεικτικά οι 20 πρώτες εγγραφές των αποτελεσμάτων σύμφωνα με την εκφώνηση:

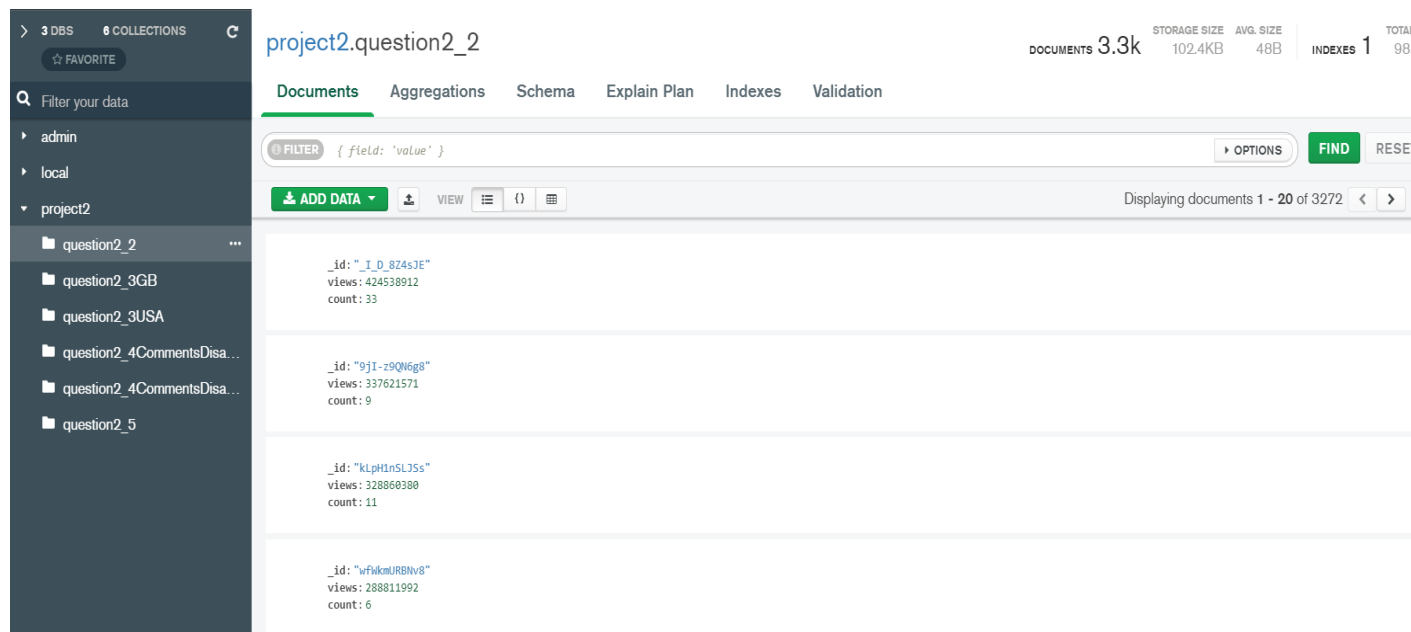
```
_id,count,views
_I_D_8Z4sJE,33,424538912
9jI-z9QN6g8,9,337621571
kLpHlnSLJSs,11,328860380
wfWkmURBNv8,6,288811992
VYOjWnS4cMY,4,259721696
xpVfcZ0ZcFM,6,258164991
ffxKSjUwKdU,10,208876887
zEf423kYfqk,22,200862743
FlsCjmMhFmw,37,169884583
sGIm0-dQd8M,34,167456025
TyHvyGVs42U,11,143408235
2Vv-BfVoq4g,10,138578860
M4ZoCHID9GI,6,138535053
Ck4xHocysLw,6,137081637
7C2z4GqgS5E,9,123010920
tCXGJQYZ9JA,7,117270304
U9BwWKXjVaI,33,106147032
au2n7VVGv_c,7,105629911
6ZfuNTqbHE8,11,100672931
fGqdIPer-ms,15,100159686
```


Τα πλήρη αποτελέσματα(results2.2.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.2

Επιπλέον για το συγκεκριμένο ερώτημα σύμφωνα με την εκφώνηση έπρεπε να κατασκευάσουμε και μια γραφική απεικόνιση. Έτσι οδηγηθήκαμε στο Atlas, το οποίο είναι παρεμφερές με το MongoDB και μπορούμε να πάρουμε δεδομένα από μια συλλογή του Compass και να δημιουργήσουμε γραφικές αναπαραστάσεις με σκοπό τη περαιτέρω μελέτη των δεδομένων μας. Έτσι πήγαμε στο <https://www.mongodb.com/atlas/database> φτιάξαμε λογαριασμό και συνδεθήκαμε σε μια βάση δεδομένων όπως φαίνεται στη συνέχεια.



Σε αυτή τη βάση δεδομένων εισαγάγαμε τα δεδομένα που εξαγάγαμε από κάθε ερώτημα, έτσι ώστε να μπορούμε να τα χρησιμοποιήσουμε από το atlas



Επομένως τώρα στο Atlas, μπορούμε να πάμε στο **charts** → **Data Sources** → **Add Data Source** και να προσθέσουμε δεδομένα για να τα αναπαραστήσουμε με διαγράμματα.

Data Sources						<div> <div>I'm an Owner</div> <div>Find Data Source...</div> <div>Add Data Source</div> </div>	
Name	Source	Added	Pipeline	Permission			
project2.question2_2	project2.question2_2 Cluster0	a few seconds ago	<div>Add Pipeline</div>	<div>ACCESS</div>	<div>...</div>		
project2.question2_4CommentsDisabled_TRUE	project2.question2_4CommentsDisabled_TRUE Cluster0	a few seconds ago	<div>Add Pipeline</div>	<div>ACCESS</div>	<div>...</div>		
project2.question2_4CommentsDisabled_FALSE	project2.question2_4CommentsDisabled_FALSE Cluster0	a few seconds ago	<div>Add Pipeline</div>	<div>ACCESS</div>	<div>...</div>		
project2.question2_3GB	project2.question2_3GB Cluster0	a few seconds ago	<div>Add Pipeline</div>	<div>ACCESS</div>	<div>...</div>		
project2.question2_3USA	project2.question2_3USA Cluster0	a few seconds ago	<div>Add Pipeline</div>	<div>ACCESS</div>	<div>...</div>		
project2.question2_5	project2.question2_5 Cluster0	a few seconds ago	<div>Add Pipeline</div>	<div>ACCESS</div>	<div>...</div>		

Η τελική μορφή του Data Sources θα είναι κάπως έτσι για όλα τα ερωτήματα που απαιτούν γραφική αναπαράσταση. Επομένως τώρα επιστρέφουμε πίσω στο **Charts → Dashboards → Add Dashboard → Add chart** και προσθέτουμε ένα chart με όνομα question2_2

Πάνω αριστερά επιλέγουμε το Data source που προσθέσαμε προηγουμένως, στο chart type το είδος chart που μας ενδιαφέρει(στο συγκεκριμένο ερώτημα scatter plot) και κάνουμε drag & drop τα fields μας από αριστερά στους άξονες X και Y

Data Source

project2.question2_2

Sample

Query

filter

A_id

count

views

Chart Type

Grid

Heatmap

Scatter

Encode

Filter

Customize

X Axis

views

Y Axis

count

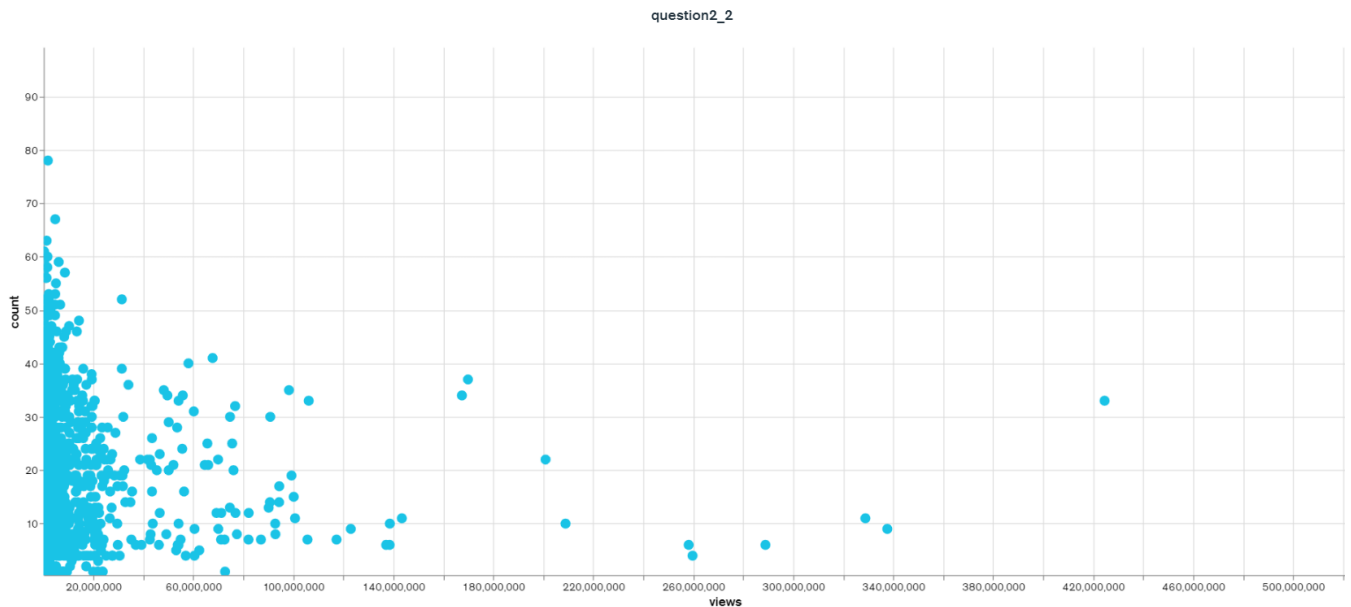
Size

+ aggregation

Color

+ category

Ακολουθώντας τα παραπάνω βήματα παίρνουμε το εξής αποτέλεσμα:



Συμπεράσματα

Παρατηρώντας τα αποτελέσματα μας, καταλήγουμε στο συμπέρασμα πως στα περισσότερα βίντεο οι προβολές είναι ανεξάρτητες των tags καθώς υπάρχουν βίντεο με αρκετά views ενώ ταυτόχρονα έχουν περίπου 10 ετικέτες ή και λιγότερες. Επίσης υπάρχουν βίντεο με 50+ ετικέτες που έχουν αισθητά λιγότερες προβολές σε σύγκριση με άλλα βίντεο που χρησιμοποιούν λιγότερες. Εξαιρέση αποτελούν ορισμένα βίντεο όπως αυτό που διακρίνεται δεξιά με περίπου 420.000.000 προβολές το οποίο ο χρησιμοποιεί 33 ετικέτες.

Ερώτημα 2.3

Για το ερώτημα 2.3 σκεφτήκαμε πως θα έπρεπε με κάποιον τρόπο να απομονώσουμε όλες τις ετικέτες που υπάρχουν και μετά να μετρήσουμε πόσες φορές εμφανίζεται η κάθε ετικέτα μέσα στα video. Τέλος θα ταξινομούσαμε το πλήθος των ετικετών σε φθίνουσα σειρά. Σύμφωνα με την εκφώνηση, θα πρέπει να εκτελέσουμε το ερώτημα που περιγράψαμε για 2 περιοχές, τη Μεγάλη Βρετανία και τις Η.Π.Α. Τα ερωτήματα θα είναι ίδια και στις 2 περιπτώσεις, μόνο τα συμπεράσματα θα διαφέρουν όπως είναι φυσικό άλλωστε. Οπότε για τη Μεγάλη Βρετανία, πάμε στο aggregation του GBData και γράφουμε τα ερωτήματα μας ως εξής: Πρώτα μέσω της εντολής **\$project** κρατάμε μόνο το πεδίο tags από το οποίο θα εξάγουμε το περιεχόμενο του πίνακα. Για την επίτευξη αυτού του σκοπού κάναμε χρήση της εντολής **\$unwind** η οποία σύμφωνα με το documentation του MongoDB βγάζει ως έξοδο ένα καινούργιο document για κάθε στοιχείο του επιλεγμένου πίνακα. Στο τρίτο μας βήμα κάναμε ένα group by σύμφωνα με την εντολή **\$group** παίρνοντας ως κλειδί την κάθε ετικέτα και υπολογίζοντας το πλήθος εμφάνισής της με χρήση της εντολής \$sum μέσα στο group by. Τελευταίο μας βήμα ήταν η ταξινόμηση του πλήθους που υπολογίσαμε στο προηγούμενο βήμα σε φθίνουσα σειρά. Οπότε χρησιμοποιήσαμε την εντολή **\$sort** για την επίτευξη του παραπάνω στόχου. Τέλος μέσω του **\$out** εξαγάγαμε τα αποτελέσματα σε ένα καινούργιο collection results2.3GB για περαιτέρω μελέτη και επεξεργασία

The image displays four screenshots of the MongoDB Aggregation Builder interface, showing the stages of an aggregation pipeline:

- \$project stage:** The query is `{ "tags": 1 }`. The output shows a document with `_id: "HKIIgYfhQIE"` and `tags: Array`.
- \$unwind stage:** The query is `{ "path": "$tags" }`. The output shows a document with `_id: "HKIIgYfhQIE"` and `tags: "BPG/RVG/RCA Records"`.
- \$group stage:** The query is `{ "_id": "$tags", "count": { "$sum": 1 } }`. The output shows a document with `_id: "dinosaur"` and `count: 2`.
- \$sort stage:** The query is `{ "count": -1 }`. The output shows a document with `_id: "funny"` and `count: 319`.

Ακολουθούν ενδεικτικά οι 20 πρώτες εγγραφές των αποτελεσμάτων από τις συνολικές 31680 σύμφωνα με την εκφώνηση:

_id	count
funny	319
comedy	263
music	206
[none]	178

humor	161
interview	151
comedian	138
video	135
late night	131
celebrities	128
jokes	126
Funny video	126
live	117
2018	112
hollywood	106
celebrity	103
clip	101
show	99
comedic	96
Pop	92

Τα πλήρη αποτελέσματα(results2.3GB.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.3/GB

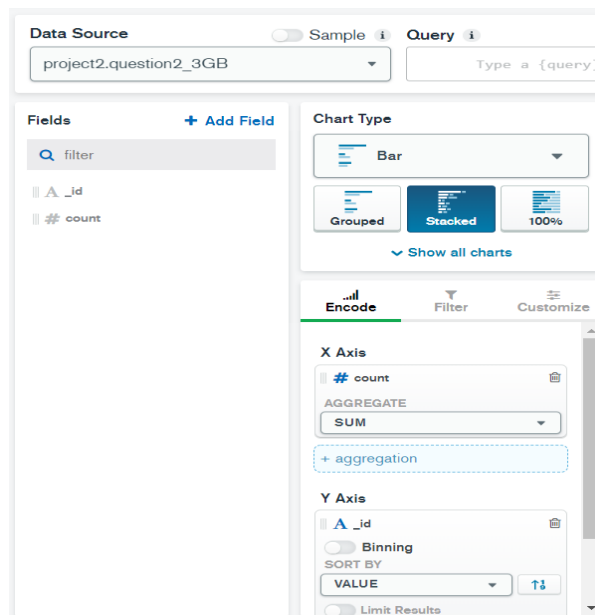
Για την περιοχή USA όπως προαναφέραμε τα ερωτήματα παραμένουν ίδια, απλά αλλάζει ο τύπος που τα εκτελούμε. Από το Aggregation του GBData,θα μεταφερθούμε στο USADataUPDATED και τα αποτελέσματα από τα ερωτήματα θα αποθηκευτούν στο collection results2.3USA. Πλέον τα αποτελέσματα αλλάζουν, ενδεικτικά οι 20 πρώτες εγγραφές από τις συνολικές 56408 θα είναι οι εξής:

_id	count
funny	636
comedy	521
humor	279
[none]	261
comedian	229
music	213
celebrities	213
interview	204
how to	200
2018	199
celebrity	197
funny video	196
video	187
jokes	184
news	178
food	174
science	173

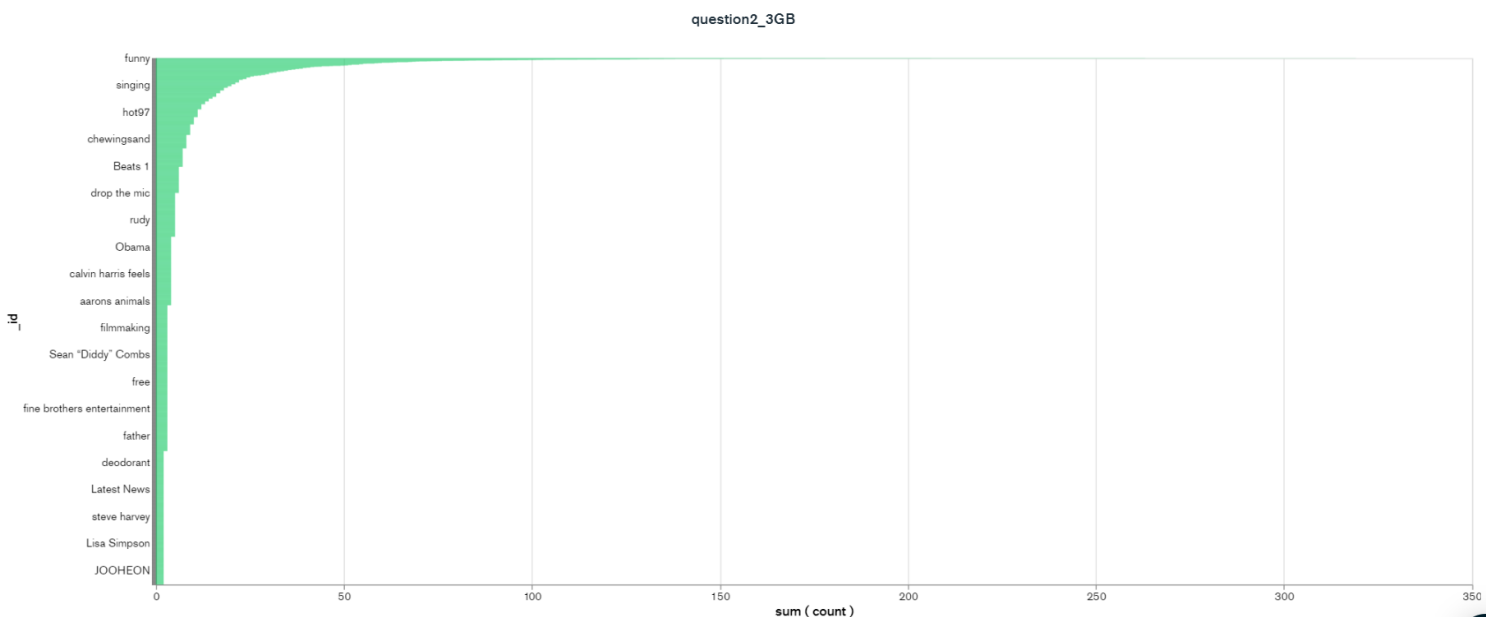
late night	167
NBC	164
live	162

Τα πλήρη αποτελέσματα(results2.3USA.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.3/USA

Τώρα θα οδηγηθούμε ξανά στο Atlas με σκοπό τη γραφική αναπαράσταση των δεδομένων μας. Σε αυτό το ερώτημα χρειαζόμαστε bar chart,οπότε επιλέγω ξανά στο Data Source το αρχείο με τα δεδομένα που με ενδιαφέρει ,στο chart type Bar και stacked.Τέλος κάνω drag & drop το count στον άξονα X και τις ετικέτες στον άξονα Y



Και παίρνω το ακόλουθο αποτέλεσμα:



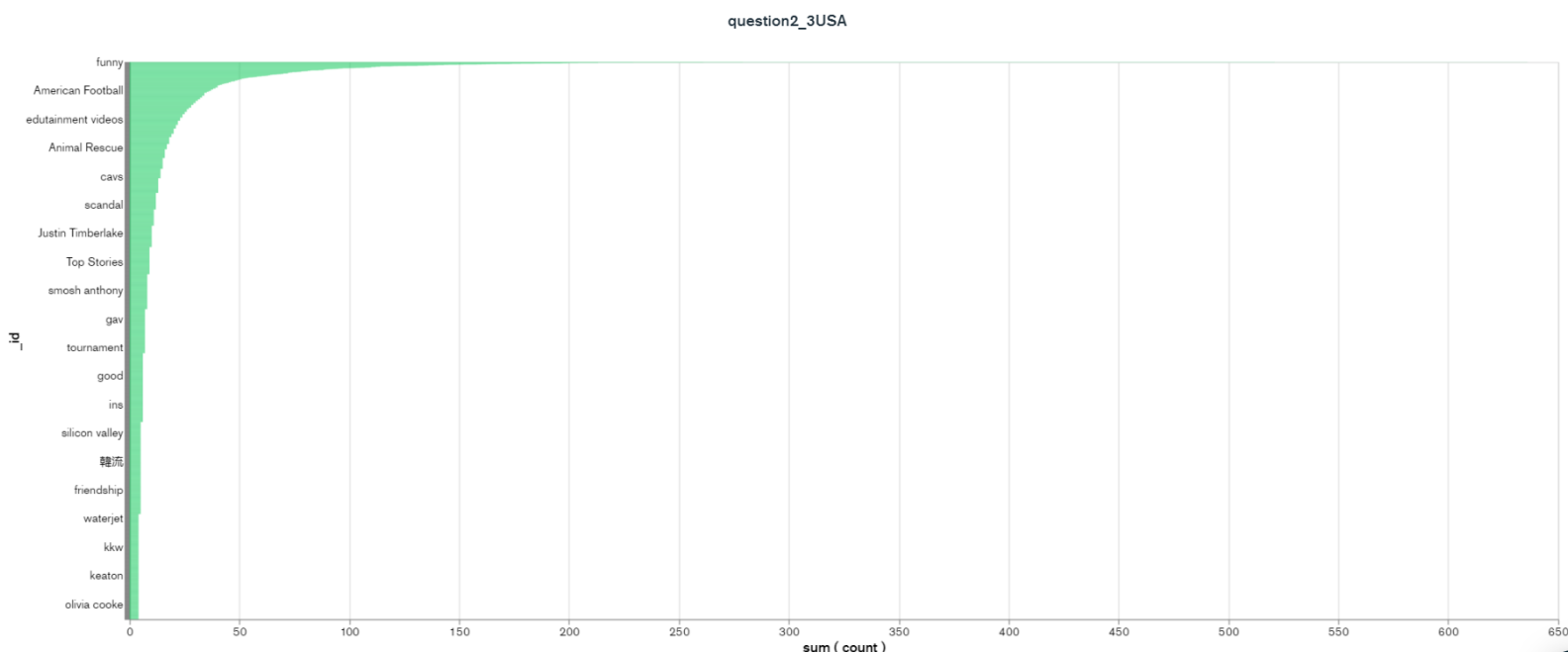
Παρατηρούμε πως δεν φαίνονται οι τιμές όλων των ετικετών, καθώς είναι πάρα πολλές. Όμως πατώντας “**view source documents**” μπορώ να τις δω αναλυτικά:

Chart Data: question2_3GB

The following table of chart data is used to render this visualization.

_id	sum (count)
funny	319
comedy	263
music	206
[none]	178
humor	161
interview	151
comedian	138
video	135

Ακολουθώντας τα ίδια βήματα παίρνω το ακόλουθο bar chart για την περιοχή USA



Συμπεράσματα

Παρατηρώντας τα αποτελέσματα, διακρίνουμε ορισμένα μοτίβα που επαναλαμβάνονται και στις 2 χώρες. Οι πιο δημοφιλείς ετικέτες και στις 2 χώρες είναι σχετικές με τη κωμωδία/χιούμορ. Για παράδειγμα η ετικέτα funny βρίσκεται στην κορυφή και στις 2 χώρες και φανερώνει πως υπάρχουν πολλά αστεία βίντεο, μιας που έχουν απήχηση στο κοινό. Βέβαια όπως είναι λογικό ορισμένες ετικέτες εμφανίζονται μόνο σε μια χώρα, για παράδειγμα στις Η.Π.Α υπάρχουν πολλές ετικέτες για το Αμερικάνικο ποδόσφαιρο μιας που πρόκειται για το εθνικό άθλημα των Αμερικάνων. Όπως και στη Μεγάλη Βρετανία η ετικέτα London εμφανίζεται με συχνότητα 40 ενώ στις Η.Π.Α δεν υπάρχει καν.

Ερώτημα 2.4

Για το ερώτημα 2.4 διακρίναμε 2 περιπτώσεις.

1. Υπολογισμός μέσου όρου views,likes,dislikes με δυνατότητα σχολιασμού(comments disabled: false)
2. Υπολογισμός μέσου όρου views,likes,dislikes χωρίς δυνατότητα σχολιασμού(comments disabled: true)

Επομένως για την 1^η περίπτωση πήγαμε στο aggregation του GBData και αρχικά κρατήσαμε μόνο τα πεδία που χρειαζόμαστε(video_id,likes,dislikes,views,comments_disabled) μέσω της **\$project**. Ακολούθως κάναμε **\$match** μόνο τις περιπτώσεις όπου υπάρχει δυνατότητα σχολιασμού. Στη συνέχεια πραγματοποιήσαμε ομαδοποίηση(**\$group**) με κλειδί null έτσι ώστε να υπολογίσουμε τους μέσους όρους που μας ενδιαφέρουν και τέλος κρατήσαμε μόνο το ακέραιο μέρος των δεδομένων μας. Οπότε εξαγάγαμε το αποτέλεσμα μέσω της **\$out** σε ένα collection **results2.4CommentsDisabled_FALSE**

The screenshot displays the MongoDB Aggregation Framework interface with four stages of an aggregation pipeline. Each stage shows a JSON query on the left and a sample output document on the right.

- Stage 1: \$project**
Query:

```
1 /**  
2  * specifications: The fields to  
3  * include or exclude.  
4  */  
5 {  
6   video_id:1,likes:1,dislikes:1,views:1,comments_disable:  
7 }
```


Output (Sample of 20 documents):

```
{  
  "_id": "HKIIgVFhQIE",  
  "views": 3905540,  
  "likes": 130790,  
  "dislikes": 1197,  
  "comments_disabled": "False"  
}
```
- Stage 2: \$match**
Query:

```
1 /**  
2  * query: The query in MQL.  
3  */  
4 {  
5   comments_disabled: "False"  
6 }
```


Output (Sample of 20 documents):

```
{  
  "_id": "HKIIgVFhQIE",  
  "views": 3905540,  
  "likes": 130790,  
  "dislikes": 1197,  
  "comments_disabled": "False"  
}
```
- Stage 3: \$group**
Query:

```
1 /**  
2  * id: The id of the group.  
3  * fieldName: The first field name.  
4  */  
5 {  
6   id: null,  
7   likesAverage: { $avg: "$likes" },  
8   dislikesAverage: { $avg: "$dislikes" },  
9   viewsAverage: { $avg: "$views" }  
10 }
```


Output (Sample of 1 document):

```
{  
  "_id": null,  
  "likesAverage": 101070.82934410941,  
  "dislikesAverage": 5955.09014609885,  
  "viewsAverage": 4840335.540876593  
}
```
- Stage 4: \$project**
Query:

```
1 /**  
2  * specifications: The fields to  
3  * include or exclude.  
4  */  
5 {  
6   id:0,  
7   likesAverage: { $toInt: "$likesAverage" },  
8   dislikesAverage: { $toInt: "$dislikesAverage" },  
9   viewsAverage: { $toInt: "$viewsAverage" },  
10 }
```


Output (Sample of 1 document):

```
{  
  "likesAverage": 101070,  
  "dislikesAverage": 5955,  
  "viewsAverage": 4840335  
}
```


Για την 2^η περίπτωση χρησιμοποιήσαμε την ίδια λογική απλά αλλάζοντας το \$match που χρησιμοποιήσαμε στο 2^ο ερώτημα προκειμένου να κρατήσουμε μόνο τα βίντεο με απενεργοποιημένα σχόλια. Έτσι αποθηκεύσαμε το αποτέλεσμα στη συλλογή **results2.4CommentsDisabled_TRUE**

Επομένως το περιεχόμενο της συλλογής results2.4CommentsDisabled_FALSE είναι

```
likesAverage: 101070
dislikesAverage: 5955
viewsAverage: 4840335
```

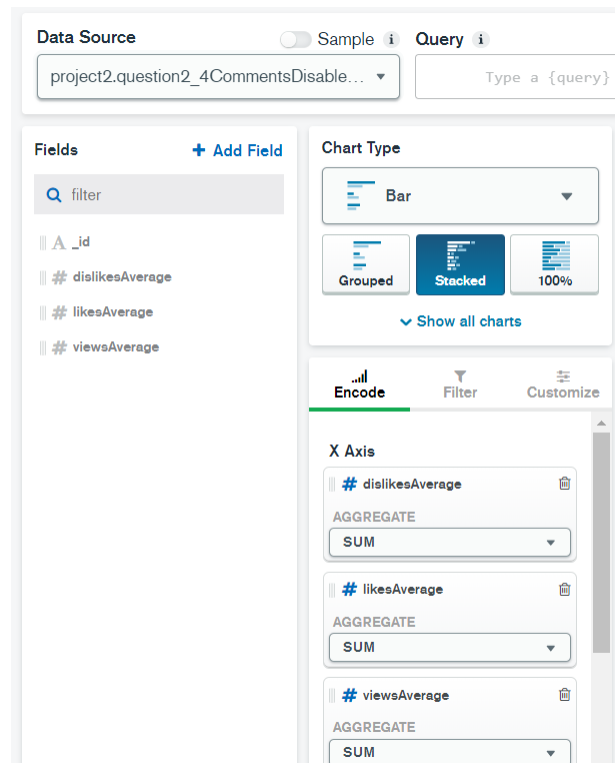
Τα πλήρη αποτελέσματα(results2.4CommentsDisabled_FALSE.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.4/CommentsDisabledFalse

Και το περιεχόμενο της συλλογής results2.4CommentsDisabled_TRUE είναι

```
likesAverage: 32531
dislikesAverage: 4881
viewsAverage: 3306934
```

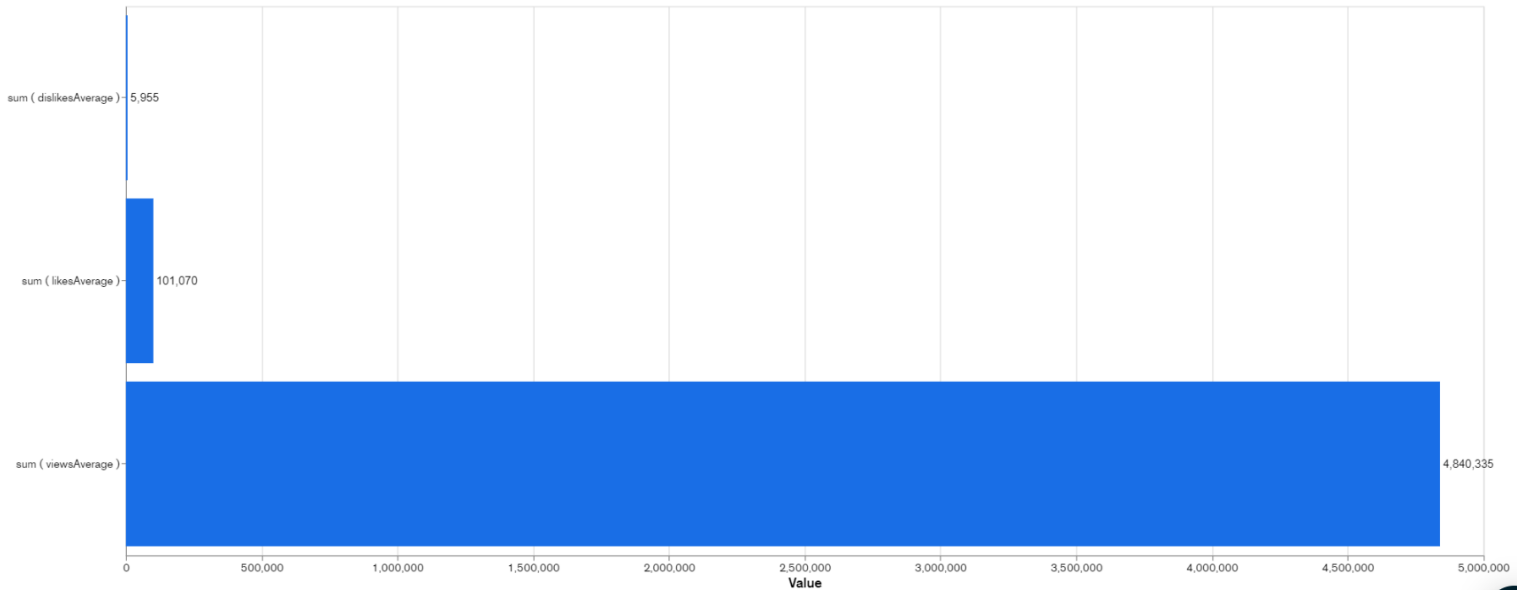
Τα πλήρη αποτελέσματα(results2.4CommentsDisabled_TRUE.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.4/CommentsDisabledTrue

Τελευταίο μας βήμα είναι η εισαγωγή των δεδομένων μας στο atlas και η δημιουργία των bar charts. Οπότε εισαγάγουμε τα δεδομένα στο atlas όπως και στο προηγούμενο ερώτημα και επιλέγουμε το bar chart:

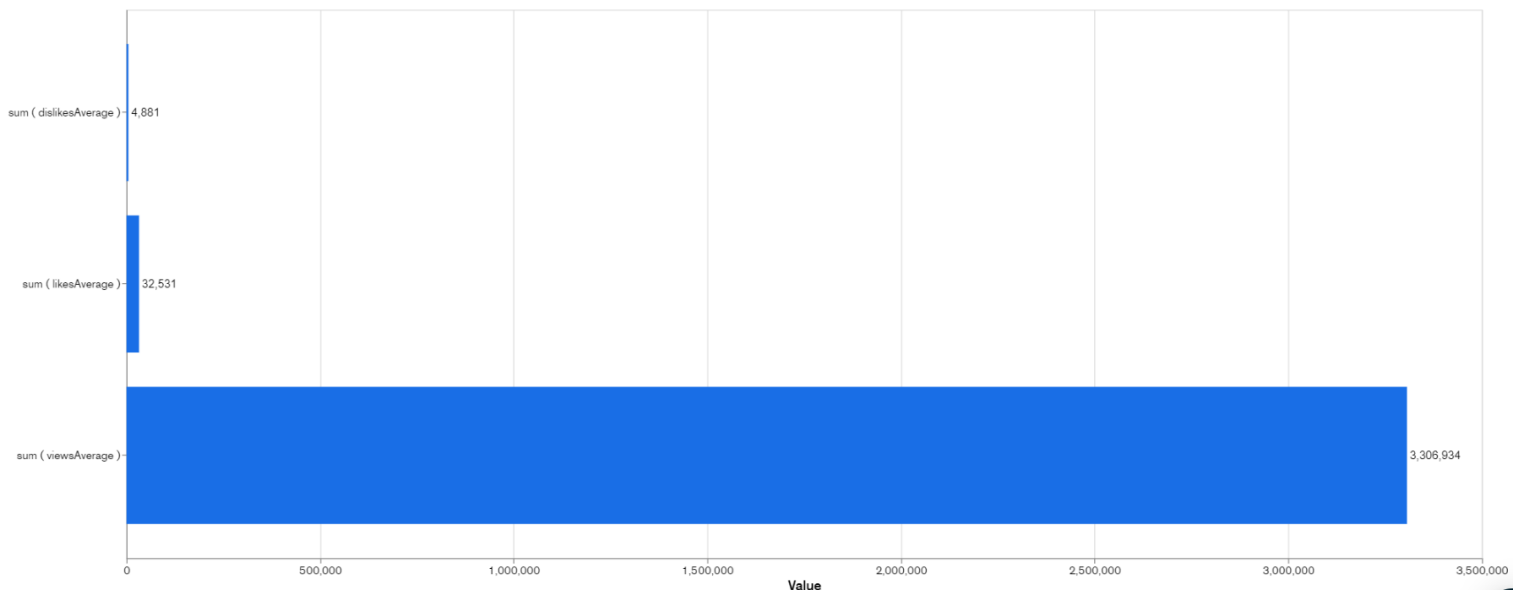


Έτσι παίρνουμε τα εξής αποτελέσματα:

question2_4CommentsDisabled_FALSE



question2_4CommentsDisabled_TRUE



Συμπεράσματα

Παρατηρώντας τα αποτελέσματα, καταλήγουμε στο συμπέρασμα πως η απενεργοποίηση των σχολίων έχει ως αποτέλεσμα τη δυσарέσκεια των viewers καθώς δεν μπορούν να εκφράσουν τη γνώμη τους και αυτό φαίνεται από την απόκλιση των likes-dislikes που φαίνονται στα bar charts. Επίσης παρατηρείται κάποια μείωση στα views αλλά δεν είναι αισθητή οπότε δεν είναι βέβαιο εάν η απενεργοποίηση των σχολίων επηρεάζει τις προβολές. Βέβαια κάποιος χρήστης πριν ανοίξει το βίντεο δε γνωρίζει εάν τα σχόλια είναι απενεργοποιημένα, οπότε ίσως ο αριθμός των προβολών είναι ανεξάρτητος των comments ως ένα σημείο, καθώς σε μερικές περιπτώσεις εάν τα σχόλια είναι απενεργοποιημένα ίσως ο χρήστης να μην παρακολουθήσει το ίδιο βίντεο 2^η φορά.

Ερώτημα 2.5

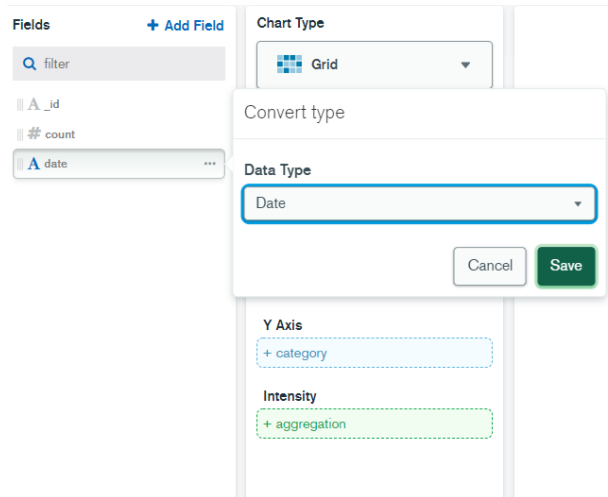
Για το ερώτημα 2.5 αρχικά σκεφτήκαμε σε πρώτο βήμα να περιορίσουμε τη χρονική περίοδο στα πλαίσια που μας ενδιαφέρει, σύμφωνα με την εκφώνηση. Επομένως μέσω ενός ταιριάσματος(**\$match**) και με χρήση των λογικών τελεστών **\$gte & \$lte** κρατήσαμε μόνο τις ημερομηνίες δημοσίευσης βίντεο που βρίσκονται στο διάστημα 5-12-2017 έως 5-3-2018. Στο επόμενο βήμα με χρήση του **\$project** επιλέγουμε μόνο τα πεδία id και publish_time. Έπειτα μέσω ενός **\$group** ομαδοποιούμε τα δεδομένα μας κρατώντας την ημερομηνία και έναν μετρητή που εκφράζει τα βίντεο που ανέβηκαν τη συγκεκριμένη μέρα. Ακολούθως λαμβάνει χώρα η ταξινόμηση(**\$sort**) σύμφωνα με την ημερομηνία μιας που βρίσκεται στο date format που θέλουμε. Επιπλέον για να κινούμαστε στα πλαίσια της εκφώνησης κάνουμε ένα **\$project** όπου κρατάμε το πλήθος που προαναφέραμε και από την ημερομηνία αφαιρούμε τη ζώνη ώρας και διατηρούμε μόνο την ημερομηνία σε μορφή **μέρα-μήνας-έτος**. Τέλος μέσω του **\$out** τα δεδομένα μας καταλήγουν στο collection **results2.5**

Ενδεικτικά οι 20 πρώτες εγγραφές από τις συνολικές 90 θα είναι οι εξής:

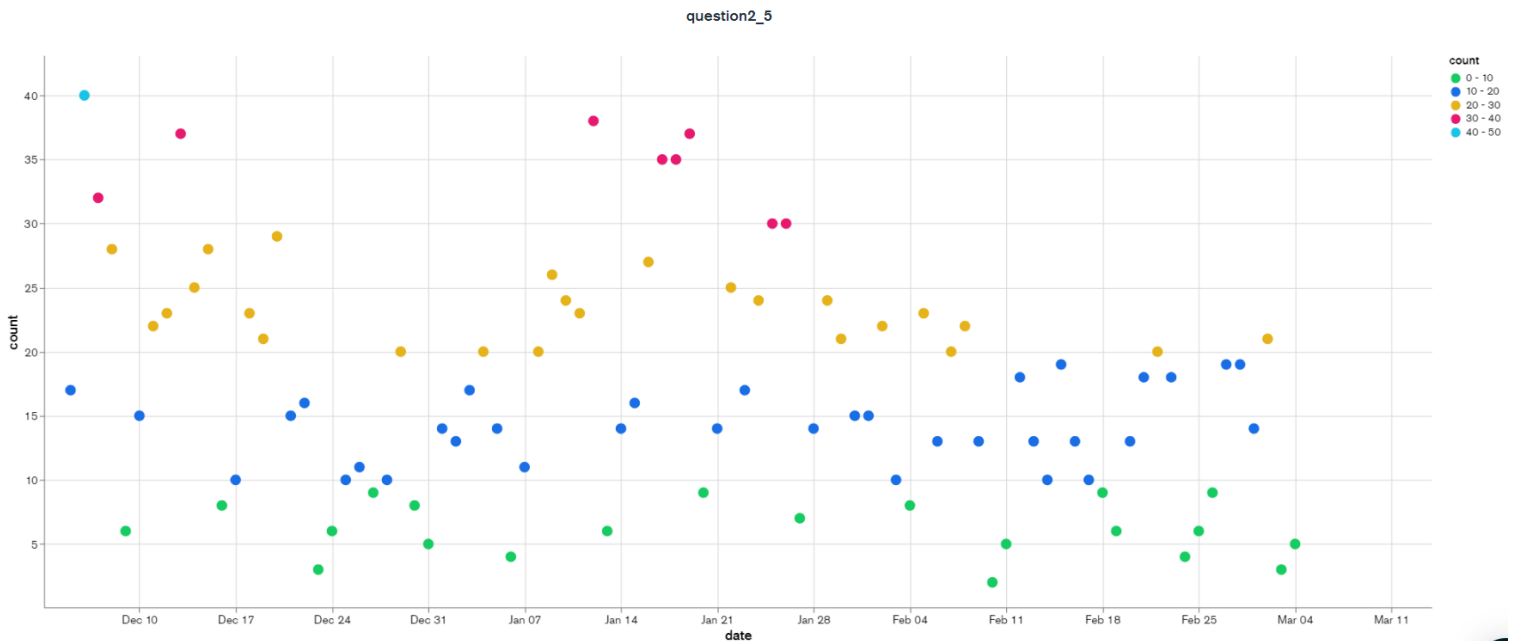
count	date
17	05-12-2017
40	06-12-2017
32	07-12-2017
28	08-12-2017
6	09-12-2017
15	10-12-2017
22	11-12-2017
23	12-12-2017
37	13-12-2017
25	14-12-2017
28	15-12-2017
8	16-12-2017
10	17-12-2017
23	18-12-2017
21	19-12-2017
29	20-12-2017
15	21-12-2017
16	22-12-2017
3	23-12-2017
6	24-12-2017

Τα πλήρη αποτελέσματα(results2.5.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο question2.5

Τώρα από το collection results2.5 θα μεταφέρουμε τα δεδομένα μας στη βάση του atlas και θα φτιάξουμε το scatter plot ακριβώς όπως το κατασκευάσαμε στο ερώτημα 2.2 με τη διαφορά πως θα πρέπει να τροποποιήσουμε το πεδίο date από string που είναι σε date προκειμένου να γίνει δεκτό από το scatter plot



Πατάμε Save, κάνουμε drag & drop τα πεδία μας στους κατάλληλους άξονες και παίρνουμε το ακόλουθο αποτέλεσμα



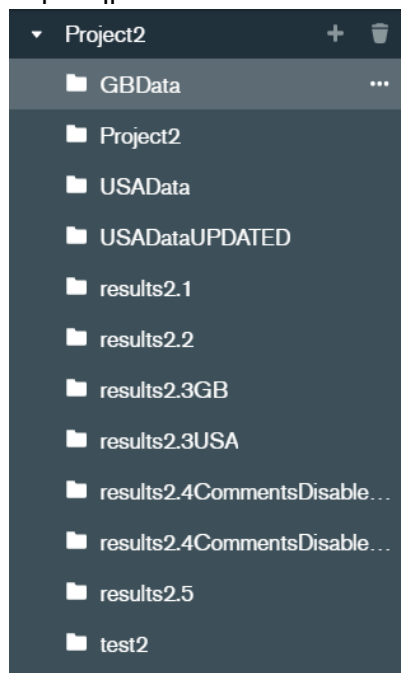
Συμπεράσματα

Παρατηρούμε πως τα περισσότερα βίντεο (40 ανέβηκαν στις 6 Δεκεμβρίου το 2017, 38 στις 12/1/2018, 37 στις 13/12/2017 και 19/1/2018 και 35 στο διήμερο 17 και 18 Ιανουαρίου.) Μετά από μια μικρή έρευνα που κάναμε βρήκαμε πως στις 6 Δεκεμβρίου το 2017 ανέβηκε το youtube rewind video από το επίσημο κανάλι του youtube και μερικά video από τα δεδομένα μας αναφέρονται πάνω σε αυτό το γεγονός. Επίσης στις 12/1/2018 αρκετά video κάνουν αναφορά σε ρατσιστικές δηλώσεις του τότε προέδρου Donald Trump σχετικά με το προσφυγικό θέμα που αντιμετώπιζαν οι Η.Π.Α. Συμπληρωματικά στις 13 Δεκεμβρίου παρατηρήσαμε πως στα descriptions και στους τίτλους των βίντεο γίνεται αναφορά στο όνομα της Taylor Swift η οποία έκανε μια ανακοίνωση εκείνη την ημέρα, ενδεχομένως με αφορμή τα γενέθλια της, τα οποία μετά από αναζήτηση διαπιστώσαμε πως ήταν εκείνη την ημέρα. Επίσης ορισμένα βίντεο αναφέρονται στην ταινία Star Wars the last jedi η οποία βρήκαμε πως έκανε πρεμιέρα στο

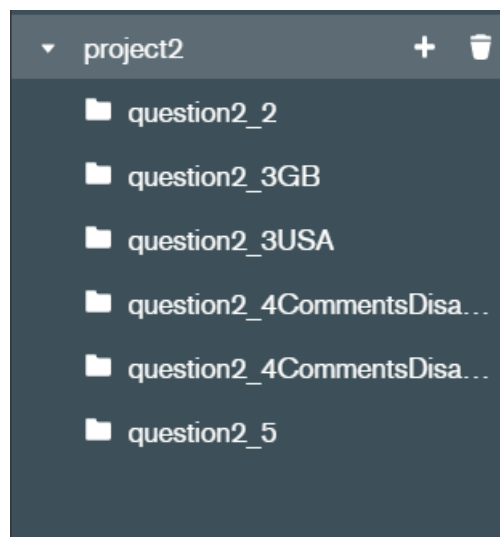
Ηνωμένο Βασίλειο την προηγούμενη μέρα. Επιπλέον μερικά βίντεο αναφέρονται στα Χριστούγεννα που πλησιάζουν. Ακολούθως στις 17/1/2018 ανέβηκαν μερικά βίντεο σχετικά με το παιχνίδι Fallout ύστερα από ένα tweet της εταιρίας. Τέλος στις 19/1/2018 με αφορμή το καινούργιο Album του Fall Out Boy ανέβηκαν μερικά βίντεο.

Εν τέλη έχουμε 2 βάσεις δεδομένων, μια για τα ερωτήματα μας και μια για το atlas όπως φαίνεται παρακάτω :

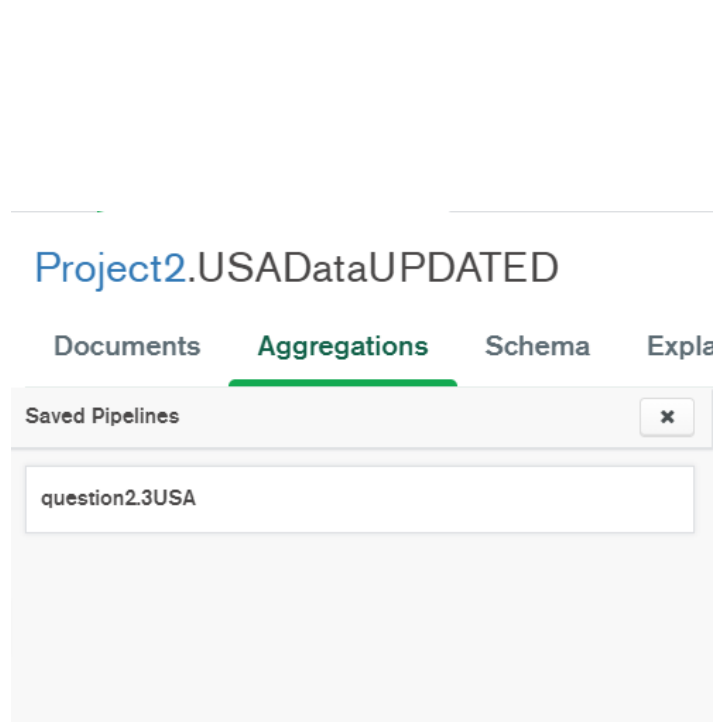
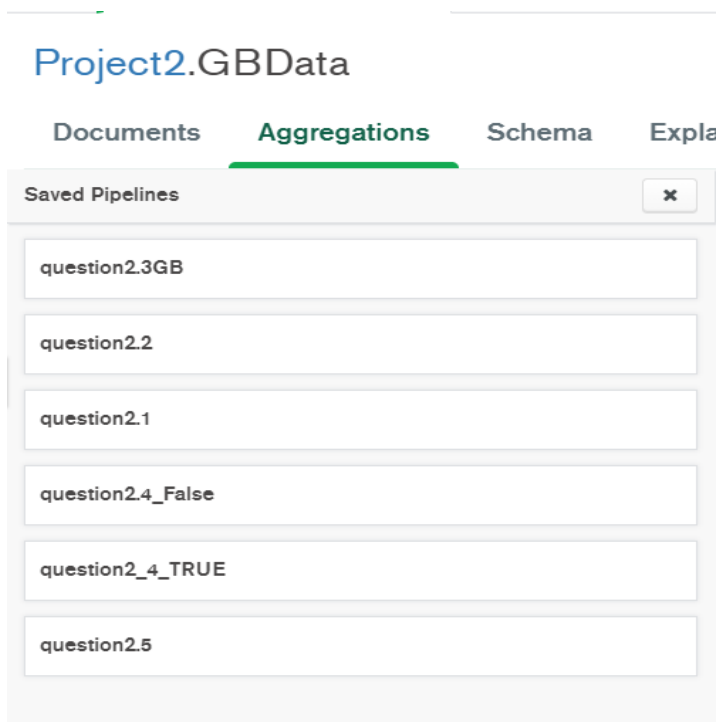
Ερωτήματα:



Atlas:



Επιπλέον σε 2 διαφορετικά collections έχουμε αποθηκεύσει σε pipelines τα ερωτήματα μας:



Bonus

Για bonus υλοποίηση η ομάδα μας επέλεξε να βρει τις μέρες που περνάνε από τη στιγμή που δημοσιευτεί ένα video μέχρι να μπει στις τάσεις, όπως μας αναφέρατε στην εκφώνηση. Αρχικά όπως είναι προφανές πρέπει να επιλέξουμε τα πεδία που μας ενδιαφέρουν μέσω ενός **\$project**. Έχοντας τα πεδία `publish_time` και `trending_date` παρατηρήσαμε πως το `trending_date` ήταν σε μορφή **year.day.month**. Επομένως εμείς αποφασίσαμε να το μετατρέψουμε στη μορφή **year-month-day**. Έτσι σε πρώτο βήμα μέσω ενός **\$replaceAll** όπου "." βάλουμε "-". Στη συνέχεια μέσω ενός **\$split** σπάσαμε το string μας σύμφωνα με τις παύλες, έτσι ώστε να απομονώσουμε την ημέρα, το μήνα και το έτος, αποθηκεύοντάς τα σε έναν πίνακα. Έτσι με χρήση της εντολής **\$arrayElemAt** πήραμε το περιεχόμενο του πίνακα και το αποθηκεύσαμε σε 3 μεταβλητές(`year, month, day`). Έπειτα πραγματοποιήσαμε ένα string concatenation(**\$concat**) ενώνοντας τις 3 μεταβλητές σε μια με όνομα `trending_date`. Όμως είναι σε μορφή string και για να βρούμε τη διαφορά ημερών από το πεδίο `publish_time` έπρεπε να την μετατρέψουμε σε date όπως και κάναμε με χρήση της **\$dateFromString**. Επομένως τώρα που και τα 2 πεδία μας βρίσκονται στη σωστή μορφή κάναμε αφαίρεση(**\$subtract**)το `publish_time` από το `trending_day`, όμως επειδή το αποτέλεσμα θα ήταν σε ms διαιρέσαμε(**\$divide**) με $24*60*60*1000= 86400000$ για να βρούμε τις μέρες. Στο τελευταίο μας βήμα απλά μετατρέψαμε το αποτέλεσμα μας σε ακέραιο προς τα πάνω(**\$ceil**)

The image displays four sequential screenshots of the MQL5 IDE, illustrating the stages of a script designed to calculate the number of days between a video's publish time and its trending date.

- Stage 1:** The initial script defines the input fields: `trending_date: 1,` and `publish_time: 1`. The output shows the initial state of the variables: `_id: "HKIIgVFhQIE"`, `trending_date: "17.19.12"`, and `publish_time: 2017-12-08T05:00:01.000+00:00`.
- Stage 2:** The script uses `$replaceAll` to replace the dots in the trending_date with hyphens. The output shows the updated `trending_date` as `"17-19-12"`.
- Stage 3:** The script uses `$split` to split the trending_date string by hyphens. The output shows the resulting array: `date: Array` with elements `0: "17"`, `1: "19"`, and `2: "12"`.
- Stage 4:** The script uses `$arrayElemAt` to extract the year, month, and day from the array. The output shows the final state of the variables: `_id: "HKIIgVFhQIE"`, `publish_time: 2017-12-08T05:00:01.000+00:00`, `year: "17"`, `month: "12"`, and `day: "19"`.

\$project

1

2

3

4

5

6

```

1 {
2   trending_date: {
3     $concat: ['$year', '-', '$month', '-', '$day']
4   },
5   publish_time: 1
6 }

```

Output after \$project stage (Sample of 20 documents)

_id: "HKIIgYFhQ1E"

publish_time: 2017-12-08T05:00:01.000+00:00

trending_date: "17-12-19"

\$project

1

2

3

4

5

6

7

8

```

1 {
2   trending_date: {
3     $dateFromString: {
4       dateString: '$trending_date'
5     }
6   },
7   publish_time: 1
8 }

```

Output after \$project stage (Sample of 20 documents)

_id: "HKIIgYFhQ1E"

publish_time: 2017-12-08T05:00:01.000+00:00

trending_date: 2017-12-19T00:00:00.000+00:00

\$project

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

```

1 {
2   publish_time: {
3     $dateToString: {
4       format: '%d-%m-%Y',
5       date: '$publish_time'
6     }
7   },
8   trending_date: {
9     $dateToString: {
10      format: '%d-%m-%Y',
11      date: '$trending_date'
12    }
13  },
14  dayssince: {
15    $divide: [{
16      $subtract: ['$trending_date', '$publish_time']
17    }, 86400000]
18  }
19 }

```

Output after \$project stage (Sample of 20 documents)

_id: "HKIIgYFhQ1E"

publish_time: "08-12-2017"

trending_date: "19-12-2017"

dayssince: 10.791655092592592

\$project

1

2

3

4

5

6

7

```

1 {
2   dayssince: {
3     $ceil: '$dayssince'
4   },
5   publish_time: 1,
6   trending_date: 1
7 }

```

Output after \$project stage (Sample of 20 documents)

_id: "HKIIgYFhQ1E"

publish_time: "08-12-2017"

trending_date: "19-12-2017"

dayssince: 11

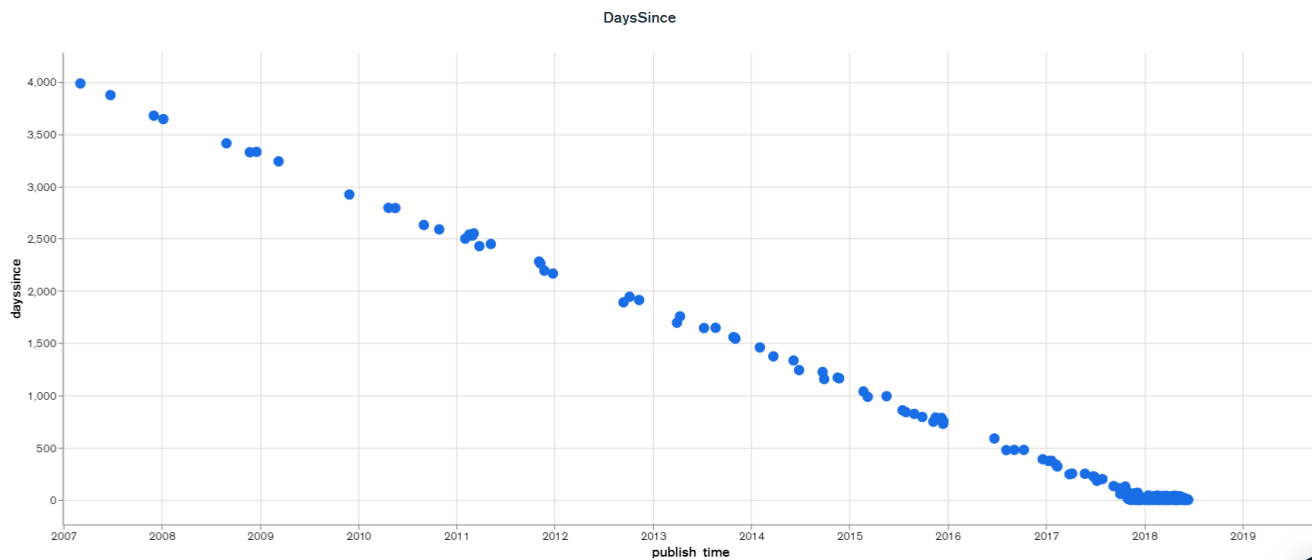
Ενδεικτικά τα πρώτα 20 αποτελέσματα είναι τα εξής:

_id	dayssince	publish_time	trending_date
HKIIgYFhQ1E	11	08-12-2017	19-12-2017
M39tO96E6B0	1	27-02-2018	28-02-2018
tcaw61zYt1Q	29	02-05-2018	31-05-2018
rKH3G023Xfw	13	07-05-2018	20-05-2018
VEoOuF_cN0Q	12	13-11-2017	25-11-2017
nkWn5vKCZj4	14	17-05-2018	31-05-2018
J-dv_DcDD_A	18	12-04-2018	30-04-2018
tjVJgR4OrgM	13	13-11-2017	26-11-2017
bNW6oR670s4	23	07-04-2018	30-04-2018
eeBMQpzoEXQ	6	31-05-2018	06-06-2018
vHCvmZ1C5jQ	20	01-11-2017	21-11-2017
nshmwFK3yd8	10	07-11-2017	17-11-2017
c7gO55FhKUQ	18	25-12-2017	12-01-2018
T0_M3t3UVao	19	21-12-2017	09-01-2018
wrzWGuLTnOA	10	14-03-2018	24-03-2018
6x3iBCfHFDc	1	03-06-2018	04-06-2018
CfyMD5rSK50	4	27-01-2018	31-01-2018

```
lp_bsz_4e3U,25,03-02-2018,28-02-2018  
USigj639wTU,11,03-12-2017,14-12-2017  
8A3ILTzaswo,14,04-01-2018,18-01-2018
```

Τα πλήρη αποτελέσματα(bonus.json) καθώς και τα ερωτήματα(questions.txt) που χρησιμοποιήσαμε βρίσκονται στο φάκελο bonus

Μέσω του atlas κατασκευάσαμε και ένα ενδεικτικό scatter plot με σκοπό να εξάγουμε συμπεράσματα.



Συμπεράσματα

Παρατηρούμε πως τα πιο πρόσφατα βίντεο χρειάζονται λιγότερες μέρες για να μπουν στις τάσεις