MongoDB

Below is a screenshot of a sample of the dataset that I used for this project. It is some data from kaggle about the top tik-tock users, mostly celebrities and some metrics about how popular they are.

	A	B	C	D	E	F	G	H
1	Account	Title	Subscribers c	Views avg.	Likes avg.	Comments a	Shares avg.	
2	billieeilish	BILLIE EILISH	41600000	95000000	18100000	151000	54800	
3	badbunny	Bad Bunny	13400000	33300000	7300000	101900	106800	
4	chipmunksof	Chipmunks o	11400000	25000000	3000000	21800	64200	
5	charlidameli	charli d,Äôar	135400000	18700000	2600000	54700	35200	
6	yessicadcs	Yessica y Mo	263700	16700000	1400000	16200	138400	
7	therock	The Rock	49700000	24700000	3000000	30900	16400	
8	the_pruld	ThePruld	433300	12200000	2700000	36300	56600	
9	lukedonkin	Luke Donkin	2000000	14000000	2400000	34000	54900	
10	mastertingu	mastertingu	1300000	13400000	2000000	32500	44100	
11	txt.bighitent	TOMORROW	15300000	7200000	2200000	57200	31000	
12	besttoks	Best of TikTc	1800000	55900000	1500000	3300	12700	
13	machinegunl	Colson	3200000	17500000	1400000	26800	23300	
14	jenniferkersh	Jennifer Kers	12500	6400000	1600000	22600	62000	
15	noahschnapp	Noah Schnap	20100000	8600000	1800000	42000	11300	
16	bdylanhollis	B. Dylan Holl	6400000	9100000	1500000	12400	46100	
17	fisherbenny	Benjamin Ba	3700000	13100000	1100000	7000	41700	
18	sylvaniandra	Sylvaniandra	1800000	5500000	1300000	32700	56500	
19	iamferv	Fernanda	24600000	12900000	1900000	15800	8300	
20	mercyfno	Mercy	404500	11800000	1400000	17400	18100	
21	userbigbaby	Byron Trema	601800	8700000	414200	23600	125400	
22	hellanearsh	Bella Nearsh	87000000	15400000	1800000	16400	3100	

I loaded this data into my mongo database using the python script below and executed a series of queries. Below is the code and output of those queries.

```
#Step 2: Create sample data
file = open('/Users/stephennelson/Projects/MongoDB/ti
csvreader = csv.reader(file)
header = next(csvreader)
rows = []
for row in csvreader:
    rows.append(row)
file.close()
for i in tqdm(range(len(rows))):
    user = { str(header[0]) : rows[i][0],
             str(header[1]) : rows[i][1],
             str(header[2]) : rows[i][2],
             str(header[3]) : rows[i][3],
             str(header[4]) : rows[i][4],
             str(header[5]) : rows[i][5],
             str(header[6]) : rows[i][6]
    }
    result = db.accounts.insert_one(user)
print("data has been updated")
```

This block was used to calculate the accounts with the highest number of subscribers and average shares.

```
cursor = db['accounts'].aggregate([
    {"$match" : {"Subscribers": { "$gte": 100000000 }}
    },
    { "$group": { "_id": "null", "Account": { "$sum": 1 }} }
    ]);
for document in cursor:
    print("number of accounts with more than 100,000,000 subscribers: ", document)
cursor = db['accounts'].aggregate([
    {"$match" : {"Subscribers": { "$gte": 100000000 }}
    }
    ]);
for document in cursor:
    print(document)
cursor = db['accounts'].aggregate([
   { "$match": { "$or": [ { "Subscribers": { "$gt": 10000000} }, { "Shares": { "$gte": 100000} } ] } },
  { "$group": { "_id": "Null", "Account": { "$sum": 1 } } }
] );
for document in cursor:
    print("number of accounts with more than 100,000,000 subscribers and average shares greater than 100,000: ", document)
```

```
(Mongo) stephennelson@Stephens-MBP MongoDB % python leaderboard.py
number of accounts with more than 100,000,000 subscribers:  {'_id': 'null', 'Account': 2}
{'_id': ObjectId('620c801e2dc63405d1cd70a7'), 'Account': 'charlidamelio', 'Title': 'charli d'amelio', 'Subscribers': 135400000, 'Vi
ews': 18700000, 'Likes': 2600000, 'Comments': 54700, 'Shares': 35200}
{'_id': ObjectId('620c801f2dc63405d1cd70bf'), 'Account': 'khaby.lame', 'Title': 'Khabane lame', 'Subscribers': 131100000, 'Views':
13800000, 'Likes': 1600000, 'Comments': 12700, 'Shares': 5900}
number of accounts with more than 100,000,000 subscribers and average shares greater than 100,000:  {'_id': 'Null', 'Account': 208}
```

This block returned the value of the accounts that averaged the most comments and the least comments.

```
cursor = db['accounts'].aggregate([
    {"$sort":{"Likes": ASCENDING}},
    {"$group" :
    {"_id" :"Title", "leastcomments" : {"$first" :"$Comments"},
                    "mostcomments"  : {"$last" :"$Comments"}

    }}
    ]);
for document in cursor:
    print("min/max number of average comments: ", document)
```

```
(Mongo) stephennelson@Stephens-MBP MongoDB % python min_max_comments.py
min/max number of average comments:  {'_id': None, 'leastcomments': 26400, 'mostcomments': 151000}
```