

# 1. Navrhnete dátový model (kolekcie a formát dokumentov) v MongoDB pre dataset tweetov, ktorý bude využívaný mobilnou aplikáciou, ktorá bude: a. Zobrazovať tweety jednotlivých používateľov vo forme feedov b. Zobrazovať jednotlivé tweety a ich retweets

Ako referenciu na to aké všetky informácie je potrebné o tweete mať zobrazené vo feede som použil tento tweet od Elona Muska. Tento tweet je čisto ukážkový v databáze neexistuje.



Pri návrhu som sa rozhodol vytvoriť dva dokumenty. Prvý dokument je dokument author v ktorom sa nachádzajú všetky informácie o autorovi. Ukážku tohto dokumentu môžeme vidieť na obrázku nižšie.

```
{
  "id": 224,
  "name": "Dave Pell",
  "username": "davepell",
  "description": "Managing Editor, Internet.",
  "followers_count": 60064,
  "following_count": 1054,
  "tweet_count": 54618,
  "listed_count": 2016
}
```

Druhý dokument ktorý som vytvoril je dokument s názvom tweet tento dokument obsahuje všetky údaje dostupne o tweete plus sa tam nachádza username, name a pomocou author\_id je naviazaný na konkrétneho autora ktorý ho tweetoval. Dôvodom pridania username a name je, že vždy pri zobrazení tweetu sa musia zobrazíť aj tieto dva údaje. To znamená, že tieto dva údaje budú vždy dopytované spolu s daným tweetom. Ostatné údaje o userovi už nebudú dopytované tak často preto ich môžeme nechať v samostatnom dokumente. Na obrázku nižšie môžeme vidieť ako vyzerá.

```

{
  "source_id": 1496931553645318144,
  "content": "I'm not a religious guy, but thank god Trump isn't president right now.",
  "language": "en",
  "source": "Twitter for iPhone",
  "retweet_count": 80,
  "reply_count": 27,
  "like_count": 910,
  "quote_count": 2,
  "possibly_sensitive": false,
  "created_at": "2022-02-24T19:34:28+01:00",
  "author_id": 224,
  "username": "davepell",
  "name": "Dave Pell",
  "hashtags": null,
  "links": null,
  "annotations": [
    {
      "type": "Person",
      "value": "Trump",
      "probability": 0.999
    }
  ],
  "context_annotations": [
    {
      "domain": {
        "name": "Person",
        "description": "Named people in the world like Nelson Mandela"
      },
      "entity": {
        "name": "Donald Trump",
        "description": "45th US President, Donald Trump"
      }
    },
    {
      "domain": {
        "name": "Politician",
        "description": "Politicians in the world, like Joe Biden"
      },
      "entity": {
        "name": "Donald Trump",
        "description": "45th US President, Donald Trump"
      }
    },
    {
      "domain": {
        "name": "Unified Twitter Taxonomy",
        "description": "A taxonomy view into the Semantic Core knowledge graph"
      },
      "entity": {
        "name": "Donald Trump",
        "description": "45th US President, Donald Trump"
      }
    },
    {
      "domain": {
        "name": "Unified Twitter Taxonomy",
        "description": "A taxonomy view into the Semantic Core knowledge graph"
      },
      "entity": {
        "name": "Politics",
        "description": "Politics"
      }
    },
    {
      "domain": {
        "name": "Unified Twitter Taxonomy",
        "description": "A taxonomy view into the Semantic Core knowledge graph"
      },
      "entity": {
        "name": "Political figures",
        "description": "Politician"
      }
    }
  ],
  "conversation_references": null
}

```

Modely ktoré môžeme vidieť hore. Vidíme, že modely sú podobné ako v predošlom zadaní. Autor obsahuje autora a tweets konverzáciu a všetky k nej patriace veci. Teraz bližšie k tweetom. Použil som takzvaný embedded prístup to znamená, že v konverzácií mám vložené polia s jsonami ktoré obsahujú hashtags, links, annotations, context\_annotations, conversation\_references. Ďalej tu je author\_id v tweete takže každý tweet ma priradený svojho autora to sa nazýva document referencing. Taktiež v conversation\_referencing máme tweet k parent tweetu to sa nazýva document referencing v rovnakej kolekcii. Rozhodol som sa pre tweet s referenciou na author\_id a nie pre authora ktorý by mal pole referencii na všetky svoje tweety, tweet\_id pretože to by mohlo viesť k veľmi veľkým dokumentom pre používateľov ktorý radi tweetuju predstavme si radovo tisíceky ideciiek. Na druhu stranu by to viedlo k užívateľom ktorý napríklad netweetuju vôbec len sa pozerajú a mali by toto pole prázdne. To znamená vytvorili by sa veľmi rozdielne veľké dokumenty.

## 1. Zobrazovať tweety jednotlivých používateľov vo forme feedov

Každý tweet obsahuje author\_id to znamená keď chcem zobrazovať tweety pre jednotlivých používateľov vo forme feedov môžeme použiť napríklad takúto query. Sortnute to je preto lebo chcem vidieť posledný tweet používateľa ktorý uverejnil. Samozrejme nechceme všetky tweety tohto používateľa ale len urcitym limit napríklad 10 nato použijeme funkciu limit.

```
db.tweets.find({
  author_id: <author_id>
}).sort( { created_at: -1 } ).limit( 10 )
```

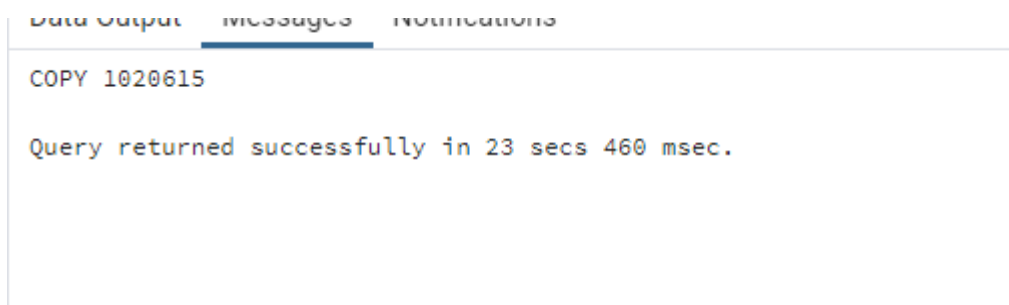
## 2. Zobrazovať jednotlivé tweety a ich retweets

Každý tweet obsahuje conversation\_references kde sa nachádza parent\_id a type to znamená, že si vieme dohľadať tweet a všetky jeho retweets pomocou nasledujúcej query. Taktiež je tam sort created\_at aby sme mali posledné tweety a nejaký limit 10 aby sme nevrátili nekonečno tweetov.

```
db.tweets.find( {
  conversation_references: {
    $elemMatch: {
      parent_id: <tweet_id>,
      type: "retweeted"
    }
  }
} ).sort( { created_at: -1 } ).limit( 10 )
```

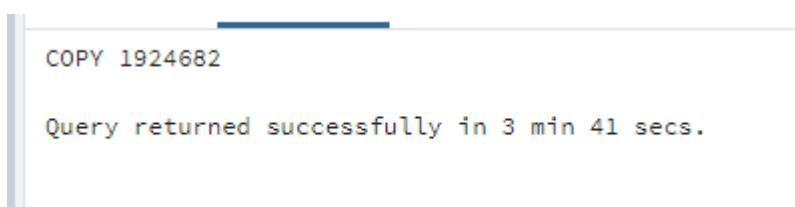
## 2. Nainštalujte alebo využite online inštanciu MongoDB servera, do ktorého importujte všetky tweets (a s nimi spojené data – anotácie, referencie, odkazy a informácie o kontexte) zo dňa 24.02.2022.

Dáta o autoroch z postgres bolo potrebné exportnúť, query ktorú som nato použil môžeme nájsť v priečinku query súbor author.txt. Export authorov trval približne pol minúty tento export môžeme vidieť na nasledujúcom obrázku.



```
Data Output  Messages  Notifications
COPY 1020615
Query returned successfully in 23 secs 460 msec.
```

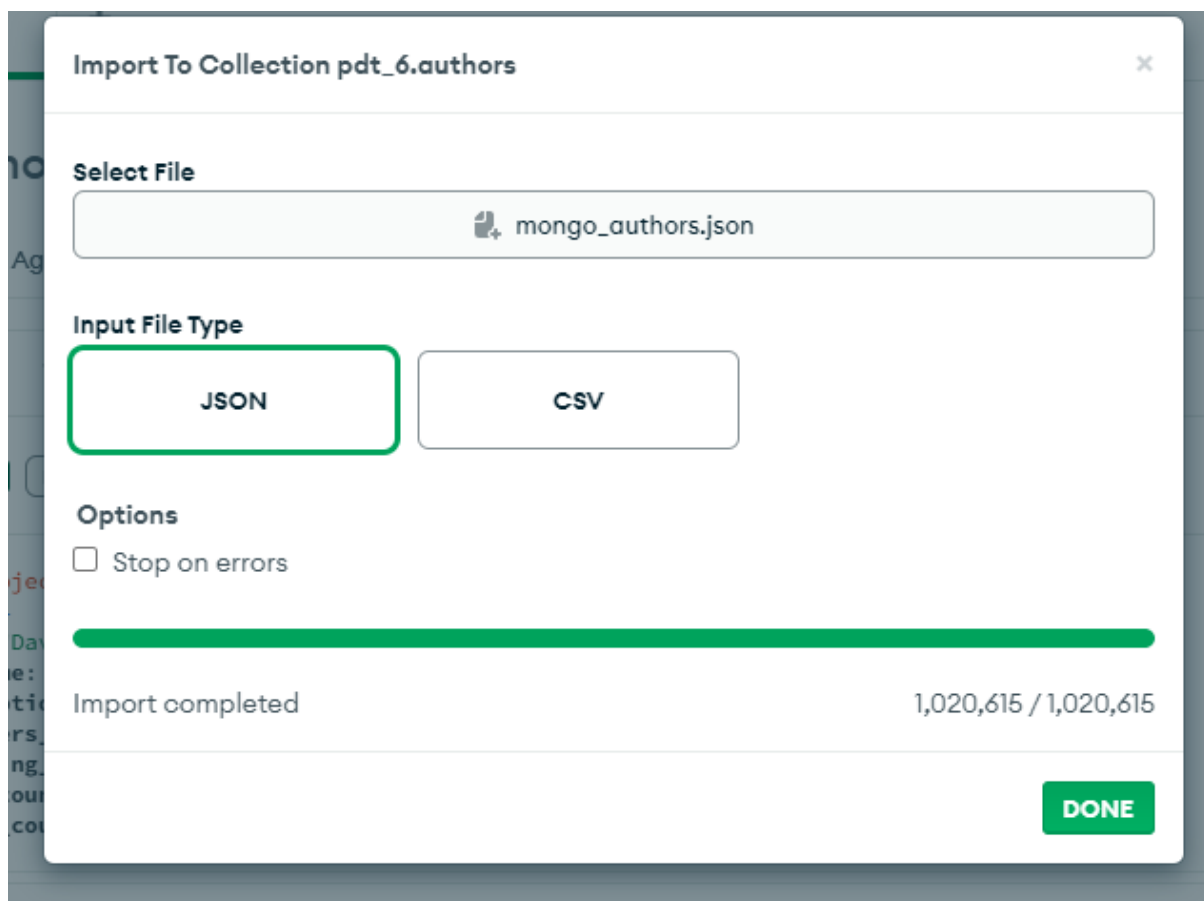
Dáta o tweetoch z postgresu bolo taktiež potrebné exportnúť, query ktorú som nato použil môžeme nájsť v priečinku query súbor tweets.txt. Export tweetov trval približne 4 minúty, tento export môžeme vidieť na nasledujúcom obrázku.



```
COPY 1924682
Query returned successfully in 3 min 41 secs.
```

Query ktoré boli použité na export z postgres do JSON formátu mali podobnú logiku ako v predchádzajúcom zadaní.

Export už bol pripravený teraz bolo na čase nainštalovať mongo a dáta importnúť. Nainštaloval som si MongoDB Community Server toto mi nainštalovalo MongoDB compass niečo na štýl pgadmin akurát pre mongo a server na ktorom mi beží databáza, tiež som potreboval niečo do čoho písať queriny tak som stiahol MongoDB Shell. Vytvoril som databázu pdt\_6 a dve kolekcie. Kolekciu authors a kolekciu tweets. Dáta som importol z jsonu. Import trval približne minútu. Na obrázku nižšie môžeme vidieť import authorov. Ako vidieť na obrázku importli sa všetky bez problémov.




Potom som importol tweety to môžeme vidieť na obrázku nižšie import trval približne 8 minút. Znova môžeme vidieť, že všetky dokumenty sa importli bez problémov.

Import To Collection pdt\_6.tweets



Select File

 mongo\_tweets.json

Input File Type

JSON

CSV

Options

☐ Stop on errors



Import completed

1,924,682 / 1,924,682

DONE

**3. Napíšte dotaz, ktorý nad importovanou databázou: a. Vypíše posledných 10 tweetov pre autora, ktorý má username Newnews\_eu b. Vypíše posledných 10 retweetov pre tweet, ktorý má id 1496830803736731649.**

Najprv som pomocou príkazu db zistil v akej db som. Potom som pomocou príkazu use pdt\_6 sa presunul do mojej databázy nad ktorou chcem vykonávať query.

**1. Úloha**

Query:

```
db.tweets.aggregate([
  { $match : { username : "Newnews_eu" } },
  { $sort: { created_at: -1 } },
  { $limit: 10 },
])
```

Response z tejto query môžeme nájsť v priečinku responses súbor uloha1.json. Query je veľmi jednoduchá matchnem podľa mena Newnews\_eu sortnem od najväčšieho po najmenší podľa fieldu created\_at a výsledok zlimitujem len na 10 tweetov.

**2. Úloha**

Query:

```
db.tweets.find( {
  conversation_references: {
    $elemMatch: {
      parent_id: "1496830803736731649",
      type: "retweeted"
    }
  }
}).sort( { created_at: -1 }).limit( 10 )
```

Response z tejto query môžeme nájsť v priečinku responses subor uloha2.json. Query je veľmi jednoduchá nájdem v conversation\_references taký tweet ktorý ma parent id "1496830803736731649", a jedna sa o retweet následne tieto údaje zoradím od najväčšieho po najmenší podľa fieldu created\_at a dám limit 10.