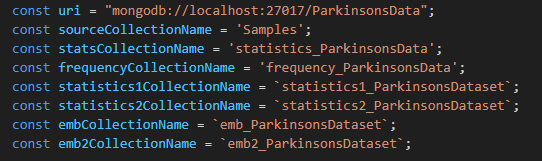
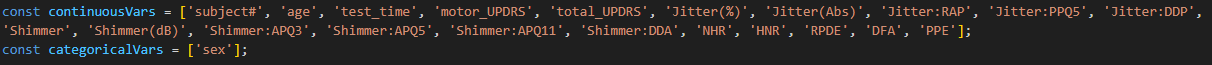
Project 1 documentation

First we import mongoDb client into our sript  
  
  
next, we define mongo connection strings, and all the collections required for this project.

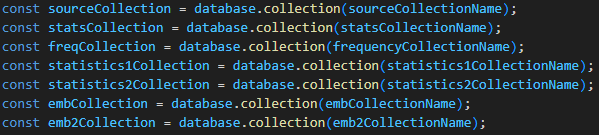
  
  
after that we need to seperate our dataset attribute types from continuous and categorical types:



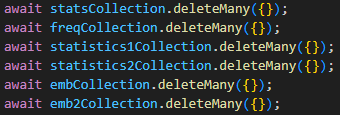
Here we're connecting to mongo using uri above, to ParkinsonsData database



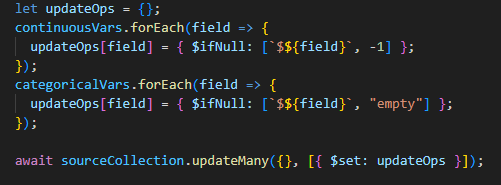
Creating collections inside the DB to be used for the future tasks:



Resetting data incase we wanted to re-execute the script (so we wont have duplicates):



Task 1: we replace all empty values of Continuous attributes with -1, and for Categorical attributes with 'Empty' string

$ifNull was used according to <https://www.mongodb.com/docs/manual/reference/operator/aggregation/ifNull/#:~:text=%24ifNull%3A%20%5B%20%22%24description%22%2C%20%22Unspecified%22%20%5D%20%7D>  
  


Task 2: we calculate mean, standard deviation, and number of elements, for this the aggragate function was used (1 pipeline) and the result was grouped using \_id

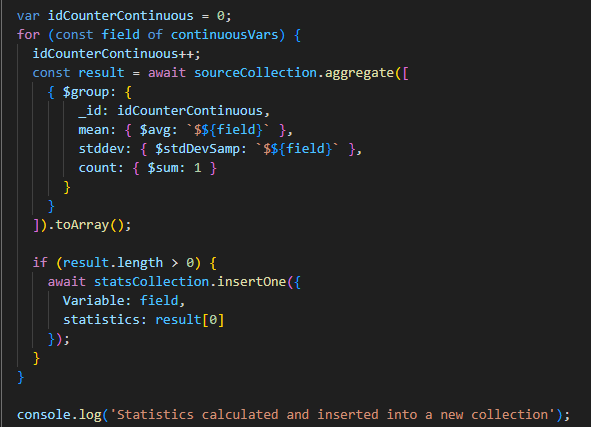
documentation for aggragate function can be found on : https://www.mongodb.com/docs/manual/aggregation/

$group function was used according to https://www.mongodb.com/docs/manual/aggregation/#:~:text=%24group%20stage.-,The%20%24group%20stage%3A,-Groups%20the%20remaining

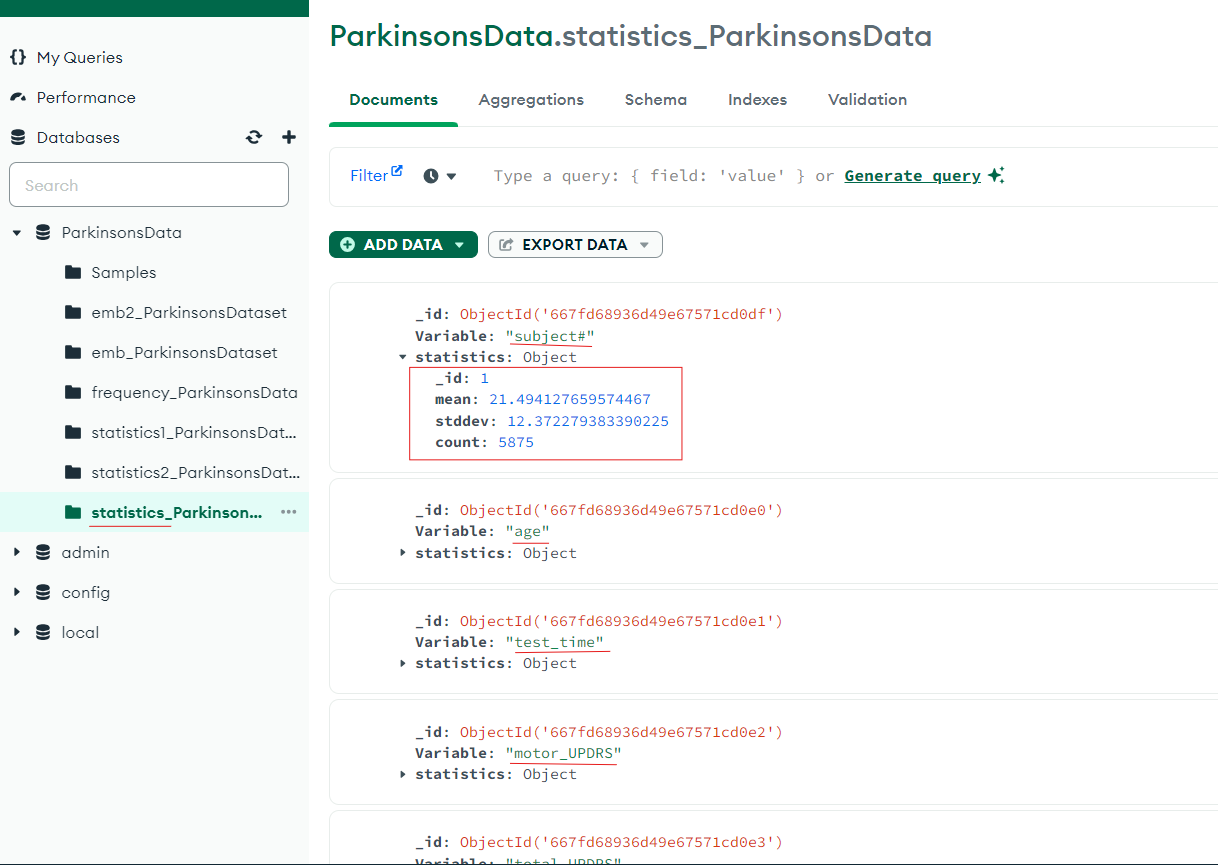
$avg function was used according to https://www.mongodb.com/docs/manual/reference/operator/aggregation/avg/#:~:text=%22%24item%22%2C-,avgAmount%3A%20%7B%20%24avg%3A,-%7B%20%24multiply%3A

$stdDevSamp was used according to https://www.mongodb.com/docs/manual/reference/operator/aggregation/stdDevSamp/#:~:text=ageStdDev%3A%20%7B-,%24stdDevSamp%3A%20%22%24age%22,-%7D%20%7D%20%7D

$sum was used according to https://www.mongodb.com/docs/manual/reference/operator/aggregation/sum/#:~:text=quantity%22%20%5D%20%7D%20%7D%2C-,count%3A%20%7B%20%24sum%3A%201%20%7D,-%7D



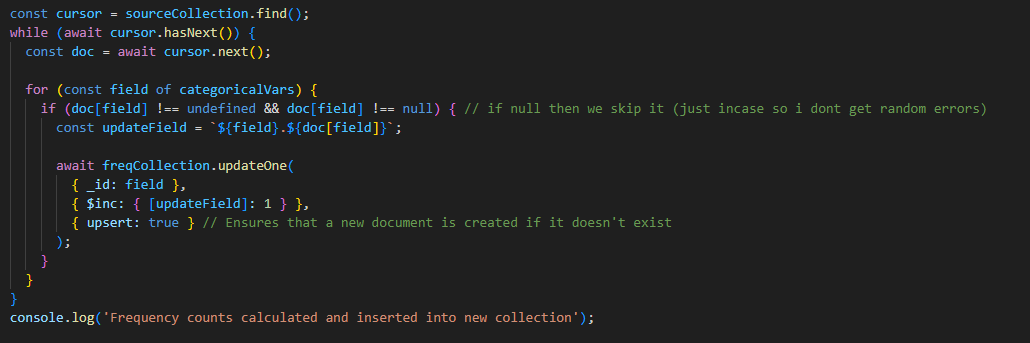
As shown in the image below, every category was inserted into new collection, and mean, standard deviation, and count was calculated for that category.



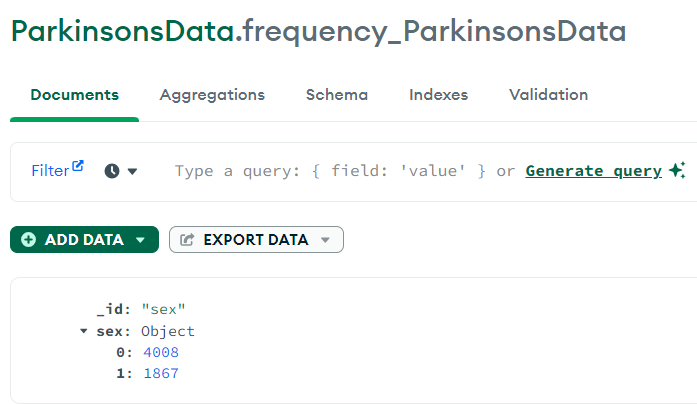
Task 3: Calculate frequency using $inc, first we go through all the data row by row(.next()) and for each row we use $inc to the related field, if field doesnt exist, we create it and fill them inside frequency document.

$inc was used according to https://www.mongodb.com/docs/manual/reference/operator/update/inc/#:~:text=%24inc%3A%20%7B%20quantity%3A%20%2D2%2C%20%22metrics.orders%22%3A%201%20%7D

updateOne was used according to <https://www.geeksforgeeks.org/mongodb-updateone-method-db-collection-updateone/#:~:text=db.collection.updateOne(%3Cfilter%3E%2C%20%3Cupdate%3E%2C%20%7B%0A%C2%A0%20%C2%A0upsert%3A%20%3Cboolean%3E%2C>



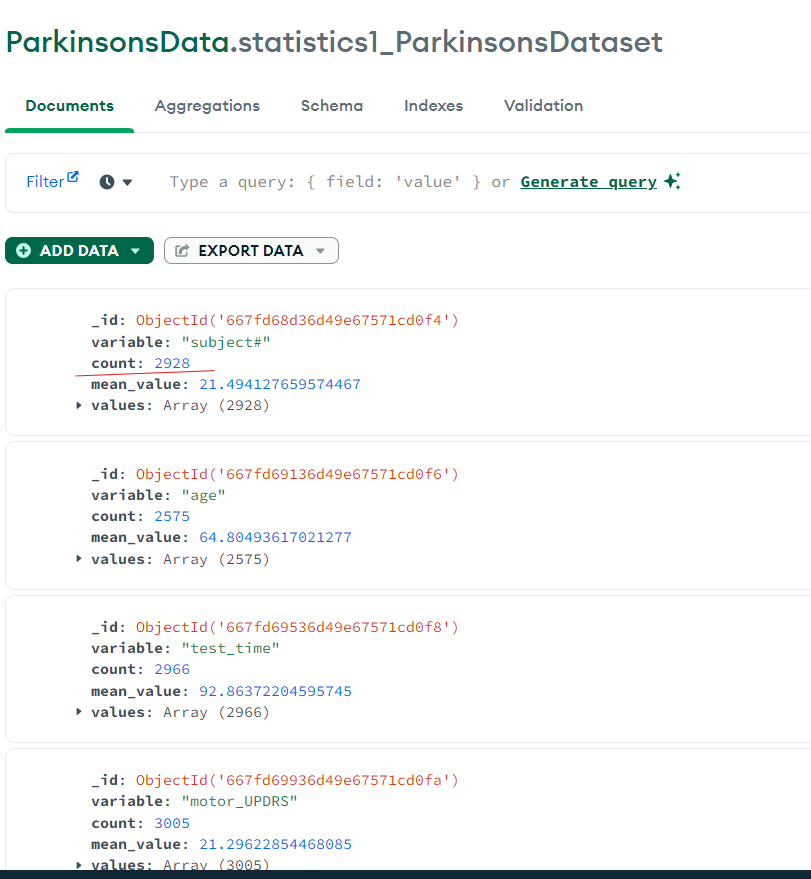
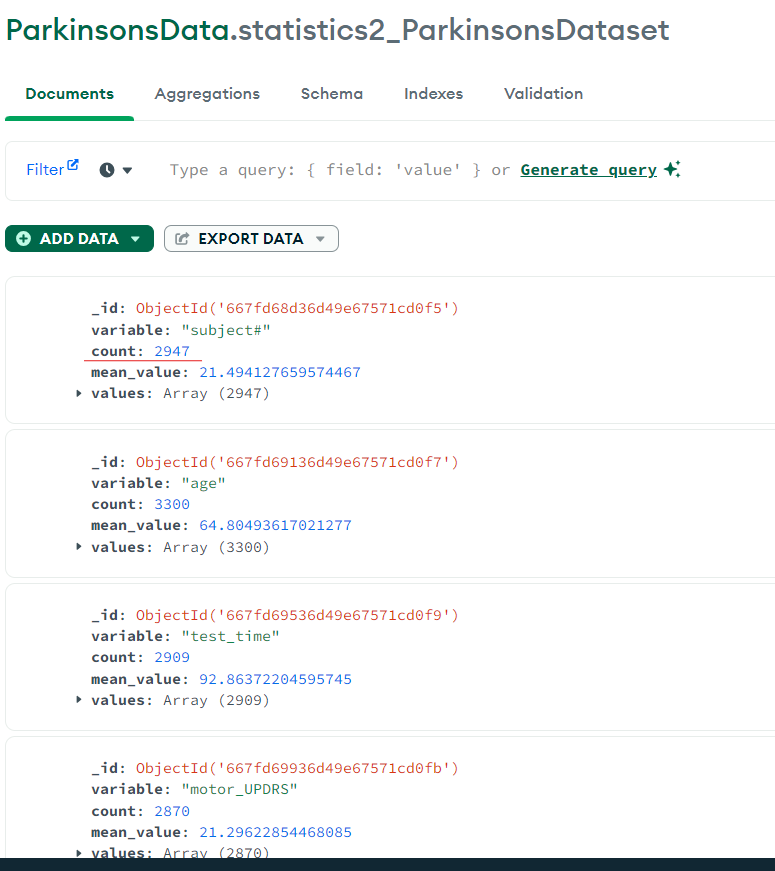
Result of this was generated in the frequency collection shown below



4008 male, and 1867 females.

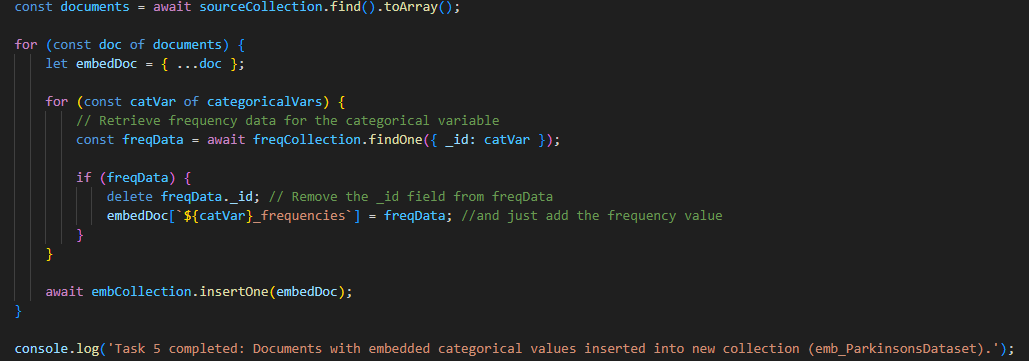
Task 4: Create 2 new documents, and fill the first one with the rows which are lower than the mean and the second one with higher.

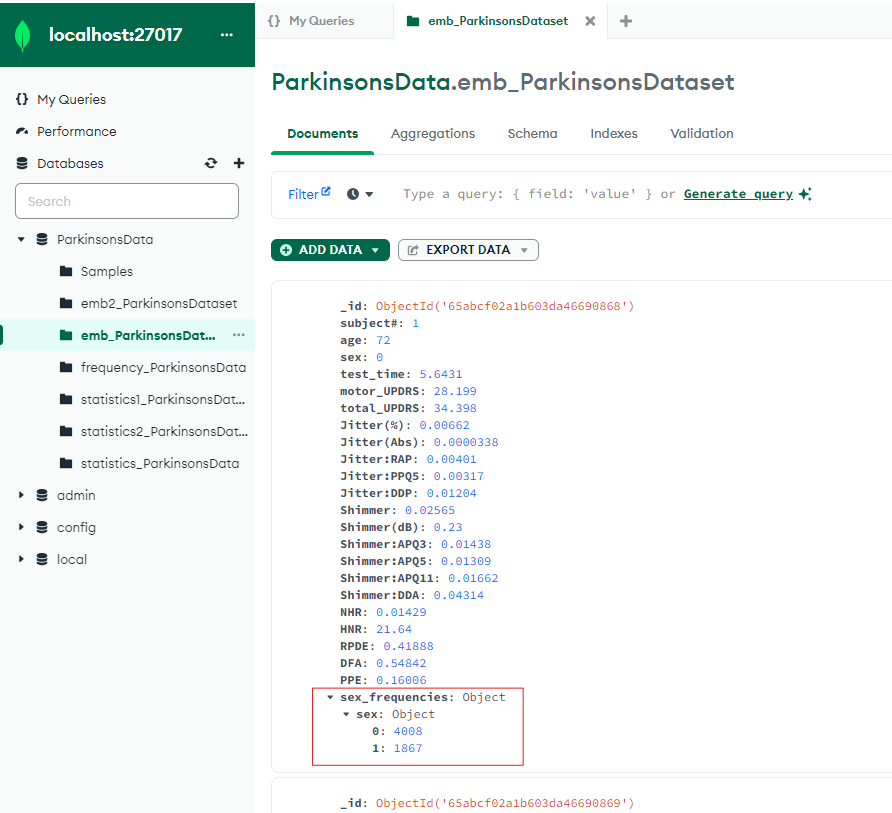


Result:  
statistic1:  
  
  
  
statistic2:  
  
  
the total that we had of for example Subject#, was 5875 which was split into 2 collections, first one was 2928, and second one was 2947, which add up to the total sum correctly.

Task 5: clone the first document and add to it the frequencies for the categorical values and insert it into new collection (emb).

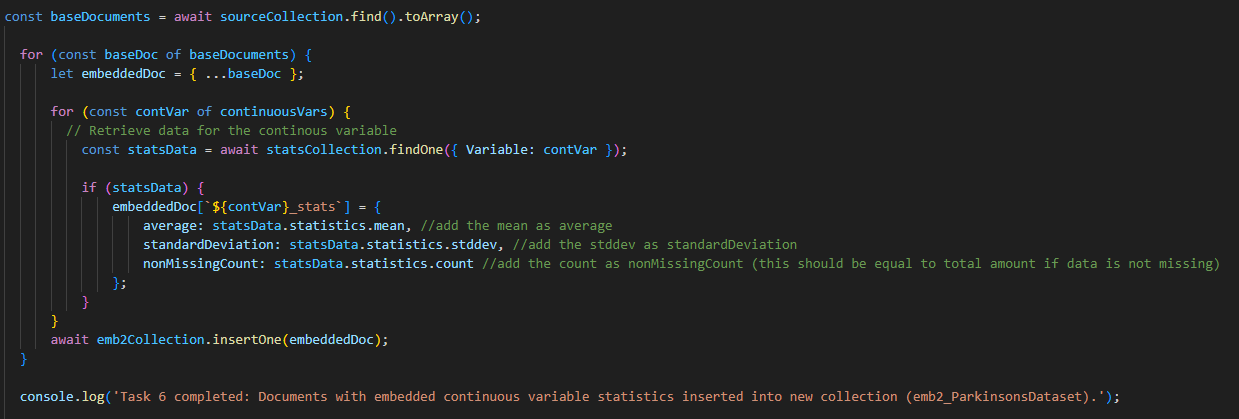
... (spread operator in js) to clone the doc

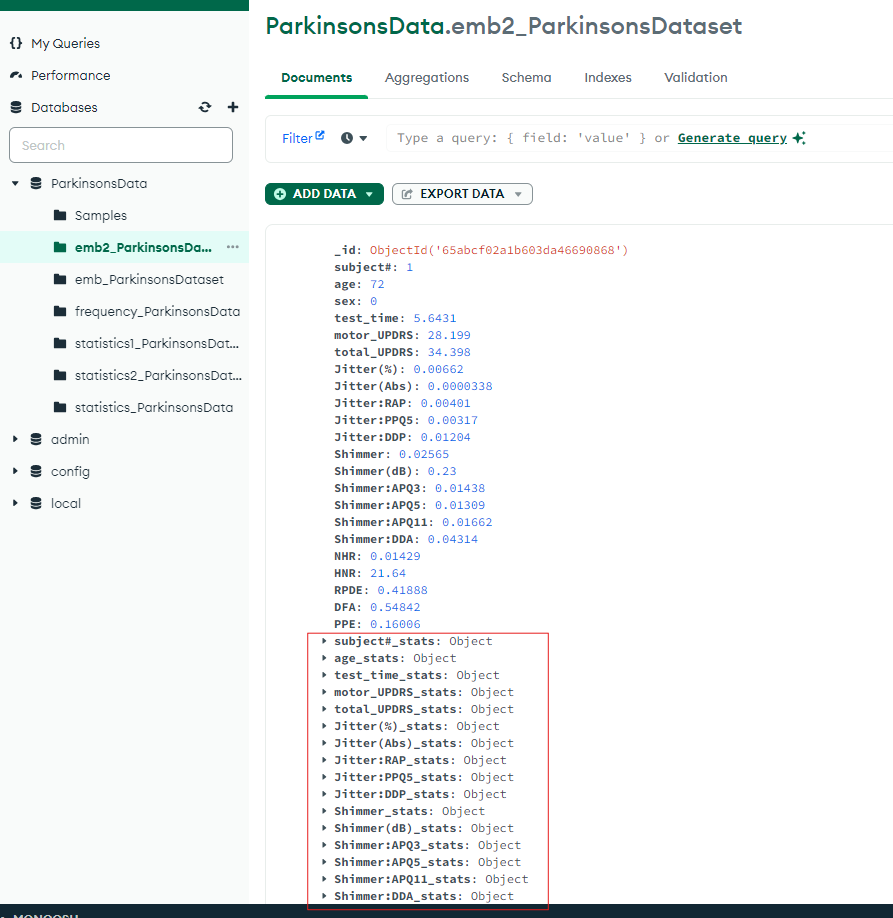


Result shown below:  


Task 6:clone the first document and add to it the mean and standard deviation and count for the continues values and insert it into new collection (emb2).

... (spread operator in js) to clone the doc

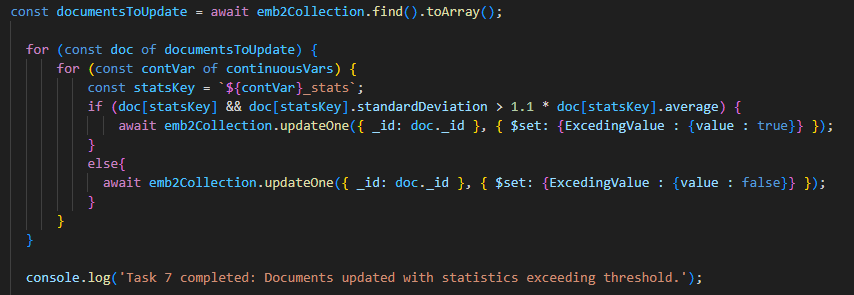


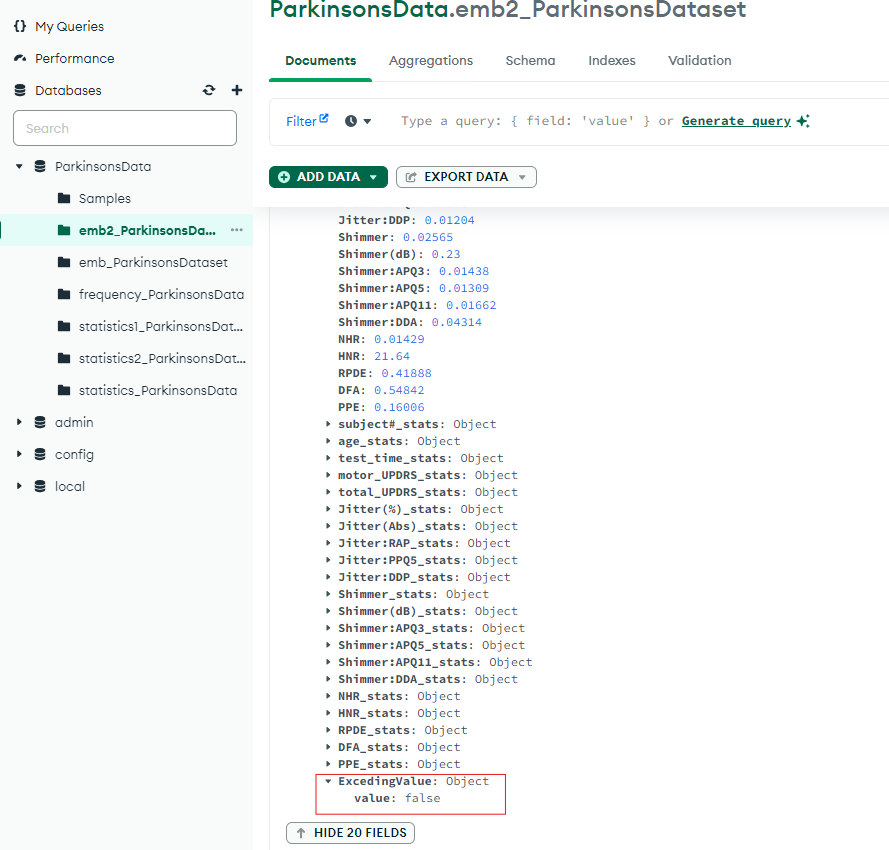
Result shown below:  


Task 7: Itirate over emb2 collection, and check if the standard deviation is 10% bigger then the mean

first we get the collection, and the for every continous attribute we have, we read the value of 1.1\*mean and we compare it to SD, if its bigger, we use $Set to add another attribute to the document which is equal to true, if not its equal to false.

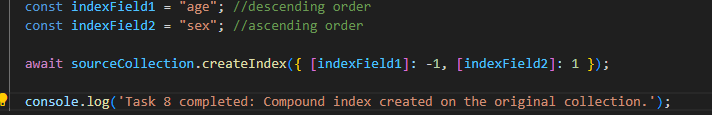
$Set was used according to <https://www.mongodb.com/docs/manual/reference/operator/update/set/#:~:text=the%20details%20document%3A-,db.products.updateOne(,),-After%20updating%2C%20the>

  
  
updating emb2 collection, and checking if value is bigger than 1.1\*mean, then i created a new attribute inside the document which is either true or false:



Task 8: Creating indexs for the collection, in this case i created index for age and for sex attributes

createIndex was used according to <https://www.mongodb.com/docs/manual/reference/method/db.collection.createIndex/#:~:text=Create%20an-,Index%20on%20a%20Multiple%20Fields,-The%20following%20example>

  
  
result shown inside mongodb as follow:

