**Exercicios – MongoDB**

**Link de referência 1**: <https://www.mongodb.com/docs/manual/reference/log-messages/>

1) Os logs no MongoDB são estruturados no formato json. Sendo assim, podem ser importados como uma collection no MongoDB. Importe os logs disponibilizados no link abaixo e escreva o script utilizado para importação

*Link:* [*mongod.log.3.gz*](https://datasidetecnologia-my.sharepoint.com/:u:/g/personal/lucas_souza_dataside_com_br/Eatu695CybJCu-ynypKQWrUBMrD-Dq0rSe8iOIcPl783pw?e=JfQFJj)

**R:** mongoimport --uri="mongodb+srv://acelera:Dts\_2022@testedatabase.eim9i.mongodb.net/exercicio" -c=breno --drop data.json

2) Com base no **link de referência 1**, podemos obter um entendimento da estrutura básica dos documentos importados no exercício 1. Baseado nisso, escreva as consultas abaixo:

* 1. Encontre os documentos onde o campo componente corresponda ao evento “NETWORK”

**R:** db.breno.find({“c”: “NETWORK”}).pretty()

* 1. Encontre os documentos onde o campo componente corresponda a um dos eventos: “ACCESS”, “CONTROL”

**R:** db.breno.find({“$or” :[{“c” : “NETWORK”}, {“c” : “ACCESS”}]}).pretty()fgçfgçlfgfçf

* 1. Encontre os documentos entre as datas **14/03/2022 01:00:00 e 15/03/2022 04:00:00** onde os componentes são “**ACCESS**” e o campo msg é igual a **Successful authentication**

**R:** db.breno.find({“t”: {“$gt: ISODate(“2022-03-14T15:01:00.00+00:00), “$lt”: ISODate(“2022-03-15T15:04:00.00+00:00)}})

* 1. Encontre os documentos onde o componente é igual a COMMAND a mensagem é igual a slow query e plano de execução é igual “COLLSCAN” e a quantidade de documentos examinados é superior ou igual a 1000

**R:** db.breno.find({“c”: “COMMAND”}, {“attr.planSummary”: “COLLSCAN”}, {“keysExamined” : {“$elemMatch”: {“keysExamined”: {“$gte”: 1000}}}})

* 1. Encontre os valores distintos para o campo do componente

**R:** db.breno.distinct(“c”)

* 1. Encontre os documentos que registram as queries que fizeram varredura de coleção apenas para o banco onetoone\_azul\_seguros e apresente apenas os campos ctx e attr.command.filter

**R:** db.breno.find({“attr.ns”: {“$regex”: “onetoone\_azulseguros”}}, {“ctx”: 1, “attr.command.filter” : 1}).pretty()

* 1. Escreva uma query que agrupe a quantidade collscans por banco de dados e imprima a saída em uma collection auxiliar chamada resumo\_collscans.

O resultado deve parecer com o exemplo abaixo:

{

"\_id" : ObjectId("627139c662fe5bdd3885dc96"),

"db"triba : "onetoone\_ncopj",

"qtd" : 2

}

Para essa questão pesquise o comando **$out**

**R:** db.breno.aggregate([{"$match": {"attr.planSummary": {"$eq": "COLLSCAN"}}}, {$group : {“\_id”:{:"$\_id", “db”:"$attr.command.$db"}, “qtd”:{$sum:1}}}, {"$out": "resumo\_collscans"}])

* 1. Exporte a collection resumo\_collscans para um arquivo csv

**R:** sudo mongoexport --uri="mongodb+srv://acelera:Dts\_2022@testedatabase.eim9i.mongodb.net/exercicio" -d=exercicio -c=breno --type=csv -o=/home/brenfern/meuscsv/logs.csv --fields= "\_id,qtd"

OBS: Suba o arquivo no OneDrive e compartilhe o link aqui:

i. agrupe a quantidade de ocorrência de todos os componentes ( campo c ). A saída deve parecer com o exemplo abaixo:

{ "\_id" : "CONNPOOL", "count" : 690 }

{ "\_id" : "NETWORK", "count" : 765192 }

{ "\_id" : "INDEX", "count" : 70 }

{ "\_id" : "WRITE", "count" : 79951 }

{ "\_id" : "-", "count" : 3677 }

{ "\_id" : "CONTROL", "count" : 8 }

{ "\_id" : "COMMAND", "count" : 220966 }

{ "\_id" : "ACCESS", "count" : 428018 }

{ "\_id" : "STORAGE", "count" : 269 }

#### **R:** db.breno.aggregate([{"$group": {"\_id": "$c", "count": {"$sum": 1}}}])