**Mongo DB**

1. **Consulta Flag = Yes**

db.Earthquake.find( { FLAG\_TSUNAMI: "Yes"} )



1. Consulta flag =yes e ( maginitude > 7 ou morte >1)

db.Earthquake.find( {

FLAG\_TSUNAMI: "Yes",

$or: [ { EQ\_MAG\_UNK:{ $gt: “7”} }, { DEATHS: {$gt: “1”} } ]

} )



1. Join Income with PoliceKillingUD

db.PoliceKillingsUS.aggregate(

[{

$lookup: {

from: 'Income',

localField: 'City',

foreignField: 'City',

as: 'inc'

}

}, {

$match: {

"state": "AL"

}

}, {

$project: {

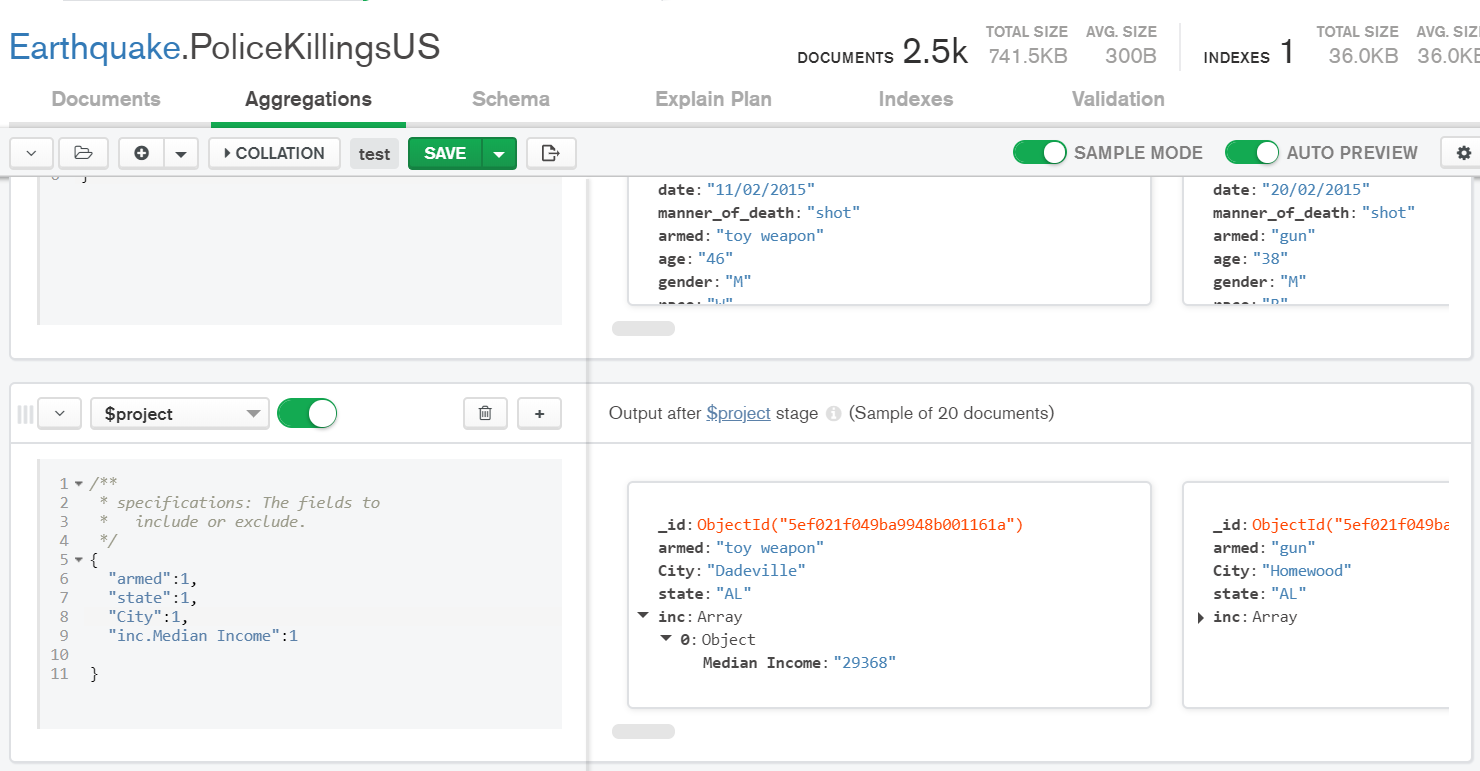
"armed": 1,

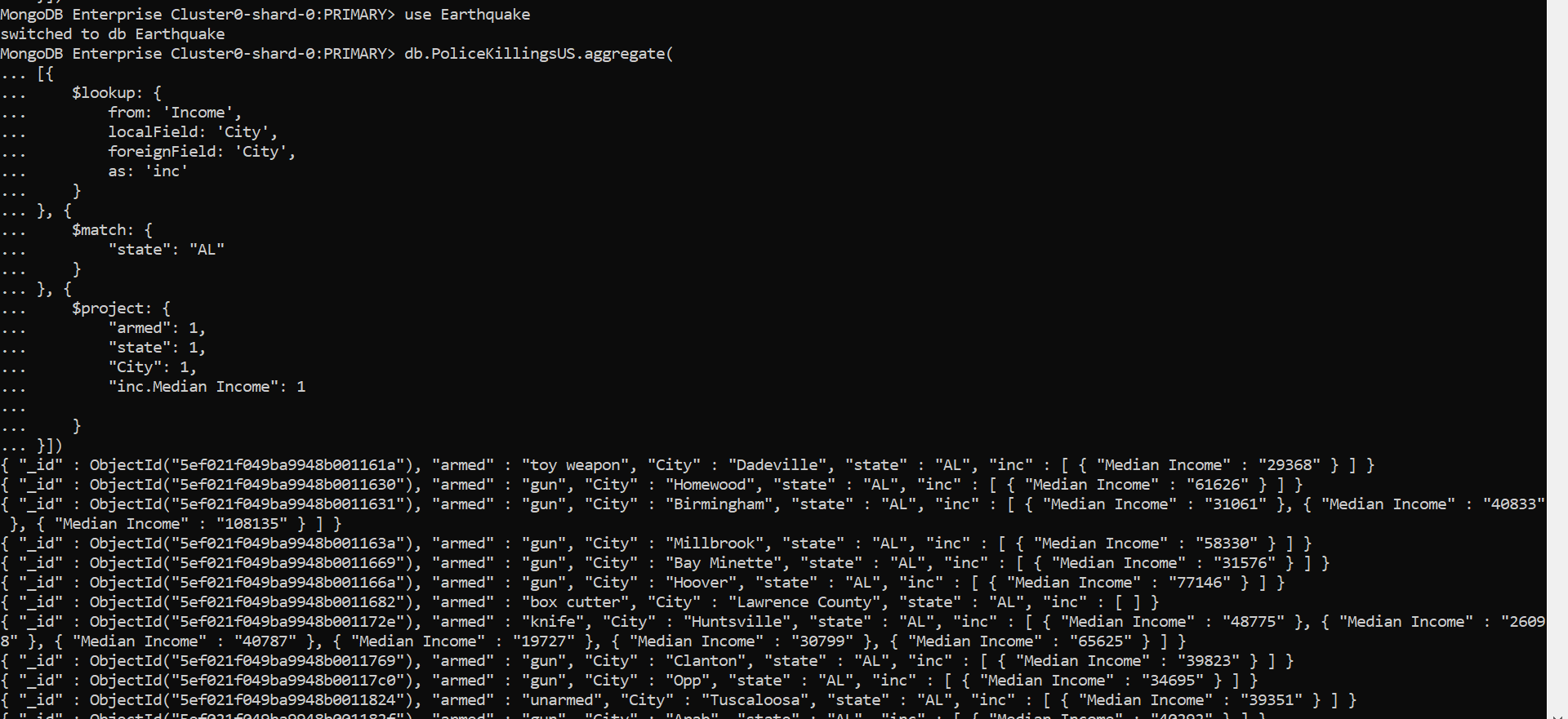
"state": 1,

"City": 1,

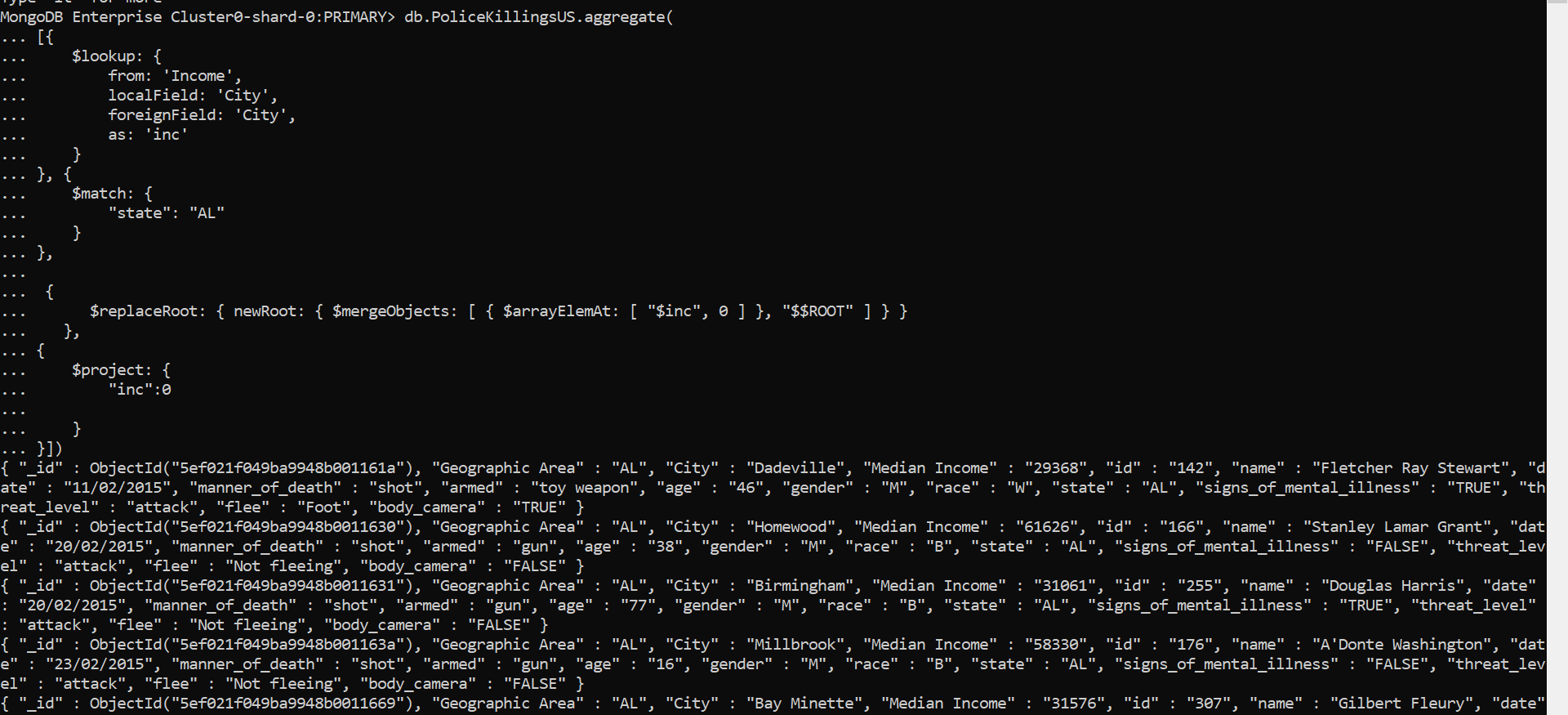
"inc.Median Income": 1

}

}])



**Utilizando merge**



1. **Geospatial Queries**

Transformar o dado no formato correto

db.Earthquake.updateMany(

{},

[

{ $set: { status: "Modified", "location":{coordinates: ["$LONGITUDE", "$LATITUDE"],"type":"Point" }} },

{ $unset: [ "LONGITUDE", "LATITUDE" ] }

]

)

Criar index

db.Earthquake.createIndex({ location: "2dsphere" })

Ver um registro

db.Earthquake.findOne()

Procurar terremotos perto da localização retornada no registro acima (testei com 15 milhas, retorno apenas ele, e com 100 milhas do ponto retornou vários registros)

-15 milhas

db.Earthquake.find({ location:

{ $geoWithin:

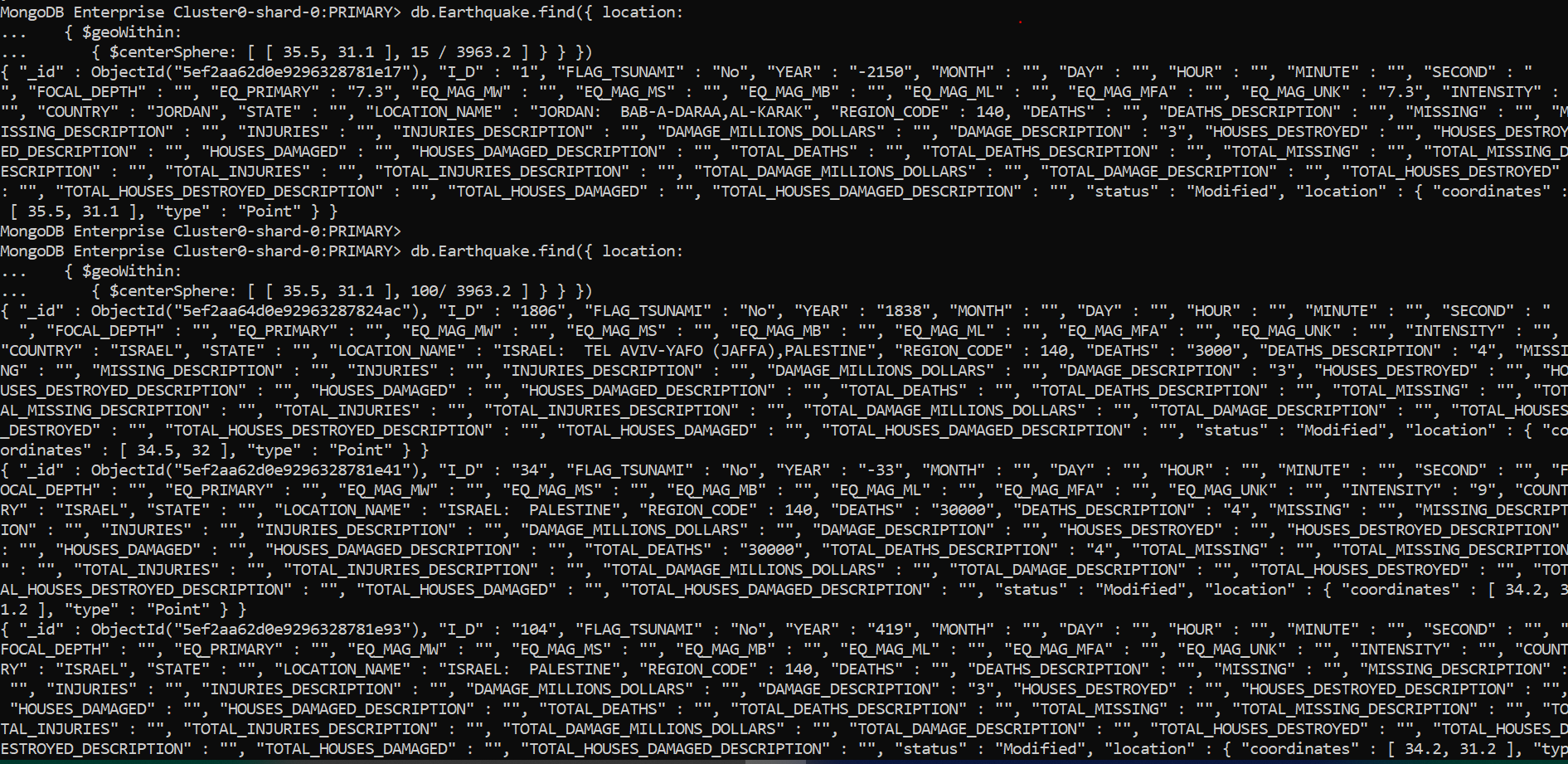
{ $centerSphere: [ [ 35.5, 31.1 ], 15/ 3963.2 ] } } })

- 100 milhas

db.Earthquake.find({ location:

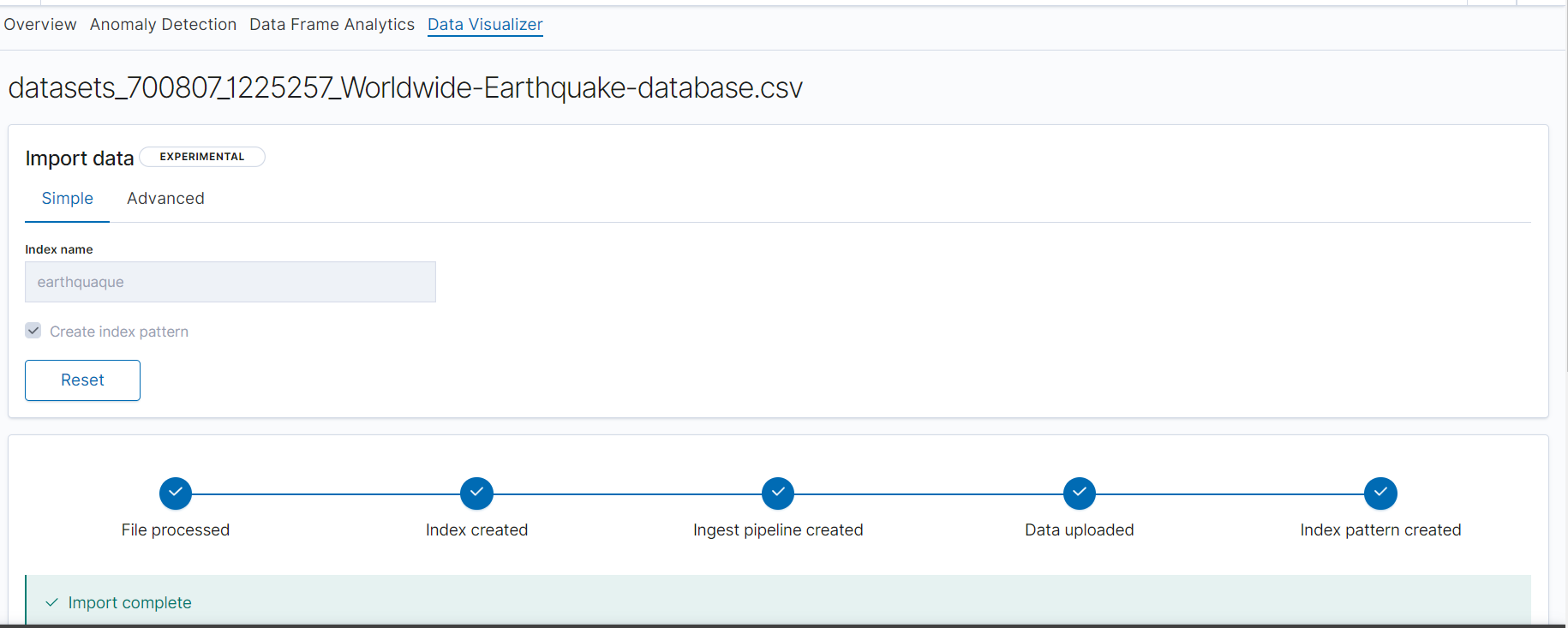
{ $geoWithin:

{ $centerSphere: [ [ 35.5, 31.1 ], 100/ 3963.2 ] } } })

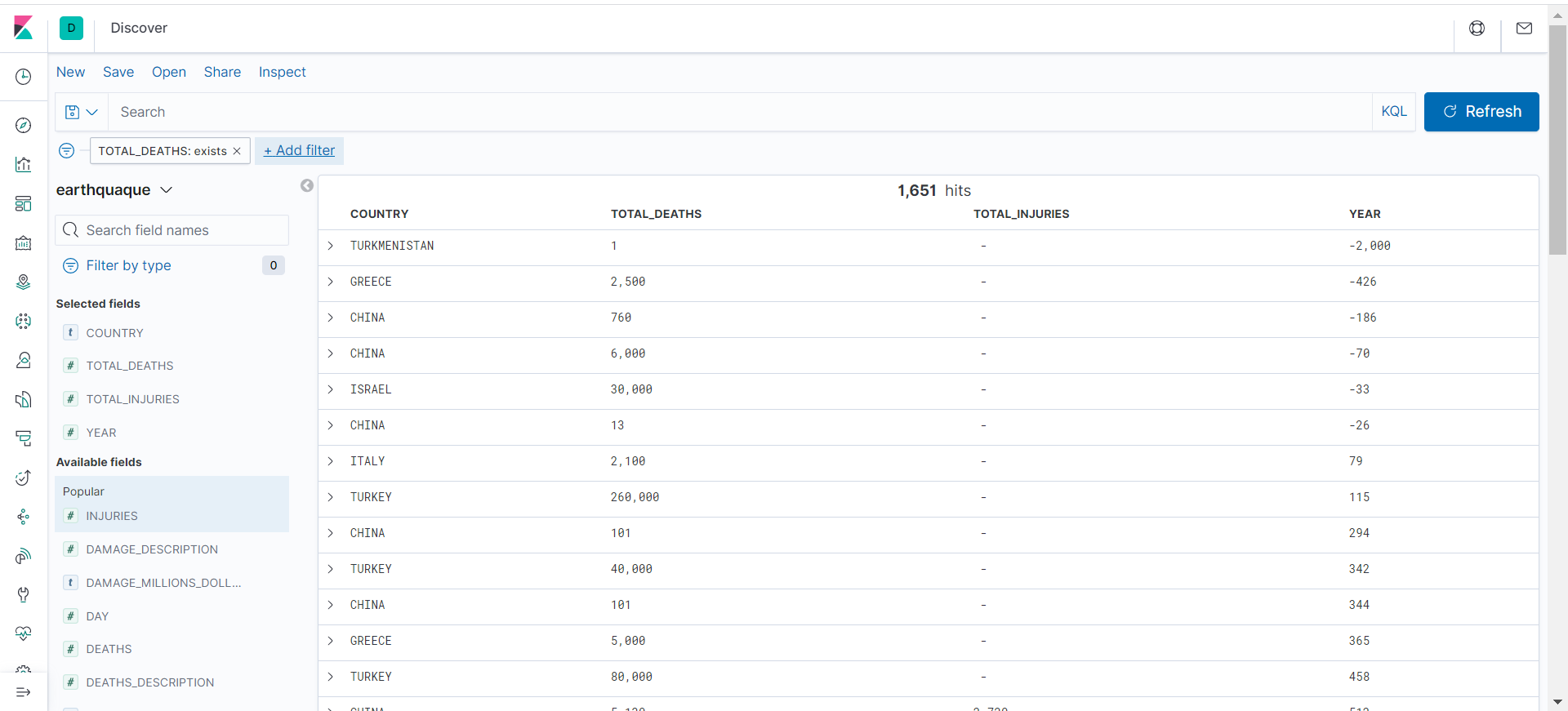


**Elastik Search**

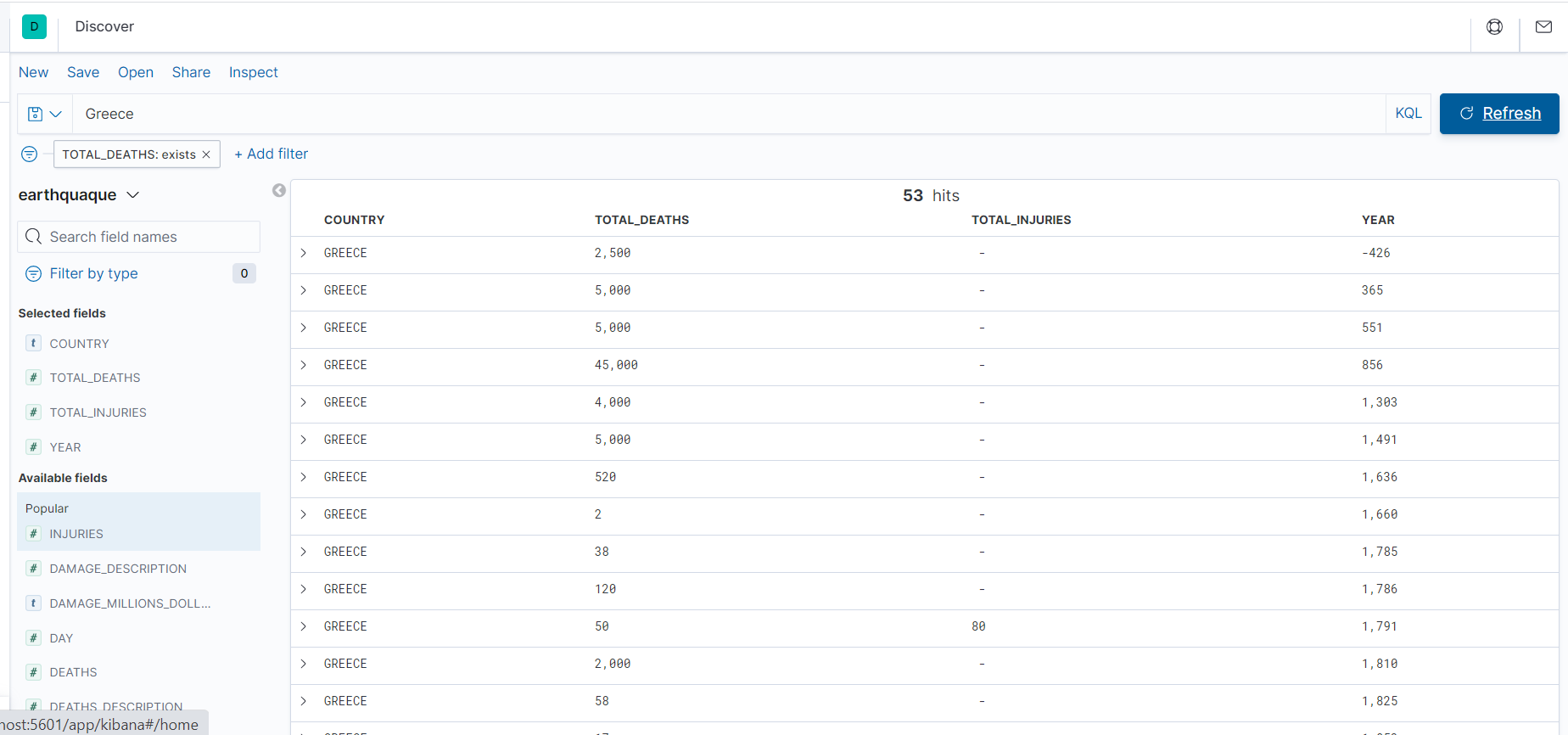
Arquivo Earthquake importado



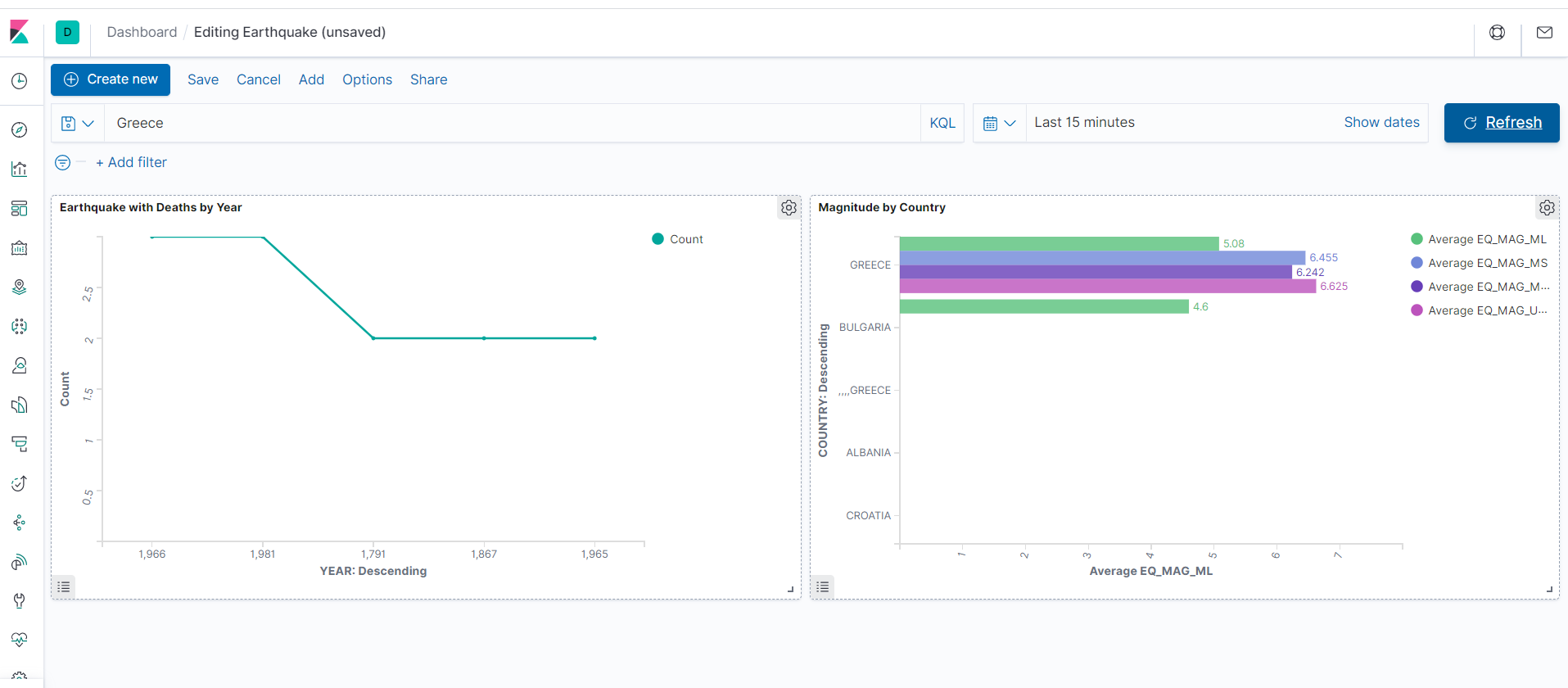
Explorando o Discover, selecionando campos e fazendo filtros.



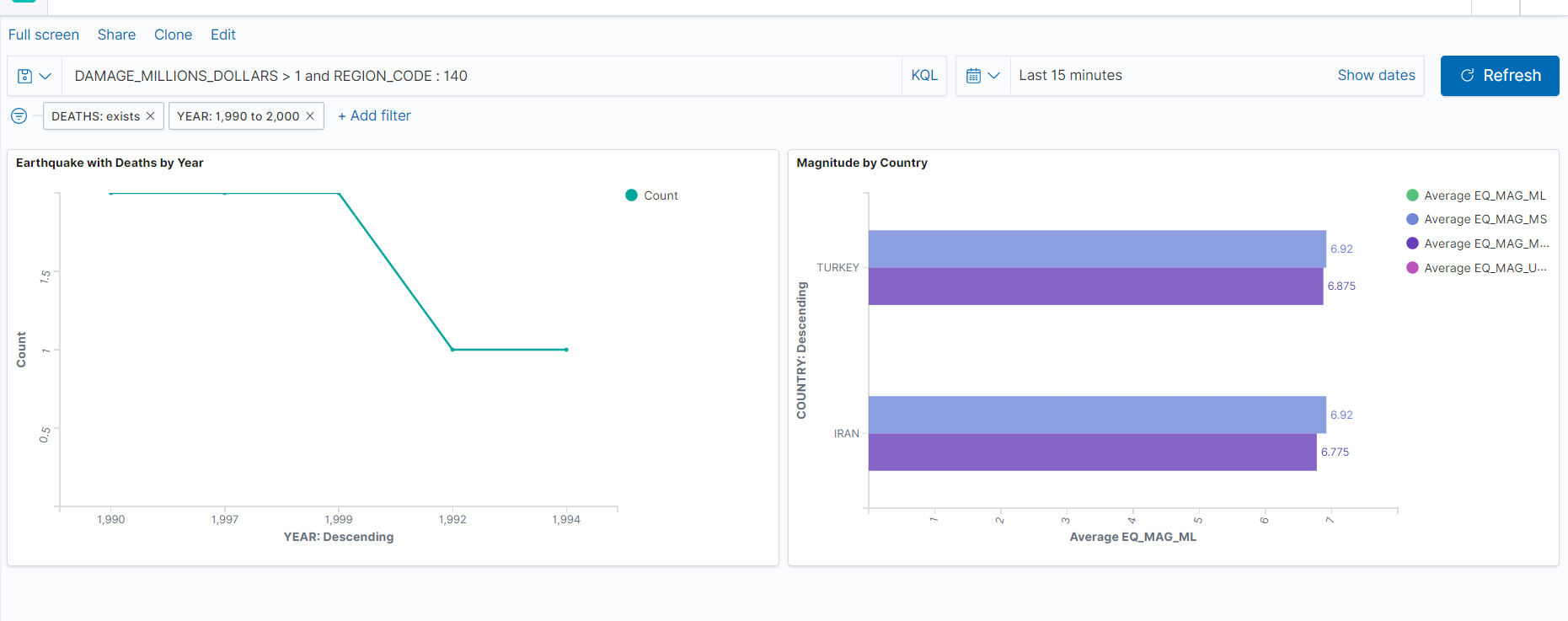
Filtrando a palavra Grécia



Criando Dashboard e filtrando o pais Grécia



Terremotos com danos acima de 1 milhão de dólares na região 140 com mortes entre os anos de 1990 e 2000.



**Neo4j**

**Dados utilizados do Kaggle**

* <https://www.kaggle.com/kwullum/fatal-police-shootings-in-the-us?select=MedianHouseholdIncome2015.csv>
* <https://www.kaggle.com/mohitkr05/global-significant-earthquake-database-from-2150bc>