

Exercici 1:

Codi python per a importar les dades de l'arxiu csv a la base de dades:

```
import pymongo
import csv

client =
pymongo.MongoClient("mongodb://localhost:27017/")
db = client["localitzacions"]
col = db["transactions"]

with open('comtrade.csv', 'r') as f:
    reader = csv.reader(f, delimiter=";")
    header = next(reader)
    for row in reader:
        doc = {}
        for i in range(len(header)):
            try:
                doc[header[i]] = float(row[i])
            except ValueError:
                doc[header[i]] = row[i]
            #print(doc)

        col.insert_one(doc)

print("Collection imported successfully")
```

1. db.transactions.find({}, { _id: 0, year: 1, reporter: 1, partner: 1, value: 1 })
2. db.transactions.updateMany({}, { \$set: { unit: "litres" } })
3. db.transactions.deleteMany({ reporter: "Mexico", partner: "Italy" })
4. db.transactions.updateMany({ wty: 0 }, { \$set: { wty: 1000 } })

Exercici 2:

- 1) Creu una col·lecció “espanya” a la bd localitzacions a partir del fitxer geonames_ES.json que trobaràs al CV i realitzeu les següents consultes. A l'hora d'importar les dades considereu les següents restriccions:

```
import json

from pymongo import MongoClient

client = MongoClient('localhost', 27017)

db = client.localitzacions

collection = db.espanya

json_file = 'geonames_ES.json'

with open(json_file, encoding="utf-8") as file:

    data = json.load(file)

    for item in data:

        item['_id'] = item['geonameid']

        item['localització'] = {

            'type': 'Point',

            'coordinates': [item['latitud'], item['long']]

        }

        del item['geonameid']

        del item['latitud']

        del item['long']
```

```
collection.insert_many(data)

print("Dades importades correctament a la base de dades
MongoDB amb les restriccions aplicades.")
```

- 2) Feu les següents consultes sense definir cap índex i poseu el resultat d'aplicar el pla d'execució en mode executionStats

d) Localitzacions amb nom Barcelona:

```
executionStats: {
  executionSuccess: true,
  nReturned: 3,
  executionTimeMillis: 26,
  totalKeysExamined: 0,
  totalDocsExamined: 79499,
  executionStages: {
    stage: 'COLLSCAN',
    filter: {
      name: {
        '$eq': 'Barcelona'
      }
    },
    nReturned: 3,
    executionTimeMillisEstimate: 0,
    works: 79500,
    advanced: 3,
    needTime: 79496,
    needYield: 0,
    saveState: 79,
    restoreState: 79,
    isEOF: 1,
    direction: 'forward',
    docsExamined: 79499
  },
  command: {
    find: 'espanya',
```

```
filter: {  
  name: 'Barcelona'  
},  
'$db': 'localitzacions'  
},
```

e) Localitzacions per sota de 500m d'elevació respecte el nivell del mar:

```
executionStats: {  
  executionSuccess: true,  
  nReturned: 71391,  
  executionTimeMillis: 26,  
  totalKeysExamined: 0,  
  totalDocsExamined: 79499,  
  executionStages: {  
    stage: 'COLLSCAN',  
    filter: {  
      elevation: {  
        '$lt': 500  
      }  
    },  
    nReturned: 71391,  
    executionTimeMillisEstimate: 0,  
    works: 79500,  
    advanced: 71391,  
    needTime: 8108,  
    needYield: 0,  
    saveState: 79,  
    restoreState: 79,  
    isEOF: 1,  
    direction: 'forward',  
    docsExamined: 79499  
  }  
},  
command: {  
  find: 'espanya',  
  filter: {  
    elevation: {  
      '$lt': 500  
    }  
  },  
  '$db': 'localitzacions'  
},
```

- 3) Per cadascuna de les consultes anteriors, construïu un índex que s'utilitzi i poseu el resultat d'aplicar el pla d'execució en mode executionStats. Compareu els resultats obtinguts amb els de l'Exercici 2:

a) Localitzacions amb nom Barcelona:

```
db.espanya.createIndex({ "name": 1 })
```

Results:

```
executionStats: {  
  executionSuccess: true,  
    nReturned: 3,  
    executionTimeMillis: 6,  
    totalKeysExamined: 3,  
    totalDocsExamined: 3,  
    executionStages: {  
      stage: 'FETCH',  
      nReturned: 3,  
      executionTimeMillisEstimate: 0,  
      works: 4,  
      advanced: 3,  
      needTime: 0,  
      needYield: 0,  
      saveState: 0,  
      restoreState: 0,  
      isEOF: 1,  
      docsExamined: 3,  
      alreadyHasObj: 0,  
      inputStage: {  
        stage: 'IXSCAN',  
        nReturned: 3,  
        executionTimeMillisEstimate: 0,  
        works: 4,  
        advanced: 3,  
        needTime: 0,  
        needYield: 0,  
        saveState: 0,  
        restoreState: 0,  
        isEOF: 1,  
        keyPattern: {  
          name: 1  
        },  
      },  
    },  
  },  
}
```

```
indexName: 'name_1',
isMultiKey: false,
multiKeyPaths: {
  name: []
},
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds: {
  name: [
    '["Barcelona", "Barcelona"]'
  ]
},
keysExamined: 3,
seeks: 1,
dupsTested: 0,
dupsDropped: 0
}
}
},
```

b) Localitzacions per sota de 500m d'elevació respecte el nivell del mar:

```
executionStats: {
  executionSuccess: true,
  nReturned: 71391,
  executionTimeMillis: 72,
  totalKeysExamined: 71391,
  totalDocsExamined: 71391,
  executionStages: {
    stage: 'FETCH',
    nReturned: 71391,
    executionTimeMillisEstimate: 2,
    works: 71392,
    advanced: 71391,
    needTime: 0,
    needYield: 0,
    saveState: 71,
    restoreState: 71,
    isEOF: 1,
    docsExamined: 71391,
    alreadyHasObj: 0,
    inputStage: {
      stage: 'IXSCAN',
      nReturned: 71391,
```

```
executionTimeMillisEstimate: 2,  
works: 71392,  
advanced: 71391,  
needTime: 0,  
needYield: 0,  
saveState: 71,  
restoreState: 71,  
isEOF: 1,  
keyPattern: {  
  elevation: 1  
},  
indexName: 'elevation_1',  
isMultiKey: false,  
multiKeyPaths: {  
  elevation: []  
},  
isUnique: false,  
isSparse: false,  
isPartial: false,  
indexVersion: 2,  
direction: 'forward',  
indexBounds: {  
  elevation: [  
    '[-inf.0, 500)'  
  ]  
},  
keysExamined: 71391,  
seeks: 1,  
dupsTested: 0,  
dupsDropped: 0  
}  
}
```

Hi ha millores en el temps d'execució i en el nombre de documents examinats. Els índexs ens permeten accedir als documents de manera més eficient en funció dels camps especificats en les consultes.

- 4) Feu les següents consultes (si cal construir cap índex, ho feu) i poseu el resultat d'aplicar el pla d'execució en mode executionStats. Per totes les consultes indiqueu: si utilitzen cap índex, l'índex; el nombre de documents que retornen; els que examinen; i el temps d'execució.
 - a) Localitzacions en un radi de 5 km al voltant de les coordenades (2.15899, 41.38879)

Consulta:

```
db.espanya.find({
  "localització": {
    $near: {
      $geometry: {
        type: "Point",
        coordinates: [2.15899, 41.38879]
      },
      $maxDistance: 5000
    }
  }
}).explain("executionStats")
```

S'ha utilitzat un índex geoespacial
totalDocsExamined: 79000
nReturned: 20501
executionTimeMillis: 23

b) db.espanya.find({"feature class": "S"}).explain("ExecutionStats")
totalDocsExamined: 79000
nReturned: 26069
executionTimeMillis: 83

c)

Consulta:

```
db.espanya.find({
  "localització": {
    $near: {
      $geometry: {
        type: "Point",
        coordinates: [2.15899, 41.38879]
      },
      $maxDistance: 5000
    }
  },
  "elevation": { $lt: 500 }
}).explain("executionStats")
```

totalDocsExamined: 79000
nReturned: 15900
executionTimeMillis: 50

d)

Consulta:

```
db.espanya.find({
  "localització": {
    $near: {
      $geometry: {
        type: "Point",
        coordinates: [2.15899, 41.38879]
      },
      $maxDistance: 5000
    }
  },
  "feature class": "H",
  "feature code": "HTL"
}).explain("executionStats")
```

S'utilitza un índex que combina l'índex geoespacial i la identificació dels hostals, s'han retornat 23 documents, examinat 103 i el temps d'execució ha estat de 23 ms

- 5) Afegeix un atribut "tipus" amb valor 'Hostal' a totes les localitzacions de la consulta b) de l'Exercici 4. (utilitza updateMany)

Consulta:

```
db.espanya.updateMany(
  {
    "feature class": "S",
  },
  {
    $set: { "tipus": "Hostal" }
  }
)
```

- 6) Amb el nou atribut, es pot trobar els hostals que estan a 5km al voltant de les coordenades (2.15899, 41.38879)? En cas afirmatiu, mostreu el pla d'execució en mode executionStats i en cas negatiu expliqueu perquè no. A més, indiqueu: si utilitza cap índex, l'índex; el nombre de documents que retorna; els que examina; i el temps d'execució.

Creem aquest índex:

```
db.espanya.createIndex({  
  "tipus": 1,  
  "localització": "2dsphere"  
})
```

```
db.espanya.find({  
  "tipus": "Hostal",  
  "localització": {  
    $near: {  
      $geometry: {  
        type: "Point",  
        coordinates: [2.15899, 41.38879]  
      },  
      $maxDistance: 5000  
    }  
  }  
}).explain("executionStats")
```

encara que l'índex és adequat no retorna res, documentsreturned = 0.