MongoDB TP CRUD - Correction

I. Emails

I.1. Shell command to import the file in collection “email”

```
> ./mongoimport --db test --collection emails --file <path>/enron.json
Returns:
2016-11-21T14:39:31.648+0100  connected to: localhost
2016-11-21T14:39:32.050+0100  imported 5929 documents
```

I.2. The global number of emails

```
> db.emails.count({})
Returns:
5929
```

I.3. The number of emails sent by address in domain enron.com

```
> db.emails.count({sender : {$regex : "@enron\.com$"}})
Returns:
2431
```

I.4. Get the forwarded emails (subject starting by “FW:”)

```
> db.emails.find({subject : {$regex:"^FW:"}})
Returns:
{ "_id" : ObjectId("52af48b6d55148fa0c19975d") ... }
...
```

I.5. Time to get the result and command to get the time

```
> db.emails.find({subject : {$regex:"^FW:"}}).explain("executionStats")
Returns:
{
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 178,
        "executionTimeMillis" : 7,
        "totalKeysExamined" : 0,
        "totalDocsExamined" : 5929,
        ....
    },
}
```

I.6. Add an index on subject

```
> db.emails.createIndex( { subject: 1 } )
Returns:
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

I.7. Time to get the result and command to get the time after indexing

```
> db.emails.find({subject : {$regex:"^FW:"}}).explain("executionStats")
Returns:
{
    "executionStats" : {
        "executionSuccess" : true,
        "nReturned" : 178,
        "executionTimeMillis" : 6,
        "totalKeysExamined" : 179,
        "totalDocsExamined" : 178,
        "executionStages" : {
            ...
            "inputStage" : {
                ...
                "indexName" : "subject_1",
                ...
                "indexBounds" : {
                    "subject" : [
                        "[\"FW:\", \"FW;\")",
                        "[/^FW:/, /^FW:/]"
                    ]
                },
                ...
            }
        }
    },
}
```

I.8. Find only dates, sender and subject of all messages sent to rosalee.fleming@enron.com

```
> db.emails.find({to : "rosalee.fleming@enron.com"},{_id:false, date:true, sender:true, subject:true})
Returns:
{
  "sender" : "natalie@layfamily.com",
  "date" : "2000-12-12 00:55:00-08:00",
  "subject" : "Fw: Zach's First Christmas with Santa"
}
```

I.9. Remove lizard_ar@yahoo.com from all the email recipient

```
> db.emails.update({},{$pull : {to : "lizard_ar@yahoo.com"}},{multi:true})
Returns:
WriteResult({ "nMatched" : 5929, "nUpserted" : 0, "nModified" : 47 })
```

I.10. Add rob.bradley@enron.com as recipient to all emails sent by rosalee.fleming@enron.com

```
> db.emails.update({sender:"rosalee.fleming@enron.com"},{$push:{to : "rob.bradley@enron.com"}},{multi : true})
Returns:
WriteResult({ "nMatched" : 849, "nUpserted" : 0, "nModified" : 849 })
```

II. ZIP Codes

II.1. List the 10 most populated zones in California and Louisiana

```
> db.zips.find({$or:[{state:"CA"},{state:"LA"}]}).sort({pop:-1}).limit(10);
Returns:
{
  "_id" : "90201",
  "city" : "BELL GARDENS",
  "loc" : [ -118.17205, 33.969177 ],
  "pop" : 99568,
  "state" : "CA"
}
...
```

II.2. Then most populated zones in California and Louisiana ranked 10 to 20

```
> db.zips.find({$or:[{state:"CA"},{state:"LA"}]}).sort({pop:-1}).limit(10).skip(10);
Returns:
{
  "_id" : "91744",
  "city" : "INDUSTRY",
  "loc" : [ -117.934098, 34.029428 ],
  "pop" : 77114,
  "state" : "CA"
}
...
```

II.3. Add a field country with the value USA to all the zips

```
> db.zips.update({},{$set :{country : "USA"}},{multi:true})
Returns:
WriteResult({ "nMatched" : 29353, "nUpserted" : 0, "nModified" : 29353 })
```

II.4. List all zones with more than 100 000 inhabitant located on the west side of meridian 110.

```
> db.zips.find({"loc.0":{$gt:-110},pop:{$gt:50000}})
Returns:
{
  "_id" : "02146",
  "city" : "BROOKLINE",
  "loc" : [ -71.128917, 42.339158 ],
  "pop" : 56614,
  "state" : "MA",
  "country" : "USA"
}
...
```

II.5. The closest zones to coordonates -73.996705, 40.74838:

Penser à vérifier que l'index sur les coordonnées à été créé.

```
> db.zips.createIndex( { loc : "2d" } )
```

```
> db.zips.find( {'loc': {$near : [ -73.996705, 40.74838 ] } } ).limit(1)
Returns:
{
  "_id" : "10001",
  "city" : "NEW YORK",
  "loc" : [ -73.996705, 40.74838 ],
  "pop" : 18913,
  "state" : "NY",
  "country" : "USA" }
```

II.6. The cities that are less than 5km away from -73.996705, 40.74838:

Penser à vérifier que l’index sur les coordonnées à été créé.

```
> db.zips.createIndex( { loc : "2dsphere" } )


> db.zips.distinct("city", {
  'loc': {
    $near : {
      $geometry: {
        type: "Point",
        coordinates: [ -73.996705, 40.74838 ]
      },
      $maxDistance: 5000
    }
  }
})
Returns:
[
  "NEW YORK",
  "HOBOKEN",
  "WEEHAWKEN",
  "JERSEY CITY",
  "BROOKLYN",
  "GUTTENBERG",
  "ASTORIA"
]
```

II.7. The cities that have more than 500 000 inhabitants.

```
> db.zips.aggregate([{ $group: { _id: "$city", totalPop: { $sum: "$pop" } } },
    {$match: { totalPop:{$gte:500000 } } },
    {$sort: { totalPop: -1} } ] )
Returns:
{ "_id" : "CHICAGO", "totalPop" : 2452177 }
{ "_id" : "BROOKLYN", "totalPop" : 2341387 }
{ "_id" : "HOUSTON", "totalPop" : 2123053 }
{ "_id" : "LOS ANGELES", "totalPop" : 2102295 }
...
```