University Of Piraeus
Department of Digital Systems

MongoDB Spatial Data Bulk Insertion

Chorologos

ΓΓΕΚ
ΓΕΝΙΚΗ ΓΡΑΜΜΑΤΕΙΑ
ΕΡΕΥΝΑΣ ΚΑΙ ΚΑΙΝΟΤΟΜΙΑΣ

ΕΛΙΔΕΚ.
Ελληνικό Ίδρυμα
Έρευνας & Καινοτομίας

MongoDB Spatial Data Bulk Insertion (1/4)

Supposing that the MongoDB container is running and we possess a CSV file that contains records that constitute spatial points (longitude, latitude values). We proceed to bulk insertion by the following steps:

☐ Copy the CSV file to the MongoDB container:

```
$docker cp mycsv.csv mongodb:/mycsv.csv
```

☐ Connect to the MongoDB container:

```
$sudo docker exec -it mongodb bash
```

MongoDB Spatial Data Bulk Insertion (2/4)

❑ Run the importing command:

```
$mongoimport  --db=test  --collection=points
--type=csv --file=./mycsv.csv
--host=localhost:27017   --columnsHaveTypes
--fields="longitude.double(),
     latitude.double()"
```

The CSV records will be inserted as documents with two double type fields, named longitude and latitude. The records will be inserted in points collection of test database.

MongoDB Spatial Data Bulk Insertion (3/4)

By default, the batch size of `mongoimport` **tool for bulk insertion is set to 100,000**

❑ To represent the longitude and latitude as points, we transform the fields in GeoJSON objects (this command should run on the mongo shell – run `mongo`), named *location*:

```
$db.points.updateMany({},
    [{"$set":{ "location" : {type:"Point",
    coordinates:["$longitude", "$latitude"]
        }   }   }   ]   )
```

MongoDB Spatial Data Bulk Insertion (4/4)

❑ Delete the longitude and latitude fields, as we have formed the GeoJSON objects

```
$db.points.updateMany({},
     {"$unset":{longitude:1, latitude:1 }})
```

Acknowledgement