Project : selectionnez, remodeler

Match : filtrer

Group : agreger

Sort : trier

Mongo db : find many

$eq : true

db.publis.distinct("type",{})

db.produits.aggregate([{$match:{nom : "Thinkpad X230"}}, {$project : {\_id:1}}])

db.produits.aggregate([{$match:{\_id:ObjectId("5c45a053ca4d40aaaeb30095")}}, {$project:{nom:1}}])

db.produits.aggregate([{$match:{prix:{$gt:1200}}}, {$project:{nom:1}}])

db.produits.find({ultrabook:"true"}).limit(1)

db.produits.find({nom:{$regex:/Mackbook/}}).limit(1)

db.produits.find({nom:{$regex:/^Mackbook/i}}).limit(1)

db.produits.remove({fabriquant:"Apple"})

db.produits.aggregate([{$match:{nom : "Thinkpad X230"}}, {$project : {\_id:1}}])

db.produits.remove({\_id:ObjectId("5c45a053ca4d40aaaeb30095")})

db.facture.insert({numero:"10012A",date:"2013-07-04",nom\_client:"Alexandre Lepetit",courriel:"mo@exemple.com",code\_P:["MACBOOKAIR","APPLESUPPORT"],nom\_P:["Macbook Air","AppleCare 1 an"],prix:[999.99,149.99],quantite:[1,1],total:899.99})

db.facture.insert({numero:"10013A",date:"2013-07-05",nom\_client:"Jacques Berger",courriel:"berger.jacques@lampe.com",code\_P:"LENOVOX230",nom\_P:"Lenovo Thinkpad X230",prix:899.99,quantite:1,total:899.99})

db.facture.find({numero:"10013A"})

db.facture.update({numero:"10013A"}, {$set:{date:"2013-07-03",courriel:"alex@example.com"}})

db.facture.find(code\_P:"LENOVOX230")

db.facture.remove({numero:"10012A"})

mongoimport --host 127.0.0.1 --port 27017 --username username --password password --db test --collection maCollection --type csv --file maCollection.json

Pour plus détail voir, [MongoDB-mongoimport](http://docs.mongodb.org/manual/reference/program/mongoimport/)

db.facture.insert({numero: "10012A", date:"2013-07-04",nom\_client:"Alexandre Lepetit",courriel:"mo@exemple.com",produits:[{code:"MACBOOKAIR",nom:"Macbook Air",prix:999.99,quantite:1},{code:"APPLESUPPORT",nom:"AppleCare 1 an",Prix:149.99,quantite:1}]})

db.facture.find({"produits":{$elemMatch:{"code":"MACBOOKAIR"}}})

Evaluation projet Santé - Crime

Nettoyage des données :

Données de superficie :

Téléchargement du jeu de données csv. Ouverture avec bloc note, suppression des doublons de « ; » et des « ; » dérangeants en en tête. Puis remplacement de tous les « ; » par des « , » et enregistrement avec extension .csv.

Données de population :

Téléchargement du jeu de données csv. Ouverture avec bloc note, remplacement de tous les «.» par « » pour avoir des valeurs en int et enregistrement avec extension .csv.

Données de crimes :

Téléchargement du jeu de données csv. Ouverture avec bloc note, remplacement de tous les «.» par « » pour avoir des valeurs en int et enregistrement avec extension .csv.

Données de police :

Téléchargement du jeu de données csv. Ouverture avec bloc note, remplacement de tous les «.» par « » pour avoir des valeurs en int et enregistrement avec extension .csv.

Données de crimes en prison :

Téléchargement du jeu de données csv. Ouverture avec bloc note, remplacement de tous les «.» par « » pour avoir des valeurs en int et enregistrement avec extension .csv.

Données sécurité sociale :

Téléchargement du jeu de données csv. Ouverture avec bloc note, remplacement de tous les «.» par « » pour avoir des valeurs en int et enregistrement avec extension .csv.

Importation du CSV superficie :

"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db pays --collection superficie --type csv --headerline --file C:\Users\Dell\Desktop\projets\EvalMongoDB\EuropeSuperficie2.csv

Superficie totale de l’Europe :

db.superficie.aggregate({$group: {\_id: '',"Superficie\_(km2)": { $sum:"$Superficie\_(km2)" }}}, {$project: {\_id: 0,"Superficie\_(km2)": "$Superficie\_(km2)"}}.

Plus petit Pays :

db.superficie.aggregate({$group:{\_id:"$États\_membres","total":{$max:"$Superficie\_(km2)"}}},{$sort:{"total":1}},{$limit:1})

Plus grand pays :

db.superficie.aggregate({$group:{\_id:"$États\_membres","total":{$max:"$Superficie\_(km2)"}}},{$sort:{"total":-1}},{$limit:1})

Importation du CSV population :

C:\Users\Dell>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db pays --collection population --type csv --headerline --file C:\Users\Dell\Desktop\projets\EvalMongoDB\population.csv

Passage des Value en int :

db.population.find().forEach( function (x) { x.Value = parseInt(x.Value); db.population.save(x); });

Population de l’Europe :

db.population.aggregate({$group: {\_id: '',"Value": { $sum:"$Value" }}}, {$project: {\_id: 0,"Value": "$Value"}})

Pays le moins peuplé :

db.population.aggregate({$group:{\_id:"$GEO","total":{$min:"$Value"}}},{$sort:{"total":1}},{$limit:1})

Pays le plus peuplé :

* Remove des “:” dans Value

db.population.remove({Value:":"})

db.population.aggregate({$group:{\_id:"$GEO","total":{$max:"$Value"}}},{$sort:{"total":-1}},{$limit:7})

Importation du csv crimes :

C:\Users\Dell>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db pays --collection crimes --type csv --headerline --file C:\Users\Dell\Desktop\projets\EvalMongoDB\crimes.csv

Suppression des “:” dans Values :

db.crimes.remove({Value:":"})

Pays européen avec le plus de cambriolages en 2007

db.crimes.aggregate({$match:{"ICCS":{$regex:/Cambriolage/}, TIME:2007}},{$sort:{"Value":-1}},{$limit:1},{$project:{GEO:1,Value:1,\_id:0}})

Pays européen avec le moins de cambriolages en 2007

db.crimes.aggregate({$match:{"ICCS":{$regex:/Cambriolage/},TIME:2007}},{$sort:{"Value":1}},{$limit:1},{$project:{GEO:1,Value:1,\_id:0}})

Pays européen et année avec le moins de meurtres

db.crimes.aggregate({$match:{"ICCS":{$regex:/Homicide/}}},{$sort:{"Value":1}},{$limit:1},{$project:{GEO:1,TIME:1,\_id:0}})

Pays européen avec le moins de meurtres cumulés (année).

db.crimes.aggregate({$match:{"ICCS":{$regex:/Homicide/}}},{$group:{\_id:{TIME:"$TIME",GEO:"$GEO"},"less\_M\_Y":{$min:"$Value"}}},{$sort:{"less\_M\_Y":1}},{$limit:1},{$project:{less\_M\_Y:1,\_id:1}})

Pays européen dans lequel on risque le moins d’être blessé en 2017:

db.crimes.aggregate([{$match:{TIME:2007,"ICCS":/atteinte/}},{$sort:{"Value":1}},{$limit:1},{$project:{GEO:1,Value:1,\_id:0}}])

Pays européen dans lequel on risque le moins d’être blessé sur toute la période d’analyse:

db.crimes.aggregate({$match:{"ICCS":{$regex:/atteinte/}}},{$group:{\_id:{GEO:"$GEO"},Value:"$Value"}}},{$sort:{"Value":1}},{$limit:1})

Pays européen dans lequel on risque le plus d’être blessé sur toute la période d’analyse:

db.crimes.aggregate({$match:{"ICCS":{$regex:/atteinte/}}},{$group:{\_id:{GEO:"$GEO"},Value:{$sum:"$Value"}}},{$sort:{"Value":-1}},{$limit:1})

Moyenne des effectifs par pays :

db.police.aggregate({$group:{\_id:"$GEO",Value:{$avg:"$Value"}}})

Pays avec l’effectif le plus grand pour l’année en 2002 :

db.police.aggregate({$match:{TIME:2002}},{$group:{\_id:"$GEO","total":{$**sum**:"$Value"}}},{$sort:{"total":-1}},{$limit:1})

Pays le plus carcéral dans le temps

db.crime.aggregate({$group:{\_id:{TIME:"$TIME",GEO:"$GEO"},total:{$max:"$Value"}}},{$sort:{total:-1}},{$group:{\_id:"$\_id.TIME",total:{$first:"$total"},GEO:{$first:"$\_id.GEO"}}})

Pays le moins carcéral dans le temps

db.crime.aggregate({$group:{\_id:{TIME:"$TIME",GEO:"$GEO"},total:{$max:"$Value"}}},{$sort:{total:1}},{$group:{\_id:"$\_id.TIME",total:{$first:"$total"},GEO:{$first:"$\_id.GEO"}}})

Année la plus carcérale :

db.crime.aggregate({$group:{\_id:{TIME:"$TIME"},total:{$sum:"$Value"}}},{$sort:{total:-1}},{$limit:1})

Dépense la plus élevé pour les soins de santé par habitant pour chaque année :

db.ps.remove({Value:":"})

db.ps.aggregate({$match:{"UNIT":{$regex:/habitant/}}},{$group:{\_id:{TIME:"$TIME",GEO:"$GEO"},total:{$max:"$Value"}}},{$sort:{total:-1}},{$group:{\_id:"$\_id.TIME",total:{$first:"$total"},GEO:{$first:"$\_id.GEO"}}

Taux de change monétaire en 2007 pour la Bulgarie :

db.ps.aggregate({$match:{"SPDEPS":{$regex:/Total/},GEO:"Bulgarie",TIME:2007}})

Code en plus :

db.crime.aggregate({$group:{\_id:"$TIME" ,uniqueValues: {$addToSet: "$TIME"},Value:{$max:"$Value"}}})

db.crimes.aggregate({$match:{"ICCS":{$regex:/Cambriolage/}}},{$group:{\_id:"","max\_C":{$max:"$Value"}}},{$sort:{"total":-1}},{$limit:1},{$project:{GEO:1,max\_C:1,\_id:0}})

db.crimes.aggregate({$match:{"ICCS":{$regex:/Homicide/}}},{$group:{\_id:"$TIME",GEO:"$GEO","less\_M\_Y":{$min:"$Value"}}},{$sort:{"less\_M\_Y":-1}},{$limit:1},{$project:{GEO:1,less\_M\_Y:1,\_id:1}})