

Rapport De Projet

Bases de données avancées

BOUROUINA Rania
181831052716

CHIBANE Ilies
181831072041

1 Partie I : Importation de la BD

Pour commencer, il faudrait d'abord installer MongoDB (comme nous l'avons fait en TP), et faire les configurations nécessaires en rajoutant une variable d'environnement et en créant les fichiers de données nécessaires.

```
C:\Users\Ranya BR>mongo
{"t":{"$date":"2022-04-27T06:16:11.366+01:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}}}}
{"t":{"$date":"2022-04-27T06:16:11.368+01:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"main", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-04-27T06:16:11.849+01:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main", "msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-27T06:16:11.850+01:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"main", "msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-04-27T06:16:11.852+01:00"},"s":"W", "c":"ASIO", "id":22601, "ctx":"main", "msg":"No TransportLayer configured during NetworkInterface startup"}
{"t":{"$date":"2022-04-27T06:16:11.852+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationDonorService", "ns":"config.tenantMigrationDonors"}}
{"t":{"$date":"2022-04-27T06:16:11.852+01:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationRecipientService", "ns":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2022-04-27T06:16:11.852+01:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"main", "msg":"Multi threading initialized"}
```

On utilise cette commande pour vérifier l'existence du fichier data.

```
C:\Users\Ranya BR>mongo
MongoDB shell version v5.0.8
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("e7b96b7c-7e9c-4a36-bd6c-5567ca4478c5") }
MongoDB server version: 5.0.8
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
https://community.mongodb.com
---
The server generated these startup warnings when booting:
```

```
PS C:\Users\Ranya BR> mongosh
Current Mongosh Log ID: 6268fe116d66d4745b08b39
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.3.1
Using MongoDB: 5.0.8
Using Mongosh: 1.3.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting:
  2022-04-26T11:05:30.168+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
test>
```

Après avoir téléchargé MongoDB Shell, on lance la commande mongosh pour pouvoir exécuter des commandes

1.1 Téléchargement du fichier et création d'une base de données BDD et une collection world

MongoDB Compass est un GUI qui nous facilite les tâches de création de base de données, importation de données et création de collection. Cependant, pour ce TP, nous allons utiliser la ligne de commandes.

```

>_MONGOSH

> show dbs
< admin    41 kB
  config  111 kB
  local   41 kB
test>

```

FIGURE 1 – Les bases de données avant la création

```

> use BDD
< 'switching to db BDD'
BDD>

```

FIGURE 2 – Création de BDD

```

> show collections
<
BDD>

```

FIGURE 3 – Liste des collections avant la création de world

```

→ > db.createCollection("world")
< { ok: 1 }
BDD>

```

FIGURE 4 – Création de la collection world

1.2 Importation des données dans cette collection

```

PS C:\Users\Ranya BR> mongoimport -d BDD -c world --drop --file D:\M1\s2\bda\Projet-BDA\Projet-BDA\world-mongodb.json
2022-04-27T11:11:19.298+0100 connected to: mongodb://localhost/
2022-04-27T11:11:19.301+0100 dropping: BDD.world
2022-04-27T11:11:19.575+0100 239 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\Ranya BR>

```

1.3 Lancement de robo3t.exe pour vérifier les documents

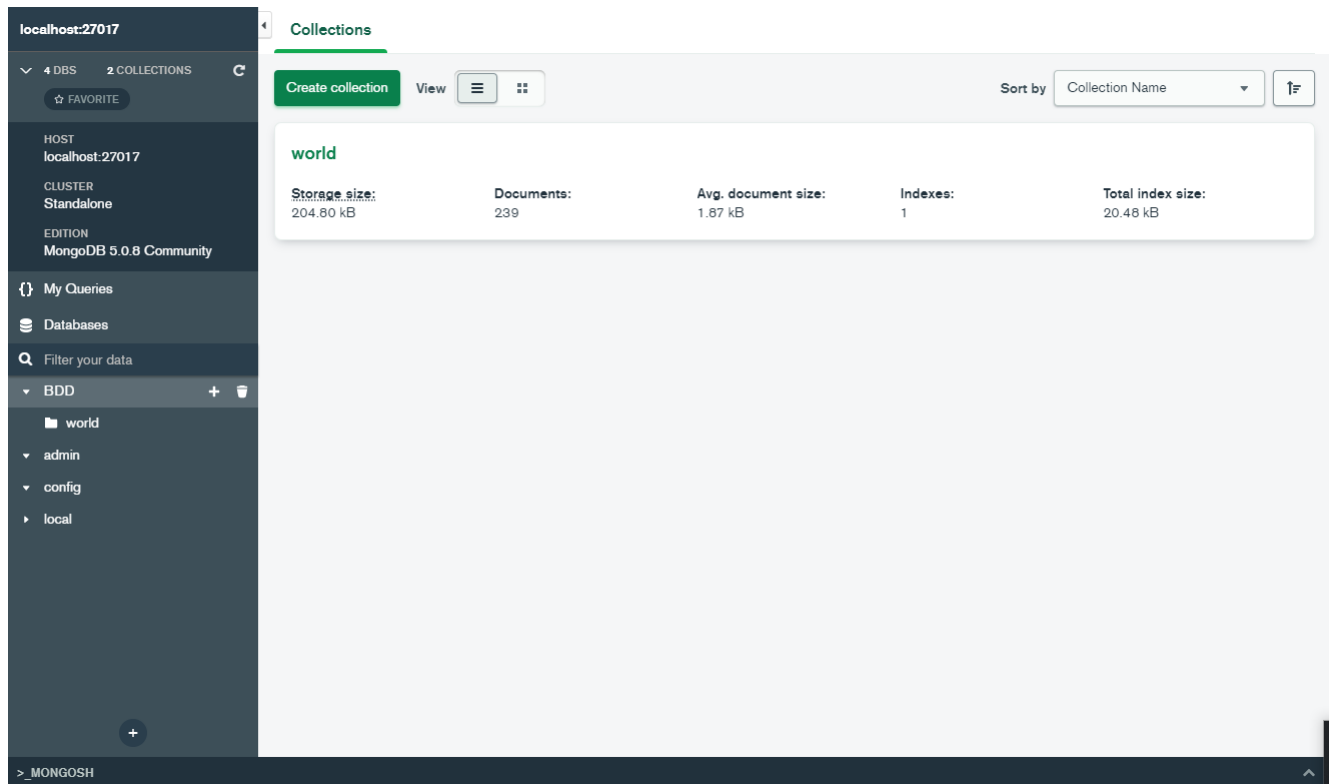


FIGURE 5 – Vérification des documents avec mongoDB Compass

2 Partie II : MongoDB – PyMongo - Python

2.1 1. Déterminer le nombre exact de pays

```
> _MONGOSH
> db.world.distinct("Name").length
< 239
BDD>
```

FIGURE 6 – Execution de la commande avec Mongo Shell

```
1 def QST1(): # Déterminer Le nombre exact de pays
2     return Len(collection.distinct("Name")) #or return collection.distinct("Name").Length
3
```

FIGURE 7 – La fonction Utilisée

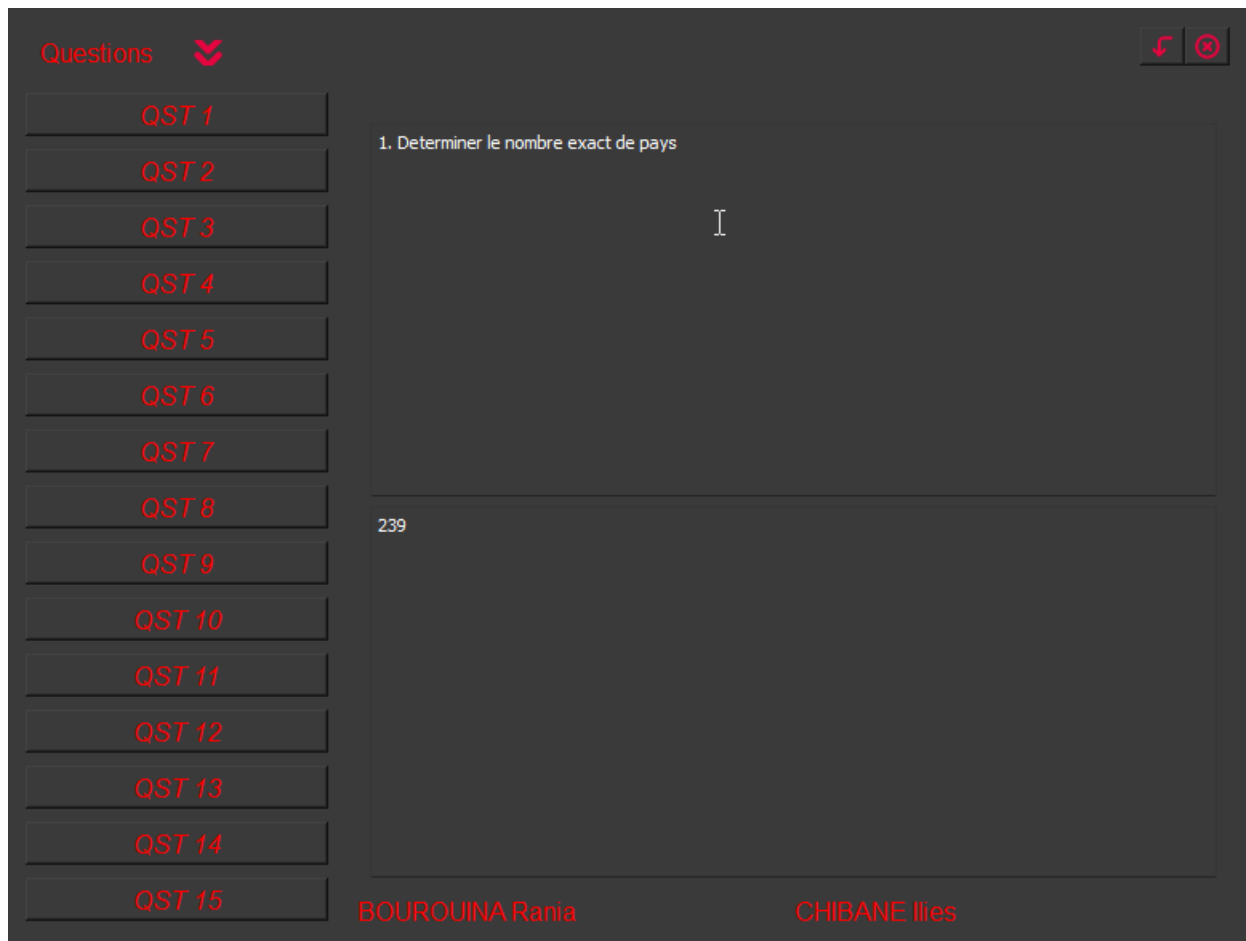


FIGURE 8 – Le résultat sur l'interface

2.2 2. Lister les différents continents

```
> _MONGOSH
> db.world.distinct("Continent")
< [
  'Africa',
  'Antarctica',
  'Asia',
  'Europe',
  'North America',
  'Oceania',
  'South America'
]
BDD>
```

FIGURE 9 – Execution de la commande avec Mongo Shell



```
1 def QST2(): # Lister les différents continents
2     return collection.distinct("Continent")
3
```

FIGURE 10 – La fonction Utilisée

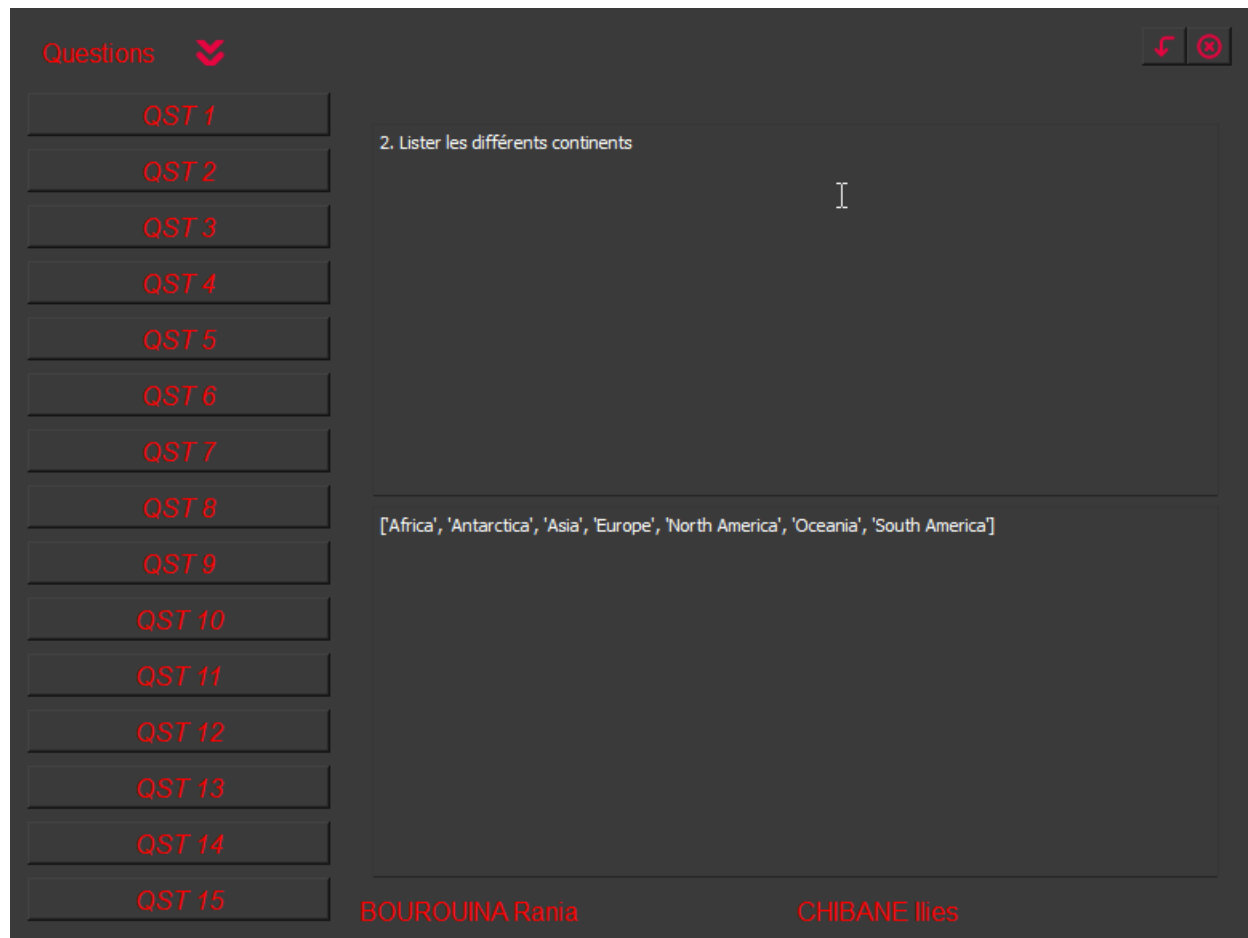


FIGURE 11 – Le résultat sur l'interface

2.3 3. Lister les informations de l'Algérie

```
> _MONGOSH
> db.world.findOne({"Name":"Algeria"})
< { _id: ObjectId("626916c7e77e9c4a05cdf687"),
  Code: 'DZA',
  Name: 'Algeria',
  Continent: 'Africa',
  Region: 'Northern Africa',
  SurfaceArea: 2381741,
  IndepYear: 1962,
  Population: 44700000,
  LifeExpectancy: 69.7,
  GNP: 49982,
  GNPOld: 46966,
  LocalName: 'Al-Jaza'ir/Algérie',
  GovernmentForm: 'Republic',
  HeadOfState: 'Abdelmadjid Tebboune',
  Capital: { ID: 35, Name: 'Alger', District: 'Alger', Population: 2168000 },
  Code2: 'DZ',
  Cities:
    [ { ID: 36, Name: 'Oran', District: 'Oran', Population: 609823 },
      { ID: 37,
        Name: 'Constantine',
        District: 'Constantine',
        Population: 443727 },
      { ID: 38, Name: 'Annaba', District: 'Annaba', Population: 222518 },
      { ID: 39, Name: 'Bordj', District: 'Bordj', Population: 1000000 },
      { ID: 40, Name: 'Blida', District: 'Blida', Population: 1000000 },
      { ID: 41, Name: 'Boumerdes', District: 'Boumerdes', Population: 1000000 },
      { ID: 42, Name: 'Chlef', District: 'Chlef', Population: 1000000 },
      { ID: 43, Name: 'Djelfa', District: 'Djelfa', Population: 1000000 },
      { ID: 44, Name: 'El Oued', District: 'El Oued', Population: 1000000 },
      { ID: 45, Name: 'Guelma', District: 'Guelma', Population: 1000000 },
      { ID: 46, Name: 'Jijel', District: 'Jijel', Population: 1000000 },
      { ID: 47, Name: 'Khenchela', District: 'Khenchela', Population: 1000000 },
      { ID: 48, Name: 'Laghouat', District: 'Laghouat', Population: 1000000 },
      { ID: 49, Name: 'Médéa', District: 'Médéa', Population: 1000000 },
      { ID: 50, Name: 'Mila', District: 'Mila', Population: 1000000 },
      { ID: 51, Name: 'Mostaganem', District: 'Mostaganem', Population: 1000000 },
      { ID: 52, Name: 'Moudania', District: 'Moudania', Population: 1000000 },
      { ID: 53, Name: 'Oum el Bouaghi', District: 'Oum el Bouaghi', Population: 1000000 },
      { ID: 54, Name: 'Saida', District: 'Saida', Population: 1000000 },
      { ID: 55, Name: 'Skikda', District: 'Skikda', Population: 1000000 },
      { ID: 56, Name: 'Tlemcen', District: 'Tlemcen', Population: 1000000 },
      { ID: 57, Name: 'Tizi Ouzou', District: 'Tizi Ouzou', Population: 1000000 },
      { ID: 58, Name: 'Widek', District: 'Widek', Population: 1000000 },
      { ID: 59, Name: 'Zouara', District: 'Zouara', Population: 1000000 } ] }
```

FIGURE 12 – Execution de la commande avec Mongo Shell

```
1 def QST3(): # Lister Les informations de L'Algérie
2     return collection.find_one({"Name": "Algeria"})
3
```

FIGURE 13 – La fonction Utilisée

Questions

QST 1

QST 2

QST 3

QST 4

QST 5

QST 6

QST 7

QST 8

QST 9

QST 10

QST 11

QST 12

QST 13

QST 14

QST 15

3. Lister les informations de l'Algérie

_id	626916c7e77e9c4a05cdf687
Code	DZA
Name	Algeria
Continent	Africa
Region	Northern Africa
SurfaceArea	2381741
IndepYear	1962
Population	44700000
LifeExpectancy	69.7
GNP	49982
GNPold	46966
LocalName	Al-Jaza'ir/Algérie
GovernmentForm	Republic

BOUROUINA Rania

CHIBANE Ilies

FIGURE 14 – Le résultat sur l'interface

2.4 4. Lister les pays du continent Africain, ayant une population inférieure à 100000 habitants

```

> db.world.find({"Continent":"Africa", "Population":{"$lt": 100000}}, {"Name" :1}).sort("Name")
< { _id: ObjectId("626916c7e77e9c4a05cdf6ac"),
  Name: 'British Indian Ocean Territory' }
{ _id: ObjectId("626916c7e77e9c4a05cdf703"),
  Name: 'Saint Helena' }
{ _id: ObjectId("626916c7e77e9c4a05cdf711"),
  Name: 'Seychelles' }
BDD>

```

FIGURE 15 – Execution de la commande avec Mongo Shell


```

1  def QST4(): # Lister Les pays du continent Africain, ayant une population inférieure à 100000 habitants
2      countries = []
3      for post in (collection.find({"Continent": "Africa", "Population": {"$lt": 100000}}, {"Name": 1}).sort("Name")):
4          countries.append(post["Name"])
5      return countries
6

```

FIGURE 16 – La fonction Utilisée

Questions

QST 1

QST 2

QST 3

QST 4

QST 5

QST 6

QST 7

QST 8

QST 9

QST 10

QST 11

QST 12

QST 13

QST 14

QST 15

4. Lister les pays du continent Africain, ayant une population inférieure à 100000 habitants

[British Indian Ocean Territory', 'Saint Helena', 'Seychelles']

BOUROUINA Rania
CHIBANE Ilies

FIGURE 17 – Le résultat sur l'interface

2.5 5. Lister les pays indépendant du continent océanique

```
> db.world.find({"Continent":"Oceania", "IndepYear":{"$ne": "NA"}},{ "Name" :1}).sort("Name")
< { _id: ObjectId("626916c7e77e9c4a05cdf659"), Name: 'Australia' }
  { _id: ObjectId("626916c7e77e9c4a05cdf68c"),
    Name: 'Fiji Islands' }
  { _id: ObjectId("626916c7e77e9c4a05cdf6b7"), Name: 'Kiribati' }
  { _id: ObjectId("626916c7e77e9c4a05cdf6cd"),
    Name: 'Marshall Islands' }
  { _id: ObjectId("626916c7e77e9c4a05cdf691"),
    Name: 'Micronesia, Federated States of' }
  { _id: ObjectId("626916c7e77e9c4a05cdf6e6"), Name: 'Nauru' }
  { _id: ObjectId("626916c7e77e9c4a05cdf6e9"),
    Name: 'New Zealand' }
  { _id: ObjectId("626916c7e77e9c4a05cdf6f2"), Name: 'Palau' }
  { _id: ObjectId("626916c7e77e9c4a05cdf6f3"),
    Name: 'Papua New Guinea' }
  { _id: ObjectId("626916c7e77e9c4a05cdf72c"), Name: 'Samoa' }
  { _id: ObjectId("626916c7e77e9c4a05cdf707"),
    Name: 'Solomon Islands' }
  { _id: ObjectId("626916c7e77e9c4a05cdf719"), Name: 'Tonga' }
  { _id: ObjectId("626916c7e77e9c4a05cdf71d"), Name: 'Tuvalu' }
  { _id: ObjectId("626916c7e77e9c4a05cdf72a"), Name: 'Vanuatu' }
BDD>
```

FIGURE 18 – Execution de la commande avec Mongo Shell

```
1 def QST5(): # Lister Les pays indépendant du continent océanique
2   countries = []
3   for post in (collection.find({"Continent":"Oceania", "IndepYear":{"$ne": "NA"}},{ "Name" :1}).sort("Name")):
4     countries.append(post["Name"])
5   return countries
6
```

FIGURE 19 – La fonction Utilisée

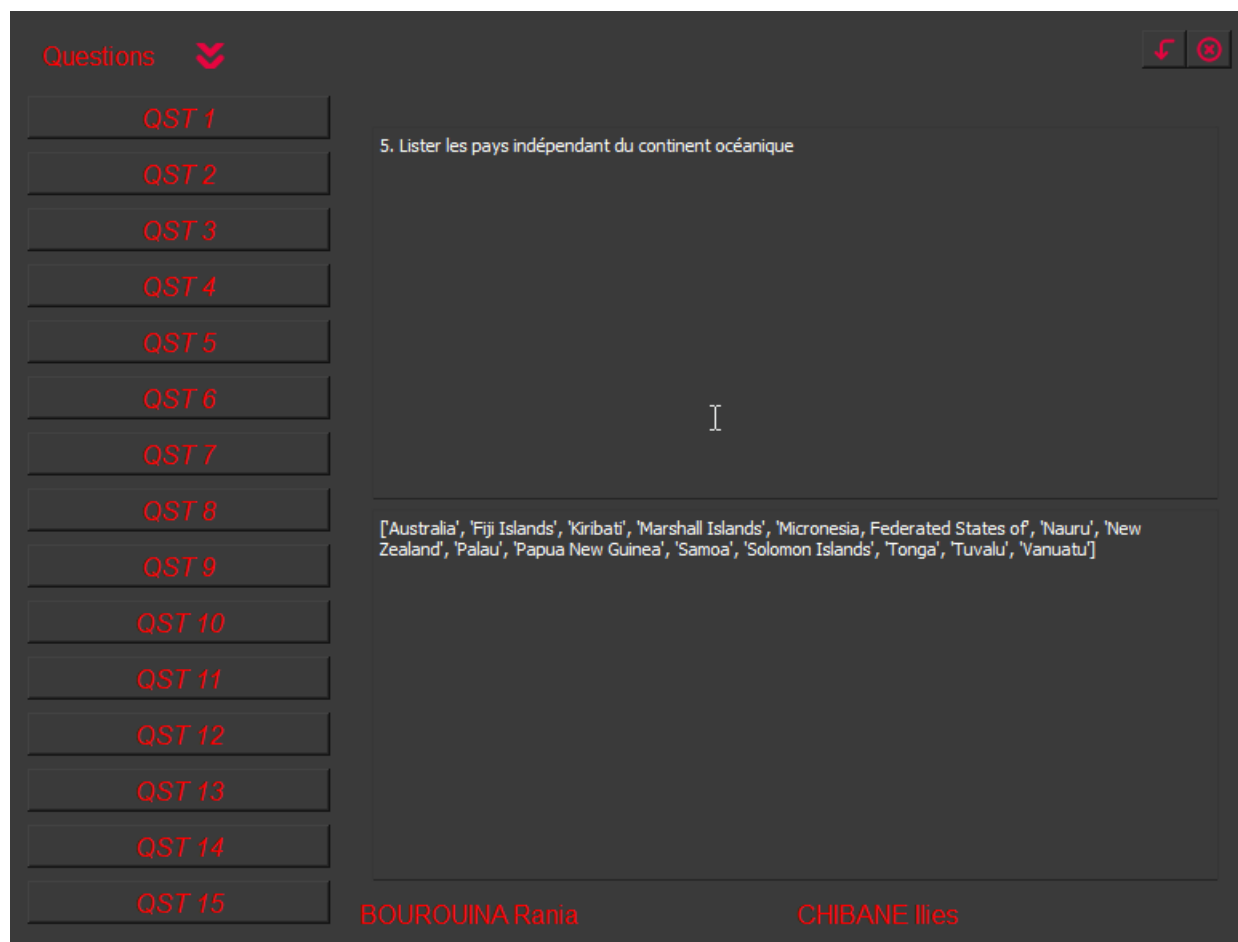


FIGURE 20 – Le résultat sur l'interface

2.6 6. Quel est le plus gros continent en termes de surface ? (un seul continent affiché à la fin)

```
> _MONGOSH

> db.world.find().sort("SurfaceArea", -1).limit(1)
< { _id: ObjectId("626916c7e77e9c4a05cdf728"),
  Code: 'VAT',
  Name: 'Holy See (Vatican City State)',
  Continent: 'Europe',
  Region: 'Southern Europe',
  SurfaceArea: 0.4,
  IndepYear: 1929,
  Population: 1000,
  LifeExpectancy: 'NA',
  GNP: 9,
  GNPOld: 'NA',
  LocalName: 'Santa Sede/Città del Vaticano',
  GovernmentForm: 'Independent Church State',
  HeadOfState: 'Johannes Paavali II',
  Capital:
    { ID: 3538,
      Name: 'Città del Vaticano',
      District: '-',
      Population: 455 },
  Code2: 'VA',
  OffLang: [ { Language: 'Italian', Percentage: 0 } ] }

BDD>
```

FIGURE 21 – Execution de la commande avec Mongo Shell

```
1 def QST6(): # Quel est le plus gros continent en termes de surface ? (un seul continent affiché à la fin)
2   countries = []
3   for post in (collection.find().sort("SurfaceArea", -1).limit(1)):
4     countries.append(post["Name"])
5   return countries
6
```

FIGURE 22 – La fonction Utilisée

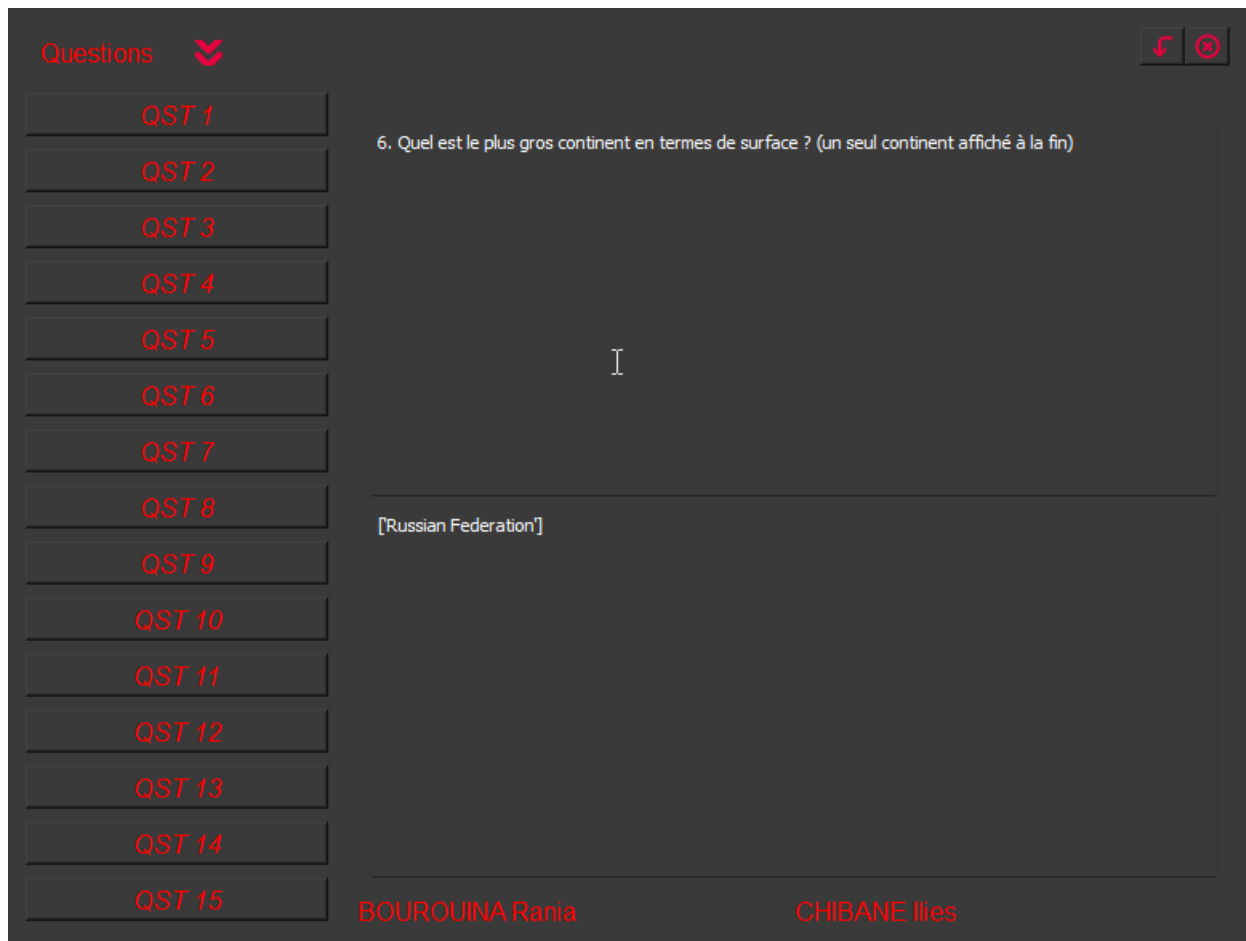


FIGURE 23 – Le résultat sur l'interface

2.7 7. Donner par continents le nombre de pays, la population totale et en bonus le nombre de pays indépendant.

```
> db.world.aggregate( [{ $group: { _id: "$Continent", 'total': { '$sum': '$Population' }, 'pays': { '$sum': 1 } } } ] )
< { _id: 'Oceania', total: 30401150, pays: 28 }
  { _id: 'Asia', total: 3705025700, pays: 51 }
  { _id: 'North America', total: 482993000, pays: 37 }
  { _id: 'Europe', total: 730074600, pays: 46 }
  { _id: 'Antarctica', total: 0, pays: 5 }
  { _id: 'Africa', total: 797704000, pays: 58 }
  { _id: 'South America', total: 345780000, pays: 14 }
BDD>
```

FIGURE 24 – Execution de la commande avec Mongo Shell


Le bonus (nombre de pays indépendants par continent sera rajouté lors de l'implémentation avec python.

```

1 def QST7(): # Donner par continents le nombre de pays, la population totale et en bonus le nombre de pays indépendant.
2     continents_infos = {}
3     for post in collection.distinct("Continent"):
4         continent_infos = {}
5         count = collection.count_documents({"Continent":post})
6         count_indi = collection.count_documents({"Continent":post, "IndepYear":{"$ne": "NA"}})
7         continent_infos["Number of countries"] = count
8         continent_infos["Size of the population"] = 0
9         continent_infos["Number of independent countries"] = count_indi
10        continents_infos[post] = continent_infos
11
12    pipe = [{'$group': {'_id': '$Continent', 'total': {'$sum': '$Population'}}}]
13    for post in collection.aggregate(pipeline=pipe):
14        continents_infos[post["_id"]]["Size of the population"] = post["total"]
15
16    return continents_infos
17

```

FIGURE 25 – La fonction Utilisée

Questions 

QST 1

QST 2

QST 3

QST 4

QST 5

QST 6

QST 7

QST 8

QST 9

QST 10

QST 11

QST 12

QST 13

QST 14

QST 15

7. Donner par continents le nombre de pays, la population totale et en bonus le nombre de pays indépendant.

I

Africa	Number of countries	58
	Size of the population	797704000
	Number of independent countries	53
Antarctica	Number of countries	5
	Size of the population	0
	Number of independent countries	0
Asia	Number of countries	51
	Size of the population	3705025700
	Number of independent countries	47
Europe	Number of countries	46
	Size of the population	730074600
	Number of independent countries	43

BOUROUINA Rania

CHIBANE Ilies

FIGURE 26 – Le résultat sur l'interface

NB : pour avoir le nombre de pays par continent, il suffit de compter le nombre d'occurences du document car le code

du pays est unique dans chacun.

2.8 8. Donner la population totale des villes d'Algérie

```
> _MONGOSH

> db.world.aggregate(
  [
    {$match: {"Name": "Algeria"}},
    {$group: {
      _id: '$Cities',
      'total': {'$sum': '$Population'}}
    ]
  )
< { _id:
  [ { ID: 36, Name: 'Oran', District: 'Oran', Population: 609823 },
    { ID: 37,
      Name: 'Constantine',
      District: 'Constantine',
      Population: 443727 },
    { ID: 38, Name: 'Annaba', District: 'Annaba', Population: 222518 },
    { ID: 39, Name: 'Batna', District: 'Batna', Population: 183377 },
    { ID: 40, Name: 'Sétif', District: 'Sétif', Population: 179055 },
    { ID: 41,
      Name: 'Sidi Bel Abbès',
      District: 'Sidi Bel Abbès',
      Population: 153106 },
    { ID: 42, Name: 'Skikda', District: 'Skikda', Population: 128747 },
    { ID: 43, Name: 'Biskra', District: 'Biskra', Population: 128281 },
    { ID: 44,
```

FIGURE 27 – Execution de la commande avec Mongo Shell

```
1 def QST8(): # Donner la population totale des villes d'Algérie
2   n = collection.find_one({"Name": "Algeria"})["Cities"]
3   count = sum(map(Lambda x: int(x['Population']), n))
4
5   return count
6
```

FIGURE 28 – La fonction Utilisée

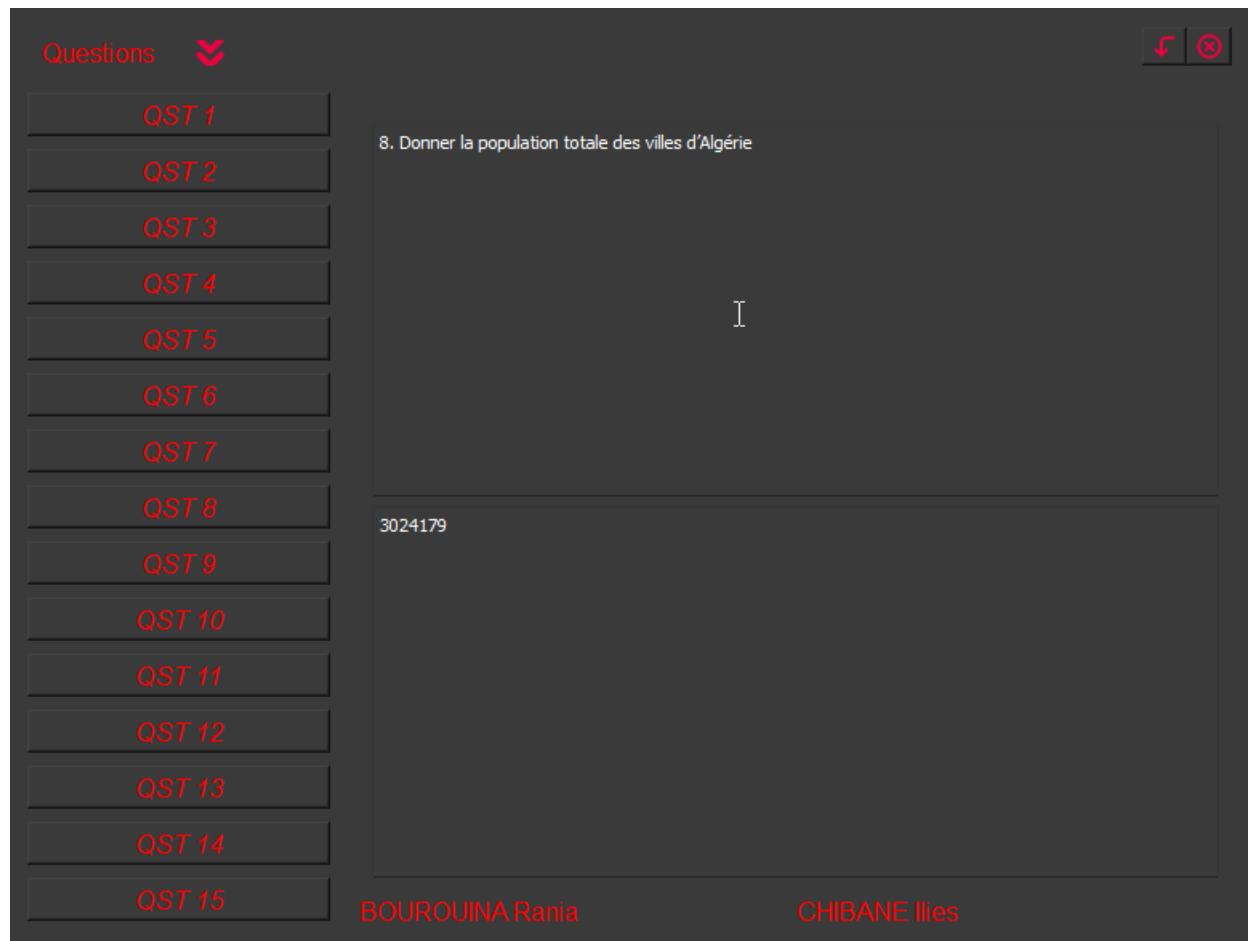


FIGURE 29 – Le résultat sur l'interface


```

>_MONGOSH
{ ID: 45, Name: 'Béjaïa', District: 'Béjaïa', Population: 117162 },
{ ID: 46,
  Name: 'Mostaganem',
  District: 'Mostaganem',
  Population: 115212 },
{ ID: 47,
  Name: 'Tébessa',
  District: 'Tébessa',
  Population: 112007 },
{ ID: 48,
  Name: 'Tlemcen (Tilimsen)',
  District: 'Tlemcen',
  Population: 110242 },
{ ID: 49, Name: 'Béchar', District: 'Béchar', Population: 107311 },
{ ID: 50, Name: 'Tiaret', District: 'Tiaret', Population: 100118 },
{ ID: 51,
  Name: 'Ech-Chleff (el-Asnam)',
  District: 'Chlef',
  Population: 96794 },
{ ID: 52,
  Name: 'Ghardaïa',
  District: 'Ghardaïa',
  Population: 89415 } ],
total: 44700000 }
BDD>

```

FIGURE 30 – Execution de la commande avec Mongo Shell

2.9 9. Donner la capitale (uniquement nom de la ville et population) d’Algérie

```

> db.world.findOne({"Name": "Algeria"}, {"Capital.Name":1, "Capital.Population" :1})
< { _id: ObjectId("626916c7e77e9c4a05cdf687"),
  Capital: { Name: 'Alger', Population: 2168000 } }
BDD>

```

FIGURE 31 – Execution de la commande avec Mongo Shell

```

1  def QST9(): # 9. Donner la capitale (uniquement nom de la ville et population) d'Algérie
2      capital = collection.find_one({"Name": "Algeria"})["Capital"]
3      keys = ["Name", "Population"]
4      capital_info = [capital[key] for key in keys]
5      return capital_info
6

```

FIGURE 32 – La fonction Utilisée

Questions

QST 1

QST 2

QST 3

QST 4

QST 5

QST 6

QST 7

QST 8

QST 9

QST 10

QST 11

QST 12

QST 13

QST 14

QST 15

9. Donner la capitale (uniquement nom de la ville et population) d'Algérie

I

[Alger', 2168000]

BOUROUINA Rania

CHIBANE Ilies

FIGURE 33 – Le résultat sur l'interface

2.10 10. Quelles sont les langues parlées dans plus de 15 pays ?

```
> _MONGOSH

> db.world.aggregate([
  {
    $addFields: { Langue : { $concatArrays: ["$OffLang","$NotOffLang" ] } }
  },
  { $unwind: "$Langue"
  },
  { $group: {
    _id:"$Langue.Language",
    "nb" : { $sum :1}
  }
  },
  { $match: { "nb": { $gt: 15 } } },
])

< { _id: 'German', nb: 17 }
  { _id: 'Spanish', nb: 25 }
  { _id: 'Chinese', nb: 17 }
  { _id: 'Russian', nb: 17 }
  { _id: 'Arabic', nb: 29 }
  { _id: 'French', nb: 18 }
  { _id: 'English', nb: 44 }

BDD>
```

FIGURE 34 – Execution de la commande avec Mongo Shell

```
1 def Q5181(): # 10. Quelles sont les langues parlées dans plus de 15 pays ?
2
3     db.world.aggregate([
4         {
5             $addFields: { Langue : { $concatArrays: ["$OffLang","$NotOffLang" ] } }
6         },
7         { $unwind: "$Langue"
8         },
9         { $group: {
10             _id:"$Langue.Language",
11             "nb" : { $sum :1}
12         }
13         },
14         { $match: { "nb": { $gt: 15 } } }
15     ])
16
17     languages = []
18     pipe = ([ $addFields: { "Langue" : { $concatArrays: ["$OffLang","$NotOffLang" ] } } ], [ $unwind: "$Langue" ], [ $group: { "_id": "$Langue.Language", "Number of Countries" : { $sum :1 } } ], [ $match: { "nb": { $gt: 15 } } ] )
19     for post in collection.aggregate(pipeline(pipe)):
20         languages.append(post)
21     return languages
22
```

FIGURE 35 – La fonction Utilisée

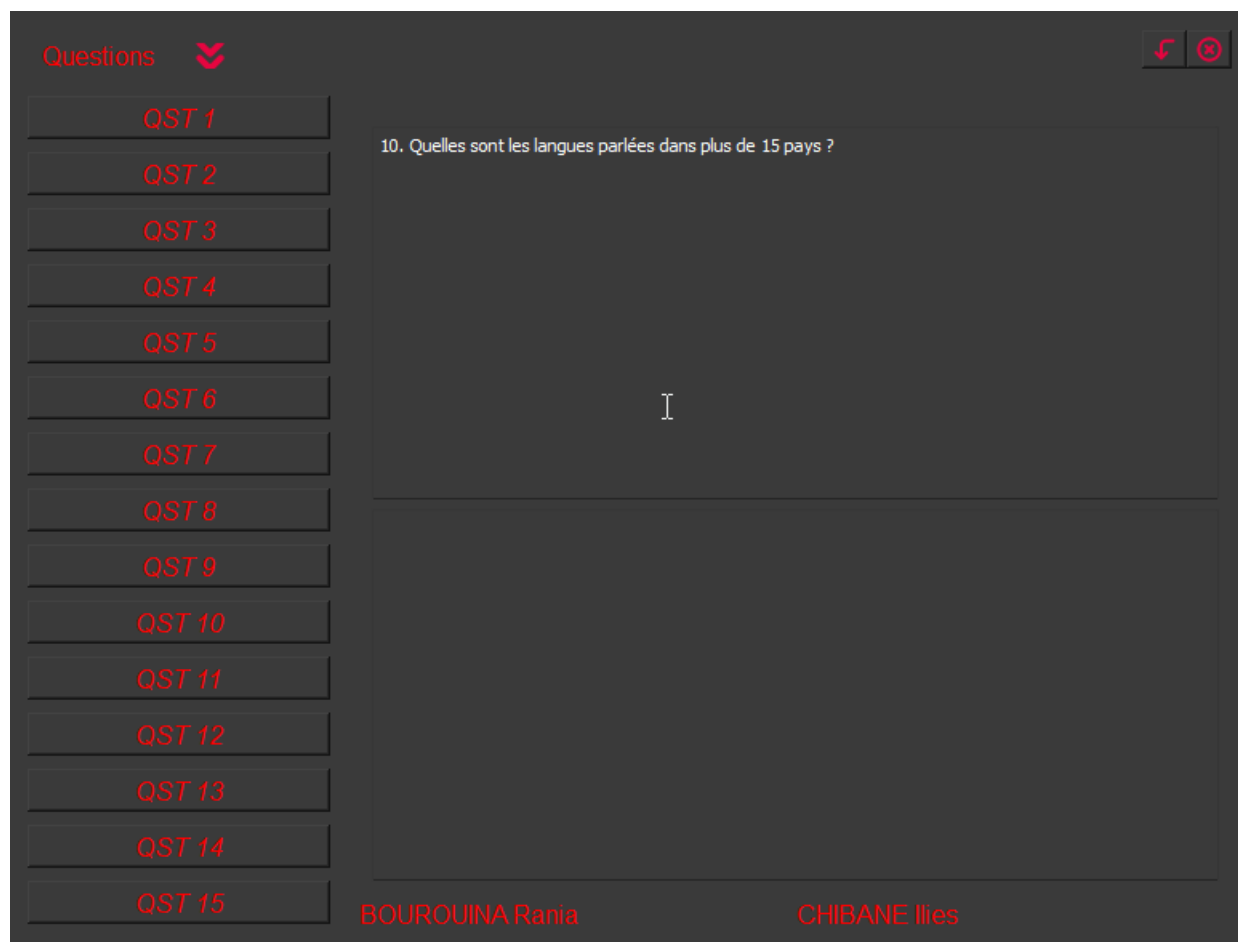


FIGURE 36 – Le résultat sur l'interface

- 2.11 11. Calculer pour chaque pays le nombre de villes (pour les pays ayant au moins 100 villes), en les triant par ordre décroissant du nombre de villes

```
>_MONGOSH

> db.world.aggregate([
  {
    $addFields: { villes : {$size: { "$ifNull": [ "$Cities", [] ] } } }
  },
  {
    $match: {
      villes: { $gt: 100 }
    }
  },
  {
    $sort: {"villes":-1}
  },
  {
    $project: {
      _id: "$Name",
      "Number of Cities" :"$villes"
    }
  }
])
< { _id: 'China', 'Number of Cities': 362 }
{ _id: 'India', 'Number of Cities': 340 }
{ _id: 'United States', 'Number of Cities': 273 }
{ _id: 'Brazil', 'Number of Cities': 249 }
{ _id: 'Japan', 'Number of Cities': 247 }
{ _id: 'Russian Federation', 'Number of Cities': 188 }
{ _id: 'Mexico', 'Number of Cities': 172 }
{ _id: 'Philippines', 'Number of Cities': 135 }
BDD>
```

FIGURE 37 – Execution de la commande avec Mongo Shell

```

1 def QST11(): # 11. Calculer pour chaque pays le nombre de villes (pour les pays ayant au moins 100 villes), en les triant par ordre décroissant du nombre de villes
2     ...
3     db.world.aggregate([
4         {
5             $addFields: { villes : { $size: { $ifNull: [ "$Cities", [] ] } } }
6         },
7         {
8             $match: {
9                 villes: { $gt: 100 }
10            }
11        },
12        {
13            $sort: { "villes": -1 } },
14        {
15            $project: {
16                _id: "$Name",
17                "Number of Cities": "$villes"
18            }
19        })
20    ...
21    pays = []
22    pipe = [{ '$addFields': { "villes" : { '$size': { '$ifNull': [ "$Cities", [] ] } } } }, { '$match': { "villes": { '$gt': 100 } } }, { '$sort': { "villes": -1 } }, { '$project': { "_id": "$Name", "Number of Cities": "$villes" } } ]
23    for post in collection.aggregate(pipeline=pipe):
24        pays.append(post)
25    return pays
26

```

FIGURE 38 – La fonction Utilisée

Questions

QST 1

QST 2

QST 3

QST 4

QST 5

QST 6

QST 7

QST 8

QST 9

QST 10

QST 11

QST 12

QST 13

QST 14

QST 15

11. Calculer pour chaque pays le nombre de villes (pour les pays ayant au moins 100 villes), en les triant par ordre décroissant du nombre de villes

_id	Number of Cities
China	362
India	340
United States	273
Brazil	249
Japan	247
Russian Federation	188
Mexico	172
Philippines	135

BOUROUINA Rania
CHIBANE Ilies

FIGURE 39 – Le résultat sur l'interface

2.12 12. Lister les 10 villes les plus habitées, ainsi que leur pays, dans l'ordre décroissant de la population

```
>_MONGOSH
> db.world.aggregate([
  { $unwind: "$Cities" },
  { $sort: {"Cities.Population":-1} },
  { $limit : 10 },
  { $sort: {"Cities.Population":1} },
  {
    $project: {
      _id:"$Cities.Name",
      "Country":"$Name",
      "City Population":"$Cities.Population"
    }
  }
])
< { _id: 'Tianjin', Country: 'China', 'City Population': 5286800 }
{ _id: 'Rio de Janeiro',
  Country: 'Brazil',
  'City Population': 5598953 }
{ _id: 'Chongqing',
  Country: 'China',
  'City Population': 6351600 }
{ _id: 'Delhi', Country: 'India', 'City Population': 7206704 }
{ _id: 'New York',
  Country: 'United States',
  'City Population': 8008278 }
{ _id: 'Istanbul',
```

FIGURE 40 – Execution de la commande avec Mongo Shell

```

>_MONGOSH

    })
< { _id: 'Tianjin', Country: 'China', 'City Population': 5286800 }
  { _id: 'Rio de Janeiro',
    Country: 'Brazil',
    'City Population': 5598953 }
  { _id: 'Chongqing',
    Country: 'China',
    'City Population': 6351600 }
  { _id: 'Delhi', Country: 'India', 'City Population': 7206704 }
  { _id: 'New York',
    Country: 'United States',
    'City Population': 8008278 }
  { _id: 'Istanbul',
    Country: 'Turkey',
    'City Population': 8787958 }
  { _id: 'Karachi',
    Country: 'Pakistan',
    'City Population': 9269265 }
  { _id: 'Shanghai', Country: 'China', 'City Population': 9696300 }
  { _id: 'São Paulo',
    Country: 'Brazil',
    'City Population': 9968485 }
  { _id: 'Mumbai (Bombay)',
    Country: 'India',
    'City Population': 10500000 }
BDD>

```

FIGURE 41 – Execution de la commande avec Mongo Shell

```

1  def Q5T12(): # 12. Lister les 10 villes les plus habitées, ainsi que leur pays, dans l'ordre décroissant de la population
2  ...
3      db.world.aggregate([
4          {'$unwind': '$Cities' },
5          {'$sort': {'Cities.Population': -1 } },
6          {'$limit': 10 },
7          {'$sort': {'Cities.Population': 1 } },
8          {'$project': {
9              '_id': '$Cities.Name',
10             'Country': '$Name',
11             'City Population': '$Cities.Population'
12         } }
13     ])
14     ...
15     villes = []
16     pipe = [{'$unwind': '$Cities'}, {'$sort': {'Cities.Population': -1 } }, {'$limit': 10 }, {'$sort': {'Cities.Population': 1 } }, {'$project': { '_id': '$Cities.Name', 'Country': '$Name', 'City Population': '$Cities.Population' }}]
17     for post in collection.aggregate(pipeline=pipe):
18         villes.append(post)
19     return villes
20
21
22
23

```

FIGURE 42 – La fonction Utilisée

2.13 13. Lister les pays pour lesquels l'Arabe est une langue officielle

```
> _MONGOSH

> db.world.find({"OffLang.Language":{"$eq: "Arabic"}},{"Name":1})
< { _id: ObjectId("626916c7e77e9c4a05cdf653"),
  Name: 'United Arab Emirates' }
{ _id: ObjectId("626916c7e77e9c4a05cdf65c"), Name: 'Bahrain' }
{ _id: ObjectId("626916c7e77e9c4a05cdf67e"), Name: 'Djibouti' }
{ _id: ObjectId("626916c7e77e9c4a05cdf684"),
  Name: 'Western Sahara' }
{ _id: ObjectId("626916c7e77e9c4a05cdf687"), Name: 'Algeria' }
{ _id: ObjectId("626916c7e77e9c4a05cdf694"), Name: 'Egypt' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6b1"), Name: 'Iraq' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6b2"), Name: 'Israel' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6b3"), Name: 'Jordan' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6bc"), Name: 'Kuwait' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6bd"), Name: 'Lebanon' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6c0"),
  Name: 'Libyan Arab Jamahiriya' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6d3"), Name: 'Morocco' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6ea"), Name: 'Oman' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6f7"), Name: 'Qatar' }
{ _id: ObjectId("626916c7e77e9c4a05cdf700"),
  Name: 'Saudi Arabia' }
{ _id: ObjectId("626916c7e77e9c4a05cdf704"), Name: 'Sudan' }
{ _id: ObjectId("626916c7e77e9c4a05cdf70d"), Name: 'Somalia' }
{ _id: ObjectId("626916c7e77e9c4a05cdf712"), Name: 'Syria' }
{ _id: ObjectId("626916c7e77e9c4a05cdf714"), Name: 'Chad' }
```

FIGURE 44 – Execution de la commande avec Mongo Shell

```

> _MONGOSH

> db.world.find({"OffLang.Language":{"$in: ["Arabic"]}}, {"Name":1})
< { _id: ObjectId("626916c7e77e9c4a05cdf653"),
  Name: 'United Arab Emirates' }
{ _id: ObjectId("626916c7e77e9c4a05cdf65c"), Name: 'Bahrain' }
{ _id: ObjectId("626916c7e77e9c4a05cdf67e"), Name: 'Djibouti' }
{ _id: ObjectId("626916c7e77e9c4a05cdf684"),
  Name: 'Western Sahara' }
{ _id: ObjectId("626916c7e77e9c4a05cdf687"), Name: 'Algeria' }
{ _id: ObjectId("626916c7e77e9c4a05cdf694"), Name: 'Egypt' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6b1"), Name: 'Iraq' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6b2"), Name: 'Israel' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6b3"), Name: 'Jordan' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6bc"), Name: 'Kuwait' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6bd"), Name: 'Lebanon' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6c0"),
  Name: 'Libyan Arab Jamahiriya' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6d3"), Name: 'Morocco' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6ea"), Name: 'Oman' }
{ _id: ObjectId("626916c7e77e9c4a05cdf6f7"), Name: 'Qatar' }
{ _id: ObjectId("626916c7e77e9c4a05cdf700"),
  Name: 'Saudi Arabia' }
{ _id: ObjectId("626916c7e77e9c4a05cdf704"), Name: 'Sudan' }
{ _id: ObjectId("626916c7e77e9c4a05cdf70d"), Name: 'Somalia' }
{ _id: ObjectId("626916c7e77e9c4a05cdf712"), Name: 'Syria' }
{ _id: ObjectId("626916c7e77e9c4a05cdf714"), Name: 'Chad' }

```

FIGURE 45 – Execution de la commande avec Mongo Shell

```

1  def QST13(): # 13. Lister les pays pour lesquels L'Arabe est une langue officielle
2      '''
3      db.world.find({"OffLang.Language":{"$eq:"Arabic"}}, {"Name":1})
4      '''
5      countries = []
6      for post in (collection.find({"OffLang.Language":{"$eq":"Arabic"}}, {"Name":1})):
7          countries.append(post["Name"])
8      return countries
9

```

FIGURE 46 – La fonction Utilisée

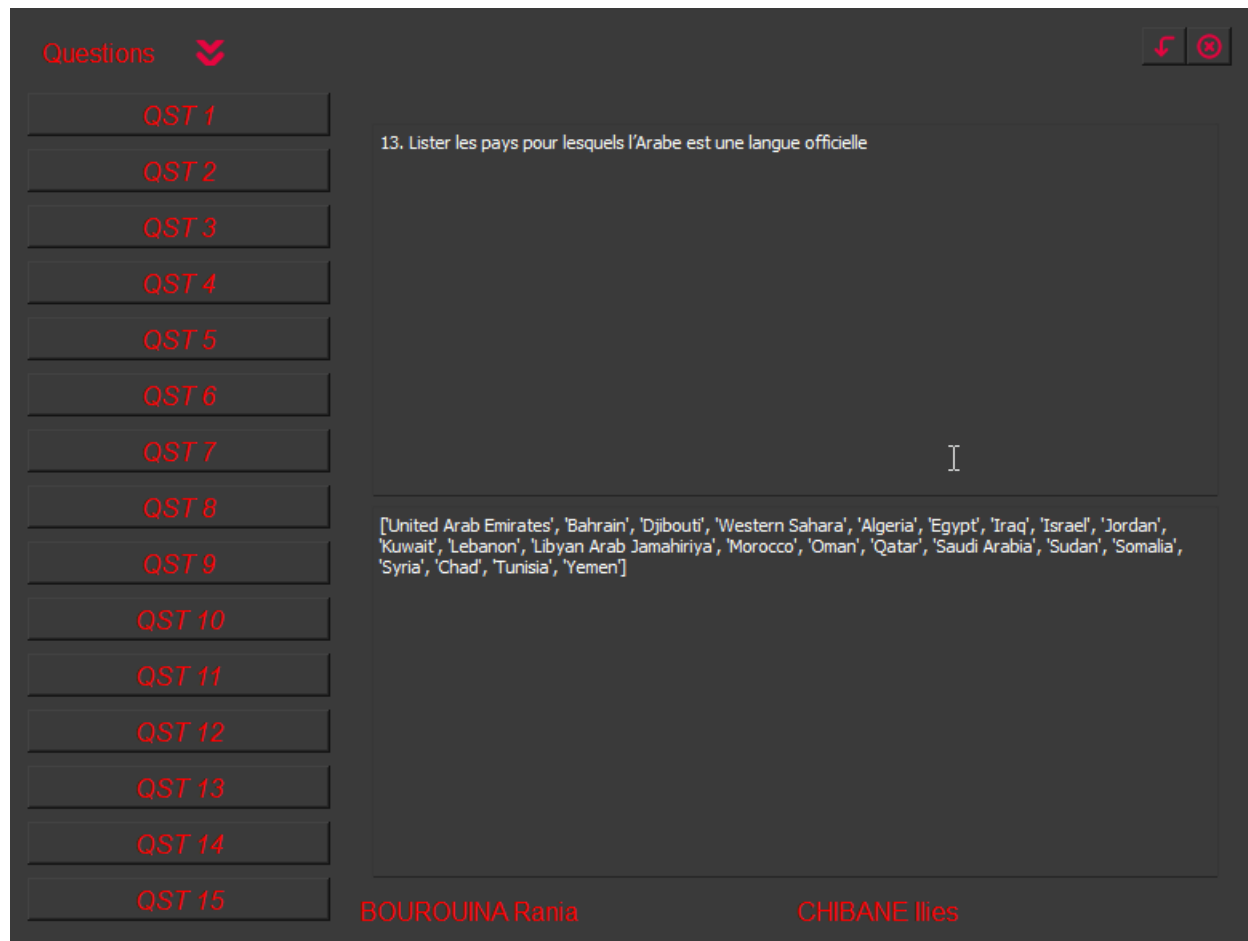


FIGURE 47 – Le résultat sur l'interface

2.14 14. Lister les 5 pays avec le plus de langues parlées

```
> _MONGOSH

> db.world.aggregate([
  {
    $addFields: { c : { $concatArrays: [ "$OffLang", "$NotOffLang" ] } }
  },
  {
    $addFields: { langs : { $size: { "$ifNull": [ "$c", [] ] } } }
  },
  {
    $sort: {"langs":-1}
  },
  { $limit : 5 },
  {
    $group: {
      _id: "$Name"
    }
  }
])
< { _id: 'China' }
{ _id: 'Russian Federation' }
{ _id: 'India' }
{ _id: 'United States' }
{ _id: 'Canada' }

BDD>
```

FIGURE 48 – Execution de la commande avec Mongo Shell

```
1  def Q514(): # 14. Lister les 5 pays avec le plus de langues parlées
2  """
3  """
4  db.world.aggregate([
5    {
6      $addFields: { c : { $concatArrays: [ "$OffLang", "$NotOffLang" ] } }
7    },
8    {
9      $addFields: { langs : { $size: { "$ifNull": [ "$c", [] ] } } }
10   },
11   {
12     $sort: {"langs":-1}
13   },
14   { $limit : 5 },
15   {
16     $group: {
17       _id: "$Name"
18     }
19   }
20 ])
21
22 countries = []
23 pipe = [ { $addFields: { "c" : { $concatArrays: [ "$OffLang", "$NotOffLang" ] } } }, { $addFields: { "langs" : { $size: { "$ifNull": [ "$c", [] ] } } } }, { $sort: {"langs":-1} }, { $limit : 5 }, { $group: { "_id": "$Name" } } ]
24
25 for post in collection.aggregate(pipeline=pipe):
26     countries.append(post["_id"])
27 return countries
```

FIGURE 49 – La fonction Utilisée

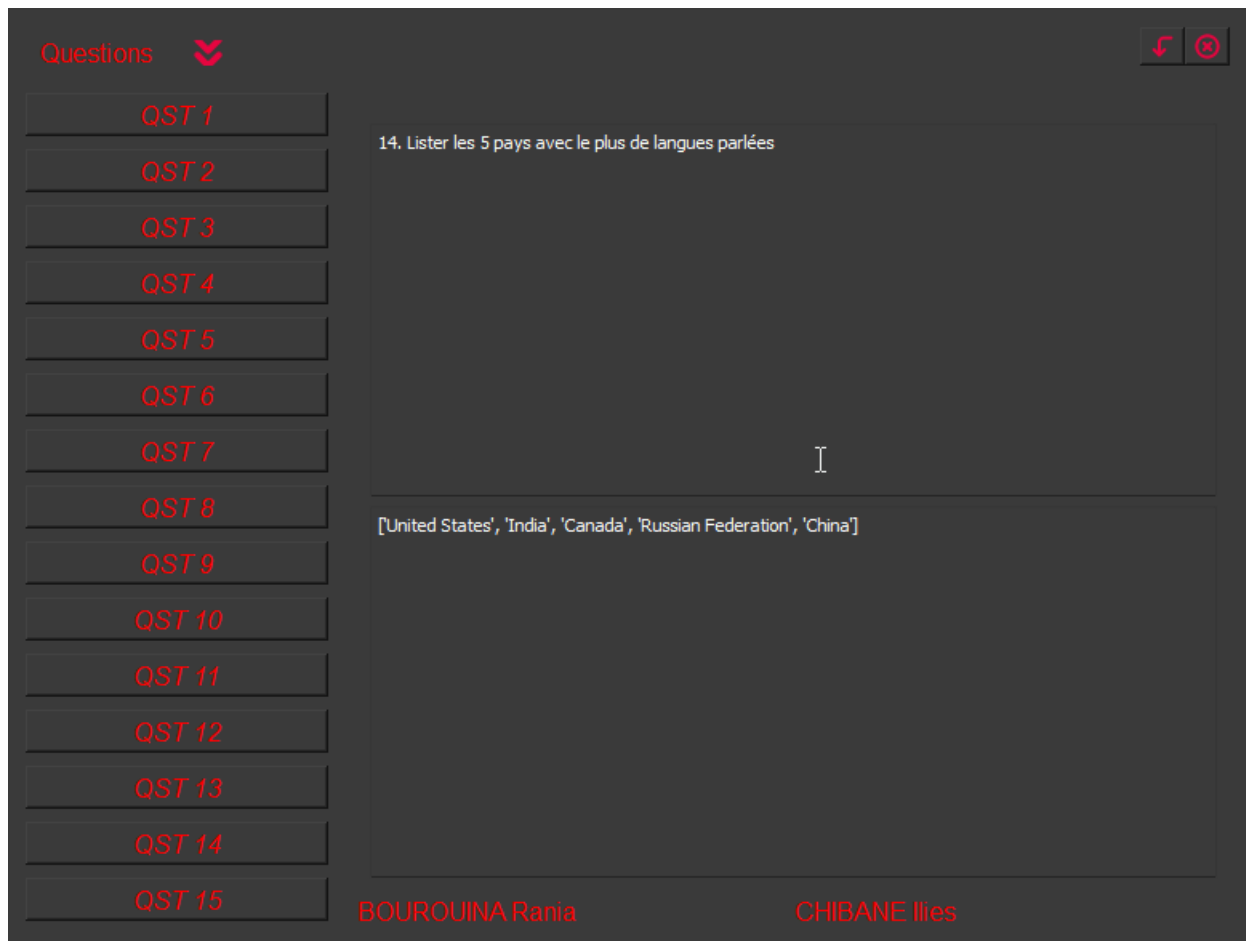


FIGURE 50 – Le résultat sur l'interface

- 2.15 15. Lister les pays pour lesquels la somme des populations des villes est supérieure à la population du pays.

```
>_MONGOSH
>
  db.world.aggregate([
    {
      $addFields: { somme : {$sum: "$Cities.Population"} }
    },
    {
      $addFields: { comp : {$cmp: ['$somme', '$Population']} }
    },
    {$match: {comp: {$gt: 1}}},
    {
      $project: {
        _id: "$Name",
        "Total Cities Population": "$somme",
        "Country Population": "$Population"
      }
    }
  ])
<
BDD>
```

FIGURE 51 – Execution de la commande avec Mongo Shell

```

1  def QST15(): # 15. Lister les pays pour lesquels la somme des populations des villes est supérieure à la population du pays.
2      '''
3          db.world.aggregate([
4              {
5                  $addFields: { somme : { $sum: "$Cities.Population" } }
6              },
7              {
8                  $addFields: { comp : { $cmp: ['$somme', '$Population'] } }
9              },
10             { $match: { comp: { $gt: 1 } } },
11             {
12                 $project: {
13                     _id: "$Name",
14                     "Total Cities Population": "$somme",
15                     "Country Population": "$Population"
16                 }
17             }
18         ])
19         '''
20         countries = []
21         pipe = [
22             {
23                 $addFields: { "somme" : { $sum: "$Cities.Population" } }
24             },
25             {
26                 $addFields: { "comp" : { $cmp: ['$somme', '$Population'] } }
27             },
28             { $match: { "comp": { $gt: 1 } } },
29             {
30                 $project: {
31                     _id: "$Name",
32                     "Total Cities Population": "$somme",
33                     "Country Population": "$Population"
34                 }
35             }
36         ]
37         for post in collection.aggregate(pipeline=pipe):
38             countries.append(post["_id"])
39         return countries
40     
```

FIGURE 52 – La fonction Utilisée

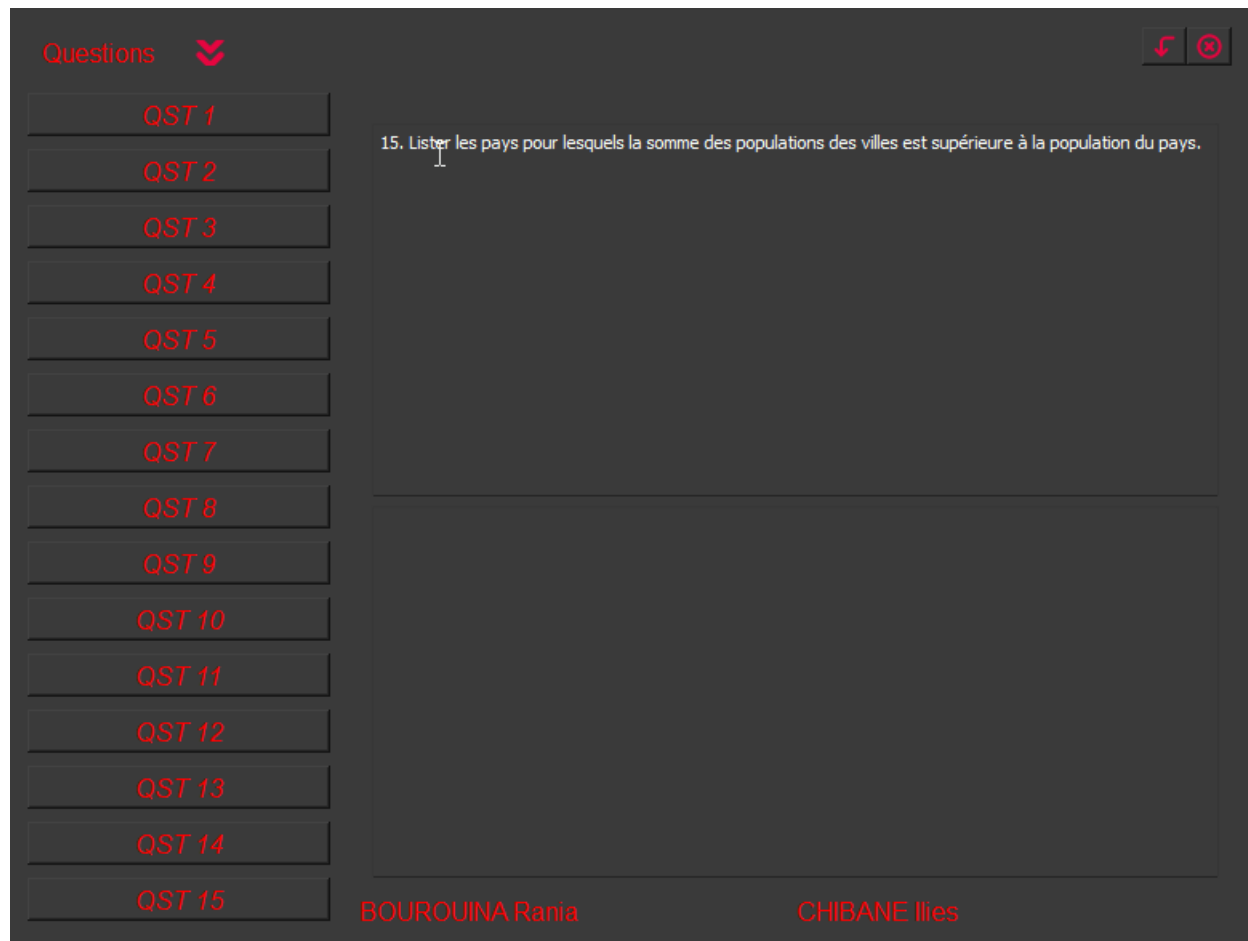


FIGURE 53 – Le résultat sur l'interface

Il n'y a aucun pays dont la somme des populations de ses villes est supérieure à sa population. Pour confirmer que ceci est vrai on exécute la commande suivante pour avoir les noms des pays qui ont une population supérieure à la somme des populations de ses villes.


```

> _MONGOSH
> db.world.aggregate([
  {
    $addFields: { somme : {$sum: "$Cities.Population"} }
  },
  {
    $addFields:{comp : {$cmp: ['$somme','$Population']}}
  },
  {$match: {comp:{$eq:-1}}},
  {
    $project: {
      _id:"$Name",
      "Total Cities Population": "$somme",
      "Country Population": "$Population"
    }
  }
])
< { _id: 'Armenia',
  'Total Cities Population': 384400,
  'Country Population': 3520000 }
{ _id: 'American Samoa',
  'Total Cities Population': 5200,
  'Country Population': 68000 }
{ _id: 'Afghanistan',
  'Total Cities Population': 552100,
  'Country Population': 22720000 }

```

FIGURE 54 – Execution de la commande avec Mongo Shell