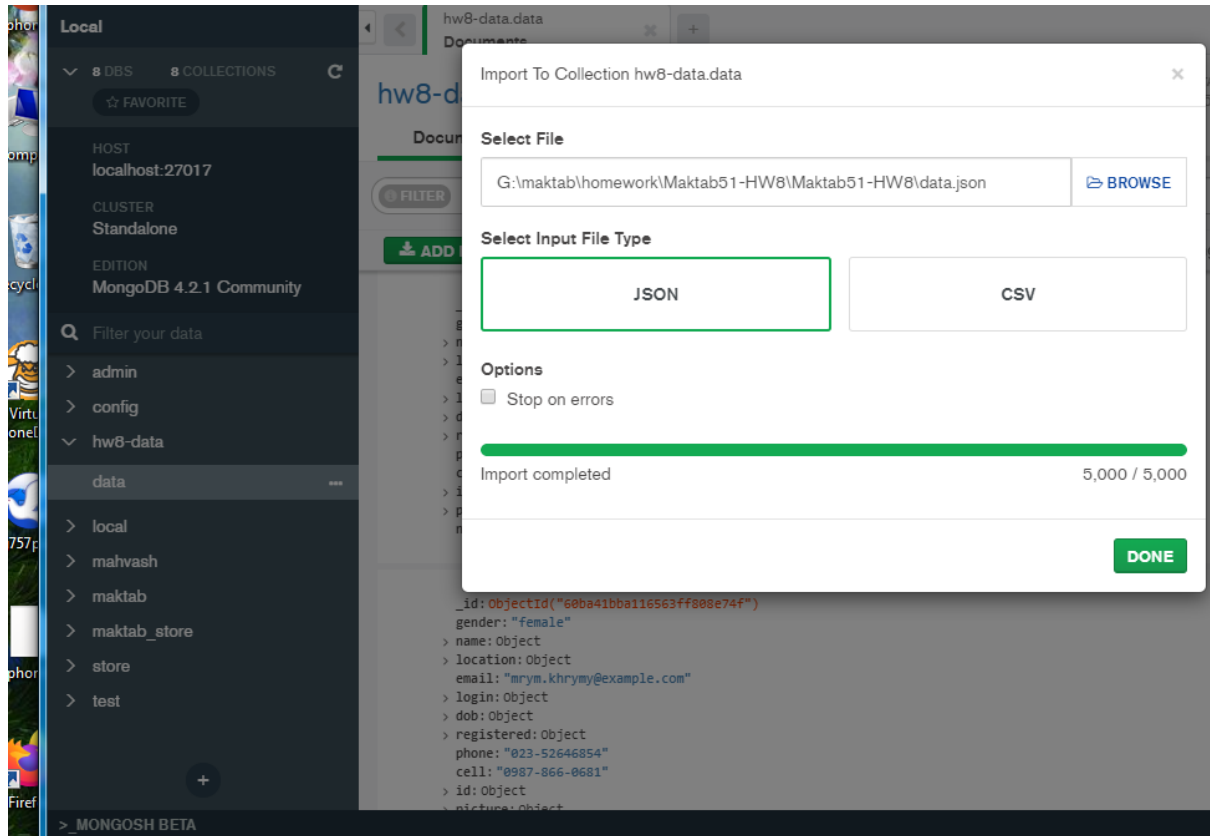


A. After downloading data, save them to the mongodb manually.

(capture one sample of adding your data to your answer(PDF))

Note: Capture your insert code and it's result. If you have many things, capture just one sample of insertion.



B. Make the following queries. Capture the results and paste them to your answer(PDF).

Note: Capture the queries and its results.

1-Find the firstname and lastname of all users that have more than 50 years and reside on (their city = 'گلستان')

```
Administrator: Command Prompt - mongo
> use hw8-data
switched to db hw8-data
> db.data.find(<<"dob.age" :<"$gt":50>,"location.city":"?0%ófδ">,<"name.first":1,"name.last":1>>).pretty()
{
  "_id" : ObjectId<"60ba41bba116563ff808e79c">,
  "name" : {
    "first" : "?ø?δf",
    "last" : "?r?Ω?"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e7b6">,
  "name" : {
    "first" : "Πr-¼nf",
    "last" : "øfΩnf-r-fδ"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e82a">,
  "name" : {
    "first" : "?ø-r?f",
    "last" : "½fδfr?"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e84d">,
  "name" : {
    "first" : "ωø?f",
    "last" : "r-αf??"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e8b3">,
  "name" : {
    "first" : "δnf",
    "last" : "½øβfδ? δ?fø"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e8b6">,
  "name" : {
    "first" : "r-fø?δ",
    "last" : "½fδfr?"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e8f1">,
  "name" : {
    "first" : "σfβΩω",
    "last" : "Ωø½ø?"
  }
}
{
  "_id" : ObjectId<"60ba41bba116563ff808e950">,
  "name" : {
    "first" : "Πø?"
  }
}
```

hw8-data.data

Documents

Aggregations

Schema

Explain Plan

FILTER `{ $and: [{ "dob.age": { $gt: 50 } }, { "location.city": "گلستان" }] }`

PROJECT `{ "name.first": 1, "name.last": 1 }`

SORT `{ field: -1 }`

COLLATION `{ locale: 'simple' }`



VIEW



`_id: ObjectId("60ba41bba116563ff808e79c")`
▼ name: Object
first: "دینا"
last: "کریمی"

`_id: ObjectId("60ba41bba116563ff808e7b6")`
▼ name: Object
first: "عرفیا"
last: "کامیاران"

`_id: ObjectId("60ba41bba116563ff808e82a")`
▼ name: Object
first: "پوریا"
last: "سالاری"

2--We want to reward those users who have registered on our site for more than 20 years. Find their last name, phone and address.

```
Administrator: Command Prompt - mongo
Type "it" for more
> db.data.find(<{"registered.age" : <{"$gt":10}>}, {"name.last":1, "phone":1, "location.city":1, "location.street":1}>).pretty()
{
  "_id" : ObjectId("60ba41bba116563ff808e74f"),
  "name" : {
    "last" : "?r?Ω?"
  },
  "location" : {
    "street" : {
      "number" : 1548,
      "name" : "%ω?ε ?ά?r? θfΩω"
    },
    "city" : "ιfωεfδ"
  },
  "phone" : "023-52646854"
}
{
  "_id" : ObjectId("60ba41bba116563ff808e750"),
  "name" : {
    "last" : "?fΩr-øf"
  },
  "location" : {
    "street" : {
      "number" : 4602,
      "name" : "ιfΩΩ ΩιfΩ σr-fωfδ?"
    },
    "city" : "fiσωfδ"
  },
  "phone" : "071-56056918"
}
{
  "_id" : ObjectId("60ba41bba116563ff808e751"),
  "name" : {
    "last" : "δξø δΓr"
  },
  "location" : {
    "street" : {
      "number" : 1174,
      "name" : "ε?όr- θøf½fδ?"
    },
    "city" : "fωøfι"
  },
  "phone" : "046-08764369"
}
{
  "_id" : ObjectId("60ba41bba116563ff808e752"),
  "name" : {
    "last" : "r-αf??"
  },
  "location" : {
    "street" : {
      "number" : 7273,
      "name" : "%ør-f"
    },
    "city" : "f?θfΩ"
  },
}
```



```

    { "$eq": [ { "$dayOfMonth": "$dob.date" }, { "$dayOfMonth": new Date() } ] },
    { "$eq": [ { "$month"      : "$dob.date" }, { "$month"      : new Date() } ] }
  ]
}

```

4-Find the number of users based on their province(=state). For example Isfahan, 30
Tehran, 50

```

> db.data.aggregate([{$group:{$_id:"$location.state",count: < $sum: 1 >}}]
...
> db.data.aggregate([{$group:{$_id:"$location.state",count: < $sum: 1 >}}])
{ "_id" : "تهران", "count" : 179 }
{ "_id" : "اصفهان", "count" : 163 }
{ "_id" : "چابهار", "count" : 175 }
{ "_id" : "گلستان", "count" : 172 }
{ "_id" : "کرمان", "count" : 173 }
{ "_id" : "آذربایجان", "count" : 164 }
{ "_id" : "کرمانشاه", "count" : 173 }
{ "_id" : "آذربایجان", "count" : 130 }
{ "_id" : "فارس", "count" : 172 }
{ "_id" : "اصفهان", "count" : 164 }
{ "_id" : "کرمان", "count" : 161 }
{ "_id" : "اصفهان", "count" : 160 }
{ "_id" : "کرمان", "count" : 171 }
{ "_id" : "کرمان", "count" : 160 }
{ "_id" : "اصفهان", "count" : 156 }
{ "_id" : "اصفهان", "count" : 151 }
{ "_id" : "اصفهان", "count" : 157 }
{ "_id" : "اصفهان", "count" : 164 }
{ "_id" : "اصفهان", "count" : 154 }
{ "_id" : "اصفهان", "count" : 174 }
Type "it" for more
>

```

hw8-data.data Aggregations

DOCUMENTS 5.0k TOTAL SIZE 5.5MB AVG. SIZE 1.1KB INDEXES 1 TOTAL SIZE 60.0KB AVG. SIZE 60.0K

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION Untitled- Modified SAVE SAMPLE MODE AUTO PREVIEW

5000 Documents in the Collection Preview of Documents in the Collection

\$group Output after \$group stage (Sample of 20 documents)

```

1 /**
2  * _id: The id of the group.
3  * fieldN: The first field name.
4  */
5 {
6   _id: "$location.state",
7   count: {$sum:1}
8 }
9

```

id: "تم" count: 173

_id: "کردستان" count: 175

ADD STAGE

5-Which city has the most number of users and which one has the least number of users?

```

Type "it" for more
> db.data.aggregate([{$group:{$_id:"$location.state",count: { $sum: 1 } }},{$sort:
{"count":-1}},{$limit:1}]
{ "_id" : "کردستان", "count" : 183 }
> db.data.aggregate([{$group:{$_id:"$location.state",count: { $sum: 1 } }},{$sort:
{"count":1}},{$limit:1}]
{ "_id" : "آمل", "count" : 130 }
>

```

Documents Aggregations Schema Explain Plan Indexes Validation

COLLATION { locale: 'simple' }

Sort

Output after `$sort` stage (Sample of 20 documents)

```
1 /**
2  * Provide any number of field/order pairs.
3  */
4  {"count": -1}
```

Output after `$limit` stage (Sample of 1 document)

```
1 /**
2  * Provide the number of documents to limit.
3  */
4  1
```

Output after `$sort` stage (Sample of 20 documents)

```
{ "_id": "تهران", "count": 183 }
```

Output after `$limit` stage (Sample of 1 document)

```
{ "_id": "تهران", "count": 183 }
```

6-Compare the average of users' age in Tehran with the average of users' age in other cities.

```
> use hw8-data
switched to db hw8-data
> db.data.aggregate([{$facet:<"a":[{$match:<"location.city":{$in:[<"تهران">]}>]}, <
  $group:<_id:null,ave: < $avg: "$dob.age" >>>]}, "b":[{$match:<"location.city":{$n
  in:[<"تهران">]}>]}, <$group:<_id:null,ave: < $avg: "$dob.age" >>>]}],<$project:<res
  ult:<"$cmp":[<"$a.ave", "$b.ave">]>>>]}]
< "result" : -1 >
```

Facet

Output after `$facet` stage (Sample of 1 document)

```
1 /**
2  * outputFieldN: The first output field.
3  * stageN: The first aggregation stage.
4  */
5  {
6  "a": [ { $match: { "location.city": { $in: [ "تهران" ] } } }, { $group: { _id: null,
7  "b": [ { $match: { "location.city": { $in: [ "تهران" ] } } }, { $group: { _id: null,
```

Project

Output after `$project` stage (Sample of 1 document)

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5  {
6  {
7  result: { "$cmp": [ "$a.ave", "$b.ave" ] }
8  }
9  }
```


7-For an advertisement, we want to categorize users based on their age. Write a query to categorize them into three groups 1)youth 2)middle-aged 3)old:

Youth for ages < 16

Middle-aged for 16 < ages <40

Old for ages >40

The screenshot shows the MongoDB Compass interface for a collection named 'hw8-data.data'. The 'Aggregations' tab is active, displaying a pipeline with a single stage named '\$bucket'. The stage configuration is as follows:

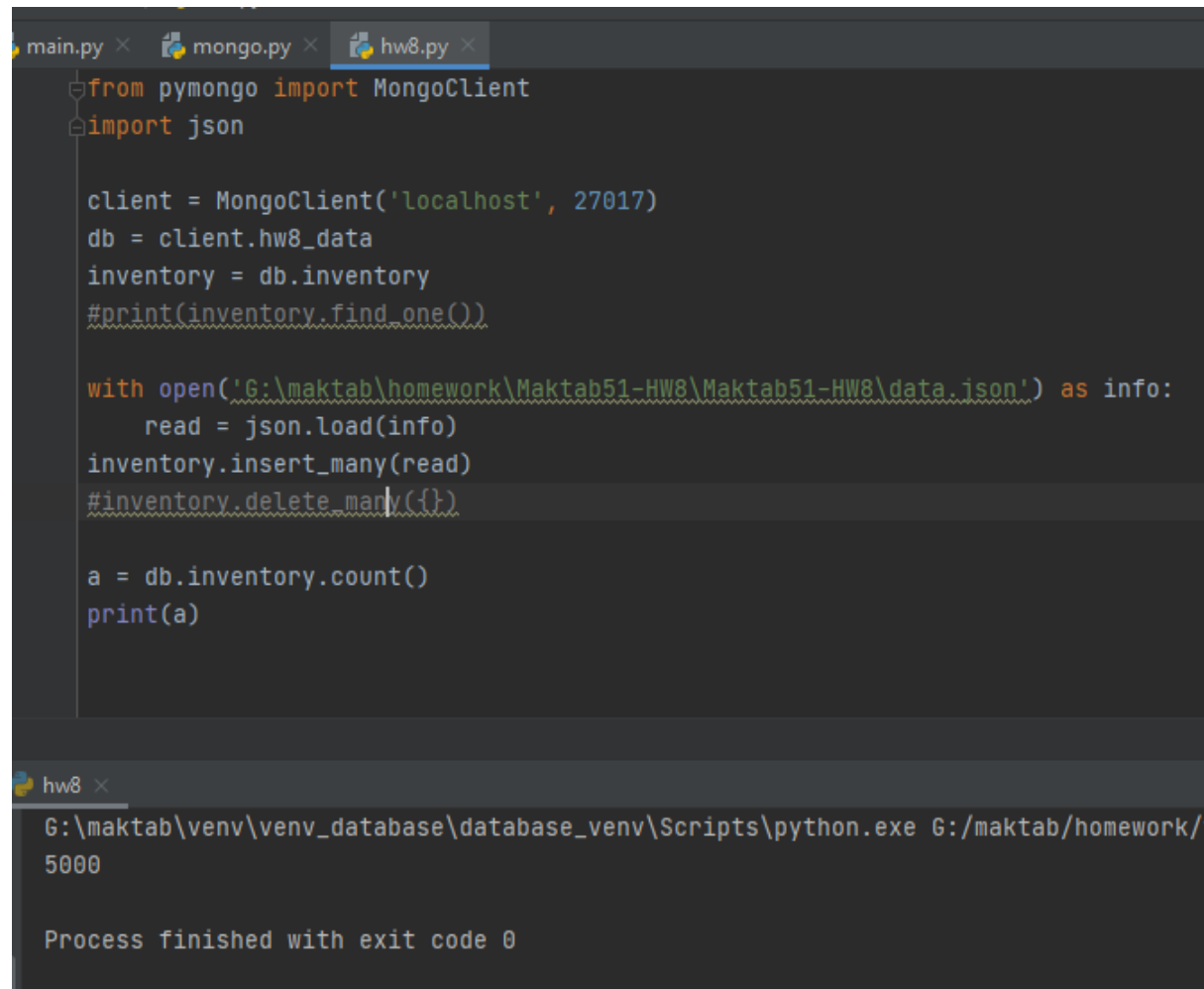
- groupBy: "\$dob.age"
- boundaries: [0, 25, 40]
- default: "old"
- output: {
 outputN: { \$push: { "name": "name.first", "age": "\$dob.age" } }

The 'Output after \$bucket stage' panel shows a sample of 3 documents. The first document is categorized as 'old' and contains an array of 5 objects, each with 'name' and 'age' fields. The second and third documents are partially visible, also showing the 'old' category and an array of objects.

```
{ "name.first": "name.first", "age": 69 }, { "name": "name.first", "age": 45 }, { "name": "name.first", "age": 72 }, { "name": "name.first", "age": 46 }, { "name": "name.first", "age": 59 }, { "name": "name.first", "age": 60 }, { "name": "name.first", "age": 56 }, { "name": "name.first", "age": 61 }, { "name": "name.first", "age": 63 }, { "name": "name.first", "age": 52 }, { "name": "name.first", "age": 76 }, { "name": "name.first", "age": 63 }, { "name": "name.first", "age": 49 }, { "name": "name.first", "age": 43 }, { "name": "name.first", "age": 50 }, { "name": "name.first", "age": 53 }, { "name": "name.first", "age": 76 }, { "name": "name.first", "age": 60 }, { "name": "name.first", "age": 76 }, { "name": "name.first", "age": 71 }, { "name": "name.first", "age": 51 }, { "name": "name.first", "age": 53 }, { "name": "name.first", "age": 70 }, { "name": "name.first", "age": 49 }, { "name": "name.first", "age": 52 }, { "name": "name.first", "age": 42 }, { "name": "name.first", "age": 76 }, { "name": "name.first", "age": 43 }, { "name": "name.first", "age": 74 }, { "name": "name.first", "age": 46 }, { "name": "name.first", "age": 77 }, { "name": "name.first", "age": 76 }, { "name": "name.first", "age": 60 }, { "name": "name.first", "age": 46 }, { "name": "name.first", "age": 53 }, { "name": "name.first", "age": 75 }, { "name": "name.first", "age": 66 }, { "name": "name.first", "age": 48 }, { "name": "name.first", "age": 59 }, { "name": "name.first", "age": 50 }, { "name": "name.first", "age": 69 }, { "name": "name.first", "age": 66 }, { "name": "name.first", "age": 41 }, { "name": "name.first", "age": 53 }, { "name": "name.first", "age": 57 }, { "name": "name.first", "age": 72 }, { "name": "name.first", "age": 53 }, { "name": "name.first", "age": 46 } ] }
> db.data.aggregate([{$bucket: {groupBy: "$dob.age", boundaries: [0, 25, 40], default: "old", output: {outputN: { $push: { "name": "name.first", "age": "$dob.age" } } } } ]})
```

By using pymongo, save the information of these users to a new collection and make sure that the info of all users is saved to your collection using “count”.

Note: Capture your code and it's result. Do not forget to append your python file to your document.



```
main.py × mongo.py × hw8.py ×
from pymongo import MongoClient
import json

client = MongoClient('localhost', 27017)
db = client.hw8_data
inventory = db.inventory
#print(inventory.find_one())

with open('G:\maktab\homework\Maktab51-HW8\Maktab51-HW8\data.json') as info:
    read = json.load(info)
    inventory.insert_many(read)
    #inventory.delete_many({})

a = db.inventory.count()
print(a)
```

```
hw8 ×
G:\maktab\venv\venv_database\database_venv\Scripts\python.exe G:/maktab/homework/
5000

Process finished with exit code 0
```