TASK 2

Sophie Fidan
21068639

UFCFU3-15-3
Advanced

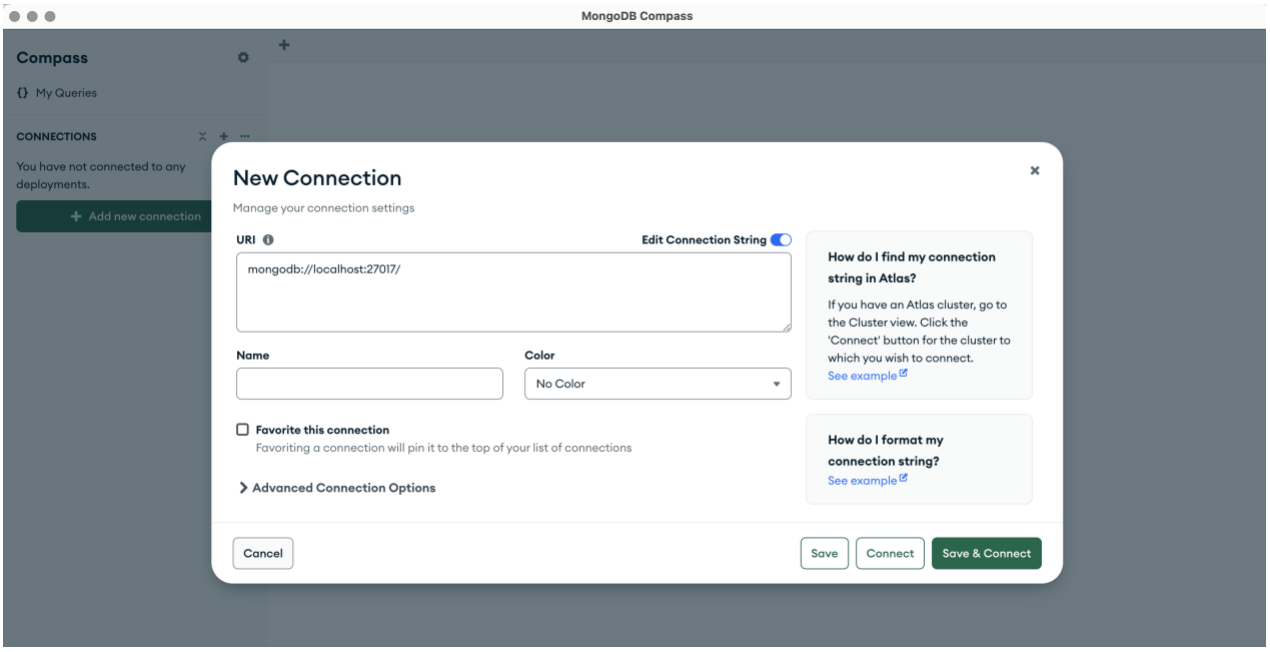
Databases

TABLE OF CONTENTS

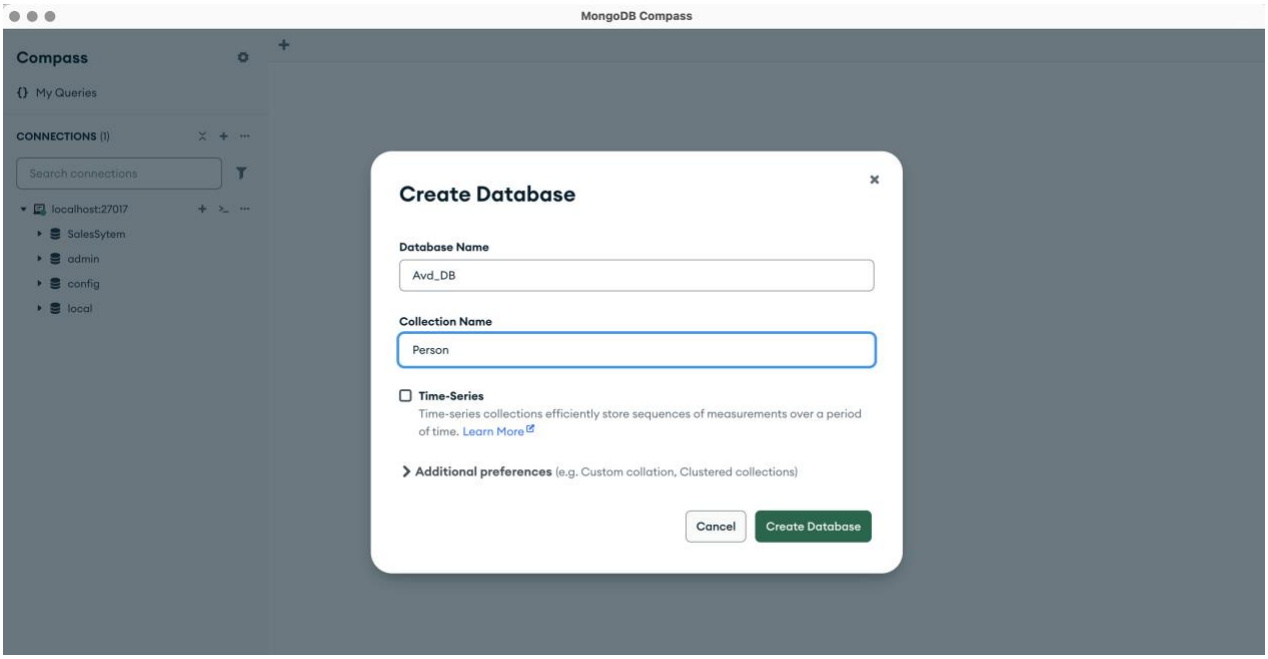
MONGODB SETUP

MongoDB setup using MongoDB Compass. In a NoSQL model, a single collection can be used to store all data to simplify data retrieval and reduce complexity. Therefore, all the data is imported straight from CSV file.

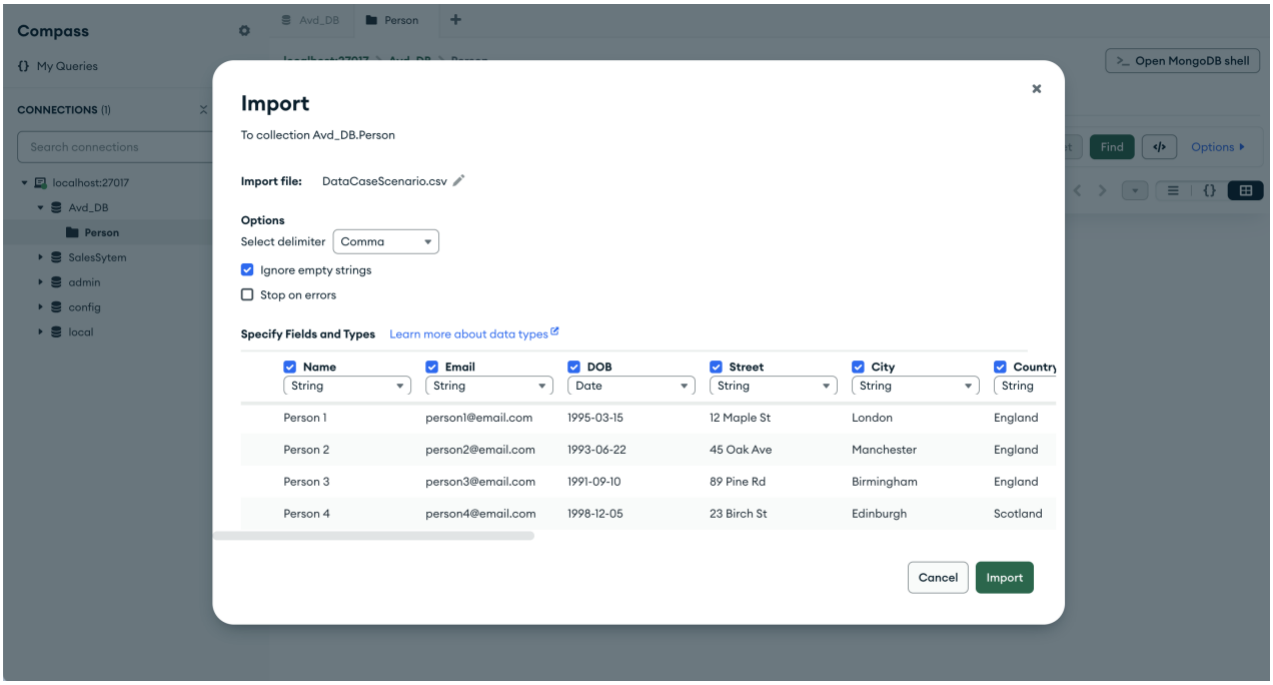
CONNECTING LOCALHOST



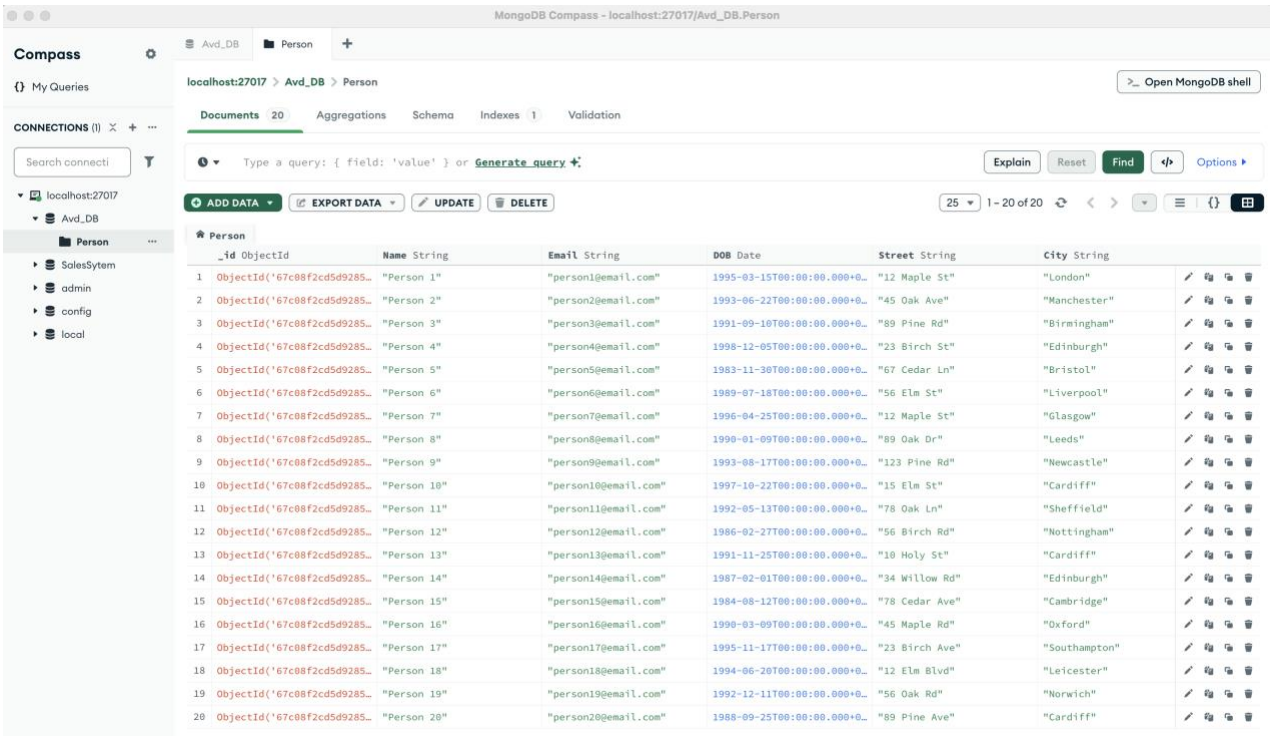
CREATING A NEW DATABASE AND COLLECTION



IMPORTING DATA FROM CSV FILE



THE PERSON COLLECTION



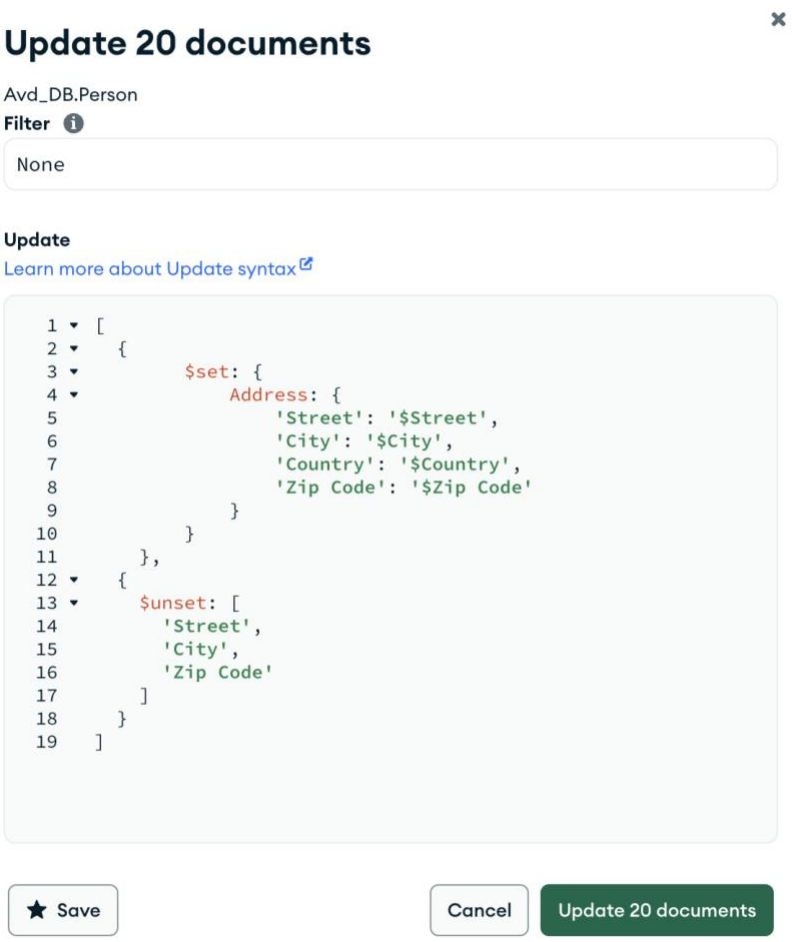
UPDATING DOCUMENTS

The data is transformed into a more structured design to improve efficiency:

- The imported CSV file had a flat structure, which made address details, favourite items and neighbours' information scattered as separate fields. By grouping related data, the database is more readable and organised, reducing complexity.
- With this design, queries become more efficient, as MongoDB can fetch entire objects instead of selecting multiple fields when retrieving data. This also improves the indexing performance of traversing the nested documents rather than scanning unrelated fields.

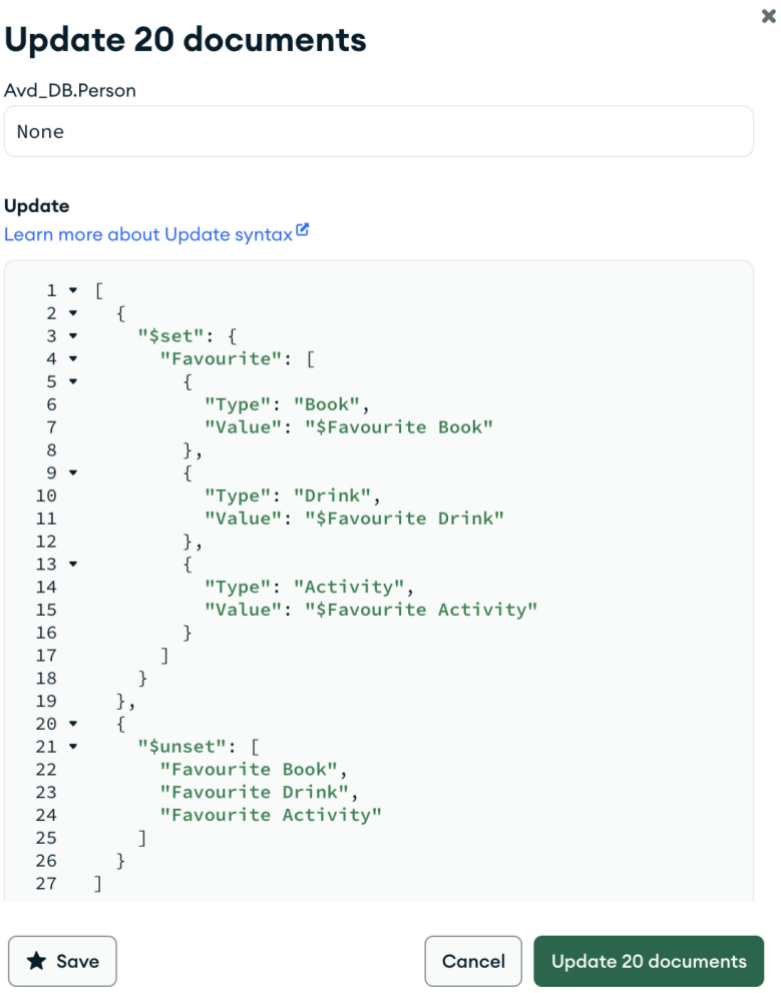
UPDATING ADDRESS FIELD

Instead of having different fields for *Street*, *City*, *Country* and *Zip Code*, the address details are reconstructed into a nested object called *Address*.



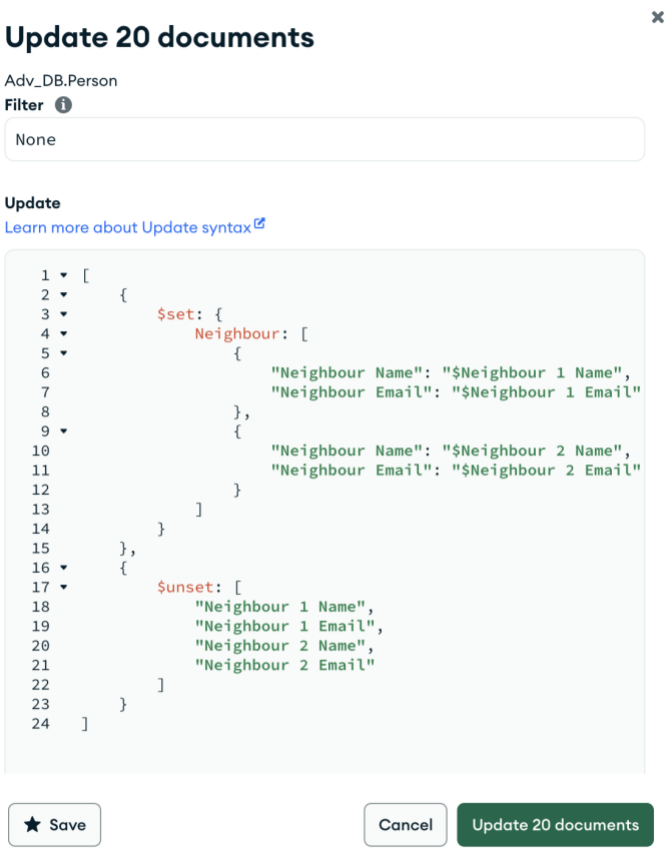
UPDATE FAVOURITE FIELD

Instead of having different fields for *Favourite Book*, *Favourite Drink*, and *Favourite Activity*, all favourite types are reconstructed into a nested object called *Favourite*.



UPDATE NEIGHBOUR FIELD

The neighbour information of name and email is stored into an array of objects called *Neighbour* to work efficiently with two neighbour objects.

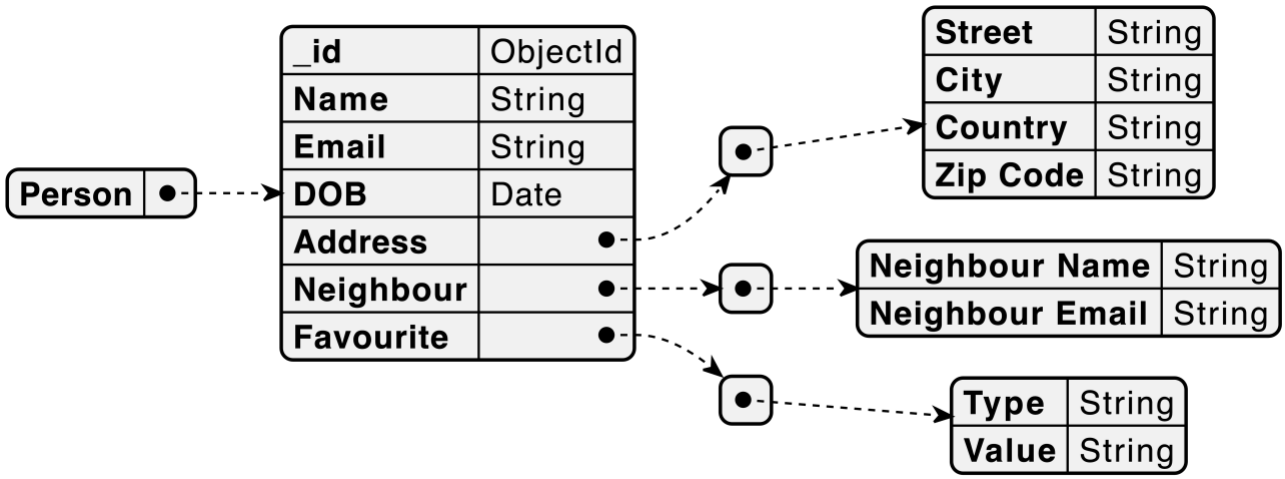


EXAMPLE OBJECT AFTER ALL UPDATES

```
_id: ObjectId('67c093f2d5d9285c0d6e0bbe')
Name : "Person 1"
Email : "person1@email.com"
DOB : 1995-03-15T00:00:00.000+00:00
▾ Address : Array (1)
  ▾ 0: Object
      Street : "12 Maple St"
      City : "London"
      Country : "England"
      Zip Code : "E1 6AN"
▾ Favourite : Array (3)
  ▾ 0: Object
      Type : "Book"
      Value : "A New Beginning"
  ▾ 1: Object
      Type : "Drink"
      Value : "Lemonade"
  ▾ 2: Object
      Type : "Activity"
      Value : "Outdoor Running"
▾ Neighbour : Array (2)
  ▾ 0: Object
      Neighbour Name : "Neighbour A"
      Neighbour Email : "neighbourA@email.com"
  ▾ 1: Object
      Neighbour Name : "Neighbour B"
      Neighbour Email : "neighbourB@email.com"
```

DOCUMENT DATA MODEL

The Document Data Model is created using PlantUML as below:



QUERIES

DISPLAY ALL PERSONS' NAME AND THEIR AGES IN YEARS

```
[
    {
        "$addFields": {
            "AgeInDays": {
                "$dateDiff": {
                    "startDate": "$DOB",
                    "endDate": "$$NOW",
                    "unit": "day"
                }
            }
        }
    },
    {
        "$addFields": {
            "Age": {
                "$floor": {
                    "$divide": ["$AgeInDays", 365]
                }
            }
        }
    },
    {
        "$project": {
            "Name": 1,
            "Age": "$Age"
    }
    }
]
```

	_id ObjectId	Name String	Age Double
1	ObjectId('67c093f2d5d9285…	"Person 1"	29
2	ObjectId('67c093f2d5d9285…	"Person 2"	31
3	ObjectId('67c093f2d5d9285…	"Person 3"	33
4	ObjectId('67c093f2d5d9285…	"Person 4"	26
5	ObjectId('67c093f2d5d9285…	"Person 5"	41
6	ObjectId('67c093f2d5d9285…	"Person 6"	35
7	ObjectId('67c093f2d5d9285…	"Person 7"	28
8	ObjectId('67c093f2d5d9285…	"Person 8"	35
9	ObjectId('67c093f2d5d9285…	"Person 9"	31
10	ObjectId('67c093f2d5d9285…	"Person 10"	27
11	ObjectId('67c093f2d5d9285…	"Person 11"	32
12	ObjectId('67c093f2d5d9285…	"Person 12"	39
13	ObjectId('67c093f2d5d9285…	"Person 13"	33
14	ObjectId('67c093f2d5d9285…	"Person 14"	38
15	ObjectId('67c093f2d5d9285…	"Person 15"	40
16	ObjectId('67c093f2d5d9285…	"Person 16"	34
17	ObjectId('67c093f2d5d9285…	"Person 17"	29
18	ObjectId('67c093f2d5d9285…	"Person 18"	30
19	ObjectId('67c093f2d5d9285…	"Person 19"	32
20	ObjectId('67c093f2d5d9285…	"Person 20"	36

GROUP PERSONS BY THEIR FAVOURITE DRINK AND RETURN AVERAGE AGE OF EACH GROUP

```
[
  {
    "$addFields": {
      "AgeInDays": {
        "$dateDiff": {
          "startDate": "$DOB",
          "endDate": "$$NOW",
          "unit": "day"
        }
      }
    }
  },
  {
    "$addFields": {
      "Age": {
        "$floor": {
          "$divide": ["$AgeInDays", 365]
        }
      }
    }
  },
  {
    "$unwind": "$Favourite"
  },
  {
    "$match": {
      "Favourite.Type": "Drink"
    }
  },
  {
    "$group": {
      "_id": "$Favourite.Value",
      "AverageAge": {
        "$avg": "$Age"
      }
    }
  },
  {
    "$project": {
      "_id": 0,
      "FavouriteDrink": "$_id",
      "AverageAge": 1
    }
  }
]
```

	FavouriteDrink String	AverageAge Double
1	"Green Tea"	35
2	"Fruit Juice"	31
3	"Coconut Water"	32.5
4	"Hot Chocolate"	31
5	"Fruit Smoothie"	27
6	"Sparkling Water"	32
7	"Iced Tea"	26
8	"Herbal Tea"	35.5
9	"Iced Coffee"	40
10	"Lemonade"	33.5
11	"Water"	35.5
12	"Coffee"	31
13	"Smoothie"	33

DISPLAY THE AVERAGE AGE OF PEOPLE WHO LIKE HIKING

```
[
  {
    "$addFields": {
      "AgeInDays": {
        "$dateDiff": {
          "startDate": "$DOB",
          "endDate": "$$NOW",
          "unit": "day"
        }
      }
    }
  },
  {
    "$addFields": {
      "Age": {
        "$floor": {
          "$divide": ["$AgeInDays", 365]
        }
      }
    }
  },
  {
    "$unwind": "$Favourite"
  },
  {
    "$match": {
      "Favourite.Type": "Activity",
      "Favourite.Value": "Hiking"
    }
  },
  {
    "$group": {
      "_id": "$Favourite.Value",
      "AverageAge": {
        "$avg": "$Age"
      }
    }
  },
  {
    "$project": {
      _id: 0,
      Favourite: "$_id",
      "AverageAge": 1
    }
  }
]
```

	Favourite String	AverageAge Double
1	"Hiking"	33

DISPLAY THE TOTAL NUMBER OF PEOPLE FROM EACH CITY AND SORT IT IN ASCENDING ORDER BY TOTAL NUMBER OF PEOPLE

```
[
  {
    $unwind: "$Address"
  },
  {
    $group: {
      _id: "$Address.City",
      "Number of People": {
        $sum: 1
      }
    }
  },
  {
    $sort: {
      "Number of People": 1
    }
  },
  {
    $project: {
      _id: 0,
      City: "$_id",
      "Number of People": 1
    }
  }
]
```

	City String	Number of People Int32
1	"London"	1
2	"Leicester"	1
3	"Liverpool"	1
4	"Birmingham"	1
5	"Nottingham"	1
6	"Southampton"	1
7	"Newcastle"	1
8	"Manchester"	1
9	"Norwich"	1
10	"Leeds"	1
11	"Sheffield"	1
12	"Cambridge"	1
13	"Bristol"	1
14	"Oxford"	1
15	"Glasgow"	1
16	"Edinburgh"	2
17	"Cardiff"	3

DISPLAY NAME OF PERSON(S) WHOSE NEIGHBOUR IS NEIGHBOUR C

```
[
  {
    "$unwind": "$Neighbour"
  },
  {
    "$match": {
      "Neighbour.Neighbour Name": "Neighbour C"
    }
  },
  {
    "$project": {
      _id: 0,
      "Name": 1,
      "Neighbour": "$Neighbour.Neighbour Name"
    }
  }
]
```

	Name String	**Neighbour** String
1	"Person 2"	"Neighbour C"