Advanced Databases UFCFU3-15-3: Task 2

Arthur Milner (21035478)

Database Model Diagram

Fig. 1 reflects how I will implement my NoSQL database.

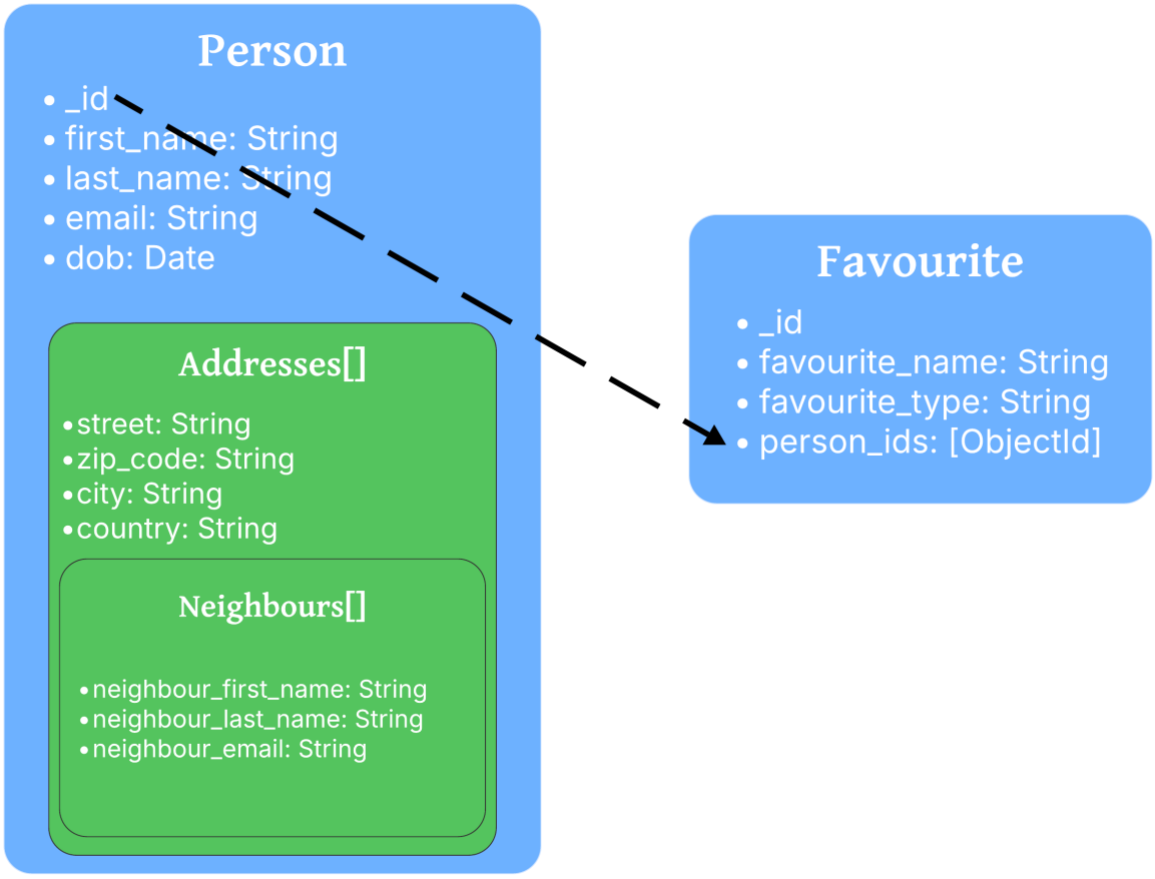


Figure 1 Diagram to display the intended database schema

MongoDB Design Decisions with Justifications

When converting the case study data into something usable by MongoDB, I heavily considered if, or to what extent, I will utilise referencing. Because much of the data will be shared across multiple people, and a single person can have multiple instances of the same object, such as a favourite, it would make sense to apply some referencing to prevent the data redundancy/duplication that is a consequence of embedding and only worsens as the databases scale increases. Referencing also simplifies changing details referenced in multiple people, for example, changing the incorrect name of a favourite would only need to be done once. The issue lies in the requirement of additional collections, and the high computational expense of NoSQL joins, resulting in slower query performance. Also introduced is the issue of managing dependencies to prevent stale references, which would require additional

overhead in a production environment, such as running scheduled jobs to clean-up references. Because of this I concluded minimal referencing, alongside more embedding, would be the most appropriate approach for NoSQL to get the benefits of both methods whilst reducing the negatives.

Favourites can be referenced by adding just a single collection, as seen in **Fig. 1**, I decided this because they are likely to be shared frequently, and updates are likely to be most common for a person's favourites. They are even seen to be shared multiple times within the small sample data. I have used an array of references to the persons who have the favourite, inside the Favourite collection, preventing the need to store any favourite data inside the person collection, making the favourite collection the centre for managing favourite details and references, maintaining simplicity.

Embedding features heavily in my design for faster reading of the data, reduced complexity in queries, the need for less collections, and consequently less joins. The general rule I followed was details which are not often shared or updated should be embedded. This is because the main negatives of data duplication and increased complexity in updating would consequently not be felt on the same scale as favourites, for an example. Unfortunately, some details of an address will be shared widely, such as country/city/zip code, but I felt referencing these values would create too much complexity in queries and require a lot of computational power to execute the joins required for a simple query retrieving basic address details. Additionally, these fields are unlikely to frequently change. Including person and address details in the same collection allows for simpler queries, particularly when address and person data is likely to be queried together.

I also decided to embed neighbours[] inside addresses[]. This allows for more flexibility should addresses require more neighbours in the future, alongside easier querying of neighbour details. You do not need to check both neighbour 1 and neighbour 2 details separately and can instead search within the embedded array for all associated neighbour details.

My database will have validation schema applied to each collection, this helps to enforce the datatypes of the fields, such as date for DoB to help with age calculation in queries, but also to apply referential integrity on referenced person IDs, and finally to mark fields as required. Additionally, unique indexes will be used to improve data validation, such as a person's email requiring uniqueness.

Overall, I believe my design aligns with capturing the benefits of using NoSQL, it promotes a flexible model which can easily be expanded. Combining minimal referencing with embedding ensures a balance between complexity, performance, and reduced data redundancy.

Validation

After creating the database using the MongoDB Compass GUI and adding the Person and Favourite collections, for each collection I created a validation schema and set them in the validation tab of MongoDB Compass with the appropriate validation action and level **(Fig. 2)**.

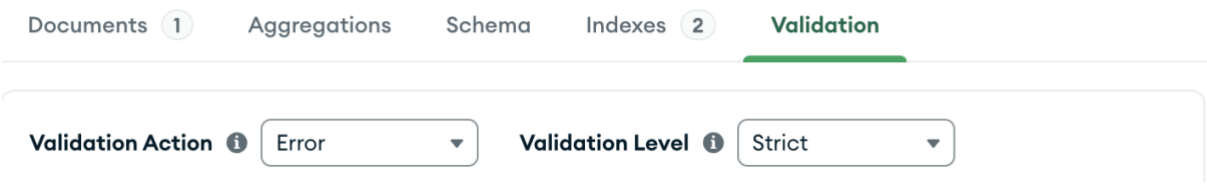


Figure 2 Validation action and level

Person Validation Schema

```
{
    "$jsonSchema": {
      "bsonType": "object",
      "required": ["first_name", "last_name", "email", "dob", "addresses"],
      "properties": {
            "first_name": {
            "bsonType": "string",
            "description": "Must be a string."
            },
            "last_name": {
            "bsonType": "string",
            "description": "Must be a string."
            },
            "email": {
            "bsonType": "string",
            "pattern": "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$",
            "description": "Must be a valid email."
            },
            "dob": {
            "bsonType": "date",
            "description": "Must be a date."
            },
            "addresses": {
                "bsonType": "array",
                "minItems": 1,
                "items": {
                    "bsonType": "object",
                    "required": ["street", "zip_code", "city", "country", "neighbours"],
                    "properties": {
                        "street": {
                            "bsonType": "string",
                            "description": "Must be a string."
                        },
                        "zip_code": {
                            "bsonType": "string",
                            "description": "Must be a string."
                        },
                        "city": {
                            "bsonType": "string",
                            "description": "Must be a string."
                        },
                        "country": {
                            "bsonType": "string",
                            "description": "Must be a string."
                        },
                        "neighbours": {
                            "bsonType": "array",
                            "minItems": 1,
                            "items": {
                                "bsonType": "object",
                                "required": ["neighbour_first_name", "neighbour_last_name", "neighbour_email"],
                                "properties": {
                                    "neighbour_first_name": {
                                        "bsonType": "string",
                                        "description": "Must be a string."
                                    },
                                    "neighbour_last_name": {
                                        "bsonType": "string",
                                        "description": "Must be a string."
                                    },
                                    "neighbour_email": {
                                        "bsonType": "string",
                                        "pattern": "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$",
                                        "description": "Must be a valid email."
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Figure 2 Validation schema for Person collection

Fig. 2 presents the Person validation schema and how it enforces regular expression for the email fields, alongside defining each field's expected data type, and stating which fields are required.

Favourites Validation Schema

```
{
    "$jsonSchema": {
      "bsonType": "object",
      "required": ["favourite_name","favourite_type","person_ids"],
      "properties": {
        "favourite_name": {
          "bsonType": "string",
          "description": "Must be a string."
        },
        "favourite_type": {
          "bsonType": "string",
          "description": "Must be a string."
        },
        "person_ids": {
        "bsonType": "array",
        "items": {
          "bsonType": "objectId"
        },
        "description": "Must contain valid person IDs."
        }
      }
    }
}
```

Figure 3 Validation schema for Favourite collection

Fig. 3 displays how the Favourite validation schema helps ensure referential integrity by enforcing checks for a valid objectId/reference ID on the favourites, alongside setting data types and fields as required.

Unique Indexes

Fig. 4 presents the code used to create the unique indexes, ensuring person email and favourite name are unique. **Figs. 5 & 6** show the unique indexes inside MongoDB compass.

```
db.Person.createIndex({ email: 1 }, { unique: true });
db.Favourite.createIndex({ favourite_name: 1 }, { unique: true });
```

Figure 4 MongoDB shell command to create unique indexes

Figure 5 Person email unique index in MongoDB compass

Figure 6 Favourite favourite_name unique index in MongoDB compass

Inserting the Data

The exported JSON of the contents of the inserted data and screenshots of an example document's structure as seen in MongoDB compass are included in Appendix A.

Inserting Person

MongoDB Command:

```
db.Person.insertMany([
    {
        "first_name": "Person",
        "last_name": "1",
        "email": "person1@email.com",
        "dob": new Date("1995-03-15"),
        "addresses": [
            {
                "street": "12 Maple St",
                "zip_code": "E1 6AN",
                "city": "London",
                "country": "England",
                "neighbours": [
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "A",
                        "neighbour_email": "neighborA@email.com"
                    },
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "B",
                        "neighbour_email": "neighborB@email.com"
                    }
                ]
            }
        ]
    },
    {
        "first_name": "Person",
        "last_name": "2",
        "email": "person2@email.com",
        "dob": new Date("1993-06-22"),
        "addresses": [
            {
                "street": "45 Oak Ave",
                "zip_code": "M1 2WD",
                "city": "Manchester",
                "country": "England",
                "neighbours": [
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "C",
                        "neighbour_email": "neighborC@email.com"
                    },
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "D",
                        "neighbour_email": "neighborD@email.com"
                    }
                ]
            }
        ]
    },
    {
        "first_name": "Person",
        "last_name": "3",
        "email": "person3@email.com",
        "dob": new Date("1991-09-10"),
        "addresses": [
            {
                "street": "89 Pine Rd",
                "zip_code": "B1 1AB",
                "city": "Birmingham",
                "country": "England",
                "neighbours": [
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "E",
                        "neighbour_email": "neighborE@email.com"
                    },
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "F",
                        "neighbour_email": "neighborF@email.com"
                    }
                ]
            }
        ]
    },
```

Figure 7 MongoDB shell command to insert into Person collection (1)

Figure 8 MongoDB shell command to insert into Person collection (2)

Figure 9 MongoDB shell command to insert into Person collection (3)

Figure 10 MongoDB shell command to insert into Person collection (4)

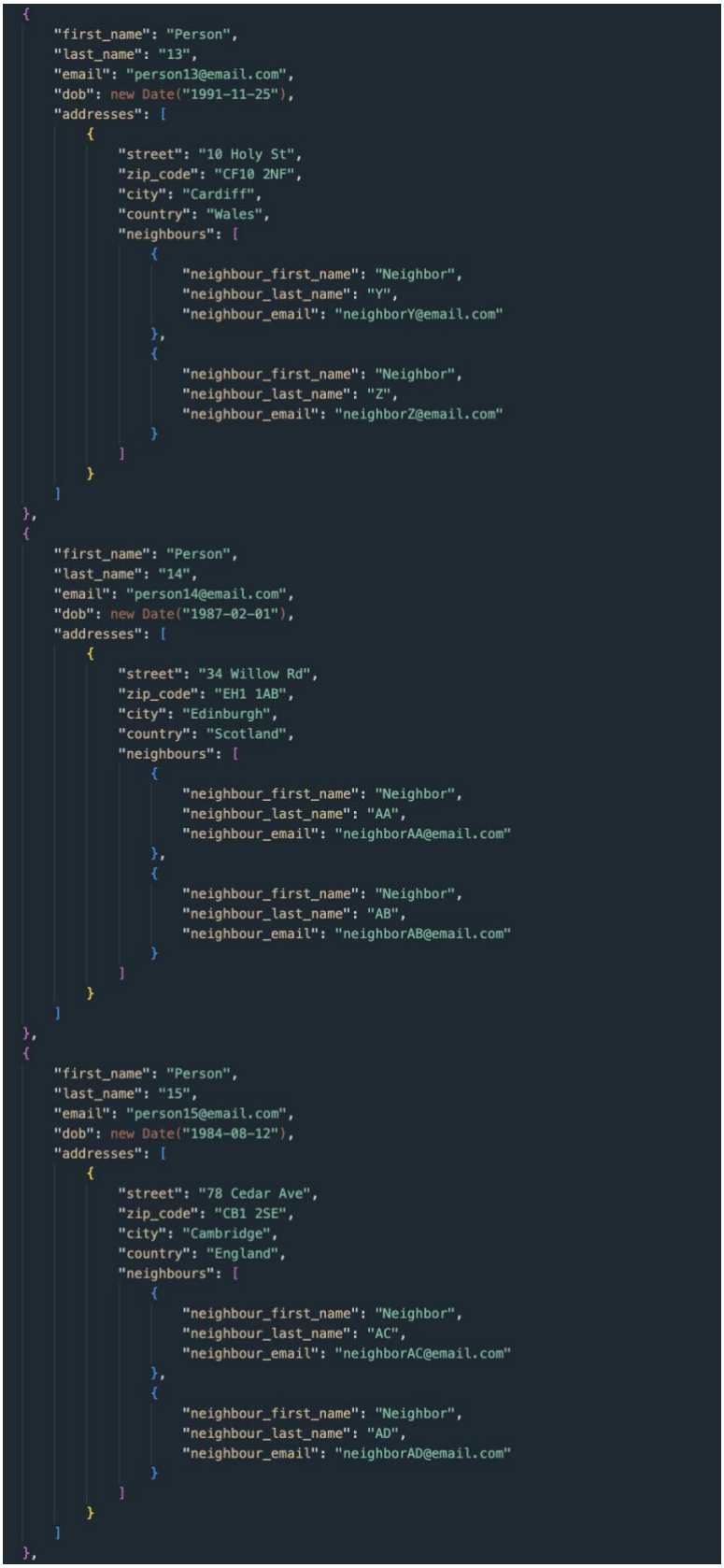


Figure 11 MongoDB shell command to insert into Person collection (5)

Figure 12 MongoDB shell command to insert into Person collection (6)

```
    {
        "first_name": "Person",
        "last_name": "19",
        "email": "person19@email.com",
        "dob": new Date("1992-12-11"),
        "addresses": [
            {
                "street": "56 Oak Rd",
                "zip_code": "NR1 4BE",
                "city": "Norwich",
                "country": "England",
                "neighbours": [
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "AK",
                        "neighbour_email": "neighborAK@email.com"
                    },
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "AL",
                        "neighbour_email": "neighborAL@email.com"
                    }
                ]
            }
        ]
    },
    {
        "first_name": "Person",
        "last_name": "20",
        "email": "person20@email.com",
        "dob": new Date("1988-09-25"),
        "addresses": [
            {
                "street": "89 Pine Ave",
                "zip_code": "CF10 3BC",
                "city": "Cardiff",
                "country": "Wales",
                "neighbours": [
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "AM",
                        "neighbour_email": "neighborAM@email.com"
                    },
                    {
                        "neighbour_first_name": "Neighbor",
                        "neighbour_last_name": "AN",
                        "neighbour_email": "neighborAN@email.com"
                    }
                ]
            }
        ]
    }
]);
```

Figure 13 MongoDB shell command to insert into Person collection (7)

MongoDB Shell Output:

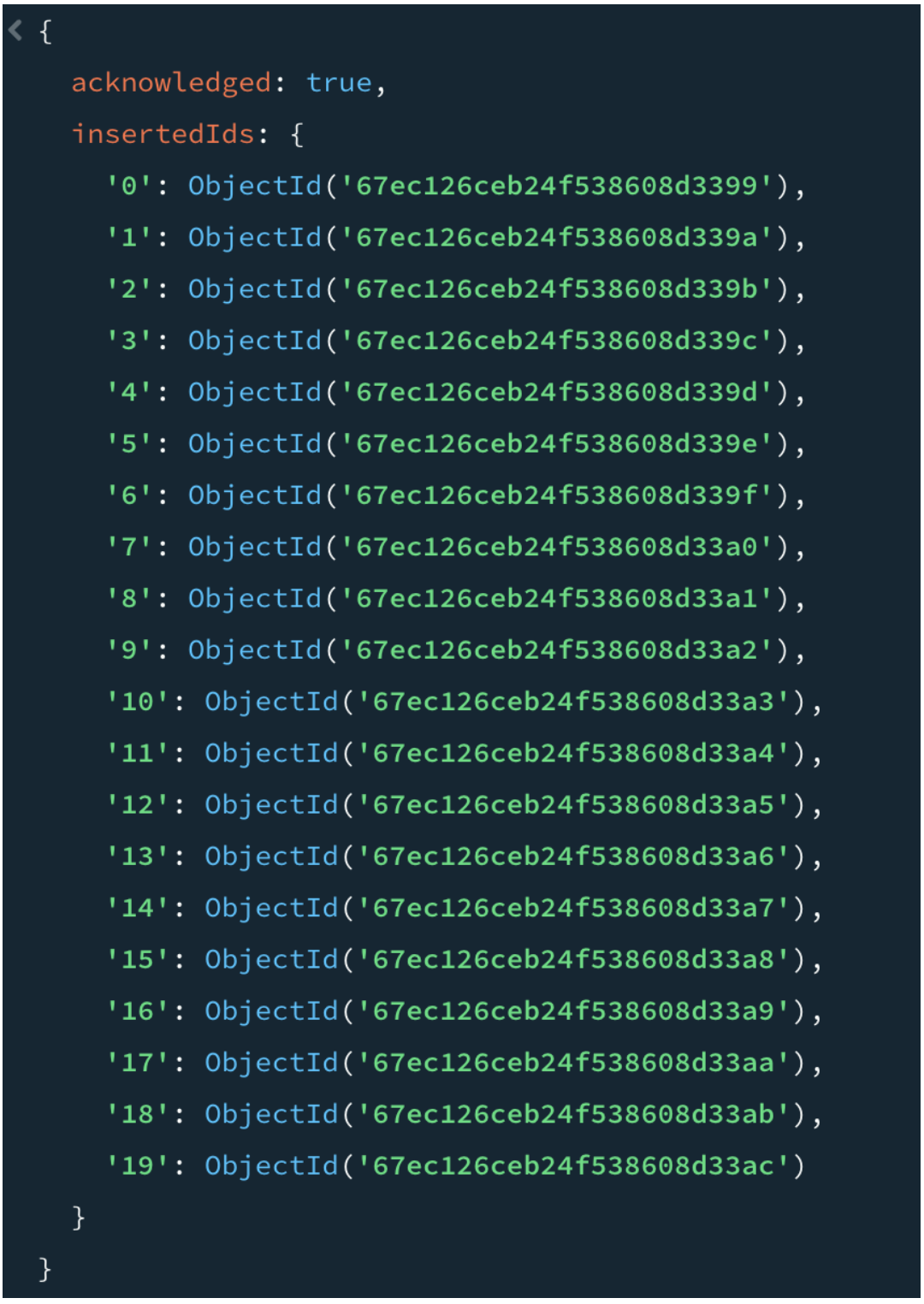


Figure 14 MongoDB shell output from inserting persons

Inserting Favourite

To insert on Favourite, I first got all the relevant person IDs and stored them in variables within MongoDB shell, allowing me to reference them in my insert many statement.

Storing the Person IDs:

```
var person1 = db.Person.findOne({ "email": "person1@email.com" })._id;
var person2 = db.Person.findOne({ "email": "person2@email.com" })._id;
var person3 = db.Person.findOne({ "email": "person3@email.com" })._id;
var person4 = db.Person.findOne({ "email": "person4@email.com" })._id;
var person5 = db.Person.findOne({ "email": "person5@email.com" })._id;
var person6 = db.Person.findOne({ "email": "person6@email.com" })._id;
var person7 = db.Person.findOne({ "email": "person7@email.com" })._id;
var person8 = db.Person.findOne({ "email": "person8@email.com" })._id;
var person9 = db.Person.findOne({ "email": "person9@email.com" })._id;
var person10 = db.Person.findOne({ "email": "person10@email.com" })._id;
var person11 = db.Person.findOne({ "email": "person11@email.com" })._id;
var person12 = db.Person.findOne({ "email": "person12@email.com" })._id;
var person13 = db.Person.findOne({ "email": "person13@email.com" })._id;
var person14 = db.Person.findOne({ "email": "person14@email.com" })._id;
var person15 = db.Person.findOne({ "email": "person15@email.com" })._id;
var person16 = db.Person.findOne({ "email": "person16@email.com" })._id;
var person17 = db.Person.findOne({ "email": "person17@email.com" })._id;
var person18 = db.Person.findOne({ "email": "person18@email.com" })._id;
var person19 = db.Person.findOne({ "email": "person19@email.com" })._id;
var person20 = db.Person.findOne({ "email": "person20@email.com" })._id;
```

Figure 15 Storing person IDs in variables within the MongoDB shell

MongoDB Command:

```
db.Favourite.insertMany([
    {
        "favourite_name": "A New Beginning",
        "favourite_type": "Book",
        "person_ids": [person1]
    },
    {
        "favourite_name": "The Road to Success",
        "favourite_type": "Book",
        "person_ids": [person2]
    },
    {
        "favourite_name": "Endless Possibilities",
        "favourite_type": "Book",
        "person_ids": [person3]
    },
    {
        "favourite_name": "Journey of Life",
        "favourite_type": "Book",
        "person_ids": [person4]
    },
    {
        "favourite_name": "The Adventure Continues",
        "favourite_type": "Book",
        "person_ids": [person5, person20]
    },
    {
        "favourite_name": "Finding Inner Peace",
        "favourite_type": "Book",
        "person_ids": [person6]
    },
    {
        "favourite_name": "Exploring New Horizons",
        "favourite_type": "Book",
        "person_ids": [person7]
    },
    {
        "favourite_name": "The Great Journey",
        "favourite_type": "Book",
        "person_ids": [person8]
    },
    {
        "favourite_name": "The Power of Change",
        "favourite_type": "Book",
        "person_ids": [person9]
    },
    {
        "favourite_name": "New Beginnings Await",
        "favourite_type": "Book",
        "person_ids": [person10, person13]
    },
    {
        "favourite_name": "Wandering Souls",
        "favourite_type": "Book",
        "person_ids": [person11]
    },
    {
        "favourite_name": "Freedom and Choice",
        "favourite_type": "Book",
        "person_ids": [person12]
    },
    {
        "favourite_name": "Chasing Dreams",
        "favourite_type": "Book",
        "person_ids": [person14]
    },
    {
        "favourite_name": "The Endless Journey",
        "favourite_type": "Book",
        "person_ids": [person15]
    },
    {
        "favourite_name": "The Future Ahead",
        "favourite_type": "Book",
        "person_ids": [person16]
    },
    {
        "favourite_name": "The Path to Glory",
        "favourite_type": "Book",
        "person_ids": [person17]
    },
    {
        "favourite_name": "Life's Adventure",
        "favourite_type": "Book",
        "person_ids": [person18]
    },
    {
        "favourite_name": "Into the Wild",
        "favourite_type": "Book",
        "person_ids": [person19]
    },
```

Figure 16 MongoDB shell command to insert into Favourite collection (1)

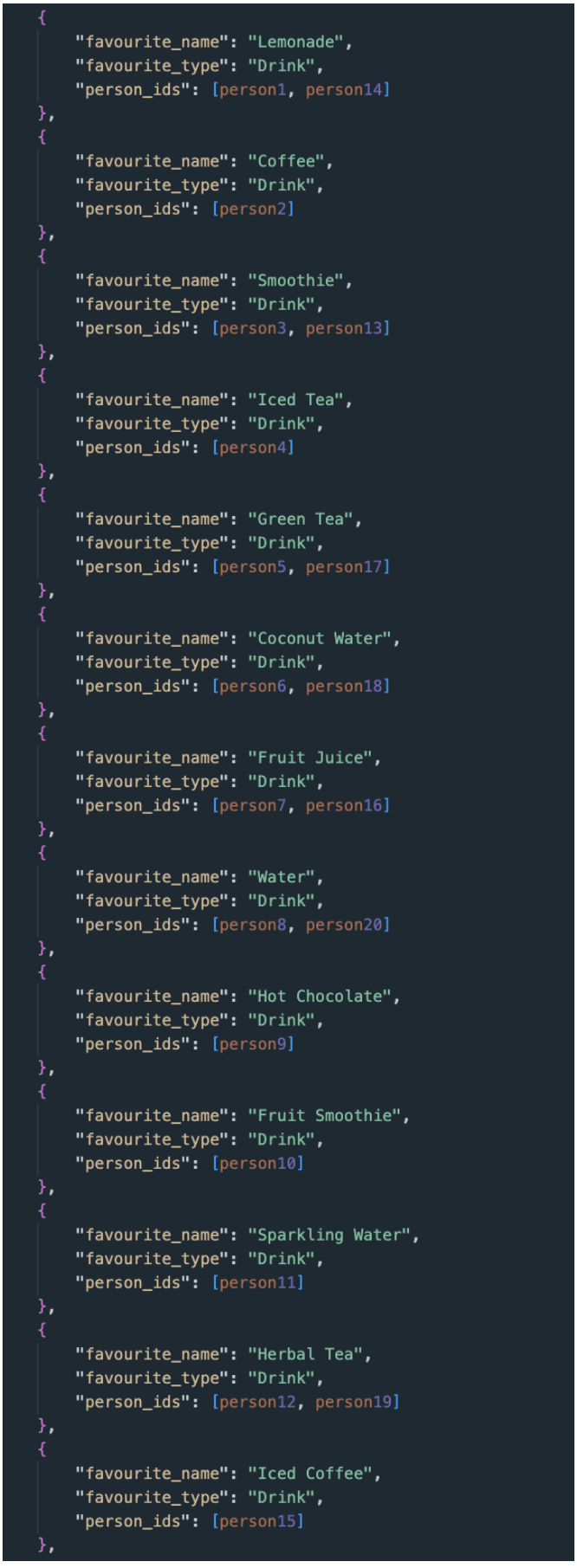


Figure 17 MongoDB shell command to insert into Favourite collection (2)

```
    {
        "favourite_name": "Outdoor Running",
        "favourite_type": "Activity",
        "person_ids": [person1]
    },
    {
        "favourite_name": "Hiking",
        "favourite_type": "Activity",
        "person_ids": [person2, person8, person13, person18, person20]
    },
    {
        "favourite_name": "Swimming",
        "favourite_type": "Activity",
        "person_ids": [person3, person19]
    },
    {
        "favourite_name": "Traveling",
        "favourite_type": "Activity",
        "person_ids": [person4]
    },
    {
        "favourite_name": "Gardening",
        "favourite_type": "Activity",
        "person_ids": [person5, person17]
    },
    {
        "favourite_name": "Reading",
        "favourite_type": "Activity",
        "person_ids": [person6]
    },
    {
        "favourite_name": "Cycling",
        "favourite_type": "Activity",
        "person_ids": [person7, person15]
    },
    {
        "favourite_name": "Skiing",
        "favourite_type": "Activity",
        "person_ids": [person9]
    },
    {
        "favourite_name": "Jogging",
        "favourite_type": "Activity",
        "person_ids": [person10]
    },
    {
        "favourite_name": "Rock Climbing",
        "favourite_type": "Activity",
        "person_ids": [person11]
    },
    {
        "favourite_name": "Yoga",
        "favourite_type": "Activity",
        "person_ids": [person12, person16]
    },
    {
        "favourite_name": "Running",
        "favourite_type": "Activity",
        "person_ids": [person14]
    }
]);
```

Figure 18 MongoDB shell command to insert into Favourite collection (3)

MongoDB Shell Output:

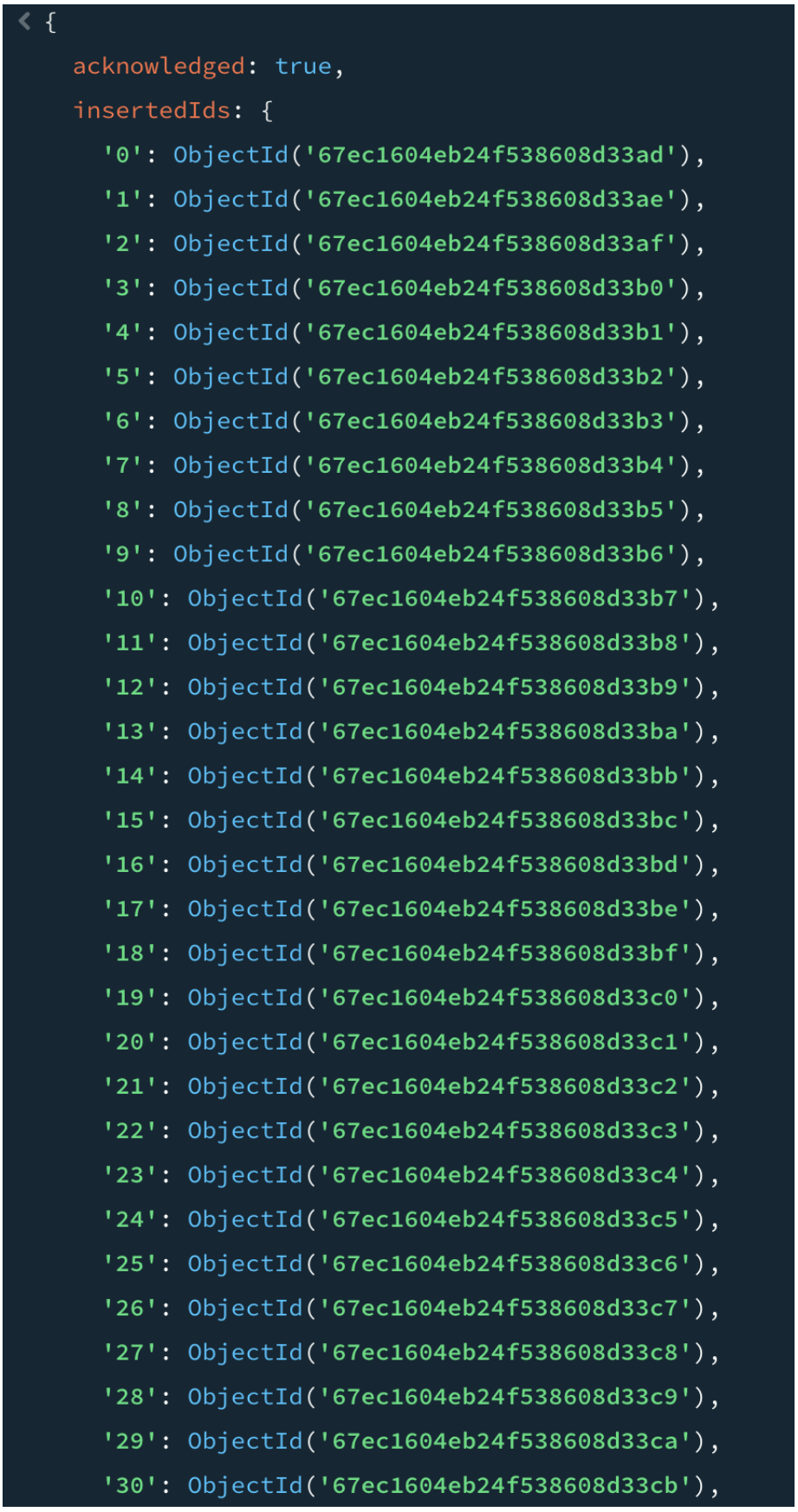


Figure 19 MongoDB shell output from inserting into Favourite collection (1)

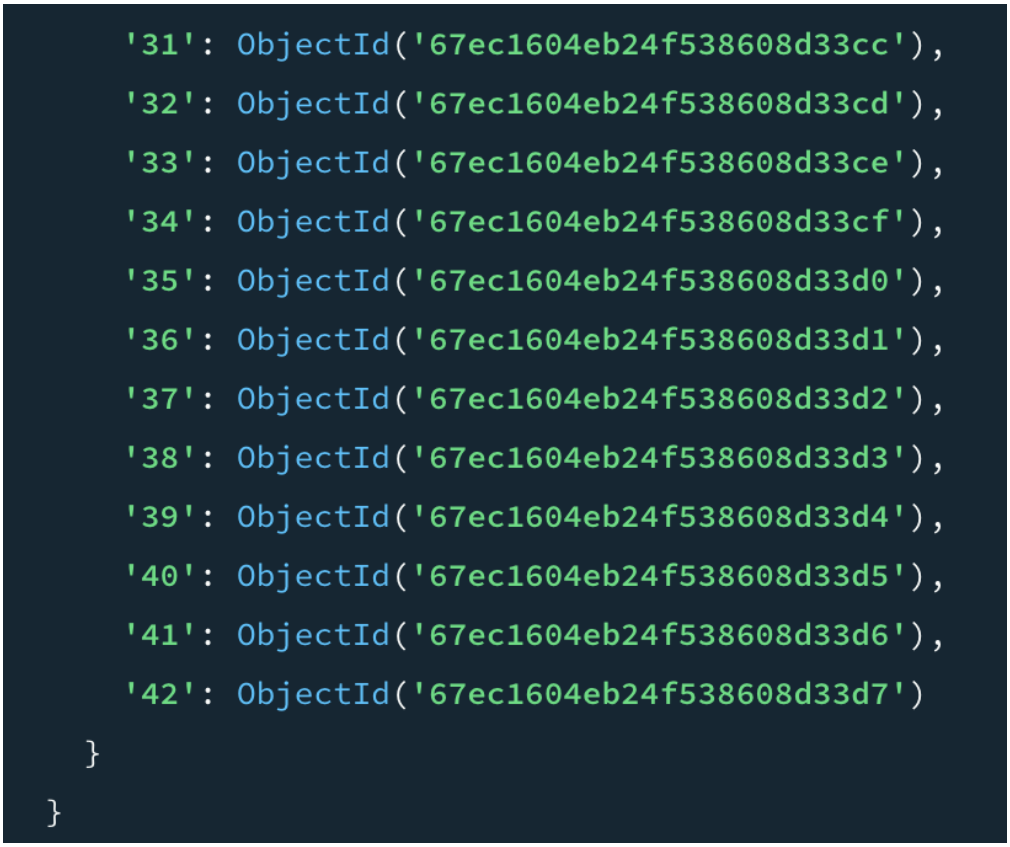


Figure 20 MongoDB shell output from inserting into Favourite collection (2)

Queries

1) Display person's name and their age in years.

$NOW is used to obtain the current date inside the query. $project is used to define what fields I want returned in my query, in this case first_name, last_name, and a new field age. The age field was calculated by combining **Stack Overflow, 2022** and **Stack Overflow, 2019**, I first attempted to use $dateDiff but it wasn't giving accurate results, I suspect due to an issue with leap years. $floor is used to ensure the age is an integer.

MongoDB Shell Command:

```
db.Person.aggregate([
  {
    $project: {
      first_name: 1,
      last_name: 1,
      age: {
        $floor: {
          $divide: [
            { $subtract: ["$$NOW", "$dob"] },
            1000 * 60 * 60 * 24 * 365.25
          ]
        }
      }
    }
  }
]);
```

Figure 21 Query 1 MongoDB shell command

MongoDB Shell Output:

Figure 22 Query 1 MongoDB shell output (1)

Figure 23 Query 1 MongoDB shell output (2)

```
{
  _id: ObjectId('67ec126ceb24f538608d33a7'),
  first_name: 'Person',
  last_name: '15',
  age: 40
}
{
  _id: ObjectId('67ec126ceb24f538608d33a8'),
  first_name: 'Person',
  last_name: '16',
  age: 35
}
{
  _id: ObjectId('67ec126ceb24f538608d33a9'),
  first_name: 'Person',
  last_name: '17',
  age: 29
}
{
  _id: ObjectId('67ec126ceb24f538608d33aa'),
  first_name: 'Person',
  last_name: '18',
  age: 30
}
{
  _id: ObjectId('67ec126ceb24f538608d33ab'),
  first_name: 'Person',
  last_name: '19',
  age: 32
}
{
  _id: ObjectId('67ec126ceb24f538608d33ac'),
  first_name: 'Person',
  last_name: '20',
  age: 36
}
```


Figure 24 Query 1 MongoDB shell output (3)

2) Group Persons by their favourite drink and return average age of each group.

Firstly, I use $match to filter the Favourite collection, this is so the query only checks favourites of the type “Drink”. Executing this first improves query efficiency as subsequent steps will involve fewer documents.

$lookup is used to join Favourite and Person, enabling the access of Person details through the array of referenced person_ids in Favourite and the _id field in Person. $unwind is then applied to the persons array returned by the $lookup, this ensures the array is split into a separate document for each person. This is so the ages are calculated for each person individually. $set is used to create the age field, with the same calculation as query 1.

$group is used to group by favourite_name, and $avg is used inside the $group to return the average age per drink name. Finally, $project is used to rename the _id from the $group to drink_name, and $round to round the average ages returned to one decimal place for visual clarity.

MongoDB Query:

```
db.Favourite.aggregate([
  {
    $match: { favourite_type: "Drink" }
  },
  {
    $lookup: {
      from: "Person",
      localField: "person_ids",
      foreignField: "_id",
      as: "persons"
    }
  },
  {
    $unwind: "$persons"
  },
  {
    $set: {
      age: {
        $floor: {
          $divide: [
            { $subtract: ["$$NOW", "$persons.dob"] },
            1000 * 60 * 60 * 24 * 365.25
          ]
        }
      }
    }
  },
  {
    $group: {
      _id: "$favourite_name",
      average_age: { $avg: "$age" }
    }
  },
  {
    $project: {
      drink_name: "$_id",
      average_age: { $round: ["$average_age", 1] },
      _id: 0
    }
  }
]);
```

Figure 25 Query 2 MongoDB shell command

MongoDB Shell Output:

Figure 26 Query 2 MongoDB shell output

3) Display average age of people who likes Hiking.

Query 3 is very similar to the structure of query 2, the key differences are using favourite_name in $match to filter by only “Hiking” and using null in the $group _id to group all results into one document/a single average value.

MongoDB Query:

```
db.Favourite.aggregate([
  {
    $match: { favourite_name: "Hiking" }
  },
  {
    $lookup: {
      from: "Person",
      localField: "person_ids",
      foreignField: "_id",
      as: "persons"
    }
  },
  {
    $unwind: "$persons"
  },
  {
    $set: {
      age: {
        $floor: {
          $divide: [
            { $subtract: ["$$NOW", "$persons.dob"] },
            1000 * 60 * 60 * 24 * 365.25
          ]
        }
      }
    }
  },
  {
    $group: {
      _id: null,
      average_age: { $avg: "$age" }
    }
  },
  {
    $project: {
      average_age: { $round: ["$average_age", 1] },
      _id: 0
    }
  }
]);
```

Figure 27 Query 3 MongoDB shell command

MongoDB Shell Output:

Figure 28 Query 3 MongoDB shell output

4) Display the total number of people from each City and sort it in ascending order by total number of people.

First $unwind ensures each address is counted individually, then $group is used to get the $sum of each unique city value. $project then renames _id to city, finally $sort on population ensures the results are sorted from low to high population.

MongoDB Query:

```
db.Person.aggregate([
  {
    $unwind: "$addresses"
  },
  {
    $group: {
      _id: "$addresses.city",
      population: { $sum: 1 }
    }
  },
  {
    $project: {
      city: "$_id",
      population: 1,
      _id: 0
    }
  },
  {
    $sort: { population: 1 }
  }
]);
```

Figure 29 Query 4 MongoDB shell command

MongoDB Shell Output:

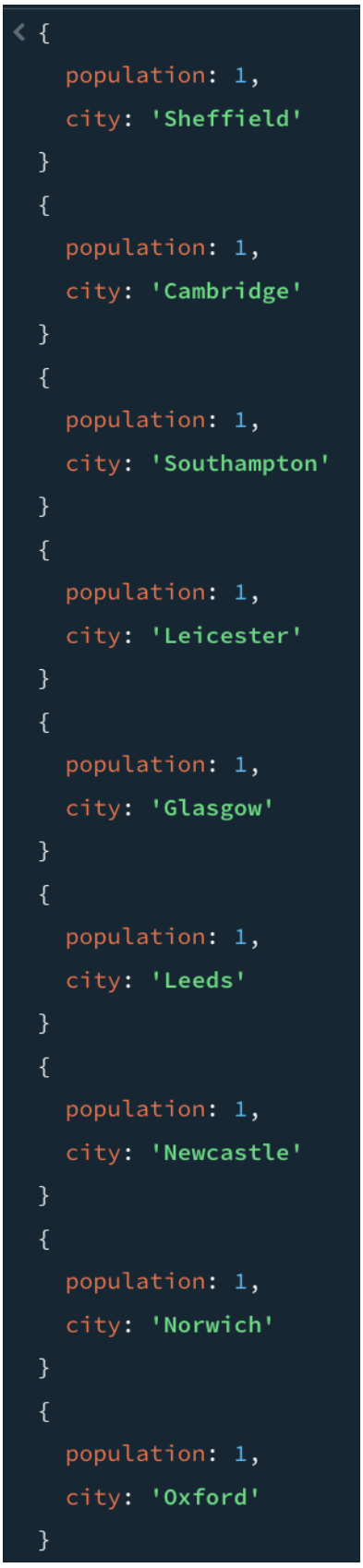


Figure 30 Query 4 MongoDB shell output (1)

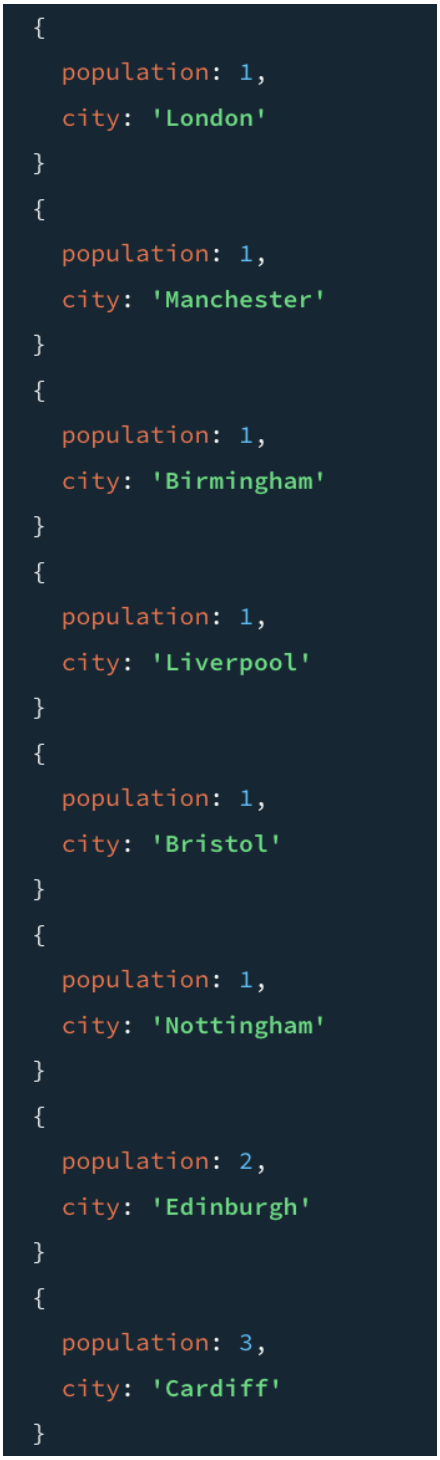


Figure 31 Query 4 MongoDB shell output (2)

5) Display name of person(s) whose neighbour is neighbour C

For this query I used find() instead of aggregate() as it satisfies the requirement whilst keeping the query simple and efficient.

MongoDB Query:

```
db.Person.find(
  {
    "addresses.neighbours.neighbour_first_name": "Neighbor",
    "addresses.neighbours.neighbour_last_name": "C"
  },
  {
    _id: 0,
    first_name: 1,
    last_name: 1,
    email: 1
  }
);
```

Figure 32 Query 5 MongoDB shell command

MongoDB Shell Output:

```
< {
    first_name: 'Person',
    last_name: '2',
    email: 'person2@email.com'
  }
```

Figure 33 Query 5 MongoDB shell output

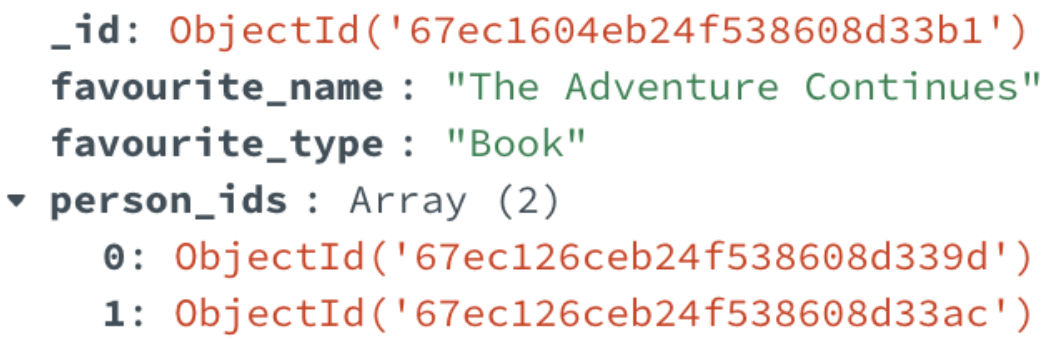
Bibliography

- Stack Overflow (2022) ***Convert date difference to years to calculate age in MongoDB***. Available from: https://stackoverflow.com/questions/39381450/convert-date-difference-to-years-to-calculate-age-in-mongodb [Accessed March 4th 2025]
- Stack Overflow (2019) ***Get age from Birthdate***. Available from: https://stackoverflow.com/questions/10008050/get-age-from-birthdate [Accessed March 4th 2025]

Appendix A (Inserted Data)

Figure 1 (Favourite Data)

Example document structure as seen in MongoDB compass:



Exported JSON of Inserted data:


```
[{
  "_id": {
    "$oid": "67ec1604eb24f538608d33ad"
  },
  "favourite_name": "A New Beginning",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d3399"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33ae"
  },
  "favourite_name": "The Road to Success",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339a"
    }
  ]
```

```
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33af"
  },
  "favourite_name": "Endless Possibilities",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339b"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b0"
  },
  "favourite_name": "Journey of Life",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339c"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b1"
  },
  "favourite_name": "The Adventure Continues",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339d"
    },
    {
      "$oid": "67ec126ceb24f538608d33ac"
    }
  ]
},
```

```
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b2"
  },
  "favourite_name": "Finding Inner Peace",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339e"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b3"
  },
  "favourite_name": "Exploring New Horizons",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339f"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b4"
  },
  "favourite_name": "The Great Journey",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a0"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b5"
  },
```

```
  "favourite_name": "The Power of Change",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a1"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b6"
  },
  "favourite_name": "New Beginnings Await",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a2"
    },
    {
      "$oid": "67ec126ceb24f538608d33a5"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b7"
  },
  "favourite_name": "Wandering Souls",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a3"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b8"
  },
  "favourite_name": "Freedom and Choice",
```

```
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a4"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33b9"
  },
  "favourite_name": "Chasing Dreams",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a6"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33ba"
  },
  "favourite_name": "The Endless Journey",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a7"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33bb"
  },
  "favourite_name": "The Future Ahead",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a8"
```

```
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33bc"
  },
  "favourite_name": "The Path to Glory",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a9"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33bd"
  },
  "favourite_name": "Life’s Adventure",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33aa"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33be"
  },
  "favourite_name": "Into the Wild",
  "favourite_type": "Book",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33ab"
    }
  ]
},
{
```

```
  "_id": {
    "$oid": "67ec1604eb24f538608d33bf"
  },
  "favourite_name": "Lemonade",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d3399"
    },
    {
      "$oid": "67ec126ceb24f538608d33a6"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c0"
  },
  "favourite_name": "Coffee",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339a"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c1"
  },
  "favourite_name": "Smoothie",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339b"
    },
    {
      "$oid": "67ec126ceb24f538608d33a5"
    }
  ]
```

```
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c2"
  },
  "favourite_name": "Iced Tea",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339c"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c3"
  },
  "favourite_name": "Green Tea",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339d"
    },
    {
      "$oid": "67ec126ceb24f538608d33a9"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c4"
  },
  "favourite_name": "Coconut Water",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339e"
    },
    {
      "$oid": "67ec126ceb24f538608d33aa"
```

```
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c5"
  },
  "favourite_name": "Fruit Juice",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339f"
    },
    {
      "$oid": "67ec126ceb24f538608d33a8"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c6"
  },
  "favourite_name": "Water",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a0"
    },
    {
      "$oid": "67ec126ceb24f538608d33ac"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c7"
  },
  "favourite_name": "Hot Chocolate",
  "favourite_type": "Drink",
  "person_ids": [
```

```
    {
      "$oid": "67ec126ceb24f538608d33a1"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c8"
  },
  "favourite_name": "Fruit Smoothie",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a2"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33c9"
  },
  "favourite_name": "Sparkling Water",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a3"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33ca"
  },
  "favourite_name": "Herbal Tea",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a4"
    },
    {
```

```
      "$oid": "67ec126ceb24f538608d33ab"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33cb"
  },
  "favourite_name": "Iced Coffee",
  "favourite_type": "Drink",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a7"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33cc"
  },
  "favourite_name": "Outdoor Running",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d3399"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33cd"
  },
  "favourite_name": "Hiking",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339a"
    },
    {
      "$oid": "67ec126ceb24f538608d33a0"
```

```
    },
    {
      "$oid": "67ec126ceb24f538608d33a5"
    },
    {
      "$oid": "67ec126ceb24f538608d33aa"
    },
    {
      "$oid": "67ec126ceb24f538608d33ac"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33ce"
  },
  "favourite_name": "Swimming",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339b"
    },
    {
      "$oid": "67ec126ceb24f538608d33ab"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33cf"
  },
  "favourite_name": "Traveling",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339c"
    }
  ]
},
{
```

```
  "_id": {
    "$oid": "67ec1604eb24f538608d33d0"
  },
  "favourite_name": "Gardening",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339d"
    },
    {
      "$oid": "67ec126ceb24f538608d33a9"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d1"
  },
  "favourite_name": "Reading",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339e"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d2"
  },
  "favourite_name": "Cycling",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d339f"
    },
    {
      "$oid": "67ec126ceb24f538608d33a7"
    }
  ]
```

```
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d3"
  },
  "favourite_name": "Skiing",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a1"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d4"
  },
  "favourite_name": "Jogging",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a2"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d5"
  },
  "favourite_name": "Rock Climbing",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a3"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d6"
```

```
  },
  "favourite_name": "Yoga",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a4"
    },
    {
      "$oid": "67ec126ceb24f538608d33a8"
    }
  ]
},
{
  "_id": {
    "$oid": "67ec1604eb24f538608d33d7"
  },
  "favourite_name": "Running",
  "favourite_type": "Activity",
  "person_ids": [
    {
      "$oid": "67ec126ceb24f538608d33a6"
    }
  ]
}]
```


Figure 2 (Person Data)

Example document structure as seen in MongoDB compass:

```
_id: ObjectId('67ec126ceb24f538608d3399')
first_name : "Person"
last_name : "1"
email : "person1@email.com"
dob : 1995-03-15T00:00:00.000+00:00
addresses : Array (1)
  0: Object
    street : "12 Maple St"
    zip_code : "E1 6AN"
    city : "London"
    country : "England"
    neighbours : Array (2)
      0: Object
        neighbour_first_name : "Neighbor"
        neighbour_last_name : "A"
        neighbour_email : "neighborA@email.com"
      1: Object
        neighbour_first_name : "Neighbor"
        neighbour_last_name : "B"
        neighbour_email : "neighborB@email.com"
```

Exported JSON of inserted data:


```
[{
 "_id": {
  "$oid": "67ec126ceb24f538608d3399"
 },
 "first_name": "Person",
 "last_name": "1",
 "email": "person1@email.com",
 "dob": {
  "$date": "1995-03-15T00:00:00.000Z"
 },
 "addresses": [
  {
   "street": "12 Maple St",
   "zip_code": "E1 6AN",
   "city": "London",
   "country": "England",
   "neighbours": [
    {
     "neighbour_first_name": "Neighbor",
```

```
        "neighbour_last_name": "A",
        "neighbour_email": "neighborA@email.com"
       },
       {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "B",
        "neighbour_email": "neighborB@email.com"
       }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d339a"
  },
  "first_name": "Person",
  "last_name": "2",
  "email": "person2@email.com",
  "dob": {
    "$date": "1993-06-22T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "45 Oak Ave",
      "zip_code": "M1 2WD",
      "city": "Manchester",
      "country": "England",
      "neighbours": [
       {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "C",
        "neighbour_email": "neighborC@email.com"
       },
       {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "D",
        "neighbour_email": "neighborD@email.com"
       }
      ]
```

```
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d339b"
  },
  "first_name": "Person",
  "last_name": "3",
  "email": "person3@email.com",
  "dob": {
    "$date": "1991-09-10T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "89 Pine Rd",
      "zip_code": "B1 1AB",
      "city": "Birmingham",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "E",
          "neighbour_email": "neighborE@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "F",
          "neighbour_email": "neighborF@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d339c"
  },
  "first_name": "Person",
  "last_name": "4",
```

```
  "email": "person4@email.com",
  "dob": {
    "$date": "1998-12-05T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "23 Birch St",
      "zip_code": "EH1 1YZ",
      "city": "Edinburgh",
      "country": "Scotland",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "G",
          "neighbour_email": "neighborG@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "H",
          "neighbour_email": "neighborH@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d339d"
  },
  "first_name": "Person",
  "last_name": "5",
  "email": "person5@email.com",
  "dob": {
    "$date": "1983-11-30T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "67 Cedar Ln",
      "zip_code": "BS1 3XE",
      "city": "Bristol",
```

```
    "country": "England",
    "neighbours": [
     {
      "neighbour_first_name": "Neighbor",
      "neighbour_last_name": "I",
      "neighbour_email": "neighborI@email.com"
     },
     {
      "neighbour_first_name": "Neighbor",
      "neighbour_last_name": "J",
      "neighbour_email": "neighborJ@email.com"
     }
    ]
   }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d339e"
  },
  "first_name": "Person",
  "last_name": "6",
  "email": "person6@email.com",
  "dob": {
    "$date": "1989-07-18T00:00:00.000Z"
  },
  "addresses": [
   {
    "street": "56 Elm St",
    "zip_code": "L1 1AA",
    "city": "Liverpool",
    "country": "England",
    "neighbours": [
     {
      "neighbour_first_name": "Neighbor",
      "neighbour_last_name": "K",
      "neighbour_email": "neighborK@email.com"
     },
     {
      "neighbour_first_name": "Neighbor",
```

```
        "neighbour_last_name": "L",
        "neighbour_email": "neighborL@email.com"
       }
     ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d339f"
  },
  "first_name": "Person",
  "last_name": "7",
  "email": "person7@email.com",
  "dob": {
    "$date": "1996-04-25T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "12 Maple St",
      "zip_code": "G1 2TF",
      "city": "Glasgow",
      "country": "Scotland",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "M",
          "neighbour_email": "neighborM@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "N",
          "neighbour_email": "neighborN@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
```

```
    "$oid": "67ec126ceb24f538608d33a0"
  },
  "first_name": "Person",
  "last_name": "8",
  "email": "person8@email.com",
  "dob": {
    "$date": "1990-01-09T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "89 Oak Dr",
      "zip_code": "LS1 3AB",
      "city": "Leeds",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "O",
          "neighbour_email": "neighborO@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "P",
          "neighbour_email": "neighborP@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a1"
  },
  "first_name": "Person",
  "last_name": "9",
  "email": "person9@email.com",
  "dob": {
    "$date": "1993-08-17T00:00:00.000Z"
  },
  "addresses": [
```

```
    {
      "street": "123 Pine Rd",
      "zip_code": "NE1 2AB",
      "city": "Newcastle",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "Q",
          "neighbour_email": "neighborQ@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "R",
          "neighbour_email": "neighborR@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a2"
  },
  "first_name": "Person",
  "last_name": "10",
  "email": "person10@email.com",
  "dob": {
    "$date": "1997-10-22T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "15 Elm St",
      "zip_code": "CF10 3AF",
      "city": "Cardiff",
      "country": "Wales",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "S",
```

```
        "neighbour_email": "neighborS@email.com"
       },
       {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "T",
        "neighbour_email": "neighborT@email.com"
       }
     ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a3"
  },
  "first_name": "Person",
  "last_name": "11",
  "email": "person11@email.com",
  "dob": {
    "$date": "1992-05-13T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "78 Oak Ln",
      "zip_code": "S1 4GT",
      "city": "Sheffield",
      "country": "England",
      "neighbours": [
       {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "U",
        "neighbour_email": "neighborU@email.com"
       },
       {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "V",
        "neighbour_email": "neighborV@email.com"
       }
     ]
    }
```

```
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a4"
  },
  "first_name": "Person",
  "last_name": "12",
  "email": "person12@email.com",
  "dob": {
    "$date": "1986-02-27T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "56 Birch Rd",
      "zip_code": "NG1 2PB",
      "city": "Nottingham",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "W",
          "neighbour_email": "neighborW@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "X",
          "neighbour_email": "neighborX@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a5"
  },
  "first_name": "Person",
  "last_name": "13",
  "email": "person13@email.com",
```

```
  "dob": {
    "$date": "1991-11-25T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "10 Holy St",
      "zip_code": "CF10 2NF",
      "city": "Cardiff",
      "country": "Wales",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "Y",
          "neighbour_email": "neighborY@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "Z",
          "neighbour_email": "neighborZ@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a6"
  },
  "first_name": "Person",
  "last_name": "14",
  "email": "person14@email.com",
  "dob": {
    "$date": "1987-02-01T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "34 Willow Rd",
      "zip_code": "EH1 1AB",
      "city": "Edinburgh",
      "country": "Scotland",
```

```
    "neighbours": [
      {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "AA",
        "neighbour_email": "neighborAA@email.com"
      },
      {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "AB",
        "neighbour_email": "neighborAB@email.com"
      }
    ]
  }
 ]
},
{
 "_id": {
  "$oid": "67ec126ceb24f538608d33a7"
 },
 "first_name": "Person",
 "last_name": "15",
 "email": "person15@email.com",
 "dob": {
  "$date": "1984-08-12T00:00:00.000Z"
 },
 "addresses": [
  {
   "street": "78 Cedar Ave",
   "zip_code": "CB1 2SE",
   "city": "Cambridge",
   "country": "England",
   "neighbours": [
    {
     "neighbour_first_name": "Neighbor",
     "neighbour_last_name": "AC",
     "neighbour_email": "neighborAC@email.com"
    },
    {
     "neighbour_first_name": "Neighbor",
     "neighbour_last_name": "AD",
```

```
        "neighbour_email": "neighborAD@email.com"
       }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a8"
  },
  "first_name": "Person",
  "last_name": "16",
  "email": "person16@email.com",
  "dob": {
    "$date": "1990-03-09T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "45 Maple Rd",
      "zip_code": "OX2 6TP",
      "city": "Oxford",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "AE",
          "neighbour_email": "neighborAE@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "AF",
          "neighbour_email": "neighborAF@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33a9"
```

```
  },
  "first_name": "Person",
  "last_name": "17",
  "email": "person17@email.com",
  "dob": {
    "$date": "1995-11-17T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "23 Birch Ave",
      "zip_code": "SO14 3HL",
      "city": "Southampton",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "AG",
          "neighbour_email": "neighborAG@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "AH",
          "neighbour_email": "neighborAH@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33aa"
  },
  "first_name": "Person",
  "last_name": "18",
  "email": "person18@email.com",
  "dob": {
    "$date": "1994-06-20T00:00:00.000Z"
  },
  "addresses": [
    {
```

```
      "street": "12 Elm Blvd",
      "zip_code": "LE1 3PL",
      "city": "Leicester",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "Al",
          "neighbour_email": "neighborAl@email.com"
        },
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "AJ",
          "neighbour_email": "neighborAJ@email.com"
        }
      ]
    }
  ]
},
{
  "_id": {
    "$oid": "67ec126ceb24f538608d33ab"
  },
  "first_name": "Person",
  "last_name": "19",
  "email": "person19@email.com",
  "dob": {
    "$date": "1992-12-11T00:00:00.000Z"
  },
  "addresses": [
    {
      "street": "56 Oak Rd",
      "zip_code": "NR1 4BE",
      "city": "Norwich",
      "country": "England",
      "neighbours": [
        {
          "neighbour_first_name": "Neighbor",
          "neighbour_last_name": "AK",
          "neighbour_email": "neighborAK@email.com"
```

```
      },
      {
        "neighbour_first_name": "Neighbor",
        "neighbour_last_name": "AL",
        "neighbour_email": "neighborAL@email.com"
      }
    ]
  }
 ]
},
{
 "_id": {
  "$oid": "67ec126ceb24f538608d33ac"
 },
 "first_name": "Person",
 "last_name": "20",
 "email": "person20@email.com",
 "dob": {
  "$date": "1988-09-25T00:00:00.000Z"
 },
 "addresses": [
  {
   "street": "89 Pine Ave",
   "zip_code": "CF10 3BC",
   "city": "Cardiff",
   "country": "Wales",
   "neighbours": [
    {
     "neighbour_first_name": "Neighbor",
     "neighbour_last_name": "AM",
     "neighbour_email": "neighborAM@email.com"
    },
    {
     "neighbour_first_name": "Neighbor",
     "neighbour_last_name": "AN",
     "neighbour_email": "neighborAN@email.com"
    }
   ]
  }
 ]
```

```
}]
```