Politechnika Krakowska
Wydział Informatyki i Telekomunikacji

Projekt z przedmiotu
Metody i narzędzia analizy
dużych zbiorów danych

Projekt nr 1

Kierunek: Informatyka

Stopień studiów: II stopnia

Specjalizacja: Data Science

Wykonali:

Marek Dalida
Maciej Gicala
Rafał Gęgotek

1) Cel projektu

Celem projektu było stworzenie aplikacji importującej dane z dowolnej strony internetowej do baz typu NoSQL. W tym celu należało wykorzystaj przynajmniej dwie bazy, tj lokalną oraz udostępnioną w ramach darmowej chmury MongoDB. Dane do minimum jednej z tych baz miały być pozyskiwane na zasadzie web scrapping'u.

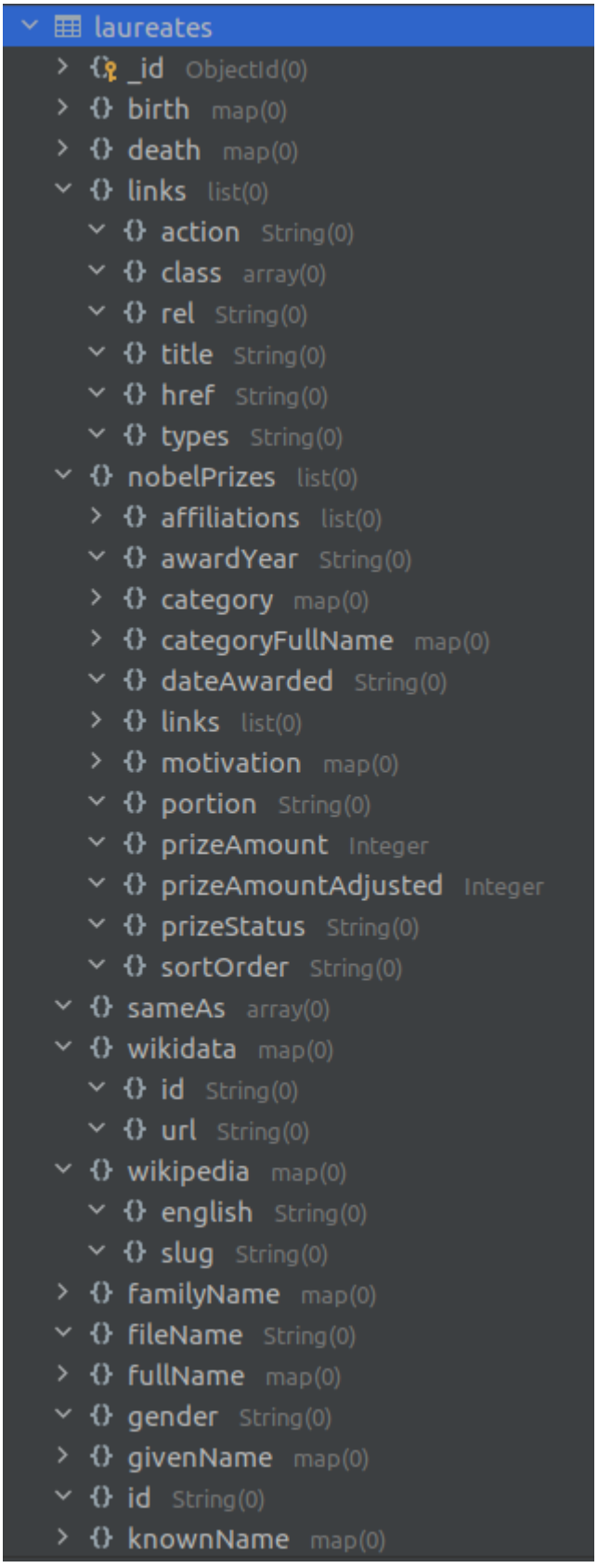
2) Zbiór danych

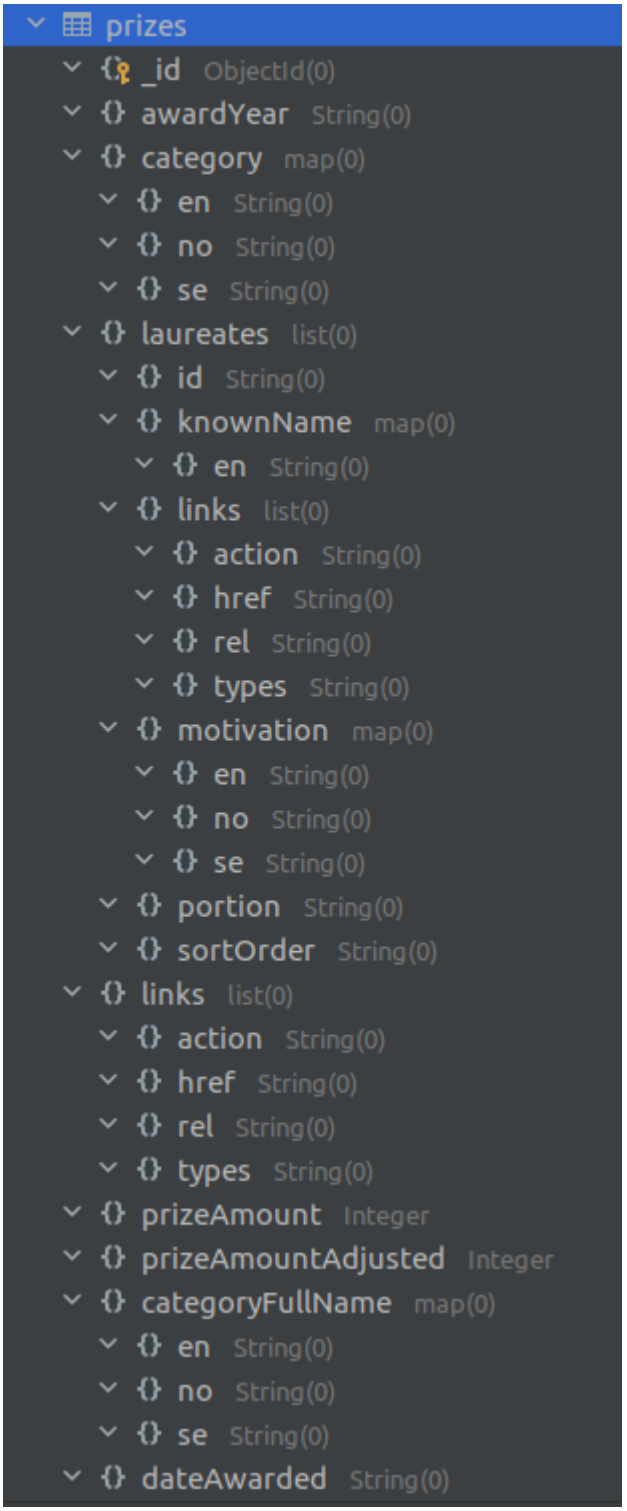
Trzynastu laureatów otrzymało Nagrodę Nobla w 2021 r. za osiągnięcia, które przyniosły ludzkości największe korzyści. Ich praca i odkrycia sięgają od klimatu Ziemi i naszego zmysłu dotyku po wysiłki na rzecz ochrony wolności słowa.
Między innymi te osiągnięcia stanowią zbiór danych użyty w projekcie.

Dane składają się z dwóch kolekcji laureates oraz nobelPrizes, które zostały załadowane do lokalnej bazy danych z ogólno dostępnego interfejsu API oparty na RES: https://www.nobelprize.org/about/developer-zone-2/
Wyniki zapytań do API jest zwracany w formacie JSON, a zawartość jest aktualizowana na bieżąco w oparciu o informację www.nobelprize.org, w momencie ogłoszenia nowych Laureatów.

Łącznie surowe dane liczą 6.5 MB, kolekcja *laureates* ma 968 rekordów, natomiast nobelPrizes 658 rekordów.

Schemat kolekcji prezentuje się następująco:





3) Implementacja

Projekt został zaimplementowany w języku JavaScript.

3.1) Uwierzytelnianie

Obie bazy danych są zabezpieczone hasłem. W programie poświadczenia wymagane do dostępu do baz są wyciągane z pliku konfiguracyjnego. W przypadku ich braku wyświetlany jest komunikat o błędzie.

```
if (process.env.NODE_ENV !== 'production') {
   require('dotenv').config();
}

if (!USERNAME || !PASSWORD) {
   throw new Error('No credentials provided. Please update .env file');
}

const local_uri = `mongodb://${USERNAME}:${PASSWORD}@localhost:27017`;
const atlas_uri =
`mongodb+srv://${USERNAME}:${PASSWORD}@cluster0.mmpd8.mongodb.net/myFirstDatabase?r
etryWrites=true&w=majority`;
```

Podobnie jeśli poświadczenia okażą się błędne, wyświetlany jest odpowiedni komunikat.

```
} catch(e) {
   if (e.codeName === 'AuthenticationFailed') {
     console.error('Authentication failed.');
   }
 }
```

3.2) Aktualizacja kolekcji

Po podłączeniu się do obu baz wywoływane są metody aktualizujące dane. Przyjmują one nazwę kolekcji do zaktualizowania, zapytanie do API zwracające aktualne dane, a także połączenie do bazy, w której ma się wykonać aktualizacja.

```
    await local_client.connect();
    console.log('Connected successfully to local db');
    await atlas_client.connect();
    console.log('Connected successfully to atlas db');

    await updateCollection(laureatesName, axios.get(`${baseApiUrl}${laureatesName}`,
{ params: { limit: 968 }}), local_client);

    await updateCollection(laureatesName, axios.get(`${baseApiUrl}${laureatesName}`,
{ params: { limit: 968 }}), atlas_client);
```

Funkcja aktualizująca zaczyna od wykonania przekazanego zapytania. Następnie usuwa wszystkie aktualnie istniejące rekordy z wybranej kolekcji i wypełnia kolekcję nowo otrzymanymi danymi. O każdym z tych kroków informuje użytkownika komunikatami.

```
async function updateCollection(name, call, client) {
 console.log('Started fetching data...');
 const { data } = await call;
 console.log('Finished fetching data.');

 const database = client.db('nobel');
 const collection = database.collection(name);
 await collection.deleteMany();

 console.log('Started inserting records...');
 const result = await collection.insertMany(data[name]);
 console.log(`Finished. Inserted ${data[name].length} records to collection
${name}.`);
}
```

Po zaktualizowaniu danych pobieranych z API następuje aktualizacja tabeli statystyk. Wykonywana jest seria zapytań do bazy lokalnej a ich wyniki są umieszczane w odpowiedniej kolekcji bazy chmurowej.

```
await updateStatsCollection(local_client, atlas_client);
```

```
async function updateStatsCollection(sourceClient, destinationClient) {
  console.log('Started filling statistics collection...');
  const source_database = sourceClient.db('nobel');
  const laureates = source_database.collection(laureatesName);
  const numberOfMen = await laureates.find({ gender: 'male' }).count();
  const numberOfWomen = await laureates.find({ gender: 'female' }).count();
  const numberOfPoles = await laureates.find({ "birth.place.country.en": "Poland" }).count();
  const avgAgeFirstPrize = await laureates.aggregate([…
  ]).toArray();
  const numberOfMultiWinners = await laureates.aggregate([…
  ]).toArray();

  const destination_database = destinationClient.db('nobel');
  const stats = destination_database.collection(statsName);
  await stats.deleteMany();
  await stats.insertOne({ name: 'Male count', value: numberOfMen });
  await stats.insertOne({ name: 'Female count', value: numberOfWomen });
  await stats.insertOne({ name: 'Pole count', value: numberOfPoles });
  await stats.insertOne({ name: 'Average age when getting first prize', value: avgAgeFirstPrize[0].avgAge });
  await stats.insertOne({ name: 'Multiple awarded count', value: numberOfMultiWinners[0].count });
  console.log('Finished filling statistics collection.');
}
```

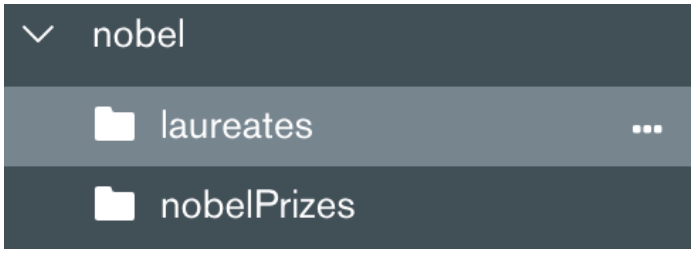
4) Prezentacja działania

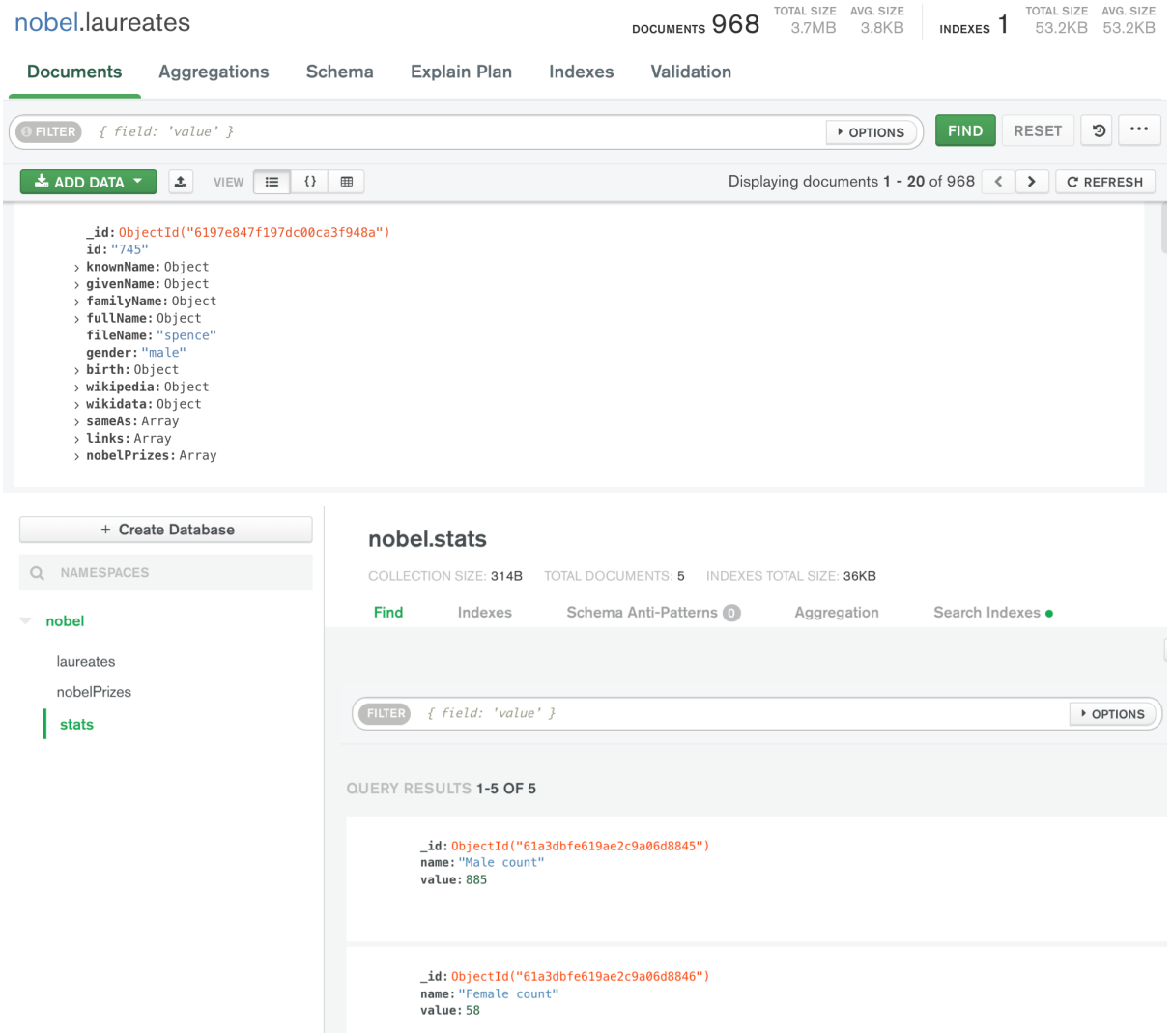
4.1) Poprawne uruchomienie

W przypadku poprawnego uruchomienia wyświetlane są komunikaty o kolejnych krokach działania.

```
gcks@gcks projekt 1 % node index.js
Connected successfully to local db
Connected successfully to atlas db
Started fetching data...
Finished fetching data.
Started inserting records...
Finished. Inserted 968 records to collection laureates.
Started fetching data...
Finished fetching data.
Started inserting records...
Finished. Inserted 658 records to collection nobelPrizes.
Started fetching data...
Finished fetching data.
Started inserting records...
Finished. Inserted 968 records to collection laureates.
Started fetching data...
Finished fetching data.
Started inserting records...
Finished. Inserted 658 records to collection nobelPrizes.
Started filling statistics collection...
Finished filling statistics collection.
Closed connection to local db
Closed connection to atlas db
```

Dane w obu bazach są aktualizowane.





4.2) Brakujące lub niepoprawne poświadczenia

W przypadku nieudanego uwierzytelniania aplikacja również wyświetla stosowne informacje o błędach.

```
gcks@gcks projekt 1 % node index.js
No credentials provided. Please update .env file
Authentication failed.
```

```
gcks@gcks projekt 1 % node index.js
Authentication failed.
```

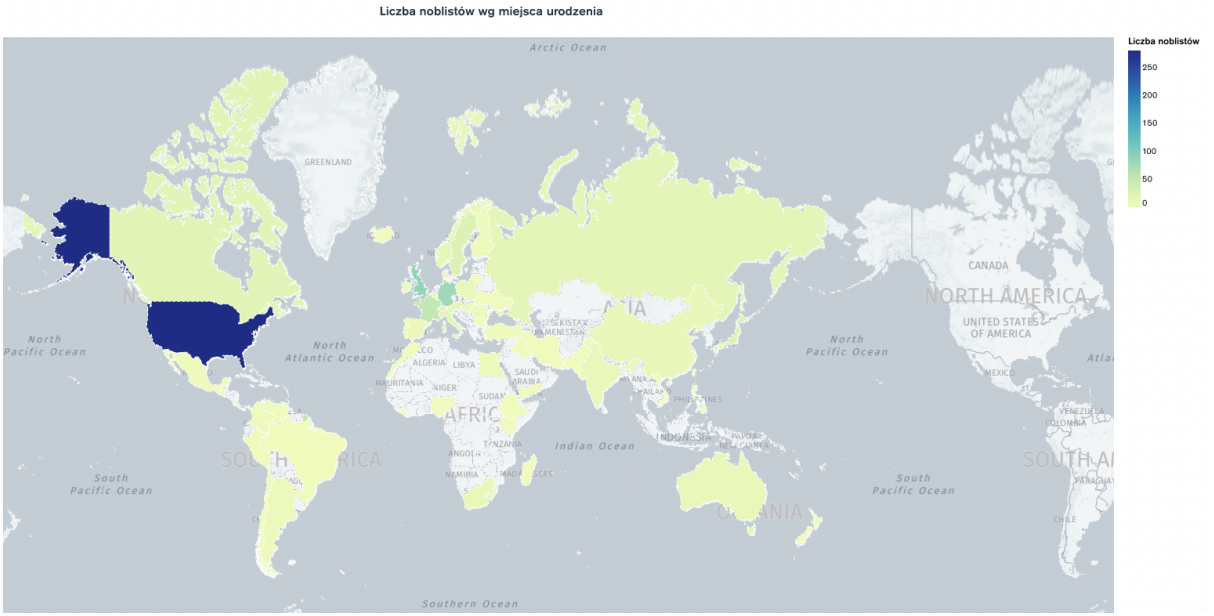
5) Analiza danych

W ramach korzystania z bazy danych MongoDB, możemy posługiwać się dedykowanymi operacjami filtrowania danych dla własnej analizy. Między innymi dzięki użyciu agregacji możemy przetwarzać dokumenty i zwracać przeliczone wyniki.
Poniżej zostały zaprezentowane przykładowe statystyki, przy wykorzystaniu technik jakie udostępnia baza MongoDB.

1) Liczba nagród z podziałem na kraje pochodzenia laureatów.

```
Laureates.aggregate( pipeline: [
    {$unwind: "$nobelPrizes"},
    {
        $group: {
            _id: "$birth.place.country.en",
            sum: {$sum: 1},
        }
    },
    {
        $sort: {sum: -1}
    }
])
```

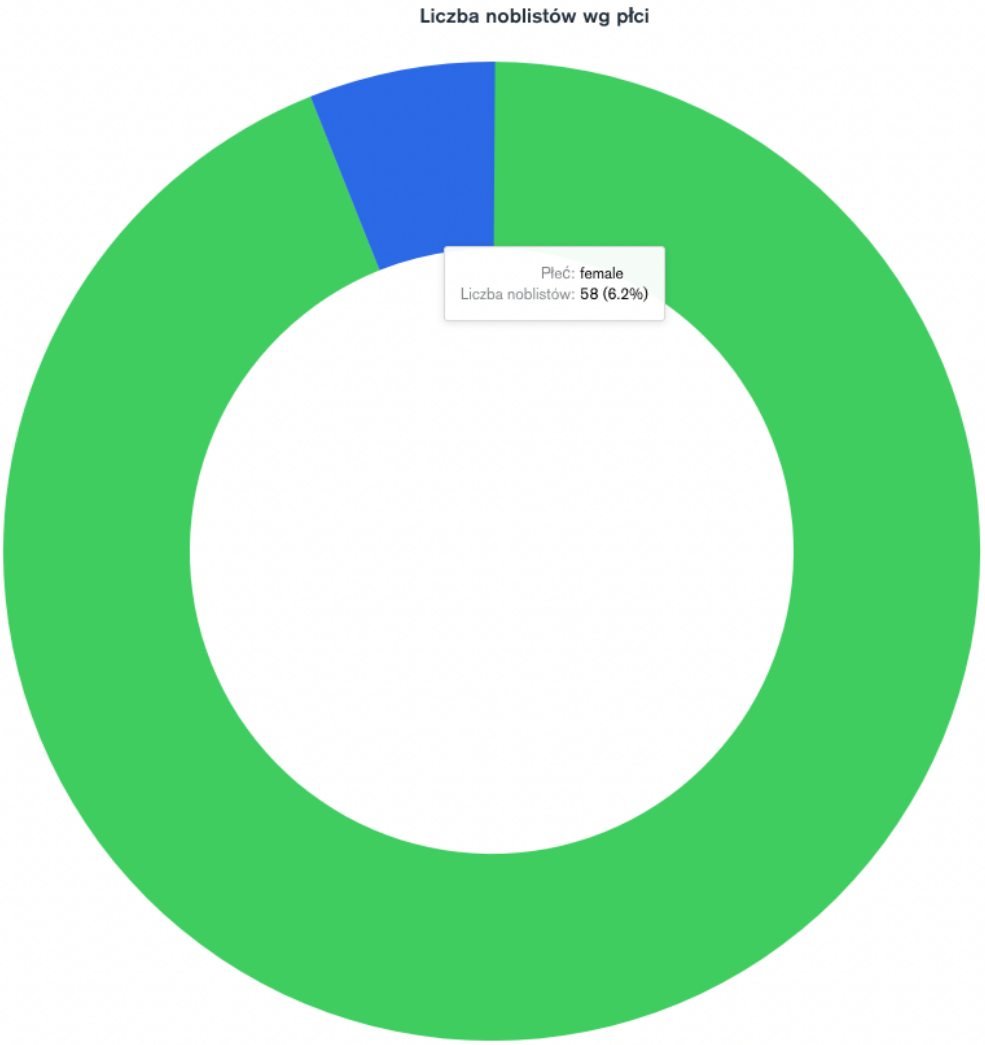

```
[
    {
        "_id": "USA",
        "sum": 283
    },
    {
        "_id": "United Kingdom",
        "sum": 89
    },
    {
        "_id": "Germany",
        "sum": 80
    },
    {
        "_id": "France",
        "sum": 54
    },
    {
        "_id": null,
        "sum": 30
    },
    {
        "_id": "Sweden",
        "sum": 29
    },
    {
        "_id": "Japan",
        "sum": 28
    },
    {
        "_id": "Canada",
        "sum": 21
    },
    {
        "_id": "the Netherlands",
        "sum": 19
```

2) Liczba nagród z podziałem na płeć laureatów.

```
Laureates.aggregate( pipeline: [
    {$unwind: "$nobelPrizes"},
    {
        $group: {
            _id: "$gender",
            sum: {$sum: 1},
        }
    },
    {
        $sort: {sum: -1}
    },
    {
        $limit: 100
    }
])
```

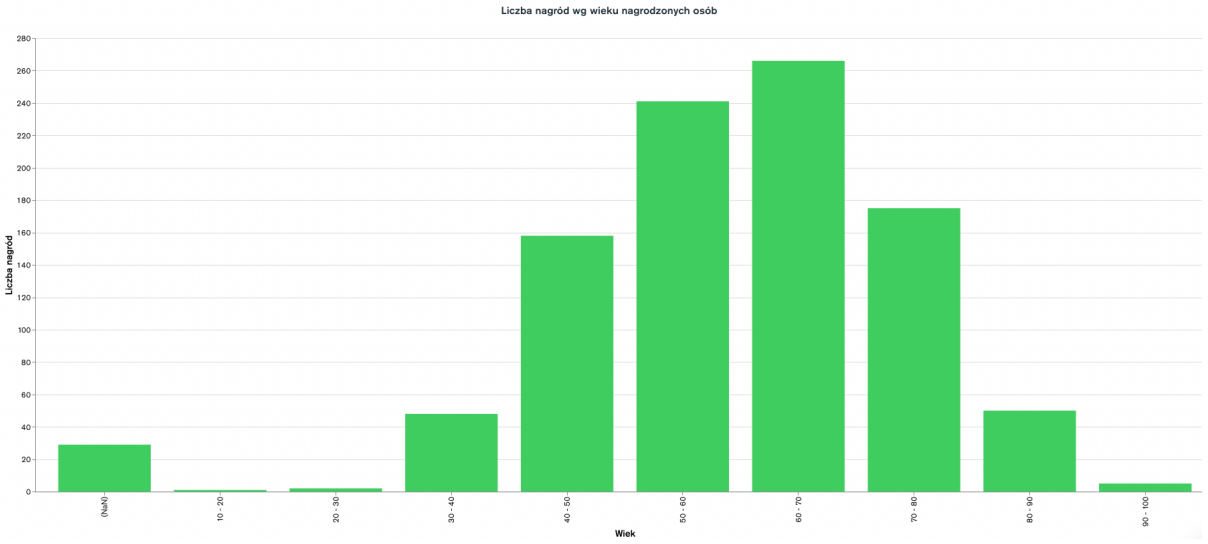
```
[
    {
        "_id": "male",
        "sum": 888
    },
    {
        "_id": "female",
        "sum": 59
    },
    {
        "_id": null,
        "sum": 28
    }
]
```



3) Liczba nagród z podziałem na wiek laureatów.

```
Laureates.aggregate( pipeline: [
    {$unwind: "$nobelPrizes"},
    {
        $project: {
            _id: 1,
            "birthYear": {
                $convert: {
                    input:  {$substr: ["$birth.date", 0, 4 ]},
                    to: "int",
                    onNull: 0
                } },
            "awardYear": {
                $convert: {
                    input:  "$nobelPrizes.awardYear", to: "int",
                    onNull: 0
                } },
            "gender": 1
        }
    },
    {
        $addFields: {
            age: {$subtract: [ "$awardYear", "$birthYear"] },
        }
    },
    {
        $group: {
            _id: '$age',
            count: {$sum: 1},
        }
    },
    {
        $sort: {count: -1}
    }
])
```

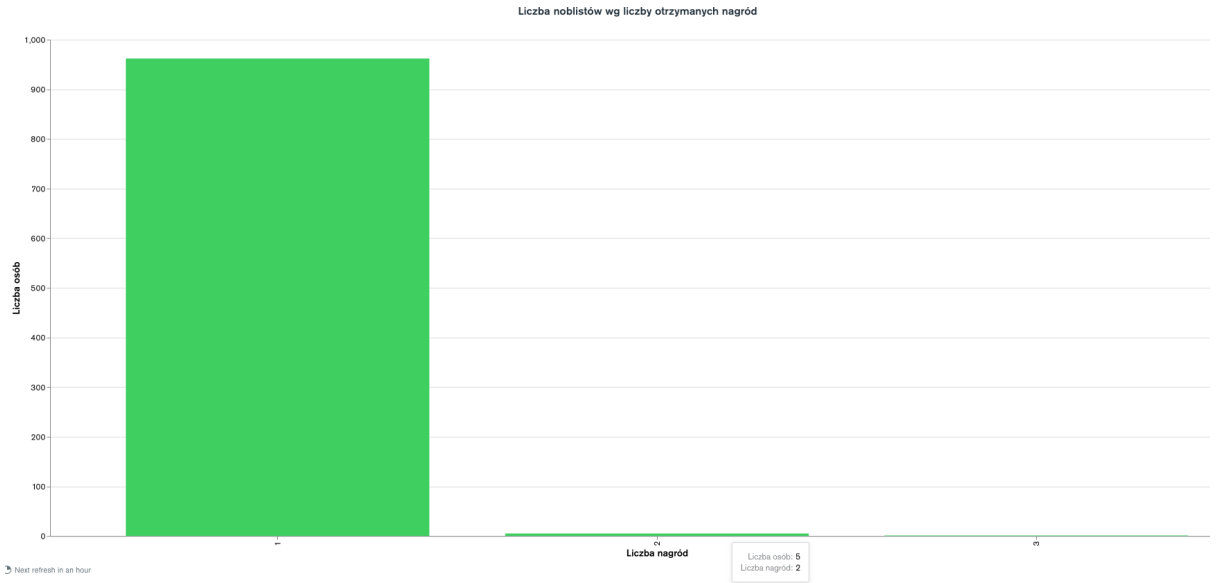

```
[
    {
        "_id": 63,
        "count": 35
    },
    {
        "_id": 61,
        "count": 34
    },
    {
        "_id": 56,
        "count": 34
    },
    {
        "_id": 60,
        "count": 32
    },
    {
        "_id": 64,
        "count": 30
    },
    {
        "_id": 54,
        "count": 30
    },
    {
        "_id": 68,
        "count": 30
    },
    {
        "_id": null,
        "count": 29
    },
    {
        "_id": 62,
        "count": 27
    },
    {
        "_id": 55,
        "count": 25
    },
    {
        "_id": 57,
        "count": 25
```

4) Liczba laureatów z podziałem na liczbę nagród.

```
Laureates.aggregate( pipeline: [
    {
        $project: {
            _id: 1,
            countOfPrize: {$size: "$nobelPrizes"}
        }
    },
    {
        $group: {
            _id: "$countOfPrize",
            count : { $sum: 1}
        }
    }
])
```

```
[
    {
        "_id": 2,
        "count": 5
    },
    {
        "_id": 3,
        "count": 1
    },
    {
        "_id": 1,
        "count": 962
    }
]
```



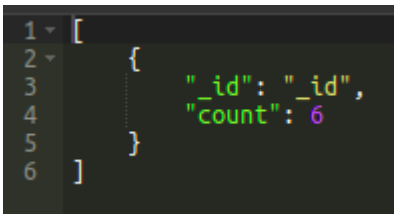
5) Średnia wieku laureatów dla ich pierwszego zdobytego tytułu.

```
Laureates.aggregate( pipeline: [
    {
        $project: {
            _id: 1,
            "birthYear": { $convert: {
                    input:  { $substr: ["$birth.date", 0, 4]},
                    to: "int",
                    onNull: 0
                } },
            "awardYear": { $convert: {
                    input:  {$first: "$nobelPrizes.awardYear"},
                    to: "int",
                    onNull: 0
                } },
        }
    },
    {
        $addFields: {
            age: {$subtract: [ "$awardYear", "$birthYear"] },
        }
    },
    {
        $group: {
            _id: '_id',
            avgAge: {$avg: "$age"},
        }
    },
])
```

```
[
    {
        "_id": "_id",
        "avgAge": 60.03503184713376
    }
]
```

6) Liczba laureatów z wieloma nagrodami.

```
Laureates.aggregate( pipeline: [
    {
        $project: {
            _id: 1,
            countOfPrize: {$size: "$nobelPrizes"}
        }
    },
    {
        $match: {
            countOfPrize : { $gt: 1 }
        }
    },
    {
        $group: {
            _id: "_id",
            count : { $sum: 1}
        }
    }
])
```

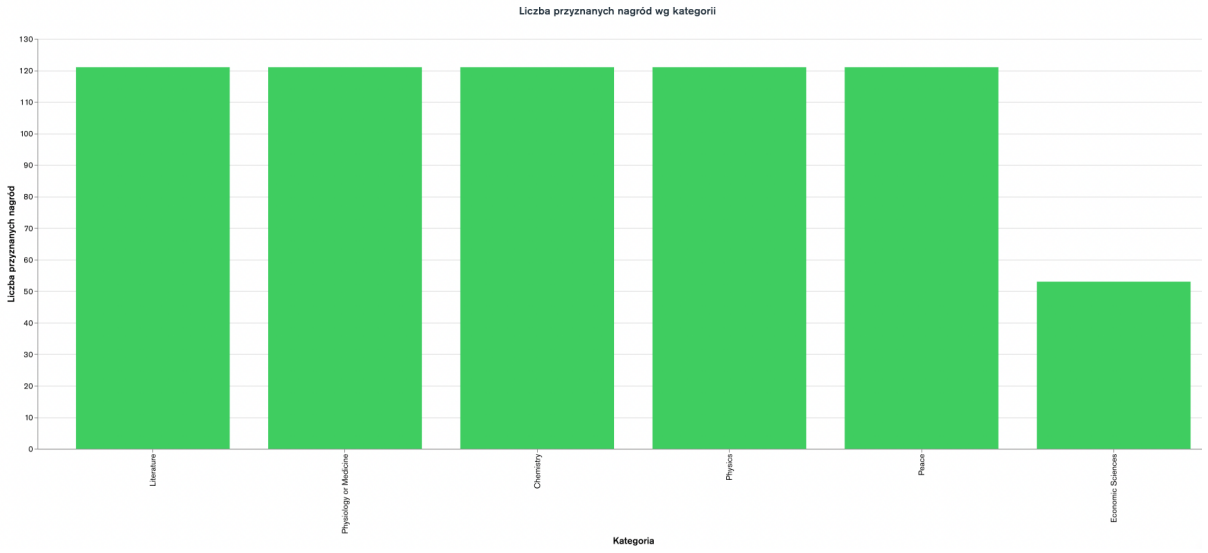


```
[
    {
        "_id": "_id",
        "count": 6
    }
]
```

7) Nagrody Nobla z podziałem na kategorie.

```
NobelPrizes.aggregate( pipeline: [
    {
        $group: {
            _id: "$category.en",
            sum: {$sum: 1},
        }
    },
    {
        $sort: {sum: -1}
    }
])
```


```
[
    {
        "_id": "Peace",
        "sum": 121
    },
    {
        "_id": "Physiology or Medicine",
        "sum": 121
    },
    {
        "_id": "Chemistry",
        "sum": 121
    },
    {
        "_id": "Literature",
        "sum": 121
    },
    {
        "_id": "Physics",
        "sum": 121
    },
    {
        "_id": "Economic Sciences",
        "sum": 53
    }
]
```

8) Średnia przyznawana kwota finansowa z podziałem na kategorie.

```
NobelPrizes.aggregate( pipeline: [
    {
        $group: {
            _id: "$category.en",
            avgOfPrizeAmount:
                {$avg: "$prizeAmount"},
            avgOfPrizeAmountAdjusted:
                {$avg: "$prizeAmountAdjusted"}
        }
    },
    {
        $sort: {avgOfPrizeAmount: -1}
    }
])
```


```
[
    {
        "_id": "Economic Sciences",
        "avgOfPrizeAmount": 5618037.735849056,
        "avgOfPrizeAmountAdjusted": 7490924.886792453
    },
    {
        "_id": "Physics",
        "avgOfPrizeAmount": 2554777.5454545454,
        "avgOfPrizeAmountAdjusted": 5702950.289256198
    },
    {
        "_id": "Literature",
        "avgOfPrizeAmount": 2554777.5454545454,
        "avgOfPrizeAmountAdjusted": 5701601.52892562
    },
    {
        "_id": "Physiology or Medicine",
        "avgOfPrizeAmount": 2554777.5454545454,
        "avgOfPrizeAmountAdjusted": 5702950.289256198
    },
    {
        "_id": "Chemistry",
        "avgOfPrizeAmount": 2554777.5454545454,
        "avgOfPrizeAmountAdjusted": 5702950.289256198
    },
    {
        "_id": "Peace",
        "avgOfPrizeAmount": 2554777.5454545454,
        "avgOfPrizeAmountAdjusted": 5702950.289256198
    }
]
```

9) Liczba przyznanych Nagród Nobla z podziałem na lata.

```
NobelPrizes.aggregate( pipeline: [
    {
        $project: {
            _id: 1,
            "dateAwarded": {
                $convert: {
                    input:  {$substr:
                            ["$dateAwarded", 0, 4]
                    },
                    to: "int",
                    onNull: 0
                } },
        }
    },
    {
        $group: {
            _id: "$dateAwarded",
            sum: {$sum: 1},
        }
    },
    {
        $sort: {_id: -1}
    }
])
```


```
[
    {
        "_id": 2021,
        "sum": 6
    },
    {
        "_id": 2020,
        "sum": 6
    },
    {
        "_id": 2019,
        "sum": 7
    },
    {
        "_id": 2018,
        "sum": 3
    },
    {
        "_id": 2017,
        "sum": 5
    },
    {
        "_id": 2016,
        "sum": 5
    },
    {
        "_id": 2015,
        "sum": 5
    },
    {
        "_id": 2014,
        "sum": 5
    },
    {
        "_id": 2013,
        "sum": 5
    },
    {
        "_id": 2012,
        "sum": 5
    },
```

6) Wnioski

Zakres funkcjonalności i zastosowanych technik został wykonany zgodnie z założeniami wymaganymi na ocenę 4.0

Z analizy danych wynika, że najwięcej laureatów nagrody Nobla pochodzi z krajów rozwiniętych o znacznej liczbie ludności takich jak Stany Zjednoczone, Wielka Brytania czy Niemcy.

Nagroda Nobla znacznie częściej otrzymywali mężczyźni niż kobiety oraz osoby zaawansowane wiekowo. Najwięcej nagród zostało przyznanych w grupie 60-70 lat. Średni wiek laureata nagrody Nobla to około 60 lat.

Niewielu laureatom nagrody udało się ponownie ją zdobyć, udało się to tylko 6 osobom.

Kwoty przyznawane w ramach nagrody Nobla, były największe w przypadku nauk ekonomicznych, w pozostałych kategoriach ich wartość nie różniła się istotnie. Należy jednak zauważyć, że liczba przyznanych nagród w przypadku nauk ekonomicznych była najmniejsza spośród wszystkich kategorii.