Dokumentowe bazy danych – MongoDB

Ćwiczenie 1

- informacja o wersji

```
db.version()
```

- wynik

```
{
  "result": "7.0.7"
}
```

- lista baz danych na serwerze

```
show dbs;
show databases;
db.adminCommand('listDatabases');
```

- wynik

```
[
  {
    "empty": false,
    "name": "admin",
    "sizeOnDisk": 40960
  },
  {
    "empty": false,
    "name": "config",
    "sizeOnDisk": 98304
  },
  {
    "empty": false,
    "name": "local",
    "sizeOnDisk": 40960
  }
]
```

- wybór bazy danych
  - baza o nazwie univ

```
use univ;
```

- informacja o bieżącej bazie danych

```
db;
```

- informacja o kolekcjach

```
show collections;
db.getCollectionNames();
```

Proste operacje/zapytania

- MongoDB simple query
- https://www.mongodb.com/docs/manual/tutorial/query-documents/

Przykład 1

- proste operacje na dokumentach

Ćw 1

- wstaw/stwórz pierwszy dokument
  - w tym momencie tworzona jest baza danych i kolekcja (jeśli wcześniej nie istniały)

```
db.student.insertOne(
    {
        "student_id": 1,
        "firstname": "John",
        "lastname": "Gold",
        "age": 25
    }
);
```

- sprawdź listę baz i kolekcji

```
show databases;
```

- wynik

```
[
  {
    "empty": false,
    "name": "admin",
    "sizeOnDisk": 40960
  },
  {
    "empty": false,
    "name": "config",
    "sizeOnDisk": 98304
  },
  {
    "empty": false,
    "name": "local",
    "sizeOnDisk": 40960
  },
  {
    "empty": false,
    "name": "univ",
    "sizeOnDisk": 8192
  }
]
```

```
show collections;
```

- wynik

```
[
  {
    "badge": "",
    "name": "student"
  }
]
```

- wyszukaj dokumenty w kolekcji `employees`

```
db.student.find();
db.student.find({});
```

- wynik

```
[
  {
    "_id": {"$oid": "67e86e45b1c69e3d7e42c9c0"},
    "age": 25,
    "firstname": "John",
    "lastname": "Gold",
    "student_id": 1
  }
]
```

Ćw 2

- stwórz indeks zapewniający unikalność atrybutu `"student_id"`

```
// tworzenie indeksu
db.student.createIndex({ "student_id" : 1 }, { "unique": true });

// inf. o indeksach w kolekcji student
db.student.getIndexes();

// usunięcie indeksu o nazwie
db.student.getIndexes("student_id_1");
```

- wstaw kilka dokumentów do kolekcji `student`
  - np

```
db.student.insertOne(
    {
        "student_id": 2,
        "firstname": "James",
        "lastname": "Bond"
    }
);


db.student.insertOne(
    {
        "firstname": "John",
        "lastname": "Bond"
    }
);
```

- wyszukaj dokumenty w kolekcji `student`

```
db.student.find();
db.student.find({});
db.student.find({"firstname": "John"});
db.student.find({"student_id": 1});

db.student.find({"_id": ObjectId("67e669647013d10c86e71c87")});
```

- zmodyfikuj wybrane dokumenty
  - po wykonaniu każdego z przykładów sprawdź wynik za pomocą `db.student.find()`
- np

```
db.student.updateOne(
    {"student_id" : 1},
    {
        $set: {
            "firstname": "Adam",
            "lastname": "Silver"
        }
    }
)
```

```
db.student.updateOne(
    {"student_id" : 1},
    {
        $inc: {"age": 5}
    }
)
```

```
db.student.updateOne(
    {"student_id" : 1},
    {
        $set: {
            "firstname": "John",
            "lastname": "Gold"
        },
        $inc: {
            "age": -2
        }
    }
)
```

```
db.student.updateOne(
    {"student_id" : 2},
    {
        $set: {
            "age":  19
        }
    }
)
```

```
db.student.updateOne(
    {"student_id" : 2},
    {
        $unset: {
            "age": ""
        }
    }
)
```

- usuń wybrane dokumenty
    - po wykonaniu każdego z przykładów sprawdź wynik za pomocą `db.student.find()`
- np

```
db.student.deleteOne({"student_id": 1})
db.student.deleteMany({"student_id": 1})
db.student.deleteOne({"_id": ObjectId("67e669647013d10c86e71c87")});
```

- `replace`

```
db.student.find({"student_id": 1});
```

```
db.student.replaceOne(
    {
        "student_id": 1
    },
    {
        "student_id": 1,
        "firstname": "Jan",
        "lastname": "Kowalski",
        "hobby": "Ski"
    }
);
```

- wynik

```
[
  {
    "_id": {"$oid": "67ec1a5b5d92f33cd84a2598"},
    "hobby": "Ski",
    "lastname": "Kowalski",
    "lirstlame": "Jan",
    "student_id": 1
  }
]
```

- tablice
    - `grades` - tablica ocen studenta

```
db.student.updateOne(
    {
        "student_id": 1
    },
    {
        $set : {
            "grades" : [3, 4, 5, 6, 8]
        }
    }
);
```

```
db.student.find({"student_id": 1});
```

- wynik


```
[
  {
    "_id": {"$oid": "67ec1a5b5d92f33cd84a2598"},
    "firstname": "Jan",
    "grades": [3, 4, 5, 6, 8],
    "hobby": "Ski",
    "lastname": "Kowalski",
    "student_id": 1
  }
]
```


- średnia ocena

```
db.student.aggregate([
    {
        $match : {
            "student_id": 1
        }
    },
    {
        $project: {
            "firstname": 1,
            "lastname": 1,
            "student_id": 1,
            "grades": 1,
            "averagegrade": { $avg: "$grades" }
        }
    }
])
```

- wynik


```
[
  {
    "_id": {"$oid": "67ec1a5b5d92f33cd84a2598"},
    "averagegrade": 5.2,
    "firstname": "Jan",
    "grades": [3, 4, 5, 6, 8],
    "lastname": "Kowalski",
    "student_id": 1
  }
]
```


- dodanie elementu do tablicy

```
db.student.updateOne(
    { student_id: 1 },
    { $push: { grades: 10 } }
);
```

```
db.student.find({"student_id": 1});
```

- wynik


```
[
  {
    "_id": {"$oid": "67ec1a5b5d92f33cd84a2598"},
    "firstname": "Jan",
    "grades": [3, 4, 5, 6, 8, 10],
    "hobby": "Ski",
    "lastname": "Kowalski",
    "student_id": 1
  }
]
```


- modyfikacja elementu w tablicy
  - indeksy zaczynają się od 0

```
db.student.updateOne(
    { student_id: 1 },
```

```
    { $set: { "grades.2": 12 } }
)
```

```
db.student.find({"student_id": 1});
```

- wynik

```
[
  {
    "_id": {"$oid": "67ec1a5b5d92f33cd84a2598"},
    "firstname": "Jan",
    "grades": [3, 4, 12, 6, 8, 10],
    "hobby": "Ski",
    "lastname": "Kowalski",
    "student_id": 1
  }
]
```

- `$map` - wykonanie operacji dla każdego elementu tablicy
  - https://www.mongodb.com/docs/manual/reference/operator/aggregation/map/
  - w przykładzie poniżej
    - dzielimy każdą ocenę przez 20

```
db.student.updateOne(
    { student_id: 1 },
    [
        {
            $set: {
                grades: {
                    $map: {
                        input: "$grades",
                        as: "g",
                        in: { $divide: ["$$g", 20] }
                    }
                }
            }
        }
  ]
)
```

```
db.student.find({"student_id": 1});
```

- wynik

```
[
  {
    "_id": {"$oid": "67ec1a5b5d92f33cd84a2598"},
    "firstname": "Jan",
    "grades": [0.15, 0.2, 0.6, 0.3, 0.4, 0.5],
    "hobby": "Ski",
    "lastname": "Kowalski",
    "student_id": 1
  }
]
```

Ćw 3

- Wykonaj kilka własnych eksperymentów z operacjami CRUD
- przydatne linki
  - https://www.mongodb.com/docs/manual/crud/
  - https://www.mongodb.com/docs/manual/reference/method/db.collection.find/
  - https://www.mongodb.com/docs/manual/reference/method/db.collection.insertOne/
  - https://www.mongodb.com/docs/manual/reference/method/db.collection.updateOne/
  - https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteOne/

Ćw 4

- usuń kolekcję `student`

```
db.student.drop();
```

- usuń bazę danych `univ`

```
db.dropDatabase();
```

Przykład 2

Ćw 4

- wybierz bazę `employees`

```
use employees;

db;
```

- wstaw dokument do kolekcji `employees`

```
db.employees.insertOne(
    {
        "EmployeeID": 1,
        "FirstName": "Nancy",
        "LastName": "Davolio",
        "Address" : {
            "Street": "507 - 20th Ave. E. Apt. 2A",
            "City": "Seattle",
            "Country": "USA"
        },
        "Title": "Sales Representative",
        "TitleOfCourtesy": "Ms.",
        "BirthDate": ISODate("1948-12-08T00:00:00.000Z"),
        "HireDate": ISODate("1992-05-01T00:00:00.000Z"),
        "Phone": ["(206) 555-9857", "(206) 555-9858"],
        "Salary": 1000
    }
)
```

- sprawdź zawartość kolekcji

```
db.employees.find();
```

- wstaw kilka kolejnych dokumentów za pomocą `insertMany`

```
db.employees.insertMany(
[
    {
        "EmployeeID": 2,
         "FirstName": "Andrew",
        "LastName": "Fuller",
        "Address" : {
            "Street": "908 W. Capital Way",
            "City": "Tacoma",
            "Country": "USA",
        },
        "Title": "Vice President, Sales",
        "TitleOfCourtesy": "Dr.",
        "BirthDate": ISODate("1952-02-19T00:00:00.000Z"),
        "HireDate": ISODate("1992-08-14T00:00:00.000Z"),
        "Phone": ["(206) 555-9482"],
        "Salary": 10000
    },
    {
        "EmployeeID": 3,
        "FirstName": "Janet",
        "LastName": "Leverling",
        "Address" : {
            "Street": "722 Moss Bay Blvd.",
            "City": "Kirkland",
            "Country": "USA",
         },
        "Title": "Sales Representative",
        "TitleOfCourtesy": "Ms.",
        "BirthDate": ISODate("1963-08-30T00:00:00.000Z"),
```

```
        "HireDate": ISODate("1992-04-01T00:00:00.000Z"),
        "Phone": ["(206) 555-3412"],
        "Salary": 1200
    },
    {
        "EmployeeID": 4,
        "FirstName": "Margaret",
        "LastName": "Peacock",
        "Address" : {
            "Street": "4110 Old Redmond Rd.",
            "City": "Redmond",
            "Country": "USA",
        },
        "Title": "Sales Representative",
        "TitleOfCourtesy": "Mrs.",
        "BirthDate": ISODate("1937-09-19T00:00:00.000Z"),
        "HireDate": ISODate("1993-05-03T00:00:00.000Z"),
        "Phone": ["(206) 555-8122"],
        "Salary": 1100
    },
     {
        "EmployeeID": 5,
        "FirstName": "Steven",
        "LastName": "Buchanan",
        "Address" : {
            "Street": "14 Garrett Hill",
            "City": "London",
            "Country": "UK",
        },
        "Title": "Sales Manager",
        "TitleOfCourtesy": "Mr.",
        "BirthDate": ISODate("1955-03-04T00:00:00.000Z"),
        "HireDate": ISODate("1993-10-17T00:00:00.000Z"),
        "Phone": ["(71) 555-4848"],
        "Salary": 2000
    },
    {
        "EmployeeID": 6,
        "FirstName": "Michael",
        "LastName": "Suyama",
        "Address" : {
            "Street": "Coventry House Miner Rd.",
            "City": "London",
            "Country": "UK",
        },
        "Title": "Sales Representative",
        "TitleOfCourtesy": "Mr.",
        "BirthDate": ISODate("1963-07-02T00:00:00.000Z"),
        "HireDate": ISODate("1993-10-17T00:00:00.000Z"),
        "Phone": ["(71) 555-7773"],
        "Salary": 1500
    },
    {
        "EmployeeID": 7,
        "FirstName": "Robert",
        "LastName": "King",
        "Address" : {
            "Street":  "Edgeham Hollow Winchester Way",
            "City": "London",
            "Country": "UK",
        },
        "Title": "Sales Representative",
        "TitleOfCourtesy": "Mr.",
        "BirthDate": ISODate("1960-05-29T00:00:00.000Z"),
        "HireDate": ISODate("1994-01-02T00:00:00.000Z"),
        "Phone": ["(71) 555-5598"],
        "Salary": 1000
    },
    {
        "EmployeeID": 8,
        "FirstName": "Laura",
        "LastName": "Callahan",
        "Address" : {
            "Street":  "4726 - 11th Ave. N.E.",
            "City": "Seattle",
            "Country": "USA",
        },
        "Title": "Inside Sales Coordinator",
        "TitleOfCourtesy": "Ms.",
        "BirthDate": ISODate("1958-01-09T00:00:00.000Z"),
        "HireDate": ISODate("1994-03-05T00:00:00.000Z"),
        "Phone": ["(206) 555-1189"],
        "Salary": 3000
    },
    {
        "EmployeeID": 9,
        "FirstName": "Anne",
        "LastName": "Dodsworth",
        "Address" : {
            "Street": "7 Houndstooth Rd.",
```

```
            "City": "London",
            "Country": "UK",
        },
        "Title": "Sales Representative",
        "TitleOfCourtesy": "Ms.",
        "BirthDate": ISODate("1966-01-27T00:00:00.000Z"),
        "HireDate": ISODate("1994-11-15T00:00:00.000Z"),
        "Phone": ["(71) 555-4444"],
        "Salary": 1400
    }
]
)
```

Cw 5

- w DataGrip możesz posłużyć się poleceniami SQL
    - podzbiór poleceń SQL - proste polecenia SELECT
- wykonaj kilka takich poleceń
    - przeanalizuj wyniki
- np.

```
select * from employees;
select * from employees where Address.Country = "USA";
select * from employees where Address.Country = "USA" and TitleOfCourtesy = "Ms.";

select * from employees where Salary > 1200;
```

- napisz odpowiadające im polecenia find
- np

```
db.employees.find({})
db.employees.find({"Address.Country": "USA"})
db.employees.find({$and: [{"Address.Country": "USA"}, {"TitleOfCourtesy" : "Ms."}]})

db.employees.find(
    {
        $and: [
            {"Address.Country": "USA"},
            {"TitleOfCourtesy" : "Ms."}
        ]
    }
)

db.employees.find({ Salary: { $gt: 1200 } })
```

- operacja projekcji (wybór atrybutów w zbiorze wynikowym)

```
db.employees.find({})
```

```
db.employees.find(
    {},
    {"FirstName": 1, "LastName":1}
)
```

- wynik

```
[
  {
    "_id": {"$oid": "67e879dbb1c69e3d7e42c9c3"},
    "FirstName": "Nancy",
    "LastName": "Davolio"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c5"},
    "FirstName": "Andrew",
    "LastName": "Fuller"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c6"},
    "FirstName": "Janet",
    "LastName": "Leverling"
  },
```

```
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c7"},
    "FirstName": "Margaret",
    "LastName": "Peacock"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c8"},
    "FirstName": "Steven",
    "LastName": "Buchanan"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c9"},
    "FirstName": "Michael",
    "LastName": "Suyama"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9ca"},
    "FirstName": "Robert",
    "LastName": "King"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9cb"},
    "FirstName": "Laura",
    "LastName": "Callahan"
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9cc"},
    "FirstName": "Anne",
    "LastName": "Dodsworth"
  }
]
```


- warunek i projekcja

```
db.employees.find(
    {
        "Address.Country": "USA"
    },
    {
        "FirstName": 1,
        "LastName":1
    }
)
```

- wynik


```
[
  {
    "FirstName": "Nancy",
    "LastName": "Davolio"
  },
  {
    "FirstName": "Andrew",
    "LastName": "Fuller"
  },
  {
    "FirstName": "Janet",
    "LastName": "Leverling"
  },
  {
    "FirstName": "Margaret",
    "LastName": "Peacock"
  },
  {
    "FirstName": "Laura",
    "LastName": "Callahan"
  }
]
```


- inne przykłady
  - 1 - atrybut pojawi się w zbiorze wynikowym
  - 0 - atrybut nie pojawi się
  - mogą wystąpić albo 1 albo 0
    - nie dotyczy atrybutu `_id_`

```
db.employees.find({},{"_id": 0, "FirstName": 1, "LastName": 1 })

db.employees.find({},{"_id": 1, "FirstName": 0, "LastName": 0})

// to jest błąd
db.employees.find({},{"_id": 1, "FirstName": 1, "LastName": 0})
```

- count

```
select count(*) from employees;
select count(*) from employees where Address.Country = "USA";
select count(*) from employees where Address.Country = "USA" and TitleOfCourtesy = "Ms.";

select count(*) from employees where Salary > 1200;
```

```
db.employees.find({}).count();
db.employees.find({"Address.Country": "USA"}).count();

db.employees.find({$and: [{"Address.Country": "USA"}, {"TitleOfCourtesy" : "Ms."}]}).count();


db.employees.find(
    {
        $and: [
            {"Address.Country": "USA"},
            {"TitleOfCourtesy" : "Ms."}
        ]
     }
).count();
```

```
db.employees.find({ Salary: { $gt: 1200 } }).count();
```

- wynik

```
[
  {
    "result": 5
  }
]
```

- $or $and

```
select * from employees where Salary > 1100  and Salary < 1500
```

```
db.employees.find({
  Salary: { $gt: 1100, $lt: 1500 }
})

db.employees.find(
    {
        $and: [
            { Salary: { $gt: 1100 } },
            { Salary: { $lt: 1500 } }
        ]
    }
);
```

- wynik

```
[
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c6"},
    "Address": {
      "Street": "722 Moss Bay Blvd.",
      "City": "Kirkland",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1963-08-30T00:00:00.000Z"},
    "EmployeeID": 3,
    "FirstName": "Janet",
    "HireDate": {"$date": "1992-04-01T00:00:00.000Z"},
    "LastName": "Leverling",
    "Phone": ["(206) 555-3412"],
    "Salary": 1200,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Ms."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9cc"},
    "Address": {
      "Street": "7 Houndstooth Rd.",
      "City": "London",
```

```
      "Country": "UK"
    },
    "BirthDate": {"$date": "1966-01-27T00:00:00.000Z"},
    "EmployeeID": 9,
    "FirstName": "Anne",
    "HireDate": {"$date": "1994-11-15T00:00:00.000Z"},
    "LastName": "Dodsworth",
    "Phone": ["(71) 555-4444"],
    "Salary": 1400,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Ms."
  }
]
```


```
select * from employees where Salary <= 1100  or Salary >= 1500
```

```
db.employees.find(
    {
        $or: [
            { Salary: { $lte: 1100 } },
            { Salary: { $gte: 1500 } }
        ]
    }
);
```

- wynik


```
[
  {
    "_id": {"$oid": "67e879dbb1c69e3d7e42c9c3"},
    "Address": {
      "Street": "507 - 20th Ave. E.\r\nApt. 2A",
      "City": "Seattle",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1948-12-08T00:00:00.000Z"},
    "EmployeeID": 1,
    "FirstName": "Nancy",
    "HireDate": {"$date": "1992-05-01T00:00:00.000Z"},
    "LastName": "Davolio",
    "Phone": ["(206) 555-9857", "(206) 555-9858"],
    "Salary": 1000,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Ms."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c5"},
    "Address": {
      "Street": "908 W. Capital Way",
      "City": "Tacoma",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1952-02-19T00:00:00.000Z"},
    "EmployeeID": 2,
    "FirstName": "Andrew",
    "HireDate": {"$date": "1992-08-14T00:00:00.000Z"},
    "LastName": "Fuller",
    "Phone": ["(206) 555-9482"],
    "Salary": 10000,
    "Title": "Vice President, Sales",
    "TitleOfCourtesy": "Dr."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c7"},
    "Address": {
      "Street": "4110 Old Redmond Rd.",
      "City": "Redmond",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1937-09-19T00:00:00.000Z"},
    "EmployeeID": 4,
    "FirstName": "Margaret",
    "HireDate": {"$date": "1993-05-03T00:00:00.000Z"},
    "LastName": "Peacock",
    "Phone": ["(206) 555-8122"],
    "Salary": 1100,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Mrs."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c8"},
    "Address": {
      "Street": "14 Garrett Hill",
```

```
      "City": "London",
      "Country": "UK"
    },
    "BirthDate": {"$date": "1955-03-04T00:00:00.000Z"},
    "EmployeeID": 5,
    "FirstName": "Steven",
    "HireDate": {"$date": "1993-10-17T00:00:00.000Z"},
    "LastName": "Buchanan",
    "Phone": ["(71) 555-4848"],
    "Salary": 2000,
    "Title": "Sales Manager",
    "TitleOfCourtesy": "Mr."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c9"},
    "Address": {
      "Street": "Coventry House Miner Rd.",
      "City": "London",
      "Country": "UK"
    },
    "BirthDate": {"$date": "1963-07-02T00:00:00.000Z"},
    "EmployeeID": 6,
    "FirstName": "Michael",
    "HireDate": {"$date": "1993-10-17T00:00:00.000Z"},
    "LastName": "Suyama",
    "Phone": ["(71) 555-7773"],
    "Salary": 1500,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Mr."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9ca"},
    "Address": {
      "Street": "Edgeham Hollow Winchester Way",
      "City": "London",
      "Country": "UK"
    },
    "BirthDate": {"$date": "1960-05-29T00:00:00.000Z"},
    "EmployeeID": 7,
    "FirstName": "Robert",
    "HireDate": {"$date": "1994-01-02T00:00:00.000Z"},
    "LastName": "King",
    "Phone": ["(71) 555-5598"],
    "Salary": 1000,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Mr."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9cb"},
    "Address": {
      "Street": "4726 - 11th Ave. N.E.",
      "City": "Seattle",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1958-01-09T00:00:00.000Z"},
    "EmployeeID": 8,
    "FirstName": "Laura",
    "HireDate": {"$date": "1994-03-05T00:00:00.000Z"},
    "LastName": "Callahan",
    "Phone": ["(206) 555-1189"],
    "Salary": 3000,
    "Title": "Inside Sales Coordinator",
    "TitleOfCourtesy": "Ms."
  }
]
```


```
select * from employees where year(HireDate) = 1992
```

```
db.employees.find(
    {
        $expr: {
            $eq: [ { $year: "$HireDate" }, 1992 ]
        }
    }
);


db.employees.find(
    {
        HireDate: {
            $gte: ISODate("1992-01-01T00:00:00Z"),
            $lt: ISODate("1993-01-01T00:00:00Z")
        }
    }
);
```

- wynik


```
[
  {
    "_id": {"$oid": "67e879dbb1c69e3d7e42c9c3"},
    "Address": {
      "Street": "507 - 20th Ave. E.\r\nApt. 2A",
      "City": "Seattle",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1948-12-08T00:00:00.000Z"},
    "EmployeeID": 1,
    "FirstName": "Nancy",
    "HireDate": {"$date": "1992-05-01T00:00:00.000Z"},
    "LastName": "Davolio",
    "Phone": ["(206) 555-9857", "(206) 555-9858"],
    "Salary": 1000,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Ms."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c5"},
    "Address": {
      "Street": "908 W. Capital Way",
      "City": "Tacoma",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1952-02-19T00:00:00.000Z"},
    "EmployeeID": 2,
    "FirstName": "Andrew",
    "HireDate": {"$date": "1992-08-14T00:00:00.000Z"},
    "LastName": "Fuller",
    "Phone": ["(206) 555-9482"],
    "Salary": 10000,
    "Title": "Vice President, Sales",
    "TitleOfCourtesy": "Dr."
  },
  {
    "_id": {"$oid": "67e879e2b1c69e3d7e42c9c6"},
    "Address": {
      "Street": "722 Moss Bay Blvd.",
      "City": "Kirkland",
      "Country": "USA"
    },
    "BirthDate": {"$date": "1963-08-30T00:00:00.000Z"},
    "EmployeeID": 3,
    "FirstName": "Janet",
    "HireDate": {"$date": "1992-04-01T00:00:00.000Z"},
    "LastName": "Leverling",
    "Phone": ["(206) 555-3412"],
    "Salary": 1200,
    "Title": "Sales Representative",
    "TitleOfCourtesy": "Ms."
  }
]
```


Ćw 6

- Wykonaj kilka własnych eksperymentów

Operacje agregacji

- aggregate
- aggregation pipline
- https://www.mongodb.com/docs/manual/core/aggregation-pipeline/

```
db.<collection>.aggregate(

[

   {stage 1},

   {stage 2},

    …..

   {stage N}

]
```

```
)
```

- https://www.mongodb.com/docs/manual/reference/operator/aggregation-pipeline/#std-label-aggregation-pipeline-operator-reference
- Stages
    - `$match`
    - `$project`
    - `$group`
    - `$unwind`
    - `$lookup`
    - `$out`
    - ...
- Aggregation vs SQL
    - https://www.mongodb.com/docs/manual/reference/sql-aggregation-comparison/

Przykład 3

Ćw 6

- baza `employees` z poprzedniego przykładu

```
use employees;

db;

db.employees.find();
```

- Operacje agregacji
    - `aggregate`

```
db.employees.aggregate([
    {
        $match: {
            "Address.Country": "USA"
        }
    },
    {
        $project: {
            "_id": 0,
            "LastName": 1,
            "FirstName": 1}
    },
    {
        $sort: {LastName: -1}
    }
])
```

- wynik

```
[
  {
    "FirstName": "Nancy",
    "LastName": "Davolio"
  },
  {
    "FirstName": "Andrew",
    "LastName": "Fuller"
  },
  {
    "FirstName": "Janet",
    "LastName": "Leverling"
  },
  {
    "FirstName": "Margaret",
    "LastName": "Peacock"
  },
  {
    "FirstName": "Laura",
    "LastName": "Callahan"
  }
]
```

- dodatkowe pole `FullName`

```
db.employees.aggregate([
    {
        $match: {
            "Address.Country": "USA"
        }
    },
    {
        $project: {
            "_id": 0,
            "First": "$LastName",
            "Last": "$LastName",
            "FullName": {
                $concat: ["$FirstName", " ", "$LastName"]
            }

       }
    },
    {
        $sort: {lastname: -1}
    }
])
```

- wynik

```
[
  {
    "First": "Davolio",
    "FullName": "Nancy Davolio",
    "Last": "Davolio"
  },
  {
    "First": "Fuller",
    "FullName": "Andrew Fuller",
    "Last": "Fuller"
  },
  {
    "First": "Leverling",
    "FullName": "Janet Leverling",
    "Last": "Leverling"
  },
  {
    "First": "Peacock",
    "FullName": "Margaret Peacock",
    "Last": "Peacock"
  },
  {
    "First": "Callahan",
    "FullName": "Laura Callahan",
    "Last": "Callahan"
  }
]
```

- grupowanie
    - $group

```
select TitleOfCourtesy, count(*)
from employees
where  Address.Country = "USA"
group by TitleOfCourtesy
```

```
db.employees.aggregate([
    {
        $match: {
            "Address.Country": "USA"
        }
    },
    {
        $group: {
            "_id": "$TitleOfCourtesy",
            "cnt": {"$sum": 1}
        }
    }
])
```

- wynik

```
[
  {
    "_id": "Dr.",
    "cnt": 1
```

```
  },
  {
    "_id": "Mrs.",
    "cnt": 1
  },
  {
    "_id": "Ms.",
    "cnt": 3
  }
]
```

- dodatkowa projekcja, zmiana nazwy

```
db.employees.aggregate([
    {
        $match: {
            "Address.Country": "USA"
        }
    },
    {
        $group: {
            "_id": "$TitleOfCourtesy",
            "cnt": {"$sum": 1}
        }
    },
    {
        $project: {
            "_id": 0,
            "TitleOfCourtesy": "$_id",
            "cnt": 1}
    },

])
```

- grupowanie wg kilku atrybutów

```
select Address.Country, TitleOfCourtesy, count(*) cnt
from employees
group by Address.Country, TitleOfCourtesy
order by cnt desc;
```

```
db.employees.aggregate([
    {
        $group: {
            _id: {
                country: "$Address.Country",
                title: "$TitleOfCourtesy"
            },
            cnt: { $sum: 1 }
        }
    },
    {
        $project: {
        _id: 0,
        country: "$_id.country",
        title: "$_id.title",
        cnt: 1
        }
    },
    {
        $sort: {cnt: -1}
    }
])
```

- wynik

```
[
  {
    "Country": "UK",
    "TitleOfCourtesy": "Mr.",
    "cnt": 3
  },
  {
    "Country": "USA",
    "TitleOfCourtesy": "Ms.",
    "cnt": 3
  },
  {
    "Country": "USA",
    "TitleOfCourtesy": "Dr.",
    "cnt": 1
```

```
  },
  {
    "Country": "UK",
    "TitleOfCourtesy": "Ms.",
    "cnt": 1
  },
  {
    "Country": "USA",
    "TitleOfCourtesy": "Mrs.",
    "cnt": 1
  }
]
```


- zapis wyniku
  - `$out`
  - zapis zbioru wynikowego to kolekcji o wskazanej nazwie

```
db.employees.aggregate([
    {
        $group: {
            _id: {
                Country: "$Address.Country",
                TitleOfCourtesy: "$TitleOfCourtesy"
            },
            cnt: { $sum: 1 }
        }
    },
    {
        $project: {
        _id: 0,
        Country: "$_id.Country",
        TitleOfCourtesy: "$_id.TitleOfCourtesy",
        cnt: 1
        }
    },
    {
        $out: "em_by_country_title"
    }
])
```

- lista kolekcji

```
show collections;
```

- wynik


```
[
  {
    "badge": "",
    "name": "em_by_country_title"
  },
  {
    "badge": "",
    "name": "employees"
  }
]
```


- zawartość kolekcji `em_by_title`

```
db.em_by_country_title.find();
```

- wynik


```
[
  {
    "_id": {"$oid": "67ec20f77b1ef941c9512dcb"},
    "Country": "USA",
    "TitleOfCourtesy": "Mrs.",
    "cnt": 1
  },
  {
    "_id": {"$oid": "67ec20f77b1ef941c9512dcc"},
    "Country": "USA",
    "TitleOfCourtesy": "Ms.",
    "cnt": 3
  },
  {
    "_id": {"$oid": "67ec20f77b1ef941c9512dcd"},
```

```
    "Country": "USA",
    "TitleOfCourtesy": "Dr.",
    "cnt": 1
  },
  {
    "_id": {"$oid": "67ec20f77b1ef941c9512dce"},
    "Country": "UK",
    "TitleOfCourtesy": "Mr.",
    "cnt": 3
  },
  {
    "_id": {"$oid": "67ec20f77b1ef941c9512dcf"},
    "Country": "UK",
    "TitleOfCourtesy": "Ms.",
    "cnt": 1
  }
]
```


Ćw 7

- Wykonaj kilka własnych eksperymentów

Ćw 8

- usuń kolekcje `employees` i `em_usa_by_titl`

```
db.employees.drop();
db.em_usa_by_title.drop();
```

- usuń bazę danych `employees`

```
db.dropDatabase();
```