

## Projekt 2

Damian Guzek s167911

1.

```
CREATE VIEW widok1 AS
SELECT nazwa, 100-stan AS 'zamowienie' FROM karma
WHERE stan<100;
```

	nazwa	zamowienie
►	hedera	16
	lobelia	42
	vinica	78

```
db.zoo.update({id:"karma"},{$inc: {"hedera":-100}})
db.zoo.update({id:"karma"},{$inc: {"allium":-100}})
db.zoo.update({id:"karma"},{$inc: {"canna":-100}})
db.zoo.update({id:"karma"},{$inc: {"dolichos":-100}})
db.zoo.update({id:"karma"},{$inc: {"lobelia":-100}})
db.zoo.update({id:"karma"},{$inc: {"pinus":-100}})
db.zoo.update({id:"karma"},{$inc: {"vinica":-100}})
db.zoo.update({id:"karma"},{$inc: {"phormium":-100}})
db.zoo.update({id:"karma"},{$inc: {"lupinus":-100}})
db.zoo.update({id:"karma"},{$inc: {"wolowina":-100}})
```

```
db.zoo.find({$and:[{"hedera":{$lt:0}},{"id":"karma"}]},{"hedera:1, _id:0})
db.zoo.find({$and:[{"allium":{$lt:0}},{"id":"karma"}]},{"allium:1, _id:0})
db.zoo.find({$and:[{"canna":{$lt:0}},{"id":"karma"}]},{"canna:1, _id:0})
db.zoo.find({$and:[{"dolichos":{$lt:0}},{"id":"karma"}]},{"dolichos:1, _id:0})
db.zoo.find({$and:[{"lobelia":{$lt:0}},{"id":"karma"}]},{"lobelia:1, _id:0})
db.zoo.find({$and:[{"pinus":{$lt:0}},{"id":"karma"}]},{"pinus:1, _id:0})
db.zoo.find({$and:[{"vinica":{$lt:0}},{"id":"karma"}]},{"vinica:1, _id:0})
db.zoo.find({$and:[{"phormium":{$lt:0}},{"id":"karma"}]},{"phormium:1, _id:0})
db.zoo.find({$and:[{"lupinus":{$lt:0}},{"id":"karma"}]},{"lupinus:1, _id:0})
db.zoo.find({$and:[{"wolowina":{$lt:0}},{"id":"karma"}]},{"wolowina:1, _id:0})
```

db.zoo.fin.. x db.zoo.fin.. x db.zoo.fin.. x		
zoo	0.095 sec.	0 50
Key	Value	Type
▼ (1)	{ 1 field }	Object
## hedera	-16.0	Double

```
/* 1 */
{
  "hedera" : -16.0
}
```

2.

```
CREATE VIEW widok2 AS
SELECT id_zwierze, data_szczenie FROM szczepienia
WHERE data_szczenie
BETWEEN '2019-09-01' AND '2019-12-31';
```

	id_zwierze	data_szczenie	
▶	20018	2019-09-13	
	20013	2019-09-24	
	20007	2019-10-30	
	20001	2019-12-16	

```
db.zoo.find({
  $or:[{
    szczepienie1: {
      $gte: ISODate("2019-09-01T00:00:00.000Z"),
      $lt: ISODate("2019-12-31T00:00:00.000Z")},
    {
      szczepienie2: {
        $gte: ISODate("2019-09-01T00:00:00.000Z"),
        $lt: ISODate("2019-12-31T00:00:00.000Z")}}
  ]
})
```

```
/* 1 */
{
  "_id" : ObjectId("5e9f74b255170ab887f9d268"),
  "id" : 20001.0,
  "gatunek" : "alpaka",
  "rok_urodzenia" : 1979.0,
  "weterynarz" : 10001.0,
  "opiekun" : 10004.0,
  "szczepienie1" : ISODate("2019-12-16T00:00:00.000Z"),
  "klatka" : 30001.0,
  "darownizna" : 1109.0,
  "kontrola" : ISODate("2019-05-31T00:00:00.000Z")
}
```

zoo 0.606 sec.	
Key	Value
> (1) ObjectId("5e9f74b255170ab887f9d...	{ 10 fields }
> (2) ObjectId("5e9f74b255170ab887f9d...	{ 10 fields }
> (3) ObjectId("5e9f74b255170ab887f9d...	{ 11 fields }
> (4) ObjectId("5e9f74b255170ab887f9d...	{ 11 fields }

3.

```
CREATE VIEW widok3 AS
```


```
SELECT pracownicy.id_pracownik, imie, nazwisko, telefon
```

```
FROM pracownicy JOIN rodo ON pracownicy.id_pracownik=rodo.id_pracownik
```

```
WHERE pracownicy.id_pracownik=10002;
```

	id_pracownik	imie	nazwisko	telefon	
▶	10002	Karol	Piasecki	116997913	

```
db.zoo.find({"id":10002},{imie:1,nazwisko:1,telefon:1, _id:0, id:1})
```

Key	Value	Type
>  (1)	{ 4 fields }	Object

```
/* 1 */
{
  "id" : 10002.0,
  "imie" : "Karol",
  "nazwisko" : "Piasecki",
  "telefon" : 116997913.0
}
```

4.

```
CREATE VIEW widok4 AS  
SELECT klatki.id_klatka, nazwa, stan  
FROM klatki JOIN karma_klatki ON klatki.id_klatka=karma_klatki.id_klatka  
JOIN karma ON karma_klatki.id_karma=karma.id_karma  
WHERE klatki.id_klatka=30001;
```

	id_klatka	nazwa	stan	
▶	30001	lobelia	58	
	30001	vinica	22	

```
db.zoo.find({"id":"karma"},{lobelia:1,vinica:1,_id:0})
```

key	value	type
> (1)	{ 2 fields }	Object

```
/* 1 */  
{  
  "lobelia" : 58.0,  
  "vinica" : 22.0  
}
```

5.

```
CREATE VIEW widok5 AS
```

```
SELECT imie, nazwisko, telefon FROM rodo WHERE id_pracownik IN  
      (SELECT id_pracownik FROM klatki WHERE id_klatka IN  
        (SELECT id_klatka FROM zwierzeta WHERE  
          rok_urodzenia>2013));
```

	imie	nazwisko	telefon	
►	Karolina	Kleina	229333986	
	Martyna	Waras	466637069	
	Aleksandra	Warmbier	316374202	

```
db.zoo.find({rok_urodzenia:{$gt: 2013}},{_id:0,opiekun:1}).sort({"opiekun":-1})
```

```
db.zoo.find({"id":{$in:[10004, 10005, 10008]}},{_id:0,imie:1,nazwisko:1,telefon:1})
```

Key	Value	Type
> { } (1)	{ 3 fields }	Object
> { } (2)	{ 3 fields }	Object
> { } (3)	{ 3 fields }	Object

```
/* 1 */  
{  
  "imie" : "Karolina",  
  "nazwisko" : "Kleina ",  
  "telefon" : 229333986.0  
}
```

6.

```
CREATE VIEW widok6 AS
```

```
SELECT
```

```
    sum(darowizna),
```

```
    avg(darowizna),
```

```
    max(darowizna),
```

```
    min(darowizna),
```

```
    (SELECT Count(*) FROM klatki WHERE darowizna<3000) as '<3000',
```

```
    (SELECT count(*) FROM klatki WHERE darowizna>3000) as '>3000'
```

```
FROM klatki;
```

	sum(darowizna)	avg(darowizna)	max(darowizna)	min(darowizna)	<3000	>3000	
▶	69846	4656.4	23347	5	10	5	

```
db.zoo.find({darownizna:{$gt: 5}},{_id:0,darownizna:1}).sort({darownizna:-1}).limit(1)
```

```
db.zoo.find({darownizna:{$gt: 5}},{_id:0,darownizna:1}).sort({darownizna:1}).limit(1)
```

```
db.zoo.find({darownizna:{$gt: 3000}}).count()
```

```
db.zoo.find({darownizna:{$lt: 3000}}).count()
```

```
/* 1 */
{
  "darownizna" : 23347.0
}
```

```
/* 1 */
{
  "darownizna" : 543.0
}
```

```
8
```

0.179 sec.

```
12
```

#warto zwrócić uwagę, że dane są lekko inne ponieważ

1. wyeliminowaliśmy pustą klatkę

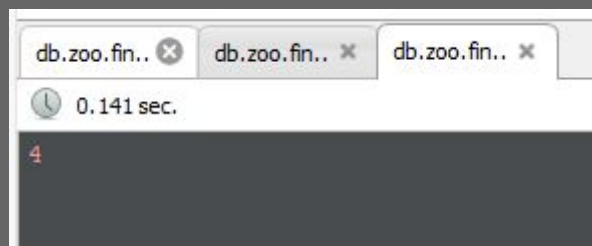
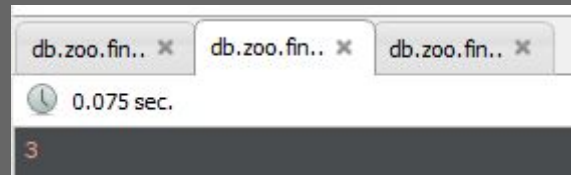
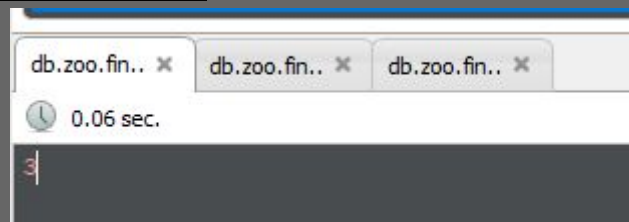
2. Na potrzeby dokumentowej bazy danych datki rozumiemy, jako datki na zwierzę w dokumentowej bazie danych, a w relacyjnej bazie danych jako datki na klatki.

7.

```
CREATE VIEW widok7 AS  
SELECT sekcja,count(*) FROM pracownicy  
GROUP BY sekcja;
```

	sekcja	count(*)	
▶	ssaki	4	
	ptaki	3	
	gady	3	

```
db.zoo.find({"sekcja":"ptaki"}).count()  
db.zoo.find({"sekcja":"gady"}).count()  
db.zoo.find({"sekcja":"ssaki"}).count()
```



8.

```
CREATE VIEW widok8 AS
```

```
SELECT imie, nazwisko FROM rodo WHERE id_pracownik IN
```

```
(SELECT id_pracownik FROM wycieczki GROUP BY id_pracownik HAVING
```

```
count(*)>3);
```

	imie	nazwisko
▶	Cezary	Centkowski

```
db.zoo.find( {"wycieczka4":{"$exists:1}}, {_id:0,imie:1,nazwisko:1} )
```

Key	Value	Type
>  (1)	{ 2 fields }	Object

```
/* 1 */
{
  "imie" : "Cezary",
  "nazwisko" : "Centkowski"
}
```



# Porównanie

MognoDB zdaje się dawać więcej swobody swoim użytkownikom. Jeżeli zarządzał bym małą firmą, która intensywnie się rozwija i ciągle szuka swojego miejsca na rynku, to MongoDB zdaje się być wygodniejszym rozwiązaniem przez swoją elastyczność. W moim odczuciu, również składnia NoSQL jest łatwiejsza do przyswojenia niż w SQL. Również zauważyłem spory potencjał dokumentowych baz danych w data science, ponieważ struktura dokumentowej bazy danych jest bardziej podobna do ramek danych niż w relacyjnych bazach danych.

Gdyby jednak była taka możliwość, to najszybciej jak to tylko możliwe przerzuciłbym się na relacyjną bazę danych. Mimo, że moim zdaniem jest ona sztywna oraz znacząco cięższa w zaprojektowaniu w porównaniu do dokumentowej bazy danych, to jednak łatwiej się w niej odnaleźć przy bardziej skomplikowanych zapytaniach.

Mówiąc już bardziej konkretnie MongoDB:

- nie ma ogólnie ustalonej struktury
- nie łączy kolekcji za pomocą kluczy głównych i kluczy obcych jak w przypadku relacji z MySQL
- nie mamy dedykowanego języka zapytań (w przypadku relacyjnych baz danych jest różnica w zapytaniach na przykład w Oraclu i MySQL)
- umożliwiają rozproszoną dystrybucję, a zatem łatwo jest zwiększyć efektywność pracy serwera poprzez dodanie kolejnego, co nie jest możliwe w relacyjnych bazach danych

Przechodząc do konkretnych przykładów z mojego raportu można zauważyć, że zapytanie numer 1, które było bardzo krótkie w MySQL, przeobraziło się w ogromne zapytanie w MongoDB. Bardzo duży plus dla MySQL. Z kolei 3,4 podpunkt był łatwiejszy do realizacji w MongoDB, co widać po długości zapytania. Pierwszy duży problem napotkałem w 5 podpunkcie, gdy miałem problem z odtworzeniem podzapytania - duży plus dla SQL. Podpunkt 6 mógł wydawać się na korzyść SQL, jednak zauważyłem w dokumentacji MongoDB, że można wykorzystać bardzo dużo fajnych funkcji agregujących, których nie widziałem wcześniej wykorzystanych w MySQL. Być może jest to kwestia dokumentacji. Podpunkt 8 moim zdaniem pokazuje wyższość baz relacyjnych nad dokumentowymi. Przy tworzeniu tej dokumentowej bazy, w celu zachowania jej prostoty byłem zmuszony do stworzenia "wycieczka1", "wycieczka2", "wycieczka3"... co mogłoby być bardzo uciążliwe przy pracy przy takiej bazie danych.