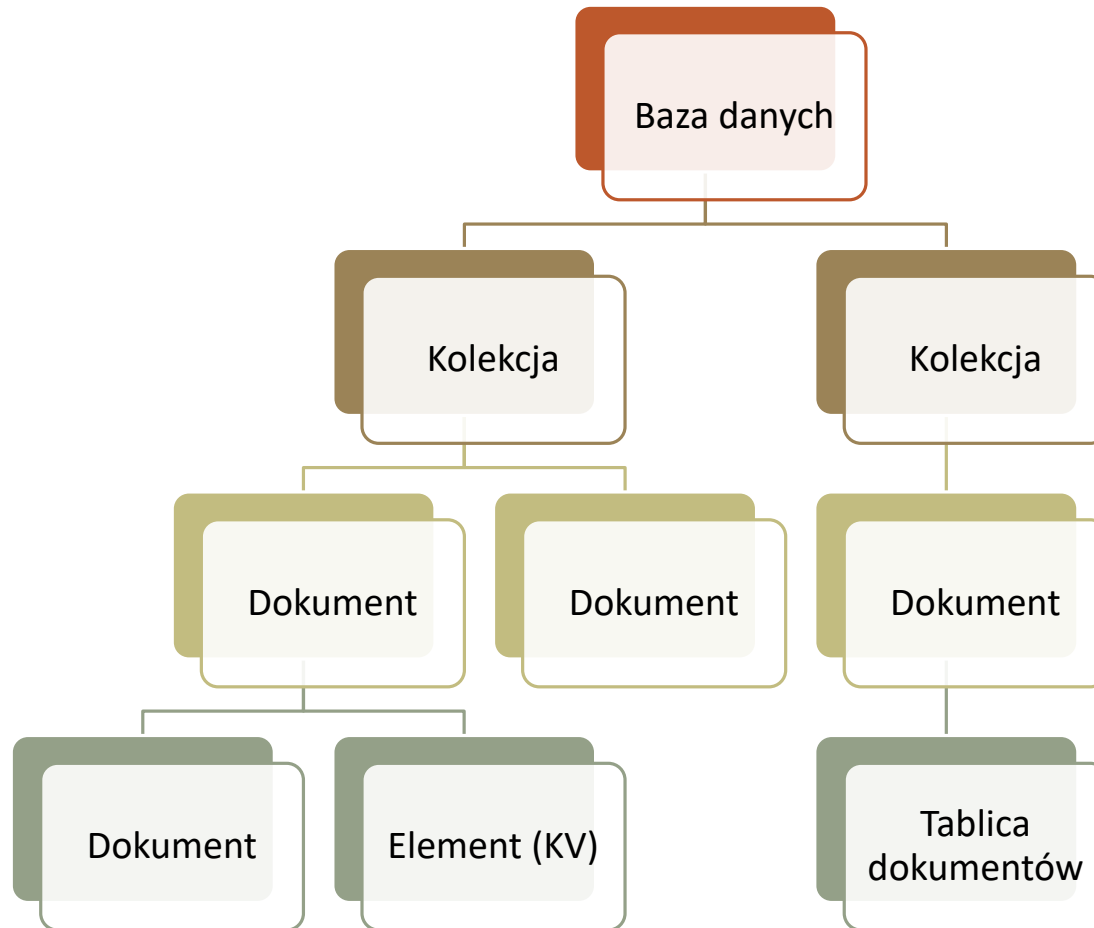


Podstawowe zapytania

Struktury danych



Typy przechowywanych danych

ObjectId – unikalny (w obrębie całej bazy) identyfikator

507f191e810c19729de864ea

4 bajty - epoch (znacznik czasowy)

3 bajty - identyfikator maszyny

2 bajty - id procesu

3 bajty - licznik

Typy przechowywanych danych

String (UTF-8)

Double

Bool

Date

Int32

Int64

Timestamp

Typy przechowywanych danych

Object – zagnieżdżony dokument

Array – lista wartości lub dokumentów

Javascript – przechowywanie skryptów w bazie

Null

Undefined

Zapytania proste - insert

```
db.<collection_name>.insert({"key" : "value"})
```

```
db.foo.insert(  
  {  
    "age": 39,  
    "name": "Dolly Leon",  
    "arrayData" : ["a", "b", "c"]  
  })
```

Zapytania proste - insert

Pozwala na dodanie nowego dokumentu do kolekcji

Przeprowadza podstawową validację – sprawdza czy `_id` nie istnieje już w bazie, czy rozmiar dokumentu nie przekracza **16MB**, validacja JSONa

Jeśli pole `_id` nie istnieje Mongo dodaje automatycznie to pole

Zapytania proste - save

```
db.<collection_name>.save({"key" : "value"})
```

```
db.foo.save(  
  {  
    _id: "4cf00c00bf5862117f55e75",  
    "age": 39,  
    "name": "Dolly Leon",  
    "arrayData" : ["a", "b", "c"]  
  })
```


Zapytania proste - save

Dodaje lub zastępuje dokument

Jeśli `_id` istnieje w bazie wykonywany jest zastąpienie istniejącego dokumentu

Jeśli `_id` nie istnieje w bazie wykonywany jest insert

Zapytania proste - remove

```
//usunięcie wszystkich rekordów z kolekcji  
db.foo.remove({})
```

```
//usunięcie rekordów spełniających zapytanie  
db.foo.remove({"key" : "value"})
```

```
//usunięcie pierwszego rekordu spełniającego zapytanie  
db.foo.remove({"name" : "Dolly Leon"}, true)
```

Zapytania proste - remove

Usuwa dokumenty z bazy

Możliwość usunięcia wszystkich dokumentów lub zgodnych z zapytaniem

Możliwość usunięcia tylko pojedynczego rekordu: `db.foo.remove(query, true)`

Usunięcie danych nie może zostać cofnięte

Zapytania proste - drop

```
db.foo.drop()
```

Zapytania proste - drop

Usuwa całą kolekcję i wszystkie dokumenty które zawiera

Usuwa indeksy zdefiniowane w kolekcji

Wielokrotnie szybsze od delete

Zapytania proste - update

```
db.foo.update(  
  {"name" : "Dolly Leon"},  
  {  
    "age": 30,  
    "name": "Dolly Leon",  
    "arrayData" : ["a", "b", "c"]  
  })
```

```
db.foo.update(  
  {"name" : "Dolly Leon"},  
  {$set:  
    {"age": 30}  
  }  
)
```

Zapytania proste - update

Aktualizuje dokument

Możliwa aktualizacja tylko jednego pola poprzez atrybut \$set

Opcja **upsert** pozwala na dodanie nowego dokumentu, jeśli dokument do modyfikacji nie został znaleziony

Opcja **multi** pozwala na aktualizacje więcej niż jednego dokumentu

Zapytania proste - update

```
db.foo.update(  
  {"name" : "Dolly Leon"},  
  {  
    "age": 30,  
    "name": "Dolly Leon",  
    "arrayData" : ["a", "b", "c"]  
  },  
  {  
    upsert:true,  
    multi:true  
  }  
)
```


Zapytania proste - find

```
//pobieranie wszystkich dokumentów z kolekcji  
db.foo.find()
```

```
//pobieranie wszystkich dokumentów z kolekcji  
//spełniających zapytanie  
db.foo.find({"name" : "Dolly Leon"})
```

Zapytania proste - find

Wyszukuje wszystkie dokumenty w kolekcji spełniające zapytanie

Zwracany jest kursor (jeśli nie zostanie przypisany do zmiennej automatycznie pobiera 20 pierwszych rezultatów)

Zapytania proste – find - kursor

Jest wskaźnikiem na wynik zapytania

Po 10 minutach nieaktywności jest zamykany (możliwość wyłączenia przez `DBQuery.Option.noTimeout`)

Nie jest izolowany – zmiany dokumentów w bazie mogą go modyfikować

Zapytania proste – find - kursor

Cursor.hasNext() – informacja czy kursor zawiera jeszcze jakieś elementy

Cursor.next() – pobiera kolejny dokument

Cursor.forEach() – iteracja po kursorze

Cursor.toArray() – izolacja kursora (pobranie wszystkich danych z kursora)

Zapytania proste – find – sortowanie

```
//sortowanie rosnące  
db.foo.find({"name" : "Dolly Leon"})  
.sort({ "age" : 1})
```

```
//sortowanie malejące  
db.foo.find({"name" : "Dolly Leon"})  
.sort({ "age" : -1})
```

```
//sortowanie wielokrotne  
db.foo.find({"name" : "Dolly Leon"})  
.sort({ "age" : -1, "name" : 1})
```

Zapytania proste – find – sortowanie

Domyślna kolejność pobieranych dokumentów nie jest gwarantowana przez Mongo (nie zawsze ostatnio dodany dokument będzie zajmował ostatnie miejsce w pobranych danych)

Za pomocą metody kursora **Cursor.sort(...)** możemy zapewnić sortowanie rosnące (1) oraz malejące (-1)

Zapytania proste – find – sortowanie

Sortowanie bez indeksów jest bardzo kosztowne (Mongo musi pobrać wszystkie rekordy i posortować je w pamięci)

Możliwe jest wielokrotne sortowanie (wspierane przez indeksy)

Zapytania proste – find – skip, limit

```
//pominięcie pierwszych 10 rekordów  
db.foo.find({"field" : "value"}).skip(10)
```

```
//pobranie pierwszych 20 rekordów  
db.foo.find({"field" : "value"}).limit(20)
```

```
//pominięcie pierwszych 10 rekordów i pobranie kolejnych 20  
db.foo.find({"field" : "value"}).skip(10).limit(20)
```


Zapytania proste – find – skip, limit

Poprzez metody kursora skip i limit jesteśmy w stanie zminimalizować liczbę pobieranych dokumentów

skip(n) – pomija n pierwszych rekordów

limit(n) – ogranicza liczbę pobieranych rekordów do n

Zapytania proste – find – selektory zapytań

//Nie równe

```
db.foo.find({"key" : {$ne: "value"}})
```

//Większe od

```
db.foo.find({"key" : {$gt: "value"}})
```

//Mniejsze od

```
db.foo.find({"key" : {$lt: "value"}})
```

//Pole zawiera przynajmniej jedną wartość z tablicy

```
db.foo.find({"field" : {$in: ["value1", "value2"]} })
```

Zapytania proste – find – selektory zapytań

\$gt – większe od

\$gte – większe lub równe

\$lt – mniejsze od

\$lte – mniejsze lub równe

\$ne – nierówne

Zapytania proste – find – selektory zapytań

\$in – wartość równa przynajmniej jednemu elementowi tablicy

\$nin – wartość nierówna żadnemu elementowi z tablicy lub pole nie istnieje

Zapytania proste – find – selektory zapytań - \$in, \$nin

Selektory \$in oraz \$nin mogą być używane dla tablic

\$in dla tablic sprawdza czy tablica zawiera przynajmniej jeden element z tablicy podanej w zapytaniu

\$nin dla tablic sprawdza czy tablica nie zawiera żadnego elementu z tablicy podanej w zapytaniu

Zapytania proste – find – warunki logiczne

```
db.foo.find(  
  {$and :  
    [  
      {"field1" : "value1"},  
      {"field2" : "value2"}  
    ]  
})
```

```
db.foo.find(  
  {$or :  
    [  
      {"field1" : "value1"},  
      {"field2" : "value2"}  
    ]  
})
```

Zapytania proste – find – warunki logiczne

\$and – dokument spełnia wszystkie warunki

\$or – dokument spełnia przynajmniej jeden warunek

\$not – dokument nie spełnia warunku

\$nor – dokument nie spełnia wszystkich warunków

Zapytania proste – find - exist

```
//pole istnieje  
db.foo.find({"field" : {$exist : 1}})
```

```
//pole nie istnieje  
db.foo.find({"field" : {$exist : 0}})
```


Zapytania proste – find - exist

Pozwala na pobranie tylko dokumentów zawierających (1) lub niezawierających (0) danego pola

Przydatne przy braku ścisłego modelu w kolekcji

Zapytania proste - findOne

```
db.foo.findOne({"name" : "Dolly Leon"})
```

Zapytania proste - findOne

Zwraca pierwszy napotkany wynik

Nie zwraca kursora, a dokument

Brak możliwości sortowania

Obsługuje te same selektory zapytań co find()