



mongoDB

NoSQL база данных



NoSQL = Not only SQL

NoSQL базы данных

- ▶ Key-value (Redis, Dynamo, MemcachedDB, Voldemort)
- ▶ Document (MongoDB, CouchDB, Riak)
- ▶ Wide Column (BigTable, Cassandra, HBase)
- ▶ Graph (Neo4j, InfiniteGraph)



MongoDB

- ▶ Быстрая база данных (C++)
- ▶ Устоявшийся проект, выпущена версия 8.0
- ▶ Open-source, но разрабатывается и поддерживается компанией 10gen
- ▶ Одно из наиболее универсальных решений



Поддержка MongoDB языками



Официальные драйверы

- ▶ Perl
- ▶ PHP
- ▶ Python
- ▶ Ruby
- ▶ C/C++
- ▶ Java
- ▶ .NET
- ▶ Javascript
- Erlang
- Scala
- Haskell

Драйверы open-source

ActionScript, Clojure, Delphi, Node.js, F#, Go, Groovy, Lua, Objective C, Smalltalk и т.д.

Кто использует MongoDB?





MongoDB

Документная база данных

Каждому ключу соответствует документ



```
_id: ObjectId("4daf...") => {  
    first_name: "Gleb",  
    last_name: "Strelchenko",  
    contacts: {  
        email: "glebspartan@gmail.com"  
    },  
    skills: ["php", "mongodb"]  
}
```


Организация данных



SQL	MongoDB
База данных	База данных
Таблица	Коллекция
Строка/запись	Документ

Преимущества

- ▶ Документно-ориентированная база данных
- ▶ Быстрое чтение и особенно запись
- ▶ Широкая по сравнению с другими NoSQL функциональность
- ▶ Масштабирование из коробки
 - ▶ По чтению (Master/slave, Replica sets)
 - ▶ По записи (Sharding)





Но зачем?

Области применения

MongoDB

- ▶ хранение разнородных данных
- ▶ хранение геоданных
- ▶ хранение логов и статистики
- ▶ хранение графовых данных



Пример использования

Система управления контентом

- ▶ HTML-страница
- ▶ Новость
- ▶ Карточка товара



Пример использования

Система управления контентом

SQL

3 таблицы: pages, news, goods

MongoDB

1 коллекция: documents



Пример использования

Система управления контентом

Страница

```
{  
  type: "page",  
  content: "Текст на странице"  
}
```



Пример использования

Система управления контентом

Новость

```
{  
  type: "news",  
  date: new Date(),  
  header: "Заголовок",  
  teaser: "Краткий текст",  
  content: "Полное содержание"  
}
```



Пример использования

Система управления контентом

Карточка товара

```
{  
  type: "good",  
  name: "Телевизор"  
  price: 30000,  
  features: {  
    lcd: 1,  
    led: 0  
  },  
  categories: [ "home", "tv" ]  
}
```





Организация данных

Типы данных BSON

- ▶ String
- ▶ Integer
- ▶ Double
- ▶ Date
- ▶ Byte array (бинарные данные)
- ▶ Boolean
- ▶ Null
- ▶ BSON Object



Ключ

Каждому добавленному документу
автоматически предоставляется уникальный ключ

`_id: ObjectId("47cc67093475061e3d95369d")`





CRUD



Create

use vldc

```
db.users.insert({ first_name: "Oleg" })
```

Read

```
db.users.find()
```

```
db.users.find({}, { first_name: 1 })
```

```
db.users.find({first_name: "Oleg" }).sort({ _id: -1 }).skip(1).limit(10)
```



Update

```
db.users.update({ first_name: "Oleg" }, { $set: { last_name: "Kachan" } })
```

Delete

```
db.users.remove({ _id: ObjectId("4df8fb81ed4cadd6271c0000") })  
db.users.remove({ first_name: "Oleg" })
```

Операторы условий

- ▶ `$gt`, `$lt`, `$gte`, `$lte`
- ▶ `$ne`
- ▶ `$in`, `$nin`
- ▶ `$mod`
- ▶ `$all`

- `$size`
- `$exists`
- `$type`
- `$not`
- `$where`



Операторы модификации

- ▶ \$set
- ▶ \$unset
- ▶ \$inc
- ▶ \$push
- ▶ \$pushAll

- \$addToSet
- \$pop
- \$pull
- \$pullAll





Создание индексов

```
db.users.ensureIndex({ first_name: 1 }) // по возрастанию  
db.users.ensureIndex({ first_name: -1 }) // по убыванию
```

Гео-индекс

```
db.places.ensureIndex({ location: "2d" })
```

Поиск при помощи операторов

- ▶ \$near – поиск объектов с сортировкой, самые близкие - первые
- ▶ \$box – поиск объектов в заданном квадрате
- ▶ \$center – поиск объектов в заданном радиусе



Есть вопросы?