

Проектирование баз данных, ч.2

ФИО преподавателя: Смирнов Михаил Вячеславович
e-mail: smirnovmgupi@gmail.com

Лекция 5

Документная модель MongoDB

Определение СУБД MongoDB.

Документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц.

Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++.

Система поддерживает ad-hoc-запросы: они могут возвращать конкретные поля документов и пользовательские JavaScript-функции. Поддерживается поиск по регулярным выражениям.

Система масштабируется горизонтально, используя технику сегментирования объектов баз данных - распределение их частей по различным узлам кластера. Администратор выбирает ключ сегментирования, который определяет, по какому критерию данные будут разнесены по узлам (в зависимости от значений хэша ключа сегментирования).

Может работать в соответствии с парадигмой MapReduce. Во фреймворке для агрегации есть аналог SQL-инструкции GROUP BY.

Архитектура СУБД MongoDB.

У MongoDB есть официальные драйверы для основных языков программирования (Си, C++, C#, Erlang, Go, Haskell, J#, Java, JavaScript, Lisp, Perl, PHP, Python, Ruby, Delphi, Scala).

Существует также большое количество неофициальных или поддерживаемых сообществом драйверов для других языков программирования и фреймворков.

Основным интерфейсом к базе данных была командная оболочка «mongo». С версии MongoDB 3.2 в качестве графической оболочки поставляется «MongoDB Compass». Существуют продукты и сторонние проекты, которые предлагают инструменты с GUI для администрирования и просмотра данных.

Варианты использования СУБД MongoDB.

1. Хранение и регистрация событий;
2. Системы управления документами и контентом;
3. Электронная коммерция;
4. Компьютерные игры;
5. Данные мониторинга, датчиков;
6. Мобильные приложения;
7. Хранилище операционных данных веб-страниц (например, хранение комментариев, рейтингов, профилей пользователей, сеансы пользователей).

Основной программный инструментарий MongoDB.

mongo - интерактивная оболочка, которая позволяет разработчикам и администраторам просматривать, вставлять, удалять и обновлять данные в своей базе данных, также позволяет настроить репликацию, сегментирование, отключить узлы, выполнить JavaScript или любые другие запросы к базе данных;

mongostat - инструмент командной строки, который суммирует список статистических данных для исполняемого экземпляра MongoDB, это позволяет визуализировать количество вставок, обновлений, удалений, запросов и команд, а также ресурсопотребление экземпляра;

mongosniff - инструмент, обеспечивающий перехват, сохранение и последующее воспроизведение команд, посылаемых в MongoDB;

mongoimport/mongoexport - средства импорта и экспорта из JSON, CSV или TSV, поддерживается ряд других форматов;

mongodump/mongorestore - инструменты создания резервной копии и восстановления базы данных из неё.

Определения элементов MongoDB.

База данных – это физический контейнер для коллекций. Каждая база данных получает свой набор файлов в файловой системе. Один сервер MongoDB обычно имеет несколько баз данных.

Коллекция представляет собой группу документов MongoDB. Это эквивалент таблицы RDBMS. Коллекция существует в одной базе данных. Коллекции не применяют схему. Документы в коллекции могут иметь разные поля. Как правило, все документы в коллекции имеют аналогичное или связанное назначение.

Документ – это набор пар ключ-значение. Документы имеют динамическую схему. Динамическая схема означает, что документы в одной коллекции не обязательно должны иметь одинаковый набор полей или структуру, а общие поля в документах коллекции могут содержать данные разных типов.

Пример документа MongoDB.

```
{
  _id: ObjectId(7df78ad8902c)
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'tutorials point',
  url: 'http://www.tutorialspoint.com',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user: 'user1',
      message: 'My first comment',
      dateCreated: new Date(2011,1,20,2,15),
      like: 0
    },
    {
      user: 'user2',
      message: 'My second comments',
      dateCreated: new Date(2011,1,25,7,45),
      like: 5
    }
  ]
}
```

_id – это 12-байтовое шестнадцатеричное число, которое обеспечивает уникальность каждого документа.

MongoDB предоставит уникальный идентификатор для каждого документа. Эти 12 байтов, первые 4 байта для текущей метки времени, следующие 3 байта для идентификатора машины, следующие 2 байта для идентификатора процесса сервера MongoDB и оставшиеся 3 байта являются простым инкрементным значением.

Схема данных MongoDB.

Проектирование моделей данных в среде СУБД mongoDB не подразумевает создания логической и физической схемы данных (что, впрочем не значит, что не существует вариантов эти СУБД визуализировать).

Вместо этого практикуется использование одной из двух типовых моделей **проектирования**:



Модель данных со встраиванием

Нормализованная модель данных

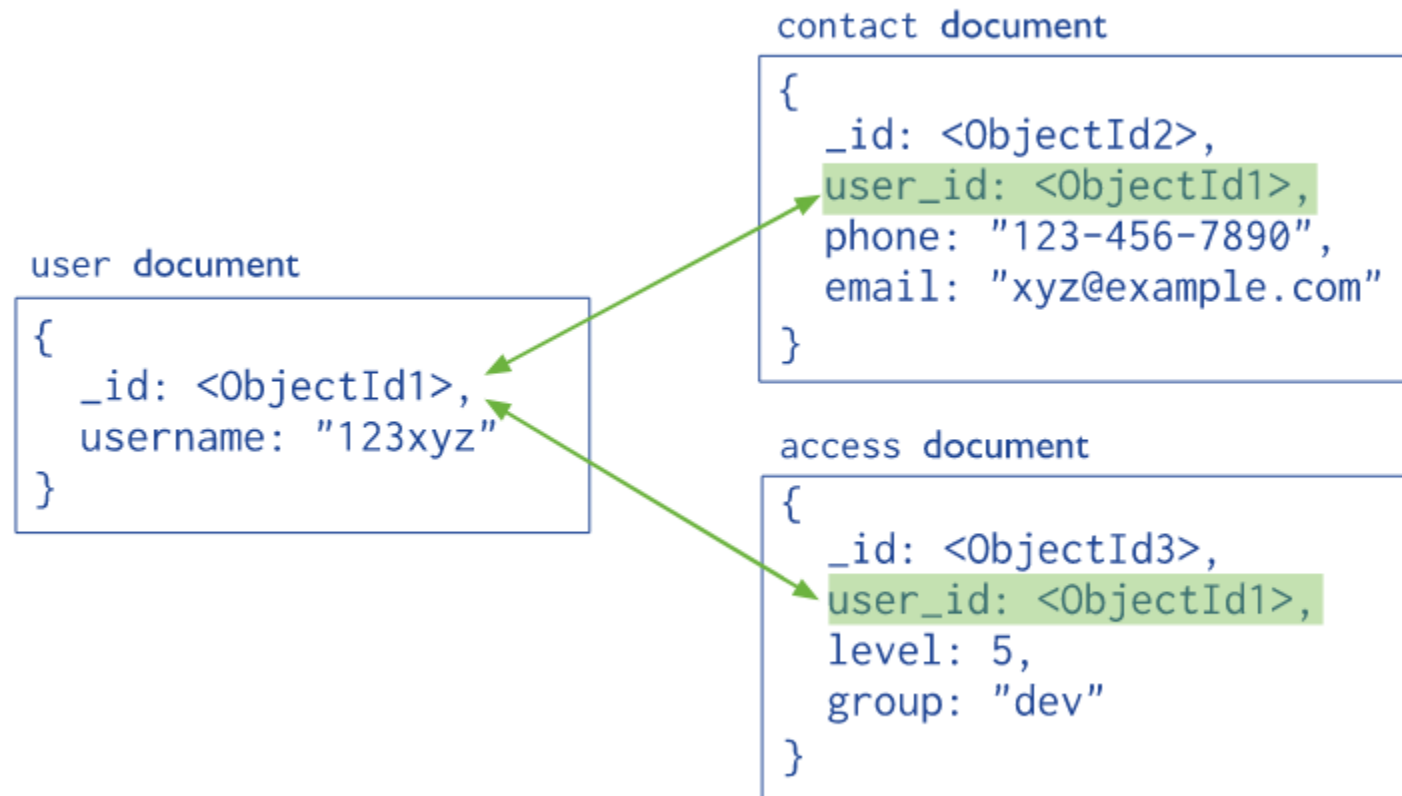
Модель со встраиванием MongoDB.

Также называется “mongodb rich document”. Все связанные данные встраиваются в соответствующий документ.



Нормализованная модель MongoDB.

Нормализованные модели описывают связи, используя ссылки между документами. По логике похоже на реляционную связь.

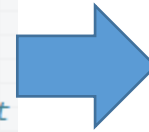


Паттерны моделей данных MongoDB. Модель связи “один-к-одному”, вложенная

Данная модель подразумевает вложение в один документ набора связанных данных.

```
// patron document
{
  _id: "joe",
  name: "Joe Bookreader"
}

// address document
{
  patron_id: "joe", // reference to patron document
  street: "123 Fake Street",
  city: "Faketon",
  state: "MA",
  zip: "12345"
}
```



```
{
  _id: "joe",
  name: "Joe Bookreader",
  address: {
    street: "123 Fake Street",
    city: "Faketon",
    state: "MA",
    zip: "12345"
  }
}
```

Паттерны моделей данных MongoDB. Модель связи “один-ко-многим”, вложенная

Данная модель использует связи между документами, с целью описать отношения один-ко-многим между связанными данными.

```
{
  title: "MongoDB: The Definitive Guide",
  author: [ "Kristina Chodorow", "Mike Dirolf" ],
  published_date: ISODate("2010-09-24"),
  pages: 216,
  language: "English",
  publisher: {
    name: "O'Reilly Media",
    founded: 1980,
    location: "CA"
  }
}

{
  title: "50 Tips and Tricks for MongoDB Developer",
  author: "Kristina Chodorow",
  published_date: ISODate("2011-05-06"),
  pages: 68,
  language: "English",
  publisher: {
    name: "O'Reilly Media",
    founded: 1980,
    location: "CA"
  }
}
```



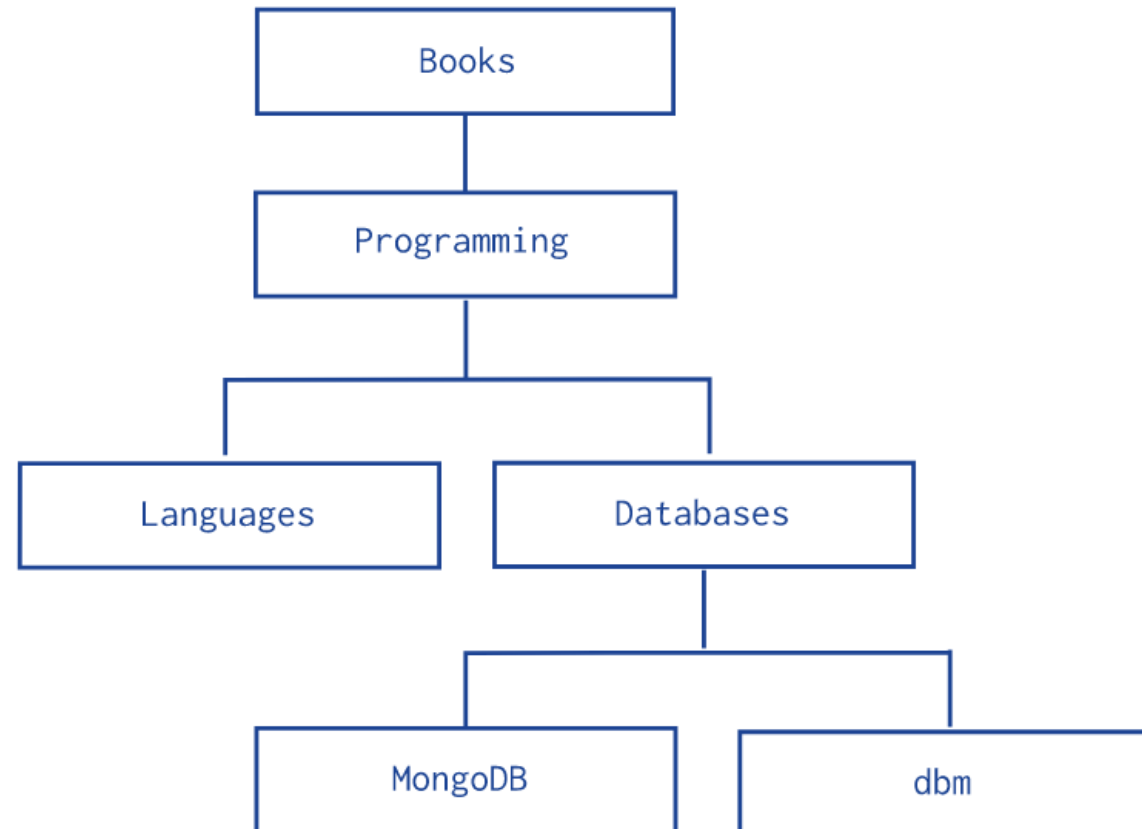
```
{
  name: "O'Reilly Media",
  founded: 1980,
  location: "CA",
  books: [123456789, 234567890, ...]
}

{
  _id: 123456789,
  title: "MongoDB: The Definitive Guide",
  author: [ "Kristina Chodorow", "Mike Dirolf" ],
  published_date: ISODate("2010-09-24"),
  pages: 216,
  language: "English"
}

{
  _id: 234567890,
  title: "50 Tips and Tricks for MongoDB Developer",
  author: "Kristina Chodorow",
  published_date: ISODate("2011-05-06"),
  pages: 68,
  language: "English"
}
```

Паттерны моделей данных MongoDB. Модель дерева.

Приведем пример структуры дерева для рассмотрения на ней разных возможных паттернов.



Паттерны моделей данных MongoDB. Модель дерева. Родительские ссылки.

В данном паттерне для всех документов прописываются их родители.

```
db.categories.insert( { _id: "MongoDB", parent: "Databases" } )
db.categories.insert( { _id: "dbm", parent: "Databases" } )
db.categories.insert( { _id: "Databases", parent: "Programming" } )
db.categories.insert( { _id: "Languages", parent: "Programming" } )
db.categories.insert( { _id: "Programming", parent: "Books" } )
db.categories.insert( { _id: "Books", parent: null } )
```

Паттерны моделей данных MongoDB. Модель дерева. Ссылки на потомков.

В данном паттерне для всех документов прописываются их потомки.

```
db.categories.insert( { _id: "MongoDB", children: [] } )
db.categories.insert( { _id: "dbm", children: [] } )
db.categories.insert( { _id: "Databases", children: [ "MongoDB", "dbm" ] } )
db.categories.insert( { _id: "Languages", children: [] } )
db.categories.insert( { _id: "Programming", children: [ "Databases", "Languages" ] } )
db.categories.insert( { _id: "Books", children: [ "Programming" ] } )
```


Паттерны моделей данных MongoDB. Паттерн для поддержки поиска по тэгам.

В данном паттерне подразумевается использование многоключевого индекса, который будет содержать тэги, доступные для поиска.

```
{ title : "Moby-Dick" ,  
  author : "Herman Melville" ,  
  published : 1851 ,  
  ISBN : 0451526996 ,  
  topics : [ "whaling" , "allegory" , "revenge" , "American" ,  
            "novel" , "nautical" , "voyage" , "Cape Cod" ]  
}
```



```
db.volumes.createIndex( { topics: 1 } )
```



```
db.volumes.findOne( { topics : "voyage" }, { title: 1 } )
```

Паттерны моделей данных MongoDB. Паттерн для модели дата-время.

В данном паттерне документы организуются вокруг временных данных, четко привязанных к минуте-часу-дню-году и т.д.

```
// temperatures collection

{
  "_id": 1,
  "sensor_id": 12345,
  "timestamp": ISODate("2019-01-31T10:00:00.000Z"),
  "temperature": 40
}
{
  "_id": 2,
  "sensor_id": 12345,
  "timestamp": ISODate("2019-01-31T10:01:00.000Z"),
  "temperature": 40
}
{
  "_id": 3,
  "sensor_id": 12345,
  "timestamp": ISODate("2019-01-31T10:02:00.000Z"),
  "temperature": 41
}
...
```



```
{
  "_id": 1,
  "sensor_id": 12345,
  "start_date": ISODate("2019-01-31T10:00:00.000Z"),
  "end_date": ISODate("2019-01-31T10:59:59.000Z"),
  "measurements": [
    {
      "timestamp": ISODate("2019-01-31T10:00:00.000Z"),
      "temperature": 40
    },
    {
      "timestamp": ISODate("2019-01-31T10:01:00.000Z"),
      "temperature": 40
    },
    ...
    {
      "timestamp": ISODate("2019-01-31T10:42:00.000Z"),
      "temperature": 42
    }
  ],
  "transaction_count": 42,
  "sum_temperature": 1783
}
```

Визуализация модели данных MongoDB.

Визуализация (логическая модель) строится с учетом особенностей структуры документов mongo и языка JSON, который используется при структурировании данных.

Документы в коллекции представлены в виде контейнеров, в которых хранятся атрибуты, некоторые из которых множатся на другие документы коллекции.

Связи в одной коллекции могут распространяться как между документами, так и между документами и мультиплицированными атрибутами (свойства разных типов связей документов mongo).

Спасибо за внимание!