



УНИВЕРСИТЕТ ПО БИБЛИОТЕКОЗНАНИЕ И
ИНФОРМАЦИОННИ ТЕХНОЛОГИИ

КУРСОВА РАБОТА
ПО
Администриране на база данни

Тема: „Администриране на MongoDB“

Изготвил:

магистърска програма
„Информационни технологии“

гр. СОФИЯ
05.01.2015 г.

Ръководител:

УВОД

База данни е система за организиране на колекции от данни. Всеки софтуерен продукт, уеб или мобилно приложение използва база данни под някаква форма. Основните два типа са релативни и нерелативни база данни. MongoDB е нерелативна база данни, лидер в областта. Популярността именно на тази база данни идва от факта, че работи с почти всички програмни езици, с основните операционни системи и следва тенденциите в програмирането. В обектно ориентирания подход в програмирането се насърчава ползване на логически модели. MongoDB позволява записа на информацията да следва този подход, вместо тя да се разбива на серия от редове в различни таблици.

I. ГЛАВА ПЪРВА

Въведение в нерелационни база данни и MongoDB

1.1 Нерелационни база данни

Характерно за нерелационните база данни е начина, по който съхраняват информацията:

- Използва документно ориентиран модел;
- Данните се организират в колекция от документи;
- Пази данните във вид на модела ключ-стойност (key-value stores);
- Документния модел няма фиксирана схема;
- Денормализирани данни.

За разлика от релационните база данни, тук не е необходимо предварително да се декларира типа на променливите при създаване на таблиците. Всеки запис в нерелационните база данни може да съдържа различни типове данни. При този вид база данни съответно няма релации между отделните записи, а цялата необходима информация се намира на едно място - фигура 1¹. Това води до повторения на някой данни, но същевременно изисква по-малко операции, които трябва да се правят за извеждане на данните и е причината този вид база данни да се сочат за по-добри от към представяне.

¹ Телерик, курс База данни - <http://telerikacademy.com/Courses/Courses/Details/21>



Фигура 1

В същото време нерелационните база данни поддържат същите стандартни операции като релационните база данни:

- CRUD операции – създаване, четене, обновяване и изтриване (create, read, update, delete - CRUD)
- Индексиране и заявки (indexing и querying)
- Транзакции и едновременност

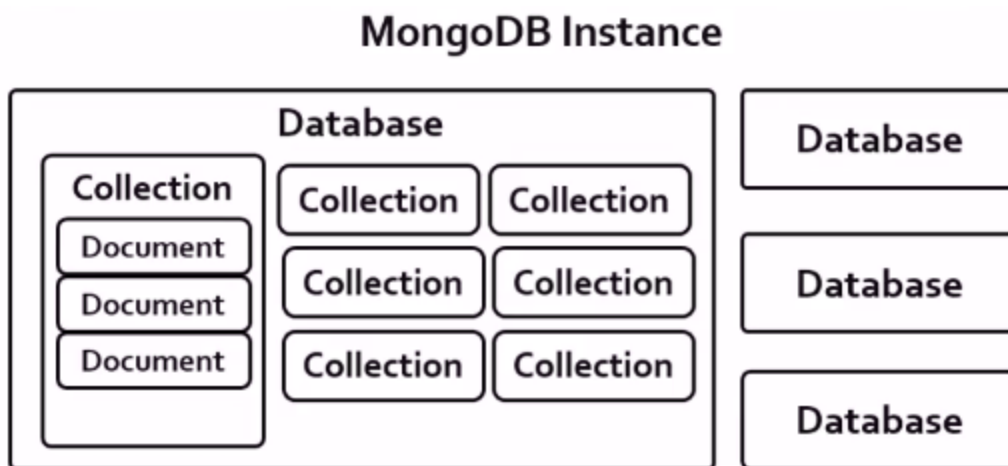
1.2 Въведение в MongoDB

MongoDB е база данни с отворен код. Името и идва от английското (американски английски) “humongous” – внушителен, огромен. Това е нерелационна база данни, ползваща документно ориентирания структурен модел. MongoDB работи на основните сървърни платформи

(Linux, OS X, Solaris, Windows), като може да се използва с почти всички програмни езици². Също така популярността ѝ расте и застава на челни позиции в класациите на *db-engines.com*³.

1.3 Терминология на MongoDB

Като нерелационна база данни, част от термините са различни. Таблицата, реда и колоната, познати ни от релационните база данни, тук са заместени от колекция, документ и поле. Колекцията представлява група от документи и е еквивалент на таблица. Колекцията обединява в себе си документи със свързано съдържание - фигура 2⁴. Всеки запис в колекцията в базата представлява документ с цялата налична информация, а колоните в MongoDB се наричат полета.



Фигура 2

² Подробни синтезирано описани характеристики може да се видят на <http://db-engines.com/en/system/MongoDB>

³ db-engines.com - <http://db-engines.com/en/ranking>

⁴ Телерик, курс База данни - <http://telerikacademy.com/Courses/Courses/Details/211>

1.4 Въведение в MongoDB CRUD операции

MongoDB съхранява информацията във формат на документ, който има JSON-подобна структура. Документа следва логиката на структурата в програмните езици, асоциирана с връзката „ключ-стойност“.

Както споменахме, документите в една колекция са обединени по някаква обща характеристика. Съответно заявките в MongoDB се обръщат към специфична колекция от документи. В заявката се описват критериите и условията, по които да се направи обръщане към конкретен документ. Също така може да се конкретизира и определено поле в документа, към което да е насочена заявката (*projection*). Има възможност към заявките да се добавят и допълнителни изисквания, като операциите: *limits* - за ограничаване на резултата от селекцията; *skips* - за прескачане на резултата от селекцията; и *sort orders* - за подредба на резултата от селекцията – фигура 3⁵.

Collection Query Criteria Modifier
`db.users.find({ age: { $gt: 18 } }).sort({age: 1 })`

Фигура 3

Заявката, с която може да се покажат данните от дадена колекция по определени критерии е: *db.collection.find()*. Това е еквивалента на SQL заявката *select* – фигура 4⁶. Тъй като в MongoDB може да се обръща само към определена колекция метода *join*, познат от SQL, липсва.

⁵ <http://docs.mongodb.org/manual/core/crud-introduction/>

⁶ <http://docs.mongodb.org/manual/core/read-operations-introduction/>

Схема на select операция в MongoDB:

<code>db.users.find(</code>	← collection
<code> { age: { \$gt: 18 } },</code>	← query criteria
<code> { name: 1, address: 1 }</code>	← projection
<code>).limit(5)</code>	← cursor modifier

Схема на същата операция в SQL:

<code>SELECT _id, name, address</code>	← projection
<code>FROM users</code>	← table
<code>WHERE age > 18</code>	← select criteria
<code>LIMIT 5</code>	← cursor modifier

Фигура 4

Операциите за промяна на данните в базата данни са: създаване - *insert*, обновяване - *update*, и изтриване - *remove*. В MongoDB тези операции могат да се обръщат към една единствена колекция от данни.

- *insert* – това е метод, с който се създава нов документ в една колекция (фигура 5⁷)

Схема на insert операцията в MongoDB:

<code>db.users.insert (</code>	← collection	
<code>{</code>		
<code> name: "sue",</code>	← field: value	} document
<code> age: 26,</code>	← field: value	
<code> status: "A"</code>	← field: value	
<code>}</code>		
<code>)</code>		

Схема на същата операция в SQL:

<code>INSERT INTO users</code>	← table
<code>(name, age, status)</code>	← columns
<code>VALUES ("sue", 26, "A")</code>	← values/row

Фигура 5

⁷ <http://docs.mongodb.org/manual/core/write-operations-introduction/#insert>

- *update* – променя съществуващ документ в дадена колекция. В метода може да се добавят определени критерии, по-които да се избере конкретния документ – фигура 6⁸. Има възможност да се обновяват няколко документа едновременно.

Схема на update операцията в MongoDB:

```
db.users.update(
  { age: { $gt: 18 } },
  { $set: { status: "A" } },
  { multi: true }
)
```

← collection
← update criteria
← update action
← update option

Схема на същата операция в SQL:

```
UPDATE users
SET status = 'A'
WHERE age > 18
```

← table
← update action
← update criteria

Фигура 6

- *remove* – метода за изтриване на документ от дадена колекция. В заявката за изтриване се подават параметрите, с които да се избере конкретния документ за изтриване – фигура 7⁹.

Схема на remove операцията в MongoDB:

```
db.users.remove(
  { status: "D" }
)
```

← collection
← remove criteria

Схема на същата операция в SQL:

```
DELETE FROM users
WHERE status = 'D'
```

← table
← delete criteria

Фигура 7

⁸ <http://docs.mongodb.org/manual/core/write-operations-introduction/#update>

⁹ <http://docs.mongodb.org/manual/core/write-operations-introduction/#remove>

II. Глава Втора

Структура и типове на данните в MongoDB

2.1 Модел на структурата на данните

Данните в MongoDB са в не структурирана схема (flexible shema), ползвайки JSON модел за съхраняване на данните. За разлика от SQL база данни, където трябва да се дефинира табличната структура преди въвеждане на данните, колекциите на MongoDB не изискват предварително задаване на модела (структурата) на документа. Това улеснява обхождането на различните стойности на документа, както и сравняването по между им, дори и да съдържат различни по тип данни. В същото време различните документи имат сходна структура използвайки JSON документната структура – фигура 8¹⁰.



Фигура 8

¹⁰ Телерик, курс База данни - <http://telerikacademy.com/Courses/Courses/Details/211>

Както при останалите база данни, така и тук при създаването на нова база, структурирането на данните трябва да бъде планира спрямо нуждите и приложението на съответното приложение. В документацията на MongoDB¹¹ са посочени два модела на структуриране на данните:

- Модел на вмъкнати данни (embedded data models) – Фигура 9.
- Нормален модел на данните – Фигура 10.



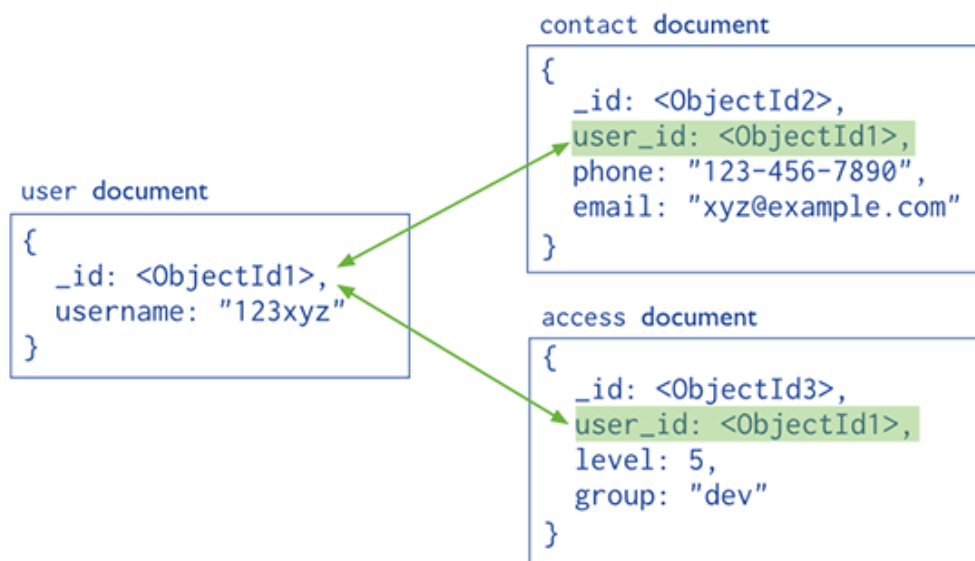
Фигура 9

Модела на вмъкнатите данни е характерен за нерелационните база данни. С този модел в един документ под формата на вмъкване се добавя допълваща информация, която в другите база данни ще се представи под формата на релация. Основното за този модел е, че осигурява по-добро представяне на операцията за четене на данни, както и извличане на съответна информация с една операция. Също така обновяването на определена информация в базата става с

¹¹ [mongodb.org - http://docs.mongodb.org/manual/core/data-modeling-introduction/](http://docs.mongodb.org/manual/core/data-modeling-introduction/)

единична оптимизирана операция наричана на английски - *single atomic write operation*.

Тъй като в този модел цялата информация се съдържа в един запис (документ), трябва да се има предвид при избора на този подход, до колко ще се разрастват данните в базата. Документа в MongoDB се съхранява в BSON формат, който сам по себе си има ограничение. Максималния размер на един BSON документ е *16 megabytes*.



Фигура 10

С нормалния модел се правят връзки между различните документи с референция. Така няма повтаряне на една и съща информация в различните документи, но изисква допълнителни заявки за показване на пълната информация. В същото време позволява по-сложна релация между документите и между повече документи едновременно (*many-to-many relationships*).

2.2 Типове данни

Данните в MongoDB се пазят във формат на BSON (*Binary JSON*), което е бинарният вариант на JSON (*JavaScript Object Notation*).

Типовете данни са¹²:

- JSON примитивни типове данни – числа, стринг / низ, булеви стойности (*true/false*)
- BSON примитивни типове данни - дата, *ObjectId*, *UUID*, регулярни изрази (*regex*)
- Обекти – наричани още вмъкнати документи
- Масиви

Така структурата на документа получава схемата „поле-стойност“.

За наименоуване на полета има определени правила, които трябва да се следват:

- Името на полето да е стринг/низ
- Името на полето не може да започва със знака за долар (\$)
- Името на полето не може да съдържа точка (.)
- Името на полето не може да е празно

Същите правила се отнасят и до наименоуването на отделните колекции.

Като в релационните база данни, така и в MongoDB има поле, което съдържа уникална стойност за всеки документ, така наречения *primary key*. Полето има име *_id* и е задължително за всеки документ. Трябва да е уникално за всеки документ, като може да е от всеки тип данни освен масив. То може както да се генерира автоматично, без да се предвижда

¹² Copeland. Rick, MongoDB Applied Design Patterns – O'Reilly, 2013г

поле за него при записа на даден документ, така и да се генерира при създаването.

2.3 Дизайн на база данни с MongoDB

При създаването на всяко приложение освен дизайна на кода, съществено за неговото правилно функциониране е правилно разпределяне на данните и структурирането им - дизайна на базата данни. MongoDB е създадена да работи с големи данни, на различни сървъри и с облачни технологии. В същото време е нерелационна база данни с динамична схема.

За да може едно приложение да работи оптимално трябва да се определи начина на употреба на данните. Факторите, които трябва да се вземат под внимание са:

- За какво ще се ползват данните – само за четене и показване или ще се обновяват непрекъснато;
- Колко големи ще са данните;
- Каква част от данните ще са най-често използвани;
- Къде ще се съхраняват.

Отговора на тези въпроси ще спомогне за най-оптималното организиране на структурата на базата.

III. Глава трета

Сигурност

3.1 Сигурност на MongoDB база данни

Освен структурата на данните за всяко приложение е важно и тяхната сигурност. Важно изискване при разработката на администрация е осигуряване на контрола на достъп. Трябва да е възможно всеки потребител да може да се легитимира, за да бъде разпознат дали има право на достъп и какви операции може да извършва. Това може да става чрез:

- а) удостоверяване на автентичността (Authentication) – механизъм за удостоверяване на правото на достъп от потребителя. В MongoDB се поддържат различни подходи (удостоверяване на достъпа чрез сертификати) - *x509 certificate*, *LDAP proxy*, *Kerberos*;
- б) ниво на достъп до базата – механизъм за ограничаване на правата на всеки потребител за операции извършващи се върху базата;
- с) кодиране (*encryption*) – може да се случва в няколко направления:
 - кодиране на връзката с помощта на SSL сертификат;
 - кодиране на данните - данните, които не се ползват (*Encryption at Rest*); кодиране на определени нива на данните спрямо поле или спрямо документ; цялостно кодиране на данните по място на съхранение или спрямо операционна система и др.

Заклучение

MongoDB е сравнително нова база данни, която бързо набира популярност. Нейната гъвкава структура позволява по-добро планиране на структурата на данните, което води до по-оптимално представяне на приложението, което я използва. Това се оказва особено необходимо с развитието на мобилните устройства, с което става ясно, че оптималното представяне на работа е от изключителна важност. Не трябва да се пропуска и факта, че MongoDB работи на различни платформи и на почти всички програмни езици. Като едно от най-голямото преимущество е, че е свързан с развитието и тенденциите в програмирането и позволява много по-лесна работа на платформи базирани на JavaScript език.

Едни от най-известните сайтове и приложения, които ползват MongoDB са – Foursquare (за съхраняване на „отбелязванията“ на потребителите), eBay (използват MongoDB при извеждане на предложенията за търсене - „*search suggestion*“), MetLife, City of Chicago, Forbs и други¹³.

¹³ Описаните в официалния сайт <http://www.mongodb.com/> и официалната страница в Wikipedia - <http://en.wikipedia.org/wiki/MongoDB>

Използвана литература

- 1) Телерик, курс база данни -
<http://telerikacademy.com/Courses/Courses/Details/211>
- 2) Телерик, курс End-to-end JavaScript приложения -
<http://academy.telerik.com/school-academy/season-2013-2014/end-to-end-javascript-apps>
- 3) db-engines.com - <http://db-engines.com/en/ranking>
- 4) Copeland. Rick, MongoDB Applied Design Patterns – O`Reilly, 2013г.
- 5) mongodb.org - <http://docs.mongodb.org/>
- 6) mongodb.com - <http://www.mongodb.com/>
- 7) tutsplus.com - <http://code.tutsplus.com/tutorials/getting-started-with-mongodb-part-1--net-22879>
- 8) Wikipedia - <http://en.wikipedia.org/wiki/MongoDB>

Съдържание

УВОД.....	2
I. ГЛАВА ПЪРВА Въведение в нерелационни база данни и MongoDB	3
1.1 Нерелационни база данни.....	3
1.2 Въведение в MongoDB.....	4
1.3 Терминология на MongoDB	5
1.4 Въведение в MongoDB CRUD операции	6
II. Глава Втора Структура и типове на данните в MongoDB	9
2.1 Модел на структурата на данните	9
2.2 Типове данни	12
2.3 Дизайн на база данни с MongoDB	13
III. Глава трета Сигурност	14
3.1 Сигурност на MongoDB база данни	14
Заклучение	15
Използвана литература	16
Съдържание	17