

## ПРАКТИЧЕСКАЯ РАБОТА 7. РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ В MONGODB

### 7.1 Цель и содержание

Цель работы: знакомство с регулярными выражениями MongoDB.

Задачи работы: научиться использовать регулярные выражения при составлении запросов.

### 7.2 Регулярные выражения в MongoDB

Регулярные выражения в MongoDB используют Perl-синтаксис (Perl-совместимые регулярные выражения). Регулярные выражения являются мощным инструментом для составления шаблонных запросов. Представляют собой строку, состоящую из метасимволов и задающую правила поиска.

Регулярные выражения – формальный язык поиска и осуществления манипуляций с [подстроками](#) в тексте, основанный на использовании метасимволов. В MongoDB регулярные выражения используются для составления запросов. При составлении регулярных выражений необходимо руководствоваться следующими правилами:

1. Любой символ обозначает себя самого, если это не метасимвол. Если вам нужно отменить действие метасимвола, то поставьте перед ним '\'.
2. Строка символов обозначает строку этих символов.
3. Множество возможных символов (класс) заключается в квадратные скобки '[]', это значит, что в данном месте может стоять один из указанных в скобках символов. Если первый символ в скобках '^' – это значит, что ни один из указанных символов не может стоять в данном месте выражения. Внутри класса можно употреблять символ '-', обозначающий диапазон символов. Например, a-z – один из малых букв латинского алфавита, 0-9 – цифра и т.д.
4. Все символы, включая специальные, можно обозначать с помощью '\' как в языке C.
5. Альтернативные последовательности разделяются символом '|'. Заметьте, что внутри квадратных скобок это обычный символ.
6. Внутри регулярного выражения можно указывать "подшаблоны" заключая их в круглые скобки и ссылаться на них как '\номер'. Первая скобка

обозначается как '\1'.<sup>14</sup>

При составлении запросов используются следующие опции:

i	не различать строчные и заглавные буквы
m	многострочная строка

В шаблонах используются следующие метасимволы:

\	следующий метасимвол считается обычным символом
^	начала строки
.	один произвольный символ. Кроме символа конца строки (\n)
\$	конец строки
	или
()	группировка
[]	класс символов

Метасимволы имеют модификаторы, которые пишутся после них:

*	повторяется 0 или большее число раз
+	повторяется 1 или большее число раз
?	1 или 0 раз
{n}	повторяется точно n раз
{n,}	повторяется, по меньшей мере, n раз
{n,m}	повторяется не меньше n, но и не больше m

---

<sup>14</sup> Введение в Perl. Регулярные выражения (шаблоны) [Электронный учебник]. – Режим доступа: [http://citforum.ru/internet/perl\\_tut/re.shtml](http://citforum.ru/internet/perl_tut/re.shtml)

Фигурные скобки во всех других случаях считаются обычными символами.

По умолчанию действие метасимволов распространяется столько раз, сколько это возможно, не учитывая результат действия следующих метасимволов.

При составлении регулярных выражений можно использовать бэкслэш-символы. Наиболее часто встречающиеся символы приведены в таблице 1.

Таблица 1 – Наиболее часто встречающиеся бэкслэш-символы

Обозначение	Описание
\t	символ табуляции
\n	новая строка
\r	перевод каретки
\v	вертикальная табуляция
\e	escape
\033	восьмеричная запись символа
\x1A	шестнадцатеричная
\l	нижний регистр следующего символа
\u	верхний регистр следующего символа
\L	все символы в нижнем регистре до \E
\U	в верхнем регистре следующего символа
\E	ограничитель смены регистра
\s	один space символ (пробел, табуляция, новая строка)
\S	один не space символ
\d	одна цифра
\D	одна не цифра
\w	алфавитно-цифровая последовательность и символ ‘_’
\W	не алфавитно-цифровая последовательность и символ ‘_’

Все вышеперечисленные бэкслэш-символы при распознавании регулярного выражения воспринимается как один символ. Для того чтобы обозначить последовательности необходимо комбинировать бэкслэш-символы с модификаторами метасимволов, например:

`\d+` - одна или больше цифр, то же, что и `[0-9]+`

`\w+` – слово, тоже что и `[a-zA-Z0-9_]+`

Для закрепления результата рассмотрим несколько примеров.

Пример 1: необходимо составить запрос, для отыскания всех документов с именем «document» или «Document» или «DOCUMenT» или другим, чередующим строчные и заглавные буквы:

```
db.test.find( { "name": /document/i })
```

Рассмотрим данный запрос внимательнее. Регулярное выражение начинается и заканчивается символом `'/'`. Далее следует имя запрашиваемого документа: «document». Далее, опция `<i>` – не различать строчные и заглавные буквы.

Пример 2: запрос, отыскивающий все документы, содержащие в имени последовательность символов «umen» в любом регистре:

```
db.test.find( { "name": /\w+umen\w/i })
```

или

```
db.test.find( { "name": /umen/i })
```

Запрос состоит из:

`/` – начало запроса;

`\w+` – любая цифробуквенная последовательность;

`umen` – часть слова;

`\w` – цифробуквенная последовательность;

`/` – конец запроса;

`i` – не различать строчные и заглавные символы.

Пример 3: запрос для нахождения всех документов, начинающихся с последовательности «doc»:

```
db.test.find( { "name": /^doc\w+/i })
```

В данном запросе явно указывается начало строки «`^doc`» и любая дальнейшая цифробуквенная последовательность («`\w+`»).

Пример 4: составить запрос для отыскания всех документов, имеющих сложное имя: первые три символа являются буквами; четвертый символ – цифра; пятый символ – любой. Любая последовательность цифробуквенных символов. Последний символ в имени документа – цифра 5.

Пример имени документа, удовлетворяющий приведенным требованиям: «Doc7sH\_u67i5».

Запрос для отыскания данного документа:

```
db.test.find( { name: /\w{3}\d\w+5$/ })
```

Пример 5: ознакомьтесь с часто встречающимися регулярными выражениями, которые представлены в таблице 2.

Таблица 2 – Наиболее часто встречающиеся регулярные выражения и их составные части

Регулярное выражение	Описание
/men/	Проверяет, есть ли в строке подстрока «men»
/^doc/	Проверяет, начинается ли строка с «doc»
/ent\$/	Проверяет, оканчивается ли строка на «ent»
/d o c/ [doc]	Проверяет, содержит ли строка символ «d» или «o» или «c»
/do{1, 3}c	Проверяет, содержит ли строка символ «c», следующие за ним 1-3 символа «o», за которыми следует символ «c»
/docu...t/	Проверяет, содержит ли строка символы «docu» и «t», разделенные тремя символами
^d/	Проверяет, содержит ли строка цифру

### 7.3 Методика и порядок выполнения работы

Индивидуальное задание:

Создайте различные типы запросов с использованием регулярных выражений:

- 1) поиск документа, начинающегося с определенной последовательности символов;
- 2) поиск документов, содержащих определенную последовательность символов;
- 3) поиск документов со сложным именем (шаблонное имя).

Составьте запросы с регулярными выражениями, примеры которых не описаны в теоретическом описании практической работы.