Lab-12 : Regular Expression (Regex) Method

- ✓ A regular expression (regex) is a pattern used to search text values in documents.
- ✓ Regex is mainly used with the **find()** method to perform pattern matching.

General Syntax of Regex:

```
db.<collection_name>.find(
  { <field_name>: { $regex: "<pattern>", $options: "<option>" } }
)
```

$regex → Specifies the pattern to search in the field.

$options → Specifies matching options such as case-insensitive search.

Another Regex Syntax (Shortcut Form):

```
db.<collection_name>.find({ field: /pattern/ })
```

Common Regex Operations:

1. **Search Values Starting With a Pattern :** db.<collection>.find({ field: /^pattern/ })

   Ex. **Find employees whose name start with E.**

   Ans. db.employee.find({ENAME:{$regex:"^E"}})

   OR

   db.employee.find({ENAME:/^E/})

   Ex. Find employees whose name starts with S or M in your collection.

   Ans. db.employee.find({ENAME:/^[SM]/})

2. **Search Values Ending With a Pattern :** db.<collection>.find({ field: /pattern$/ })

   Ex. Find employees where city name ends in 'ney'.

   Ans. db.employee.find({CITY:/ney$/})

3. **Search Values Containing a Pattern : db.<collection>.find({ field: /pattern/ })**

Ex. Display employee name whose name contains n. (Both uppercase(N) and lowercase(n)) – case insensitive

Ans. db.employee.find({ENAME:/n/i})

OR

db.employee.find({ENAME:{$regex:"n",$options:"i"}})

Ex. Find employees whose names do not contain 'a'

Ans. db.Employee.find({ ENAME: { $not: /a/ } })

4. **Search Values Based on Length or Character Pattern:**

Ex. Display employee name whose name starts with E and having 5 characters.

Ans. db.employee.find({ENAME:/^E.{4}$/})

Ex. Find employees whose names start with 'J' and end with 'n'.

Ans. db.Employee.find({ ENAME: /^J.*n$/ })

Ex. Find employees whose names have exactly 4 letters.

Ans. db.Employee.find({ ENAME: /^.{4}$/ })

Ex. Find employees whose names have three or more letters. – At least 3 letters.

Ans. db.Employee.find({ ENAME: /^.{3,}$/ })

Ex. Find employees whose names have two consecutive vowels (a, e, i, o, u).

Ans. db.Employee.find({ ENAME: /[aeiou]{2}/ })

Ex. Find employees whose names contain exactly one vowel.

Ans. db.Employee.find({ ENAME: /^[^aeiou]*[aeiou][^aeiou]*$/i })

	(1)	(2)	(3)	
/^	**[^aeiou]***	**[aeiou]**	**[^aeiou]***	**$/i**
	(non-vowels)	**(one vowel)**	**(non-vowels)**	

(1) **[^aeiou]** means any character except vowels.
***** means **zero or more occurrences**.

(2) **[aeiou]** Matches exactly one vowel (must contain one vowel at this position)

(3) **[^aeiou]*** Again allows zero or more non-vowel characters after the vowel.

Remember These Important Regex Symbols and Patterns

Symbol / Pattern	Meaning	Example Usage
^	Starts with pattern	^A
$	Ends with pattern	n$
.	Any single character	A.n
*	Zero or more occurrences	A*
+	One or more occurrences	A+
.*	Any number of characters	^J.*n$
.{n}	Exactly n characters	^.{4}$
.{n,}	At least n characters	^.{3,}$
[abc]	Matches a, b, or c	/^[SM]/
[^abc]	Not a, b, or c	/[^a]/
[0-9] or \d	Any digit	/\d/
[A-Za-z]	Any alphabet letter	/^[A-Za-z]+$/
\s	Any whitespace	/\s/
i	Case-insensitive match	/rajkot/i

Darshan Halani