

# REGULARNI SKAZI

---

- Regularni iskaz (regular expression) je specijalni tekst string koji opisuje patern(uzorak) koji se traži.
- Regularni iskazi su slični ali moćniji od wildcard iskaza.
- Idealni su za validaciju podataka i traženje željenog podatka u bazama podataka.
- Pretraga zasnovana na regexp daje rezultat na osnovu samo jednog prolaza kroz podatke.

# OSOBI NE REGEXP

## Patern (^)

---

- REGEXP **nije case sensitive**, osim ukoliko se koristi **binary** string.
- MySQL upit prikazuje imena studenata čija imena počinju sa 'k'.
- '^' se koristi za mečovanje početka imena.

```
SELECT *  
FROM Student  
WHERE Ime REGEXP '^k';
```

MySql iskaz prikazuje imena studenata koja počinju sa 'k' vodeći računa o malim tj. velikim slovima jer se koristi **BINARY** operator.

```
SELECT *  
FROM Student  
WHERE Ime REGEXP BINARY '^w';
```

# OSOBI NE REGEXP

## Patern (\$)

---

- MySQL iskaz pronalazi imena studenata čija se imena završavaju na 'on'.
  - '\$' se koristi za mečovanje završetka stringa.

```
SELECT *  
FROM Student  
WHERE Ime REGEXP "on$";
```

- MySQL iskaz pronalazi autore čija imena sadrže 't' u imenu.

```
SELECT *  
FROM Student  
WHERE Ime REGEXP "t";
```

# OSOBI NE REGEXP

## Patern ([...])

---

- MySql iskaz pronalazi imena studenata koja sadrže sledeća slova u imenu 'z' ili 'v' ili 'm'.

```
SELECT *  
FROM Student  
WHERE Ime REGEXP "[zvm]";
```

- MySql iskaz pronalazi imena studenata koja sadrže karaktere od b do g

```
SELECT *  
FROM Student  
WHERE Ime REGEXP "[b-g]" ;
```

# OSOBI NE REGEXP

## Patern (.) i ({...})

---

- MySql iskaz pronalazi imena studenata koja sadrže tačno 12 karaktera.
- Koristi se '^' i '\$' podudaranje za početak i kraj imena i **12 instanci '.'** za podudaranje sa 12 karaktera.
- Tačka (.) je patern koji odgovara jednom bilo kom karakteru

```
SELECT *  
FROM Student  
WHERE Ime REGEXP '^.....$';
```

- MySql iskaz pronalazi imena studenata koja sadrže tačno 12 karaktera.
- Koristi se '^' i '\$' podudaranje za početak i kraj imena i instanca '.' .
- Srednje zagrade {} za definisanje koliko puta se tačka ponavlja.

```
SELECT * FROM Student  
WHERE Ime REGEXP '^.{12}$';
```

# OSOBI NE REGEXP

## Paterni

---

Patern	Šta patern meču je
^	Početak stringa
\$	Kraj stringa
.	Jedan bilo koji karakter
[...]	Bilo koji karakter koji se nalazi u uglastim zagradama
[^...]	Bilo koji karakter koji se ne nalazi u uglastim zagradama
p1 p2 p3	Alternativa; meču je ili p1 ili p2 ili p3 patern
*	Nula ili više instanci prethodnog elementa
+	Jedna ili više instanci prethodnog elementa
{n}	n instanci prethodnog elementa
{m,n}	m do n instanci prethodnog elementa

# OSOBI NE REGEXP

## Podudaranje po tipu klase

Klase	Opis mečovanja
<code>[:alnum:]</code>	Alfanumerički – bilo koji broj ili slovo. Odgovara [a-z], [A-Z] i [0-9]
<code>[:alpha:]</code>	Alfa – bilo koje slovo. Odgovara [a-z] ili [A-Z]
<code>[:blank:]</code>	Space ili Tab. Odgovara [\t] ili [ ]
<code>[:cntrl:]</code>	ASCII Kontrolni karakteri
<code>[:digit:]</code>	Numerički. Odgovaraju [0-9]
<code>[:graph:]</code>	Bilo koji karakter sa izuzetkom space
<code>[:lower:]</code>	Mala slova. Odgovara [a-z]
<code>[:punct:]</code>	Karakter koji nisu kontrolni ni alfanumerički (interpunkcijski znakovi)
<code>[:space:]</code>	Bilo koji razmak (tab, novi red, space ...)
<code>[:upper:]</code>	Velika slova. Odgovara [A-Z]
<code>[:xdigit:]</code>	Bilo koje hexa vrednosti. Odgovara [A-F], [a-f] i [0-9]

`SELECT * FROM Zemlje where Ime regexp '[:blank:]'`

# OSOBI NE REGEXP

## Escape Sekvenca

---

- Želimo da pretražimo oznaku CD modela [7543], problem je što se uglaste zagrade koriste i za podudaranje karaktera koji se nalaze u uglastim zagradama.
- Escape sekvencu koristimo ukoliko želimo da mečujemo same uglaste zagrade a ne njenu REGEXP sintaksu.

```
SELECT *  
FROM Proizvod  
WHERE Ime_proivoda REGEXP 'CD-RW Model \[7543\]';
```



# OSOBI NE REGEXP

## Repetition Match

Koristi se kada želimo da mečujemo ponavljanje određene instance više puta

Patern	Opis
*	Nula ili više instanci prethodnog elementa
+	Jedna ili više instanci prethodnog elementa
{n}	n instanci prethodnog elementa
{n,}	minimum n instanci prethodnog elementa
{n1,n2}	Minimum n1 a maksimum n2 instance
?	Nula ili jedna instanca prethodnog elementa

```
SELECT *  
FROM product  
WHERE opis REGEXP 'Drives?';
```

kod	ime	opis
3	WildTech 250Gb 1700	SATA Disk Drive
20	MasterSlave Multi-pack	5 SATA Disk Drives

# OSOBI NE REGEXP

## Podudaranje na osnovu pozicije u tekstu

---

Patern	Opis
^	Početak teksta
\$	Kraj teksta
[:<:]	Početak reči
[:>:]	Kraj reči

```
SELECT ime  
FROM Proizvod  
WHERE ime REGEXP '[:<:]One[:>:]';
```

# OSOBI NE REGEXP

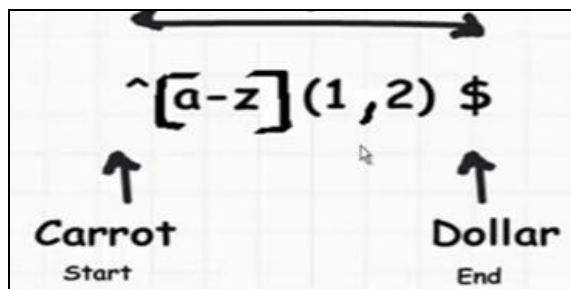
Za mečovanje kopleksnih iskaza tj paterna potrebno je razumeti tri simbola koja se koriste kod REGEXP (zagrada, \$(meču je kraj stringa) i ^(meču je početak stringa))

## Postoje tri vrste zagrada

[] -> definišu karaktere koje treba mečovati

{ } -> definišu koliko karaktera treba mečovati

() -> koriste se za grupisanje



Šta meču je ovaj regexp

```
^[a-zA-Z0-9]{1,10}@[a-zA-Z]{1,10}.(com|org)$
```

# PRIMERI

---

1. Mečuj karakter između a-g bilo gde u iskazu
  - `[a-g]`
2. Mečuj karaktere između a-g sa dužinom 3. Potrebno je definisati dužinu validacije
  - `[a-g]{3}`
  - `^[a-g]{3}$` -> dozvoljavamo samo tri bilo koja slova iz opsega a-g u iskazu
3. Mečuj karaktere između a-g sa minimum 1 karakterom a maximum 3
  - `^[a-g]{1,3}$`
4. Mečovanje 8 digita fiksne dužine (npr 12345678)
  - `^[0-9]{8}$`
5. Mečovanje minimum 3 a maksimalno 7 digita
  - `^[0-9]{3,7}$`
6. Mečovanje računa gde su prva tri karaktera tekstualna a naredna 8 su brojevi
  - `^[a-zA-Z]{3}[0-9]{8}$`
7. Prosta validacija web adrese
  - `^[w]{3}\.[A-Za-z0-9]{1,12}\.(com|edu|rs)`

# PRIMERI

---

1. Napisati REGEXP koji hvata mala slova, cifre, donju i srednju crtu. Dozvoljen broj karaktera je između 3 i 16.
  - `^[a-z0-9_-]{3,16}$`
2. Prikaz hekza vrednosti sa '#' opcionim karakterom i sa mogućnošću da se prikažu 6 ili 3 hekza vrednosti.
  - `^#?([a-f0-9]{6}|[a-f0-9]{3})$`
3. Traženje email adrese
  - `^([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.-]{2,6})$`
4. Traženje web adrese
  - `^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.-]{2,6})([\/\w \.-]*)*\/?$`