MASARYKOVA UNIVERZITA
INSTITUT BIOSTATISTIKY A ANALÝZ

Databázové systémy a SQL

Lekce 7 – Vyhledávání v textu

Daniel Klimeš

Vyhledávání v textu

- Operátor LIKE
  - zástupné znaky
    - _ = 1 libovolný znak
    - % = 0 nebo n libovolných znaků
    - ESCAPE '\'

  - Příklad:
  - Pracoviště Ústí
  - SELECT * FROM sites WHERE site LIKE ‘%Ústí%’
  - Text obsahující znak procento
  - SELECT * FROM eav_string WHERE values LIKE '%\%%' ESCAPE '\';
  - Jednoznakové texty
  - SELECT * FROM eav_string WHERE values LIKE ‘_’;
  - Text podobný datumu kdekoliv v textu
  - SELECT * FROM eav_string WHERE values LIKE ‘%__.__.____%’;

Regulární výrazy

Regulární výraz = šablona/vzor (pattern)

- Pochází z programovacích jazyků pro zpracování textu
- Nejen pro databáze

Skládá se:

- z hledaných znaků, textu
- zástupných znaků
- kvantifikátorů
- modifikátory
- operátory

PostgreSQL

https://www.postgresql.org/docs/current/static/functions-matching.html

Operator	Description	Example
~	Matches regular expression, case sensitive	'thomas' ~ '.*thomas.*'
~*	Matches regular expression, case insensitive	'thomas' ~* '.*Thomas.*'
!~	Does not match regular expression, case sensitive	'thomas' !~ '.*Thomas.*'
!~*	Does not match regular expression, case insensitive	'thomas' !~* '.*vadim.*'

- WHERE sloupec ~ ‘vyraz’

- SUBSTRING (string FROM pattern)
- REGEXP_REPLACE(string text, pattern text, replacementtext [, flags text])

Zástupné znaky

Znak	Význam
. (tečka)	Jakýkoliv znak
^	Začátek řetězce
$	Konec řetězce
\d	Číslice
\D	Vše kromě číslice
\w	Písmeno, číslice, podtržítko
\W	Doplněk k \w
\s	Bílý znak – mezera, tabulátor
\S	Doplněk k \s

Hledání datumu:

```
SELECT values FROM eav_string
WHERE values ~ '\d\d\.\d\d\.\d\d\d\d'
```

Kvantifikátory, modifikátory

Znak	Význam
*	0 – n opakování
+	1 – n opakování
?	0 nebo 1 opakování
{m}	Přesně m opakování
{m,}	m nebo více opakování
{m,n}	Minimálně m, maximálně n opakování

```
SELECT values FROM eav_string
WHERE values ~ '\d{1,2}\.\d{1,2}\.\d{4}'
```

Greedy x non-greedy

```
SELECT values,
SUBSTRING(values from  '\d.*\d') greedy,
SUBSTRING(values, '\d.*?\d') non_greedy
FROM eav_string WHERE values ~ '\d.*\d'
```

Znak	Význam
*	0 – n opakování
+	1 – n opakování
?	0 nebo 1 opakování
{m,}	m nebo více opakování
{m,n}	Minimálně m, maximálně n opakování

Operátory

Znak	Význam
[abc]	Jeden z uvedených znaků (a nebo b nebo c)
[^abc]	Libovolný znak kromě uvedených (vše kromě a b c)
(abc)	Uzavření skupiny znaků-blok
|	nebo
\1	Odkaz na první blok
\	Ruší speciální význam znaku např.: „\.“ = tečka

```
SELECT values FROM eav_string
WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}'
```

Dvě stejné číslice za sebou (11, 22, 33,…)

```
SELECT values FROM eav_string
WHERE values ~ '(\d)\1'
```

Extrakce řetězce

Extrakce subřetězce:
SUBSTRING (string FROM pattern)

```
SELECT SUBSTRING (values from '[0123]?\d\.[01]?\d\.\d{4}'), values
FROM eav_string
WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}‘
--pouze první výskyt
```

```
SELECT REGEXP_MATCHES (values, '[0123]?\d\.[01]?\d\.\d{4}', 'g'),  values
FROM eav_string
WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}‘
-- pro každý výskyt nový řádek
```

```
SELECT REGEXP_MATCHES (values,
'([0123]?\d\.[01]?\d\.\d{4}).*?([0123]?\d\.[01]?\d\.\d{4})'),  values
FROM eav_string
--WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}’
--dva výskyty => pole (array)
```

Pole (array)

- Na položky se odkazujeme indexem v hranatých závorkách
- Index od 1

```
SELECT datumy, datumy[1] prvni_datum, datumy[2]  druhe_datum FROM (
SELECT REGEXP_MATCHES (values,
'([0123]?\d\.[01]?\d\.\d{4}).*?([0123]?\d\.[01]?\d\.\d{4})') datumy,  values
FROM eav_string
) a
```

Extrakce a konverze

Konverze na datum:

```
SELECT TO_DATE(SUBSTRING (values from '[0123]?\d\.[01]?\d\.\d{4}'),
'dd.mm.yyyy'), values
FROM eav_string
WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}'
```

Pokus o konverzi může selhat, pokud nejde o platné datum

```
SELECT datum, age(datum) FROM (
SELECT to_date(SUBSTRING (values FROM '[0123]?\d\.[01]?\d\.\d{4}'),
'dd.mm.yyyy') datum, values
FROM eav_string
WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}'
and is_date(SUBSTRING (values FROM '[0123]?\d\.[01]?\d\.\d{4}')) = true
) x
```

IS DATE

```
create or replace function is_date(s varchar) returns boolean as
$$
begin
  perform s::date;
  return true;
exception when others then
  return false;
end;
$$ language plpgsql;
```

REGEXP_REPLACE

Nahrazení nalezeného vzoru za jiný text:
REGEXP_REPLACE(sloupec, pattern, novy_text, modifikator)

modifikator– ‘g’ = všechny výskyty

```
SELECT REGEXP_REPLACE(values, '([0123]?\d)\.([01]?\d)\.(\d{4})', '\3-\2-\1') datum,
values
FROM eav_string
WHERE values ~ '[0123]?\d\.[01]?\d\.\d{4}'
```

Rozklad pomocí reg. výrazu

```
SELECT foo FROM REGEXP_SPLIT_TO_TABLE('the quick brown fox
jumps over the lazy dog', '\s+') AS foo;
```

Odkazy

- http://www.regularnivyrazy.info/
- http://www.regexlib.com
- Jan Goyvaerts: **Regulární výrazy**

Cvičení

- Obsahuje tabulka PSČ?
- Obsahuje tabulka Rodná čísla?

```
SELECT values FROM eav_string WHERE
values ~ '^[1-7]\d{2}\s?\d{2}\s*$'
```

```
SELECT values FROM eav_string
WHERE values ~ '^\d{6}/\d{4}'
```