Reguláris kifejezések Oracle-ben

Jánosi-Rancz Katalin Tünde

Reguláris kifejezések - *Regular Expression - regexp*

- szövegmintákat leíró nyelv, amellyel karakterláncokban található mintákat írunk le
- standardizált
- minta illesztéssel nagyon gyors keresés, szűrés megvalósítására szolgál

- **Mintaillesztés:** ha ellenőrizni szeretnénk, hogy egy sztring megfelel-e bizonyos szabályoknak vagy megtalálhatók-e benne bizonyos szabályoknak eleget tévő rész sztringek

- pl. Valós email cím

Operátor	Leírás
*	Nulla vagy több előfordulás
+	Egy vagy több előfordulás
?	Nulla vagy egy előfordulás
|	Választás operator alternatív minták megadásához
^	Illeszkedés a minta elején
$	Illeszkedés a minta végén
.	Bármilyen karaketer a támogatott karakterkészletből, kivéve a NULL
[…]	Illeszkedik a listában szereplő valamelyik karakterrel.
[^ …]	**[^]** karakternek speciális jelentése van, **a nem illeszkedést** jelenti. Illeszkedik a listában **nem** szereplő karakterekkel.
( )	Csoportképző operátor, amelyben szereplő kifejezésre később \n alakban hivatkozhatunk, ahol n a teljes reguláris kifejezésben részkifejezés sorszáma.
{m}	Pontosan m-szeres ismétlődés.
{m,}	Legalább m-szeres ismétlődés.
{m,n}	Legalább m, de legfeljebb n-szeres ismétlődés.
\	A következő metakaraktert szószerint értelmezi, pl. \.a.
\n	Többszöri előfordulás, Hivatkozás az n-edik ’(’ ’)’ zárójelek közé zárt kifejezésre Az n 1 és 9 között lehet.
[..] [f]	Karakterek egy csoportja vagy egy több karakterből álló szimbólum pl.: ’ny’, ’ly’
[: :] [g]	Karakterosztály megadása (pl.: [:alpha]). A karakterosztályon belül bármely karakterre illeszkedik
[==] [h]	Ekvivalens osztályok meghatározása.

Karakterosztály	Jelentés
[:alnum:]	Minden alfanumerikus karakter
[:alpha:]	Minden betű karakter
[:blank:]	Minden fehérelválasztó.
[:cntrl:]	Minden kontroll karakter (nem nyomtatható)
[:digit:]	Minden numerikus szám
[:graph:]	[:punct:],[:upper:],[:lower:], és [:digit:] karakterek.
[:lower:]	Minden kisbetű.
[:print:]	Nyomtatható karakterek.
[:punct:]	Írásjelek.
[:space:]	Minden helyköz karakter (nem nyomtatható).
[:upper:]	Minden nagybetű.
[:xdigit:]	Minden érvényes 16-os számrendszerbeli szám.

\d	Matches a digit character.
\D	Matches a nondigit character.
\w	Matches a word character.
\W	Matches a nonword character.
\s	Matches a whitespace character.
\S	matches a non-whitespace character.
\A	Matches the beginning of a string or matches at the end of a string before a newline character.
\Z	Matches at the end of a string.
*?	Matches the preceding pattern zero or more occurrences.
+?	Matches the preceding pattern one or more occurrences.
??	Matches the preceding pattern zero or one occurrence.
{n}?	Matches the preceding pattern n times.
{n,}?	Matches the preceding pattern at least n times.
{n,m}?	Matches the preceding pattern at least n times, but not more than m times.

Példák

- abc -> abc
- a(b|c)d -> abd és acd
- a.c -> abc, adc, a1c, a&c, DE NEM abb
- a+ -> a, aa, aaa, DE NEM bbb
- ab?c -> abc, ac, DE NEM adc, abbc
- ab*c -> ac, abc, abbc, abbbbc, DE NEM adc
- a{5} ->aaaaa, DE NEM aaaa
- a{3,} ->aaa, aaaaa, DE NEM aa
- a{3,5} ->aaa, aaaa, aaaaa, DE NEM aa

Példák

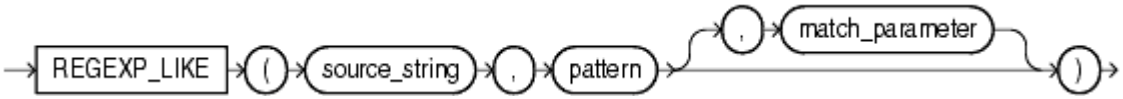
- [abc] -> at, bet, cot, DE NEM def
- [^abc] ->abcdef, ghi, DE NEM abc
- [^a-i] -> hijk, lmn, DE NEM abcdefghi
- **Subexpression**
- (abc)?def -> abcdefghi, defghi, DE NEM ghi
- **Backreference**
- (abc|def)\1 ->abcabc, defdef, de NEM abcdef, abc
- ^(.*)\1$ -> 2 egyforma karakter egymás mellett

Példák

- \+ -> abc+def, DE NEM abcdef
- ^def -> defghi, DE NEM abcdef
- def$ -> abcdef, DE NEM defghi
- [[:upper:]]+ -> abcDEFghi, DE NEM abcdefghi

REGEXP_LIKE

- Karakterlánc és minta összehasonlítására alkalmas
- Hasonló a LIKE - hoz, de sokkal bonyolultabb reguláris kifejezéseket adhatunk meg



- Regexp_like(karakterlánc, minta [, találati paraméterek])

- 1. Steven/ Stephen.

```
SELECT name FROM emp WHERE REGEXP_LIKE (last_name, '^Ste(v|ph)en$');
```

- 2. Akinek 2 magánhangzó van a nevében, ‘i’ = case-insensitive

```
SELECT name FROM emp WHERE REGEXP_LIKE (last_name, '([aeiou])\1', 'i');
```

- 3. Három hosszú nevek

```
SELECT name FROM emp
WHERE regexp_like (name, '^[a-z]{3}$' );.
```

- 4. Második betű a,c vagy f

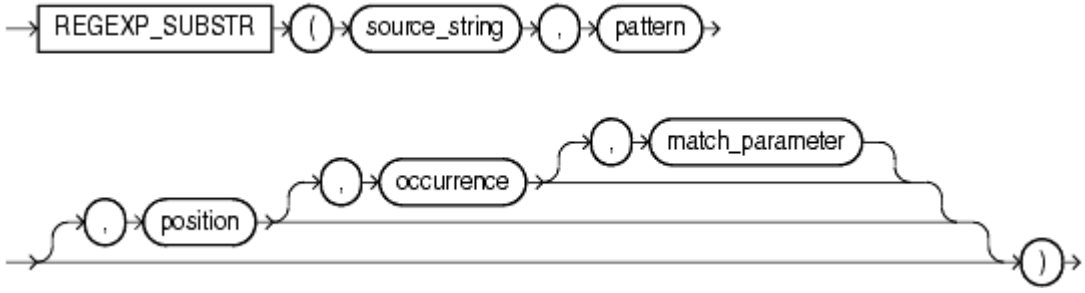
```
SELECT name FROM emp
where  REGEXP_LIKE( name, '^.[acf]' ) ;
```

- 5. Valós email cím

```
SELECT email FROM emp
WHERE REGEXP_LIKE (email, '[A-Z0-9._%-]+@[A-Z0-9._%-]+\.[A-Z]{2,4}' );
```

REGEXP_SUBSTR

A SUBSTR műveletet egészíti ki reguláris kifejezésekkel. Magával a rész-karakterlánccal tér vissza.



Regexp_substr(karakterlánc, minta [, kezdő pozíció [, keresett előfordulás [, találati paraméterek]]])

- 1. A 'call', 'caller', 'called, vagy 'calling'

```
SELECT REGEXP_SUBSTR('You have called me too often, she said.,
,'call((ing)|(er)|(ed))') FROM dual;
        --Eredmény: called
```

- 2. A 3-ik mezőt : elválasztóval

```
SELECT REGEXP_SUBSTR('system/pwd@orabase:1521:sidval', '[^:]+', 1, 3)
FROM DUAL; --Eredmény: Sidval
```

- 3. Az első szám

```
SELECT REGEXP_SUBSTR ('2, 5, and 10 are numbers in this example', '\d')
FROM dual;   --Eredmény: 2
```

- 4. Az első kétjegyű szám

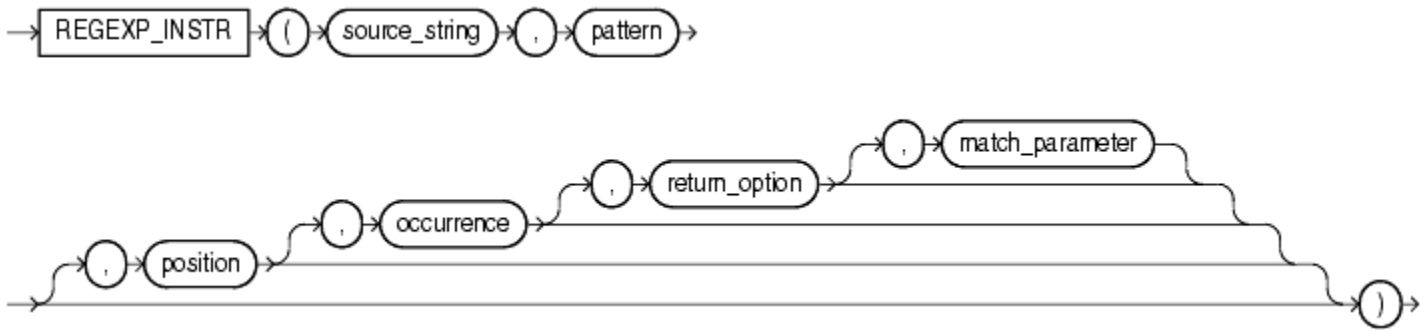
```
SELECT REGEXP_SUBSTR ('2, 5, and 10 are numbers in this example', '(\d)(\d)')
FROM dual; --Eredmény: 10.
```

- 5. Második magánhagnzó előfordulás

```
SELECT REGEXP_SUBSTR ('TechOnTheNet', 'a|e|i|o|u', 1, 2, 'i') FROM dual;
 -- Eredmény: 'O'
```

REGEXP_INSTR

- nem a talált karakterlánccal, hanem annak kezdőpozíciójával tér vissza, egy integerrel



- Regexp_instr(karakterlánc, minta [, kezdő pozíció [, keresett előfordulás [, visszatérési opció[, találati paraméterek]]]])

- 1. Megkeresi a 6-ik előfordulását egy non blank karakterenek (válasz 37)
  - SELECT REGEXP_INSTR('500 Oracle Parkway, Redwood Shores, CA', '[^]+', 1, 6) FROM DUAL;
- 2. Megkeresi az első karakter pozicióját a címben

  SELECT REGEXP_INSTR(street_address,'[[:alpha:]]') FROM locations;
- 3. s,r,p-vel kezdődő 7 hosszú szavak 3 poziciótól kezdve, 2 előfordulás utáni

  SELECT REGEXP_INSTR('500 Oracle Parkway, Redwood Shores, CA', '[s|r|p][[:alpha:]]{6}', 3, 2, 1, 'i') "REGEXP_INSTR" FROM DUAL;

  --Eredmény: 28
- 4. az ow első előfordulása a 16-ik poziciótól kezdve

  SELECT REGEXP_INSTR ('The example shows how to use the REGEXP_INSTR function', 'ow', 16, 1, 0, 'i') FROM dual;
- 5. Valós email cím

  SELECT REGEXP_INSTR(,valami sdsds@cx sds@sdf.ro', '\w+@\w+(\.\w+)+') FROM DUAL;

```
SELECT
  REGEXP_INSTR
① ('0123456789',       -- source char or search value
② '(123)(4(56)(78))', -- regular expression patterns
③ 1,                   -- position to start searching
④ 1,                   -- occurrence
⑤ 0,                   -- return option
⑥ 'i',                 -- match option (case insensitive)
⑦ 1)              -- sub-expression on which to search
   "Position"
FROM dual;
```

	Position
1	2

- You may need to find a specific subpattern that identifies a protein needed for immunity in the mouse DNA.

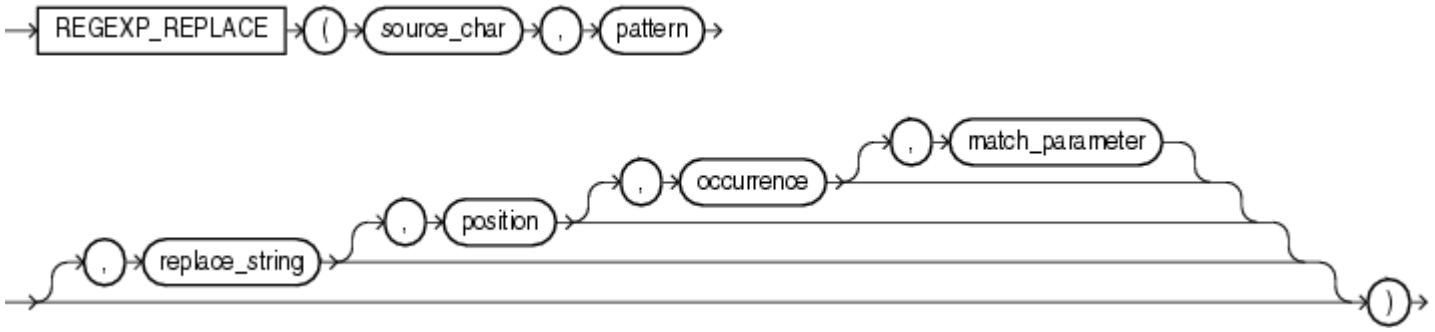
```
SELECT
  REGEXP_INSTR('ccacctttccctccactcctcacgttctcacctgtaaagcgtccctc
  cctcatccccatgccccttaccctgcagggtagagtaggctagaaaccagagagctccaa
  gctccatctgtggagaggtgccatccttgggctgcagagagaggagaatttgccccaaagc
  tgcctgcagagcttcaccacccttagtctcacaaagccttgagttcatagcatttcttgag
  ttttcaccctgcccagcaggacactgcagcacccaaagggcttcccaggagtagggttgcc
  ctcaagaggctcttgggtctgatggccacatcctggaattgttttcaagttgatggtcaca
  gccctgaggcatgtaggggcgtggggatgcgctctgctctgctctcctctcctgaacccct
  gaaccctctggctaccccagagcacttagagccag',
    '(gtc(tcac)(aaag))', 1, 1, 0, 'i', 1) "Position"
FROM dual;
```

```
Position
--------------------
195

1 rows selected
```

REGEXP_REPLACE

- Egy string-ben kereshetünk reguláris mintát és cserélhetjük ki, amire szeretnénk. A függvény minden behelyettesített karakterláncot visszaad.



- Regexp_replace(karakterlánc, minta [, helyettesítő szöveg [, induló pozíció [, keresett előfordulás [, találati paraméterek]]]])

- 1. A . ot – á alakítja

```
SELECT REGEXP_REPLACE(phone_number, '\.','-') AS phone FROM emp;
```

- 2. Keresi a xxx.xxx.xxxx és átalakítja (xxx) xxx-xxxx

```
SELECT REGEXP_REPLACE(testcol, '([[:digit:]]{3})\.([[:digit:]]{3})\.([[:digit:]]{4})',
'(\1) \2-\3') FROM test;
```

- 3. Betesz egy szünetet ha kisbetűt nagy követi

```
SELECT REGEXP_REPLACE('felhívott KisPista', '([[:lower:]])([[:upper:]])', '\1 \2')
FROM DUAL;
```

- Eredmény: 'felhívott Kis Pista'

- 4. A kétjegyű számok helyett #

```
SELECT REGEXP_REPLACE ('2, 5, and 10 are numbers in this example', '(\d)(\d)', '#')
FROM dual;
```

Eredmény: '2, 5, and # are numbers in this example'

REGEXP_COUNT

- A mintának megfelelő előfordulások számát adja vissza
- REGEXP_COUNT(karakterlánc, minta [, induló pozíció [, találati paraméterek]])
- 1. Az a,c vagy f, karakterek száma

```
select  REGEXP_COUNT( name, '[acf]' ) from emp
```

- 2. Hány t van

```
SELECT REGEXP_COUNT (last_name, 't', 1, 'i') AS total FROM emp;
```

- 3. Hány szóból áll

```
select REGEXP_COUNT( szöveg, '[^ ]+' ) from texts;
```

REGEXP_COUNT

```
SELECT REGEXP_COUNT(
  'ccacctttccctccactcctcacgttctcacctgtaaagcgtcctccctcatc
  cccatgccccctttaccctgcagggtagagtaggctagaaaccagagagctccaag
  ctccatctgtggagaggtgccatccttgggctgcagagagaggagaatttgcccc
  aaagctgcctgcagagcttcaccacccttagtctcacaaagccttgagttcatag
  catttcttgagttttcaccctgcccagcaggacactgcagcacccaaagggcttc
  ccaggagtagggttgccctcaagaggctcttgggtctgatggccacatcctggaa
  ttgttttcaagttgatggtcacagccctgaggcatgtaggggcgtggggatgcgc
  tctgctctgctctcctctcctgaacccctgaaccctctggctaccccagagcact
  tagagccag','gtc' ) AS Count
FROM dual;
```



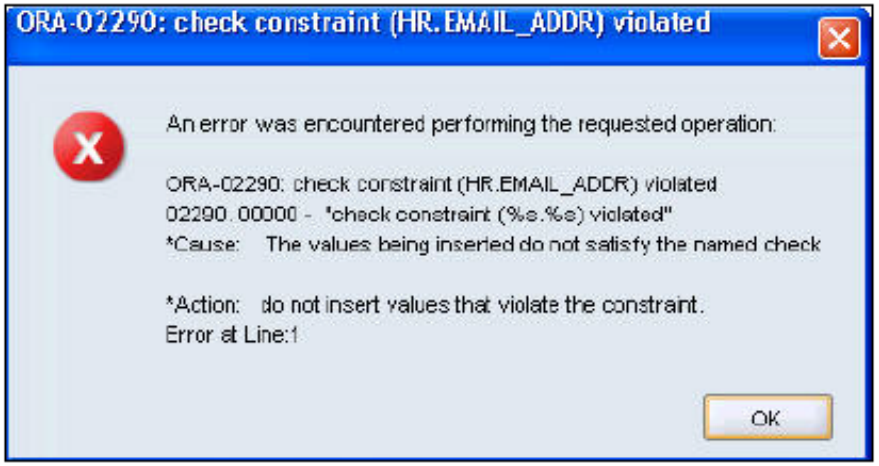
```
COUNT
----------
4

1 rows selected
```

Regular Expressions and Check Constraints: Examples

```
ALTER TABLE emp8
  ADD CONSTRAINT email_addr
  CHECK(REGEXP_LIKE(email,'@')) NOVALIDATE;
```

```
INSERT INTO emp8 VALUES
  (500,'Christian','Patel','ChrisP2creme.com',
   1234567890,'12-Jan-2004','HR_REP',2000,null,102,40);
```



Emlékeztető - Constraints

1. ENABLE ensures that all incoming data conforms to the constraint
2. DISABLE allows incoming data, regardless of whether it conforms to the constraint
3. VALIDATE ensures that existing data conforms to the constraint
4. NOVALIDATE means that some existing data may not conform to the constraint

- Online gyakorlási lehetőseg!!!
  - regexr.com

- https://docs.oracle.com/cd/B19306_01/appdev.102/b14251/adfns_regexp.htm