

Oracle

正则表达式函数介绍

数据库吧
database8.com

<http://www.database8.com>

2011-3-1

Oracle 中的支持正则表达式的函数主要有下面四个：

- 1, REGEXP_LIKE : 与 LIKE 的功能相似
- 2, REGEXP_INSTR : 与 INSTR 的功能相似
- 3, REGEXP_SUBSTR : 与 SUBSTR 的功能相似
- 4, REGEXP_REPLACE : 与 REPLACE 的功能相似

它们在用法上与 Oracle SQL 函数 LIKE、INSTR、SUBSTR 和 REPLACE 用法相同，

但是它们使用 POSIX 正则表达式代替了老的百分号 (%) 和通配符 (_) 字符。

POSIX 正则表达式由标准的元字符 (metacharacters) 所构成：

'^' 匹配输入字符串的开始位置，在方括号表达式中使用，此时它表示不接受该字符集合。

'\$' 匹配输入字符串的结尾位置。如果设置了 RegExp 对象的 Multiline 属性，则 \$ 也匹配 '\n' 或 '\r'。

'.' 匹配除换行符之外的任何单字符。

'?' 匹配前面的子表达式零次或一次。

'+' 匹配前面的子表达式一次或多次。

'*' 匹配前面的子表达式零次或多次。

'|' 指明两项之间的一个选择。例子 '^([a-z]+|[0-9]+)\$' 表示所有小写字母或数字组合成的字符串。

'()' 标记一个子表达式的开始和结束位置。

'[]' 标记一个中括号表达式。

'{m,n}' 一个精确地出现次数范围, $m \leq \text{出现次数} \leq n$, '{m}' 表示出现 m 次, '{m,}' 表示至少

出现 m 次。

\num 匹配 num, 其中 num 是一个正整数。对所获取的匹配的引用。

字符簇:

[[:alpha:]] 任何字母。

[[:digit:]] 任何数字。

[[:alnum:]] 任何字母和数字。

[[:space:]] 任何白字符。

[[:upper:]] 任何大写字母。

[[:lower:]] 任何小写字母。

[[:punct:]] 任何标点符号。

[[:xdigit:]] 任何 16 进制的数字, 相当于[0-9a-fA-F]。

各种操作符的运算优先级

\转义符

(), (?:), (?=), [] 圆括号和方括号

*, +, ?, {n}, {n,}, {n,m} 限定符

^, \$, anymetacharacter 位置和顺序

regexp like 函数

```
1. --创建表
2. create table fzq
3. (
4. id varchar(4),
5. value varchar(10)
6. );
7. --数据插入
8. insert into fzq values
9. ('1','1234560');
10. insert into fzq values
11. ('2','1234560');
12. insert into fzq values
13. ('3','1b3b560');
14. insert into fzq values
15. ('4','abc');
16. insert into fzq values
17. ('5','abcde');
18. insert into fzq values
19. ('6','ADREasx');
20. insert into fzq values
21. ('7','123 45');
22. insert into fzq values
23. ('8','adc de');
24. insert into fzq values
25. ('9','adc,.de');
26. insert into fzq values
27. ('10','1B');
28. insert into fzq values
29. ('10','abcbvbnb');
30. insert into fzq values
```

```

31. ('11','11114560');
32. insert into fzq values
33. ('11','11124560');
34.
35.
36. --查询 value 中以 1 开头 60 结束的记录并且长度是 7 位
37. select * from fzq where value like '1____60';
38. select * from fzq where regexp_like(value,'1....60');
39. --查询 value 中以 1 开头 60 结束的记录并且长度是 7 位并且全部是数字的记录。
40. --使用 like 就不是很好实现了。
41. select * from fzq where regexp_like(value,'1[0-9]{4}60');
42. -- 也可以这样实现，使用字符集。
43. select * from fzq where regexp_like(value,'1[[:digit:]]{4}60');
44. -- 查询 value 中不是纯数字的记录
45. select * from fzq where not regexp_like(value,'^[[:digit:]]+$');
46. -- 查询 value 中不包含任何数字的记录。
47. select * from fzq where regexp_like(value,'^[^[:digit:]]+$');
48. --查询以 12 或者 1b 开头的记录.不区分大小写。
49. select * from fzq where regexp_like(value,'^1[2b]','i');
50. --查询以 12 或者 1b 开头的记录.区分大小写。
51. select * from fzq where regexp_like(value,'^1[2B]');
52. -- 查询数据中包含空白的记录。
53. select * from fzq where regexp_like(value,'[[:space:]]');
54. --查询所有包含小写字母或者数字的记录。
55. select * from fzq where regexp_like(value,'^([a-z]+|[0-9]+)$');
56. --查询任何包含标点符号的记录。
57. select * from fzq where regexp_like(value,'[[:punct:]]');

```

regexp replace 函数

SQL> SELECT REGEXP_REPLACE('坐火车 去 西藏 玩 看 秀丽 山川','(){2,}','(',')'
所有连续两个以上的空白改成逗号 FROM DUAL;

所有连续两个以上的空白改成逗号

坐火车, 去, 西藏 玩, 看, 秀丽 山川

```
SQL> SELECT REGEXP_REPLACE('坐火车 去 西藏 玩 看 秀丽 山川','( )+',',',')
所有连续的空白改成逗号 FROM DUAL;
```

所有连续的空白改成逗号

坐火车, 去, 西藏, 玩, 看, 秀丽, 山川

或者

```
SQL> SELECT REGEXP_REPLACE('坐火车 去 西藏 玩 看 秀丽山川','\s+',',',')
所有连续的空白改成逗号 FROM DUAL;
```

所有连续的空白改成逗号

坐火车, 去, 西藏, 玩, 看, 秀丽, 山川

regexp instr 函数

regexp_instr 是 oracle 正则表达式中的一个函数。这个函数其实与 instr 函数其实相似。

regexp_instr 的原型如下:

```
regexp_instr(x, pattern[, start[, occurrence[, return_option[, match_option]]]])
```

这里每个参数分别含义如下:

x 待匹配的字符串

pattern 待匹配的模式

start 开始匹配的位置, 如果不指定默认为 1

occurrence 匹配的次数, 如果不指定, 默认为 1

return_option 指定返回值的类型, 如果该参数为 0, 则返回值为匹配位置的第一个字符, 如果该值为非 0 则返回匹配值的最后一个位置。

match_option 可以用这个参数来修改一些默认的配置设置。这个值与前面所说的 regexp_like 函数中的 match_option 参数的意义是一样的。

上例子:

```
Select INSTR('avdsdabasdad','a',1,1) FROM DUAL;
```

1

--从给定字段的第一个字符开始查找 a 字符串, 返回查找到第一个 a 的位置

```
Select INSTR('avdsdabasdad','a',1,2) FROM DUAL;
```

6

--从给定字段的第一个字符开始查找 a 字符串，返回查找到第二个 a 的位置

```
Select INSTR('avdsdabasdad','a',6,1) FROM DUAL;
```

6

--从给定字段的第六个字符开始查找 a 字符串，返回查找到第一个 a 的位置

regexp_substr 函数

5 个参数

第一个是输入的字符串

第二个是正则表达式

第三个是标识从第几个字符开始正则表达式匹配。（默认为 1）

第四个是标识第几个匹配组。（默认为 1）

第五个是取值范围：

i：大小写不敏感；

c：大小写敏感；

n：点号 . 不匹配换行符号；

m：多行模式；

x：扩展模式，忽略正则表达式中的空白字符。

全部测试数据

```
SQL> select * from test_reg_substr;
```

A

ABC123XYZ

ABC123XYZ456

<Name>Edward</Name>

检索中间的数字

```
SQL> SELECT
```

```
2   REGEXP_SUBSTR(a, '[0-9]+')
```

```
3 FROM
```

```
4   test_reg_substr
```

```
5 WHERE
```

```
6   REGEXP_LIKE(a, '[0-9]+');
```

```
REGEXP_SUBSTR(A, '[0-9]+')
```

```
-----  
123
```

```
123
```

检索中间的数字（从第一个字母开始匹配，找第 2 个匹配项目）

```
SQL> SELECT
```

```
2   NVL(REGEXP_SUBSTR(a, '[0-9]+', 1, 2), '-') AS a
```

```
3 FROM
```

```
4   test_reg_substr
```

```
5 WHERE
```

```
6   REGEXP_LIKE(a, '[0-9]+');
```


A

-

456

取得“字符集合”

```
SQL> SELECT
```

```
2   REGEXP_SUBSTR(a, '\w+')
```

```
3 FROM
```

```
4   test_reg_substr
```

```
5 WHERE
```

```
6   REGEXP_LIKE(a, '\w+');
```

```
REGEXP_SUBSTR(A, '\w+')
```

ABC123XYZ

ABC123XYZ456

Name

取得“字符集合”（从第一个字母开始匹配，找第 2 个匹配项目）

```
SQL> SELECT
```

```
2   NVL(REGEXP_SUBSTR(a, '\w+', 1, 2), '-') AS a
```

```
3 FROM
```

4 test_reg_substr

5 WHERE

6 REGEXP_LIKE(a, '\w+');

A

Oracle

正则表达式函数介绍

数据库吧
database8.com

<http://www.database8.com>