(https://ramkedem.com/en/)

Oracle REGEXP_LIKE Examples

Regular expressions are patterns used to match character combinations in strings. Oracle 10g introduced support for regular expressions using different functions. This post focuses on the Oracle **REGEXP_LIKE** function, and explains how to use it.

Description

the Oracle REGEXP_LIKE is used to perform a regular expression matching (rather than a simple pattern matching performed by LIKE).

syntax

```
REGEXP_LIKE ( string expression, pattern [, matching parameter ] )
```

- *string expression* – the string expression.
- *pattern (http://docs.oracle.com/cd/B12037_01/server.101/b10759/ap_posix001.htm#i690819)* – the regular expression matching pattern
- *match parameter (http://docs.oracle.com/cd/B12037_01/server.101/b10759/conditions018.htm)* – lets you to change the default matching behaviour of the Oracle REGEXP_LIKE function (for example, change the search from case sensitive to case insensitive).

Basic Oracle REGEXP_LIKE Examples

We'll start by creating a table called *Names*, based on its values, the following Oracle REGEXP_LIKE examples will perform different regular expression searches.

```
CREATE TABLE names
AS
SELECT last_name AS NAME
FROM hr.employees
ORDER BY salary ;
```

The following Oracle REGEXP_LIKE example would retrieve all of the names that contain the letter 'z'. This Oracle SELECT statement actually puts no lower or upper limit on the number of letters before or after the letter 'z' (any number of characters is allowed), but requires the word to contain the letter 'z'.

```
SELECT *
FROM names
WHERE regexp_like (name , 'z') ;


NAME
-------------------------
Lorentz
Gietz
Ozer
```

The next Oracle REGEXP_LIKE example would retrieve all of the names that contain the letter-sequence 'be'. Again, this Oracle SELECT statement actually puts no lower or upper limit on the number of letters before or after the letter-sequence 'be' (any number of characters is allowed), but requires the word to contain the letter-sequence 'be'.

```
SELECT *
FROM names
WHERE regexp_like (name , 'be') ;

NAME
--------------------------
Abel
Greenberg
```


using the pipe (|) operator

The Pipe operator (|) is used to specify alternative matches. In the next Oracle REGEXP_LIKE example we would use the pipe operator (|) in order to retrieve all of the names that contain the letter-sequence ‘be’ or ‘ae’. This Oracle SELECT statement actually puts no lower or upper limit on the number of letters before or after the letter-sequence ‘be’ or ‘ae'(any number of characters is allowed), but requires the word to contain these sequences.

```
SELECT *
FROM names
WHERE regexp_like (name , 'be|ae') ;

NAME
------------------------
Baer
Abel
Raphaely
Greenberg
```

By specifying the letter ‘c’ (as the third argument of the REGEXP_LIKE function) we can make a case sensitive search (the default in Oracle).

```
SELECT *
FROM names
WHERE regexp_like (name , 'be|ae' , 'c' ) ;

NAME
------------------------
Baer
Abel
Raphaely
Greenberg
```


And by specifying the letter ‘i’ (as the third argument of the REGEXP_LIKE function) we can make a case insensitive search.

```
SELECT *
FROM names
WHERE regexp_like (name , 'be|ae' , 'i' ) ;

NAME
------------------------
Bell
Bernstein
Baer
Abel
Raphaely
Greenberg
```

Using the Caret(^) operator

We can use the caret (^) operator to indicate a beginning-of-line character, in this REGEXP_LIKE example we would retrieve all names that start with the letter-sequence ‘be’ or ‘ba’ (case insensitive search)

```
SELECT *
FROM names
WHERE regexp_like (name , '^be|^ba' , 'i' ) ;

NAME
-------------------------
Baida
Bell
Banda
Bates
Bernstein
Baer
```


Using the Dollar ($) operator

We can use the dollar ($) operator to indicate an end-of-line character, in this REGEXP_LIKE example we would retrieve all names that end with the letter-sequence ‘es’ or ‘er’ (case insensitive search).

```
SELECT *
FROM names
WHERE regexp_like (name , 'es$|er$' , 'i' ) ;

NAME
-------------------------
Philtanker
Colmenares
Jones
Gates
Davies
Nayer
Stiles
Dellinger
Bates
Baer
```


Using Square Brackets

We can use the Square Brackets to specify a matching list that should match any one of the expressions represented in it. The next Oracle REGEXP_LIKE example would retrieve all names that contain the letters ‘j’ or ‘z’.

```
SELECT *
FROM names
WHERE regexp_like (name , '[jz]') ;

NAME
-------------------------
Rajs
Lorentz
Gietz
Ozer
Errazuriz
```


This REGEXP_LIKE example would retrieve all names that contain the letters ‘b’ or ‘z’ or ‘E’ (case sensitive search)

```
SELECT *
FROM names
WHERE regexp_like (name , '[bzE]') ;


NAME
-------------------------
Tobias
Cabrio
Everett
Lorentz
Pataballa
Ernst
Cambrault
Gietz
McEwen
Cambrault
```


Next, we'll modify our last query and make it a case insensitive search :

```
SELECT *
FROM names
WHERE regexp_like (name , '[bzE]' , 'i') ;

NAME
-------------------------
Philtanker
Zachary
Markle
Gee
Perkins
Colmenares
Patel
OConnell
Mikkilineni
Tobias
Seo
```


This Oracle REGEXP_LIKE example would retrieve all the names that contain the letters ‘a’, ‘b’, or ‘c’ :

```
SELECT *
FROM names
WHERE regexp_like (name , '[abc]') ;

NAME
-------------------------
Philtanker
Markle
Landry
Colmenares
Patel
Vargas
Sullivan
Marlow
Grant
Matos
```


And instead of specifying the letters ‘a’, ‘b’ and ‘c’ separately, we can specify a range :

```
SELECT *
FROM names
WHERE regexp_like (name , '[a-c]') ;

NAME
-------------------------
Philtanker
Markle
Landry
Colmenares
Patel
Vargas
Sullivan
Marlow
Grant
Matos
```


The next Oracle REGEP_LIKE example would retrieve all names that contain a letter in the range of ‘d’ and ‘g’, followed by the letter ‘a’.

```
SELECT *
FROM names
WHERE regexp_like (name , '[d-g]a') ;

NAME
-------------------------
Vargas
Baida
Fleaur
Banda
```


Using the Period (.) Operator

The period (.) operator matches any character except NULL, the next Oracle REGEXP_LIKE example would retrieve all names that contain a letter in the range of ‘b’ and ‘g’, followed by any character, followed by the letter ‘a’.

```
SELECT *
FROM names
WHERE regexp_like (name , '[b-g].[a]') ;

NAME
-------------------------
Colmenares
Tobias
McCain
Sarchand
Sewall
Cambrault
Sciarra
Cambrault
```


We can use the Period Operator to represent more than one character, the next Oracle REGEXP_LIKE example would retrieve all names that contain a letter in the range of ‘b’ and ‘g’, followed by any two characters, followed by the letter ‘a’.

```
SELECT *
FROM names
WHERE regexp_like (name , '[b-g]..[a]') ;

NAME
-------------------------
De Haan
Kochhar
```

Using the curly brackets

The curly brackets are used to specify an exact number of occurrences, for example display all names that contain double ‘o’ letters.

```
SELECT *
FROM names
WHERE regexp_like (name , '[o]{2}') ;

NAME
--------------------------
Khoo
Bloom
```

UpScale Analytics is one of the largest platforms in the world for learning SQL by doing, consisting **over 300 SQL exercises** at different levels (including solutions), by topics, across over 100 different datasets. More… (https://ramkedem.com/en/about/)