

Expresiones Regulares en Clientix



CONTENIDO

INTRODUCCIÓN	4
EJEMPLOS DE EXPRESIONES REGULARES	5
<i>Nombres de archivos simples</i>	<i>5</i>
<i>Nombres de archivos con número de secuencia</i>	<i>5</i>
<i>Nombres de archivos con fechas</i>	<i>6</i>
<i>Nombres de archivos con fechas y secuenciadores</i>	<i>7</i>
<i>Otros ejemplos</i>	<i>8</i>
PRUEBA DE EXPRESIONES REGULARES	9
METACARACTERES ADMITIDOS EN EXPRESIONES REGULARES	10
CARACTERES DE ESCAPE	11
CLASE DE CARACTERES	11
CUANTIFICADORES	12
CONSTRUYENDO EXPRESIONES REGULARES.....	13
COINCIDENCIA DE CADENAS BÁSICAS CON EXPRESIONES REGULARES	13
OPERACIONES DE EXPRESIÓN REGULARES EN SUBEXPRESIONES	13
OPERADOR DE EXPRESIONES REGULARES Y USO DE METACARACTERES.....	14
COINCIDIR CON CUALQUIER CARACTER - PUNTO.....	14
CERO O UNO - OPERADOR DE SIGNO DE INTERROGACIÓN	15
CERO O MÁS - ESTRELLA	15
INTERVALO - RECUENTO EXACTO	16
INTERVALO: AL MENOS CONTAR	16
INTERVALO - ENTRE EL RECUENTO	17
LISTA DE CARACTERES COINCIDENTES	17
LISTA DE CARACTERES NO COINCIDENTES	18
OPERADOR "O"	19
SUBEXPRESIÓN	19
BACKREFERENCE (RETRO-REFERENCIA)	19
CARACTER DE ESCAPE	20
PRINCIPIO DE ANCLA DE LÍNEA	21
ANCLA DE FINAL DE LÍNEA	21
CLASE DE CARÁCTER POSIX.....	21
SECUENCIA DE CLASIFICACIÓN DE POSIX	22
CLASE DE EQUIVALENCIA DE CARACTERES POSIX.....	23
SOPORTE DE EXPRESIONES REGULARES EN ORACLE DATABASE	25
FUNCIONES DE ORACLE DATABASE SQL PARA EXPRESIONES REGULARES	25
FUNCIONES DE EXPRESIONES REGULARES DE SQL	26
<i>REGEXP_LIKE</i>	<i>26</i>
<i>REGEXP_REPLACE.....</i>	<i>26</i>
<i>REGEXP_INSTR.....</i>	<i>26</i>
<i>REGEXP_SUBSTR.....</i>	<i>26</i>
REFERENCIAS.....	27

REVISIONES DE ESTE DOCUMENTO	28
------------------------------------	----

INTRODUCCIÓN

Este instructivo explica el uso de las expresiones regulares en el sistema Clientix.

Las expresiones regulares especifican patrones para buscar en cadenas de caracteres utilizando convenciones de sintaxis estandarizadas. Una expresión regular puede especificar patrones complejos de secuencias de caracteres.

Por ejemplo, la siguiente expresión regular:

`a(b|c)d`

Busca el patrón: 'a', seguido de 'b' o 'c', seguido de 'd'.
Esta expresión regular coincide con 'abd' y 'acd'.

Una expresión regular se especifica utilizando dos tipos de caracteres:

- Metacaracteres: operadores que especifican algoritmos para realizar la búsqueda.
- Literales: los Caracteres reales para buscar.

Para ver el resumen de los Metacaracteres admitidos en las expresiones regulares, vea la sección [Metacaracteres admitidos en expresiones regulares](#).

Para una referencia completa de construcción de expresiones regulares, vea la sección [Construyendo Expresiones Regulares](#).

Existen una serie de herramientas para verificar expresiones regulares. Vea la sección para [Prueba de Expresiones Regulares](#) más detalles.

A continuación presentamos algunos ejemplos para el reconocimiento de nombres de archivos y otros ejemplos útiles.

Ejemplos de expresiones regulares

Nombres de archivos simples

Expresión Regular:
File_Name123\.xls

Se espera que el nombre sea "File_Name123.xls". Esto es un literal simple sin variaciones. Cualquier otro nombre es descartado.

Nombres de archivos con número de secuencia

Expresión Regular:
^File_Name\[0-9]{3}\.txt\$

Secuencia	Explicación
^File_Name	Se espera que comience con el literal "File_Name"
\.	Seguido de un punto "."
[0-9]{3}	Seguido de 3 caracteres numéricos (ni mas, ni menos) para el número de secuencia
\.txt\$	Seguido del literal ".txt" con la extensión del archivo al final

Esta expresión encuentra:

File_Name.123.txt
File_Name.555.txt
File_Name.291.txt

No encuentra:

File_Name.12.txt
Porque la **secuencia** "12" debe tener justo 3 dígitos numéricos, y tiene sólo dos.

File_Name.1234.txt
Porque la **secuencia** "1234" debe tener justo 3 dígitos numéricos, y tiene cuatro.

File_Name.txt

Porque le falta un punto + secuencia de 4 dígitos numéricos después de "File_Name".

Nombres de archivos con fechas

Expresión Regular:

`^File_Name345\[0-9]{2,4}\-[0-9]{1,2}\-[0-9]{1,2}\.txt$`

Secuencia	Explicación
<code>^File_Name345</code>	Se espera que comience con el literal "File_Name345"
<code>\-</code>	Seguido de un guión "-"
<code>[0-9]{2,4}</code>	Seguido de entre 2 y 4 caracteres numéricos para el año
<code>\-</code>	Seguido de un guión "-"
<code>[0-9]{1,2}</code>	Seguido de entre 1 y 2 caracteres numéricos para el mes
<code>\-</code>	Seguido de un guión "-"
<code>[0-9]{1,2}</code>	Seguido de entre 1 y 2 caracteres numéricos para el día
<code>\.</code>	Seguido de un punto "."
<code>txt\$</code>	Seguido del literal "txt" con la extensión del archivo al final

Esta expresión encuentra:

File_Name345-1995-12-31.txt

File_Name345-1995-2-1.txt

File_Name345-95-12-1.txt

No encuentra:

File_Name345.1995-12-31.txt

Por el **punto "."** después de "File_Name345"; debe ser un guion "-"

File_Name345.1995-12-31.xlsx

Por la **extensión "xlsx"**; debe ser "txt"

File_Name345-1995-12-31.003.txt

Porque tiene un **numero ".003"** entre el día "-31." y la extensión ".txt" del archivo, y no debería.

File_Name.txt

Porque no tiene la secuencia de caracteres "345-" después de "File_Name" ni las secuencias de fecha antes de la **extensión ".txt"**

Nombres de archivos con fechas y secuenciadores

Expresión Regular:

`^File_Name678\.[0-9]{2,4}\-[0-9]{1,2}\-[0-9]{1,2}\.[0-9]{1,4}\.(xls|xlsx)$`

Secuencia	Explicación
<code>^File_Name678</code>	Se espera que comience con el literal "File_Name678 ",
<code>\.</code>	Seguido de un punto "."
<code>[0-9]{2,4}</code>	Seguido de entre 2 y 4 caracteres numéricos para el año,
<code>\-</code>	Seguido de un guión "-"
<code>[0-9]{1,2}</code>	Seguido de entre 1 y 2 caracteres numéricos para el mes
<code>\-</code>	Seguido de un guión "-"
<code>[0-9]{1,2}</code>	Seguido de entre 1 y 2 caracteres numéricos para el día
<code>\.</code>	Seguido de un punto "."
<code>[0-9]{1,4}</code>	Seguido de entre 1 y 4 caracteres numéricos de un número secuencial
<code>\.</code>	Seguido de un punto "."
<code>(xls xlsx)\$</code>	Seguido del literal "xls" o "xlsx" correspondiente la extensión del archivo al final.

Esta expresión encuentra:

File_Name678.2019-04-31.0001.xls

File_Name678.1995-2-1.002.xlsx

File_Name678.2018-12-1.03.xls

File_Name678.018-12-1.3.xls

No encuentra:

File_Name678-1995-12-31.003.xls

Por el **guion "-"** después de "File_Name678"; debe ser un punto "."

File_Name345.1995-12-31.0001.xlsx

Porque **comienza con "File_Name345"**; debe comenzar con "File_Name678".

File_Name678.1995-12-31.xls

Porque le falta el número de secuencia entre el día "-31" y la extensión ".xls".

File_Name678.txt

Porque le falta el punto y la fecha "AAAA-MM-DD" luego de "File_Name678", el punto y secuencia numérica de 4 dígitos "9999" luego de la fecha. Además la extensión no es ".xls" o ".xlsx".

Otros ejemplos

Descripción	Expresión	Comentario
Número de Seguro Social	\d{3}-\d{2}-\d{4}	Se espera que el número de la seguridad social tenga tres dígitos, seguidos por un -, seguido por 2 dígitos, seguidos por un - y seguido de nuevo por 4 dígitos.
Código Postal	\d{5}	Los códigos postales de EE.UU. están formados por 5 dígitos.
Código Zip + 4	\d{5}(-\d{4})	Un código postal basado en EE.UU. seguido por - y 4 dígitos.
Carácter Alfanumérico	[a-zA-Z0-9]	El carácter puede estar formado por todos los caracteres entre a a z, A a Z y 0 a 9.
Caracteres de palabra de dirección de correo electrónico	\w+([-+.] \w+)*@ \w+([-+.] \w+)*\ . \w+([-+.] \w+)*	Una palabra, "(" de manera opcional seguida por "-", "+" o "." y una palabra las veces que sea necesario ")", seguida por la "@", seguida por una palabra, "(" de manera opcional seguida por "-" o "." y una palabra las veces que sea necesario ")", seguida por un ".", una palabra, "(" de manera opcional seguida por un - o . y una palabra las veces que sea necesario ")". Nota: \w+ equivale a una palabra que contiene 1 o más
Número Telefónico de EE.UU.	((\(\d{3}\))?) (\d{3}-))?\d{3}-\d{4}	De manera opcional un "(", seguido por 3 dígitos, seguidos por ")" y un espacio o, de manera opcional, 3 dígitos seguidos por un "-", seguido por 3 dígitos, un "-" y 4 dígitos.

Limita el número de caracteres que un usuario puede escribir en un campo de texto	<code>^. {1,10}\$</code>	Esta expresión limitaría el número de caracteres que se pueden escribir en un campo de texto a 10.
Limita los caracteres escritos en un campo de entrada del usuario a solo caracteres de texto	<code>^[a-z]{1,10}\$</code> o <code>^[A-Z]{1,10}\$</code> o <code>^[A-Za-z]{1,10}\$</code>	Estas expresiones limitan los caracteres escritos en un campo de entrada del usuario a minúsculas, mayúsculas o una combinación de letras mayúsculas y minúsculas.
Limita los caracteres escritos en un campo de entrada del usuario a dígitos (es decir, que no sean letras)	<code>^[0-9]{1,10}\$</code>	Esta expresión limita los caracteres escritos en un campo de entrada del usuario a números.
Política de contraseñas	<code>^[a-zA-Z]\w{4,15}\$</code>	La contraseña contiene solamente caracteres alfanuméricos, la longitud mínima es 4 y la máxima es 15.

Prueba de Expresiones Regulares

Es posible verificar expresiones regulares con nombres de archivos y otras cadenas de caracteres en esta dirección: <https://www.regextester.com/100077>

Otras direcciones para realizar verificaciones:

<https://www.debuggex.com/>
<https://regex101.com/>

Metacaracteres admitidos en expresiones regulares

La siguiente Tabla enumera los metacaracteres admitidos para su uso en expresiones regulares pasadas a funciones de expresiones regulares de SQL. Los detalles sobre el comportamiento de coincidencia de estos metacaracteres se encuentran en "[Construir expresiones regulares](#)".

Sintaxis de Meta-caracter	Nombre Operador	Descripción
.	Cualquier Carácter - Punto	Coincide con cualquier Carácter
+	Uno o más - Más cuantificador	Coincide con una o más apariciones de la subexpresión anterior
?	Cero o uno - Cuantificador de interrogación	Coincide con cero o una ocurrencia de la subexpresión anterior
*	Cero o más - Cuantificador de estrellas	Coincide con cero o más apariciones de la subexpresión anterior
{m}	Intervalo - Recuento exacto	Coincide exactamente con las ocurrencias de la subexpresión anterior
{m,}	Intervalo: al menos contar	Coincide con al menos m apariciones de la subexpresión anterior
{m,n}	Intervalo - Entre el recuento	Coincide al menos con m, pero no más de n ocurrencias de la subexpresión anterior
[...]	Lista de Caracters coincidentes	Coincide con cualquier Carácter de la lista ...
[^ ...]	Lista de caracteres no coincidentes	Coincide con cualquier Carácter que no esté en la lista ...
	O	'a b' coincide con el carácter 'a' o 'b'.
(...)	Subexpresión o agrupamiento	Tratar la expresión ... como una unidad. La subexpresión puede ser una cadena de literales o una expresión compleja que contiene operadores.
\n	Backreference	Coincide con la enésima subexpresión anterior, donde n es un número entero de 1 a 9.
\	Carácter de escape	Trate el metacarácter subsiguiente en la expresión como un literal.
^	Principio de ancla de línea	Haga coincidir la expresión siguiente solo cuando ocurra al comienzo de una línea.
\$	Ancla de final de línea	Haga coincidir la expresión anterior solo cuando ocurra al final de una línea.
[:clase:]	Clase de carácter POSIX	Haga coincidir cualquier carácter que pertenezca a la clase de caracteres

		especificada. Se puede usar dentro de cualquier expresión de lista.
[.elemento.]	Secuencia de clasificación de POSIX	Especifica una secuencia de clasificación para usar en la expresión regular. El elemento que utiliza debe ser una secuencia de clasificación definida, en la configuración regional actual.
[=Caracter=]	Clase de equivalencia de caracteres POSIX	Coincide con los caracteres que tienen el mismo carácter base que el Caracter que especifique.

Carateres de Escape

Se pueden usar números de caracteres de escape para denotar ciertas teclas, por ejemplo `\t` significa una tabulación. La tabla siguiente contiene una lista de caracteres de escape.

Carácter de Escape	Descripción
Caracteres ordinarios	Los caracteres distintos a <code>. \$ ^ { [() * + ? \</code> se corresponden a sí mismos.
<code>\b</code> :	<code>\b</code> denota un límite de palabra (entre los caracteres <code>\w</code> y <code>\W</code>) excepto dentro de una clase de caracteres <code>[]</code> , donde <code>\b</code> equivale al carácter de retroceso. Ver la tabla abajo.
<code>\t</code>	Coincide con una tabulación.
<code>\r</code>	Coincide con un retorno de carro.
<code>\v</code>	Coincide con una tabulación vertical.
<code>\f</code>	Coincide con un avance de página.
<code>\n</code>	Coincide con una nueva línea.
<code>\e</code>	Coincide con un escape.
<code>\</code>	<code>\</code> Cuando va acompañado por un carácter que no sea reconocido como un carácter de escape, tiene coincidencia con ese carácter. Por ejemplo, <code>*</code> es lo mismo que <code>\x2A</code> .

Clase de Caracteres

Una clase de caracteres es un grupo de caracteres que encontrará una coincidencia si cualquiera de los caracteres incluido en el conjunto coincide. La tabla siguiente resume la sintaxis de correspondencia de caracteres.

Clase de Caracteres	Descripción
.	Los caracteres distintos a <code>. \$ ^ { [() * + ? \</code> se corresponden a sí mismos.
[aeiou]	Coincide con cualquier carácter individual incluido en el conjunto de caracteres especificado, en este caso "a", "e", "i", "o" o "u".
[0-9a-fA-F]	El uso de un guión (-) permite la especificación de rangos de caracteres contiguos, 0 a 9, "a" a "f" y "A" a "F".
\w	Coincide con cualquier carácter que no sea una palabra. \W equivale a <code>[^a-zA-Z_0-9]</code> .
\s	Tiene coincidencia con cualquier carácter blanco-espacio. \s equivale a <code>[\f\n\r\t\v]</code> .
\S	Tiene coincidencia con cualquier carácter que no sea blanco-espacio. \S equivale a <code>[^ \f\n\r\t\v]</code> .
\d	Tiene coincidencia con cualquier dígito decimal. \d equivale a <code>[0-9]</code> .
\D	Tiene coincidencia con cualquier carácter que no sea dígito. \D equivale a <code>[^0-9]</code> .

Cuantificadores

Los cuantificadores agregan datos opcionales de cantidad a una expresión regular. Una expresión con cuantificador se aplica al carácter, grupo o clase de caracteres que inmediatamente la precede. Las expresiones regulares .NET Framework soportan cuantificadores de coincidencia mínima ("flojos").

La tabla siguiente describe los metacaracteres que afectan a la cantidad de coincidencias.

Cuantificador	Descripción
*	Especifica cero o más coincidencias; por ejemplo, <code>\w*</code> o <code>(abc)*</code> . Equivale a <code>{0,}</code> .

+	Especifica una o más coincidencias; por ejemplo, \w+ o (abc)+. Equivale a {1,}.
?	Especifica cero o una coincidencia; por ejemplo, \w? o (abc)?. Equivale a {0,1}.
{n}	Especifica exactamente n coincidencias; por ejemplo, (pizza){2}.
{n,}	Especifica al menos n coincidencias; por ejemplo, (abc){2,}.
{n,m}	Especifica al menos n, pero no más de m, coincidencias.
*?	Especifica la primera coincidencia que consume el mínimo de repeticiones posibles (equivale a flojo *).
+?	Especifica el mínimo de repeticiones posibles, pero al menos una (equivale a flojo +).
??	Especifica cero repeticiones si es posible, o una (flojo ?).
{n}?	Equivale a {n} (flojo {n}).
{n,}?	Especifica el mínimo de repeticiones posibles, pero al menos n (flojo {n,}).
{n,m}?	Especifica el mínimo de repeticiones posibles entre n y m (flojo {n,m}).
	Coincidencia con cualquiera de los términos separados por el carácter (barra vertical); por ejemplo, gato perro tigre. La coincidencia exitosa más a la izquierda gana.

Construyendo Expresiones Regulares

Esta sección trata la construcción de expresiones regulares con algunos ejemplos.

Coincidencia de cadenas básicas con expresiones regulares

La coincidencia más simple que puede realizar con expresiones regulares es la coincidencia de cadena básica. Para este tipo de coincidencia, la expresión regular es una cadena de literales sin metacaracteres. Por ejemplo, para encontrar la secuencia 'abc', especifique la expresión regular:

abc

Operaciones de expresión regulares en subexpresiones

Como se mencionó anteriormente, las expresiones regulares se construyen usando metacaracteres y literales.

Metacaracteres que operan en un solo literal, como '+' y '?' también puede operar en una secuencia de literales o en una expresión completa. Para hacerlo, use el operador de agrupación para encerrar la secuencia o subexpresión.

Consulte "[Subexpresión](#)" para obtener más información sobre las agrupaciones.

Operador de expresiones regulares y uso de metacaracteres

Esta sección proporciona ejemplos de uso para cada metacarácter soportado o operador de expresión regular.

Coincidir con cualquier Caracter - Punto

El operador punto '.' coincide con cualquier carácter individual en el juego de caracteres actual. Por ejemplo, para encontrar la secuencia - 'a', seguido de cualquier carácter, seguido de 'c' - use la expresión:

a.c

Esta expresión coincide con todas las siguientes secuencias:

abc
adc
a1c
a&c

La expresión no coincide:

abb

Uno o más - Plus

El operador uno o más '+' coincide con una o más apariciones de la expresión anterior. Por ejemplo, para encontrar una o más apariciones del carácter 'a', usa la expresión regular:

a+

Esta expresión coincide con todo lo siguiente:

a
aa

aaa

La expresión no coincide:

bbb

Cero o uno - operador de signo de interrogación

El signo de interrogación coincide con cero o uno, y solo uno, con la aparición del carácter o subexpresión anterior. Puede pensar que este operador especifica una expresión que es opcional en el texto fuente.

Por ejemplo, para encontrar - 'a', opcionalmente seguido de 'b', seguido de 'c' - usa la siguiente expresión regular:

ab?c

Esta expresión coincide con

abc
ab

La expresión no coincide:

adc
abbc

Cero o más - Estrella

El operador cero o más '*', coincide con cero o más ocurrencias del carácter anterior o subexpresión. Por ejemplo, para encontrar - 'a', seguido de cero o más ocurrencias de 'b', seguido de 'c' - use la expresión regular:

ab*c

Esta expresión coincide con todas las siguientes secuencias:

ab
abc
abbc
abbbbc

La expresión no coincide:

adc

Intervalo - Recuento exacto

El operador de intervalo de recuento exacto se especifica con un solo dígito entre llaves. Utiliza este operador para buscar un número exacto de ocurrencias del carácter anterior o subexpresión.

Por ejemplo, para encontrar que 'a' aparece exactamente 5 veces, especifique la expresión regular:

a{5}

Esta expresión coincide con

aaaaa

La expresión no coincide:

aaaa

Intervalo: al menos contar

Utiliza el operador de intervalo al menos-conteo para buscar un número especificado de apariciones, o más, del carácter anterior o subexpresión. Por ejemplo, para encontrar que 'a' aparece al menos 3 veces, usa la expresión regular:

a{3,}

Esta expresión coincide con todo lo siguiente:

aaa

aaaaa

La expresión no coincide:

aa

Intervalo - Entre el recuento

Utiliza el operador de intervalo entre recuentos para buscar una cantidad de ocurrencias dentro de un rango especificado. Por ejemplo, para encontrar donde 'a' aparece al menos 3 veces y no más de 5 veces, usa la siguiente expresión regular:

`a{3,5}`

Esta expresión coincide con todas las siguientes secuencias:

aaa
aaaa
aaaaa

La expresión no coincide:

aa

Lista de Caracters coincidentes

Utiliza la lista de caracteres coincidentes para buscar una ocurrencia de cualquier carácter en una lista. Por ejemplo, para encontrar 'a', 'b' o 'c', use la siguiente expresión regular:

`[abc]`

Esta expresión coincide con el primer carácter en cada una de las siguientes cadenas:

a
bet
cot

La expresión no coincide:

def

Los siguientes operadores de expresiones regulares están permitidos dentro de la lista de caracteres, cualquier otro metacaracter incluido en una lista de caracteres pierde su significado especial (se tratan como literales):

- Operador de rango '-'
- Clase de caracteres POSIX [::]
- Secuencia de clasificación POSIX [. .]
- Clase de equivalencia de caracteres POSIX [= =]

Lista de caracteres no coincidentes

Use la lista de caracteres que no coincidan para especificar los caracteres que no desea que coincidan. Los caracteres que no están en la lista de caracteres que no coinciden se devuelven como una coincidencia. Por ejemplo, para excluir los caracteres 'a', 'b' y 'c' de sus resultados de búsqueda, use la siguiente expresión regular:

[^abc]

Esta expresión coincide con los caracteres 'd' y 'g' en las siguientes cadenas:

abcdef
ghi

La expresión no coincide:

abc

Al igual que con la lista de caracteres coincidentes, los siguientes operadores de expresiones regulares están permitidos dentro de la lista de caracteres no coincidentes (cualquier otro metacaracter incluido en una lista de caracteres se ignora):

- Operador de rango '-'
- Clase de caracteres POSIX [::]
- Secuencia de clasificación POSIX [. .]
- Clase de equivalencia de caracteres POSIX [= =]

Por ejemplo, la siguiente expresión regular excluye cualquier carácter entre 'a' e 'i' del resultado de la búsqueda:

[^a-i]

Esta expresión coincide con los caracteres 'j' y 'l' en las siguientes cadenas:

hijk
lmn

La expresión no coincide con los caracteres:

abcdefghi

Operador "O"

Use el operador '|' para especificar una expresión alternativa. Por ejemplo, para que coincida con 'a' o 'b', use la siguiente expresión regular:

a|b

Subexpresión

Puede usar el operador de subexpresión para agrupar caracteres que desea buscar como una cadena o para crear una expresión compleja. Por ejemplo, para encontrar la cadena opcional 'abc', seguida de 'def', use la siguiente expresión regular:

(abc)?def

Esta expresión coincide con las cadenas 'abcdef' y 'def' en las siguientes cadenas:

abcdefghi
defghi

La expresión no coincide con la cadena:

ghi

Backreference (retro-referencia)

La retro-referencia le permite buscar una expresión repetida. Usted especifica una referencia inversa con '\n', donde n es un número entero de 1 a 9 que indica la enésima subexpresión anterior en su expresión regular.

Por ejemplo, para encontrar una repetición de cualquiera de las cadenas 'abc' o 'def', use la siguiente expresión regular:

`(abc|def)\1`

Esta expresión coincide con las siguientes cadenas:

abcabc
defdef

La expresión no coincide con las siguientes cadenas:

abcdef
abc

La referencia inversa cuenta las subexpresiones de izquierda a derecha comenzando con el paréntesis de apertura de cada subexpresión anterior.

La retro-referencia le permite buscar una cadena repetida sin conocer la cadena real con anticipación. Por ejemplo, la expresión regular:

`^(.*)\1$`

Coincide con una línea que consta de dos apariencias adyacentes de la misma cadena.

Caracter de escape

Utilice el carácter de escape `\` para buscar un Caracter que normalmente se trata como un metacarácter. Por ejemplo, para buscar el carácter '+', use la siguiente expresión regular:

`\+`

Esta expresión coincide con el carácter más '+' en la siguiente cadena:

abc+def

La expresión no coincide con ningún carácter en la cadena:

abcdef

Principio de ancla de línea

Use el principio del anclaje de línea `^` para buscar una expresión que aparece solo al comienzo de una línea. Por ejemplo, para encontrar una ocurrencia de la def de cadena al comienzo de una línea, use la expresión:

`^def`

Esta expresión coincide con def en la cadena:

defghi

La expresión no coincide con def en la siguiente cadena:

abcdef

Ancla de final de línea

El metacaracter de anclaje de final de línea `$` le permite buscar una expresión que aparece solo al final de una línea. Por ejemplo, para encontrar una ocurrencia de def que ocurra al final de una línea, use la siguiente expresión:

`def$`

Esta expresión coincide con def en la cadena:

abcdef

La expresión no coincide con def en la siguiente cadena:

defghi

Clase de carácter POSIX

El operador de clase de caracteres POSIX le permite buscar una expresión dentro de una lista de caracteres que sea miembro de una clase de caracteres POSIX específica. Puede usar este operador para buscar caracteres con un formato específico, como mayúsculas, o puede buscar caracteres especiales, como dígitos o caracteres de puntuación. El conjunto completo de caracteres POSIX es aceptado.

Para usar este operador, especifique la expresión usando la sintaxis `[: clase:]` donde clase es el nombre de la clase de caracteres POSIX que se debe buscar. Por ejemplo, para buscar uno o más caracteres en mayúscula consecutivos, use la siguiente expresión regular:

```
[[:upper:]]+
```

Esta expresión coincide con 'DEF' en la cadena:

```
abcDEFghi
```

La expresión no devuelve una coincidencia para la siguiente cadena:

```
abcdefghi
```

Tenga en cuenta que la clase de caracteres debe aparecer dentro de una lista de caracteres, por lo que la clase de caracteres siempre se anida entre corchetes para la lista de caracteres en la expresión regular.

Ver también:

"Dominio de expresiones regulares" o "Mastering Regular Expressions" publicado por O'Reilly & Associates, Inc. para obtener más información sobre las clases de caracteres POSIX.

Secuencia de clasificación de POSIX

El operador del elemento de secuencia de clasificación POSIX `[. .]` te permite usar una secuencia de clasificación en tu expresión regular. El elemento que especifique debe ser una secuencia de clasificación definida en la configuración regional actual.

Este operador le permite usar una secuencia de clasificación de múltiples caracteres en su expresión regular donde solo se permitiría un carácter. Por ejemplo, puede usar este operador para asegurarse de que la secuencia de clasificación 'ch', cuando se define en una configuración regional como el español, se trate como un carácter en las operaciones que dependen del orden de los caracteres.

Para usar el operador de secuencia de clasificación, especifique `[.element.]` Donde element es la secuencia de clasificación que desea buscar. Puede usar cualquier

secuencia de clasificación que esté definida en la configuración regional actual, incluidos los elementos de un solo carácter y los elementos de varios caracteres.

Por ejemplo, para encontrar la secuencia de clasificación 'ch', use la siguiente expresión regular:

```
[.ch.]
```

Esta expresión coincide con la secuencia 'ch' en la siguiente cadena:

chabc

La expresión no coincide con la siguiente cadena:

cdefg

Puede usar el operador de secuencia de clasificación en cualquier expresión regular donde se necesite intercalación. Por ejemplo, para especificar el rango de 'a' a 'ch', puede usar la siguiente expresión:

```
[a-.[ch.]]
```

Clase de equivalencia de caracteres POSIX

Utilice el operador de clase de equivalencia de caracteres POSIX para buscar caracteres en la configuración regional actual que sean equivalentes. Por ejemplo, para encontrar el carácter español 'ñ' así como 'n'.

Para usar este operador, especifique [= character =], para encontrar todos los caracteres que son miembros de la misma clase de equivalencia de caracteres que el carácter especificado.

Por ejemplo, la siguiente expresión regular podría usarse para buscar caracteres equivalentes a 'n' en una configuración regional en español:

```
[=n=]
```

Esta expresión coincide con 'N' y 'ñ' en la siguiente cadena:

El Niño

Soporte de expresiones regulares en Oracle Database

Oracle Database implementa soporte de expresiones regulares que cumple con la especificación POSIX Extended Regular Expression (ERE).

El soporte de expresiones regulares se implementa con un conjunto de funciones de Oracle Database SQL que le permiten buscar y manipular datos de cadenas. Puede usar estas funciones en cualquier entorno en el que se utilice Oracle Database SQL. Consulte "[Funciones de Oracle Database SQL para expresiones regulares](#)" más adelante en este capítulo para obtener más información.

Oracle Database es compatible con un conjunto de metacaracteres comunes utilizados en expresiones regulares. El comportamiento de metacaracteres admitidos y características relacionadas se describe en "Metacaracteres admitidos en expresiones regulares".

Nota:

La interpretación de los metacaracteres difiere entre las herramientas que admiten expresiones regulares en la industria. Si transfiere expresiones regulares de otro entorno a la base de datos Oracle, asegúrese de que la sintaxis de expresiones regulares sea compatible y el comportamiento sea el esperado.

Funciones de Oracle Database SQL para expresiones regulares

La base de datos proporciona un conjunto de funciones SQL que le permiten buscar y manipular cadenas utilizando expresiones regulares. Puede usar estas funciones en cualquier tipo de datos que contenga datos de caracteres como CHAR, NCHAR, CLOB, NCLOB, NVARCHAR2 y VARCHAR2.

Una expresión regular debe estar encerrada o envuelta entre comillas simples. Al hacerlo, se asegura de que toda la expresión sea interpretada por la función SQL y puede mejorar la legibilidad de su código.

Nota:

Al igual que con todos los literales de texto utilizados en funciones de SQL, las expresiones regulares deben estar encerradas o envueltas entre comillas simples. Si su expresión regular incluye el carácter de comillas simples, ingrese dos comillas simples para representar una sola comilla dentro de su expresión.

Funciones de expresiones regulares de SQL

REGEXP_LIKE

Esta función busca una columna de caracteres para un patrón. Utilice esta función en la cláusula WHERE de una consulta para devolver filas que coincidan con la expresión regular que especifique.

Consulte la Referencia SQL de Oracle Database para obtener detalles de la sintaxis sobre la función REGEXP_LIKE.

https://docs.oracle.com/cd/B13789_01/server.101/b10759/conditions018.htm#SQLRF00501

REGEXP_REPLACE

Esta función busca un patrón en una columna de caracteres y reemplaza cada aparición de ese patrón con el patrón que especifique.

Consulte la Referencia SQL de Oracle Database para obtener detalles de la sintaxis sobre la función REGEXP_REPLACE.

https://docs.oracle.com/cd/B13789_01/server.101/b10759/functions115.htm#SQLRF06302

REGEXP_INSTR

Esta función busca una cadena para una ocurrencia dada de un patrón de expresión regular. Usted especifica qué ocurrencia desea encontrar y la posición de inicio para buscar. Esta función devuelve un número entero que indica la posición en la cadena donde se encuentra la coincidencia.

Consulte la Referencia SQL de Oracle Database para obtener detalles de la sintaxis sobre la función REGEXP_INSTR.

https://docs.oracle.com/cd/B13789_01/server.101/b10759/functions114.htm#SQLRF06300

REGEXP_SUBSTR

Esta función devuelve la subcadena real que coincide con el patrón de expresión regular que especifique.

Consulte la Referencia SQL de Oracle Database para obtener detalles de la sintaxis sobre la función REGEXP_SUBSTR.

https://docs.oracle.com/cd/B13789_01/server.101/b10759/functions116.htm#SQLRF06303

Referencias

Traducción del original:

https://docs.oracle.com/cd/B13789_01/appdev.101/b10795/adfns_re.htm#1007267

REVISIONES DE ESTE DOCUMENTO

FECHA	REVISADO POR	DESCRIPCIÓN DE CAMBIOS
2018-10-08	C. Ramirez	Se mejora los ejemplos de nombres de archivos. Se agrega la sección Prueba de Expresiones Regulares
2018-09-23	C. Ramirez	Desarrollo inicial