



 **ACADEMY**

# Programación de bases de datos con SQL

17-3

Expresiones regulares



**ORACLE** ACADEMY

Copyright © 2017, Oracle y/o sus filiales. Todos los derechos reservados.

# Objetivos

En esta lección se abordan los siguientes objetivos:

- Describir las expresiones regulares
- Utilizar expresiones regulares para buscar, hacer coincidir y sustituir cadenas en sentencias SQL
- Construir y ejecutar expresiones regulares y restricciones de control

# Objetivo

- En ocasiones, puede que tenga que buscar o sustituir una parte concreta de texto en una columna, cadena de texto o documento.
- Ya sabe cómo realizar una coincidencia de patrón simple mediante LIKE y comodines.
- En ocasiones puede que deba buscar cadenas de texto muy complejas como, por ejemplo, la extracción de todas URL de un fragmento de texto.
- Otras veces, es posible que se le solicite que realice una búsqueda más compleja como buscar todas las palabras cuyo segundo carácter es una vocal.

# Objetivo

- Las expresiones regulares son un método de descripción de patrones tanto simples como complejos para la búsqueda y la manipulación.
- Se utilizan ampliamente en el sector informático y no están limitadas a Oracle.
- La implantación de expresiones regulares de Oracle es una extensión de POSIX (Portable Operating System for UNIX) y, como tal, es completamente compatible con el estándar POSIX, controlado por Institute of Electrical and Electronics Engineers (IEEE).

Las expresiones regulares son un tema muy detallado y profundo. El único objetivo de esta lección es proporcionar una breve visión general de su finalidad y su uso.

# Expresiones regulares

- El uso de expresiones normales se basa en el uso de metacaracteres.
  - Los metacaracteres son caracteres con un significado especial, como un carácter comodín, un carácter repetitivo, un carácter no coincidente o un rango de caracteres.
  - Puede utilizar varios símbolos de metacaracteres predefinidos en la coincidencia de patrones.
  - En las siguientes diapositivas se muestran algunos de los metacaracteres y se proporciona una breve explicación de cada uno de ellos.

# Metacaracteres

Símbolo	Description
. (punto)	Coincide con cualquier carácter en el juego de caracteres soportados, excepto NULL
?	Coincide con ninguna o una incidencia
*	Coincide con cero o más incidencias
+	Coincide con una o más incidencias
()	Expresión de agrupación, que se trata como una sola subexpresión
\	Carácter de escape
	Operador de alternancia para especificar coincidencias alternativas
^/\$	Coincide con el principio de línea/final de línea
[]	Expresión de corchetes para una lista coincidente que coincide con cualquier expresión representada en la lista

La lista completa de metacaracteres con ejemplos de su uso se puede encontrar en [http://docs.oracle.com/cd/B19306\\_01/appdev.102/b14251/adfns\\_regexp.htm](http://docs.oracle.com/cd/B19306_01/appdev.102/b14251/adfns_regexp.htm)

# Ejemplos de Expresiones Normales

- Una expresión regular simple es muy similar a la búsqueda con comodines con la que ya está familiarizado.
- Veamos un ejemplo: vamos a utilizar el operador de punto para buscar la letra "a" seguida de cualquier carácter seguido de la letra "c".
- Como expresión regular, esto se haría como: 'a.c'.
- La misma expresión como búsqueda con comodines SQL estándar sería: WHERE column LIKE 'a\_c'.



# Ejemplos de Expresiones Normales

- ¿Cuáles de las siguientes cadenas coincidirá con 'a.c'?

1. 'ABC',
2. 'abc',
3. 'aqx',
4. 'axc',
5. 'aBc',
6. 'abC'
7. 'Amc'
8. 'amrc'



# Ejemplos de Expresiones Normales

- Las cadenas en rojo coinciden con la cadena de búsqueda 'a.c.'
- 'ABC', 'abc', 'aqx', 'axc', 'aBc', 'abC', 'Amc', 'amrc'
- El resto de ejemplos falla debido a que tienen el carácter en la posición incorrecta o con las mayúsculas/minúsculas incorrectos (mayúsculas no minúsculas según lo especificado en la cadena de búsqueda).

# Funciones de Expresiones Regulares

- Oracle proporciona un juego de funciones SQL que puede utilizar para buscar y manipular cadenas mediante expresiones regulares.
- Puede utilizar estas funciones en cualquier tipo de dato que contenga datos de caracteres como CHAR, CLOB y VARCHAR2.
- Una expresión regular debe ir entre comillas simples.

# Funciones de Expresiones Regulares

Name	Description
REGEXP_LIKE	Similar al operador LIKE, pero utiliza la coincidencia de expresiones normales en lugar de la coincidencia de patrones simple.
REGEXP_REPLACE	Busca un patrón de expresión regular y lo reemplaza por una cadena de sustitución.
REGEXP_INSTR	Busca una cadena dada para un patrón de expresión regular y devuelve la posición de la coincidencia encontrada.
REGEXP_SUBSTR	Busca un patrón de expresión regular dentro de una cadena dada y devuelve la subcadena coincidente.
REGEXP_COUNT	Devuelve el número de veces que aparece un patrón en una cadena. Especifique la cadena y el patrón. También puede especificar la posición de inicio y opciones de coincidencia (por ejemplo, c para sensibilidad a mayúsculas/minúsculas).

# Ejemplos de Funciones de Expresiones Regulares

- Supongamos que se le ha pedido que enumere todos los empleados con el nombre Stephen o Steven.

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

- Con expresiones regulares, solo tiene que utilizar la función REGEXP\_LIKE y la cadena de búsqueda: '^Ste(v|ph)en\$'
  - "^" especifica el inicio de la cadena que se está buscando
  - "S" mayúscula, seguida de
  - "t" minúscula, seguida de
  - "e" minúscula, seguida de

# Ejemplos de Funciones de Expresiones Regulares

- Con expresiones regulares, solo tiene que utilizar la función REGEXP\_LIKE y la cadena de búsqueda: '^Ste(v|ph)en\$'
  - "(" inicia una subexpresión
  - "v" minúscula
  - "|" especifica un O
  - "p" minúscula seguida de "h" mayúscula
  - ")" termina con el grupo de opciones,
  - "e" minúscula
  - "n" minúscula
  - "\$" especifica el final de la cadena que se está buscando

# Ejemplos de Funciones de Expresiones Regulares

- Ejemplo:

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

- Resultado:

FIRST_NAME	LAST_NAME
Steven	King

# Ejemplos de Funciones de Expresiones Regulares

- La función REPLACE de expresión regular sustituirá un patrón de cadena por otro.
- En este ejemplo se busca una "H" seguida de cualquier vocal y se sustituyen por dos símbolos "\*\*".

```
SELECT last_name, REGEXP_REPLACE(last_name, '^H(a|e|i|o|u)', '**')  
      AS "Name changed"  
FROM employees;
```

LAST_NAME	Name changed
Abel	Abel
Davies	Davies
...	...
Hartstein	**rtstein
Higgins	**ggins
...	...



# Ejemplos de Funciones de Expresiones Regulares

- La función COUNT de expresión regular devuelve el número de veces que aparece un patrón en una cadena.
- En este ejemplo se busca la subexpresión "ab"

```
SELECT country_name, REGEXP_COUNT(country_name, '(ab)') AS "Count of 'ab'"  
FROM wf_countries  
WHERE REGEXP_COUNT(country_name, '(ab)')>0;
```

COUNTRY_NAME	Count of 'ab'
Republic of Zimbabwe	1
Arab Republic of Egypt	1
Great Socialist Peoples Libyan Arab Jamahiriya	1
Kingdom of Saudi Arabia	1
Syrian Arab Republic	1
Gabonese Republic	1
United Arab Emirates	1

Subexpresión: expresión que forma parte de una expresión mayor.

# Expresiones Regulares en Restricciones de Control

- Las expresiones regulares también se pueden utilizar como parte del código de la aplicación para garantizar que solo se almacenen los datos en la base de datos.
- Se pueden incluir una llamada a una función de expresión regular, por ejemplo, una restricción CHECK.

# Expresiones Regulares en Restricciones de Control

- Por lo tanto, si desea asegurarse de que no se captura ninguna dirección de correo electrónico sin "@" en una tabla en la base de datos, solo tiene que agregar las siguientes restricciones de control:

```
ALTER TABLE employees  
ADD CONSTRAINT email_addr_chk  
CHECK(REGEXP_LIKE(email, '@'));
```

- Esto garantizará que todas las direcciones de correo electrónico incluyan un signo "@".

# Expresiones Regulares en Restricciones de Control

- Mediante expresiones regulares, puede comprobar el formato de las direcciones de correo electrónico más a fondo para comprobar que son válidas.
- Una dirección de correo electrónico válida tendrá uno o más caracteres y, a continuación, @, seguido de uno o más caracteres y, a continuación, un . (punto), seguido de uno o más caracteres.

```
CREATE TABLE my_contacts  
(first_name VARCHAR2(15),  
last_name VARCHAR2(15),  
email VARCHAR2(30) CHECK(REGEXP_LIKE(email, '.*@.*\..*')));
```

# Expresiones Regulares en Restricciones de Control

- Definiciones de sintaxis:

.+        significa uno o más caracteres

@        un símbolo @

\.        a . (un punto) (aquí la barra invertida es un carácter de escape)

```
CREATE TABLE my_contacts
(first_name VARCHAR2(15),
last_name VARCHAR2(15),
email VARCHAR2(30) CHECK(REGEXP_LIKE(email, '.*@.*\..*'));
```

# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- EXPRESIONES REGULARES
- Metacaracteres
- Subexpresiones

# Resumen

En esta lección, debe haber aprendido lo siguiente:

- Describir las expresiones regulares
- Utilizar expresiones regulares para buscar, hacer coincidir y sustituir cadenas en sentencias SQL
- Construir y ejecutar expresiones regulares y restricciones de control



 **ACADEMY**