

# 16

## Suporte a Expressões Regulares

## Objetivos deste Capítulo

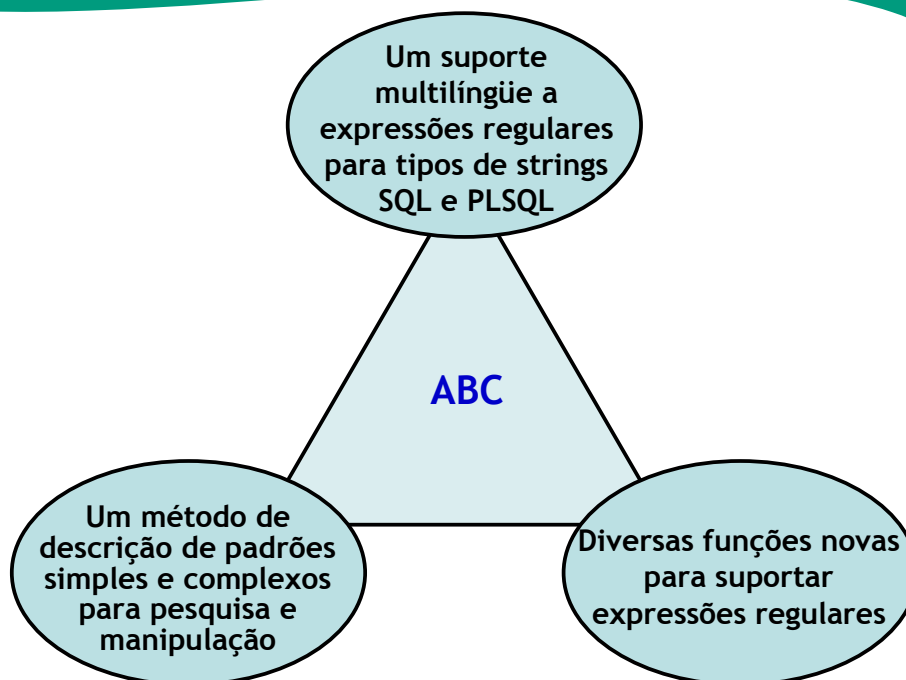
- Ao concluir este capítulo, você poderá usar o suporte a expressões SQL regulares para pesquisar, substituir e estabelecer uma correspondência com strings específicas.

16-2

### Objetivos deste Capítulo

Neste capítulo, você aprenderá a usar o recurso de suporte a expressões regulares que foi apresentado no Banco de Dados Oracle 10g.

# Expressões Regulares



16-3

## Visão Geral de Expressões Regulares

O Banco de Dados Oracle 10g apresenta o suporte a Expressões Regulares. Esta implementação é compatível com o padrão POSIX (Portable Operating System for UNIX), controlado pelo IEEE (Institute of Electrical and Electronics Engineers), para semântica e sintaxe de correspondência de dados ASCII. Os recursos multilíngües do Oracle ampliam os recursos de correspondência dos operadores, superando o padrão do POSIX. As expressões regulares são um método de descrição de padrões simples e complexos para pesquisa e manipulação de dados.

A manipulação e a pesquisa de strings contribuem para um grande percentual de lógica em uma aplicação baseada na Web. O uso varia desde a tarefa mais simples (encontrar as palavras "San Francisco" em um texto especificado) até a mais complexa (extrair todas as URLs do texto), ou tarefas ainda mais complexas (encontrar todas as palavras cujo segundo caractere é uma vogal).

Quando conjugado com o SQL nativo, o uso de expressões regulares possibilita operações de pesquisa e manipulação muito eficientes com os dados armazenados no banco de dados Oracle. É possível usar esse recurso para solucionar problemas que, de outro modo, teriam uma programação muito complexa.

# Metacaracteres

Símbolo	Descrição
*	Estabelece correspondência com zero ou mais ocorrências
	Operação de alteração para especificar correspondências alternativas
^/\$	Estabelece correspondência com o início de linha/fim de linha
[ ]	Expressão entre colchetes para uma lista de correspondências que contenha qualquer uma das expressões representadas na lista
{m}	Estabelece correspondência exatamente <i>m</i> vezes
{m,n}	Estabelece correspondência pelo menos <i>m</i> vezes, mas não mais do que <i>n</i> vezes
[ : ]	Especifica uma classe de caracteres e corresponde a qualquer caractere dessa classe
\	Pode ter 4 significados diferentes: 1. Significa uma barra propriamente dita. 2. Colocar o próximo caractere entre aspas. 3. Introduzir um operador. 4. Não fazer nada.
+	Corresponde a uma ou mais ocorrências
?	Estabelece correspondência com zero ou uma ocorrência
.	Corresponde a qualquer caractere no conjunto de caracteres suportado, exceto NULL
()	Expressão de agrupamento, tratada como uma única subexpressão
[ = ]	Especifica as classes de equivalência
\n	Expressão de referência retroativa
[..]	Especifica um elemento de comparação, como um elemento de multicaracteres

16-4

## Metacaracteres

Os metacaracteres são caracteres especiais que têm um significado especial, como um curinga, um caractere repetitivo, um caractere não correspondente ou uma faixa de caracteres. Você pode usar diversos símbolos de metacaracteres predefinidos no padrão de correspondência.

# Usando Metacaracteres

Problema: Localizar 'abc' em uma string:

Solução:

'abc'

Corresponde a: abc

Não corresponde a: 'def'

1

Problema: Localizar um 'a' seguido de qualquer caractere, seguido de 'c'

Metacaractere: qualquer caractere é definido por '.'

Solução:

'a.c'

Corresponde a: abc

Corresponde a: adc

Corresponde a: alc

Corresponde a: a&c

Não corresponde a: abb

2

Problema: Localizar uma ou mais ocorrências de 'a'

Metacaractere: Use o sinal de adição '+' para estabelecer uma correspondência com um ou mais dos caracteres anteriores

Solução:

'a+'

Corresponde a: a

Corresponde a: aa

Não corresponde a: bbb

3

16-5

## Usando Metacaracteres

1. No primeiro exemplo, é executada uma correspondência simples. 2. No segundo exemplo, qualquer caractere é definido como '.'. Este exemplo pesquisa o caractere "a" seguido de qualquer caractere, seguido do caractere "c". 3. O terceiro exemplo pesquisa uma ou mais ocorrências da letra "a". O caractere "+" é usado aqui para indicar uma correspondência de um ou mais dos caracteres dos caracteres anteriores.

Você também pode pesquisar listas de caracteres não correspondentes. Uma lista de caracteres não correspondentes possibilita definir um conjunto de caracteres para os quais uma correspondência é inválida. Por exemplo, para localizar todos os caracteres, exceto "a", "b" e "c", você pode definir "^" para indicar uma não correspondência.

Expressão: [^abc]  
Corresponde a: abcdef  
Corresponde a: ghi  
Não corresponde a: abc

Para estabelecer uma correspondência com qualquer letra que não esteja entre "a" e "i", você pode usar:

Expressão: [^a-i]  
Corresponde a: hijk  
Corresponde a: lmn  
Não corresponde a: abcdefghi

### Usando Metacaracteres (continuação)

Sintaxe do Metacaractere	Nome do Operador	Descrição
.	Qualquer caractere - Ponto	Estabelece correspondência com qualquer caractere
+	Um ou Mais - Quantificador Sinal de Adição	Estabelece correspondência com uma ou mais ocorrências da subexpressão precedente
?	Zero ou Um - Quantificador Ponto de Interrogação	Estabelece correspondência com zero ou uma ocorrência da subexpressão precedente
*	Zero ou Mais - Quantificador Asterisco	Estabelece correspondência com zero ou mais ocorrências da subexpressão precedente
{m} {m,} {m,n}	Intervalo - Contagem Exata	Estabelece correspondência com <ul style="list-style-type: none"> <li>• exatamente <i>m</i> ocorrências</li> <li>• pelo menos <i>m</i> ocorrências</li> <li>• pelo menos <i>m</i>, mas não mais do que <i>n</i> ocorrências da subexpressão precedente</li> </ul>
[...]	Lista de Caracteres de Correspondência	Estabelece correspondência com qualquer caractere da lista ...
[^...]	Lista de Caracteres Não Correspondentes	Estabelece correspondência com qualquer caractere que não esteja na lista ...
	Ou	'a b' corresponde a 'a' ou 'b'.
(...)	Subexpressão ou Agrupamento	Trata a expressão ... como uma unidade.
\n	Referência retroativa	Estabelece correspondência com a subexpressão precedente de número, <i>n</i> é um inteiro entre 1 e 9
\	Caractere de Escape	Trata o metacaractere subsequente como um literal.

### Usando Metacaracteres (continuação)

Sintaxe do Metacaractere	Nome do Operador	Descrição
<b>^</b>	Início de Âncora de Linha	Estabelece correspondência com a expressão subsequente quando ele ocorre no início de uma linha.
<b>\$</b>	Fim de Âncora de Linha	Estabelece correspondência apenas quando ocorre no fim de uma linha.
<b>[class:]</b>	Classe de Caracteres do POSIX	Estabelece correspondência com qualquer caractere que pertença à classe de caracteres especificada.

# Funções de Expressões Regulares

Nome da Função	Descrição
<b>REGEXP_LIKE</b>	É semelhante ao operador <code>LIKE</code> , mas executa a correspondência de expressões regulares em vez de fazer a correspondência de padrão simples
<b>REGEXP_REPLACE</b>	Pesquisa um padrão de expressão regular e o substitui por uma string de substituição
<b>REGEXP_INSTR</b>	Procura em determinada string um padrão de expressão regular e retorna a posição em que o correspondente foi localizado
<b>REGEXP_SUBSTR</b>	Pesquisa um padrão de expressão regular em determinada string e retorna a substring correspondente

16-8

## Funções de Expressões Regulares

O Banco de Dados Oracle 10g oferece um conjunto de funções SQL que podem ser usadas para pesquisar e manipular strings usando expressões regulares. É possível usar essas funções com qualquer tipo de dados que contenha dados de caractere como `CHAR`, `NCHAR`, `CLOB`, `NCLOB`, `NVARCHAR2` e `VARCHAR2`. Uma expressão regular deve ser delimitada ou encapsulada em aspas simples. Assim, garante-se que a expressão inteira será interpretada pela função SQL e que haverá maior legibilidade do código.

**REGEXP\_LIKE:** Esta função pesquisa um padrão em uma coluna de caracteres. Use esta função na cláusula `WHERE` de uma consulta para retornar linhas que correspondam à expressão regular especificada.

**REGEXP\_REPLACE:** Esta função pesquisa um padrão em uma coluna de caracteres e substitui cada ocorrência pelo padrão especificado.

**REGEXP\_INSTR:** Esta função pesquisa determinada ocorrência de um padrão de expressão regular em uma string. Você especifica qual ocorrência deseja localizar e a posição de início da pesquisa. Esta função retorna um inteiro indicando a posição na string em que o correspondente foi localizado.

**REGEXP\_SUBSTR:** Esta função retorna a substring real que corresponde ao padrão de expressão regular especificado.



## A Sintaxe da Função REGEXP

```
REGEXP_LIKE (texto_pesq, padrão [,opção_acerto])
```

```
REGEXP_INSTR (srcstr, padrão [, posição [, ocorrência  
[, opção_retorno [, opção_acerto]]]])
```

```
REGEXP_SUBSTR (srcstr, padrão [, posição  
[, ocorrência [, opção_acerto]]])
```

```
REGEXP_REPLACE(srcstr, padrão [,texto_subst [, posição  
[, ocorrência [, opção_acerto]]]])
```

16-9

### A Sintaxe da Função REGEXP

A tabela a seguir contém descrições dos termos mostrados no slide de sintaxe.

texto_pesq	Valor de pesquisa
padrão	Expressão regular
ocorrência	Ocorrência a ser pesquisada
posição	Posição de início da pesquisa
opção_retorno	Posição de início ou fim da ocorrência
texto_subst	Padrão de substituição da string de caracteres
opção_acerto	Opção para alterar a correspondência padrão. Pode incluir um ou mais dos seguintes valores: "c" —usa correspondência com distinção entre maiúsculas e minúsculas (padrão) "i" —usa correspondência sem distinção entre maiúsculas e minúsculas "n" — permite operador de correspondência com qualquer caractere "m" —trata a string de origem como uma linha múltipla

# Executando Pesquisas Básicas

```
SELECT nome, sobrenome  
FROM funcionario  
WHERE REGEXP_LIKE (sobrenome, '^C(arlo|have)s$');
```

NOME	SOBRENOME
Roberto	Carlos
Pedro	Chaves

16-10

## Exemplo de REGEXP\_LIKE

Nesta consulta à tabela `FUNCIONARIO`, serão exibidos todos os funcionários cujos sobrenomes contenham Carlos ou Chaves. Na expressão usada,

`'^C(arlo|have)s$'` :

- `^` indica o início da sentença
- `$` indica o fim da sentença
- `|` indica ou

# Verificando a Presença de um Padrão

```
SELECT endereco,  
       REGEXP_INSTR(endereco, '^[:alpha:]')  
FROM   localidade  
WHERE  
       REGEXP_INSTR(endereco, '^[:alpha:]') > 1;
```

ENDEREÇO	REGEXP_INSTR(ENDEREÇO, '^[:ALPHA:]')
Rua da Aurora 174	4
Rua Haddock Lobo 1048	4
Rua Duque de Caxias 1187	4
Magdalen Centre, The Oxford Science Park	9

16-11

## Verificando a Presença de um Padrão

Neste exemplo, a função `REGEXP_INSTR` é usada para pesquisar o endereço a fim de encontrar o local do primeiro caractere não alfabético, independentemente se está em maiúsculas ou minúsculas. A pesquisa é executada apenas para os endereços que não começam com um número. Observe que `[:<classe>:]` implica uma classe de caracteres e corresponde a qualquer caractere dessa classe; `[:alpha:]` corresponde a qualquer caractere alfabético. Os resultados são exibidos.

Na expressão usada na consulta `'^[:alpha:]'`:

- `[` inicia a expressão
- `^` indica NOT
- `[:alpha:]` indica a classe de caracteres alfa
- `]` finaliza a expressão

O operador de classe de caracteres do POSIX permite pesquisar uma expressão em uma lista de caracteres que é membro de uma classe de caracteres específica do POSIX. É possível usar este operador para pesquisar uma formatação específica, como caracteres maiúsculos, ou para pesquisar caracteres especiais, como dígitos ou caracteres de pontuação. Todo o conjunto de classes de caracteres do POSIX é suportado. Use a sintaxe `[:classe:]` em que *classe* é o nome da classe de caracteres do POSIX a ser pesquisada. A expressão regular abaixo pesquisa um ou mais caracteres maiúsculos consecutivos: `[:upper:]+`.

## Exemplo de Extração de Substrings

```
SELECT REGEXP_SUBSTR(endereco , ' [^ ]+ ') "Rua"  
FROM localidade;
```

Rua
da
Haddock
Duque
Bloor
Centre,

16-12

### Exemplo de uma Extração de Substring

Neste exemplo, os nomes de ruas são extraídos da tabela `LOCALIDADE`. Para isso, o conteúdo da coluna `ENDERECO` que está antes do primeiro espaço é retornado usando a função `REGEXP_SUBSTR`. Na expressão usada na consulta `' [^ ]+ '`:

- `[` inicia a expressão
- `^` indica NOT
- indica espaço
- `]` finaliza a expressão
- `+` indica 1 ou mais
- indica espaço

## Substituindo Padrões

```
SELECT REGEXP_REPLACE( nome_pais, '(.)',  
                        '\1 ') "REGEXP_REPLACE"  
FROM pais;
```

REGEXP_REPLACE
Brasil
Canadá
Reino Unido

16-13

### Substituindo Padrões

Este exemplo examina `NOME_PAIS`. O banco de dados Oracle reformata este padrão com um espaço após cada caractere não nulo da string. Os resultados são mostrados.

# Expressões Regulares e Constraints de Verificação

```
ALTER TABLE funcionario  
ADD CONSTRAINT func_email_ck  
CHECK (REGEXP_LIKE(email, '@')) NOVALIDATE ;
```

1

```
INSERT INTO funcionario VALUES  
(500, 'Cristino', 'Pascoal',  
'CrisPascoal.com', 1234567890,  
'12-Jan-2004', 'AD_VP', 2000, null, 102, 40) ;
```

2

```
INSERT INTO funcionario VALUES  
*  
ERRO na linha 1:  
ORA-02290: restrição de verificação  
(PROFESSOR.FUNC_EMAIL_CK) violada
```

16-14

## Expressões Regulares e Constraints de Verificação

As expressões regulares também podem ser usadas em constraints de verificação. Neste exemplo, uma constraint de verificação é adicionada à coluna `EMAIL` da tabela `FUNCIONARIO`. Isso garantirá que apenas as strings que contiverem um símbolo "@" serão aceitas. A constraint é testada. A constraint de verificação é violada porque o endereço de email não contém o símbolo solicitado. A cláusula `NOVALIDATE` garante que os dados existentes não serão verificados.

# Exercício 16

16-15

## Exercício 16

1. Crie uma consulta para pesquisar, na tabela `FUNCIONARIO`, todos os funcionários cujos nomes começam com "Ro" ou "Ra".

NOME	SOBRENOME
Roberto	Carlos
Rafael	Miranda
Roberto	Nascimento

2. Crie uma consulta que remova, na exibição, os espaços da coluna `ENDEREÇO` da tabela `LOCALIDADE`.

ENDEREÇO
RuadaAurora174
RuaHaddockLobo1048
RuaDuquedeCaxias1187
460BloorSt.W.
MagdalenCentre,TheOxfordSciencePark

3. Crie uma consulta que, na exibição, substitua "St" por "Street" na coluna `ENDEREÇO` da tabela `LOCALIDADE`. Tenha cuidado para não afetar as linhas que já incluam "Street". Exiba apenas as linhas que forem afetadas.

<code>REGEXP_REPLACE(ENDEREÇO,'ST','STREET')</code>
460 Bloor Street. W.