

FIAP GRADUAÇÃO

Tecnologia em Análise e Desenvolvimento de Sistemas

Application Development For Databases

PROF. MILTON

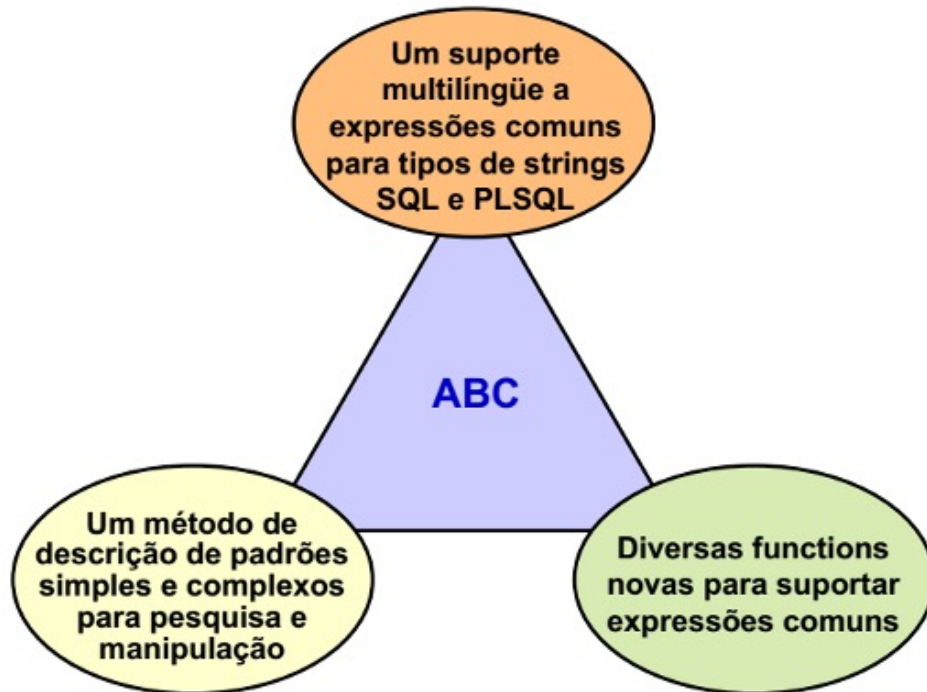
Expressões Regulares

Objetivos

Ao concluir esta lição, você será capaz de usar o suporte a expressões SQL comuns para pesquisar, substituir e estabelecer uma correspondência com strings específicas.



Visão Geral de Expressões Comuns



Functions de Expressões Comuns

Nome da Function	Descrição
REGEXP_LIKE	É semelhante ao operador LIKE, mas executa a correspondência de expressões comuns em vez de fazer a correspondência de padrão simples
REGEXP_REPLACE	Pesquisa um padrão de expressão comum e o substitui por uma string de substituição
REGEXP_INSTR	Procura em determinada string um padrão de expressão comum e retorna a posição em que o correspondente foi localizado
REGEXP_SUBSTR	Pesquisa um padrão de expressão comum em determinada string e retorna a substring correspondente

A Sintaxe da Function REGEXP

```
REGEXP_LIKE (srcstr, pattern [,match_option])
```

```
REGEXP_INSTR (srcstr, pattern [, position [, occurrence  
[, return_option [, match_option]]]])
```

```
REGEXP_SUBSTR (srcstr, pattern [, position  
[, occurrence [, match_option]]])
```

```
REGEXP_REPLACE(srcstr, pattern [,replacestr [, position  
[, occurrence [, match_option]]]])
```

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

Usando Metacaracteres

Prepare o seu ambiente de testes:

```
CREATE TABLE E1 AS  
  SELECT ENAME nome, HIREDATE  
  FROM EMP;
```

```
INSERT INTO E1  
  SELECT INITCAP(ENAME) nome, HIREDATE  
  FROM EMP;
```

```
INSERT INTO E1  
  SELECT LOWER(ENAME) nome, HIREDATE  
  FROM EMP;
```

Usando Metacaracteres

Prepare o seu ambiente de testes:

```
INSERT INTO e1
VALUES ('www.oracle.com', sysdate +1);

INSERT INTO e1
VALUES ('', sysdate +2);

INSERT INTO e1
VALUES (''||chr(9)||chr(10)||chr(11)||chr(12)||chr(13), sysdate +3);

INSERT INTO e1
VALUES ('192.168.0.15', sysdate +4);

INSERT INTO e1
VALUES ('oracle@oracle.com', sysdate + 5);
```

```
INSERT INTO e1 VALUES ('www.o.com', sysdate +1);
```

```
INSERT INTO e1 VALUES ('', sysdate +2);
```

```
INSERT INTO e1 VALUES (''||chr(9)||chr(10)||chr(11)||chr(12)||chr(13), sysdate +3);
```

```
INSERT INTO e1 VALUES ('192.15', sysdate +4);
```

```
INSERT INTO e1 VALUES ('o@o.com', sysdate + 5);
```

Usando Metacaracteres

O metacaracter **^** corresponde ao início da cadeia de caracteres.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, '^K');
```

'c' indica “case sensitive”.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, '^K', 'c');
```

'i' indica “case insensitive”.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, '^K', 'i');
```

Usando Metacaracteres

O metacaracter **^** corresponde ao fim da cadeia de caracteres.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, 'G$');
```

'c' indica “case sensitive”.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, 'G$', 'c');
```

'i' indica “case insensitive”.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, 'G$', 'i');
```

Usando Metacaracteres

O metacaracter [] indica uma lista

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, '^ST[ie]', 'i');
```

```
SELECT hiredate FROM e1  
WHERE REGEXP_LIKE  
  (TO_CHAR(hiredate, 'YYYY'), '^198[0-5]$');
```

Usando Metacaracteres

O metacaracter '.' indica qualquer caractere.

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, '^S....$', 'i');
```

```
SELECT nome FROM e1  
WHERE REGEXP_LIKE (nome, '^.....$', 'i');
```

Usando Metacaracteres

O metacaracter '{ }' indica um número de caracteres.

```
SELECT nome  
  FROM e1  
 WHERE REGEXP_LIKE (nome, '^S.{5}$', 'i');
```

Usando Metacaracteres

O metacaracter '|' alternancia.

```
SELECT nome
  FROM e1
 WHERE REGEXP_LIKE (nome, '(scott|king)', 'i');
```

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


Classes de Metacaracteres

Classe	Significado
[:alnum:]	<i>Caracteres Alfanuméricos</i>
[:alpha:]	Maiúsculas/Minúsculas
[:cntrl:]	<i>Caracteres de controle</i>
[:digit:]	Números
[:lower:]	<i>Caracteres Minúsculos</i>
[:punct:]	Sinais de pontuação
[:space:]	<i>Caracteres brancos</i>
[:upper:]	Caracteres Maiúsculos

Usando Classes de Metacaracteres

```
SELECT nome, length(nome)
  FROM e1
 WHERE REGEXP_LIKE(nome, '^[:space:]*$');
```

```
SELECT nome
  FROM e1
 WHERE REGEXP_LIKE(nome, '[:digit:]+');
```

Verificando a Presença de um Padrão

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

STREET_ADDRESS	REGEXP_INSTR(STREET_ADDRESS,'^[[:ALPHA:]]')
Magdalen Centre, The Oxford Science Park	9
Schwanthalerstr. 7031	16
Rua Frei Caneca 1360	4
Murtenstrasse 921	14
Pieter Breughelstraat 837	7
Mariano Escobedo 9991	8

Exemplo de Extração de Substrings

```
SELECT REGEXP_SUBSTR(street_address , ' [^ ]+ ' )  
"Road" FROM locations;
```

Road
Via
Calle
Jabberwocky
Interiors
Zagora
Charade

...

Substituindo Padrões

```
SELECT REGEXP_REPLACE( country_name, '(.)',  
                        '\1 ') "REGEXP_REPLACE"  
FROM countries;
```

REGEXP_REPLACE(COUNTRY_NAME,'(',')','\1')
Argentina
Australia
Belgium
Brazil
Canada
Switzerland
China

...

Expressões Comuns e Constraints de Verificação

```
ALTER TABLE emp8  
  ADD CONSTRAINT email_addr  
  CHECK (REGEXP_LIKE(email, '@')) NOVALIDATE ;
```

1

```
INSERT INTO emp8 VALUES  
  (500, 'Christian', 'Patel',  
   'ChrisP2creme.com', 1234567890,  
   '12-Jan-2004', 'HR_REP', 2000, null, 102, 40) ;
```

2

```
INSERT INTO emp8 VALUES  
*
```

ERROR at line 1:
ORA-02290: check constraint (ORA20.EMAIL_ADDR) violated

Expressões Regulares e Constraints de Verificação

```
CREATE TABLE T1 (  
  COL1 VARCHAR(2) NOT NULL  
  CONSTRAINT COL1_UF_FORMAT  
  CHECK(REGEXP_LIKE(COL1, '[A-Z]{2}', 'c'))  
);  
  
insert into t1 values ('DF');  
insert into t1 values ('df');  
insert into t1 values ('D');  
insert into t1 values ('f');
```

Expressões Regulares e Constraints de Verificação

```
CREATE TABLE T2 (  
  COL1 VARCHAR2(11) NOT NULL  
  CONSTRAINT COL1_NUMBER_FORMAT  
    CHECK(REGEXP_LIKE(COL1, '\d{11}'))  
);  
  
insert into t2 values ('01234567890');  
insert into t2 values ('0123456789');  
insert into t2 values ('012345678XX');
```


Expressões Regulares e Constraints de Verificação

```
CREATE TABLE contatos(  
  email VARCHAR(40) CONSTRAINT chk_email  
    CHECK (REGEXP_LIKE(email, '^([a-zA-Z][[:alnum:]]_.-  
)*@[a-zA-Z][[:alnum:]]_.-[.]([a-zA-Z]+)$')),  
  cep8 CHAR(8) CONSTRAINT chk_cep8  
    CHECK (REGEXP_LIKE(cep8, '^([[:digit:]]{8})$')),  
  cep9 CHAR(9) CONSTRAINT chk_cep9  
    CHECK (REGEXP_LIKE(cep9, '^([[:digit:]]{5}-  
[[:digit:]]{3})$')),  
  uf CHAR(2) CONSTRAINT chk_uf  
    CHECK  
(REGEXP_LIKE(uf, '^A(C|L|M|P)|BA|CE|DF|ES|GO|M(A|G|S|T)|P(A  
|B|E|I|R)|R(J|N|O|R|S)|S(C|E|P)|TO$'))  
);
```

```
CREATE TABLE contatos(  
  email VARCHAR(40)  
    CONSTRAINT chk_email  
      CHECK (REGEXP_LIKE(email, '^([a-zA-Z][[:alnum:]]_.-  
)*@[a-zA-Z][[:alnum:]]_.-  
[.]([a-zA-Z]+)$')),  
  cep8 CHAR(8)  
    CONSTRAINT chk_cep8  
      CHECK (REGEXP_LIKE(cep8, '^([[:digit:]]{8})$')),  
  cep9 CHAR(9)  
    CONSTRAINT chk_cep9  
      CHECK (REGEXP_LIKE(cep9, '^([[:digit:]]{5}-[[:digit:]]{3})$')),  
  uf CHAR(2)
```

```
CONSTRAINT chk_uf
CHECK
(REGEXP_LIKE(uf,'^A(C|L|M|P)|BA|CE|DF|ES|GO|M(A|G|S|T)|P(A|B|E|I|R)|R(J|N|
O|R|S)|S(C|E|P)|TO$'))
);
```

Sumário

Nesta lição, você aprendeu a usar o suporte a expressões SQL comuns para pesquisar, substituir e estabelecer uma correspondência com strings específicas.