

字符串、字符串函數、操作符

Oracle數據庫提供了大量的字符串數據類型

如：

CHAR

NCHAR

VARCHAR2

NVARCHAR2

CLOB

NCLOB

如果需要聲明一個可變長度的字符串時，必須提供該字符串的最大長度。

如：

varchar2(30)

=====

範例：

```
DECLARE
  name varchar2(20);
  company varchar2(30);
  introduction clob;
  choice char(1);
BEGIN
  name := 'John Smith';
  company := 'Infotech';
  introduction := ' Hello! I''m John Smith from Infotech.';
  choice := 'y';
  IF choice = 'y' THEN
    dbms_output.put_line(name);
    dbms_output.put_line(company);
    dbms_output.put_line(introduction);
  END IF;
END;
/
```

db<>fiddle

Oracle

21c

run

markdown

By using db<>fiddle, you agree to license everything you submit by [Creative Commons CC0](#)

+

≡

```

1 DECLARE
2   name varchar2(20);
3   company varchar2(30);
4   introduction clob;
5   choice char(1);
6 BEGIN
7   name := 'John Smith';
8   company := 'Infotech';
9   introduction := ' Hello! I''m John Smith from Infotech.';
10  choice := 'y';
11  IF choice = 'y' THEN
12    dbms_output.put_line(name);
13    dbms_output.put_line(company);
14    dbms_output.put_line(introduction);
15  END IF;
16 END;
17 /

```

1 rows affected

```

dbms_output:
John Smith
Infotech
Hello! I'm John Smith from Infotech.

```

=====

字符串函數和操作符：

1	ASCII(x) ; 返回字符 x 的 ASCII 值		15	NLS_UPPER(x) ; 相同，不同之處在於它可以使用不同的排序方法所指定 NLSSORT UPPER 函數
2	CHR(x) ; 返回字符 x 的 ASCII 值		16	NLSSORT(x) ; 改變排序的字符的方法。任何NLS函數之前必須指定該參數;否則，默認的排序被使用
3	CONCAT(x, y) ; 連接字符串x和y，並返回附加的字符串		17	NVL(x, value) ; 返回如果x為null返回null; 否則返回x
4	INITCAP(x) ; 每個單詞的首字母x中轉換為大寫，並返回該字符串		18	NVL2(x, value1, value2) ; 如果x不為null返回value1; 如果x為null，則返回value2
5	INSTR(x, find_string [, start] [,		19	REPLACE(x, search_string, replace_string) ;

	occurrence]); 搜索find_string在x中並返回它出現的位置			搜索x對於SEARCH_STRING並替換使用replace_string它
6	INSTRB(x); 返回另一個字符串中字符串的位置，但返回以字節為單位的值		20	RPAD(x, width [, pad_string]); 填充x到右側
7	LENGTH(x); 返回x中的字符數		21	RTRIM(x [, trim_string]); 從x右邊修剪
8	LENGTHB(x); 返回為單字節字符集的字節的字符串的長度		22	SOUNDEX(x); 返回包含x的拼音表示形式的字符串
9	LOWER(x); 在x轉換為小寫字母，並返回該字符串		23	SUBSTR(x, start [, length]); 返回x的一個子開始於由start指定的位置。可選長度為子字符串
10	LPAD(x, width [, pad_string]); X用空格向左填充，把字符串的總長度達寬字符		24	SUBSTRB(x); 相同SUBSTR除外的參數均以字節代替字符的單字節字符的係統
11	LTRIM(x [, trim_string]); 從x的左修剪字符		25	TRIM([trim_char FROM] x); 從左側和右側修剪x字符
12	NANVL(x, value); 如果x匹配NaN的特殊值（非數字）則返回其值，否則返回x		26	UPPER(x); x轉換為大寫字母，並返回該字符串
13	NLS_INITCAP(x); 相同INITCAP函數，但它可以使用不同的排序方法所指定NLSSORT			
14	NLS_LOWER(x); 同樣的，不同的是它可以使用不同的			

排序方法所指定 NLSSORT LOWER函數			
-------------------------------	--	--	--

範例1.：

```

DECLARE
  greetings varchar2(11) := 'hello world';

BEGIN
  dbms_output.put_line('ASCII = '||ASCII(greetings));

  dbms_output.put_line('UPPER = '||UPPER(greetings));

  dbms_output.put_line('LOWER = '||LOWER(greetings));

  dbms_output.put_line('INITCAP = '||INITCAP(greetings));

  dbms_output.put_line ( '從第1個字母開始，向後 顯示出1個字元 = '||SUBSTR (greetings, 1, 1));
  dbms_output.put_line ( '從最後1個字母開始，向後 顯示出1個字元 = '||SUBSTR (greetings, -1, 1));
  dbms_output.put_line ( '從第7個字母開始，向後 顯示出5個字元 = '||SUBSTR (greetings, 7, 5));
  dbms_output.put_line ( '從第4個字母開始，向後 顯示出所有字元 = '||SUBSTR (greetings, 4));

  dbms_output.put_line ( '顯示出"e"這個字元，總共有幾個 = '||INSTR (greetings, 'e'));

END;
/

```

db<>fiddle
Oracle
21c
run
markdown

By using db<>fiddle, you agree to license everything you submit by [Creative Commons CC0](#).

```

1 DECLARE
2   greetings varchar2(11) := 'hello world';
3
4 BEGIN
5   dbms_output.put_line('ASCII = '||ASCII(greetings));
6
7   dbms_output.put_line('UPPER = '||UPPER(greetings));
8
9   dbms_output.put_line('LOWER = '||LOWER(greetings));
10
11  dbms_output.put_line('INITCAP = '||INITCAP(greetings));
12
13  dbms_output.put_line ( '從第1個字母開始，向後 顯示出1個字元 = '||SUBSTR (greetings, 1, 1)
14 |
15  dbms_output.put_line ( '從最後1個字母開始，向後 顯示出1個字元 = '||SUBSTR (greetings, -1, 1)
16 |
17  dbms_output.put_line ( '從第7個字母開始，向後 顯示出5個字元 = '||SUBSTR (greetings, 7, 5)
18 |
19  dbms_output.put_line ( '從第4個字母開始，向後 顯示出所有字元 = '||SUBSTR (greetings, 4));
20
21  dbms_output.put_line ( '顯示出"e"這個字元，總共有幾個 = '||INSTR (greetings, 'e'));
22
23 END;
24 /

```

1 rows affected
dbms_output:
ASCII = 104
UPPER = HELLO WORLD
LOWER = hello world
INITCAP = Hello World
從第1個字母開始，向後 顯示出1個字元 = h
從最後1個字母開始，向後 顯示出1個字元 = d
從第7個字母開始，向後 顯示出5個字元 = world
從第4個字母開始，向後 顯示出所有字元 = lo world
顯示出"e"這個字元，總共有幾個 = 2

範例2.：

```
DECLARE
  greetings varchar2(30) := '...~..Hello World...!..';
BEGIN
  -- 從字串"右邊"開始，移除掉"."字元，直到出現其他字元
  dbms_output.put_line(RTRIM(greetings, '.'));

  -- 從字串"左邊"開始，移除掉"."字元，直到出現其他字元
  dbms_output.put_line(LTRIM(greetings, '.'));

  -- 分別從字串"左邊"和"右邊"開始，移除掉"."字元，直到出現其他字元
  dbms_output.put_line(TRIM( '.' from greetings));

END;
/
```

db<>fiddle Oracle 21c run markdown

By using db<>fiddle, you agree to license everything you submit by [Creative Commons CC0](#).

```
1 DECLARE
2   greetings varchar2(30) := '...~..Hello World...!..';
3 BEGIN
4   -- 從字串"右邊"開始，移除掉"."字元，直到出現其他字元
5   dbms_output.put_line(RTRIM(greetings, '.'));
6
7   -- 從字串"左邊"開始，移除掉"."字元，直到出現其他字元
8   dbms_output.put_line(LTRIM(greetings, '.'));
9
10  -- 分別從字串"左邊"和"右邊"開始，移除掉"."字元，直到出現其他字元
11  dbms_output.put_line(TRIM( '.' from greetings));
12
13 END;
14 /
```

```
1 rows affected
dbms_output:
...~..Hello World..!
~..Hello World...!..
~..Hello World..!
```