

Postgres PL/pgSQL String and Formating Functions

Source: <http://www.postgresql.org/docs/8.1/static/functions-string.html>
<http://www.postgresql.org/docs/8.1/static/functions-formatting.html>
<http://www.postgresql.org/docs/8.1/static/functions-array.html>
<http://www.postgresql.org/docs/8.1/static/functions-aggregate.html>
<http://www.postgresql.org/docs/9.1/static/functions-string.html>
 Practical <http://www.adderpit.com/practical-postgresql/book1.htm>

STRING FUNCTIONS

Function	Example	Return Type	Result
Description			
string string concatenation	'Post' 'greSQL'	text	String PostgreSQL
char_length(string) in string	char_length('jose');	int	Number of chars 4
lower(string) string to lower case	lower('TOM')	text	Convert tom
overlay(string placing string from int [for int]) substring	overlay('Txxxxas' placing 'hom' from 2 for 4)	text	Replace Thomas
position(substring in string) specified substring	position('om' in 'Thomas')	int	Location of 3
substring(string [from int] [for int]) substring	substring('Thomas' from 2 for 3)	text	Extract hom
trim([leading trailing both] [characters] from string) longest string	trim(both 'x' from 'xTomxx')	text	Remove the Tom
only the chars			containing (space
defaults) form start/end			/both ends of
string			
upper(string) string to uppercase	upper('tom')	text	Convert TOM
initcap(text) letter of each	initcap('hi THOMAS')	text	Convert first Hi Thomas
uppercase and rest			word to to lower case
length(string text) characters in string	length('jose')	int	Number of 4
md5(string text) the MD5 hash	md5('abc')	text	Calculates 900150983cd24fb0
d6963f7d28e17f72 strpos(string, substring) specified substr	strpo('high','ig')	int	Location of 2
substr(string, from [, count]) substring	substr('alphabet',3,2)	int	Extract ph
replace(string text, from text, to text) occurrences in	replace('abcdefabcdef', 'cd', 'XX')	text	Replace all abXXefabXXef
substring from with			string of substring to
e.g. UPDATE dataelement SET name=replace(name,'IPD2012 ','DTC2012 ') WHERE			

dataelement.name LIKE 'DTC2012 %';

FORMATTING FUNCTIONS

Function	Example	Return Type
Description to_char(timestamp, text) stamp to string	to_char(current_timestamp, 'HH12:MI:SS')	text convert time

Array Functions and Operators

Operator	Description	Result
Example = equal [1,2,3]	t	ARRAY[1.1,2.1,3.1]::int[] = ARRAY
<> equal [1,2,4]	not	ARRAY[1,2,3] <> ARRAY
< than [1,2,4]	less	ARRAY[1,2,3] < ARRAY
> than [1,2,4]	greater	ARRAY[1,4,3] > ARRAY
<= equal [1,2,3]	less than or	ARRAY[1,2,3] <= ARRAY
>= equal [1,4,3]	greater than or	ARRAY[1,4,3] >= ARRAY
 concatenation [4,5,6]	array-to-array ARRAY[1,2,3] ARRAY	{1,2,3,4,5,6}
 concatenation [7,8,9]	array-to-array ARRAY[1,2,3] ARRAY[[4,5,6], {1,2,3},{4,5,6},{7,8,9}}	{1,2,3,4,5,6,7,8,9}
 concatenation [4,5,6]	element-to-array 3 ARRAY	{3,4,5,6}
 concatenation 7	array-to-element ARRAY[4,5,6]	{4,5,6,7}

Operator	Return type
Description Example array_cat (anyarray, anyarray) arrays [4,5])	anyarray concatenate two array_cat(ARRAY[1,2,3], ARRAY {1,2,3,4,5})
array_append (anyarray, anyelement) an array array_append(ARRAY[1,2], 3)	anyarray append an element to the end of {1,2,3}
array_prepend (anyelement, anyarray) beginning of an array array_prepend(1, ARRAY [2,3])	anyarray append an element to the {1,2,3}
array_dims (anyarray) representation of array's dimensions [4,5,6]])	text returns a text array_dims(ARRAY[[1,2,3], [1:2][1:3])
array_to_string (anyarray, text) elements using provided delimiter '~^~')	text concatenates array array_to_string(ARRAY[1, 2, 3], 1~^~2~^~3)
string_to_array (text, text) elements using provided delimiter '~^~')	text[] splits string into array string_to_array('xx~^~yy~^~zz', {xx,yy,zz})