Functions and Operators

NGUYEN Hong Phuong
phuongnh@soict.hut.edu.vn

1

Contents

- Logical operators
- Comparison operators
- Mathematical functions and operators
- String functions and operators
- Binary functions and operators
- Bit functions and operators
- Pattern matching
- Data type formatting functions
- Date/Time functions and operators

2

1. Logical operators

- The usual logical operators are available: AND, OR, NOT

a	b	a AND b	a OR b
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL
NULL	NULL	NULL	NULL

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

3

2. Comparison operators

Operator	Description
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
=	equal
<> or !=	not equal

- BETWEEN
  - a BETWEEN x AND y ⇔ a>=x AND a<=y
  - a NOT BETWEEN x AND y ⇔ a<x OR a>y
- Check whether a value is or is not null
  - Expression IS NULL
  - Expression IS NOT NULL

4

3. Mathematical Functions and Operators

- +, -, *, /, %,
- ^ (exponentiation): 2.0^3.0 = 8
- |/ (square root): |/ 25.0 = 5
- ||/ (cube root): ||/ 27.0 = 3
- ! (factorial): 5! = 120
- !! 5 = 120
- @ -5.0 = 5
- & (bitwise AND): 91 & 15 = 11
- | (bitwise OR): 32 | 3 = 35
- # (bitwise XOR): 17 # 5 = 20
- <<, >>

5

3. Mathematical Functions….(cont)

- Mathematical Functions

Function	Return Type	Description	Example	Result
abs(x)	(same as x)	absolute value	abs(-17.4)	17.4
cbrt(dp)	dp	cube root	cbrt(27.0)	3
ceil(dp or numeric)	(same as input)	smallest integer not less than argument	ceil(-42.8)	-42
ceiling(dp or numeric)	(same as input)	smallest integer not less than argument (alias for ceil)	ceiling(-95.3)	-95
degrees(dp)	dp	radians to degrees	degrees(0.5)	28.6478897565412
exp(dp or numeric)	(same as input)	exponential	exp(1.0)	2.71828182845905

6

3. Mathematical Functions….(cont)

floor(dp or numeric)	(same as input)	largest integer not greater than argument	floor(-42.8)	-43
ln(dp or numeric)	(same as input)	natural logarithm	ln(2.0)	0.693147180559945
log(dp or numeric)	(same as input)	base 10 logarithm	log(100.0)	2
log(b numeric, x numeric)	numeric	logarithm to base b	log(2.0, 64.0)	6.0000000000
mod(y, x)	(same as argument types)	remainder of y/x	mod(9,4)	1
pi()	dp	"π" constant	pi()	3.14159265358979
power(a dp, b dp)	dp	a raised to the power of b	power(9.0, 3.0)	729
power(a numeric, b numeric)	numeric	a raised to the power of b	power(9.0, 3.0)	729

3. Mathematical Functions….(cont)

radians(dp)	dp	degrees to radians	radians(45.0)	0.785398163397448
random()	dp	random value between 0.0 and 1.0	random()	
round(dp or numeric)	(same as input)	round to nearest integer	round(42.4)	42
round(v numeric, s int)	numeric	round to s decimal places	round(42.4382, 2)	42.44
setseed(dp)	void	set seed for subsequent random() calls (value between 0 and 1.0)	setseed(0.54823)	

3. Mathematical Functions….(cont)

sign(dp or numeric)	(same as input)	sign of the argument (-1, 0, +1)	sign(-8.4)	-1
sqrt(dp or numeric)	(same as input)	square root	sqrt(2.0)	1.4142135623731
trunc(dp or numeric)	(same as input)	truncate toward zero	trunc(42.8)	42
trunc(v numeric, s int)	numeric	truncate to s decimal places	trunc(42.4382, 2)	42.43

Trigonometric Functions

Function	Description
acos(x)	inverse cosine
asin(x)	inverse sine
atan(x)	inverse tangent
atan2(y, x)	inverse tangent of y/x
cos(x)	cosine
cot(x)	cotangent
sin(x)	sine
tan(x)	tangent

4. String Functions and Operators

- Types: character, character varying, text

Function	Return Type	Description	Example	Result
string || string	text	String concatenation	'Post' || 'greSQL'	PostgreSQL
string || non-string or non-string || string	text	String concatenation with one non-string input	'Value: ' || 42	Value: 42
bit_length(string)	int	Number of bits in string	bit_length('jose')	32

4. String Functions and…(cont)

char_length(string) or character_length(string)	int	Number of characters in string	char_length('jose')	4
lower(string)	text	Convert string to lower case	lower('TOM')	tom
octet_length(string)	int	Number of bytes in string	octet_length('jose')	5
overlay(string placing string from int [for int])	text	Replace substring	overlay('Txxxxas' placing 'hom' from 2 for 4)	Thomas

4. String Functions and…(cont)

position(substring in string)	Location of specified substring	position('om' in 'Thomas')	3
substring(string [from int] [for int])	Extract substring	substring('Thomas' from 2 for 3)	hom
substring(string from pattern)	Extract substring matching POSIX regular expression. See Section 9.7 for more information on pattern matching.	substring('Thomas' from '...$')	mas

13

4. String Functions and…(cont)

substring(string from *pattern* for *escape*)	text	Extract substring matching SQL regular expression. See Section 9.7 for more information on pattern matching.	substring('Thomas' from '%#"o_a#"_' for '#')	oma
trim([leading | trailing | both] [characters] from string)	text	Remove the longest string containing only the characters (a space by default) from the start/end/both ends of the string	trim(both 'x' from 'xTomxx')	Tom
upper(string)	text	Convert string to uppercase	upper('tom')	TOM

14

Pattern matching

- LIKE

```
'abc' LIKE 'abc'    true
'abc' LIKE 'a%'     true
'abc' LIKE '_b_'    true
'abc' LIKE 'c'      false
```

- SIMILAR

```
'abc' SIMILAR TO 'abc'       true
'abc' SIMILAR TO 'a'         false
'abc' SIMILAR TO '%(b|d)%'   true
'abc' SIMILAR TO '(b|c)%'    false
```

15

5. Data types formatting functions

- text to_char(timestamp/interval/int/double/numeric, text): convert …..to string
- date to_date(text, text): convert string to date
- Numeric to_number(text, text)

16

6. date/time functions

- current_date
- current_time
- date_part(text, timestamp/interval)
- now()
- timeofday()
  - Select date_part('year', current_date);
  - Select Extract(Year from current_date);
  - Select date_part('hours',current_time);
  - Select Extract(Hours from current_time);

Year/Month/Day

H/Hours/hour/M/Minutes/Minute/S/Seconds/Second

17