

abs(x)

Descripció: calcula el valor absolut d'un nombre

Paràmetre x: qualsevol tipus numèric

Retorna: el mateix tipus que x

Exemples:

```
template1=> select abs(5.4);
abs
-----
5.4
(1 row)
```

```
template1=> select abs(-5.4);
abs
-----
5.4
(1 row)
```

age(t)

Descripció: Calcula el temps que ha passat des de la data a mitja nit introduïda fins la data d'avui.

Paràmetre t: timestamp

Retorna: string

Exemples:

```
training=> select age(timestamp '1997-08-07');
age
-----
18 years 5 mons 13 days
(1 row)
```

```
training=> select age(timestamp '1997-12-05');
age
-----
18 years 1 mon 15 days
(1 row)
```

age(timestamp dateA, timestamp dateB)

Descripció: retorna el resultat de la data del primer paràmetre menys la data del segon paràmetre. El format del resultat será en anys (years), messos (mons) i dies (days) si també té un número de dies inferior a 30.

Paràmetre dateA: date

Paràmetre dateB: date

Retorna: interval

Exemples:

```
training=> SELECT age(timestamp '2016-01-20', timestamp '1994-01-20');
age
-----
22 years
(1 row)
```

```
training=> SELECT age(timestamp '2016-01-20', timestamp '1898-12-06');
age
-----
117 years 1 mon 14 days
(1 row)
```

ascii(s)

Descripció: obté el valor ASCII del primer caràcter d'una cadena

Paràmetre s: string

Retorna: int

Exemples:

```
template1=> select ascii('x');
ascii
-----
120
(1 row)

template1=> select ascii('a');
ascii
-----
97
(1 row)

template1=> select ascii('arbre');
ascii
-----
97
(1 row)
```

asin(x)

Descripció: Inverteix un sinus.

Paràmetre x: Radiants (entre -1 i 1)

Retorna: Radiants

Exemples:

```
training=> select asin(-1);
      asin
-----
-1.5707963267949
(1 row)
training=> select asin(1);
      asin
-----
1.5707963267949
```

atan(x)

Descripció: retorna l'angle que té x com a tangent

Paràmetre x: qualsevol tipus numèric

Retorna: double

Exemples:

```
SELECT atan(90);
      atan
-----
1.55968567289729
(1 row)
```

ceil(x)

Descripció: enter immediatament posterior a x

Paràmetre x: double precision o numeric

Retorna: el mateix tipus que x

Exemples:

```
template1=> select ceil(42.8);
 ceil
-----
      43
(1 row)

template1=> select ceil(-42.8);
 ceil
-----
     -42
(1 row)
```

char_length(s)

Descripció: calcula el nombre de caràcters d'una cadena

Paràmetre s: string

Retorna: int

Exemples:

```
template1=> SELECT char_length('hola');
 char_length
-----
           4
(1 row)

template1=> SELECT char_length('hola, com estàs?');
 char_length
-----
          16
(1 row)
```

chr(x)

Descripció: Caràcter amb el codi donat. Per UTF8 l'argument és tractat com un punt de codi UNICODE. Per a altres codificacions multi-byte l'argument ha de designar un caràcter ASCII. La NULL (0) caràcter no està permès perquè els tipus de dades de text no poden emmagatzemar aquests bytes.

Paràmetre x: int

Retorna: text

Exemples:

```
template1=> select chr(65);
chr
-----
A
(1 row)

template1=> select chr(104);
chr
-----
h
(1 row)

template1=> select chr(55);
chr
-----
7
(1 row)
```

concat(s1, s2...)

Descripció: Concatenar les representacions de text de tots els arguments. Els arguments NULL s'ignoren.

Paràmetre s: string

Retorna: text

Exemples:

```
training=> select concat('qwert', 123, NULL, 'asdf');
concat
-----
qwert123asdf
(1 fila)

training=> select concat('abc', NULL, NULL, 'defg');
concat
-----
abcdefg
(1 fila)
```

convert(s, src, dest)

Descripció: Agafa una cadena (primer paràmetre) del codi que sigui (segon paràmetre) i el converteix al codi que vulguis (tercer paràmetre)

Paràmetre s: string bytea,

Paràmetre src: src_encoding name

Paràmetre dest: dest_encoding name

Retorna: bytea

Exemples:

```
template1=> SELECT convert('text_in_utf8', 'UTF8', 'LATIN1');
              convert
```

```
-----
\x746578745f696e5f75746638
(1 row)
```

```
template1=> SELECT convert('Si codificas los numeritos...', 'UTF8', 'LATIN1');
              convert
```

```
-----
\x536920636f64696669636173206c6f73206e756d657269746f732e2e2e
(1 row)
```

```
template1=> SELECT convert('leeras exactamente lo que escribo', 'UTF8', 'LATIN1');
              convert
```

```
-----
\x6c6565726173206578616374616d656e7465206c6f20717565206573637269626f
(1 row)
```

convert_from(s, src)

Descripció: converteix strings en UTF8 i ho converteix en el tipus que té la base de dades per defecte

Paràmetre s: string

Paràmetre src: src_encoding

Retorna: string

Exemples:

```
training=> SELECT convert_from('text_in_utf8', 'UTF8');
convert_from
-----
text_in_utf8
(1 row)
training=> SELECT convert_from('Me va a escribir lo mismo?', 'UTF8');
convert_from
-----
Me va a escribir lo mismo?
(1 row)

training=> SELECT convert_from('Que útil...', 'UTF8');
convert_from
-----
Que útil...
(1 row)
```

convert_to(t,d_e)

Descripció: Converteix qualsevol text a una altra codificació.

Paràmetre t: qualsevol text.

Paràmetre d_e: tipus de codificació que vulguis traduir el text. (dest_encoding)

Retorna: el text traduït al tipus de codificació que li has introduït.

Exemples:

```
training=> select convert_to('Algun text', 'UTF8');  
convert_to
```

```
-----  
\x416c67756e2074657874  
(1 row)
```


cos(x)

Descripció: Calcula el cosinus d'un nombre.

Paràmetre x: qualsevol tipus numèric

Retorna: el mateix tipus que x

Exemples:

```
training=> select cos(8);
cos
-----
-0.145500033808614
(1 row)
```

```
training=> select cos(existencias) from productos;
cos
-----
-0.883877473182372
0.991202811863474
0.955073644047295
1
-0.911130261884677
0.941036507442989
0.717964101410472
-0.989992496600445
0.424179007336997
-0.359072421071653
0.843853958732492
-0.325809805219964
-0.998591672156699
-0.857803093244988
0.857787599307056
0.283662185463226
-0.962605866313567
0.765414051945343
-0.879688592495152
0.843853958732492
-0.962605866313567
0.765414051945343
-0.759687912858821
0.83422336050651
0.136737218207834
(25 rows)
```

current_date

Descripció: Per obtenir-ne la data actual

Retorna: La data actual que tenim al sistema operatiu

Exemples:

```
template1=> SELECT current_date;
      date
-----
 2016-01-20
(1 row)

training=> SELECT current_date AS FechaConsultaTabla, COUNT(num_empl) AS NumeroVendedores FROM repventas;
      fechaconsultatabla | numerovendedores
-----+-----
 2016-01-20              |                10
(1 row)
```

current_time

Descripció: Mostra l'hora actual

Retorna: L'hora actual amb la zona horaria.

Exemples:

```
training=> SELECT current_time;
      timetz
-----
 13:33:02.028845+01
(1 row)
```

date_part(s,i)

Descripció: Mostra la part que tu passis pel paràmetre 's' de l'interval que posis al paràmetre 'i'

Paràmetre s: string

Paràmetre i: string

Retorna: int

Exemples:

```
training=> SELECT date_part('month', interval '2 years 3 months');
date_part
-----
          3
(1 row)

training=> SELECT date_part('month', interval '2 years 4 months');
date_part
-----
          4
(1 row)

training=> SELECT date_part('day', interval '2 years 4 months 5 days');
date_part
-----
          5
(1 row)

training=> SELECT date_part('hour', interval '2 years 4 months 5 days 450 minutes');
date_part
-----
          7

training=> SELECT date_part('minute', interval '2 years 4 months 5 days 450 minutes'
);
date_part
-----
         30
(1 row)
```

date_part(t, timestamp)

Descripció: retorna l'unitat de temps desitjada d'una marca de temps.

Paràmetre t: temps

Paràmetre timestamp: marca de temps

Retorna: doble precisió

Exemple:

```
training=# select date_part('hour', timestamp '2001-02-16 20:38:40');
date_part
-----
         20
(1 row)
```

```
training=# select date_part('minute', timestamp '2001-02-16 20:38:40');
date_part
-----
        38
(1 row)
```

```
training=# select date_part('second', timestamp '2001-02-16 20:38:40');
date_part
-----
        40
(1 row)
```

date_trunc(s, t)

Descripció: Mostra el temps amb el format indicat al primer String('hour' mostra hora, 'minuts' mostra minuts...)

Paràmetre s: string

microseconds

milliseconds

second

minute

hour

day

week

month

quarter

year

decade

century

millennium

Paràmetre t: timestamp

Retorna: timestamp

Exemples:

```
training=> SELECT date_trunc('hour', timestamp '2001-02-02 21:20:43');
           date_trunc
-----
2001-02-02 21:00:00
(1 row)

training=> SELECT date_trunc('minute', timestamp '2001-02-02 21:20:40');
           date_trunc
-----
2001-02-02 21:20:00
(1 row)
training=> SELECT date_trunc('year', timestamp '2015-12-31 21:20:40');
           date_trunc
-----
2015-01-01 00:00:00
(1 row)
```

degrees(x)

Descripció: Conversió de radiants a graus:

Per convertir radiants a graus, fem ús del fet que p radiants és igual a un mitjà cercle, o 180° .

Això vol dir que si dividim radiants per p, la resposta és el nombre de semicercles. Multiplicant això pel 180° ens dirà la resposta en graus.

Per tant, per convertir radiants a graus, multipliqui per $180 / p$, com això:

$\text{degrees} = \text{radians} \times (180/p)$

Paràmetre x: double precision

Retorna: double precision

Exemples:

```
template1=> SELECT degrees(0.5);
           degrees
-----
 28.6478897565412
(1 row)

template1=> SELECT degrees(450.47575);
           degrees
-----
25810.3592479904
(1 row)
```

E

exp(num)

Descripció: Calcula la funció exponencial.

Paràmetre num: qualsevol tipus numèric

Retorna: El resultat de la funció

Exemples:

```
training=> SELECT exp(8);
           exp
-----
2980.95798704173
(1 row)
```

```
training=> SELECT exp(0);
           exp
-----
1
(1 row)
```

extract(f from t)

Descripció: extreu un camp de la data i el temps.

Paràmetre f: field

Paràmatre t: timestamp

Retorna: double precision

Exemples:

```
SELECT EXTRACT(CENTURY FROM TIMESTAMP '2016-01-20') AS "SIGLO";
SIGLO
```

```
-----
```

```
21
```

```
(1 row)
```

```
SELECT EXTRACT(CENTURY FROM TIMESTAMP '1789-07-14');
date_part
```

```
-----
```

```
18
```

```
(1 row)
```

extract(s from interval)

Descripció: mostra per pantalla el parametre s i l'extreu de un interval de temps.

Paràmetre s: nom de unitat de temps(hour, month...)

Parametre interval: una data i/o hora.

Retorna: double precision.

Exemples:

```
select extract(hour from interval '09:23:02');
date_part
```

```
-----
```

```
9
```

format(t1,t2,...)

Descripció: Imprimeix una cadena amb el format que tu li introdueixis.

Paràmetre t1: text

Paràmetre t2: text

Retorna: text

Exemples:

```
training=> SELECT format('Testing %3$s, %2$s, %1$s', 'one', 'two', 'three');
```

```
format
```

```
-----
```

```
Testing three, two, one
```

```
(1 row)
```

```
training=> SELECT format('Testing %2$s, %1$s, %3$s', 'one', 'two', 'three');
```

```
format
```

```
-----
```

```
Testing two, one, three
```

```
(1 row)
```

I

initcap(s)

Descripció: retorna la cadena d'entrada posant la primera lletra de les paraules en la cadena en majúscules i la resta en minúscules. S'utilitzen caràcters no alfanumèrics per separar les paraules ('.', '?', ' ', etc)

Paràmetre s: string

Retorna: text

Exemples:

```
template1=> SELECT initcap('HELLO WORLD!HELLO,hello.123A a123');
```

```
initcap
```

```
-----
```

```
Hello World!Hello,Hello.123a A123
```

```
(1 row)
```

L

left(s, n)

Descripció: retorna els primers n caràcters de la cadena s , en el cas que n sigui negativa, retornara tots els caràcters menys els n caràcters del final.

Paràmetre s: string

Paràmetre n: int

Retorna: string

Exemples:

```
training=> select left('hola mon', 2);
```

```
left
-----
ho
(1 row)
-----
```

```
training=> select left('hola mon', -2);
```

```
left
-----
hola m
(1 row)
```

length(s)

Descripció: obté el nombre de caràcters de la cadena.

Paràmetre s: string

Retorna: int

Exemples:

```
training=> SELECT length(nombre), nombre FROM repventas LIMIT 2;
length |  nombre
-----+-----
      10 | Bill Adams
      10 | Mary Jones
(2 rows)
```

length(s, e)

Descripció: obté el nombre de caràcters de la cadena especificant la codificació.

- **Paràmetre s:** string
- **Paràmetre e:** encoding name

Retorna: int

Exemples:

```
template1=> SELECT length('jose', 'UTF8');
length
-----
         4
(1 row)

template1=> SELECT length('Han Solo', 'UTF8');
length
-----
         8
(1 row)
```

ln(dp,n)

Descripció: logaritme neperià

Paràmetre dp: doble precisió

Paràmetre n: numeric

Retorna: el mateix que l'entrada

Exemples:

```
training=# select ln(1);
ln
----
 0
(1 row)

training=# select ln(43);
ln
-----
3.76120011569356
(1 row)
```

log(n)

Descripció: calcula el logaritme en base 10 del número que li passiis

Paràmetre n: db o numèric

Retorna: double

Exemples:

```
training=> SELECT log(1000.0);
          log
-----
 3.0000000000000000
(1 row)
```

```
training=> SELECT log(10.0);
          log
-----
 1.0000000000000000
(1 row)
```

```
training=> SELECT log(1.0);
          log
-----
 0.0000000000000000
(1 row)
```

```
training=> SELECT log(10982.0);
          log
-----
 4.0406814393733580
(1 row)
```

lower(s)

Descripció: Converteix a minúscules una cadena.

Paràmetre s: string

Retorna: string

Exemples:

```
template1=> select lower('HOLA');
  lower
-----
  hola
(1 row)

template1=> select lower('HoLa, QuE tAl?');
  lower
-----
 hola, que tal?
(1 row)
```

mod (y, x)

Descripció: calcula el modul de dos nombres

Paràmetre x: int

Paràmetre y: int

Retorna: int

Exemples:

```
training=> SELECT mod(10,5);
mod
-----
0
(1 row)
training=> SELECT mod(9,5);
mod
-----
4
(1 row)
training=> SELECT mod(9,4);
mod
-----
1
(1 row)
```

overlay(s1 placing s2 from x1 [for x2])

Descripció: Substitueix una cadena parcialment

Paràmetre s1: string

Paràmetre s2: string

Paràmetre x1: int

Paràmetre x2: int

Retorna: string

Exemples:

```
training=> SELECT overlay('Alxxxt' placing 'ber' from 2 for 4);
overlay
-----
Abert
(1 row)

training=> SELECT overlay('Los perros' placing 'gatos' from 5 for 6);
overlay
-----
Los gatos
(1 row)

training=> SELECT overlay('XXXXXXX' placing 'Angel' from 1);
overlay
-----
AngelXX
(1 row)
```

pi()

Descripció: mostra la constant Pi "π".

Retorna: double precision.

Exemples:

```
SELECT pi();
      pi
-----
3.14159265358979
(1 row)

SELECT pi() / 2 AS "Medio Pi";
      Medio Pi
-----
1.5707963267949
(1 row)
```

position(s1 in s2)

Descripció: Retorna un enter que representa la posició de la cadena de caràcters que s'inclou a la columna donada (o cadena subministrada)

Paràmetre s1: string

Paràmetre s2: string

Retorna: int

Exemples:

```
training=> SELECT position('om' in 'Thomas');  
position
```

```
-----  
          3  
(1 row)
```

```
training=> SELECT position('ol' in 'Thomas');  
position
```

```
-----  
          0  
(1 row)
```

```
training=> SELECT position('a' in 'patata');  
position
```

```
-----  
          2  
(1 row)
```

```
training=> SELECT nombre FROM repventas WHERE position ('o' in nombre ) = 2;  
nombre
```

```
-----  
Bob Smith  
Tom Snyder  
(2 rows)
```


radians(dp)

Descripció: transforma els graus a radiants.

Paràmetre dp: graus

Retorna: radiants

Exemple:

```
select radians(45);
      radians
-----
0.785398163397448
```

replace(s,f,t)

Descripció: Substitueix de la cadena (s) on aparegui la subcadena (f) sera substituïda per la subcadena (t)

- **Paràmetre s:** String text
- **Paràmetre f:**
- o text

Retorna: text

Exemples:

```
template1=> select replace('abcdefabcdef', 'cd', 'XX');
      replace
-----
abXXefabXXef
(1 row)
```

```
template1=> select replace('abcdefabcdef', 'ab', 'YY');
      replace
-----
YYcdefYYcdef
(1 row)
```

```
template1=> select replace('abcdefabcdef', 'f', 'T');
      replace
-----
abcdeTabcdeT
(1 row)
```

reverse(s)

Descripció: retorna la cadena de forma invertida

Paràmetre s: string

Retorna: text

Exemples:

```
training=> select reverse('qwerty');
reverse
-----
ytrewq
(1 fila)

training=> select reverse('12345');
reverse
-----
54321
(1 fila)

training=> select reverse('asdfg');
reverse
-----
gfdsa
(1 fila)
```

round(dp or numeric)

Descripció: Arrodoneix al número enter més proper

Paràmetre x: qualsevol tipus numèric

Retorna: el mateix tipus que x

Exemples:

```
training=> select round(4.12345);
round
-----
4
(1 row)
training=> select round(4.5);
round
-----
5
(1 row)
training=> select round(4.4);
round
-----
4
(1 row)
```

round(number,dec)

round(11,2) devuelve 11.00

Establece cantidad de decimales

rpads1, i[, s2]

Descripció: Agafa la longitud que li indiquis en el segon paràmetre de l'String que has indicat en el primer paràmetre, en cas que l'String del primer paràmetre sigui més petit, omple el que queda amb l'String del tercer paràmetre, repetint-se els cops que faci falta per omplir la longitud del segon paràmetre.

Paràmetre s1: string

Paràmetre i: int

Paràmetre s2: string

Retorna: string

Exemples:

```
template1=> SELECT rpad('hola', 9, 'adios');
           rpad
-----
holaadios
(1 row)

template1=> SELECT rpad('hola', 2, 'adios');
           rpad
-----
ho
(1 row)

template1=> SELECT rpad('hola', 18, 'adios');
           rpad
-----
holaadiosadiosadio
(1 row)

training=> SELECT rpad('hola', 20, '.');
           rpad
-----
hola.....
(1 row)
```

setseed(n)

Descripció: Serveix per assegurar que la funció random retorna nombres aleatoris. el paràmetre pot ser entre -1.0 i 1.0. Cada nombre en aquest rang té la seva propia seqüència fixa de nombres per retornar amb random().

Paràmetre n: un nombre entre -1.0 i 1.0

Retorna:

Exemples:

```
template1=> SELECT setseed(0.5);
```

```
setseed
```

```
-----
```

```
(1 row)
```

-- setseed (0.5) sempre tindrà aquests tres resultats de random() com els tres primers nombres de la seva seqüència

```
template1=> SELECT random();
```

```
random
```

```
-----
```

```
0.798512778244913
```

```
(1 row)
```

```
template1=> SELECT random();
```

```
random
```

```
-----
```

```
0.518533017486334
```

```
(1 row)
```

```
template1=> SELECT random();
```

```
random
```

```
-----
```

```
0.0734698106534779
```

```
(1 row)
```

```
template1=> SELECT setseed(0.3);
```

```
setseed
```

```
-----
```

```
(1 row)
```

```
template1=> SELECT random();
```

```
random
```

```
-----
```

```
0.939957889728248
```

```
(1 row)
```

```
template1=> SELECT random();
```

```
random
```

```
-----
```

```
0.483513379469514
```

```
(1 row)
```

```
template1=> SELECT setseed(0.5);
```

```
setseed
```

```
-----
```

```
(1 row)
```

-- aquí podem veure que després de tornar a posar setseed(0.5), ens retorna els tres mateixos nombres quan fem random() tres vegades.

```
template1=> SELECT random();
```

```
random
```

```
-----
```

```
0.798512778244913
```

```
(1 row)
```

```
template1=> SELECT random();
random
-----
0.518533017486334
(1 row)
template1=> SELECT random();
random
-----
0.0734698106534779
(1 row)
```

sign(x)

Descripció: Calcula si un nombre és positiu o negatiu.

Paràmetre x: qualsevol tipus numèric.

Retorna: el mateix tipus que x.

Exemples:

```
training=> select sign(-655);
sign
-----
-1
(1 row)
```

```
training=> select sign(655);
sign
-----
1
(1 row)
```

split_part(s, d, f)

Descripció: divideix la cadena s amb el delimitador d i et retorna el camp número f

Paràmetre s: string

Paràmetre d: text

Paràmetre f: int

Retorna: text

Exemples:

```
training=> SELECT split_part('Espejo de lo que somos','o', 3); split_part
-----
      que s
(1 row)

training=> SELECT split_part('Espejo de lo que somos','pe', 2);
      split_part
-----
      jo de lo que somos
(1 row)

training=> SELECT split_part('estas montando un espectaculo albert','a', 4); split_p
art
-----
      culo
(1 row)

training=> SELECT split_part('Espejo de lo que somos',' ', 2);
      split_part
-----
      de
(1 row)
```

sqrt(x)

Descripció: raíz cuadrada

Paràmetre x: double precision o qualsevol tipus numèric

Retorna: igual que el de entrada

Exemples:

```
template1=> SELECT sqrt(2.0);
           sqrt
-----
1.414213562373095
(1 row)

template1=> SELECT sqrt(15.55);
           sqrt
-----
3.943348830626071
(1 row)
```

strpos(s,sbs)

Descripció: calcula la posició d'un substring dins d'un string.

Paràmetre s: qualsevol string

Paràmetre sbs: qualsevol substring

Retorna: un enter amb la posició del substring dins l'string

Exemples:

```
training=> select strpos('high', 'ig');
           strpos
-----
2
(1 row)

training=> select strpos('high', 'ab');
           strpos
-----
0
(1 row)
```


substring(s from ptrn for esc)

Descripció:

Paràmetre s: string

Paràmetre ptrn: pattern

Paràmetre esc: escape

Retorna: text

Exemples:

```
template1=> select substring('Thomas' from '%#"o_a#"_' for '#');
 substring
-----
      oma
(1 row)
```

substring(s from ptrn)

Descripció: Retorna caràcters de la cadena s. Retorna el que li especifiqueu en el patró (expressions regulars).

Paràmetre s: string

Paràmetre ptrn: pattern

Retorna: text

Exemples:

```
template1=> select substring('Thomas' from '..');
 substring
-----
      Th
(1 row)

template1=> select substring('Thomas' from '...$');
 substring
-----
      mas
(1 row)
```

substring(s from x1 for x2)

Descripció: Extreu una part de la cadena s. Extreu caràcters des del caràcter en la x1, i extreu tants caràcters com especifica x2.

Paràmetre s: string

Paràmetre x1: integer

Paràmetre x2: integer

Retorna: text

Exemples:

```
template1=> select substring('Thomas' from 2 for 2);
```

```
substring
```

```
-----
```

```
ho
```

```
(1 row)
```

```
template1=> select substring('Thomas' from 4 for 3);
```

```
substring
```

```
-----
```

```
mas
```

```
(1 row)
```

```
template1=> select substring('Thomas' from 2);
```

```
substring
```

```
-----
```

```
homas
```

```
(1 row)
```

```
template1=> select substring('Thomas' for 2);
```

```
substring
```

```
-----
```

```
Th
```

```
(1 row)
```

T

tan(number)

tan(number) es la funcion trigonometrica de tangente. Recibe un parametro tipo double en **radianes**. Devuelve un numero.

to_char(d, s)

Descripció: Converteix un nombre real a un String amb el format que tu li passis com a paràmetre.

Paràmetre d: double

Paràmetre s: string

Retorna: string

Exemples:

```
training=> select to_char(125.8, '999D9');
to_char
-----
    125.8
(1 row)

training=> select to_char(125.8, '999D999');
to_char
-----
    125.800
(1 row)

training=> select to_char(125.8, '0999D999');
to_char
-----
   0125.800
(1 row)

training=> select to_char(125.8, '99D9');
to_char
-----
    ##.##
(1 row)
```

to_char(i, t)

Descripció: converteix un nombre a una cadena amb format.

Paràmetre i: numeric

Paràmetre t: text

Retorna: text

Exemple:

```
select to_char(125.9, '999');
to_char
-----
    126
(1 row)
```

```
select to_char(125.9, '999D9');
to_char
-----
   125.9
(1 row)
```

```
select to_char(125.4867575855, '0999');
to_char
-----
   0125
(1 row)
```

to_char(intv,txt)

Descripció: Agafa un interval de temps, i t'ho retorna com una cadena amb format. La sortida és en el format que vulguis. HH12/24 per les hores, MI pels minuts, SS pels segons, Day per al dia, Month pel mes. Els separadors poden ser :, /, ...

Paràmetre intv: interval

Paràmetre txt: text

Retorna: Els camps que vulguis en el format que vols

Exemples:

```
SELECT to_char(interval '15h 2m 12s', 'HH24:MI:SS');
to_char
-----
15:02:12
(1 row)
```

to_char(tmstp,txt)

Descripció: Converteix la "marca de temps" a string en el format que escullis.

Paràmetre tmstp: timestamp (current_date, current_timestamp)

Paràmetre txt: text en el format que vulguis de sortida. HH12/24 per les hores (12 o 24 depenent de com la vulguis), MI pels minuts, SS pels segons, Day per al dia, Month pel mes. Els separadors poden ser: /, :, -, ...

Retorna: Els camps que hagi escollit en el format que escullis.

Exemples:

```
training=> SELECT to_char(current_timestamp, 'HH24:MI:SS');
to_char
-----
13:25:38
(1 row)

training=> SELECT to_char(current_timestamp, 'Month Day HH24:MI:SS');
to_char
-----
January Wednesday 13:27:57
(1 row)
```

to_char(x, s)

Descripció: converteix els numèrics a String canviant el format

Paràmetre x: qualsevol tipus numèric

Paràmetre s: String

Retorna: String

Exemples:

```
training=> SELECT to_char(123456789, '999999999D99');
           to_char
-----
123456789.00
(1 row)
training=> SELECT to_char(1234567.89, '999999999D99');
           to_char
-----
1234567.89
(1 row)
training=> SELECT to_char(1234567.89, '999999999D99L');
           to_char
-----
1234567.89$
(1 row)

training=> SELECT to_char(123, '999S');
           to_char
-----
123+
(1 row)

training=> SELECT to_char(-123, '999S');
           to_char
-----
123-
(1 row)
SELECT to_char(12323, '999PL99');
           to_char
-----
123+23
(1 row)
training=> SELECT to_char(12323, '999MI99');
           to_char
-----
123 23
(1 row)
```

to_date(t1, t2)

Descripció: converteix un *string* a *date*.

Paràmetre t1: *text*. Data que vols convertir a *date*.

Paràmetre t2: *text*. Format de la data

Retorna: date.

Exemples:

```
SELECT to_date('12 Apr 1', 'YY Mon DD');
to_date
-----
2012-04-01
(1 row)

SELECT to_date('05 Dec 2000', 'DD Mon YYYY');
to_date
-----
2000-12-05
(1 row)
```

to_number('s','n')

Descripció: converteix una cadena(s) en un numero(n) en el format escollit.

Paràmetre s: string.

Paràmetre n: format del numero final.

Exemple:

```
SELECT TO_NUMBER('12,4bc54.8-', '99999999D9S');
to_number
-----
-12454.8

-----

SELECT TO_NUMBER('12,454.8-', '999999D9S');
to_number
-----
-12454.8
(1 row)

-----

SELECT TO_NUMBER('12,454.8-', '999999D9');
to_number
-----
12454.8
(1 row)
```


to_timestamp(double precision)

Descripció: Converteix unix epoch en timestamp.

Paràmetre: Unix epoch (segons passats desde UTC (Coordinated Universal Time) '1970-1-1')

Retorna: Timestamp with time zone.

Exemples:

```
training=> select to_timestamp(1284352323);
to_timestamp
-----
2010-09-13 06:32:03+02
(1 row)

\q
[iaw47951368@h23 ~]$ date +%s
1453293311
training=> select to_timestamp(1453293311);
to_timestamp
-----
2016-01-20 13:35:11+01
(1 row)
```

translate(s, t_f, t_t)

Descripció: Reemplaça els caràcters del primer paràmetre (s) que són iguals als caràcters del segon paràmetre (t_f) amb els caràcters del tercer paràmetre (t_t). Si el segon paràmetre és més llarg que el tercer, s'eliminarà els caràcters del primer paràmetre que són iguals als caràcters addicionals del segon paràmetre.

Paràmetre s: string

Paràmetre t_f: string (caràcters per buscar en el primer paràmetre)

Paràmetre t_t: string (caràcters que van a reemplaçar els caràcters trobats en el primer paràmetre)

Retorna: text

Exemples:

```
template1=> SELECT translate('template','pla','123');
 translate
-----
 tem123te
(1 row)
```

```
template1=> SELECT translate('template','apl','123');
 translate
-----
 tem231te
(1 row)
```

```
template1=> SELECT translate('template','aplt','123');
 translate
-----
 em231e
(1 row)
```

trim([leading | trailing | both] [characters] from string)

Descripció: borra caràcters del començament, final o ambdós costats de la cadena, per defecte borra els espais en blanc sinó indiquem el caràcter o cadena.

Pàrametres:

leading: elimina els espais en blanc o caràcters del principi.

trailing: elimina els espais en blanc o caràcters del final .

both: elimina els espais en blanc o caràcters de ambdós costats.

characters = caràcters

string = cadena

Retorna: text

Exemples:

```
select trim(both 'x' from 'xTomxx');
      btrim
-----
      Tom
(1 row)
```

```
-----
select trim(leading 'x' from 'xTomxx');
      ltrim
-----
      Tomxx
(1 row)
```

```
-----
select trim(trailing 'x' from 'xTomxx');
      rtrim
-----
      xTom
(1 row)
```

trunc(number,[dec])

trunc(number,[dec]) corta la parte decimal, o hasta los decimales especificados.

Ej: trunc(42.8365) devuelve 42

trunc(42.8365,2) devuelve 42.83

Nota: Su enemigo natural es round(number,[dec])

upper(s)

Descripció: converteix les lletres de una cadena en lletres majúscules.

Paràmetre s: string

retorna: string

Exemples:

```
training=> select upper('hola');
```

```
upper
```

```
-----
```

```
  HOLA
```

```
(1 row)
```

```
-----
```

```
training=> select upper('hola mon');
```

```
upper
```

```
-----
```

```
  HOLA MON
```

```
(1 row)
```