

# Funciones SQL

Antonio Espín Herranz

# Funciones integradas

- Fecha / Hora
- Matemáticas
- Cadenas
- Formatear tipos
- Expresiones y funciones condicionales
- Otros tipos de funciones: Arrays, XML, Json

# Funciones SQL

- Para ejecutar una función SQL de postgres lo podemos hacer dentro de una sentencia de SQL o en un bloque de pl/pgsql.
- La sintaxis para ejecutar sólo la función: **select nombre\_función()**
- Las funciones están organizadas en tipos.
- También se pueden añadir nuevas funciones programadas en pl/pgsql que se graben en la BD.
- Las funciones se pueden utilizar tanto en la parte de los campos como en las condiciones o para agrupar y las condiciones de grupo.

# Funciones de fecha / hora

- **Current\_date**
  - La fecha actual
- **Current\_timestamp**
  - La fecha y la hora actual
- **Current\_time**
  - La hora actual
- **Extract(valor from timestamp)**
  - Valor representa una palabra clave que indica la parte del timestamp que queremos obtener.
  - Year, month, day, hour, minute, second

# Ejemplos

- `select extract(year from timestamp'2009-12-31 12:25:50');`
- Retorna el año '2009'
  
- `select extract(month from timestamp'2009-12-31 12:25:50');`
- Retorna el mes '12'
  
- `select extract(day from timestamp'2009-12-31 12:25:50');`
- Retorna el día '31'
  
- `select extract(hour from timestamp'2009-12-31 12:25:50');`
- Retorna la hora '12'
  
- `select extract(minute from timestamp'2009-12-31 12:25:50');`
- Retorna el minuto '25'
  
- `select extract(second from timestamp'2009-12-31 12:25:50');`
- Retorna el segundo '50'

# Funciones de fecha / hora

- Similar a la función extract, tenemos: date\_part
- Select **date\_part**('month',e.fechaentrega) ...
- Para imprimir el nombre del día o del mes en español:
- **set lc\_time to 'es\_ES.utf8';** -- Indicar el locale del país.
- select to\_char(now(), '**TMDay, DD TMMonth YYYY**');
- select to\_char(current\_date, '**TMMonth**');

# Funciones de fecha / hora

- **Now()**
  - Devuelve un timestamp de la fecha / hora actual.
- Las fechas también soportan operaciones de suma y resta de un entero:
  - `select current_date+5, current_date-5;`
- Se pueden indicar intervalos:
  - `Select current_date + Interval '1 hour';`
- **Listado completo de funciones y operaciones:**
- <https://www.postgresql.org/docs/16/functions-datetime.html>

# Detener la ejecución

- `pg_sleep ( double precision )`
  - `pg_sleep_for ( interval )`
  - `pg_sleep_until ( timestamp with time zone )`
- 
- `SELECT pg_sleep(1.5);`
  - `SELECT pg_sleep_for('5 minutes');`
  - `SELECT pg_sleep_until('tomorrow 03:00');`



# Matemáticas

- **Abs(x)**
  - El valor absoluto de x. `Select abs(-20);`
- **Cbrt(x)**
  - La raíz cuadrada de x. `Select cbrt(27)`
- **Ceiling(x)**
  - Redondeo al alza: `select ceiling(12.34) → 13`
- **Floor(x)**
  - Redondeo a la baja
  - `Select floor(12.34) → 12`

# Matemáticas

- **Power(x, y)**
  - Potencia de x elevado a y
  - Select power(2,3)  $\rightarrow$  8
- **Round(numero)**
  - Redondea el número al valor más próximo
  - Select round(12.4)  $\rightarrow$  12
- **Sign(numero)**
  - Devuelve el signo del número: -1 es menor que 0, 0 si es 0, 1 es mayor que 0.

# Matemáticas

- **Sqrt(x):**

- Devuelve la raíz cuadrada de un número.
- `Select sqrt(9) → 3`

- **Mod(x, y)**

- Devuelve el resto de dividir x con respecto a y
- `Select mod(11,2) → 1`

- **Pi()**

- Devuelve el valor de pi
- El valor del número de pi.

# Matemáticas

- **Random()**
  - Devuelve un número aleatorio entre 0 y 1.
  - `Select random()`
- **Trunc(x):** Retorna la parte entera del parámetro
  - `Select trunc(34.7);`
  - Retorna 34
- **Trunc(x, decimales)**
  - Trunca con el número decimales indicado
  - `Select trunc(34.77777, 2) → 34.77`

# Matemáticas

- Funciones trigonométricas: sin, cos, tan
- Select **sin(x)**, **cos(x)**, **tan(x)**

# Cadenas

- Funciones que se aplican a cadenas de texto.
- **char\_length(string)**
  - Retorna la longitud de la cadena.
  - `Select char_length('hola') → 4`
- **Upper(string)**
  - Retorna la cadena en mayúsculas
  - `Select upper('hola') → 'HOLA'`
- **Lower(string)**
  - Retorna la cadena en minúsculas
  - `Select lower('HOLA') → 'hola'`

# Cadenas

- **Position**(string in string)
  - Retorna la posición de un string dentro de otro. Si no está contenido retorna un 0. Ejemplo:
  - `select position('Mundo' in 'Hola Mundo');`
  - retorna 6.
  - `select position('MUNDO' in 'Hola Mundo');`
  - retorna 0 (ya que no coinciden mayúsculas y minúsculas)
- **Substring**(string [from int][for int])
  - Retorna un substring, le indicamos la posición inicial y la cantidad de caracteres a extraer desde dicha posición. Ejemplo:
  - `select substring('Hola Mundo' from 1 for 2);`
  - retorna 'Ho'.
  - `select substring('Hola Mundo' from 6 for 5);`
  - retorna 'Mundo'.

# Cadenas

- **trim**([leading|trailing|both] [string] from string): Elimina caracteres del principio o o final de un string. Por defecto elimina los espacios en blanco si no indicamos el caracter o string. Ejemplo:
- `select char_length(trim(' Hola Mundo '));`
- retorna un 10. Esto es debido a que primero se ejecuta la función trim que elimina los dos espacios iniciales y los dos finales.
- `select char_length(trim(leading ' ' from ' Hola Mundo '));`
- retorna un 12. Esto es debido a indicamos que elimine los espacios en blanco de la cadena solo del comienzo (leading).
- `select trim(trailing '-' from '--Hola Mundo----');`
- retorna '--Hola Mundo'. Esto es debido a indicamos que elimine los guiones del final del string.



# Cadenas

- **ltrim(string,string)**: Elimina los caracteres de la izquierda según el dato del segundo parámetro de la función. Ejemplo:
- `select char_length(ltrim(' Hola'));`
- retorna un 4.
- `select ltrim('---Hola','-');`
- retorna 'Hola'.

# Cadenas

- **rtrim**(string,string): Elimina los caracteres de la derecha según el dato del segundo parámetro de la función. Ejemplo:
- `select char_length(rtrim('Hola '));`
- retorna un 4.
- `select rtrim('Hola----','-');`
- retorna un 'Hola'.

# Cadenas

- **substr**(text,int[,int]): Retorna una subcadena a partir de la posición que le indicamos en el segundo parámetro hasta la posición indicada en el tercer parámetro. Ejemplo:
- `select substr('Hola Mundo',2,4);`
- retorna 'ola'.
- `select substr('Hola Mundo',2);`
- retorna 'ola Mundo' (si no indicamos el tercer parámetro retorna todo el string hasta el final)

# Cadenas

- **lpad**(text,int,text): Rellena de caracteres por la izquierda. El tamaño total de campo es indicado por el segundo parámetro y el texto a insertar se indica en el tercero. Ejemplo:
  - `select lpad('Hola Mundo',20,'-');`
  - retorna `'-----Hola Mundo'`.
- **rpadd**(text,int,text): Rellena de caracteres por la derecha. El tamaño total de campo es indicado por el segundo parámetro y el texto a insertar se indica en el tercero. Ejemplo:
  - `select rpadd('Hola Mundo',20,'-');`
  - retorna `'Hola Mundo-----'`.
- `select rpadd('Hola Mundo',20,'-*');`
- retorna `'Hola Mundo-*-*-*-*'`.

# Formatear tipos

- Disponemos de funciones para convertir a **texto**, a **número** o **fecha**.
- <https://www.postgresql.org/docs/16/functions-formatting.html>
- A estas funciones se les indica un formato
  - **To\_char, to\_date, to\_number, to\_timestamp**
- Formatos y ejemplos de las funciones:
  - `to_char(125, '999') → 125`
  - `to_char(125.8::real, '999D9') → 125.8`
  - `to_char(-125.8, '999D99S') → 125.80-`
  - `to_date( text, text ) → date`
  - `to_date('05 Dec 2000', 'DD Mon YYYY') → 2000-12-05`

# Formatear tipos

- `to_number ( text, text )` → numeric
- `to_number('12,454.8-', '99G999D9S')` → -12454.8
- `to_timestamp ( text, text )` → timestamp with time zone
- `to_timestamp('05 Dec 2000', 'DD Mon YYYY')` → 2000-12-05 00:00:00-05

# Formatos fecha/hora

Pattern	Description
HH	hour of day (01–12)
HH12	hour of day (01–12)
HH24	hour of day (00–23)
MI	minute (00–59)
SS	second (00–59)
MS	millisecond (000–999)
US	microsecond (000000–999999)
FF1	tenth of second (0–9)
FF2	hundredth of second (00–99)
FF3	millisecond (000–999)
FF4	tenth of a millisecond (0000–9999)
FF5	hundredth of a millisecond (00000–99999)
FF6	microsecond (000000–999999)
SSSS, SSSSS	seconds past midnight (0–86399)
AM, am, PM or pm	meridiem indicator (without periods)
A.M., a.m., P.M. or p.m.	meridiem indicator (with periods)
Y, YYYY	year (4 or more digits) with comma

# Formatos fecha/hora

YYYY	year (4 or more digits)
YYY	last 3 digits of year
YY	last 2 digits of year
Y	last digit of year
IYYY	ISO 8601 week-numbering year (4 or more digits)
IYY	last 3 digits of ISO 8601 week-numbering year
IY	last 2 digits of ISO 8601 week-numbering year
I	last digit of ISO 8601 week-numbering year
BC, bc, AD or ad	era indicator (without periods)
B.C., b.c., A.D. or a.d.	era indicator (with periods)
MONTH	full upper case month name (blank-padded to 9 chars)
Month	full capitalized month name (blank-padded to 9 chars)
month	full lower case month name (blank-padded to 9 chars)
MON	abbreviated upper case month name (3 chars in English, localized lengths vary)
Mon	abbreviated capitalized month name (3 chars in English, localized lengths vary)
mon	abbreviated lower case month name (3 chars in English, localized lengths vary)
MM	month number (01–12)
DAY	full upper case day name (blank-padded to 9 chars)



# Formatos fecha/hora

Day	full capitalized day name (blank-padded to 9 chars)
day	full lower case day name (blank-padded to 9 chars)
DY	abbreviated upper case day name (3 chars in English, localized lengths vary)
Dy	abbreviated capitalized day name (3 chars in English, localized lengths vary)
dy	abbreviated lower case day name (3 chars in English, localized lengths vary)
DDD	day of year (001–366)
IDDD	day of ISO 8601 week-numbering year (001–371; day 1 of the year is Monday of the first ISO week)
DD	day of month (01–31)
D	day of the week, Sunday (1) to Saturday (7)
ID	ISO 8601 day of the week, Monday (1) to Sunday (7)
W	week of month (1–5) (the first week starts on the first day of the month)
WW	week number of year (1–53) (the first week starts on the first day of the year)
IW	week number of ISO 8601 week-numbering year (01–53; the first Thursday of the year is in week 1)
CC	century (2 digits) (the twenty-first century starts on 2001-01-01)
J	Julian Date (integer days since November 24, 4714 BC at local midnight; see <a href="#">Section B.7</a> )
Q	quarter
RM	month in upper case Roman numerals (I–XII; I=January)
rm	month in lower case Roman numerals (i–xii; i=January)

# Formatos fecha/hora

rm	month in lower case Roman numerals (i-xii; i=January)
TZ	upper case time-zone abbreviation (only supported in to_char)
tz	lower case time-zone abbreviation (only supported in to_char)
TZH	time-zone hours
TZM	time-zone minutes
OF	time-zone offset from UTC (only supported in to_char)

# Expresiones y funciones condicionales

- **Case**

- Se puede aplicar en un bloque de código o dentro de un SQL.

```
CASE WHEN condition THEN result  
  [WHEN ...]  
  [ELSE result]  
END
```

```
SELECT a,  
  CASE WHEN a=1 THEN 'one'  
        WHEN a=2 THEN 'two'  
        ELSE 'other'  
  END  
FROM test;
```

# Expresiones y funciones condicionales

- **Coalesce**(value, ...)
  - Devuelve el primer valor que no sea nulo, de todos los argumentos que le pasamos.
- **Nullif**(value1, value2)
  - Devuelve un valor nulo si value1 es igual a value2.
- **Greatest**(valor1, valor2, ..., valorN):
  - Devuelve el valor más grande
- **Least**(valor1, valor2, ... valorN):
  - Devuelve el valor más pequeño de la lista

# Otros tipos de funciones:

- Arrays:
  - <https://www.postgresql.org/docs/16/functions-array.html>
- Json:
  - <https://www.postgresql.org/docs/16/functions-json.html>
- XML:
  - <https://www.postgresql.org/docs/16/functions-xml.html>

# Enlaces

- Documentación oficial: funciones
  - <https://www.postgresql.org/docs/16/functions.html>
- Otros enlaces:
  - <https://www.tutorialesprogramacionya.com/postgresqlya/temarios/descripcion.php?cod=181&punto=23&inicio=>
  - <https://www.postgresql.org/docs/16/functions-datetime.html>