

FUNCIONES DE LIBRERIA

1.- Formatos

1.01.- Numérico: `To_Char(Campo_Número, '999G999G999D99')`

Toma el contenido del campo numérico y lo convierte en una salida de formato numérico de tipo 999,999,999.99

```
SELECT Campo1, ..., to_char( Campo_Número, '999G999G999D99' )  
FROM tabla
```

```
WHERE Condición – criterio ORDER BY Campo1, CampoN... .;
```

Ejemplo 01: Se desea mostrar todos los registros de la tabla `tmcargos`, donde el sueldo se debe presentar con el formato numérico 999.999.999,99

```
SELECT pkcodcar, dcargo, to_char(sueldo, '999G999G999D99' ) AS "Sueldo"  
FROM tmcargos ;
```

```
bdretardados=# select * from tmcargos;  
pkcodcar | dcargo | sueldo | fkcds  
-----  
1 | PRESIDENTE | 5000000.01 | 1  
2 | VICE PRESIDENTE | 4500000.02 | 0  
3 | GERENTE | 4000000.03 | 1  
4 | SUB-GERENTE | 3500000.04 | 0  
5 | COORDINADOR | 3000000.05 | 1  
6 | SUPERVISOR | 2500000.06 | 0  
(6 filas)
```

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, '999G999G999D99' ) AS "Sueldo"  
bdretardados=# FROM tmcargos ;  
pkcodcar | dcargo | Sueldo  
-----  
1 | PRESIDENTE | 5.000.000,01  
2 | VICE PRESIDENTE | 4.500.000,02  
3 | GERENTE | 4.000.000,03  
4 | SUB-GERENTE | 3.500.000,04  
5 | COORDINADOR | 3.000.000,05  
6 | SUPERVISOR | 2.500.000,06  
(6 filas)
```

Nota (GD): Las unidades de miles, millardos... con punto(.); el separador decimal (,)

Ejemplo 02: Se desea mostrar todos los campos y los registros de la tabla `tmcargos`, donde el sueldo se debe mostrar con el formato numérico de moneda así `999,999,999.99`

```
bdretardados=# select * from tmcargos;
pkcodcar |      dcargo      |      sueldo      | fkcods
-----+-----+-----+-----
1 | PRESIDENTE | 5000000.01 | 1
2 | VICE PRESIDENTE | 4500000.02 | 0
3 | GERENTE | 4000000.03 | 1
4 | SUB-GERENTE | 3500000.04 | 0
5 | COORDINADOR | 3000000.05 | 1
6 | SUPERVISOR | 2500000.06 | 0
(6 filas)
```

```
SELECT pkcodcar, dcargo, to_char( sueldo, 'FM999,999,999.99' ) AS "Sueldo"
FROM tmcargos ;
```

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, 'FM999,999,999.99' ) AS "Sueldo"
bdretardados=# FROM tmcargos ;
pkcodcar |      dcargo      |      Sueldo
-----+-----+-----
1 | PRESIDENTE | 5,000,000.01
2 | VICE PRESIDENTE | 4,500,000.02
3 | GERENTE | 4,000,000.03
4 | SUB-GERENTE | 3,500,000.04
5 | COORDINADOR | 3,000,000.05
6 | SUPERVISOR | 2,500,000.06
(6 filas)
```

Nota (FM): Las unidades de miles, millardos... con coma(,); el separador decimal punto(.)

Ejemplo 03: Se desea mostrar la sumatoria de todos los pagos hecho por el estudiante cuya cédula es '1000'. Donde el sueldo se debe mostrar con el **formato numérico de moneda** así 999,999,999.99

```
SELECT to_char( SUM(monto), 'FM999,999,999.99' ) AS "Monto Pagado:"
FROM tdpagos
```

WHERE fkcedalum = '1000' ; --> Monto Pagado: 9,000,000.00

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, 'FM999,999,999.99' ) AS
"Sueldo"
bdretardados=# FROM tmcargos ;
```

pkcodcar	dcargo	Sueldo
1	PRESIDENTE	5,000,000.01
2	VICE PRESIDENTE	4,500,000.02
3	GERENTE	4,000,000.03
4	SUB-GERENTE	3,500,000.04
5	COORDINADOR	3,000,000.05
6	SUPERVISOR	2,500,000.06

(6 filas)

```
bdretardados=# select * from tmcargos;
pkcodcar | dcargo | sueldo | fkcds
```

pkcodcar	dcargo	sueldo	fkcds
1	PRESIDENTE	5000000.01	1
2	VICE PRESIDENTE	4500000.02	0
3	GERENTE	4000000.03	1
4	SUB-GERENTE	3500000.04	0
5	COORDINADOR	3000000.05	1
6	SUPERVISOR	2500000.06	0

(6 filas)

Nota (FM): Las unidades de miles, millardos... con coma(,); el separador decimal punto(.)

Ejemplo 04:

```
SELECT pkcodcar, dcargo, to_char( sueldo, 'L999G999G999D99' ) AS "Sueldo"
```

FROM tmcargos ;

```
bdretardados=# SELECT pkcodcar, dcargo, to_char( sueldo, 'L999G999G999D99' ) AS "Sueldo"
bdretardados=# FROM tmcargos ;
```

pkcodcar	dcargo	Sueldo
1	PRESIDENTE	\$ 5.000.000,01
2	VICE PRESIDENTE	\$ 4.500.000,02
3	GERENTE	\$ 4.000.000,03
4	SUB-GERENTE	\$ 3.500.000,04
5	COORDINADOR	\$ 3.000.000,05
6	SUPERVISOR	\$ 2.500.000,06

(6 filas)

Nota (LG): L Símbolo de pesos \$; Las unidades de miles, millardos... con punto(.); el separador decimal coma (,)

1.02.- Fecha: to_char(DATE 'YYYY/MM/DD' / Campo_Fecha , 'DD/MM/YYYY')

Toma el contenido del campo fecha y lo convierte en el formato fecha DD/MM/YYYY.

```
SELECT Campo1, ..., to_char( DATE 'fecha' / Campo_Fecha , 'DD/MM/YYYY' )  
FROM tabla
```

```
WHERE Pregunta1, ..., PreguntaN ORDER BY Campo1, CampoN... ;
```

Ejemplo01: Se desea mostrar todos los registros de la templeados, donde la fecha se debe mostrar en formato DD/MM/YYYY

```
SELECT pkcedemple, nomemple, to_char( fecha, 'DD/MM/YYYY' ) As "Fecha"  
FROM templeados ;
```

```
bdretardados=# select pkcedemple, nomemple, fecha  
bdretardados=# from templeados;  
pkcedemple | nomemple | fecha  
-----  
1000 | ROBERTO JAIMES | 1978-03-03  
2000 | ZILA CONTRERAS | 1980-10-05  
3000 | MARTHA | 1996-11-22  
4000 | CARLOS | 2000-07-15  
5000 | MATIAS | 2005-09-25  
6000 | NATASHA | 2007-10-28  
7000 | PEGGY CARTER | 1999-08-29  
8000 | YSA CAPACHO | 1993-07-23  
9000 | MARIA MORALES | 2003-07-07  
17 | VALERIA | 1981-04-07  
13 | ANGELY | 1978-06-21  
7 | SERGIO CAPACHO | 1967-11-22  
4020 | YSAURA | 2018-02-05  
4017 | MESSI | 1996-06-09  
4011 | RONALDO | 2011-12-25  
9057 | CARMENZA | 1988-08-09  
9026 | FANNY | 2005-07-10  
10000 | MERCY | 2013-07-07  
(18 filas)
```

```
bdretardados=# SELECT pkcedemple, nomemple,  
bdretardados=# to_char( fecha, 'DD/MM/YYYY' ) As "Fecha"  
bdretardados=# FROM templeados;  
pkcedemple | nomemple | Fecha  
-----  
1000 | ROBERTO JAIMES | 03/03/1978  
2000 | ZILA CONTRERAS | 05/10/1980  
3000 | MARTHA | 22/11/1996  
4000 | CARLOS | 15/07/2000  
5000 | MATIAS | 25/09/2005  
6000 | NATASHA | 28/10/2007  
7000 | PEGGY CARTER | 29/08/1999  
8000 | YSA CAPACHO | 23/07/1993  
9000 | MARIA MORALES | 07/07/2003  
17 | VALERIA | 07/04/1981  
13 | ANGELY | 21/06/1978  
7 | SERGIO CAPACHO | 22/11/1967  
4020 | YSAURA | 05/02/2018  
4017 | MESSI | 09/06/1996  
4011 | RONALDO | 25/12/2011  
9057 | CARMENZA | 09/08/1988  
9026 | FANNY | 10/07/2005  
10000 | MERCY | 07/07/2013  
(18 filas)
```

SELECT to_char(DATE '22/11/1967', 'MM/YYYY/DD') AS fecha ;

```
bdretardados=# SELECT to_char( DATE '22/11/1967', 'MM/YYYY/DD') AS fecha;
 fecha
-----
 11/1967/22
(1 fila)
```

SELECT to_char(DATE '22/11/1967', 'Mon/YYYY/DD') AS fecha ;

```
bdretardados=# SELECT to_char( DATE '22/11/1967', 'Mon/YYYY/DD') AS fecha;
 fecha
-----
Nov/1967/22
(1 fila)
```

SELECT (NOW() - DATE '1967/11/22') AS "Dias Transcurridos" ;

SELECT to_char(CURRENT_TIMESTAMP, 'HH24:MI:SS') ; --> Hora formato 24H

SELECT to_char(CURRENT_TIMESTAMP, 'HH12:MI:SS AM') ; Hora formato 12H AM o PM

SELECT to_char(CURRENT_TIMESTAMP, 'DD-MM-YYYY') ; --> Fecha formato Dia/Mes/Año

SELECT to_char(CURRENT_TIMESTAMP, 'DD-MON-YYYY HH12:MI:SS AM') ;

Mon-> mes tipo Ene(Jan)/Feb/Mar....

SELECT to_char(date '1967-11-22', 'DD-Mon-YYYY H12:MI:SS AM') ;

--> Formato 22-Nov-1967 H12:00:00 AM

2.- Funciones de fecha

2.1.- `current_date`: retorna la fecha actual del servidor.

```
SELECT current_date;
```

```
bdretardados=# SELECT current_date;
current_date
-----
2025-03-13
(1 fila)
```

2.2.- `current_time`: retorna la hora actual del servidor.

```
SELECT current_time;
```

```
bdretardados=# SELECT current_time;
current_time
-----
21:44:50.148208-05
(1 fila)
```

2.3.- `current_timestamp`: retorna la fecha y la hora actual del servidor.

```
SELECT current_timestamp; / SELECT NOW();
```

```
bdretardados=# SELECT current_timestamp;
current_timestamp
-----
2025-03-13 21:46:31.992633-05
(1 fila)
```

2.4.- extract(**Parametro** from timestamp 'Dato_Fecha_Hora'): retorna una parte de la fecha / hora según el parámetro que se indiquemos antes de la clausula from. Y después del from se debe iniciar la clausula timestamp (fecha / hora).

Ejemplo 01: Retorna el año actual del servidor. ('1967')

```
SELECT extract( Year from timestamp '1967-11-22 12:25:50' );
```

```
SELECT extract( Year from current_timestamp );
```

Ejemplo 02: Retorna el mes actual del servidor. ('11' Noviembre)

```
SELECT extract( Month from timestamp '1967-11-22 12:25:50' );
```

```
SELECT extract( Month from current_timestamp );
```

Ejemplo 03: Retorna el día actual del servidor. ('22')

```
SELECT extract( Day from timestamp '1967-11-22 12:25:50' );
```

```
SELECT extract( Day from current_timestamp );
```

Ejemplo 04: Retorna la hora actual del servidor. ('12')

Nota: las horas se expresan en formato de hora militar; es decir de 0 a 23 horas.

```
SELECT extract( Hour from timestamp '1967-11-22 12:25:50' );
```

```
SELECT extract( Hour from current_timestamp );
```

Ejemplo 05: Retorna el minuto actual del servidor. ('25')

```
SELECT extract( Minute from timestamp '1967-11-22 12:25:50' ) ;
```

```
SELECT extract( Minute from current_timestamp ) ;
```

Ejemplo 06: Retorna el segundo actual del servidor. ('50')

```
SELECT extract( Second from timestamp '1967-11-22 12:25:50' ) ;
```

```
SELECT extract( Second from current_timestamp ) ;
```

Ejemplo 07: Retorna la cantidad de milisegundos actuales del servidor

```
SELECT extract( MILLISECONDS from timestamp '1967-11-22 12:25:50' ) ;
```

```
SELECT extract( MILLISECONDS from current_timestamp ) ;
```

Ejemplo 08: Retorna el siglo actual del servidor. ('20')

```
SELECT extract( Century from timestamp '1967-11-22 12:25:50' ) ;
```

```
SELECT extract( Century from current_timestamp ) ;
```

Ejemplo 09: Retorna el día de la semana actual.

Nota: 0 -> Domingo / 1 -> Lunes / 2 -> Martes

```
SELECT extract( Dow from timestamp '1967-11-22 12:25:50') ;
```

```
SELECT extract( Dow from current_timestamp) ;
```

Ejemplo 10: Retorna la cantidad de días transcurridos del año

```
SELECT extract( Doy from timestamp '1967-11-22 12:25:50') ;
```

```
SELECT extract( Doy from current_timestamp) ;
```

Ejemplo 11: Retorna el número de semanas transcurridos del año'

```
SELECT extract( Week from timestamp '1967-11-22 12:25:50') ;
```

```
SELECT extract( Week from current_timestamp) ;
```

Ejemplo 12: Retorna el número del trimestre del año donde se ubica la fecha.

```
SELECT extract( quarter from timestamp '1967-11-22 12:25:50') ;
```

```
SELECT extract( quarter from current_timestamp) ;
```

Ejemplo 13: se realiza una consulta a la **TMEMPLEADOS**, donde se muestra: la cedula, el nombre, la fecha de nacimiento, el año de nacimiento, el mes de nacimiento, el día de nacimiento y **la edad actual del empleado**.

```

SELECT cedemple, nomemple, dcargo, dstatus, fecha,
       Extract(Year from fecha) AS "Año",
       Extract(Month from fecha) AS "Mes",
       Extract(Day from fecha) AS "Día",
       Extract(Year from current_date) – Extract(Year from fecha) AS "Edad"
FROM tmempleados AS E , tmcargos AS C, tmstatus AS S
WHERE ( E.fkcods = S.cods ) and ( E.fkcodcar = C.codcar );

```

cedemple	nomemple	dcargo	dstatus	fecha	Año	Mes	Día	Edad
1000	ROBERTO JAIMES	GERENTE	ACTIVO	1978-03-03	1978	3	3	47
2000	ZILA CONTRERAS	VICE PRESIDENTE	ACTIVO	1980-10-05	1980	10	5	45
3000	MARTHA	GERENTE	ACTIVO	1996-11-22	1996	11	22	29
4000	WILMARY FLOREZ	SUB-GERENTE	ACTIVO	2000-07-15	2000	7	15	25
5000	MATIAS	SUB-GERENTE	ACTIVO	2005-09-25	2005	9	25	20
6000	NATASHA	PRESIDENTE	ACTIVO	2007-10-28	2007	10	28	18
7000	PEGGY CARTER	VICE PRESIDENTE	ACTIVO	1999-08-29	1999	8	29	26
8000	YSA CAPACHO	VICE PRESIDENTE	ACTIVO	1993-07-23	1993	7	23	32
9000	MARIA	SUB-GERENTE	ACTIVO	2003-07-07	2003	7	7	22

(9 filas)

Ejemplo 14:

```

select cbio, apebio, nombio ,
       extract(year from NOW()) - extract(year from fenac) as Edad
from tmbioanalistas ;

SELECT NOW() ; -> Muestra la fecha y hora actual del servidor

```

2.5.- Función DATE_PART() --> retorna una parte de la fecha.

SELECT campo1, campo2, , DATE_PART('Parametro' , campo_fecha)

FROM tabla WHERE Condición – criterio ORDER BY Campo1, campo2... ;

Parametros:

- year = Año
- quarter = Cuarto del año actual (Enero..Marzo: 1; Abril...Junio:2; Julio...Sept:3)
- month = mes
- day = día del mes
- week = Cantidad de semanas transcurridas del año

Ejemplo 01:

SELECT pkcedemple, nomemple, fecha AS "Fe Nac" ,

(DATE_PART('year', current_date) - DATE_PART('year', fecha)) AS "Edad"

FROM templeados

WHERE fecha BETWEEN '1993-01-01' AND '2005-12-31'

ORDER BY fecha ;

```
bdretardados=# SELECT pkcedemple, nomemple, fecha AS "Fe Nac",
bdretardados=# (DATE_PART( 'year', current_date) - DATE_PART( 'year', fecha) ) AS "Edad"
bdretardados=# FROM templeados
bdretardados=# WHERE fecha BETWEEN '1993-01-01' AND '2005-12-31'
bdretardados=# ORDER BY fecha ;
 pkcedemple | nomemple | Fe Nac | Edad
-----+-----+-----+-----
 8000      | YSA CAPACHO | 1993-07-23 | 32
 3000      | MARTHA     | 1996-11-22 | 29
 7000      | PEGGY CARTER | 1999-08-29 | 26
 4000      | CARLOS     | 2000-07-15 | 25
 9000      | MARIA MORALES | 2003-07-07 | 22
 5000      | MATIAS     | 2005-09-25 | 20
(6 filas)
```

2.6.- Función Age → retorna la cantidad de años, meses y días que existe de diferencias existe entre dos fechas.

```
SELECT campo1, campo2, . .... , AGE( current_date, campo_fecha)
FROM tabla
WHERE Condición – criterio

ORDER BY Campo1, campo2... ;
```

current_date -----→ fecha actual del servidor

Ejemplo 01: Se desea mostrar le número de cédula, el nombre y la edad actual de los empleados que nacieron entre los años 2000 (Inclusive) y 2005 (Inclusive).

```
SELECT pkcedemple AS "Cédula",
       nomemple AS "Nombre",
       EXTRACT( YEAR FROM AGE(current_date, fecha) ) AS "Edad"
FROM templeados
WHERE EXTRACT( YEAR FROM fecha ) BETWEEN 2000 and 2005 ;
```

```
bdretardados=# SELECT pkcedemple AS "Cédula",
bdretardados=# nomemple AS "Nombre",
bdretardados=# EXTRACT(YEAR FROM AGE(current_date, fecha)) AS "Edad"
bdretardados=# FROM templeados
bdretardados=# WHERE EXTRACT(YEAR FROM fecha) BETWEEN 2000 and 2005;
Cédula | Nombre | Edad
-----+-----+-----
4000 | CARLOS | 24
5000 | MATIAS | 19
9000 | MARIA MORALES | 21
(3 filas)
```

Ejemplo 02: Se desea consultar del empleado cuya cedula es 6000, lo siguiente: la cantidad de años, meses y días, que han transcurrido desde la fecha de registro de sus retardos a la fecha actual del servidor.

```
SELECT nr, fkcedemple, fkcodexcu, current_date AS "Fe Act", fecha,
       Age( current_date, fecha ) AS "Tiempo"
FROM tdretardos
WHERE ( fkcedemple = '6000' )

ORDER BY fecha ;
```

nr	fkcedemple	fkcodexcu	Fe Act	fecha	Tiempo
62	6000	4	2025-03-13	2022-01-05	3 years 2 mons 8 days
23	6000	6	2025-03-13	2022-01-15	3 years 1 mon 29 days
69	6000	2	2025-03-13	2022-01-21	3 years 1 mon 23 days
74	6000	3	2025-03-13	2022-01-25	3 years 1 mon 19 days
38	6000	3	2025-03-13	2022-02-07	3 years 1 mon 6 days
45	6000	6	2025-03-13	2022-02-27	3 years 14 days
50	6000	7	2025-03-13	2022-03-11	3 years 2 days
58	6000	5	2025-03-13	2022-03-25	2 years 11 mons 19 days
61	6000	5	2025-03-13	2022-03-28	2 years 11 mons 16 days
5	6000	7	2025-03-13	2022-04-11	2 years 11 mons 2 days
15	6000	5	2025-03-13	2022-04-25	2 years 10 mons 18 days
18	6000	7	2025-03-13	2022-04-28	2 years 10 mons 15 days
27	6000	3	2025-03-13	2022-05-07	2 years 10 mons 6 days
34	6000	6	2025-03-13	2022-05-27	2 years 9 mons 17 days
79	6000	3	2025-03-13	2022-06-07	2 years 9 mons 6 days
87	6000	6	2025-03-13	2022-06-27	2 years 8 mons 16 days
86	6000	6	2025-03-13	2022-06-27	2 years 8 mons 16 days
105	6000	4	2025-03-13	2022-07-16	2 years 7 mons 28 days
94	6000	6	2025-03-13	2022-07-16	2 years 7 mons 28 days
96	6000	2	2025-03-13	2022-07-26	2 years 7 mons 18 days
107	6000	7	2025-03-13	2022-09-26	2 years 5 mons 17 days
123	6000	6	2025-03-13	2022-11-03	2 years 4 mons 10 days
112	6000	7	2025-03-13	2022-11-05	2 years 4 mons 8 days
119	6000	3	2025-03-13	2022-11-12	2 years 4 mons 1 day
117	6000	4	2025-03-13	2022-11-16	2 years 3 mons 27 days
124	6000	1	2025-03-13	2022-11-19	2 years 3 mons 24 days
120	6000	7	2025-03-13	2022-11-26	2 years 3 mons 17 days
121	6000	4	2025-03-13	2022-11-27	2 years 3 mons 16 days

(28 filas)

Ejemplo 03: Se desea obtener el listado de las personas que llegaron tarde en noviembre 2022. Donde es importante observar la cantidad de años, meses y días, que han transcurrido desde la fecha de registro de sus retardos a la fecha actual del servidor;

```

SELECT M.pknr,
       M.fkcedemple,
       T.nomemple,
       M.fkcodexcu,
       X.dexcusa,
       current_date AS "Fe Act",
       M.fecha,
       Age( current_date, M.fecha ) AS "Tiempo"
FROM tdretardos AS M, templeados AS T, tmexcusas AS X
WHERE ( M.fkcedemple = T.pkcedemple ) AND
      ( M.fkcodexcu = X.pkcodexcu ) AND
      ( M.fecha BETWEEN '2022-11-01' AND '2022-11-30' )

ORDER BY M.fecha ;

```

```

bdretardados=#
bdretardados=# SELECT M.pknr,
bdretardados=#       M.fkcedemple,
bdretardados=#       T.nomemple,
bdretardados=#       M.fkcodexcu,
bdretardados=#       X.dexcusa,
bdretardados=#       current_date AS "Fe Act",
bdretardados=#       M.fecha,
bdretardados=#       Age( current_date, M.fecha ) AS "Tiempo"
bdretardados=# FROM tdretardos AS M, templeados AS T, tmexcusas AS X
bdretardados=# WHERE ( M.fkcedemple = T.pkcedemple ) AND
bdretardados=# ( M.fkcodexcu = X.pkcodexcu ) AND
bdretardados=# ( M.fecha BETWEEN '2022-11-01' AND '2022-11-30' )
bdretardados=# ORDER BY M.fecha

```

pknr	fkcedemple	nomemple	fkcodexcu	dexcusa	Fe Act	fecha	Tiempo
110	1000	ROBERTO JAIMES	1	ME QUEDE DORMIDO	2025-06-01	2022-11-01	2 years 7 mons
123	6000	NATASHA	6	MUCHO TRAFICO	2025-06-01	2022-11-03	2 years 6 mons 28 days
111	4000	CARLOS	3	NO ME BUSCARON	2025-06-01	2022-11-03	2 years 6 mons 28 days
112	6000	NATASHA	7	EL CARRO NO ME QUIERE	2025-06-01	2022-11-05	2 years 6 mons 26 days
113	5000	MATIAS	6	MUCHO TRAFICO	2025-06-01	2022-11-07	2 years 6 mons 24 days
116	7000	PEGGY CARTER	4	PELIE CON EL AGUILA	2025-06-01	2022-11-10	2 years 6 mons 21 days
119	6000	NATASHA	3	NO ME BUSCARON	2025-06-01	2022-11-12	2 years 6 mons 19 days
114	3000	MARTHA	2	SALI TARDE DE LA CASA	2025-06-01	2022-11-13	2 years 6 mons 18 days
115	2000	ZILA CONTRERAS	8	ME SECUESTRARON	2025-06-01	2022-11-15	2 years 6 mons 16 days
117	6000	NATASHA	4	PELIE CON EL AGUILA	2025-06-01	2022-11-16	2 years 6 mons 15 days
124	6000	NATASHA	1	ME QUEDE DORMIDO	2025-06-01	2022-11-19	2 years 6 mons 12 days
118	1000	ROBERTO JAIMES	6	MUCHO TRAFICO	2025-06-01	2022-11-19	2 years 6 mons 12 days
120	6000	NATASHA	7	EL CARRO NO ME QUIERE	2025-06-01	2022-11-26	2 years 6 mons 5 days
121	6000	NATASHA	4	PELIE CON EL AGUILA	2025-06-01	2022-11-27	2 years 6 mons 4 days
97	7000	PEGGY CARTER	5	MUCHA LLUVIA	2025-06-01	2022-11-27	2 years 6 mons 4 days

(15 filas)

Ejemplo 04: Dada la tabla **TMClientes** con un campo llamado **FeNac_Clie** (Fecha de nacimiento del cliente). Se desea agregar una fecha de nacimiento aleatoria para cada cliente

Solución A: Recorre toda la tabla **TMClientes** y pregunta en cada fila **si FeNac es Nula**; en caso de que la respuesta sea verdadera

a.- Genera un valor aleatorio (**RANDOM()** --> **Ejemplo:0.8843472626092566**),

b.- **multiplica dicho valor** por 15000 ($0.8843472626092566 * 15000$ -> 13265.208939138849)

c.- Extrae la parte entera del valor (13265.2089391388490000 -> 13265)

d.- Asigna una fecha inicial en este caso 01/Ene/1975 (DATE '1975-01-01')

e.- Suma 13265 días a la fecha DATE '1975-01-01' --> 2011-04-27

UPDATE tmclientes

SET FeNac = DATE '1975-01-01' + (RANDOM() * 15000)::INT

WHERE FeNac IS NULL ;

Solución B:

UPDATE tmclientes

SET FeNac = DATE '1980-01-01' + ((RANDOM() * (DATE '2005-12-31' - DATE '1980-01-01'))::INT)

WHERE FeNac IS NULL ;

3.- Funciones matemáticas

3.1.- `SELECT Abs(x) ... ;` --> Retorna el valor absoluto del valor "X".

Ejemplo: `SELECT Abs(-20);` ---> Retorna 20

3.2.- `SELECT cbrt (x) ... ;` --> Retorna la raíz cubica del valor "X".

Ejemplo: `SELECT cbrt (27);` --> Retorna 3.

3.3.- `SELECT ceiling (x) ... ;` --> Redondea hacia arriba el valor del argumento "X".

Ejemplo 01: `SELECT ceiling (12.34);` --> Retorna 13.

Ejemplo 02: `SELECT ceiling (12.57);` ---> Retorna 13.

3.4.- `SELECT floor (x) ... ;` --> Redondea hacia abajo el valor del argumento "X".

Ejemplo 01: `SELECT floor (12.34);` --> Retorna 12.

Ejemplo 02: `SELECT floor (12.57);` ---> Retorna 12.

3.5.- `SELECT round (x) ... ;` --> Redondea el valor "X" al valor más próximo.

Ejemplo 01: `SELECT round(10.4);` --> Retorna 10.

Ejemplo 02: `SELECT round(10.5);` --> Retorna 11.

`SELECT round (x , d) ... ;` --> Redondea el valor "X" con d decimales

Ejemplo 03: `SELECT round(10.44758 , 2);` --> Retorna 10.45

Ejemplo 04: `SELECT round(2015.033337 , 2);` --> Retorna 2015.03

3.6.- `SELECT power (x , y) ;` --> Retorna el valor de "x" elevado a la potencia "y".

Ejemplo 01: `SELECT power(2 , 3);` --> Retorna 8.

3.7.- `SELECT sign (x) ... ;` --> Retorna si el valor "X" es positivo o negativo.

Nota: si $x < 0$ --> -1; si $x = 0$ --> 0; si $x > 0$ --> 1

Ejemplo 01: `SELECT sign(-23.4);` --> Retorna -1.

3.8.- `SELECT sqrt (x) ... ;` --> Retorna la raíz cuadrada del valor "X".

Ejemplo 01: `SELECT sqrt(9);` --> Retorna 3.

3.9.- `SELECT mod (x, y) ... ;` --> Retorna el resto de la división de “X” entre “Y”.

Ejemplo 01: `SELECT mod(11, 2);` --> Retorna 1.

3.10.- `SELECT pi() ... ;` --> Retorna el valor de pi.

Ejemplo 01: `SELECT pi();` --> Retorna 3.14159265358979.

3.11.- `SELECT random() ... ;` --> Retorna un valor aleatorio entre 0 y 1
(Sin incluir el 0 o el 1).

Ejemplo 01: `SELECT random();` --> Retorna 0.895562474101578.

3.12.- `trunc()` -->... Recorta o trunca el valor X.

`SELECT trunc() ... ;` --> Retorna la parte entera del valor “X”.

Ejemplo 01: `SELECT trunc(34.7);` --> Retorna 34.

Ejemplo 02: `SELECT trunc(10.44758);` --> Retorna 10

Ejemplo 03: `SELECT trunc(2015.033337);` --> Retorna 2015

`SELECT trunc(x , d);` --> Retorna la parte el valor "X" con n decimales.

Ejemplo 04: `SELECT trunc(67.77874525 , 2);` --> Retorna 67.77.

Ejemplo 02: `SELECT trunc(10.44758 , 2);` --> Retorna 10.44

Ejemplo 03: `SELECT trunc(2015.033337 , 2);` --> Retorna 2015.03

3.13.- `SELECT factorial(x);` --> Retorna el factorial del valor "X".

Ejemplo 01: `SELECT factorial(5);` --> Retorna 120

3.14.- `SELECT COALESCE(Valor1, Valor2,, Valor N);`

--> Seleccione el primer valor no nulo que encuentre de izquierda a derecha.

Ejemplo 01: Dada la Tabla de Verdad de COALESCE

Valor 1	Valor 2	Valor 3	Resultado
10	20	30	10
NULL	500	900	500
NULL	NULL	'Hola Mundo'	'Hola Mundo'
NULL	NULL	NULL	NULL

`SELECT COALESCE(10, 20, 30);` --> 10

`SELECT COALESCE(NULL, 500, 900);` --> 500


`SELECT COALESCE(NULL, NULL, 'Hola Mundo');` --> Hola Mundo


Ejemplo 2: Determinar el monto máximo de lo pagado a los empleados en formato de miles y dos decimales

```
SELECT TO_CHAR( COALESCE( MAX(sueldo) , 0 ) , '999G999G999D99') AS "Monto Maximo" FROM tdpagos;
```

4.- Funciones Trigonómicas


4.1.- SELECT sin(x) ...  --> Retorna el valor del seno de "X".

Ejemplo 01: SELECT sin(0) ;  Retorna 0.

4.2.- SELECT cos(x) ...  --> Retorna el valor del coseno de "X".

Ejemplo 01: SELECT cos(0) ;  Retorna 1.

4.3.- SELECT tan(x) ...  --> Retorna el valor de la tangente de "X".

Ejemplo 01: SELECT tan(0) ;  Retorna 0.

4.4.- SELECT cot(x) ...  --> Retorna el valor de la cotangente de "X".

Ejemplo 01: SELECT cot(15) ;  Retorna -1.1682333052318372.

4.5.- SELECT asin(x) ...  --> Retorna el valor el arco seno de "X".

Ejemplo 01: SELECT asin(-0.5) ;  Retorna -0.5235987755982989.

4.6.- `SELECT acos(x);` --> Retorna el valor el arco coseno de "X".

Ejemplo 01: `SELECT acos(-0.5);` --> Retorna 2.0943951023931957.

4.7.- `SELECT atan(x);` --> Retorna el valor del arco tangente de "X".

Ejemplo 01: `SELECT atan(-0.5);` --> Retorna 0.4636476090008061.

4.8.- `SELECT atan2(x , Y);` --> Retorna el valor del arco tangente de los valores "X" y "Y".

Ejemplo 01: `SELECT atan2(-0.8 , 2);` --> Retorna -0.3805063771123649

4.9.- `SELECT radians(x);` --> Convierte un valor de grados a radianes.

Ejemplo 01: `SELECT radians(180);` --> Retorna 3.141592653589793

Ejemplo 02: `SELECT radians(85.94366926962348);` --> Retorna 1.5

4.10.- `SELECT degrees(x);` --> Convierte un valor de radianes a grados.

Ejemplo 01: `SELECT degrees(3.141592653589793);` --> Retorna 180

Ejemplo 02: `SELECT degrees(1.5);` --> Retorna 85.94366926962348

4.11.- `SELECT ln(x) ... ;` --> Retorna el valor del logaritmo natural del valor "X"

Ejemplo 01: `SELECT ln(2.718281828459045);` --> 0.9999999999999999

4.12.- `SELECT exp(x) ... ;` --> Retorna el valor de la constante e elevado a la potencia "X". Donde la constante e es igual a 2.718281, base de los algoritmos naturales

Ejemplo 01: `SELECT exp(0.9999999999999999);` --> Retorna 2.718281828459045

4.13.- `log()` --> Retorna el logaritmo natural del valor "X".

`SELECT log(x) ... ;` --> Retorna el logaritmo natural del valor "X".

Ejemplo 01: `SELECT log(2) ;` --> Retorna 0.3010299956639812

`SELECT log(x , y) ... ;` --> Retorna el logaritmo natural del valor "X" con en base "Y".

Ejemplo 01: `SELECT log(2 , 4) ;` --> Retorna 2.0000000

5.- Funciones string o de carácter

5.1.- `SELECT char_length(String_Texto);`

--> Retorna la cantidad de caracteres del String; en otras palabras, **retorna la longitud del texto**

Ejemplo01: `SELECT char_length('Hola Mundo');` --> Retorna 10

Ejemplo02: `SELECT character_length('Hola Mundo');` --> Retorna 10

Ejemplo03: `SELECT LENGTH ('PostgreSQL Tutorial'), LENGTH (NULL);` -->

length | length

-----+-----
19 |

5.2.- `SELECT upper('Texto');` --> Convierte todo el texto a mayúsculas

Ejemplo: `SELECT upper('Hola Mundo');` --> Retorna HOLA MUNDO

5.3.- `SELECT lower('Texto');` --> Convierte todo el texto a minúsculas

Ejemplo: `SELECT lower('Hola Mundo');` --> Retorna hola mundo

5.4.- SELECT substr('Texto_Base' , Pi, Can);

--> Extrae un sub-string de un Texto_Base; para ello se indica la **Pi(Posición Inicial)** y la **Can(Cantidad)** a extraer.

Ejemplo 01: SELECT substr('####My name is PostgreSQL#####', 5, 11); -->

Retorna: My name is

5.5.- SELECT substring('Texto_base' from PP for N);

--> Extrae un sub-string (Sub-Cadena) del Texto_base; para ello se indica la Posición de partida (**PP**) y la cantidad de caracteres a extraer (**N**).

Ejemplo 01: SELECT substring('My name is PostgreSQL' from 12 for 7); -->

Retorna Postgre

5.6.- SELECT position('Texto1' in 'Texto2');

--> Retorna la posición donde se encuentra por primera vez el texto1 en el texto2.

Nota: Si texto1 no existe en el texto2, retorna 0. Hace diferencia entre mayúsculas y minúsculas.

Ejemplo 01: SELECT position('Mundo' in 'Hola Mundo'); --> Retorna 6

Ejemplo 02: SELECT position('Mundo' in 'Hola Mundo, estoy en mi casa llamada Hola Mundo'); --> Retorna 6

5.7.- SELECT ltrim(' Texto') ... ;

--> Elimina todos los espacios en blanco que estén a la izquierda de String-Texto.

Ejemplo 01:

SELECT char_length(' My name is PostgreSQL'); --> Retorna 27

SELECT ltrim(' My name is PostgreSQL'); --> Retorna: My name is PostgreSQL'

SELECT char_length(**ltrim(' My name is PostgreSQL')**); ----> Retorna 21

5.8.- SELECT rtrim('Texto ') ... ;

--> Elimina todos los espacios en blanco que estén a la derecha de String-Texto.

Ejemplo 01:

SELECT char_length('My name is PostgreSQL ') AS "Total Inicio",

rtrim('My name is PostgreSQL ') AS "Texto Base",

char_length(rtrim('My name is PostgreSQL ')) As "Total Final"; -->

Total Inicio	Texto Base	Total Final
-----+-----+-----		
32	My name is PostgreSQL	21

5.9.- SELECT ascii('Caracter') ... ; --> Retorna el número entero ASCII asociado al carácter dado.

Ejemplo 01: SELECT Ascii('A') AS "A Mayúscula", Ascii('a') AS "a Minúscula"; -->

A Mayúscula	a Minúscula
-----+-----	
65	97

5.10.- SELECT chr(Número_Caracter);

--> Retorna el carácter Ascii asociado al número entero dado.

Ejemplo 01: SELECT chr(65), chr(97) ; -->

Retorna:

chr | chr

-----+-----

A | a

5.11.- SELECT trim([leading / trailing / both 'Caracter_a_eliminar']

from 'Texto_Base');

--> Elimina del texto dado, el(los) caracteres indicados del inicio, final o en ambos extremos.

Ejemplo 01: leading ---> Elimina del inicio del texto base; el carácter "#".

SELECT trim(leading '#' from '#####My name is PostgreSQL#####'); ->

Retorna: My name is PostgreSQL#####

Ejemplo 02: trailing ---> Elimina del final del texto base; el carácter "#".

SELECT trim(trailing '#' from '#####My name is PostgreSQL#####');

Retorna: My name is PostgreSQL#####

Ejemplo 03: both ---> Elimina del inicio y final del texto base; el carácter "#".

SELECT trim(both '#' from '#####My name is PostgreSQL#####');

Retorna: My name is PostgreSQL

5.12.- Concatenar → Existen dos formas de concatenar dos o más cadenas de texto.

Ejemplo 01: Usando el canalizador ||

```
SELECT 'Mi Nombres es:' || 'Sergio' || ' ' || 'Capacho' ; --→
```

Retorna: Mi Nombre es: Sergio Capacho

Ejemplo 02: Usando la función **concat()**

```
SELECT concat( 'Mi Nombres es:', 'Sergio', ' ', 'Capacho') AS "Texto"; --→
```

Retorna: Mi Nombres es: Sergio Capacho

5.13.- SELECT initcap('Texto') ; → Convierte la primera letra de cada palabra a mayúscula

Ejemplo 01: **SELECT initcap('my name is postgresQL ') ;** --→

Retorna: My Name Is Postgresql

5.14.- SELECT REPLACE(Cadena_Original , 'Texto_a_Buscar' , 'Reemplazo') ;

→ Es una herramienta de manipulación de texto que busca todas las apariciones de una subcadena específica ('Texto_a_Buscar') dentro de un texto dado (Cadena_Original) y las sustituye por un texto nuevo ('Reemplazo')

Es extremadamente útil para limpiar datos, corregir errores masivos o cambiar formatos de visualización

Ejemplo 01: Supongamos que tenemos una tabla llamada TMClientes con un campo llamado celular(varchar 12), donde los números se almacenaron inicialmente así: 57-300-726-63-70; y por necesidad de trabajo el número se necesita con el siguiente formato: 573007266370

```
SELECT REPLACE( celular , '-' , '' ) As "Celular Nuevo" FROM TMClientes ;
```

```
SELECT REPLACE( '57-300-726-63-70' , '-' , '' ) As "Celular Nuevo" ; --→
```

Entrada: 57-300-726-63-70

Salida: 573007266370

Ejemplo 02: Supongamos que se necesita actualizar el número celular de todos los clientes con el formato sin guiones (-).

```
UPDATE TMClientes
```

```
SET celular = REPLACE( celular , '-' , '' ) ;
```

Ejemplo 03: Corregir Nombres, Imagina que en tu base de datos de productos, alguien escribió mal la palabra "Premium" como "Prem":

```
SELECT REPLACE( nombre_producto , 'Prem' , 'Premium' ) As "Nombre Nuevo"
```

```
FROM TMProductos ; --→
```

Entrada: "Café Prem Arábica"

Salida: "Café Premium Arábica"

Reglas de Oro de REPLACE()

- 1.- Distingue entre mayúsculas y minúsculas; es decir, **SI buscas 'abc'**, no encontrará 'ABC'.
- 2.- Sustitución Global: Si la palabra aparece 3 veces en la misma celda, reemplazará las 3 veces.
- 3.- **No modifica la cadena original**, sino que devuelve una cadena con la sustitución. Para lograr una modificación en el punto de origen se debe usar UPDATE.
- 4.- Manejo de NULLs: Si cualquiera de los tres argumentos es NULL, el resultado suele ser NULL.
- 5.- Se puede utilizar en declaraciones SELECT para modificar la salida dinámicamente sin alterar los datos almacenados; como se puede observar en los ejemplos 01 y 03.
- 6.- Se puede combinar con otras funciones SQL como UPPER(), LOWER(), TRIM(), etc.

Ejemplo 03: Cambio en el formato de la salida numérica

Nota: Para colocar un carácter de escape en PostgreSQL, se utiliza principalmente la **barra invertida (\)** dentro de cadenas con el prefijo, ejemplo: 'cadena\ncon\tsalto'. Para insertar una comilla simple literal, se duplica (')', y para una barra invertida literal, se usa doble barra (\\) en cadenas de escape

```
SELECT REPLACE('123,456,789.67', ',', ' ') As "Sueldo Nuevo"; -->
```

Entrada: 123,456,789.67'

Salida: 123' 456' 789.67

```
SELECT REPLACE( REPLACE('123,456,789.67', ',', ' ') , '.', ',') )
```

```
As "Sueldo Nuevo"; -->
```

Entrada: 123,456,789.67'

Salida: 123' 456' 789,67

Fuentes

- 1.- <https://www.postgresql.org/docs/current/>
- 2.- <https://neon.tech/postgresql/tutorial>
- 3.- <https://www.w3schools.com/postgresql/>
- 5.- <https://www.todopostgresql.com/comandos-postgresql-alter/>
- 6.- <https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-alter-table/>
- 7.- <https://www.postgresql.org/docs/current/ddl-alter.html>
- 8.- <https://www.postgresql.org/docs/current/sql-altertable.html>
- 9.- https://www.w3schools.com/sql/sql_ref_group_by.asp
- 10.-
<https://www.tutorialesprogramacionya.com/postgresqlya/temarios/descripcion.php?inicio=0&cod=181&punto=23>
- 11.- <https://www.w3schools.com/sql/>
- 12.- Fuente: <https://programacionymas.com/blog/como-funciona-inner-left-right-full-join>
- 13.- LISTADO DE FUNCIONES DE LIBRERIA SQL DE POSTGRESQL
<https://learn.microsoft.com/es-es/sql/t-sql/functions/functions?view=sql-server-ver16>
- 14.- <https://www.pgadmin.org/docs/pgadmin4/development/genindex.html>

