

[CASA](#)[PROCEDIMIENTOS ALMACENADOS](#)[ÍNDICES](#)[PUNTOS DE VISTA](#)[GATILLOS](#)[INTERFACES](#) ▾[FUNCIONES](#) ▾

**Resumen** : en este tutorial, aprenderá cómo usar la `STRING_AGG()` función PostgreSQL para concatenar cadenas y colocar un separador entre ellas.

## Introducción a la `STRING_AGG()` función PostgreSQL .

La `STRING_AGG()` función PostgreSQL es una [función agregada](#) que concatena una lista de cadenas y coloca un separador entre ellas. La función no agrega el separador al final de la cadena.

A continuación se muestra la sintaxis de la `STRING_AGG()` función:

```
1 STRING_AGG ( expression, separator [order_by_clause] )
```

La `STRING_AGG()` función acepta dos argumentos y una `ORDER BY` cláusula opcional .

`expression` es cualquier expresión válida que pueda resolverse en una cadena de caracteres. Si utiliza otros tipos que no sean el tipo de cadena de caracteres, debe [convertir](#) explícitamente estos valores de ese tipo al tipo de cadena de caracteres.

`separator` Es el separador para cuerdas concatenadas.

El `order_by_clause` es una cláusula opcional que especifica el orden de los resultados concatenados. Tiene la siguiente forma:

```
1 ORDER BY expression1 {ASC | DESC}, [...]
```

El `STRING_AGG()` es similar a la `ARRAY_AGG()` función, excepto para el tipo de retorno. El tipo de retorno de la `STRING_AGG()` función es la cadena, mientras que el tipo de retorno de la `ARRAY_AGG()` función es la [matriz](#) .

Al igual que otras funciones de agregación tales como `AVG()` , `COUNT()` , `MAX()` , `MIN()` , y `SUM()` , la `STRING_AGG()` función se utiliza a menudo con la `GROUP BY` cláusula.

## `STRING_AGG()` Ejemplos de funciones de PostgreSQL

CASA

PROCEDIMIENTOS ALMACENADOS

ÍNDICES

PUNTOS DE VISTA

GATILLOS

INTERFACES ▾

FUNCIONES ▾

```

1 SELECT
2     f.title,
3     STRING_AGG (
4     a.first_name || ' ' || a.last_name,
5         ','
6     ORDER BY
7         a.first_name,
8         a.last_name
9     ) actors
10 FROM
11     film f
12 INNER JOIN film_actor fa USING (film_id)
13 INNER JOIN actor a USING (actor_id)
14 GROUP BY
15     f.title;

```

Aquí está la salida parcial:

title	actors
Academy Dinosaur	Christian Gable,Johnny Cage,Lucille Tracy,Mary Keitel,Mena Temple,Oprah Kilmer,Penelope Guinness,Rock Dukakis,Sandra Peck,Warren Nolte
Ace Goldfinger	Bob Fawcett,Chris Depp,Minnie Zellweger,Sean Guinness
Adaptation Holes	Bob Fawcett,Cameron Streep,Julianne Dench,Nick Wahlberg,Ray Johansson
Affair Prejudice	Fay Winslet,Jodie Degeneres,Kenneth Pesci,Oprah Kilmer,Scarlett Damon
African Egg	Dustin Tautou,Gary Phoenix,Matthew Carrey,Matthew Leigh,Thora Temple
Agent Truman	Jayne Neeson,Kenneth Hoffman,Kirsten Paltrow,Morgan Williams,Reese West,Sandra Kilmer,Warren Nolte
Airplane Sierra	Jim Mostel,Mena Hopper,Michael Bolger,Oprah Kilmer,Richard Penn
Airport Pollock	Fay Kilmer,Gene Willis,Lucille Dee,Susan Davis
Alabama Devil	Christian Gable,Elvis Marx,Greta Keitel,Mena Temple,Meryl Allen,Rip Crawford,Rip Winslet,Warren Nolte,William Hackman
Aladdin Calendar	Alec Wayne,Greta Malden,Jada Ryder,Judy Dean,Ray Johansson,Renee Tracy,Rock Dukakis,Val Bolger
Alamo Videotape	Johnny Cage,Michael Bening,Scarlett Damon,Sean Guinness

B) Uso de la `STRING_AGG()` función para generar una lista de correos electrónicos.

El siguiente ejemplo utiliza la `STRING_AGG()` función para crear una lista de correo electrónico para cada país. El correo electrónico en cada lista separado por un punto y coma.

```

1 SELECT
2     country,
3     STRING_AGG (email, ';') email_list
4 FROM
5     customer
6 INNER JOIN address USING (address_id)
7 INNER JOIN city USING (city_id)
8 INNER JOIN country USING (country_id)

```

[CASA](#)[PROCEDIMIENTOS ALMACENADOS](#)[ÍNDICES](#)[PUNTOS DE VISTA](#)[GATILLOS](#)[INTERFACES](#) ▾[FUNCIONES](#) ▾

Algeria	judy.gray@sakilacustomer.org;june.carroll@sakilacustomer.org;mario.cheatham@sakilacustomer.org
American Samoa	anthony.schwab@sakilacustomer.org
Angola	martin.bales@sakilacustomer.org;claudel.herzog@sakilacustomer.org
Anguilla	bobby.boudreau@sakilacustomer.org
Argentina	kimberly.lee@sakilacustomer.org;julia.flores@sakilacustomer.org;florence.woods@sakilacustomer.org;
Armenia	stephanie.mitchell@sakilacustomer.org
Austria	jill.hawkins@sakilacustomer.org;audrey.ray@sakilacustomer.org;nora.herrera@sakilacustomer.org
Azerbaijan	andrew.purdy@sakilacustomer.org;raymond.mcwhorter@sakilacustomer.org
Bahrain	seth.hannon@sakilacustomer.org
Bangladesh	michelle.dark@sakilacustomer.org;frank.waggoner@sakilacustomer.org;stephen.qualls@sakilacustome
Belarus	clara.shaw@sakilacustomer.org;cory.meehan@sakilacustomer.org
Bolivia	joel.francisco@sakilacustomer.org;jon.wiles@sakilacustomer.org
Brazil	debra.nelson@sakilacustomer.org;frances.parker@sakilacustomer.org;denise.kelly@sakilacustomer.or
Brunei	lois.butler@sakilacustomer.org
Bulgaria	jessie.banks@sakilacustomer.org;tyrone.asher@sakilacustomer.org
Cambodia	allison.stanley@sakilacustomer.org;elmer.noel@sakilacustomer.org

En este tutorial, ha aprendido cómo usar la `STRING_AGG()` función PostgreSQL para concatenar cadenas y colocar un separador entre ellas.

¿Fue útil este tutorial?



Si



no



Tutorial Anterior:

[Función PostgreSQL ARRAY\\_AGG](#)

INICIO RÁPIDO DE POSTGRESQL

[CASA](#)[PROCEDIMIENTOS ALMACENADOS](#)[ÍNDICES](#)[PUNTOS DE VISTA](#)[GATILLOS](#)[INTERFACES ▾](#)[FUNCIONES ▾](#)[Explorar objetos de servidor y base de datos](#)[FUNDAMENTOS DE POSTGRESQL](#)[SELECCIONAR](#)[ORDEN POR](#)[SELECCIONAR DISTINTO](#)[DÓNDE](#)[LÍMITE](#)[HA PODIDO RECUPERAR](#)[EN](#)[ENTRE](#)[ME GUSTA](#)[ES NULO](#)[ALIAS](#)[Se une](#)[UNIR INTERNAMENTE](#)[UNIRSE IZQUIERDO](#)[Auto-unirse](#)[ÚNETE COMPLETO](#)[Cruzar](#)[Unir Natural](#)[AGRUPAR POR](#)[TENIENDO](#)[UNIÓN](#)[INTERSECARSE](#)

[CASA](#)[PROCEDIMIENTOS ALMACENADOS](#)[ÍNDICES](#)[PUNTOS DE VISTA](#)[GATILLOS](#)[INTERFACES ▾](#)[FUNCIONES ▾](#)

---

[ALGUNA](#)[TODOS](#)[EXISTE](#)[INSERTAR](#)[ACTUALIZAR](#)[ACTUALIZACIÓN Únete](#)[Upsert](#)[BORRAR](#)[GESTIÓN DE LA ESTRUCTURA DE LA MESA](#)[Tipos de datos PostgreSQL](#)[CREAR MESA](#)[SELECCIONAR EN](#)[CREAR TABLA COMO](#)[Columna de incremento automático con SERIE](#)[Columna de identidad](#)[ALTERAR MESA](#)[MESA PLEGABLE](#)[TABLA DE TRUNCAS](#)[VERIFICAR Restricción](#)[Restricción no nula](#)[Clave externa](#)[Clave primaria](#)[Restricción ÚNICA](#)

[CASA](#)[PROCEDIMIENTOS ALMACENADOS](#)[ÍNDICES](#)[PUNTOS DE VISTA](#)[GATILLOS](#)[INTERFACES ▾](#)[FUNCIONES ▾](#)

---

[Vista Recursiva PostgreSQL](#)[POSTGRESQL TRIGGERS](#)[Introducción a Trigger](#)[Creando un gatillo](#)[Gestionando Desencadenadores PostgreSQL](#)

## SOBRE POSTGRESQL TUTORIAL

PostgreSQLTutorial.com es un sitio web dedicado a desarrolladores y administradores de bases de datos que trabajan en el sistema de administración de bases de datos PostgreSQL.

Constantemente publicamos útiles tutoriales de PostgreSQL para mantenerlo actualizado con las últimas características y tecnologías de PostgreSQL. Todos los tutoriales de PostgreSQL son simples, fáciles de seguir y prácticos.

## TUTORIALES RECIENTES DE POSTGRESQL

[Función PostgreSQL STRING\\_AGG](#)[Función PostgreSQL ARRAY\\_AGG](#)[PostgreSQL CREAR PROCEDIMIENTO](#)[Transacción PostgreSQL](#)[PostgreSQL ISNULL](#)[PostgreSQL CTE](#)[PostgreSQL es nulo](#)[Índices de columnas múltiples de PostgreSQL](#)[Tipos de índices de PostgreSQL](#)[PostgreSQL Listar índices](#)[MÁS TUTORIALES](#)[PostgreSQL hoja de trucos](#)[Administración de PostgreSQL](#)

[CASA](#)[PROCEDIMIENTOS ALMACENADOS](#)[ÍNDICES](#)[PUNTOS DE VISTA](#)[GATILLOS](#)[INTERFACES ▾](#)[FUNCIONES ▾](#)

---

## INFORMACION DEL SITIO

[Casa](#)[Sobre nosotros](#)[Contáctenos](#)[Política de privacidad](#)

Copyright © 2019 by [PostgreSQL Tutorial del](#) sitio web. Todos los derechos reservados.