

# Funciones PL/PGSQL

Antonio Espín Herranz

# Contenidos

- Funciones almacenadas:
  - Funciones de sql o de lenguaje de consulta
  - Funciones de lenguaje procedural
  - Funciones de lenguaje de programación
- Procedimientos almacenados

# Funciones de SQL

- Sintaxis:
- Create [or replace] function nombreFuncion([arg1 argTipo argDefault, ...]) return tipoVar as 'consultaSQL' **language 'sql'**;
- consultaSQL: Rellenamos el comando sql.
- Indicamos que es una función del lenguaje SQL.

# Ejemplo

- **Create function** “nombre\_esquema”.nombre\_función(origen integer, destino integer, valor numeric) returns integer
  - **As ‘una consulta SQL’** → dentro de este sql puedo utilizar los parámetros que pasamos a la función. Los parámetros se identifican con el signo de \$ seguido de un número. Pueden ser consultas de acción, además separando las queries por un ;
  - Dentro de la parte de la consulta el valor devuelto de un integer se puede realizar con **select 1**
  - Los parámetros se numeran **\$1, \$2, \$3** según la posición que ocupan en la cabecera de la función.
  - Después de la query se especifica en que lenguaje se ha escrito la función.
  - **Language ‘sql’;**
  - **Terminamos con un ;**
  - **Se puede colocar SOLO el tipo de los parámetros**

# Ejecutar la función de SQL

- Para **ejecutar** la consulta se utiliza un select
  - **Select** “esquema”.nombreFuncion(param1, param2, param3);
- Para eliminar una función del esquema:
  - **Drop function** “nombreEsquema”.nombreFuncion;
- Dentro de una function si ejecuto algún sql; insert, update o delete puede utilizar como valor retorno FOUND (es un boolean) si la query se ejecuta bien. Found retorna 0 y si no retorna 1 (si no encuentra nada)

# Funciones de lenguaje de procedural

- Sintaxis:

Create [or replace] function nombreFuncion ([arg1 argTipo argDefault, ])

As

\$\$

Declare

    [variable tipo]

Begin

    Código

End

\$\$

Language 'plpgsql'; → Se indica el tipo de lenguaje

# Funciones de lenguaje de procedural

- Create function “nombreEsquema”.nombreFuncion() returns integer as
  - \$\$
  - Declare
    - Suma integer;
  - Begin
    - Select sum(saldo) into suma from cuentas;
    - Raise info ‘suma de saldos es: %’, suma;
    - Return suma;
  - End;
  - \$\$
  - Language ‘plpgsql’;

# Funciones de lenguaje procedural

- Ejecutar la función anterior:
- `Select "nombreEsquema".nombreFuncion();`
- Tiene que salir el mensaje de info y además retorna el valor de la suma calculada.



# Procedimientos almacenados

- Tenemos dos tipos:
  - Procedimientos de lenguaje de consulta
  - Procedimientos de lenguaje procedural.

# Procedimientos de lenguaje de consulta

- Sintaxis:

Create or replace procedure nombre([arg1 argTipo argDefault...])

As 'consulta SQL'

Language 'sql';

- Para llamar al procedimiento almacenado:
- **Call** nombreEsquema.nombreProc(param1, ... paramN)

# Ejemplo

```
create or replace procedure nombreEsquema.nombreProc(param1  
tipo1, param2 tipo2)
```

```
language sql
```

```
as
```

```
$$
```

```
    sql 1
```

```
    sql 2
```

```
    sql 3
```

```
$$
```

# Procedimientos de lenguaje procedural

Create or replace procedure nombre([arg1 argTipo, ...])

Language **plpgsql** -- **Se puede poner al final ...**

As

\$\$

Declare

Variables

Begin

Código

End

\$\$

# Funciones PL/PGSQL

- <https://www.postgresql.org/docs/current/plpgsql-declarations.html>