

# Set-Returning Functions

Álvaro Herrera

13 de noviembre de 2004

# Set-returning Functions (SRF)

- ▶ Funciones que “retornan tablas”
- ▶ En cláusula FROM
- ▶ hacer JOIN
  - ▶ con otras tablas
  - ▶ otras funciones
  - ▶ subconsultas
- ▶ Definir tipo de retorno

# Ejemplo 1

```
SELECT * FROM
    una_srf(arg1, ...)
JOIN otra_srf( ... ) USING (columna)
JOIN una_tabla ON (condicion ...)
```

## Ejemplo 2

```
SELECT * FROM
    una_tabla,
    una_srf(...),
    (SELECT tres, cuatro
     FROM otra_srf(...))
WHERE ...
```

# Tipos de retorno

- ▶ Lo especial de una SRF
- ▶ RETURNS SETOF foo
- ▶ foo puede ser
  - ▶ un tipo
  - ▶ una tabla
  - ▶ una definición anónima

# Sintaxis para retorno

1. 

```
CREATE FUNCTION una_srf (...)  
  RETURNS SETOF tabla ...;  
SELECT * FROM una_srf (foo, bar, ...);
```
2. 

```
CREATE TYPE foo (a int, b int);  
CREATE FUNCTION otra_srf (...)  
  RETURNS SETOF foo ...;  
SELECT * FROM otra_srf(...);
```

## Sintaxis para retorno (cont.)

```
3. CREATE FUNCTION tercera_srf(...)
    RETURNS SETOF RECORD ...;

SELECT * FROM tercera_srf(...)
    AS tercera(definicion de tipo);
```

# PL/pgSQL: sintaxis

**RETURN NEXT foo** acumula la tupla foo para retornar

**RETURN** termina la función y devuelve todas las tuplas retornadas

```
FOR SELECT ... LOOP
    ...
    RETURN NEXT ...
END LOOP;
RETURN;
```



# Dividir un intervalo en trozos

```
DECLARE
    inicio ALIAS FOR $1;
    valor  TIMESTAMP WITH TIME ZONE;
    final  ALIAS FOR $2;
    paso   ALIAS FOR $3;
    ret    RECORD;
```

## Dividir un intervalo en trozos (cont.)

```
BEGIN
valor := inicio;
LOOP
    IF valor >= final THEN
        RETURN;
    END IF;
    SELECT INTO ret valor, valor + paso;
    RETURN NEXT ret;
    valor := (valor + paso);
END LOOP;
END;
```

# Dividir un intervalo en trozos

```
SELECT * FROM intervalos
  ('2003-01-01', '2003-01-02', '3 hour')
 AS foo(inicio TIMESTAMP WITH TIME ZONE,
        fin TIMESTAMP WITH TIME ZONE);
```

<i>inicio</i>	<i>fin</i>
2003-01-01 00:00:00-03	2003-01-01 03:00:00-03
2003-01-01 03:00:00-03	2003-01-01 06:00:00-03
2003-01-01 06:00:00-03	2003-01-01 09:00:00-03
⋮	⋮
2003-01-01 21:00:00-03	2003-01-02 00:00:00-03

# SRF en SQL

```
CREATE FUNCTION getfoo(int)
  RETURNS setof foo AS '
    SELECT * FROM foo
    WHERE fooid = $1;'
  LANGUAGE SQL;
```

```
SELECT * FROM getfoo(1);
```