

Introducción a PostgreSQL

Álvaro Herrera

1er Foro Mundial de Tecnología Libre

LISTEN/NOTIFY: ¿qué es?

- ▶ Mecanismo para notificar cambios hacia el exterior
- ▶ Asegura transaccionalidad
 - ▶ sólo se dispara al momento de hacer COMMIT
 - ▶ sólo si la transacción ha sido exitosa

Ejemplos de uso

- ▶ Envío de correo
- ▶ Refrescar vistas materializadas asincrónicas
- ▶ Invocar programas externos de limpieza

Veremos cómo se hace correctamente un sistema que envíe un mail a un usuario cuando sucede algo.

Demostración

Acá hago la demostración del programa que envía mail.

El Esquema

Tabla «maildirs»

<i>Columna</i>	<i>Tipo</i>	<i>Modificadores</i>
maildir_id	serial	PRIMARY KEY
direccion	text	NOT NULL

Tabla «eventos_email»

<i>Columna</i>	<i>Tipo</i>	<i>Modificadores</i>
evento_id	serial	PRIMARY KEY
maildir_id	integer	not null REFERENCES maildir
contenido	text	not null
enviado	boolean	not null

El proceso Listener

- ▶ Listener: escucha permanentemente
- ▶ Recibe las notificaciones
- ▶ Debe actuar en respuesta
- ▶ Al recibir la notificación, la base de datos ya fue actualizada con la nueva información

Listener: El Código Fuente

Acá muestro el código fuente del Listener
<http://alvherre.atentus.cl/charlas/listener.c>

El proceso Notificador

```
CREATE FUNCTION notifica_evento_email()  
RETURNS TRIGGER LANGUAGE plpgsql AS $$  
    BEGIN  
        NOTIFY eventos_mail;  
        RETURN null;  
    END $$;
```

```
CREATE TRIGGER trig_notif  
AFTER INSERT ON eventos_email  
FOR EACH ROW  
EXECUTE PROCEDURE notifica_evento_email();
```