

Llamar procedures/functions with call.

crea un procedimiento que permita crear una vista de todas las tareas que tiene asociada una categoría de tareas.

create or replace procedure nombre (nombre\_categoria TEXT)

Language plpgsql

As \$\$

Begin

execute format ('create or replace view vista\_tareas\_%.1s as

select \* from tareas where categoria = '%.1s',

nombre\_categoria, nombre\_categoria

);

END;

\$\$;

select \* from tareas;

• crear vista materializada con usuarios que tengan tareas pendientes.

create materialized view vista\_usuarios\_tareas\_pendientes as

select u.id, u.nombre, t.estado from usuarios u

inner join tareas t on u.id = t.usuario\_id

where t.estado = 'pendiente';

• crear una transacción que permita cerrar las tareas que ya ha 'expirado' por fecha.

begin;

update tareas set estado = 'cerrado'

where fecha\_limite < current\_date and estado != 'cerrado';

commit;

Cree una transacción que permita clonar las tareas y sean asignadas a otro usuario.

DO \$\$

Declare

tarea RECORD;

nuevo-usuario-id INT := 5;

Begin

FOR tarea IN select \* <sup>from</sup> tareas LOOP

Insert into tareas (a, b, c, usuario-id)

values (tarea.a, tarea.b, tarea.c, nuevo-usuario-id);

End LOOP;

END;

\$\$;

• Cree una tabla temporal con toda la información de las tareas.

Create Temp table info.tareas AS

Select \* FROM tareas;

• Cree un procedimiento que permita llenar cualquier tabla en cualquier BD utilizando los ciclos

Create or Replace procedure llenar-tabla (tabla TEXT, veces INT)

Language plpgsql - AS \$\$ - declare i INT := 1;

Begin

while i <= veces LOOP

EXECUTE format('Insert into %I default values', tabla);

i := i + 1;

END LOOP;

END;

\$\$;

Cree una función que devuelva el nombre de la persona, junto a tarea  
 create or REPLACE function a-v ()

RETURNS Table (nombre text, tarea, text)

AS \$\$

Begin

RETURN QUERY

select u.nombre, t.descripcion

from usuarios u

join tareas t on u.id = t.usuario\_id;

END;

\$\$ LANGUAGE plpgsql;

Llamado select \* FROM Function-nombre();

• Concatenar atributos

create or replace procedure resumen-tareas()

LANGUAGE plpgsql

AS \$\$

DECLARE

Tarea RECORD;

Begin

for tarea in select atributos(columnas) from tareas t

join usuarios u on t.usuario\_id = u.id

Loop

Raise notice '# % - % - % - % - %';

tarea.a, tarea.b

END LOOP;

END;

\$\$;



Procedimiento para crear una vista normal o materializada  
con case when.

```
create or replace procedure gca_DB (nombre TEXT, TIPO TEXT)  
LANGUAGE plpgsql.
```

```
AS $$
```

```
Begin
```

```
EXECUTE format (
```

```
case
```

```
when tipo = 'normal' then
```

```
'CREATE VIEW %I AS select * FROM Tareas'
```

```
else
```

```
'CREATE MATERIALIZED VIEW %I AS select * from  
Tareas'
```

```
END;
```

```
nombre
```

```
);
```

```
END;
```

```
$$;
```

• clonar una tabla en una tabla temporal

```
CREATE OR REPLACE PROCEDURE clonar_tabla (tabla-origen  
text)
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
Begin
```

```
EXECUTE format ('CREATE TEMP Table temp_%I AS  
table %I', tabla-origen, tabla-origen);
```

```
EO;
```

```
$$;
```

```
call, select * from temp-tareas;
```