

Bases de Datos II

POSTGRES, PL/PGSQL, INTEGRIDAD Y SEGURIDAD DE LOS DATOS

UNPSJB - Sede Trelew
2018

Cátedra:

Profesor: Lic. Gabriel Ingravallo

JTP: Lic. Cristian Parise

Integrantes:

Pablo Adrian Toledo Margalef

1. Crear la tabla "audit" con los siguientes atributos

- Tabla (investigar si se puede crear una restricción para que solamente permita en este campo un nombre de tabla que exista en la base de datos)
- Tipo Operacion (crear una restricción para que solamente permita el siguiente dominio 'SELECT', 'INSERT', 'DELETE', 'UPDATE')
- Usuario
- Fecha/Hora

```
-- cjo de tablas del esquema public en la base actual
select 'in' in (select
table_name
from
information_schema.tables tbls
where
tbls.table_catalog=(select current_database())
and
tbls.table_schema='public');

-- usuario actual
select current_user;

create or replace function tabla_valida(tabla varchar) returns
boolean as
$$
begin
    return (select tabla in (select
        table_name
        from
        information_schema.tables tbls
        where
        tbls.table_catalog=(select current_database())
        and
        tbls.table_schema='public'));
end
$$ language plpgsql;

create or replace function operacion_valida(varchar) returns boolean
as
$$
begin
    return (select lower($1) in
('update', 'insert', 'delete', 'update'));
end
$$ language plpgsql;
select operacion_valida('UPDATE');
```

```
-- drop si existe
drop table if exists audit;
-- creacion tabla audit
create table audit (
    tabla varchar(20) not null constraint nombre_valido_check
check(tabla_valida(tabla)),
    tipo_operacion varchar(20) not null constraint
operacion_valida_check check(operacion_valida(tipo_operacion)),
    usuario varchar(30) not null,
    fecha_hora timestamp with time zone default current_timestamp
);
```

2. Crear los siguientes TRIGGERS

- para INSERCIÓN sobre Tabla avion que inserte en la tabla audit la operacion realizada
- para ELIMINACIÓN sobre Tabla avion que inserte en la tabla audit , la operación realizada

```
create or replace function audit_insert_avion() returns trigger
as
$$
begin
    insert into audit values('avion','insert', current_user);
    return new;
end
$$ language plpgsql;
```

```
create trigger insercion_avion after
insert on avion
execute procedure audit_insert_avion();
```

```
create or replace function audit_delete_avion() returns trigger
as
$$
begin
    insert into audit values('avion','delete', current_user);
    return new;
end
$$ language plpgsql;
```

```
create trigger eliminacion_avion after
delete on avion
execute procedure audit_delete_avion();
```

3. Crear dos usuarios en la base de datos TP1-Aviones, denominados userA y userB

```
create user user_a;
create user user_b;
```

4. Investigar y documentar la gramática y diferentes modificadores de las sentencias GRANT y REVOKE en Postgres.

La sintaxis y el uso de las sentencias grant y revoke en postgres son extremadamente extensas, por lo que me limitaré a comentar brevemente aquellas que me llamaron la atención.

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES
| TRIGGER }
[, ...] | ALL [ PRIVILEGES ] }
ON { [ TABLE ] table_name [, ...]
| ALL TABLES IN SCHEMA schema_name [, ...] }
TO role_specification [, ...] [ WITH GRANT OPTION ]
```

El uso de grant on table nos permite restringir a nivel de tabla completa las operaciones que se pueden realizar, también pudiendo en la misma sentencia otorgar permisos para todas las tablas de cierto esquema de la base.

```
GRANT { EXECUTE | ALL [ PRIVILEGES ] }
ON { FUNCTION function_name ( [ [ argmode ] [ arg_name ]
arg_type [, ...] ] ) [, ...]
| ALL FUNCTIONS IN SCHEMA schema_name [, ...] }
TO role_specification [, ...] [ WITH GRANT OPTION ]
```

Otorgamiento curioso e interesante de permisos, ya que permite especificar qué función puede utilizar el usuario al que se le otorgan. Se especifica qué parametros recibe y de qué tipo para poder señalar unívocamente a la función. También se puede otorgar para todas las funciones del esquema.

```
GRANT { USAGE | ALL [ PRIVILEGES ] }
ON LANGUAGE lang_name [, ...]
```

```
TO role_specification [, ...] [ WITH GRANT OPTION ]
```

Otorgar permisos sobre tal o cual lenguaje.

Con respecto al revoke se observa que respeta la misma diferenciación en la granularidad que se observa en el grant. Pero con la siguiente particularidad.

```
REVOKE [ GRANT OPTION FOR ]
    { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES |
      TRIGGER }
      [, ...] | ALL [ PRIVILEGES ] }
    ON { [ TABLE ] table_name [, ...]
        | ALL TABLES IN SCHEMA schema_name [, ...] }
    FROM { [ GROUP ] role_name | PUBLIC } [, ...]
    [ CASCADE | RESTRICT ]
```

Se pueden revocar los permisos sobre una tabla, esquema o función, pero también se puede revocar el permiso de otorgar permisos a otros usuarios.

5. Otorgar los siguientes permisos:

- de seleccion al userA sobre todas las tablas y permisos de INSERT sobre la tabla audit con la opción WITH GRANT OPTION
- Otorgar permisos de seleccion, insert y update al userB sobre las siguientes tablas: avion, piloto, pilotoAvion

```
grant all privileges
on table
avion,
falla,
jefe,
localidad,
"modeloAvion",
piloto,
"pilotoAvion",
trabajador,
"trabajadorReparacion"
to user_a with grant option;
```

```
grant insert on table audit to user_a with grant option;
```

```
grant select, insert, update on table avion, piloto,"pilotoAvion" to
user_b with grant option;
```

6. Conectarse con el usuario userB a la base de datos y hacer una insercion sobre la tabla avion, documentar lo que arroja el motor. Si no fue posible realizar la operación, solucionar el inconveniente conectándose con el usuario userA

```

/*
-- Inserción sobre avion utilizando user_b
tp1-aviones=> insert into avion values (1015,10,1980,120)
tp1-aviones-> ;
ERROR: permiso denegado a la relación audit
CONTEXT0: sentencia SQL: «insert into audit
values('avion','insert', current_user)»
función PL/pgSQL audit_insert_avion() en la línea 4 en sentencia SQL
*/

/*
-- Conectado con user_a
tp1-aviones=> grant insert on table audit to user_b;
GRANT
*/

/*
--Otra vez conectado con user_b
tp1-aviones=> insert into avion values (1015,10,1980,120);
INSERT 0 1
tp1-aviones=> select * from avion where "nroAvion"=1015;
  nroAvion | tipoModelo | año   | horasVuelo
-----+-----+-----+-----
      1015 |           | 10    | 1980      |
(1 fila)
*/

```