Creación de Usuarios, Roles yadministración de permisos

Oscar Gutierrez Blanco
Lorena Lozano Plata
José Miguel Alonso
Jesús Escobar

Universidad de Alcalá

Creación de un usuario

- Se crea con el comand SQL:

  CREATE USER ***name*** [[WITH] ***option***

- Donde las opciones son:
- Para más detalle de cada opció

```
    SUPERUSER | NOSUPERUSER
  | CREATEDB | NOCREATEDB
  | CREATEROLE | NOCREATEROLE
  | INHERIT | NOINHERIT
  | LOGIN | NOLOGIN
  | REPLICATION | NOREPLICATION
  | BYPASSRLS | NOBYPASSRLS
  | CONNECTION LIMIT connlimit
  | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
  | VALID UNTIL 'timestamp'
  | IN ROLE role_name [, ...]
  | IN GROUP role_name [, ...]
  | ROLE role_name [, ...]
  | ADMIN role_name [, ...]
  | USER role_name [, ...]
  | SYSID uid
```

Creación de un role

- Se crea con el comand SQL:

  CREATE ROLE ***name*** [[WITH] ***option*** [...]]

- La única diferencia entre la creación de roles y usuarios es que por defecto se establece la poción NOLOGIN, con lo que el ROLE no se puede conectar y se utiliza para administrar privilegios.
- Estableciendo la opción LOGIN un role es igual un usuario

Administración de permisos

- Concesión de privilegios a un role o un usuario:

```
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 ]
    [ GRANTED BY role_specification ]
```

- Los privilegeios que se pueden conc
  - Má detalles de cada privilegio en:
  - https://www.postgresql.org/docs/15/sql

```
SELECT
INSERT
UPDATE
DELETE
TRUNCATE
REFERENCES
TRIGGER
CREATE
CONNECT
TEMPORARY
EXECUTE
USAGE
SET
ALTER SYSTEM
```


Universidad de Alcalá

Administración de permisos

- Revocación de privilegios a un role:

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

- Las opciones son igual que en el comando GRANT, salvo la opción CASCADE que permite revocar los provilegios en cascada. Es decir, si se revocan privilegios a un roles, si este había concedido esos mismos privilegios a otros usuarios, estos se revocan automáticamente :
  - Má detalles en:
  - https://www.postgresql.org/docs/15/sql-revoke.html

Universidad de Alcalá

Administración de permisos

- Revocación de privilegios a un role:

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

- Las opciones son igual que en el comando GRANT, salvo la opción CASCADE que permite revocar los provilegios en cascada. Es decir, si se revocan privilegios a un roles, si este había concedido esos mismos privilegios a otros usuarios, estos se revocan automáticamente :
  - Má detalles en:
  - https://www.postgresql.org/docs/15/sql-revoke.html

Universidad de Alcalá

Ejemplo

- Vamos a crear un role que se conecta a la base de datos de congresos.
- Le vamos a conceder únicamente permisos de lectura en todas las tablas.
- En primer lugar creamos dos roles con la opción LOGIN y les a
- A

```
congresos=# CREATE ROLE oscar WITH LOGIN PASSWORD 'oscar';
CREATE ROLE
congresos=#
```

```
congresos=# REVOKE ALL PRIVILEGES ON DATABASE congresos FROM oscar;
REVOKE
congresos=#
```


Universidad de Alcalá

Ejemplo

- A Continuación salimos de psql y nos conectamos con el usuario oscar

```
oscargutierrez@MacBook-Air-de-oscar ~ % psql -U oscar -p 5434 -d congresos -W
Password:
psql (13.4, server 13.2)
Type "help" for help.

congresos=> \d
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | ponencia | table | postgres
 public | ponente  | table | postgres
 public | sala     | table | postgres
 public | sesion   | table | postgres
(4 rows)
```

- In sulta sencilla

```
congresos=> SELECT * FROM ponente;
ERROR:  permission denied for table ponente
congresos=>
```


Universidad de Alcalá

Ejemplo

- Concedemos los permisos de lectura en todas las tablas y a continuación realizamos la misma consulta:

```
congresos=> \c congresos postgres
Password for user postgres:
psql (13.4, server 13.2)
You are now connected to database "congresos" as user "postgres".
congresos=# GRANT SELECT ON ALL TABLES IN SCHEMA public TO oscar;
GRANT
congresos=# \c congresos oscar
Password for user oscar:
psql (13.4, server 13.2)
You are now connected to database "congresos" as user "oscar".
congresos=> SELECT * FROM sesion;
    dia     | nums | nomsala |   dni    |         titulo
------------+------+---------+----------+------------------------
 2006-01-06 |    1 | BOSQUE  | 30506080 | HARDWARE
 2006-01-06 |    2 | BOSQUE  | 30777780 | HARDWARE
 2006-02-06 |    1 | ALAMO   | 00003443 | COMUNICACIONES
 2006-02-06 |    2 | ALAMO   | 35467080 | COMUNICACIONES
 2006-02-06 |    3 | SAUCE   | 30506080 | COMUNICACIONES
 2006-02-06 |    4 | BOSQUE  | 33344550 | COMUNICACIONES
 2006-03-06 |    1 | SAUCE   | 30777780 | BASES DE DATOS
 2006-03-06 |    2 | BOSQUE  | 30777780 | INTELIGENCIA ARTIFICIAL
 2006-04-06 |    1 | ROBLE   | 30506080 | OFIMATICA
 2006-04-06 |    2 | ROBLE   | 00003443 | OFIMATICA
 2006-04-06 |    3 | ALAMO   | 35467080 | SISTEMAS OPERATIVOS
(11 rows)
```


Universidad de Alcalá

Ejemplo

- Comprobamos los permisos de escritura intentando insertar una fila en la tabla sala:

```
congresos=> \d sala
                        Table "public.sala"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 nomsala   | character varying(15) |           | not null |
 capacidad | integer               |           |          |
 precio    | numeric(10,0)         |           |          |
Indexes:
    "salas_pkey" PRIMARY KEY, btree (nomsala)
Check constraints:
    "salas_capacidad_check" CHECK (capacidad >= 25 AND capacidad <= 99)
    "salas_precio_check" CHECK (precio >= 100000::numeric AND precio <= 200000::numeric)
Referenced by:
    TABLE "sesion" CONSTRAINT "sala_fk" FOREIGN KEY (nomsala) REFERENCES sala(nomsala) MATCH Fl
UPDATE CASCADE ON DELETE RESTRICT

congresos=> INSERT INTO sala VALUES('sala_nueva',30,200);)
ERROR:  permission denied for table sala
congresos->
```


Universidad de Alcalá