PostgreSQL szerepkörök

Általános tulajdonságok

PostgreSQL az adatbázis-hozzáférési jogosultságokat a szerepkörök (role) alapján kezeli. Egy szerepkör felfogható adatbázis-felhasználóként vagy adatbázis-felhasználók csoportjaként, attól függően, hogyan van beállítva. A szerepkörök birtokolhatnak adatbázis-objektumokat (például táblákat és függvényeket), és jogosultságokat adhatnak ezekre az objektumokra más szerepköröknek annak szabályozására, hogy ki mely objektumokhoz férhet hozzá. Emellett lehetőség van arra is, hogy egy szerepkör tagságát egy másik szerepkörnek adjuk, így a tag szerepkör használhatja egy másik szerepkörhöz rendelt jogosultságokat.

A szerepkörök fogalma magába foglalja a „felhasználók” és „csoportok” korábbi fogalmait. A PostgreSQL 8.1 előtti verzióiban a felhasználók és csoportok különálló entitások voltak, de ma már csak szerepkörök léteznek. Bármely szerepkör működhet felhasználóként, csoportként, vagy akár mindkettőként. PostgreSQL szerverben.

Gyakorlati példák

Egyszerű „felhasználó” szerepkör létrehozása. Felhasználónak az a szerepkör tekinthető, amelynek van `LOGIN` tulajdonsága. A szerepkör létrehozása többféle módszerrel történhet.

```
[root@gepnev ~]# su - postgres

[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS


[postgres@gepnev ~]$ createuser test

[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
 test      |

[postgres@gepnev ~]$ dropuser test

[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
```

```
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
```

Másik módszerrel ugyanez

```
[root@gepnev ~]# su - postgres

[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS

[postgres@gepnev ~]$ psql -c 'CREATE ROLE test LOGIN'
CREATE ROLE
[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
 test      |

[postgres@gepnev ~]$ psql -c 'DROP ROLE test'
DROP ROLE
[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
```

A `CREATE USER name` megegyezik a `CREATE ROLE name LOGIN` paranccsal, azzal a különbséggel, hogy automatikusan tartalmazza a LOGIN attribútumot.

```
[postgres@gepnev ~]$ psql -c "CREATE ROLE test LOGIN PASSWORD 'test'"
CREATE ROLE
[postgres@gepnev ~]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
 test      |

[postgres@gepnev ~]$ psql -c "SELECT * FROM pg_roles WHERE rolname = 'test'"
 rolname | rolsuper | rolinherit | rolcreaterole | rolcreatedb | rolcanlogin
| rolreplication | rolconnlimit | rolpassword | rolvaliduntil | rolbypassrls
| rolconfig |  oid
---------+----------+------------+---------------+-------------+------------
-+----------------+--------------+-------------+---------------+------------
--+-----------+-------
 test    | f        | t          | f             | f           | t
| f              |           -1 | ********    |               | f
|           | 16390
```

```
(1 row)

[postgres@gepnev data]$ cat ${PGDATA}/pg_hba.conf
# TYPE DATABASE USER ADDRESS METHOD
local all all trust
host all all 127.0.0.1/32 trust
host all all ::1/128 trust

local replication all peer
host replication all 127.0.0.1/32 ident
host replication all ::1/128 ident


[postgres@gepnev ~]$ psql -U test -d postgres -c "SELECT 'z' < 'á'"
 ?column?
----------
 f
(1 row)

[postgres@gepnev data]$ cat ${PGDATA}/pg_hba.conf
# TYPE DATABASE USER ADDRESS METHOD
local all test md5
local all all trust
host all all 127.0.0.1/32 trust
host all all ::1/128 trust

local replication all peer
host replication all 127.0.0.1/32 ident
host replication all ::1/128 ident

[postgres@gepnev data]$ pg_ctl reload
server signaled

[postgres@gepnev data]$ psql -U test -d postgres -c "SELECT 'z' < 'á'"
Password for user test: xxx
psql: error: connection to server on socket "/run/postgresql/.s.PGSQL.5432"
failed: FATAL:  password authentication failed for user "test"

[postgres@gepnev data]$ psql -U test -d postgres -c "SELECT 'z' < 'á'"
Password for user test:
 ?column?
----------
 f
(1 row)

[postgres@gepnev data]$ psql -c 'DROP ROLE test'
DROP ROLE

[postgres@gepnev data]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
```

```
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
```

Szerepkör hierarchia

Szerepkörök létrehozása

```
[postgres@gepnev data]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS

[postgres@gepnev data]$ psql -c 'CREATE ROLE db_read'
CREATE ROLE

[postgres@gepnev data]$ psql -c 'CREATE ROLE db_write'
CREATE ROLE

[postgres@gepnev data]$ psql -c 'CREATE ROLE db_admin'
CREATE ROLE

[postgres@gepnev data]$ psql -c '\du'
                             List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 db_admin  | Cannot login
 db_read   | Cannot login
 db_write  | Cannot login
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
```

Jogosultságok hozzáadása

```
[postgres@gepnev data]$ psql -c "GRANT SELECT ON ALL TABLES IN SCHEMA public
TO db_read"
GRANT

[postgres@gepnev data]$ psql -c "GRANT INSERT, UPDATE, DELETE ON ALL TABLES
IN SCHEMA public TO db_write"
GRANT

[postgres@gepnev data]$ psql -c "GRANT USAGE, CREATE ON SCHEMA public TO
db_admin"
GRANT

[postgres@gepnev data]$ psql -c "GRANT ALL PRIVILEGES ON ALL TABLES IN
SCHEMA public TO db_admin"
```

```
GRANT

[postgres@gepnev data]$ psql -c "GRANT USAGE, SELECT ON ALL SEQUENCES IN
SCHEMA public TO db_read"
GRANT

[postgres@gepnev data]$ psql -c "GRANT USAGE, SELECT, UPDATE ON ALL
SEQUENCES IN SCHEMA public TO db_write"
GRANT

[postgres@gepnev data]$ psql -c "GRANT ALL PRIVILEGES ON ALL SEQUENCES IN
SCHEMA public TO db_admin"
GRANT

[postgres@gepnev data]$ psql -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT SELECT ON TABLES TO db_read"
ALTER DEFAULT PRIVILEGES

[postgres@gepnev data]$ psql -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT INSERT, UPDATE, DELETE ON TABLES TO db_write"
ALTER DEFAULT PRIVILEGES

[postgres@gepnev data]$ psql -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON TABLES TO db_admin"
ALTER DEFAULT PRIVILEGES

[postgres@gepnev data]$ psql -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT USAGE, SELECT ON SEQUENCES TO db_read"
ALTER DEFAULT PRIVILEGES

[postgres@gepnev data]$ psql -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT USAGE, SELECT, UPDATE ON SEQUENCES TO db_write"
ALTER DEFAULT PRIVILEGES

[postgres@gepnev data]$ psql -c "ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON SEQUENCES TO db_admin"
ALTER DEFAULT PRIVILEGES
```

Szerepkörök egymásba ágyazása

```
[postgres@gepnev data]$ psql -c "GRANT db_read TO db_write"
GRANT ROLE

[postgres@gepnev data]$ psql -c "GRANT db_write TO db_admin"
GRANT ROLE
```

Felhasználók szerepkörei

```
[postgres@gepnev data]$ psql -c "GRANT db_read TO db_write"
GRANT ROLE

[postgres@gepnev data]$ psql -c "GRANT db_write TO db_admin"
GRANT ROLE

[postgres@gepnev data]$ psql -c "CREATE ROLE admin LOGIN"
CREATE ROLE

[postgres@gepnev data]$ psql -c "GRANT db_admin TO admin"
GRANT ROLE

[postgres@gepnev data]$ psql -c "CREATE ROLE developer LOGIN"
CREATE ROLE

[postgres@gepnev data]$ psql -c "GRANT db_write TO developer"
GRANT ROLE

[postgres@gepnev data]$ psql -c "CREATE ROLE monitor LOGIN"
CREATE ROLE

[postgres@gepnev data]$ psql -c "GRANT db_read TO monitor"
GRANT ROLE

[postgres@note02 data]$ psql -c "\du"
                            List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 admin     |
 db_admin  | Cannot login
 db_read   | Cannot login
 db_write  | Cannot login
 developer |
 monitor   |
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
```

Teszteles

```
[postgres@gepnev data]$ psql -U admin -d postgres -c "\dt"
Did not find any tables.

[postgres@gepnev data]$ psql -U admin -d postgres -c "CREATE TABLE users (id
SERIAL PRIMARY KEY, name TEXT)"
CREATE TABLE

[postgres@gepnev data]$ psql -U admin -d postgres -c "\dt"
         List of tables
 Schema | Name  | Type  | Owner
--------+-------+-------+-------
```

```
 public | users | table | admin
(1 row)

[postgres@gepnev data]$ psql -U admin -d postgres -c "DROP TABLE users"
DROP TABLE

[postgres@gepnev data]$ psql -U admin -d postgres -c "\dt"
Did not find any tables.

[postgres@gepnev data]$ psql -U developer -d postgres -c "CREATE TABLE users
(id SERIAL PRIMARY KEY, name TEXT)"
ERROR:  permission denied for schema public
LINE 1: CREATE TABLE users (id SERIAL PRIMARY KEY, name TEXT)
                    ^
[postgres@gepnev data]$ psql -U monitor -d postgres -c "CREATE TABLE users
(id SERIAL PRIMARY KEY, name TEXT)"
ERROR:  permission denied for schema public
LINE 1: CREATE TABLE users (id SERIAL PRIMARY KEY, name TEXT)
                    ^
```

From:
https://wiki.r-l.hu/ - **Reverse-Logic wiki**

Permanent link:
https://wiki.r-l.hu/doku.php?id=postgresql:szerepkorok

Last update: **2026/04/21 01:02**