BSM211 Veritabanı Yönetim Sistemleri - Celal ÇEKEN, İsmail ÖZTEL, Veysel Harun ŞAHİN

Veritabanı Güvenliği, Yetkilendirme

psql Kullanımı

psql, postgresql sunucuya bağlanıp işlemler yapmamızı sağlayan konsol uygulamasıdır.

psql Uygulamasını Çalıştırma

```
psql -U postgres -h localhost

psql -U ayse -d pagila -h localhost
```

```
psql -U postgres

VTYS_Comp:~ vtys$ psql -U postgres
Password for user postgres:
psql (9.5.4)
Type "help" for help.

postgres=#
```

psql ile veritabanlarını görüntüleme

```
postgres=# \l
                                  List of databases
         Name         |  Owner   | Encoding |   Collate   |    Ctype    |   Access
privileges
----------------------+----------+----------+-------------+-------------+---------
--------------
 AlisVerisUygulamasi  | postgres | UTF8     | tr_TR.UTF-8 | tr_TR.UTF-8 |
 Musteri              | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 Northwind            | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 OgrenciBilgiSistemi  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
```

psql ile veritabanına bağlanma

```
postgres=# \c Northwind
psql (10.3, server 12.1)
WARNING: psql major version 10, server major version 12.
```

```
         Some psql features might not work.
You are now connected to database "Northwind" as user "postgres".
```

psql ile tabloları görüntüleme

```
Northwind=# \d+
                                    List of relations
 Schema |              Name               |   Type   |  Owner   |    Size    |
Description
--------+---------------------------------+----------+----------+------------+----
---------
 public | CustomersContactName            | table    | postgres | 16 kB      |
 public | Musteriler                      | table    | postgres | 0 bytes    |
 public | Musteriler_musteriNo_seq        | sequence | postgres | 8192 bytes |
 public | OrderCustomerEmployee           | view     | postgres | 0 bytes    |
 public | OrderCustomerEmployee1          | view     | postgres | 0 bytes    |
 public | OrderCustomerEmployee3          | view     | postgres | 0 bytes    |
 public | OrderCustomerEmployee4          | view     | postgres | 0 bytes    |
 public | SiparisMusteriSatisTemsilcisi   | view     | postgres | 0 bytes    |
 public | UrunDegisikligiIzle             | table    | postgres | 8192 bytes |
 public | UrunDegisikligiIzle_kayitNo_seq | sequence | postgres | 8192 bytes |
 public | categories                      | table    | postgres | 16 kB      |
 public | customercustomerdemo            | table    | postgres | 8192 bytes |
 public | customerdemographics            | table    | postgres | 8192 bytes |
 public | customers                       | table    | postgres | 56 kB      |
 public | employees                       | table    | postgres | 16 kB      |
 public | employeeterritories             | table    | postgres | 8192 bytes |
 public | order_details                   | table    | postgres | 120 kB     |
 public | orders                          | table    | postgres | 144 kB     |
 public | products                        | table    | postgres | 8192 bytes |
```

psql ile sql sorgusu çalıştırma

```
Northwind=# select "ProductID", "ProductName"  from products;
 ProductID |           ProductName
-----------+---------------------------------
         5 | Chef Anton's Gumbo Mix
         7 | Uncle Bob's Organic Dried Pears
        12 | Queso Manchego La Pastora
        13 | Konbu
```

psql - Kullanıcıları/Rolleri Listeleme

```
postgres=# \du
                                   List of roles
```

```
 Role name |                         Attributes                         | Member
of

-----------+------------------------------------------------------------+---------
--

 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

 testk1    |                                                            | {}
```

psql Çıkış Örneği

```
postgres=# \q
postgres=#
VTYS_Comp:~ vtys$
```

Yetki/Yetkilendirme

- Nesnelerle ilgili sahip olunan haklara yetki denir.
- Kullanıcıların/rollerin, veritabanı yönetim sistemi ve veritabanı nesneleri (tablo, görünüm, fonksiyon vb.) üzerinde hangi haklara sahip olacağının belirlenmesine yetkilendirme adı verilir.
- İki tür yetki vardır:
    - Temel yetkiler (rol özellikleri)
    - Nesne yetkileri

Temel Yetkiler (Rol Özellikleri)

- Kullanıcıların/rollerin veritabanı yönetim sistemi üzerindeki haklarını ifade eder.
    - SUPERUSER, CREATEDB, CREATEROLE, CREATEUSER, INHERIT, LOGIN, REPLICATION, BYPASSRLS, NOSUPERUSER, NOCREATEDB, NOCREATEROLE, NOCREATEUSER, NOINHERIT, NOLOGIN, NOREPLICATION, NOBYPASSRLS

Nesne Yetkileri

- Kullanıcıların/rollerin veritabanı nesneleri (tablo, görünüm, fonksiyon vb.) üzerindeki haklarını ifade eder.
- Nesnelerin türüne göre (tablo, görünüm, fonksiyon) aşağıdaki yetkiler verilir.
    - SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER,
    - CREATE, CONNECT, TEMPORARY, EXECUTE, USAGE ...

Kullanıcı/Rol Katalogları

- Oturum yetkilendirmesini postgres rolü olarak ayarla (Bu işlemin yapılabilmesi için superuser yetkisi gereklidir).
- Böylelikle işlemler, postgres rolünün yetkileriyle yapılabilecektir.

```
SET SESSION AUTHORIZATION "postgres";
```

- pg_authid ve pg_roles kataloğunu sorgula.
- Bu kataloglarda roller hakkında bilgi mevcuttur.

```
SELECT * FROM "pg_authid";
```

```
SELECT * FROM "pg_roles";
```

- pg_user kataloğunu sorgula.

```
SELECT * FROM "pg_user";
```

Kullanıcı/Rol Oluşturma/Değiştirme İşlemleri

- Hiç bir yetkisi olmayan rol oluştur.
- Rolün aynı zamanda şifesi de mevcut değildir.

```
CREATE ROLE "rol1";
```

- SUPERUSER yetkisi olan rol oluştur.
- SUPERUSER, nesnelerle ilgili herseyi yapma yetkisine sahiptir.

```
CREATE ROLE "rol2" WITH SUPERUSER;
```

- Roller oluşturulduktan sonra düzenlenebilir.

```
ALTER ROLE "rol1" WITH SUPERUSER CREATEDB;
```

```
ALTER ROLE "rol1" WITH NOSUPERUSER;
```

```
ALTER ROLE "rol1" WITH LOGIN;
```

```
ALTER ROLE "rol1" WITH NOLOGIN;
```

- abc şifresine sahip "kullanici1" adında bir kullanıcı oluştur.
- abc şifresi MD5 algoritması ile kodlanır (Kullanılacak algoritma postgresql.conf içerisinde değiştirilebilir).

```
CREATE USER "kullanici1" WITH PASSWORD 'abc';
```

- CREATE USER, CREATE ROLE ifadesinin bir takma isimdir.
- Aralarındaki fark LOGIN seçeneğidir.
- CREATE USER ifadesi varsayılan olarak LOGIN yetkili rol oluşturur.
- CREATE ROLE ifadesi varsayılan olarak LOGIN yetkisi olmayan rol oluşturur.
- 8.1 ile birlikte user ve group kavramı yerine rol kavramı getirildi.
- Bir rol, user olabilir, group olabilir veya ikisi birden olabilir.

```
CREATE ROLE "rol3" WITH PASSWORD 'abc' LOGIN;
```

- Şifre kodlanarak saklanır.
- Şifrenin son geçerlilik tarihi de belirtilir.

```
CREATE ROLE "kullanici4" WITH PASSWORD 'abc' VALID UNTIL '2020-01-01';
```

psql - Kullanıcı/Rol Oluşturma/Değiştirme İşlemleri

- Örnek

```
postgres=# CREATE USER testk1 WITH PASSWORD '111111';

CREATE ROLE

postgres=#

postgres=# \du

                                   List of roles

 Role name |                         Attributes                         | Member
of

-----------+------------------------------------------------------------+---------
```

```
--
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testk1    |                                                            | {}

postgres=#
```

- Örnek
    - `CREATE USER` ifadesi, `CREATE ROLE` ifadesinin bir takma isimdir.
    - Aralarındaki fark `LOGIN` seçeneğidir.
    - https://www.postgresql.org/docs/current/sql-createrole.html

```
postgres=# CREATE ROLE testk2 WITH PASSWORD '111111';
CREATE ROLE
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member
of
-----------+------------------------------------------------------------+---------
--
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testk1    |                                                            | {}
 testk2    | Cannot login                                               | {}

postgres=#
```

- Örnek

```
postgres=# CREATE ROLE testk3 WITH PASSWORD '111111' LOGIN;
CREATE ROLE
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member
of
-----------+------------------------------------------------------------+---------
--
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testk1    |                                                            | {}
 testk2    | Cannot login                                               | {}
 testk3    |                                                            | {}

postgres=#
```

- Örnek

```
postgres=# CREATE ROLE testk4 WITH PASSWORD '111111' LOGIN CREATEDB;
CREATE ROLE
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member
of
-----------+------------------------------------------------------------+---------
--
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testk1    |                                                            | {}
 testk2    | Cannot login                                               | {}
 testk3    |                                                            | {}
 testk4    | Create DB                                                  | {}

postgres=#
```

- Örnek

```
postgres=# ALTER ROLE testk2 WITH LOGIN;
ALTER ROLE
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member
of
-----------+------------------------------------------------------------+---------
--
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testk1    |                                                            | {}
 testk2    |                                                            | {}
 testk3    |                                                            | {}
 testk4    | Create DB                                                  | {}

postgres=#
```

- Örnek

```
postgres=# DROP ROLE testk2;
DROP ROLE
postgres=# \du
                                   List of roles
 Role name |                         Attributes                         | Member
of
-----------+------------------------------------------------------------+---------
--
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 testk1    |                                                            | {}
 testk3    |                                                            | {}
 testk4    | Create DB                                                  | {}
```

```
postgres=#
```

Kullanıcı/Rol Silme İşlemleri

```
DROP USER "kullanici1";
```

```
DROP ROLE "rol1";
```

- Nesne oluşturulurken, CREATE komutunu çalıştıran rol, bu nesnenin sahibi olarak belirlenir.
- Nesne sahibi (ya da SUPERUSER) nesne üzerindeki tüm haklara sahiptir.
- Veritabanı sahibi olan bir rolü silmeden önce veritabanı sahipliğini başka bir role aktarmalıyız.
- Bu işlemi ALTER DATABASE ile yapabiliriz.

```
CREATE ROLE "rol1";
```

```
ALTER DATABASE "NorthWind" OWNER TO "rol1";
```

```
DROP ROLE "rol1";

-- Kernel error: ERROR:  role "rol1" cannot be dropped because some objects depend
on it DETAIL: owner of database NorthWind
```

```
ALTER DATABASE "NorthWind" OWNER TO "postgres";
```

```
DROP ROLE "rol1";
```

- Veritabanı sahibi olan bir rolü silmeden önce veritabanı sahipliğini başka bir role aktarmalıyız.
- Bu işlemi REASSIGN ile de yapabiliriz.

```
CREATE ROLE "rol1";
```

```
ALTER DATABASE "NorthWind" OWNER TO "rol1";
```

```
DROP ROLE "rol1";

-- Kernel error: ERROR:  role "rol1" cannot be dropped because some objects depend
on it DETAIL:  owner of database NorthWind
```

- "rol1" in sahibi olduğu tüm nesnelerin yeni sahibini "postgres" olarak belirle.

```
REASSIGN OWNED BY "rol1" TO "postgres";
```

```
DROP ROLE "rol1";
```

- Bir rolü grup gibi kullanabiliriz.
- Diğer rollerin bu rolden yetkilerini kalıtım olarak almasını temin edebiliriz.

```
CREATE ROLE "gruprol";
```

```
CREATE ROLE "rol1";
```

- rol1 isimli rolün yetkilerine gruprol isimli rolün yetkilerini de ekle.
- Temel yetkiler kalıtım olarak alınmaz. Yalnızca grubun nesneler üzerindeki yetkileri kalıtım olarak alınır.

```
GRANT "gruprol" TO "rol1";
```

- Bunun yapılabilmesi için rol1 isimli rolün kalıtım alma özelliğine sahip olması gerekir. (Postgresql in yeni sürümlerinde rol oluşturulduğunda INHERIT yetkisi veriliyor)
- Diğer bir deyişle INHERIT yetkisine sahip olması gerekir.
- Bu yetki yoksa, yetkiler kalıtım alınmaz.

```
ALTER ROLE "rol1" WITH INHERIT;
```

```
CREATE ROLE "rol2" WITH INHERIT;
```

```
GRANT "gruprol" TO "rol2";
```

- rol1 isimli role verilmiş yetkilerin (gruprol yetkileri) geri alınması.

```
REVOKE "gruprol" FROM "rol1";
```

Yetkilendirme İşlemleri

> PUBLIC: Tüm roller / kullanıcılar.
> kullaniciAdi: Tek bir kullanıcı.
> ALL: Tüm yetkiler.

- rol1 isimli role customers tablosu üzerinde seçim yapma yetkisi ver.

```
GRANT SELECT ON "customers" TO "rol1";
```

- Tüm rollere customers tablosu üzerinde kayıt ekleme yetkisi ver.

```
GRANT INSERT ON "customers" TO PUBLIC;
```

- rol1 isimli kullanıcıya customers tablosu üzerinde tüm yetkileri ver.

```
GRANT ALL ON "customers" TO "rol1";
```

- rol1 isimli rolün customers tablosu üzerindeki güncelleme yetkisini geri al.

```
REVOKE UPDATE ON "customers" FROM "rol1";
```

- rol1 isimli rolün customers tablosu üzerindeki tüm yetkilerini geri al.

```
REVOKE ALL ON "customers" FROM "rol1";
```

- rol1 kullanicisinin Sema1 icerisindeki nesnelere ait tüm yetkileri geri alinir.

```
REVOKE ALL ON SCHEMA "Sema1" FROM "rol1";
```

- Herhangi bir nesne üzerinde yetkiye sahip olan bir rolü silemeyiz.

```
CREATE ROLE "rol1";
```

```
GRANT SELECT ON "customers" TO "rol1";
```

```
DROP ROLE "rol1";

-- Kernel error: ERROR:  role "rol4" cannot be dropped because some objects depend
on it DETAIL:  privileges for table customers
```

- rol1 in sahibi olduğu tüm nesneleri sil (kısıtlar ihlal edilemez)

```
DROP OWNED BY "rol1";
```

```
DROP ROLE "rol1";
```

Örnek (Northwind Veri Tabanı)

```
CREATE ROLE "rol1";
```

```
SET SESSION AUTHORIZATION "rol1";
```

```
SELECT * FROM "customers";

-- Kernel error: ERROR:  permission denied for relation customer
```

- rol1 seçme hakkına sahip olmadigi icin hata oluşur.

```
GRANT SELECT ON "customers" TO "rol1";
```

- Yetkilendirme yapabilmek için oturum yetkilendirmesini "postgres" kullanıcısı şeklinde ayarla.

```
SET SESSION AUTHORIZATION "postgres";
```

- rol1, kullanıcısına "customers" tablosu üzerinde seçme yetisi ver.

```
GRANT SELECT ON "customers" TO "rol1";
```

```
SET SESSION AUTHORIZATION "rol1";
```

- Sorgu çalışır.

```
SELECT * FROM "customers";
```

Fonksiyonlar ve Yetkilendirme

```
CREATE OR REPLACE FUNCTION "milKMDonustur"(degerMil REAL, OUT degerKM REAL)
AS $$
BEGIN
    degerKM = degerMil * 1.6;
END;
$$
LANGUAGE plpgsql;
```

```
SET SESSION AUTHORIZATION "rol1";
```

```
SELECT * FROM "milKMDonustur"(3);
```

```
SET SESSION AUTHORIZATION "postgres";
```

```
REVOKE ALL ON FUNCTION "milKMDonustur"(REAL, OUT REAL) FROM "rol1";
```

- Aşağıdaki ifade çalışır. Fonksiyonlar PUBLIC grubu için varsayılan olarak çalıştırılırlar.

```
SELECT * FROM milKMDonustur(3);
```

```
SET SESSION AUTHORIZATION "postgres";
```

```
REVOKE ALL ON FUNCTION "milKMDonustur"(REAL, OUT REAL) FROM PUBLIC;
```

```
SET SESSION AUTHORIZATION "rol1";
```

- Aşağıdaki ifade çalışmaz.

```
SELECT * FROM milKMDonustur(3);
```

Şifreleme

- Kullanici şifreleri ve gizli bilgiler açık olarak saklanmamalıdır.
- Linux

```
sudo apt-get install postgresql-contrib
```

- Kripto eklentisini oluştur.

```
CREATE EXTENSION "pgcrypto";
```

- "sifrem" şifresini sha512 algoritması ile kodla.

```
SELECT ENCODE(DIGEST('sifrem', 'sha512'), 'hex');
```

- "sifrem" şifresini md5 algoritması ile kodla ve sonucu "md5" ifadesi ile birleştir.

```
SELECT 'md5'|| MD5('sifrem');
```