

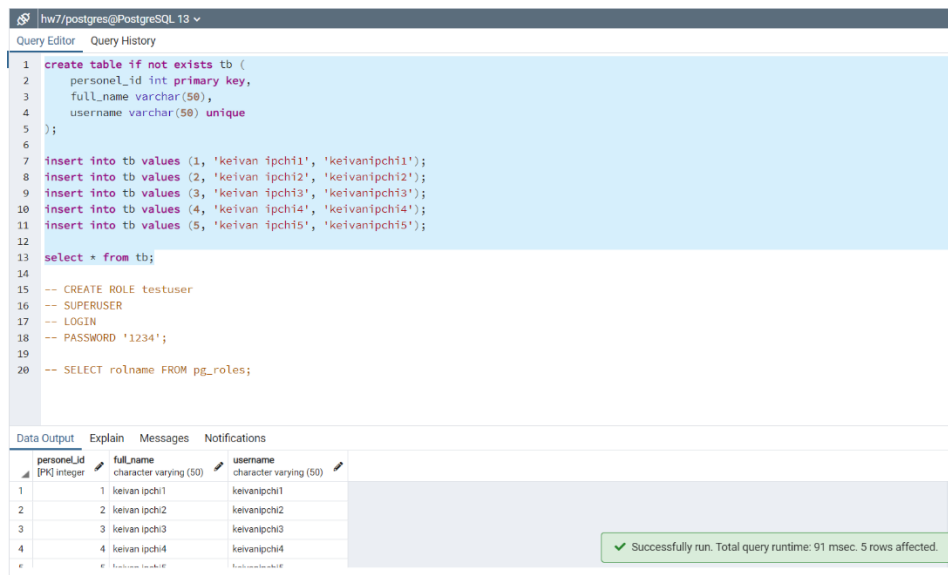
Database Lab – HW7

(سوال اول)

تفاوت بین این دو دستور آن است که CREATE USER یک کاربر می سازد ولی این کاربر هیچ دسترسی اولیه ای به پایگاه داده ها ندارد. دستور CREATE ROLE یک نقش جدید میسازد که سطح دسترسی کاربران را مشخص میکند. میتوانید بعد تر ROLE را به USER دهیم.

(سوال دو)

ساخت جدول، اضافه کردن دیتا:



```

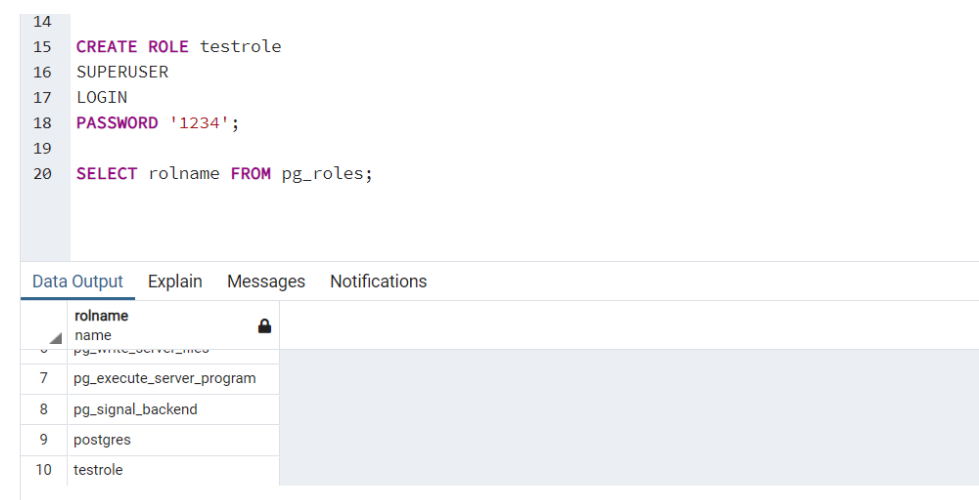
1 create table if not exists tb (
2     personel_id int primary key,
3     full_name varchar(50),
4     username varchar(50) unique
5 );
6
7 insert into tb values (1, 'keivan ipchi1', 'keivanipchi1');
8 insert into tb values (2, 'keivan ipchi2', 'keivanipchi2');
9 insert into tb values (3, 'keivan ipchi3', 'keivanipchi3');
10 insert into tb values (4, 'keivan ipchi4', 'keivanipchi4');
11 insert into tb values (5, 'keivan ipchi5', 'keivanipchi5');
12
13 select * from tb;
14
-- CREATE ROLE testuser
-- SUPERUSER
-- LOGIN
-- PASSWORD '1234';
-- SELECT rolname FROM pg_roles;

```

personel_id [PK] integer	full_name character varying (50)	username character varying (50)
1	keivan ipchi1	keivanipchi1
2	keivan ipchi2	keivanipchi2
3	keivan ipchi3	keivanipchi3
4	keivan ipchi4	keivanipchi4

Successfully run. Total query runtime: 91 msec. 5 rows affected.

ساخت یک role:



```

14
15 CREATE ROLE testrole
16 SUPERUSER
17 LOGIN
18 PASSWORD '1234';
19
20 SELECT rolname FROM pg_roles;

```

rolname name
pg_wal_server_name
pg_execute_server_program
pg_signal_backend
postgres
testrole

Successfully run. Total query runtime: 91 msec. 5 rows affected.

ساخت یک group role:

Database Lab – HW7

```

21
22 -- SELECT rolname FROM pg_roles;
23
24
25
26 create role group_role
27 SUPERUSER
28 VALID UNTIL '2030-02-03';
29
30 SELECT rolname FROM pg_roles;

```

Data Output Explain Messages Notifications

rolname	name
pg_signal_backend	
postgres	
testrole	
group_role	

✓ Successfully run. Total q

ساخت دو role و اضافه کردنشان به group role:

```

33
34 create role testrole1 LOGIN;
35 create role testrole2 LOGIN;
36 GRANT group_role to testrole1;
37 GRANT group_role to testrole2;

```

Data Output Explain Messages Notifications

GRANT ROLE

Query returned successfully in 36 msec.

✓ Query returned successfully in 36 msec.

دریافت ارور برای select کردن:

```

hw7-> \q
hw7-> \q
C:\Users\Keivan>psql -U testrole1 hw7
Password for user testrole1:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

hw7=> select * from tb;
ERROR: permission denied for table tb
hw7=>

```

ied suc

لیست نقش ها:

Database Lab – HW7

```

ERROR: permission denied for table tb
hw7=> \du

               List of roles
Role name | Attributes | Member of
-----+-----+-----
group_role | Superuser, Cannot login | + | {}
           | Password valid until 2030-02-03 00:00:00+03:30 | |
newtstrole2 | | | {group_role}
postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS | | {}
testrole   | Superuser | | {}
testrole1  | Bypass RLS | | {group_role}

hw7=>

```

دادن RLS به سطر ها:

```

43
44 ALTER TABLE tb ENABLE ROW LEVEL SECURITY;
45 CREATE POLICY test_roles
46 ON tb
47 TO group_role;

```

Data Output Explain Messages Notifications

CREATE POLICY

Query returned successfully in 95 msec.

این سیاست روی test_roles که یک group role بود دسترسی ایجاد میکند. نقش testrole1 که از test_roles ارث بری میکند (خودمان ست کردیم در بخش قبلی) پس دسترسی دارد.

پاک کردن newtstrole2:

```

48
49 drop role newtstrole2;

```

Data Output Explain Messages Notifications

DROP ROLE

Query returned successfully in 35 msec.

✓ Query returned successfully in 35 msec.

سوال سه)

اگر فقط یک superuser داشته باشیم که توسط backend نقش ها را مدیریت کند در صورت هک شدن یا باگ امنیتی هکر یا آن شخص میتواند همه نقش ها را کنترل کند. در نتیجه خود را superuser کند و دسترسی بقیه را بگیرد. متن قبلی یک مثال بود.