

```

postgres=# \du
                                List of roles
 Role name |
Member    | Attributes
of
-----+-----
postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
{}

```

```

postgres=#
postgres=# select * from pg_shadow;
 username | usesysid | usecreatedb | usesuper | userepl | usebypassrls |
          |          |             |          |         |              |
-----+-----+-----+-----+-----+-----+-----
 postgres |         10 | t           | t        | t       | t           |
 md50005  |
 a4b0ccaccbd18bcf64e90e242c90 |
 (1 row)

```

```

postgres=#
postgres=#
postgres=#
postgres=# create role developer;
CREATE ROLE
postgres=# \q
-bash-4.2$ psql -U developer
Password for user developer:
psql: error: fe_sendauth: no password supplied
-bash-4.2$
-bash-4.2$ psql
Password for user postgres:
psql (12.8)
Type "help" for help.

```

```

postgres=#
postgres=# alter user developer password 'postgres';
ALTER ROLE
postgres=#
ALTER ROLE
postgres=# \q
-bash-4.2$
-bash-4.2$ psql -U developer
Password for user developer:
psql: error: FATAL:  role "developer" is not permitted to log in
-bash-4.2$
-bash-4.2$ psql
Password for user postgres:
psql (12.8)
Type "help" for help.
postgres=#

```

```

postgres=# create table t1(id int);
CREATE TABLE
postgres=# select current_user,current_database();
  current_user | current_database
-----+-----
 postgres     | postgres
(1 row)

postgres=#
postgres=# \d
          List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
 public | t1   | table | postgres
(1 row)

postgres=# \du
                                List of roles
 Role name |                               Attributes
Member
of
-----+-----
+-----
----
 developer | Cannot login
{}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
{}

postgres=#
postgres=# grant insert,select on table t1 to developer;
GRANT
postgres=#
postgres=# -----create users dev1,dev2 and grant developer role to
the users-----;
postgres=#
postgres=# create user dev1 password 'postgres';
CREATE ROLE
postgres=# create user dev2 password 'postgres';
CREATE ROLE
postgres=#
postgres=# \du
                                List of roles
 Role name |                               Attributes
Member
of
-----+-----
+-----
----
 dev1      |
{}
 dev2      |
{}

```

```

    developer | Cannot login |
{}
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS |
{}

```

```

postgres=# ---no special privilege nor member of any role---;
postgres=#
postgres=# grant developer to dev1,dev2;
GRANT ROLE
postgres=#
postgres=#
postgres=# \du

```

```

                                List of roles
Role name |                               Attributes |
Member of |
-----+-----
dev1      |                               |
{developer}
dev2      |                               |
{developer}
developer | Cannot login |
{}
postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS |
{}

```

```

postgres=# ---u can see users are member of developer role---;
postgres=#
postgres=# \q
-bash-4.2$ psql -U dev1
Password for user dev1:
psql: error: FATAL:  database "dev1" does not exist
-bash-4.2$
-bash-4.2$ psql -U dev1
Password for user dev1:
psql: error: FATAL:  database "dev1" does not exist
-bash-4.2$ psql -U dev1 -d postgres
Password for user dev1:
psql (12.8)
Type "help" for help.

```

```

postgres=> -----notice the error, i have to mention db name to login---;
postgres=> select current_user,current_database();
 current_user | current_database
-----+-----
dev1          | postgres
(1 row)

```

```
postgres=>
```

```

postgres=> \d
          List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----

```

```
public | t1 | table | postgres
(1 row)
```

```
postgres=> select * from t1;
 id
----
(0 rows)
```

```
postgres=> delete from t1;
ERROR: permission denied for table t1
postgres=>
postgres=> ---select operation allowed but not delete---;
postgres=>
postgres=> insert into t1 values(1);
INSERT 0 1
postgres=> ---insert is also allowed---;
postgres=>
postgres=> create table t2(id int);
CREATE TABLE
postgres=> \d
```

```
      List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
 public | t1  | table | postgres
 public | t2  | table | dev1
(2 rows)
```

```
postgres=> ---look the t2 table(owner dev1 and schema public)----;
postgres=>
postgres=#
```

```
*****
****
```