

How to Change Ownership in PostgreSQL?

written by sysadmin | 19 November 2025

Ownership of a database and the tables in it in PostgreSQL is very crucial because this has a big influence on users who want to carry out CRUD (Create, Read, Update, Delete) activities in a PostgreSQL database. If the user is not the owner of the database and the tables in it, then the user will not be able to perform CRUD.

Problem

How to change ownership in PostgreSQL?

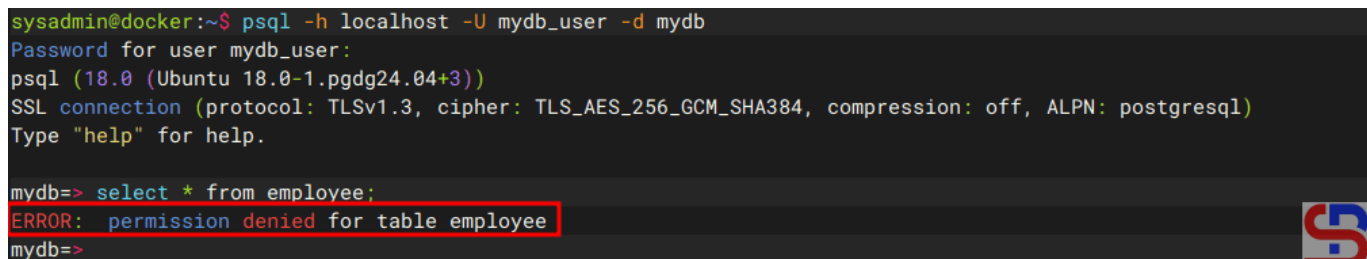
Solution

For example, I have a mydb database and mydb_user as a user. When I access the database using the user mydb_user and perform a query:

```
select * from employee;
```

There is an error as shown below:

ERROR: permission denied for table employee

A screenshot of a terminal window showing a PostgreSQL session. The user 'sysadmin' is connected to a 'docker' environment. They run 'psql -h localhost -U mydb_user -d mydb'. The prompt changes to 'psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))'. They enter the query 'select * from employee;'. The output is 'ERROR: permission denied for table employee'. The error message is highlighted with a red box. A small logo is visible in the bottom right corner of the terminal window.

```
sysadmin@docker:~$ psql -h localhost -U mydb_user -d mydb
Password for user mydb_user:
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

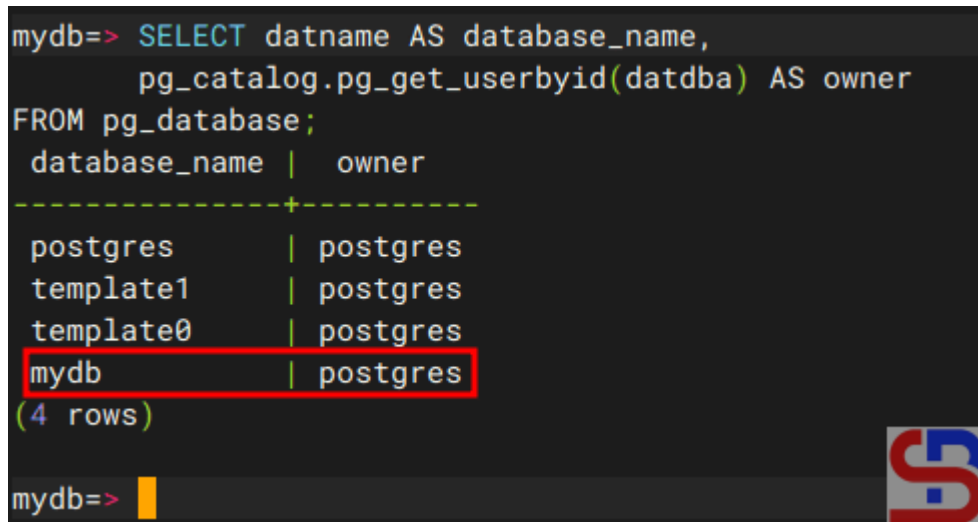
mydb=> select * from employee;
ERROR: permission denied for table employee
mydb=>
```

Error permission denied

After I searched on the internet, it turned out that this was caused by the user mydb_user not being the owner of the mydb database and the tables in it. To see the owner of the

database, you can use the query below:

```
SELECT datname AS database_name,  
       pg_catalog.pg_get_userbyid(datdba) AS owner  
FROM pg_database;
```



The screenshot shows a PostgreSQL terminal window with the following output:

```
mydb=> SELECT datname AS database_name,  
       pg_catalog.pg_get_userbyid(datdba) AS owner  
FROM pg_database;  
 database_name | owner  
-----+-----  
 postgres      | postgres  
 template1     | postgres  
 template0     | postgres  
 mydb          | postgres  
(4 rows)
```

The row for 'mydb' is highlighted with a red rectangle. The terminal prompt 'mydb=>' is followed by a yellow cursor bar. A small logo is visible in the bottom right corner of the terminal window.

List the owners

You can see in the picture above, the postgres user is the owner of the mydb database. This is because when creating the database uses the postgres user and not the mydb_user.

INFO

You can also use an alternative query like the one below to see the owner of your database:

```
SELECT d.datname AS database_name, r.rolname AS owner  
FROM pg_database d  
JOIN pg_roles r ON d.datdba = r.oid  
WHERE d.datname = 'your_database_name';
```

or using the simple query:

```
\l
```

Similarly, if you want to see the owners of the tables in the database (but you must first log in to that database to run this query), use the query below:

```
SELECT tablename, tableowner
```

```
FROM pg_tables
WHERE schemaname = 'public';
```

Then there will be a display like the following:

```
postgres=# \c mydb;
You are now connected to database "mydb" as user "postgres".
mydb=# SELECT tablename, tableowner
FROM pg_tables
WHERE schemaname = 'public';
 tablename | tableowner
-----+-----
 employee  | postgres
(1 row)

mydb=#
```



Show the owner of the tables

You can see in the image above that the owner of the table employee in the mydb database is a postgres user. This is because when creating the tables uses user postgres and user mydb.

INFO

You can also use a query like the one below to see the ownership of all tables in PostgreSQL:

```
SELECT
    schemaname,
    tablename,
    tableowner
FROM pg_tables
ORDER BY schemaname, tablename;
```

A. Change database owner

If you want to change ownership of the mydb database from user postgres to user mydb_user, then you have to log in to PostgreSQL as an existing user, which in the case of a

postgres user:

```
ALTER DATABASE mydb OWNER to mydb_user;
```

Run the query below to check ownership of the database:

```
\l
```

```
mydb=# \l
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
mydb	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres + postgres=CTc/postgres + mydb_user=CTc/postgres
postgres	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres +
template0	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			postgres=CTc/postgres +
template1	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres + postgres=CTc/postgres

(4 rows)

```
mydb=# ALTER DATABASE mydb OWNER to mydb_user;
```

```
ALTER DATABASE
```

```
mydb=# \l
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	Locale	ICU Rules	Access privileges
mydb	mydb_user	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=Tc/mydb_user + mydb_user=CTc/mydb_user
postgres	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres +
template0	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			postgres=CTc/postgres +
template1	postgres	UTF8	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres + postgres=CTc/postgres

(4 rows)

```
mydb=#
```

Change the database owner

You can see that the owner of the mydb database has changed to user mydb_user. But even so, you still can't run CRUD on the database because the tables in the database still belong to the postgres user.

B. Change the owner of the table

Still using the postgres user, log in to the mydb database using the command:

```
\c mydb;
```

Then run the query below to change the table, sequence, and view to mydb_user:

```

DO $$
DECLARE
    r RECORD;
BEGIN
    -- Change the owner of all tables
    FOR r IN SELECT schemaname, tablename FROM pg_tables WHERE schemaname =
'public'
    LOOP
        EXECUTE format('ALTER TABLE %I.%I OWNER TO mydb_user;', r.schemaname,
r.tablename);
    END LOOP;

    -- Change the owner of all sequences
    FOR r IN SELECT sequence_schema, sequence_name FROM
information_schema.sequences WHERE sequence_schema = 'public'
    LOOP
        EXECUTE format('ALTER SEQUENCE %I.%I OWNER TO mydb_user;',
r.sequence_schema, r.sequence_name);
    END LOOP;

    -- Change the owner of all views
    FOR r IN SELECT table_schema, table_name FROM information_schema.views
WHERE table_schema = 'public'
    LOOP
        EXECUTE format('ALTER VIEW %I.%I OWNER TO mydb_user;',
r.table_schema, r.table_name);
    END LOOP;
END $$;

```

```

postgres=# \c mydb;
You are now connected to database "mydb" as user "postgres".
mydb=# DO $$
DECLARE
    obj RECORD;
BEGIN
    -- Change the owner of all tables
    FOR obj IN
        SELECT 'TABLE', schemaname, tablename
        FROM pg_tables
        WHERE schemaname = 'public'
    LOOP
        EXECUTE format('ALTER TABLE %I.%I OWNER TO mydb_user;', obj.schemaname, obj.tablename);
    END LOOP;

    -- Change the owner of all sequences
    FOR obj IN
        SELECT sequence_schema, sequence_name
        FROM information_schema.sequences
        WHERE sequence_schema = 'public'
    LOOP
        EXECUTE format('ALTER SEQUENCE %I.%I OWNER TO mydb_user;', obj.sequence_schema, obj.sequence_name);
    END LOOP;

    -- Change the owner of all views
    FOR obj IN
        SELECT table_schema, table_name
        FROM information_schema.views
        WHERE table_schema = 'public'
    LOOP
        EXECUTE format('ALTER VIEW %I.%I OWNER TO mydb_user;', obj.table_schema, obj.table_name);
    END LOOP;
END;
$$;
DO
mydb=#

```

Query to change the owner of the tables

After that, run the command below to view the ownership of the tables in the mydb database:

```

SELECT tablename, tableowner
FROM pg_tables
WHERE schemaname="public";

```

```

mydb=# SELECT tablename, tableowner
FROM pg_tables
WHERE schemaname = 'public';
 tablename | tableowner
-----+-----
 employee  | mydb_user
(1 row)

mydb=#

```

Change the owner of the tables

From the image above, you can see the owner of the tables in the mydb database is not a postgres user but a mydb_user. So, you should be able to do CRUD using the mydb_user user as shown in the image below:

```
sysadmin@docker:~$ psql -h localhost -U mydb_user -d mydb
Password for user mydb_user:
psql (18.0 (Ubuntu 18.0-1.pgdg24.04+3))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off, ALPN: postgresql)
Type "help" for help.

mydb=> INSERT INTO employee (name,age,city) VALUES ('queen',23,'Florida');
INSERT 0 1
mydb=> select * from employee;
 name | age | city
-----+----+-----
 bob  |  21 | New York
 John |  22 | Chicago
 steve | 22 | Kansas
 trump |  20 | Orlando
 paul  |  20 | Texas
 queen |  23 | Florida
(6 rows)
```

Do CRUD on the database

Note

As a reminder, to create databases and tables in the database, you should not use postgres user but create a new user because it is very dangerous to use postgres user to carry out daily activities in the PostgreSQL database.

References

commandprompt.com
stackoverflow.com
w3resource.com