# Database Security Project

This is a project that I have been working on with one of my colleague at Masters.

## Database: Postgres database under docker container

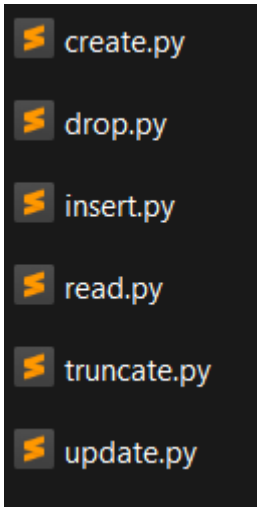
- Run docker container

docker run --name pdb -e POSTGRES_HOST=postgres -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=root -d -p 5432:5432 postgres

```
PS D:\Obsidian\general\Data\CSML\Database Security\project\project> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED       STATUS       PORTS                    NAMES
b470cb325a6a   postgres   "docker-entrypoint.s…"   2 hours ago   Up 2 hours   0.0.0.0:5432->5432/tcp   pdb
```

- Copy scripts in the container

docker cp my_scripts_folder pdb:/my_scripts_folder




```
root@1c3ca7d880aa:/my_scripts_folder# ls -l
total 28
-rwxr-xr-x 1 root root 1435 Jan 25 22:03 create.py
-rwxr-xr-x 1 root root  370 Jan 25 22:03 drop.py
-rwxr-xr-x 1 root root 3238 Jan 26 08:39 insert.py
-rwxr-xr-x 1 root root 1085 Jan 25 22:03 read.py
-rwxr-xr-x 1 root root  540 Jan 25 22:03 truncate.py
-rwxr-xr-x 1 root root  750 Jan 25 22:03 update.py
```

Script Example:

```python
import psycopg2
import os

host = 'localhost'
username = 'postgres'
password = 'root'

conn = psycopg2.connect(
    host=host,
    user=username,
    password=password
)


conn.autocommit = True
cur = conn.cursor()
cur.execute("CREATE DATABASE mydatabase;")
```

- We need to install psycopg2 module for python in order for the scripts to manipulate the database , tables , inserting , reading , truncating and dropping the tables.

```
apt-get update \
    && apt-get -y install nano libpq-dev gcc python3 python3-pip \
    && pip install psycopg2 && pip install psycopg2-binary
```

## Database design and tables

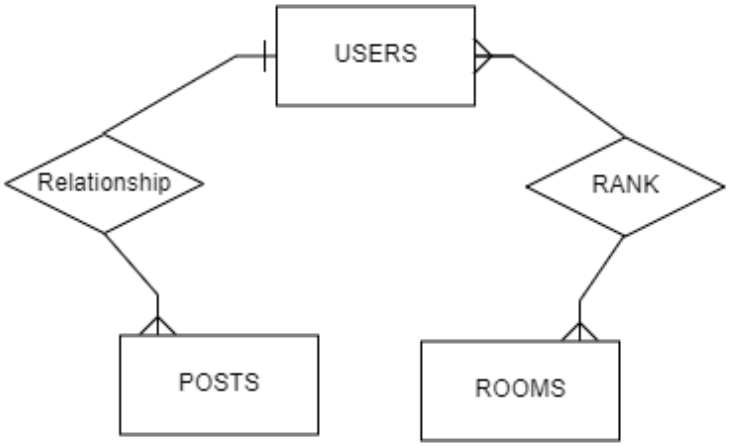
- Users
  - Id : Unique identifier for each user.
  - Username : The alias of an user.
  - Password : Password of the user.
  - Public_key : Public_key of user.
  - Rank : What rank does the user have in every room.
- Posts
  - Post_id : Unique identifier for each post.
  - Creator_id : The user that created the post.
  - Pubdate : Date of publication.
  - Description : Description of publication.
  - Title : Title of publication.
  - Url : Url where it can be found the post.
- Rooms
  - Id_room : Unique identifier for each room.

    - Name_of_room : The name of the chat room.
    - Topic_of_room : The topic of the room.
    - Rules : Rules that should be followed when chatting on the server's room.
- Rank
    - U_id : Unique identifier for each user.
    - Id_room : Unique identifier for each room.
    - Admin :  Flag if the user is an admin on a room or not .

## Entities and Relationships Diagrams:

- Users : one to many with Posts.
- Users: Many to many with Rooms, trough the soon to be created Rank.

A)




B)

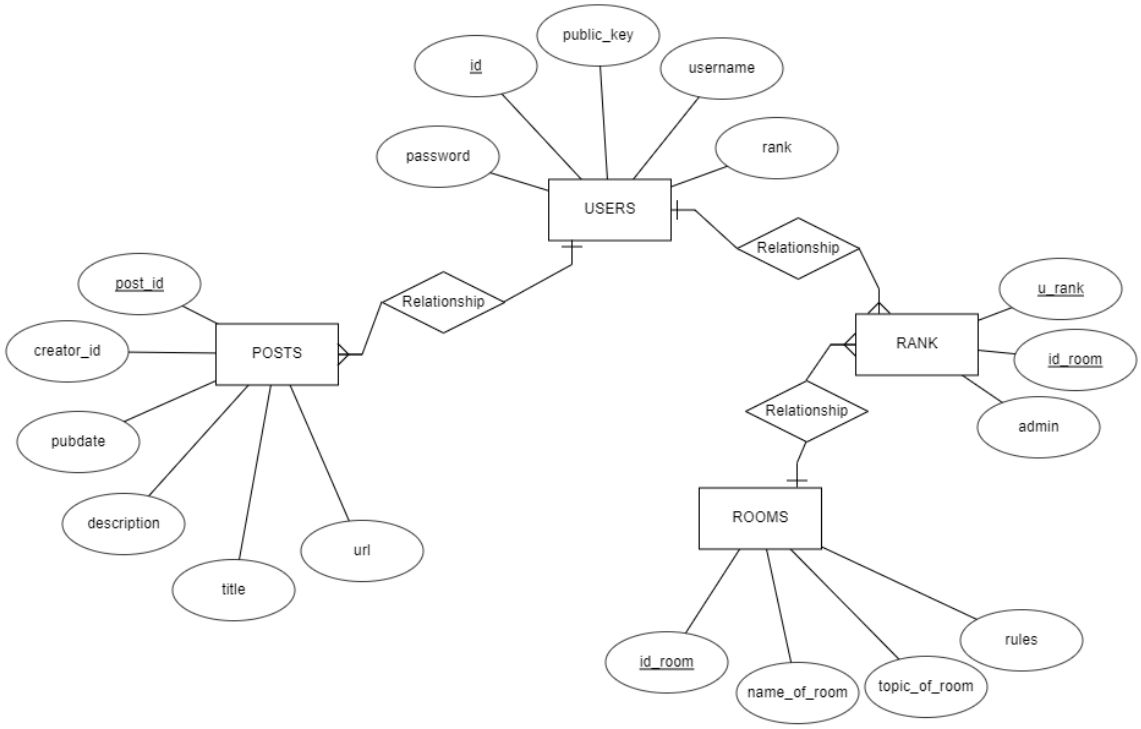
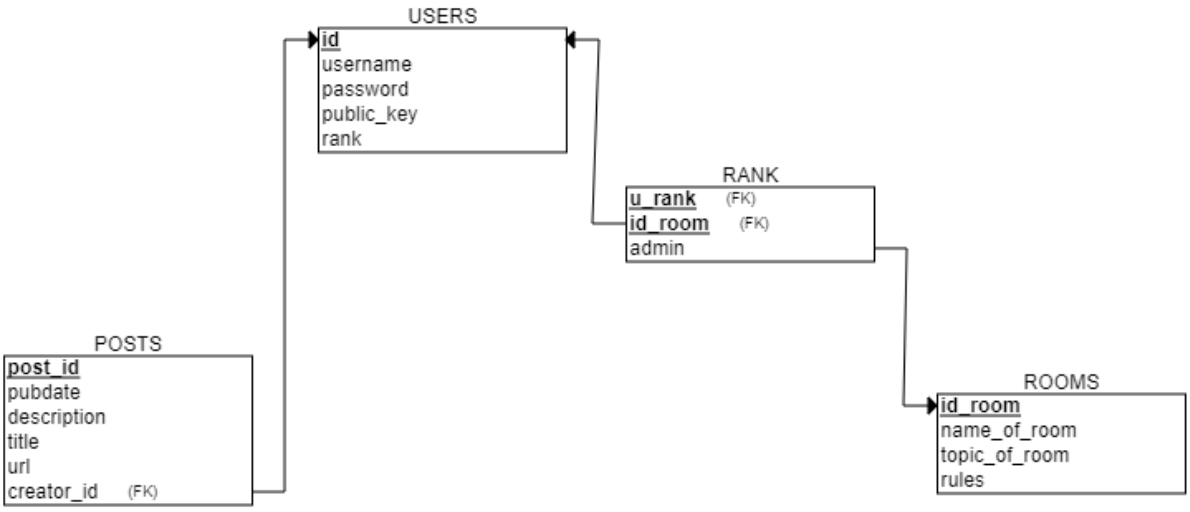



Diagram for the columns, primary keys , foreign keys:




Roles:

- Admin: Admin of the whole database .
- Modder: Will have privileges the same as Admin.
- Helper: Will have just a few privileges on all the tables.
- Student: Will have just SELECT statement over some tables.

SQL script:

- CREATE ROLE admin;
- 
- CREATE ROLE modder;
- 
- CREATE ROLE helper;
- 
- CREATE ROLE student;

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

## Role Based/Discretionary Access Control/Row Level Security

## Sql script:

- ALTER ROLE admin WITH SUPERUSER CREATEDB CREATEROLE LOGIN BYPASSRLS REPLICATION;
- 
- GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO admin WITH GRANT OPTION;
- 
- GRANT SELECT, INSERT, UPDATE, DELETE ON posts,rank,rooms,users TO modder;

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

## Column Level Security:

- GRANT SELECT(username,public_key,rank) ON users to helper;
- GRANT SELECT on posts,rooms,rank to helper;

- GRANT SELECT(id,username) ON users TO student;
- GRANT SELECT(creator_id,pubdate,description,title,url) ON posts TO student;
- GRANT SELECT(name_of_room,topic_of_room,rules) ON rooms TO student;

```
postgres=# \dp
                                  Access privileges
 Schema |       Name        |   Type   |     Access privileges     |  Column privileges   | Policies
--------+-------------------+----------+---------------------------+----------------------+----------
 public | posts             | table    | postgres=arwdDxt/postgres+| creator_id:         +|
        |                   |          | admin=a*r*w*d*/postgres  +|   student=r/postgres+|
        |                   |          | modder=arwd/postgres     +| pubdate:            +|
        |                   |          | helper=r/postgres          |   student=r/postgres+|
        |                   |          |                           | description:        +|
        |                   |          |                           |   student=r/postgres+|
        |                   |          |                           | title:              +|
        |                   |          |                           |   student=r/postgres+|
        |                   |          |                           | url:                +|
        |                   |          |                           |   student=r/postgres  |
 public | posts_post_id_seq | sequence |                           |                       |
 public | rank              | table    | postgres=arwdDxt/postgres+|                       |
        |                   |          | admin=a*r*w*d*/postgres  +|                       |
        |                   |          | modder=arwd/postgres     +|                       |
        |                   |          | helper=r/postgres          |                       |
 public | rooms             | table    | postgres=arwdDxt/postgres+| name_of_room:       +|
        |                   |          | admin=a*r*w*d*/postgres  +|   student=r/postgres+|
        |                   |          | modder=arwd/postgres     +| topic_of_room:      +|
        |                   |          | helper=r/postgres          |   student=r/postgres+|
        |                   |          |                           | rules:              +|
        |                   |          |                           |   student=r/postgres  |
 public | rooms_id_room_seq | sequence |                           |                       |
 public | users             | table    | postgres=arwdDxt/postgres+| id:                 +|
        |                   |          | admin=a*r*w*d*/postgres  +|   student=r/postgres+|
        |                   |          | modder=arwd/postgres      | username:           +|
        |                   |          |                           |   helper=r/postgres +|
        |                   |          |                           |   student=r/postgres+|
        |                   |          |                           | public_key:         +|
        |                   |          |                           |   helper=r/postgres +|
        |                   |          |                           | rank:               +|
        |                   |          |                           |   helper=r/postgres   |
 public | users_id_seq      | sequence |                           |                       |
(7 rows)
```

## Users:

- Chiri: - Modder
- Paul: - Helper
- Chiri_clona: - Student
- Paul_clona: - Student
- Tom: - Admin

## SQL script:

- CREATE USER chiri WITH PASSWORD 'pass';
- 
- CREATE USER paul WITH PASSWORD 'pass';
- 
- CREATE USER chiri_clona WITH PASSWORD 'pass';
- 
- CREATE USER paul_clona WITH PASSWORD 'pass';
- 
- CREATE USER tom WITH PASSWORD 'pass';

## For granting roles for the users:

- GRANT admin TO tom;
- 
- GRANT modder TO chiri;
- 
- GRANT helper TO paul;
- 

- GRANT student TO chiri_clona, paul_clona;

```
postgres=# \du
                                   List of roles
  Role name   |                         Attributes                         | Member of
--------------+------------------------------------------------------------+-----------
 admin        | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 chiri        |                                                            | {modder}
 chiri_clona  |                                                            | {student}
 helper       | Cannot login                                               | {}
 modder       | Cannot login                                               | {}
 paul         |                                                            | {helper}
 paul_clona   |                                                            | {student}
 postgres     | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 student      | Cannot login                                               | {}
 tom          |                                                            | {admin}
```

## Policy:

- CREATE POLICY student_rls_posts ON posts FOR SELECT TO student USING (creator_id = (select id from users where username=current_user));
- ALTER TABLE posts ENABLE ROW LEVEL SECURITY;

```
postgres=# \dp
                                                  Access privileges
 Schema |       Name        |   Type   |     Access privileges     |  Column privileges   |                    Policies
--------+-------------------+----------+---------------------------+----------------------+-----------------------------------------------
 public | posts             | table    | postgres=arwdDxt/postgres+| creator_id:         +| student_rls_posts (r):                      +
        |                   |          | admin=a*r*w*d*/postgres  +|   student=r/postgres+|   (u): (creator_id = ( SELECT users.id       +
        |                   |          | modder=arwd/postgres     +| pubdate:            +|    FROM users                               +
        |                   |          | helper=r/postgres         |   student=r/postgres+|   WHERE ((users.username)::text = CURRENT_USER))+
        |                   |          |                           | description:        +|   to: student
        |                   |          |                           |   student=r/postgres+|
        |                   |          |                           | title:              +|
        |                   |          |                           |   student=r/postgres+|
        |                   |          |                           | url:                +|
        |                   |          |                           |   student=r/postgres |
 public | posts_post_id_seq | sequence |                           |                      |
 public | rank              | table    | postgres=arwdDxt/postgres+|                      |
        |                   |          | admin=a*r*w*d*/postgres  +|                      |
        |                   |          | modder=arwd/postgres     +|                      |
        |                   |          | helper=r/postgres         |                      |
 public | rooms             | table    | postgres=arwdDxt/postgres+| name_of_room:       +|
        |                   |          | admin=a*r*w*d*/postgres  +|   student=r/postgres+|
        |                   |          | modder=arwd/postgres     +| topic_of_room:      +|
        |                   |          | helper=r/postgres         |   student=r/postgres+|
        |                   |          |                           | rules:              +|
        |                   |          |                           |   student=r/postgres |
 public | rooms_id_room_seq | sequence |                           |                      |
 public | users             | table    | postgres=arwdDxt/postgres+| id:                 +|
        |                   |          | admin=a*r*w*d*/postgres  +|   student=r/postgres+|
        |                   |          | modder=arwd/postgres      | username:           +|
        |                   |          |                           |   helper=r/postgres +|
        |                   |          |                           |   student=r/postgres+|
        |                   |          |                           | public_key:         +|
        |                   |          |                           |   helper=r/postgres +|
        |                   |          |                           | rank:               +|
        |                   |          |                           |   helper=r/postgres  |
 public | users_id_seq      | sequence |                           |                      |
(7 rows)
```

Demo:

```
postgres=# select * from users;
 id |  username   | password  | public_key | rank
----+-------------+-----------+------------+------
  1 | chiri       | password1 | publickey1 |    1
  2 | paul        | password2 | publickey2 |    2
  3 | chiri_clona | password3 | publickey3 |    3
  4 | paul_clona  | password4 | publickey4 |    4
  5 | tom         | password5 | publickey5 |    5
(5 rows)

postgres=# select * from posts;
 post_id | creator_id |       pubdate       |       description       |  title  |        url
---------+------------+---------------------+-------------------------+---------+--------------------
       1 |          1 | 2022-01-01 12:00:00 | Description chiri       | Title 1 | http://example1.com
       2 |          2 | 2022-01-02 12:00:00 | Description paul        | Title 2 | http://example2.com
       3 |          3 | 2022-01-03 12:00:00 | Description chiri_clona | Title 3 | http://example3.com
       4 |          4 | 2022-01-04 12:00:00 | Description paul_clona  | Title 4 | http://example4.com
       5 |          5 | 2022-01-04 12:00:00 | Description tom         | Title 5 | http://example5.com
(5 rows)
```

```
postgres=# \c postgres chiri_clona
You are now connected to database "postgres" as user "chiri_clona".
postgres=> select creator_id,pubdate,description,title,url from posts;
 creator_id |       pubdate       |       description       |  title  |         url
------------+---------------------+-------------------------+---------+---------------------
          3 | 2022-01-03 12:00:00 | Description chiri_clona | Title 3 | http://example3.com
(1 row)

postgres=> \c postgres postgres
You are now connected to database "postgres" as user "postgres".
postgres=# select * from posts;
 post_id | creator_id |       pubdate       |       description       |  title  |         url
---------+------------+---------------------+-------------------------+---------+---------------------
       1 |          1 | 2022-01-01 12:00:00 | Description chiri       | Title 1 | http://example1.com
       2 |          2 | 2022-01-02 12:00:00 | Description paul        | Title 2 | http://example2.com
       3 |          3 | 2022-01-03 12:00:00 | Description chiri_clona | Title 3 | http://example3.com
       4 |          4 | 2022-01-04 12:00:00 | Description paul_clona  | Title 4 | http://example4.com
       5 |          5 | 2022-01-04 12:00:00 | Description tom         | Title 5 | http://example5.com
(5 rows)

postgres=# select * from users;
 id |  username   | password  | public_key | rank
----+-------------+-----------+------------+------
  1 | chiri       | password1 | publickey1 |    1
  2 | paul        | password2 | publickey2 |    2
  3 | chiri_clona | password3 | publickey3 |    3
  4 | paul_clona  | password4 | publickey4 |    4
  5 | tom         | password5 | publickey5 |    5
(5 rows)
```

```
postgres=> \c postgres paul_clona;
You are now connected to database "postgres" as user "paul_clona".
postgres=> select * from rooms;
ERROR:  permission denied for table rooms
postgres=> select u_id, id_room from rooms;
ERROR:  column "u_id" does not exist
LINE 1: select u_id, id_room from rooms;
               ^
postgres=> select name_of_room,topic_of_room,rules from rooms;
 name_of_room | topic_of_room | rules
--------------+---------------+--------
 Room 1       | Topic 1       | Rule 1
 Room 2       | Topic 2       | Rule 2
 Room 3       | Topic 3       | Rule 3
 Room 4       | Topic 4       | Rule 4
 Room 5       | Topic 5       | Rule 5
(5 rows)
```

## Encryption:

- CREATE EXTENSION pgcrypto;
- 
- UPDATE users SET password = md5(password) WHERE id = 1;
- 
- UPDATE users SET password = md5(password) WHERE id = 2;
- 
- UPDATE users SET password = crypt(password,gen_salt('bf')) WHERE id = 3;
- 
- UPDATE users SET password = md5(password) WHERE id = 4;
- 
- UPDATE users SET password = crypt(password,gen_salt('des')) WHERE id = 5;

```
You are now connected to database "postgres" as user "postgres".
postgres=#
postgres=# CREATE EXTENSION pgcrypto; CREATE EXTENSION pgcrypto; ^C
postgres=#
postgres=# CREATE EXTENSION pgcrypto;
CREATE EXTENSION
postgres=# UPDATE users SET password = md5(password) WHERE id = 1;
UPDATE 1
postgres=# UPDATE users SET password = md5(password) WHERE id = 2;
UPDATE 1
postgres=# UPDATE users SET password = crypt(password,gen_salt('bf')) WHERE id = 3;
UPDATE 1
postgres=# UPDATE users SET password = md5(password) WHERE id = 4;
UPDATE 1
postgres=# UPDATE users SET password = crypt(password,gen_salt('des')) WHERE id = 5;
UPDATE 1
postgres=# select * from users;
 id |  username   |                           password
    | public_key | rank
----+-------------+--------------------------------------------------------------+------------+------
  1 | chiri       | 7c6a180b36896a0a8c02787eeafb0e4c
    | publickey1 |    1
  2 | paul        | 6cb75f652a9b52798eb6cf2201057c73
    | publickey2 |    2
  3 | chiri_clona | $2a$06$blDqVWYODpYFGlvQUywxz.xyTGJLi5ZGQDb8ZQ8/fl9gb1i9cP.uS | publickey3 |    3
  4 | paul_clona  | 34cc93ece0ba9e3f6f235d4af979b16c
    | publickey4 |    4
  5 | tom         | kFWfYZ9gO60Jw
    | publickey5 |    5
(5 rows)
```

## Backup & Recovery:

- pg_dump > mydb.sql
- 
- createdb -T template0 postgres_restaured
- 
- psql postgres_restaured < mydb.sql

```
postgres=# \l
                                                List of databases
    Name    |  Owner   | Encoding |  Collate   |   Ctype    | ICU Locale | Locale Provider |   Access privileges
------------+----------+----------+------------+------------+------------+-----------------+-----------------------
 mydatabase | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            |
 postgres   | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            |
 template0  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            |
     | =c/postgres          +
     | postgres=CTc/postgres
 template1  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            |
     | =c/postgres          +
     | postgres=CTc/postgres
(4 rows)
```

```
postgres@1c3ca7d880aa:/my_scripts_folder/s$ pg_dump > mydb.sql
postgres@1c3ca7d880aa:/my_scripts_folder/s$ cat mydb.sql
--
-- PostgreSQL database dump
--

-- Dumped from database version 15.1 (Debian 15.1-1.pgdg110+1)
-- Dumped by pg_dump version 15.1 (Debian 15.1-1.pgdg110+1)

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;
```

```
postgres@1c3ca7d880aa:/my_scripts_folder/s$ createdb -T template0 postgres_restaured
postgres@1c3ca7d880aa:/my_scripts_folder/s$ psql postgres_restaured < mydb.sql
SET
SET
SET
SET
SET
 set_config
------------

(1 row)

SET
SET
SET
SET
CREATE EXTENSION
COMMENT
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
```

```
CREATE POLICY
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
postgres@1c3ca7d880aa:/my_scripts_folder/s$
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
postgres@1c3ca7d880aa:/my_scripts_folder/s$ _
```

```
postgres@1c3ca7d880aa:/my_scripts_folder/s$ psql -U postgres -d postgres_restaured
psql (15.1 (Debian 15.1-1.pgdg110+1))
Type "help" for help.

postgres_restaured=# \dt
         List of relations
 Schema | Name  | Type  |  Owner
--------+-------+-------+----------
 public | posts | table | postgres
 public | rank  | table | postgres
 public | rooms | table | postgres
 public | users | table | postgres
(4 rows)
```