# PostgreSQL
# Database Administrator

- ❖ Create database with sample tables and data
- ❖ Create users
- ❖ Set and verify password expiry dates
- ❖ Monitor currently logged-in users
- ❖ Grant full access on database
- ❖ Connect via pgAdmin
- ❖ Take dump backups of database
- ❖ Delete database
- ❖ Restore them from backups

## Create database with sample tables and data

1. To create a new database named db1, execute the following SQL command:

   ❖ CREATE DATABASE db1;

2. Once the database is created, connect to it using:

   ❖ \c db1

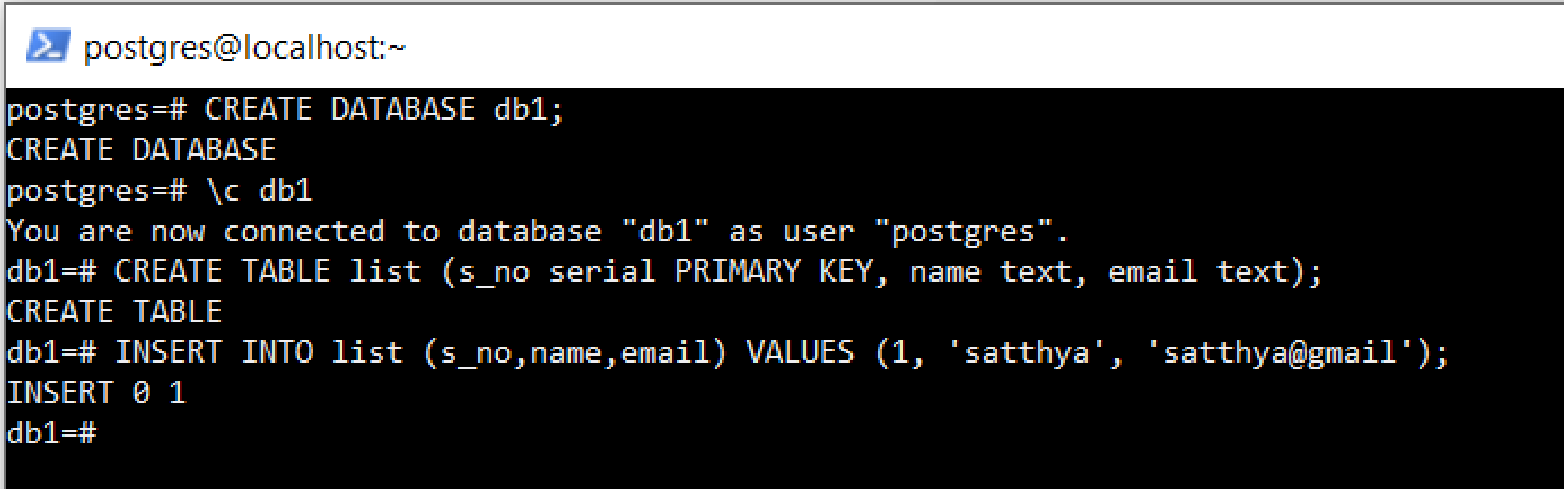
3. Then create a table named list with the following command:

   ❖ CREATE TABLE list (s_no serial PRIMARY KEY, name text, email text);

4. To insert a record into the list table, use the following SQL command:

   ❖ INSERT INTO list (s_no, name, email) VALUES (1,’satthya’, ’satthya@gmail.com’)

## Create database with sample tables and data

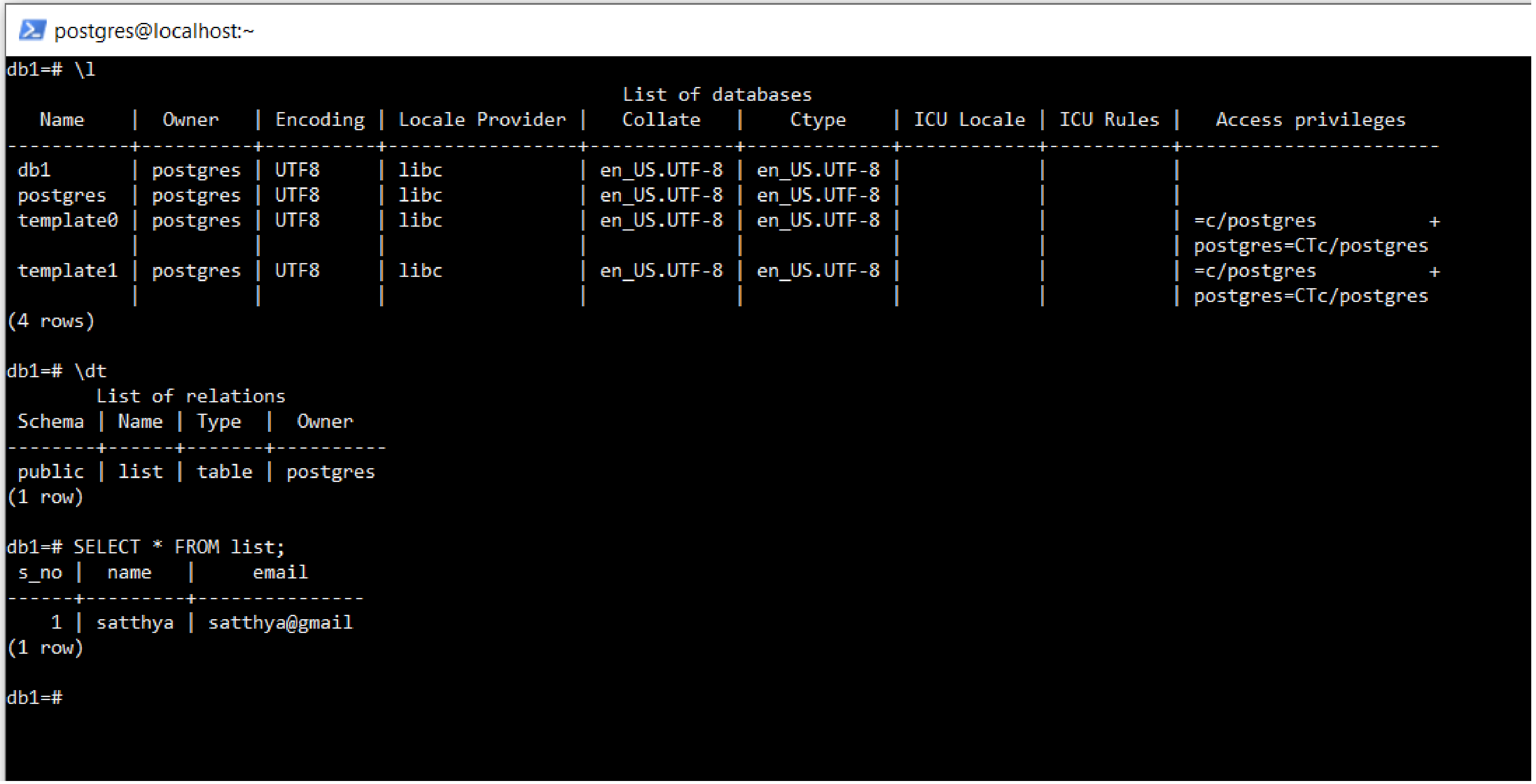


```
postgres=# CREATE DATABASE db1;
CREATE DATABASE
postgres=# \c db1
You are now connected to database "db1" as user "postgres".
db1=# CREATE TABLE list (s_no serial PRIMARY KEY, name text, email text);
CREATE TABLE
db1=# INSERT INTO list (s_no,name,email) VALUES (1, 'satthya', 'satthya@gmail');
INSERT 0 1
db1=#
```

## Verify the database with sample tables and data

1. To view the created database:
    - ❖ \l
2. To view the created table
    - ❖ \dt
3. To view the data in the table
    - ❖ SELECT * FROM list;

## Verify the database with sample tables and data



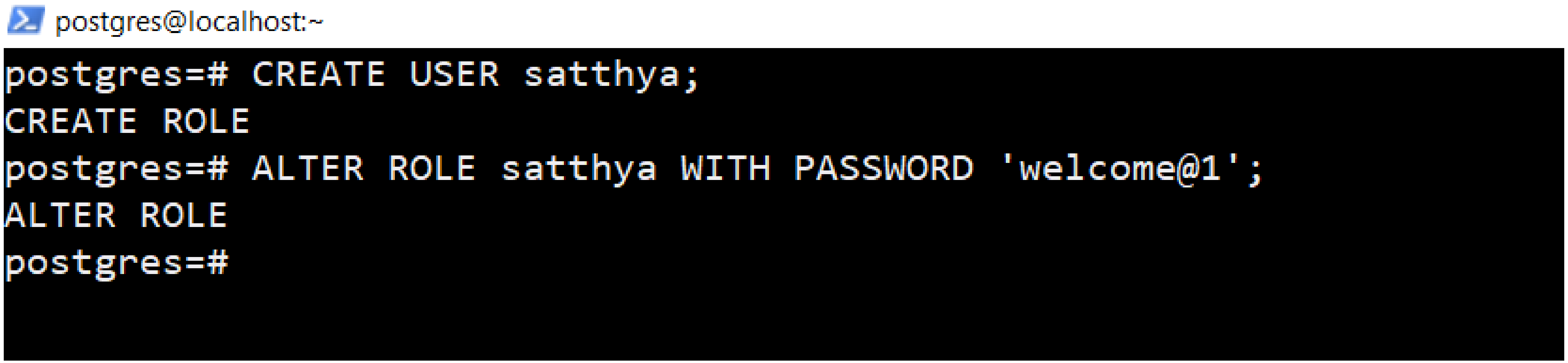
postgres@localhost:~

```
db1=# \l
                                                      List of databases
   Name    |  Owner   | Encoding | Locale Provider |   Collate   |    Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+-------------+-------------+------------+-----------+-----------------------
 db1       | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 postgres  | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.UTF-8 | en_US.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |             |             |            |           | postgres=CTc/postgres
(4 rows)

db1=# \dt
        List of relations
 Schema | Name | Type  |  Owner
--------+------+-------+----------
 public | list | table | postgres
(1 row)

db1=# SELECT * FROM list;
 s_no |  name   |     email
------+---------+---------------
    1 | satthya | satthya@gmail
(1 row)

db1=#
```

## Create users

1. To create users, please execute below SQL command:
    - ❖ CREATE USER satthya;
2. To create user with a password:
    - ❖ CREATE USER satthya WITH PASSWORD ‘Welcome@1’;
3. To change or assigned password:
    - ❖ ALTER ROLE satthya WITH PASSWORD ‘Welcome@1’;

## Create users

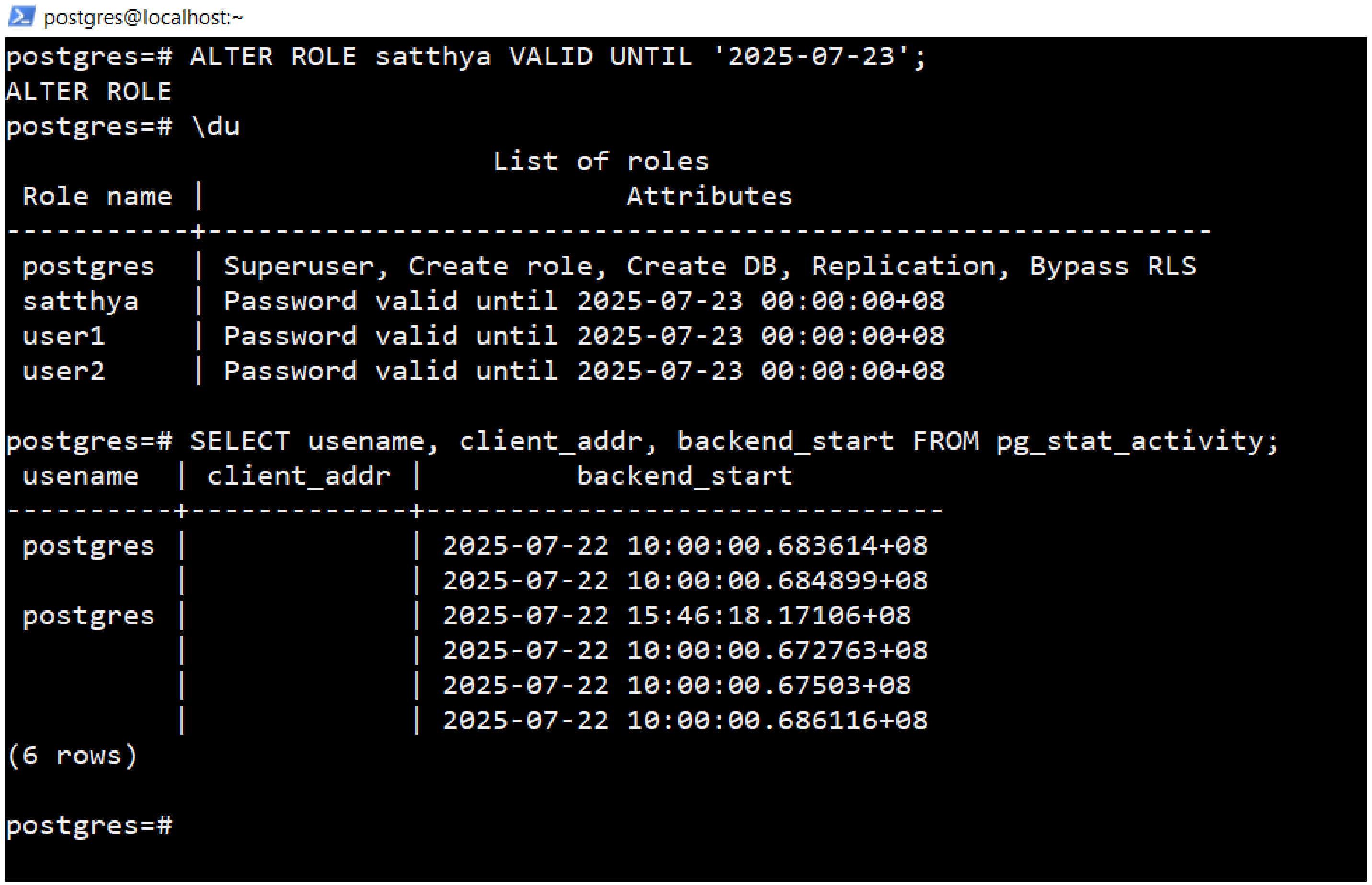


```
postgres=# CREATE USER satthya;
CREATE ROLE
postgres=# ALTER ROLE satthya WITH PASSWORD 'welcome@1';
ALTER ROLE
postgres=#
```

## Assigned Password Age and Validate

1. Assigned Password expiry date:
   - ❖ ALTER ROLE satthya VALID UNTIL ‘2025-07-23’;
2. Verify User:
   - ❖ \du
3. Monitor currently logged in users:
   - ❖ SELECT usename, client_addr, backend_start FROM pg_stat_activity;

## Assigned Password Age and Validate


postgres@localhost:~

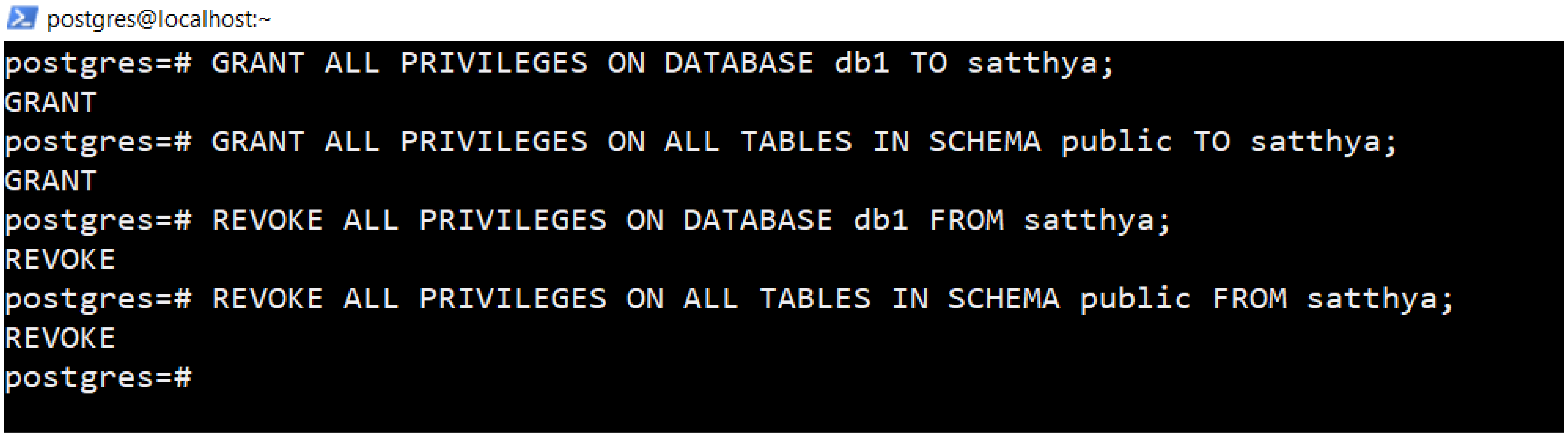
```
postgres=# ALTER ROLE satthya VALID UNTIL '2025-07-23';
ALTER ROLE
postgres=# \du
                               List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS
 satthya   | Password valid until 2025-07-23 00:00:00+08
 user1     | Password valid until 2025-07-23 00:00:00+08
 user2     | Password valid until 2025-07-23 00:00:00+08

postgres=# SELECT usename, client_addr, backend_start FROM pg_stat_activity;
 usename  | client_addr |         backend_start
----------+-------------+-------------------------------
 postgres |             | 2025-07-22 10:00:00.683614+08
          |             | 2025-07-22 10:00:00.684899+08
 postgres |             | 2025-07-22 15:46:18.17106+08
          |             | 2025-07-22 10:00:00.672763+08
          |             | 2025-07-22 10:00:00.67503+08
          |             | 2025-07-22 10:00:00.686116+08
(6 rows)

postgres=#
```

## Grant/Revoke full access on database

To grant full access on database, access should be given on database and table level.

1. Grant access on database:
   - ❖ GRANT ALL PRIVILEGES ON DATABASE db1 TO satthya;
2. Grant access on table:
   - ❖ GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO satthya;
3. To remove the access on database:
   - ❖ REVOKE ALL PRIVILEGES ON DATABASE db1 FROM satthya;
4. To remove the access on table
   - ❖ RERVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM satthya;

## Grant/Revoke full access on database



```
postgres=# GRANT ALL PRIVILEGES ON DATABASE db1 TO satthya;
GRANT
postgres=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO satthya;
GRANT
postgres=# REVOKE ALL PRIVILEGES ON DATABASE db1 FROM satthya;
REVOKE
postgres=# REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM satthya;
REVOKE
postgres=#
```

## Connect via pgAdmin

To connect pgAdmin to a PostgreSQL server remotely, we first need to allow remote connections on the server.

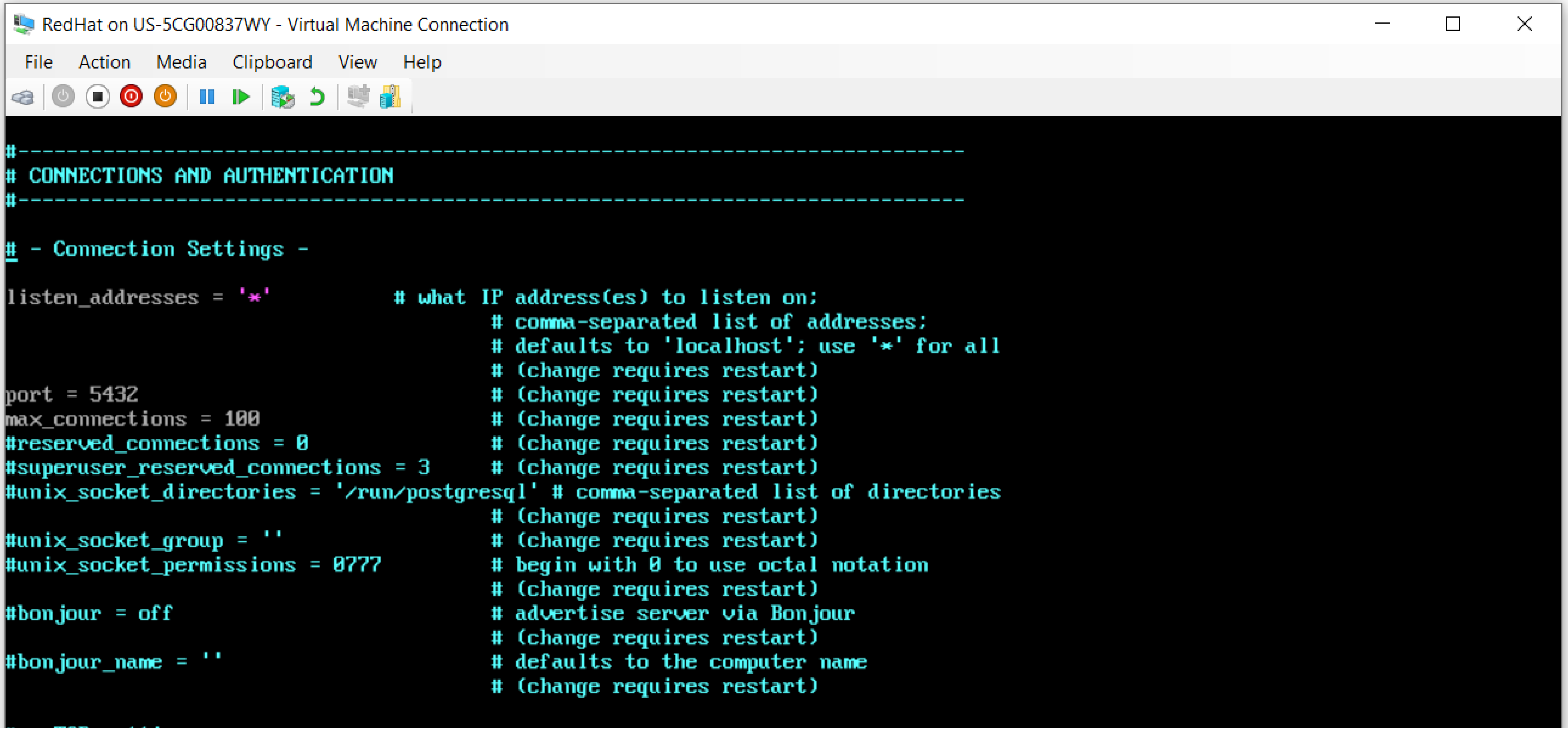
1. Edit postgresql.conf

    ❖ cd /var/lib/pgsql/16/data
    ❖ vi postgresql.conf

    Update or uncomment the following lines:

    ❖ Listen_addresses = ‘*’
    ❖ Port = 5432

# Connect via pgAdmin


RedHat on US-5CG00837WY - Virtual Machine Connection
File Action Media Clipboard View Help

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#reserved_connections = 0               # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/run/postgresql' # comma-separated list of directories
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                                        # (change requires restart)
```

## Connect via pgAdmin

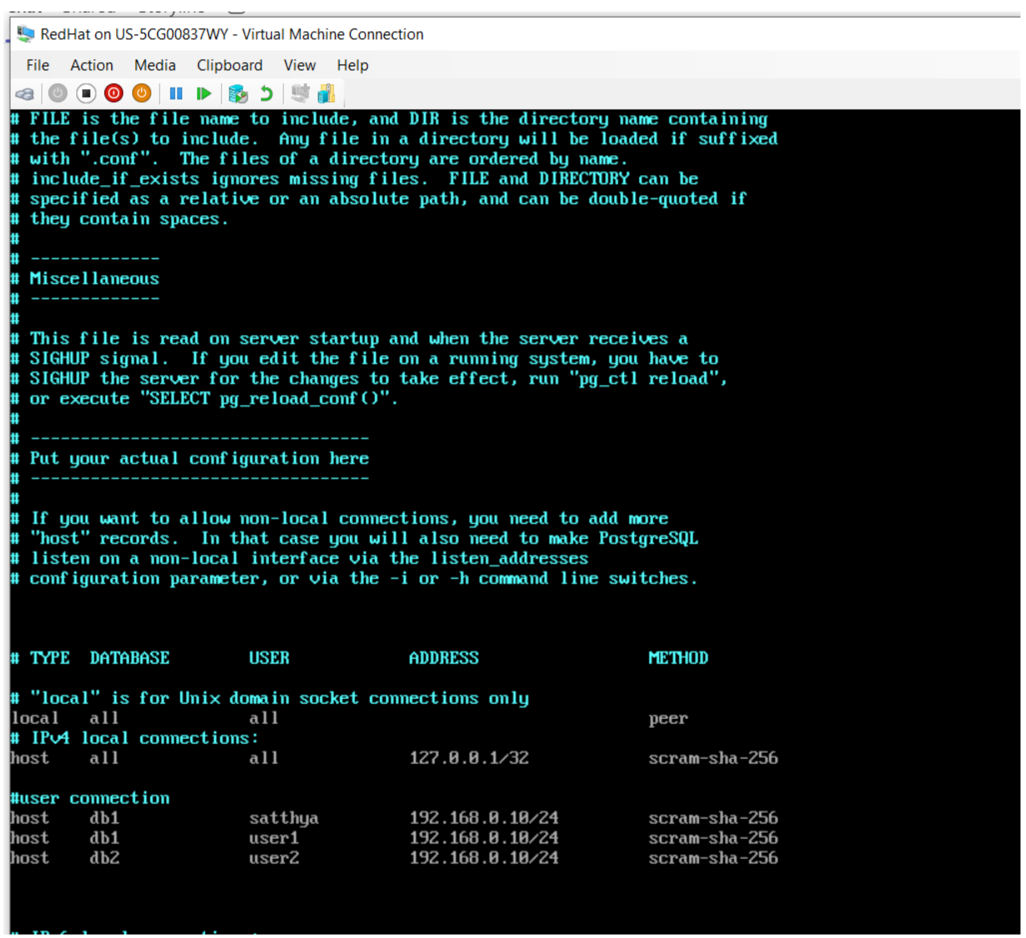
2. Edit pg_hba.conf to allow client Ips:

   - ❖ vi pg_hba.conf

| TYPE | DATABASE | USER | ADDRESS | METHOD |
|---|---|---|---|---|
| host | db1 | satthya | 192.168.0.1/24 | scram-sha-256 |

Remote machine IP Address

# Connect via pgAdmin


RedHat on US-5CG00837WY - Virtual Machine Connection
File Action Media Clipboard View Help

```
# FILE is the file name to include, and DIR is the directory name containing
# the file(s) to include.  Any file in a directory will be loaded if suffixed
# with ".conf".  The files of a directory are ordered by name.
# include_if_exists ignores missing files.  FILE and DIRECTORY can be
# specified as a relative or an absolute path, and can be double-quoted if
# they contain spaces.
#
# -------------
# Miscellaneous
# -------------
#
# This file is read on server startup and when the server receives a
# SIGHUP signal.  If you edit the file on a running system, you have to
# SIGHUP the server for the changes to take effect, run "pg_ctl reload",
# or execute "SELECT pg_reload_conf()".
#
# ----------------------------------
# Put your actual configuration here
# ----------------------------------
#
# If you want to allow non-local connections, you need to add more
# "host" records.  In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.



# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            scram-sha-256

#user connection
host    db1             satthya         192.168.0.10/24         scram-sha-256
host    db1             user1           192.168.0.10/24         scram-sha-256
host    db2             user2           192.168.0.10/24         scram-sha-256
```

# Connect via pgAdmin

3. Allowed firewall rules:

   - firewall-cmd --add-service=postgresql –permanent
   - firewall-cmd --add-port=5432/tcp –permanent
   - firewall-cmd --reload

4. Restart the service

   - systemctl restart postgresql-16.service

## Connect via pgAdmin

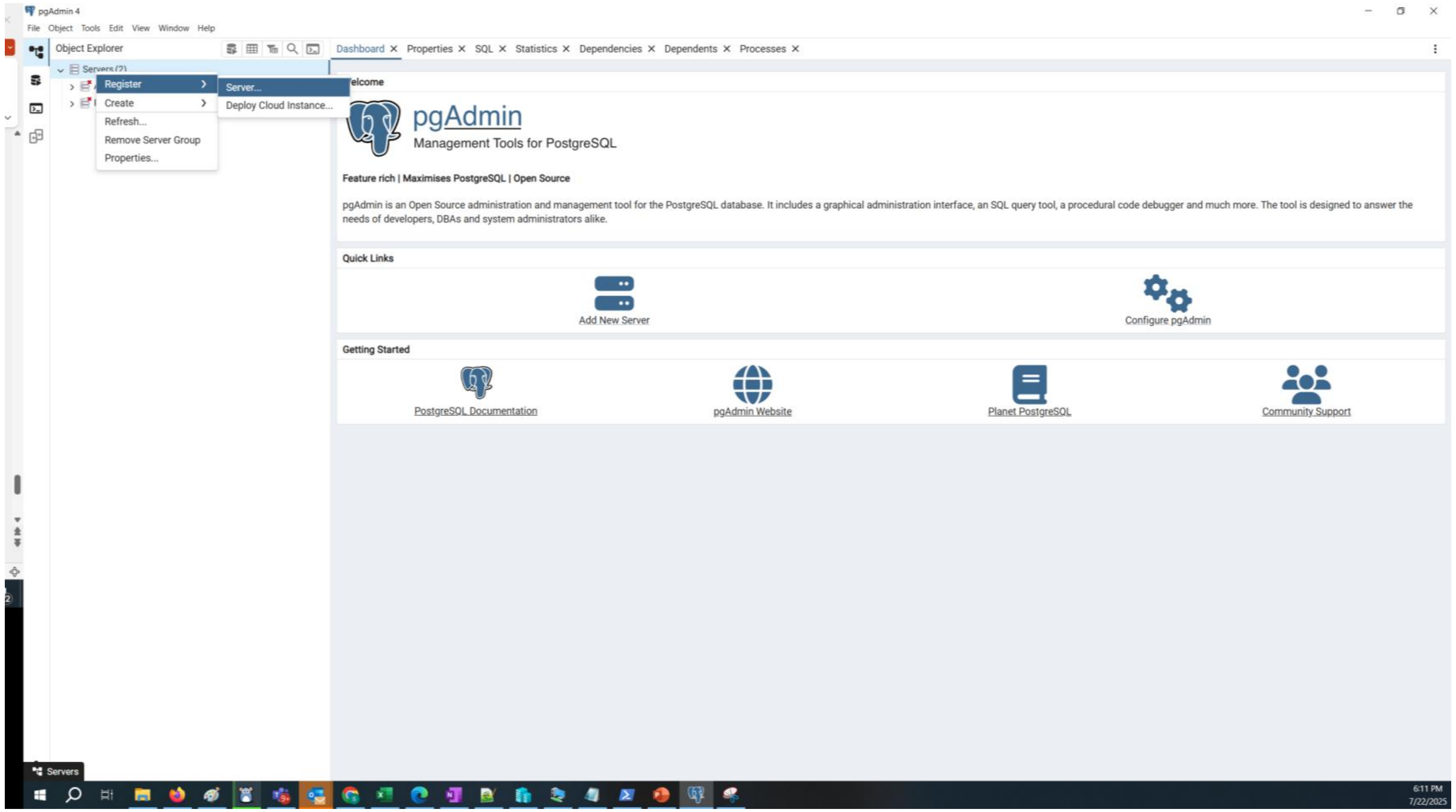
5. Verify the connection from pgAdmin

    - ❖ Click on Register Server
        - ❖ On the general tab enter the database name

        - ❖ Key in below details on connection tab

            - ❖ Database server ip address
            - ❖ Port
            - ❖ Maintenance database
            - ❖ Username
            - ❖ password

# Connect via pgAdmin

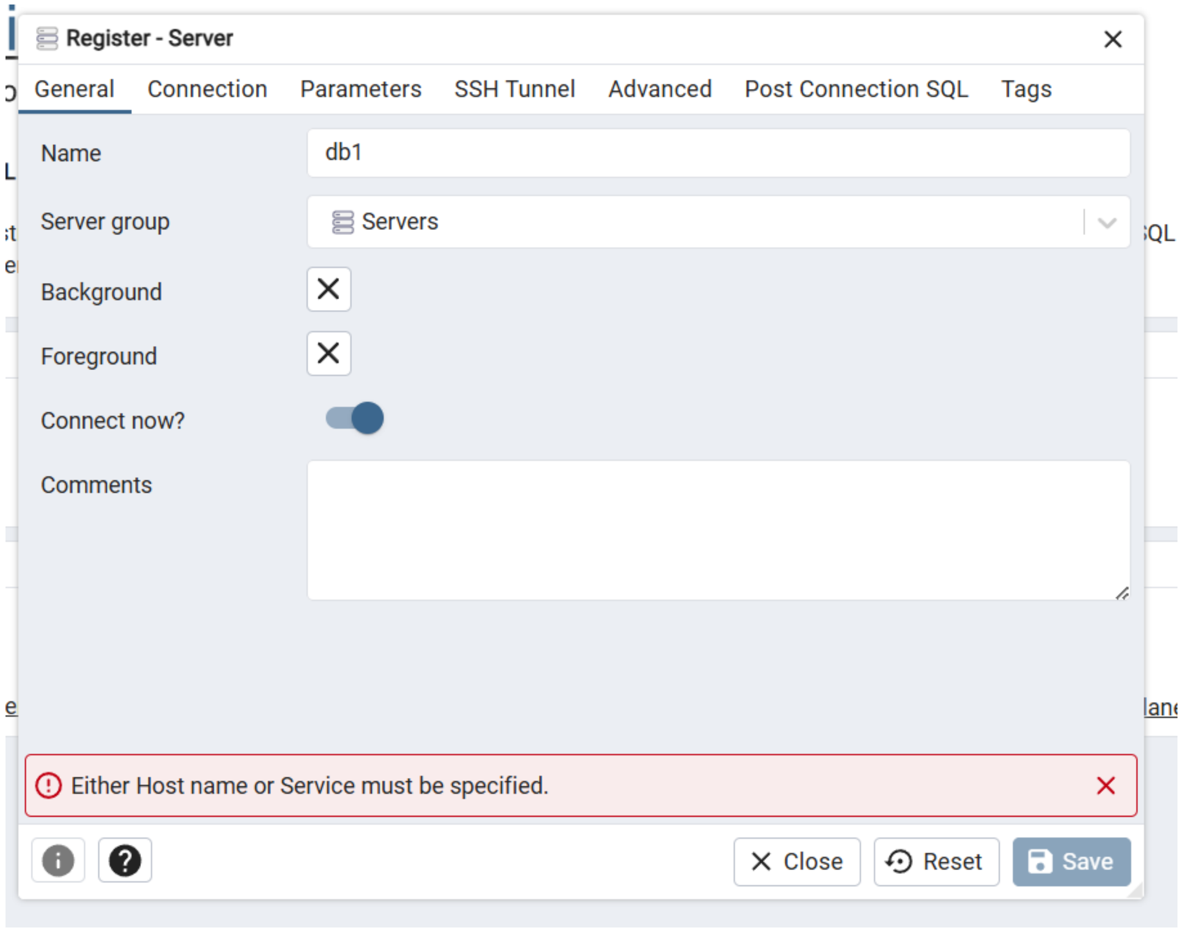


Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.

# Connect via pgAdmin



Register - Server

Name: db1

Server group: Servers

Background: ✗

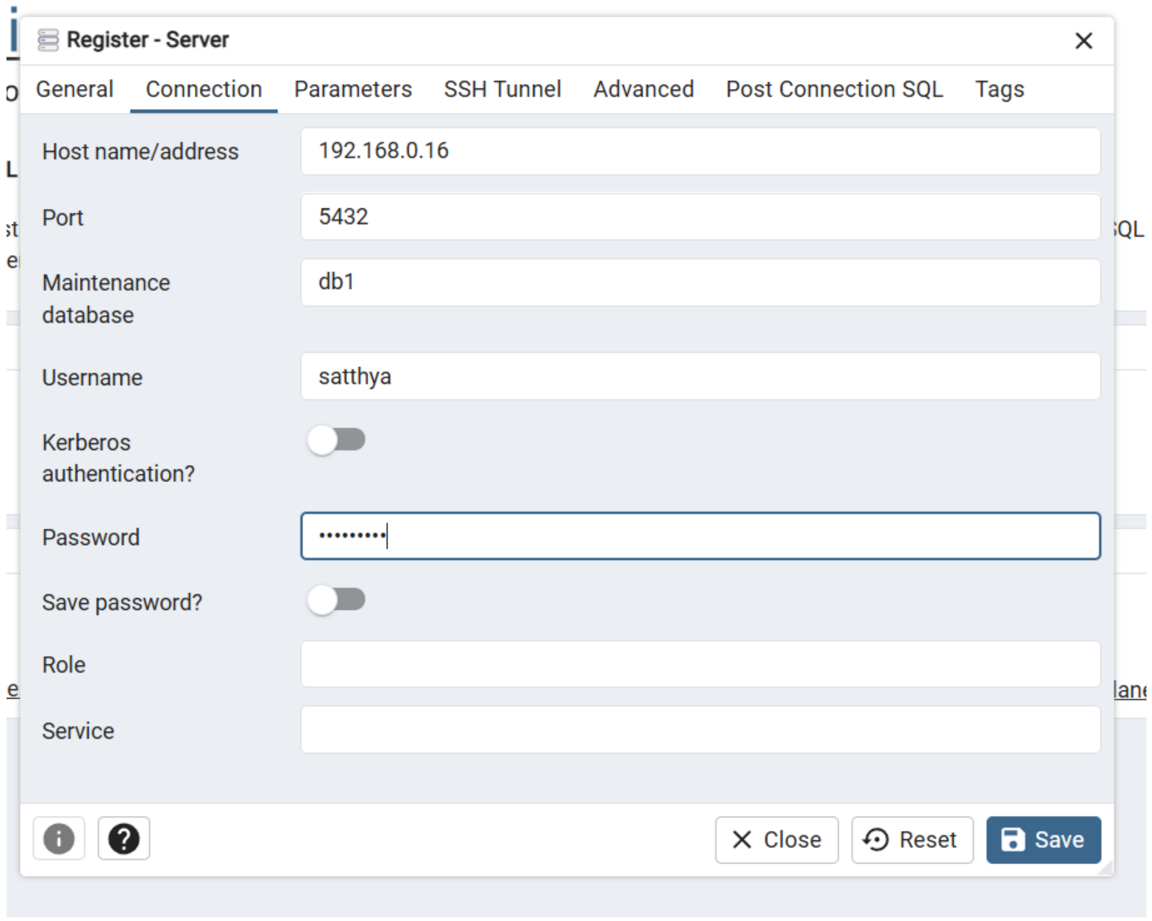
Foreground: ✗

Connect now? ●

Comments:

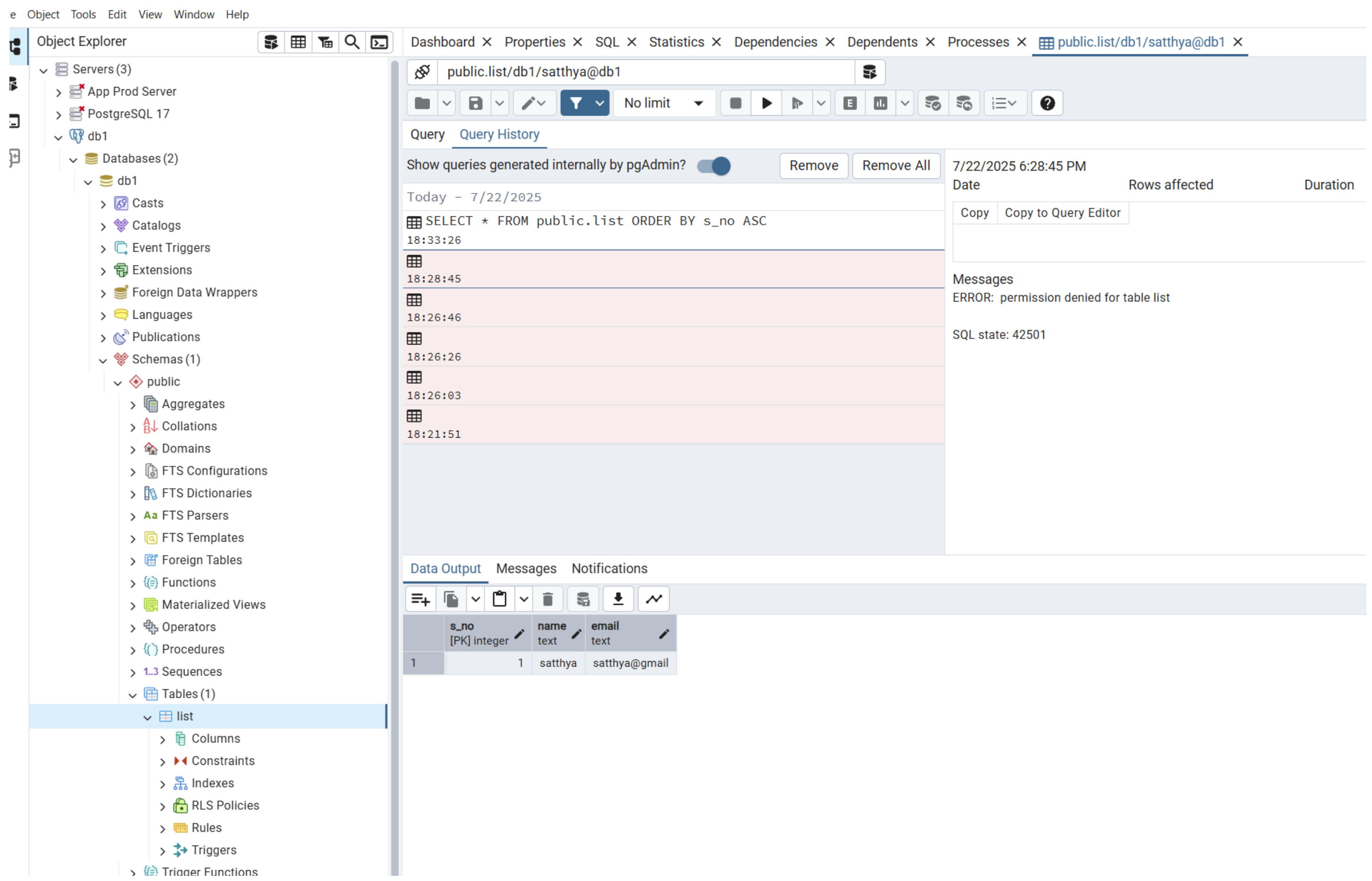
Either Host name or Service must be specified.

# Connect via pgAdmin



| | |
|---|---|
| Host name/address | 192.168.0.16 |
| Port | 5432 |
| Maintenance database | db1 |
| Username | satthya |
| Kerberos authentication? | off |
| Password | ••••••••• |
| Save password? | off |
| Role | |
| Service | |

# Connect via pgAdmin



Show queries generated internally by pgAdmin?  Remove  Remove All

```
Today - 7/22/2025
SELECT * FROM public.list ORDER BY s_no ASC
18:33:26
18:28:45
18:26:46
18:26:26
18:26:03
18:21:51
```

7/22/2025 6:28:45 PM

Date  Rows affected  Duration

Copy  Copy to Query Editor

Messages

```
ERROR:  permission denied for table list

SQL state: 42501
```

| | s_no [PK] integer | name text | email text |
|---|---|---|---|
| 1 | 1 | satthya | satthya@gmail |

## Take dump backups of database

1. Create directory
    - ❖ mkdir –p /backup/dump_back
2. Change ownership to the postgres user
    - ❖ chown –R postgres:postgres /backup/dump_back
3. Switch to postgres userg
    - ❖ pg_dump –d db1 –f /backup/dump_back/db1_bck.sql

## Delete & Restore Database

1. Connect to the PostgreSQL server
    - ❖ psql
2. Delete the database db1:
    - ❖ DROP DATABASE db1
3. Restore the database db1:
    - ❖ Recreate the database with same name
        - ❖ CREATE DATABASE db1;
4. Switch to postgres user
    - ❖ psql -d db1 -f /backup/dump_back/db1_bck.sql

END