Introduction to SQL

Part-2

Hitesh Kumar Sharma
Instructor, Pluralsight

Loading a Dataset in PostgreSQL

PLURALSIGHT

Loading a Database in PostgreSQL

- we will be using a sample database which is **DVD rental database.**
- You can download the sample ***dvdrental*** database from following link

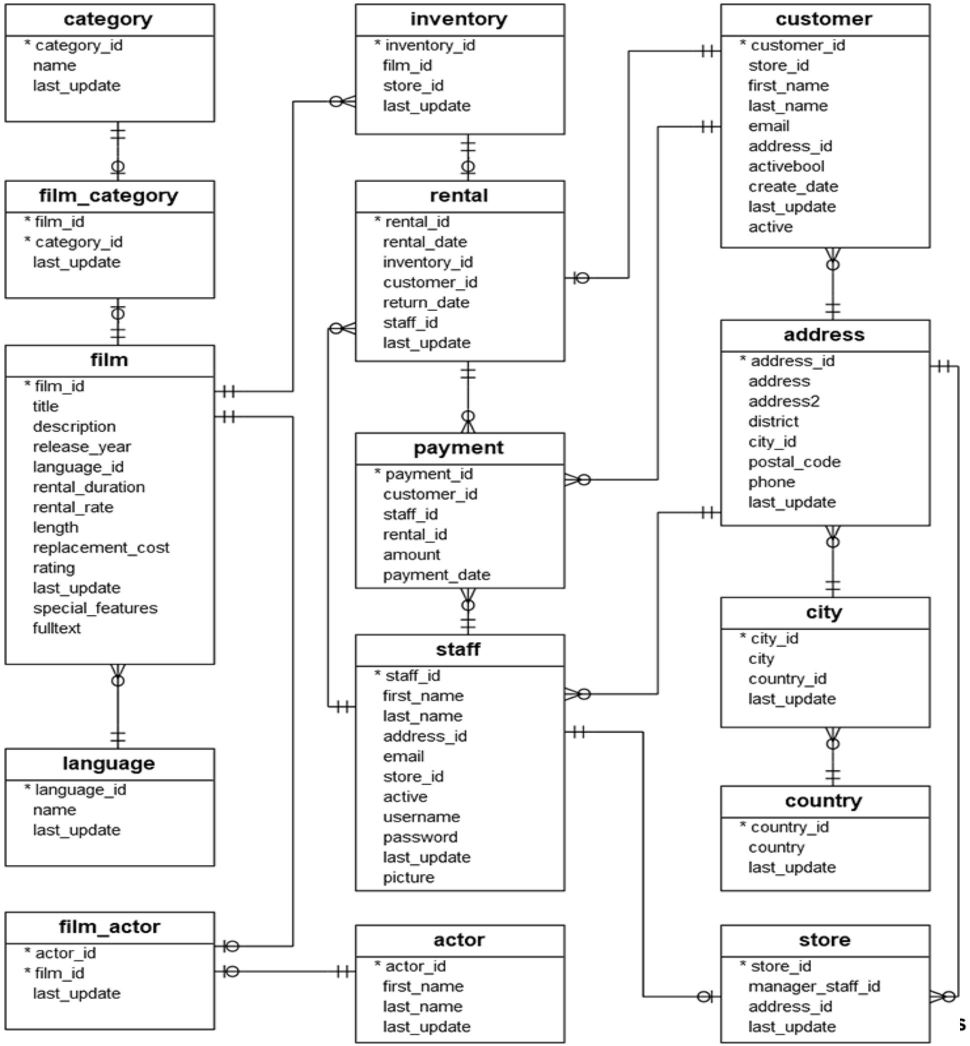
https://github.com/imkumaraju/dvdrenat-sample-databse

The Sample Database:

So, the DVD rental database that we will be using ahead in the article represents a DVD rental store. The objects in the database includes:

- 15 tables
- 1 trigger
- 8 functions
- 1 domain
- 7 views
- 13 sequences

ER Diagram of Sample Database (dvdrental)



category
* category_id
name
last_update

film_category
* film_id
* category_id
last_update

film
* film_id
title
description
release_year
language_id
rental_duration
rental_rate
length
replacement_cost
rating
last_update
special_features
fulltext

language
* language_id
name
last_update

film_actor
* actor_id
* film_id
last_update

inventory
* inventory_id
film_id
store_id
last_update

rental
* rental_id
rental_date
inventory_id
customer_id
return_date
staff_id
last_update

payment
* payment_id
customer_id
staff_id
rental_id
amount
payment_date

staff
* staff_id
first_name
last_name
address_id
email
store_id
active
username
password
last_update
picture

actor
* actor_id
first_name
last_name
last_update

customer
* customer_id
store_id
first_name
last_name
email
address_id
activebool
create_date
last_update
active

address
* address_id
address
address2
district
city_id
postal_code
phone
last_update

city
* city_id
city
country_id
last_update

country
* country_id
country
last_update

store
* store_id
manager_staff_id
address_id
last_update

Tables in Sample Database

There are 15 tables in our sample database which are listed below:

1.actor – stores actors data including first name and last name.

2.film – stores films data such as title, release year, length, rating, etc

3.film_actor – stores the relationships between films and actors.

4.category – stores film's categories data.

5.film_category– stores the relationships between films and categories.

6.store – contains the store data including manager staff and address.

7.inventory – stores inventory data.

8.rental – stores rental data.

9.payment – stores customer's payments.

10.staff – stores staff data.

11.customer – stores customers data.

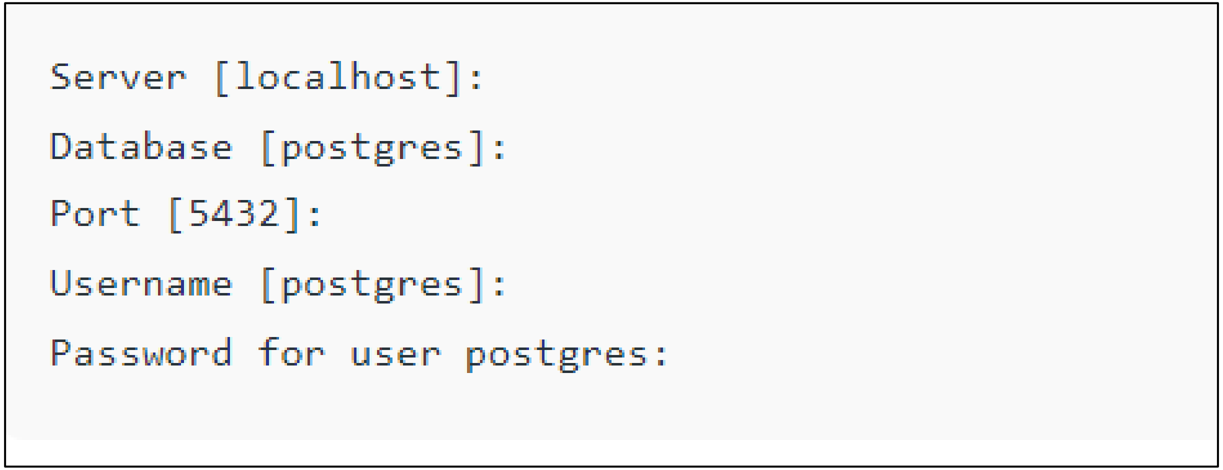
12.address – stores address data for staff and customers

13.city – stores the city names.

14.country – stores the country names

Steps to load Sample Database

Step 1: Open the SQL shell and create a database for renting DVDs. You must enter your database's credentials once you've opened the shell. They should resemble the following in some way:

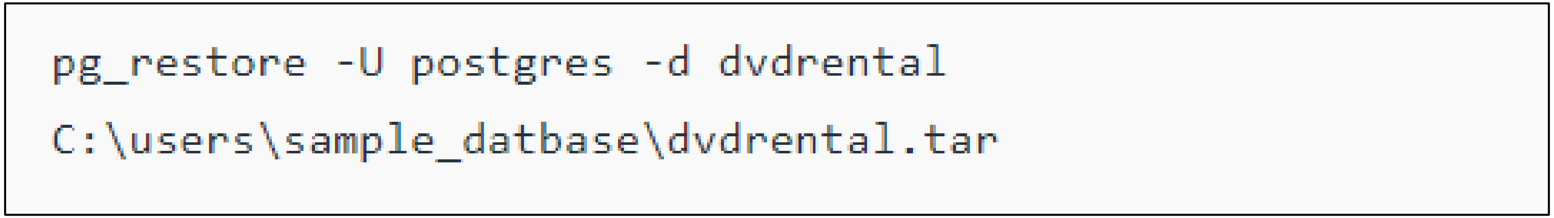


```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
```

Steps to load Sample Database

Step 2: Create a folder at the location of your choice (for example, c:\users\sample_database\dvdrental.tar) and load the database file into it. Launch the command prompt now, and go as follows to the PostgreSQL installation folder's bin folder:

Use the pg_restore tool to load data into the dvdrental database that we had just created as using the command:

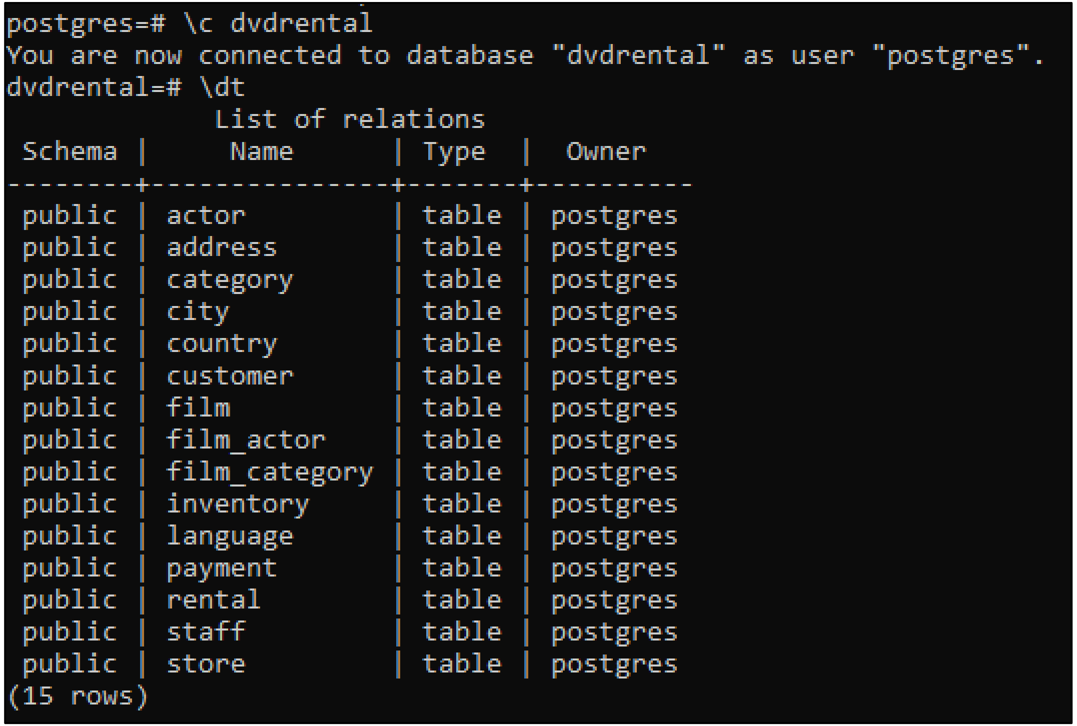


```
pg_restore -U postgres -d dvdrental
C:\users\sample_datbase\dvdrental.tar
```

Now enter your database user **Password** and your sample database will be loaded.

Verify Loading of Sample Database

Now if you need to verify if the sample database is loaded, use the below command to get into the database in SQL shell:



```
postgres=# \c dvdrental
You are now connected to database "dvdrental" as user "postgres".
dvdrental=# \dt
             List of relations
 Schema |     Name      | Type  |  Owner
--------+---------------+-------+----------
 public | actor         | table | postgres
 public | address       | table | postgres
 public | category      | table | postgres
 public | city          | table | postgres
 public | country       | table | postgres
 public | customer      | table | postgres
 public | film          | table | postgres
 public | film_actor    | table | postgres
 public | film_category | table | postgres
 public | inventory     | table | postgres
 public | language      | table | postgres
 public | payment       | table | postgres
 public | rental        | table | postgres
 public | staff         | table | postgres
 public | store         | table | postgres
(15 rows)
```

Database Commands in PostgreSQL

PLURALSIGHT

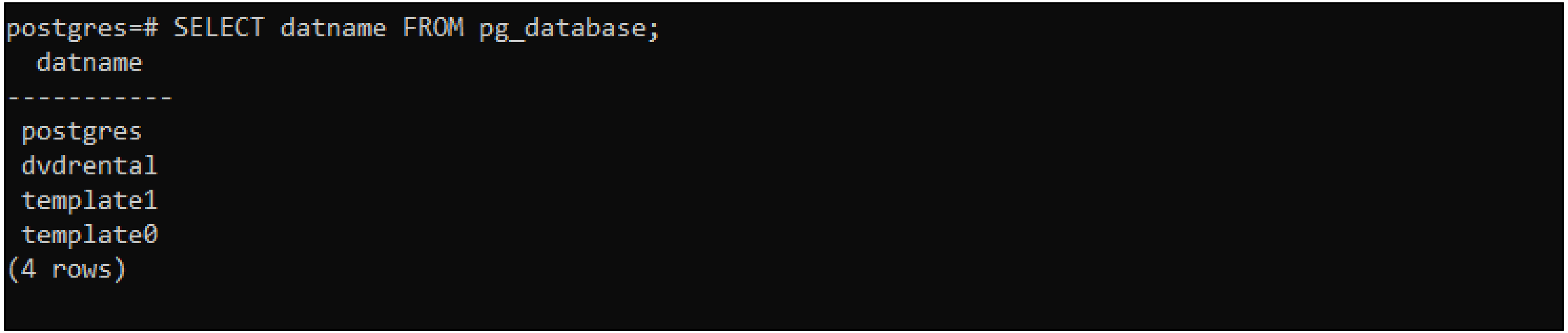
Show Database

Using SELECT statement:

The SELECT statement can also be used to list all the database present on the server:

Syntax: ***SELECT datname FROM pg_database;***

Example: ***SELECT datname FROM pg_database;***



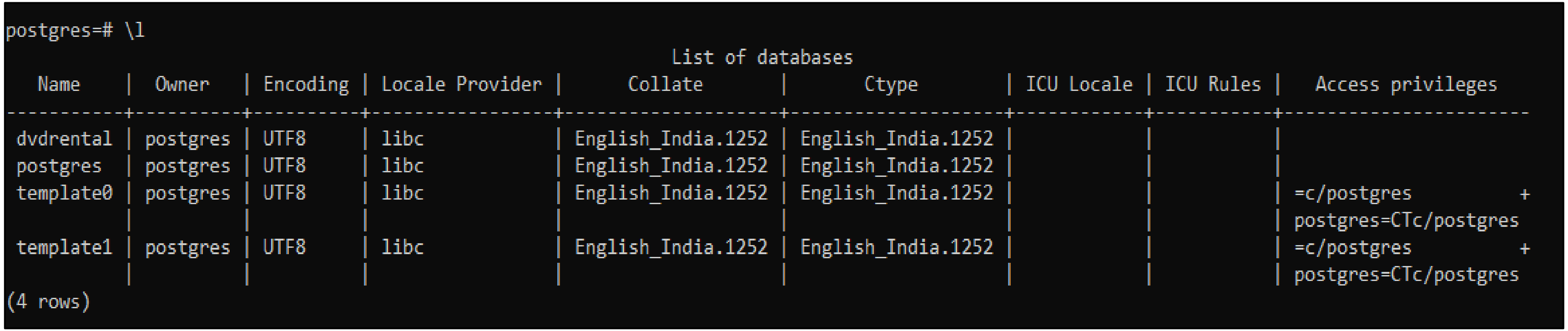
```
postgres=# SELECT datname FROM pg_database;
  datname
-----------
 postgres
 dvdrental
 template1
 template0
(4 rows)
```

Show Database

use the below command to list all databases using a superuser such as postgres:

\l

This will lead to the following:



```
postgres=# \l
                                                              List of databases
   Name    |  Owner   | Encoding | Locale Provider |      Collate       |       Ctype        | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+--------------------+--------------------+------------+-----------+-----------------------
 dvdrental | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 postgres  | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 template0 | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
           |          |          |                 |                    |                    |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
           |          |          |                 |                    |                    |            |           | postgres=CTc/postgres
(4 rows)
```

Create User in PostgreSQL

Creating a New User:

- Create a new user using the CREATE USER command. For example:

CREATE USER new_user WITH PASSWORD 'password';

- Replace new_user with the desired username and 'password' with the actual password for the user.

Grant Permission to User in PostgreSQL

Granting Permissions:

- Grant necessary permissions to the user as required. For example, to grant the CREATE privilege on a specific schema, use the following command:

```
ALTER ROLE your_username WITH SUPERUSER;
```

- You can grant other permissions depending on the requirements of your exercise.
- To switch to another user within the PostgreSQL console, you can use the following command:

```
\c - username
```

Hands-On Lab Exercise-1

(Topic: Create User)

Create Database

To create a database through the psql shell we make the use of the CREATE DATABASE statement as below:

CREATE DATABASE db_name

OWNER = role_name

ENCODING = encoding

CONNECTION LIMIT = max_concurrent_connection

Create Database

The various options provided by the CREATE DATABASE statement are explained below:

•***db_name:*** *It is the name of the new database that you want to create.It must always be a unique name.*

•***role_name:*** *It is the role name of the user who will own the new database.*

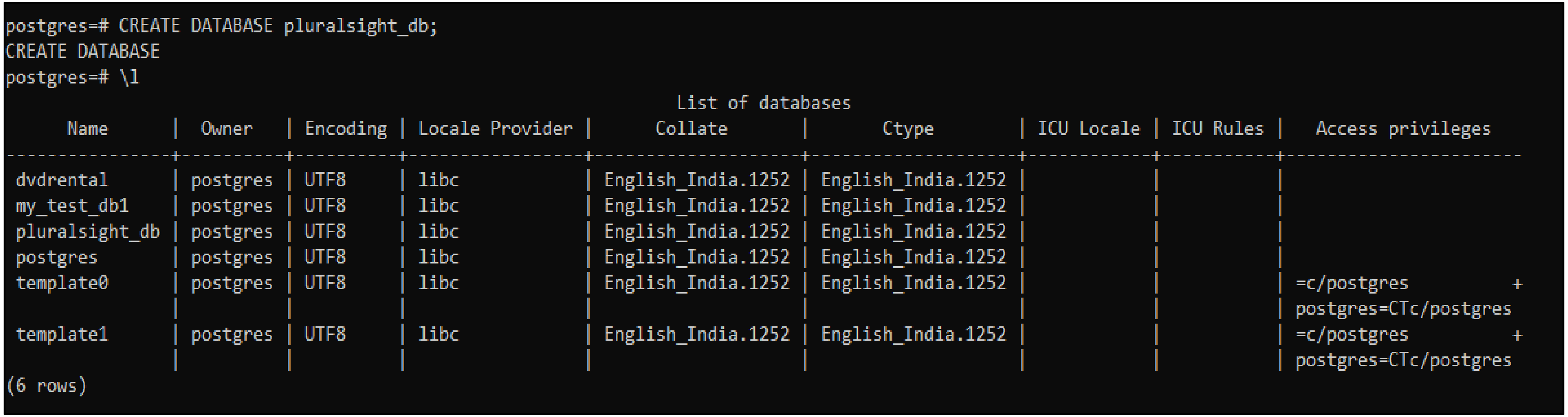
•***encoding:*** *It specifies the character set encoding for the new database. By default, it is the encoding of the template database.*

•***max_concurrent_connection:*** *It specifies the maximum concurrent connections to the new database.*

Create Database (Example)

Here we will create a test database with all default settings.

CREATE DATABASE pluralsight_db;



```
postgres=# CREATE DATABASE pluralsight_db;
CREATE DATABASE
postgres=# \l
                                                          List of databases
      Name      |  Owner   | Encoding | Locale Provider |      Collate       |       Ctype        | ICU Locale | ICU Rules |   Access privileges
----------------+----------+----------+-----------------+--------------------+--------------------+------------+-----------+-----------------------
 dvdrental      | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 my_test_db1    | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 postgres       | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 template0      | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                |          |          |                 |                    |                    |            |           | postgres=CTc/postgres
 template1      | postgres | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                |          |          |                 |                    |                    |            |           | postgres=CTc/postgres
(6 rows)
```

Create Database (Example 2)

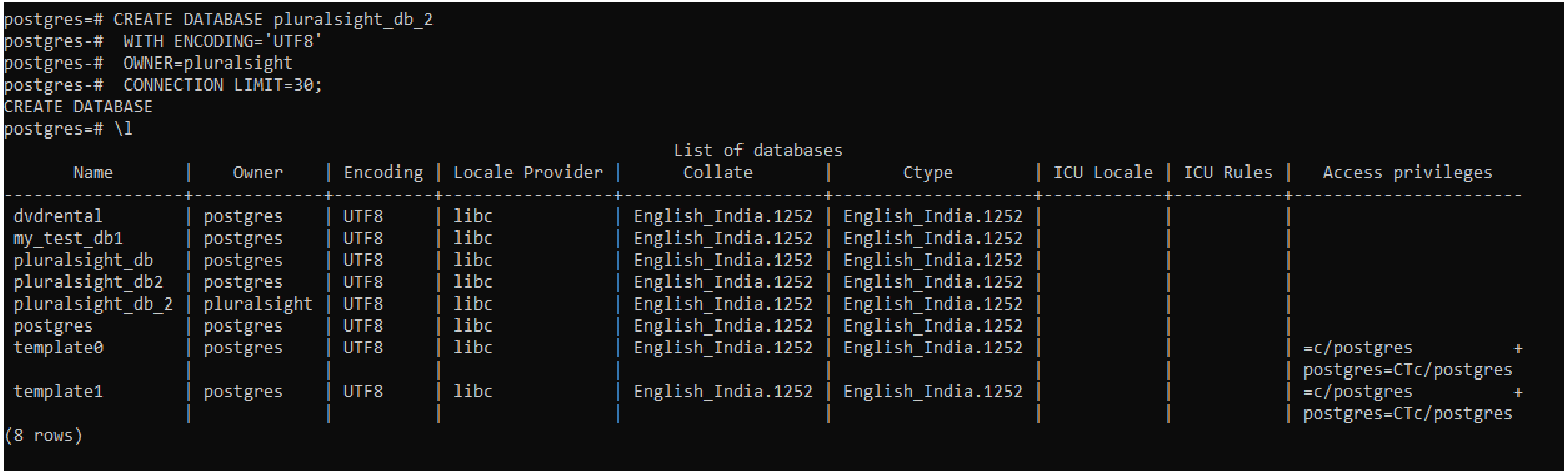
Here we will create a test database with all default settings.

CREATE DATABASE pluralsight_db_2

WITH ENCODING='UTF8'

OWNER=pluralsight

CONNECTION LIMIT=30;



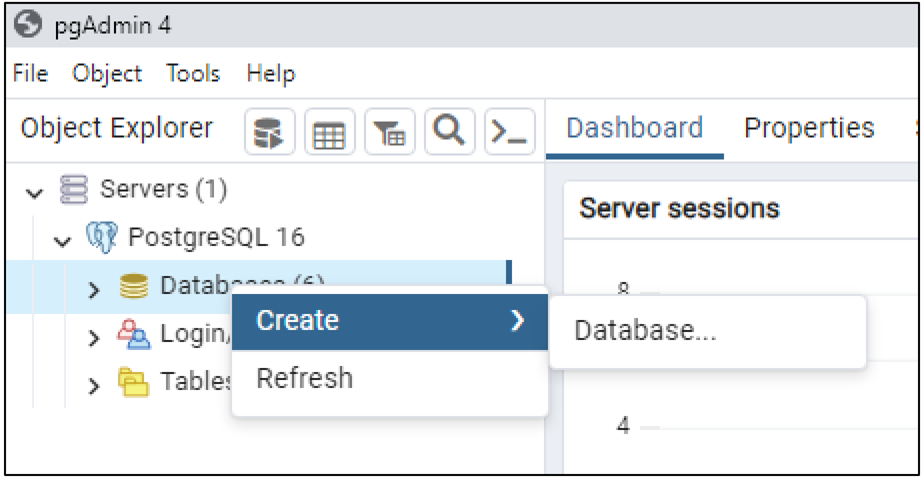
```
postgres=# CREATE DATABASE pluralsight_db_2
postgres-#  WITH ENCODING='UTF8'
postgres-#  OWNER=pluralsight
postgres-#  CONNECTION LIMIT=30;
CREATE DATABASE
postgres=# \l
                                                              List of databases
       Name       |    Owner    | Encoding | Locale Provider |      Collate       |       Ctype        | ICU Locale | ICU Rules |   Access privileges
------------------+-------------+----------+-----------------+--------------------+--------------------+------------+-----------+-----------------------
 dvdrental        | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 my_test_db1      | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db   | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db2  | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db_2 | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 postgres         | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 template0        | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                  |             |          |                 |                    |                    |            |           | postgres=CTc/postgres
 template1        | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                  |             |          |                 |                    |                    |            |           | postgres=CTc/postgres
(8 rows)
```

Create Database (using pgAdmin4)

Follow the below steps to create a new database using pgAdmin.

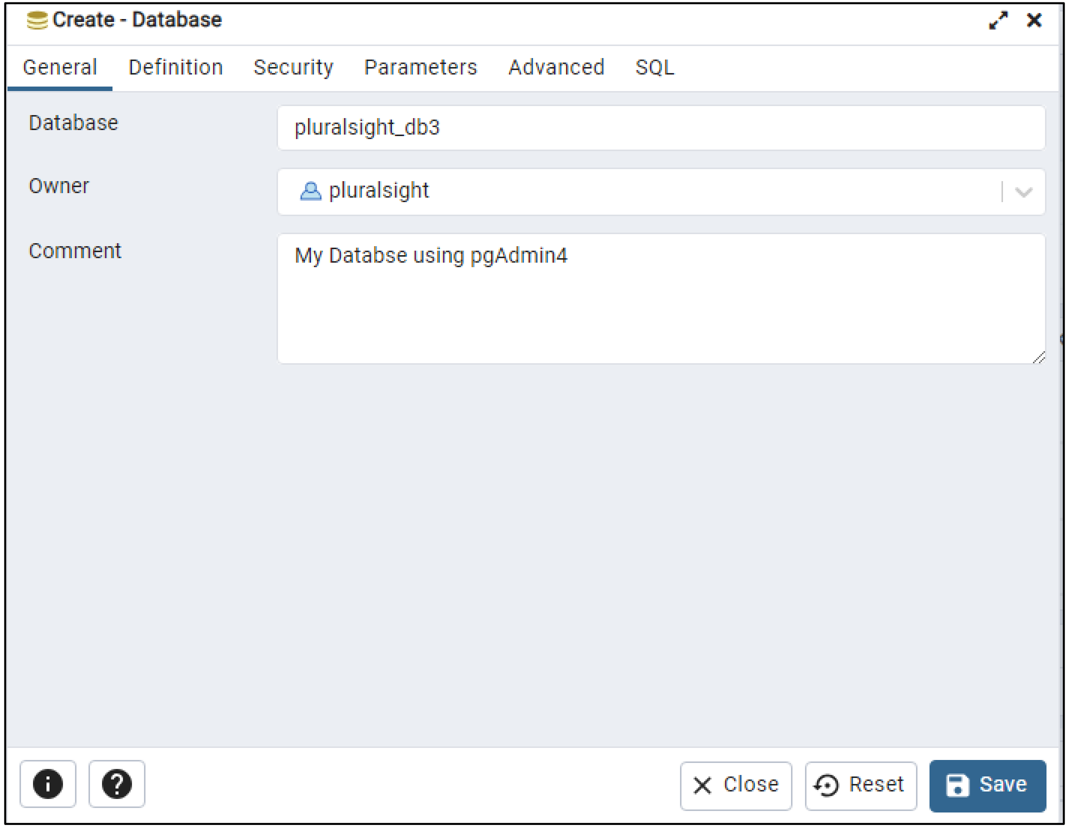
Step 1: Log in to PostgreSQL via pgAdmin.

Step 2: Right click on the Databases menu and then click on Create-> Database... sub-menu item as depicted below:



Create Database (using pgAdmin4)

Step 3: Now enter the new database name, owner, and configure parameters and click the OK button as depicted below:



Create - Database

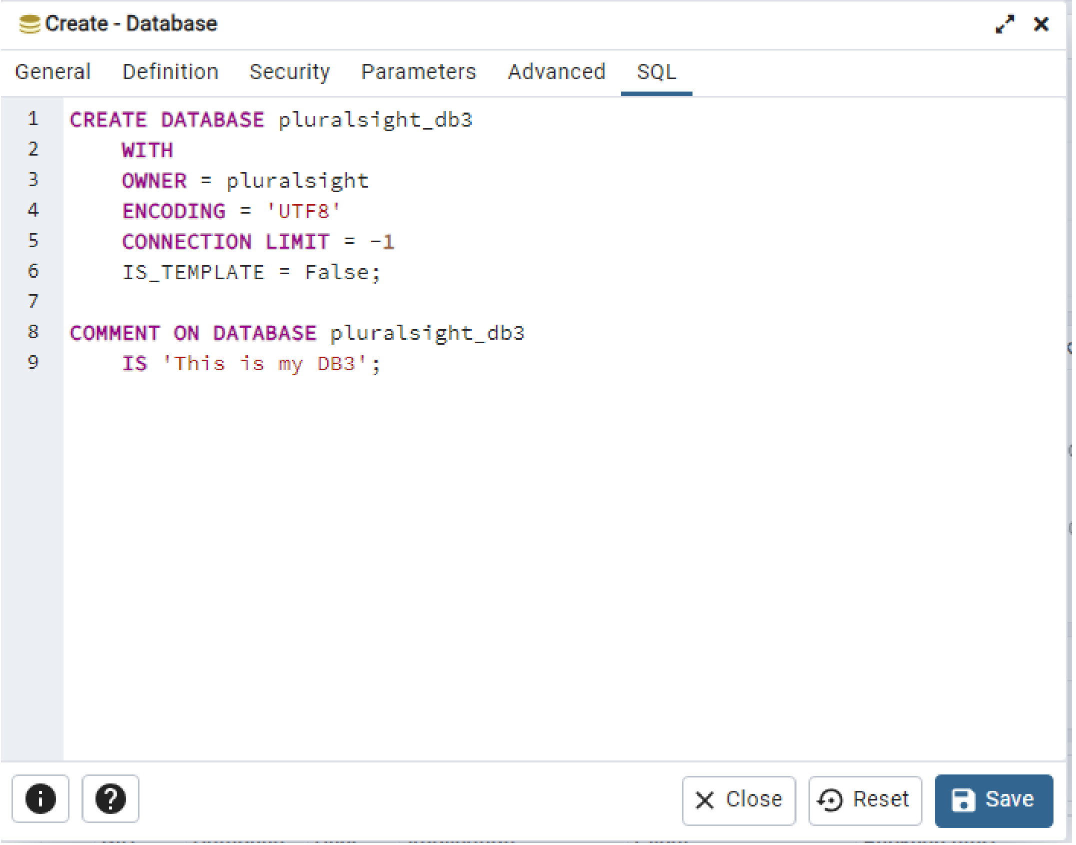
Database: pluralsight_db3

Owner: pluralsight

Comment: My Databse using pgAdmin4

Create Database (using pgAdmin4)

Step 3: Now enter the new database name, owner, and configure parameters and click the OK button as depicted below:



```
CREATE DATABASE pluralsight_db3
    WITH
    OWNER = pluralsight
    ENCODING = 'UTF8'
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;

COMMENT ON DATABASE pluralsight_db3
    IS 'This is my DB3';
```

Hands-On Lab Exercise-2

(Topic: Create Database)

PLURALSIGHT

Rename Database

A database can be renamed in PostgreSQL using the ALTER DATABASE RENAME TO statement.

When renaming a database, the following procedures must be followed:

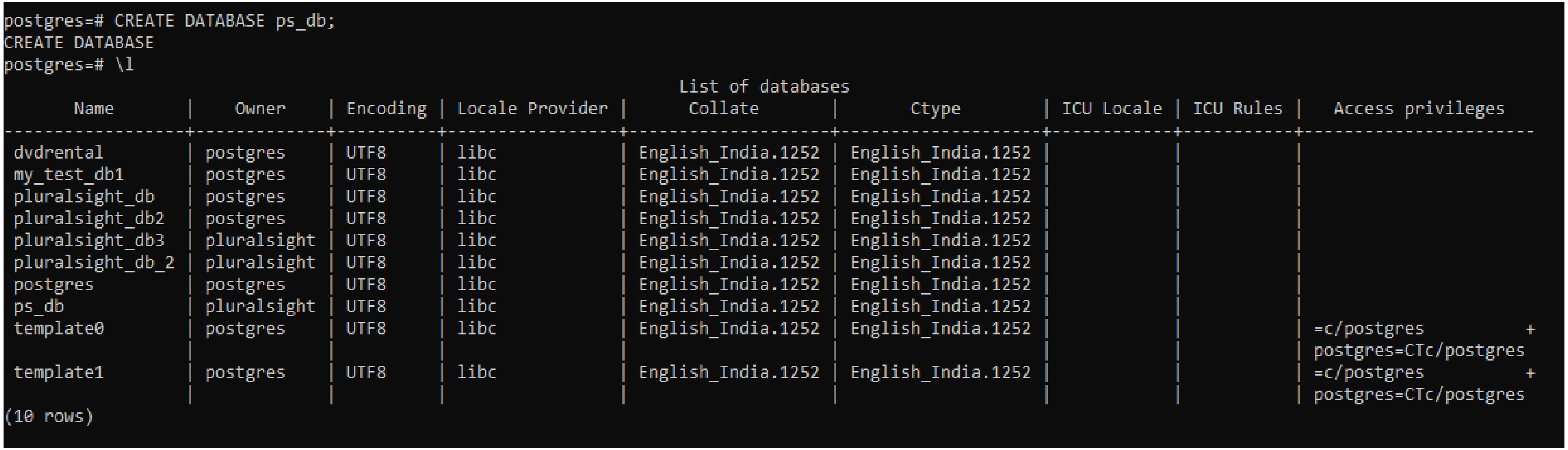
- Open a new database connection to disconnect from the one you wish to rename.
- Cut off any connections to the database that has to be renamed.
- The ALTER DATABASE statement can now be used to rename the database.

Rename Database

Step 1: Create a database named “ps_db” using the below commands:

CREATE DATABASE ps_db;

\connect ps_db;



```
postgres=# CREATE DATABASE ps_db;
CREATE DATABASE
postgres=# \l
                                                                      List of databases
       Name       |    Owner    | Encoding | Locale Provider |      Collate       |       Ctype        | ICU Locale | ICU Rules |   Access privileges
------------------+-------------+----------+-----------------+--------------------+--------------------+------------+-----------+-----------------------
 dvdrental        | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 my_test_db1      | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db   | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db2  | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db3  | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db_2 | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 postgres         | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 ps_db            | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 template0        | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                  |             |          |                 |                    |                    |            |           | postgres=CTc/postgres
 template1        | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                  |             |          |                 |                    |                    |            |           | postgres=CTc/postgres
(10 rows)
```

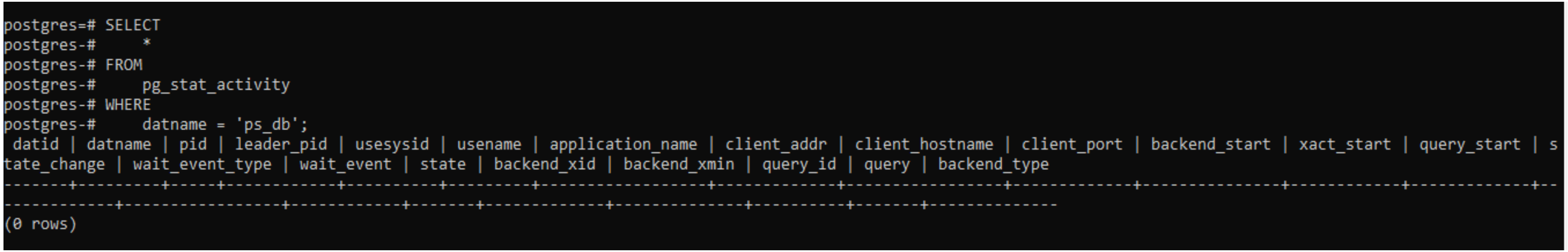
Rename Database

Step 2: Disconnect and connect to another db:

\connect postgres;

Step 3: Use the below query to check all active connections to the “test_db” database:

*SELECT **
FROM
pg_stat_activity
WHERE
datname = ‘ps_db';

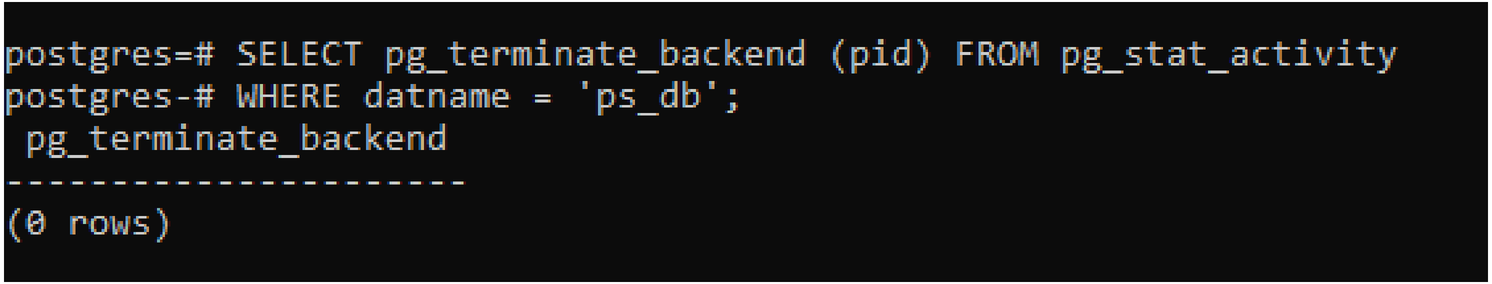


```
postgres=# SELECT
postgres-#     *
postgres-# FROM
postgres-#     pg_stat_activity
postgres-# WHERE
postgres-#     datname = 'ps_db';
 datid | datname | pid | leader_pid | usesysid | usename | application_name | client_addr | client_hostname | client_port | backend_start | xact_start | query_start | s
tate_change | wait_event_type | wait_event | state | backend_xid | backend_xmin | query_id | query | backend_type
-------+---------+-----+------------+----------+---------+------------------+-------------+-----------------+-------------+---------------+------------+-------------+--
------------+-----------------+------------+-------+-------------+--------------+----------+-------+--------------
(0 rows)
```

Rename Database

Step 4: Use the below query to terminate all the connections to the test_db database:

SELECT pg_terminate_backend (pid) FROM pg_stat_activity
WHERE datname = 'ps_db';

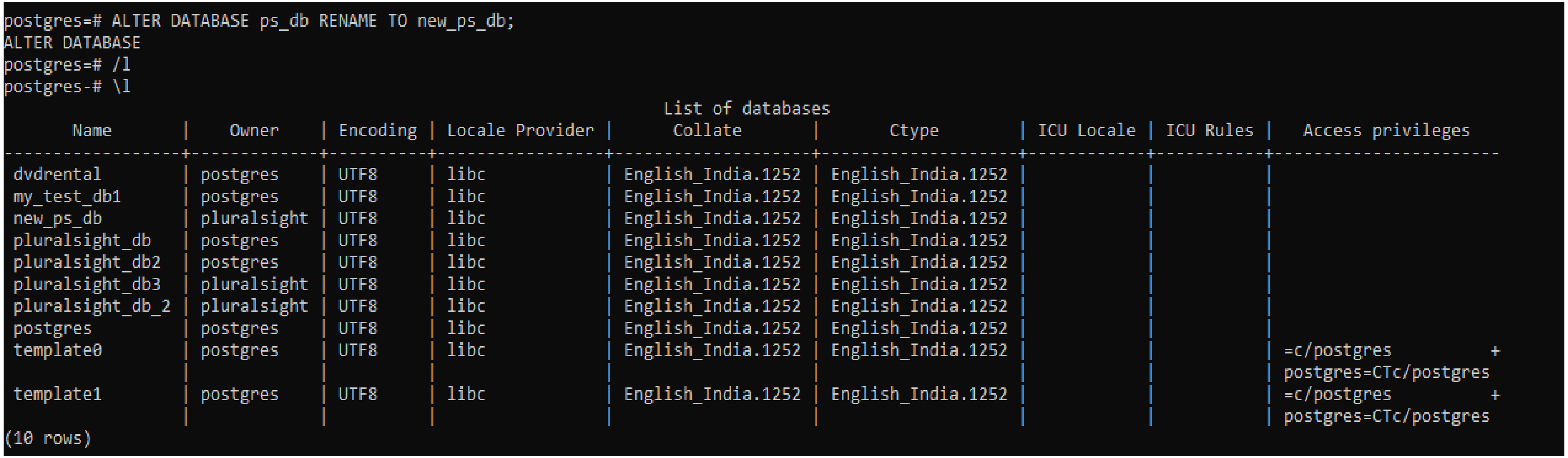


```
postgres=# SELECT pg_terminate_backend (pid) FROM pg_stat_activity
postgres-# WHERE datname = 'ps_db';
 pg_terminate_backend
----------------------
(0 rows)
```

Rename Database

Step 5: Now use the ALTER DATABASE RENAME TO statement to rename the database as “new_test_db”(say) as follows:

ALTER DATABASE ps_db RENAME TO new_ps_db;



```
postgres=# ALTER DATABASE ps_db RENAME TO new_ps_db;
ALTER DATABASE
postgres=# /l
postgres-# \l
                                                              List of databases
       Name        |    Owner    | Encoding | Locale Provider |      Collate       |       Ctype        | ICU Locale | ICU Rules |   Access privileges
-------------------+-------------+----------+-----------------+--------------------+--------------------+------------+-----------+-----------------------
 dvdrental         | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 my_test_db1       | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 new_ps_db         | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db    | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db2   | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db3   | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 pluralsight_db_2  | pluralsight | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 postgres          | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           |
 template0         | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                   |             |          |                 |                    |                    |            |           | postgres=CTc/postgres
 template1         | postgres    | UTF8     | libc            | English_India.1252 | English_India.1252 |            |           | =c/postgres          +
                   |             |          |                 |                    |                    |            |           | postgres=CTc/postgres
(10 rows)
```