

Setting up a PostgreSQL DB using Command Line interface on Ubuntu Distribution

The following documentation help's in seting up a PostgreSQL from scratch, which means you'll need to install PostgreSQL, create a database, and then begin interacting with it using the command line. Here's a step-by-step guide to get you from installation to creating your first database and table.

Step 1: Install PostgreSQL

The exact command to install PostgreSQL depends on your operating system. Assuming you're using a Debian-based Linux distribution (like Ubuntu), you can install PostgreSQL using the following commands:

```
sudo apt update
sudo apt install postgresql postgresql-contrib
```

Step 2: Start and Enable PostgreSQL Service

Ensure the PostgreSQL service is running and enabled to start on boot:

```
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

Step 3: Switch to the PostgreSQL User

PostgreSQL creates a user named `postgres` during installation for managing databases. Switch to this user:

```
sudo -i -u postgres
```

Step 4: Create a New Role

You might want to create a new role (user) to manage your databases. To do so, use the `createuser` command. Replace `your_username` with your desired username:

```
createuser --interactive
```

When prompted, enter your new username and decide whether it should have superuser permissions.

Step 5: Create Your First Database

Now, create your first database. Replace `your_database_name` with the name you wish to give your database:

```
createdb your_database_name
```

Step 6: Connect to Your Database

Connect to your newly created database using the `psql` utility:

```
psql -d your_database_name
```

If you want to connect using the new user you created, you might need to set a password for that user and then use it to connect. First, set the password:

```
psql -d postgres -c "ALTER USER your_username WITH PASSWORD 'your_password';"
```

Then, connect to your database:

```
psql -d your_database_name -U your_username
```

Step 7: Create a Table

Once inside the `psql` command-line interface, you can start defining your database schema. Here's an example command to create a simple table:

```
CREATE TABLE companies (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    website VARCHAR(100)  
);
```

Step 8: Insert Data

Insert some data into your new table:

```
INSERT INTO companies (name, website) VALUES ('Example Company', 'https://example.com')
```

Step 9: Query Data

Retrieve the data you've inserted:

```
SELECT * FROM companies;
```

Step 10: Exit `psql`

Exit the `psql` interface:

```
\q
```

Additional Information

- **Documentation:** For more detailed information, refer to the PostgreSQL [official documentation](#).
- **Management:** You can manage PostgreSQL with the `postgres` user or the role you created. For web-based management, pgAdmin 4 is a popular choice.
- **Security:** Ensure your PostgreSQL installation is secured, especially if it's accessible over a network. This includes setting strong passwords and considering firewall rules.

This guide covers the basics to get you started with PostgreSQL on Linux, from installation to creating your first database and table.