

Postgres Configuration for StockEcom 241015

1. **Connecting to PostgreSQL:**

To connect to PostgreSQL as the superuser 'postgres':

```
``bash
```

```
psql -U postgres
```

```
...
```

To connect to the specific StockEcom database as 'nickbiy':

```
``bash
```

```
psql -U nickbiy -d StockEcom
```

```
...
```

2. **Dropping the Database:**

To drop the StockEcom database:

```
``sql
```

```
DROP DATABASE IF EXISTS "StockEcom";
```

```
...
```

3. **Dropping Users:**

To drop the user 'nickbiy':

```
``sql
```

```
DROP USER IF EXISTS nickbiy;
```

```
...
```

4. **Creating the Database:**

To create a new database called StockEcom:

```
```sql
```

```
CREATE DATABASE "StockEcom";
```

```
```
```

5. ****Creating a New User:****

To create the user 'nickbiy' with a password:

```
```sql
```

```
CREATE USER nickbiy WITH PASSWORD 'your_password';
```

```
```
```

6. ****Granting Privileges to User:****

To grant all privileges on StockEcom to 'nickbiy':

```
```sql
```

```
GRANT ALL PRIVILEGES ON DATABASE "StockEcom" TO nickbiy;
```

```
```
```

7. ****Granting Permissions on Schema:****

To grant usage and creation privileges on the public schema to 'nickbiy':

```
```sql
```

```
GRANT USAGE, CREATE ON SCHEMA public TO nickbiy;
```

```
```
```

To make 'nickbiy' the owner of the public schema:

```
```sql
```

```
ALTER SCHEMA public OWNER TO nickbiy;
```

```
```
```

8. ****Listing Tables and Relations:****

To list all tables (relations) in the StockEcom database:

```
```sql
\dt
```
```

9. ****Describing Tables:****

To describe the structure of a table (e.g., auth_user):

```
```sql
\d auth_user
```
```

10. ****Querying for Users:****

To check if a user exists by email:

```
```sql
SELECT id, username, email FROM auth_user WHERE email = 'admin@gmail.com';
```
```

11. ****Updating User Roles:****

To make 'admin@gmail.com' a superuser and give them admin privileges:

```
```sql
UPDATE auth_user
SET is_superuser = TRUE, is_staff = TRUE
WHERE email = 'admin@gmail.com';
```
```

12. ****Password Hashing in Django:****

To generate a hashed password in Django shell:

```
``python  
  
from django.contrib.auth.hashers import make_password  
  
hashed_password = make_password('admin@123')  
``
```