Database Roles

PostgreSQL **manages database access permissions** using the concept of *roles*.

A role can be thought of as **either a database user, or a group of database users**, depending on how the role is set up.

Roles can **own database objects (for example, tables and functions)** and can assign privileges on those objects to other roles to control who has access to which objects.

Furthermore, it is possible to **grant *membership* in a role to another role**, thus allowing the member role to use privileges assigned to another role.

The concept of roles subsumes the concepts of “users” and “groups”. In PostgreSQL versions before 8.1, users and groups were distinct kinds of entities, but **now there are only roles**. **Any role can act as a user, a group, or both**.

Create Database Roles

> *CREATE ROLE name;*
>
> *DROP ROLE name;*

Experiment 1:

```
CREATE ROLE usrtest;
CREATE USER usr_test2;

select rolname from PG_ROLES;--show roles in pgsql

DROP ROLE usrtest;
DROP USER usr_test2;

CREATE USER "usrtest3";
DROP ROLE "usrtest3";
```

Tips:

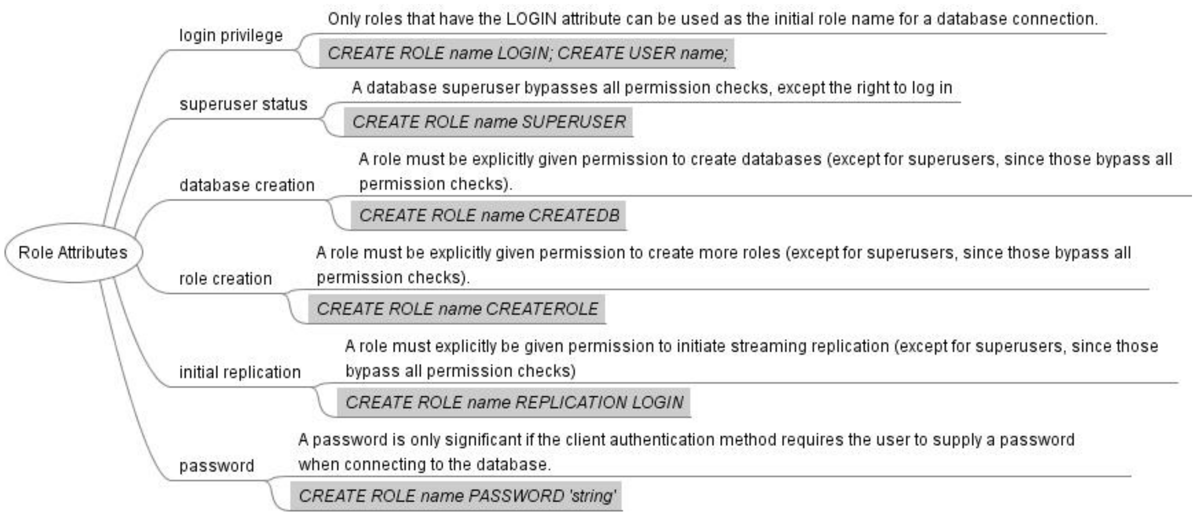
- `name` follows the rules for SQL identifiers: either **unadorned without special characters**, or **double-quoted**.
- In order to bootstrap the database system, a freshly initialized system always contains one predefined role. This role is always a “superuser”, and by default (unless altered when running `initdb`) it will have the same name as the operating system user that initialized the database cluster. Customarily, this role will be named `postgres`.

Role Attributes

A database role can have a number of attributes that define its privileges and interact with the client authentication system.

A role's attributes can be modified after creation with `ALTER ROLE`.

See the reference pages for the CREATE ROLE , ALTER ROLE and DROP ROLE commands for details.


Role Attributes
login privilege
Only roles that have the LOGIN attribute can be used as the initial role name for a database connection.
CREATE ROLE name LOGIN; CREATE USER name;
superuser status
A database superuser bypasses all permission checks, except the right to log in
CREATE ROLE name SUPERUSER
database creation
A role must be explicitly given permission to create databases (except for superusers, since those bypass all permission checks).
CREATE ROLE name CREATEDB
role creation
A role must be explicitly given permission to create more roles (except for superusers, since those bypass all permission checks).
CREATE ROLE name CREATEROLE
initial replication
A role must explicitly be given permission to initiate streaming replication (except for superusers, since those bypass all permission checks)
CREATE ROLE name REPLICATION LOGIN
password
A password is only significant if the client authentication method requires the user to supply a password when connecting to the database.
CREATE ROLE name PASSWORD 'string'

Tips: It is good practice to create a role that has the `CREATEDB` and `CREATEROLE` privileges, but is not a superuser, and then use this role for all routine management of databases and roles. This approach avoids the dangers of operating as a superuser for tasks that do not really require it.