Create Db/User/Schema

Create database Psql / createdb utility:

* Syntax from psql: Create database databasename owner ownername;
* Syntax from command line: Createdb <dbname>.
* Syntax for help: createdb --help

Drop database – Psql/ dropdb utility:

* We can’t drop the database which we are connected.
* Example:

scott=# drop database scott;

ERROR: cannot drop the currently open database

* Syntax from psql: Drop database <dbname>.
* Syntax from command line: dropdb <dbname>.
* Syntax for dropdb help: dropdb –help

Create user – Psql/ createuser utility/ Interactive:

* Syntax from psql: create user scott login superuser password 'welcome';
* Syntax from command line: createuser <username>
* Syntax for interactive user creation from command line:
* Example:
* createuser --interactive joe

Shall the new role be a superuser? (y/n) n

Shall the new role be allowed to create databases? (y/n) y

Shall the new role be allowed to create more new roles? (y/n) y

* Syntax for createuser help: createuser --help

Drop user - Psql/ dropuser utility:

* Syntax from psql: drop user <username>
* Syntax from command line: dropuser <username>
* Dropping a user with objects or privileges will return an error.

Example:

postgres=# drop user test1;

ERROR: role "test1" cannot be dropped because some objects depend on it

* Assign the user privileges to another user before dropping the user.

Example:

REASSIGN OWNED BY user to postgres;

Drop role username;

Grant:

* Grant CONNECT to the database:

GRANT CONNECT ON DATABASE database\_name TO username;

* Grant USAGE on schema:

GRANT USAGE ON SCHEMA schema\_name TO username;

* Grant on all tables for DML statements: SELECT, INSERT, UPDATE, DELETE

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA schema\_name TO username;

* Grant all privileges on all tables in the schema:

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema\_name TO username;

* Grant all privileges on all sequences in the schema:

GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA schema\_name TO username;

* Grant permission to create database:

ALTER USER username CREATEDB;

* Make a user superuser:

ALTER USER myuser WITH SUPERUSER;

* Remove superuser status:

ALTER USER username WITH NOSUPERUSER;

* Column Level access:

GRANT SELECT (col1), UPDATE (col1) ON mytable TO user;

Revoke Examples

* Revoke Delete/update privilege on table from user

REVOKE DELETE, UPDATE ON products FROM user;

* Revoke all privilege on table from user

REVOKE ALL ON products FROM user;

* Revoke select privilege on table from all users (Public)

REVOKE SELECT ON products FROM PUBLIC;

Create & Drop Schema

* Create Schema

CREATE schema <schema\_name>;

* Create Schema for a user, the schema will also be named as the user

Create schema authorization <username>;

* Create Schema named John, that will be owned by brett

CREATE schema IF NOT EXISTS john AUTHORIZATION brett;

* Drop a Schema

Drop schema <schema\_name>;

(We cannot drop schema if there are any object associate with it.)

Schema Search Path:

* Show search path can be used to find the current search path.

Example:

postgres=# show search\_path;

search\_path

-----------------

"$user", public

( 1 row)

* Default "$user" is a special option that says if there is a schema that matches the current user (i.e SELECT SESSION\_USER;), then search within that schema.
* Search path can be set at session level, user level, database level and cluster level

Example:

Test1=# SET search\_path TO test1,public;

Test1=# \dt

List of relations

Schema | Name | Type | Owner

--------+---------+-------+----------

test1 | abc | table | test1

(1 rows)