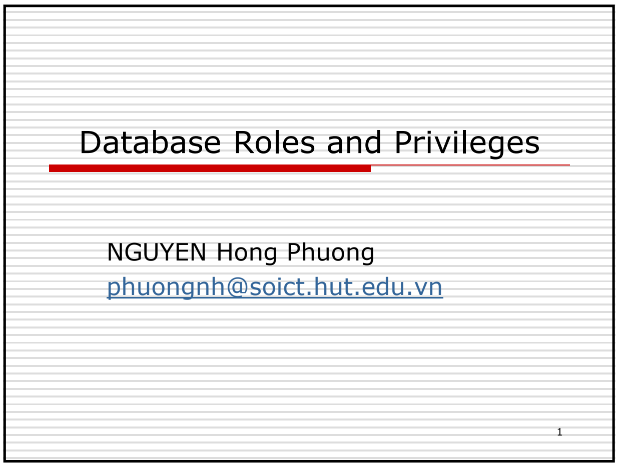
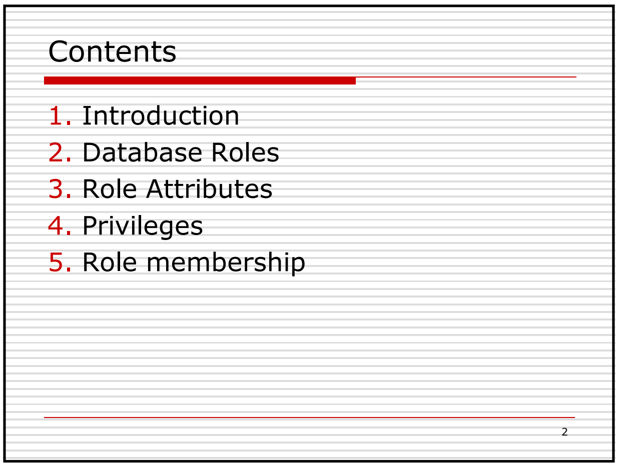


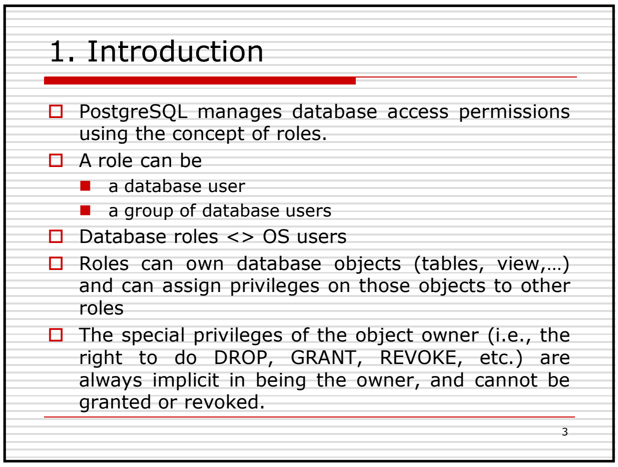
Database Roles and Privileges

NGUYEN Hong Phuong
phuongnh@soict.hut.edu.vn



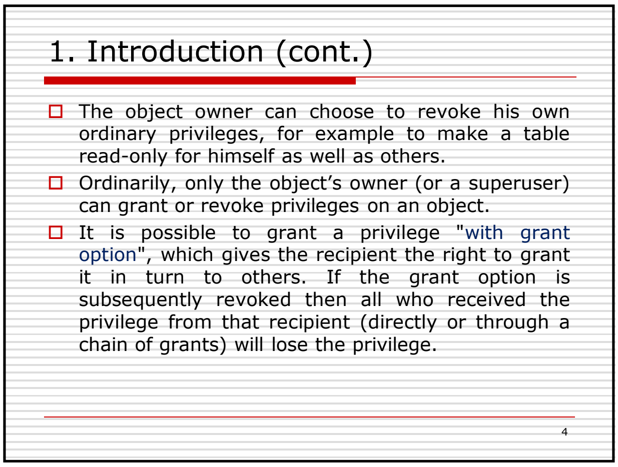
Contents

1. Introduction
2. Database Roles
3. Role Attributes
4. Privileges
5. Role membership



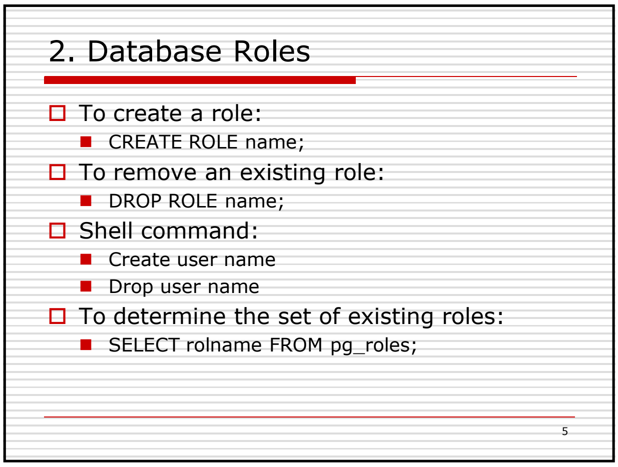
1. Introduction

- PostgreSQL manages database access permissions using the concept of roles.
- A role can be
  - a database user
  - a group of database users
- Database roles <> OS users
- Roles can own database objects (tables, view,…) and can assign privileges on those objects to other roles
- The special privileges of the object owner (i.e., the right to do DROP, GRANT, REVOKE, etc.) are always implicit in being the owner, and cannot be granted or revoked.



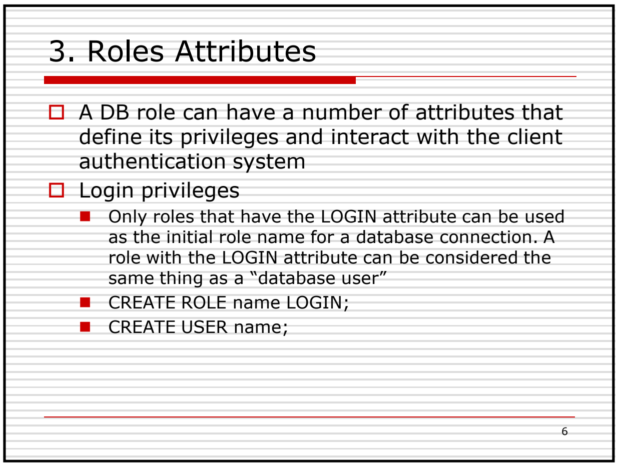
1. Introduction (cont.)

- The object owner can choose to revoke his own ordinary privileges, for example to make a table read-only for himself as well as others.
- Ordinarily, only the object's owner (or a superuser) can grant or revoke privileges on an object.
- It is possible to grant a privilege "with grant option", which gives the recipient the right to grant it in turn to others. If the grant option is subsequently revoked then all who received the privilege from that recipient (directly or through a chain of grants) will lose the privilege.



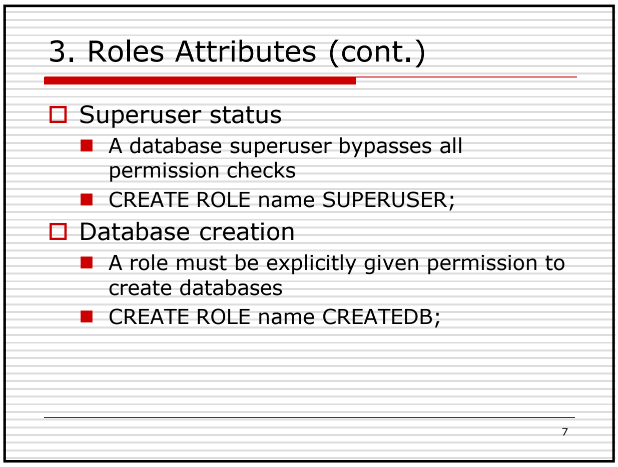
2. Database Roles

- To create a role:
  - CREATE ROLE name;
- To remove an existing role:
  - DROP ROLE name;
- Shell command:
  - Create user name
  - Drop user name
- To determine the set of existing roles:
  - SELECT rolname FROM pg_roles;



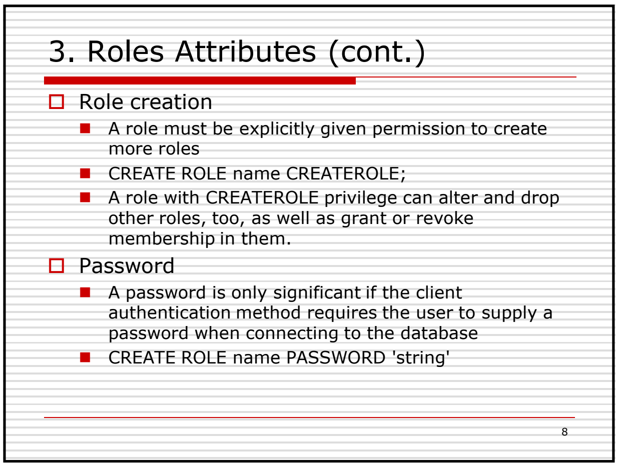
3. Roles Attributes

- A DB role can have a number of attributes that define its privileges and interact with the client authentication system
- Login privileges
  - Only roles that have the LOGIN attribute can be used as the initial role name for a database connection. A role with the LOGIN attribute can be considered the same thing as a "database user"
  - CREATE ROLE name LOGIN;
  - CREATE USER name;



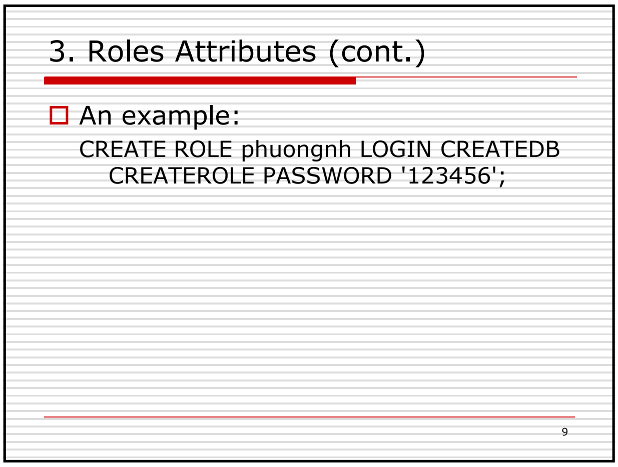
3. Roles Attributes (cont.)

- Superuser status
  - A database superuser bypasses all permission checks
  - CREATE ROLE name SUPERUSER;
- Database creation
  - A role must be explicitly given permission to create databases
  - CREATE ROLE name CREATEDB;



3. Roles Attributes (cont.)

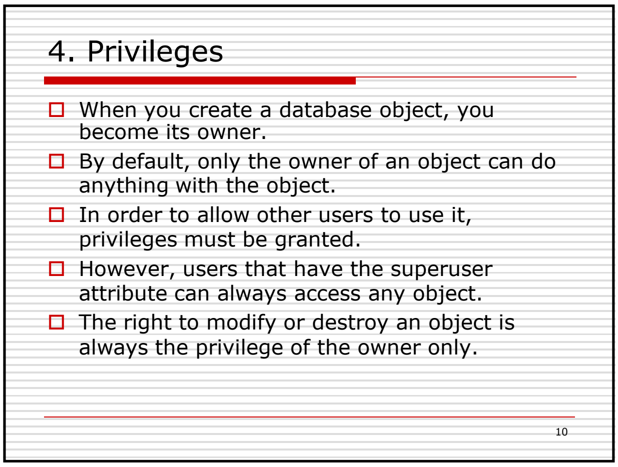
- Role creation
  - A role must be explicitly given permission to create more roles
  - CREATE ROLE name CREATEROLE;
  - A role with CREATEROLE privilege can alter and drop other roles, too, as well as grant or revoke membership in them.
- Password
  - A password is only significant if the client authentication method requires the user to supply a password when connecting to the database
  - CREATE ROLE name PASSWORD 'string'



3. Roles Attributes (cont.)

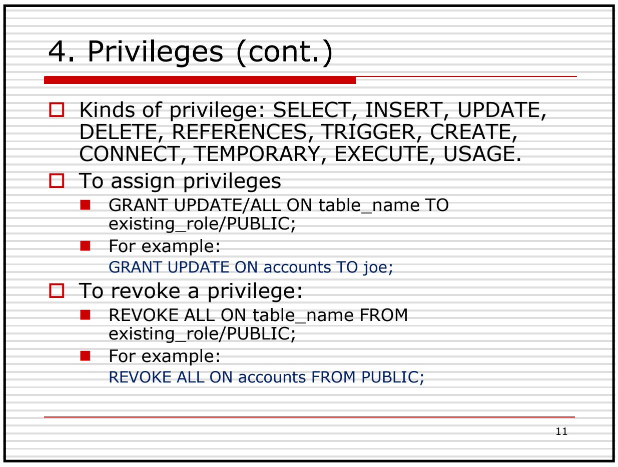
- An example:

  CREATE ROLE phuongnh LOGIN CREATEDB CREATEROLE PASSWORD '123456';



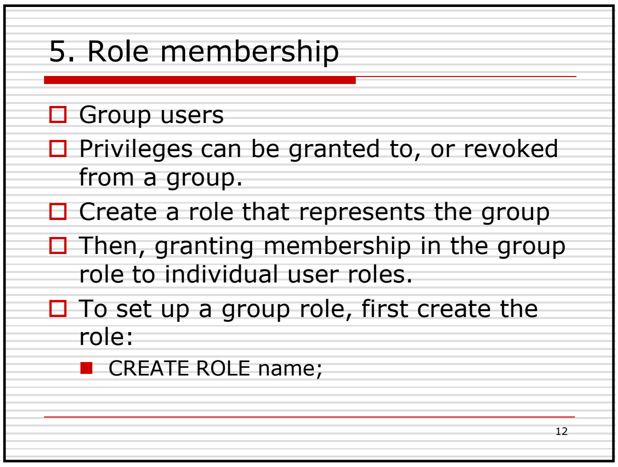
4. Privileges

- When you create a database object, you become its owner.
- By default, only the owner of an object can do anything with the object.
- In order to allow other users to use it, privileges must be granted.
- However, users that have the superuser attribute can always access any object.
- The right to modify or destroy an object is always the privilege of the owner only.



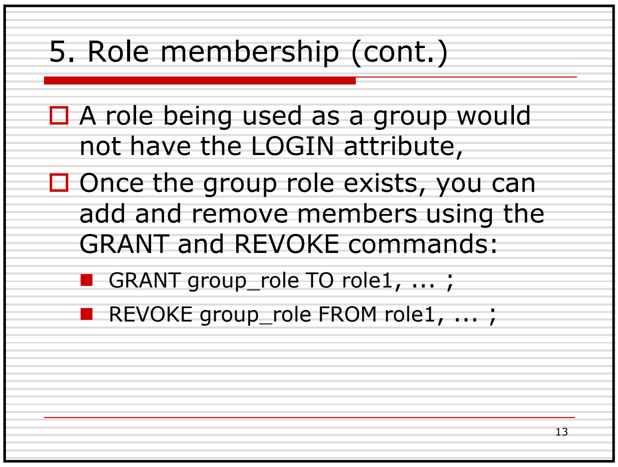
4. Privileges (cont.)

- Kinds of privilege: SELECT, INSERT, UPDATE, DELETE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE, USAGE.
- To assign privileges
  - GRANT UPDATE/ALL ON table_name TO existing_role/PUBLIC;
  - For example:
    GRANT UPDATE ON accounts TO joe;
- To revoke a privilege:
  - REVOKE ALL ON table_name FROM existing_role/PUBLIC;
  - For example:
    REVOKE ALL ON accounts FROM PUBLIC;



5. Role membership

- Group users
- Privileges can be granted to, or revoked from a group.
- Create a role that represents the group
- Then, granting membership in the group role to individual user roles.
- To set up a group role, first create the role:
  - CREATE ROLE name;



5. Role membership (cont.)

- A role being used as a group would not have the LOGIN attribute,
- Once the group role exists, you can add and remove members using the GRANT and REVOKE commands:
  - GRANT group_role TO role1, ... ;
  - REVOKE group_role FROM role1, ... ;

13