Role và User

NGUYEN HongPhuong
Email: phuongnh@soict.hust.edu.vn
Site: https://users.soict.hust.edu.vn/phuongnh
Face: https://www.facebook.com/phuongnhbk
Hanoi University of Science and Technology

Nội dung

- Trong bất kỳ hệ thống quản trị cơ sở dữ liệu nào, thì vấn đề an ninh - bảo mật luôn được đặt lên hàng đầu.
- Bước đầu tiên là xác định rõ ràng những tài khoản nào sẽ được quyền truy cập, xem hoặc chỉnh sửa dữ liệu.
  - Ví dụ, các trưởng bộ phận có thể xem được tài khoản lương của nhân viên, cấp quản lý cao hơn có quyền xem và chỉnh sửa, trong khi nhân viên chỉ có thể xem được tài khoản của chính họ.

- Về mặt bản chất, role là 1 phần của tiered security model:
  - Login security: thực hiện quá trình kết nối tới server
  - Database security: nhận quyền truy cập tới cơ sở dữ liệu
  - Database object: nhận quyền truy cập tới từng đối tượng và dữ liệu riêng biệt trong toàn hệ thống

- Các role của server thường được giám sát và quản lý bởi Database Administrator – DBA. Ở chế độ mặc định, các role này được thiết lập public đối với tất cả các tài khoản, và toàn bộ những tài khoản sau khi thêm vào SQL Server cũng sẽ tự động được gán role public.

SQL Server Roles

- A role is a group of permissions. Roles help you simplify permission management. For example, instead of assigning permissions to users individually, you can group permissions into a role and add users to that role:
  - First, create a role.
  - Second, assign permissions to the role.
  - Third, add one or more users to the role.

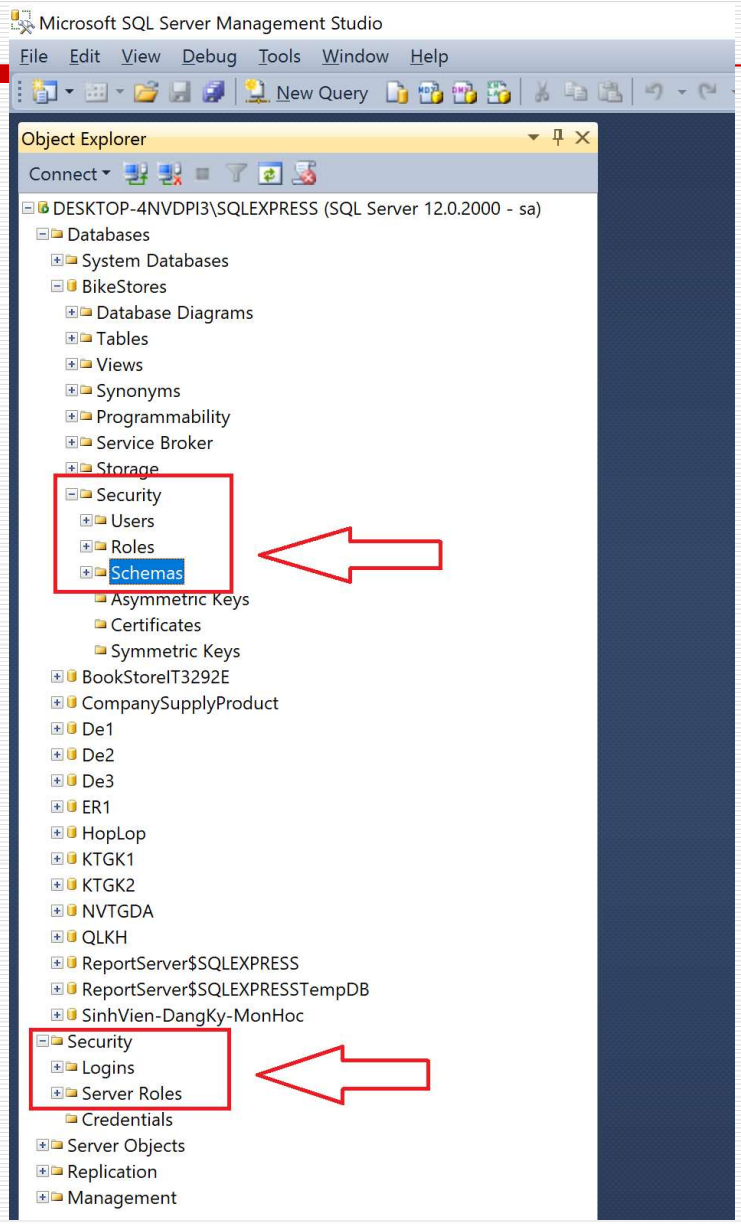
SQL Server Roles

- SQL Server provides you with three main role types:
  - **Server-level roles**: manage the permissions on SQL Server-like changing server configuration.
  - **Database-level roles**: manage the permissions on databases like creating tables and querying data.
  - **Application-level roles**: allow an application to run with its own, user-like permissions.

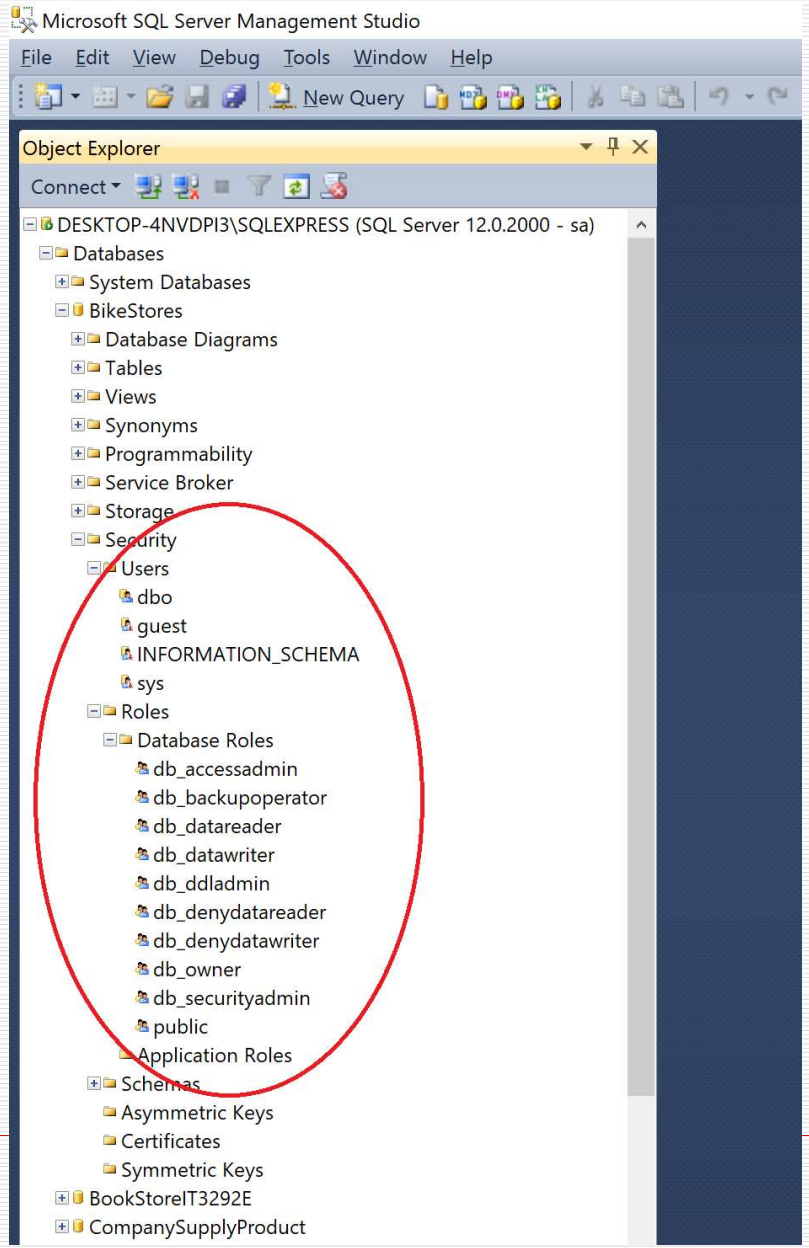
SQL Server Roles

- For each type, SQL Server provides two types:
  - **Fixed server roles**: are the built-roles provided by SQL Server. These roles have a fixed set of permissions.
  - **User-defined roles**: are the roles you define to meet specific security requirements.

- 2 security
  - Database
    - Users
    - Roles
    - Schemas
  - Server instance
    - Logins
    - Server Roles



Users và Roles của database



Users và Roles của database

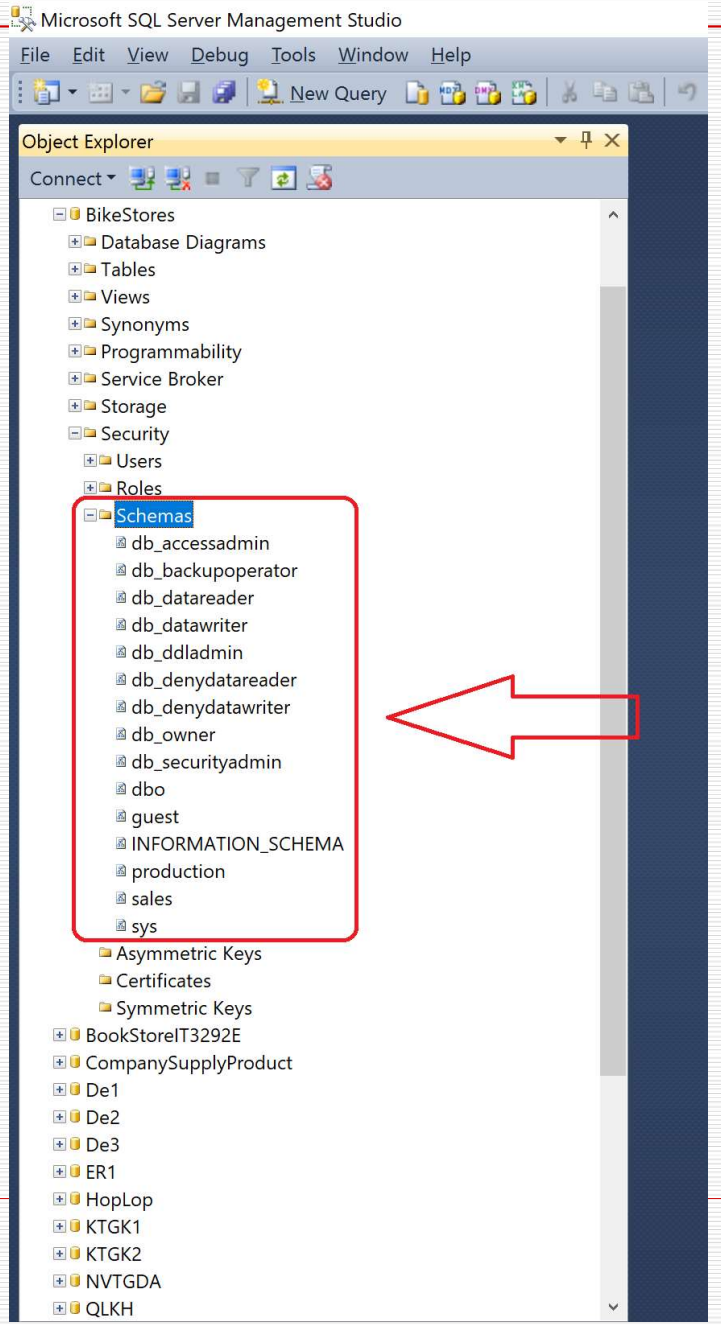
- Việc tạo CSDL là trách nhiệm của người quản trị. Lưu ý một số điểm sau khi tạo bảng:
  - db_owner: toàn bộ người dùng có quyền full – access
  - db_accessadmin: người dùng có quyền quản lý các Windows Group và tài khoản SQL Server đăng nhập
  - db_datareader: người dùng có thể đọc được toàn bộ dữ liệu
  - db_datawriter: người dùng có quyền thêm, xóa hoặc chỉnh sửa dữ liệu trong bảng

Users và Roles của database

- db_ddladmin: người dùng có thể sử dụng các file dynamic – link library (DLL)
- db_securityadmin: người dùng có thể chỉnh sửa vai trò role và quản lý các bậc quản lý, phân quyền khác
- db_bckupoperator: người dùng có thể sao lưu cơ sở dữ liệu
- db_denydatareader: người dùng không thể xem dữ liệu trong bảng
- db_denydatawriter: người dùng không thể xem, thay đổi hoặc xóa dữ liệu trong bảng

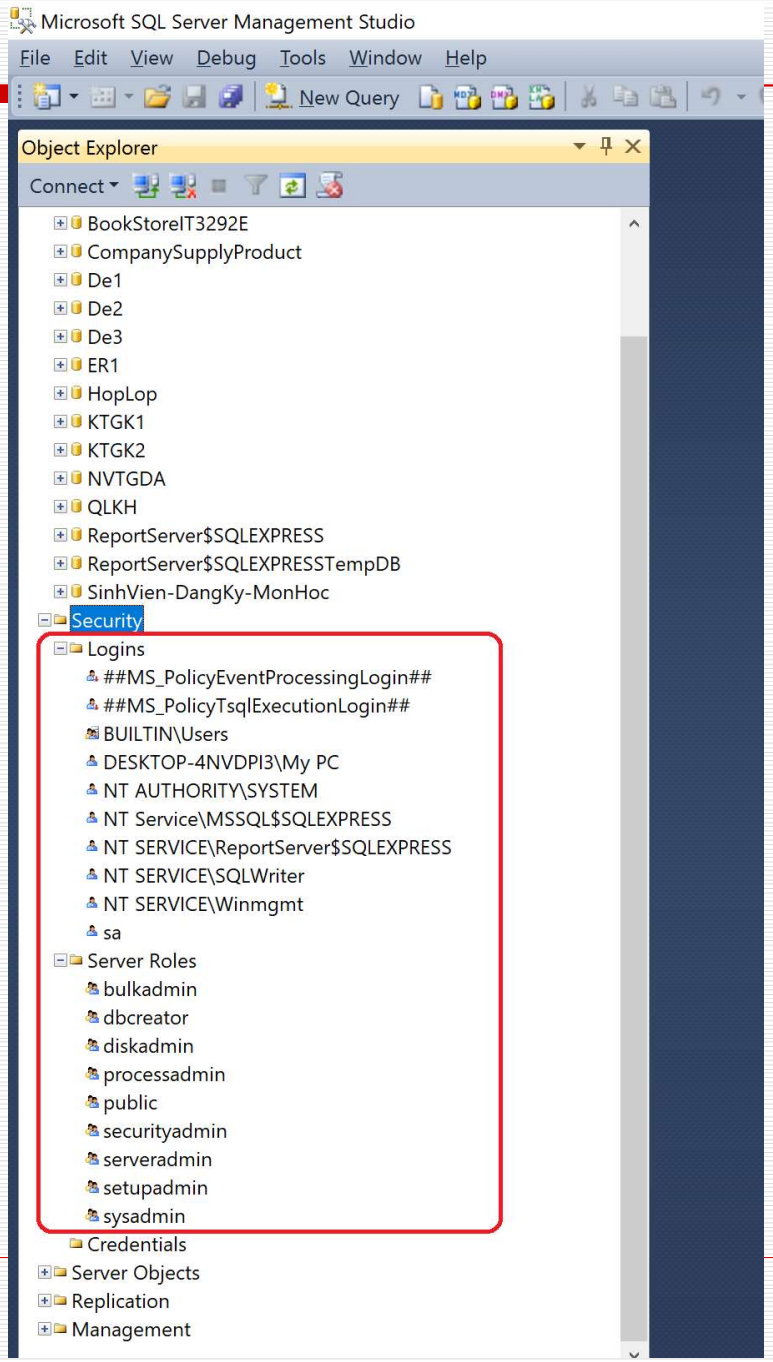
Schemas của database

- Database Schemas giống như **namespace** trong lập trình. Nó là 1 thùng chứa các Object(Table, SP, View, ...).
- Có thể giới hạn quyền đăng nhập người dùng bằng cách chỉ cho phép truy cập tới 1 Schema nhất định của Database.



Logins và Server Roles của Server instance

- Có 9 server roles cố định, không thể thêm/xóa bất kỳ roles nào



Giải thích về các Server Role

Server Role	Giải thích
sysadmin	Có thể làm bất kỳ điều gì trong SQL Server
serveradmin	Có thể tuỳ chỉnh cấu hình máy chủ và tắt máy chủ
setupadmin	Có thể thêm và xóa các máy chủ được liên kết bằng cách sử dụng các câu lệnh Transact-SQL
securityadmin	Có thể GRANT, DENY, REVOKE với cơ sở dữ liệu được cấp quyền truy cập. Có thể tự thay đổi mật khẩu
processadmin	Có thể tắt hoặc tạm dừng bất kỳ tiến trình nào hoạt động trên SQL Server
dbcreator	Có thể tạo, thay đổi, xoá và khôi phục bất kỳ cơ sở dữ liệu nào
diskadmin	Có thể quản lý các file của SQL Server
bulkadmin	Có thể thực thi các câu lệnh BULK INSERT
public	Không thể làm bất kì điều gì tác động tới cơ sở dữ liệu. Chỉ có thể truy cập tới các Object được public bên trong cơ sở dữ liệu

Two authentication modes in SQL Server

- Help to login and connect with the SQL Server.
  - Windows Authentication
  - SQL Server Authentication

Windows Authentication

- Windows authentication mode enables local Windows authentication with SQL Server, where you can login with your local Windows credentials.
- This is the default authentication mode and is more secure than SQL Server authentication. It is also referred to as integrated security which is tightly integrated with Windows. Users who are already authenticated with Windows and need not provide any additional credentials while connecting to SQL Server. It is also called a trusted connection. The user account is confirmed by Windows.

Windows Authentication

- In Windows authentication, login can be created in SQL Server for an entire Windows group which simplifies managing account administration.

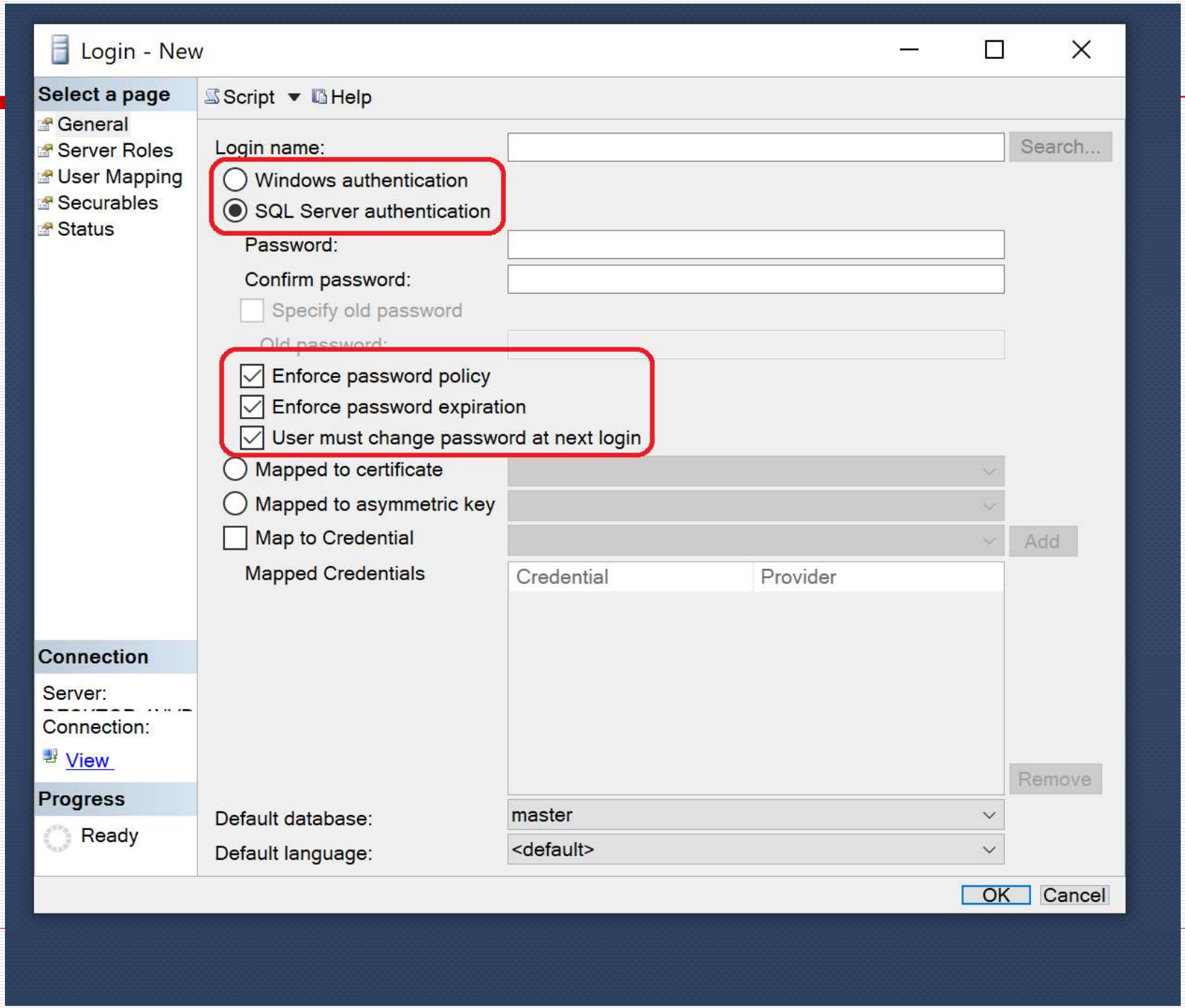
- Windows authentication uses Kerberos security protocol, provides password policy enforcement, and supports password expiration.

SQL Server Authentication

- You 've learned about creating a login using Windows user account. Here, you will learn to create a login using SQL Server user.
- At the time of installing SQL Server, if you select Mixed Mode, then you must provide a password for the built-in System Administrator or sa account. It is highly recommended to create a strong password for the sa account; otherwise disable this account as it is mapped to the sysadmin server role and has administrative rights on the whole server. Hence it is vulnerable to attack by hackers.

SQL Server Authentication

- Select three optional password policies:
  - **Enforce password policy**: The Windows password policies of the computer are enforced for SQL Server Logins.
  - **Enforce password expiration**: The maximum password age policy of the computer is enforced.
  - **User must change password at next login**: If this option is selected, the user is required to change their SQL Server login password the next time they login.


Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
master
Default language:
<default>
Connection
Server:
Connection:
View
Progress
Ready
OK
Cancel

Tạo tài khoản login

- Dùng thủ tục lưu trữ hệ thống sp_addlogin
- Dùng lệnh CREATE LOGIN
- Dùng wizard

Tạo tài khoản login - dùng sp_addlogin

- Cú pháp

```
sp_addlogin [ @loginame = ] 'login'
    [ , [ @passwd = ] 'password' ]
    [ , [ @defdb = ] 'database' ]
    [ , [ @deflanguage = ] 'language' ]
    [ , [ @sid = ] sid ]
    [ , [ @encryptopt = ] 'encryption_option' ]
[;]
```

- Các tham số
  - [@loginame=] 'login' là tên của login, login là sysname, không có mặc định
  - [@passwd=] 'password' là mật khẩu login, password là sysname, mặc định là NULL.

Tạo tài khoản login - dùng sp_addlogin

- Các tham số
  - [@defdb=] 'database' là CSDL mặc định của login (CSDL mà login được kết nối tới đầu tiên sau khi login vào); database là sysname, mặc định là **master**.
  - [@deflanguage=] 'language' là ngôn ngữ mặc định của login; language là sysname, mặc định là NULL. Nếu không chỉ rõ, mặc định ngôn ngữ là ngôn ngữ mặc định hiện tại của server.
  - [@sid=] 'sid' là số định danh an ninh (SID), kiểu varbinary(16), giá trị mặc định là NULL. Nếu sid là NULL, hệ thống sinh ra một SID cho tài khoản login mới. Mặc dù kiểu dữ liệu là varbinary, các giá trị khác NULL có chiều dài chính xác là 16 bytes, và giá trị đó không được có rồi. Việc xác định sid rất có ích, ví dụ khi viết các đoạn mã hoặc di chuyển tài khoản login từ server Specifying sid is useful, for example, when you are này sang server khác mà vẫn muốn cùng SID trên các server khác nhau.

Tạo tài khoản login - dùng sp_addlogin

- [@encryptopt=] 'encryption_option’ chỉ định xem mật khẩu được chuyển dưới dạng văn bản rõ ràng hay dưới dạng băm của mật khẩu văn bản rõ ràng. Lưu ý rằng không có mã hóa nào diễn ra. Từ "mã hóa" được sử dụng trong cuộc thảo luận này vì mục đích tương thích ngược. Nếu một mật khẩu văn bản rõ ràng được chuyển vào, nó sẽ được băm. Hàm băm được lưu trữ. encryption_option là varchar (20) và có thể là một trong các giá trị sau

Value	Description
NULL	The password is passed in clear. This is the default.
skip_encryption	The password is already hashed. The Database Engine should store the value without re-hashing it.
skip_encryption_old	The supplied password was hashed by an earlier version of SQL Server. The Database Engine should store the value without re-hashing it. This option is provided for upgrade purposes only.

Tạo tài khoản login - dùng sp_addlogin

- Một số thủ tục hay đi cùng sp_addlogin

Stored procedure	Description
sp_grantlogin	Adds a Windows user or group.
sp_password	Changes the password of a user.
sp_defaultdb	Changes the default database of a user.
sp_defaultlanguage	Changes the default language of a user.

Tạo tài khoản login - dùng sp_addlogin - Ví dụ

- Tạo 1 SQL Server login cho phuongnh với mật khẩu là '123456'

```
EXEC sp_addlogin 'phuongnh', '123456';
```

- Tạo 1 SQL Server login cho phuongnh với mật khẩu là '123456', có CSDL mặc định

```
EXEC sp_addlogin 'phuongnh', '123456', 'BikeStores';
```

Tạo tài khoản login - dùng CREATE LOGIN

- create login phuongnh with password = '123456'
- drop login phuongnh

Tạo tài khoản login – dùng wizard

Lệnh tạo ROLE

```
CREATE ROLE role_name
[AUTHORIZATION owner_name];
```

- The owner_name is a *database user* or *role* that owns the new role. If you omit the AUTHORIZATION clause, the *user who executes* the CREATE ROLE statement will own the new role.
- Note that the owner of the role and any member of an owning role can add or remove members of the role.
- Typically, you create a new role, grant the permissions to it using the GRANT statement, and add members to the role using the ALTER ROLE statement.

Creating a new role example

- First, create the new login called james in the master database:

```
CREATE LOGIN james
WITH PASSWORD = 'Ux!sa123ayb';
```

- Next, create a new user for the login james:

```
CREATE USER james
FOR LOGIN james;
```

- Then, create a new role called sales:

```
CREATE ROLE sales;
```

Creating a new role example

- After that, grant the SELECT, INSERT, DELETE, and UPDATE privileges on the sales schema to the sales role:

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON SCHEMA::sales
TO sales;
```

- Finally, add the user james to the sales role:

```
ALTER ROLE sales
ADD MEMBER james;
```

Creating a new role owned by a fixed database role example

- The following example uses the CREATE ROLE statement to create a new role owned by the db_securityadmin fixed database role:

```
CREATE ROLE sox_auditors
AUTHORIZATION db_securityadmin;
```

Examining the roles

- The roles and their members are visible in the sys.database_principals and sys.database_role_members views.
- The following shows the information on the sales and sox_auditors roles:

```
SELECT name, principal_id, type, type_desc, owning_principal_id
FROM sys.database_principals
WHERE name in ('sales', 'sox_auditors');
```

Lệnh GRANT

```
GRANT { ALL [ PRIVILEGES ] }
      | permission [ ( column [ ,...n ] ) ] [ ,...n ]
      [ ON [ class :: ] securable ] TO principal [ ,...n ]
      [ WITH GRANT OPTION ] [ AS principal ]
```

```
-- Execute the following as a database owner
GRANT EXECUTE ON TestProc TO TesterRole WITH GRANT OPTION;
EXEC sp_addrolemember TesterRole, User1;
-- Execute the following as User1
-- The following fails because User1 does not have the permission as
GRANT EXECUTE ON TestMe TO User2;
-- The following succeeds because User1 invokes the TesterRole member
GRANT EXECUTE ON TestMe TO User2 AS TesterRole;
```

Lệnh GRANT

```
USE AdventureWorks;
GRANT CREATE TABLE TO MelanieK;
GO

USE AdventureWorks2012;
GRANT SHOWPLAN TO AuditMonitor;
GO

USE AdventureWorks2012;
GRANT CREATE VIEW TO CarmineEs WITH GRANT OPTION;
GO

USE AdventureWorks2012;
GRANT CONTROL ON DATABASE::AdventureWorks2012 TO Sarah;
GO
```

Lệnh GRANT

```
GRANT SELECT ON Person.Address TO RosaQdM;
GO

--permission on stored procedure
USE AdventureWorks2012;
GRANT EXECUTE ON OBJECT::HumanResources.uspUpdateEmployeeHireInfo
    TO Recruiting11;
GO

CREATE ROLE newrole ;
GRANT EXECUTE ON dbo.uspGetBillOfMaterials TO newrole ;
GO

GRANT SELECT ON SCHEMA :: Person TO WilJo WITH GRANT OPTION;
```

Lệnh REVOKE

```
REVOKE [ GRANT OPTION FOR ]
       {
         [ ALL [ PRIVILEGES ] ]
         |
                permission [ ( column [ ,...n ] ) ] [ ,...n ]
       }
       [ ON [ class :: ] securable ]
       { TO | FROM } principal [ ,...n ]
       [ CASCADE] [ AS principal ]
```

Lệnh REVOKE

```
CREATE SCHEMA Sales;
GO
CREATE USER Joe without login;
GO
CREATE ROLE Vendors;
GO
ALTER ROLE Vendors ADD MEMBER Joe;
GO
GRANT SELECT ON SCHEMA :: Sales TO Vendors;
GO
REVOKE SELECT ON SCHEMA :: Sales TO Vendors;
GO
```