Principales

Los Principales pueden representar a un usuario específico, un rol que aglutine varios usuarios o una aplicación.

SQL Server los divide en tres categorías:

- **Windows**: cuentas de usuario o grupos autenticados por Windows
- **SQL Server**: Login's de servidor o grupos autenticados por SQL Server
- **Base de Datos**: usuarios de BD's y roles de BD's y de aplicación

Principales - Windows

```
CREATE LOGIN [dominio_instancia\usuario]
    FROM WINDOWS
    WITH DEFAULT_DATABASE = AdventureWorks;
```

Principales – SQL Server

SQL Server Logins

Se usa cuando no es posible la autenticación Windows (accesos externos)

Roles de Servidor

Son principales predefinidos a nivel de servidor

Los Logins de Windows y de SqlServer pueden ser añadidos como miembros de un rol de servidor

Principales – SQL Server

Logins desde certificados y claves asimétricas

Cuando la conexión se hace con un login de este tipo (por ejemplo vía HTTPS o un Service Broker Endpoint), ésta se hallará dentro del contexto de este login.

Un login creado de esta forma, no puede ser usado para conectar a SQL Server por la vía normal a través de ADO.NET o Management Studio.

Políticas de Windows Server 2003 y 2008

SQL Server puede usar las políticas de seguridad de Windows para la autenticación.

El administrador del sistema puede determinar que la seguridad de SQL server siga los mismos estándares que la de Windows.

No funciona con Windows 2000, XP, etc., incluso si la máquina está en un dominio Windows Server 2003.

Principales - Base de Datos

Son los objetos que representan a los usuarios a los que podemos asignar permisos para acceder a la Base de Datos o a los objetos contenidos dentro de ellas.

Operan en el nivel de Base de Datos .

Se reconocen tres tipos:

- Usuarios de Base de Datos
- Roles de Base de Datos
- Roles de aplicación

Principales - Base de Datos

Usuarios de Base de Datos

No se debe usar el procedimiento ***sp_adduser*** (obsoleto).

Se puede asociar a:

- Un login
- Un certificado
- Una clave asimétrica

Ejemplo - 1:

```
CREATE USER usuario1
FOR LOGIN [iniciosesion];
GO
```

Principales - Base de Datos

Ejemplo - 2:

```
CREATE USER usuario2
FOR CERTIFICATE certificado;
GO
```

En el ejemplo 2, si un usuario o una aplicación conecta a un servicio Web, la consulta será ejecutada en el contexto de seguridad del usuario asociado con la clave o certificado.

Roles Base de Datos

sp_addrole está obsoleto

```
CREATE ROLE [mirol]
AUTHORIZATION dbo;
```

Roles de aplicación

Permiten definir un contexto de seguridad para una aplicación específica.
sp_addapprole está obsoleto.

Principales - Propiedad

Propietario de objeto

Es importante comprender dos utilidades clave relacionadas con este concepto.

La cláusula AUTHORIZATION en los mandatos CREATE. Se asigna un propietario para el objeto creado. ALTER AUTHORIZATION permite el cambio de propietario.

La segunda es que podemos en lugar de suministrar el nombre de un principal como propietario, especificar SCHEMA OWNER, que asignará la propiedad del objeto al propietario del esquema al que el objeto pertenece.

```
ALTER AUTHORIZATION ON OBJECT::Sales.SalesOrderHeader
TO SCHEMA OWNER;
```

(2.2.1 Seguridad (Transferencia de Propiedad))

Principales - Suplantación

Suplantación (Impersonation)

Un usuario puede suplantar a otro en un bloque específico de código.

SETUSER no se puede usar con usuarios asociados a login's Windows

Se consigue con la cláusula EXECUTE AS que se añade a los mandatos CREATE/ALTER de los procedimientos, funciones, triggers y colas de Service Broker.

A diferencia de SETUSER, este mandato no está restringido a los miembros del rol de servidor sysadmin o al rol de base de datos db_owner.

Esto es importante, ya que la cuenta de suplantación no necesita tener acceso a los objetos referenciados en el módulo (procedimiento, función, etc.).

Principales - Suplantación

Ejemplo

```
USE AdventureWorks
GO

CREATE LOGIN analog WITH PASSWORD = ‘contrasena1';
CREATE LOGIN jorgelog WITH PASSWORD = ‘contrasena2';
CREATE USER ana FOR LOGIN analog;
CREATE USER jorge FOR LOGIN jorgelog ;
GO

EXEC sp_addrolemember 'db_datareader', ‘ana ';
EXEC sp_addrolemember 'db_ddladmin', ‘ana ';
GO
```

Principales - Suplantación

Creamos un procedimiento almacenado:

```
CREATE PROCEDURE prog_CogerDirecciones
WITH EXECUTE AS 'ana '
AS
        SELECT * FROM Person.Address;
GO
```

Cambiamos la propiedad del procedimiento almacenado:

```
ALTER AUTHORIZATION ON prog_CogerDirecciones
TO ana ;
```

Principales - Suplantación

Concedemos a ***Jorge*** el permiso de ejecución del procedimiento

```
SETUSER
GRANT EXECUTE ON prog_CogerDireccionesTO jorge ;
```

Probamos este código

```
SETUSER  ‘jorge '
SELECT * FROM Person.Address;
```

Vemos que no se puede leer la tabla.

2.2.2 Seguridad(Impersonate.sql)