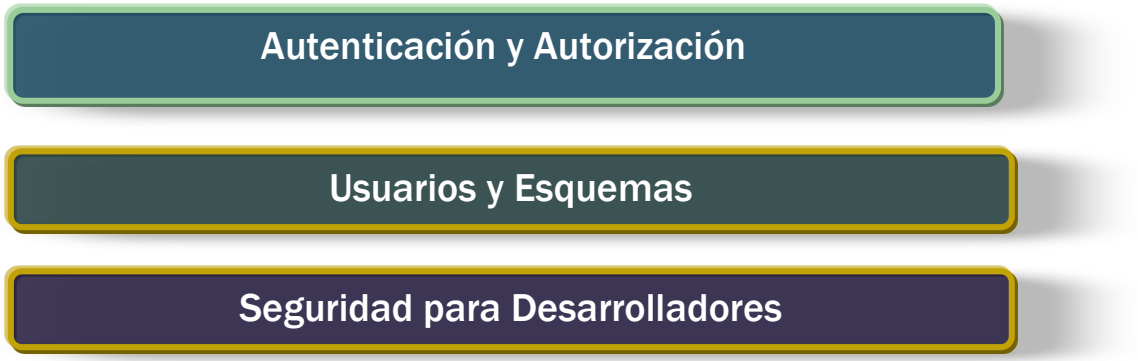
	UNIVERSIDAD DON BOSCO **FACULTAD DE ESTUDIOS TECNOLÓGICOS** **ESCUELA DE COMPUTACIÓN**
CICLO 1-2016	**Clase N°11**
	Tema: Diseño de seguridad de una Base de datos **Materia:** Base de datos **Docentes:** Blanca Iris Cañas

La protección de la información (controlar el acceso a los datos de una organización) se parece mucho a la protección de una estructura física. Por ejemplo, imagine que tiene su propio negocio y el edificio que lo alberga también es de su propiedad no querrá que el público en general pueda acceder al edificio; solo deberían tener acceso los empleados. Sin embargo, también necesita restricciones para las zonas a las que los empleados pueden acceder, porque solo los contables deberían tener acceso al departamento de contabilidad y casi nadie debería tener acceso a su despacho; debe instalar diversos sistemas de seguridad. La protección de SQL Server (su "edificio") se basa en este concepto; nadie puede entrar a menos que se le conceda acceso y, una vez que los usuarios están dentro, los diferentes sistemas de seguridad mantienen las aéreas confidenciales a salvo de miradas indiscretas.



SQL Server define 4 conceptos básicos:

1. Login de SQL Server
2. Usuario de la Base de datos
3. Role de la BD
4. Role de una aplicación

Inicios de sesión (Login): Un login es la habilidad de utilizar una instancia del Servidor SQL, está asociado con un usuario de Windows o con un usuario de SQL. Son autenticados contra SQL Server por lo tanto son los accesos al servidor, pero esto no quiere decir que puedan acceder a las bases de datos o a otros objetos. Para poder acceder a cada una de las bases de datos se necesita de un usuario (user).

Usuario de la base de datos (User):

El usuario de la base de datos es la identidad del inicio de sesión cuando está conectado a una base de datos. El usuario de la base de datos puede utilizar el mismo nombre que el inicio de sesión, pero no es necesario.

- Los Logins son asignados a los usuarios
- Los grants se les asignan a los usuarios.
- A los usuarios se le asignan sus propios Esquemas(schemas)

Usuarios por defecto en una BD

dbo: Propietario. No puede ser borrado de la BD

Guest: Permite a usuarios que no tienen cuenta en la BD, que accedan a ella, pero hay que hacerle permiso explícitamente

Information_schema
Permite ver los metadatos de SQL Server

sys
Permite consultar las tablas y vistas del sistema, procedimientos extendidos y otros objetos del catálogo del sistema

Mostrar usuarios de una base de datos:

```
USE master
GO
SELECT * FROM sys.database_principals
```

Los usuarios pueden pertenecer a Roles.

- Todos los usuarios son miembros del Role “Public”
- El login “sa” está asignado al usuario dbo en todas las base de datos.

Da acceso a la base de datos, pero esto tampoco quiere decir que pueda hacer cualquier operación sobre la base de datos, en principio no puede hacer casi nada, salvo que se le vaya asignando roles y otros privilegios para hacerle permisos de acceso a los objetos de esa base de datos.

Roles:

Los Roles pueden existir a nivel de instancia o base de datos.

A nivel de Instancia:

- Los logins pueden ser otorgados roles llamados “server roles”.
- No se pueden crear Roles nuevos

A nivel de Base de Datos

- Los usuarios de base de datos pueden ser otorgados roles.
- Se pueden crear roles nuevos.

Role de una Aplicación

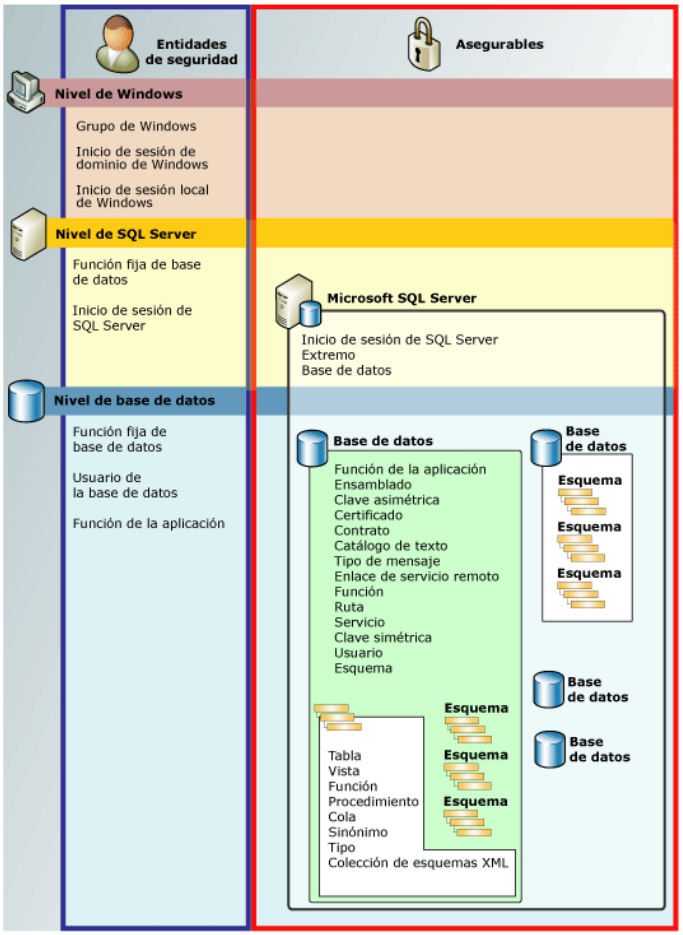
Un role de aplicación sirve para asignarle permisos a una aplicación:

- Tiene un password
- No contiene usuarios

Jerarquía de permisos

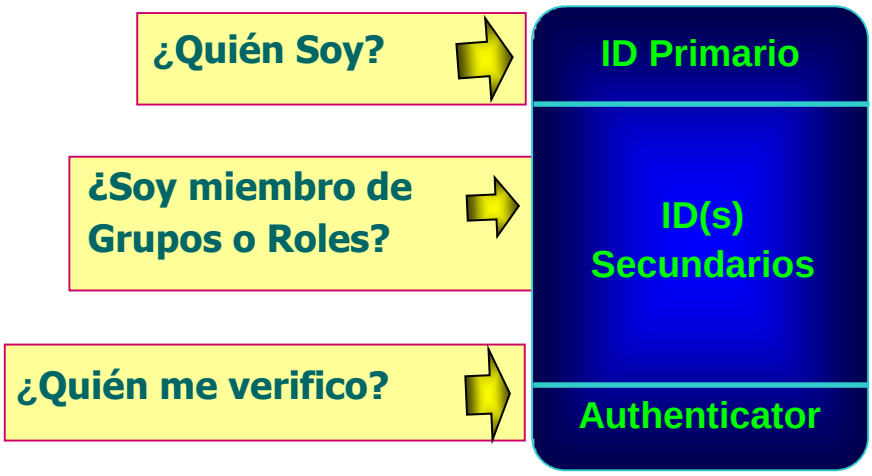
El Motor de base de datos administra un conjunto jerárquico de entidades que se pueden proteger mediante permisos. Estas entidades se conocen como elementos protegibles. Los protegibles más prominentes son los servidores y las bases de datos, pero los permisos discretos se pueden establecer en un nivel mucho más específico.

En la siguiente figura se muestra las relaciones entre las jerarquías de permisos del Motor de base de datos



Autenticación

¿Qué es Autenticación?
Es básicamente el proceso de determinar que alguien es realmente quien dice ser.



En SQL Server nos encontramos con tres niveles o capas en los cuales podemos gestionar la seguridad. El primero de ellos se encuentra a **nivel de servidor**, en él podemos gestionar quién tiene acceso al servidor y quién no, y además gestionamos que roles va a desempeñar. Para que alguien pueda acceder al servidor **debe tener un inicio de sesión (login) asignado**, y a éste se asigna los roles o funciones que puede realizar sobre el servidor.

El que alguien tenga acceso al servidor no quiere decir que pueda acceder a las bases de datos que se encuentran en él. Para ello hay que tener acceso a la siguiente barrera de seguridad, que es a **nivel de base de datos**. Para que un login tenga acceso a una base de datos, tenemos que crear en ella un **usuario (user)**. Se debe crear un usuario en cada una de las bases de datos a las que queramos que acceda un login.

Análogamente, el que un usuario tenga acceso a una base de datos no quiere decir que tenga acceso a todo su contenido, ni a cada uno de los objetos que la componen. Para que esto ocurra tendremos que irle **concediendo o denegando permisos sobre cada uno de los objetos que la componen**.

A continuación se puede observar un gráfico que refleja este modelo.

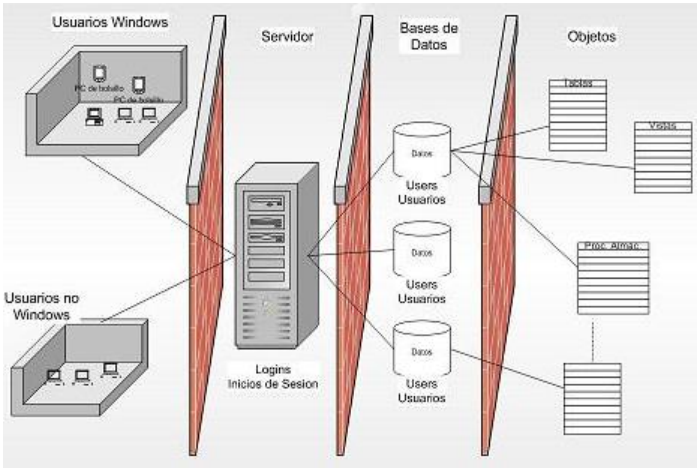


Figura tomada de: http://www.sqlserversi.com/2009/01/seguridad-en-sql-server.html

Mejoras en la Autenticación sobre SQL2000

- La información de Login no se envía como texto plano a través de la red.
- La nueva política para los passwords esta activada por default. Esta política incluye:
  - Complejidad de passwords. No más password como: “sa”, “password”, “Admin”, “Administrator”, “sysadmin” o “ ”
  - Maneja el bloqueo de las cuentas en base a intentos fallidos
  - Expiración de passwords

Usuarios de BD y esquemas

Colección de objetos de la BBDD cuyo propietario es un único principal y forma un único espacio de nombres (conjunto de objetos que no pueden tener nombres duplicados)

servidor.basededatos.esquema.objeto

Los objetos ahora pertenecen al esquema de forma independiente al usuario
Beneficios:

- El borrado de un usuario no requiere que tengamos que renombrar los objetos
- Resolución de nombres uniforme

- Gestión de permisos a nivel de esquema

Una BBDD puede contener múltiples esquemas
Cada esquema tiene un propietario (principal): usuario o rol
Cada usuario tiene un default schema para resolución de nombres
La mayoría de los objetos de la BBDD residen en esquemas
Creación de objetos dentro de un esquema requiere permisos
CREATE y ALTER o CONTROL sobre el esquema

CREATE SHEMA

Crea un esquema en la base de datos actual. La transacción CREATE SCHEMA también puede crear tablas y vistas dentro del nuevo esquema, así como establecer la autorización, denegación o revocación (GRANT, DENY o REVOKE) de permisos en esos objetos.

Sintaxis:

CREATE SHEMA Nombre_Esquema

```
CREATE SCHEMA RecursosHumanos
```

Creando una tabla bajo el esquema RecursosHumanos:

```
CREATE TABLE RecursosHumanos.Empleado
      ( codigo int primary key,
        nombre varchar(25),
        apellido varchar(25)
      )
```

Creación de Inicio de Sesión y usuarios de Base de datos

Se necesita tener 2 Usuarios de BD y 2 Inicio de sesión para la Base de datos Northwind

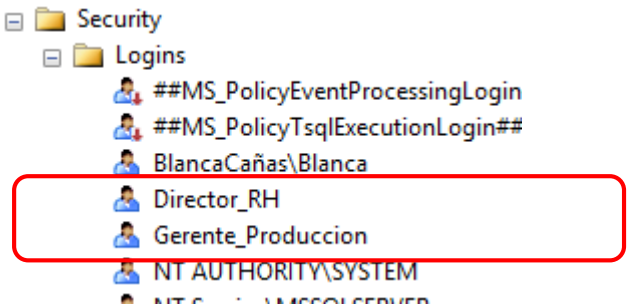
Sintaxis:

CREATE LOGIN nombre_login
WITH PASSWORD = 'clave_usuario';

Creando de los inicios de sesión:

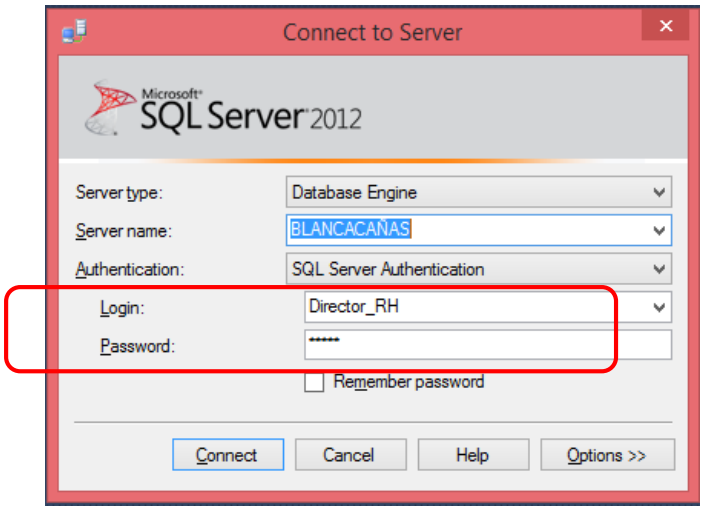
```
CREATE LOGIN Director_RH
WITH PASSWORD = '12345';

CREATE LOGIN Gerente_Produccion
WITH PASSWORD = 'abcd';
```

Los inicios de sesión se encuentran en la carpeta Login de la carpeta Security, la cual se encuentra en el nodo del nombre del servidor

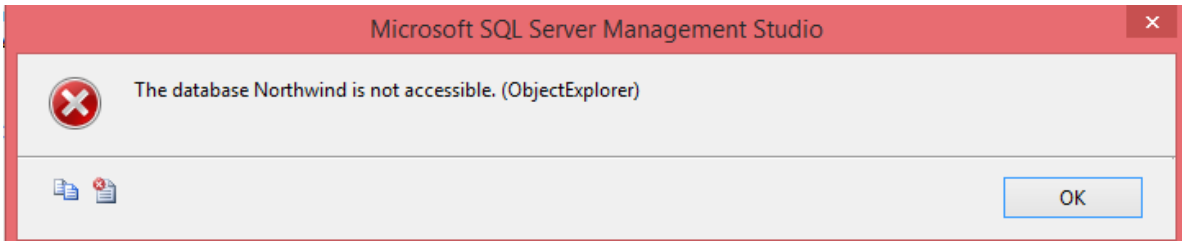


Comprobando los inicios de sesión:

Entrar a SQL Server 2012 con el nuevo inicio de sesión creado anteriormente



Hacer doble clic sobre la base de datos Northwind y obtendremos la siguiente ventana:



En el cual se indica que no se puede acceder a la base de datos, con la cual queremos trabajar.

Para poder tener acceso a las bases de datos primero debemos crear los USUARIOS de BASES DE DATOS, para la creación de usuarios de las bases de datos nos ayuhaceremos del siguiente código.

Sintaxis:

USE BASE_DE_DATOS;

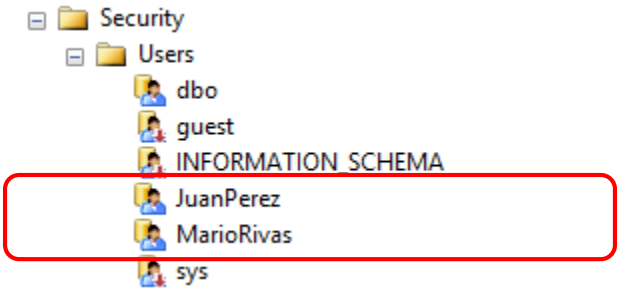
CREATE USER nombre_usuario **FOR LOGIN** nombre_login
WITH DEFAULT_SCHEMA = algun_esquema;

Ejemplo de creación de usuarios de base de datos:

```
USE Northwind
```

```
CREATE USER JuanPerez FOR LOGIN Director_RH
WITH DEFAULT_SCHEMA = RecursosHumanos

CREATE USER MarioRivas FOR LOGIN Gerente_Produccion
WITH DEFAULT_SCHEMA = Produccion
```

Los usuarios de base de datos se encuentran en la carpeta Users de la carpeta Security a nivel del nombre de la base de datos



Asignando permisos sobre esquemas

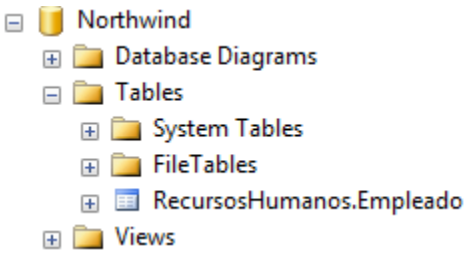
Vamos a asignarle un esquema al usuario, por ejemplo el esquema RecursosHumanos.

```
GRANT SELECT
ON SCHEMA :: RecursosHumanos
TO JuanPerez
WITH GRANT OPTION
GO
```

GRANT: La instrucción GRANT se utiliza para conceder determinados permisos genéricos o bien permisos sobre objetos a usuarios de la base de datos

GRANT OPTION: permite que el usuario al que le han concedido permisos pueda a su vez concederlos a otros usuarios.

Si nos conectamos con el inicio de sesión Director_RH, observamos las siguientes tablas en la base de datos Northwind:



Si hacemos un SELECT a la tabla Empleado, podemos obtener la información almacenada en esa tabla

```
SELECT * FROM RecursosHumanos.Empleado
```

100 %

Results | Messages

	codigo	nombre	apellido
1	1	Maria Luisa	Martinez
2	2	Roberto Carlos	Carranza
3	3	Iñaki	Aguirre

Pero si hacemos un INSERT a la tabla, nos mostrara un mensaje de error indicando que no se tiene permisos de agregar datos a la tabla.

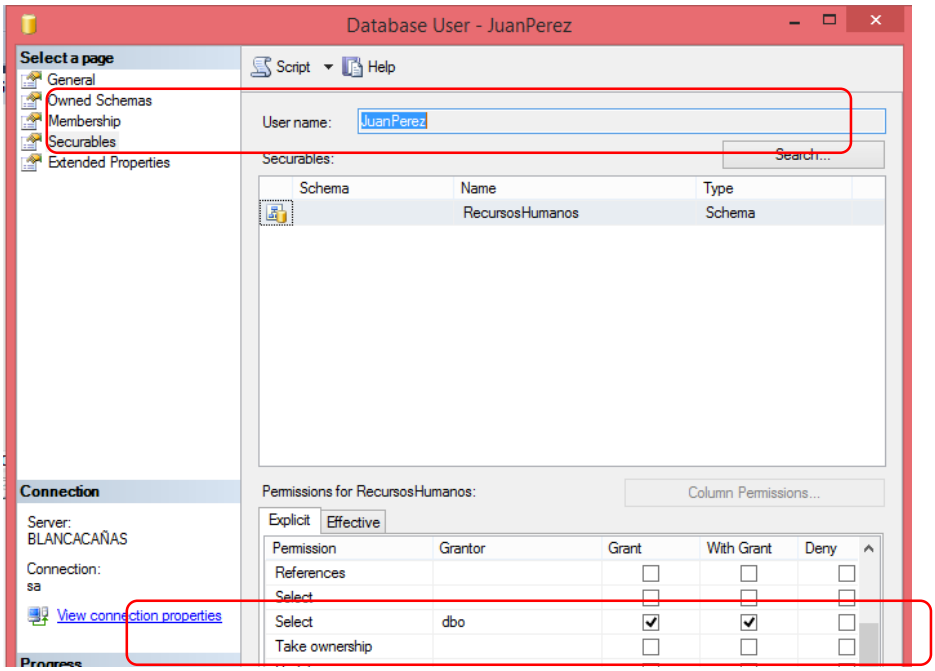
```
INSERT INTO RecursosHumanos.Empleado VALUES (4, 'Marisol', 'Abarca')
```

00 %

Messages

```
Msg 229, Level 14, State 5, Line 1
The INSERT permission was denied on the object 'Empleado', database 'Northwind', schema 'RecursosHumanos'.
```

En conclusión: Como se puede apreciar el usuario efectivamente puede realizar un SELECT, pero no podrá realizar un DELETE o un INSERT o un UPDATE ya que por defecto cuando se crea un usuario de base de datos el unico permiso asignado es la instruccion SELECT

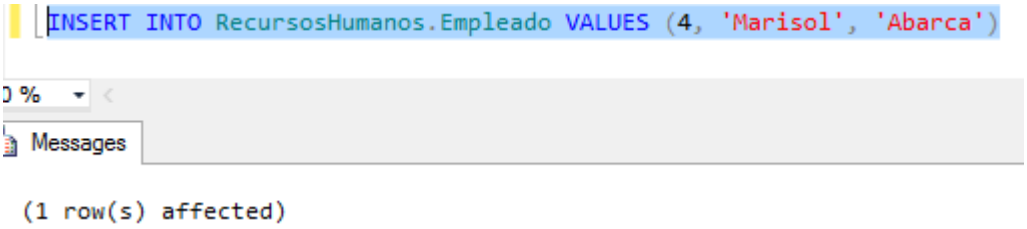


Asignando permisos sobre tablas

Asignar permisos tabla por tabla

```
GRANT INSERT ON OBJECT::RecursosHumanos.Empleado
TO JuanPerez
```

Ahora con el inicio de sesion Director_RH, ejecutar de nuevo la consulta



Y ahora si se podra agregar el nuevo registro a la tabla

Quitar permisos sobre tablas

```
REVOKE INSERT ON OBJECT::RecursosHumanos.Empleado
TO JuanPerez
```

Quitar los permisos sobre un esquema

Esto sería como deshacer la otorgación del permiso, para esto se utiliza la palabra REVOKE que significa revocar o quitar.

```
REVOKE SELECT
ON SCHEMA :: RecursosHumanos
TO JuanPerez CASCADE
```

Ahora ya no se tiene ningun permiso para los objetos almacenados bajo el esquema RecursosHumanos