

Laboratorio de seguridad

Tabla de contenidos

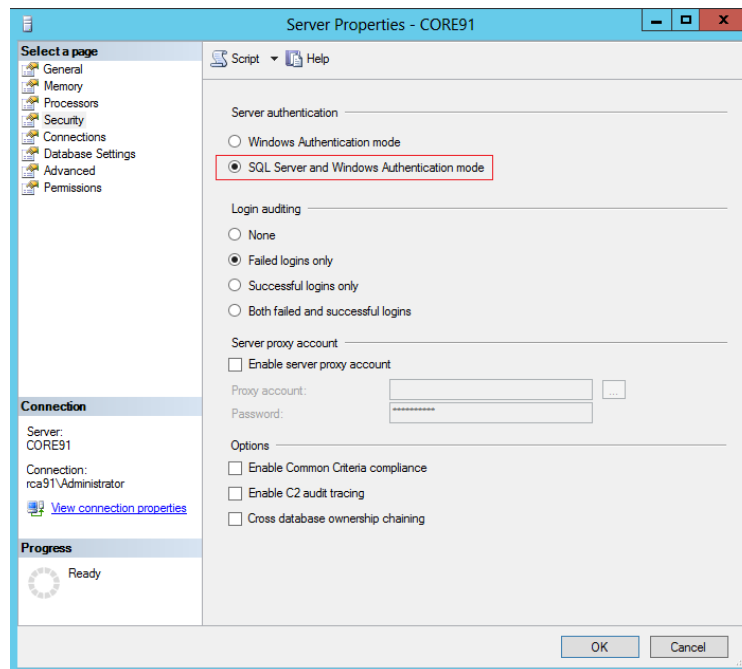
Configure las propiedades de Security del SQL Server, para permitir los posibles métodos de autenticación	2
Identifique las diferencias existentes entre cada tipo de autenticación y la política de complejidad y expiración de contraseñas.	3
Crear cuentas en forma correcta	5
<i>Autenticación SQL Server</i>	5
<i>Autenticación de dominio de Windows</i>	13
Excepciones	22
Genera el reporte detallado e interpreta los resultados.....	25
<i>Roles a nivel de servidor por usuario</i>	25
<i>Roles a nivel de base de datos por usuario</i>	25
<i>Permisos de usuario a nivel de objetos de una base de datos</i>	26
Conclusiones	28
Recomendaciones.....	29
Bibliografía	30

CONFIGURE LAS PROPIEDADES DE SECURITY DEL SQL SERVER, PARA PERMITIR LOS POSIBLES MÉTODOS DE AUTENTICACIÓN

SQL Server ofrece dos métodos de autenticación:

- Windows Authentication mode; utiliza cuentas del dominio de directorio activo (Active Directory).
- SQL Authentication mode; permite la configuración de cuentas directamente administradas por el SQL Server, independientemente de su existencia o no en el dominio activo. La configuración de las cuentas, las contraseñas y sus permisos, recaen directamente sobre el administrador de la base de datos.

Para seleccionar entre ellas, se utiliza el tab de preferencias del servidor. Se puede escoger entre utiliza exclusivamente autenticación basada en windows o habilitar también la autenticación de SQL Server, cómo puede verse en la siguiente interfaz.



¿Cómo habilitar la cuenta sa?

Durante la configuración estándar de las ediciones más recientes de SQL server esta cuenta se encuentra deshabilitado. Los siguientes comandos de TSQL permiten activarlas y definir una contraseña.

```
ALTER LOGIN sa ENABLE;
ALTER LOGIN sa WITH PASSWORD = 'Pa$$word';
```

IDENTIFIQUE LAS DIFERENCIAS EXISTENTES ENTRE CADA TIPO DE AUTENTICACIÓN Y LA POLÍTICA DE COMPLEJIDAD Y EXPIRACIÓN DE CONTRASEÑAS.

Autenticación SQL Server	Autenticación Windows
El usuario debe usar un usuario/contraseña para entrar al servidor, y luego un usuario / contraseña distinto para conectarse a SQL Server	El usuario puede conectarse a SQL Server usando las credenciales del usuario de la sesión activa de Windows en el momento en que se establece la conexión
Intercambio de handshake y challenge menos seguro, al no poder utilizar el protocolo de seguridad Kerberos.	Utiliza el protocolo de seguridad Kerberos para establecer la autenticación.
No ofrece todas las políticas de seguridad que ofrece los Dominios de Windows, pero usa la configuración local del servidor para manejar aspectos como la expiración de contraseñas y así como las políticas de contraseñas disponibles desde Windows Server 2003.	Permite definir políticas de complejidad de contraseñas y expiración de contraseñas que pueden ser aplicadas a uno o más usuarios del dominio, de forma consistente.
La contraseña encriptada de SQL Server debe ser transmitida por la red al realizar la conexión. Algunas aplicaciones que se conectan automáticamente ofrecen la posibilidad de guardar la contraseña, lo cual crea el riesgo local de que dicha contraseña se almacena de forma no encriptada. Ejemplo, conexiones de Excel 2007 donde el usuario almacena la contraseña en el odt. La contraseña almacena no es encriptada.	SQL Server no pide la contraseña. La validación de la identidad ocurre a través de tokens y la garantía de que la sesión fue previamente autorizada por el servidor de dominio.
Permite soportar aplicaciones heredadas (legacy), por ejemplo aquellas aplicaciones que solamente aceptan autenticación de SQL Server.	No se puede utilizar con algunas aplicaciones heredadas.
Soporta ambientes con sistemas operativos mixtos, cuando no todos los usuarios están autenticados por el dominio de Windows.	Solo funciona con Windows

Autenticación SQL Server	Autenticación Windows
Permite que los usuarios se conecten desde dominios desconocidos o no certificados.	Solo se pueden conectar desde dominios de confianza.
Permite el soporte a aplicaciones web donde los usuarios pueden crear sus propias identidades.	Los usuarios no pueden crear sus propias identidades, hasta deben ser creadas previamente en los servidores de dominio.

CREAR CUENTAS EN FORMA CORRECTA

Autenticación SQL Server

T-SQL

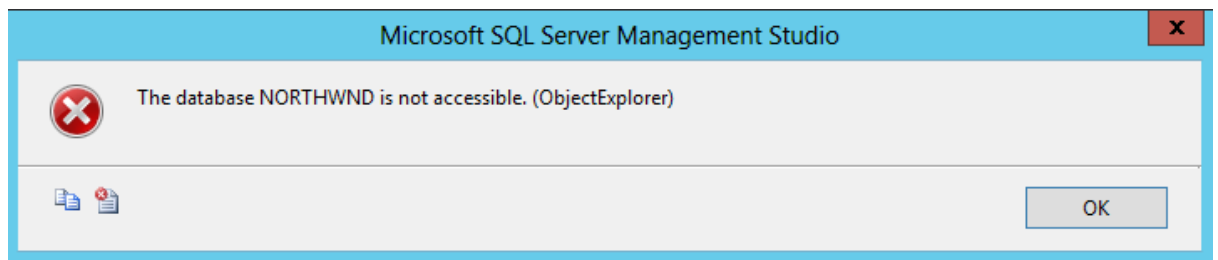
1. Crear un LOGIN con autenticación SQL

```
CREATE LOGIN appUser  
WITH PASSWORD = 'Pa$$word'
```

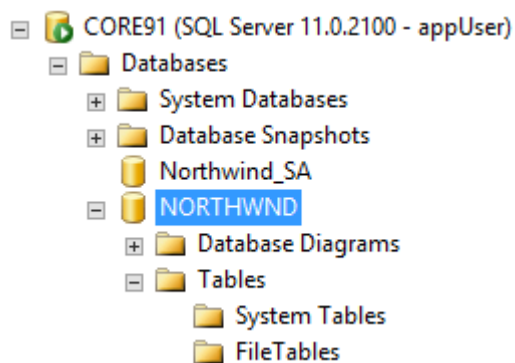
2. Asignar rol de servidor

```
ALTER SERVER ROLE diskadmin ADD MEMBER appUser;
```

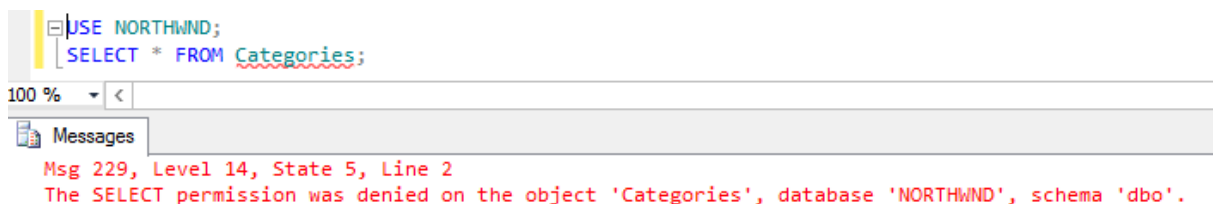
3. Crear un USER para la base de datos NORTHWND



```
USE NORTHWND;  
CREATE USER appUser FROM LOGIN appUser;
```



4. Asignar rol de base de datos



```
USE NORTHWND;
ALTER ROLE db_datareader ADD MEMBER appUser;
```

USE NORTHWND;
SELECT * FROM Categories;

100 %

Results Messages

	CategoryID	CategoryName	Description	Picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	0x151C2F000200000...
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F000200000...
3	3	Confections	Desserts, candies, and sweet breads	0x151C2F000200000...
4	4	Dairy Products	Cheeses	0x151C2F000200000...
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F000200000...
6	6	Meat/Poultry	Prepared meats	0x151C2F000200000...
7	7	Produce	Dried fruit and bean curd	0x151C2F000200000...
8	8	Seafood	Seaweed and fish	0x151C2F000200000...

5. Asignar privilegios

USE NORTHWND;
INSERT INTO Categories VALUES ('Prueba', 'Categoria de prueba', NULL);

100 %

Messages

Msg 229, Level 14, State 5, Line 2
The INSERT permission was denied on the object 'Categories', database 'NORTHWND', schema 'dbo'.

```
USE NORTHWND;
GRANT INSERT ON dbo.Categories TO appUser;
```

USE NORTHWND;

INSERT INTO Categories VALUES ('Prueba','Categoria de prueba',NULL);

SELECT * FROM Categories;

100 %

<

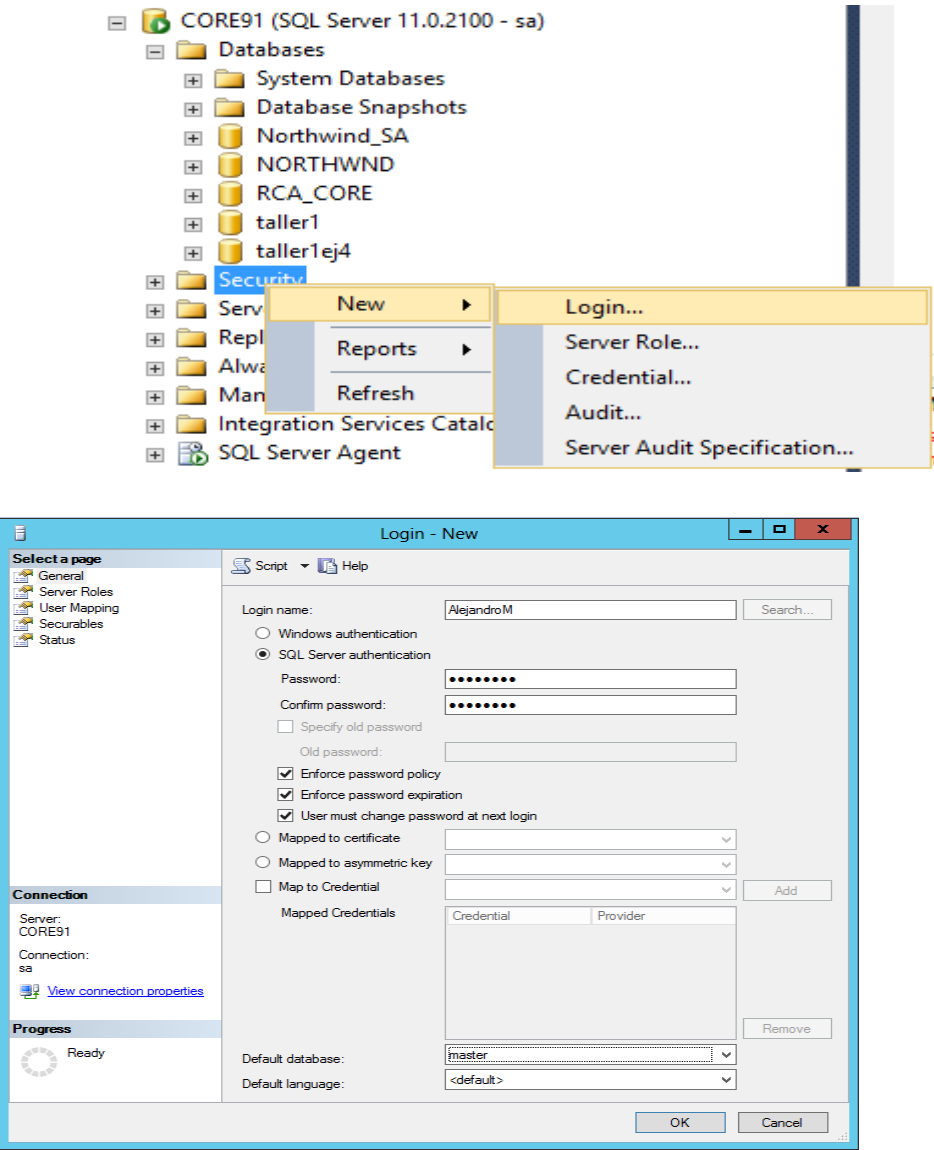
Results

Messages

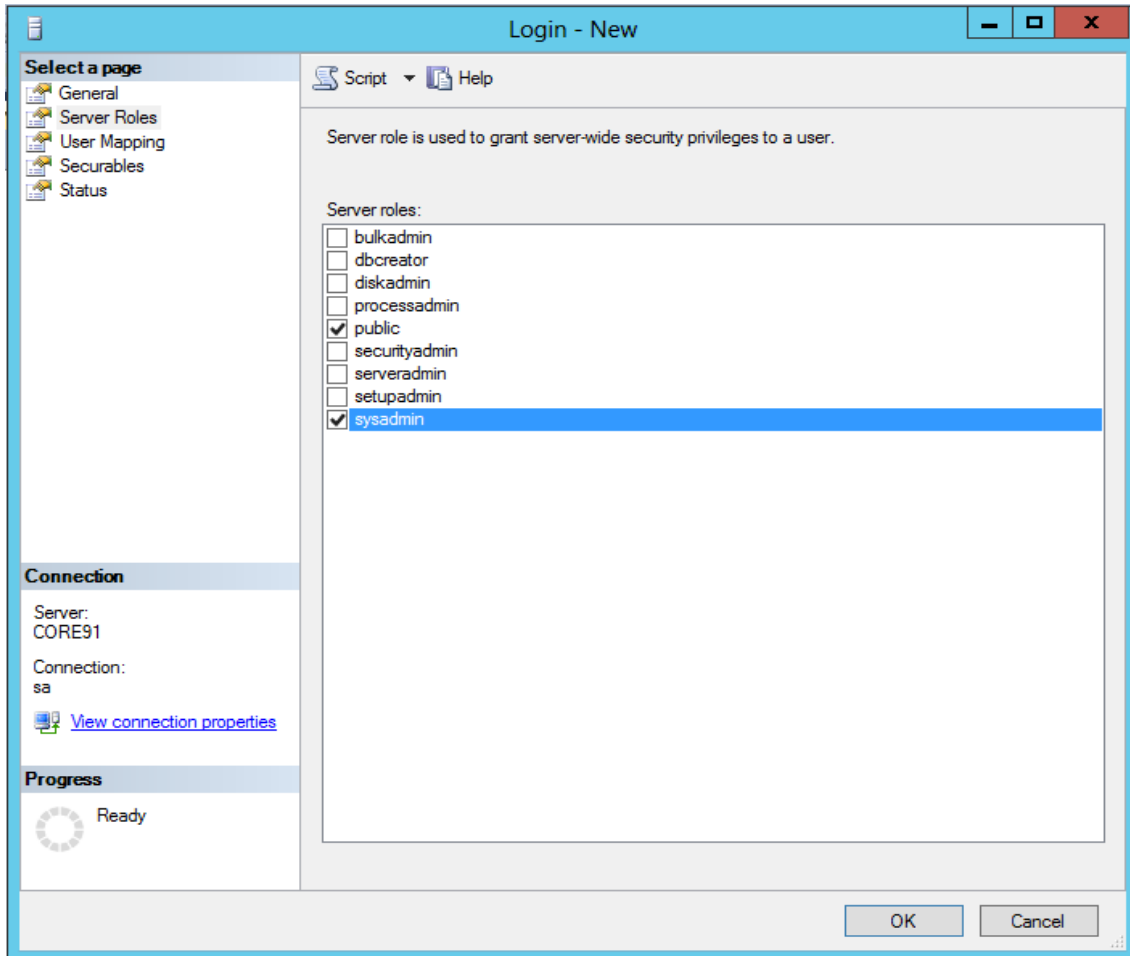
	CategoryID	CategoryName	Description	Picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	0x151C2F0002000000...
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F0002000000...
3	3	Confections	Desserts, candies, and sweet breads	0x151C2F0002000000...
4	4	Dairy Products	Cheeses	0x151C2F0002000000...
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F0002000000...
6	6	Meat/Poultry	Prepared meats	0x151C2F0002000000...
7	7	Produce	Dried fruit and bean curd	0x151C2F0002000000...
8	8	Seafood	Seaweed and fish	0x151C2F0002000000...
9	9	Prueba	Categoria de prueba	NULL

Utilizando GUI

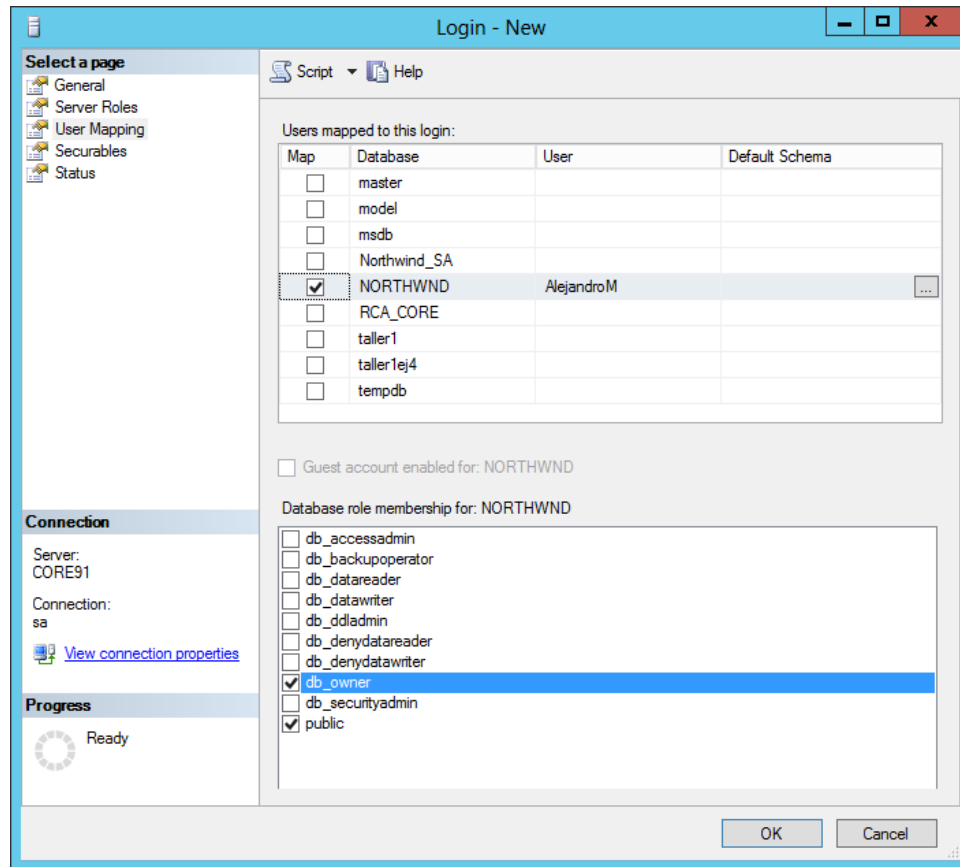
1. Crear un LOGIN con autenticación SQL



2. Asignar el rol fijo de servidor pertinente a la cuenta de login creada.



3. Crear un USER para la base de datos NORTHWND y asignar rol de base de datos db_owner.



4. Asignar privilegios

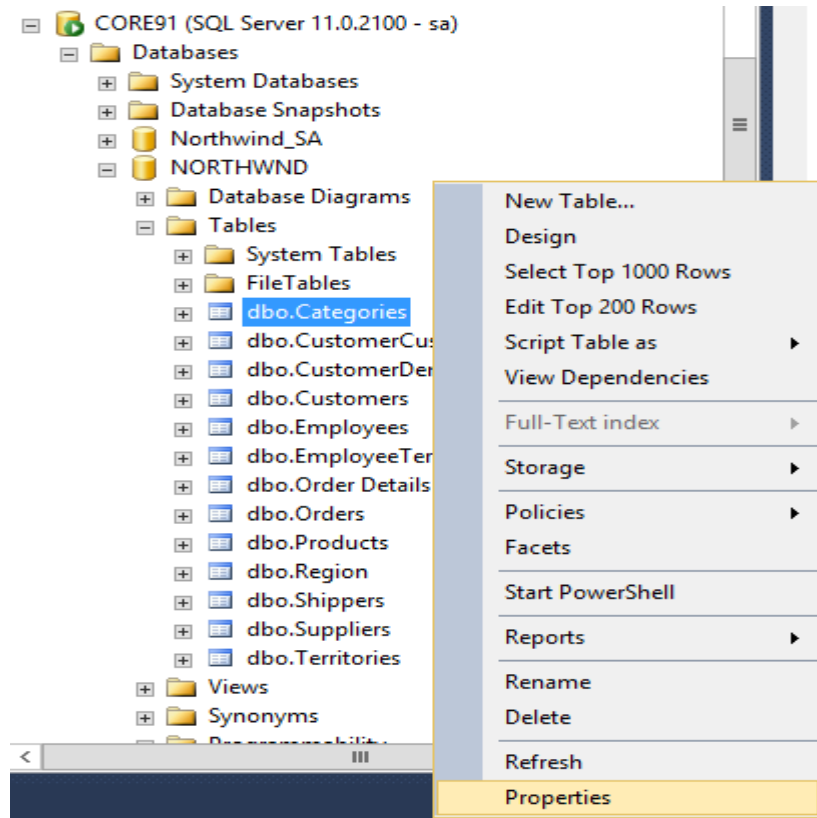


Table Properties - Categories

Script Help

Schema:

[View schema permissions](#)

Table name:

Users or roles:

	Name	Type
	appUser	User
	rca91\AnaC	User

Connection

Server: CORE91

Connection: sa

[View connection properties](#)

Progress

Ready

Permissions for appUser:

Explicit Effective

Permission	Grantor	Grant	With Grant	Deny
Alter		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Control		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delete		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Insert		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Insert	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
References		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Select		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

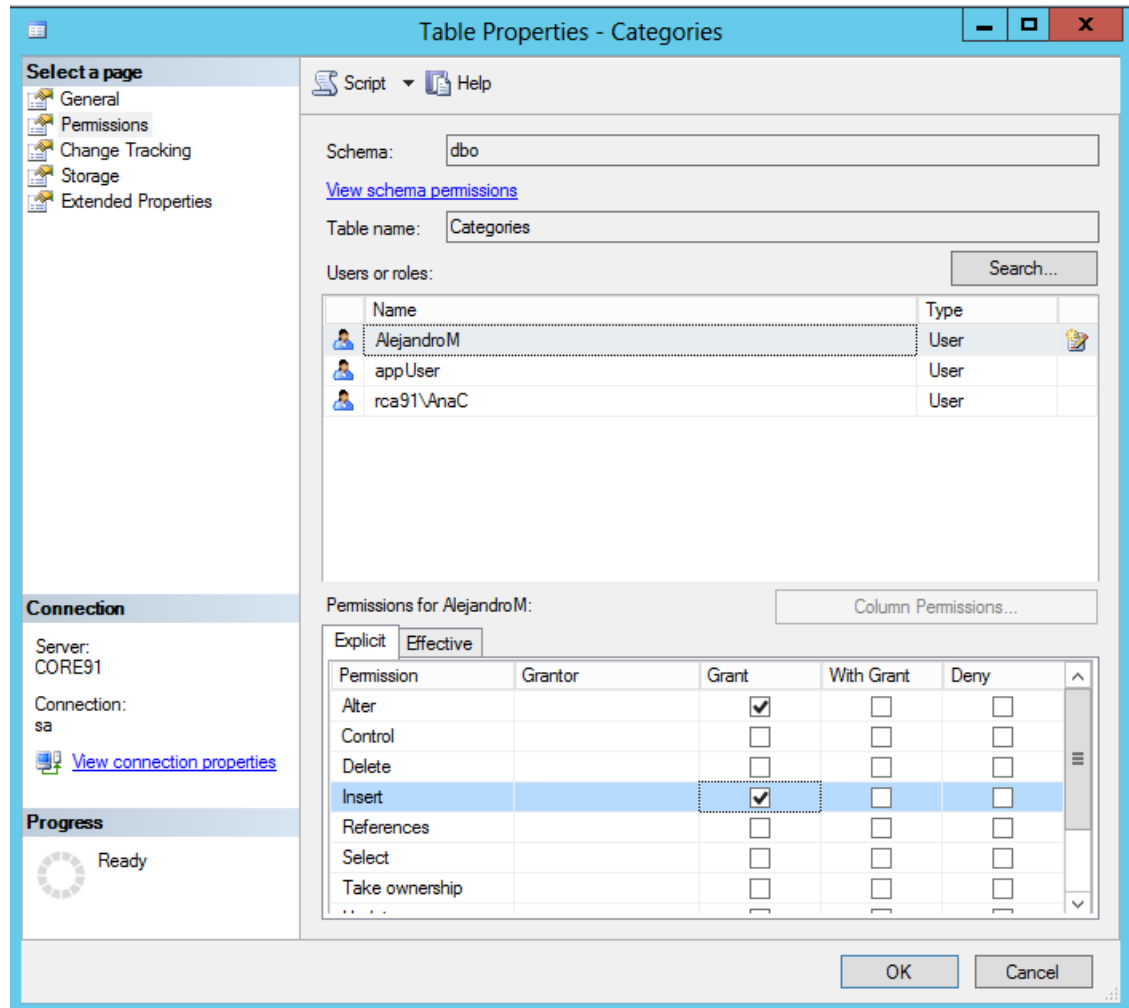
OK Cancel

Select Users or Roles

Select these object types:

Enter the object names to select ([examples](#)):

OK Cancel Help



Autenticación de dominio de Windows

T-SQL

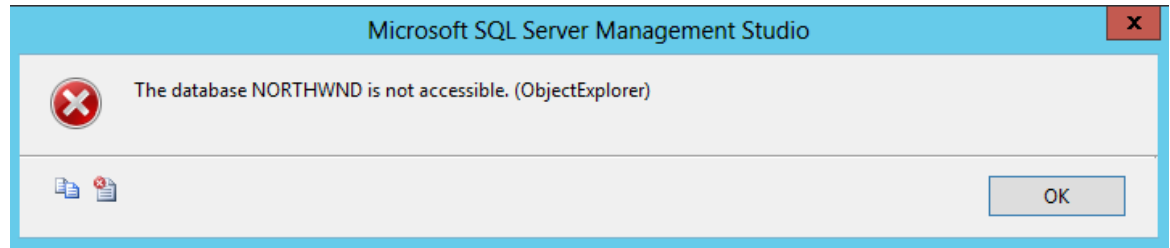
1. Crear un LOGIN con autenticación Windows

```
CREATE LOGIN [rca91\AnaC] FROM WINDOWS;
```

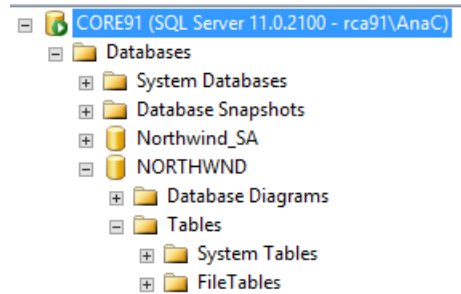
2. Asignar rol de servidor

```
ALTER SERVER ROLE bulkadmin ADD MEMBER [rca91\AnaC];
```

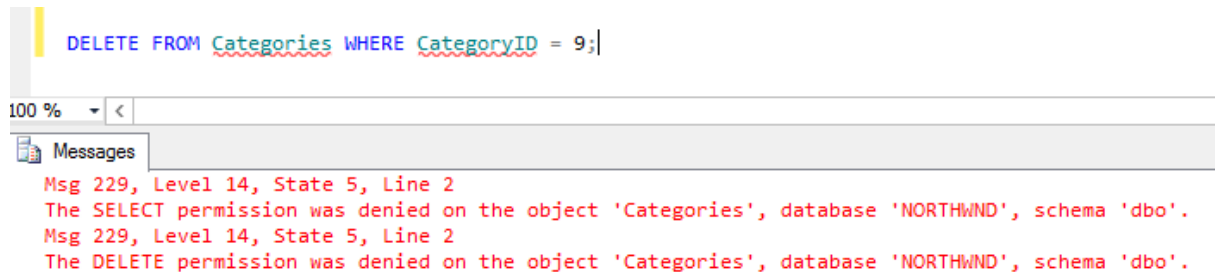
3. Crear un USER para la base de datos NORTHWND



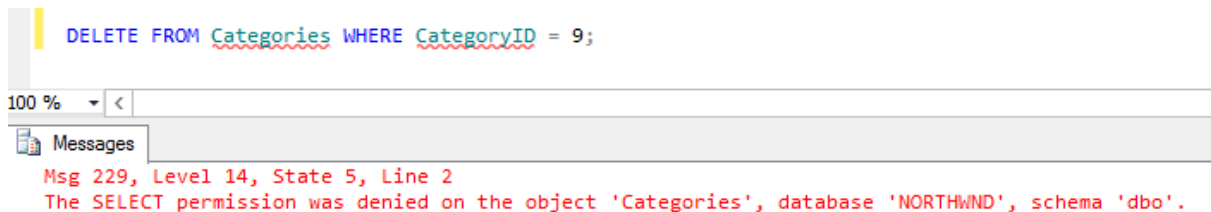
```
USE NORTHWND
CREATE USER [rca91\AnaC] FOR LOGIN [rca91\AnaC]
```



4. Asignar rol de base de datos



```
USE NORTHWND;
ALTER ROLE db_datawriter ADD MEMBER [rca91\AnaC];
```



5. Asignar privilegios

```
USE NORTHWND;
GRANT SELECT ON dbo.Categories TO [rca91\AnaC];
```

```
DELETE FROM Categories WHERE CategoryID = 9;  
SELECT * FROM Categories;
```

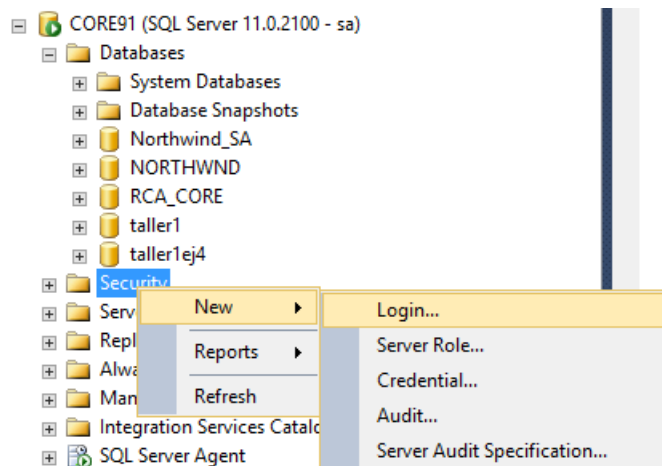
100 %

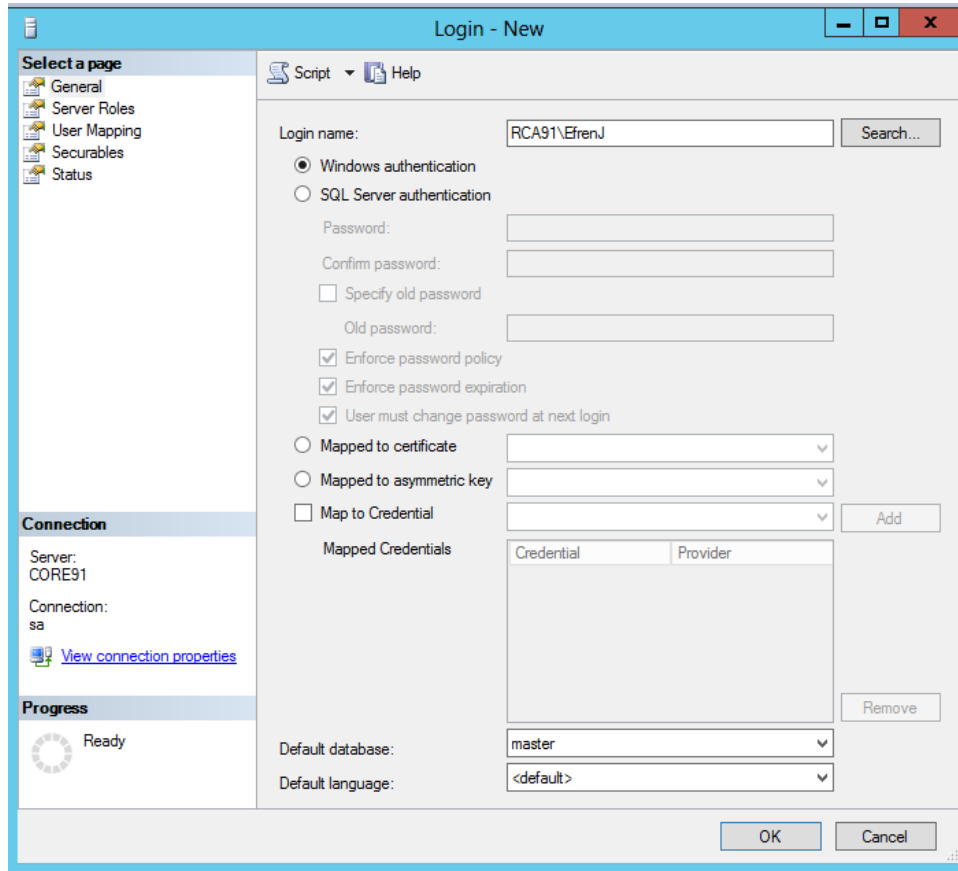
Results Messages

	CategoryID	CategoryName	Description	Picture
1	1	Beverages	Soft drinks, coffees, teas, beers, and ales	0x151C2F00020000...
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and ...	0x151C2F00020000...
3	3	Confections	Desserts, candies, and sweet breads	0x151C2F00020000...
4	4	Dairy Products	Cheeses	0x151C2F00020000...
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal	0x151C2F00020000...
6	6	Meat/Poultry	Prepared meats	0x151C2F00020000...
7	7	Produce	Dried fruit and bean curd	0x151C2F00020000...
8	8	Seafood	Seaweed and fish	0x151C2F00020000...

Utilizando GUI

1. Crear un LOGIN con autenticación Windows





Login - New

Script Help

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Connection

Server: CORE91

Connection: sa

[View connection properties](#)

Progress

Ready

Login name: RCA91\EfrenJ 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
------------	----------

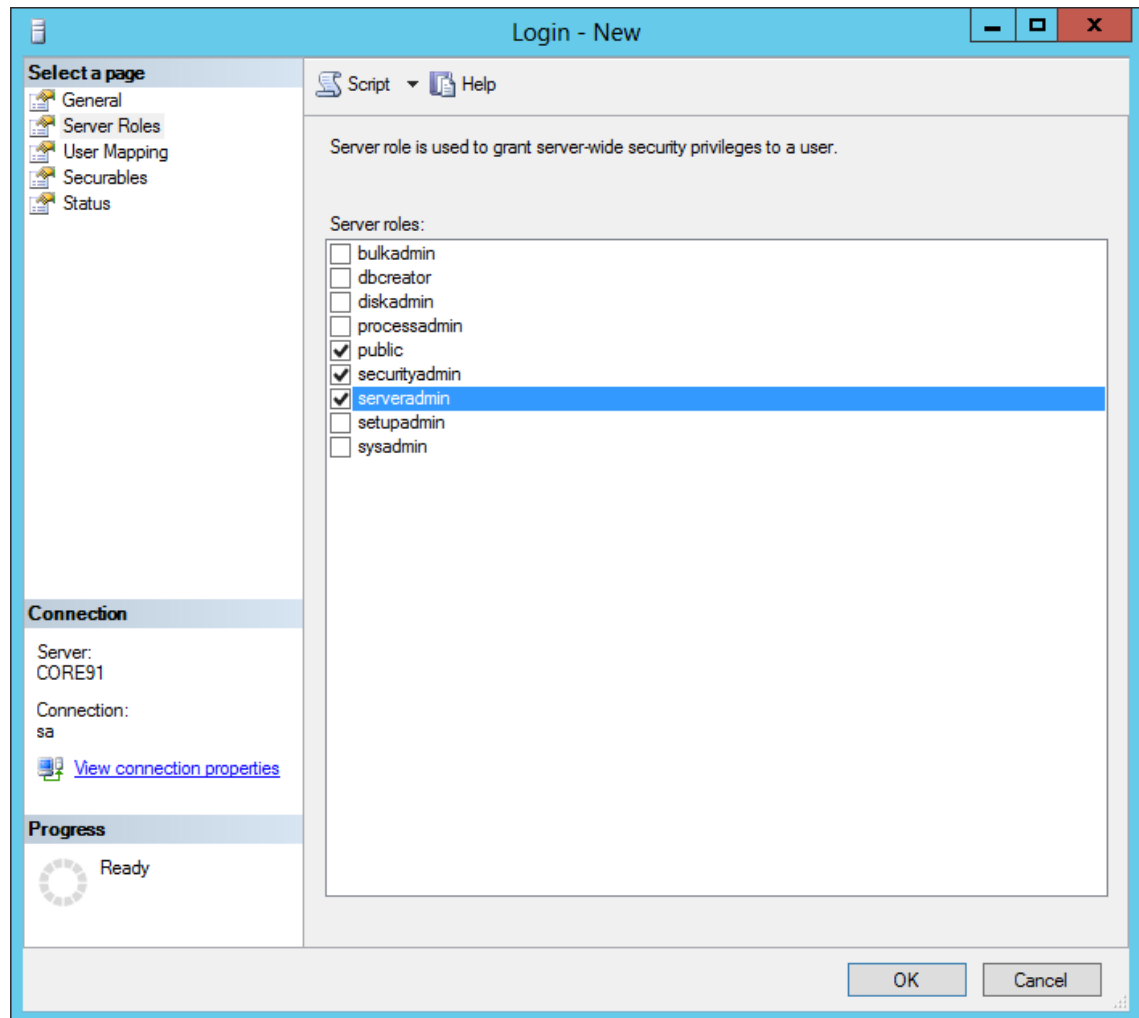
Remove

Default database: master

Default language: <default>

OK Cancel

2. Asignar rol de servidor



3. Crear un USER para la base de datos NORTHWND y asignar roles de base de datos

PAGE 18

4. Asignar privilegios

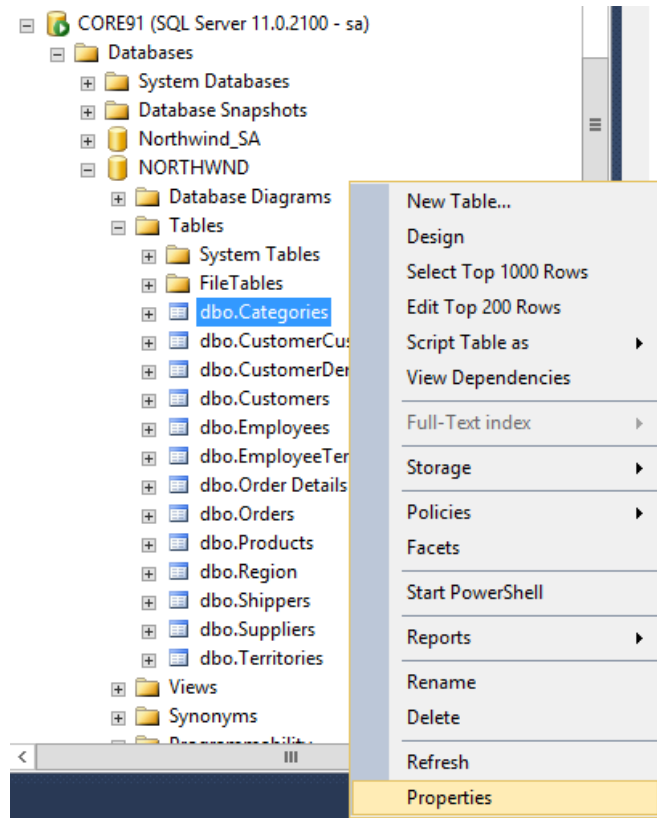


Table Properties - Categories

Script Help

Schema:

[View schema permissions](#)

Table name:

Users or roles:

	Name	Type
<input checked="" type="checkbox"/>	AlejandroM	User
<input type="checkbox"/>	appUser	User
<input type="checkbox"/>	rca91\AnaC	User

Permissions for AlejandroM:

Explicit Effective

Permission	Grantor	Grant	With Grant	Deny
Alter		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alter	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Control		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delete		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Insert		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Insert	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
References		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OK Cancel

Select Users or Roles

Select these object types:

Enter the object names to select ([examples](#)):

OK Cancel Help

PAGE 21

EXCEPCIONES

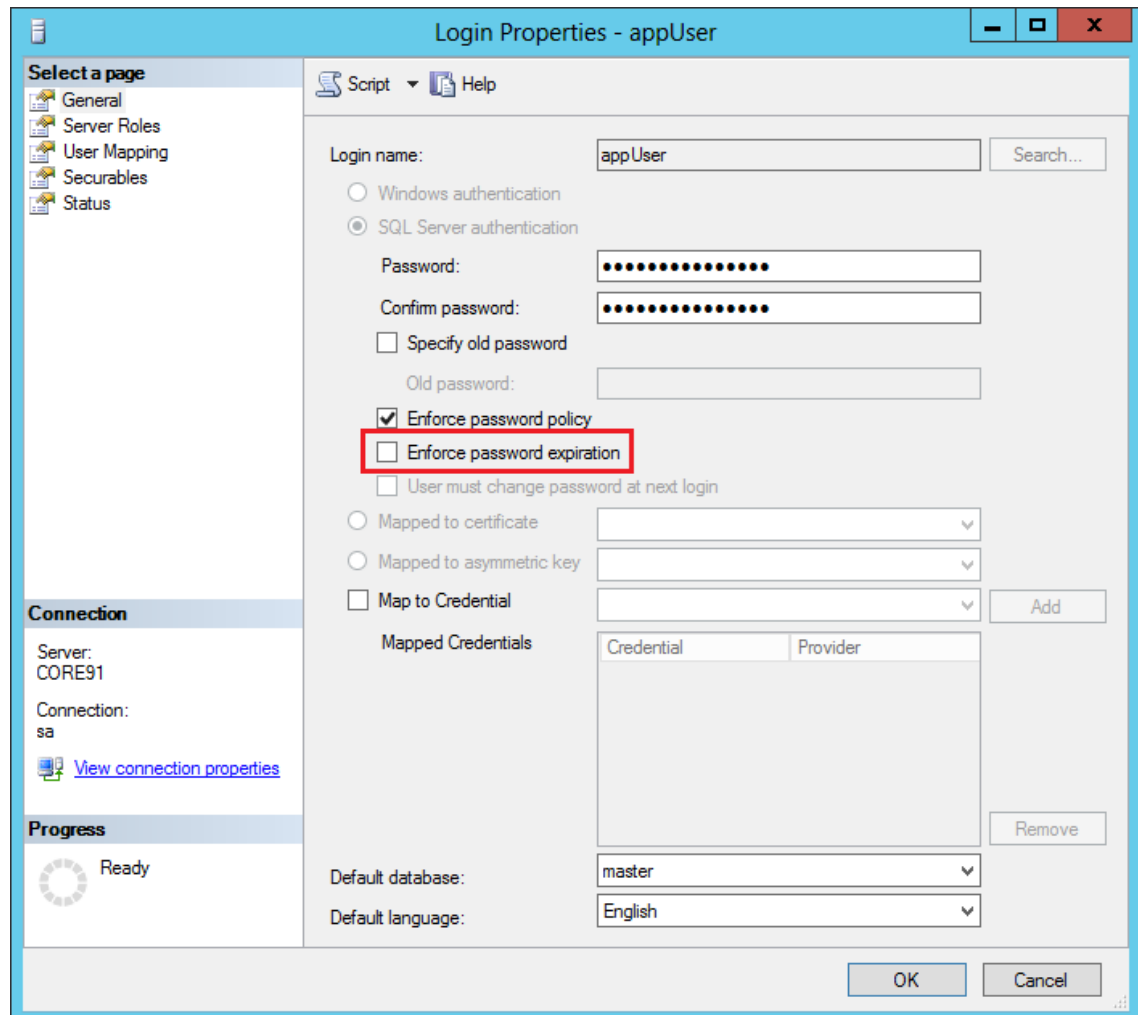
1. Deshabilitar expiración de contraseña

En algunos casos puede ser necesaria la deshabilitación de la expiración de la contraseña de un Login de SQL. Por ejemplo, si la combinación de usuario y contraseña es usada por una aplicación externa y la contraseña expira, la aplicación no podría acceder a la base de datos hasta que se realice el cambio de contraseña del lado de la aplicación. En este caso, es posible que sea mejor deshabilitar la expiración de la contraseña para que no suceda.

T-SQL:

```
ALTER LOGIN [appUser] WITH CHECK_EXPIRATION=OFF
```

GUI:



The screenshot shows the 'Login Properties - appUser' dialog box. The 'General' tab is active. Under 'SQL Server authentication', the 'Enforce password expiration' checkbox is highlighted with a red rectangle. Other options include 'Enforce password policy' (checked), 'User must change password at next login' (unchecked), 'Mapped to certificate', 'Mapped to asymmetric key', and 'Map to Credential'. The 'Connection' section shows 'Server: CORE91' and 'Connection: sa'. The 'Progress' section shows 'Ready'.

2. El usuario no debe cambiar la contraseña la próxima vez que haga login

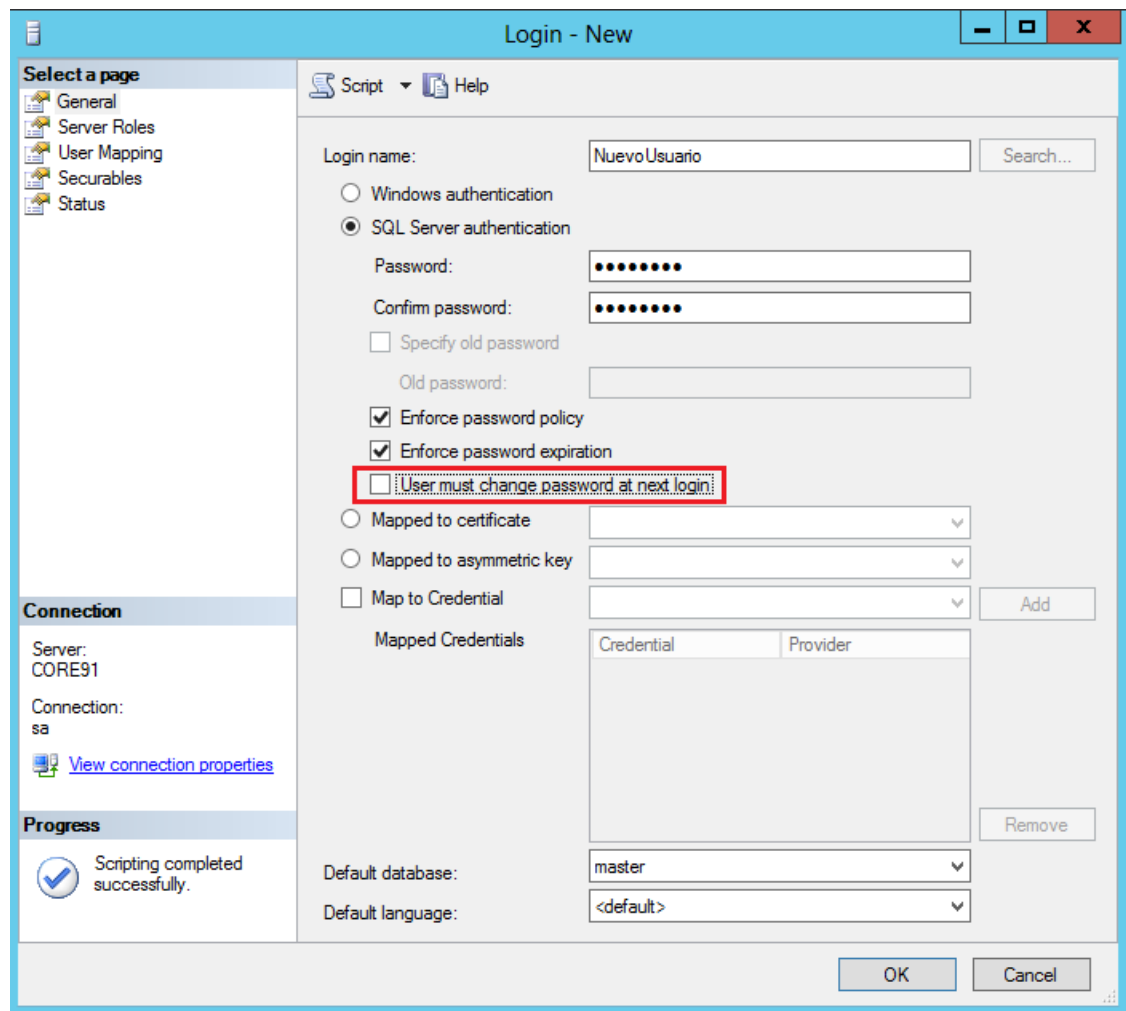
Si la contraseña debe ser compartida, guardada o recordada por varias personas, o si la próxima persona que ingrese utilizando el nuevo login no será la persona que utilizará la cuenta todos los días, entonces es preferible deshabilitar la opción de “cambiar la contraseña la próxima vez que haga login”.

T-SQL

*/*Si no se especifica la opción "MUST_CHANGE", entonces no será necesario cambiar la contraseña*/*

```
CREATE LOGIN [NuevoUsuario] WITH PASSWORD=N'Pa$$word' --MUST_CHANGE
```

GUI



The screenshot shows the 'Login - New' dialog box in SQL Server Enterprise Manager. The 'General' tab is active. The 'Login name' is 'NuevoUsuario'. The 'Authentication' section has 'SQL Server authentication' selected. The 'Password' and 'Confirm password' fields are filled with dots. The 'Specify old password' checkbox is unchecked. The 'Old password' field is empty. The 'Enforce password policy' and 'Enforce password expiration' checkboxes are checked. The 'User must change password at next login' checkbox is highlighted with a red rectangle and is unchecked. The 'Mapped to certificate', 'Mapped to asymmetric key', and 'Map to Credential' options are also visible but not selected. The 'Mapped Credentials' table is empty. The 'Default database' is set to 'master' and the 'Default language' is set to '<default>'. The 'OK' and 'Cancel' buttons are at the bottom right.

3. El usuario no debe tener políticas de contraseña

Si debe existir algún usuario a el cual no se le debe solicitar políticas de seguridad

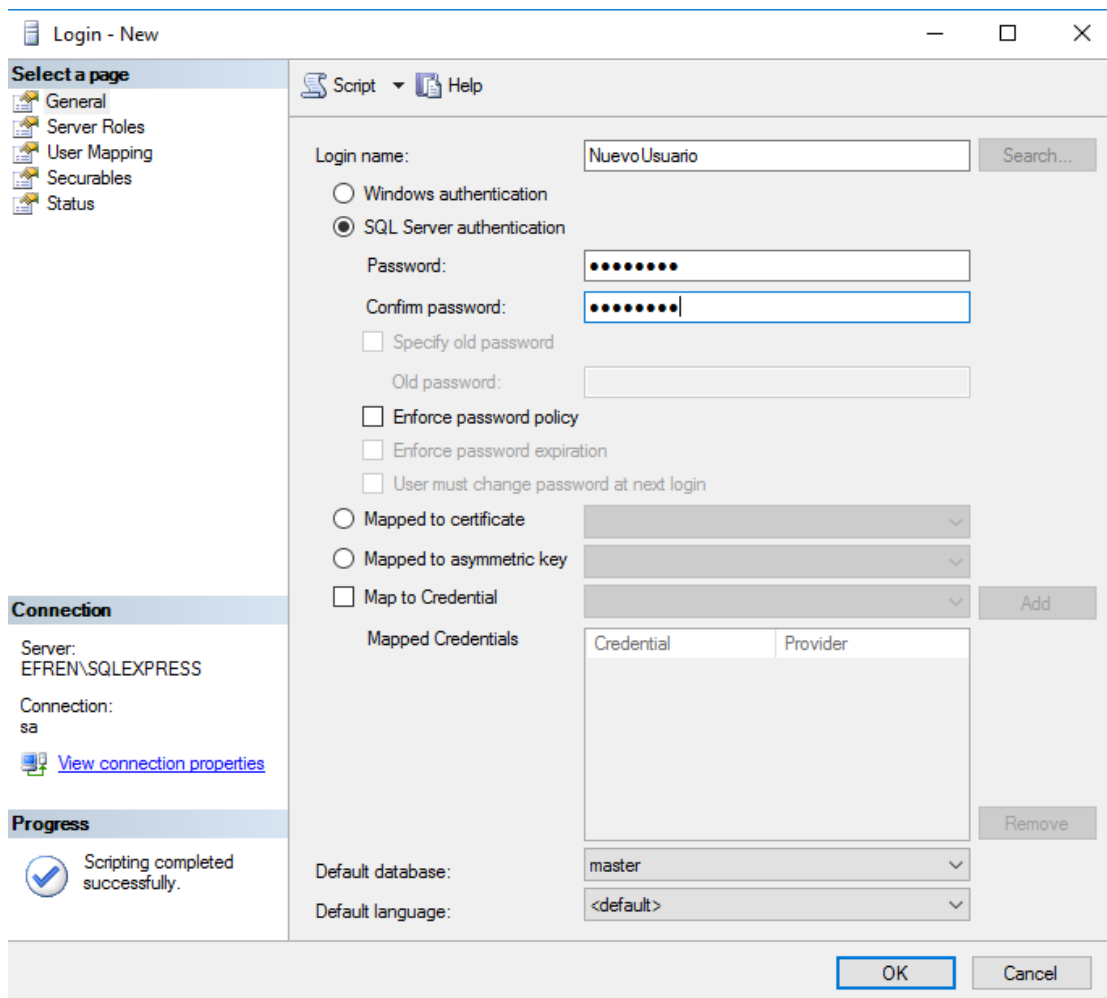
debido a conexiones existentes con sistemas ya sean de respaldos o de auditoría, y estos demanden mantener conexiones y contraseñas constantes debido a su uso automático se plantea la no utilización de políticas de contraseña.

T-SQL

*/*No se debe especifica la opción "CHECK_EXPIRATION" y "CHECK_POLICY", entonces no será necesario utilizar políticas de contraseña*/*

```
CREATE LOGIN [NuevoUsuario] WITH PASSWORD=N'Pa$$word' CREATE LOGIN
[NuevoUsuario] PASSWORD=N'Pa$$word', CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
```

GUI



Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Login name: NuevoUsuario 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

Mapped Credentials

Credential	Provider
------------	----------

Add Remove

Default database: master

Default language: <default>

OK Cancel

Connection

Server: EFREN\SQLEXPRESS

Connection: sa

[View connection properties](#)

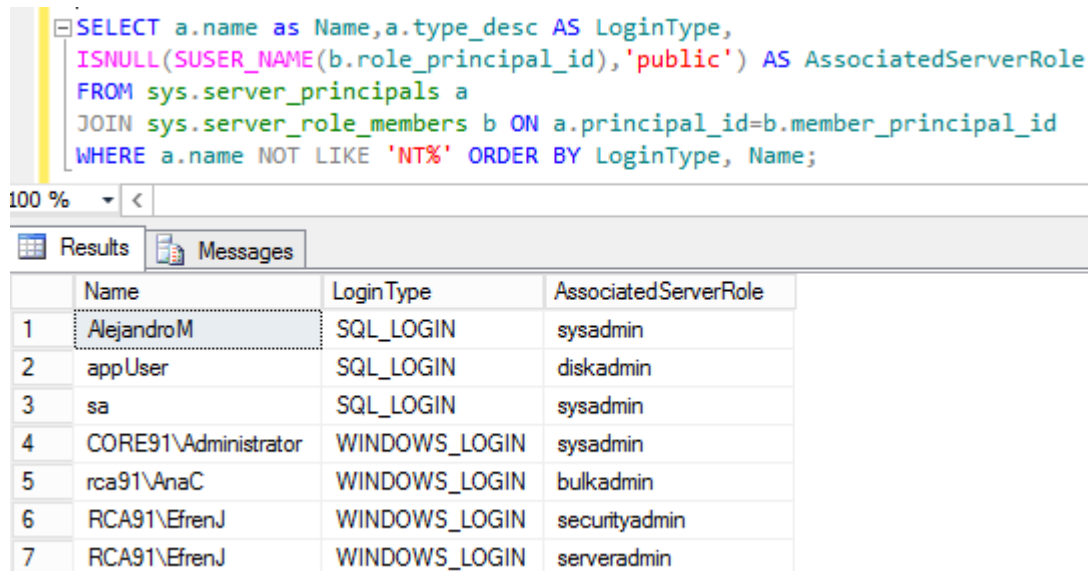
Progress

Scripting completed successfully.

GENERA EL REPORTE DETALLADO E INTERPRETA LOS RESULTADOS.

Roles a nivel de servidor por usuario

```
SELECT
    a.name as Name
    , a.type_desc AS LoginType
    , ISNULL(SUSER_NAME(b.role_principal_id),'public') AS AssociatedServerRole
FROM sys.server_principals a
JOIN sys.server_role_members b
    ON a.principal_id=b.member_principal_id
WHERE a.name NOT LIKE 'NT%'
ORDER BY LoginType, Name;
```



The screenshot shows a SQL query window with the following query:

```
SELECT a.name as Name,a.type_desc AS LoginType,
ISNULL(SUSER_NAME(b.role_principal_id),'public') AS AssociatedServerRole
FROM sys.server_principals a
JOIN sys.server_role_members b ON a.principal_id=b.member_principal_id
WHERE a.name NOT LIKE 'NT%' ORDER BY LoginType, Name;
```

Below the query window, the 'Results' tab is active, displaying the following data:

	Name	LoginType	AssociatedServerRole
1	AlejandroM	SQL_LOGIN	sysadmin
2	appUser	SQL_LOGIN	diskadmin
3	sa	SQL_LOGIN	sysadmin
4	CORE91\Administrator	WINDOWS_LOGIN	sysadmin
5	rca91\AnaC	WINDOWS_LOGIN	bulkadmin
6	RCA91\EfrenJ	WINDOWS_LOGIN	securityadmin
7	RCA91\EfrenJ	WINDOWS_LOGIN	serveradmin

Roles a nivel de base de datos por usuario

```
SELECT
    pr.name AS UserName
    , pr.type_desc AS LoginType
    , USER_NAME(me.role_principal_id) AS AssociatedDatabaseRole
    , DB_NAME() AS 'Database'
FROM sys.database_principals pr
LEFT OUTER JOIN sys.database_role_members me
    ON pr.principal_id=me.member_principal_id
WHERE me.role_principal_id IS NOT NULL
ORDER BY LoginType, UserName;
```

```

SELECT pr.name AS UserName, pr.type_desc AS LoginType,
USER_NAME(me.role_principal_id) AS AssociatedDatabaseRole, DB_NAME() AS 'Database'
FROM sys.database_principals pr
LEFT OUTER JOIN sys.database_role_members me ON pr.principal_id=me.member_principal_id
WHERE me.role_principal_id IS NOT NULL
ORDER BY LoginType, UserName;

```

100 % <

Results Messages

	UserName	LoginType	AssociatedDatabaseRole	Database
1	AlejandroM	SQL_USER	db_owner	NORTHWND
2	appUser	SQL_USER	db_datareader	NORTHWND
3	dbo	SQL_USER	db_owner	NORTHWND
4	rca91\AnaC	WINDOWS_USER	db_datawriter	NORTHWND
5	RCA91\EfrenJ	WINDOWS_USER	db_datawriter	NORTHWND
6	RCA91\EfrenJ	WINDOWS_USER	db_datareader	NORTHWND
7	RCA91\EfrenJ	WINDOWS_USER	db_accessadmin	NORTHWND
8	RCA91\EfrenJ	WINDOWS_USER	db_securityadmin	NORTHWND

Permisos de usuario a nivel de objetos de una base de datos

```

SELECT
    pr.type_desc AS LoginType
    , pr.name as Name
    , pe.permission_name AS 'Action', pe.state_desc AS 'Permission'
    , CASE class
        WHEN 0 THEN 'Database::' + DB_NAME()
        WHEN 1 THEN OBJECT_NAME(major_id)
        WHEN 3 THEN 'Schema::' + SCHEMA_NAME(major_id)
    END AS 'Securable'
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
ON pe.grantee_principal_id = pr.principal_id
WHERE pr.name <> 'guest' AND pr.name <> 'public'
ORDER BY LoginType, Name;

```

```

SELECT pr.type_desc AS LoginType, pr.name as Name,
       pe.permission_name AS 'Action', pe.state_desc AS 'Permission',
       CASE class
         WHEN 0 THEN 'Database::' + DB_NAME()
         WHEN 1 THEN OBJECT_NAME(major_id)
         WHEN 3 THEN 'Schema::' + SCHEMA_NAME(major_id) END AS 'Securable'
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
ON pe.grantee_principal_id = pr.principal_id
WHERE pr.name <> 'guest' AND pr.name <> 'public' ORDER BY LoginType, Name;

```

100 % <

Results Messages

	LoginType	Name	Action	Permission	Securable
1	SQL_USER	AlejandroM	CONNECT	GRANT	Database::NORTHWND
2	SQL_USER	AlejandroM	ALTER	GRANT	Categories
3	SQL_USER	AlejandroM	INSERT	GRANT	Categories
4	SQL_USER	appUser	INSERT	GRANT	Categories
5	SQL_USER	appUser	CONNECT	GRANT	Database::NORTHWND
6	SQL_USER	dbo	CONNECT	GRANT	Database::NORTHWND
7	WINDOWS_USER	rca91\AnaC	CONNECT	GRANT	Database::NORTHWND
8	WINDOWS_USER	rca91\AnaC	SELECT	GRANT	Categories
9	WINDOWS_USER	RCA91\EfrenJ	CONNECT	GRANT	Database::NORTHWND
10	WINDOWS_USER	RCA91\EfrenJ	CONTROL	DENY	Categories

CONCLUSIONES

1. El nivel de integración que proporciona el Microsoft SQL Server con las tecnologías de autenticación empresarial del mismo fabricante, permite una experiencia transparente en el proceso de conexión con los usuarios. Es muy clara la división de las siguientes funciones:
 - a. Creación de usuarios, mantenimiento de contraseñas (Active Directory).
 - b. Habilitar la conexión de usuarios existentes con cuentas de la base de datos. Se observa que el uso de cuentas LOGIN y cuentas USER es el resultado de soportar el sistema de autenticación basado en SQL SERVER, que no deja de ser un requisito estándar de las implementaciones de SQL (SQL Server).
 - c. Administración de los permisos de acceso de los usuarios a los componentes de la base de datos (SQL Server).
2. La existencia de la autenticación independiente de los usuarios (Microsoft SQL Authentication) no debe verse simplemente como el resultado de utilizar tecnologías más antiguas, sino también con la necesidad de soportar la conectividad y autenticación cuando se trata de clientes distintos de la cartera de productos de Microsoft, por ejemplo:
 - a. A nivel de sistemas operativos diferentes como Linux, UNIX.
 - b. A nivel de productos no provistos por Microsoft en estos ambientes, donde se requiere autenticación mediante ODBC, JDBC, etc.
3. La asignación de permisos en SQL Server no solamente permite la granularidad requerida por los estándares de SQL (usuario, tabla), sino también extender los conceptos de grupo del dominio activo, para simplificar y normalizar el manejo a lo largo de una organización. Esto permite autorizar o denegar a los usuarios diferentes funciones y accesos a nivel de tabla y de base de datos. Estos permisos se pueden asignar a ambos tipos de usuario, o pueden ser heredados según los roles que le han sido asignados.

RECOMENDACIONES

1. La fase de diseño de los diferentes accesos en SQL Server requiere un análisis previo de las prácticas de la organización, para diseñar apropiadamente aquellas políticas que mejor se adapten a estos requerimientos. Por ejemplo, algunas organizaciones, pueden poseer una estructura bastante plana cuando se trata de las unidades de desarrollo y soporte de sus bases de datos, un solo grupo de TI puede estar a cargo de dicha organización. Otras, por su parte, pueden por el contrario tener una gran cantidad de grupos de negocio y sus áreas de soporte pueden encontrarse alineadas a dichas unidades de negocio o pueden tener más bien tener una organización matricial. En estos casos, diseñar una estructura de seguridad, requerirá el manejo de un mayor número de jerarquías y niveles de acceso, así como conceptos de cuentas de servicio.
2. Uno de los retos más importantes que algunas organizaciones pueden enfrentar, es la designación de las cuentas de usuario bajo las cuales ejecutan los procesos críticos, como el SQL server. En el pasado, dos modelos básicos han operado:
 - a. Utilizar cuentas locales en los servidores de base de datos, bajo las cuales los procesos principales de SQL server son ejecutados. Esto tiene la desventaja de la falta de centralización de las políticas de ejecución y la necesidad de mantener la renovación de contraseñas en cada equipo individual.
 - b. Crear cuenta de pseudo usuario, cuyo único propósito es la ejecución de los servicios de SQL server en los servidores. Si bien, esto elimina el problema de la descentralización, no obstante, se queda corto cuando se trata de manejar la vida de las contraseñas y limitar la ejecución de tales o cuales servicios a los servidores previamente definidos.
 - c. La aparición en Windows 2008 R2 y el subsiguiente desarrollo en 2012, de las cuentas globales de administración, o cuentas grupales de administración, vienen a atacar de frente este problema. Con este modelo, es posible crear usuarios cuyo único propósito es correr servicios en uno o más equipos previamente definidos. El manejo de la vida de las contraseñas recae sobre el sistema de directorio activo, sin requerir la intervención del administrador. De tal forma, se obtiene lo mejor de ambos mundos.
3. Las configuraciones por defecto del base de datos toman en cuenta una serie de compromisos entre la seguridad y la operación normal de los sistemas en las bases instaladas de los clientes, especialmente consideraciones de soporte de ediciones anteriores o productos populares. Por este motivo, muchas veces las decisiones de configuración por defecto no pueden asumirse como “seguras”, especialmente si se trata de sistemas donde la exposición de la información acarrea riesgos sobre los datos financieros de los clientes. Debido a esto, una de las tareas del administrador y los grupos de seguridad, es evaluar las configuraciones por defecto de forma detallada, definir el nivel de apetito de riesgo de sus compañías y los recursos materiales con los cuáles cuentan, para tomar las decisiones de configuración que mejor se alineen con su realidad.

BIBLIOGRAFÍA

ALTER ROLE (Transact-SQL). (s.f.). Recuperado de [https://msdn.microsoft.com/en-us/library/ms189775\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ms189775(v=sql.110).aspx)

ALTER SERVER ROLE (Transact-SQL). (s.f.). Recuperado de [https://msdn.microsoft.com/en-us/library/ee677634\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ee677634(v=sql.110).aspx)

Auditing SQL Server User and Role Permissions for Databases. (s.f.). Recuperado de <https://www.mssqltips.com/sqlservertip/2132/auditing-sql-server-user-and-role-permissions-for-databases/>

Choose an Authentication Mode. (s.f.). Recuperado de <https://msdn.microsoft.com/en-us/library/ms144284.aspx>

Change Server Authentication Mode. (s.f.). Recuperado de <https://msdn.microsoft.com/en-us/library/ms188670.aspx>

Create a Login. (s.f.). Recuperado de [https://msdn.microsoft.com/en-us/library/aa337562\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/aa337562(v=sql.110).aspx)

Permissions of Fixed Database Roles (Database Engine). (s.f.). Recuperado de <https://msdn.microsoft.com/library/ms189612.aspx>

Permissions of Fixed Server Roles (Database Engine). (s.f.). Recuperado de <https://msdn.microsoft.com/library/ms175892.aspx>

Shyamsundar, A. (2014). Managed Service Accounts (MSA) and SQL 2012: Practical Tips. Recuperado de <http://blogs.msdn.com/b/arvindsh/archive/2014/02/03/managed-service-accounts-msa-and-sql-2012-practical-tips.aspx>

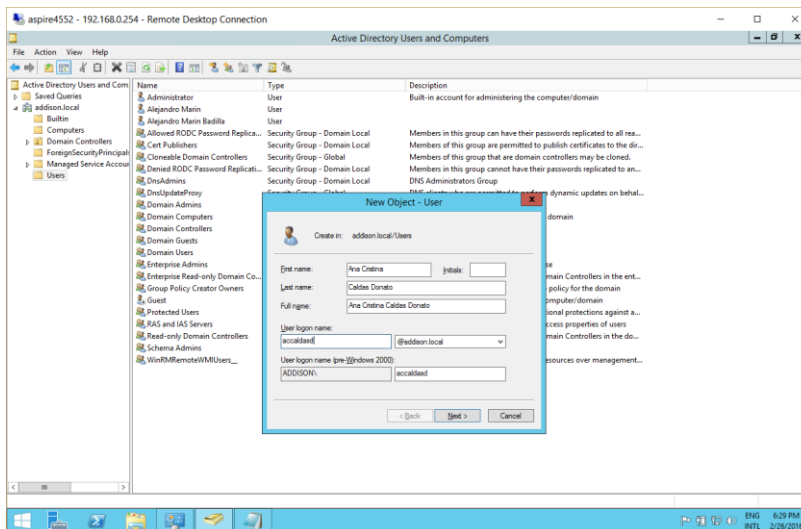
T-SQL: Retrieve all users and associated roles for ALL databases. (2012). Recuperado de <https://www.pythian.com/blog/httpconsultingblogs-emc-comjamiethomsonarchive20070209sql-server-2005-3a00-view-all-permissions-2800-2-2900.aspx/>

Anexo 1 - Ejemplo de Creación de cuentas desde el Active Directory y autorización de SQL Server paso a paso

Configuración de los usuarios

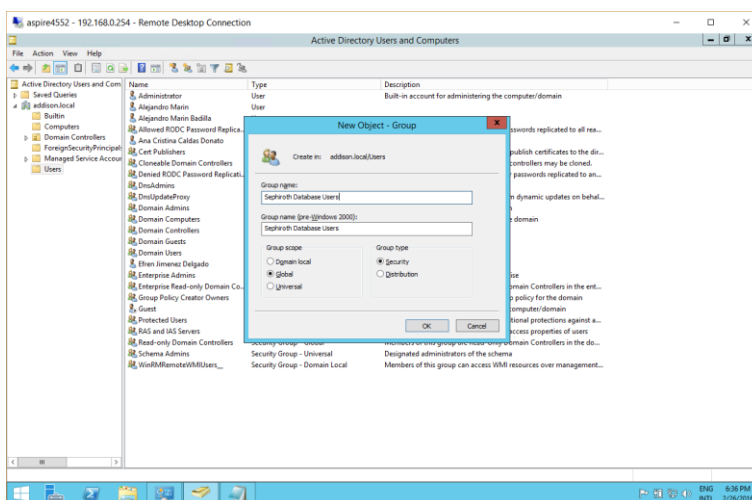
1. Crear el usuario en el Dominio Activo.

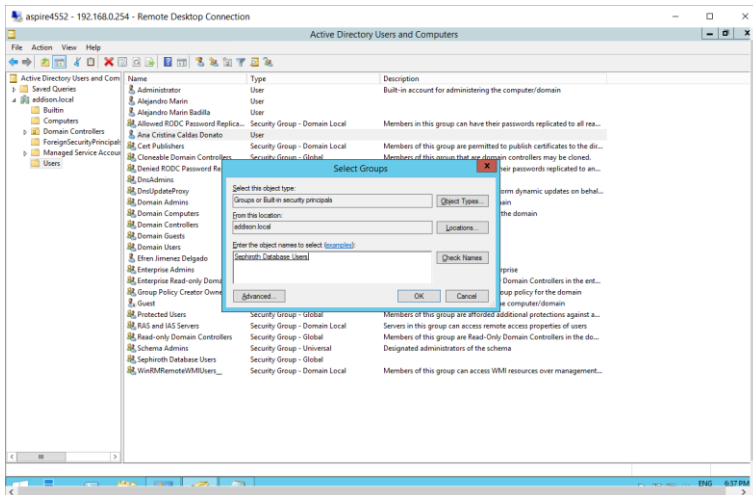
Por ejemplo, crear el usuario accaldasd, asociado a la persona Ana Cristina Caldas Donato, con la cuenta de correo caldas.donato@gmail.com.



2. Crear un Grupo para contener a los usuarios de la base de datos, y asignar los usuarios al grupo

Por ejemplo, crear el grupo “Sephiroth Database Users”

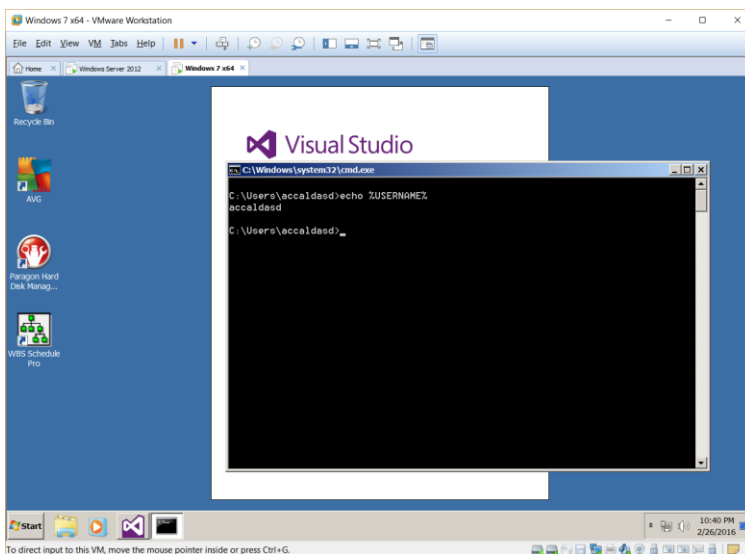


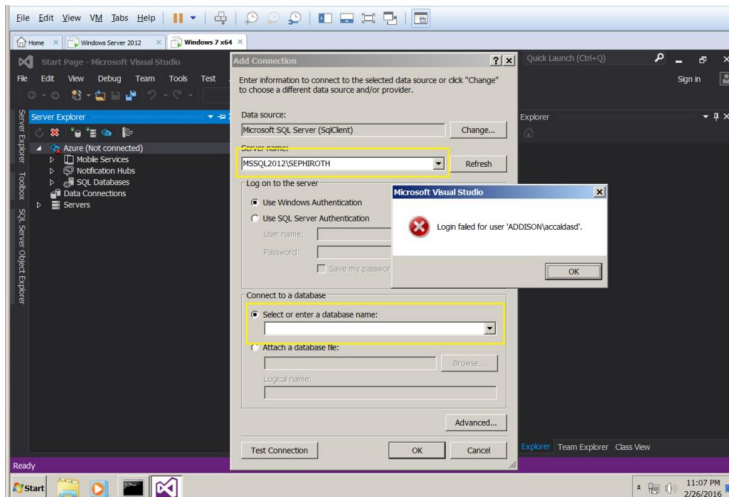


El proceso de autorización de cuentas de usuario basada en el servidor de dominio de activo activo, conlleva los siguientes pasos:

1. Verificar que el usuario no es capaz de iniciar una sesión autenticada con el SQL Server

Procedemos a iniciar una sesión con el nuevo usuario en una estación de trabajo (equipo terminal). El siguiente pantallazo muestra que se está utilizando el usuario creado en el dominio.





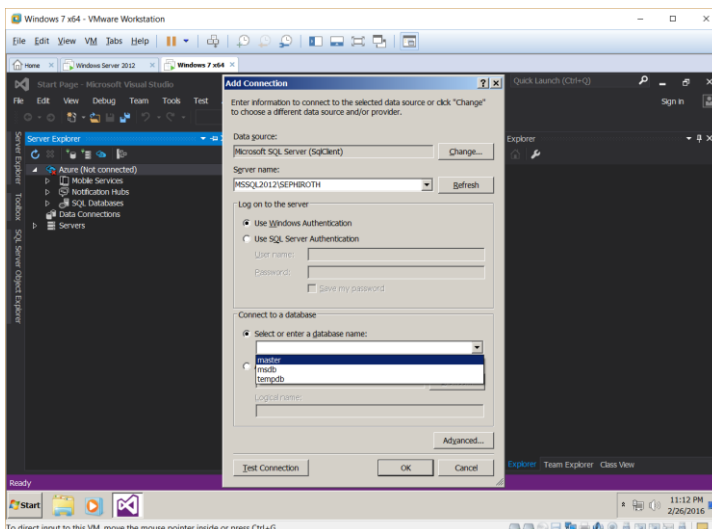
1. Crear un LOGIN asociado a la cuenta de dominio.

```
CREATE LOGIN [domain\user] FROM WINDOWS;
```

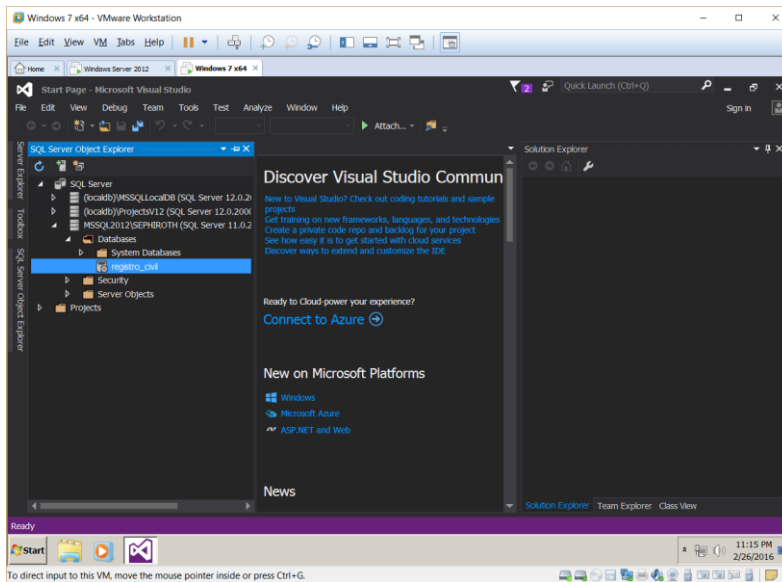
Observe que al crearse el LOGIN, el servidor permite al usuario ver la lista de base de datos creadas por default.

Ejemplo:

```
CREATE LOGIN [addison\accaldasd] FROM WINDOWS;
```



Sin embargo, si hacemos uso de la herramienta de exploración de objetos, notaremos que no es posible establecer una conexión con el schema registro_civil, el cual ya está configurado en el sistema y posee una tabla. Observe como sobre el icono del schema registro_civil, está superpuesta una (x), que señala que no es posible establecer la conexión.

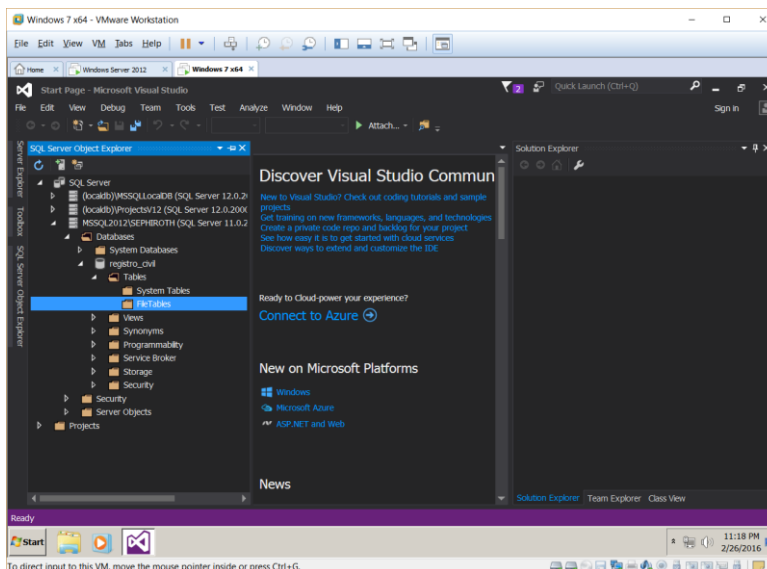


3. Crear un USER asociado al login

```
CREATE USER [domain\user] FROM LOGIN [domain\user];
```

Ejemplo:

```
CREATE USER [addison\accaldas] FROM LOGIN [addison\accaldas];
```



Con la configuración anterior se ha asociado un LOGIN a una cuenta de USER en el servidor el SQL Server. En esta ocasión, el cliente puede establecer la conexión con el schema, sin embargo, no le es posible ver los objetos o tablas configurados en el schema.

4. Habilitar el acceso del usuario de dominio al servidor de base de datos.

(<https://msdn.microsoft.com/en-us/library/ms173463.aspx>)

```
GRANT CONNECT TO [domain\user];
```

Ejemplo:

Nota: en versiones anteriores de SQL server, se utilizaba el stored procedure sp_grantdbaccess para habilitar el permiso de conexión a la base de datos. Este procedimiento se considera obsoleto y se recomienda no utilizarse ya que va a ser removido en futuras versiones de SQL Server.

(<https://msdn.microsoft.com/en-us/library/ms178013.aspx>)

```
EXEC sp_grantdbaccess [domain\user], [domain\user]
```

Observe que, de esta manera, el usuario puede establecer una sesión con el servidor de SQL Server, no obstante, podrá notar que este no puede más que establecer una sesión, sin embargo, no puede acceder a las operaciones comunes de la base de datos

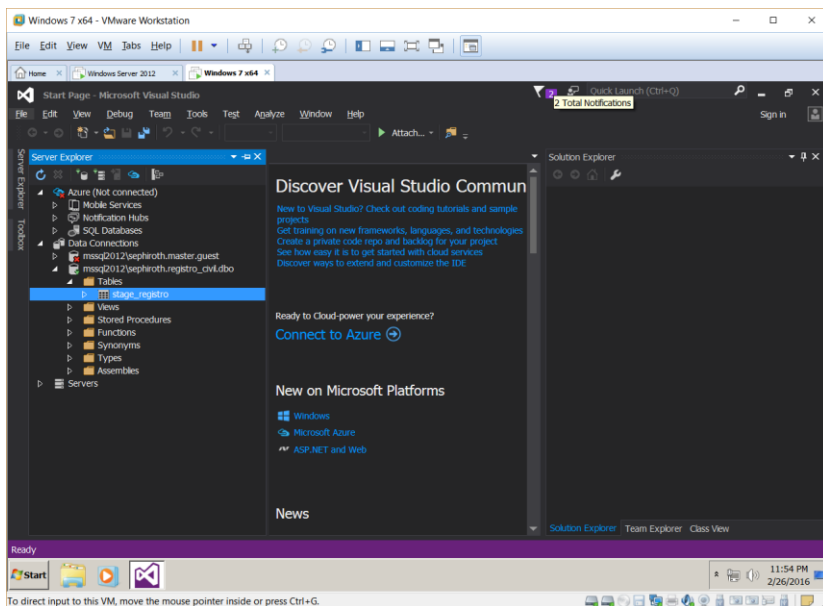
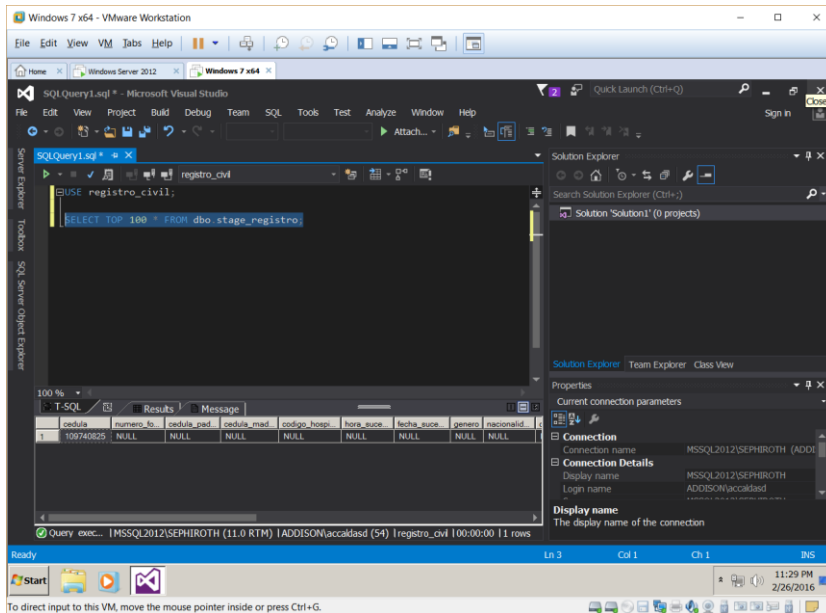
5. Habilitar los permisos apropiados en la tabla.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Tablename TO [domain\user];
```

Ejemplo:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.stage_registro TO  
[addison\accaldas];
```

De esta manera, ya es posible ejecutar un comando de selección sobre la tabla. Observe, sin embargo, que el browser de objetos, no es posible listar las tablas, debido a que estos es un permiso separada.



Reporte permisos de usuarios (servidor)

```

SELECT
  a.name as Name
  , a.type_desc AS LoginType
  , a.default_database_name AS DefaultDBName
  , ISNULL(SUSER_NAME(b.role_principal_id), 'public') AS AssociatedServerRole
  , c.class_desc AS ClassDesc, c.permission_name AS ServerLevelPermission
  , c.state_desc AS PermissionState
FROM sys.server_principals a
LEFT JOIN sys.server_role_members b
  ON a.principal_id=b.member_principal_id
JOIN sys.server_permissions c

```

```

        ON a.principal_id = c.grantee_principal_id
WHERE LEFT(a.name,2) <> '##'
        AND a.name <> 'public'
ORDER BY Name, LoginType;

```

Name	LoginType	DefaultDBName	AssociatedServerRole	ClassDesc	ServerLevelPermission	PermissionState
addison\accaldasd	WINDOWS_LOGIN	master	public	SERVER	CONNECT SQL	GRANT
addison\accaldasd	WINDOWS_LOGIN	master	public	SERVER	VIEW ANY DATABASE	GRANT
ADDISON\nbkypeu	WINDOWS_LOGIN	master	sysadmin	SERVER	CONNECT SQL	GRANT
ADDISON\SQL2012MSA\$	WINDOWS_LOGIN	master	sysadmin	SERVER	CONNECT SQL	GRANT
ADDISON\SQLServerSecurityAdmins	WINDOWS_GROUP	DB1PROD	securityadmin	SERVER	CONNECT SQL	GRANT
NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	master	public	SERVER	ALTER ANY AVAILABILITY GROUP	GRANT
NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	master	public	SERVER	CONNECT SQL	GRANT
NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	master	public	SERVER	VIEW SERVER STATE	GRANT
NT SERVICE\SQLAgent\$SEPHIROTH	WINDOWS_LOGIN	master	sysadmin	SERVER	CONNECT SQL	GRANT
NT SERVICE\SQLWriter	WINDOWS_LOGIN	master	sysadmin	SERVER	CONNECT SQL	GRANT
NT SERVICE\Winmgmt	WINDOWS_LOGIN	master	sysadmin	SERVER	CONNECT SQL	GRANT
sa	SQL_LOGIN	master	sysadmin	SERVER	CONNECT SQL	GRANT

--Reporte permisos de usuarios (base de datos):

```

SELECT
    pr.name AS Name
    ,pr.type_desc AS LoginType
    , pr.authentication_type_desc AS AuthType
    , pe.class_desc AS ClassDesc,pe.permission_name AS DatabaseLevelPermission
    ,pe.state_desc AS PermissionState
FROM sys.database_permissions pe
JOIN sys.database_principals pr
    ON pr.principal_id = pe.grantee_principal_id
WHERE pr.name <> 'public'
AND LEFT(pr.Name,2) <> '##'
ORDER BY Name, LoginType;

```

Name	LoginType	AuthType	ClassDesc	DatabaseLevelPermission	PermissionState
addison\accaldasd	WINDOWS_USER	WINDOWS	DATABASE	CONNECT	GRANT
dbo	SQL_USER	INSTANCE	DATABASE	CONNECT	GRANT
guest	SQL_USER	NONE	DATABASE	CONNECT	GRANT