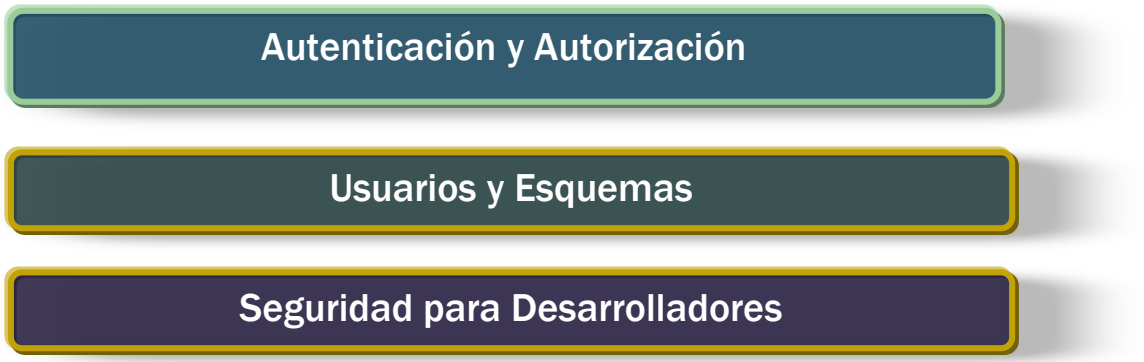
	UNIVERSIDAD DON BOSCO **FACULTAD DE ESTUDIOS TECNOLÓGICOS** **ESCUELA DE COMPUTACIÓN**
CICLO 1-2016	
	Diseño de seguridad de una Base de datos

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

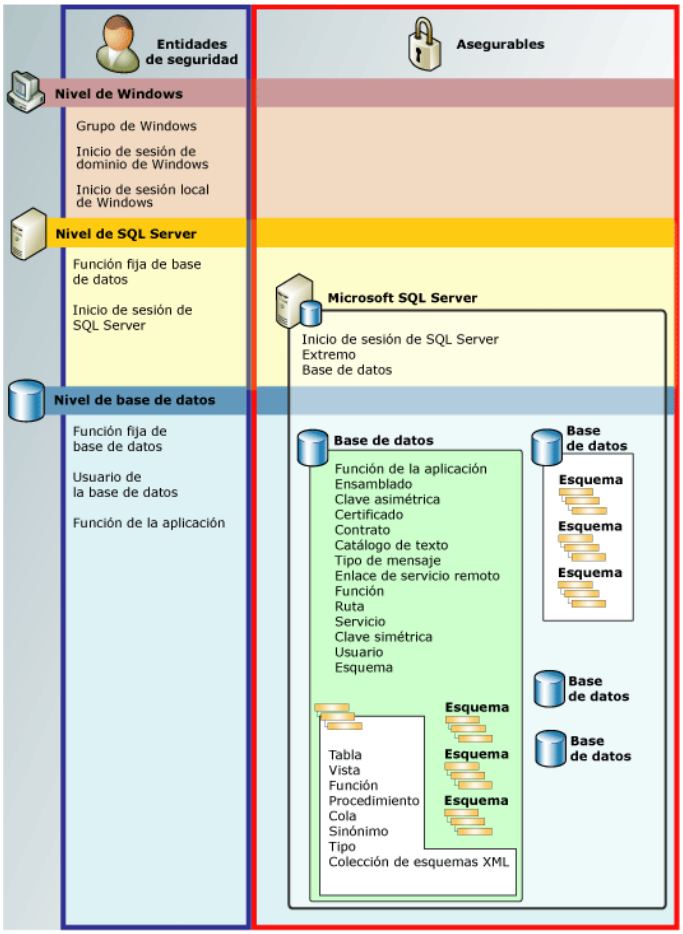
Un role de aplicación sirve para asignarle permisos a una aplicación:

- Tiene un password
- No contiene usuarios

Jerarquía de permisos

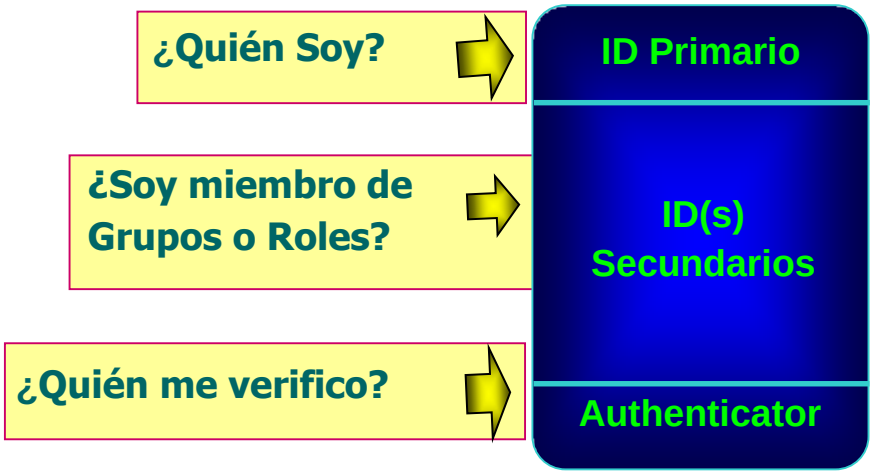
El Motor de base de datos administra un conjunto jerárquico de entidades que se pueden proteger mediante permisos. Estas entidades se conocen como elementos protegibles. Los protegibles más prominentes son los servidores y las bases de datos, pero los permisos discretos se pueden establecer en un nivel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

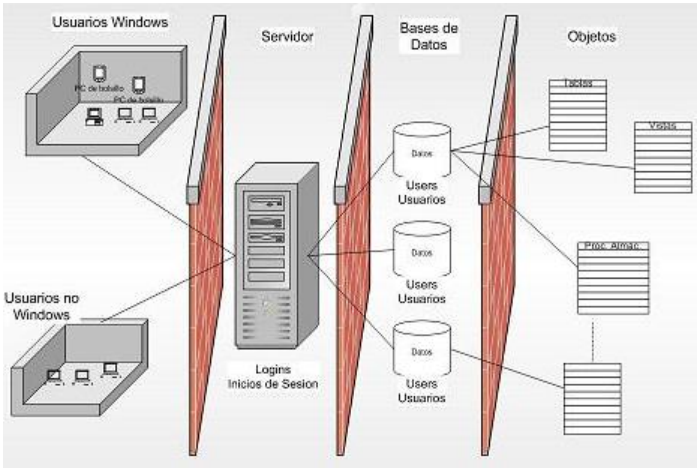


Figura tomada de: http://www.sqlserversi.com/2009/01/seguridad-en-sql-server.html

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

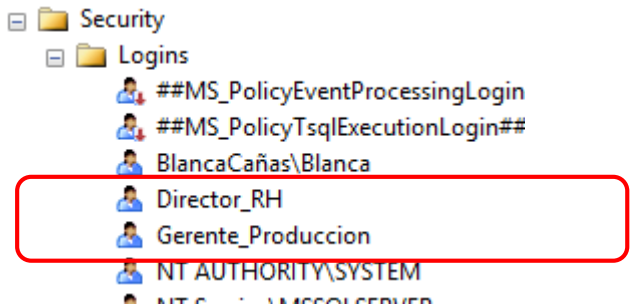
Sintaxis:

CREATE LOGIN nombre_login
WITH PASSWORD = 'clave_usuario';

Creando de los inicios de sesión:

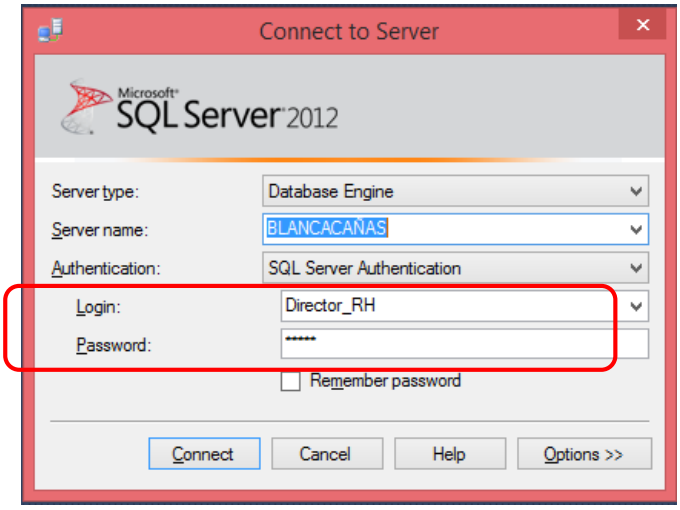
```
CREATE LOGIN Director_RH
WITH PASSWORD = '12345';

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

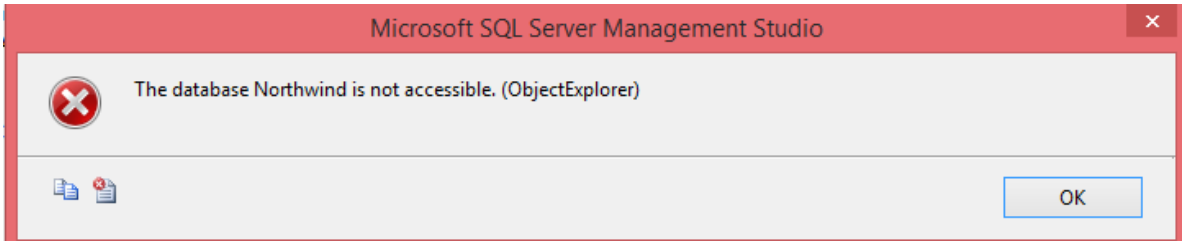


Comprobando los inicios de sesión:

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:

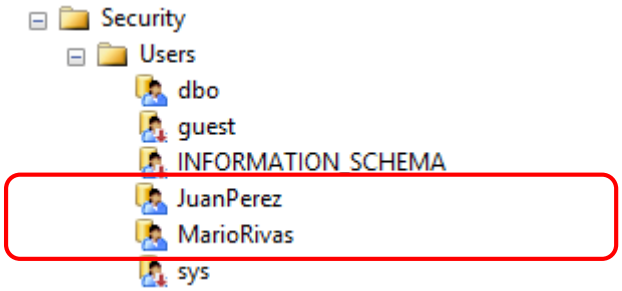
USE BASE_DE_DATOS;

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

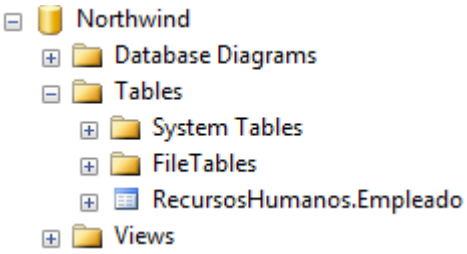
Vamos a asignarle un esquema al usuario, por ejemplo el esquema RecursosHumanos.

```
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.

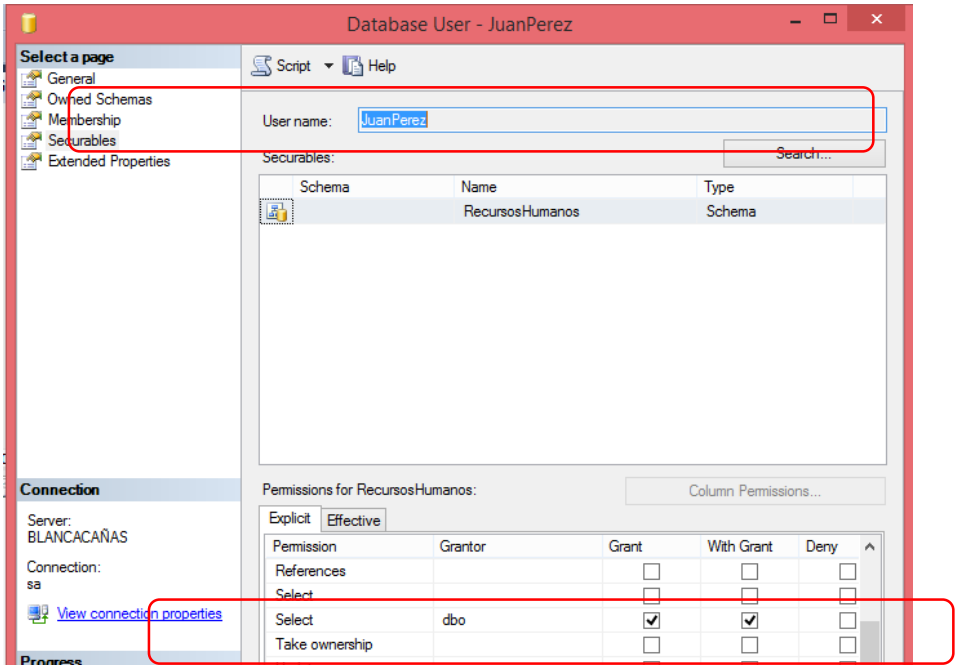
```
INSERT INTO RecursosHumanos.Empleado VALUES (4, 'Marisol', 'Abarca')
```

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

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

Messages

```
(1 row(s) affected)
```

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

Ejercicios

Creación de la base de datos, esquemas y tablas

1. Crear la base de datos Alumnos_SuCarnet

```
CREATE DATABASE Alumnos_SuCarnet
GO

USE Alumnos_SuCarnet
GO
```

Nota: Sustituir la palabra SuCarnet por su número de carnet

2. Ahora se van a crear dos esquemas en la base de datos

```
--Creacion de esquemas
CREATE SCHEMA alumno
GO

CREATE SCHEMA nota
GO
```

3. Crear las tablas de la base de datos, cada una en un esquema distinto

```
--Creacion de la tabla  notas en el esquema alumno
CREATE TABLE alumno.alumnos(
carnet int PRIMARY KEY,
nombres varchar(25),
apellidos varchar(25))
GO


--Creacion de la tabla notas en el esquema nota
CREATE TABLE nota.notas(
idnotas int identity,
carnet int FOREIGN KEY REFERENCES alumno.alumnos(carnet)
ON UPDATE CASCADE
ON DELETE CASCADE,
nota1 decimal(10,2),
nota2 decimal(10,2),
nota3 decimal(10,2),
promedio as (nota1+nota2+nota3)/3
)
```

4. Agregar datos a la base de datos

```
--Agregando datos
INSERT INTO alumno.alumnos VALUES(111,'Juan Jose','Perez')
INSERT INTO alumno.alumnos VALUES(222,'Maria Luisa ','Flores')
INSERT INTO alumno.alumnos VALUES(333,'Carlos Francisco ','Gavidia')
INSERT INTO alumno.alumnos VALUES(444,'Claudia Evelyn','Rivas')

INSERT INTO nota.notas VALUES(111,7.6,10,5.5)
INSERT INTO nota.notas VALUES(222,8.5,9,10)
INSERT INTO nota.notas VALUES(333,9.3,8.5,5.7)
```

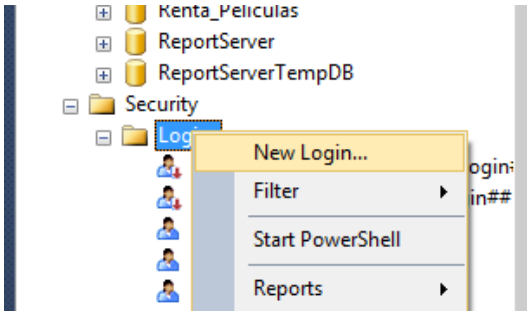
5. Seleccionar los datos de la base de datos

```
--Seleccione los datos
SELECT * FROM alumno.alumnos
SELECT * FROM nota.notas
```

Creación de inicio de sesión con SQL Management Studio y Transact SQL

1. En SQL Server Management Studio, abrir el Explorador de objetos y expanda la carpeta que tiene el nombre del servidor.

2. Expandir la carpeta Seguridad (Security), en la carpeta Logins hacer clic derecho y seleccionar la opción Nuevo Inicio de sesión (New Login)

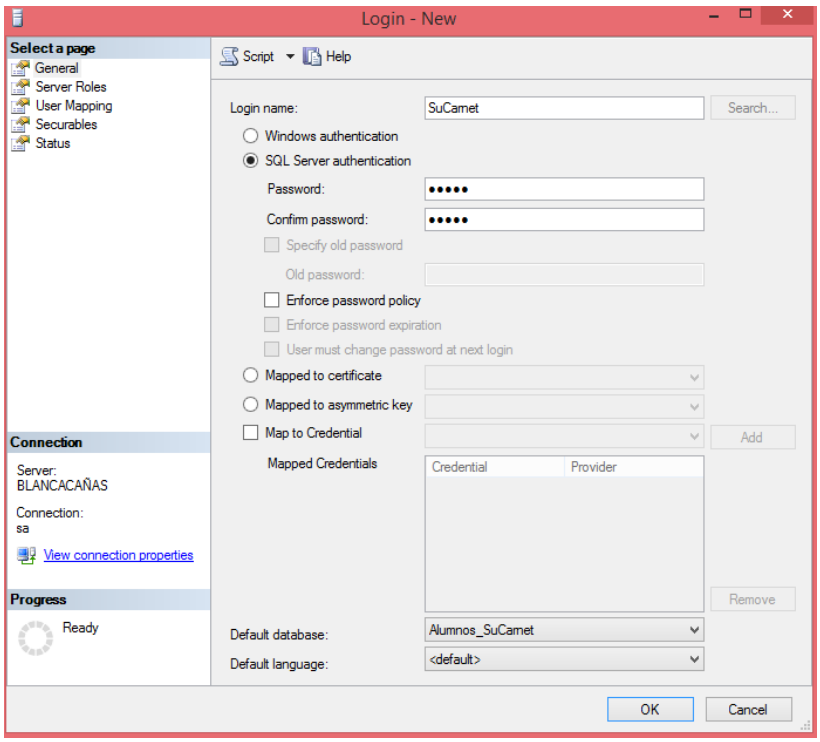


3. En la página General, escribir un nombre para el nuevo inicio de sesión en el cuadro Nombre de inicio de sesión, para este caso su carnet

4. Seleccionar Autenticación de SQL Server.

5. Escribir una contraseña de inicio de sesión, puede ser también el carnet.

6. Desmarcar la opción de Exigir directivas de contraseña (Enforce password policy), no es recomendable hacerlo, pero para efectos de práctica lo haremos.



7. Seleccionar en base de datos predeterminada (Default database), la base de datos Alumnos_SuCarnet

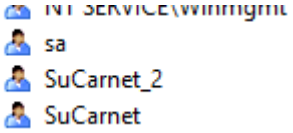
8. Hacer clic en Aceptar (Ok).

Con Transact SQL

9. En el Editor de consultas, escriba el siguiente comando Transact-SQL

```
CREATE LOGIN SuCarnet_2 WITH PASSWORD = '12345'
```

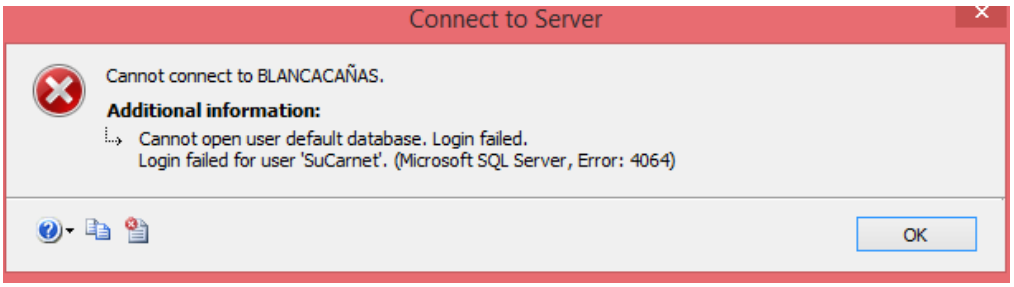
10. Verifique que se crearon los inicios de sesión



Creación de usuarios de base de datos

1. Conectarse al servidor de SQL Server e iniciar sesión con el login **SuCarnet**

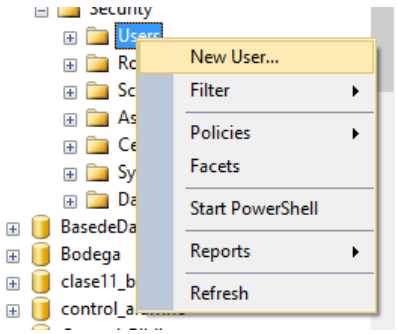
2. Digitar el nombre de Login y contraseña y mostrará un mensaje de error el cual indica que no se tiene acceso a la base de datos, para tener acceso debemos crear usuarios de BD.



3. Cambiarse al usuario **sa**

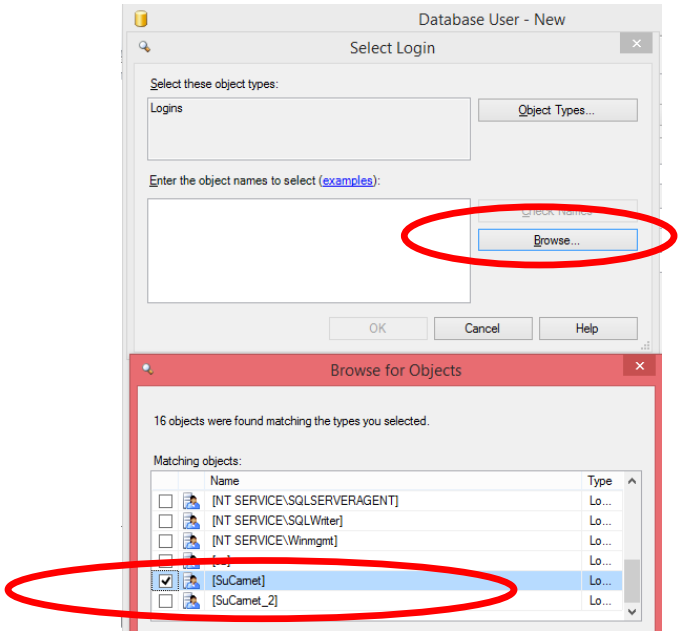
4. Expandir la base de datos en la que se quiere crear el usuario, en esta práctica será **Alumnos_SuCarnet**

5. En la carpeta Seguridad (Security), seleccionar la carpeta Usuarios (Users), hacer clic derecho y seleccionar la opción nuevo usuario (New User…)



6. En Nombre de usuario digitamos un nombre de usuario por ejemplo **SuNombre** (cambiar por su nombre)

7. Luego hacer clic en los puntos suspensivos al lado de Nombre de inicio de sesión (Login name), para asociar el nombre de usuario a un Inicio de sesión existente

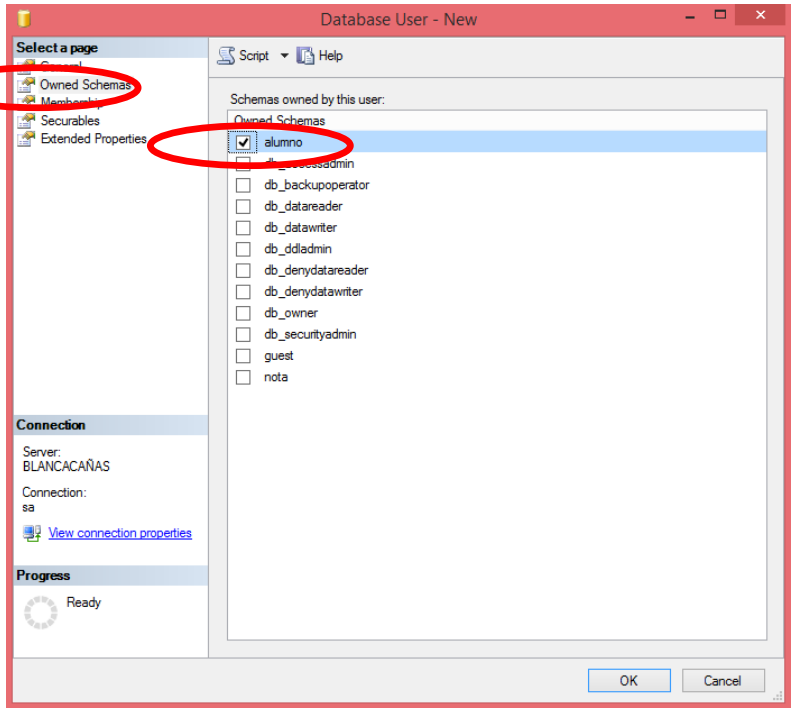


8. Hacer clic en la opción OK (en las dos ventanas)

   Esto quiere decir que cuando alguien inicie su sesión, se activarán las cuentas de la Base de Datos asociadas a este login.

9. En **esquema predeterminado** (Default schema) digitar **alumno**

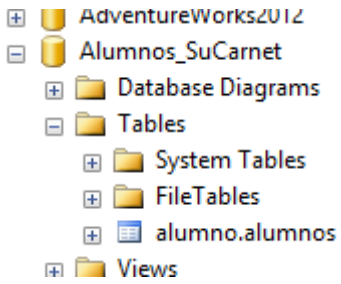
10. En la opción de menú del lado izquierdo, seleccionar la opción **Esquemas propietarios** (Owned Schemas), seleccionamos el esquema que deseamos que el usuario pueda tener acceso y control sobre él, en este caso sería alumno



11. Hacer clic en aceptar (Ok).

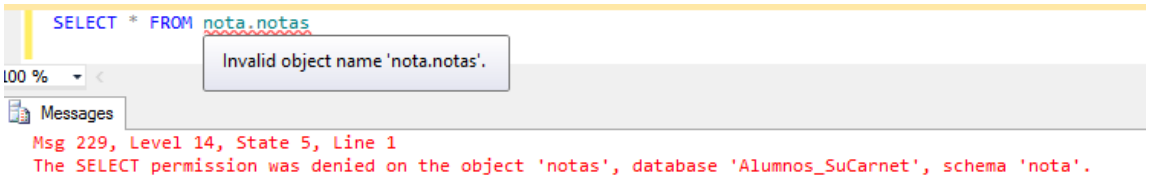
12. Ahora accedemos con el inicio de sesión que creamos para que se active el usuario de base de datos.

13. Hacer doble clic en la base de datos Alumnos_SuCarnet y verificamos a que tablas tenemos acceso



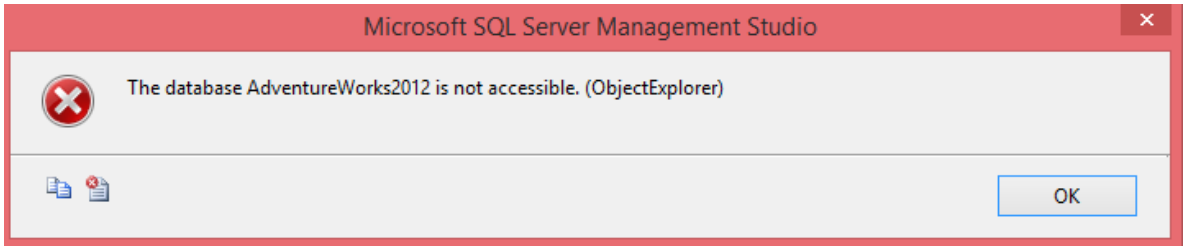
Como se observa solo se puede acceder a la tabla o tablas bajo el esquema alumno, así como se realizó en el punto 10

14. Si queremos acceder a la tabla Notas con una instrucción SELECT, SQL Server nos manhacería el siguiente mensaje de error.
Por ejemplo hacer un SELECT a la tabla nota.notas



Esto se debe a que el usuario no tiene permiso para usar la tabla nota.notas

15. Hacer doble clic en cualquier base de datos, diferente a la de Alumnos_SuCarnet, y se mostrará el siguiente mensaje



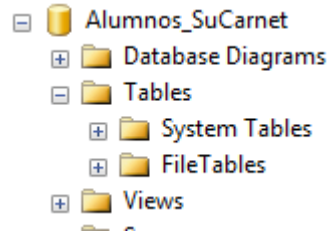
El usuario no tiene permiso para hacer uso de otra base de datos

Creación de un usuario con Transact SQL

16. Digitar la siguiente consulta

```
USE Alumnos_SuCarnet
GO
CREATE USER SuApellido FOR LOGIN SuCarnet_2
WITH DEFAULT_SCHEMA = nota
```

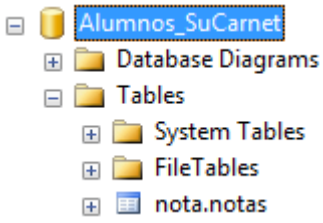
17. Ahora entrar a SQL Management Studio con el login SuCarnet_2 , password 12345, y observar que no tiene asignada ninguna tabla



18. Digitar la siguiente consulta para asignarle a un usuario permiso sobre un esquema:

```
--Asignar permiso sobre un esquema
GRANT SELECT
ON SCHEMA :: nota
TO SuApellido
WITH GRANT OPTION
GO
```

19. Actualizar la carpeta Tables de la base de datos y observar los cambios



Asignación de permisos

1. Cambiarse al usuario **sa**

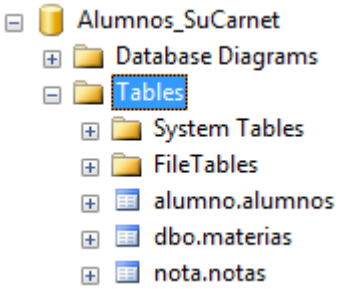
2. En la base de datos **Alumno_SuCarnet** ,digitar la consulta para crear la siguiente tabla:

```
CREATE TABLE materias
( ID INT,
  nombre VARCHAR(50)
)
GO
```

3. Agregar los siguientes registros

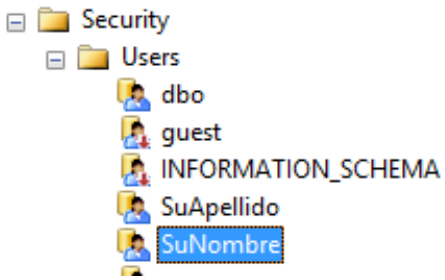
```
INSERT INTO materias VALUES(101,'Fisica')
INSERT INTO materias VALUES(102,'Programacion')
INSERT INTO materias VALUES(103,'Diseño Web')
INSERT INTO materias VALUES(104,'Algebra')
```

4. La tabla materias se crea en la cuenta dbo

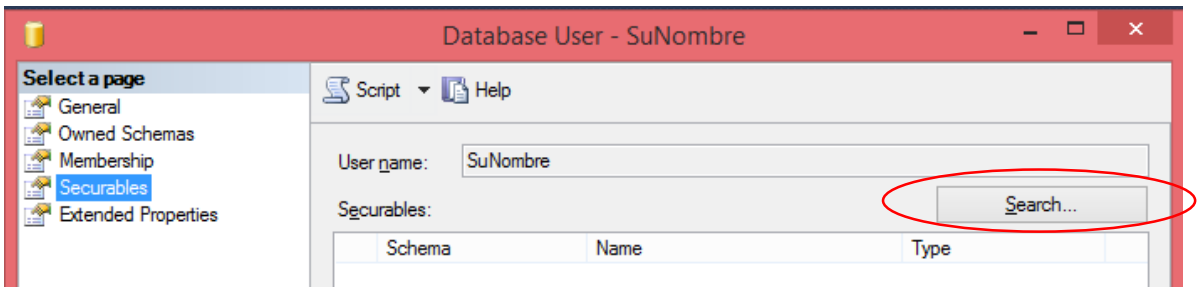


SQL Server denomina como dbo a aquellas cuentas de usuario que han creado bases de datos en la instancia así como también a aquellos usuarios que pertenecen al rol “sysadmin”.

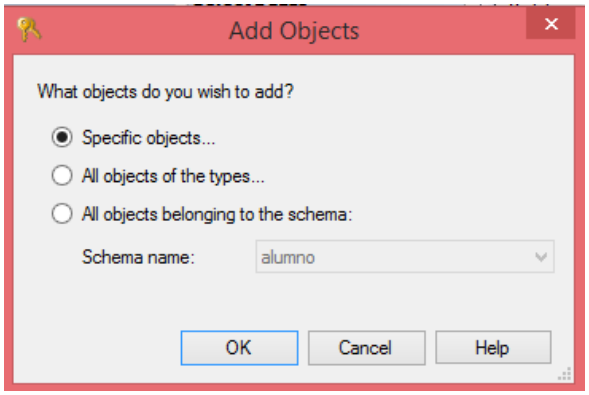
5. Hacer doble clic en la carpeta seguridad (security) de la base de datos Alumnos_SuCarnet, hacer doble clic sobre el usuario de base de datos que se creó anteriormente



6. En la parte izquierda de la pantalla seleccionar la opción **Securables,** hacer clic en la opción **Search…**



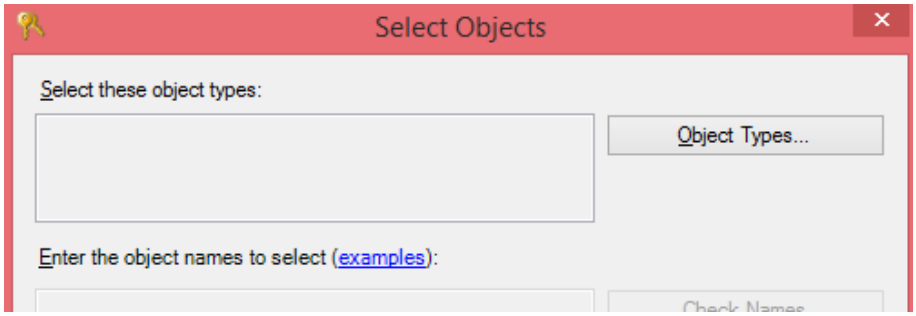
7. Se abre la siguiente ventana



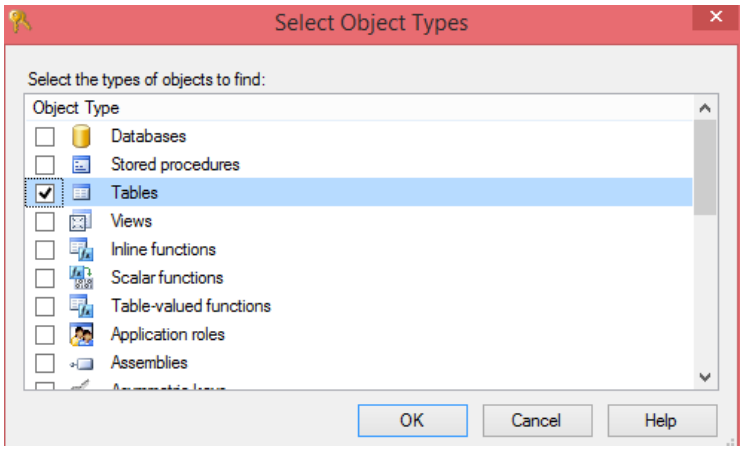
La cual posee las siguientes opciones:

- **Objetos específicos (Specific objects…):** crear permisos para uno o varios objetos de SQL el usuario tiene la oportunidad de agregar los objetos de su preferencia.
- **Todos los objetos de los tipos (All objects of the types…)**: se seleccionaran todos los tipos de objetos, es decir si selecciona el objeto tabla, se seleccionaran todas las tablas que contenga la BD
- **Todos los objetos que pertenecen al esquema (All objects belonging to the schema):** para seleccionar los objetos que pertenezcan a un esquema en específico.

8. En esta ocasión el ejercicio es solo asignar ciertos permisos a la tabla que se acaban de crear, como solo se quiere tabla seleccionar la primera opción, **objetos específicos (Specific Objects)**, en la ventana hacer clic en **Tipos de objeto (Object Types…)**

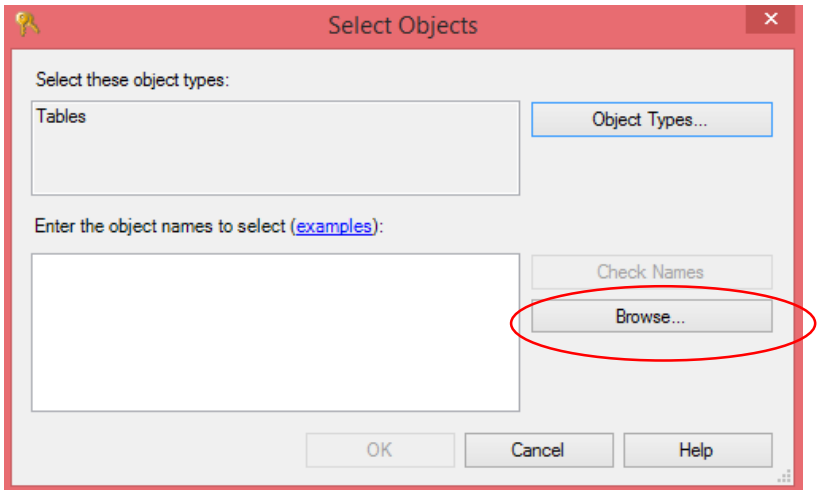


9. Seleccionar **Tablas (Tables)**

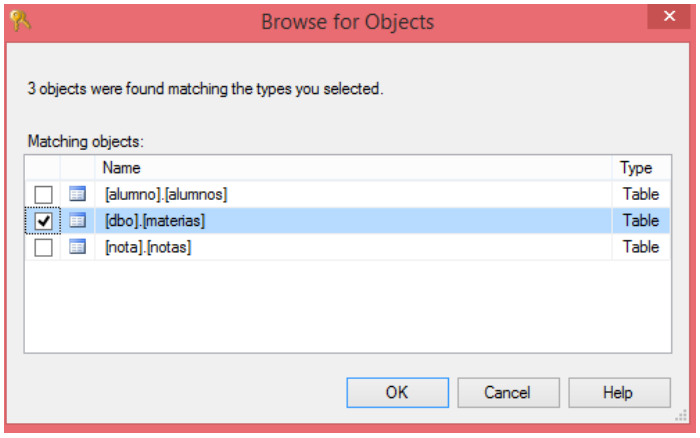


10. Hacer clic en aceptar (Ok)

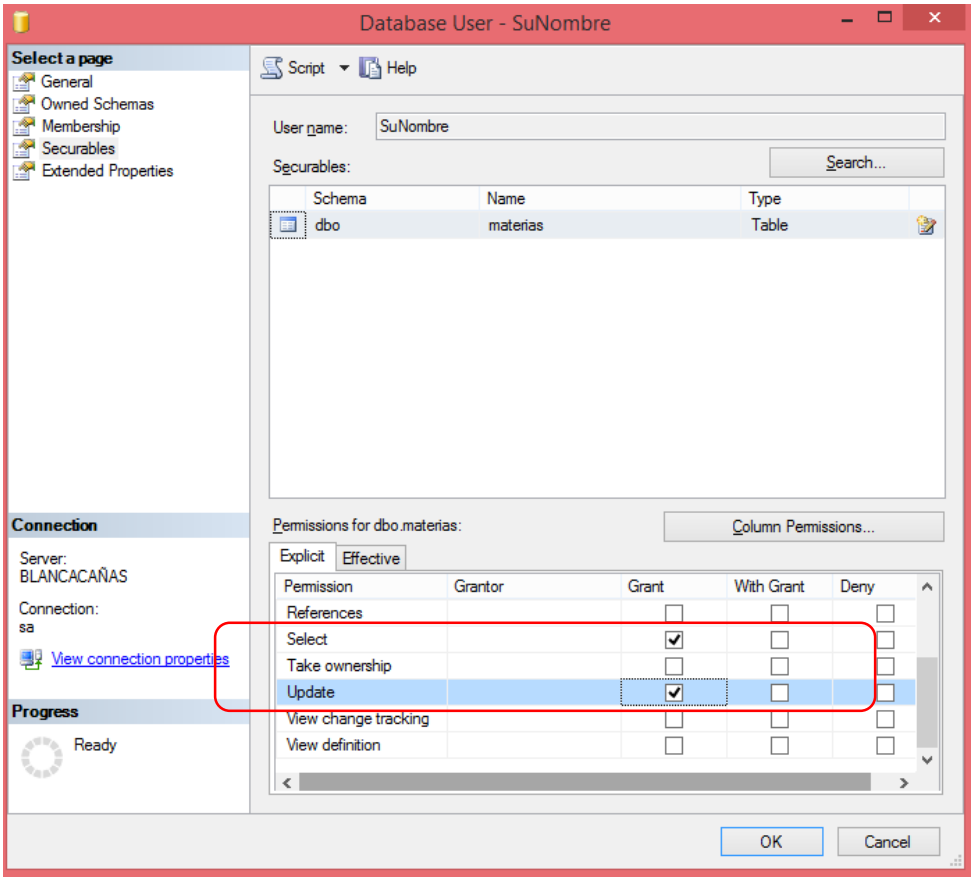
11. Hacer clic en el botón examinar (Browse…) para buscar las tablas existentes en la base de datos



12. Seleccionar **la tabla Materias**



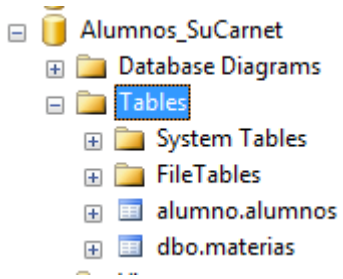
13. Hacer clic en aceptar (Ok) en dos ocasiones, y nos encontraremos con la siguiente pantalla en la cual marcamos las casillas UPDATE y SELECT, lo que quiere decir que el usuario solo podrá hacer consultas y actualizar datos.



14. Para finalizar hacer clic en aceptar (Ok)

15. Conectarse de nuevo al servidor e ingresar con el inicio de sesión (SuCarnet) que creo anteriormente para ver probar los permisos en esa tabla.

16. Observar que ahora le aparecen las dos tablas alumno.alumnos y materias



La tabla alumno.alumnos que fue asignada cuando se creó el usuario de la base de datos y dbo.materias la cual se acaba de asignar en los pasos anteriores

17. Probar las siguientes consultas SQL y colocar en el script por medio de un comentario si se obtuvieron o no resultados y porque

```
--Haciendo uso de la cláusula SELECT
SELECT * FROM dbo.materias

--Haciendo uso de la cláusula UPDATE
UPDATE dbo.materias SET nombre='Fisica Tecnica' WHERE ID=101

--Haciendo uso de la cláusula INSERT
INSERT INTO dbo.materias VALUES(105,'SQL Server')
```

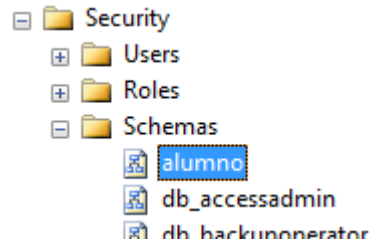
Agregando permisos a esquemas

En los ejercicios anteriores, un usuario de base de datos podía realizar consultas y actualizaciones de datos, ahora se hará que ese usuario de base de datos tenga el permiso para crear tablas las cuales quehaceran en el esquema que se le asigno cuando este se creó.

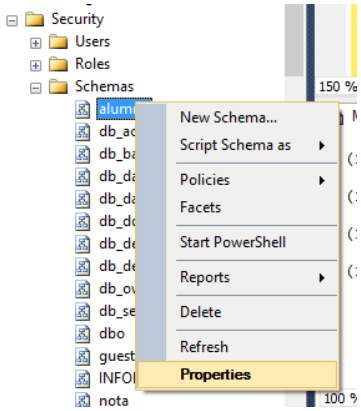
1. Iniciar sesión con el login **sa**

2. Expandir la base de datos en la que se quiere otorgar los permisos, en esta práctica será **Alumnos_SuCarnet.**

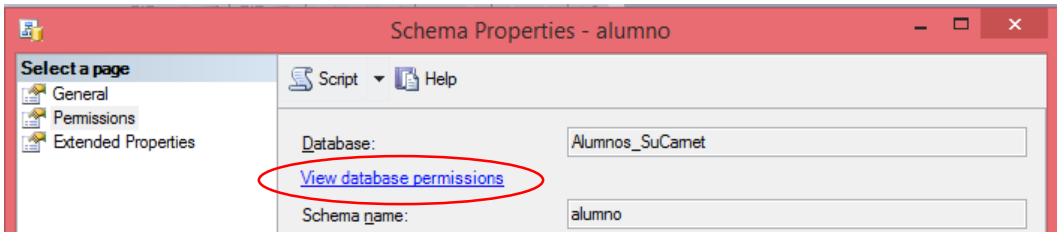
3. Expandir la carpeta seguridad (Security) y luego expandir la carpeta esquemas (Schemas)



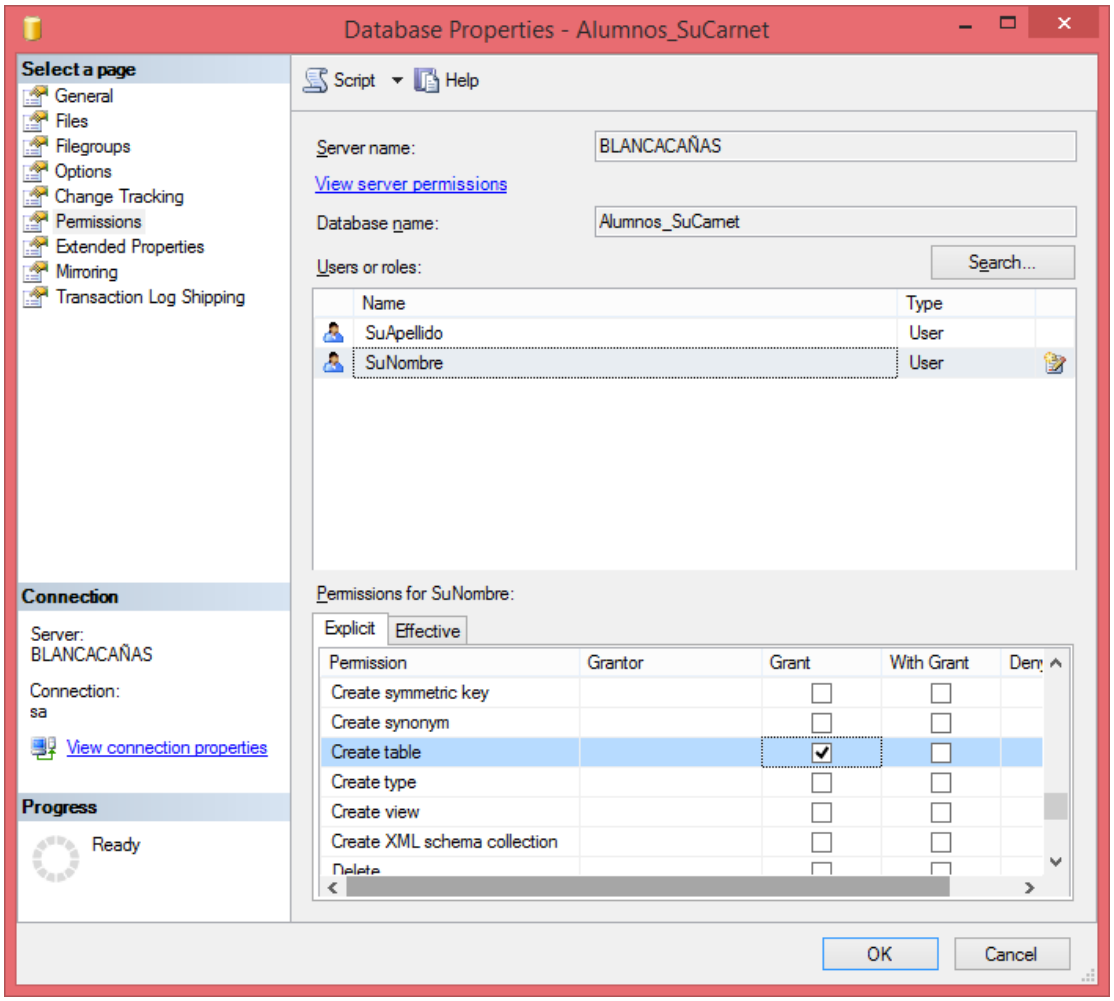
4. Hacer clic derecho sobre el esquema en el que se otorgaran permisos al usuario, en este ejercicio será al esquema **alumno** y hacer clic en propiedades (Properties)



5. En la parte izquierda hacer clic en la página de permisos (Permissions), y luego hacer clic en la opción **Ver permisos de base de datos (View database permissions)**



6. En la ventana que se abre ir a la sección de permisos explícitos que se encuentra en la parte de abajo, seleccionar los permisos que deseamos que posea el usuario (SuNombre), en este ejercicio agregamos el permiso para creación de tablas y procedimientos, hacer clic en la casilla conceder (Grant) a la par del permiso **CREATE PROCEDURE y CREATE TABLE**



7. Hacer clic en OK

8. Hacer clic en OK

9. Probar permisos, iniciar sesión con el **login SuCarnet**, ejecutar las siguientes instrucciones en la ventana de consultas.

Crear una tabla:

```
CREATE TABLE Prueba
(column1 INT NOT NULL,
 column2 CHAR(10) NOT NULL
)
```

¿Pudo crear la tabla, por qué?

Crear una vista

```
CREATE VIEW Reporte
AS
 SELECT carnet,nombres,apellidos FROM alumno.alumnos
```

¿Pudo crearlo, por qué?

7. Iniciar sesión con el login **sa** y digitar el siguiente código:

```
GRANT CREATE VIEW
TO SuNombre
```

10. Iniciar sesión con el login SuCarnet, ejecutar de nuevo la consulta de la creación de la vista ¿Se pudo crear sí, no y porque?

Eliminando inicio de sesión y usuario de Base de datos

1. Eliminar un Login utilizando Interfaz:
   - Expandir la carpeta Seguridad (Security), la cual se encuentra a nivel del servidor
   - Seleccionar el inicio de sesión (Login) que desea eliminar
   - Hacer clic derecho
   - Seleccionar eliminar (Delete)

2. Utilizando Transact-SQL, digitar la siguiente consulta

```
DROP LOGIN SuCarnet_2
```

3. Eliminar un usuario de base de datos utilizando Interfaz:
   - Expandir la carpeta Seguridad (Security), la cual se encuentra a nivel de la base de datos
   - Seleccionar el usuario (Users) que desea eliminar
   - Hacer clic derecho
   - Seleccionar eliminar (Delete)

4. Utilizando Transact-SQL, digitar la siguiente consulta

```
DROP USER SuApellido
```