Pregrado

Programa de Ingeniería de Sistemas

GESTIÓN DE DATOS E INFORMACIÓN II

Sesión 13

Tema:

Seguridad de base de datos

Resultado de aprendizaje

Maneja herramientas administrativas e implementa seguridad de una base de datos SQL Server para una organización

Evidencia de aprendizaje

Los estudiantes demostrarán sus habilidades y conocimientos en la implementación de medidas de seguridad en una base de datos SQL Server, centrándose en aspectos como la gestión de usuarios y roles, la aplicación de permisos a nivel de objetos, y la auditoría de eventos.

Contenido

Nombre del tema

- Seguridad de base de datos
- Gestión de Roles y Usuarios
- Auditoria de eventos
- Implementación de políticas de contraseñas
- Control de acceso a datos sensibles
- Gestión de permisos a nivel de sistema

Revisa el siguiente video:

https://youtu.be/h3yn79WUEb4?si=DnLO_NNQHnJhnYUe

Después de haber visualizado el video en la slide anterior, reflexionamos y respondemos las siguientes interrogantes:

01 ¿En qué consiste la seguridad de base de datos?

02 ¿Qué aspectos deben considerarse en la seguridad de la base de datos?

03 ¿Qué problemas traería en una empresa la perdida de la información?

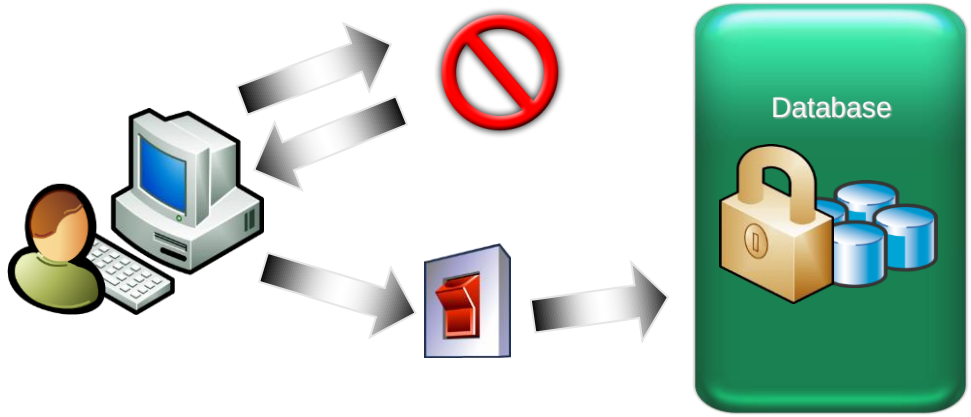
Tema

Seguridad de base de datos

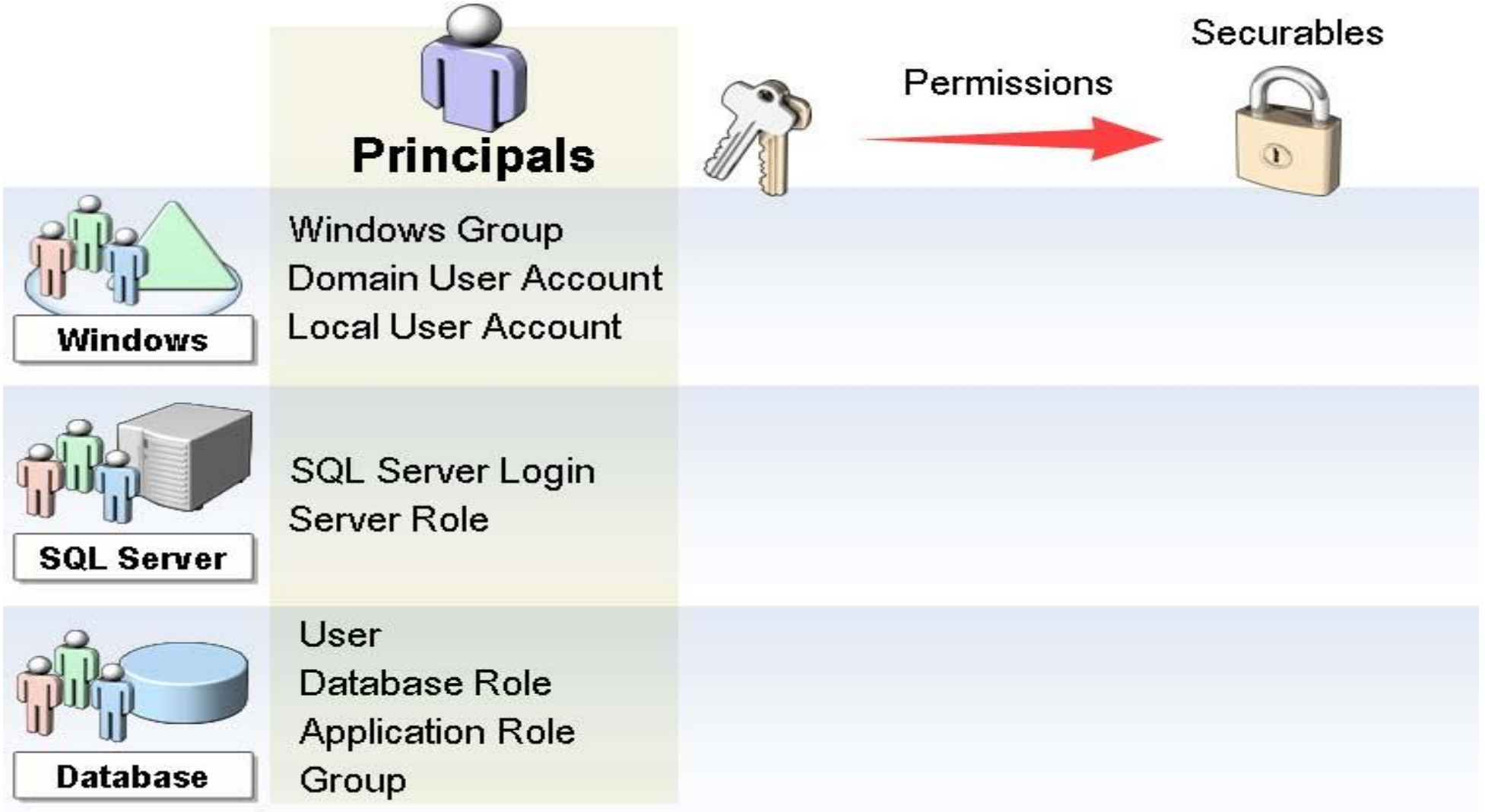
SEGURIDAD DE BASE DE DATOS

Es una parte esencial de la administración de bases de datos, especialmente en entornos empresariales donde la integridad y confidencialidad de la información son prioritarias.

En SQL Server, se implementa un robusto sistema de seguridad que abarca la gestión de usuarios, roles, permisos y auditoría de eventos.



SEGURIDAD DE BASE DE DATOS



SEGURIDAD DE BASE DE DATOS

Nivel	Principal
Windows	Cuenta de usuario local Windows Cuenta de usuario de dominio Windows Grupo Windows
SQL Server	SQL Server login SQL Server role
Base de datos	Usuario de base de datos, Rol de base de datos Grupo de base de datos, Rol de la aplicación

SEGURIDAD SQL SERVER

Son recursos a los cuales el Sistema de Autorización controla el acceso

Nivel	Principal
Servidor	Logins HTTP Encode Certificados
Base de Datos	Usuarios Roles Roles de Aplicacion Certificados, Assemblies
Schema	Tablas Vistas Funciones, Procedimientos Almacenados Reglas, Defaults

AUTENTICACIÓN DE WINDOWS

Cuando un usuario se conecta a través de una cuenta de usuario de **Microsoft Windows, SQL Server** valida el nombre de cuenta y la contraseña con el token de la entidad de seguridad de Windows del sistema operativo. Esto significa que Windows confirma la identidad del usuario. SQL Server no pide la contraseña y no realiza la validación de identidad.

La **autenticación de Windows** es el modo de autenticación predeterminado y es mucho más seguro que la autenticación de SQL Server.

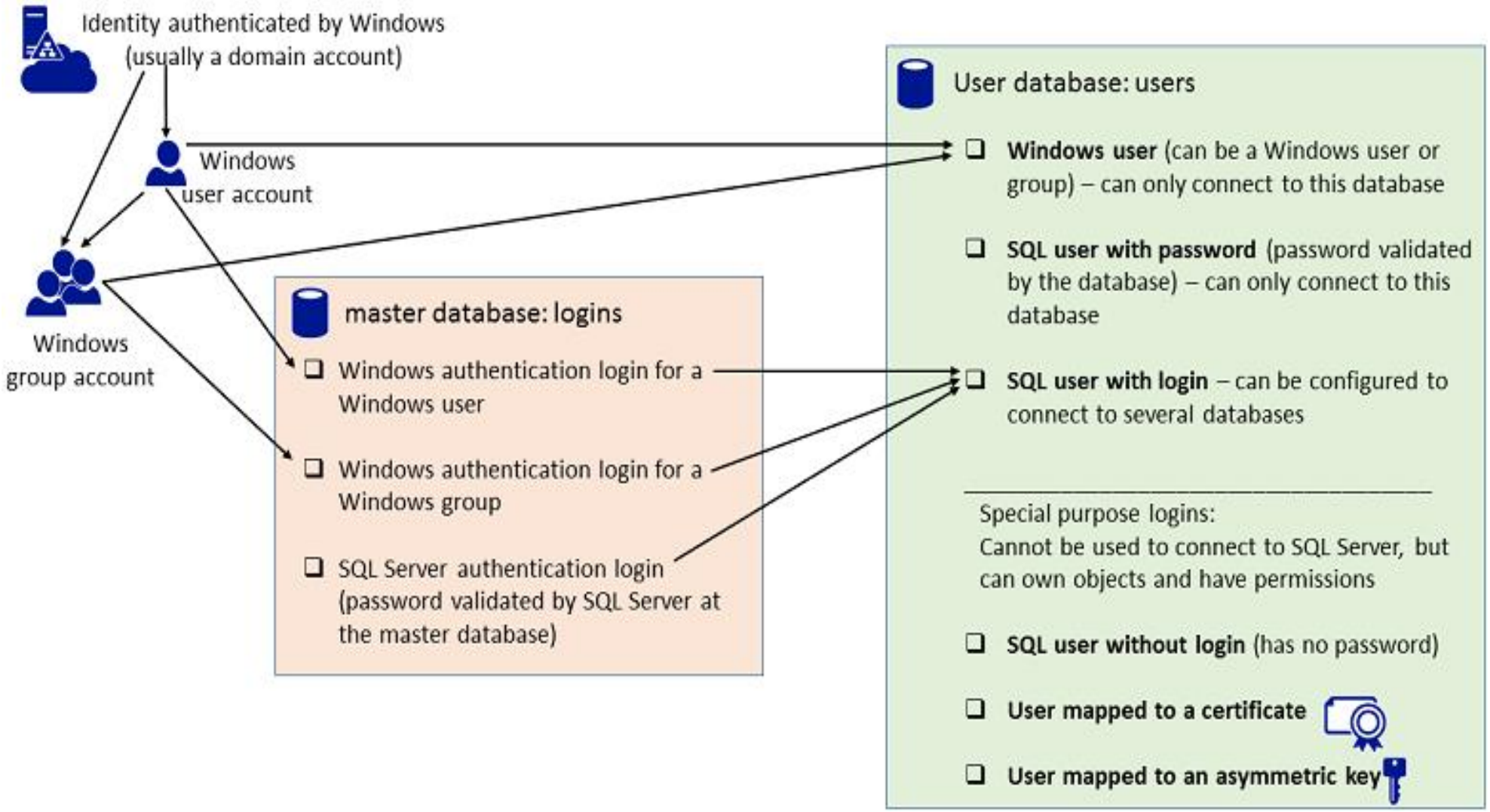
AUTENTICACIÓN DE WINDOWS

La **autenticación de Windows** usa el protocolo de seguridad de **Kerberos**, proporciona la aplicación de directivas de contraseñas en cuanto a la validación de la complejidad de las contraseñas seguras, ofrece compatibilidad para el bloqueo de cuentas y admite la expiración de las contraseñas.

Una conexión realizada utilizando la **autenticación de Windows** se denomina a veces conexión de confianza, porque SQL Server confía en las credenciales proporcionadas por Windows.

Cuando se emplea la **autenticación de Windows**, se pueden crear grupos de Windows en el nivel de dominio y se puede crear un inicio de sesión en SQL Server para todo el grupo. La administración del acceso desde el nivel de dominio puede simplificar la administración de cuentas.

AUTENTICACIÓN DE WINDOWS



AUTENTICACIÓN SQL SERVER

Cuando se utiliza la **autenticación de SQL Server** , los inicios de sesión se crean en SQL Server y no se basan en cuentas de usuario de Windows. El nombre de usuario y la contraseña se crean utilizando SQL Server y se almacenan en SQL Server.

Los usuarios que se conectan usando la **autenticación de SQL Server** deben indicar sus credenciales (inicio de sesión y contraseña) cada vez que se conectan. Al utilizar la autenticación de SQL Server , debe establecer contraseñas seguras para todas las cuentas de SQL Server .

Microsoft SQL Server crea una cuenta de administrador por defecto denominada SA. Esta cuenta dispone de privilegios de administrador completos así como de propiedad de tablas de sistema.

AUTENTICACIÓN SQL SERVER

Cuando se utiliza la **autenticación de SQL Server** , los inicios de sesión se crean en SQL Server y no se basan en cuentas de usuario de Windows. El nombre de usuario y la contraseña se crean utilizando SQL Server y se almacenan en SQL Server.

Los usuarios que se conectan usando la **autenticación de SQL Server** deben indicar sus credenciales (inicio de sesión y contraseña) cada vez que se conectan. Al utilizar la autenticación de SQL Server , debe establecer contraseñas seguras para todas las cuentas de SQL Server .

Microsoft SQL Server crea una cuenta de administrador por defecto denominada SA. Esta cuenta dispone de privilegios de administrador completos así como de propiedad de tablas de sistema.

Tema

Gestión de Roles y Usuarios

ROLES DE NIVEL DE BASE DE DATOS

Para administrar con facilidad los permisos en las bases de datos, **SQL Server** proporciona varios roles , que son las entidades de seguridad que agrupan a otras entidades de seguridad. Son como los grupos del sistema operativo Microsoft Windows.

Los **roles de nivel de base de datos** se aplican a toda la base de datos en lo que respecta a su ámbito de permisos.

ROLES DE NIVEL DE BASE DE DATOS

Para administrar con facilidad los permisos en las bases de datos, **SQL Server** proporciona varios roles , que son las entidades de seguridad que agrupan a otras entidades de seguridad. Son como los grupos del sistema operativo Microsoft Windows.

Los **roles de nivel de base de datos** se aplican a toda la base de datos en lo que respecta a su ámbito de permisos.

ROLES DE NIVEL DE BASE DE DATOS

db_owner Los miembros del rol fijo de base de datos db_owner pueden realizar todas las actividades de configuración y mantenimiento en la base de datos y también pueden quitar la base de datos en SQL Server. (En SQL Database y Azure Synapse, algunas actividades de mantenimiento requieren permisos de nivel de servidor y los roles db_owners no las pueden realizar).

db_securityadmin Los miembros del rol fijo de base de datos db_securityadmin pueden modificar la pertenencia a roles únicamente para roles personalizados y administrar permisos. Los miembros de este rol pueden elevar potencialmente sus privilegios y se deben supervisar sus acciones.

ROLES DE NIVEL DE BASE DE DATOS

db_accessadmin Los miembros del rol fijo de base de datos db_accessadmin pueden agregar o quitar el acceso a la base de datos para inicios de sesión de Windows, grupos de Windows e inicios de sesión de SQL Server .

db_backupoperator Los miembros del rol fijo de base de datos db_backupoperator pueden crear copias de seguridad de la base de datos.

db_ddladmin Los miembros del rol fijo de base de datos db_ddladmin pueden ejecutar cualquier comando del lenguaje de definición de datos (DDL) en una base de datos.

ROLES DE NIVEL DE BASE DE DATOS

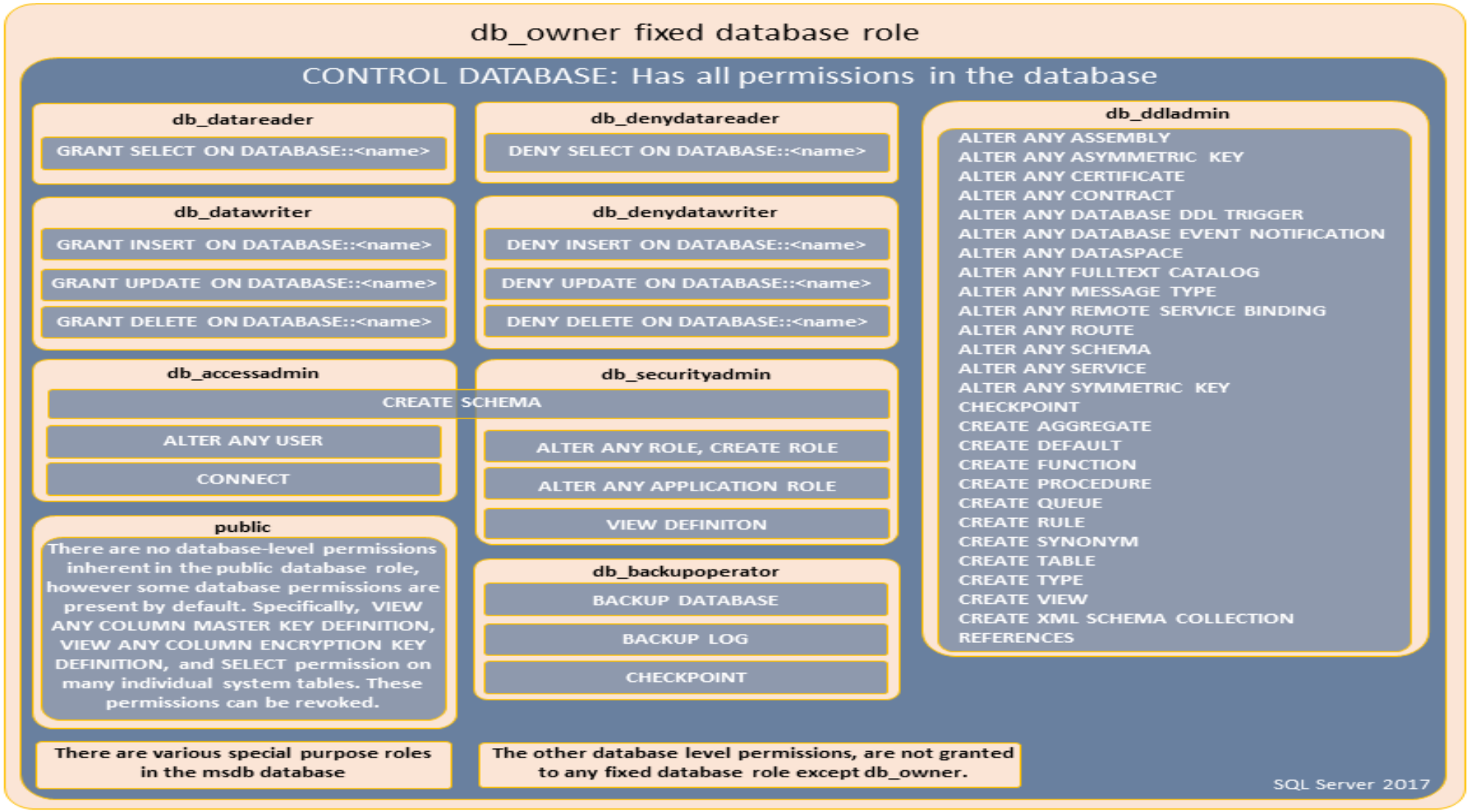
db_datawriter Los miembros del rol fijo de base de datos db_datawriter pueden agregar, eliminar o cambiar datos en todas las tablas de usuario.

db_datareader Los miembros del rol fijo de base de datos db_datareader pueden leer todos los datos de todas las tablas y vistas de usuario. Los objetos de usuario pueden existir en cualquier esquema, excepto sys e INFORMATION_SCHEMA.

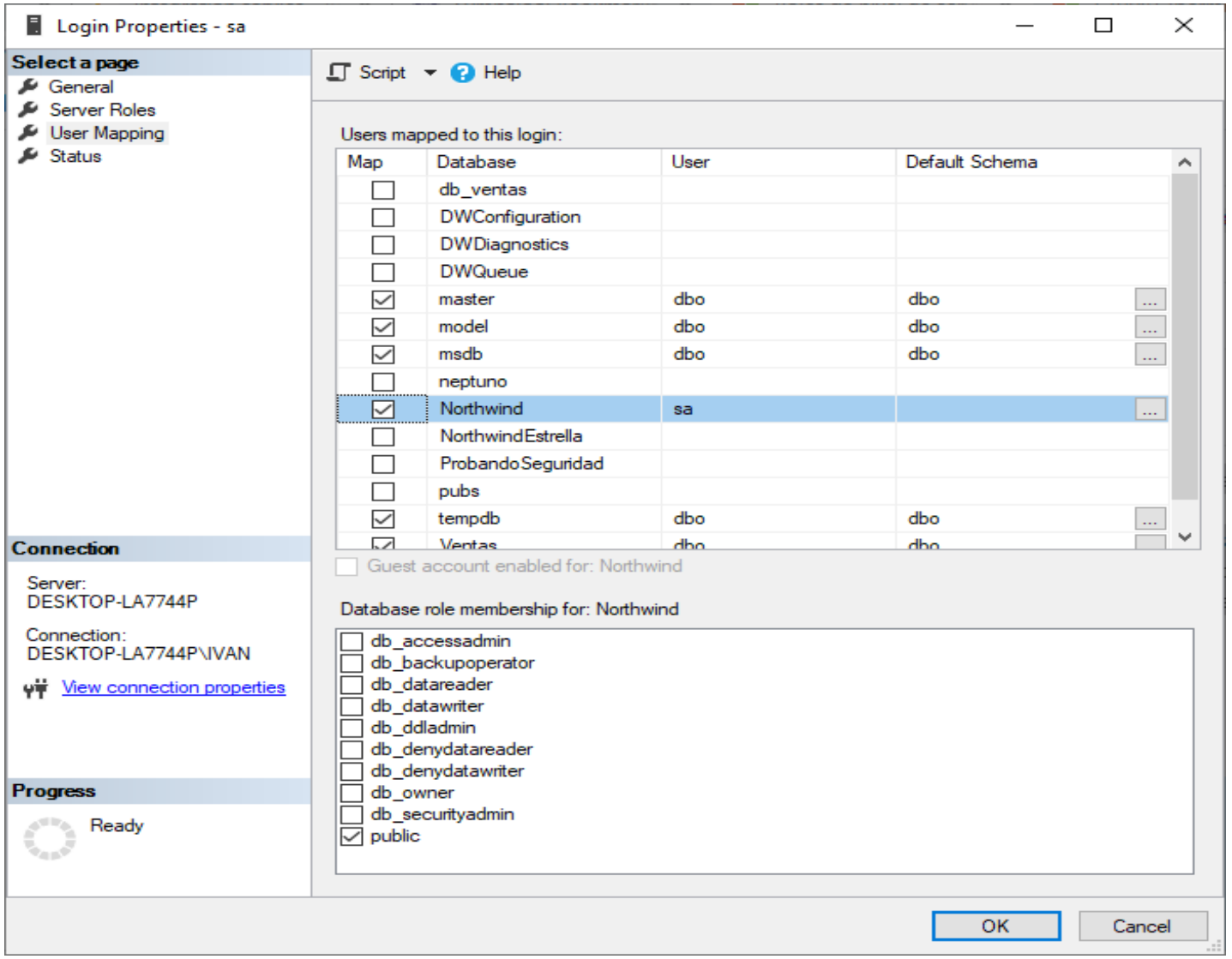
db_denydatawriter Los miembros del rol fijo de base de datos db_denydatawriter no pueden agregar, modificar ni eliminar datos de tablas de usuario de una base de datos.

db_denydatareader Los miembros del rol fijo de base de datos db_denydatareader no pueden leer datos de las tablas y vistas de usuario dentro de una base de datos.

ROLES DE NIVEL DE BASE DE DATOS



ROLES DE NIVEL DE BASE DE DATOS



CUENTA DE USUARIO DBO

El **dbo** o **base de datos owner**, es una cuenta de usuario que tiene permisos implícitos para realizar todas las actividades en la base de datos. Los miembros del **sysadmin** rol de servidor fijo se asignan automáticamente a dbo.

La **dbo** cuenta de usuario se confunde con frecuencia con el **db_owner** rol fijo de la base de datos. El alcance de **db_owne**r es una base de datos; el alcance de sysadmin es todo el servidor. La pertenencia al db_**owner** rol no confiere dbo privilegios de usuario.

PERMISOS SQL SERVER

Asegurable	Permiso	Descripción
Server	CONNECT_SQL	Conecta al servidor.
	CREATE LOGIN	Crea un login.
	ALTER ANY LOGIN	Altera cualquier login en el rango del servidor.
	CONTROL SERVER	Control completo de administración del sistema.
Login	ALTER	Altera el login.
	IMPERSONATE	Impersonar el login.
Base de Datos	CREATE TABLE	Crea una tabla en la base de datos.
	ALTER ANY USER	Altera cualquier usuario en la base de datos.
	CONTROL	Control completo de la base de datos.
Usuario	ALTER	Altera el usuario especificado.
Schema	SELECT	Selecciona rows para cualquier objeto del schema.
	ALTER	Altera cualquier objeto en el schema.
	TAKE OWNERSHIP	Toma propiedad del schema.
Tabla	SELECT	Selecciona rows de la tabla.
	ALTER	Altera la tabla.
	CONTROL	Control total de la tabla.

GESTIÓN DE USUARIOS

Usuarios: En SQL Server, un usuario representa a una entidad (usuario o aplicación) que puede autenticarse en el servidor.

Por ejemplo, para crear un usuario llamado 'UsuarioEjemplo':

```
CREATE LOGIN UsuarioEjemplo WITH PASSWORD = 'ContraseñaSegura';
CREATE USER UsuarioEjemplo FOR LOGIN UsuarioEjemplo;
```

GESTIÓN DE ROLES

Roles: Los roles agrupan usuarios y simplifican la asignación de permisos.

Ejemplo de creación de un rol y asignación de un usuario:

```
CREATE ROLE RolEjemplo;
EXEC sp_addrolemember 'RolEjemplo', 'UsuarioEjemplo';
```

ASIGNACIÓN DE PERMISOS A NIVEL DE OBJETOS

SQL Server permite asignar permisos específicos a objetos como tablas, vistas o procedimientos almacenados.

Por ejemplo, otorgar permisos **SELECT** a un usuario en una tabla:

```
GRANT SELECT ON dbo.TablaEjemplo TO UsuarioEjemplo;
```

EJEMPLOS

CREACIÓN DE LOGIN

```
CREATE LOGIN [LoginFinanciera] WITH PASSWORD = N'Protect2020'
GO
CREATE LOGIN [LoginFinanciera] WITH PASSWORD = N'Protect2020',
DEFAULT_DATABASE=[FinancieraDemoBD]
-- creación de login desde Windows
CREATE LOGIN [LoginFinanciera] FROM WINDOWS WITH
DEFAULT_DATABASE=[FinancieraDemoBD]
```

ASIGNACIÓN DE ROLES A LOGIN

```
-- Agregar rol dbcreator al login
EXEC sp_addsrvrolemember 'LoginFinanciera', 'dbcreator'
```

EJEMPLOS

CREACIÓN DE USUARIO DE BASE DE DATOS

```
USE FinancieraDemoBD
GO
CREATE USER [UsuarioBDFinanciera] FOR LOGIN [LoginFinanciera]
GO
-- conceder permisos al usuario para que realice Select sobre una
tabla de la BD.
GRANT SELECT ON dbo.CLIENTE TO UsuarioBDFinanciera
GO
```

ELIMINAR LOGIN Y USUARIO DE BASE DE DATOS

```
-- Eliminar Login y Usuario
DROP LOGIN LoginFinanciera
DROP USER UsuarioBDFinanciera
```

EJEMPLOS

CREACIÓN DE LOGIN Y USUARIO DE BASE DE DATOS, PREVIA VERIFICACIÓN DE EXISTENCIA

```
-- CREACION DE LOGIN
USE [master]
GO
IF NOT EXISTS (SELECT name FROM sys.server_principals WHERE name = N'LoginFinanciera')
CREATE LOGIN [LoginFinanciera] WITH PASSWORD = N'Protect2020',
DEFAULT_DATABASE=[FinancieraDemoBD]
-- CREACION DE USUARIO DE BASE DE DATOS
USE [FinancieraDemoBD]
GO
IF NOT EXISTS(SELECT name FROM sys.sysusers WHERE name = 'UsuarioBDFinanciera')
CREATE USER [UsuarioBDFinanciera] FOR LOGIN [LoginFinanciera]
GO
```

EJEMPLOS

ASIGNACIÓN DE ROLES A USUARIO DE BASE DE DATOS Y OTORGAR PERMISOS A OBJETOS: PROCEDIMIENTOS ALMACENADOS, TABLAS

```
-- ASIGNACION DE PERMISOS
EXEC sp_addrolemember N'db_datareader', N'UsuarioBDFinanciera'
GO
EXEC sp_addrolemember N'db_datawriter', N'UsuarioBDFinanciera'
GO
GRANT EXECUTE TO [UsuarioBDFinanciera]
--DAR PERMISOS AL USUARIO DE BASE DE DATOS SOBRE PROCEDIMIENTOS ALMACENADOS Y TABLAS
USE [FinancieraDemoBD]
GO
GRANT EXECUTE ON [spListar_Clientes] TO [UsuarioBDFinanciera];
GRANT EXECUTE ON [spInsertar_Cliente] TO [UsuarioBDFinanciera];
GRANT EXECUTE ON [spListar_Proveedor] TO [UsuarioBDFinanciera];
GRANT EXECUTE ON [spInsertar_Proveedor] TO [UsuarioBDFinanciera];
GRANT SELECT,INSERT,UPDATE,DELETE ON [CLIENTE] TO [UsuarioBDFinanciera];
GRANT SELECT,INSERT,UPDATE,DELETE ON [PROVEEDOR] TO [UsuarioBDFinanciera];
GRANT SELECT,INSERT,UPDATE,DELETE ON "dbo"."COLABORADOR" TO "UsuarioBDFinanciera";
-- PERMISO ESPECIAL --> EVALUAR CONSIDERACIONES DE SEGURIDAD
GRANT ALTER ON "dbo"."CLIENTE" TO "UsuarioBDFinanciera";
GO
```

EJEMPLOS

CONSULTAR LOGIN Y USUARIOS DE BASE DE DATOS

```
-- Ver los inicios de Sesion
select * from sys.server_principals

-- *** Ver los Inicios de Sesión de la Instancia de SQL Server ***
select sid, name, dbname, password, loginname
from master..syslogins

-- *** Ver los Usuarios de la Base de Datos actual de SQL Server ***
use FinancieraDemoBD
GO
select uid, name, sid, *
from sysusers
where islogin=1
select * from sys.database_principals
```

Tema

Auditoria de eventos

AUDITORIA DE EVENTOS

La auditoría de eventos registra actividades en la base de datos.

Configuración de la auditoría para rastrear eventos de inicio de sesión:

```
CREATE SERVER AUDIT AuditoriaInicioSesion
TO FILE (FILEPATH = 'C:\Auditoria\', MAXSIZE = 100 MB);
ALTER SERVER AUDIT AuditoriaInicioSesion WITH (STATE = ON);
```

Tema

Implementación de Políticas de Contraseña

IMPLEMENTACIÓN DE POLITICAS DE CONTRASEÑA

SQL Server permite definir políticas de contraseña para mejorar la seguridad.

Creación de una política de contraseña que requiere caracteres especiales y caducidad:

```
ALTER LOGIN UsuarioEjemplo
WITH PASSWORD = 'NuevaContraseña'
CHECK_POLICY = ON,
CHECK_EXPIRATION = ON;
```

Tema

Control de Acceso a Datos Sensibles

CONTROL DE ACCESO A DATOS SENSIBLES

Para proteger datos sensibles, se pueden utilizar técnicas como la encriptación de columnas.

Ejemplo de encriptación de una columna con Always Encrypted:

```
ALTER TABLE dbo.TablaSensible ADD ColumnaEncriptada varchar(100)
ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = LlaveEncriptacion, ENCRYPTION_TYPE
= Determinista,
ALGORITHM = AEAD_AES_256_CBC_HMAC_SHA_256);
```

Tema

Gestión de permisos a nivel de sistema

GESTIÓN DE PERMISOS A NIVEL DE SISTEMA

La gestión de permisos a nivel de sistema garantiza que solo los usuarios autorizados puedan administrar el servidor.

Ejemplo de asignación de permisos al usuario 'UsuarioEjemplo':

```
GRANT ALTER ANY LOGIN TO UsuarioEjemplo;
```

Autoevaluación

Sesión 13

Pregunta 1

¿Cómo se crea un usuario en SQL Server y cuál es la relación entre un usuario y un login?

- CREATE USER; un usuario está vinculado directamente a un login.
- CREATE LOGIN; un usuario es independiente de un login.
- CREATE USER; un usuario es lo mismo que un login.
- Ninguna.

Pregunta 2

¿Cuál es la función de la instrucción GRANT en SQL Server?

- Conceder permisos a un usuario o rol.
- Crear un nuevo objeto en la base de datos.
- Eliminar un objeto existente.
- Ninguna

Pregunta 3

¿Qué tipo de eventos se pueden rastrear mediante la auditoría de eventos en SQL Server?

- ◯ Solo eventos de inicio de sesión.
- ◯ Cualquier evento relacionado con la base de datos.
- ◯ Eventos exclusivos de administración del servidor.
- ◯ Ninguna

Pregunta 4

¿Cómo se habilita una política de contraseña para un usuario en SQL Server?

- ALTER LOGIN; estableciendo CHECK_POLICY en ON.
- CREATE USER; definiendo una política de contraseña.
- GRANT PERMISSIONS; configurando CHECK_PASSWORD_POLICY en YES.
- Ninguna

Autoevaluación

¡Vamos por más logros!

¡Felicitaciones!

Ha concluido la autoevaluación

Conclusiones

La gestión efectiva de usuarios y roles en **SQL Server** es fundamental para garantizar que cada entidad tenga acceso solo a los recursos necesarios. La vinculación entre usuarios y logins permite un control preciso sobre la autenticación y autorización.

La asignación cuidadosa de permisos a nivel de objetos es esencial para controlar el acceso a datos y funciones específicas en la base de datos.

La **auditoría de eventos en SQL** Server ofrece una herramienta poderosa para monitorear actividades en la base de datos. Rastrear eventos como inicios de sesión y cambios en datos sensibles permite mantener la integridad de la información y cumplir con regulaciones de seguridad y privacidad.

La implementación de **políticas de contraseña** fortalece la seguridad al establecer estándares para contraseñas robustas. La capacidad de establecer requisitos de complejidad y períodos de caducidad contribuye a prevenir accesos no autorizados y proteger la información sensible.

La **asignación de permisos a nivel de sistema** garantiza que solo usuarios autorizados puedan realizar tareas de administración del servidor. La correcta asignación de roles y permisos evita posibles riesgos asociados con la manipulación no autorizada de configuraciones del servidor.

Aplicando lo aprendido:

Desarrollar la Guía de Laboratorio N°13

Referencias

CAPACHO, José y Wilson NIETO. Diseño de Bases de Datos [en línea]. Barranquilla: Universidad del Norte, 2017. ISBN 9789587418255. Disponible en: https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1690049&lang=es&site=ehost-live

WANUMEN Luis, RIVAS Edwin, Mosquera Darín. Bases de datos en SQL Server [en línea]. Bogotá: Ecoe Ediciones, 2017. ISBN 9789587715705. Disponible en: https://www.digitaliapublishing.com/a/66605

HUESO Luis. Bases de datos [en línea]. Madrid: Rama Editorial, 2014. ISBN 9788499641577.
Disponible en: https://www.digitaliapublishing.com/a/109943

PRIETO, Rafael. SGBD e instalación: administración de bases de datos (UF1469) [en línea]. Antequera : IC Editorial. ISBN 9788416433360. Disponible en: https://www.digitaliapublishing.com/a/86830

Pregrado