



“Año del Bicentenario, de la consolidación de nuestra Independencia, y de la
conmemoración de las heroicas batallas de Junín y Ayacucho”

“UNIVERSIDAD PERUANA LOS ANDES”



UPLA

FACULTAD DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

CURSO:

Base de Datos II

DOCENTE:

Mg.Raúl Fernandez Bejarano

CICLO:

V

AULA:

A1

INTEGRANTES:

Luis Ramirez Brayan Esnyder

HUANCAYO – PERU

2024

Implementación de Restricciones de Asignación de Horarios y Encriptación en SQL Server

1. Introducción

Este informe describe el proceso de implementación de restricciones de asignación de horarios para usuarios y la configuración de encriptación de datos en SQL Server. Estas medidas están destinadas a mejorar el control de acceso a los horarios de trabajo y a proteger los datos sensibles de usuarios.

2. Objetivos

- **Implementar restricciones de horarios:** Establecer reglas que permitan la asignación de horarios a los usuarios según su tipo.
- **Configurar encriptación de datos:** Proteger la confidencialidad de información sensible a través de la encriptación de columnas en la base de datos.

3. Restricciones de Asignación de Horarios

Para limitar los horarios asignables a los usuarios según su tipo, se ha creado una estructura de base de datos con dos tablas: `Usuarios` y `Horarios`. Se incluye una restricción que previene que ciertos usuarios asignen horarios fuera del rango permitido.

3.1. Diseño de Tablas

Las siguientes tablas se crearon para gestionar los usuarios y sus horarios de trabajo:

- **Tabla Usuarios:** Contiene información básica de los usuarios, incluyendo su tipo.
- **Tabla Horarios:** Define los horarios asignados a cada usuario, incluyendo el día de la semana y el rango de horas.

Definición de tablas SQL:

```
SQLQuery1.sql - D:\3GC\USUARIO (57))* X
IF OBJECT_ID('Usuarios', 'U') IS NOT NULL
    DROP TABLE Usuarios;
GO

CREATE TABLE Usuarios (
    UsuarioID INT PRIMARY KEY,
    Nombre NVARCHAR(50),
    TipoUsuario NVARCHAR(50) -- Ejemplo: 'Admin', 'Empleado', etc.
);
GO

IF OBJECT_ID('Horarios', 'U') IS NOT NULL
    DROP TABLE Horarios;
GO

CREATE TABLE Horarios (
    HorarioID INT PRIMARY KEY,
    UsuarioID INT FOREIGN KEY REFERENCES Usuarios(UsuarioID),
    DiaSemana NVARCHAR(20), -- Ejemplo: 'Lunes', 'Martes', etc.
    HoraInicio TIME,
    HoraFin TIME
);
GO

100 %
Messages
Commands completed successfully.

Completion time: 2024-11-14T11:07:25.8331060-05:00
```

3.2. Insertar Datos de Ejemplo

Para visualizar el contenido de las tablas, se insertaron registros de ejemplo en las tablas *Usuarios* y *Horarios*.

```
-- Insertar datos en Usuarios
INSERT INTO Usuarios (UsuarioID, Nombre, TipoUsuario)
VALUES (1, 'Juan Perez', 'Admin'),
       (2, 'Maria Lopez', 'Empleado'),
       (3, 'Carlos Gomez', 'Empleado');

-- Insertar datos en Horarios
INSERT INTO Horarios (HorarioID, UsuarioID, DiaSemana, HoraInicio, HoraFin)
VALUES (1, 2, 'Lunes', '08:00', '17:00'),
       (2, 2, 'Martes', '09:00', '18:00'),
       (3, 3, 'Miercoles', '08:00', '16:00');

00 %
Messages
(3 rows affected)
(3 rows affected)

Completion time: 2024-11-14T11:14:20.2006257-05:00
```

3.3. Visualización de las Tablas

Para consultar y visualizar los datos en las tablas `Usuarios` y `Horarios`, se utilizarán las siguientes consultas:

```
-- Mostrar todos los usuarios
SELECT * FROM Usuarios;

-- Mostrar todos los horarios asignados a los usuarios
SELECT * FROM Horarios;
```

100 %

Results Messages

	UsuarioID	Nombre	TipoUsuario
1	1	Juan Perez	Admin
2	2	Maria Lopez	Empleado
3	3	Carlos Gomez	Empleado

	HorarioID	UsuarioID	DiaSemana	HorasInicio	HorasFin
1	1	2	Lunes	08:00:00.0000000	17:00:00.0000000
2	2	2	Martes	09:00:00.0000000	18:00:00.0000000
3	3	3	Miercoles	08:00:00.0000000	16:00:00.0000000

4. Encriptación de Datos en SQL Server

Para proteger datos sensibles, como los nombres de los usuarios, se implementó una encriptación a nivel de columna utilizando claves simétricas.

4.1. Creación de Clave Maestra y Certificado

```
USE master;
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'ContraseñaSegura123!';
GO

CREATE CERTIFICATE CertificadoEjemplo
WITH SUBJECT = 'Certificado para encriptar datos';
GO
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-11-14T11:20:29.8158910-05:00

4.2. Creación de Clave Simétrica para Encriptación

```
CREATE SYMMETRIC KEY ClaveSimetricaEjemplo
WITH ALGORITHM = AES_256
ENCRIPTION BY CERTIFICATE CertificadoEjemplo;
GO
```

Messages

Commands completed successfully.

Completion time: 2024-11-14T11:21:19.4579953-05:00

	UsuarioID	Nombre	TipoUsuario
1	1	Administrador	Admin
2	2	Empleado	Empleado
3	3	Empleado	Empleado

4.3. Encriptación de Datos Sensibles

```
-- Abrir la clave simétrica
OPEN SYMMETRIC KEY ClaveSimetricaEjemplo
DECRYPTION BY CERTIFICATE CertificadoEjemplo;

-- Encriptar la columna Nombre
UPDATE Usuarios
SET Nombre = ENCRYPTBYKEY(KEY_GUID('ClaveSimetricaEjemplo'), Nombre);

-- Cerrar la clave simétrica
CLOSE SYMMETRIC KEY ClaveSimetricaEjemplo;
```

Messages

(3 rows affected)

Completion time: 2024-11-14T11:21:50.5257034-05:00

4.4. Descriptación de Datos para Lectura

```
OPEN SYMMETRIC KEY ClaveSimetricaEjemplo  
DECRYPTION BY CERTIFICATE CertificadoEjemplo;  
  
SELECT UsuarioID,  
       CONVERT(NVARCHAR(50), DECRYPTBYKEY(Nombre)) AS Nombre  
FROM Usuarios;  
  
CLOSE SYMMETRIC KEY ClaveSimetricaEjemplo;
```

Results		Messages
UsuarioID	Nombre	
1	Juan Perez	
2	Maria Lopez	
3	Carlos Gomez	