UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

ASIGNATURA: Base de Datos II

ESTUDIANTE: Espejo Quispe Luis Enrique

DOCENTE: Mg. Raúl Fernandez Bejarano

CICLO: V

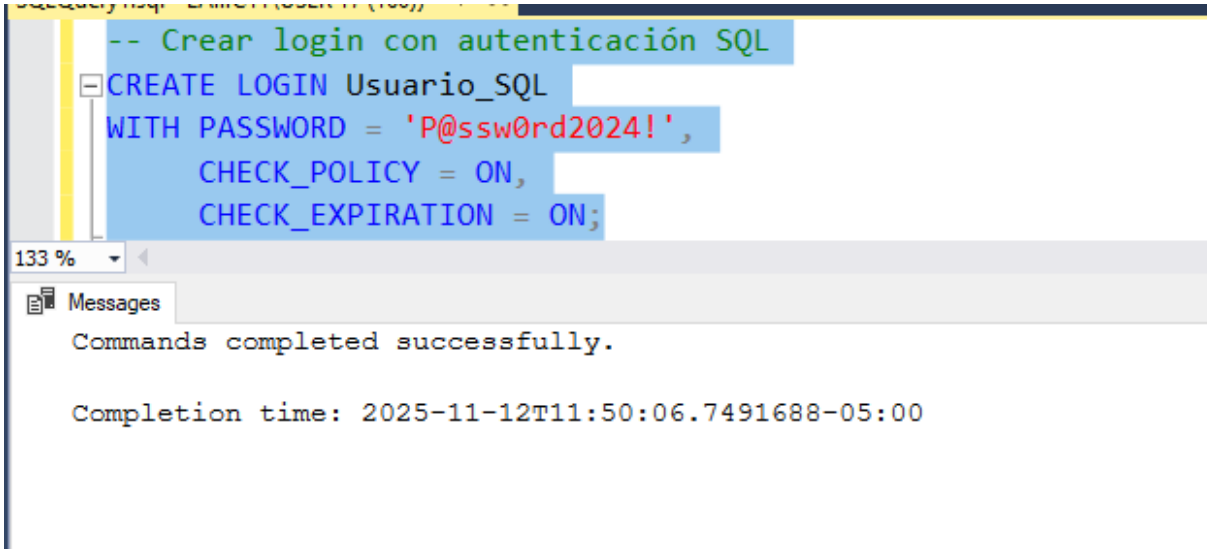
SECCIÓN: A1

HYO-2025

1. Autenticación SQL y Windows:
   La autenticación es el proceso por el cual SQL Server verifica la identidad de un usuario que intenta conectarse. Existen dos métodos principales:
   - ✓ Autenticación Windows (Recomendada): Utiliza las credenciales del sistema operativo del usuario (generalmente a través de Active Directory). El usuario es validado una vez por Windows y SQL Server confía en esa identidad. Es el método preferido porque aprovecha las políticas de seguridad centralizadas del dominio, como la complejidad y la caducidad de las contraseñas.
   - ✓ Autenticación SQL Server: Requiere un nombre de usuario y una contraseña que se almacenan directamente dentro de SQL Server. La autenticación se realiza en cada conexión.
   - ✓ Modo Mixto: Permite utilizar ambos métodos.

   Prácticas Seguras: Se debe priorizar la Autenticación Windows y utilizar Autenticación SQL Server solo cuando sea estrictamente necesario (por ejemplo, para aplicaciones que no pueden usar credenciales de Windows). En el modo SQL, se deben imponer contraseñas robustas y desactivar o renombrar la cuenta predeterminada del administrador (sa).

   SCRIPT SQL:



```sql
-- Crear login con autenticación Windows
CREATE LOGIN [LAB04-PC11\USER 19]
FROM WINDOWS;

-- Verificar logins existentes
SELECT name, type_desc, create_date, is_disabled
FROM sys.server_principals
WHERE type IN ('S', 'U', 'G');
```

121 %

Results | Messages

	name	type_desc	create_date	is_disabled
1	sa	SQL_LOGIN	2003-04-08 09:10:35.460	0
2	##MS_PolicyEventProcessingLogin##	SQL_LOGIN	2022-10-08 06:32:02.537	1
3	##MS_PolicyTsqlExecutionLogin##	SQL_LOGIN	2022-10-08 06:32:02.543	1
4	LAB04-PC14\USER 17	WINDOWS_LOGIN	2023-03-28 11:31:18.333	0
5	NT SERVICE\SQLWriter	WINDOWS_LOGIN	2023-03-28 11:31:18.337	0
6	NT SERVICE\Winmgmt	WINDOWS_LOGIN	2023-03-28 11:31:18.340	0
7	NT Service\MSSQLSERVER	WINDOWS_LOGIN	2023-03-28 11:31:18.340	0
8	BUILTIN\Usuarios	WINDOWS_GROUP	2023-03-28 11:31:18.343	0
9	NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	2023-03-28 11:31:18.343	0
10	NT AUTHORITY\NETWORK SERVICE	WINDOWS_LOGIN	2023-03-28 11:31:19.563	0
11	NT SERVICE\SQLTELEMETRY	WINDOWS_LOGIN	2023-03-28 11:31:20.650	0
12	Usuario_SQL	SQL_LOGIN	2025-11-12 11:50:06.757	0

2. Cuentas de Servicio y Configuración del Servidor
   Las cuentas de servicio son las identidades bajo las cuales se ejecutan los servicios de SQL Server (Motor de Base de Datos, SQL Server Agent, etc.) a nivel del sistema operativo Windows.
   Importancia: Los permisos que se otorgan a estas cuentas en el sistema operativo determinan lo que SQL Server puede hacer (por ejemplo, acceder a la red, leer archivos de disco). Si una cuenta de servicio tiene permisos excesivos (ej. si usa una cuenta de administrador de dominio), un atacante que comprometa el servidor SQL podría obtener control total sobre la red.
   Práctica Segura (Principio del Mínimo Privilegio): Se deben utilizar cuentas de dominio separadas y con los privilegios mínimos necesarios para cada servicio de SQL Server. Lo ideal es usar Cuentas de Servicio Administradas (MSA o gMSA), que manejan automáticamente las contraseñas.

SCRIPT SQL:

```sql
2. Cuentas de servicio y configuración del servidor
-- Crear cuenta de servicio dedicada
CREATE LOGIN [LAB04-PC11\USER 17]
FROM WINDOWS;

-- Asignar rol de servidor específico
ALTER SERVER ROLE sysadmin ADD MEMBER [LAB04-PC11\USER 17];

-- Configurar opciones de servidor
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;

-- Ver configuración actual del servidor
EXEC sp_configure;

-- Deshabilitar login remoto del SA (buena práctica)
ALTER LOGIN sa DISABLE;

-- Renombrar cuenta SA
ALTER LOGIN sa WITH NAME = [Admin_Principal];

-- Configurar política de contraseñas
ALTER LOGIN Usuario_SQL
WITH PASSWORD = 'NuevaP@ssw0rd2024!',
     CHECK_POLICY = ON,
     CHECK_EXPIRATION = ON;
```

83 %

Results | Messages

	name	minimum	maximum	config_value	run_value
1	access check cache bucket count	0	65536	0	0
2	access check cache quota	0	2147483647	0	0
3	Ad Hoc Distributed Queries	0	1	0	0
4	ADR cleaner retry timeout (min)	0	32767	15	15
5	ADR Cleaner Thread Count	1	32767	1	1
6	ADR Preallocation Factor	0	32767	4	4
7	affinity I/O mask	-2147483648	2147483647	0	0
8	affinity mask	-2147483648	2147483647	0	0
9	affinity64 I/O mask	-2147483648	2147483647	0	0
10	affinity64 mask	-2147483648	2147483647	0	0
11	Agent XPs	0	1	0	0
12	allow filesystem enumeration	0	1	1	1
13	allow polybase export	0	1	0	0
14	allow updates	0	1	0	0
15	automatic soft-NUMA disabled	0	1	0	0
16	backup checksum default	0	1	0	0
17	backup compression algorithm	0	2	0	0
18	blocked process threshold (s)	0	86400	0	0
19	c2 audit mode	0	1	0	0
20	clr enabled	0	1	0	0
21	clr strict security	0	1	1	1
22	column encryption enclave type	0	2	0	0
23	contained database authentication	0	1	0	0
24	cost threshold for parallelism	0	32767	5	5
25	cross db ownership chaining	0	1	0	0
26	cursor threshold	-1	2147483647	-1	-1
27	Data processed daily limit in TB	0	2147483647	2147483647	214748...
28	Data processed monthly limit in TB	0	2147483647	2147483647	214748...
29	Data processed weekly limit in TB	0	2147483647	2147483647	214748...
30	Database Mail XPs	0	1	0	0

Results | Messages

	name	minimum	maximum	config_value	run_value
31	default full-text language	0	2147483647	1033	1033
32	default language	0	9999	0	0
33	default trace enabled	0	1	1	1
34	disallow results from triggers	0	1	0	0
35	external scripts enabled	0	1	0	0
36	filestream access level	0	2	0	0
37	fill factor (%)	0	100	0	0
38	ft crawl bandwidth (max)	0	32767	100	100
39	ft crawl bandwidth (min)	0	32767	0	0
40	ft notify bandwidth (max)	0	32767	100	100
41	ft notify bandwidth (min)	0	32767	0	0
42	hadoop connectivity	0	8	0	0
43	hardware offload config	0	255	0	0
44	hardware offload enabled	0	1	0	0
45	hardware offload mode	0	255	0	0
46	index create memory (KB)	704	2147483647	0	0
47	in-doubt xact resolution	0	2	0	0
48	lightweight pooling	0	1	0	0
49	locks	5000	2147483647	0	0
50	max degree of parallelism	0	32767	0	0
51	max full-text crawl range	0	256	4	4
52	max server memory (MB)	128	2147483647	2147483647	214748...
53	max text repl size (B)	-1	2147483647	65536	65536
54	max worker threads	128	65535	0	0
55	media retention	0	365	0	0
56	min memory per query (KB)	512	2147483647	1024	1024
57	min server memory (MB)	0	2147483647	0	16
58	nested triggers	0	1	1	1
59	network packet size (B)	512	32767	4096	4096
60	Ole Automation Procedures	0	1	0	0

Results | Messages

	name	minimum	maximum	config_value	run_value
61	open objects	0	2147483647	0	0
62	openrowset auto_create_statistics	0	1	1	1
63	optimize for ad hoc workloads	0	1	0	0
64	PH timeout (s)	1	3600	60	60
65	polybase enabled	0	1	0	0
66	polybase network encryption	0	1	1	1
67	precompute rank	0	1	0	0
68	priority boost	0	1	0	0
69	query governor cost limit	0	2147483647	0	0
70	query wait (s)	-1	2147483647	-1	-1
71	recovery interval (min)	0	32767	0	0
72	remote access	0	1	1	1
73	remote admin connections	0	1	0	0
74	remote data archive	0	1	0	0
75	remote login timeout (s)	0	2147483647	10	10
76	remote proc trans	0	1	0	0
77	remote query timeout (s)	0	2147483647	600	600
78	Replication XPs	0	1	0	0
79	scan for startup procs	0	1	0	0
80	server trigger recursion	0	1	1	1
81	set working set size	0	1	0	0
82	show advanced options	0	1	1	1
83	SMO and DMO XPs	0	1	1	1
84	suppress recovery model errors	0	1	0	0
85	transform noise words	0	1	0	0
86	two digit year cutoff	1753	9999	2049	2049
87	user connections	0	32767	0	0
88	user instance timeout	5	65535	60	60
89	user instances enabled	0	1	1	1
90	user options	0	32767	0	0

91	version high part of SQL Server	-2147483648	2147483647	0	0
92	version low part of SQL Server	-2147483648	2147483647	0	0
93	xp_cmdshell	0	1	0	0

```
-- Configurar política de contraseñas
ALTER LOGIN Usuario_SQL
WITH PASSWORD = 'NuevaP@ssw0rd2024!',
     CHECK_POLICY = ON,
     CHECK_EXPIRATION = ON;
```

3 %

Messages

```
Commands completed successfully.

Completion time: 2025-11-12T12:06:28.4743256-05:00
```

3. Creación de Roles Fijos y Personalizados
   La gestión de permisos en SQL Server se simplifica mediante el uso de Roles, que son contenedores de permisos que agrupan a usuarios o inicios de sesión.
   - ✓ Roles Fijos de Servidor: Son roles predefinidos con permisos inalterables a nivel de servidor (ej. sysadmin, securityadmin). El rol sysadmin es el más poderoso, otorgando control completo. Solo debe asignarse a administradores esenciales.
   - ✓ Roles Fijos de Base de Datos: Son roles predefinidos con permisos a nivel de base de datos (ej. db_owner, db_datareader, db_datawriter).
   - ✓ Roles de Base de Datos Personalizados: Son creados por el administrador para agrupar usuarios que requieren un conjunto específico de permisos para realizar su trabajo. Esta es la práctica recomendada para la mayoría de los usuarios, ya que permite la segregación de funciones.

SCRIPT SQL

```
    -- 3. ROLES FIJOS DE SERVIDOR
-- Asignar rol fijo de servidor
ALTER SERVER ROLE securityadmin ADD MEMBER Usuario_SQL;

-- Roles fijos disponibles:
-- sysadmin, securityadmin, serveradmin, setupadmin,
-- processadmin, diskadmin, dbcreator, bulkadmin

USE QhatuPeru;
GO

-- 3.1. Asignar Rol Fijo de Servidor (Solo para Logins de Servidor)
-- Se otorga el permiso de administrador de procesos al Login 'Usuario_SQL_01'
ALTER SERVER ROLE processadmin ADD MEMBER Usuario_SQL;

-- 3.2. Asignar Roles Fijos de Base de Datos (Solo para Usuarios de Base de Datos)
CREATE USER Usuario_BD_Lector FOR LOGIN Usuario_SQL;
ALTER ROLE db_datareader ADD MEMBER Usuario_BD_Lector; -- Acceso de solo lectura a la BD

-- 3.3. Crear y configurar un Rol Personalizado
CREATE ROLE Rol_Gestor_Inventario;
GO

-- Asignar el nuevo rol personalizado a un usuario
ALTER ROLE Rol_Gestor_Inventario ADD MEMBER Usuario_BD_Lector;
GO
```

```
%
Messages
Commands completed successfully.

Completion time: 2025-11-12T12:47:32.1547973-05:00
```

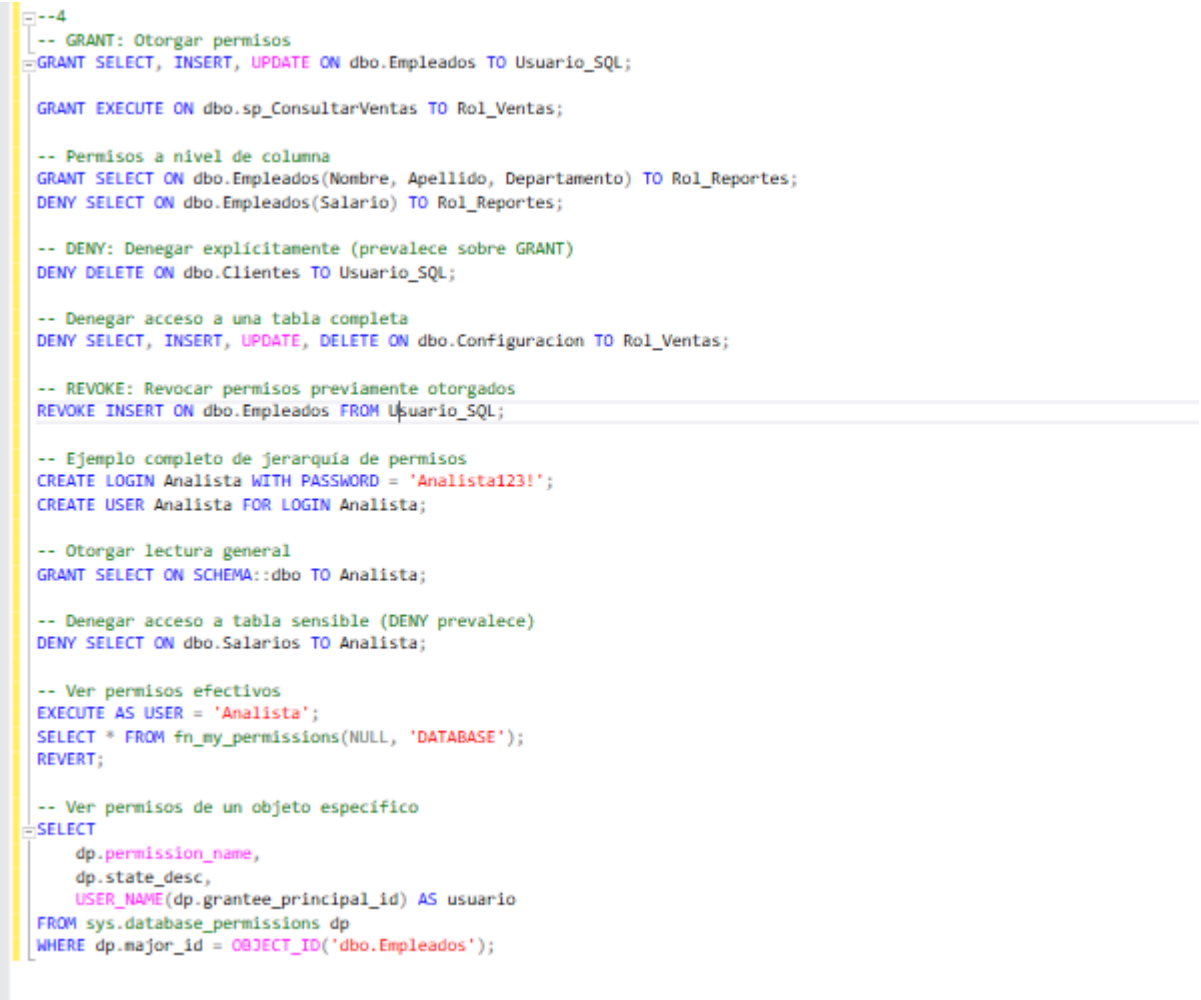
4. Control de Acceso mediante GRANT, DENY y REVOKE

Estos comandos del Lenguaje de Control de Datos (DCL) permiten especificar y modificar los permisos de los usuarios sobre los objetos de la base de datos (tablas, procedimientos, etc.).

GRANT: Otorga un permiso específico a un usuario o rol. Por ejemplo, permite leer datos de una tabla (SELECT).

DENY: Deniega explícitamente un permiso a un usuario o rol. Es una medida poderosa porque un DENY siempre tiene prioridad sobre cualquier permiso GRANT heredado de otro rol.

REVOKE: Elimina un permiso GRANT o DENY que se haya aplicado previamente. Esto devuelve el permiso al estado en el que estaba antes de que se aplicara el GRANT o DENY (por ejemplo, si el permiso se hereda de un rol, la revocación hace que la herencia vuelva a ser efectiva).



```sql
--4
-- GRANT: Otorgar permisos
GRANT SELECT, INSERT, UPDATE ON dbo.Empleados TO Usuario_SQL;

GRANT EXECUTE ON dbo.sp_ConsultarVentas TO Rol_Ventas;

-- Permisos a nivel de columna
GRANT SELECT ON dbo.Empleados(Nombre, Apellido, Departamento) TO Rol_Reportes;
DENY SELECT ON dbo.Empleados(Salario) TO Rol_Reportes;

-- DENY: Denegar explícitamente (prevalece sobre GRANT)
DENY DELETE ON dbo.Clientes TO Usuario_SQL;

-- Denegar acceso a una tabla completa
DENY SELECT, INSERT, UPDATE, DELETE ON dbo.Configuracion TO Rol_Ventas;

-- REVOKE: Revocar permisos previamente otorgados
REVOKE INSERT ON dbo.Empleados FROM Usuario_SQL;

-- Ejemplo completo de jerarquía de permisos
CREATE LOGIN Analista WITH PASSWORD = 'Analista123!';
CREATE USER Analista FOR LOGIN Analista;

-- Otorgar lectura general
GRANT SELECT ON SCHEMA::dbo TO Analista;

-- Denegar acceso a tabla sensible (DENY prevalece)
DENY SELECT ON dbo.Salarios TO Analista;

-- Ver permisos efectivos
EXECUTE AS USER = 'Analista';
SELECT * FROM fn_my_permissions(NULL, 'DATABASE');
REVERT;

-- Ver permisos de un objeto específico
SELECT
    dp.permission_name,
    dp.state_desc,
    USER_NAME(dp.grantee_principal_id) AS usuario
FROM sys.database_permissions dp
WHERE dp.major_id = OBJECT_ID('dbo.Empleados');
```

5. Cifrado y Protección de Datos (TDE, Always Encrypted)

El cifrado se utiliza para proteger los datos sensibles contra el acceso no autorizado.

TDE (Transparent Data Encryption): Cifra y descifra automáticamente los archivos de la base de datos, los archivos de log y los respaldos mientras están almacenados en el disco duro (cifrado en reposo). Es transparente para las aplicaciones, pero solo protege contra el robo físico de los archivos de la base de datos, ya que los datos se descifran en la memoria para su uso.

Always Encrypted: Esta característica está diseñada para garantizar que los datos sensibles permanezcan cifrados tanto en reposo como en tránsito. La clave de cifrado reside solo con el cliente o la aplicación, nunca en SQL Server. Esto significa que ni el administrador de la base de datos (DBA) ni el sistema operativo pueden ver los datos en texto sin formato, solo la aplicación cliente puede descifrarlos.

```
Transparent Data Encryption (TDE)
-- CONFIGURAR TDE (Cifrado Transparente de Datos)

-- Paso 1: Crear Master Key en master
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'MasterKey2024!';
GO

-- Paso 2: Crear certificado
CREATE CERTIFICATE TDE_Cert
WITH SUBJECT = 'Certificado TDE para Base de Datos';
GO

-- Paso 3: Usar la base de datos objetivo
USE MiBaseDatos;
GO

-- Paso 4: Crear clave de cifrado de base de datos
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDE_Cert;
GO

-- Paso 5: Activar TDE
ALTER DATABASE MiBaseDatos
SET ENCRYPTION ON;
GO

-- Verificar estado de cifrado
SELECT
    DB_NAME(database_id) AS DatabaseName,
    encryption_state,
    CASE encryption_state
        WHEN 0 THEN 'Sin cifrado'
        WHEN 1 THEN 'Sin cifrar'
        WHEN 2 THEN 'Cifrado en progreso'
        WHEN 3 THEN 'Cifrado'
        WHEN 4 THEN 'Cambio de clave'
        WHEN 5 THEN 'Descifrado en progreso'
        WHEN 6 THEN 'Cambio de protección'
    END AS encryption_state_desc,
    percent_complete
FROM sys.dm_database_encryption_keys;

-- Backup del certificado (MUY IMPORTANTE)
BACKUP CERTIFICATE TDE_Cert
TO FILE = 'C:\Backup\TDE_Cert.cer'
WITH PRIVATE KEY (
    FILE = 'C:\Backup\TDE_Cert_Key.pvk',
    ENCRYPTION BY PASSWORD = 'BackupKey2024!'
);
```

```
Always Encrypted (Cifrado de columnas)
-- ALWAYS ENCRYPTED (requiere configuración desde aplicación cliente)

-- Crear tabla con columnas cifradas
CREATE TABLE Empleados_Sensibles (
    EmpleadoID INT PRIMARY KEY,
    Nombre NVARCHAR(100),
    NumeroSeguroSocial NVARCHAR(11)
        ENCRYPTED WITH (
            COLUMN_ENCRYPTION_KEY = CEK_Auto,
            ENCRYPTION_TYPE = DETERMINISTIC,
            ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'
        ),
    Salario DECIMAL(10,2)
        ENCRYPTED WITH (
            COLUMN_ENCRYPTION_KEY = CEK_Auto,
            ENCRYPTION_TYPE = RANDOMIZED,
            ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'
        )
);
Cifrado a nivel de columna (tradicional)
-- Cifrado manual con funciones

-- Crear clave simétrica
CREATE SYMMETRIC KEY Clave_Empleados
WITH ALGORITHM = AES_256
ENCRYPTION BY PASSWORD = 'ClaveSegura2024!';
GO

-- Crear tabla con columna para datos cifrados
CREATE TABLE DatosSensibles (
    ID INT PRIMARY KEY,
    DatoCifrado VARBINARY(MAX)
);

-- Insertar datos cifrados
OPEN SYMMETRIC KEY Clave_Empleados
DECRYPTION BY PASSWORD = 'ClaveSegura2024!';

INSERT INTO DatosSensibles (ID, DatoCifrado)
VALUES (1, EncryptByKey(Key_GUID('Clave_Empleados'), 'Información Confidencial'));

CLOSE SYMMETRIC KEY Clave_Empleados;

-- Leer datos cifrados
OPEN SYMMETRIC KEY Clave_Empleados
DECRYPTION BY PASSWORD = 'ClaveSegura2024!';
```

```
SELECT
    ID,
    CONVERT(NVARCHAR(MAX), DecryptByKey(DatoCifrado)) AS DatoDescifrado
FROM DatosSensibles;

CLOSE SYMMETRIC KEY Clave_Empleados;
```

6. Auditoría y Monitoreo de Eventos con SQL Server Audit

La auditoría es esencial para el cumplimiento de normativas de seguridad y la detección de actividades sospechosas.

SQL Server Audit: Es la herramienta integrada para registrar y monitorear eventos a nivel de instancia y base de datos.

Función: Registra acciones importantes como inicios de sesión fallidos, cambios en los permisos, modificaciones a objetos de la base de datos (DDL) y accesos a datos (DML).

Componentes: Se define un SQL Server Audit (que especifica dónde y cómo almacenar los registros) y luego se crean Especificaciones de Auditoría de Servidor o de Base de Datos para seleccionar qué eventos específicos se deben registrar.

Ventaja: Proporciona un registro cronológico e inmutable para determinar quién realizó una acción específica, cuándo y dónde.