

PRACTICA DE LA SEMANA 11

```
-- 1A. Crear el Login SQL. Usar CHECK_POLICY=ON y CHECK_EXPIRATION=ON
-- Reemplaza 'tu_password_segura' con una contraseña real y segura.
CREATE LOGIN login_sql_alumno
    WITH PASSWORD = 'tu_password_segura',
    CHECK_POLICY = ON,
    CHECK_EXPIRATION = ON;
GO
```

creacion de login windows simulado

```
CREATE LOGIN [VICTUSHP\elvis] FROM WINDOWS;
GO
```

2. Comprobar Políticas y Forzar Expiración (Demostración)

Esta sección demuestra que las políticas están activas y cómo forzar una expiración, lo cual es parte del requisito del proyecto.

```

1 SELECT name, is_policy_checked, is_expiration_checked
2 FROM sys.sql_logins
3 WHERE name = 'login_sql_alumno';
4 GO
5
6
7 -- El script original del proyecto ya lo hace al crearlo, pero esta es la forma de forzarlo después.
8 ALTER LOGIN login_sql_alumno WITH CHECK_EXPIRATION = OFF; -- Paso temporal
9 ALTER LOGIN login_sql_alumno WITH CHECK_EXPIRATION = ON; -- Paso para forzar el estado
10 GO
11
12 -- 2C. Verificar que la expiración forzada se haya aplicado.
13 SELECT name, is_expiration_checked,
14        is_disabled
15 FROM sys.sql_logins
16 WHERE name = 'login_sql_alumno';
17 GO

```

5 % No se encontraron problemas. Línea: 1 Car.

name	is_policy_checked	is_expiration_checked
login_sql_alumno	1	1

name	is_expiration_checked	is_disabled
login_sql_alumno	1	0

xp_cmdshell es un procedimiento almacenado extendido que permite a los usuarios (con los permisos adecuados) ejecutar comandos del sistema operativo (Windows o Linux) directamente desde T-SQL. Por seguridad, debe estar deshabilitado, a menos que sea estrictamente necesario.

```

1 -- Habilitar opciones avanzadas para cambiar la configuración
2 EXEC sp_configure 'show advanced options', 1;
3 RECONFIGURE;
4 GO
5
6 -- Deshabilitar el procedimiento almacenado extendido xp_cmdshell
7 EXEC sp_configure 'xp_cmdshell', 0;
8 RECONFIGURE;
9 GO
10
11 -- (Opcional) Deshabilitar opciones avanzadas de nuevo
12 EXEC sp_configure 'show advanced options', 0;
13 RECONFIGURE;
14 GO
15
16 -- Para documentar la configuración (demostrar que está deshabilitado):
17 SELECT name, value, value_in_use
18 FROM sys.configurations
19 WHERE name = 'xp_cmdshell';

```

76 % No se encontraron problemas.

name	value	value_in_use
xp_cmdshell	0	0

El valor 1 indica que la característica para crear bases de datos contenidas está **Habilitada** a nivel de la instancia de SQL Server.

El valor NONE indica que la base de datos QhatuPeru **aún no permite** usuarios contenidos. Necesitas cambiar su configuración explícitamente.

```
1  -- 1. Revisar si la autenticación contenida está habilitada a nivel de instancia
2  SELECT name, value_in_use
3  FROM sys.configurations
4  WHERE name = 'contained database authentication';
5  GO
6
7  -- 2. Revisar si la base de datos QhatuPeru permite usuarios contenidos
8  USE master;
9  GO
10 SELECT name, containment_desc
11 FROM sys.databases
12 WHERE name = 'QhatuPeru';
13 GO
14
15
```

6 % No se encontraron problemas.

Resultados Mensajes

	name	value_in_use
1	contained database authentication	1

	name	containment_desc
1	QhatuPeru	NONE

Crear Credencial y Proxy para SQL Agent

Esta configuración permite a los Jobs (tareas programadas) de SQL Agent ejecutar comandos del OS (CmdExec) utilizando credenciales de baja prioridad (VICTUSHP\elvis) en lugar de usar la cuenta de servicio de SQL Server.

```

1  -- **IMPORTANTE:** Reemplaza 'Tu_Password_Secreta_Aqui' con la contraseña de tu usuario 'elvis'
2
3  -- A. Crear la Credencial (almacena el login y password de VICTUSHP\elvis)
4  USE master;
5  CREATE CREDENTIAL [Credencial_OS_Agente]
6      WITH IDENTITY = 'VICTUSHP\elvis',
7      SECRET = 'Tu_Password_Secreta_Aqui';
8  GO
9
10 -- B. Crear el Proxy de SQL Agent, vinculándolo a la credencial
11 EXEC msdb.dbo.sp_add_proxy
12     @proxy_name = N'Proxy_CmdExec_Seguro',
13     @credential_name = N'Credencial_OS_Agente',
14     @description = N'Proxy seguro para ejecutar comandos OS via SQL Agent Jobs.';
15 GO
16
17 -- C. Habilitar el proxy para el subsistema CmdExec (ID 3)
18 EXEC msdb.dbo.sp_grant_proxy_to_subsystem
19     @proxy_name = N'Proxy_CmdExec_Seguro',
20     @subsystem_id = 3;
21 GO

```

3

Preparación y Creación del Rol Personalizado

Si el rol ventas_readwrite o las tablas de prueba no existían, estos comandos las crean.

```

1  USE QhataPeru;
2  GO
3
4  -- 1A. Creación de las tablas de prueba (si no existen)
5  IF OBJECT_ID('GUIA_ENVIO') IS NULL
6  BEGIN
7      CREATE TABLE GUIA_ENVIO (GuiaID INT PRIMARY KEY IDENTITY(1,1), Direccion NVARCHAR(200));
8  END
9  IF OBJECT_ID('GUIA_DETALLE') IS NULL
10 BEGIN
11     CREATE TABLE GUIA_DETALLE (DetalleID INT PRIMARY KEY IDENTITY(1,1), GuiaID INT);
12 END
13 GO
14
15 -- 1B. Crear el rol de base de datos personalizado (si no existe)
16 IF DATABASE_PRINCIPAL_ID('ventas_readwrite') IS NULL
17 BEGIN
18     CREATE ROLE ventas_readwrite;
19 END
20 GO

```

5  No se encontraron problemas. 

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T08:18:32.0546560-05:00

Otorgar Permisos Granulares

Este paso otorga los permisos específicos (SELECT/INSERT/UPDATE) requeridos por el rol personalizado **ventas_readwrite**.



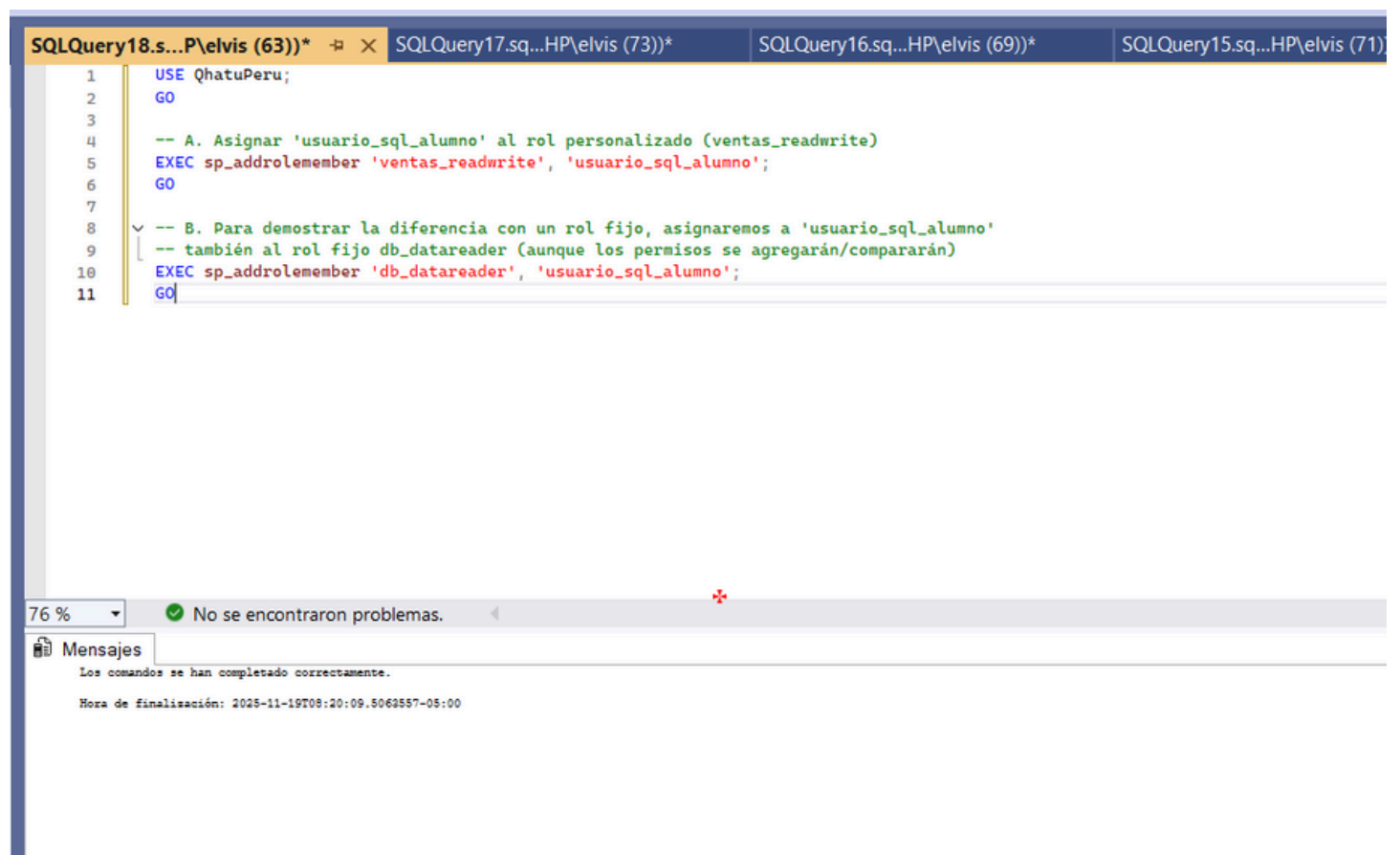
The screenshot shows a SQL query window titled 'SQLQuery3.sq...P\elvis (54))' containing the following SQL code:

```
1 USE QhatuPeru;
2 GO
3
4 -- Mostrar los permisos del rol personalizado
5 SELECT
6     dp.permission_name,
7     OBJECT_NAME(major_id) AS GrantedToObject,
8     dr.name AS GrantedToRole
9 FROM
10     sys.database_permissions dp
11 JOIN
12     sys.database_principals dr ON dp.grantee_principal_id = dr.principal_id
13 WHERE
14     dr.name = 'ventas_readwrite';
15
```

Below the query window, the 'Resultados' (Results) pane displays a table with the following data:

permission_name	GrantedToObject	GrantedToRole
INSERT	GUIA_ENVIO	ventas_readwrite
SELECT	GUIA_ENVIO	ventas_readwrite
UPDATE	GUIA_ENVIO	ventas_readwrite
INSERT	GUIA_DETALLE	ventas_readwrite
SELECT	GUIA_DETALLE	ventas_readwrite
UPDATE	GUIA_DETALLE	ventas_readwrite

Asignamos el usuario de SQL (usuario_sql_alumno) al rol personalizado. No podemos usar tu usuario de Windows (VICTUSHP\elvis) para la comparación porque es el dbo (administrador) y tiene permisos totales.



The screenshot shows a SQL query window titled 'SQLQuery18.s...P\elvis (63))' containing the following SQL code:

```
1 USE QhatuPeru;
2 GO
3
4 -- A. Asignar 'usuario_sql_alumno' al rol personalizado (ventas_readwrite)
5 EXEC sp_addrolemember 'ventas_readwrite', 'usuario_sql_alumno';
6 GO
7
8 -- B. Para demostrar la diferencia con un rol fijo, asignaremos a 'usuario_sql_alumno'
9 -- también al rol fijo db_datareader (aunque los permisos se agregarán/compararán)
10 EXEC sp_addrolemember 'db_datareader', 'usuario_sql_alumno';
11 GO
```

Below the query window, the 'Mensajes' (Messages) pane displays the following message:

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T08:20:09.5063557-05:00

Mostrar Diferencias y Documentación

Esta sección documenta la pertenencia a los roles y explica la diferencia clave requerida por el ejercicio.

```
1 USE QhatuPeru;
2 GO
3
4 -- A. Verificar la pertenencia de 'usuario_sql_alumno' a los roles para documentar
5 SELECT DP1.name AS RoleName,
6        DP2.name AS MemberName
7 FROM sys.database_role_members AS DRM
8 JOIN sys.database_principals AS DP1 ON DRM.role_principal_id = DP1.principal_id
9 JOIN sys.database_principals AS DP2 ON DRM.member_principal_id = DP2.principal_id
10 WHERE DP2.name = 'usuario_sql_alumno';
11 GO
12
13
```

76 % No se encontraron problemas.

Resultados Mensajes

	RoleName	MemberName
1	ventas_readwrite	usuario_sql_alumno
2	db_datareader	usuario_sql_alumno

4


Preparación: Creación de Objetos de Prueba


Necesitamos una tabla, un *login* y un *rol* para la demostración.

```
1 USE QhatuPeru;
2 GO
3
4 -- 1A. Crear la tabla de prueba 'Inventario' (si no existe)
5 IF OBJECT_ID('Inventario') IS NULL
6 BEGIN
7     CREATE TABLE Inventario (
8         ProductoID INT PRIMARY KEY,
9         NombreProducto NVARCHAR(100),
10        Cantidad INT,
11        PrecioProveedor MONEY, -- Columna sensible
12        PrecioVenta MONEY      -- Columna sensible
13    );
14 END
15 GO
16
17 -- 1B. Crear el rol de base de datos 'analista_inventario'
18 CREATE ROLE analista_inventario;
19 GO
20
21 -- 1C. Crear un login SQL y un usuario de base de datos para el analista
22 CREATE LOGIN login_analista WITH PASSWORD = 'P@sswordAnalista', CHECK_POLICY = ON;
23 GO
24 CREATE USER usuario_analista FOR LOGIN login_analista;
25 GO
```

Aquí aplicamos la lógica: se **otorga** la lectura a la tabla completa (GRANT SELECT) y luego se **deniega** explícitamente la lectura de las columnas sensibles (DENY SELECT).

```
1  USE QhatuPeru;
2  GO
3
4  -- 2A. Asignar el rol al usuario analista
5  EXEC sp_addrolemember 'analista_inventario', 'usuario_analista';
6  GO
7
8  -- 2B. OTORGAR SELECT a toda la tabla Inventario (para ver inventario)
9  GRANT SELECT ON Inventario TO analista_inventario;
10 GO
11
12 -- 2C. DENEGAR SELECT en las columnas de precios (DENY anula GRANT)
13 DENY SELECT ON Inventario (PrecioProveedor, PrecioVenta) TO analista_inventario;
14 GO
```

76 %  No se encontraron problemas.

 Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T08:24:52.1981552-05:00

Demostración del Control de Acceso (Prueba)

Para documentar el proyecto, ejecute esta sección para mostrar la diferencia entre una consulta permitida y una negada.

SQL

```

1  USE QhatuPeru;
2  GO
3
4  -- 3A. Simular ser el 'usuario_analista'
5  EXECUTE AS USER = 'usuario_analista';
6  GO
7
8  -- PRUEBA 1 (DEBE FUNCIONAR): Ver columnas no sensibles (Cantidad, Nombre)
9  -- Demuestra que el analista puede ver el inventario.
10 -- SELECT ProductoID, NombreProducto, Cantidad
11 -- FROM Inventario;
12 -- GO
13
14 -- PRUEBA 2 (DEBE FALLAR): Intentar ver columnas sensibles (Precios)
15 -- Esto DEBE generar un error de permisos (porque DENY tiene precedencia).
16 -- SELECT ProductoID, NombreProducto, PrecioProveedor, PrecioVenta
17 -- FROM Inventario;
18 -- GO
19
20 -- 3B. Volver al usuario administrador
21 REVERT;
22 GO

```



76 %



9



0



Resultados

Mensajes

(0 filas afectadas)

Mens. 230, Nivel 14, Estado 1, Línea 16

The SELECT permission was denied on the column 'PrecioProveedor' of the object 'Inventario', database 'QhatuPeru', schema 'dbo'.

Mens. 230, Nivel 14, Estado 1, Línea 16

The SELECT permission was denied on the column 'PrecioVenta' of the object 'Inventario', database 'QhatuPeru', schema 'dbo'.

Hora de finalización: 2025-11-19T08:25:56.1415405-05:00

5

Crear la Master Key del Servidor

Este paso crea la clave maestra que protege el certificado. Se debe ejecutar en la base de datos master.


```
SQLQuery18.s...P\elvis (65))  SQLQuery17.sq...HP\elvis (75))  SQLQuery16.sq...HP\elvis (69))  SQLQuery15.sq...HP\elvis (63))
1  USE master;
2  GO
3
4  -- 1A. Crear la Clave Maestra de la base de datos Master
5  -- **IMPORTANTE:** Reemplaza 'ContraseñaSeguraParaMasterKey' con una contraseña fuerte y guárdala.
6  CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'ContraseñaSeguraParaMasterKey';
7  GO
```

76 % 9 0

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T08:31:04.6233173-05:00

Crear el Certificado de Servidor

Este certificado, creado a partir de la Master Key, protegerá la clave de cifrado de la base de datos. Se ejecuta en master.

```
1  USE master;
2  GO
3
4  -- 2A. Crear el Certificado del Servidor
5  CREATE CERTIFICATE TdeCertificado
6  WITH SUBJECT = 'Certificado para TDE en QhatuPeru';
7  GO
```

76 %

9

0

↑

↓

◀

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T08:32:32.2133266-05:00

Crear la Clave de Cifrado de la Base de Datos (DEK)

Este es el primer paso específico en la base de datos QhatuPeru. La DEK es la clave simétrica real que cifrará los datos.

QLQuery18.s...P\elvis (63))* SQLQuery17.sq...HP\elvis (73))* SQLQuery16.sq...HP\elvis (6

1
2
3
4
5
6
7
8

USE QhatuPeru;
GO

-- 3A. Crear la Clave de Cifrado de la Base de Datos (DEK)
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCPTION BY SERVER CERTIFICATE TdeCertificado;
GO

5 %
No se encontraron problemas.

Mensajes

Mens. 33103, Nivel 16, Estado 1, Línea 5
A database encryption key already exists for this database.
Hora de finalización: 2025-11-19T08:33:31.7408295-05:00

Finalmente, se activa el cifrado en la base de datos QhatuPeru. El proceso de cifrado de los archivos físicos (MDF/LDF) comenzará automáticamente.

```
SQLQuery 16.s...P\elvis (65))  SQLQuery 17.sq...HP\elvis (75))  SQLQuery 16.sq...HP\elvis (69))  SQLQ...
1  USE QhatuPeru;
2  GO
3
4  -- 4A. Habilitar el cifrado TDE en la base de datos
5  ALTER DATABASE QhatuPeru
6  SET ENCRYPTION ON;
7  GO
```

76 % No se encontraron problemas.

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T08:34:13.4678727-05:00

Para demostrar que TDE se ha implementado correctamente, utiliza esta consulta para verificar el estado de cifrado:

```

1  -- 5A. Verificar el estado del cifrado (Encryption State)
2  -- Estado 2 = Encrypted (Completado)
3  -- Estado 3 = Encryption in progress (En progreso)
4  SELECT DB_NAME(database_id) AS DatabaseName,
5         encryption_state_desc
6  FROM sys.dm_database_encryption_keys
7  WHERE DB_NAME(database_id) = 'QhatuPeru';
8  GO

```

76 %

1

0

↑ ↓

Resultados Mensajes

	DatabaseName	encryption_state_desc
1	QhatuPeru	ENCRYPTED

6

Creación de las Claves de Cifrado (CMK y CEK)

La Columna Master Key (CMK) protege la Columna Encryption Key (CEK), y ambas se necesitan antes de cifrar una columna.

```

SQLQuery1.sql...P\elvis (63))
1  USE QhatuPeru;
2  GO
3
4  -- A. Crear la Column Master Key (CMK)
5  -- **IMPORTANTE:** Este paso asume que un certificado ha sido creado en el almacén de Windows.
6  -- Reemplace 'Certificado_AE_QhatuPeru' por el nombre de su certificado.
7  IF NOT EXISTS (SELECT name FROM sys.column_master_keys WHERE name = 'CMK_PRECIO_PROVEEDOR')
8  BEGIN
9      CREATE COLUMN MASTER KEY [CMK_PRECIO_PROVEEDOR]
10     WITH (
11         KEY_STORE_PROVIDER_NAME = 'MSSQL_CERTIFICATE_STORE',
12         KEY_PATH = 'Certificado_AE_QhatuPeru'
13     );
14 END
15 GO
16
17 -- B. Crear la Column Encryption Key (CEK) - SINTAXIS CORREGIDA
18 -- Esta clave es la que cifra los datos y es protegida por la CMK.
19 IF NOT EXISTS (SELECT name FROM sys.column_encryption_keys WHERE name = 'CEK_PrecioProveedor')
20 BEGIN
21     CREATE COLUMN ENCRYPTION KEY [CEK_PrecioProveedor]
22     WITH VALUES
23     (
24         COLUMN_MASTER_KEY = [CMK_PRECIO_PROVEEDOR],
25         ALGORITHM = 'RSA_OAEP'
26     );
27 END
28 GO

```

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2025-11-19T16:30:12.0492080-05:00

dbo.CLIENTES	dbo.GUÍA_DETALLE	dbo.GUÍA_ENVIO	dbo.LINEA	dbo.ORDEN_COMPRA	dbo.ORDEN_DETALLE	dbo.Pedidos	dbo.PROVEEDOR	dbo.Reportes	dbo.TIENDA	dbo.TRANSPORTISTA	dbo.Ventas	Tablas de libro de contabilidad descart	Vistas	Recursos externos	Sinónimos	Programación	Almacén de consultas	Service Broker	Almacenamiento
--------------	------------------	----------------	-----------	------------------	-------------------	-------------	---------------	--------------	------------	-------------------	------------	---	--------	-------------------	-----------	--------------	----------------------	----------------	----------------

Stock	FechaCreacion	PrecioProveedor_ENC	nombre_cmk	proveedor	ruta_clave
1	2025-11-19 16:30:12.0492080-05:00	1500.00	CMK_Qhatu	MSSQL_CERTIFICATE_STORE	CurrentUser/My/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...

ProductoID	Nombre	PrecioProveedor	PrecioProveedor_ENC	PrecioVenta	Stock
1	Laptop Gamer	1500.00	NULL	2000.00	10
2	Mouse Inalámbrico	25.00	NULL	45.00	50
3	Teclado Mecánico	80.00	NULL	120.00	30

7 El objetivo es **configurar una auditoría de seguridad** que consta de dos partes:

1. **Un Server Audit (Auditoría de Servidor):** Debe registrar tanto los intentos de inicio de sesión (**login**) que son **fallidos** como los que son **exitosos**.
2. **Un Database Audit Specification (Especificación de Auditoría de Base de Datos):** Esta debe configurarse específicamente para la base de datos **QhatuPeru** y tiene que registrar los cambios en la estructura de la base de datos (lenguaje DDL) como son las operaciones de **CREATE**, **ALTER**, y **DROP** (crear, modificar o eliminar objetos) aplicados a los objetos considerados **críticos**.

```
41 BEGIN
42 CREATE DATABASE AUDIT SPECIFICATION EspecificacionBDQhatu
43 FOR SERVER AUDIT AuditoriaQhatu
44 ADD (SELECT, INSERT, UPDATE, DELETE ON DATABASE::QhatuPeru BY PUBLIC),
45 ADD (EXECUTE ON DATABASE::QhatuPeru BY PUBLIC)
46 WITH (STATE = ON);
47 PRINT 'Especificación de BD creada';
48 END
49 ELSE
50 PRINT 'Especificación de BD ya existe';
51
52 PRINT '☑ Auditoría configurada exitosamente';
53 GO
```

Mensajes

Auditoría creada en Application Log
Auditoría habilitada
Especificación de servidor creada
Especificación de BD creada
? Auditoría configurada exitosamente

Hora de finalización: 2025-11-19T12:28:51.8573098-05:00

100 % No se encontraron problemas. Línea: 8 Carácter: 1 TABULACIONES MIXTO

8 Objetivo: Configurar una sesión de **Extended Events (Eventos Extendidos)**.

- **Funcionalidad:** Esta sesión debe **capturar** eventos de **deadlocks** (bloqueos mutuos) y eventos de **login failed** (intentos de inicio de sesión fallidos).
- **Resultado esperado:** Los eventos deben **guardarse en un archivo**, y se debe crear una **vista** que permita consultar y analizar esos datos de **XEvent** directamente desde la base de datos.

```
SQLQuery7.sql...P\elvis (79))* SQLQuery6.sql ...HP\elvis (78))* SQLQuery5.sql ...HP\elvis (85))* SQLQuery4.sql ...HP\elvis (69))*
1 USE QhatuPeru;
2 GO
3
4 -- 1. Crear una función para simplificar la lectura de archivos .xel
5 CREATE FUNCTION dbo.fn_ReadXEvents (@FilePath NVARCHAR(4000))
6 RETURNS TABLE
7 AS
8 RETURN
9 (
10     -- sys.fn_xe_file_target_read_file lee los archivos de eventos y devuelve el contenido XML
11     SELECT
12         CAST(event_data AS XML) AS EventXML
13     FROM sys.fn_xe_file_target_read_file(@FilePath, NULL, NULL, NULL)
14 );
15 GO
16
17 -- 2. Crear la vista que consulta los eventos y extrae datos clave
18 CREATE VIEW Monitoreo_XE_QhatuPeru AS
19 SELECT
20     -- Extraer el nombre del evento
21     T.EventXML.value('/event/@name[1]', 'NVARCHAR(100)') AS EventName,
22     -- Extraer la marca de tiempo del evento
23     T.EventXML.value('/event/@timestamp[1]', 'datetimeoffset') AS EventTime,
24     -- Intentar extraer el nombre de la BD (disponible en deadlocks)
25     T.EventXML.value('/event/action[@name="database_name"]/value[1]', 'NVARCHAR(128)') AS DatabaseName
```

76 % No se encontraron problemas.

Mensajes

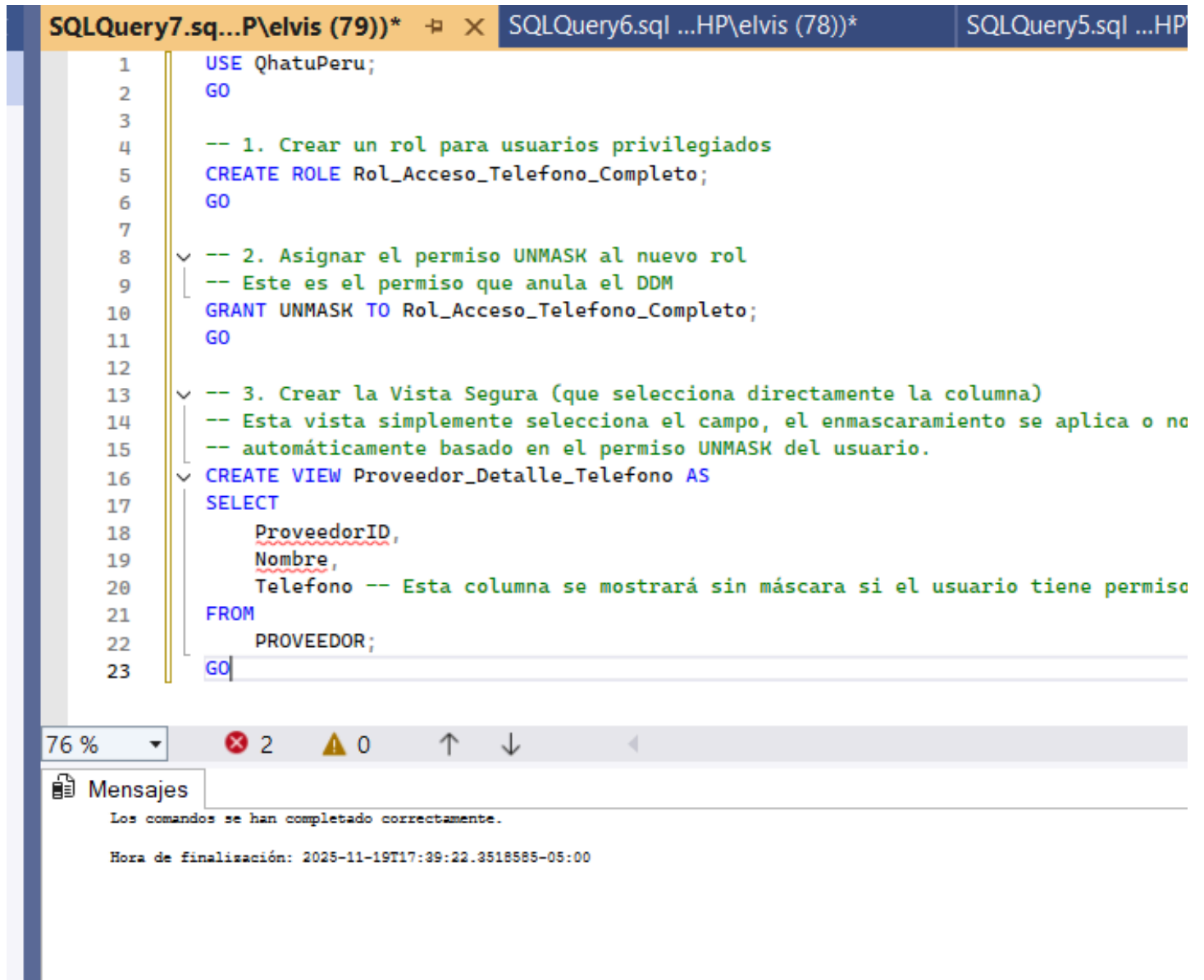
Vista Monitoreo_XE_QhatuPeru creada. Consulta SELECT * FROM Monitoreo_XE_QhatuPeru;

Hora de finalización: 2025-11-19T17:37:49.2788928-05:00

9 Objetivo: Implementar técnicas de ocultación de datos y control de acceso.

Funcionalidad 1 (Dynamic Data Masking): Aplicar **Dynamic Data Masking (Enmascaramiento Dinámico de Datos)** a columnas que contengan información sensible (el ejemplo dado es el número de **Teléfono en la tabla PROVEEDOR**), para que el dato real no sea visible para la mayoría de los usuarios.

Funcionalidad 2 (Acceso Controlado): Crear una **vista segura** que permita a un conjunto limitado de usuarios (validado mediante una **función que valide su rol**) ver los **datos completos y sin enmascarar**.



```
SQLQuery7.sql...P\elvis (79))* SQLQuery6.sql ...HP\elvis (78))* SQLQuery5.sql ...HP
1  USE QhatsuPeru;
2  GO
3
4  -- 1. Crear un rol para usuarios privilegiados
5  CREATE ROLE Rol_Acceso_Telefono_Completo;
6  GO
7
8  -- 2. Asignar el permiso UNMASK al nuevo rol
9  -- Este es el permiso que anula el DDM
10 GRANT UNMASK TO Rol_Acceso_Telefono_Completo;
11 GO
12
13 -- 3. Crear la Vista Segura (que selecciona directamente la columna)
14 -- Esta vista simplemente selecciona el campo, el enmascaramiento se aplica o no
15 -- automáticamente basado en el permiso UNMASK del usuario.
16 CREATE VIEW Proveedor_Detalle_Telefono AS
17 SELECT
18     ProveedorID,
19     Nombre,
20     Telefono -- Esta columna se mostrará sin máscara si el usuario tiene permiso
21 FROM
22     PROVEEDOR;
23 GO
```

76 % 2 0

Mensajes

Los comandos se han completado correctamente.

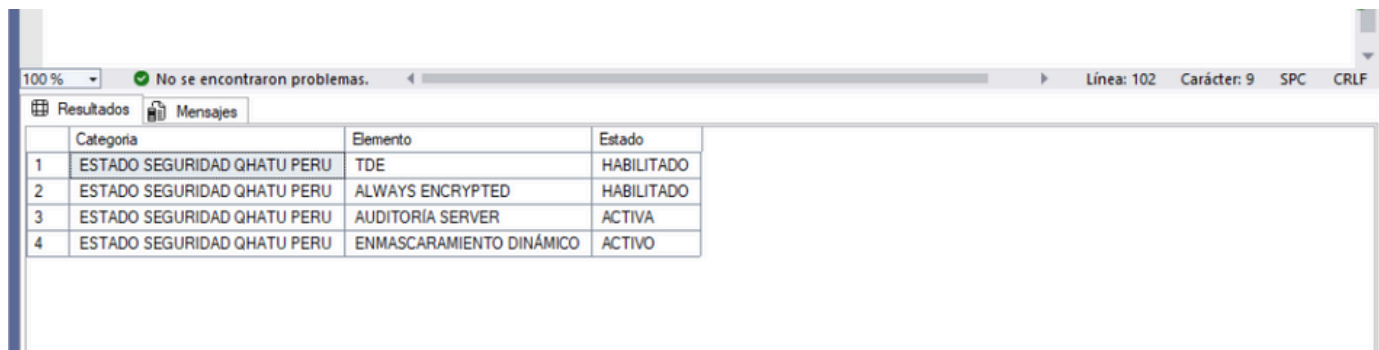
Hora de finalización: 2025-11-19T17:39:22.3518585-05:00

10 Objetivo: Este es un proyecto integrador que combina múltiples características de seguridad avanzadas.

Pasos:

1. Crear un **rol auditor_seguridad** (para segregación de funciones).
2. **Habilitar TDE (Transparent Data Encryption)** en la base de datos (si no está habilitado).

3. **Preparar Always Encrypted** para una **columna sensible** (cifrado persistente de la columna).
4. **Configurar auditoría** para una tabla específica (auditar el acceso a esa tabla).
5. Dejar un **procedimiento almacenado** que registre cambios críticos (esto sugiere una **traza** o registro de auditoría personalizado).



The screenshot shows a SQL Server Enterprise Manager console window. The status bar at the top indicates '100 %' zoom, 'No se encontraron problemas.' (No problems found), and 'Línea: 102 Carácter: 9 SPC CRLF'. Below the status bar, there are two tabs: 'Resultados' (Results) and 'Mensajes' (Messages). The 'Resultados' tab is active, displaying a table with four columns: 'Id', 'Categoria', 'Elemento', and 'Estado'. The table contains four rows of data, all with the same category 'ESTADO SEGURIDAD QHATU PERU'.

	Categoria	Elemento	Estado
1	ESTADO SEGURIDAD QHATU PERU	TDE	HABILITADO
2	ESTADO SEGURIDAD QHATU PERU	ALWAYS ENCRYPTED	HABILITADO
3	ESTADO SEGURIDAD QHATU PERU	AUDITORIA SERVER	ACTIVA
4	ESTADO SEGURIDAD QHATU PERU	ENMASCARAMIENTO DINÁMICO	ACTIVO